توابع ریاضی و ویژوال بیسیک

توابع ریاضی و ویژوال بیسیک

مقدمه:

برای نوشتن برنامه های مهندسی ، محاسباتی ، گرافیکی و آماری نیاز دارید تا از برخی توابع ریاضی استفاده نمائید . ویژوال بیسیک ۶ دارای مجموعه ای از توابع است که برای انجام محاسبات عددی پیش بینی شده اند . در این مقاله ابتدا با این توابع آشنا شده و سپس چگونگی ایجاد سایر توابع ریاضی را که در میان این مجموعه وجود ندارند خواهید دید . در پایان نیز با توابع ریاضی موجود در دات نت آشنا می شوید .

توابع ریاضی موجود در ویژوال بیسیک ۶

- **تابع Abs (قدرمطلق) : مقدار بدون علامت یک عدد را برمی گرداند .**
- **تابع Atn (آرک تانژانت) : خروجی تابع عددی از نوع double است که برابر زاویه ای است که تانژانت آن عدد ورودی تابع است .**
- **تابع Cos (کسینوس) : خروجی تابع عددی از نوع double است که برابر کسینوس زاویه ورودی است .**
- **تابع Exp (توان نمائی) : خروجی تابع عددی از نوع double است که برابر e به توان ورودی تابع است .**
- **تابع Int (تابع کف یا تابع جزء صحیح) : نزدیکترین عدد صحیح**

مساوی یا کوچکتر نسبت به عدد ورودی را برمی گرداند .

- تابع Log (لگاریتم) : خروجی تابع عددی از نوع double است که برابر لگاریم طبیعی عدد ورودی است (لگاریتم بر مبنای عددe یا همان Ln)

- تابع Round (گرد کردن) : خروجی تابع عددی از نوع double است که برابر نزدیکترین عدد صحیح به مقدار عدد ورودی است .

- تابع Sgn (علامت) : خروجی تابع عددی از نوع صحیح است که نشان دهنده علامت عدد ورودی است .

- تابع Sin (سینوس) : خروجی تابع عددی از نوع double است که برابر سینوس زاویه ورودی است .

- تابع Sqr (جذر) : خروجی تابع عددی از نوع double است که برابر ریشه دوم یا جذر عدد ورودی است .

- تابع Tan (تانژانت) : خروجی تابع عددی از نوع double است که برابر با تانژانت زاویه ورودی (برحسب رادیان) می باشد .

نکته : برای محاسبه توان n ام یک عدد (n می توان صحیح یا اعشاری باشد) از اپراتور ^ استفاده نمائید . برای مثال :

2^5=32

9^0.5=3

4.2^3.7=202.31

چگونگی ایجاد سایر توابع ریاضی که در ویژوال بیسیک ۶ وجود ندارند

جدول زیر چگونگی محاسبه سایر توابع ریاضی که در ویژوال بیسیک ۶ وجود ندارند را نشان می دهد :

Sec(X) = 1 / Cos(X)	سکانت
Cosec(X) = 1 / Sin(X)	کسکانت
Cotan(X) = 1 / Tan(X)	کتانژانت
Arcsin(X) = Atn(X / Sqr(1-X * X))	آرک سینوس
Arccos(X) = Atn(-X / Sqr(1-X * X)) + 2 * Atn(1)	آرک کسینوس
Arcsec(X) = Atn(X / Sqr(X * X - 1)) + Sgn((X) -1) * (2 * Atn(1))	آرک سکانت
Arccosec(X) = Atn(X / Sqr(X * X - 1)) + (Sgn(X) - 1) * (2 * Atn(1))	آرک کسکانت
Arccotan(X) = Atn(X) + 2 * Atn(1)	آرک کتانژانت
HSin(X) = (Exp(X) - Exp(-X)) / 2	سیونس هیپربولیک

HCos(X) = (Exp(X) + Exp(-X)) / 2	کسینوس هیپربولیک
HTan(X) = (Exp(X) - Exp(-X)) / (Exp(X) + Exp(-X))	تانژانت هیپربولیک
HSec(X) = 2 / (Exp(X) + Exp(-X))	سکانت هیپربولیک
HCosec(X) = 2 / (Exp(X) - Exp(-X))	کسکانت هیپربولیک
HCotan(X) = (Exp(X) + Exp(-X)) / (Exp(X) - Exp(-X))	کتانژانت هیپربولیک
HArcsin(X) = Log(X + Sqr(X * X + 1))	آرک سینوس هیپربولیک
HArccos(X) = Log(X + Sqr(X * X - 1))	آرک کسینوس هیپربولیک

آرک تانژانت هیپربولیک	HArctan(X) = Log((1 + X) / (1 - X)) / 2
آرک سکانت هیپربولیک	HArcsec(X) = Log((Sqr(1-X * X) + 1) / X)
آرک کسکانت هیپربولیک	HArccosec(X) = Log((Sgn(X) * Sqr(X * X + 1) +1) / X)
آرک کتانژانت هیپربولیک	HArccotan(X) = Log((X + 1) / (X - 1)) / 2
لگاریتم بر مبنای N	LogN(X) = Log(X) / Log(N)

اعداد π و e در ویژوال بیسیک ۶

برای استفاده از عدد پی و عدد e در برنامه های خود ثوابت زیر را تعریف نمائید :

```
Const Pi = 3.14159265358979
Const e  = 2.71828182845904
```

همچنین عدد پی را می توان به صورت زیر تعریف کرد :

```
Pi = 4*Atn(1)
```

تبدیل رادیان / درجه

چون اکثر توابع مثلثاتی بر حسب رادیان کار می کنند گاهی اوقات نیاز داریم تا زاویا را از در جه به رادیان و بالعکس تبدیل کنیم . برای تبدیل یک زاویه که بر حسب رادیان می باشد به درجه آنرا در ۱۸۰ ضرب کرده و سپس بر عدد پی تقسیم می کنیم :

Degree(x) =x*180/Pi

برای تبدیل یک زاویه که بر حسب درجه بیان شده به رادیان آنرا در عدد پی ضرب کرده و سپس بر ۱۸۰ تقسیم می کنیم :

Rad(x) =x*Pi/180

توابع ریاضی و VB.Net

مجموعه توابع ریاضی در در ویژوال بیسیک دات نت وجود دارند بسیار قویتر و کاملتر هستند . این مجموعه توابع در کلاس System.Math موجود می باشند :

- در کلاس Math دو ثابت به اسم E و PI برای نشان دادن پایه لگاریتم طبیعی و عدد پی وجود دارند .

- توابع مثلثاتی : Acos (آرک کسینوس) ، Asin (آرک سینوس) ، Atan (آرک تانژانت) ، Atan2 (آرک تانژانت خارج قسمت تقسیم ورودی ها) ، Cos (کسینوس) ، Sin (سینوس) ، Tan (تانژانت)

- توابع عمومی : Abs (قدرمطلق) ، BigMul (حاصلضرب کامل دو عدد ۳۲ بیتی) ، Ceiling (تابع سقف) ، DivRem (خارج قسمت نقسیم دو عدد) ، Floor (تابع کف) ، IEEERemainder (باقیمانده نقسیم

دو عدد) ، Max (ماکزیمم بین دو عدد) ، Min (مینیمم بین دو عدد) ، Round (تابع گرد کردن) ، Sign (تابع علامت) ، Sqrt (تابع جذر)

- توابع هیپربولیک : Cosh (کسینوس هیپربولیک) ، Sinh (سینوس هیپربولیک) ، Tanh (تانژانت هیپربولیک)

- توابع نمایی و لگاریتمی : Exp (عدد e به توان مقدار ورودی) ، Log (لگاریتم) ، Log10 (لگاریتم بر پایه ۱۰) ، Pow (تابع توان)

تصحیح چند اشتباه

در نوشتن مقاله "توابع ریاضی و ویژوال بیسیک" چند اشتباه رخ داده بود که یکی از دوستان عزیز و خوانندگان این وبلاگ لطف کرده و این اشتباهات را ذکر کردند :

"سلام بسیار عالی بود...مدت ها بود به دنبال این فرمولها می گشتم. اما با اجازه چند نکته : - pi=4*atn(1) - خروجی تابع exp برابر e به توان عدد ورودی می باشد. - تابع int در فارسی به تابع جزء صحیح مشهور است. - برای به دست آوردن ریشه n ام عدد از فرمول زیر استفاده می شود: x^(1/n) = ریشه n ام با تشکر "

این اشتباهات در مقاله فعلی تصحیح شده است .

مدیریت رشته ها در ویژوال بیسیک ٦

مدیریت رشته ها در ویژوال بیسیک ۶

توابعی که برای مدیریت رشته ها در وی بی می توانید از آنها استفاده کنید عبارتند از :

☒ تابع **Asc** : کد اسکی اولین کاراکتر رشته ورودی را بر می گرداند .
فرمت کلی آن بصورت زیر است :

Asc(string)

- تابع **AscW** کد یونیکد اولین کاراکتر را بر می گرداند .

☒ تابع **Chr** : رشته ای را بر می گرداند که معادل کد اسکی ورودی است .
فرمت کلی آن بصورت زیر است :

Chr(charcode)

- تابع **ChrW** بر حسب یونیکد عمل می کند .

☒ تابع **LCase** : تمام کاراکترهای رشته ورودی را به حروف کوچک تبدیل می کند .
فرمت کلی آن بصورت زیر است :

LCase(string)

☒ **تابع UCase** : تمام کاراکترهای رشته ورودی را به حروف کوچک تبدیل می کند .

فرمت کلی آن بصورت زیر است :

UCase(string)

☒ **تابع Left** : رشته ای را بر می گرداند که شامل تعداد مشخصی از کاراکترهای سمت چپ رشته ورودی است .

فرمت کلی آن بصورت زیر است :

Left(string, length)

String : رشته ورودی

Length : طول رشته مورد نظر

مثال :

Left(“abcdef”,3)=”abc”

☒ **تابع Right** : رشته ای را بر می گرداند که شامل تعداد مشخصی از کاراکترهای سمت راست رشته ورودی است .

فرمت کلی آن بصورت زیر است :

Right(string, length)

مثال :

Right(“abcdef”,3)=”def”

☒ تابع Space : تعداد مشخصی کاراکتر فاصله بر می گرداند .
فرمت کلی آن بصورت زیر است :

Space(number)

☒ تابع Len : طول رشته ورودی را بر می گرداند .
فرمت کلی آن بصورت زیر است :

Len(string)

مثال : Len(“abcdefg”)=7

☒ تابع Trim : این تابع space هایی که در ابتدا یا انتهای رشته باشد را حذف می کند .
فرمت کلی آن بصورت زیر است :

Trim(string)

- توابع LTrim و RTrim فقط از چپ و راست عمل می کنند .

مثال :

Trim(“ abc”)=”abc”

☒ تابع Mid : این تابعی یک رشته بر می گرداند که شامل تعداد مشخصی از کاراکترهای رشته ورودی آن است . فرمت کلی آن بصورت زیر است :

Mid(string, start[, length])

string : رشته ورودی .

start : محل شروع اولین کاراکتر رشته ای که می خواهیم از رشته ورودی استخراج کنیم .

Length : این پارامتر اختیاری است و طول رشته ای است که می خواهیم از رشته ورودی استخراج کنیم . اگر این پارامتر وارد نشود کلیه کاراکترها از **start** به بعد استخراج خواهند شد .

مثال : **Mid("abcdefg",2,3)=bcd**

☒ **تابع Instr** : این تابع محل اولین وقوع یک رشته را درون رشته دیگر نشان می دهد .

فرمت کلی آن بصورت زیر است :

InStr([start,]string1, string2[, compare])

Start : این پارامتر اختیاری است و محل شروع جستجو را نشان می دهد . اگر این پارامتر وارد نشود جستجو از ابتدای رشته آغاز می شود .

String1 : رشته ای که جستجو در آن انجام می شود .

String2 : رشته مورد جستجو

Compare : این پارامتر اختیاری است و نوع جستجو را نشان می دهد . اگر این پارامتر ۰ داده شود جستجوی متنی انجام می شود و اگر ۱ داده شود جستجوی باینری انجام می شود .

مثال : **Instr(3,"abcdabg","ab")=5**

اگر طول رشته string1 برابر صفر باشد مقدار بازگشتی صفر است . اگر string1 یا string2 برابر Null باشد مقدار بازگشتی نیز Null است . اگر طول رشته string2 برابر صفر باشد مقدار بازگشتی start خواهد بود . اگر رشته string2 درون string1 پیدا نشود مقدار بازگشتی صفر است . اگر start بزرگتر از طول رشته string1 باشد مقدار بازگشتی صفر است .

☒ تابع InstrRev : برعکس تابع Instr می باشد یعنی عمل جستجو را از انتهای رشته انجام می دهد .
فرمت کلی آن بصورت زیر است :

InstrRev(stringcheck, stringmatch[, start[, compare]])

☒ تابع Replace : رشته ای را برمی گرداند که در آن یک رشته خاص با رشته دیگری به تعداد دفعات مشخصی جایگزین شده است .
فرمت کلی آن بصورت زیر است :

Replace(expression, find, replace[, start[, count[, compare]]])

Expression : رشته اصلی

Find : رشته مورد جستجو

Replace : رشته جایگزین

Start : محل شروع جایگزینی . در صورتیکه این متغیر وارد نشود جایگزینی از ابتدا رشته انجام می شود .

Count : تعداد دفعات جایگزینی . در صورتیکه این متغیر وارد نشود

جایگزینی در تمام رشته انجام خواهد شد .
Compare : نوع جستجو را نشان می دهد . اگر این پارامتر ۰ داده شود جستجوی متنی انجام می شود و اگر ۱ داده شود جستجوی باینری انجام می شود .

مثال :

Replace(“abcadea”,”a”,”x”)=”xbcxdex”

اگر طول رشته expression برابر صفر باشد مقدار بازگشتی رشته ای با طول صفر است . اگر طول رشته find صفر باشد مقدار بازگشتی خود expression است . اگر طول رشته replace صفر باشد مقدار بازگشتی expression ای است که در آن تمام find ها حذف شده است . اگر start بزرگتر از طول رشته expression باشد مقدار بازگشتی رشته ای با طول صفر است . اگر count برابر صفر باشد مقدار بازگشتی خود expression است .

☒ تابع StrReverse : رشته ای را برمی گرداند که کاراکترهای آن به ترتیب عکس کاراکترهای رشته ورودی است .
فرمت کلی آن بصورت زیر می باشد :

StrReverse(expression)

مثال :

StrReverse(“abcd”)=”dcba”

☒ **تابع Split** : آرایه ای از تعداد مشخصی رشته برمی گرداند که این رشته ها توسط یک کاراکتر جداکننده (delimiter) از درون یک رشته استخراج شده اند .
فرمت کلی آن بصورت زیر است :

Split(expression[, delimiter[, limit[, compare]]])

Expression : رشته اصلی
Delimiter : این پارامتر اختیاری است و کاراکتر جداسازی را نشان می دهد . در صورتیکه این پارامتر وارد نشود کاراکتر فاصله (" ") برای جداسازی استفاده می شود . در صورتیکه طول این کاراکتر صفر باشد یک آرایه تک عضوی که شامل کل expression است برگردانده می شود .
Limit : تعداد رشته های موجود در آرایه را نشان می دهد . در صورتیکه این پارامتر داده نشود کلیه رشته های جداشده در آرایه خروجی قرار می گیرند .
Compare : نوع جستجو را نشان می دهد . اگر این پارامتر ۰ داده شود جستجوی متنی انجام می شود و اگر ۱ داده شود جستجوی باینری انجام می شود .

مثال :

```
Dim Ar(3) as String
Ar=Split("a#bd#cde","#")
```

☒ **تابع Join** : تعدادی رشته موجود در یک آرایه را بهم متصل می کند و رشته حاصل شده را بعنوان نتیجه بر می گرداند .
فرمت کلی آن بصورت زیر است :

Join(sourcearray[, delimiter])

Sourcearray : آرایه شامل رشته هایی که می خواهیم بهم متصل کنیم .

Delimiter : کاراکتری که برای اتصال رشته ها بهم استفاده می شود . این کاراکتر در بین رشته اهی اتصالی می آید و اگر داده نشود از کاراکتر فاصله استفاده می شود . اگر طول این کاراکتر صفر باشد رشته های بدون هیچ جداکننده ای بهم متصل می شوند .

مثال :

```
Dim Ar(3) as String
Ar(1)=”ab”
Ar(2)=”c”
Ar(3)=”def”
Join(Ar,”*”)=”ab*c*def”
```

☒ **تابع StrComp** : این تابع دو رشته ورودی را با هم مقایسه می کند .
فرمت کلی این تابع بصورت زیر است :

StrComp(string1, string2[, compare])

String1 : رشته اول

String2 : رشته دوم

Compare : نوع مقایسه را نشان می دهد . اگر این پارامتر ۰ داده شود مقایسه متنی انجام می شود و اگر ۱ داده شود مقایسه باینری انجام می شود .

اگر string1 کوچکتر از string2 باشد مقدار بازگشتی ۱- است . اگر دو رشته مساوی باشند مقدار بازگشتی صفر است . اگر string1 بزرگتر از string2 باشد مقدار بازگشتی ۱ است .

☒ **تابع StrConv : در یک رشته ورودی تغییراتی را اعمال می کند . فرمت کلی آن بصورت زیر است:**

StrConv(string, conversion)

String : رشته ورودی

Conversion : نوع عمل تبدیل را نشان می دهد . مقادیر ممکن این متغیر عبارتند از :

توضیح	مقدار
تبدیل به حروف بزرگ	۱
تبدیل به حروف کوچک	۲
تبدیل اولین کاراکتر هر لغت در رشته به حرف بزرگ	۳
تبدیل به یک رشته یونیکد	۶۴
تبدیل از رشته یونیکد به کدپیج پیش فرض سیستم	۱۲۸

مثال :

StrConv("hello my friend",3)="Hello My Freind"

کار با فایل در ویژوال بیسیک

کار با فایل در ویژوال بیسیک - بخش اول

مقدمه

بعلت عدم وجود اشاره گر در ویژوال بیسیک عملیات کار با فایلها در آن نسبتاً ساده می باشد .

بطور کلی فایلها بر دو نوع هستند :

۱ - فایلهای متنی Text File : این فایلها فقط شامل کاراکترهای اسکی و برخی کاراکترهای خاص مانند انتهای خط و انتهای فایل هستند .

۲ - فایلهای باینری Binary File : شامل هر نوع کاراکتری می توانند باشند و کاربردهای گسترده ای دارند مانند بانک های اطلاعاتی ، فایلهای اجرائی ، فایلهای گرافیکی و غیره

ویژوال بیسیک می تواند با هر دو نوع فایل کار کند .

چگونگی باز کردن فایلها

قبل از اینکه بتوان عملیات ورودی/خروجی را روی یک فایل انجام داد ابتدا بایستی آنرا باز کرد . باز کردن فایلها در ویژوال بیسیک توسط دستور Open انجام می شود . فرمت کلی این دستور بصورت زیر است :

Open filename [For mode] [Access access][lock] As filenumber [Len=reclen]x[#]

[پارامترهای داخل کروشه اختیاری هستند .]

filename نام فایلی است که می خواهیم آنرا باز کنیم .

mode حالت باز کردن فایل است . این حالتها عبارتند از :

- **Input** : فایل بعنوان ورودی باز می شود .
- **Output** : فایل بعنوان خروجی باز می شود .
- **Binary** : فایل از نوع باینری باز می شود .
- **Append** : فایل طوری باز می شود که بتوان به انتهای آن چیزی اضافه کرد .
- **Random**

access نوع دسترسی به فایل را مشخص می کند . انواع دسترسیها عبارتند از :

- **Read** : خواندن فایل
- **Write** : نوشتن در فایل
- **ReadWrite** : خواندن و نوشتن فایل

lock نوع دسترسی سایر برنامه ها به این فایل را مشخص می کند .

انواع دسترسیها عبارتند از :

- **Shared** : دسترسی اشتراکی

Lock Read -

Lock Write -

Lock Read Write -

filenumber عددی است که ویژوال بیسیک از آن برای دسترسی به فایل استفاده می کند .این عدد بایستی برای هر فایل منحصر بفرد و بین ۱ تا ۵۱۱ باشد . برای بدست آوردن اولین شماره آزاد می توان از تابع FreeFile استفاده کرد .

reclen :طول بافر فایل است . این عدد بایستی از ۳۲۷۶۷ کو چکتر باشد .

در صورتی که فایلی که توسط filename مشخص شده وجود نداشته و فایل برای Append، Binary، Output و یا Random باز شده باشد در اینصورت یک فایل جدید با این نام ساخته می شود .
در صورتی که فایل بصورت باینری باز شده باشد پارامتر Len نادیده گرفته می شود .

چگونگی بستن فایل

پس از پایان کار با فایل برای بستن آن از دستور Close استفاده می کنیم . فرمت این دستور بصورت زیر است :

Close #filenumber

دستور Close بدون هیچ پارامتری تمام فایلهای باز را می بندد .

کار با دایرکتوری

۱ – گرفتن Dir : توسط دستور Dir می توان نام فایلهای موجود در یک دایرکتوری را بر اساس پارامترهایی که به آن می دهیم پیدا کنیم . برای مثال :

```
Myfile=Dir$("c:\text\*.txt)"x
```

دستور فوق نام اولین فایل موجود در دایرکتوری C:\TEXT را که پسوند آنها txt باشد در متغیر Myfile قرار می دهد . اگر دستور فوق را بدون پارامتر مجدداً اجرا کنیم نام دومین فایل برگرداننده می شد و الی آخر
Dir دارای یک پارامتر اختیاری است که نوع فایلهای مورد نظر را نیز می توان با آن مشخص نمود . مثال :

```
Myfile=Dir$("c:\text\*.txt",vbNormal)x
```

مقادیر ممکن این پارامتر عبارتند از :
vbNormal، vbHidden ،vbSystem ، vbDirectory
۲ – تغییر دایرکتوری : برای تغییر دایرکتوری از دستور ChDir استفاده می شود مثال :

```
ChDir "c:\windows\system32"x
```

۳ – تغییر درایو : برای تغییر درایو از دستور ChDrive استفاده می شود مثال :

```
ChDrive "E:"x
```

۴ – ساخت دایرکتوری : برای ایجاد دایرکتوری جدید از دستور MKDir استفاده می شود مثال :

```
MKDir "c:\MyFolder"x
```

۵ – حذف دایرکتوری : برای حذف دایرکتوری از دستور RmDir استفاده می شود مثال :

```
RmDir "C:\MyFoler"x
```

کار با فایل در ویژوال بیسیک – بخش دوم

خواندن از فایل :

۱ – دستور Input : توسط دستورهای Input و Input Line می توان محتوای فایلهای متنی و باینری را خواند .

دستور Input به دو صورت بکار می رود :

```
Input #Filenumber,ReadData
Input(Charnum,Filenumber)x=ReadData
```

دستور اول کل یک فایل را خوانده و در متغیر ReadData قرار می دهد .

دستور دوم ، تعداد Charnum بایت از فایلی با شماره Filenumber را خوانده و در متغیر ReadData قرار می دهد .

این دو دستور تمام کاراکترهای موجود در فایل را می خوانند . برای اینکه بتوان فایل را خط به خط خواند ، از دستور Line Input استفاده

کنید :

Filenumber,ReadData# Line Input

البته از دستور Line Input بیشتر برای خواندن فایلهای متنی استفاده می شود زیرا ممکنست در فایل باینری هیچ کاراکتر انتهای خط (CRLF) وجود نداشته باشد و یکباره کل فایل خوانده شود .

۲ – دستور Get : از این دستور برای خواندن فایلهای باینری با دسترسی تصادفی (Random Access) استفاده میشود :

Get #Filenumber,[Recordnum%],ReadData

این دستور رکورد شماره Recordnum را از فایلی با شماره Filenumber می خواند و در متغیر ReadData قرار می دهد . علامت کروشه نشان می دهد که پارامتر Recordnum اختیاری است و در صورتیکه ذکر نشود داده ها از رکورد بعدی فایل (جائیکه اشاره گر فایل آنجا قرار دارد) خوانده می شوند .

نوشتن در فایل :

۱ – دستور Print : توسط این دستور می توان اطلاعاتی را در فایل قرار داد :

Print #FileNumber,WriteData

دستور فوق محتویات متغیر WriteData را در فایلی با شماره FileNumber می نویسد . بوسیله دستور Print می توان اطلاعات را بصورت خط به خط در فایل نوشت برای مثال :

Print #۱,"Hello Visual Basic"+Vbcrlf

عبارت Vbcrlf نشان دهنده کاراکتر انتهای خط (CRLF) در ویژوال بیسیک می باشد .

۲ – دستور Put : این دستور برای نوشتن داده ها در فایلهای باینری با دسترسی تصادفی بکار می رود :

Put #FileNumber,[Recordnum],WriteData

این دستور محتویات متغیر WriteData را در رکورد شماره Recordnum قرار می دهد .

تعیین محل رکورد در فایلهای با دسترسی تصادفی :

برای منتقل کردن اشاره گر فایل به یک رکورد در یک فایل باینری با دسترسی اتفاقی ، از دستور Seek استفاده می شود . این دستور محل یک بایت را در فایل می گیرد و اشاره گر فایل را در آنجا قرار می دهد بنابراین دستورات Put و Get می توانند با این رکورد کار کنند :

Seek #FileNumber,RecordNumber

تشخیص انتهای فایل :

برای اینکه متوجه شویم به انتهای یک فایل رسیده ایم از دستور EOF استفاده می کنیم . این دستور یکی از مقادیر True یا False را بر می گرداند که نشان می دهد به انتهای فایلرسیده ایم یا نه . از این تابع در حلقه های Do-While استفاده می شود :

Do While Not (EOF(FileNumer))x

.

.

.

Loop

حلقه فوق تا زمانیکه فایل موردنظر به انتها نرسیده باشد اجرا خواهد شد .

بدست آوردن طول یک فایل :

بوسیله دستور LOF می توان طول محتویات یک فایل را بدست آورد :

```
FileSize=LOF(FileNumber)x
```

بدست آوردن محل اشاره گر فایل :

توسط دستور Loc می توان محل جاری اشاره گر فایل را بدست آورد .

اجرا شدن این دستور محل اشاره گر را تغییر نمی دهد :

```
FilePosition=Loc(FileNumber)x
```

کار با فایل در ویژوال بیسیک - بخش سوم

سایر عملیات کار با فایل :

۱ - حذف فایل : برای حذف یک یا چند فایل از دستور Kill استفاده می شود :

```
Kill "C:\Temp\MyFile.txt"x
C:\Temp\*.txt"x" Kill
```

۲ - انتقال فایل : برای انتقال یک فایل از یک دایرکتوری به دایرکتوری دیگر از دستور Name استفاده می شود . مبدا و مقصد بایستی روی یک درایو باشند . اگر دایرکتوری مبدا و مقصد یکی باشد فایل تغییر نام داده می شود :

```
C:\Temp\File1.txt" To "C:\Temp2\File2.txt"x" Name
```

۳ – کپی کردن فایل : برای کپی کردن یک فایل از یک دایرکتوری به دایرکتوری دیگر از دستور FileCopy استفاده می شود :

```
FileCopy "\File1.txt\ To "C:\Temp\File2.txt"x
```

۴ – بدست آوردن تاریخ و زمان آخرین تغییر فایل و یا زمان ایجاد فایل : برای این کار از دستور FileDateTime استفاده می شود . ابتدا بایستی یک متغیر از نوع Variant تعریف کرده و سپس توسط این دستور تاریخ و زمان موردنظر را استخراج کنیم :

```
Variant Dim FileInfo As
FileInfo=FileDateTime("C:\Temp\MyFile.txt")x
```

۵ – استخراج طول فایل : برای بدست آوردن طول یک فایل بر حسب بایت از دستور FileLen استفاده می شود :

```
FileSize=FileLen("C:\MyFile.txt")x
```

۶ – تغییر صفت یک فایل : برای تغییر صفت یک فایل از دستور SetAttr استفاده می شود . پارامترهای این دستور عبارتند از :

۰ : فایل معمولی

۲ : فایل مخفی

۴ : فایل سیستمی

```
FileNumber,FileAttrib SetAttr
```

مقابله با خطاهای کار با فایل :

در زمان کار با فایلهای احتمال زیادی وجود دارد که خطا بوجود آید . بنابراین بایستی در زمان کار با فایلها در صورت ممکن از روتینهای

مقابله با خطا استفاده کنیم . شایع ترین خطاهای کار با فایل عبارتند از :

۵۲ : شماره یا نام فایل صحیح نیست

۵۳ : فایل پیدا نشد

۵۴ : حالت فایل صحیح نیست

۵۵ : فایل قبلاً باز شده

۵۸ : فایل از قبل وجود دارد

۵۹ : طول رکورد صحیح نیست

۶۱ : دیسک پر است

۶۲ : عبور از انتهای فایل

۶۳ : شماره رکورد صحیح نیست

۷۰ : دسترسی ممنوع است

۷۱ : دیسک آماده نیست

۷۶ : مسیر پیدا نشد

در هنگام مقابله با خطا بهتراست از یک ساختار Select-Case استفاده کنید :

```
Err Select Case
Case 71
MsgBox "Drive is Not Ready"x
.
.
.
End Select
```

API های ویندوز

API های ویندوز

امروز قصد دارم در مورد API های ویندوز و چگونگی استفاده از آنها در ویژوال بیسیک بطور خلاصه توضیح دهم و همچنین دو مثال پراستفاده را نیز در این زمینه بیان کنم که عبارتند از چگونگی پخش فایلهای Wav و ساخت یک تایمر با دقت بالا :

۱ – آشنایی با Windows API : واژه API مخفف Application Programming Interface می باشد . API های ویندوز مجموعه ای از توابع از پیش آماده موجود در سیستم عامل هستند که شما می توانید آنها را در برنامه های خود فراخوانی کنید . این توابع در چندین کتابخانه DLL ویندوز ذخیره شده اند . برای دسترسی به این توابع در ویژوال بیسیک ابتدا باید آنها را برنامه خود declare کنید . برای مثال :

```
Declare Function sndPlaySound Lib "winmm.dll" Alias
sndPlaySoundA" (ByVal lpszSoundName As String, "
Long ByVal uFlags As Long) As
```

همانطور که می بینید مثال فوق یک Declare از تابع sndPlaySound می باشد که این تابع در کتابخانه Winmm.dll موجود است . کلمه Alias نشان می دهد که این تابع نام دیگری در dll دارد . سایر بخشها مربوط به تعریف پارامترهای تابع می باشند که در مورد مثال فوق ، این تابع دو پارامتر ورودی و یک خروجی از نوع Long دارد .
پس از Delare کردن API در برنامه می توانید از آن استفاده نمائید .

۲ – پخش فایلهای Wav : تابعی که برای پخش فایلهای Wav استفاده می شود تابع sndPlaySound است که در بالا با آن آشنا شدید .
پارامتر lpzSoundName نام و مسیر فایل Wavو پارامتر uFlags چگونگی پخش فایل را مشخص می کند . مقادیر ممکن این پارامتر عبارتند از :
– SND_ASYNC : اجازه می دهد طوری فایل Wav پخش شود که آنرا بتوان وقفه داد . بعبارت دیگر قادر خواهید بود فایل Wav تان را هر زمان که بخواهید پخش کنید و مطمئن باشید که حتماً شنیده می شود .
– SND_LOOP : فایل Wav را بطور ممتد پخش می کند .
– SND_NODEFAULT : اگر فایل Wav پیدا نشود صدای دیگری پخش نخواهد شد (مثلاً برخی صداهای default ویندوز)
– SND_SYNC : در طول پخش فایل Wav کنترل به برنامه داده نمی شود . این پارامتر در زمانیکه می خواهید فایل Wav ای را در پس زمینه برنامه تان پخش کنید مناسب نمی باشد .
– SND_NOSTOP : اگر فایل Wav ای قبلاً در حال پخش باشد ، فایل Wav شما آنرا دچار وقفه نمی کند . از این پارامتر زمانی استفاده می شود که بخواهیم فایل Wav مان هیچوقت در وسط کار قطع نشود .
اگر بخواهید از بیش از یکی از این پارامترها استفاده کنید توسط Or آنها را ترکیب نمائید مثال :

```
or  ding.wav", SND_ASYNC\" & sndPlaySound App.path
SND_LOOP
```

نکته : برای استفاده از توابع صوتی پیچیده تر بایستی از DirectSound که یکی از اجزای DirectX می باشد استفاده کنید . در

مورد DirectSound بعداً صحبت خواهم کرد .

۳ – ساخت یک تایمر با دقت بالا : شاید تا بحال از کنترل تایمر موجود در نوار ابزار ویژوال بیسیک استفاده کرده باشید . این تایمر دارای دقت حدود ۵۵ میلی ثانیه است . برای دستیابی به زمانهای با دقت بالاتر این کنترل مفید نخواهد بود .
تابع GetTickCount یک API موجود در کتابخانه Kernel32.dll است . این تابع طول زمانی را که سیستم شروع به کار کرده است را برحسب میلی ثانیه برمی گرداند :

```
() "Private Declare Function GetTickCount Lib "kernel32
As Long
```

برای بررسی طی شدن یک مدت زمانی خاص شما ابتدا باید مقدار این تابع را در یک متغیر کمکی مثل TempTime قرار دهید سپس در یک حلقه Do-Loop باید اختلاف زمان GetTickCount جدید و زمان TempTime را با مقدار زمانی که می خواهید سپری شود مقایسه کنید :

```
TempTime = GetTickCount()x
GetTickCount() - TempTime > Do While DesiredTime
'Do some things
Loop
```

توسط کد بالا می توان یک عملیات خاص را برای یک مدت زمانی مشخص اجرا کرد .
کد زیر نشان می دهد که چگونه می توان دستورات خاصی را در فواصل زمانی خاص اجرار کرد :

```
ExitFunction = False
GetTickCount()x = TempTime
Do While not(ExitFunction)x
GetTickCount() - TempTime then > If DesiredTime
'Reset the temporary variable
GetTickCount()x = TempTime
'Do some things
End If
Loop
```

همچنین از تابع GetTickCount می توان برای benchmark برنامه ها استفاده کرد . بعبارت دیگر می توان زمان اجرای یکسری دستورات خاص را بدست آورد .

ایجاد ساختارهای داده ای در ویژوال بیسیک

بخش اول

ایجاد ساختارهای داده ای در ویژوال بیسیک - بخش اول

مقدمه :

ساختارهای داده ای از نظر تعداد اعضا به دو دسته استاتیک و دینامیک تقسیم می شوند . ساختارهای استاتیک مثل آرایه های یک بعدی و آرایه های دو بعدی ، تعداد اعضای آنها در زمان طراحی برنامه مشخص می

شود و در طول اجرای برنامه ثابت است اما تعداد اعضای ساختارهای داده ای دینامیک در طول اجرای برنامه تغییر می کند . لیست پیوندی (LinkList) ، پشته (Stack) ، صف (Queue) و درختهای باینری (Tree Binary) ، نمونه هایی از ساختارهای داده ای دینامیک هستند .
لیست پیوندی شامل مجموعه ای از عناصر داده ای است که اضافه و حذف اعضا در هر جای لیست ممکن است .
پشته یک ساختار داده ای مهم در کامپایلرها و سیستم های عامل است که عمل اضافه و حذف عناصر از ابتدای آن انجام می شود .
صف یک ساختار داده ای است که عمل اضافه کردن از انتها و عمل حذف کردن از ابتدای آن انجام می شود .
درختهای دودویی برای جستجوی بسیار سریع ، ذخیره سازی داده ها و کامپایل عبارات استفاده می شوند .

نوع داده Variant :

نوع داده variant برای متغیرهایی بکار می رود که بطور صریح نوع آنها تعریف نشده است مثال :

```
Variant Dim value As
```

این نوع داده می تواند هر نوع داده ای را در خود ذخیره کند . همچنین برای ایجاد ساختارهای داده ای مثل لیست های پیوندی ، صف ، پشته و درخت مناسب است .
نوع داده موجود در variant می توان توسط توابع VarType و TypeName تعیین کرد . تابع VarType یک مقدار صحیح برمی گرداند که نشان دهنده نوع ذخیره شده در variant است .
مثال :

```
Dim value as Variant
```

value="Hello"x
در اینصورت مقدار بازگشتی (VarType(value برابر ۴ خواهد بود .
تابع TypeName یک رشته برمی گرداند که نشان دهنده نام نوع داده ذخیره شده در variant است .

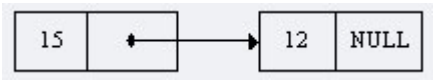
اخذ حافظه بطور دینامیک Dynamic Memory Allocation :

برای ایجاد و نگهداری ساختارهای داده ای دینامیک بایستی در هنگام اجرای برنامه بتوان فضای بیشتری برای نگهداری داده های جدید بدست آورد . با استفاده از کلمه کلیدی New می توان در ویژوال بیسیک حاقظه دینامیک گرفت :
Set NewNode=New ListNode
که ListNode یک شی از ساختار داده ای مورد نظر ماست .

کلاسهای خود ارجاعی :

کلاس خودارجاعی نوعی کلاس است که دارای یک اشاره گر (Pointer) به یک شی از همان نوع کلاس باشد . برای مثال اگر کلاس ما به اسم ClistNode باشد و متغیر زیر را در آن تعریف کنیم ، این کلاس یک کلاس خود ارجاعی است :
ClistNode Private mNextNode as
از mNextNode برای لینک دادن اعضای یک ساختار داده ای دینامیک بهم استفاده می شود (بعبارت دیگر گره زدن یک شی از کلاس ClistNode به یک شی دیگر از همان کلاس) . شی های خودارجاعی می توانند به همدیگر لینک شوند و ساختارهای داده ای مثل لیست پیوندی ، صف ، پشته و درخت را ایجاد کنند .

شکل زیر دو شی خود ارجاعی را نشان می دهد که بصورت یک لیست بهم لینک شده اند . عبارت NULL بدین معنا است که شی خودارجاعی به شی دیگری اشاره نمی کند (Nothing) و نشان دهنده انتهای ساختار داده است .



ایجاد ساختارهای داده ای در ویژوال بیسیک – بخش دوم

لیست پیوندی

همانطور که گفته شد لیست پیوندی مجموعه ای از یکسری داده است که این داده ها از نوع اشیا خودارجاعی هستند . (هر شی خودارجاعی دارای یک متغیر نوع variant برای نگهدار مقدار و یک اشاره گر به شی بعدی است) . هر عضو لیست پیوندی را یک گره گویند . هر لیست پیوندی از طریق یک اشاره گر به اولین گره قابل دسترسی است . گره های بعدی از طریق قسمت لینک موجود در هر گره قابل دسترس هستند . همچنین لینک آخرین گره با Nothing تنظیم می شود که انتهای لیست را نشان می دهد .

مزیت اصلی لیست های پیوندی نسبت به آرایه اینست که تعداد عناصر لیست پیوندی قابل تغییر است . بعبارت دیگر لیست های پیوندی بصورت دینامیک هستند و طول آنها قابل تغییر است اما سایز آرایه ثابت است . (البته ویژوال بیسطک از آرایه های با سایز متغیر نیز پشتیبانی می کند اما این عمل تغییر سایز اتوماتیک نیست .)

عمل درج در لیست پیوندی ساده است و تنها بایستی دو اشاره گر تغییر یابد .

لیست های پیوندی را می توان به سادگی با قراردادن هر عضو جدید در محل صحیح بصورت sortشده نگهداری کرد .

اعضای لیست پیوندی در حافظه بصورت پیوسته ذخیره نمی شوند بنابراین نمی توان فوراً به هر عضو لیست دسترسی داشت (بر خلاف آرایه) .

برای ایجاد لیست پیوندی در ویژوال بیسیک نیاز به سه کلاس است :

۱ – کلاس ClistNode : کلاسی است که هر گره از لیست را توصیف می کند :

```
private mNodeData as Variant
ClistNode private mNextNode as
public Property Get Data() as Variant
Data=mNodeData
Property End
Variant)x Public Property Let Dta(ByVal vNewValue as
MNodeData=vNewValue
End Property
NextNode() as ClistNode Public Property Get
Set NextNode=mNextNode
End Property
Property Let NextNode(Byval vNewValue as  Public
Variant)x
mNextNode=vNewValue Set
End Property
```

۲ – کلاس Clist برای توصیف لیست پیوندی .

mFirstNode برای اشاره به اولین ClistNode و mLastNode برای اشاره به آخرین ClistNode در یک شی clist بکار می رود . زمانیکه یک

Clsit ایجاد می شود این دو متغیر با Nothing تنظیم می شوند . روال Property Get Iterator یک شی ClistIterator برمی گرداند که می توان از آن برای حرکت در بین اعضای لیست استفاده کرد .

```
Private mFirstNode as ClistNode
as ClistNode Private mLastNode
Public Function IsEmpty() as boolean
Is Nothing,True,False)x IsEmpty=IIf(mFirstNode
End function
as variant)x Public Sub InsertAtFront(insertItem
Dim tempNode as ClistNode
If IsEmpty() then
mFirstNode=New ClistNode Set
Set mLastNode=mFirstNode
Else
tempNode=mFirstNode Set
ClistNode Set mFirstNode=New
MFirstNode.NextNode=tempNode
if End
MFirstNode.Data=insertItem
End sub
InsertAtBack(insertItem as Variant)x Public sub
Dim tempNode as ClistNode
IsEmpty() then If
Set mLastNode=New ClistNode
mFirstNode=mLastNode Set
Else
Set tempNode=mLastNode
ClistNode Set mLastNode=New
TempNode.NextNode=mLastNode
if End
MLastNode.Data=insertItem
End sub
RemoveFromFront()x Public function
Dim removeItem as Variant
If IsEmpty() then
```

```
list is empty Msgbox
RemoveFromFront=Null
Exit function
if End
RemoveItem=mFirstNode.Data
If mFirstNode Is mLastNode then
mFirstNode=Nothing Set
Set mLastNode=Nothing
Else
mFirstNode=mFirstNode.NextNode Set
End if
RemoveFromFront=removeItem
function End
Public Function RemoveFromBack()x
Variant Dim removeItem as
Dim current as ClistNode
If IsEmpty() then
empty Msgboc list is
RemovefromBack=Null
Exit function
if End
RemoveItem=mLastNode.Data
If mFirstNode Is mLastNode then
mFirstNode=nothing Set
Set mLastNode=Nothing
Else
current=mFirstNode Set
While Not current.NextNode Is mLastNode
current=current.NextNode Set
Wend
mLastNode=current Set
Current.NextNode=nothing
if End
RemoveFromBack=removeItem
End function
Iterator() as variant Public property Get
Dim iter as ClistIterator
```

```
ClistIterator Set iter=New
Iter.StartNode=mFirstNode
Set Iterator=iter
property End
```

عملکرد روال InsertAtFront :

a – فراخوانی IsEmpty برای تعیین خالی بودن لیست

b – اگر لیست خالی باشد mFirstNode و mLastNode به New ClsitNode اشاره می کنند .

c – اگر لیست خالی نباشد گره جدید توسط اشاره دادن tempNode به اولین گره لیست و سپس اشاره دادن mFirstNode به گره New ClsitNode و سپس اشاره دادن mFirstNode.NextNode به tempNode ساخته می شود .

d – تنظیم mFirstNode.Data با مقدار مورد نظر

عملکرد روال InsertAtBack :

a – فراخوانی IsEmpty برای تعیین خالی بودن لیست

b – اگر لیست خالی باشد mFirstNode و mLastNode به New ClsitNode اشاره می کنند .

c – اگر لیست خالی نباشد گره جدید توسط اشاره دادن tempNode به آخرین گره لیست و سپس اشاره دادن mLastNode به گره New ClsitNode و سپس اشاره دادن tempNode.NextNode به mLastNode ساخته می شود .

d – تنظیم mLastNode.Data با مقدار مورد نظر

عملکرد روال RemoveFromFront :

a – اگر لیست خالی باشد Null برگشت داده می شود .

b – اگر لیست خالی نباشد داده mFirstNode به removeItem

اختصاص داده می شود .

c – اگر لیست فقط یک گره داشته باشد mFirstNode و mLastNode با Nothing مقدار دهی می شوند و گره از لیست حذف می شود .

d – اگر گره بیش از یک عضو داشته باشد mFirstNode برابر mFirstNode.NextNode می شود .

e – مقدار removeItem برگشت داده می شود .

عملکرد روال RemoveFromBack :

a – اگر لیست خالی باشد Null برگشت داده می شود .

b – اگر لیست خالی نباشد داده mLastNode به removeItem اختصاص داه می شود .

c – اگر لیست یک گره داشته باشد mFirstNode و mLastNode با Nothing مقدار دهی می شوند و گره از لیست حذف می شود .

d – اگر لیست بیش از یک گره داشته باشد متغیر current برابر mFirstNode می شود . سپس با استفاده از current روی گره های لیست حرکت می کنیم تا به گره ای برسیم که به آخرین گره اشاره می کند . سپس mLastNode را به گره ای که current به آن اشاره می کند قرار می دهیم و مقدار current.NextNode را Nothing می کنیم تا بعنوان آخرین گزه لیست معرفی شود .

e – مقدار removeItem برگشت داده می شود .

۳ – کلاس ClistIterator : این کلاس برای حرکت روی گره های لیست و دستکاری هر گره بکار می رود . از حرکت کننده ها برای چاپ لیست و یا انجام دادن عملی بر روی هر عضو Clist می توان استفاده کرد . این کلاس دارای دو متغیر از نوع ClistNode به نامهای mBookmark و mFirstNode است . متغیر mFirstNode به اولین گره در Clist

اشاره می کند و متغیر mBookmark موقعیت فعلی حرکت کننده بر روی Clist را نشان می دهد . روال StartNode Property Let این دو متغیر را مقدار دهی اولیه می کند . تابع NextItem اگر مقدار mBookmark برابر Null باشد ، Null برگشت می دهد و در غیراینصورت مقدار tempData را برابر mBookmark.Data و مقدار mBookmark را برابر mBookmark.NextNode قرار می دهد . تابع HasMoreItems اگر لیست دارای چندین عضو باشد True برمی گرداند . روال ResetBookmark حرکت کننده را به ابتدای لیست منتقل می کند .

```
Private mBookmark as ClistNode
as ClistNode Private mFirstNode
Public Property Let StartNode(Byval vNewValue as
variant)x
Set mFirstNode=vNewValue
Set mBookmark=mFirstNode
property End
Public function NextItem()x
Dim tempData as varaint
then mBookmark Is nothing If
NextItem=Null
Else
TempData=mBookmark.Data
mBookmark=mBookmark.NextNode Set
NextItem=tempData
End if
function End
Public function HasMoreItems() as boolean
mBookmark Is  HasMoreItems=IIf(Not
nothing,True,False)x
End function
ResetmBookmark()x Public sub
```

```
MBookmark=mFirstNode
End sub
```

ایجاد ساختارهای داده ای در ویژوال بیسیک - بخش سوم

مثالی از استفاده از کلاسهای لیست پیوندی :

ابتدا کلاسهایی که در جلسه قبل معرفی شد را به پروژه تان اضافه کنید .

سپس در بخش کدنویسی فرمتان ، ابتدا یک شی از نوع کلاس Clist بصورت زیر تعریف کنید :

```
Dim list as New Clist
```

در فرمتان سه CommandButton با نامهای AddFirst، AddLast و ShowList و نیز یک TextBox با نام ListMember قرار دهید .

کد زیر را برای رویداد کلیک شدن دکمه AddFirst بنویسید :

```
Call list.InsertAtFront(ListMember.text)x
```

کد زیر را برای رویداد کلیک شدن دکمه AddLast بنویسید :

```
Call list.InsertAtBack(ListMember.text)x
```

کد زیر را برای رویداد کلیک شدن دکمه ShowList بنویسید :

```
Dim elements as New ClistIterator
elements=list.Iterator Set
If elements.HasMoreItems=false then msgbox ("list is
empty")x
Else
elements.HasMoreItems While
```

```
Msgbox(elements.NextItem)x
Wend
if end
```

پشته :

پشته نوعی لیست پیوندی است که گره های جدید ، فقط به انتهای آن می توانند اضافه شوند . بهمین دلیل به پشته ، ساختمان داده LIFO می گویند . قسمت لینک آخرین گره پشته با Nothing مقدار دهی می شود که نشان دهنده پایین پشته است .

روالهای اصلی پشته Push و Pop هستند .

Push یک گره جدید به بالای پشته اضافه می کند و Pop از بالای پشته گره ای را حذف کرده و مقدار داده آن را بر می گرداند .

ایجاد ساختارهای داده ای در ویژوال بیسیک - بخش چهارم

کلاس پشته :

همانطور که در بخش قبل گفته شد پشته نوعی لیست پیوندی است که گره های جدید فقط به انتهای آن اضافه شوند . روالهای اصلی پشته Push و Pop هستند .

Push یک گره جدید به بالای پشته اضافه می کند و Pop از بالای پشته گره ای را حذف کرده و مقدار داده آن را بر می گرداند .

یک کلاس پشته را با استفاده از کلاس Clist و بصورت زیر پیاده سازی می کنیم :

```
Private list As New Clist
as Variant)x Public Sub Push(value
List.InsertAtFront(value)x
```

```
End sub
Variant Public Function Pop As
Pop=list.RemoveFromFront()x
End Function
IsStackEmpty() As Boolean Public Function
IsStackEmpty=list.IsEmpty()x
function End
Public Property Get Iterator() as variant
Iterator=list.Iterator Set
End Property
```

در این کلاس ابتدا یک شی از نوع کلاس Clist تعریف شده است . سپس متدهای Push توسط متد InsertAtFront و Pop توسط متد RemoveFromFront پیاده سازی شده اند .

یک برنامه نمونه :

برای نوشتن یک برنامه برای کار با پشته ابتدا کلاس Stack را که کد آن را در بالا دیدید به پروژه تان اضافه کنید . سپس در بخش کد مربوط به فرمتان ابتدا یک شی از نوع کلاس Stack بصورت زیر تعریف کنید :

```
Dim mStack as New Stack
```

سپس در فرمتان سه CommandButton با نامهای Push و Pop و ShowStack و نیز یک TextBox با نام StackMember قرار دهید .

کد زیر را برای کلیک شدن دکمه Push بنویسید :

```
mStack.push(StackMember.text)x
```

کد زیر را برای کلیک شدن دکمه Pop بنویسید :

StackMember.text=mStack.Pop()x

کد زیر را برای کلیک شدن دکمه ShowStack بنویسید :

```
Dim elements as New ClistIterator
elements=mStack.Iterator Set
If elements.HasMoreItems=false then msgbox "stack is
empty"x
Else
While elemets.HasMoreItems
elements.NextItem Msgbox
Wend
```

ایجاد ساختارهای داده ای در ویژوال بیسیک - بخش پنجم

صف :

صف نوعی ساختار داده ای است که گره ها از ابتدای صف (سر صف head) حذف می شوند و از انتهای صف (ته صف tail) اضافه می شوند . بنابر این ، صف یک ساختار داده ای FIFO است . صف دارای دو متد به نامهای AddQueue و DelQueue است که اولین متد ، عنصری را به انتهای صف اضافه می کند و دومین متد ، عنصری را از ابتدای صف حذف می کند .

برای ایجاد کلاس Cqueue از کلاس Clist استفاده می کنیم :

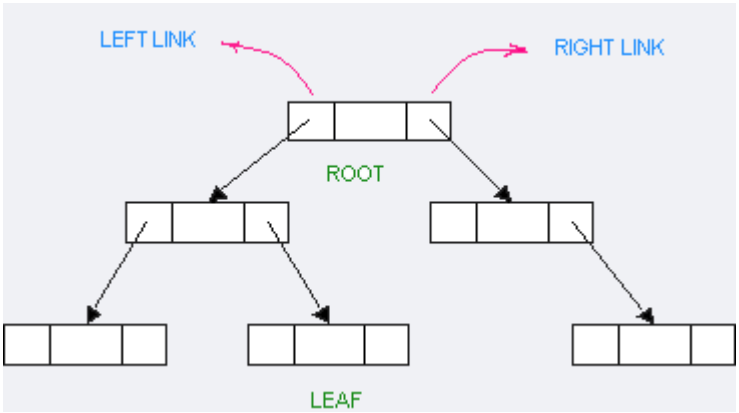
```
Private list as New Clist

AddQueue(value as Variant)x Public Sub
(List.InsertAtBack(value
End sub

Function DelQueue() as Variant Public
```

```
DelQueue=list.RemoveFromFront
function End

Public property Get Iterator() as Variant
Iterator=list.Iterator Set
End Property
```

درخت :

لیستهای پیوندی ، پشته ها و صف ها جزو ساختارهای داده ای خطی هستند در حالیکه یک درخت ، یک ساختار داده ای دو بعدی با خصوصیات ویژه ای است . گره های درخت دارای دو یا چند لینک هستند . در اینجا در مورد درختهای دودویی یا باینری بحث می کنیم که در آن همه گره ها دارای دو لینک هستند . گره ریشه اولین گره در درخت است . هر لینک گره ریشه ، به یک فرزند اشاره می کند . به فرزندان یک گره Siblings می گویند . به گره بدون فرزند ، برگ یا Leaf گفته می شود .

درختهای جستجوی باینری درخت هایی هستند که در آنها مقدار فرزند چپ هر گره کمتر از گره پدر و مقدار فرزند سمت راست هر گره بیشتر از گره پدر می باشد .



ایجاد ساختارهای داده ای در ویژوال بیسیک - بخش ششم

کلاس درختهای جستجوی باینری

برای ایجاد درختهای جستجوی باینری در ویژوال بیسیک نیاز به ایجاد دو کلاس داریم :

۱ – کلاس CTreeNode که هر ند درخت دودویی را توصیف می کند . این کلاس دارای یک متغیر به نام mNodeData از نوع Variant برای نگهداری داده هر گره است . همچنین دارای دو متغیر اشاره گر به نامهای mLeft و mRight می باشد که به ترتیب به فرزند چپ و فرزند راست درخت اشاره می کنند .

متد Get Data مقدار داده هر گره را بر می گرداند و متد Let Data مقدار داده هر گره را تنظیم می کند .

متد Get Left آدرس فرزند چپ هر گره را برمی گرداند و متد Left Let فرزند چپ هر گره را تنظیم می کند .

متد Get Right آدرس فرزند راست هر گره را برمی گرداند و متد Let Right فرزند راست هر گره را تنظیم می کند .

متد Insert برای اضافه کردن فرزند به یک گره به کار می رود . اگر مقدار گره ای که می خواهیم بعنوان فرزند به درخت اضافه کنیم کوچکتر از مقدار خود گره باشد بعنوان فرزند چپ و در غیر اینصورت بعنوان فرزند راست به گره اضافه می شود . اضافه شدن نیز بدین صورت است که ابتدا بررسی می شود آیا گره قبلاً فرزندی داشته است یا نه ؟ اگر نداشته باشد (mLeft و یا mRight برابر Nothing باشد) این گره جدید مستقیماً به گره متصل می شود اما اگر گره قبلاً فرزندی داشته باشد متد Insert برای آن فرزند اضافه می شود و اینکار تا جایی ادامه می یابد که به گره ای برسیم که فرزندی نداشته باشد :

```
Private mLeft as CtreeNode
CtreeNode Private mRight as
Private mNodeData as Variant


variant Public Property Get Data() as
Data=mNodeData
End property
vNewValue as Variant)x Public Property Let Data(Byval
MNodeData=vNewValue
End property
Get Left() as variant Public Property
Set Left=mLeft
End property
Left(Byval vNewValue as variant)x Public Property Let
Set mLeft=vNewValue
property End


Public Property Get Right() as variant
Right=mRight Set
End Property
variant)x Public Property Let Right(Byval vNewValue as
Set mRight=vNewValue
End Property


as variant)x Public Sub Insert(value
If value
If mLeft Is Nothing Then
mLeft=New CtreeNode Set
MLeft.Data=value
Else
MLeft.Insert(value)x
if End
mNodeData then<Elseif value
If mRight Is Nothing then
CtreeNode mRight=New Set
MRight.Data=value
```

```
Else
MRight.Insert(value)x
End if
if End
End sub
```

۲ – کلاس CTree : این کلاس برای ایجاد درخت بکار می رود . این کلاس دارای متغیری بنام mRoot از نوع CTreeNode برای تعریف ریشه درخت است . همچنین یک متغیر mOutputString برای نمایش دادن اعضای درخت دارد .

```
Private mRoot as CtreeNode
as String Private mOutputString

Public Sub InsertNode(value as Varaint)x
then If mRoot Is Nothing
CtreeNode Set Mnode=New
MRoot.Data=value
Else
MRoot.Insert(value)x
End if
sub End

Public PreorderTraversal()x
MOutputString=””x
PreorderHelper(mRoot)x Call
End sub

CtreeNode)x Private Sub PreorderHelper(node As
If node Is nothing Then
Exit sub
if End
x“ “ & node.Data & MOutputString=mOutputString
```

```
PreorderHelper(node.left)x Call
Call PreorderHelper(node.right)x
sub End

Public Sub InorderTraversal()x
MOutputString=””x
InorderHelper(mRoot)x Call
End sub

CtreeNode)x Private Sub InorderHelper(node as
If node Is nothing then
Exit sub
End if
InorderHelper(node.Left)x Call
x“ “ & node.Data & MOutputString=mOutputString
Call InorderHelper(node.Right)x
End sub

PostorderTraversal()x Public
MOutputString=””x
PostorderHelper(mRoot)x Call
End sub

CtreeNode)x Private Sub PostorderHelper(node as
If node Is Nothing then
Exit sub
End if
PostorderHelper(node.Left)x Call
PostorderHelper(node.Right)x Call
x“ “ & node.Data & MOutputString=mOutputString
End sub

Varaint Public Property Get Output() as
```

```
Output=mOutputString
End Property
```

ایجاد ساختارهای داده ای در ویژوال بیسیک - بخش پایانی

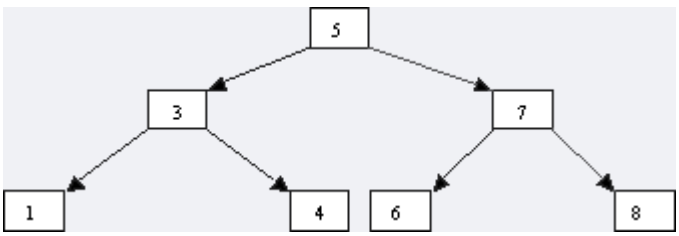
انواع روشهای پیمایش عناصر درخت :

۱ - روش InOrder : در این روش ابتدا عناصر نیمه سمت چپ درخت ، سپس ریشه و در آخر عناصر نیمه سمت راست درخت نمایش داده می شوند .

۲ - روش PreOrder : در این روش ابتدا ریشه درخت ، سپس عناصر نیمه سمت چپ و در پایان عناصر نیمه سمت راست درخت نمایش داده می شوند .

۳ - روش PostOrder : در این روش ابتدا عناصر نیمه سمت چپ درخت ، سپس عناصر نیمه سمت راست درخت و در پایان ریشه درخت نمایش داده می شوند .

مثال : درخت زیر را در نظر بگیرید :



نتیجه پیمایش InOrder درخت : ۱،۳،۴،۵،۶،۷،۸

نتیجه پیمایش PreOrder درخت : ۵،۳،۱،۴،۷،۶،۸

نتیجه پیمایش PostOrder درخت : ۱،۴،۳،۶،۸،۷،۵

بررسی متدهای کلاس CTree :

متد InsertNode : اگر گره ریشه برابر Null باشد value را برابر مقدار گره ریشه قرار می دهد . در غیر اینصورت متد Insert مربوط به گره ریشه فراخوانی می شود .

متد PreorderTraversal : رشته چاپ عناصر ریشه را خالی می کند و سپس متد پیمایش Preorder را فراخوانی می کند .

متد InorderTraversal : رشته چاپ عناصر ریشه را خالی می کند و سپس متد پیمایش Inorder را فراخوانی می کند .

متد PostorderTraversal : رشته چاپ عناصر ریشه را خالی می کند و سپس متد پیمایش Postorder را فراخوانی می کند .

متد Get Output : عناصر پیمایش شده درخت را برمی گرداند .

یک برنامه نمونه :

ابتدا کلاسهای CTreeNode و CTree را به پروژه تان اضافه کنید .

سپس متغیر زیر را در قسمت کدنویسی فرمتان تعریف کنید :

```
Dim mTree as New Ctree
```

سپس در فرمتان یک Textbox با نام Value و دو Command Button با نامهای Insert و Show قرار دهید .

کد زیر را برای وارد کردن عنصر به درخت برای دکمه Insert بنویسید :

```
mTree.InsertNode(Value.Text)x
```

کد زیر را برای پیمایش InOrder درخت برای دکمه Show بنویسید :

```
Call mTree.InorderTraversal
```

شی Collection :

ویژوال بیسیک دارای شی پیش ساخته ای به نام Collection است که می تواند مجموعه ای از مقادیر با هر نوع داده ای را در خود ذخیره کند . در واقع عناصر موجود در یک Collection می توانند دارای نوعهای داده ای متفاوت باشند . شی Collection قابلیت رشد دینامیک دارد . شی Collection توسط کلمه کلیدی New ایجاد می شوند . توسط متد Add می توان به Cllection عضو اضافه کرد و توسط متد Remove می توان عضوی را از آن حذف کرد . هر عضو از Collection توسط متد Item قابل دستیابی است . با استفاده از خاصیت Count می توان تعداد اعضای موجود در Collection را تعیین نمود . بصورت پیش فرض اعضای جدید به انتهای Collection اضافه می شوند ولی توسط آرگومانهای اختیاری متد Add می توان محل اضافه شدن را تغییر داد . متد Remove یک شماره می گیرد که موقعیت عضوی را که می خواهیم آنرا حذف کنیم مشخص می کند .

توسط دستورات زیر می توان اعضای یک Collection را نمایش داد :

```
Dim mCollection as New Collection
Variant Dim element as
.
.
.
For Each element In mCollection
element Msgbox
```

element متغیری از نوع variant برای اشاره به هر عضو Collection می باشد .

کار با رجیستری در ویژوال بیسیک - قسمت اول

کار با رجیستری در ویژوال بیسیک - قسمت اول

رجیستری چیست ؟

سیستم عامل ویندوز تنظیمات سخت افزاری و نرم افزاری خود را بطور مرکزی در یک بانک اطلاعاتی با ساختار سلسله مراتبی ذخیره می کند که رجیستری نام دارد . رجیستری جایگزینی برای بسیاری از فایلهای پیکربندی INI، SYS و COM است که در نسخه های اولیه ویندوز موجود بود . رجیستری ، سیستم عامل را با مهیا کردن اطلاعات موردنیز برای اجرای برنامه ها و load شدن component ها ، کنترل می کند .

رجیستری شامل انواع مختلفی از اطلاعات می باشد مثل :

- اطلاعات سخت افزارهای نصب شده روی سیستم

- اطلاعات درایورهای نصب شده روی سیستم

- اطلاعات برنامه های نصب شده روی سیستم

- اطلاعات پروتکلهای شبکه ای مورد استفاده در سیستم

ساختار رجیستری شامل چندین مجموعه رکورد است که داده های این رکوردها توسط بسیاری از برنامه ها و اجزای سیستم عامل خوانده و یا نوشته می شود .

اجزای رجیستری

اجزای تشکیل دهنده رجیستری عبارتند از :

۱ – Subtree : subtree ها همانند folder های موجود در ریشه یک درایو هارد هستند . رجستری ویندوز دارای پنج subtree می باشد :

- HKEY_LOCAL_MACHINE : شامل تمام داده های پیکربندی برای کامپیوتر می باشد و شامل ۵ key است : Hardware، SAM، Security، Software و System

- HKEY_USERS : شامل داده های مربوط به تنظیمات سیستم عامل برای هر user است مثل تنظیمات desktop و محیط ویندوز

- HKEY_CURRENT_USER : شامل داده های کاربر فعلی سیستم

- HKEY_CLASSES_ROOT : شامل اطلاعات پیکربندی نرم افزار است مثل داده های OLE و داده های کلاسهای متناظر با فایل

- HKEY_CURRENT_CONFIG : شامل اطلاعات مورد نیاز برای تنظیمات داریورهای سخت افزاری و غیره

۲ – key : Key ها همانند folder ها و subfolder های روی هارد هستند . هر key متناظر با object های نرم افزاری یا سخت افزاری می باشد . subkey ها key هایی هستند که درون یکسری key قراردارند .

۳ – Entry : هر key دارای یک یا چند entry است . هر entry دارای سه بخش می باشد :

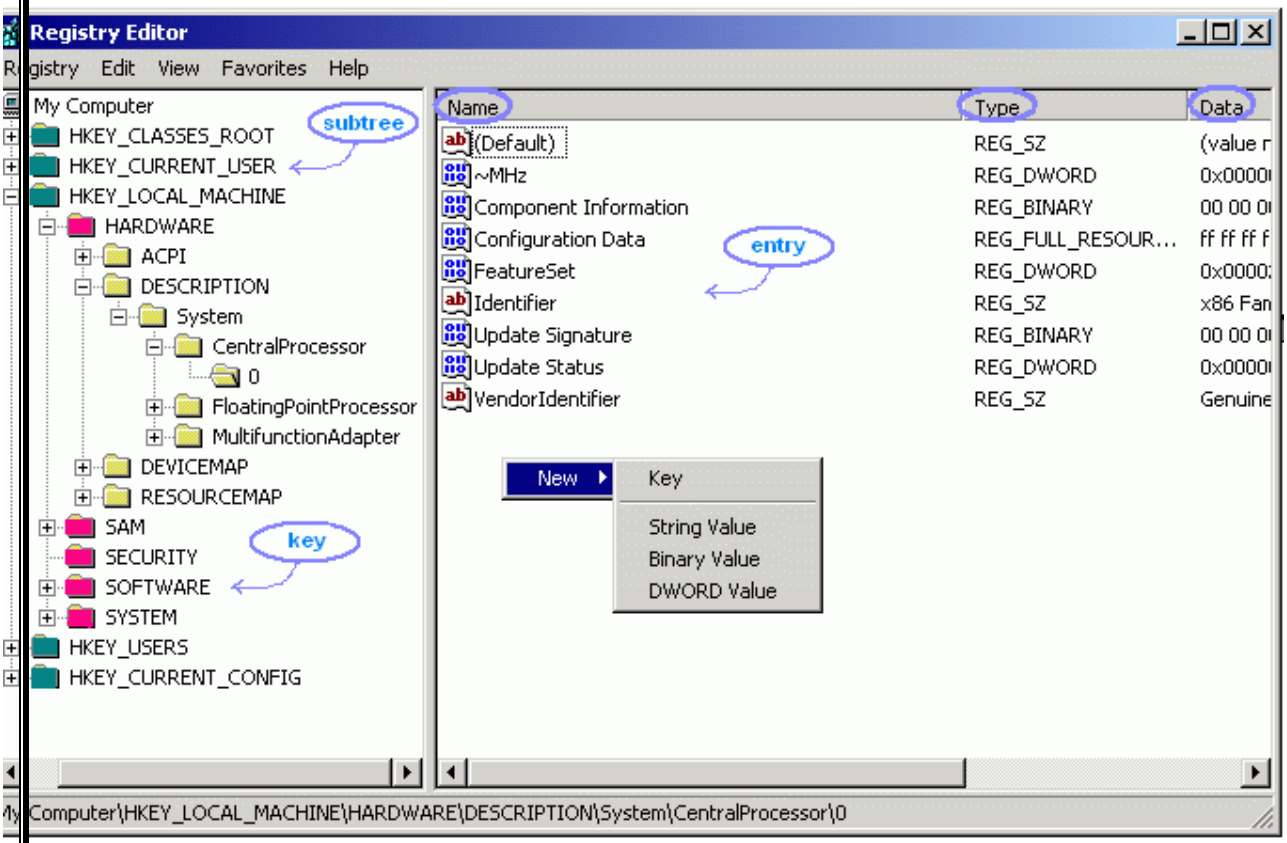
- نام Name

- نوع داده ای Data Type : مقدار هر entry یکی از انواع داده های زیر است :

REG_DWORD، REG_SZ، REG_EXPAND_SZ،
REG_BINARY،
REG_MULTI_SZ،
REG_FULL_RESOURCE_DESCRIPTOT

- مقدار Value

!Error



نکته ۱ : برای مشاهده رجیستری و اعمال تغییرات در آن (لطفاً اگر هیچ تجربه ای در تنظیم کردن رجیستری ندارید اطلاعات آنرا تغییر ندهید) ، می توانید از برنامه regedit.exe و یا regedt32.exe موجود در ویندوز استفاده کنید . برای اینکار کافیست نام برنامه را در کادر Run وارد کنید .

نکته ۲ : در قسمت دوم این مقاله کلاسی برای خواندن و نوشتن از/به رجیستری در ویژوال بیسیک خواهم نوشت و سپس نمونه ای از کاربردهای نوشتن یکسری اطلاعات خاص در رجیستری را خواهیم دید .

کار با رجیستری در ویژوال بیسیک – قسمت دوم

برای کار با رجیستری در ویژوال بیسیک کلاس Registery.bas را مطابق مطالب زیر ایجاد کرده و در پروژه های خود از آن استفاده کنید :

۱ – تعریف ثابتهای مورد نیاز : برای نوشتن این کلاس نیاز به تعریف چهار دسته ثابت داریم :

– ثابتهای مربوط به تعریف data type های entry های رجیستری :

```
Global Const REG_SZ As Long = 1
REG_DWORD As Long = 4 Global Const
```

– ثابتهای مربوط به تعریف key های رجیستری

```
H80000000& = Global Const HKEY_CLASSES_ROOT
H80000001& = Global Const HKEY_CURRENT_USER
H80000002& = Const HKEY_LOCAL_MACHINE Global
H80000003& = Global Const HKEY_USERS
```

– ثابتهای مربوط به خطاهای کار با رجیستری

```
Global Const ERROR_NONE = 0
ERROR_BADDB = 1 Global Const
Global Const ERROR_BADKEY = 2
۳ = Global Const ERROR_CANTOPEN
Global Const ERROR_CANTREAD = 4
۵ = Global Const ERROR_CANTWRITE
```

Global Const ERROR_OUTOFMEMORY = 6
٧ = Global Const ERROR_INVALID_PARAMETER
Global Const ERROR_ACCESS_DENIED = 8
ERROR_INVALID_PARAMETERS = 87 Global Const
Global Const ERROR_NO_MORE_ITEMS = 259

- ثابتهای متفرقه

H3F& = Global Const KEY_ALL_ACCESS
Const REG_OPTION_NON_VOLATILE = 0 Global

۲ - Declare کردن Api های مورد نیاز : برای کار با رجیستری از توابع کتابخانه Advapi32.dll استفاده می کنیم . این توابع عبارتند از :

- تابع RegCloseKey : آزاد کردن handle مربوط به یک key

ByVal) "Declare Function RegCloseKey Lib "advapi32.dll
hKey As Long) As Long

- تابع RegCreateKeyEx : ساخت یک key در رجیستری (اگر key قبلاً وجود داشته باشد ، این تابع آنرا باز می کند) :

"Declare Function RegCreateKeyEx Lib "advapi32.dll
Alias "RegCreateKeyExA" (ByVal hKey As Long, ByVal
Reserved As Long, ByVal lpSubKey As String, ByVal
lpClass As String, ByVal dwOptions As Long, ByVal
samDesired As Long, ByVal lpSecurityAttributes As Long,
lpdwDisposition As Long) As Long ,phkResult As Long

- تابع RegOpenKeyEx : باز کردن یک key

"Declare Function RegOpenKeyEx Lib "advapi32.dll Alias "RegOpenKeyExA" (ByVal hKey As Long, ByVal ulOptions As Long, ByVal lpSubKey As String, ByVal samDesired As Long, phkResult As Long) As Long

- تابع RegQueryValueExLong : استخراج type و data ی یک نام متناظر با یک key باز شده

Declare Function RegQueryValueExString Lib advapi32.dll" Alias "RegQueryValueExA" (ByVal hKey " As String, ByVal As Long, ByVal lpValueName lpReserved As Long, lpType As Long, ByVal lpData As lpcbData As Long) As Long ,String

Declare Function RegQueryValueExLong Lib advapi32.dll" Alias "RegQueryValueExA" (ByVal hKey " As String, ByVal As Long, ByVal lpValueName lpReserved As Long, lpType As Long, lpData As Long, Long) As Long lpcbData As

Declare Function RegQueryValueExNULL Lib advapi32.dll" Alias "RegQueryValueExA" (ByVal hKey " As String, ByVal As Long, ByVal lpValueName lpReserved As Long, lpType As Long, ByVal lpData As lpcbData As Long) As Long ,Long

- تابع RegSetValueEx : ذخیره یک مقدار در فیلد value یک کلید باز

Declare Function RegSetValueExString Lib advapi32.dll" Alias "RegSetValueExA" (ByVal hKey As " String, ByVal Reserved As Long, ByVal lpValueName As ,Long, ByVal dwType As Long, ByVal lpValue As String ByVal cbData As Long) As Long

advapi32.dll" " Declare Function RegSetValueExLong Lib Alias "RegSetValueExA" (ByVal hKey As Long, ByVal String, ByVal Reserved As Long, ByVal lpValueName As cbData As dwType As Long, lpValue As Long, ByVal Long) As Long

- تابع RegDeleteKey : پاک کردن یک کلید و کلیه اطلاعات مرتبط با آن

Lib &Private Declare Function RegDeleteKey advapi32.dll" Alias "RegDeleteKeyA" (ByVal hKey As " (String Long, ByVal lpSubKey As

- تابع RegDeleteValue : حذف مقدار یک key

Lib &Private Declare Function RegDeleteValue advapi32.dll" Alias "RegDeleteValueA" (ByVal hKey As " (String Long, ByVal lpValueName As

۳ – توابع کمکی : برای نوشتن توابع اصلی کار با رجیستری نیاز به نوشتن توابع کمکی زیر است :

– تابع SetValueEx : با توجه به نوع داده یک کلید ، مقدار موجود در آنرا در یک متغیر ذخیره می کند :

```
,Public Function SetValueEx(ByVal hKey As Long
sValueName As String, lType As Long, vValue As
Variant) As Long
Long Dim lValue As
Dim sValue As String
Select Case lType
is string Case REG_SZ ' type of value
sValue = vValue
sValueName,  ,SetValueEx = RegSetValueExString(hKey
lType, sValue, Len(sValue))x ,&0
value is Double word Case REG_DWORD ' type of
lValue = vValue
sValueName,  ,SetValueEx = RegSetValueExLong(hKey
lType, lValue, 4)x ,&0
End Select
End Function
```

– تابع QueryValueEx : سایز و نوع داده ای یک داده را که باید خوانده شود مشخص می کند .

```
Function QueryValueEx(ByVal lhKey As Long, ByVal
szValueName As String, vValue As Variant) As Long
Dim cch As Long
```

```
As Long Dim lrc
Dim lType As Long
Dim lValue As Long
String Dim sValue As
,&lrc = RegQueryValueExNULL(lhKey, szValueName, 0
cch)x ,&lType, 0
Select Case lType
For strings '
:Case REG_SZ
String(cch, 0)x = sValue
,&lrc = RegQueryValueExString(lhKey, szValueName, 0
lType, sValue, cch)x
If lrc = ERROR_NONE Then
cch)x ,vValue = Left$(sValue
Else
vValue = Empty
End If
For DWORDS '
:REG_DWORD Case
,&lrc = RegQueryValueExLong(lhKey, szValueName, 0
lValue, cch)x ,lType
If lrc = ERROR_NONE Then vValue = lValue
Case Else
other data types not supported all'
lrc = -1
Select End
:QueryValueExExit
QueryValueEx = lrc
Function Exit
:QueryValueExError
Resume QueryValueExExit
Function End
```

۴ – توابع اصلی : توابع مربوط به پاک کردن یک کلید از رجیستری ، ساخت یک کلید جدید در رجیستری و مقداردهی به یک کلید :

– تابع DeleteKey : این تابع یک کلید از رجیستری را حذف می کند . دارای دو پارامتر ورودی است :
Location که یکی از مقادیر HKEY_CLASSES_ROOT، HKEY_CURRENT_USER ، HKEY_LOCAL_MACHINE و یا HKEY_USERS است .
KeyName که نام کلیدی است که باید از رجیستری حذف شود . این کلید ممکنست شامل subkey هایی نیز باشد مثلاً Key1\SubKey1

```
,Public Function DeleteKey(lPredefinedKey As Long
sKeyName As String)x
Dim lRetVal As Long
RegDeleteKey(lPredefinedKey, sKeyName)x = lRetVal
value DeleteKey = lRetVal ' return function
End Function
```

– تابع DeleteValue : این تابع یک entry را از کلید حذف می کند . دارای سه پارامتر ورودی است : Location، KeyName و ValueName که نام آن value را مشخص می کند .

```
,Public Function DeleteValue(lPredefinedKey As Long
x(sKeyName As String, sValueName As String
Dim lRetVal As Long
As Long Dim hKey
lRetVal = RegOpenKeyEx(lPredefinedKey, sKeyName, 0,
```

```
hKey)x ,KEY_ALL_ACCESS
lRetVal = RegDeleteValue(hKey, sValueName)x
hKey)x) RegCloseKey
DeleteValue = lRetVal
End Function
```

- تابع CreateNewKey : این تابع یک کلید جدید ایجاد می کند . دارای دو پارامتر ورودی است : Location و KeyName

```
,Public Function CreateNewKey(lPredefinedKey As Long
sNewKeyName As String)x
Dim hNewKey As Long
Dim lRetVal As Long
lRetVal = RegCreateKeyEx(lPredefinedKey,
,vbNullString ,&sNewKeyName, 0
REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS,
lRetVal)x ,hNewKey ,&0
RegCloseKey (hNewKey)x
CreateNewKey = lRetVal
Function End
```

- تابع SetKeyValue : این تابع پارامتر data یک entry را تنظیم می کند . دارای ۵ پارامتر ورودی است : Location، KeyName، ValueName، ValueSetting و ValueType

```
,Public Function SetKeyValue(lPredefinedKey As Long
sKeyName As String, sValueName As String,
As Long)x vValueSetting As Variant, lValueType
Dim lRetVal As Long
Dim hKey As Long
```

```
RegOpenKeyEx(lPredefinedKey, sKeyName, 0,  = lRetVal
KEY_ALL_ACCESS, hKey)x
SetValueEx(hKey, sValueName, lValueType,  = lRetVal
vValueSetting)x
hKey)x) RegCloseKey
SetKeyValue = lRetVal
End Function
```

- تابع QueryValue : این تابع فیلد داده یک entry را برمی گرداند .
دارای سه پارامتر ورودی است : Location ، KeyName و
ValueName

```
,Public Function QueryValue(lPredefinedKey As Long
sKeyName As String, sValueName As String)x
Dim lRetVal As Long
As Long Dim hKey
Dim vValue As Variant
sKeyName, 0,  ,lRetVal = RegOpenKeyEx(lPredefinedKey
KEY_ALL_ACCESS, hKey)x
vValue)x ,lRetVal = QueryValueEx(hKey, sValueName
QueryValue = vValue
RegCloseKey (hKey)x
End
Function
```

کار با رجیستری در ویژوال بیسیک - قسمت سوم

ساخت یک انتصاب فایل یا File Association به یک برنامه

در این درس می خواهم با استفاده از کلاسی که در درس قبل معرفی شد تابعی بسازیم که توسط آن بتوانیم فایلهای با پسوندی مشخص را به یک برنامه اختصاص دهیم . بعبارت دیگر تابعی بنویسیم که اطلاعات لازم برای باز شدن فایلهایی با پسوند xxx را توسط برنامه MyApp.exe در رجیستری ثبت کند .

```
,Public Sub CreateAssociation(sExtension As String
sApplication As String, sAppPath As String)x
String Dim sPath, sAppExe As
sExtension,  & "." CreateNewKey
HKEY_CLASSES_ROOT
sExtension,  & "." ,HKEY_CLASSES_ROOT SetKeyValue
REG_SZ ,"Document." & "", sApplication
& CreateNewKey sApplication
,"Document\shell\open\command."
HKEY_CLASSES_ROOT
& SetKeyValue HKEY_CLASSES_ROOT, sApplication
Document", REG_SZ " & Document", "", sApplication."
x"١٪ " & sPath = sAppPath
exe"x." & sAppExe = sApplication
&HKEY_CLASSES_ROOT, sApplication SetKeyValue
REG_SZ ,Document\shell\open\command", "", sPath."
CreateNewKey
"
Software\Microsoft\Windows\CurrentVersion\Explorer\Fi
sExtension, HKEY_CURRENT_USER & ".\leExts
,SetKeyValue HKEY_CURRENT_USER
```

```
"
Software\Microsoft\Windows\CurrentVersion\Explorer\Fi
sExtension, "Application", sAppExe, REG_SZ & ".\leExts
& sAppExe & "\CreateNewKey "Applications
shell\open\command", HKEY_CLASSES_ROOT\"
"\HKEY_CLASSES_ROOT, "Applications SetKeyValue
sPath, REG_SZ ,"" ,"shell\open\command\" & sAppExe &
End Sub
```

کاربرد این تابع بصورت زیر است :

```
CreateAssociation("xxx","MyApp","c:\MyApp.exe")x
```

کار با رجیستری در ویژوال بیسیک – قسمت چهارم

اجرا شدن یک برنامه در هنگام راه اندازی سیستم

فرض کنید می خواهیم برنامه ای بنویسیم که هر بار در هنگام راه اندازی سیستم بطور خودکار اجرا شود. البته نمی خواهم در startup ویندوز دیده شود .
برای این کار باید برنامه موردنظر را در StartUp رجیستری قرار دهیم . به این ترتیب که در یکی از کلیدهای زیر یک مقدار رشته ای جدید(String Value) ایجاد کنیم و آدرس برنامه را در آن وارد کنیم :

```
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windo
ws\CurrentVersion\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Win
dows\CurrentVersion\Run
```

برای مثال اگه اسم برنامه مورد نظر MyApp و مسیرش C:\Windows\MyApp.exe است باید بصورت زیر عمل کرد :

,SetKeyValue HKEY_LOCAL_MACHINE SOFTWARE\Microsoft\Windows\CurrentVersion\Run", " REG_SZ ,""MyApp", "C:\MyApp.exe

نکته : البته دو تا راه دیگر برای اینکار وجود دارد که برخی تروجان ها هم از این روشها استفاده می کنند تا روی سیستم باقی بمانند : یکی استفاده از win.ini و نوشتن نام فایل جلوی = run و دیگری استفاده از system.ini و نوشتن نام برنامه جلوی خط explorer.exe .

برنامه نویسی سخت افزار در ویژوال بیسیک

برنامه نویسی سخت افزار در ویژوال بیسیک

مقدمه

شاید تابحال خواسته باشید از طریق پورت پرینتر و یا پورت سریال یا باس ISA با سخت افزاری که خودتان ساخته اید ارتباط برقرار کنید . برای این کار شما نیاز به برنامه ای دارید که بوسیله دستورات I/O با سخت افزار شما ارتباط برقرار کند . چون ویژوال بیسیک دارای دستورات کار با پورتهای I/O نیست بایستی توسط ویژوال سی یک Dll برای کار با این پورتها بسازیم .

DLL و کاربردهای آن

اگر شما در DOS برنامه نویسی کرده باشید با دستورات INPUT و OUPUT در زبان QuickBasic و با دستورات inp و outp در C آشنا هستید . این توابع در VB پیاده سازی نشده اند . این توابع برای سازندگان سخت افزار برای PC و نیز برنامه نویسان سیستم حیاتی هستند زیرا به شما اجازه خواندن از پورت و نوشتن به پورت را می دهند . برای ایجاد امکان استفاده ار پورت در VB باید از Dll یا کتابخانه های دینامیک استفاده کنیم . Dll ها به VB اجازه می دهند تا یک کد نوشته شده به یک زبان دیگر را در زمان اجرا (بطور دینامیک) به برنامه لینک شود . چون ویژوال سی دارای توابع خواندن و نوشتن پورت است بنابراین ما از این زبان برای ساخت Dll موردنظر استفاده می کنیم .

نوشتن DLL در VC

برای ساخت Dll ما بایستی دو فایل را ایجاد کنیم . اولین فایل یک فایل define یا DEF است و دومین فایل ، یک فایل C++ source یا CPP می باشد . هر دو فایل را می توانید توسط Notepad ایجاد کنید . لیست کد این دو فایل بصورت زیر است :

: Port.def file

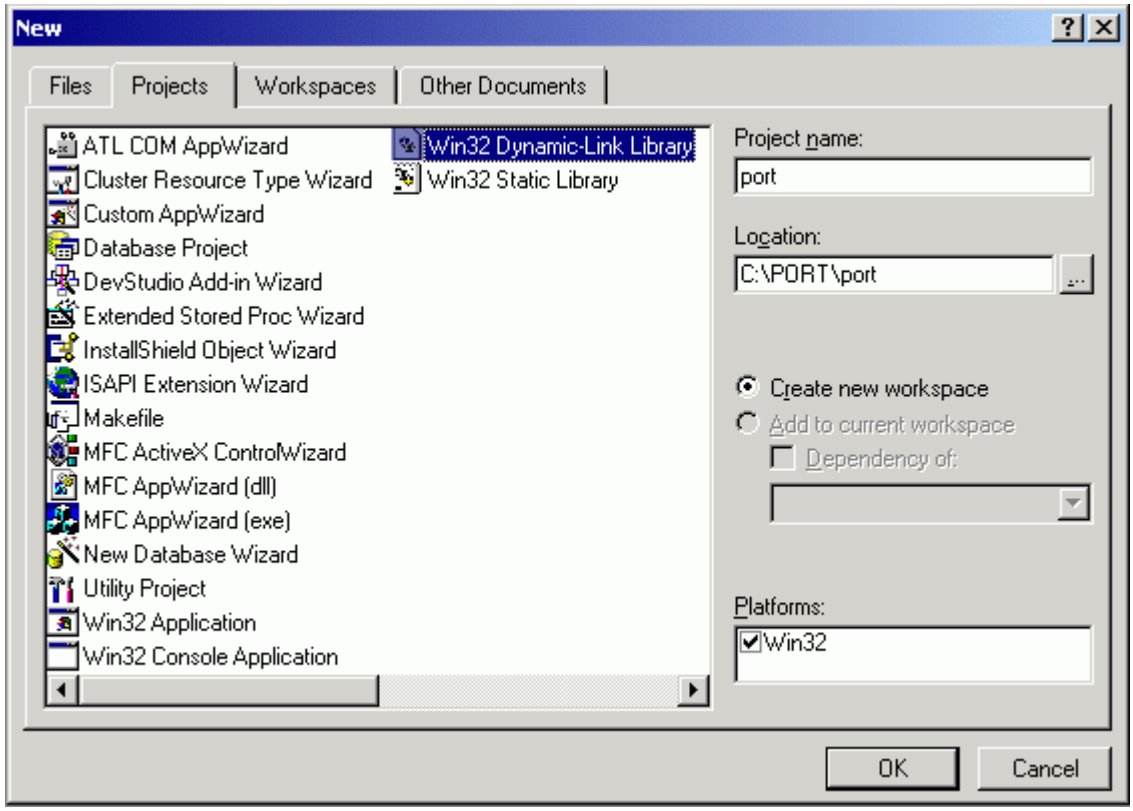
```
LIBRARY Port
DLL FOR I/O PORT DESCRIPTION
EXPORTS
Outp @1
Inp @2
```

: Port.cpp file

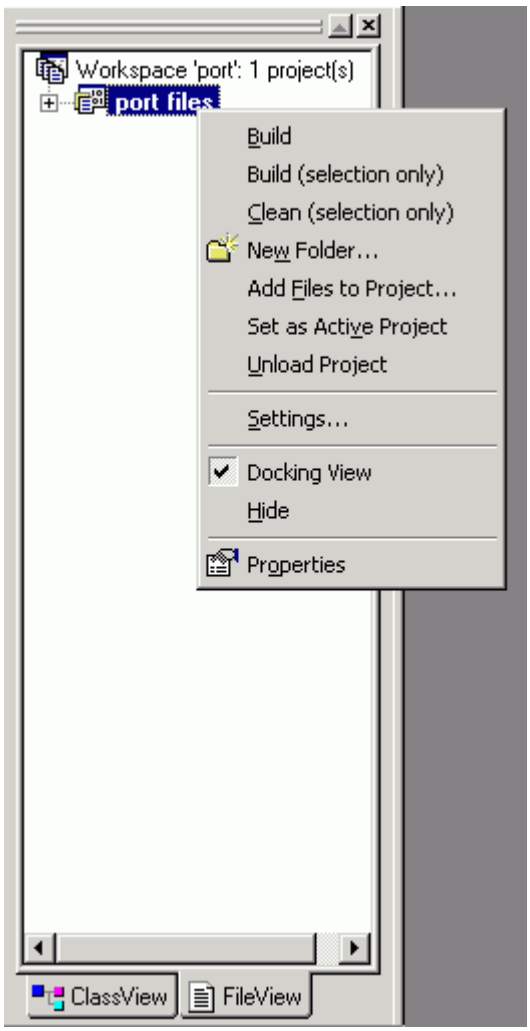
```
#include "conio.h"x
PortData)x short _stdcall Outp(int PortAddress,int
{
Dummy;x short
Dummy=(short)(_outp(PortAddress,PortData));x
return(Dummy);x
};
stdcall Inp(int PortAddress)x_ short
{
PortData;x short
PortData=(short)(_inp(PortAddress));x
return(PortData);x
};
```

پس از نوشتن این دو فایل آنها را در یک دایرکتوری قرار دهید (مثلاً C:\port) و سپس وارد محیط ویژوال سی شوید . از منوی File مورد New را انتخاب کنید و در کادری که ظاهر می شود در قسمت Projects مورد Win32 Dynamic-Link Library را انتخاب کنید . همچنین در سمت راست همان صفحه در کادر Project name عبارت port را وارد کنید و در کادر Location عبارت C:\port\port را وارد کنید و آنگاه OK را کلیک کنید .

!Error

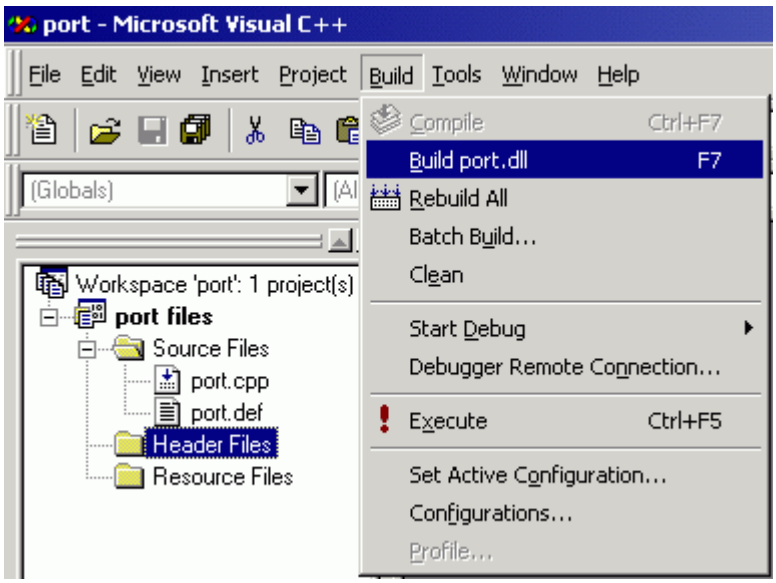


در مرحله بعدی بایستی فایل های def و cpp را به workspace ایجاد شده اضافه کنیم . برای اینکار در برگه FileView روی عبارت Port files کلیک راست کرده و مورد Project Add Files to را انتخاب کنید .



در کادری که ظاهر می شود فایل port.cpp را انتخاب کنید . با همین روش فایل port.def را نیز به پروژه اضافه نمائید .

حال وارد منوی Build شده مورد Build port.dll را انتخاب کنید تا dll مورد نظر ساخته شود .



dll ساخته شده را در دایرکتوری system ویندوزتان کپی کنید .

استفاده از Dll در ویژوال بیسیک

برای استفاده از توابع dll ساخته شده باید ابتدا توابع Out و In را declare کنید :

```
Private Declare Function Outp Lib “port.dll” (Byval
PortAddress as Integer,Byval PortData as Integer) as
Integer
Function Inp Lib “port.dll” (Byval  Private Declare
PortAddress as Integer) as Integer
```

حال در برنامه تان برای نوشتن به یک پورت از دستور زیر استفاده کنید :

```
Dummy=Outp(port_number,data)x
```

و برای خواندن از پورت از دستور زیر استفاده کنید :

Portvalue=Inp(port_number)x

نکته : برای دریافت port.dll و یک برنامه نمونه استفاده از این dll در ویژوال بیسیک ، در بخش نظر خواهی آدرس ایمیل خود را بگذارید .

پورت Parallel و برنامه نویسی آن در ویژوال بیسیک

تذکر : در مقاله های قبلی بخشی تحت عنوان " برنامه نویسی سخت افزار در ویژوال بیسیک " نوشتم که در آن روشی برای دسترسی به پورتها از طریق ویژوال بیسیک ذکر شد . لازم به تذکر است که این روش تنها در ویندوزهای ۹۵ و ۹۸ امکان پذیر است .

مبانی پورت موازی (Parallel)

پورت پارالل استاندارد کامپیوتر یک درگاه ۲۵ پینی است که ۱۲ پین آن خروجی می باشد . از این ۱۲ خط ، ۸ خط بعنوان داده خروجی (DATA Port) و ۴ خط بعنوان خط کنترل (CONTROL Port) می باشند . ۵ پین نیز ورودی بوده و بعنوان خطوط وضعیت (STATUS Port) استفاده می شوند .
۸ پین باقیمانده نیز زمین (GROUND) هستند .

خطوط پورت پارالل توسط سه آدرس I/O که هرکدام متناظر با یکی از سه پورت داده ، کنترل و و ضعیت است قابل دسترسی می باشند .
آدرس پایه پورت پارالل در اکثر کامپیوترها x378۰ می باشد (LPT 1) . بنابراین آدرس I/O برای پورت داده برابر x378۰ ، برای پورت status برابر x379۰ و برای پورت command برابر x37A۰ می باشد .

پورت پارالل استاندارد دارای دو حالت توسعه یافته به نامهای ECP و EPP نیز می باشد .

بوسیله پورت پارالل می توانید مدارهای جانبی خود را به کامپیوتر متصل کنید . تنها نکته ای که در این بین وجود دارد چگونگی برنامه نویسی پورت پارالل و در نتیجه برقراری ارتباط با مدار جانبی است . برای اطلاعات بیشتر در این زمینه با من تماس بگیرید .

برنامه نویسی پورت پارالل در محیطهای مختلف ویندوز

استفاده از کتابخانه Inpout32.dll : با استفاده از این dll می توان به پورتهای سیستم در محیطهای Win 9x/NT/2000/XP دسترسی داشت .

برای دریافت این dll به این آدرس مراجعه کنید . برای استفاده از این dll کافی است آنرا در دایرکتوری System32 ویندوزتان کپی کنید . سپس یک ماژوال به پروژه تان اضافه کرده و عبارت زیرا را در آن قرار دهید :

```
Public Declare Function Inp Lib "inpout32.dll" Alias
Inp32" (ByVal PortAddress As Integer) As Integer"
inpout32.dll" Alias "Out32" " Public Declare Sub Out Lib
Integer)x (ByVal PortAddress As Integer, ByVal Value As
```

حال برای مثال با دستور زیر می توانید اطلاعاتی را روی پینهای DATA ی مربوط به پورت پارالل بنویسید :

```
H378, your_data)x&)Call Out
```

آشنایی با شی پرینتر در ویژوال بیسیک

آشنایی با شی پرینتر در ویژوال بیسیک ۶

مقدمه

شی پرینتر ، شیی است که پرینتر پیش فرض سیستم را کنترل می کند . استفاده از شی پرینتر در ویژوال بیسیک ۶ مانند کار با سایر اشیا است و بایستی از خواص و متدهای آن استفاده کرد . در ادامه با برخی از این خواص و متدها آشنا خواهید شد .

چاپ متن توسط شی پرینتر

برای چاپ متن توسط شی پرینتر کافیست خواص CurrentX و CurrentY که محل قرار گرفتن کرسر می باشد را تنظیم نوده و سپس با استفاده از متد Print متن مورد نظر را چاپ نموده و در پایان با استفاده از متد EndDoc صفحه چاپی را از پرینتر بیرون بدهیم . مثال :

```
Printer.CurrentX=150
Printer.CurrentY=200
Printer.Print "Visual Basic Printer Object Test"
Printer.EndDoc
```

در مثال فوق فرض شده که ScaleMode برابر Pixel قرار داده شده است . توجه داشته باشید که تا قبل از اجرای متد EndDoc عمل چاپ

انجام نمی شود و فقط بعد از این متد است که چاپ انجام شده و کاغذ بیرون می آید .

اگر پس از یک دستور Print ، دستور Print دیگری را استفاده کنیم متن روی خط بعدی چاپ خواهد شد . اگر بخواهیم متن بلافاصله بعد از متن اول چاپ شود باید بعد از دستور Print اول از علامت ; استفاده کنیم .

نکته : برای کنترل دقیق محل چاپ از CurrentX و CurrentY استفاده نمائید .

چاپ گرافیک توسط شی پرینتر

به ۴ روش می توان اشکال گرافیکی را توسط شی پرینتر چاپ کنید :
۱ – چاپ دایره : با استفاده از متد Circle می توان یک دایره ، قوس و یا بیضی را در صفحه چاپ کرد . فرمت کلی این متد بصورت زیر است :

Circle (x,y),radius,[color],[start],[end],[aspect]

که x و y مختصات مرکز دایره و radius شعاع آن می باشد .
پارامترهای color، start، end و aspect اختیاری هستند و بترتیب رنگ ، محل شروع قوس ، محل خاتمه قوس و نسبت شعاع بیضی را نشان می دهند .

۲ – چاپ خط : با استفاده از متد Line می توان یک خط و مستطیل را در صفحه چاپ کرد . فرمت کلی این متد بصورت زیر است :

Line (x1,y1)-(x2,y2),[color],[B[F]]

که x1 و y1 مختصات شروع خط (یا مستطیل) و x2 و y2 مختصات انتهای خط (یا مستطیل) هستند .

پارامتر color اختیاری بوده و رنگ خط (یا مستتطیل) را نشان می دهد .

پارامتر B اختیاری بوده و نشان می دهد یک مستیل رسم شود .
پارامتر F اختیاری بوده و بهمراه B می آید و نشان می دهد یک مستطیل توپر رسم شود .

۳ – چاپ نقطه : با استفاده از متد PSet می توان نقطه ای روی صفحه چاپ کرد و فرمت کلی آن بصورت زیر است :

```
PSet (x,y),[color]
```

که x و y مختصات نقطه می باشند .
پارامتر color اختیاری بوده و رنگ نقطه را نشان می دهد .

۴ – چاپ تصویر : با استفاده از متد PaintPicture می توان محتویات یک فایل گرافیکی را چاپ کرد . فرمت کلی این متد بصورت زیر است :

```
Printer.PaintPicture picture, x1, y1, [width1], [height1],
[x2], [y2], [width2], [height2], [opcode]
```

x1 و y1 مختصات قرارگرفتن تصویر در صفحه بوده و picture یک شی از کلاس IPictureDisp است . این شی را می توان از یک PictureBox یا از خاصیت Picture فرم گرفت و یا از دستور LoadPicture استفاده کرد .

مثال ۱ :

```
Printer.PaintPicture Picture1.Picture, 100, 100
```

مثال ۲ :

PaintPicture LoadPicture("C:\sample.jpg"), 100, 100

width1 و height1 طول و عرض تصویر چاپی می باشند . x2 و y2 نیز بهمراه width2 و height2 می توانند میزان برش از تصویر اصلی برای چاپ را مشخص کنند .

سایر خواص مهم شی پرینتر

ColorMode : اگر پرینتر رنگی باشد ، رنگی یا تک رنگ بودن چاپ را تعیین می کند .
Copies : تعداد چاپ را مشخص می کند .
Font : نوع فونت چاپ متن را مشخص می کند .
FontSize : سایز فونت چاپ متن را مشخص می کند .
PrintQuality : کیفیت چاپ را مشخص می کند .

سایر متدهای مهم شی پرینتر

KillDoc : پرینت در حال چاپ را از صف چاپ حذف می کند .
NewPage : صفحه جاری را به پایان برده و صفحه جدیدی را برای چاپ آماده می کند .
Scale : سیستم مختصات کاربر را تعیین می کند .
TextHeight : ارتفاع متن پس از چاپ شدن در مختصات Scale را تعیین می کند .
TextWidth : عرض متن پس از چاپ شدن در مختصات Scale را تعیین می کند .

کار با اسکنر در وی بی

+ در بخش پرسش و پاسخ شماره ۵ سوالی در مورد کار با اسکنر و دریافت تصویر از آن در ویژوال بیسیک عنوان شد . همان طور که گفتم بایستی از کتابخانه هایی که برای این منظور ارائه شده اند استفاده کنید . یکی از این کتابخانه ها EZ Twain می باشد . برای دریافت dll مربوط به این کتابخانه و نیز دریافت یک برنامه نمونه کار با این dll در ویژوال بیسیک این لینک را کلیک کنید .

برنامه نویسی شبکه و اینترنت در VB

برنامه نویسی شبکه و اینترنت در VB بخش اول

مروری بر TCP/IP

نکته : مطالب زیر تنها در حد یک یادآوری می باشد . اگر اطلاعات کمی در مورد TCP/IP دارید به کتابهای موجود مراجعه کنید .

پروتکل Protocol : قراردادی است برای برقراری ارتباط در شبکه

مدل TCP/IP : مجموعه ای از پروتکلهای ارتباطی مرتبط بهم است که مکانیزمها و سرویسهای مورد نیاز جهت برقراری ارتباط در اینترنت را مهیا می کنند . این مدل شامل ۴ لایه است :

۱ – لایه کاربرد Application Layer : شامل برنامه های کاربردی و پروتکلهایی مثل Http، Ftp، Smtp، Pop و Telnet می باشد .

۲ – لایه انتقال Transport Layer : این لایه شامل دو پروتکل TCP و UDP است . پروتکل TCP وظیفه کنترل رسیدن بسته های داده به مقصد (TCP/IP داده ها را به بسته های کوچکی تقسیم می کند که هر بسته حاوی آدرس فرستنده ، گیرنده و شماره بسته می باشد) ، تصحیح خطا و مرتب سازی بسته ها را برعهده دارد . UDP پروتکلی شبیه TCP است با این تفاوت که هیچ ضمانتی برای رسیدن بسته های اطلاعاتی در آن وجود ندارد و معمولاً در انتقال صوت و ویدئو روی اینتزنت استفاده می شود .

۳ – لایه اینترنت Internet Layer : شامل پروتکل IP است که مسئول مسیریابی بسته های اطلاعاتی می باشد .

۴ – لایه دسترسی به شبکه Link Layer : شامل بخشی از هسته سیستم عامل و نیز درایورهای واسط شبکه برای کار با سخت افزار شبکه می باشد .

سوکت Socket و پورت Port : سوکت یک ورودی انتزاعی در لایه انتقال می باشد که برای ایجاد ارتباطات مختلف TCP/IP بکار می رود . اغلب برنامه های کاربردی که از TCP و UDP استفاده می کنند ، عملیات انتقال اطلاعات خود را با ساخت یک سوکت و سپس انجام یکسری عملیات روی آن انجام می دهند . این عملیات عبارتند از :

۱ – عملیات کنترلی : شامل اختصاص یک شماره پورت به سوکت ، initiate کردن یا accpet کردن یک ارتباط ، از بین بردن سوکت

۲ – عملیات انتقال داده : شامل نوشتن داده روی سوکت و خواندن داده از سوکت

۳ – عملیات بررسی وضعیت : مثل پیدا کردن آدرس IP مربوط به سوکت ، پیدا کردن شماره پورت سوکت و غیره

HTTP : پروتکل انتقال داده برای وب است .

FTP : پروتکل انتقال فایل روی اینترنت است .

۱ – فایل Region.dll که در مطالب قبلی معرفی شد را می توانید از اینجا بگیرید .

TAPI در ویژوال بیسیک

TAPI در ویژوال بیسیک – مقدمه

TAPI چیست ؟

TAPI یا Telephony API یک کتابخانه استاندارد برای کار با مودم و نوشتن برنامه های تلفنی می باشد . برای نمونه می توان از برنامه های Phone Dialer (شماره گیر تلفن) ، برنامه شبکه سازی تلفنی (Dialup Networking) ، برنامه تشخیص پالس مودم برای ضبط اطلاعات وارد شده از طرف کاربران و کاربردهای دیگر در این زمینه نام برد . این کتابخانه به شما کمک می کند تا بدون درگیر شدن با برنامه نویسی سخت افزار مودم و درایور آن بطور مستقیم بتوانید برنامه های کاربردیی در این زمینه بنویسید .

در این سلسه مقالات سعی می کنم تا مفاهیم TAPI و چگونگی استفاده از آنرا در ویژوال بیسیک آموزش بدهم

TAPI در ویژوال بیسیک - بخش اول

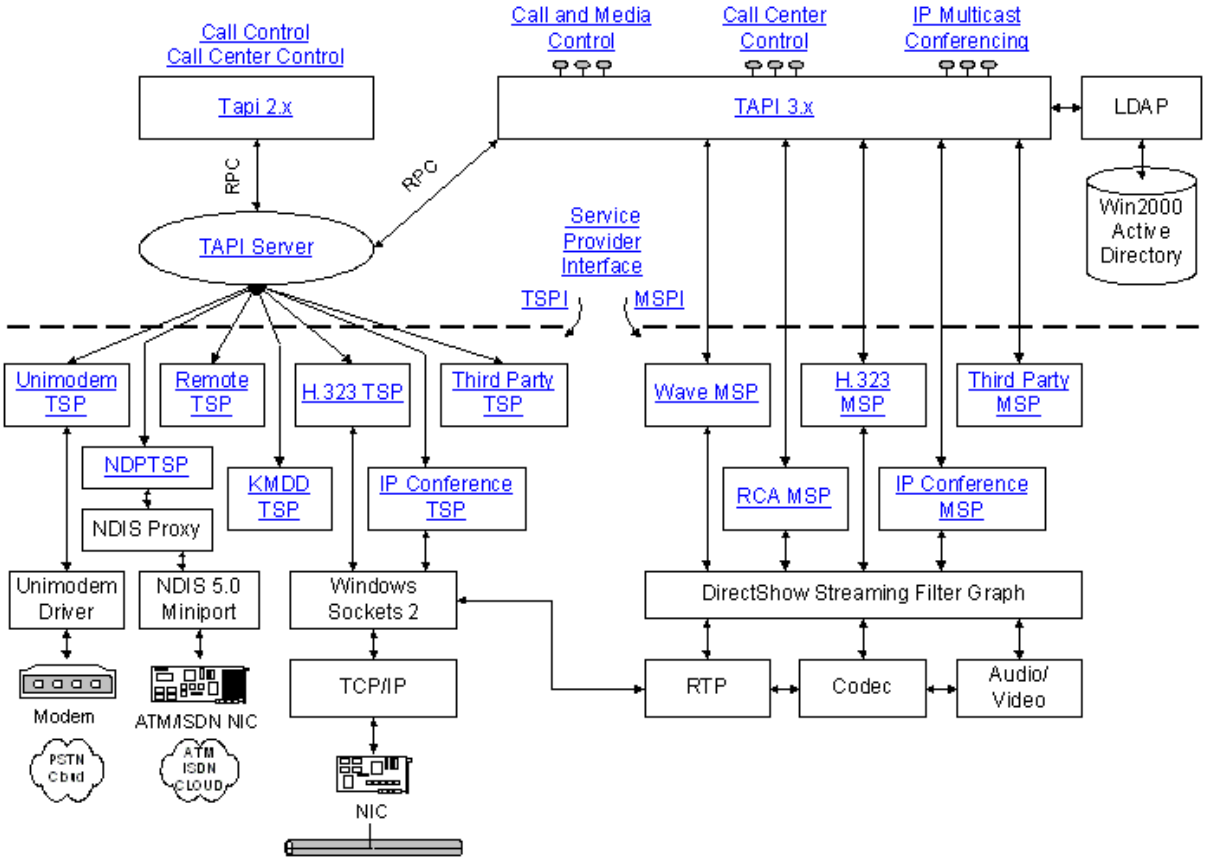
مروری بر Microsoft Telephony :

Telephony امکان مجتمع سازی کامپیوترها با دستگاههای ارتباطی و شبکه ها را فراهم نموده است . معمولاً دستگاه ارتباطی یک مودم و خط ارتباطی نیز شبکه PSTN (شبکه عمومی تلفن سوئیچینگ) می باشد . برخی از کاربردهای Telephony عبارتند از :

۱ - کنفرانسهای مالتی مدیا بصورت Multicast
۲ - VoIP
۳ - مرکز پاسخ گویی اتوماتیک
۴ - تماس تلفنی از طریق کامپیوتر روی شبکه PSTN

دیاگرام زیر معماری Microsoft Telephony را نشان می دهد :

!Error



برنامه های TAPI :

برای نوشتن برنامه های کاربردی با استفاده از TAPI بایستی ابتدا در مورد سطح سرویسی که می خواهیم ارائه دهیم تصمیم گیری کنیم . برای مثال برای نوشتن یک برنامه شماره گیر تلفن نیاز به استفاده کامل از TAPI نیست و می توان از قابلیتهای خود ویندوز در این زمینه استفاده کرد (Assisted Telephony) . در بخشهای بعدی در مورد سطوح مختلف سرویس در TAPI بیشتر صحبت خواهم کرد .

دومین مطلبی که باید مورد توجه قرار داد اینست که می خواهیم از TAPI 2.x استفاده کنیم یا از TAPI 3.x . تفاوت ایندو آنست که TAPI ورژن ۲ یک API برمبنای C است در حالیکه ورژن ۳ آن بر مبنای تکنولوژی

COM می باشد . در بخشهای بعدی مطالب بیشتری در مورد تفاوتهای این دو نسخه بیان خواهم کرد .
بخشهای اصلی یک برنامه کامل TAPI عبارتند از :

۱ – TAPI Initialization : شامل load کردن TAPI dll ، اتصال به TAPI Server ، مذاکره در مورد ورژن TAPI و برپاسازی سیستم اطلاع رسانی event می باشد .

۲ – Session Control : مقداردهی اولیه ، دریافت و کنترل تماسها

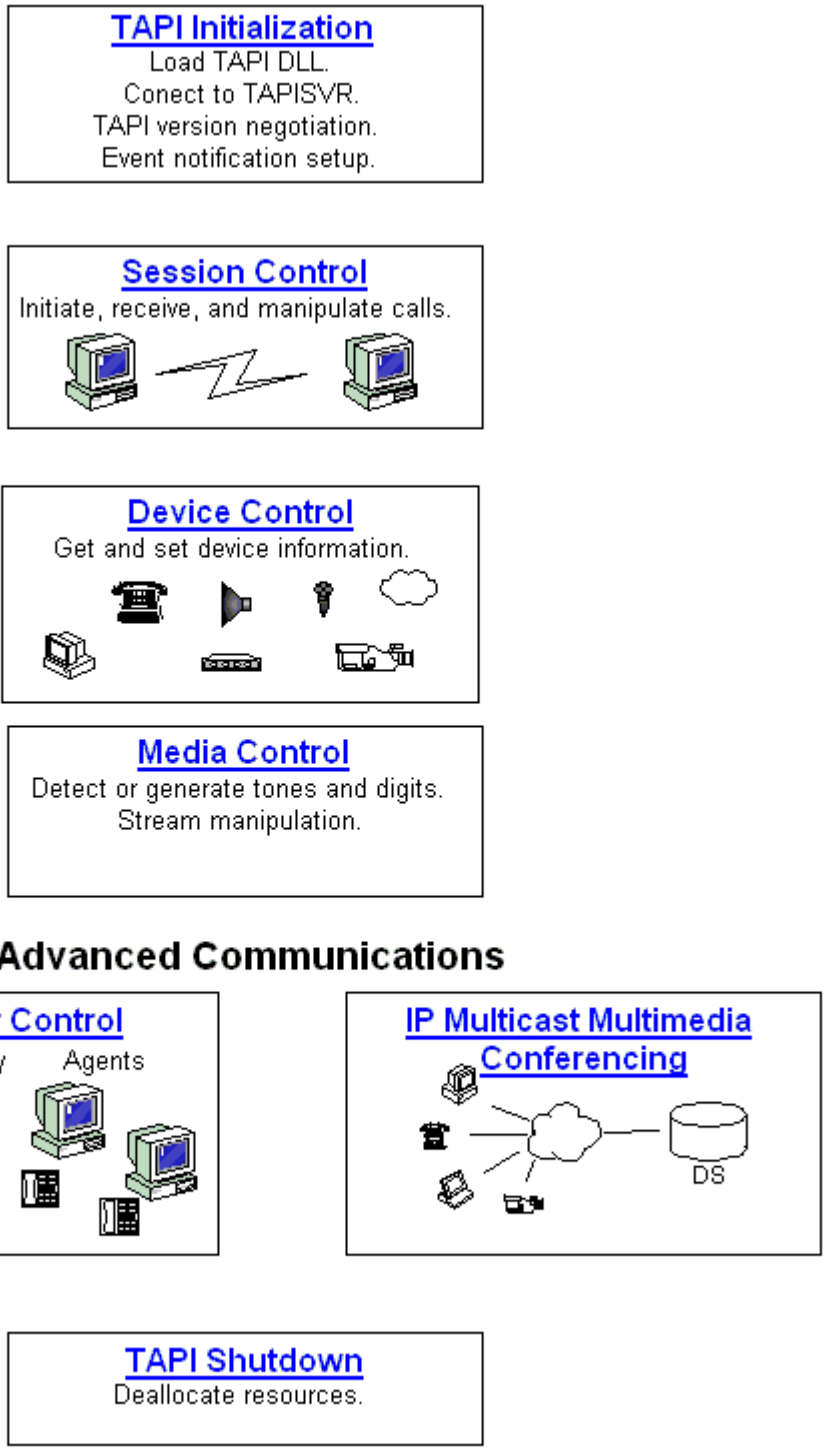
۳ – Device Control : دریافت و تنظیم اطلاعات دستگاه

۴ – Media Control : تشخیص و یا تولید تونها و ارقام ، کنترل stream

۵ – TAPI Shutdown : آزاد سازی منابع

دیاگرام زیر این مراحل را بهتر نشان می دهد :

!



TAPI در ویژوال بیسیک - بخش دوم

مقداردهی اولیه TAPI :

عملکرد درست اجزای TAPI نیاز به برپاسازی محیط ارتباطی روی کامپیوتر مورد نظر دارد . مراحل این امر عبارتند از :

۱ – نصب TAPI : زمانیکه سخت افزار و یا نرم افزار برای اولین بار به کامپیوتر اضافه می شود انجام می گیرد . جزئیات کار به سیستم عامل و نرم افزار بستگی دارد .

۲ – مقداردهی ابتدائی : ساخت اشیا و مسیرهای ارتباطی

۳ – مذاکره در مورد ورژن TAPI : برای اطمینان از اینکه اجزای TAPI قادر به تبادل داده ها باشند .

۴ – استخراج اطلاعات منابع : بدست آوردن اطلاعاتی در مورد دستگاهی که می توان از آن در برنامه TAPI مورد نظرمان استفاده نمود .

۵ – Event notification : برپاسازی سیستم اطلاع رسانی event

TAPI در ویژوال بیسیک – بخش سوم

مقداردهی اولیه TAPI در ویژوال بیسیک :

از منوی Project گزینه References را انتخاب کرده و از لیست مربوطه مورد Microsoft TAPI 3.0 Type Library را انتخاب کنید .

حال وارد بخش کد نویسی فرمتان شوید و متغیر objTAPI را بصورت زیر تعریف کنید :

```
Dim objTapi As TAPI
```

سپس در بخش مربوط به Load Form شی objTAPI را بصورت زیر ایجاد می کنیم :

```
Set objTapi = New TAPI
```

همانطور که در بخشهای قبلی گفته شد ، قبل از فراخوانی هر تابع TAPI ابتدا بایستی آنرا مقداردهی اولیه کنیم . برای مقداردهی اولیه کردن شی TAPI عبارت زیر را بنویسید :

```
Call objTapi.Initialize
```

TAPI در ویژوال بیسیک - بخش چهارم

انتخاب یک آدرس :

کد زیر نشان می دهد که چگونه می توان با استفاده از شی TAPI در ویژوال بیسیک منابع تلفنی در دسترس را برای یک آدرس که بتواند یک مجموعه مشخص از نیازها را مدیریت کند ، بررسی کرد .

توجه داشته باشید که قبل از انجام این کار بایستی عمل مقداردهی اولیه TAPI را که در بخش قبل رررسی شد ، انجام دهید .

نکته : در کد زیر عمل error checking انجام نگرفته است و برای استفاده از کد زیر در برنامه های واقعی بایستی بخش بررسی خطا را به

آن اضافه کنید .

۱ – تعریف یک شی آدرس و یک شی مجموعه آدرس :

```
Dim gobjAddress As ITAddress
As ITCollection Dim objCollAddresses
```

۲ – تنظیم شی objCollAddress بعنوان یک مجموعه آدرس از شی objTapi :

```
Set objCollAddresses = objTapi.Addresses
```

۳ – پیدا کردن آدرسی که بتواند از واسط مورد نظر ما پشتیبانی کند :

```
bFound = False
objCollAddresses.Count For indexAddr = 1 To
objCollAddresses.Item(indexAddr)x = Set objCrtAddress
Set objMediaSupport = objCrtAddress
objAddressCapabilities = objCrtAddress Set

nSelectedType ) x )If objMediaSupport.QueryMediaType
bFound = True
End If

Nothing = Set objAddressCapabilities
Set objMediaSupport = Nothing
Nothing = Set objCrtAddress

If bFound = True Then Exit For
Next indexAddr
```

در صورتیکه آدرس مورد نظر پیدا شود برنامه از حلقه خارج شده و gobjAddress یک آدرس قابل استفاده خواهد بود :

objcollAddresses.Item(indexAddr)x = Set gobjAddress

TAPI در ویژوال بیسیک - بخش پنجم

انجام Event Handling در TAPI :

کد زیر شامل یک event handler ساده برای TAPI ، رجیستر کردن واسط event ، تنظیم فیلتر event و رجیستر کردن تمام فراخوانیهای دادن اخطار است . هدف اصلی از این کد اینست که مطمئن شویم بخشی از TAPI که event ها را دریافت می کند پردازشی را قبل از انتقال به بخشهای دیگر انجام دهد .

تعاریفها :

```
TAPI Dim WithEvents gobjTapiWithEvents As
Attribute gobjTapiWithEvents.VB_VarHelpID = -1
glRegistrationToken As Long Dim
```

```
Const TAPI3_CALL_EVENTS =TE_CALLMEDIA Or
TE_CALLNOTIFICATION Or TE_CALLSTATE
```

تنظیم eventfilter بصورتیکه تمام event های تعریف شده برای TAPI را بپذیرد :

```
TAPI3_CALL_EVENTS = objTapi.EventFilter
```

رجیستر کردن event ها :

```
Set gobjTapiWithEvents = objTapi
Boolean, fMonitor As Boolean Dim fOwner As
Long Dim lMediaTypes As Long, lCallbackInstance As

fOwner = True
fOwner = True
fMonitor = False
TAPIMEDIATYPE_AUDIO = lMediaTypes
lCallbackInstance = 1

= glRegistrationToken
,gobjTapi.RegisterCallNotifications(gobjAddress,fMonitor
fOwner,lMediaTypes,lCallbackInstance)x
```

TAPI در ویژوال بیسیک - بخش ششم

انتخاب یک ترمینال :

+ قبل از اینکه یک ترمینال را برای برقراری ارتباط انتخاب کنید بایستی TAPI Initialization و عمل انتخاب آدرس را انجام داده باشید .

ابتدا یک متغیر از نوع ITBasicCallControl (واسط کنترل تماس) تعریف می کنیم :

```
Dim objCallControl As ITBasicCallControl
objCallControl = gobjReceivedCallInfo Set
```

سپس یک متغیر از نوع ITTerminalSupport (کوئری از شی آدرس) تعریف می کنیم :

```
Dim objTerminalSupport As ITTerminalSupport
objTerminalSupport = gobjAddress Set
```

سپس متغیر ترمینال را تعریف کرده و توسط شی objTerminalSupport یک ترمینال را برای آن استخراج می کنیم :

```
Dim objTerminal As ITTerminal
= Set objTerminal
objTerminalSupport.GetDefaultStaticTerminal(lMediaTy
pe, dir)x
```

در اینجا دیگر نیازی به شی objTerminalSupport نیست بنابراین آنرا آزاد می کنیم :

```
Set objTerminalSupport = Nothing
```

سپس نیاز به تعریف شی objStreamControl برای کنترل ترمینال است :

```
Dim objStreamControl As ITStreamControl
objStreamControl = objCallControl Set
```

در صورتیکه این شی ایجاد شود ، به ازای استریم های موجود در ITCollection امکان ایجاد ترمینال در یک حلقه for بررسی می شود و ترمینال مناسب انتخاب می گردد :

```
If Not (objStreamControl Is Nothing) Then
objITCollStreams As ITCollection Dim

objStreamControl.Streams = Set objITCollStreams

ITStream Dim nIndex As Long, objCrtStream As

For nIndex = 1 To objITCollStreams.Count
objITCollStreams.Item(nIndex)x = Set objCrtStream
Then If objCrtStream.MediaType = lMediaType
If objCrtStream.Direction = dir Then
objCrtStream.SelectTerminal(objTerminal)x Call
End If
End If
objCrtStream = Nothing Set
Next nIndex

Nothing = Set objITCollStreams
Set objStreamControl = Nothing
End If
```

ایجاد یک تماس (Make a Call) :

+ قبل از این بخش بایستی مراحل TAPI Initialization و عمل انتخاب آدرس انجام شده باشد .

این بخش برای ایجاد یک شی تماس ، بررسی و مشخص کردن استریمی که با این تماس در ارتباط است ، انتخاب و ایجاد ترمینالهای مناسب و کامل کردن ارتباط استفاده می شود .

قبل TAPI Initialization و عمل انتخاب آدرس و انتخاب ترمینال انجام شده باشد .

در ابتدا با استفاده از متد CreateCall یک شی تماس ساخته می شود :

```
= Set gobjCall
,gobjOrigAddress.CreateCall(strDestAddress
nSelectedType,lMediaTypes)x
```

سپس در اینجا بایستی کدی که در بخش اول این درس برای انتخاب ترمینال نوشته شد آورده شود :

```
{
Select Terminal Code
}
```

سپس بایستی دستور Connect اجرا شود :

```
gobjCall.Connect (False)x
```

False بدین معناست که ارتباط بصورت آسنکرون برقرار می شود .

TAPI در ویژوال بیسیک - بخش پایانی

دریافت یک تماس :

کد زیر برای یافتن و یا ایجاد یک ترمینال مناسب برای دریافت یک تماس بکار می رود . بایستی توجه داشته باشید که قبل از اجرای کد زیر بایستی مراحل مقداردهی اولیه ، انتخاب یک آدرس و رجیسر کردن event ها را انجام دهید . همچنین در کد زیر بایستی مرحله انتخاب ترمینال را نیز انجام دهید . توجه داشته باشید که در کد زیر متغیر pEvent یک اشاره گر برای واسط ITCallNotificationEvent است که توسط TAPI به event Handler داده می شود :

```
If TapiEvent = TE_CALLNOTIFICATION Then
objCallNotificationEvent As ITCallNotificationEvent Dim
objCallNotificationEvent = pEvent Set
ITCallInfo Dim gobjReceivedCallInfo As
Set gobjReceivedCallInfo = objCallNotificationEvent.Call
objCallControl As ITBasicCallControl Dim
gobjReceivedCallInfo = Set objCallControl
objCallControl.Answer
End If
```

+ بخاطر طولانی شدن این سری مطالب و نیز تخصصی بودن آن که باعث می شود مخاطبین کمتری داشته باشد بحث TAPI را به همین جا خاتمه می دهم . اما برای دوستان علاقمند که بخواهند مطالب بیشتری در این زمینه آموخته و نیز مثالهای عملی از نوشتن برنامه های TAPI را در

اختیار داشته باشند لینکهای زیر را معرفی می کنم :

آموزش TAPI در سایت MSDN

سوالات مختلف در مورد TAPI

سوالات مختلف در مورد TAPI

Call Center Active

یک کنترل ActiveX برای استفاده از TAPI

نمونه ای از یک برنامه کامل TAPI در ویژوال بیسیک

TAPI در ویژوال بیسیک

کتابی در مورد برنامه نویسی TAPI در ویژوال بیسیک

+ مبحث بعدی : نوشتن کلاسهای اکتیو ایکس سمت سرور (Server-Side ActiveX Dll) برای استفاده در صفحات ASP

آشنایی با RAS API و WinInet API

آشنایی با RAS API و WinInet API – بخش اول

مقدمه

ویندوز برای برقراری ارتباط با Internet Service Provide- ISP- شما از طریق مودم و خط تلفن در اتصالات dial-up networking ، از سرویسی خاص به اسم RAS (Remote Access Service) استفاده می کند . این سرویس دارای یک واسط برنامه نویسی است که RAS

API نام دارد . این واسط شامل مجموعه ای از توابع است که شما می توانید آنها را در برنامه خود صدا بزنید . RAS API ابزاری بسیار قدرتمند و قابل انعطاف است همچنین بسیار پیچیده می باشد .
خوشبختانه برای استفاده راحتتر ، مایکروسافت تعدادی تابع را در مجموعه ای به اسم WinInet API قرار داده تا بتوان از آنها برای برقراری ارتباط و کنترل اتصال استفاده کرد .

در این مجموعه آموزشی سعی خواهم کرد تا ابتدا در مورد WinInet API و سپس RAS API مطالبی را بیان کنم .

آشنایی با RAS API و WinInet API – بخش دوم

آشنایی با WinInet API :

WinInet API مجموعه ای از توابع است که امکان ایجاد و توسعه برنامه های اینترنتی را بصورتی ساده ، سریع و کارآمد برای برنامه نویسان مهیا می کند . با استفاده از این مجموعه توابع شما می توانید برنامه هایی بنویسید که از منابع اینترنتی با استفاده از پروتکلهایی چون HTTP و FTP استفاده کنند . همچنین WinInet به شما اجازه می دهد تا بتوانید ارتباطی dial-up با یک ISP ایجاد نموده و آنرا کنترل کنید .
مزیت اصلی توابع WinInet آینست که شما نیازی به دانستن ساختار پروتکلهای ارتباطی و نیز برنامه نویسی Socket نخواهید داشت . بعبارت دیگر WinInet یک واسط سطح بالا را برای کار با منابع اینترنتی ارائه می دهد .

امکانات Dial-Up موجود در WinInet :

تا قبل از ارائه اینترنت اکسپلورر ورژن ۴ ، WinInet تنها دارای دو تابع dial-up بود :

تابع InternetAttemptConnect : برای بررسی اینکه آیا یک ارتباط به اینترنت وجود دارد یا نه استفاده می شد . اگر هیچ اتصالی به اینترنت وجود نداشت این برنامه کادر تبادلی dial-up networking را نمایش می داد و کاربر اجازه داشت تا یک اتصال را برای وصل شدن به اینترنت انتخاب کند .

تابع InternetCheckConnection : تابع با استفاده از انجام یک دستور ping به url ای که به تابع داده شده ، بررسی می کرد که آیا ارتباطی به اینترنت وجود دارد یا نه .

این دو تابع دارای محدودیتهای فراوانی بودند . برای مثال تابع اول نمی تواند بطور اتوماتیک اتصال به اینترنت را برقرار کند و تابع دوم نیز نمی تواند هیچ اطلاعاتی در مورد نوع ارتباط به ما بدهد .

IE نسخه ۴ ، تعدادی تابع جدید برای WinInet معرفی کرد که برخی از آنها عبارتند از :

تابع InternetGetConnectedState : اطلاعاتی در مورد نوع ارتباط استفاده شده را بیان می کند . برای مثال این تابع اطلاع می دهد که نوع ارتباط به اینترنت از طریق مودم است یا شبکه LAN و یا از طریق پروکسی .

تابع InternetAutodial : این امکان را فراهم می سازد تا یک ارتباط اینترنتی اتوماتیک از طریق مودم را با استفاده از مدخل اتصال پیش

فرض که کاربر آنرا در dial-up networking مشخص کرده ایجاد کنید .

تابع InternetDial : این تابع کارآمدتر از تابع InternetAutodial است و کادری را نمایش می دهد که کاربر می تواند نوع مدخل مورد نظر خود برای ارتباط تلفنی با اینترنت را انتخاب کند .

تابع InternetAutodialHangup : برای قطع کردن اتصالی مودمی که از طریق تابع InternetAutodial برقرار شده استفاده می شود .

تابع InternetHangUp : برای قطع کردن اتصالی مودمی که از طریق تابع InternetDialبرقرار شده استفاده می شود .

تابع InternetSetDialState : برای تنظیم کردن وضعیت جاری ارتباط اینترنتی استفاده می شود .

در قسمت بعدی این سلسه مباحث جزئیات این توابع را بررسی کرده و نهایتاً برنامه ای کاربردی برای کار با این توابع در ویژوال بیسیک ارائه خواهم داد .

اطلاعات بیشتری در مورد WinInet :

در این بخش ما تنها توابع dial-up موجود در WinInet API را بررسی کردیم اما همانطور که در ابتدا گفته شد WinInet دارای امکانات فراوانی در زمینه کار با اینترنت است . برای آشنایی بیشتر با این امکانات در زیر جداولی ارائه شده که به اختصار امکانات مختلف این مجموعه تابع را نشان می دهد :

توابع Dial-Up :

Description	Name
Retrieves the current state of the Internet connection	InternetGetConnectedState
Initiates an unattended dial-up connection	InternetAutodial
Disconnects a modem connection initiated by	InternetAutodialHangup
Initiates a dial-up connection	InternetDial
Disconnects a modem connection initiated by InternetDial	InternetHangUp
Prompts the user for permission to initiate a dial-up connection to the given URL	InternetGoOnline
Sets the current state of the Internet connection	InternetSetDialState

توابع عمومی اینترنت :

Name	Description
InternetOpen	Initializes the Win32 Internet functions
InternetConnect	Opens an FTP, Gopher, or HTTP session for a given site
InternetCloseHandle	Closes a single Internet handle or a subtree of Internet handles
InternetErrorDlg	Displays a dialog box for the error that is passed to InternetErrorDlg
InternetFindNextFile	Continues a file search started as a result of a previous call to FtpFindFirstFile or GopherFindFirstFile
InternetGetLastResponseInfo	Retrieves the last Win32 Internet function error description or server response on the thread calling this function
InternetLockRequestFile	Allows the user to place a lock on the file being used
InternetQueryDataAvailable	Queries the amount of data available
InternetQueryOption	Queries an Internet option on the specified handle
InternetReadFile	Reads data from a handle opened by the InternetOpenURL, FtpOpenFile, GopherOpenFile, or

HttpOpenRequest function	
Reads data from a handle opened by the InternetOpenURL, FtpOpenFile, GopherOpenFile, or HttpOpenRequest function	InternetReadFileEx
Sets a file position for InternetReadFile	InternetSetFilePointer
Sets an Internet option	InternetSetOption
Sets up a callback function that Win32 Internet functions can call as progress is made during an operation	InternetSetStatusCallback
Placeholder for the application-defined status callback function	InternetStatusCallback
Formats a date and time according to the specified RFC format (as specified in the HTTP version 1.0 specification)	InternetTimeFromSystemTime
Takes an HTTP time/date string and converts it to a SYSTEMTIME structure	InternetTimeToSystemTime
Unlocks a file that was locked using InternetLockRequestFile	InternetUnlockRequestFile
Writes data to an open Internet file	InternetWriteFile
Checks for changes between secure and	InternetConfirmZoneCrossing

nonsecure URLs	

توابع URL :

Name	Description
InternetCanonicalizeUrl	Canonicalizes a URL, which includes converting unsafe characters and spaces into escape sequences.
InternetCombineUrl	Combines a base and relative URL into a single URL. The resultant URL will be canonicalized.
InternetCrackUrl	Cracks a URL into its component parts.
InternetCreateUrl	Creates a URL from its component parts.
InternetOpenUrl	Begins reading a complete FTP, Gopher, or HTTP URL.

توابع FTP :

Name	Description

Creates a new directory on the FTP server	FtpCreateDirectory
Deletes a file stored on the FTP server	FtpDeleteFile
Searches the specified directory of the given FTP session	FtpFindFirstFile
Retrieves the current directory for the given FTP session	FtpGetCurrentDirectory
Retrieves a file from the FTP server and stores it under the specified file name, creating a new local file in the process	FtpGetFile
Stores a file on the FTP server	FtpPutFile
Removes the specified directory on the FTP server	FtpRemoveDirectory
Renames a file stored on the FTP server	FtpRenameFile
Changes to a different working directory on the FTP server	FtpSetCurrentDirectory

توابع HTTP :

Name	Description
HttpAddRequestHeaders	Adds one or more HTTP request headers to the HTTP request handle
HttpEndRequest	Ends an HTTP request
HttpOpenRequest	Opens an HTTP request handle
HttpQueryInfo	Queries for information about an HTTP request
HttpSendRequest	Sends the specified request to the HTTP server
HttpSendRequestEx	Sends the specified request to the HTTP server

آشنایی با RAS API و WinInet API – بخش سوم

بررسی جزئیات توابع Dial-Up موجود در WinInet :

۱ – تابع InternetAutodial : بطور اتوماتیک باعث شماره گیری اتصال پیش فرض اینترنت توسط مودم می شود . اگر اتصال با موفقیت

انجام شود تابع مقدار true و در غیر اینصورت false بر می گرداند .

پارامترهای ورودی تابع :

dwFlags : فلگ کنترل کننده عملیات اتصال می باشد و یکی از مقادیر زیر را می تواند داشته باشد :

- INTERNET_AUTODIAL_FORCE_ONLINE
- INTERNET_AUTODIAL_FORCE_UNATTENDED

dwReserved : پارامتری رزرو شده است و بایستی صفر باشد .

چگونگی declare کردن تابع :

```
Public Declare Function InternetAutodial Lib
"wininet.dll" (ByVal dwFlags As Long, ByVal dwReserved
As Long) As Long
```

۲ – تابع InternetAutodialHangup : باعث قطع کردن یک اتصال dial-up اتوماتیک می شود . اگر قطع اتصال با موفقیت انجام شود تابع مقدار true و در غیر اینصورت false برمی گرداند . تابع دارای یک پارامتر ورودی به اسم dwReserved است که رزرو شده بود و بایستی صفر باشد .

چگونگی declare کردن تابع :

```
Public Declare Function InternetAutodialHangup Lib
"wininet.dll" (ByVal dwReserved As Long) As Long
```

۳ – تابع InternetDial : یک اتصال به اینترنت را با استفاده از یک ارتباط مودم مقداردهی اولیه می کند . پارامترهای ورودی آن عبارتند از :

hwndParent : هندل مربوط به پنجره parent

lpszConnectoid : نام ارتباط dial-up مورد استفاده

dwFlags : فلگ کنترل اتصال که یکی از مقادیر زیر را می تواند داشته باشد :

- INTERNET_AUTODIAL_FORCE_ONLINE

- INTERNET_AUTODIAL_FORCE_UNATTENDED

- INTERNET_DIAL_UNATTENDED : اتصال به اینترنت از طریق مودم بدون نمایش واسط کاربر

lpdwConnection : آدرس داده ای که شامل عدد متناظر با اتصال است .

dwReserved : پارامتری رزرو شده است و بایستی صفر باشد .

چگونگی declare کردن تابع :

```
Public Declare Function InternetDial Lib "wininet.dll"
(ByVal hwndParent As Long, ByVal lpszConnectoid As
String, ByVal dwFlags As Long, lpdwConnection As Long,
ByVal dwReserved As Long) As Long
```

۴ – تابع InternetGetConnectedState : این تابع وضعیت اتصال جاری به اینترنت را بر می گرداند . اگر اتصال برقرار باشد تابع مقدار true و در غیر اینصورت false برمی گرداند .

پارامترهای ورودی تابع عبارتند از :

lpdwFlags : توصیف وضعیت اتصال . این پارامتر یکی از مقادیر زیر را می تواند داشته باشد :

- INTERNET_CONNECTION_MODEM

- INTERNET_CONNECTION_LAN

- INTERNET_CONNECTION_PROXY

- INTERNET_CONNECTION_MODEM_BUSY

dwReserved : پارامتری رزرو شده است و بایستی صفر باشد .

چگونگی declare کردن تابع :

```
Public Declare Function InternetGetConnectedState Lib
"wininet.dll" (ByRef lpdwFlags As Long, ByVal
dwReserved As Long) As Long
```

۵ – تابع InternetGoOnline : پیغامی به کاربر برای دادن مجوز برای مقداردهی اولیه اتصال به یک URL را می دهد . اگر اینکار موفقیت آمیز باشد مقدار true و در غیر اینصورت false برمی گرداند . پارامترهای ورودی تابع عبارتند از :

URL : lpszURL وب سایت مورد نظر برای اتصال

hwndParent : هندل پنجره parent

dwReserved : پارامتری رزرو شده است و بایستی صفر باشد .

چگونگی declare کردن تابع :

```
Public Declare Function InternetGoOnline Lib
"wininet.dll" (ByVal lpszURL As String, ByVal
hwndParent As Long, ByVal dwReserved As Long) As
Long
```

۶ – تابع InyernetHangUp : به مودم می گوید که اتصال به اینترنت را قطع کند . پارامترهای این تابع عبارتند از :

dwConnection : شماره مربوط به اتصالی که می خواهیم آنرا قطع کنیم .

dwReserved : پارامتری رزرو شده است و بایستی صفر باشد .

چگونگی declare کردن تابع :

```
Public Declare Function InternetHangUp Lib
"wininet.dll" (ByVal dwConnection As Long, ByVal
dwReserved As Long) As Long
```

۷ – تابع InternetSetDialState : تنظیم نمودن وضعیت شماره گیری مودم . اگر تنظیم با موفقیت انجام شود تابع true و در غیراینصورت false برمی گرداند . پارامترهای ورودی تابع عبارتند از :

lpszConnectoid : نام اتصال dial-up

dwState : وضعیت مربوط به اتصال dial-up . در حال حاضر این پارامتر تنها مقدار INTERNET_DIALSTATE_DISCONNECTED را می تواند داشته باشد .

dwReserved : پارامتری رزرو شده است و بایستی صفر باشد .

چگونگی declare کردن تابع :

```
Public Declare Function InternetSetDialState Lib
"wininet.dll" (ByVal lpszConnectoid As String, ByVal
dwState As Long, ByVal dwReserved As Long) As Long
```

بررسی فلگهای مورد استفاده در توابع dial-up :

۱ – فلگهای تابع InternetDial :

```
= Public Const INTERNET_DIAL_UNATTENDED
x8000۰' &H8000&
= Public Const INTERENT_GOONLINE_REFRESH
x00000001۰'  H1&
H1& = Public Const INTERENT_GOONLINE_MASK
x00000001۰'
```

۲ – فلگهای تابع InternetAutoDial :

```
Public Const
INTERNET_AUTODIAL_FORCE_ONLINE = 1
Public Const
INTERNET_AUTODIAL_FORCE_UNATTENDED = 2
Public Const
INTERNET_AUTODIAL_FAILIFSECURITYCHECK =
4
```

۳ – فلگهای تابع InternetGetConnectedState :

```
Public Const INTERNET_CONNECTION_MODEM = 1
INTERNET_CONNECTION_LAN = 2 Public Const
۴ = Public Const INTERNET_CONNECTION_PROXY
Public Const
INTERNET_CONNECTION_MODEM_BUSY = 8
```

۴ – فلگهای مربوط به dial handler اختصاصی :

```
Public Const INTERNET_CUSTOMDIAL_CONNECT =
0
Public Const
INTERNET_CUSTOMDIAL_UNATTENDED = 1
Public Const
INTERNET_CUSTOMDIAL_DISCONNECT = 2
```

۵ – فلگهای عملیاتی پشتیبانی شده برای dial handler اختصاصی :

```
Const  Public
INTERNET_CUSTOMDIAL_SAFE_FOR_UNATTENDE
D = 1
Public Const
INTERNET_CUSTOMDIAL_WILL_SUPPLY_STATE =
2
```

```
Public Const
INTERNET_CUSTOMDIAL_CAN_HANGUP = 4
```

۶ - وضعیتهای مربوط به InternetSetDialState :

```
Public Const
INTERNET_DIALSTATE_DISCONNECTED = 1
```

+ برای اطلاعات بیشتر در مورد این توابع به این آدرس مراجعه کنید .

مثالی از کار با توابع DialUp موجود در کتابخانه WinInet

در این بخش که آخرین بخش از مباحث WinInet API است برنامه ای نمونه برای کار با توابع مودمی این کتابخانه ارائه خواهیم داد :

برای نوشتن برنامه ای که بتوان از طریق آن با استفاده از مودم به اینترنت متصل شد بصورت زیر عمل می کنیم :

در ابتدا بایستی تابع InternetDial را Declare کنیم :

```
Private Declare Function InternetDial Lib "wininet.dll"
Alias "InternetDialA" (ByVal hwndParent As Long,
ByVal lpszConnectoid As String, ByVal dwFlags As Long,
lpdwConnection As Long, ByVal dwReserved As Long) As
Long
```

سپس وضعیت شماره گیری را در متغیری به اسم lOption قرار می دهیم . این متغیر می تواند مقادیر زیر را داشته باشد :

- DF_FORCE_ONLINE
- DF_FORCE_UNATTENDED
- DF_DIAL_FORCE_PROMPT
- DF_DIAL_UNATTENDED

حال نام اتصالی را که می خواهیم از آن استفاده شود در متغیری به اسم ConnectionName قرار می دهیم .
همچنین دو متغیر به اسم ConnectionID و RetVal را از نوع long تعریف می کنیم .

حال تابع InternetDial را بصورت زیر صدا می کنیم :

```
RetVal = InternetDial(Me.hwnd, ConnectionName,
lOption, ConnectionID, 0)
```

اگر RetVal مخالف صفر باشد عمل Dial بدرستی انجام شده است .

برای قطع اتصال فوق بایستی از تابع InternetHangUp استفاده کنیم . برای اینکار ابتدا تابع فوق را Declare می کنیم :

```
Private Declare Function InternetHangUp Lib
"wininet.dll" (ByVal dwConnection As Long, ByVal
dwReserved As Long) As Long
```

سپس این تابع را بصورت زیر فراخوانی می کنیم :

```
RetVal = InternetHangUp(ConnectionID, 0)
```

برای اینکه مودم را مجبور کنیم تا بطور اتوماتیک از اتصال پیش فرض سیستم برای شماره گیری استفاده کند از تابع InternetAutodial استفاده می کنیم .
برای اینکار ابتدا تابع را Declare می کنیم :

```
Private Declare Function InternetAutodial Lib
"wininet.dll" (ByVal dwFlags As Long, ByVal
hwndParent As Long) As Long
```

سپس تابع را بصورت زیر فراخوانی می کنیم :

```
RetVal =
InternetAutodial(ADF_FORCE_UNATTENDED,
Me.hwnd)
```

اگر RetVal مخالف صفر باشد عمل AutoDial بدرستی انجام شده است .

برای قطع اتصالی که توسط AutoDial ایجاد شده از تابع InternetAutodialHangup استفاده می کنیم . ابتدا این تابع را Declare می کنیم :

```
Private Declare Function InternetAutodialHangup Lib
"wininet.dll" (ByVal dwReserved As Long) As Long
```

فراخوانی این تابع بصورت زیر است :

```
Call InternetAutodialHangup(0)
```

برای اینکه بفهیم آیا اتصال به اینترنت وجود دارد یا نه از تابع InternetGetConnectedStateEx استفاده می کنیم . برای اینکار ابتدا تابع را Declare می کنیم :

```
Private Declare Function InternetGetConnectedStateEx
Lib "wininet.dll" Alias "InternetGetConnectedStateExA"
(lpdwFlags As Long, lpszConnectionName As Long,
dwNameLen As Long, ByVal dwReserved As Long) As
Long
```

سپس تابع را بصورت زیر فراخوانی می کنیم :

```
strConnectionName = Space(256)
lNameLen = 256
lPtr = StrPtr(strConnectionName)
lNameLenPtr = VarPtr(lNameLen)
```

RetVal = InternetGetConnectedStateEx(lConnectionFlags, ByVal lPtr, ByVal lNameLen, 0)

که strConnectionName از نوع String و بقیه متغیرها از نوع Long هستند .

اگر RetVal مخالف صفر باشد اتصال برقرار است .

ثابتهایی که در کدهای فوق استفاده شده عبارتند از :

```
Private Const
INTERNET_AUTODIAL_FORCE_ONLINE = 1&
Private Const
INTERNET_AUTODIAL_FORCE_UNATTENDED = 2&
Private Const
INTERNET_AUTODIAL_FAILIFSECURITYCHECK =
4&

Private Const INTERNET_DIAL_FORCE_PROMPT =
&H2000
Private Const INTERNET_DIAL_SHOW_OFFLINE =
&H4000
Private Const INTERNET_DIAL_UNATTENDED =
&H8000
```

Server-Side ActiveX Dll Programming

Server-Side ActiveX Dll Programming – بخش اول

مقدمه

با قراردادن کدهای ASP درون component های server side ، برنامه

نویس نه تنها می تواند از قابلیت های ویژوال بیسیک در نوشتن کدهای خود استفاده کند بلکه سرعت load صفحات ASP وی نیز افزایش می یابد . همچنین این روش راهکاری برای کپسوله سازی و حفاظت از کدهای ASP می باشد .

در این درس یک نمونه اکتیویکس server-side را توسط ویژوال بیسیک ایجاد نموده و از آن در صفحات ASP استفاده خواهیم کرد .

اجزای Server-Side

اکتیوکس های server-side بر خلاف اکتیوکس های clict-side بر روی سرور وب اجرا می شوند و بنابراین بایستی وب سرور مورد استفاده با این تکنولوژی سازگار باشد . زمانیکه وب سرور دستوری را برای پردازش یکسری اطلاعات درون یک صفحه ASP دریافت می کند دستوراتی که درون تگهای ٪> قرار دارند بررسی می شوند . با استفاده از ویژوال بیسیک می توان یک اکتیوکس dll ساخت که جایگزین این کدهای ASP شود . در اینصورت تنها کافیست یک شی از کلاسهای موجود در این dll ساخته شود تا بتوان از قابلیتهای آن استفاده نمود .

ایجاد Dll ActiveX در ویژوال بیسیک

برای ساخت یک اکتیواکس ویژوال بیسیک را اجرا کرده و توسط گزینه New Project پروژه ای از نوع ActiveX Dll ایجاد کنید . پس از اینکه شما روی آیکون ActiveX dll کلیک کنید ویژوال بیسیک پروژه ای پیش فرض بهمراه ی: کلاس خالی برای شما ایجاد می کند . می توانید هم نام پروژه و هم نام کلاس را تغییر دهید همچنین می توانید کلاسهای دیگری به پروژه اضافه کنید .

حال برای اینکه بتوان دستورات ASP را استفاده نموده بایستی از

منوی Project وارد بخش References شده و مورد Microsoft Active Server Pages Object Library را انتخاب کنید .

استفاده از متدهای ASP در کلاس های ActiveX

بمنظور استفاده از متدهای ASP در کلاسهای ActiveX بایستی ابتدا روتینی به اسم OnStartPage در داخل کلاس تعریف کنید . ساختار این روتین بصورت زیر است :

```
Public Sub OnStartPage(PassedScriptingContext As
ScriptingContext)x

End Sub
```

زمانیکه کاربر یک صفحه ASP را که شامل شی ای از کلاس ما باشد فراخوانی کند IIS، SciptingContext را به شی ما پاس می دهد . ScriptingContext حاوی تمام متدها و خصوصیات ASP می باشد که برای استفاده در دسترس هستند . حال بایستی در روتین OnStartPage تمام اشیای ASP که توسط ScriptingContext در دسترس هستند را به اشیایی از همان نوع assign کنیم تا در صورت لزوم بتوانیم از آنها استفاده نمائیم . بنابراین قبل از نوشتن روتین OnStartPage متغیرهای زیر را تعریف می کنیم :

```
ScriptingContext Private MyScriptingContext As
Private MyApplication As Application
Request Private MyRequest As
Private MyResponse as Response
Server Private MyServer As
Private MySession As Session
```

حال در روتین OnStartPage بایستی اشیا فوق را مقداردهی کنید :

```
Public Sub OnStartPage(PassedScriptingContext As
ScriptingContext)x
Set MyScriptingContext=PassedScriptingContext
MyApplication=MyScriptingContext.Application Set
MyRequest=MyScriptingContext.Request Set
MyResponse=MyScriptingContext.Response Set
MyServer=MyScriptingContext.Server Set
MySession=MyScriptingContext.Session Set
End Sub
```

از تمام اشیا فوق مشابه نوشتن صفحات ASP می توانیم در متدهایی که برای کلاس می نویسیم استفاده کنیم . برای مثال کد ASP زیر را در نظر بگیرید :

```
MyTempVar=Request.Form(“username”)x
MyTempVar)x & “ : Entered Response.Write(“You
<%
```

حال فرض کنید می خواهیم همین دستورات را در یک متد از کلاس بنویسیم :

```
Public Sub MyMethod()x
String Dim MyTempVar As
MyTempVar=MyRequest.Form(“username”)x
```

```
MyTempVar)x & " : MyResponse.Write("You Entered
End Sub
```

نکته دیگری که باید در نظر داشت نوشتن روتینی است که در زمان پایان کار با شی فراخوانی می شود . این روتین OnEndPage نام دارد و در آن اشیایی که در روتین OnStartPage مقداردهی کرده ایم را آزاد می کنیم :

```
Public Sub OnEndPage()x
MyScriptingContext=Nothing Set
Set MyApplication= Nothing
Nothing =Set MyRequest
Set MyResponse= Nothing
Set MyServer= Nothing
Nothing =Set MySession
End Sub
```

پس از نوشتن متدهای موردنظرتان پروژه را ذخیره کنید . حال برای کامپایل پروژه از منوی File مورد File/Make Dll را انتخاب نمائید تا پروژه کامپایل شده و فایل dll موردنظرتان ساخته شود . این فایل را در دایرکتوریی که صفحات ASP شما در آنجا قرار دارد کپی کنید .
نکته ای که باید مورد توجه قرار داد اینست که در صورتیکه می خواهید از این dll در سیستم دیگری استفاده کنید ابتدا بایستی آنرا رجیستر کنید . برای رجیستر کردن یک dll از برنامه regsvr32.exe موجود در دایرکتوری سیستم ویندوز استفاده می شود :

```
C:\InetPub\wwwroot\Example\Example.dll Regsvr32.exe
```

استفاده از ActiveX Dll در صفحات ASP

برای استفاده از کلاس نوشته شده در فایل dll در صفحات ASP ابتدا بایستی یک شی از آن کلاس ایجاد کنیم :

```
= Set ObjRefrence
Server.CreateObject(“ProjectName.ClassName”)x
<%
```

پس از ساخت شی می توانیم از متدهای موجود در کلاس استفاده کنیم :

```
ObjReference.MyMethod()x
<%
```

بخش دوم – Server-Side ActiveX Dll Programming

COM+ چیست ؟

روشی که برای دسترسی به Object های ASP در درس گذشته بیان شد تا IIS 3.0 استفاده می شد . این روش استفاده از متدهای OnStart و OnEnd بود . البته گرچه هنوز این روش از سوی IIS ورژن ۵ پشتیبانی می شود اما دارای یک مشکل است :

اگر بخواهید از یک کلاس در کلاس دیگری استفاده کنید نمی توانید در

کلاس مورد استفاده ، دو متد ذکر شده را قرار دهید و بنابراین به اشیای ASP دسترسی نخواهید داشت .

راه حلی که برای این مشکل ارائه شد تکنولوژی COM+ می باشد .

بطور خلاصه در این تکنولوژی شیی به اسم ObjectContext وجود دارد که از طریق آن می توانید به اشیای ASP دسترسی داشته باشید .

برای استفاده از تکنولوژی COM+ ابتدا از منوی Project، References را انتخاب کرده و مورد زیر را انتخاب کنید :

COM+ Services Type Library

سپس در کلاسهای خود متغیرهای زیر را تعریف کنید :

```
Dim Request As ASPTypeLibrary.Request
Response As ASPTypeLibrary.Response Dim
ASPTypeLibrary.Server Dim Server As
Dim Session As ASPTypeLibrary.Session
Application As ASPTypeLibrary.Application Dim
```

حال در متد Initialize هر کلاس بایستی شی ObjectContext را تعریف کرده و مقداردهی کنید . سپس متغیرهای بالا را با استفاده از این شی مقداردهی نمائید :

```
Private Sub Class_Initialize()x
ObjectContext Dim objCtx As
Set objCtx = GetObjectContext
objCtx.item("Request")x = Set Request
Set Response = objCtx.item("Response")x
objCtx.item("Server")x = Set Server
Set Session = objCtx.item("Session")x
Application = objCtx.item("Application")x Set
End Sub
```

پاسخ به سوالات شما :

۱ – چه جوری دکمه ها و لیست باکس ها در ویژوال بیسیک را شکل اکس پی کنیم؟اصلان می شه؟

پاسخ : ؟؟؟؟

۲ – تویه vb چطوری میشه فایله اجراییdllهاشو نخاد و بدونه اونا اجرا بشه ؟

پاسخ : برنامه های اجرایی ویژوال بیسیک برای اجرا شدن به یکسری فایلهای دیگه نیاز دارند مثلاً Vb Runtime Dll . برای اینکه به این فایلها نیازی نباشد بایستی یک برنامه نصب setup file برای پروژه تان بسازید تا بتوان برنامه را روی هر کامپیوتری نصب و اجرا کرد . ساده تری راه استفاده از ابزار Deployment & Package موجود در ویژوال استدیو است . ابزارهای حرفه ای تر عبارتند از : InstallShield، InstallWise، Setup Factory و ...

۳ – اگه ممکنه یه توضیحی درباره ی دی کد کردن دی ال ال mpr دهید .

پاسخ : ؟؟؟؟

۴ – چطوری میشه تو یه تکس باکس در ویژوال بیسیک فرمان داد اینتر شود یعنی به خط بعدی رود؟ (یعنی زمانی که مولتی لاین است)

پاسخ : استفاده از کاراکتر vbCrLf

۵ – من می خوام با زبانهایی PHP و ASP برنامه بنویسیم ولی تمامی این زبانها Server_side هستند و من باید روی سرور این کار ها رو انجام بدم من می خوامم بدونم که چه طوری می تونم کامپیوتر خودمو وب سرور کنم البته یه چیزهای می دونم که باید IIS رو نصب کنم ولی نه به طور کامل خواهش می کنم کمکم کنید

پاسخ : بایستی ویندوز ۲۰۰۰ یا XP نصب کرده و از IIS آنها استفاده کنید . البته در ویندوزز ۲۰۰۰ IIS بطور اتوماتیک نصب نمی شود و باید آنرا خودتان اضافه کنید . برای اطلاعات بیشتر در مورد کار با IIS به کتاب ها و مراجع اینترنتی مراجعه کنید مراجعه کنید .

بخش سوم – Server-Side ActiveX Dll Programming

خلاصه ای بر چگونگی کار با بانک های اطلاعاتی (Database) در وی بی :

+ برای آشنایی کامل با چگونگی کار با بانک های اطلاعاتی در ویژوال بیسیک و بطور کلی Database Programming به کتاب برنامه نویسی بانک های اطلاعاتی در ویژوال بیسیک انتشارات نص رجوع کنید .

قبل از اینکه چگونگی نوشتن یک کلاس Database برای استفاده در ASP را آموزش دهم ، مقدمه ای بر چگونگی کار با بانک های اطلاعاتی در وی بی را شروع می کنم .

برای کار با بانک های اطلاعاتی در ویژوال بیسیک روشها و امکانات مختلفی وجود دارد که یکی از بهترین آنها استفاده از تکنولوژی (ADO ActiveX Data Object) می باشد . بدون هیچ توضیحی در مورد ساختار این تکنولوژی و نیز سایر روشهای دیگر ، به سراغ روش استفاده از این تکنولوژی می روم :

– برای کار با ADODB ابتدا بایستی از Reference ها مورد Microsoft ActiveX Data Object را انتخاب کرد .

– قدم بعد تعریف یک شی ADO Connection برای اتصال به بانک اطلاعاتی است :

Dim cn As ADODB.Connection

- سپس بایستی این شی ADO Connection را ایجاد نمود :

Set cn = New ADODB.Connection

- همچنین بایستی یک شی ADO Recordset برای گرفتن مجموعه ای از رکوردهای بانک اطلاعاتی تعریف کرد :

Dim rs As ADODB.Recordset

- حال بایستی اتصال به بانک اطلاعاتی را باز نمود . در این مرحله با توجه به نوع بانک اطلاعاتی و اتصالی که می خواهیم داشته باشیم عبارت اتصال ممکن است متفاوت باشد . فرض کنیم عبارت اتصال را در یک متغیر نوع String به اسم ConnString قرار دهیم :

• در صورتی که بانک اطلاعاتی مقصد SQL Server باشد و بخواهیم بطور مستقیم و بدون استفاده از واسط ODBC به آن متصل شویم :

**ConnString="Provider=SQLOLEDB.1;Password=yourpas
Security Info=True;User ID=yourusername; sowrd;Persist
Initial Catalog=yourDatabaseName;Data
"Source=yourServerName**

• در صورتی که بانک اطلاعاتی مقصد SQL Server باشد و بخواهیم با استفاده از واسط ODBC به آن متصل شویم :

**ConnString="Provider=MSDASQL.1;Password=yourpass
Security owrd;Persist**

Info=True;UserID=yourusername;DataSource=yourODB
"C_DataSourceName;Mode=ReadWrite

• در صورتی که بانک اطلاعاتی مقصد Access باشد :

ConnString=" Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=yourDatabaseFilePath;Persist Security
"Info=False

حال بایستی این اتصال را باز نمود :

cn.Open(ConnString)x

- قدم بعدی ایجاد شی ADO Recordset می باشد :

Set rs = New ADODB.Recordset

- سپس بایستی با توجه به کاری که می خواهیم با جداول آن بانک اطلاعاتی انجام دهیم یک sql query را توسط شی Recordset به آن بفرستیم :

adLockOptimistic ,rs.Open yourSQLquery,cn,OpenKeyset

بعبارت دیگر نوع باز کردن Recordset متفاوت است و توسط query مورد نظر مشخص می شود برای مثال برای انتخاب فیلدهای یک Table :

SELECT DISTINCT yourfields from yourtable WHERE yourcondition

نکته : نوع قفل کردن مجموعه رکورد می تواند adLockReadOnly نیز باشد .

- حال با استفاده از این Recordset می توان یکسری کار را روی رکوردهای موجود در جداول انجام داد برای مثال :

۱ – حرکت به ابتدای مجموعه رکورد :

```
rs.MoveFirst
```

۲ – حرکت در طول مجموعه رکورد :

```
Do
rs.fields(0)x = tmp
.
.
.
rs.MoveNext
Loop Until (rs.EOF)x
```

۳ – بستن مجموعه رکورد و بستن اتصال به بانک اطلاعاتی :

```
rs.Close
cn.Close
```

۴ – حذف رکورد جاری از مجموعه رکورد :

```
rs.delete
```

۵ – ایجاد رکورد جدید در مجموعه رکورد :

```
Dim fields(RecordsetFieldsCount) As Variant
values(RecordsetFieldsCount) As Variant Dim
fields(0) = Field 1 Name
Field 2 Name = (fields(1
.
.
.
fields(n) = Field n Name

Field 1 Value = (values(0
values(1) = Field 2 Value
.
.
.
Value values(n)= Field n

rs.AddNew fields, values
rs.update
```

بخش چهارم – Server-Side ActiveX Dll Programming

آشنایی با چند query برای کار با جداول بانک های اطلاعاتی

۱- SELECT : برای انتخاب رکوردها از یک جدول استفاده می شود .
فرمت کلی این دستور بصورت زیر است :

condition **WHERE** ***tablename*** **FROM** ***fieldnames*** **SELECT** ***fieldnames*** **ORDER BY**

مثال : فرض کنید یک جدول به اسم mytable داریم که دارای دو فیلد به نامهای id از نوع integer و name از نوع string باشد :

- انتخاب تمامی رکوردهای جدول :

```
mytable"x query="Select * from
```

- انتخاب فیلد name تمام رکوردهای جدول :

```
mytable"x query="Select name from
```

- انتخاب رکوردهایی از جدول که فیلد id آنها برابر ۲ باشد :

```
query="Select * from mytable where id=2"x
```

- انتخاب رکوردهایی از جدول که فیلد name آنها برابر a باشد :

```
name='a'"x query="Select * from mytable where
```

توجه داشته باشید که چون فیلد name از نوع string است در دستور فوق از ' برای مقدار فیلد name استفاده شده است .

- انتخاب رکوردهایی از جدول که فیلد id آنها برابر ۲ و فیلد name آنها برابر a باشد و بر حسب id مرتب شده باشند .

```
query="Select * from mytable where id=2 and name='a'
order by id"x
```

نکته : در صورتیکه بخواهیم از یک متغیر برای مقداردهی به یک فیلد در query استفاده کنید با توجه به اینکه آن متغیر از نوع integer و یا string است باید بصورت زیر عمل کنیم :

```
Dim mId as integer
```

```
string Dim mName as
mId=1
mName="a"x
" & (str(mId & "=query="Select * from mytable where id
x"'" & mName & "'=and name
```

۲ – INSERT : این دستور برای قرار دادن یک رکورد در جدول استفاده می شود . فرمت کلی این دستور بصورت زیر است :

```
(…,field2name,field1name) tablename INSERT INTO
x(…,field2value,field1value) VALUES
```

مثال :

```
mytable (id,name) values (1,'a')"x query="Insert into
```

۳ – UPDATE : این دستور برای تغییر مقادیر یک رکورد از جدول استفاده می شود . فرمت کلی این دستور بصورت زیر است :

```
,field1value=field1name SET tablename UPDATE
…,field2value=field2name
```

مثال :

```
mytable set id=2 , name='b'"x query="Update
```

۴ – DELETE : این دستور برای حذف یک یا چند رکورد از جدول استفاده می شود . فرمت کلی این دستور بصورت زیر است :

DELETE FROM *tablename* WHERE *condition*

مثال :

id=1"x query="Delete from mytable where

حال که با تکنولوژی ADODB و نیز query های مختلف برای کار با جداول بانک های اطلاعاتی آشنا شدید می توانیم کلاسی برای کار با بانک های اطلاعاتی بنویسیم و از آن در صفحات asp استفاده کنیم اما همانطور که در قسمت قبل دیدید برای اتصال به یک بانک اطلاعاتی نیاز به یکسری اطلاعات مثل نام سرور ، نام بانک اطلاعاتی و ... داریم . چند روش برای دادن این اطلاعات وجود دارد :

۱ – در کلاسی که می نویسیم این مقادیر را مشخص کنیم . اشکال این روش اینست که از کلاس نوشته شده تنها برای یک کاربرد خاص می توانیم استفاده کنیم و در صورتیکه سرور بانک اطلاعاتی و یا نام بانک اطلاعاتی ویا username و password اتصال تغییر کند بایستی در کلاس نوشته شده نیز تغییرات را اعمال کرده و مجدداً آنرا کامپایل کنیم .

۲ – راه حل دوم اینست که پارامترها را از طریق asp به متد اتصال موجود در کلاس بفرستیم برای مثال :

db=server.createobject(“dbclass.database)x Set Db.connect(servername,databasename,username,password) x

که coonect متد اتصال به بانک اطلاعاتی در کلاس database می باشد .

۳ – روش سوم آنست که این پارامترها را در یک فایل XML قراردهیم و

در متد connect آنها را از فایل بخوانیم . مزیت این روش اینست که پارامترها هم به آسانی قابل تغییر بوده و هم براحتی آنها می توان آنها را استخراج نمود .

برای این منظور بایستی ابتدا یک کلاس برای خواندن اطلاعات از فایل xml بنویسیم که موضوع درس بعد می باشد .

نکته : در صورتیکه با فرمت فایلهای xml آشنا نیستید پیشنهاد می کنم یک مطالعه مقدماتی در این زمینه انجام دهید .

Server-Side ActiveX Dll Programming – بخش پنجم

نوشتن کلاس Database – بخش اول :

پس از مباحثی که در مورد شی ADODB و چگونگی استفاده از آن در وی بی و نیز استفاده از فایلهای XML داشتیم اکنون می توانیم یک کلاس کامل و قدرتمند برای کار با بانکهای اطلاعاتی در ASP بنویسیم .

مراحل کار بصورت زیر می باشد :

۱ – ابتدا یک پروژه از نوع ActiveX Dll ایجاد کنید و نام آنرا DBase بگذارید .

۲ – از بخش References مواردی را که در مباحث قبلی گفته شد به پروژه اضافه کنید .

۳ – متغیر Cn را برای کلاس بصورت زیر تعریف کنید :

```
Private Cn As ADODB.Connection
```

۴ – ابتدا یک متد به اسم InitialConnection برای کلاس می نویسیم . در این متد ابتدا پارامترهای اتصال به بانک اطلاعاتی را مشابه آنچه در درس قبل گفته شد از یک فایل XML به اسم config.xml می خوانیم و با استفاده از آنها اتصال به بانک اطلاعاتی را باز می کنیم :

```
Public Sub InitialConnection()x
database_name, server_name ,Dim userName, Password
Dim xmlf As NewXMLReader
xmlf.Initiate("config.xml")x Call
x ("userName = xmlf.getvalue("DataBaseID
Password = xmlf.getvalue("DataBasePassword") x
xmlf.getvalue("DataBaseName") x = database_name
xmlf.getvalue("ServerAddress")x = server_name
CreateObject("ADODB.Connection")x = Set Cn
= Cn.ConnectionString
& Password & "=Provider=SQLOLEDB.1;Password"
& userName & "=Info=True;User ID Persist Security;"
"=Data Source;" & database_name & "=Initial Catalog;"
server_name &
Cn.Mode = adModeReadWrite
Cn.Open
End Sub
```

لازم به ذکر است که XMLReader کلاس کار با فایلهای XML است که در قسمت قبلی در مورد آن صحبت کردیم .

۵ – برای بستن اتصال متد زیر را به کلاس اضافه کنید :

```
EndConnection()x Public Sub
Cn.Close
Set Cn = Nothing
End Sub
```

۶ – برای اجرای query هایی که نتیجه آنها از نوع Boolean است (مثل Insert و Delete) متدی به اسم ExecuteUpdate را به کلاس اضافه کنید :

```
ExecuteUpdate(ByVal squery As String)  Public Function
As Boolean
Recordset Dim myrs As New
If Not (makesInjection(squery)) Then
adLockOptimistic = myrs.LockType
Set myrs = Cn.Execute(squery) x
ExecuteUpdate = True
Exit Function
Else
ExecuteUpdate = False
Exit Function
If End
End Function
```

۶ – برای اجرای query هایی که نتیجه آنها از نوع RecordSet است (

مثل Select) متدی به اسم ExecuteQuery را به کلاس اضافه کنید :

```
Public Function ExecuteQuery(ByVal squery As String)
As Recordset
Not (makesInjection(squery)) Then If
Cn.Execute(squery)x = Set ExecuteQuery
Exit Function
Else
Nothing = Set ExecuteQuery
Exit Function
End If
End Function
```

همانطور که می بینید در دو متد ExecuteUpdate و ExecuteQuery از تابعی به اسم makesInjection استفاده شده است . این تابع بررسی می کند که آیا در query ورودی SQL-Injection وجود دارد یا نه .

Server-Side ActiveX Dll Programming – بخش ششم

نوشتن کلاس Database : بخش دوم

sql-injection چیست ؟

همانطور که می دانید در اغلب برنامه های کاربردی تحت وب از بانکهای اطلاعاتی استفاده می شود . این برنامه ها داده های ورودی کاربر را از طریق فرمهای html دریافت کرده و بر اساس آن یک query تولید کرده و آنرا به بانک اطلاعاتی ارسال می کنند . در واقع ارتباط بین برنامه تحت وب با بانک اطلاعاتی بر اساس تولید query از داده های

کاربر برقرار می شود . اکثر این برنامه ها از زبان **SQL** برای این ارتباط استفاده می کنند . اما نکته ای که در این بین وجود دارد اینست که تولید **query** بر اساس داده هایی که کاربر مستقیماً در فیلدهای ورودی صفحه وب وارد کرده می تواند خطرناک باشد . بعبارت دیگر اگر برنامه محتویات یک فیلد را که توسط کاربر وارد شده در جلوی یک دستور **SQL** بچسباند و آنرا جهت اجرا روی بانک اطلاعاتی بفرستد در اینصورت یک هکر ماهر که با زبان **SQL** آشنا باشد می تواند محتویات این فیلدها را طوری با دستورات **SQL** پر کند و چون ای داده ها مستقیماً برای تولید **query** استفاده می شود ممکنست آن **query** تبدیل به یک فرمان مخرب شده و پس از اجرا ، اهداف نفوذگر را برآورده نماید .

مثال : فرض کنید دو فیلد به اسمهای **username** و **password** در یک فرم وب قرار دارد که برای ورود به یک سایت استفاده می شود . همچنین فرض کنید از اطلاعات این فیلدها بطور مستقیم یک **query** بصورت زیر برای بانک اطلاعاتی ارسال شود :

```
us=request.form("username")
psw=request.form("password")
query="SELECT * FROM Users WHERE
username='"&us&"' AND password='"&psw&"'"
```

حال در صورتیکه هکر یک **username** صحیح (مثلاً **xxx**) از سیستم را بداند و در فیلد **username** مقدار صحیح را وارد کرده و در فیلد **password** عبارت زیر را وارد کند :

```
1111111' or  username='xxx'
```

در اینصورت **query** بصورت زیر در می آید :

SELECT * FROM Users WHERE username='xxx' AND password='111111' or username='xxx'

در اینصورت هکر بدون دانستن یک password مجاز می تواند به سیستم وارد شود .
این امر بخاطر آنست که چون فیلدهای وارد شده توسط کاربر بطور مستقیم در query قرار داده شده اند هکر توانسته کاراکتر ' را که در زبان SQL یک کاراکتر کنترلی بوده و عملکرد خاصی دارد (عمل خاتمه دادن به عبارت SQL) را در در query بگنجاند و سپس با دادن دستورات SQL مناسب کنترل را بدست بگیرد .

این سناریو می تواند بسیار خطرناکتر باشد زیرا هکر می تواند از سایر دستورات SQL مثل INSERT و DELETE نیز استفاده کند .

– در نگارش مطالب فوق از کتاب " نفوذگری در شبکه و روشهای مقابله با آن " نوشته مهندس احسان ملکیان استفاده شده است . برای آشنایی بیشتر با sql-injection و روشهای مقابله با آن به صفحات ۳۲۰ تا ۳۲۸ این کتاب مراجعه کنید .

برای مقابله با این حملات بایستی از داده های ارسال شده توسط کاربر مستقیماً query تولید نکنیم بلکه ابتدا عدم وجود کاراکترهای کنترلی مثل ' و " و ; و *و غیره را در آن بررسی کنیم . برنامه ابتدا باید در query وجود چنین کاراکترهایی را در مکانهای غیرمجاز بررسی کند . در بخش بعدی برنامه ای را جهت بررسی query های SQL بمنظور مقابله با sql-injection ارائه خواهم داد .

بخش هفتم – Server-Side ActiveX Dll Programming

تابع بررسی وجود sql-injection که در قسمت قبل در آن صحبت کردیم بصورت زیر است :

```
Private Function makesInjection(ByVal query As String)
As Boolean
  Dim specialCharacters() As String
  Dim inQoute As Boolean
  specialCharacters = "-- ;,"
  inQoute = False
  For i = 1 To Len(query)
    Char = Mid(query, i, 1)
    If Mid(query, i, 1) = "'" And inQoute = False Then
      inQoute = True
      GoTo EndFor
    End If
    If Mid(query, i, 1) = "'" And inQoute = True Then
      inQoute = False
      GoTo EndFor
    End If
    If inQoute = False Then
      For Index = 1 To UBound(specialCharacters)
        schar = specialCharacters(Index)
        cchar = Mid(query, i, Len(schar))
        If schar = cchar Then
          Exit For
        End If
      Next
      If Index < UBound(specialCharacters) Then
        makesInjection = True
        Exit Function
      End If
    End If

EndFor:
      Next
   If inQoute = True Then
    makesInjection = True
```

```
    Else
      makesInjection = False
    End If
End Function
```

ورودی این تابع query شما و خروجی آن false یاtrue است .

عملکرد تابع بصورت زیر است :

این تابع در طول رشته query شروع به حرکت می کند و هر کاراکتر از آنرا بررسی می نماید . در صورتیکه کاراکتر جاری ' باشد و داخل ' ' نباشیم متغیر مربوط به آن true شده و حلقه یکی بجلو می رود . اما در صورتیکه کاراکتر جاری ' باشد و داخل ' ' باشیم متغیر مربوط به آن false شده و حلقه یکی بجلو می رود . سرانجام در صورتیکه داخل ' ' نباشیم بررسی می شود که این کاراکتر یکی از کاراکترهای غیر مجاز (کاراکترهای موجود در رشته specialCharacters) نباشد که اگر باشد تابع true بر می گرداند .

پس از اتمام حلقه متغیر مربوط به ' بررسی می شود که اگر true باشد در صورت injection وجود داسته و تابع نیز true بر می گرداند .

بخش هشتم – Server-Side ActiveX Dll Programming

نکته ای در مورد شی Recordset :

متد ExecuteQuery که در کلاس Database نوشتیم یک رکوردست را بعنوان نتیجه انجام query ورودی روی بانک اطلاعاتی شما برمی گرداند .

همانطور که می دانید توسط خصوصیت RecordCount می توان تعداد رکوردهای نتیجه شده از یک query را که در رکوردست قرار

دارند بدست آورد .
اما مشکلی وجود دارد اینست که با روشی که ما در قسمتهای قبل برای اجرای query در این متد استفاده کرده بودیم (myrs = Cn.Execute query) نمی توان از خاصیت Recordcount رکوردست استفاده نمود زیرا همیشه ۱- برمی گرداند . بعبارت دیگر در عبارت زیر مقدار count همیشه ۱- خواهد بود :

```
myrs.execute(query)
count=myrs.RecordCount
```

برای حل این مشکل بایستی رکوردست را با CursorType مساوی adOpenStatic باز کرد . بعبارت دیگر بجای دستورات فوق از دستور زیر استفاده کنید :

```
adLockOptimistic ,myrs.Open squery, Cn, adOpenStatic
count=myrs.RecordCount
```

+ برای اطلاعات بیشتر در این زمینه به این صفحه مراجعه کنید .

کنترل Web Browser – ساخت مرورگر صفحات وب

برنامه Internet Explorer یا iexplore.exe در واقع برنامه کوچکی است که وظیفه اصلی آن ایجاد چارچوبی برای بهم پیوستن عناصر مختلف است و این عناصر هستند که وظایف اصلی مثل load کردن صفحات وب ، اجرای کدهای Html و غیره را انجام می دهند . اصلی ترین عنصری که مستقیماً توسط iexplore.exe استفاده می شود کنترل Webbrowser (موجود در فایل shdocrw.dll) می باشد . وظیفه این فایل dll ، عبارت است از حرکت بین صفحات وب ، مدیریت تاریخچه صفحات دیده شده و غیره . این فایل خود از فایل دیگری بنام Mshtml.dll استفاده می کند که وظیفه آن بررسی و اجرای فایلهای html است . مایکروسافت به برنامه نویسان این امکان را داده که بتوانند در برنامه هایشان از کنترل webbrowser استفاده کنند . با استفاده از این کنترل می توان به سادگی یک مرورگر وب تقریباً کامل ساخت .

خصوصیات کنترل Webbrowser :

Webbrowser علاوه بر خواص استانداردی مثل width، height و ...

خواص زیر را دارد :

۱ – Busy : اگر در حال load کردن یک صفحه یا در حال جستجو در وب باشد این خاصیت True است . توسط متد Stop می توان عملیات جاری را متوقف کرد .

۲ – Container : ارجاع به شی نگهدارنده کنترل webbrowser

۳ – Document : ارجاع به صفحه html فعلی . برای کار با این صفحه html می توان از خواص و متدهابی شی Document استفاده کرد .

۴ – LocationName : حاوی آدرس محلی است که اکنون در کنترل webbrowser، load شده است . اگر این محل یک صفحه html باشد عنوان آن صفحه خواهد بود و اگر این محل یک فایل در شبکه باشد مسیر کامل آن فایل خواهد بود .

۵ – LocationURL : حاوی url محلی است که فعلاً در کنترل webbrowser، load شده است .

۶ – Offline : اگر کنترل webbrowser در حالت عدم اتصال باشد مقدار آن True و در غیراینصورت False است .

۷ – Parent : فرمی را نشان می دهد که کنترل webbrowser در آن قرار دارد .

۸ – ReadyState : وضعیت کنترل webbrowser را برمی گرداند .

متدهای کنترل webbrowser : این متدها مربوط به مرور در صفحات وب هستند :

۱ – GoBack : در لیست تاریخچه url ها ، یکی به عقب برمی گردد .

۲ – GoForward : در لیست تاریخچه url ها ، یکی به جلو می رود .

۳ – GoHome : به homepage مرورگر می رود .

۴ – Navigate : به یک url یا فایل می رود . ساختار این متد بصورت زیر است :

Navigate URL
Flags,][TargetFrameName,][PostData,][Headers]x]

URL آدرس مقصد می باشد . Flags نحوه باز شدن آدرس مقصد را تعیین می کند . اگر این پارامتر ذکر نشود آدرس جدید در پنجره فعلی باز خواهد شد و به لیست تاریخچه اضافه شده و اگر کپی آن در cache temperory موجود باشد از آنجا خوانده می شود . مقادیر پارامتر Flags عبارتند از :

– NavOpenInNewWindow : آدرس جدید را در پنجره جدیدی باز می کند .

– NavNoHistory : به لیست تاریخچه اضافه نمی شود بلکه جایگزین صفحه فعلی می شود .

– NavNoReadFromCache : صفحه جدید از cache خوانده نمی شود .

– NavNoWriteToCache : صفحه جدید روی cache نوشته نمی شود

Event های کنترل webbrowser : این event ها مربوط به مرور در وب و تغییر حالت آن هستند :

۱ – CommandStateChange : برای فعال یا غیرفعال کردن دکمه های Forward و Back در مرورگر استفاده می شود . شکل کلی فراخوانی این event بصورت زیر است :

Private Sub WebBrowser1_CommandStateChange(ByVal (Command As Long, ByVal Enable As Boolean

که command فرمانی است که حالت فعال آن تغییر کرده است و دو مقدار می گیرد : ۱ و ۳ که بترتیب معادل فرمانهای GoForward و GoBack هستند .

Enable فعال یا غیرفعال بودن فرمان را تعیین می کند .

۲ – DocumentComplete : این event زمانی فعال می شود که صفحه در حال load شدن به حالت ReadyState_Complete برود . شکل کلی فراخوانی این event بصورت زیر است :

Private Sub WebBrowser1_DocumentComplete(ByVal (Variant pDisp As Object, URL As

که pDisp ارجاعی به کنترل webbrowser است که event در آن رخ داده است و URL آدرس صفحه در حال load شدن است .

۳ – DownloadBegin : این event در آغاز حرکت به صفحه جدید روی می دهد و هیچ پارامتری نمی گیرد . مرورگر می تواند در این event پیغامی برای شروع عملیات جدید نشان می دهد .

۴ – DownloadComplete : این event در پایان عملیات یا در صورت انصراف کاربر یا بروز خطا روی می دهد .

۵ – ProgressChange : با بروز هر تغییری در وضعیت load ، این event روی می دهد . شکل کلی فراخوانی آن بصورت زیر است :

Private Sub WebBrowser1_ProgressChange(ByVal (Progress As Long, ByVal ProgressMax As Long

که Progress نشان دهنده پیشرفت عملیات (بایتهای load شده) است .

پارامتر ProgressMax تعداد کل بایتهایی که باید load شوند را نشان می دهد بنابر این :

(Progress/ProgressMax)*100=درصد پیشرفت عملیات load

یک مثال ساده :

از منوی project مورد components را انتخاب کنید و از لیست کنترلها ، Microsoft Internet Controls را به toolbar خود اضافه کنید . یک کنترل WebBrowser روی فرم قرار دهید و سایز آنرا به اندازه ابعاد فرم خود قرار دهید . یک textbox و یک دکمه روی فرم قرار دهید . کد زیر را برای event مربوط به کلیک دکمه بنویسید :

```
WebBrowser.Navigate textbox.text
```

کنترل Internet Transfer – قسمت اول

مقدمه : کنترل Internet Transfer نسبت به کنترل WebBrowser که در روزهای قبلی معرفی شد در سطح پایینتری قرار دارد . این کنترل با استفاده از دو پروتکل HTTP و FTP می تواند داده ها را منتقل کند . این کنترل زمانیکه از پروتکل HTTP استفاده می کند با همان روش کنترل WebBrowser به سرویس دهنده صفحات وب متصل می شود اما بجای آنکه صفحه وب را نمایش دهد متن Html صفحه را بازیابی می کند . همچنین زمانیکه این کنترل از پروتکل FTP استفاده می کند قادرست فایلها را بین کامپیوترهای روی شبکه منتقل سازد .

اتصالات HTTP : همانطور که می دانید ، پروتکل HTTP استاندارد وب

می باشد . صفحات وب با زبان Html نوشته می شوند و انتقال آنها از server به client توسط پروتکل HTTP صورت می گیرد .

متد OpenURL : ساده ترین راه استفاده از کنترل IT متد OpenURL است . شکل کلی این متد بصورت زیر است :

```
Inet.OpenURL(url,DataType)x
```

که url آدرس صفحه وب و DataType نوع داده بازیابی شونده است و دو مقدار icString (داده متنی) یا icByteArray (داده باینری) را می گیرد . مقدار بازگشتی این متد ، داده های منتقل شده است .

این متد بصورت سنکرون کار می کند یعنی در تمام مدت کار آن برنامه نمی تواند کار دیگری انجام دهد . اگر از icByteArray استفاده کنید باید مقدار بازگشتی آنرا در یک ارایه بایت قرار دهید .

مثال ۱ : از بخش Component در منوی Project مورد Microsoft Internet Transfer Control 6.0 را به toolbar خود اضافه کنید . سپس یک کنترل IT روی فرم قرار دهید و همچنین یک Rich Textbox و یک دکمه روی فرم قرار دهید و کد زیر را برای event مربوط به کلیک دکمه بنویسید :

```
TextBox.text=Inet.OpenURL("www.microsoft.com",icStri
ng)x
```

مثال ۲ : کد زیر داده های باینری را از اینترنت خوانده و آنها را در یک فایل ذخیره می کند :

```
byte Dim b() as
B()=Inet.OpenURL(ftp://ftp.microsoft.com/test.zip,icByte
Array)x
test.zip" For Access Write As #1\" & App.path Open
```

```
Put #1,b()x
١# Close
```

رویداد StateChanged : کنترل IT فقط یک event دارد که StateChanged می باشد . این event زمانی روی می دهد که State کنترل تغییر کند . State هر اتفاقی است که برنامه باید از آن مطلع شود . تعریف کلی این event بصورت زیر است :

```
Inet_StateChanged(ByVal NewState As Integer)x
```

که NewState مقداری است که حالت جدید را بیان می کند . مقادیر ممکن این پارامتر عبارتند از :

icNone : حالت تغییر نکرده است .

icResolvingHost : در حال جستجوی آدرس IP کامپیوتر موردنظر .

icHostResolved : آدرس IP کامپیوتر موردنظر یافت شد .

icConnecting : در حال اتصال به کامپیوتر مقصد

icConnected : اتصال به کامپیوتر مقصد برقرار شد .

icRequesting : در حال ارسال درخواست به کامپیوتر مقصد

icRequestSent : درخواست به کامپیوتر مقصد ارسال شد .

icReceivingResponse : در حال دریافت پاسخ از کامپوتر مقصد .

icResponseReceived : پاسخ کامپیوتر مقصد دریافت شد .

icDisconnecting : در حال قطع اتصال با کامپیوتر مقصد .

icDisconnected : اتصال مقصد با موفقیت قطع شد .

icError : در ارتباط با کامپیوتر مقصد خطایی رخ داده است .

icResponseCompleted : تکمیل پاسخ – تمام داده ها دریافت شد .

تشخیص خطا در عملیات انتقال داده اهمیت بالایی دارد و

StateChanged در صورت بروز هر خطایی مقدار icError را برمی گرداند و اطلاعات خطا را در دو خاصیت ResponseCode و ResponseInfo برمی گرداند .

انتقال داده بصورت آسنکرون : کنترل IT متدهای انعطاف پذیر دیگری هم دارد که آسنکرون هستند و اجازه می دهند تا همزمان با عملیات انتقال داده ، برنامه به وظایف دیگری هم بپردازد . این متدها با استفاده از Event Driven Model کار می کنند . بدین معنی که وقتی برنامه درخواست انتقال داده ای را می دهد کنترل IT درخواست را در زمینه برنامه انجام می دهد و برنامه ازاد است تا به کارهای دیگرش بپردازد . زمانیکه داده ها بازیابی شود ، داده ها را از بافر داخلی کنترل IT می خواند .

متد GetChunk : در عملیات انتقال آسنکرون ، بایستی داده را توسط این متد از بافر داخلی کنترل IT بگیریم :

```
Inet.GetChunk(datasize[,datatype])x
```

که پارامتر datasize از نوع long بوده و تعیین می کند چند بایت از بافر خوانده شود و پارامتر اختیاری datatype نوع داده را مشخص می کند و می تواند مقادیر icString و icByteArray را بگیرد .
زمانیکه StateChanged وارد حالتهای icResponseReceived و یا icResponseCompleted شد باید از GetChunk استفاده کنید . بدین صورت که از یک حلقه استفاده می کنیم تا کل بافر را بخوانیم :

```
Integer)x Private Sub Inet_StateChanged(Byval State as
Dim temp1,temp2
Select Case State
icResponseCompleted Case
temp1=””x
temp2=””x
```

```
Do
temp1=Inet.GetChunk(512,icString)x
temp1 & temp2=temp2
Loop Until temp1=””x
End Select
End Sub
```

برای بالابردن کارایی ، بهتر است از قطعات کوچک (بین ۵۱۲ تا ۱۰۲۴ بایتی) استفاده کنید .

متد Execute : و اما انعطاف پذیرترین متد کنترل IT ، متد Execute است . فرمت کلی این متد بصورت زیر است :

```
Inet.Execute(url,Command,Data,RequestHeaders)x
```

که url آدرس مقصد ، Command فرمانی است که به کامپیوتر مقصد داده می شود و Data و RequestHeaders اطلاعات اضافی لازم برای اجرای فرمان داده شده است . فرمانهای Command همان فرمانهای HTTP هستند که عبارتند از :

- GET : دریافت داده ها از کامپیوتر مقصد
- HEAD : دریافت اطلاعات header از کامپیوتر مقصد
- POST : ارسال اطلاعات لازم برای تکمیل درخواست
- PUT : ارسال فایل برای کامپیوتر میزبان (upload)

فرمان GET پرکاربردترین فرمان متد Execute است و داده های خوانده شده را در بافر داخلی بافر کنترل IT قرار می دهد تا بتوان با متد GetChunk آنها را بازیابی نمود .

مثال :

```
Inet.Execute http://www.microsoft.com,”GET”x
```

سایر خواص کنترل IT :

- AccessType : نوع دسترسی کنترل IT به اینترنت را مشخص می کند و سه مقدار می تواند بگیرد :

icUseDefault : استفاده از تنظیمات رجیستری برای دسترسی به اینترنت

icDirect : اتصال مستقیم کنترل IT به اینترنت

icNamedProxy : اتصال به اینترنت توسط پروکسی

- Document : نام صفحه پیش فرض که در متد Execute از آن استفاده می شود . اگر به متد Execute پارامتر url را ندهید از این صفحه پیش فرض استفاده می کند .

- Password : کلمه رمز عبور کامپیوتر میزبان FTP

- Procotol : نوع پروتکل مورد استفاده در متد Execute را مشخص می کند و ۵ مقدار می تواند بگیرد :

icUnknown : نامعلوم

icDefault : پروتکل پیش فرض

icFTP : پروتکل FTP

icHTTP : پروتکل HTTP

icHTTP : پروتکل حفاظت شده HTTP

- Proxy : نام میزبان پروکسی

- RequestTimeOut : مدت زمانی که کنترل IT صبر می کند تا اطلاعات را دریافت کند . اگر این خاصیت صفر باشد کنترل تا هر زمان که لازم باشد برای دریافت پاسخ صبر می کند . در حالت سنکرون (متد OpenURL) بعد از سپری شدن این مدت زمان ، یک خطا تولید می شود و در حالت آسنکرون (متد Execute) رویداد StateChanged مقدار خطا را بر می گرداند

- ResponseCode : بعد از بروز حالت icError این خاصیت کد خطا را می دهد .
- ResponseInfo : توضیحی درباره خطا
- StillExecuting : اگر True باشد یعنی کنترل مشغول انجام کار است .
- URL : آدرس مقصد در متدهای OpenURL و یا Execute
- UserName : نام کاربر برای ورود به کامپیوتر میزبان FTP

نکته ۱ : برای دریافت برنامه نمونه برای متد Execute با من تماس بگیرید .

نکته ۲ : موضوع روزهای بعد :

۱ – اتصالات FTP با استفاده از کنترل IT

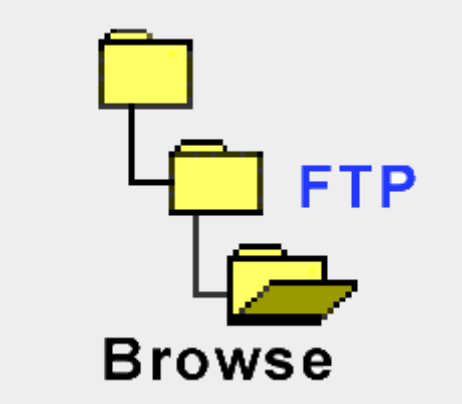
۲ – آشنایی با تکنیک Detection Collision در ساخت بازیهای دوبعدی

۳ – آشنایی با کنترل WinSock

نکته ۳ : شاید این سوال پیش بیاید که چرا همزمان با آموزش Internet Programming ، آموزش Game Programming را نیز شروع کرده ام ؟ علت اش اینست که تصمیم دارم پس از پایان یافتن این دو مبحث ، روش ساخت یک بازی دوبعدی چند نفره تحت شبکه را برایتان آموزش دهم . نظری ندارید ؟

کنترل Internet Transfer – قسمت دوم

اتصالات FTP



پروتکل FTP علاوه بر نقل و انتقال فایل بین دو کامپیوتر ، می تواند نوعی مدیریت فایل (مثل حذف فایل یا ایجاد پوشه) روی کامپیوتر مقصد را انجام دهد . FTP در انتقال فایل بسیار قویتر از HTTP است ولی به مراتب پیچیده تر از HTTP می باشد اما کنترل IT این پیچیدگیها را از دید برنامه نویس مخفی کرده است .

برای کار با سرورهای FTP باید به آنها Login نمود . نوع خاصی از Login به نام Anonymous Login (ورود ناشناس) وجود دارد که با آن کاربران می توانند بدون محدودیت از سایت FTP استفاده کنند .

توجه کنید که حتی برای ورود ناشناس هم نیاز به نام کاربر و کلمه عبور است . برای ارسال نام کاربر و کلمه عبور از خواص username و password کنترل IT استفاده می شود . اگر خاصیت username خالی باشد (blank) ، کنترل IT بطور خودکار از anonymous استفاده می کند و آدرس email کاربر بعنوان passowrd استفاده می شود .

استفاده از متد OpenURL : متد OpenURL ساده ترین راه انجام عملیات FTP است . دستور زیر از یک سایت FTP لیست می گیرد :

```
Text.text=Inet.OpenURL("ftp://ftp.microsoft.com",icStrin
g)x
```

برای خواندن فایل از یک سایت FTP باید در حالت باینری کار کرد :

b()=Inet.OpenURL("ftp://ftp.microsft.com/test.zip",icByte Array)x

استفاده از متد Execute : متد Execute قابلیتهای بیشتری دارد و اجرای آن در FTP نیاز به دو پارامتر دارد :

Inet.Execute(url,operation)x

که url آدرس سایت FTP بهمراه نام و مسیر فایل و پارامتر operation یک فرمان FTP است . کنترل IT با داده های خوانده شده FTP به دو طریق رفتار می کند :
برخی از داده ها مثل پاسخ فرمان DIR در بافر کنترل IT قرار می گیرد و باید آنها را با متد GetChunk خواند .
برخی دیگر از داده ها مثل فایل خوانده شده با فرمان GET مستقیماً روی دیسک نوشته می شوند و دیگر نیازی به استفاده از متد GetChunk نیست .
فرامین FTP بسیار قوی هستند و حتی به شما این امکان را می دهند که فایلها را به روی کامپیوتر مقصد کپی کنید ، به پوشه های کامپیوتر مقصد بروید ، فایلها را حذف کنید و یا تغییر نام دهید . البته باید توجه داشت که فرامین قابل اجرا به نوع ورود به سیستم FTP بستگی دارد . اگر با کاربر anonymous به یک سایت FTP وارد شوید تنها می تواند فایلها را ببیند و آنها را download کنید .
مهمترین فرامین FTP عبارتند از :
CD path : به دایرکتوری path می روید .
CDUP : به یک دایرکتوری بالاتر می رود .
CLOSE : بستن اتصال FTP

DELETE file1 : حذف فایل file1

DIR file1 : جستجوی فایل file1 روی دایرکتوری جاری

MKDIR path : ایجاد یک دایرکتوری با نام path

PUT file1 file2 : فایل file1 را از کامپیوتر مبدا روی فایل file2 در کامپیوتر مقصد کپی می کند .

PWD : نام دایرکتوری جاری در کامپیوتر مقصد را برمی گرداند .

QUIT : قطع اتصال FTP

GET file1 file2 : فایل file1 را از کامپیوتر مقصد روی فایل file2 در کامپیوتر مبدا کپی می کند .

file1 RENAME file2 : تغییر نام فایل file1 به file2

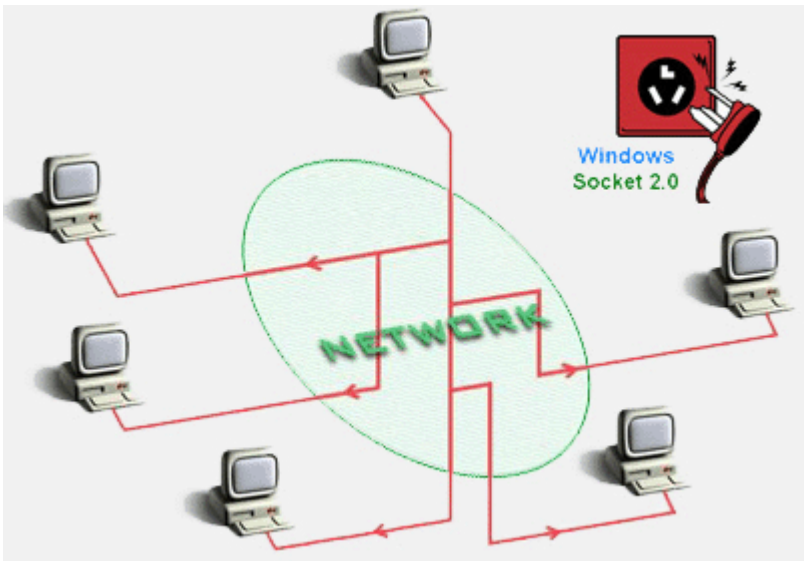
RMDIR path : حذف دایرکتوری path در کامپیوتر مقصد

SIZE file1 : بدست آوردن تعداد بایتهای فایل یا دایرکتوری file1

مثال :

```
test.zip")x Inet.Execute("ftp://ftp.microsoft.com","GET
```

کنترل WinSock – قسمت اول



مقدمه :

کنترل WinSock نسبت به تمام کنترلهای اینترنت در سطح پایینتری قرار دارد . این کنترل امکان ایجاد سرویسهای شبکه ای مبتنی بر پروتکلهای TCP و UDP را مهیا می کند . بعبارت دیگر توسط این کنترل می توان برنامه های کاربردی Client/Server (سرویس گیرنده / سرویس دهنده) ایجاد و با استفاده از پروتکل TCP و یا UDP بین آنها ارتباط برقرار نمود .

با تنظیم خصوصیات و فراخوانی متدهای این کنترل می توانید به راحتی به یک کامپیوتر راه دور متصل شوید و داده ها را در هر دو جهت جابجا نمائید . نمونه کاربرهایی که می توان با این کنترل ایجاد نمود : Client-server chat ، Mail client ، Mail server ، Proxy Server ، Network Game ، Port Scanner ، پیاده سازی الگوریتم های موازی و ...

مبانی TCP :

پروتکل کنترل اینترنت (Transfer Control Protocol) اجازه می دهد یک اتصال (Connection) را از طریق سوکت (socket) به یک کامپیوتر راه دور (Remote Computer) ساخته و استفاده کنید . با استفاده از این اتصال ، هر دو کامپیوتر می توانند داده ها را بین خودشان انتقال دهند . برقراری ارتباط از طریق TCP همانند صحبت کردن با تلفن است که باید حتماً اتصالی بین دو کامپیوتر صورت گیرد تا بتوانند با هم ارتباط برقرار کنند .

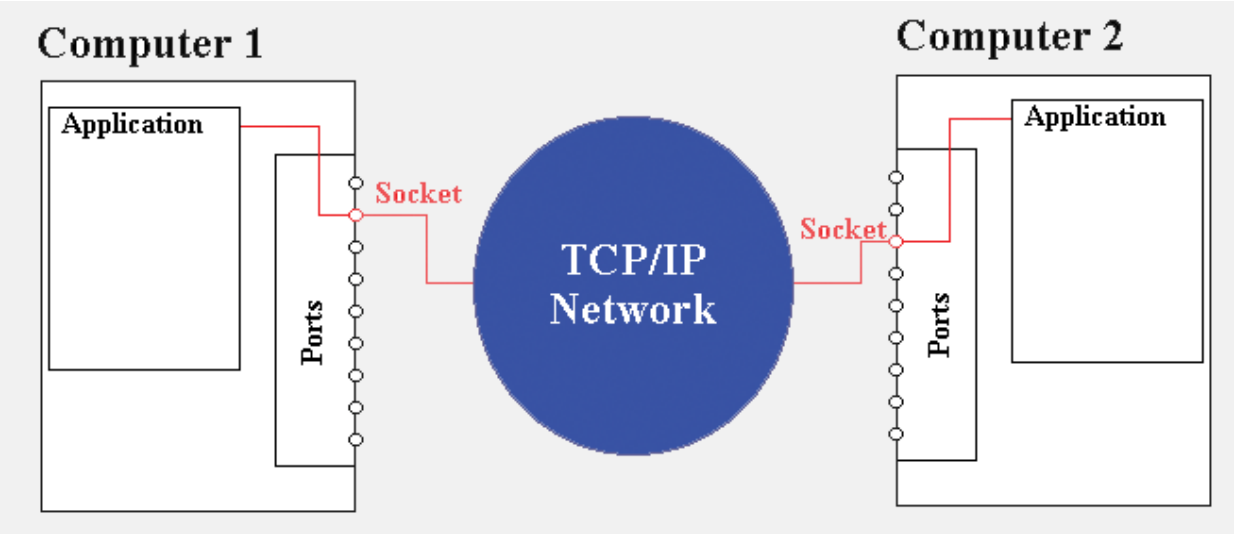
اگر یک برنامه Client می سازید بایستی بدانید که نام یا آدرس IP کامپیوتر Server چیست (Remote Host IP) و همچنین از طریق چه پورتی می توانید به آن متصل شوید (Port Remote) . حال بایستی به آن پورت Connect کنید .

همچنین اگر یک برنامه Server می سازید بایستی پورتی را که روی آن به درخواستها گوش می دهید مشخص کنید (LocalPort) و سپس به پورت گوش دهید (Listen) .

زمانیکه یک کامپیوتر Client تقاضای یک اتصال را می دهد Server این درخواست را Accept می کند .

زمانیکه یک اتصال ساخته می شود ، هر دو کامپیوتر می توانند داده را فرستاده و دریافت کنند .

!Error



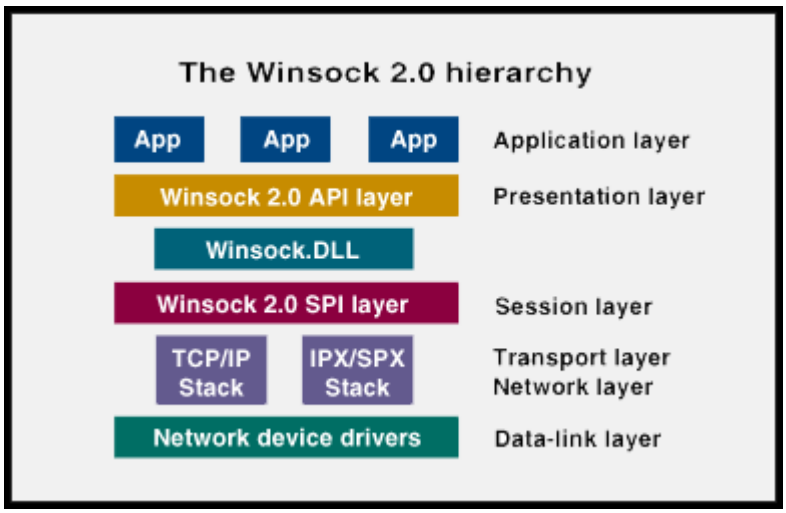
مبانی UDP :

پروتکل دیتاگرام کاربر (User Datagram Protocol) پروتکلی بدون اتصال (Connectionless) است . برخلاف TCP ، کامپیوترها نیاز به برپا کردن یک اتصال ندارند بنابراین یک برنامه می تواند یک client و یا یک server باشد . برقراری ارتباط در UDP شبیه ارسال نامه از طریق پست است .

برای انتقال داده توسط UDP ابتدا باید Local Port کامپیوتر Client

تنظیم گردد . کامپیوتر Server تنها بایستی RemoteHost را برابر آدرس کامپیوتر Client قرار دهد و همچنین Remote Port را همان Local Port کامپیوتر Client قرار دهد . سپس دو کامپیوتر می توانند داده ها را بین خود جابجا کنند .

استفاده از کنترل WinSock :



۱ – انتخاب پروتکل: در زمان استفاده از کنترل WinSock اولین کاری که باید انجام دهید انتخاب یکی از پروتکلهای TCP یا UDP است . طبیعت برنامه ای که شما می سازید نوع پروتکلی را که باید استفاده کنید مشخص می کند . چند سوال زیر به شما کمک می کند که پروتکل مورد نیازتان را انتخاب کنید :

– آیا برنامه شما در زمانیکه داده فرستاده می شود یا دریافت می شود نیاز به اطلاعاتی از طرف Server یا Client دارد ؟ اگر چنین است بایستی یک اتصال TCP قبل از ارسال یا دریافت داده ایجاد شود .

– آیا داده بسیار بزرگ است (مثل تصویر یا فایلهای صوتی) ؟ زمانیکه یک اتصال TCP ساخته می شود پروتکل TCP اتصال را باقی نگه می دارد و درستی ارسال داده تضمین شده است . این اتصال در هر حال به

منابع محاسباتی بیشتری نیاز دارد و بنابراین پرهزینه تر است .
- آیا داده متناوب ارسال می شود یا در یک نشست (Session) ارسال خواهد شد ؟ برای مثال اگر شما یک برنامه می سازید که کامپترهای مشخصی را در یک زمان خاص از انجام شدن عملیاتی مطلع می کند پروتکل UDP مناسب تر است . پروتکل UDP همچنین برای ارسال مقادیر کوچک داده ای مناست تر می باشد .

۲ – تنظیم پروتکل : برای تنظیم پروتکلی که می خواهید در برنامه تان از آن استفاده کنید در زمان طراحی برنامه خاصیت Protocol کنترل WinSock را برابر sckTCPProtocol و یا sckUDPProtocol قرار دهید . همچنین می توانید پروتکل خود را توسط کد زیر تنظیم کنید :

```
WinSock.Protocol=sckTCPProtocol
```

۳ – مشخص کردن نام کامپیوتان : برای اتصال به کامپیوتر راه دور بایستی آدرس IP و یا نام کامپوتر را بدانید .
نام کامپیوتر در Control Panel/Network/Identification موجود است . در صورتیکه می خواهید دو برنامه Client و Server خود را روی یک کامپیوتر تست کنید از آدرس IP 127.0.0.1 برای هر دو استفاده کنید اما اگر دو برنامه را روی دو کامپیوتر مجزا در شبکه قرار داده اید با اجرای دستور ipconfig در DOS Prompt می توانید آدرس IP کامپیوتر ها را بدست آورید .

۴ – ایجاد اتصال TCP : در زمان ساخت برنامه ای که از پروتکل TCP استفاده می کند ابتدا باید تصمیم بگیرید که این برنامه Client است یا Server . برای ساخت یک برنامه Server بایستی روی یک پورت خاص Listen کنید . زمانیکه Client تقاضای یک اتصال را می دهد ، برنامه Server می تواند آنرا Accept کند و بنابراین اتصال کامل شده است .

حال Client و Server می توانند با هم ارتباط داشته باشند .

مراحل زیر ساخت یک سرور چت ساده بر مبنای TCP را نشان می دهد :

- از منوی Project گزینه Components را انتخاب کنید و در لیست Component ها مورد Microsoft WinSock 6.0 را انتخاب کنید .
- یک کنترل WinSock در فرم خود قرار دهید و نام آنرا tcpserver بگذارید
- دو textbox با نامهای txtSendData و txtReceiveData و نیز یک دکمه در فرم قرار دهید .
- کد زیر را در رویداد Form_Load بنویسید :

```
Tcpserver.LocalPort=1000
tcpserver.Listen
```

- زمانیکه درخواستی از طرف Client می آید رویداد ConnectionRequest اجرا می شود . در این رویداد ابتدا باید چک کنید که حالت کنترل بسته باشد . اگر چنین نیست اتصال را قبل از پذیرفتن اتصال جدید ببندید . سپس تقاضا را بر اساس پارامتر requestID می پذیریم :

```
Private Sub tcpserver_ConnectionRequest(ByVal
(requestID As Long
tcpserver.Close sckClosed Then <> If tcpserver.State
tcpserver.Accept requestID
End Sub
```

- حال اتصال بین Client و Server برقرار شده است . کد زیر را برای event مربوط به کلیک دکمه Send بنویسید :

Tcpserver.SendData txtSendData.text

- اگر داده ای از طرف Client بیاید رویداد DataArrival اجرا می شود . کد زیر را برای این رویداد بنویسید :

```
As  Private Sub tcpserver_DataArrival(ByVal bytesTotal
(Long
Dim strData As String
strData tcpserver.GetData
txtReceiveData.Text = strData
End Sub
```

- کد زیر را برای رویداد Form_Unload بنویسید :

Tcpserver.Close

مراحل ساخت یک TCP Client بصورت زیر است :

- یک کنترل WinSock در فرم قرار دهید و نام آنرا tcpclient بگذارید .

- دو textbox با نامهای txtsend و txtreceive و نیز یک دکمه با نام sendدر فرم قرار دهید .

- یک دکمه با نام connect در فرم قرار دهید .

- کد زیر را برای متد Form_Load بنویسید :

```
tcpclient.RemoteHost=”yourservername”x
tcpclient.RemotePort=1000
```

- کد زیر را برای رویداد کلیک شدن دکمه connect بنویسید :

tcpclient.Connect

- کد زیر را برای رویداد کلیک شدن دکمه send بنویسید :

tctclient.SendData txtsend.Text

- کد زیر را برای رویداد DataArrival بنویسید :

As Private Sub tcpclient_DataArrival(ByVal bytesTotal (Long

Dim strData As String
strData tcpclient.GetData
txtreceive.Text = strData
End Sub

- کد زیر را باری رویداد Form_Unload بنویسید :

Tcpclient.Close

کدهای فوق یک سیستم Client-Server ساده را نشان می دهد . فایل exe هر دو برنامه را بسازید و آنها را اجرا کنید تا بتوانید سیستم خود را تست کنید .

۵ – پذیرفتن بیش از یک تقاضای اتصال : Server ای که در بالا ساخته شد تنها می تواند تقاضای یک اتصال را بپذیرد . با استفاده از ایجاد یک آرایه از کنترل WinSock می توان چندین تقاضای اتصال را پذیرفت . برای اینکار کافی است یک کپی (instance) از کنترل بسازیم (با تنظیم خاصیت Index) و متد Accept را برای instance جدید بکار ببریم . فرض کنید یک کنترل WinSock با نام sckServer در فرم داریم که خاصیت Index آنرا صفر قرار داده ایم . همچنین یک متغیر intMax از نوع Long تعریف می کنیم که تعداد اتصالات همزمان به Server را نگه می دارد . در event مربوط به Form_Load کد زیر را بنویسید :

```
intMax=0
sckServer(0).LocalPort=1000
sckServer(0).Listen
```

هر بار که تقاضای یک اتصال می رسد کد ابتدا تست می کند که مقدار Index چقدر است . اگر مقدار Index صفر باشد متغیر intMax یکی افزایش می یابد و از intMax برای ساخت یک instance جدید از کنترل استفاده می شود . حال از این instance برای پذیرفتن تقاضای اتصال استفاده می گردد . برای اینکار کد زیر را برای رویداد ConnectionRequest بنویسید :

```
Private Sub sckServer_ConnectionRequest(Index As
(Integer, ByVal requestID As Long
If Index = 0 Then
۱ + intmax = intmax
Load sckServer(intmax)x
۰ = sckServer(intmax).LocalPort
sckServer(Index).Accept requestID
End If
End Sub
```

۶ – ایجاد اتصال UDP : ساخت یک برنامه UDP ساده تر از برنامه های TCP است زیرا پروتکل UDP به اتصال نیاز ندارد . در برنامه TCP بالا یک کنترل WinSock بایستی حتماً Listen می کرد و یک کنترل دیگر یک اتصال را توسط متد Connect ایجاد نمود . در عوض پروتکل UDP نیازی به اتصال ندارد . برای ارسال داده بین دو کنترل WinSock سه مرحله بایستی انجام شود :

- پارامتر RemoteHost برابر نام کامپیوتر مقابل است .
- پارامتر RemotePort برابر پارامتر LocalPort کامپیوتر مقابل

- استفاده از متد Bind برای مشخص کردن LocalPort

چون هر دو کامپیوتر از نظر ارتباط مساوی هستند ، این نوع برنامه ها را Peer-to-Peer گویند . برای نمونه از کد زیر برای ساخت یک برنامه chat استفاده می کنیم :

- یک کنترل WinSock در فرم قرار دهید و نام آنرا udppeerA بگذارید .

- خاصیت Protocol آنرا UDPProtocol قرار دهید .

- دو textbox با نامهای txtsend و txtreceive و نیز یک دکمه در فرم قرار دهید .

- کد زیر را برای متد Form_Load بنویسید :

```
udppeerA.RemoteHost=”nameofpeerB”x
udppeerA.RemotePort=1001
۱۰۰۲ udppeerA.Bind
```

- کد زیر را برای event مربوط به کلیک دکمه بنویسید :

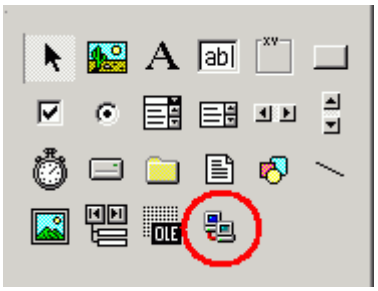
```
udppeerA.SendData txtsend.text
```

- کد زیر را برای رویداد DataArrival بنویسید :

```
Dim strData as String
strData udppeerA.GetData
txtreceive.Text=strData
```

برای ساخت UDP peerB مشابه مراحل بالا عمل کنید فقط خاصیت RemoteHost آنرا نام کامپیوتر PeerA و خاصیت RemotePort آنرا ۱۰۰۲ و خاصیت Bind آنرا ۱۰۰۱ قرار دهید .

کنترلWinSock – قسمت دوم



بررسی خواص کنترل WinSock :

ByteReceived : مقدار داده دریافت شده (موجود در بافر receive) را نشان می دهد . توسط متد GetData می توان این داده را دریافت نمود .

LocalHostName : نام ماشین محلی را نشان می دهد . این پارامتر فقط خواندنی است .

LocalIP : آدرس IP ماشین محلی را بصورت یک string برمی گرداند . این پارامتر فقط خواندنی است .

LocalPort : برای خواندن و یا تنظیم شماره پورت محلی بکار می رود .

Protocol : برای خواندن و یا تنظیم پروتوکل مورد استفاده توسط کنترل WinSock بکار می رود .

RemoteHost : برای خواندن و یا تنظیم نام یا آدرس IP ماشین راه دور بکار می رود .

RemoteHostIP : آدرس IP ماشین راه دور را برمی گرداند :

۱- برای برنامه های Client بعد از زمانیکه یک اتصال توسط متد Connect پذیرفته شد ، این خاصیت حاوی آدرس IP ماشین راه دور است .

۲ – برای برنامه Server ، بعد از آمدن یک Request Connection این خاصیت شامل آدرس IP ماشین راه دور است .

۳ – در زمان استفاده از پروتکل UDP بعد از اینکه رویداد Data

Arrival رخ داد این خاصیت حاوی آدرس IP ماشینی است که داده را فرستاده .

RemotePort : برای خواندن و یا تنظیم شماره پورت ماشین راه دوری که می خواهید به آن متصل شوید بکار می رود .

SocketHandle : مقداری را برمی گرداند که مرتبط با سوکتی است که کنترل WinSock را مدیریت می کند و برای ارتباط با لایه WinSock بکار می رود . این پارامتر فقط خواندنی است و تنها برای ارسال به API های WinSock طراحی شده است .

State : وضعیت کنترل WinSock را نشان می دهد . وضعیتهای ممکن برای State عبارتند از :

۱ – sckClosed : اتصال بسته است .

۲ – sckOpen : اتصال باز است .

۳ – sckListening : حالت گوش دادن به پورت

۴ – sckConnectionPending : معلق شدن اتصال

۵ – sckResolvingHost : تصمیم گیری در مورد میزبان

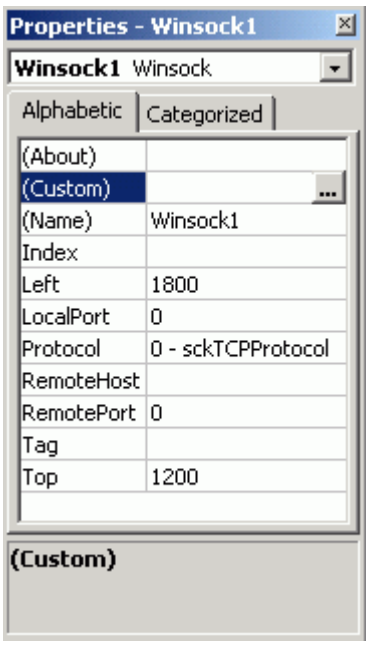
۶ – sckHostResolved : در مورد میزبان تصمیم گیری شد .

۷ – sckConnecting : حالت برقراری ارتباط

۸ – sckConnected : ارتباط برقرار شد .

۹ – sckClosing : حالت قطع اتصال

۱۰ – sckError : حالت خطا



بررسی متدهای کنترل WinSock :

متد Accept : تنها برای برنامه های TCP Server بکار می رود . این متد برای پذیرفتن یک اتصال در زمان مدیریت رویداد ConnectionRequest استفاده می شود .

متد Bind : این پارامتر LocalPort و LocalIP یک اتصال را مشخص می کند .

متد Close : برای بستن یک اتصال TCP و یا بستن یک listening socket بکار می رود .

متد GetData : بلوک جاری داده دریافت شده را گرفته و آنرا در متغیری از نوع Variant ذخیره می کند . شکل کلی این متد بصورت زیر است :

WinSock.GetData data[,type][,maxlen]x

که data داده دریافتی است . اگر داده کافی موجود نباشد data برابر empty خواهد بود .

type نوع داده دریافتی است که می تواند مقادیر زیر باشد :

vbByte – – vbInteger - vbLong - vbSingle - vbDouble -

vbError - vbBoolean - vbDate – – vbString -

vbByte+vbArray

maxlen حداکثر سایز را در زمان دریافت یک byte Array و یا یک string مشخص می کند .

متد Getdata در رویداد Data Arrival استفاده می شود که این رویداد یک پارامتر با نام TotalBytes دارد . اگر maxlen ای که شما تعیین کرده اید کمتر از TotalBytes باشد پیغام هشدار شماره ۱۰۰۴۰ دریافت می کنید بدین معنی که بایتهای باقیمانده گم خواهند شد .

متد Listen : یک سوکت می سازد و آنرا در حالت Listen قرار می دهد . این متد تنها در اتصالات TCP بکار میرود .

متد PeekData : مشابه GetData است با این تفاوت که داده را از صف ورودی حذف نمی کند . این متد تنها برای اتصالات TCP بکار می رود .

متد SendData : برای ارسال داده به کامپیوتر راه دور بکار می رود .

بررسی event های کنترل WinSock :

رویداد Close : زمانی رخ می دهد که کامپیوتر راه دور اتصال را ببندد .

رویداد Connect : بعد از اینکه یک اتصال به Server ایجاد شد روی می دهد . شکل کلی آن بصورت زیر است :

```
Private Sub WinSock_Connect(ErrorOccurred As
Boolean)x
```

که پارامتر ErrorOccurred دو مقدار دارد : اگر True باشد یعنی اتصال Fail شده است و اگر False باشد یعنی اتصال با موفقیت انجام شده است .

با رویداد Connect می توانید error هایی که در زمان فرایند باز کردن

اتصال برگردانده شده را چک کنید .

رویداد ConnectionRequest : زمانی رخ می دهد که یک کامپیوتر راه دور تقاضای یک اتصال را بدهد . این رویداد فقط برای برنامه های TCP Server بکار می رود .

رویداد DataArrival : زمانی رخ می دهد که داده جدیدی بیاید .

رویداد Error : زمانی رخ می دهد که یک خطا در فرایند ارتباط رخ دهد (مثلاً Failed to Connect و یا Failed to Send) . شکل کلی آن بصورت زیر است :

```
as  Private WinSock_Error(number as Integer,description
String,scode as Long,source as String,helpfile as
Long,canceldisplay as Boolean)x String,helpcontext as
```

number شماره کد خطا است .

description توضیحی در مورد خطا است .

source توصیف منبع خطا

canceldisplay : مشخص می کند آیا پیغام خطای پیش فرض نشان داده شود یا نه

رویداد SendComplete : زمانی رخ می دهد که یک عمل Send تکمیل شده باشد .

رویداد SendProgress : زمانی رخ می دهد که کنترل شروع به ارسال داده نماید . شکل کلی آن بصورت زیر است :

```
,WinSock_SendProgress (bytesSent As Long
bytesRemaining As Long)x
```

که bytesSent تعداد بایتهای ارسال شده و bytesRemaining تعداد بایتهای باقیمانده است

آشنایی با کتابخانه Windows Packet Capture – بخش اول

مقدمه :

برای آشنایی با مبانی شبکه های TCP/IP به بحث مروری بر TCP/IP مراجعه کنید .

معرفی :

کتابخانه WinPcap یک معماری برای استخراج Packet های TCP/IP و آنالیز شبکه در محیطهای ۳۲ بیتی ویندوز می باشد . این کتابخانه شامل سه بخش است :

۱ – یک فیلتر Packet در سطح هسته سیستم عامل (Kernel)

۲ – یک کتابخانه dll سطح پایین (low-level) با نام packet.dll

۳ – یک کتابخانه مستقل از سیستم عامل و سطح بالا (high-level) با نام wpcap.dll

فیلتر packet یک درایور دستگاه (device driver) است که به

ویندوزهای ۹۵ ، ۹۸ ، ME، NT و ۲۰۰۰ قابلیت استخراج و capture کردن و نیز ارسال داده خام (raw data) از یک کارت شبکه را می دهد . همچنین این امکان را دارد که packet های capture شده را در یک بافر ذخیره کند و یا آنها را فیلتر نماید .

packet.dll یک API است که بمنظور دسترسی مستقیم به عملکرد درایور packet استفاده می شود . بنابراین packet.dll یک واسط برنامه نویسی مستقل از سیستم عامل های مایکروسافت را مهیا می کند .

Wpcap.dll مجموعه ای از ابزارهای سطح بالای اصلی برای capture را مهیا می کند که این توابع با کتابخانه libpcap (کتابخانه capture در سیستم عامل UNIX) سازگار می باشند . این توابع اجازه capture کردن packet ها را با روشی مستقل از سخت افزار شبکه و مستقل از سیستم عامل مهیا می کنند .

دریافت کتابخانه WinPcap :

نسخه WinPcap 2.2 :

برای دریافت برنامه auto-installer (شامل درایور و DLL های مربوطه) برای سیستم های Windows 95/98/ME/NT/2000 به آدرس زیر مراجعه کنید :

(auto-installer (driver +DLLs WinPcap

برای دریافت بسته توسعه دهنده این کتابخانه که شامل برنامه های نمونه ایجاد شده توسط packet capture driver و packet.dll و libpcap است ، به آدرس زیر مراجعه کنید :

pack Developer's

برای دریافت source code کتابخانه WinPcap به آدرس زیر مراجعه

کنید :

source code WinPcap

آشنایی با کتابخانه Windows Packet Capture - بخش دوم

پاسخ به سوالاتی در مورد کتابخانه WinPcap :

۱ - WinPcap چگونه روی سیستم نصب می شود ؟

در صورت اجرای فایل نصب ، فایل درایور و Dll های مربوطه در دایرکتوری سیستم شما کپی می شوند . همچنین در Control Panel بخش Add/Remove Program می توانید عبارت WinPcap را مشاهده کنید .

۲ - آیا WinPcap 2.2 آخرین نسخه WinPcap است ؟

خیر . آخرین نسخه WinPcap نسخه ۲,۳ می باشد که از آدرس زیر قابل دریافت است :

Support Windows 95/98/ME/NT/2000/XP
auto-installer:driver +DLLs WinPcap
pack Developer's
code WinPcap source

همچنین WinPcap 3.0 Alpha 4 را می توانید از آدرس زیر دریافت کنید :

auto-installer:driver +DLLs WinPcap

۳ - چگونه می توان فهمید که WinPcap در حال اجرا روی یک کامپیوتر Win2k/XP است ؟

در بخش Run دستور msinfo32 را وارد کنید . سپس در بخش

Software Environment و سپس بخش System Drivers بایستی عبارت NPF را ببینید .

۴ – برنامه های بر مبنای WinPcap (مثل Windump که بعداً در مورد آن صحبت خواهم کرد) بدرستی اجرا نمی شوند . آیا اشکال از WinPcap است ؟

برنامه Windump را نصب کنید . دستور windump -D لیست آداپتورهای مجاز را گزارش می دهد و همچنین نشان می دهد که آیا WinPcap قادر به شناسایی درست سخت افزار شما بوده است یا نه . اگر Windump درست کار کند ، اشکال از برنامه شماست نه از WinPcap .

۵ – آیا می توان از WinPcap در اتصالات PPP (اتصالات مودمی) استفاده کرد ؟

در ویندوز ۹۵ بخاطر باگی که در NDIS است WinPcap گاهی اوقات اتصال PPP را reset می کند . در ویندوزهای ۹۸ و ME این مشکل وجود ندارد اما گاهی اوقات قادر به ارسال packet نیست . در ویندوزهای NT/2k/XP نیز برخی مشکلات در فرایند binding وجود دارد که مانع از درست کار کردن درایور پروتکل روی آداپتور WAN می شود . این مشکل بخاطر درایور PPP با نام ndiswan است که یک واسط استاندارد را برای capture کردن مهیا نمی کند .

۶ – Security در WinPcap چگونه است ؟

مدل Security این برنامه ضعیف است و نویسندگان این برنامه درحال کار روی آن هستند .

۷ – آیا از WinPcap می توان در محیطهای برنامه نویسی Borland استفاده کرد ؟

برای اینکه بتوان از امکانات WinPcap در C++ Builder استفاده کرد بایستی توسط برنامه COFF2OMF.EXE که در دایرکتوری Borland وجود دارد ، کتابخانه های Packet.lib و wpcap.lib که بر اساس استانداردهای مایکروسافت (استاندارد COFF) هستند را به استاندارد OMF تبدیل کنید مثال : COFF2OMF input.lib output.lib

۸ – آیا می توان از WinPcap در ویژوال بیسیک استفاده کرد ؟

بطور مستقیم نمی توان از این کتابخانه در VB استفاده کرد . شرکت BeeSync یک کنترل ActiveX با نام PacketX ایجاد کرده که بوسیله آن می توان از تواناییهای WinPcap در ویژوال بیسک یا هر محیط دیگری که Microsoft ActiveX technology را پشتیبانی کند استفاده کرد .

بحث در مورد این ActiveX موضوع بخش بعدی این سلسه مقالات می باشد .

۹ – آیا WinPcap می تواند با فایروال کار می کند ؟

ممکنست گاهی اوقات کار نکند .

۱۰ – آیا می توان از WinPcap در جاوا استفاده کرد ؟

WinPcap بطور مستقیم از جاوا پشتیبانی نمی کند اما در این آدرس می توانید یک Java wrapper را برای اینکار دریافت کنید .

۱۱ – آیا WinPcap از device های loopback پشتیبانی می کند ؟

خیر و علت آن بخاطر محدودیتهای ویندوز است .

۱۲ – WinPcap از چه سخت افزارهایی پشتیبانی می کند ؟
driver NPF device طوری توسعه یافته که بتواند با آداپتورهای اترنت درست کار کند اما نمی تواند بدرستی با آداپتورهای PPP WAN (مودمها) کار کند . همچنین نبایستی با کارتهای FDDI, ARCNET, ATM و Token Ring مشکلی داشته باشد اما این امر بطور قطعی تائید شده نیست . با آداپتورهای Wireless نیز نمی تواند درست کار کند و در بهترین حالت قادرست یک Ethernet emulation را ببیند .

۱۳ – آیا می توان توسط WinPcap ، بسته های اطلاعاتی ورودی را Drop کرد ؟
خیر فقط می توان بسته ها را capture کرد .

۱۴ – آیا می توان توسط امکانات WinPcap یک Firewall ساخت ؟
همانطور که در بالا گفته شد WinPcap قابلیت Drop کردن بسته ها را ندارد و تواناییهای Filtering در WinPcap تنها روی بسته های sniff شده است . برای جلوگیری کردن از ورود یک بسته به پشته TCP/IP بایستی خودتان یک درایور میانی ایجاد کنید .

۱۵ – چگونه می توان کاری کرد که WinPcap بطور اتوماتیک در زمان بوت سیستم فعال گردد ؟
مقدار کلید زیر را در رجیستری از x3۰ به x2۰ تغییر دهید . این کار فقط در ویندوزهای NTx جواب می دهد :
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NPF\Start

۱۶ – آیا با کامپایل مجدد سورس WinPcap نتیجه درستی حاصل می شود ؟

بله اما ابتدا بایستی Service Pack 5 برنامه Microsoft Visual Studio را از سایت مایکروسافت دریافت و آنرا نصب کنید .

۱۷ – آیا درایورهای دیگری بجز WinPcap برای کار با packet های TCP/IPوجود دارد ؟

بله . برای مثال برنامه Zx Sniffer یک کتابخانه capture کردن packet دارد که مشابه WinPcap است .

۱۸ – آیا Document ها و Tutorial های کاملی برای کار با Packet Driver API و Packet.dll و wpcap.dll در ویژوال سی وجود دارد ؟

بله . برای اینکار می توانید از مطالب زیر استفاده کنید :

How to use wpcap.dll
specific extensions wpcap
low level capture library Packet Driver API. How to use the
of how to write and compile a Instructions and examples driver's capture application using libpcap calls or packet (PACKET.DLL) calls
Compiling WinPcap

آشنایی با کتابخانه Windows Packet Capture – بخش سوم

بررسی یکی از برنامه های نوشته شده با WinPcap : معرفی WinDump

مقدمه :

برنامه WinDump نسخه تحت ویندوز برنامه مشهور TcpDump می باشد که در سیستم عامل UNIX وجود دارد . این برنامه یکی از بهترین ابزارهای Network Sniffer/Analyzer می باشد .
WinDump کاملاً با TcpDump سازگار می باشد و می توان از آن برای مانیتور کردن ترافیک شبکه بر اساس قوانین مختلف و پیچیده ای که قابل تنظیم هستند ، استفاده کرد .
WinDump روی سیستم عاملهای ME/۹۸/۹۵ و نیز NT/2000/XP قابل استفاده می باشد .
WinDump از کتابخانه libpcap موجود در WinPcap استفاده می کند .

دریافت برنامه :

برای دریافت نسخه ۳,۶,۲ به آدرس زیر مراجعه کنید :
WinDump.exe
source code WinDump
نکته : قبل از اجرای این برنامه بایستی برنامه WinPcap ۲,۳ را روی سیستم خود نصب کنید .
برای دریافت نسخه alpha ۳,۸ به آدرس زیر مراجعه کنید :
WinDump.exe
code WinDump source
نکته : قبل از اجرای این برنامه بایستی برنامه WinPcap ۳,۰ را روی سیستم خود نصب کنید .

پاسخ به سوالاتی در مورد برنامه WinDump :

۱ – چگونه می توان با استفاده از WinDump لیست آداپتورهای شبکه را بدست آورد ؟

با اجرای دستور WinDump –D

توسط دستور WinDump –i adaptername می توان WinDump را روی آداپتور خاصی اجرا کرد .

۲ – آیا می توان از WinDump روی اتصالات PPP (اتصالات مودمی) استفاده کرد ؟

WinDump از همان device هایی پشتیبانی می کند که WinPcap می کند . به بخش قبل مراجعه کنید .

۳ – چگونه می توان اطلاعاتی در مورد TcpDump بدست آورد ؟

به آدرس زیر مراجعه کنید :

http://www.tcpdump.org/

۴ – با وجود نصب آخرین نسخه برنامه WinDump ، چرا این برنامه قادر به capture کردن همه packet ها نیست ؟

توسط سوئیچ B می توانید سایز بافر درایور را افزایش دهید برای مثال دستور Windump –B 5000 سایز بافر را ۵ مگابایت می کند . سایز بافر برنامه در حالت عادی ۱ مگابایت است .

۵ – آیا می توان چندین WinDump را روی یک ماشین اجرا کرد ؟

بله

۶ – چرا WinDump در زمان capture کردن برای مدتی hang می کند ؟

علت آن فراخوانی تابع gethostbyaddr در برنامه می باشد که برای استخراج name host ها بکار می رود . می توان توسط سوئیچ n- از name resolution جلوگیری کرد .

۷ – آیا اطلاعات کاملی در مورد برنامه WinDump وجود دارد ؟
به آدرس زیر مراجعه کنید :
Manual WinDump

آشنایی با PacketX

مقدمه

PacketX مجموعه ای از کلاسهای اکتیو ایکس است که امکانات WinPcap را در ویژوال بیسیک و هر زبان برنامه نویسی دیگری که از تکنولوژی ActiveX Microsoft پشتیبانی کند مهیا می کند .
بطور خلاصه PacketX از WinPcap برای capture کردن و فیلتر کردن packet های شبکه استفاده می کند . علاوه بر capture استاندارد ، شما می توانید از PacketX برای گردآوری اطلاعات ترافیک شبکه و ارسال raw packet استفاده کنید .

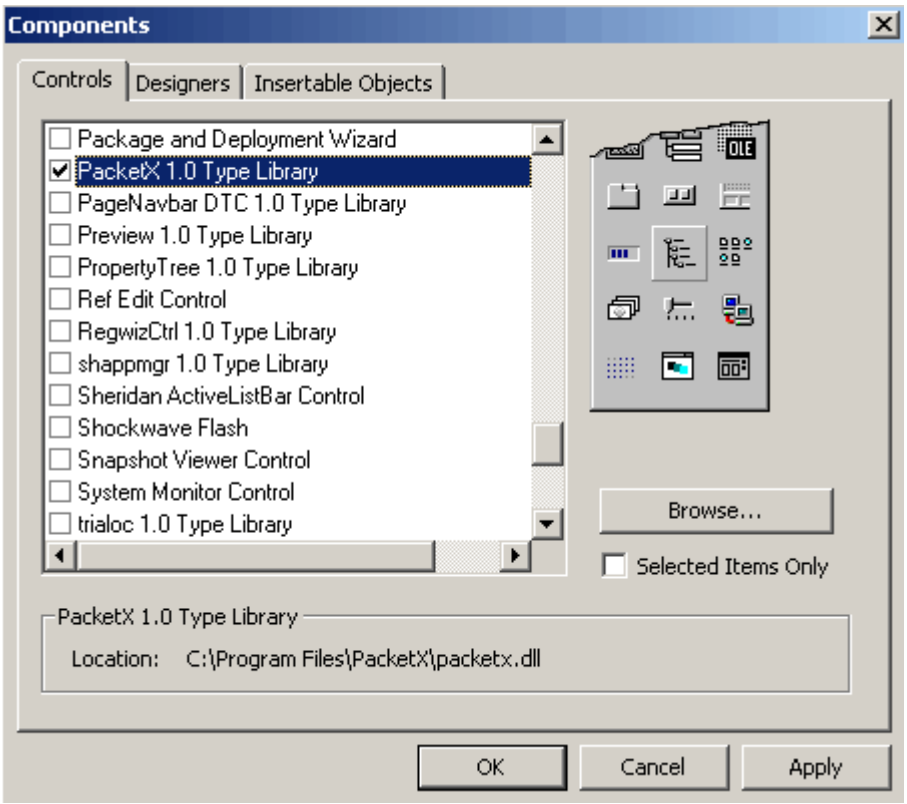
دریافت PacketX

برای دریافت نسخه ۱,۳ این کتابخانه به آدرس زیر مراجعه کنید :
PacketX 1.3 Download
قبل از استفاده از PacketX بایستی WinPcap 2.3 را که آدرس آن در

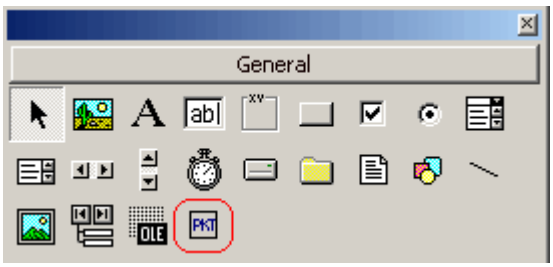
بخش قبل گفته شد دریافت و نصب نمائید .

استفاده از PacketX

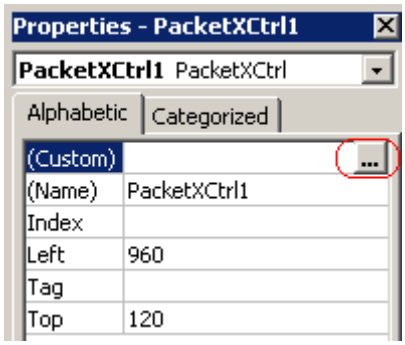
با یک مثال ساده سعی می کنم روش استفاده از این کتابخانه را به شما آموزش دهم .
پس از نصب PacketX وارد محیط ویژوال بیسیک شده و از منوی Project مورد Components را انتخاب کنید .



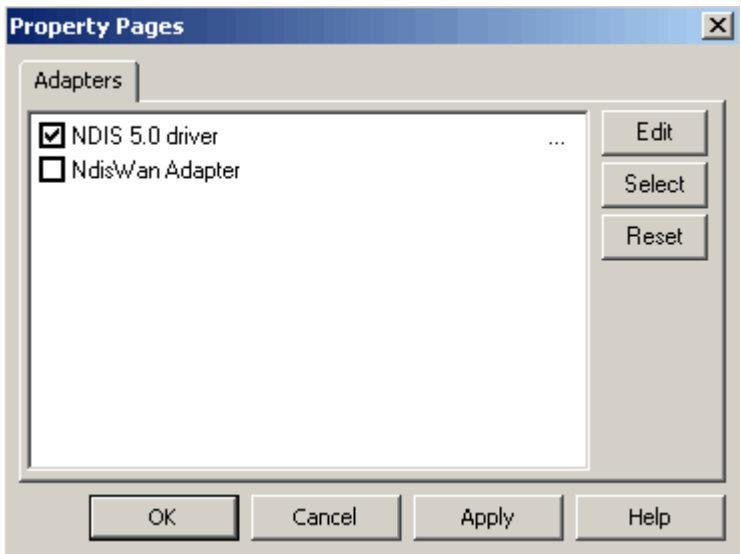
سپس مورد PacketX Type Library را انتخاب کنید تا به ToolBar اضافه شود .



این ActiveX را در فرمتان قرار دهید و سپس در قسمت properties آن روی Custom کلیک نموده و نوع آداپتور را مشخص نمائید .



پس از انتخاب آداپتور با کلیک روی دکمه Edit می توانید اطلاعاتی از قبیل مد کاری و سایر بافر و نوع فیلترینگ را مشخص نمائید .



نام این شی را نیز PacketX1 می گذاریم .
در فرمتان دو دکمه به نامهای Start و Stop قرار دهید . همچنین یک listBox با نام IPList در فرمتان قرار دهید .
در کد مربوط به Form_load عبارت زیر را بنویسید :

```
PacketX1.Adapter.BPFilter = "port 80"x
```

عبارت فوق مشخص می کند که می خواهیم packet های پورت ۸۰ را capture کنیم .
کد زیر را برای event مربوط به کلیک شدن دکمه Start بنویسید :

```
PacketX1.start
```

کد زیر را برای متد OnPacket مربوط به PacketX1 بنویسید :

```
Private Sub PacketX1_OnPacket(ByVal pPacket As
PACKETXLibCtl.IPktXPacket)x
IPList.AddItem (pPacket.DestIpAddress)x
Sub End
```

کد فوق آدرس IP مقصد packet ها را به یک لیست اضافه می کند . شما می توانید اطلاعات دیگری از قبیل داده موجود در packet ، سایز داده ، آدرس مبدا ، تاریخ ارسال داده و غیره را استخراج کنید .
کد زیر را برای event مربوط به کلیک شدن دکمه Stop بنویسید :

```
PacketX1.stop
```

برای اینکه بتوانید در یک مدت خاص وضعیت دریافت و ارسال packet را بدست آورید باید ابتدا توسط دستور زیر مد آداپتور را در حالت Statistic قرار دهید :

```
PacketX1.Adapter.Mode=PktXModeStatistics
sec ۲ //' PacketX1.Adapter.ReadTimeout=2000
```

سپس PacketX1 را start نموده و توسط متد OnStatistics اطلاعات مربوطه را استخراج کنید .

برای مشاهده اطلاعات کامل در مورد ساختار PacketX و ماژولهای آن به آدرس زیر مراجعه کنید :
Documentation PacketX

ایجاد کلاسی برای کار با فایلهای XML

ایجاد کلاسی برای کار با فایلهای XML

مقدمه :

همانطور که در قسمت سوم مباحث برنامه نویسی اکتیوایکس های سرور سایت گفته شد برای قرار دادن پارامترهای اتصال به بانک اطلاعاتی از فایل XML استفاده خواهیم نمود . بنابراین ابتدا بایستی

کلاسی برای کار با فایلهای XML بنویسیم . توجه داشته باشید که کلاسی که در این بخش معرفی می شود کلاسی ساده می باشد که فقط با آن می توان مقدار یک ند Node وجود در فایل xml را خواند . در صورت نیاز ، می توانید خودتان متدهای دیگری را به آن اضافه کنید . برای این منظور نکاتی را در انتهای همین بخش آورده ام .

XML یک زبان نشانه ای توسعه پذیر (eXtensible Markup Language) است که در سال ۱۹۹۸ توسط کنسرسیوم وب جهانی W3C ایجاد شد . XML واقعاً یک زبان نیست بلکه یک متا-زبان است و برای توصیف سایر زبانها بکار می رود . داده ها در فایلهای XML براحتی قابل تعریف و استفاده هستند .

مثالی از یک فایل XML :

```
<user>
<name/>ali<name>
<id/>۱۲<id>
<user/>
```

کار با فایلهای XML در وی بی :

برای کار با فایلهای xml در ویژوال بیسیک بایستی ابتدا از بخش References مورد Microsoft XML 3.0 را انتخاب کنید . سپس یک Modules Class به پروژه تان اضافه کنید و نام آنرا XMLReader بگذارید . در این کلاس ابتدا یک متغیر از نوع شی xml برای کار با فایلهای xml تعریف می کنیم :

```
Private xml
```

سپس متدی برای مقداردهی اولیه شی xml می نویسیم . این متد دارای یک متغیر ورودی است که نام فایل xml مورد نظر می باشد :

```
String)x Public Sub Initiate(ByVal filename As
Set xml = CreateObject("Microsoft.XMLDOM")x
xml.async = False
xml.Load (server.MapPath(filename))x
End Sub
```

توجه کنید که در کد فوق از شی server برای یافتن مسیر فیزیکی فایل XML استفاده شده است بنابراین ابتدا بایستی در Class_Initialize این شی را مطابق مطالب درس دوم مقداردهی کنید .

حال بایستی متدی برای خواندن مقدار یک ند از فایل xml بنویسیم . در این متد توسط یک حلقه for each ندهای فایل را بررسی می کنیم تا ندی را بیابیم که نامش مشابه با متغیر ورودی متد است . سپس با استفاده از خاصیت nodeValue می توانیم مقدار آنرا بخوانیم .

```
Public Function getvalue(ByVal NName As String) As
String
Dim x
getvalue = ""x
xml.documentElement.childNodes For Each x In
If x.nodeName = NName Then
x.childNodes(0).nodeValue = getvalue
Exit For
End If
Next
Function End
```

مثالی از کار با کلاس XMLReader :

همانطور که گفته شد می توانیم پارامترهای اتصال به بانک اطلاعاتی را در فایل XML قرار دهیم و در زمان Initiate کردن ADODB برای اتصال به بانک اطلاعاتی ، آنها را بخوانیم :

```
Dim xmlf As New XMLReader
xmlf.Initiate("config.xml")x Call
xmlf.getvalue("DataBaseID")x = userName
xmlf.getvalue("DataBasePassword")x = Password
xmlf.getvalue("DataBaseName")x = database_name
xmlf.getvalue("ServerAddress")x = server_name
```

ساختار یک فایل نمونه config.xml بصورت زیر می باشد :

```
<Application/>testIt<Application>
<ServerAddress/>۱۹۲,۱۶۸,۰,۱<ServerAddress>
<DataBaseName/>Edatabase<DataBaseName>
<DataBaseID/>Euser<DataBaseID>
<DataBasePassword/>Epass<DataBasePassword>
```

سایر نکات برای توسعه کلاس فوق :

نکات زیر شما را در نوشتن کلاسی کاملتر راهنمایی می کنند :

۱ – توجه داشته باشید که xml.documentElement بعنوان ریشه

فایل xml محسوب می شود . بنابراین برای دسترسی به ریشه می توان یک شی ریشه نیز تعریف کرد :

```
Dim root
Set root = xml.documentElement
```

۲ – در صورتیکه یک فایل xml دارای چندین ند در ریشه اش باشد و هر ند ریشه نیز دارای چندین ند درونی باشد توسط خاصیت root.childNodes.length و با استفاده از یک حلقه for می توان به این ندها دسترسی داشت . برای مثال فایل زیر را درنظر بگیرید :

```
<people>
<user>
<name/>ali<name>
<id/>۱<id>
<user/>
<user>
<name/>reza<name>
<id/>۲<id>
<user/>
<people/>
```

حلقه زیر روش دسترسی را به این فایل نشنان می دهد :

```
For I = 0 TO (root.childNodes.length - 1)x
thisChild = root.childNodes(I)x Set
name = thisChild.childNodes(0).Text
```

```
thisChild.childNodes(1).Text = id
Next
```

۳ – اضافه کردن ند به فایل : برای اضافه کردن ند از متدهای createNode و appendChild استفاده می شود برای مثال برای اضافه کردن یک user جدید به مثال فوق :

```
x("" ,"Set newuser = xml.createNode("element", "people
Dim name,id
x("" ,"Set newname = xml.createNode("element", "name
newname.text = yourname
x("" ,"Set newid = xml.createNode("element", "id
yourid = newid.text
newuser.appendChild(newname)x
newuser.appendChild(newid)x
root.appendChild(newuser)x
```

در انتها نیز بایستی فایل را ذخیره نمود :

```
xml.save(Server.Mappath(filename))x
```

۴ – حذف یک ند از فایل : برای حذف یک ند از فایل توسط یک حلقه for بایستی ند مورد نظر را یافته و سپس توسط متد removeChild آنرا حذف کنیم :

```
found = False
TO 0 STEP -1 (۱ – For I = (root.childNodes.length
Set thisChild = root.childNodes(I)x
```

```
thisChild.childNodes(0).Text = name
If name = searchname Then
root.removeChild(thisChild)x
found = True
End If
Next
```

در انتها نیز فایل xml را ذخیره کنید .

آشنایی با BitBlt

آشنایی با BitBlt

هدف از این مبحث آموزشی ، آشنایی با تابع BitBlt و برخی دیگر از توابع کتابخانه Win32 GDI برای انجام برخی عملیات گرافیکی مثل double buffering و خواندن sprite از فایل است .

نکته : sprite به کاراکترهای متحرکی گفته می شود که در بازیها وجود دارد .

اولین چیزی که به آن نیاز دارید ایجاد یک فرم است . خاصیت ScaleMode آنرا برابر ۳-Pixel قرار دهید . پیشنهاد می کنم که همیشه در هنگام استفاده از فرم بهمراه API از pixel برای scalemode استفاده کنید .

سپس سایز فرم را به اندازه ای افزایش دهید تا ScaleWidth برابر ۳۲۰ و ScaleHeight برابر ۲۵۶ شود . توجه کنید که خاصیت HasDC فرم را True قرار دهید . همچنین از خاصیت AutoRedraw برای فرم

استفاده نمی کنیم زیرا می خواهیم از Double Buffering استفاده کنیم که بسیار سریعتر و کارامدتر می باشد .

مرحله بعدی declare کردن API هایی است که به آنها نیاز داریم :

```
blitting'
gdi32" (ByVal " Private Declare Function BitBlt Lib
hDestDC As Long, ByVal x As Long, ByVal y As Long,
As Long, ByVal nHeight As Long, ByVal  ByVal nWidth
ySrc As  hSrcDC As Long, ByVal xSrc As Long, ByVal
Long, ByVal dwRop As Long) As Long
code timer'
Function GetTickCount Lib "kernel32" ()  Private Declare
As Long
sprites creating buffers / loading'
Private Declare Function CreateCompatibleBitmap Lib
hdc As Long, ByVal nWidth As Long,  "gdi32" (ByVal
ByVal nHeight As Long) As Long
Declare Function CreateCompatibleDC Lib  Private
Long "gdi32" (ByVal hdc As Long) As
Private Declare Function GetDC Lib "user32" (ByVal
Long hwnd As Long) As
loading sprites'
ByVal ) "Private Declare Function SelectObject Lib "gdi32
hdc As Long, ByVal hObject As Long) As Long
cleanup'
Declare Function DeleteObject Lib "gdi32" (ByVal  Private
Long hObject As Long) As
Private Declare Function DeleteDC Lib "gdi32" (ByVal
Long hdc As Long) As
```

سوال : DC چیست ؟ DC و یا بعبارت دیگر hDC، Device Context یک عدد است که به یک آدرس در حافظه اشاره می کند که داده ای در آن ذخیره شده است . در هنگام استفاده از BitBlt برای اشاره کردن به

آدرسی که داده گرافیکی در آنجا ذخیره شده ، استفاده می شود .
در مرحله بعدی نیاز به ذخیره آدرسهای DC داریم که می سازیم .
آدرسهای DC مقادیر Long هستند همچنین آنها را بصورت Public
تعریف می کنیم :

```
our Buffer's DC'
Long Public myBackBuffer As
Public myBufferBMP As Long
The DC of our sprite/graphic'
mySprite As Long Public
coordinates of our sprite/graphic on the screen'
SpriteX As Long Public
Public SpriteY As Long
```

حال باید تابعی بسازیم که تصاویر گرافیکی درون حافظه load کند .
نکته مهمی که باید به آن توجه کنید اینست که یک device context
خودش به تنهایی هیچ داده گرافیکی ندارد و بایستی یک bitmap
موجود باشد تا درون آن load شود برای مثال یک فایل bmp یا یک
bitmap خالی که از آن بعنوان back buffer استفاده می کنید .
تابعی که خواهیم نوشت یک device context منطبق با صفحه می
سازد سپس فایلهای گرافیکی مورد نظر را درون device context قرار
می دهد :

```
As (Public Function LoadGraphicDC(sFileName As String
Long
temp variable to hold our DC address'
Long Dim LoadGraphicDCTEMP As
create the DC address compatible with'
screen the DC of the'
LoadGraphicDCTEMP =
```

```
((CreateCompatibleDC(GetDC(0
...file into the DC load the graphic'
,SelectObject LoadGraphicDCTEMP
(LoadPicture(sFileName
return the address of the file'
LoadGraphicDCTEMP = LoadGraphicDC
End Function
```

سوال : double-buffering چیست ؟ زمانیکه یک محیط گرافیکی می سازید تا درون آن چیزی را ترسیم کنید ، شما sprite ها / گرافیکها / متن را درون حافظه blit می کنید (offscrean) سپس نتیجه نهایی را روی صفحه blit می کنید . این عمل از لرزش تصویر یا flickering جلوگیری می کند (زمانی رخ می دهد که چندین sprite مستقیماً روی صفحه blit شوند) و بسیار سریعتر از AutoRedraw است .

قبل از اینکه مثالی برای این تابع ذکر کنم تابع BitBlt را توضیح خواهم داد :

BitBlt تابعی از کتابخانه gdi32" dll" است . این تابع یک انتقال bit-block از داده های مرتبط به یک مستطیل از پیکسلها به یک device context مقصد انجام می دهد . بعبارت دیگر داده های گرافیکی را از محیط گرافیکی (یک bitmap) به محیط گرافیکی دیگری (screen یا یک form) کپی می کند . فرم کلی این تابع بصورت زیر است :

```
_ "Declare Function BitBlt Lib "gdi32" Alias "BitBlt
_ ,ByVal hDestDC As Long)
_ ,ByVal x As Long
_ ,ByVal y As Long
```

```
_ ,ByVal nWidth As Long
_ ,ByVal nHeight As Long
_ ,Long ByVal hSrcDC As
_ ,ByVal xSrc As Long
_ ,ByVal ySrc As Long
Long) As Long ByVal dwRop As
```

اولین خط بیان می کند که ما بوسیله gdi32 DLL به تابع BitBlt دسترسی خواهیم داشت . خطوط دیگر پارامترهایی هستند که این تابع می گیرد :

hDC : hDestDC مربوط به محیط مقصد (اگر می خواهید مقصد یک فرم باشد از form.hDC استفاده کنید و یا اینکه آدرس یک backbuffer را که ساخته اید بدهید)

x : مختصات افقی محلی که می خواهید گرافیک شما ظاهر شود .

y : مختصات عمدی محلی که می خواهید گرافیک شما ظاهر شود .

nWidth : عرض گرافیک شما

nHeight : ارتفاع گرافیک شما

hDC : hSrcDC مربوط به محیط مبدا

xSrc : افست 0 . x زمانی استفاده می شود که بخواهید از سمت چپترین گوشه گرافیک مبدا عمل blit را انجام دهید .

ySrc : افست y

dwRop : مد draw ای که در زمان blitting گرافیکتان می خواهید استفاده کنید (Raster Operations یا ROP) . این پارامتر مقادیر زیر را می تواند بگیرد :

- vbSrcCopy : داده تصویر مبدا را مستقیماً در مقصد کپی می کند .

- vbSrcPaint : داده های تصاویر مبدا و مقصد را با هم OR می کند (pseudo-alphablending effect)
- vbSrcAnd : داده های تصاویر مبدا و مقصد را با هم AND می کند (pseudo-gamma effect)
- vbSrcInvert : داده های تصاویر مبدا و مقصد را با هم XOR می کند
- vbSrcErase : ابتدا داده تصویر مقصد را invert می کند سپس آنرا با داده تصویر مبدا AND می کند .
- vbDstInvert : داده تصویر مقصد را invert می کند و داده تصویر مبدا را در نظر نمی گیرد .
- vbNotSrcCopy : داده تصویر مبدا را invert می کند و آنرا مستقیماً در مقصد کپی می کند .
- vbNotSrcErase : داده تصاویر مبدا و مقصد را OR کرده و نتیجه را invert می کند .

مثالی از کاربرد BitBlt :

```
BitBlt Form1.hDC, PlayerX, PlayerY, 48, 48,
picPlayer.hDC, 0, 0, vbSrcCopy
```

حال می خواهیم از BitBlt در یک حلقه استفاده کنیم تا یک image را در فرم حرکت دهیم :

۱ – یک فایل bmp با ابعاد x32۳۲ بسازید و با نام sprite1.bmp در دایرکتوری پروژه ذخیره کنید .

۲ – یک دکمه در فرم قرار دهید و نام آنرا cmdTest بگذارید .

۳ – دکمه را در گوشه بالایی فرم و در سمت راست قرار دهید .

۴ – کد زیر را برای event مربوط به کلیک شدن دکمه بنویسید :

```
...Timer variables'
Long Dim T1 As Long, T2 As
'backbuffer برای DC ساخت
((CreateCompatibleDC(GetDC(0 = myBackBuffer
'DC برای bitmap ساخت یک سطح
CreateCompatibleBitmap(GetDC(0), 320, = myBufferBMP
(256
'buffer خالی درون bitmap کردن سطح load
SelectObject myBackBuffer, myBufferBMP
'پر کنیم black با آنرا باید بافر درون کردن blit از قبل
vbWhiteness ,BitBlt myBackBuffer, 0, 0, 320, 256, 0, 0, 0
'نوشتیم بالا در که تابعی توسط split کردن load
("sprite1.bmp\" & LoadGraphicDC(App.Path = mySprite
cmdTest.Enabled = False
'== شروع حلقه اصلی ==
'جاری tickcount خواندن
GetTickCount = T2
Do
DoEvents
T1 = GetTickCount
'شود شروع بعدی فریم بود گذشته ثانیه میلی ۱۵ اگر
Then ۱۵ =< (If (T1 - T2
' black با آنجا کردن پر بوسیله sprite قبلی محل کردن پاک
SpriteY - 1,32, 32, 0, 0, ,BitBlt myBackBuffer, SpriteX - 1
0, vbBlackness
'buffer back درون sprite کردن Blit
BitBlt myBackBuffer, SpriteX, SpriteY, 32, 32,mySprite, 0,
vbSrcPaint ,0
```

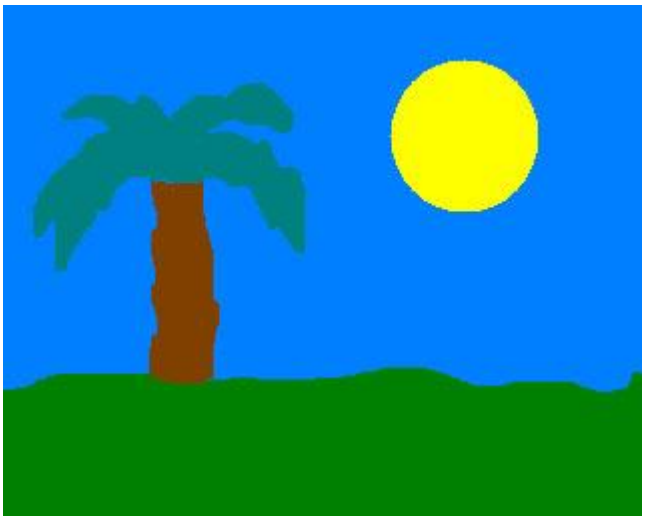
```
'Blit کردن backbuffer روی فرم
myBackBuffer,0, 0,  ,BitBlt Me.hdc, 0, 0, 320, 256
vbSrcCopy
'حرکت دادن sprite روی صفحه
SpriteX = SpriteX + ۱
SpriteY = SpriteY + 1
update timer'
T2 = GetTickCount
If End
Loop Until SpriteX = 320
```

سپس باید یک cleanup code بنویسید تا حافظه های را که برای نگهداری تصاویر گرافیکی و buffer ها استفاده کرده اید آزاد کنید :

```
(Private Sub Form_Unload(Cancel As Integer
DeleteObject myBufferBMP
DeleteDC myBackBuffer
mySprite DeleteDC
End
End Sub
```

آشنایی با تکنیک Masking در متحرک سازی

حتماً بازیهای دوبعدی را دیده اید که یک کاراکتر متحرک توسط شما روی یک زمینه تصویری حرکت داده می شود . تکنیکی که برای قراردادن یک کاراکتر متحرک روی یک زمینه ثابت استفاده می شود Masking نام دارد . Masking عبارت است از حذف بخشهایی از کاراکتر متحرک که نمی خواهیم نشان داده شوند و قرار دادن کاراکتر روی یک زمینه تصویری . برای آشنایی بیشتر با این تکنیک دو تصویر زیر را مشاهده کنید :



می خواهیم کاراکتر یک هواپیما را روی زمینه تصویری حرکت دهیم .
برای اینکار نیاز به دو Mask برای کاراکتر هواپیما داریم :
۱ – Mask اول بدین صورت ساخته می شود که نقاطی از کاراکتر را که می خواهیم روی زمینه ظاهر شوند را به رنگ سیاه و سایر نقاط را به رنگ سفید در می آوریم :

۲ – Mask دوم معکوس ماسک اول است یعنی نقاطی از کاراکتر را که می خواهیم روی زمینه ظاهر شوند را به رنگ سفید و سایر نقاط را به رنگ سیاه در می آوریم :

حال برای قرار دادن کاراکتر روی زمینه باید مراحل زیر انجام شود :

۱ – قرار دادن ماسک اول روی زمینه با استفاده از MergePaint

۲ – قرار دادن ماسک دوم روی کاراکتر با استفاده از MergePaint

۳ – قرار دادن نتیجه مرحله اول روی نتیجه مرحله دوم با استفاده از And

مراحل برنامه نویسی :

۱ – تعریف بافرهای مورد نیاز :

```
buffer1 = CreateCompatibleDC(GetDC(0))x
CreateCompatibleBitmap(GetDC(0), 50, 45)x = mybuffer1
CreateCompatibleDC(GetDC(0))x = buffer2
,۳۲۰ ,(mybuffer2 = CreateCompatibleBitmap(GetDC(0
x(۲۵۶
buffer3 = CreateCompatibleDC(GetDC(0))x
CreateCompatibleBitmap(GetDC(0), 320, = mybuffer3
256)x

mybuffer1 ,SelectObject buffer1

SelectObject buffer2, mybuffer2

mybuffer3 ,SelectObject buffer3
```

ابعاد زمینه x256۳۲۰ و ابعاد کاراکتر هواپیما x4550 است . buffer1 برای ذخیره کاراکتر و انجام عملیات شماره ۲ بکار می رود . buffer2 برای نگهداری زمینه بکار می رود و buffer3 برای انجام عملیاتهای ۱ و

۳ بکار می رود .

۲ – Load کردن تصاویر مورد نیاز :

```
& mySprite = LoadGraphicDC(App.Path
sprite1.bmp")x\"
& myMask1 = LoadGraphicDC(App.Path
mask1.bmp")x\"
& myMask2 = LoadGraphicDC(App.Path
mask2.bmp")x\"
```

زمینه تصویر Picture فرم شما می باشد پس ابتدا باید به فرمتان یک picture را بعنوان background بدهید .

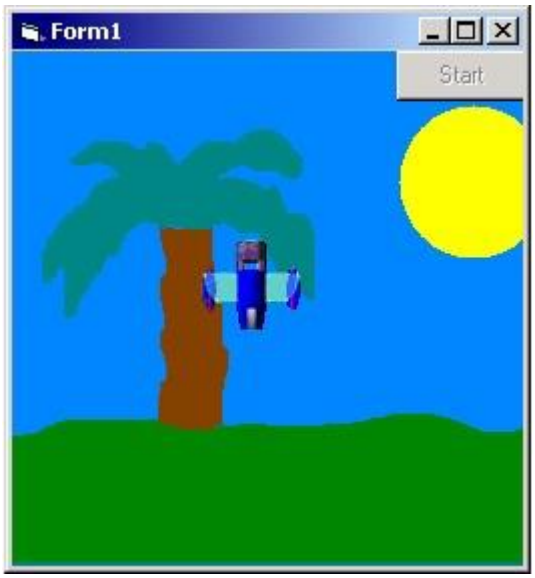
۳ – ذخیره زمینه در buffer2 :

```
vbSrcCopy ,BitBlt buffer2, 0, 0, 320, 256, Form1.hdc, 0, 0
```

۴ – شروع حلقه متحرک سازی و انجام Masking :

```
Do
DoEvents
'قرار دادن زمینه در یک بافر کمکی
BitBlt buffer3, 0, 0, 320, 256, buffer2, 0, 0, vbSrcCopy
'Merge کردن ماسک اول با بافر کمکی زمینه
myMask1, 0, 0,  ,BitBlt buffer3, SpriteX, SpriteY, 50, 45
vbMergePaint
'قرار دادن کاراکتر در یک بافر کمکی
buffer1, 0, 0, 50, 45, mySprite, 0, 0, vbSrcCopy BitBlt
```

```
' Merge کردن ماسک دوم با بافر کمکی کاراکتر
vbMergePaint ,BitBlt buffer1, 0, 0, 50, 45, myMask2, 0, 0
' And کردن بافرهای کمکی با هم
SpriteY, 50, 45, buffer1, 0, 0,  ,BitBlt buffer3, SpriteX
vbSrcAnd
' قرار دادن نتیجه نهای روی فرم
BitBlt Form1.hdc, 0, 0, 320, 256, buffer3, 0, 0, vbSrcCopy
SpriteX + 1 = SpriteX
SpriteY = SpriteY + 1
Loop Until SpriteX = 320
```



راهنمای برنامه نویسی OpenGL در ویژوال بیسیک

OpenGL یک کتابخانه low-level برای ساخت برنامه های گرافیکی می باشد که این کتابخانه در دو فایل dll پیاده سازی شده است . دو ورژن مختلف از این کتابخانه وجود دارد : نسخه Microsoft و نسخه SGI .

نسخه Microsoft دارای دو فایل dll به نامهای opengl32.dll و glu32.dll است و نسخه SGI دارای دو فایل به نامهای opengl.dll و glu.dll می باشد .

برای استفاده از OpenGL در ویژوال بیسیک تنها چیزی که لازمست نصب یک کتابخانه نمونه می باشد :

VBOpenGL 1.2 for Microsoft
VBOpenGL 1.2 for SGI

شما می توانید هم از نسخه MS و هم SGI استفاده کنید . "

اطلاعات فوق را من از سایت Visual Programming OpenGL with Basic برایتان ترجمه کرده ام

Collision Detection الگوریتم

Collision Detection الگوریتم

نکته : اگر مباحث آشنایی با تابع BitBlt و آشنایی با تکنیک Masking را مطالعه نکرده اید ، پیشنهاد می کنم ابتدا آنها را مطالعه کنید .

تکنیک collision detection ، تکنیک تشخیص برخورد دو sprite متحرک در صفحه می باشد . این تکنیک برای ساخت بازیهای کامپیوتری بسیار استفاده می شود و الگوریتمهای زیادی برای تشخیص برخورد دو بعدی و سه بعدی وجود دارد . در این بخش من شما را با چند تکنیک تشخیص برخورد دوبعدی آشنا خواهم کرد :

یکی از ساده ترین الگوریمتهای تشخیص برخورد ، الگوریتم bounding box می باشد . در این الگوریتم مرزهای دو sprite یا bounding rectangles چک می شوند تا برخورد تشخیص داده شود .

یکی دیگر از الگوریتمهای تشخیص برخورد ، الگوریتم bounding sphere است . در این الگوریتم دایره های محیطی دو sprite با هم مقایسه می شوند .

اما یکی از بهترین الگوریتم های تشخیص برخورد دو بعدی الگوریتمی است که ابتدا مستطیل محیطی دو sprite را با هم مقایسه می کند و در صورت وجود تداخل دو مستطیل ، به سراغ چک کردن تداخل در سطح پیکسلهای دو sprite می رود . برای بررسی تداخل در سطح پیکسل از mask های دو sprite استفاده می شود .

قبل از معرفی این الگوریتم ابتدا چند تابع را که در این الگوریتم استفاده می شوند معرفی خواهم کرد :

۱ – تابع CreateCompatibleDC موجود در کتابخانه gdi32 : این تابع یک حافظه DC (device context) مطابق با مقصد مشخص شده می سازد .

۲ – تابع CreateCompatibleBitmap موجود در کتابخانه gdi32 : این تابع یک bitmap مطابق با مقصدی که توسط DC مشخص شده ، می سازد .

۳ – تابع DeleteDC موجود در کتابخانه gdi32 : یک DC را از حافظه پاک می کند .

۴ – تابع GetPixel موجود در کتابخانه gdi32 : مقدار rgb رنگ یک پیکسل را برمی گرداند .

۵ – تابع SelectObject موجود در کتابخانه gdi32 : یک شی را از درون یک DC انتخاب می کند .

۶ – تابع DeleteObject موجود در کتابخانه gdi32 : یک شی را حذف کرده و کلیه منابع اختصاص یافته به آنرا آزاد می کند .

۷ – تابع IntersectRect موجود در کتابخانه user32 : تداخل دو مستطیل را محاسبه می کند و مختصات مستطیل تداخلی را در یک مستطیل مقصد قرار می دهد .

حال که با این توابع آشنا شدید به سراغ معرفی الگوریتم می روم . در این الگوریتم ابتدا باید یک type از نوع مستطیل تعرف کنیم :

```
Type RECT
Left As Long
Top As Long
Long Right As
Bottom As Long
End Type
```

همچنین نیاز به تعریف متغیرهای زیر داریم :

```
Dim MaskRect1 As RECT
RECT Dim MaskRect2 As
Dim DestRect As RECT
Dim i As Long
Dim j As Long
As Boolean Dim Collision
Dim MR1SrcX As Long
Dim MR1SrcY As Long
Long Dim MR2SrcX As
Dim MR2SrcY As Long
Dim hNewBMP As Long
Long Dim hPrevBMP As
Dim tmpObj As Long
Dim hMemDC As Long
```

سپس باید کلیه توابع فوق را بهمراه تابع BitBlt در پروژه خود declare کنید . با استفاده از API viewer می توانید فرمت این توابع را پیدا کنید .

شکل کلی تابع تشخیص برخورد بصورت زیر است :

```
Public Function CollisionDetect(ByVal x1 As Long, ByVal
y1 As Long, ByVal X1Width As Long, ByVal Y1Height As
Long, ByVal Mask1LocX As Long, ByVal Mask1LocY As
Long, ByVal Mask1Hdc As Long, ByVal x2 As Long,
ByVal y2 As Long, ByVal X2Width As Long, ByVal
Y2Height As Long, ByVal Mask2LocX As Long, ByVal
Mask2LocY As Long, ByVal Mask2Hdc As Long) As
Boolean
```

که x1 و y1 و x2 و y2 بترتیب مختصات نقطه بالایی سمت چپ sprite اول و دوم هستند .

x1width و y1height و x2width و y2height بترتیب ابعاد مستطیا محیطی دو sprite هستند .

Mask1Hdc و Mask2Hdc، DC های ماسکهای دو sprite هستند .

Mask1LocX و Mask1LocY مختصات افست ماسک اول و
Mask2LocX و Mask2LocY مختصات افست ماسک دوم می باشند .
ابتدا ابعاد مستطیلهای محیطی دو sprite را تنظیم می کنیم :

```
MaskRect1.Left = x1
y1 = MaskRect1.Top
MaskRect1.Right = x1 + X1Width
Y1Height + MaskRect1.Bottom = y1
MaskRect2.Left = x2
MaskRect2.Top = y2
X2Width + MaskRect2.Right = x2
MaskRect2.Bottom = y2 + Y2Height
```

سپس توسط تابع IntersectRect تداخل مستطیلهای محیطی بررسی
می شود و در صورت تداخل ، مستطیل تداخل استخراج می شود :

```
MaskRect2)x ,i = IntersectRect(DestRect, MaskRect1
If i = 0 Then
CollisionDetect = False
```

حال به سراغ بررسی پیکسل به پیکسل می رویم . برای اینکار بایستی
مقادیر sourceX و sourceY را برای HDC های دو ماسک بدست
آوریم :

```
Else
x2 Then < If x1
MR1SrcX = 0
```

```
x1 - x2 = MR2SrcX
Else
MR2SrcX = 0
MR1SrcX = x2 - x1
End If
Then y2 < If y1
MR2SrcY = y1 - y2
MR1SrcY = 0
Else
here ' MR2SrcY = 0
MR1SrcY = y2 - y1 - 1
End If
```

سپس حافظه های DC و Bitmap را برای انجام مقایسه تخصیص می دهیم :

```
= hMemDC
CreateCompatibleDC(Screen.ActiveForm.hdc)x
= hNewBMP
CreateCompatibleBitmap(Screen.ActiveForm.hdc,
DestRect.Bottom - ,DestRect.Right- DestRect.Left
DestRect.Top)x
hNewBMP)x ,hPrevBMP = SelectObject(hMemDC
```

اولین sprite را در حافظه hMemDc قرار می دهیم :

```
– i = BitBlt(hMemDC, 0, 0, DestRect.Right
DestRect.Left,DestRect.Bottom- DestRect.Top,Mask1Hdc,
MR1SrcY + ,MR1SrcX + Mask1LocX
Mask1LocY,vbSrcCopy) x
```

sprite دوم را با sprite اول OR می کنیم :

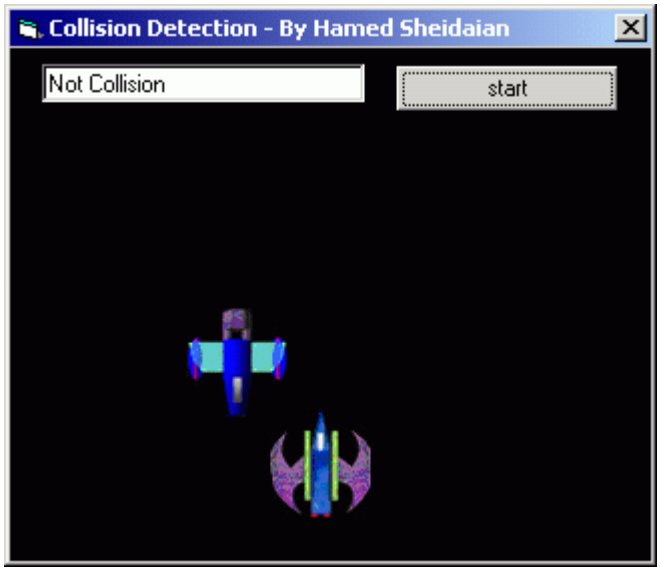
```
– i = BitBlt(hMemDC, 0, 0, DestRect.Right
DestRect.Left,DestRect.Bottom- DestRect.Top,Mask2Hdc,
MR2SrcY + Mask2LocY,  ,MR2SrcX + Mask2LocX
vbSrcPaint)x
```

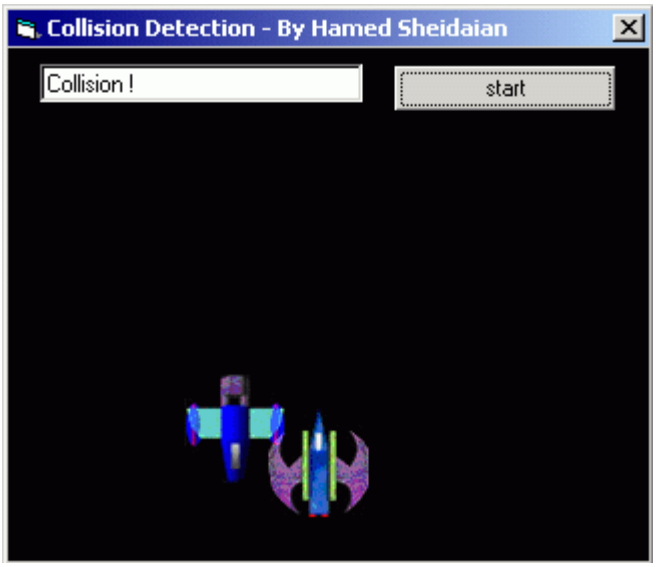
حال تمام پیکسل های hMemDC را بررسی می کنیم . اگر رنگ یکی از این پیکسلها سیاه باشد یعنی تداخل وجود دارد :

```
Collision = False
DestRect.Top - 1 – For i = 0 To DestRect.Bottom
For j = 0 To DestRect.Right - DestRect.Left - 1
GetPixel(hMemDC, j, i) = 0 Then If
Collision = True
Exit For
If End
Next
If Collision = True Then
Exit For
If End
Next
CollisionDetect = Collision
```

در پایان تمام object ها و DC ها را از بین می بریم :

```
tmpObj = SelectObject(hMemDC, hPrevBMP)x
DeleteObject(tmpObj)x = tmpObj
tmpObj = DeleteDC(hMemDC)x
```





آموزش DirectX-Graphic قسمت اول

آموزش DirectX-Graphic قسمت اول

موضوع : ساخت یک واسط direct3D

DirectX8 ابزاری برای ساخت تصاویر ثابت و متحرک دو بعدی و سه بعدی می باشد .

برای کار با DirectX8 ابتدا بایستی آنرا روی سیستم خود نصب کنید .

سپس در محیط vb از منوی project گزینه References را انتخاب کنید . در فرمی که ظاهر می شود اطمینان حاصل کنید که گزینه DirectX8 for VB type library فعال باشد .

برای کار با DirectX8 بایستی از تعریف نمودن شی پایه DirectX8 شروع نمود :

```
Dim Dx as DirectX8
```

شی Direct3D8 برای کنترل اشیا سه بعدی بکار می رود :

```
Dim D3D as Direct3D8
```

شی Direct3DDevice8 ، سخت افزار مربوط به رندر تصاویر را مشخص می کند :

```
Dim D3DDevice as Direct3DDevice8
```

حال برای شروع کار با Direct3D ، تابع ( ) initialise را تعریف می کنیم . اگر اینکار درست انجام شود تابع ، مقدار true را برمی گرداند :

```
public function initialise () as boolean
DispMode as D3DISPLAYMODE Dim
```

شی D3DISPLAYMODE حالت نمایش را مشخص می نماید .

```
Dim D3Dwindow as D3DPRESENT_PARAMETERS
```

شی فوق مشخص می کند که viewport شما چگونه باشد .
حال شی اصلی DirectX8 را می سازیم :

```
Set Dx=New DirectX8
```

سپس شی اصلی ساخت واسط سه بعدی را می سازیم :

```
set D3D.Dx.Direct3Dcreate()
```

سپس حالت فعلی نمایش را با دستور زیر استخراج می کنیم :

D3D.getadapterdisplaymode
D3DADAPTER_DEFAULT,dispmode

حال دو حالت برای کار با DirectX داریم :

۱ – windowed mode

۲ – fullscrean mode

۱ – برای کار با حالت پنجره ای ابتدا این موضوع را به DirectX اطلاع می دهیم :

D3Dwindow.windowed=1

سپس نوع referesh تصویر را مشخص می کنیم (در اینجا چند انتخاب وجود دارد که در صورت نیاز به اطلاعات بیشتر با من تماس بگیرید .) :

D3Dwindow.swapeffect=D3DSWAPEFFECT_COPY_VSYNC

سپس بایستی فرمت بافر نگهدارنده تصاویر را مشخص کنیم :

D3Dwindow.backbufferformat=dispmode.format

۲ – برای کار با حالت تمام صفحه ، ابتدا نوع refresh را مشخص کرده

سپس تعداد بافر های تصویر و سرانجام نوع و سایز بافر را مشخص می نمائیم :

```
D3Dwindow.swapeffect=D3DSWAPEFFECT_DISCARD
D3Dwindow.backbuffercount=1
D3Dwindow.backbufferformat=dispmode.format
D3Dwindow.backbufferheight=dispmode.height
D3Dwindow.backbufferwidth=dispmode.width
```

سپس پنجره نمایش مشخص می گردد :

```
D3Dwindow.hdevicewindow=frmMain.hwnd
```

@حال بایستی یک device ساخته شود که یا از طریق سخت افزار و یا نرم افزار تصاویر را رندر نماید :

```
Set
D3DDevice=D3Dcreatedevice(D3DADAPTER_DEFAULT
,D3DDEVTYPE_HAL,
frmMain.hwnd,D3DCREATE_SOFTWARE_VERTEXPR
,OCESSING
,D3Dwindow)x
sub end
```

درصورتی که کارت گرافیک شما امکانات رندر سخت افزاری تصاویر را ندارد از D3DDEVTYPE_REF بجای D3DDEVTYPE_HAL استفاده کنید .

حال بایستی روتین render را بنویسیم . البته در این درس تصویری برای رندر نداریم و تنها چگونگی نوشتن این روتین را بیان خواهم کرد :

۱ – ابتدا بایستی device مربوط به رندر ، قبل از کشیدن تصویر در آن پاک شود :

```
D3DDevice.clear 0,byval
H0,1#,0&,D3DCLEAR_TARGET,۰
```

عدد hex ای که در دستور فوق آمده رنگ زمینه صفحه را مشخص می کند

۲ – سپس بایستی تصاویر مورد نظر را رندر کنیم . اینکار توسط دستورات زیر انجام می شود :

```
D3DDevice.beginscence
' between these two lines all rendering calls go
D3DDEvice.endscence
```

۳ – در پایان بایستی صفحه را update کنید :

```
D3DDevice.present byval 0,byval 0,0,byval 0
```

آموزش DirectX-Graphic قسمت دوم

موضوع : بدست آوردن مشخصات و تواناییهای گرافیکی یک سیستم توسط DirectX-Graphic

۱ – شمارش تعداد آداپتورهای گرافیکی یک سیستم : فرض کنید متغیر nAdapters متغیری از نوع long باشد . همچنین شی D3DADAPTER_IDENTIFIER8 یک ساختار است که اطلاعات مربوط به آداپتور را نگه می دارد . در اینصورت روتین enumerateAdapters بصورت زیر خواهد بود :

```
D3DADAPTER_IDENTIFIER8 Dim adapterinfo as
Private Sub EnumerateAdapters
integer Dim i as
nadapters=D3D.Getadaptercount
```

برای بدست آوردن جزئیات آداپبورها بصورت زیر عمل می کنیم :

```
nadapters-1 for i=0 to
D3D.GetadapterIdentifier i ,0,adapterinfo
```

نام این آداپتور بصورت لیستی از کدهای اسکی است که بایستی آنها را درون یک string قرار دهیم :

```
for j=0 to 511
chr$(adapterinfo.description(j)) x & name=name
next j
x (" ",(name=replace(name,chr$(0
end sub
```

بنابراین در متغیر name نام آداپتور قرار خواهد گرفت .

۲ - مشخص کردن نوع Rendering : فرض کنید شی D3DCAPS8 توانایی rendering آداپتور را نشان دهد . در اینصورت روتین EnumerateDevices بصورت زیر خواهد بود :

```
Private EnumerateDevices
resume next On Local Error
Dim Caps as D3DCAPS8
Example deviceindex=0 'For
D3D.Getdevicecaps
deviceindex,D3DDEVTYPE_HAL,caps
err.number=D3DERR_NOTAVAILABLE then if
```

اگر آداپتور امکان رندر سخت افزاری نداشته باشد در اینصورت :

```
x ("(MsgBox("Reference Rasterizer(REF
else
MsgBox("Hardware Acceleration(HAL)+Reference
x ("(Rasterizer(REF
```

```
end if
end sub
```

۳ – شمارش تعداد Mode نمایشی آداپتور :

فرض کنید در صورت REF بودن امکان رندر ، متغیر r=2 و در غیراینصورت r=1

باشد . همچنین شی D3DDISPLAYMODE اطلاعات مدهای نمایشی را در خود

دارد . همچنین فرض کنید متغیر nModes از نوع longباشد . در اینصورت روتین enumeratedispmodes بصورت زیر خواهد بود :

```
Long) x Private Sub EnumerateDispModes(r as Long,n as
Dim i as integer
Dim mode_tmp as D3DDISPLAYMODE
For Example' deviceindex=0
nModes=D3D.Getadaptermodecount(deviceindex) x
nModes-1 for i=0 to
D3D.EnumAdapterModes(deviceindex,i,mode_tmp) x
```

ابتدا Mode ها را به دو گروه ۱۶ بیتی و ۳۲ بیتی تقسیم می کنیم :

```
if mode_tmp.format=D3DFMT_R8G8B8 or
mode_tmp=D3DFMT_X8R8G8B8 or
mode_tmp=D3DFMT_A8R8G8B8 then
```

حال چک می کنیم که device قابل پذیرش و معتبر است یا نه :

```
if
D3D.checkdevicetype(deviceindex,r,mode_tmp.format,mo
then ۰=<(de_tmp.format,Flase
& mode_tmp.height & "X" & MsgBox(mode_tmp.width
Bit ۳۲"
" & mode_tmp.format ) x & ":FMT
end if
else
if
D3D.checkdevicetype(deviceindex,r,mode_tmp.format,mo
then ۰=<(de_tmp.format,Flase
& mode_tmp.height & "X" & MsgBox(mode_tmp.width
Bit ۱۶"
" & mode_tmp.format ) x & ":FMT
end if
end if
next i
```

۴ – مشخص کردن توانایی های آداپتور گرافیکی : فرض کنید در صورت بودن امکان رندر ، متغیر r=2 و در غیراینصورت r=1 باشد : REF

```
x (Private Sub EnumerateHardware(r as long
Dim caps as D3DCAPS8
D3D.Getdevicecaps deviceindex,r,caps
Caps.MaxActiveLights = -1 Then If
x "MsgBox "Maximum Active Lights: Unlimited
Else
& " :MsgBox "Maximum Active Lights
Caps.MaxActiveLights
```

```
If End
& " :MsgBox "Maximum Point Vertex size
Caps.MaxPointSize
& " :Maximum Texture Size" MsgBox
& "X" & Caps.MaxTextureWidth
Caps.MaxTextureHeight
& " :MsgBox "Maximum Primatives in one call
Caps.MaxPrimitiveCount
If Caps.TextureCaps And
Then D3DPTEXTURECAPS_SQUAREONLY
MsgBox "Textures must always be square" x
End If
Caps.TextureCaps And  If
D3DPTEXTURECAPS_CUBEMAP Then
Cube Mapping" x MsgBox "Device Supports
End If
If Caps.TextureCaps And
Then D3DPTEXTURECAPS_VOLUMEMAP
MsgBox "Device Supports Volume Mapping" x
End If
And D3DDEVCAPS_PUREDEVICE  If Caps.DevCaps
Then
Option" x MsgBox "Device supports the Pure Device
End If
If Caps.DevCaps And
Then D3DDEVCAPS_HWTRANSFORMANDLIGHT
MsgBox "Device supports hardware transform and
lighting" x
If End
If Caps.DevCaps And
D3DDEVCAPS_HWRASTERIZATION Then
use Hardware Rasterization" x MsgBox "Device can
End If
If Caps.Caps2 And
D3DCAPS2_CANCALIBRATEGAMMA Then
MsgBox "Device can Calibrate Gamma" x
```

```
If End
If Caps.Caps2 And
D3DCAPS2_CANRENDERWINDOWED Then
Render in Windowed Mode" x MsgBox "Device can
End If
D3DCAPS2_FULLSCREENGAMMA  If Caps.Caps2 And
Then
fullscreen mode"  MsgBox "Device can calibrate gamma in
x
End If
If Caps.RasterCaps And
Then D3DPRASTERCAPS_FOGRANGE
MsgBox "Device supports range based fog calculations" x
End If
Caps.RasterCaps And  If
D3DPRASTERCAPS_ANISOTROPY Then
Anisotropic Filtering" x MsgBox "Device supports
End If
If Caps.RasterCaps And
D3DPRASTERCAPS_ZBUFFERLESSHSR Then
Z-Buffer/Depth  MsgBox "Device does not require a
Buffer" x
End If
```

آموزش DirectX-Graphic قسمت سوم

موضوع : رسم اشکال دو بعدی

مروری بر object های DirectX8

۱ – DirectX8 : این شی ، شی مرکزی برای directX است و به شما امکان دسترسی به توابع و اشیا DirectX را می دهد .

۲ – Direct3D8 : شی اصلی برای کار با محیط سه بعدی می باشد . هدف از آن ، ساخت Direct3DDevice8 است و همچنین شامل توابعی

برای مشخص کردن توانایی های کارت گرافیک است .

۳ – Direct3DDevice8 : این شی مسئول ساخت بافتها textures ، مدیریت نورها در یک صحنه ، مدیریت مواد materials و همچنین render صحنه است . در واقع این شی ، قلب نمایشی کار شماست .

۴ – D3DX8 : گر چه همیشه نیازی به استفاده از این شی نیست ، اما این شی شامل توابعی برای ساخت برنامه های userfriendly تر توسط DirectX است . مثلاً ساخت اشیا سه بعدی (مثل کره ، مکعب و ...) ، ساخت بافتها ، ساخت سطوح و غیره

شروع کار برای رسم اشیا دوبعدی

ابتدا ثابت FVF را تعریف می کنیم . این ثابت توصیف " فرمت قابل انعطاف نقطه flexible-vertex-format " برای یک vertex دو بعدی انتقال یافته و ساده شده می باشد .

سپس بایستی یک ساختار برای توصیف این vertex معرفی کنیم :

```
Const FVF = D3DFVF_XYZRHW Or D3DFVF_TEX1 Or
D3DFVF_DIFFUSE Or D3DFVF_SPECULAR
Private Type TLVERTEX
X As Single
As Single Y
Z As Single
rhw As Single
color As Long
Long specular As
tu As Single
tv As Single
End Type
```

فرض کنید بخواهیم یک مربع را در صفحه رسم کنیم . برای رسم آن نیاز

به ۴ عدد vertex داریم . بنابراین آرایه TriStrip را از نوع TLVERTEX تعریف میکنیم :

```
Dim TriStrip (0 To 3) As TLVERTEX
```

حال به سراغ تابع initialize که در درس ۱ با آن آشنا شدید می رویم و دستورات زیر را به آن اضافه می کنیم :

```
boolean Private Function Initialize as
.
.
.
```

ابتدا سیستم سایه زنی vertex را طوری تنظیم می کنیم که از FVF استفاده کند .

```
D3DDevice.SetVertexShader FVF
```

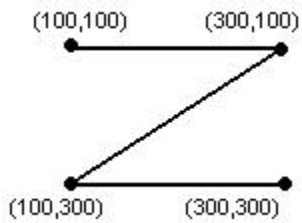
حال سیستم lighting را برای vertex های دو بعدی غیر فعال می کنیم زیرا نیازی به آن نداریم :

```
D3DRS_LIGHTING,false D3DDevice.SetRenderState
```

حال بایستی تابع initializeGeometry را اجرا کنیم . این تابع را در ادامه توضیح خواهم داد . اگر نتیجه این تابع true باشد دراینصورت initialize به درستی انجام شده است :

```
initialize=true if initializeGeometry()=true then
end function
```

تابع initializeGeometry در این درس ، تابعی ساده است که تنها آرایه Vertex ها را مقدار دهی می کند . برای رسم یک مربع نیاز به مقداردهی ۴ vertex در جهت عقربه های ساعت داریم (این مربع شامل ۲ مثلث است)




```
Boolean Private Function InitialiseGeometry() As
:On Error GoTo BOut
(color = RGB(200, 100, 0
(CreateTLVertex(100, 100, 0, 1, color, 0, 0, 0 = (TriStrip(0
(CreateTLVertex(300, 100, 0, 1, color, 0, 0, 0 = (TriStrip(1
(CreateTLVertex(100, 300, 0, 1, color, 0, 0, 0 = (TriStrip(2
(CreateTLVertex(300, 300, 0, 1, color, 0, 0, 0 = (TriStrip(3
True = InitialiseGeometry
```

```
Exit Function
:BOut
InitialiseGeometry = False
End Function
```

همانطور که مشاهده می کنید برای تعریف vertex از تابع CreateTLVERTEX استفاده شده است . این تابع صرفاً مقادیر ساختار TLVERTEX را مقداردهی می کند :

```
As  Private Function CreateTLVertex(X As Single, Y
Single, Z As Single, rhw As Single, color As Long, specular
Single, tv As Single) As TLVERTEX As Long, tu As
```

نکته : ضمن اینکه شما می توانید مقادیر اعشاری floating point را برای مختصاتهای x و y و z بکار ببرید ، Direct3D مختصاتها را با گردکردن آنها تخمین می زند و بنابراین ممکنست باعث ایجاد نتایج ناخواسته شود .

```
CreateTLVertex.X = X
Y = CreateTLVertex.Y
CreateTLVertex.Z = Z
CreateTLVertex.rhw = rhw
color = CreateTLVertex.color
CreateTLVertex.specular = specular
tu = CreateTLVertex.tu
CreateTLVertex.tv = tv
End Function
```

حال بایستی تابع Render را بنویسیم :

```
()Public Sub Render
,۰ ,D3DDevice.Clear 0, ByVal 0, D3DCLEAR_TARGET
۰ ,#۱
D3DDevice.BeginScene
D3DDevice.DrawPrimitiveUP
,(D3DPT_TRIANGLESTRIP, 2, TriStrip(0
Len(TriStrip(0))x
D3DDevice.EndScene
ByVal 0 ,۰ ,D3DDevice.Present ByVal 0, ByVal 0
End Sub
```

ساختار اصلی برای اجرای توابع فوق بصورت زیر است :

```
--Main part--
Initialize
yourevent=true Do While
Render
DoEvents
Loop
```

آموزش DirectX-Graphic قسمت چهارم

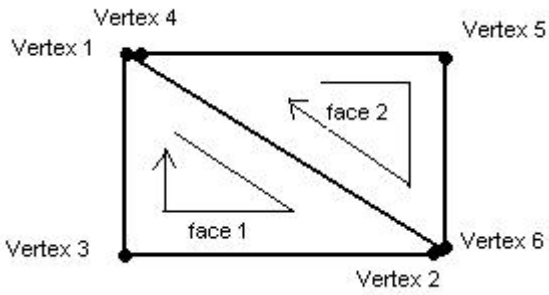
موضوع : آشنایی با برخی اصطلاحات

۱- Mesh : مش ، مجموعه ای از face ها است که یک شی سه بعدی را روی صفحه تشکیل می دهند .

۲ – Face : یک چند ضلعی است که توسط مجموعه ای از نقاط به نام vertex ساخته می شود .

۳ – Vertex : یک نقطه در فضای سه بعدی است که برای دادن موقعیت ، scale و زاویه یک face استفاده می شود .

۴ – Direct3D از شیی بنام D3DVERTEX برای نمایش یک Vertex استفاده می کند . برای ساخت face نیز از آرایه ای از vertex ها استفاده می شود . آرایه همیشه بایستی قابل تقسیم به سه باشد زیرا اشکال از face های مثلثی ساخته می شوند . هنگامیکه این مثلثها کنار هم گذاشته شوند ، شی سه بعدی را می سازند . Direct3D از بافری با نام Index Buffer استفاده می کند که با direct3D می گوید که با چه ترتیبی vertex ها را رسم نماید . Index ها بایستی همیشه در جهت عقربه های ساعت مشخص شوند .



آموزش DirectX-Graphic قسمت پنجم

موضوع : اختصاص بافت Texture به اشکال دو بعدی

در این درس می خواهیم یک مربع که دارای بافت می باشد را رسم کنیم . برای اینکار از کتابخانه کمکی D3DX8 استفاده می کنیم . همچنین شی Direct3DTexture8 را نیز استفاده می نمائیم .

```
Dim D3DX as D3DX8
Direct3DTexture8 Dim Texture as
```

حال بایستی در تابع Initialize بافت مربوطه را از روی یک فایل تصویری load کنیم :

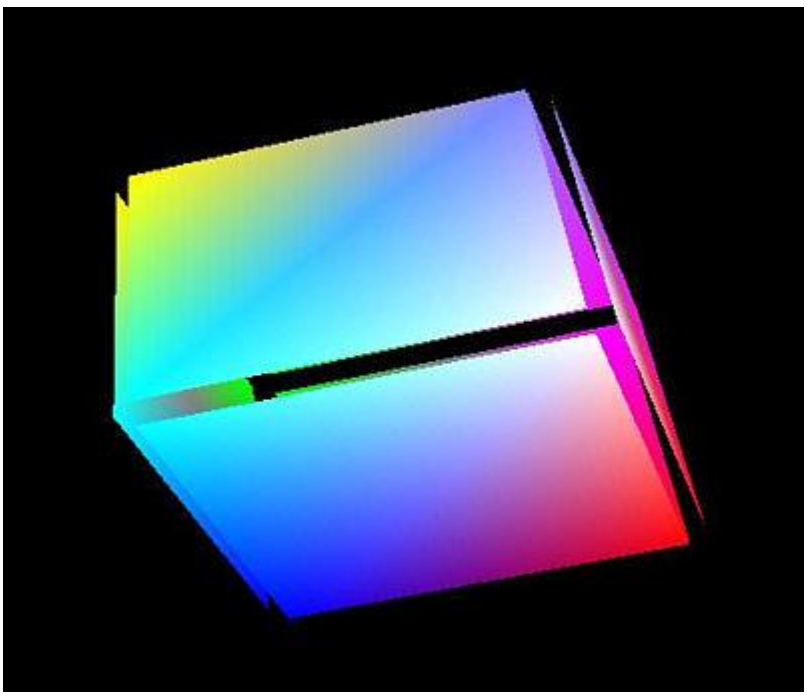
```
Private Function Initialize as boolean
.
.
.
Set
Texture=D3DX8.CreateTextureFromFile(D3DDevice,app.
x (yourfilename & path
end function
```

تابع Render نیز بصورت زیر خواهد بود :

```
Private Sub Render
D3DCLEAR_TARGET,0,1#,0,۰ D3DDevice.clear 0,byval
```

D3DDevice.beginscence
Texture,، D3DDevice.SetTexture
D3DDevice.DrawprimitiveUP
D3DPT_TRIANGLESTRIP,2,Tripstrip(0),len(Tristrip(0))
x
.
.
.
function end

آموزش DirectX-Graphic قسمت ششم



موضوع : مفاهیم اولیه رسم اشکال سه بعدی در DirectX 8

در این درس با استفاده از Direct3D یک مکعب را رسم می کنیم . برای

این منظور ابتدا نیاز به یک بافر داریم که بتوانیم شکل مورد نظر خود را در آن ذخیره کنیم :

```
Dim VBuffer as Direct3DVertexBuffer8
```

برای رسم مکعب از vertex های سه بعدی استفاده می کنیم . برای اینکار نیاز به تعریف یک تایپ جدید داریم :

```
Private Type LITVERTEX
x as single
single y as
z as single
color as long
specular as long
tu as single
as single tv
end type
```

توصیف گر این فرمت ، بصورت زیر است :

```
Const Lit_FVF = (D3DFVF_XYZ Or D3DFVF_DIFFUSE
D3DFVF_SPECULAR Or D3DFVF_TEX1)x Or
```

برای توصیف مکعب در این درس از روشی غیرکارامد استفاده شده است . به این ترتیب که از ۳۶ عدد vertex استفاده شده (در درسهای

بعدی متدهایی معرفی خواهند شد که اجازه می دهند از ۸ عدد vertex باری توصیف مکعب استفاده کنید) .

```
Dim cube(35) as LITVERTEX
```

سپس باید یکسری ماتریس سه بعدی تعریف کنیم : اولین ماتریس ، matworld است که نشان می دهد چگونه vertex ها در فضای سه بعدی قرار گرفته اند . دومین ماتریس ، matview است که نشان می دهد دوربین (نقطه دید) در کجا قرار گرفته و سومین ماتریس ، matproj است که نشان می دهد دوربین چگونه دنیای سه بعدی را روی صفحه دو بعدی نشان می دهد :

```
Dim matworld as D3DMATRIX
D3DMATRIX Dim matview as
Dim matproj as D3DMATRIX
```

در تابع Initialize قبل از ساخت device بایستی چک کنیم که آیا می توانیم از یک بافر Z شانزده بیتی استفاده کنیم یا نه ؟

```
,If D3D.CheckDeviceFormat(D3DADAPTER_DEFAULT
D3DDEVTYPE_HAL, DispMode.Format,
,D3DUSAGE_DEPTHSTENCIL, D3DRTYPE_SURFACE
D3DFMT_D16) = D3D_OK Then
D3DWindow.AutoDepthStencilFormat = D3DFMT_D16
bit Z-Buffer '16
```

حال بایستی متد D3DCreateDevice را اجرا کنید . سپس باید سیستم سایه زنی vertex را با فرمت vertex مان تنظیم کنیم :

```
D3DDevice.SetVertexShader Lit_FVF
```

همچنین سیستم نورپردازی را غیر فعال می کنیم :

```
False ,D3DDevice.SetRenderState D3DRS_LIGHTING
```

Direct3D هیچ مثلثی را که در دید شما نباشد رسم نخواهد کرد . برای متوقف کردن این امر بایستی حالت culling آنرا متوقف کنید همچنین vertex ها را بترتیب عقربه های ساعت معرفی کنید :

```
,D3DDevice.SetRenderState D3DRS_CULLMODE
D3DCULL_NONE
```

سپس باید فرمت بافر Z را فعال سازید :

```
D3DDevice.SetRenderState D3DRS_ZENABLE, 1
```

حال به بخش تعریف ماتریسها می رسیم :

آموزش DirectX-Graphic قسمت هفتم

در بحث قبلی به تعریف ماترسها رسیدیم .

۱ – World Matrix : این ماتریس برای نگهداری تمام vertex هایی که برای رندر فرستاده می شوند بکار می رود . مقادیر موجود در این ماتریس ، موقعیت یک vertex را می تواند تغییر دهد . یکی از کاربردهای آن انجام دوران rotation ، انتقال transmittion و تغییر اندازه scaling است .
برای ساخت این ماتریس از دستور زیر استفاده می کنیم :

```
D3DXMatrixIdentify matworld
```

حال این ماتریس را برای device مربوطه تایید می کنیم :

```
D3DDevice.SetTransform D3DTS_WORLD,matworld
```

۲ – View Matrix : این ماتریس را بعنوان یک دوربین در نظر بگیرید که بوسیله یک نقطه شروع و یک نقطه پایانی مشخص می شود (مشابه یک up vector که معمولاً در طول محور y رو به بالاست) :

```
,(D3DXMatrixLookAtLH matView, MakeV(0, 5, 9
MakeV(0, 0, 0),MakeV(0, 1, 0) x
matView ,D3DDevice.SetTransform D3DTS_VIEW
```

تابع MakeV که در اینجا استفاده شده بصورت زیر است :

```
Private Function MakeV(x As Single, y As Single, z As Single) As D3DVECTOR
MakeV.x = x
MakeV.y = y
MakeV.z = z
Function End
```

۳ – Projection Matrix : این ماتریس مشخص می کند چه منطقه ای از فضای جهانی برای رندر کردن visible باشد . همچنین مشخص می کند چه مقدار می توانیم بطور افقی ببینیم (زاویه دید بزرگتر منجر به دید بزرگتر می شود) :

```
D3DXMatrixPerspectiveFovLH matProj, pi / 4, 1, ۰,۱, ۵۰۰
```

در دستور فوق از زاویه دید pi/4 رادیان استفاده شده همچنین نسبت ۱:۱ استفاده شده است . قسمتهای سوم و چهارم مشخص می کنند فقط مثلثهایی کشیده شوند که با ابعاد بزرگتر از یکدهم دوربین و کوچکتر از ۵۰۰ برابر دوربین هستند .
حال دستور اختصاص به device را خواهیم داشت :

```
D3DDevice.SetTransform D3DTS_PROJECTION, matProj
```

بعد از تعریف ماتریسها بایستی تابع InitializeGeometry را صدا کنیم . در این تابع از یک ثابت با نام DFC استفاده شده است . اگر DFC=1 باشد مکعب بطور کامل کشیده می شود و اگر بزرگتر از یک باشد ، face های آن جدا از هم دیده خواهند شد . همچنین توجه کنید که از بافرهای vertex برای ذخیره داده vertex ها استفاده شده است . ساختار این تابع بصورت زیر خواهد بود :

۱ - پر کردن ساختارهای vertex

```
Front'
color, 0, 0, 0)x ,Cube(0) = CreateLitVertex(-1, 1, DFC
x(۰ ,Cube(1) = CreateLitVertex(1, 1, DFC, color, 0, 0
Cube(2) = CreateLitVertex(-1, -1, DFCcolor, 0, 0, 0)x
CreateLitVertex(-1, -1, DFC, color, 0, 0, 0)x = (Cube(4
DFC, color, 0, 0, 0)x ,۱- ,Cube(5) = CreateLitVertex(1
Back'
color, 0, 0, 0)x ,Cube(6) = CreateLitVertex(-1, 1, -DFC
x(۰ ,Cube(7) = CreateLitVertex(1, 1, -DFC, color, 0, 0
Cube(8) = CreateLitVertex(-1, -1, -DFC, color, 0, 0, 0)x
CreateLitVertex(1, 1, -DFC, color, 0, 0, 0)x = (Cube(9
DFC, color, 0, 0, 0)x- ,۱- ,Cube(10) = CreateLitVertex(-1
x(۰ ,۰ ,Cube(11) = CreateLitVertex(1, -1, -DFC, color, 0
Right'
x(۰ ,Cube(12) = CreateLitVertex(-DFC, 1, -1, color, 0, 0
Cube(13) = CreateLitVertex(-DFC, 1, 1, color, 0, 0, 0)x
CreateLitVertex(-DFC, -1, -1, color, 0, 0, 0)x = (Cube(14
CreateLitVertex(-DFC, 1, 1, color, 0, 0, 0)x = (Cube(15
```

```
color, 0, 0, 0)x ,۱- ,۱- ,Cube(16) = CreateLitVertex(-DFC
x(۰ ,Cube(17) = CreateLitVertex(-DFC, -1, 1, color, 0, 0
Left'
x(۰ ,Cube(18) = CreateLitVertex(DFC, 1, -1, color, 0, 0
Cube(20) = CreateLitVertex(DFC, -1, -1, color, 0, 0, 0)x
CreateLitVertex(DFC, 1, 1, color, 0, 0, 0)x = (Cube(21
color, 0, 0, 0)x ,۱- ,۱- ,Cube(22) = CreateLitVertex(DFC
x(۰ ,Cube(23) = CreateLitVertex(DFC, -1, 1, color, 0, 0
Top'
x(۰ ,Cube(24) = CreateLitVertex(-1, DFC, 1, color, 0, 0
Cube(25) = CreateLitVertex(1, DFC, 1, color, 0, 0, 0)x
CreateLitVertex(-1, DFC, -1, color, 0, 0, 0)x = (Cube(26
DFC, 1, cocolor, 0, 0, 0)x ,Cube(27) = CreateLitVertex(1
x(۰ ,۰ ,Cube(29) = CreateLitVertex(1, DFC, -1, color, 0
Bottom'
x(۰ ,Cube(30) = CreateLitVertex(-1, -DFC, 1, color, 0, 0
Cube(31) = CreateLitVertex(1, -DFC, 1, color, 0, 0, 0)x
CreateLitVertex(-1, -DFC, -1, color, 0, 0, 0)x = (Cube(32
DFC, 1, color, 0, 0, 0)x- ,Cube(33) = CreateLitVertex(1
x(۰ ,۰ ,Cube(34) = CreateLitVertex(-1, -DFC, -1, color, 0
Cube(35) = CreateLitVertex(1, -DFC, -1, color, 0, 0, 0)x
```

۲ – ساخت یک بافر vertex خالی با سایز مورد نظر :

```
= Set VBuffer
D3DDevice.CreateVertexBuffer(Len(Cube(0)) * 36, 0,
D3DPOOL_DEFAULT)x ,Lit_FVF
```

۳ – پر کردن بافر مربوطه با داده ها :

,۳۶ * ((D3DVertexBuffer8SetData VBuffer, 0, Len(Cube(0
Cube(0)x ,۰

حال به سراغ روتین Render می رویم :

```
Public Sub Render
D3DCLEAR_TARGET Or  ,D3DDevice.Clear 0, ByVal 0
black D3DCLEAR_ZBUFFER, 0, 1#, 0 '//Clear the screen
D3DDevice.BeginScene
Len(Cube(0))x ,D3DDevice.SetStreamSource 0, VBuffer
,D3DDevice.DrawPrimitive D3DPT_TRIANGLELIST, 0
۱۲
D3DDevice.EndScene
۰ D3DDevice.Present ByVal 0, ByVal 0, 0, ByVal
End Sub
```

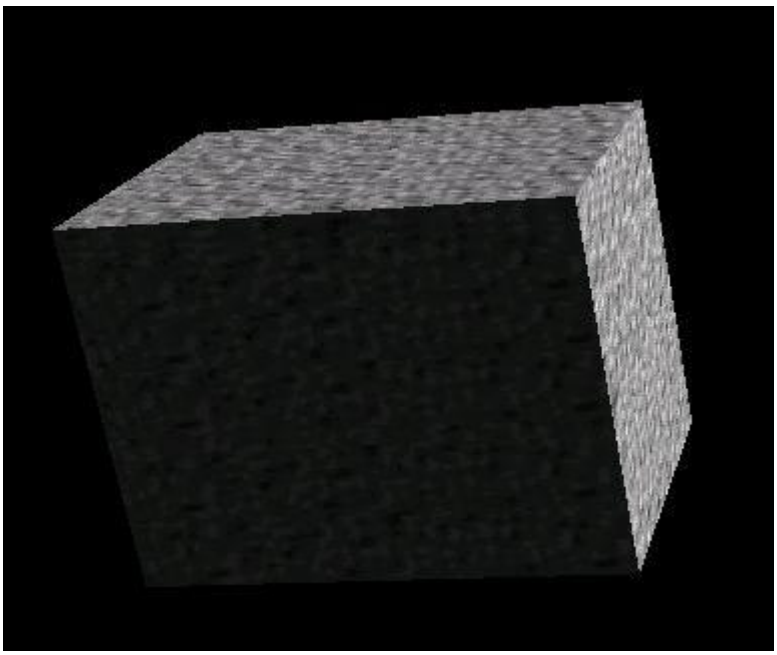
ساختار اصلی برنامه بصورت زیر خواهد بود :

```
Dim RotateAngle As Single
D3DMATRIX '//To hold temporary Dim matTemp As
call Initialize
bRunning Do While
RotateAngle = RotateAngle + 0.1
```

```
RotateAngle = RotateAngle -  Then ۳۶۰ =< If RotateAngle
360
world matrix D3DXMatrixIdentity matWorld '//Reset our
D3DXMatrixIdentity matTemp
RotateAngle * (pi / 180)  ,D3DXMatrixRotationX matTemp
x
matTemp ,D3DXMatrixMultiply matWorld, matWorld
D3DXMatrixIdentity matTemp
RotateAngle * (pi / 180)  ,D3DXMatrixRotationZ matTemp
x
matTemp ,D3DXMatrixMultiply matWorld, matWorld
matWorld ,D3DDevice.SetTransform D3DTS_WORLD
Render
DoEvents
Loop
```

آموزش DirectX-Graphic قسمت هشتم

موضوع : نورپردازی و اختصاص بافت به اشیا سه بعدی



در این درس می خواهیم به مکعب درس قبل بافت اختصاص داده و نیز آنرا با یک منبع نور ، نورپردازی کنیم .

ابتدا تایپ vertex ها را بصورت زیر تعریف می کنیم :

```
Private Type UnlitVertex
X As Single
Single Y As
Z As Single
nx As Single
ny As Single
nz As Single
Single tu As
tv As Single
End Type
```

توصیفگر این فرمت بصورت زیر خواهد بود :

```
Const Unlit_FVF = (D3DFVF_XYZ Or
(Or D3DFVF_TEX1 D3DFVF_NORMAL
```

همچنین مکعب ما توسط ارایه زیر مشخص می شود :

```
Dim Cube2(35) As UnlitVertex
```

دو ثابت pi و rad را نیز بصورت زیر تعریف می کنیم :

```
Const pi As Single = 3.141592
۱۸۰ / Const Rad = pi
```

برای اختصاص بافت به مکعب ، از شی Direct3DTexture8 استفاده می شود :

```
Dim CubeTexture As Direct3DTexture8
```

برای نورپردازی ، از شی D3DLIGHT8 استفاده می شود :

```
Dim Lights As D3DLIGHT8
```

تغییرات مورد نیاز در تابع Initialize

بعد از ساخت شی D3DDevice در این تابع ، پارامترهای آنرا بصورت زیر تنظیم می کنیم :

```
Unlit_FVF D3DDevice.SetVertexShader
١ ,D3DDevice.SetRenderState D3DRS_LIGHTING
D3DDevice.SetRenderState D3DRS_ZENABLE, 1
,D3DRS_AMBIENT D3DDevice.SetRenderState
H202020&
```

مقدار ambient یک کد هگزا RRGGBB است .

بعد از دستورات فوق ماتریسهای matworld، matview و matproj مطابق مطابل درس قبل تعریف می شوند . پس از آن بایستی بافت مکعب را از درون فایل تصویری مورد نظرتان load کنید :

```
= Set CubeTexture
D3DX.CreateTextureFromFileEx(D3DDevice,
,yourfilename, 128, 128, D3DX_DEFAULT, 0
DispMode.Format, D3DPOOL_MANAGED,
,D3DX_FILTER_LINEAR, D3DX_FILTER_LINEAR, 0
ByVal 0, ByVal 0)x
```

حال بایستی تابع InitializeGeometry صدا زده شود و سپس تابع

SetupLights فراخوانی شوند . ابتدا به توضیح تابع
InitializeGeometry می پردازیم :

```
Boolean Private Function InitialiseGeometry() As
```

ابتدا یک بردار نرمال تعریف می کنیم :

```
Dim vN As D3DVECTOR
```

سپس آرایه cube2 را با مقادیر عددی پر می کنیم . نرمالهای تمام
vertex ها را ابتدا با بردار
[۰،۰،۰] تعریف می کنیم . این مقدا بعداً تغییر خواهد کرد :

```
(۰ ,Cube2(0) = CreateVertex(-1, -1, 1, 0, 0, 0, 0
(Cube2(1) = CreateVertex(1, 1, 1, 0, 0, 0, 1, 1
(CreateVertex(-1, 1, 1, 0, 0, 0, 0, 1 = (Cube2(2
Cube2(1),  ,(vN = GenerateTriangleNormals(Cube2(0
((Cube2(2
= Cube2(0).nx = vN.X: Cube2(0).ny = vN.Y: Cube2(0).nz
vN.Z
= Cube2(1).nx = vN.X: Cube2(1).ny = vN.Y: Cube2(1).nz
vN.Z
= Cube2(2).nx = vN.X: Cube2(2).ny = vN.Y: Cube2(2).nz
vN.Z
```

```
(Cube2(3) = CreateVertex(1, 1, 1, 0, 0, 0, 1, 1
(CreateVertex(-1, -1, 1, 0, 0, 0, 0, 0 = (Cube2(4
(۰ ,۱ ,۰ ,۰ ,Cube2(5) = CreateVertex(1, -1, 1, 0
,(vN = GenerateTriangleNormals(Cube2(3), Cube2(4
((Cube2(5
= Cube2(3).nx = vN.X: Cube2(3).ny = vN.Y: Cube2(3).nz
vN.Z
= Cube2(4).nx = vN.X: Cube2(4).ny = vN.Y: Cube2(4).nz
vN.Z
= Cube2(5).nx = vN.X: Cube2(5).ny = vN.Y: Cube2(5).nz
vN.Z

Back'
(۱ ,Cube2(6) = CreateVertex(-1, 1, -1, 0, 0, 0, 0
(Cube2(7) = CreateVertex(1, 1, -1, 0, 0, 0, 1, 1
(CreateVertex(-1, -1, -1, 0, 0, 0, 0, 0 = (Cube2(8
GenerateTriangleNormals(Cube2(6), Cube2(7), = vN
((Cube2(8
Cube2(6).ny = vN.Y: Cube2(6).nz = :Cube2(6).nx = vN.X
vN.Z
vN.Y: Cube2(7).nz = = Cube2(7).nx = vN.X: Cube2(7).ny
vN.Z
= Cube2(8).nx = vN.X: Cube2(8).ny = vN.Y: Cube2(8).nz
vN.Z
```

```
(Cube2(9) = CreateVertex(1, -1, -1, 0, 0, 0, 1, 0
(CreateVertex(-1, -1, -1, 0, 0, 0, 0, 0 = (Cube2(10
(۱ ,۱ ,۰ ,۰ ,Cube2(11) = CreateVertex(1, 1, -1, 0
,(vN = GenerateTriangleNormals(Cube2(9), Cube2(10
((Cube2(11
= Cube2(9).nx = vN.X: Cube2(9).ny = vN.Y: Cube2(9).nz
vN.Z
Cube2(10).nx = vN.X: Cube2(10).ny = vN.Y: Cube2(10).nz
vN.Z =
Cube2(11).nx = vN.X: Cube2(11).ny = vN.Y: Cube2(11).nz
vN.Z =

Right'
(۰ ,Cube2(12) = CreateVertex(-1, -1, -1, 0, 0, 0, 0
(Cube2(13) = CreateVertex(-1, 1, 1, 0, 0, 0, 1, 1
(CreateVertex(-1, 1, -1, 0, 0, 0, 1, 0 = (Cube2(14
GenerateTriangleNormals(Cube2(12), Cube2(13), = vN
((Cube2(14
Cube2(12).ny = vN.Y: Cube2(12).nz :Cube2(12).nx = vN.X
= vN.Z
vN.Y: Cube2(13).nz = Cube2(13).nx = vN.X: Cube2(13).ny
= vN.Z
Cube2(14).nz :Cube2(14).nx = vN.X: Cube2(14).ny = vN.Y
= vN.Z

(۱ ,Cube2(15) = CreateVertex(-1, 1, 1, 0, 0, 0, 1
(Cube2(16) = CreateVertex(-1, -1, -1, 0, 0, 0, 0, 0
```

```
(CreateVertex(-1, -1, 1, 0, 0, 0, 0, 1 = (Cube2(17
GenerateTriangleNormals(Cube2(15), Cube2(16),  = vN
((Cube2(17
Cube2(15).ny = vN.Y: Cube2(15).nz  :Cube2(15).nx = vN.X
= vN.Z
vN.Y: Cube2(16).nz  = Cube2(16).nx = vN.X: Cube2(16).ny
= vN.Z
Cube2(17).nz  :Cube2(17).nx = vN.X: Cube2(17).ny = vN.Y
= vN.Z

Left'
(۰ ,۱ ,Cube2(18) = CreateVertex(1, 1, -1, 0, 0, 0
(Cube2(19) = CreateVertex(1, 1, 1, 0, 0, 0, 1, 1
(CreateVertex(1, -1, -1, 0, 0, 0, 0, 0 = (Cube2(20
GenerateTriangleNormals(Cube2(18), Cube2(19),  = vN
((Cube2(20
Cube2(18).ny = vN.Y: Cube2(18).nz  :Cube2(18).nx = vN.X
= vN.Z
vN.Y: Cube2(19).nz  = Cube2(19).nx = vN.X: Cube2(19).ny
= vN.Z
Cube2(20).nz  :Cube2(20).nx = vN.X: Cube2(20).ny = vN.Y
= vN.Z

(۱ ,Cube2(21) = CreateVertex(1, -1, 1, 0, 0, 0, 0
(Cube2(22) = CreateVertex(1, -1, -1, 0, 0, 0, 0, 0
(CreateVertex(1, 1, 1, 0, 0, 0, 1, 1 = (Cube2(23
Cube2(22),  ,(vN = GenerateTriangleNormals(Cube2(21
```

```
((Cube2(23
Cube2(21).nx = vN.X: Cube2(21).ny = vN.Y: Cube2(21).nz
vN.Z =
Cube2(22).nx = vN.X: Cube2(22).ny = vN.Y: Cube2(22).nz
vN.Z =
Cube2(23).nx = vN.X: Cube2(23).ny = vN.Y: Cube2(23).nz
vN.Z =

Top'
(١ ,Cube2(24) = CreateVertex(-1, 1, 1, 0, 0, 0, 0
(Cube2(25) = CreateVertex(1, 1, 1, 0, 0, 0, 1, 1
(CreateVertex(-1, 1, -1, 0, 0, 0, 0, 0 = (Cube2(26
GenerateTriangleNormals(Cube2(24), Cube2(25),  = vN
((Cube2(26
Cube2(24).ny = vN.Y: Cube2(24).nz  :Cube2(24).nx = vN.X
= vN.Z
vN.Y: Cube2(25).nz  = Cube2(25).nx = vN.X: Cube2(25).ny
= vN.Z
Cube2(26).nz  :Cube2(26).nx = vN.X: Cube2(26).ny = vN.Y
= vN.Z

(٠ ,Cube2(27) = CreateVertex(1, 1, -1, 0, 0, 0, 1
(Cube2(28) = CreateVertex(-1, 1, -1, 0, 0, 0, 0, 0
(CreateVertex(1, 1, 1, 0, 0, 0, 1, 1 = (Cube2(29
Cube2(28),  ,(vN = GenerateTriangleNormals(Cube2(27
((Cube2(29
Cube2(27).nx = vN.X: Cube2(27).ny = vN.Y: Cube2(27).nz
vN.Z =
```

```
Cube2(28).nx = vN.X: Cube2(28).ny = vN.Y: Cube2(28).nz
vN.Z =
Cube2(29).nx = vN.X: Cube2(29).ny = vN.Y: Cube2(29).nz
vN.Z =

Top'
(٠ ,Cube2(30) = CreateVertex(-1, -1, -1, 0, 0, 0, 0
(Cube2(31) = CreateVertex(1, -1, 1, 0, 0, 0, 1, 1
(CreateVertex(-1, -1, 1, 0, 0, 0, 0, 1 = (Cube2(32
GenerateTriangleNormals(Cube2(30), Cube2(31),  = vN
((Cube2(32
Cube2(30).ny = vN.Y: Cube2(30).nz  :Cube2(30).nx = vN.X
= vN.Z
vN.Y: Cube2(31).nz  = Cube2(31).nx = vN.X: Cube2(31).ny
= vN.Z
Cube2(32).nz  :Cube2(32).nx = vN.X: Cube2(32).ny = vN.Y
= vN.Z

(١ ,Cube2(33) = CreateVertex(1, -1, 1, 0, 0, 0, 1
(Cube2(34) = CreateVertex(-1, -1, -1, 0, 0, 0, 0, 0
(CreateVertex(1, -1, -1, 0, 0, 0, 1, 0 = (Cube2(35
GenerateTriangleNormals(Cube2(33), Cube2(34),  = vN
((Cube2(35
Cube2(33).ny = vN.Y: Cube2(33).nz  :Cube2(33).nx = vN.X
= vN.Z
vN.Y: Cube2(34).nz  = Cube2(34).nx = vN.X: Cube2(34).ny
= vN.Z
```

Cube2(35).nz :Cube2(35).nx = vN.X: Cube2(35).ny = vN.Y = vN.Z

سپس یک بافر vertex خالی با سایر موردنظر می سازیم :

= Set VBuffer
D3DDevice.CreateVertexBuffer(Len(Cube2(0)) * 36, 0, D3DPOOL_DEFAULT)x ,Unlit_FVF

سپس این بافر vertex ساخته شده را با داده های cube2 پر می کنیم :

Len(Cube2(0)) * 36, ,D3DVertexBuffer8SetData VBuffer, 0 0, Cube2(0)x

در دستورات فوق تابعی با نام GenerateTraingleNormals استفاده شده است . این تابع دو بردار را از روی سه vertex داده شده با آن می سازد و سپس ضرب برداری ایندو را حساب می کند و سپس بردار حاصله را نرمال می نماید :

Private Function GenerateTriangleNormals(p0 As UnlitVertex, p1 As UnlitVertex, p2 As UnlitVertex) As D3DVECTOR
D3DVECTOR 'Vector from points 0 to 1 Dim v01 As to 2 ، Dim v02 As D3DVECTOR 'Vector from points
Dim vNorm As D3DVECTOR 'The final vector

```
from points 0 to 1 and 0 to 2 Create the vectors'
p1.Z),  ,D3DXVec3Subtract v01, MakeVector(p1.X, p1.Y
(MakeVector(p0.X, p0.Y, p0.Z
p2.Y, p2.Z),  ,D3DXVec3Subtract v02, MakeVector(p2.X
(MakeVector(p0.X, p0.Y, p0.Z


product Get the cross'
D3DXVec3Cross vNorm, v01, v02


vector Normalize this'
D3DXVec3Normalize vNorm, vNorm


value Return the'
GenerateTriangleNormals.X = vNorm.X
vNorm.Y = GenerateTriangleNormals.Y
GenerateTriangleNormals.Z = vNorm.Z
End Function
```

حال به توضیح تابع SetupLights می پردازیم . در این تابع دو شی D3DMATERIAL8 و D3DCOLORVALUE استفاده شده است :

```
Private Function SetupLights() As Boolean
Mtrl As D3DMATERIAL8, Col As  Dim
D3DCOLORVALUE
Col.b = 1 :Col.a = 1: Col.r = 1: Col.g = 1
Mtrl.Ambient = Col
Mtrl.diffuse = Col
```

```
Mtrl D3DDevice.SetMaterial

Lights.Type = D3DLIGHT_DIRECTIONAL
١ = Lights.diffuse.r
Lights.diffuse.g = 1
Lights.diffuse.b = 1
(MakeVector(1, -1, 0 = Lights.Direction

D3DDevice.SetLight 0, Lights

True = SetupLights
End Function
```

تابع Render بصورت زیر است :

```
()Public Sub Render
D3DCLEAR_TARGET Or  ,٠ D3DDevice.Clear 0, ByVal
black D3DCLEAR_ZBUFFER, 0, 1#, 0 '//Clear the screen
D3DDevice.BeginScene
Draw the cube'
CubeTexture ,D3DDevice.SetTexture 0
((Len(Cube2(0 ,D3DDevice.SetStreamSource 0, VBuffer
,D3DDevice.DrawPrimitive D3DPT_TRIANGLELIST, 0
١٢
D3DDevice.EndScene
٠ D3DDevice.Present ByVal 0, ByVal 0, 0, ByVal
End Sub
```

ساختار اصلی برنامه بصورت زیر است :

```
Call Initialise
bRunning Do While
RotateAngle = RotateAngle + 0.1
RotateAngle = RotateAngle -  Then ۳۶۰ =< If RotateAngle
360

matWorld D3DXMatrixIdentity

D3DXMatrixIdentity matTemp
(RotateAngle * (pi / 180 ,D3DXMatrixRotationX matTemp
matTemp ,D3DXMatrixMultiply matWorld, matWorld

D3DXMatrixIdentity matTemp
(RotateAngle * (pi / 180 ,D3DXMatrixRotationY matTemp
matTemp ,D3DXMatrixMultiply matWorld, matWorld

D3DXMatrixIdentity matTemp
(RotateAngle * (pi / 180 ,D3DXMatrixRotationZ matTemp
matTemp ,D3DXMatrixMultiply matWorld, matWorld


matWorld ,D3DDevice.SetTransform D3DTS_WORLD
```

D3DDevice.LightEnable 0, 0 خاموش کردن light

D3DDevice.LightEnable 0, 1 روشن کردن light

Render
DoEvents
Loop

در متد D3DDevice.LightEnable پارامتر اول شماره منبع نور و پارمتر دوم enable بودن آنرا نشان می دهد

آموزش DirectX8 قسمت نهم

موضوع :ترسیم متن دو بعدی در DirectX

در این درس روش ترسیم متن با دو نوع فونت را نشان خواهم داد :
برای رسم یک متن با فونت تعریف شده در سیستم از شی D3DXFont استفاده می کنیم :

```
Dim MainFont as D3DXFont
IFont Dim MainFontDesc as
Dim TextRect as RECT
Dim fnt as new stdFont
```

در حالیکه برای ایجاد یک متن با فونت custom ابتدا یک texture تعریف می کنیم :

Dim fntTex as Direct3DTexture8

همچنین برای ترسیم هر کاراکتر یک آرایه vertex ای را از نوع TLVERTEX تعریف می نمائیم :

Dim vertchar(3) as TLVERTEX

حال به سراغ تابع Initialize می رویم . در این تابع ابتدا دستورات مربوط به ایجاد اشیا D3D و D3Dx را قرا دهید سپس دستورات مربوط به اختصاص آداپتور و نیز ایجاد شی D3DDevice را انجام می دهیم . حال دستورات تنظیم shader و rendering را می آوریم :

TL_FVF D3DDevice.SetVertexShader
D3DDevice.SetRenderState D3DRS_LIGHTING, False

سپس تنظیمات پارامترهای transparency برای rendering را انجام می دهیم :

,D3DDevice.SetRenderState D3DRS_SRCBLEND
D3DBLEND_SRCALPHA
,D3DDevice.SetRenderState D3DRS_DESTBLEND
D3DBLEND_INVSRCALPHA

```
D3DDevice.SetRenderState
True ,D3DRS_ALPHABLENDENABLE
```

حال بایستی textureرا طوری فیلتر کنیم که در زمان stretch شدن یا squash شدن بهتر بنظر برسد :

```
,D3DDevice.SetTextureStageState 0
D3DTSS_MAGFILTER, D3DTEXF_LINEAR
,D3DDevice.SetTextureStageState 0
D3DTSS_MINFILTER, D3DTEXF_LINEAR
```

حال فیلتر Z را فعال می کنیم :

```
D3DDevice.SetRenderState D3DRS_ZENABLE, 1
```

سپس ماتریسهای world، view و projection را تنظیم می کنیم :

```
matWorld D3DXMatrixIdentity
D3DDevice.SetTransform D3DTS_WORLD, matWorld
matView, MakeVector(0, 9, -9),  D3DXMatrixLookAtLH
(۰ ,MakeVector(0, 0, 0), MakeVector(0, 1
D3DDevice.SetTransform D3DTS_VIEW, matView
matProj, pi / 4, 1, 0.1, 500 D3DXMatrixPerspectiveFovLH
,D3DDevice.SetTransform D3DTS_PROJECTION
matProj
```

حال به بخش تنظیم پارامترهای فونت می رسیم . در مورد فونت دو بعدی عادی :

```
fnt.Name = "Verdana"x
۱۸ = fnt.Size
fnt.Bold = True
Set MainFontDesc = fnt
D3DX.CreateFont(D3DDevice,  = Set MainFont
MainFontDesc.hFont)x
```

و در مورد فونت custom :

```
D3DX.CreateTextureFromFileEx(D3DDevice,  = Set fntTex
,yourfilename, 256, 128, D3DX_DEFAULT, 0
D3DFMT_UNKNOWN, D3DPOOL_MANAGED,
,D3DX_FILTER_POINT, D3DX_FILTER_POINT
HFF00FF00, ByVal 0, ByVal 0)x&
end function
```

روتین Render بصورت زیر خواهد بود :

```
Public Sub Render()x
D3DCLEAR_TARGET Or  ,۰ D3DDevice.Clear 0, ByVal
D3DCLEAR_ZBUFFER, 0, 1#, 0
D3DDevice.BeginScene
```

برای رندر متن با فونت عادی بصورت زیر عمل می کنیم :

```
TextRect.Top = 440
۱ = TextRect.Left
TextRect.bottom = 480
TextRect.Right = 640
HFFCCCCFF, "Current & ,D3DX.DrawText MainFont
FPS_Current, TextRect, DT_TOP Or & " :Frame Rate
DT_CENTER
```

برای رندر متن با فونت custom بصورت زیر عمل می کنیم :

```
,"RenderStringFromCustomFont_2D "Hamed Sheidaian
۱, ۱, ۱۶, ۱۶
D3DDevice.EndScene
ByVal 0 ,D3DDevice.Present ByVal 0, ByVal 0, 0
End Sub
```

همانطور که مشاهده می کنید از روتینی با نام
RenderStringFromCustomFont_2D استفاده شده است :

```
RenderStringFromCustomFont_2D(strText  Private Sub
Single, Height As  As String, startX As Single, StartY As
Integer, Width As Integer)x
Dim I As Integer
```

```
As Integer, CharY As Integer Dim CharX
Dim Char As String
Integer Dim LinearEntry As
If Len(strText) = 0 Then Exit Sub
For I = 1 To Len(strText)x
```

۱ – ابتدا بایستی مختصات texture را انتخاب کنیم . برای اینکار بایستی هر entry را در texture جدا کنیم :

```
Char = Mid$(strText, I, 1)x
Then ۹۰ => (And Asc(Char ۶۵ =< (If Asc(Char
LinearEntry = Asc(Char) - 65
Then ۱۲۲ => (And Asc(Char ۹۷ =< (Asc(Char ElseIf
LinearEntry = Asc(Char) - 71
Then ۵۷ => (And Asc(Char ۴۸ =< (ElseIf Asc(Char
Asc(Char) + 4 = LinearEntry
ElseIf Char = " " Then
LinearEntry = 63
Then "." = ElseIf Char
LinearEntry = 62
ElseIf Char = ";" Then
LinearEntry = 66
Char = "/" Then ElseIf
LinearEntry = 64
ElseIf Char = "," Then
۶۵ = LinearEntry
End If
```

بعد از مقداردهی LinearEntry بایستی مختصات grid کاراکتر را پردازش کنیم :

```
Then ۱۵ => If LinearEntry
۰ = CharY
CharX = LinearEntry
End If
Then ۳۱ => And LinearEntry ۱۶ =< If LinearEntry
CharY = 1
CharX = LinearEntry - 16
End If
Then ۴۷ => And LinearEntry ۳۲ =< LinearEntry If
CharY = 2
LinearEntry - 32 = CharX
End If
Then ۶۳ => And LinearEntry ۴۸ =< If LinearEntry
CharY = 3
CharX = LinearEntry - 48
End If
Then ۷۹ => And LinearEntry ۶۴ =< If LinearEntry
CharY = 4
CharX = LinearEntry - 64
If End
```

۲ – حال بایستی vertex های مورد نیاز برای رسم کاراکتر را تولید کنیم :

```
I),  * vertChar(0) = CreateTLVertex(startX + (Width
HFFFFF& ,StartY, 0, 1
F,0,(1/16)*CharX,(1/8)*CharY)
CreateTLVertex(startX + (Width * I) +  = (vertChar(1
HFFFFF& ,Width, StartY, 0, 1
CharX) + (1 / 16), (1 / 8) * CharY * (۱۶ / ۱)),۰ ,F)
CreateTLVertex(startX + (Width * I),  = (vertChar(2
HFFFFF& ,StartY + Height, 0, 1
CharX, ((1 / 8) * CharY) + (1 / 8 * (۱۶ / ۱) ,۰ ,F))
CreateTLVertex(startX + (Width * I) +  = (vertChar(3
* (۱۶ / ۱)) ,Width, StartY + Height, 0, 1, HFFFFFF, 0
CharX) + (1 / 16), ((1 / 8) * CharY) + (1 / 8))x
```

۳ – رندر vertex ها :

```
fntTex ,D3DDevice.SetTexture 0
D3DDevice.DrawPrimitiveUP D3DPT_TRIANGLESTRIP,
Len(vertChar(0))x ,(2, vertChar(0
Next I
End Sub
```

آموزش DirectX-Graphic قسمت دهم

موضوع : ترسیم اشیا سه بعدی با استفاده از شی Mesh

شی Mesh که جزو اشیا D3DX می باشد امکان ترسیم اشیا سه بعدی

پایه و همچنین ترسیم مش های custom دلخواه را به شما می دهد . در این درس از شی Mesh برای ترسیم یک کره (sphere) استفاده می کنیم . ابتدا متغیر sphere را بصورت زیر تعریف کنید :

```
Dim sphere as D3DXMesh
```

همچنین برای نورپردازی و اختصاص material به کره به متغیرهای زیر نیاز داریم :

```
Dim d3dLight As D3DLIGHT8
D3DMATERIAL8 Dim material As
Dim Col As D3DCOLORVALUE
```

در تابع Initial پس از ساخت اشیا D3D و D3DX و D3DDevice بایستی پارامترهای رنگ ، نورپردازی و اختصاص ماده (material) به کره را بصورت زیر تنظیم کنید :

```
Col.a = 1
Col.b = 1
Col.g = 1
۱ = Col.r
d3dLight.Type = D3DLIGHT_DIRECTIONAL
Col = d3dLight.diffuse
d3dLight.Direction = vec(-1, -1, -1)x
```

نورپردازی از نوع جهت دار با رنگ col و بردار جهت (۱-,۱-,۱-) است .

نکته :

رنگ ambient رنگی است که هنگامیکه جسم در سایه باشد به خود می گیرد . بعبارت دیگر این رنگ را جسم وقتی که در معرض یک نور ambient باشد از خود منعکس می کند .

رنگ diffuse رنگی است که هنگامیکه جسم در معرض نور مستقیم قرار بگیرد از خود منعکس می کند .

```
material.Ambient = Col
Col = material.diffuse
d3dDevice.SetMaterial material
d3dLight ,d3dDevice.SetLight 0
d3dDevice.LightEnable 0, 1
```

سپس بایستی پارامترهای rendering را تنظیم کنید :

```
۱ ,d3dDevice.SetRenderState D3DRS_LIGHTING
d3dDevice.SetRenderState D3DRS_ZENABLE, 1
D3DRS_LIGHTING, 1 d3dDevice.SetRenderState
۱ ,d3dDevice.SetRenderState D3DRS_ZENABLE
,d3dDevice.SetRenderState D3DRS_SHADEMODE
D3DSHADE_GOURAUD
H202020& ,d3dDevice.SetRenderState D3DRS_AMBIENT
d3dDevice.SetTextureStageState 0,
D3DTEXF_LINEAR ,D3DTSS_MAGFILTER
```

```
d3dDevice.SetTextureStageState 0,
D3DTEXF_LINEAR ,D3DTSS_MINFILTER
```

حال بایستی شی sphere را بسازیم :

```
,۲۰ ,۱۰۰۰ ,Set Sphere = d3dx.CreateSphere(d3dDevice, 2
Nothing)x
```

که ۲ شعاع کره و ۱۰۰۰ تعداد slice هایی است که کره با آن ساخته می شود .

سپس بردارهای نقطه دید و مکان دوربین و رنگ زمینه را تنظیم کنید (viewpoint و camerapoint از نوع D3DVECTOR هستند) .

```
(ViewPoint = vec(0, 0, 0
(۴ ,۴ ,CameraPoint = vec(4
H404040& = BackColor
```

در روتین Render ابتدا ماتریسها و بردارهای صحنه را تنظیم می کنیم :

```
matWorld D3DXMatrixIdentity
d3dDevice.SetTransform D3DTS_WORLD, matWorld
matView, Rotation D3DXMatrixRotationY
D3DXMatrixLookAtLH matTemp, CameraPoint,
```

```
(۰ ,۱ ,ViewPoint, vec(0
D3DXMatrixMultiply matView, matView, matTemp
D3DTS_VIEW, matView d3dDevice.SetTransform
۵۰۰ ,D3DXMatrixPerspectiveFovLH matProj, pi / 4, 1, 0.1
d3dDevice.SetTransform D3DTS_PROJECTION, matProj
```

در پایان نیز شروع به رندر صحنه می کنیم :

```
d3dDevice.Clear 0, ByVal 0, D3DCLEAR_TARGET Or
۰ ,D3DCLEAR_ZBUFFER, BackColor, 1
d3dDevice.BeginScene
۰ Sphere.DrawSubset
d3dDevice.EndScene
۰ d3dDevice.Present ByVal 0, ByVal 0, 0, ByVal
```

مباحث پیشرفته Direct3D – مقدمه

موضوع : مروری بر مباحث قبلی – ساخت یک موتور گرافیکی سه بعدی

قبل از شروع مباحث جدید برنامه نویسی Direct3D ، با هم مروری بر مباحث قبلی خواهیم داشت . (مباحث قبلی در آرشیو موجود می باشند) .

در این درس با استفاده از مطالب قبلی یک Engine سه بعدی ساخته و از امکانات آن در یک برنامه نمونه استفاده خواهیم کرد .

این engine دارای دو کلاس است :

۱ – کلاس MainD3D

۲ – کلاس D3Dobject

در کلاس MainD3D متغیرها و توابع لازم برای ساخت یک device سه بعدی ، تنظیمات ماتریسی ، تابع رندر و غیره موجود می باشد .

متغیرهای عمومی این کلاس عبارتند از :

```
Public g_DX As New DirectX8
Direct3D8 Public g_D3D As
Public g_D3DX As New D3DX8
Direct3DDevice8 Public g_D3DDevice As
Public NTextures As Long
```

روتین ها و توابع این کلاس عبارتند از :

۱ – InitD3D : این روتین ، اشیا D3D و D3Ddevice را می سازد و پارامترهای آنها را تنظیم می کند .

۲ – ApplyCameraChanges : روتین ایجاد ماتریس View

۳ – SetupMatrices : روتین ایجاد ماتریس Projection

۴ – StartRender : در این روتین عملیات لازم برای شروع عمل رندر صورت می گیرد .

۵ – RenderObject : این تابع ، یک شی سه بعدی از نوع کلاس D3Dobject را می گیرد و بردارهای مورد نیاز و نیز بافت شی را تنظیم می کند و در پایان شی را ترسیم می کند .

۶ – FinishRender : در این روتین به عملیات رندر پایان داده می شود .

۷ – Cleanup: روتین از بین بردن اشیا Direct3D

۸ – CreateVector : تابع ساخت یک بردار سه بعدی

۹ – CreateTextures : روتین ساخت یک بافت جدید

۱۰ – InitTexture: تابع مقداردهی به یک بافت

در کلاس D3Dobject متغیرها و توابع لازم برای ایجاد یک شی سه بعدی و اختصاص بافت به آن موجود می باشد .

در این کلاس دو type عمومی تعریف شده است :

۱ – NormalVERTEX

۲ – TeturedVERTEX

همچنین روتین ها و توابع این کلاس عبارتند از :

۱ – InitObject : تابعی که تنظیمات اولیه vertex ها و بافت شی را انجام می دهد .

۲ – Vertex : روتین ایجاد vertex های مورد نیاز

۳ – GetRenderingMode: تابعی که مد رندر را مشخص می کند .

و نیز یکسری تابع ساخت vertex نرمال و ساخت vertex دارای بافت و غیره

این دو کلاس در یک پروژه ویژوال بیسیک قرارداده شده و پروژه با نام D3Dengine.dll کامپایل شده است .

حال با استفاده از این engine می خواهیم یک منظره سه بعدی را ایجاد کنیم :

این منظره شامل سه object است : دیوار ، آسمان و زمین.

ابتدا باید یک شی از کلاس MainD3D تعریف کنیم :

```
Dim D3D8Main As MainD3D8
```

در متد Form Load نیز سه شی Floor، Sky و Wall را بصورت زیر تعریف می کنیم :

```
Dim Floor As D3DObject
D3DObject Dim Sky As
Dim Walls As D3Dobject
```

سپس این سه شی را به اضافه شی D3D8Main ، ایجاد می کنیم :

```
Set D3D8Main = New D3DEngine.MainD3D8
New D3DEngine.D3DObject = Set Floor
Set Sky = New D3DEngine.D3DObject
D3DEngine.D3Dobject Set Walls = New
```

در ابتدا شی MainD3D را Initial می کنیم و سپس بافتهای مورد نیز خود را می سازیم :

```
Me.hWnd ,D3D8Main.InitD3D True
D3D8Main.CreateTextures 3
"floor.jpg\" + D3D8Main.InitTexture 1, App.Path
"sky.bmp\" + D3D8Main.InitTexture 2, App.Path
"D3D8Main.InitTexture 3, App.Path + "\wall.bmp
```

حال به سراغ ایجاد و مقداردهی vertex های floor می رویم . floor شامل شش vertex می باشد و بنابراین دو face مثلثی دارد :

۱ ,Floor.InitObject 6, 2, TriangleList, True

Floor.Vertex 0, -55, -2, -55, vbWhite, 0, 10
vbWhite, 10, 10 ,۵۵- ,Floor.Vertex 1, 55, -2
۰ ,Floor.Vertex 2, 55, -2, 55, vbWhite, 10
Floor.Vertex 3, -55, -2, -55, vbWhite, 0, 10
vbWhite, 10, 0 ,Floor.Vertex 4, 55, -2, 55
Floor.Vertex 5, -55, -2, 55, vbWhite, 0, 0

سپس به سراغ ایجاد و مقداردهی vertex های wall می رویم . wall شامل بیست و چهار vertex می باشد و بنابراین هشت face مثلثی دارد :

۳ ,Walls.InitObject 24, 8, TriangleList, True

HBCE8FC, 0, 1& ,Walls.Vertex 0, -55, -2, -55
HBCE8FC, 5, 1& ,۵۵- ,۲- ,Walls.Vertex 1, 55
۰ ,HBCE8FC, 5& ,Walls.Vertex 2, 55, 8, -55
HBCE8FC, 0, 1& ,Walls.Vertex 3, -55, -2, -55
HBCE8FC, 5, 0& ,۵۵- ,Walls.Vertex 4, 55, 8
۰ ,HBCE8FC, 0& ,Walls.Vertex 5, -55, 8, -55

HBCE8FC, 0, 1& ,Walls.Vertex 6, -55, -2, 55
HBCE8FC, 5, 1& ,۵۵ ,۲- ,Walls.Vertex 7, 55
۰ ,HBCE8FC, 5& ,Walls.Vertex 8, 55, 8, 55
HBCE8FC, 0, 1& ,Walls.Vertex 9, -55, -2, 55
HBCE8FC, 5, 0& ,۵۵ ,Walls.Vertex 10, 55, 8

۰ ,HBCE8FC, 0& ,Walls.Vertex 11, -55, 8, 55

HBCE8FC, 0, 1& ,Walls.Vertex 12, -55, -2, 55

HBCE8FC, 5, 1& ,۵۵- ,۲- ,۵۵- ,Walls.Vertex 13

۰ ,۵ ,HBCE8FC& ,Walls.Vertex 14, -55, 8, -55

HBCE8FC, 0, 1& ,Walls.Vertex 15, -55, -2, 55

HBCE8FC, 5, 0& ,۵۵- ,۸ ,۵۵- ,Walls.Vertex 16

۰ ,HBCE8FC, 0& ,Walls.Vertex 17, -55, 8, 55

HBCE8FC, 0, 1& ,Walls.Vertex 18, 55, -2, 55

HBCE8FC, 5, 1& ,۵۵- ,۲- ,Walls.Vertex 19, 55

۰ ,HBCE8FC, 5& ,Walls.Vertex 20, 55, 8, -55

HBCE8FC, 0, 1& ,Walls.Vertex 21, 55, -2, 55

HBCE8FC, 5, 0& ,۵۵- ,Walls.Vertex 22, 55, 8

۰ ,HBCE8FC, 0& ,Walls.Vertex 23, 55, 8, 55

حال به سراغ ایجاد و مقداردهی vertex های sky می رویم . sky شامل شش vertex می باشد و بنابراین دو face مثلثی دارد :

۲ ,Sky.InitObject 6, 2, TriangleList, True

HBCE8FC, 0, 1& ,Sky.Vertex 0, -55, 8, -55

HBCE8FC, 0, 1& ,۵۵- ,Sky.Vertex 1, 55, 8

۱ ,HBCE8FC, 0& ,Sky.Vertex 2, 55, 8, 55

HBCE8FC, 0, 1& ,Sky.Vertex 3, -55, 8, -55

```
HBCE8FC, 0, 1& ,Sky.Vertex 4, 55, 8, 55
HBCE8FC, 0, 1& ,Sky.Vertex 5, -55, 8, 55
```

در پایان تابع رندر را صدا می کنیم . البته در هر بار عمل رندر کردن ، دوربین یک درجه در صفحه X-Z دوران می کند تا کل دیوار قابل مشاهده باشد :

```
Dim Angle As Double
PI = 3.1415
۰ = Angle
Do
DoEvents
D3D8Main.StartRender vbBlack
Sky D3D8Main.RenderObject
D3D8Main.RenderObject Floor
Walls D3D8Main.RenderObject
D3D8Main.FinishRender
۰ = If Sqr(Angle ^ 2) = 360 Then Angle
Angle = Angle + 1
(۳۶۰ / (D3D8Main.CamLookAtX = Sin((Angle * 2 * PI
(۳۶۰ / (D3D8Main.CamLookAtZ = Cos((Angle * 2 * PI
D3D8Main.ApplyCameraChanges
Loop
```

نکته : برای دریافت این برنامه و نیز دریافت D3DEngine.dll و سورس آن ، پیغامی را بهمراه آدرس ایمیل خود در بخش نظرخواهی قراردهید

مباحث پیشرفته Direct3D – درس اول

موضوع : استفاده از object های D Studio Max۳ در Direct3D

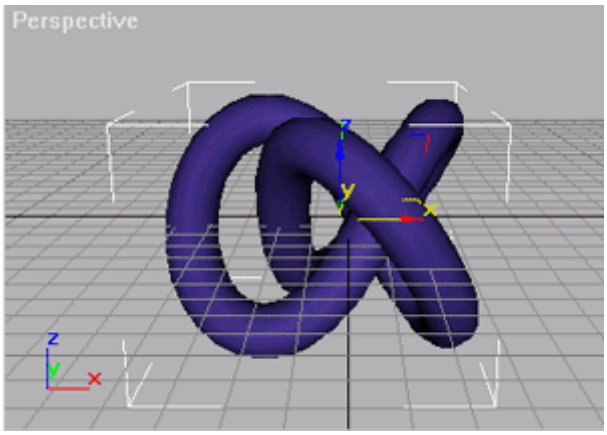
تا بحال ما هر شیی را که می خواستیم در Direct3D بسازیم خودمان بوسیله کد نویسی آنرا توصیف کرده ایم . ممکنست این سوال برایتان پیش آمده باشد که بازیهای تجاری برای تولید کاراکترهای و اشیا پیچیده سه بعدی چگونه عمل می کنند ؟

منطقی بنظر نمی رسد که اینگونه مدلهای پیچیده بصورت کد وارد برنامه شده اند زیرا نیاز به هزاران خط برنامه برای هر فریم خواهد بود . بجای اینکار ما object های خود را توسط برنامه های دیگری می سازیم و آنها را در برنامه خودمان load می کنیم سپس بافتها و material های مورد نظر را به آنها اختصاص داده و در پایان آنها را رندر می کنیم . مزیت دیگر اینکار اینست که شما می توانید براحتی فایل object خود را تغییر دهید و مدلهایی با جزئیات متفاوت برای برنامه خود قرار دهید .

مراحل ساخت چنین برنامه هایی بصورت زیر است :

۱ – ساخت object سه بعدی :

اولین چیزی که بایستی بدانید داشتن دانش پایه ای از چگونگی مدلسازی سه بعدی است . همچنین نیاز به یک نرم افزار مدلسازی مثل D Studio Max۳ دارید .



بعد از ساخت مدل خود در Max نیاز به یک Convertor دارید تا فایلهای Max را به فایلهای Direct3D که با فرمت ".X" هستند تبدیل کنید .

Convertor های زیادی برای تبدیل فایلهای نرم افزارهای مدلسازی به فایلهای ".X" وجود دارند که برخی از آنها عبارتند از :

- برنامه PolyTrans3D System Translation

- برنامه Deep Exploration 2.0

- برنامه Quick3D

- برنامه DWin۳

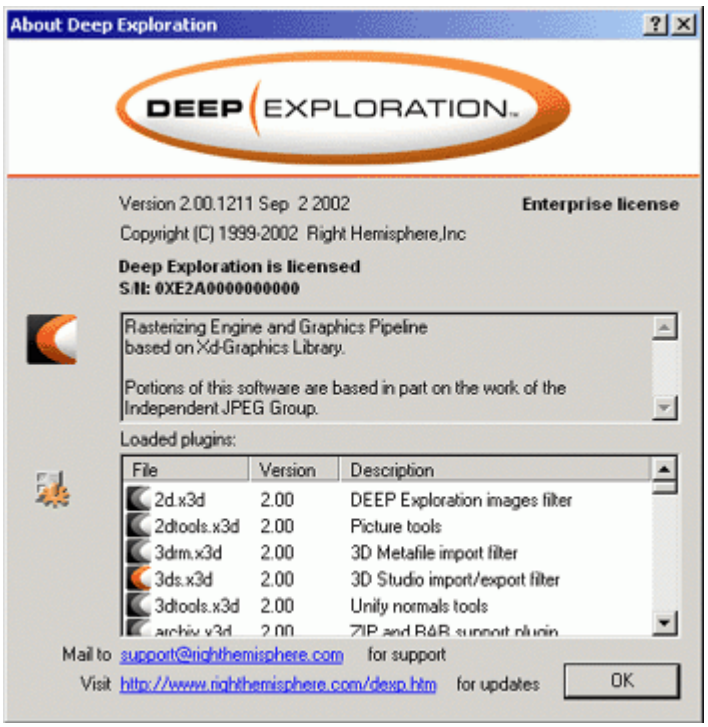
- DirectX Explorer Plugin

- ابزارهای موجود در DirectX 8.0 SDK که عبارتند از :

برنامه Conv3DS برای تبدیل فایلهای DS۳ به فایلهای X

DX SDK Exporter Plugin برای تبدیل فایلهای DS۳ و Max به فایلهای X

از بین این برنامه ها و plugin ها من برنامه Deep Exploration را به شما پیشنهاد می کنم .



در آدرس زیر می توانید اطلاعات بیشتری در مورد این برنامه بدست آورید و همچنین آنرا Download کنید :

Exploration 2.0 Deep
s/n: 0XE2A0000000000
REJ1HYXSR1A77Q10 :Authorization s/n

۲ – Load کردن یک Object ساخته شده :

زمانیکه فایل X شی مورد نظر را ساختید ، load کردن آن در direct3D ساده است . برای اینکار نیاز به یک مش داریم که اطلاعات شی ما را نگهداری کند :

```
Dim Mesh As D3DXMesh
```

همچنین برای اختصاص material و texture به شی ، نیاز به تعریف متغیرهای زیر داریم :

```
Dim MeshMaterial As D3DMATERIAL8
As Direct3DTexture8 Dim MeshTexture
```

حال به سراغ بازنویسی روتین InitGeometry می رویم :

- تعریف متغیرهای مورد نیاز :

```
Dim mtrlBuffer as D3DXBuffer
String Dim TextureFile as
Dim n as Long
```

- گرفتن داده های شی از فایل X :

```
Set Mesh=D3DX.LoadMeshFromX
"&"\"&app.path
yourfilename",D3DMESH_MANAGED,D3DDevice,Nothi
ng,mtrlBuffer,n
```

- استخراج اطلاعات materiasl شی و تنظیم پارامتر Ambient :

```
mtrlBuffer,0,MeshMaterial D3DX.BufferGetMaterial
MeshMaterial.Ambient=MeshMaterial.Diffuse
```

- استخراج نام بافت بکار رفته برای شی :

```
TextureFile=D3DX.BufferGetTextureName(mtrlBuffer,0)
x
```

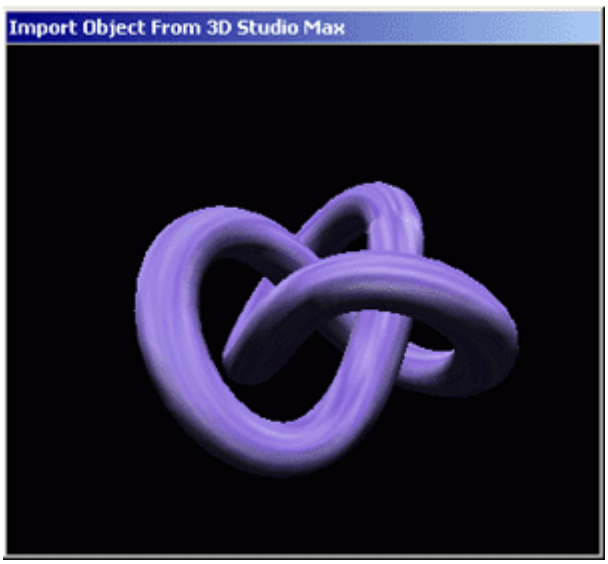
- ساخت بافت :

```
Then ""<>If TextureFile
MeshTexture=D3DX.CreateTextureFromFile Set
```

```
&"\"&D3DDevice,app.path
,TextureFile,128,128,D3DX_DEFAULT,0
D3DFMT_UNKNOWN,D3DPOOL_MANAGED,D3DX_F
ILTER_LINEAR,D3DX_FILTER_LINEAR,0,Byval
Byval 0,۰
End If
```

۳ – رندر نمودن شی : رندر نمودن شی چندان مشکل نیست اما همچنان باید ماتریسها و تبدیلاتی را که می خواهید ، خودتان مدیریت کنید .

```
MeshMaterial D3DDevice.SetMaterial
D3DDevice.SetTexture 0,MeshTexture
Mesh.DrawSubset 0
```



مباحث پیشرفته Direct3D – درس دوم

موضوع : مباحث تکمیلی نورپردازی در Direct3D

در بخش اول آموزش Direct3D با مبانی نورپردازی آشنا شدید . در این درس قصد دارم آن مباحث را کاملتر برایتان مطرح کنم .

نورپردازی یکی از بخشهای مهم طراحی یک بازی و یا یک انیمیشن سه بعدی است . بمنظور پیاده سازی نورپردازی یک صحنه ابتدا باید با تئوری آن آشنا شوید .

تئوری نورپردازی : نورپردازی در Direct3D تخمینی از چگونگی عملکرد نور در دنیای واقعی می باشد . چهار نوع اصلی نور در Direct3D قابل استفاده است (همچنین شما می توانید خودتان انواع جدیدی از نور ایجاد کنید که موضوع ما نیست) :

۱ – Point Light : توسط یک نقطه در فضای سه بعدی ایجاد می شود و دارای سه پارامتر رنگ ، دامنه و تضعیف می باشد . دامنه یک نور مسافتی است که نور می تواند طی کند . تضعیف ، مقدار کاهش نور در اثر افزایش مسافت می باشد . نور نقطه ای در تمام جهات تشعشع می کند – شبیه یک لامپ حبابی و یا یک شمع

۲ – Spot Light : دارای یک موقعیت و یک جهت است و تنها نور را در یک جهت خاص می تاباند – شبیه یک چراغ قوه . این نور دارای یک زاویه مخروطی و یک دامنه است .

۳ – Directional Light : دارای موقعیت نیست و برای پیاده سازی نورهایی که از فاصله بسیار دور می آیند – مثل خورشید – مناسب است .

۴ – Ambient Light : این نور تضمین می کند که تمام vertex های یک صحنه تاریکتر از یک رنگ خاص نباشند .

عملی کردن نورپردازی : ضمن اینکه اغلب کارت های گرافیک سه بعدی از نورپردازی پشتیبانی می کنند اما این نکته باید مورد توجه قرار گیرد که با افزایش تعداد نور در یک صحنه محاسبات Direct3D بیشتر می شود

و این باعث کند شدن رندر صحنه خواهد شد و بنابراین کارت های گرافیکی سه بعدی نیز دارای یک ماکزیمم تعداد نور هستند – مثلاً ۱۶ نور در GeForce 2 – همچنین توجه داشته باشید که نورهای مختلف دارای زمان پردازشی متفاوتی هستند . نور ambient سریعترین زمان پردازشی را دارد ، سپس نورdirectional ، سپس نور point و کندترین آنها Spot Light است .

همچنین نکته دیگری که باید توجه کنید دامنه نور است . اگر نور ، یک منطقه بزرگی را پوشش دهد بر تعداد زیادی از vertex ها تاثیر می گذارد و این باعث افزایش محاسبات می شود .

نورپردازی Specular – که در درسهای بعدی در مورد آن صحبت می کنم و برای ایجاد اشیا درخشان استفاده می شود – نیز زمان پردازشی زیادی دارد و بهتر است کمتر از آن استفاده شود .

پارامتر دیگری که باید در نظر بگیرید جزئیات هندسه شما می باشد . هر چه پیچیدگی صحنه بیشتر باشد ، نورپردازی نیز زمان بیشتری را مصرف می کند .

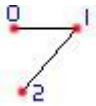
سایه زنی نیز یک بخش بسیار پیچیده در مدل سازی نور است و محاسبات آن بسیار زمان گیر خواهد بود بنابراین Direct3D مستقیماً محاسبات سایه زنی را انجام نمی دهد بلکه رنگ نور را بر مبنای جهت هر مثلث scale می کند بنابراین قسمت پشتی یک شی که رو به نور نیست ، هیچ نوری را دریافت نمی کند .

بردار نرمال : Direct3D هر vertex را بر مبنای بک بردار نرمال نورپردازی می کند و نوری که یک vertex دریافت می کند به زاویه بین نور و بردار نرمال آن vertex بستگی دارد . بردار نرمال توسط سه vertex یک face مثلثی ایجاد می شود و این بردار نرمال ساخته شده به

vertex ها اختصاص می یابد . بردار نرمال در واقع سمت یک مثلث را مشخص می کند بنابراین اگر نور پشت مثلث باشد ، مثلث هیچ نوری را دریافت نمیکند . بردار نرمال بایستی دارای طول ۱ باشد .

مراحل تولید بردار نرمال یک face مثلثی :

۱ – مطمئن شوید که face در جهت عقربه های ساعت ساخته شده است .



۲ – یک بردار از vertex شماره صفر به vertex شماره یک بسازید .

۳ – یک بردار از vertex شماره صفر به vertex شماره دو بسازید .

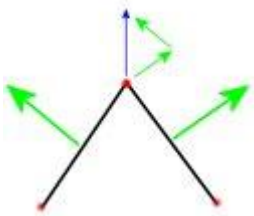
۴ – حاصلضرب برداری (cross droduct) این دو بردار را بدست آورید .

۵ – نتیجه حاصلضرب را نرمال کنید .

```
Private Function GenerateTriangleNormals(p0 As
UnlitVertex, p1 As UnlitVertex, p2 As UnlitVertex) As
D3DVECTOR
D3DVECTOR Dim v01 As
Dim v02 As D3DVECTOR
Dim vNorm As D3DVECTOR
v01, MakeVector(p1.X, p1.Y, p1.Z),  D3DXVec3Subtract
p0.Z)x ,MakeVector(p0.X, p0.Y
D3DXVec3Subtract v02, MakeVector(p2.X, p2.Y, p2.Z),
p0.Y, p0.Z)x ,MakeVector(p0.X
D3DXVec3Cross vNorm, v01, v02
vNorm ,D3DXVec3Normalize vNorm
GenerateTriangleNormals.X = vNorm.X
vNorm.Y = GenerateTriangleNormals.Y
```

```
GenerateTriangleNormals.Z = vNorm.Z
End Function
```

اگر دو face در یک vertex مشترک باشند (مثل گوشه دو دیوار) برای تولید نرمال این vertex ابتدا نرمال دو face را با روش فوق بدست آورید سپس دو بردار نرمال را با هم جمع کنید و در پایان بردار حاصلجمع را نرمال کنید .



برپاسازی نورپردازی : اولین چیزی که قبل از برپاسازی نورپردازی بایستی اعمال کنیم تغییر ساختار vertex است . برای اینکار باید پارامتر color را از ساختار vertex حذف و سه پارامتر را برای نگهداری نرمال اضافه کنیم :

```
Private Type UnlitVertex
X As Single
Single Y As
Z As Single
nx As Single
ny As Single
nz As Single
Single tu As
tv As Single
End Type
Const Unlit_FVF = (D3DFVF_XYZ Or
D3DFVF_NORMAL Or D3DFVF_TEX1)x
```

همچنین باید برای تمام vertex های شی خود بردار نرمال را محاسبه کنید برای مثال اگر شی شما یک مکعب است برای هر ۱۲ face آن بردار نرمال را بدست آورید . در زیر من کد لازم برای ساخت نرمال یکی از این face ها را نوشته ام :

```
x(۰ ,Cube2(0) = CreateVertex(-1, -1, 1, 0, 0, 0, 0
Cube2(1) = CreateVertex(1, 1, 1, 0, 0, 0, 1, 1)x
CreateVertex(-1, 1, 1, 0, 0, 0, 0, 1)x = (Cube2(2
Cube2(1),  ,(vN = GenerateTriangleNormals(Cube2(0
Cube2(2))x
= Cube2(0).nx = vN.X: Cube2(0).ny = vN.Y: Cube2(0).nz
vN.Z
= Cube2(1).nx = vN.X: Cube2(1).ny = vN.Y: Cube2(1).nz
vN.Z
Cube2(2).nx = vN.X: Cube2(2).ny = vN.Y: Cube2(2).nz =
vN.Z
```

برای برپا سازی نور ابتدا بایستی یک material به device خود اضافه کنید :

```
Dim Mtrl As D3DMATERIAL8, Col As
D3DCOLORVALUE
Col.a = 1: Col.r = 1: Col.g = 1: Col.b = 1
Col = Mtrl.Ambient
Mtrl.diffuse = Col
D3DDevice.SetMaterial Mtrl
```

سپس بایستی طوری device خود را تنظیم کنید که نور شما را بشناسد

- lights یک شی از نوع D3DLight8 است - یکبار که این خط را بنویسید می توانید از نور استفاده کنید اما اگر خصوصیات نور را تغییر دهید بایستی دوباره این دستور را فراخوانی کنید :

```
D3DDevice.SetLight 0, Lights
```

حال باید نور را روشن کنید :

```
D3DDevice.LightEnable 0, 1
```

و در پایان باید به Direct3D بگوئید که نورپردازی را برای شما انجام دهد :

```
D3DDevice.SetRenderState D3DRS_LIGHTING, ۱
```

چگونگی ایجاد یک نور : برای ایجاد هر یک از ۴ نوع اصلی نور باید به روشی خاص عمل کنید :

۱ - نورپردازی Ambient : این نوع نورپردازی بسیار ساده است و تنها با فراخوانی تابع SetRenderState ایجاد می شود . رنگ ambient یک عدد هگزادسیمال بصورت RRGGBB است :

```
D3DDevice.SetRenderState D3DRS_AMBIENT,
H202020&
```

۲ - نورپردازی Directional : دارای دو پارامتر رنگ و جهت می باشد :

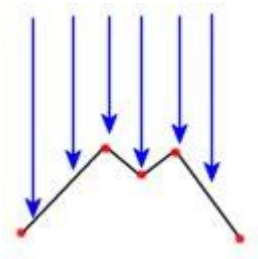
D3DLIGHT_DIRECTIONAL = Lights.Type

Lights.diffuse.r = 1

١ = Lights.diffuse.g

Lights.diffuse.b = 1

Lights.Direction = MakeVector(0, -1, 0)x



٣ – نورپردازی Point : دارای سه پارامتر موقعیت ، رنگ و تضعیف می باشد :

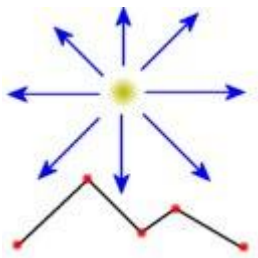
Lights.Type = D3DLIGHT_POINT

MakeVector(5, 0, 2)x = Lights.position

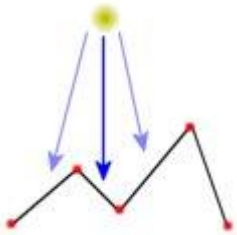
Lights.diffuse.b = 1

١٠٠ = Lights.Range

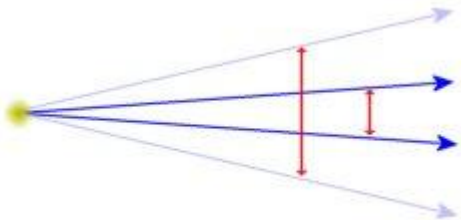
Lights.Attenuation1 = 0.05



۴ – نورپردازی Spot : این نور دارای دو مخروط است که نقاط خارج مخروط اول روشنتر از نقاط داخل آن هستند . دو زاویه برای مخروط وجود دارد – زاویه داخلی theta و زاویه خارجی phi – که برحسب رادیان هستند :



```
D3DLIGHT_SPOT = Lights.Type
Lights.position = MakeVector(-4, 0, 0)x
۱۰۰ = Lights.Range
Lights.Direction = MakeVector(1, 0, 0)x
x(۱۸۰ / Lights.Theta = 30 * (Pi
Lights.Phi = 50 * (Pi / 180)x
۱ = Lights.diffuse.g
Lights.Attenuation1 = 0.05
```



مباحث پیشرفته Direct3D - درس سوم

موضوع : استفاده از Index Buffer برای ذخیره سازی اشکال سه بعدی

مقدمه : مکعبی که در درسهای قبلی ساختیم را درنظر بگیرید . با دانشی که اکنون دارید ، دو راه برای ساخت یک مکعب داریم : ۱ - استفاده از ۳۶ عدد vertex برای تعریف face های مکعب ۲ - ساخت مکعب با استفاده از یک مدلساز و ذخیره آن با فرمت X

روش اول غیرکارامد است زیرا شما بایستی از تعداد زیادی vertex برای یک شکل بسیار ساده استفاده کنید . روش دوم مناسب است اما زمانیکه بخواهیم رنگها و بافتها را تغییر دهیم دچار مشکل خواهیم شد . روش جدیدی که امروز در مورد آن صحبت می کنم استفاده از Index Buffer است .

Index Buffer شامل یکسری عدد integer است که این اعداد مرجعی برای vertex های ذخیره شده در یک Vertex Buffer هستند . برای مثال فرض کنید یک Vertex Buffer شامل ۸ عدد vertex داریم که یک مکعب را برای ما توصیف می کند . ما می توانیم یک Index Buffer با ۳۶ عضو بسازیم بطوریکه ترتیب اتصال vertex ها را برای ما مشخص کنند . مثلاً Index های ۰ و ۱و ۳ برای مشخص کردن face شماره ۱ مکعب بکار می روند . بنابراین بجای استفاده از ۳۶ عدد vertex می توانیم مکعب را با ۸ عدد vertex و یک Index Buffer بسازیم .

گرچه استفاده از Index Buffer بسیار کارامد است اما چندین محدودیت در استفاده از آن وجود دارد . مهمترین آنها اینست که تمام اندیسهایی که یک vertex مشابه را share می کنند بایستی خصوصیات مشابهی داشته باشند - موقعیت ، رنگ ، بافت و نرمال یکسان - برای مثال نمی

توانید مکعبی بسازید که هر face آن یک رنگ داشته باشد .

ساخت Index Buffer : ابتدا به متغیرهای زیر نیاز داریم :

```
Dim VBuffer as Direct3DVertexBuffer8
as Direct3DIndexBuffer8 Dim IBuffer
Dim Vlist(0 to 7) as LITVERTEX
as Integer (Dim Ilist(0 to 35
```

تابع InitGeometry بصورت زیر بازنویسی می شود:

۱- تولید هشت vertex برای مکعب :

```
x(۰ ,۰ ,۰ ,HFF0000& ,Vlist(0) = CreateLitVertex(-1, -1, -1
x(۰ ,۰ ,۰ ,&HFF00& ,Vlist(1) = CreateLitVertex(-1, 1, -1
x(۰ ,۰ ,۰ ,&HFF& ,Vlist(2) = CreateLitVertex(1, -1, -1
HFF00FF, 0, 0, 0)x& ,Vlist(3) = CreateLitVertex(1, 1, -1
HFFFF00, 0, 0, 0)x& ,CreateLitVertex(-1, -1, 1 = (Vlist(4
HFFFF, 0, 0, 0)x& ,CreateLitVertex(-1, 1, 1 = (Vlist(5
HFFCC00, 0, 0, 0)x& ,۱ ,۱- ,Vlist(6) = CreateLitVertex(1
HFFFFFF, 0, 0, 0)x& ,Vlist(7) = CreateLitVertex(1, 1, 1
```

۲ - ایجاد Vertex Buffer توسط تابع CreateVertexBuffer :

```
= Set VBuffer
D3DDevice.CreateVertexBuffer(Len(Vlist(0)) * 8, 0,
D3DPOOL_DEFAULT)x ,Lit_FVF
```

,D3DVertexBuffer8SetData VBuffer, 0, Len(Vlist(0)) * 8, 0
Vlist(0)x

۳ - تولید index ها :

' front
۲ = (Ilist(0) = 0: Ilist(1) = 1: Ilist(2
Ilist(3) = 1: Ilist(4) = 3: Ilist(5) = 2
' Right
Ilist(7) = 3: Ilist(8) = 6 :Ilist(6) = 2
Ilist(9) = 3: Ilist(10) = 7: Ilist(11) = 6
' Back
Ilist(12) = 6: Ilist(13) = 7: Ilist(14) = 4
Ilist(17) = 4 :۵ = (Ilist(15) = 7: Ilist(16
' Left
۰ = (Ilist(18) = 4: Ilist(19) = 5: Ilist(20
Ilist(21) = 5: Ilist(22) = 1: Ilist(23) = 0
' Top
Ilist(25) = 5: Ilist(26) = 3 :Ilist(24) = 1
۳ = (Ilist(27) = 5: Ilist(28) = 7: Ilist(29
' Bottom
Ilist(30) = 2: Ilist(31) = 6: Ilist(32) = 0
Ilist(34) = 4: Ilist(35) = 0 :Ilist(33) = 6

۴ - ایجاد Index Buffer توسط تابع CreateIndexBuffer :

D3DDevice.CreateIndexBuffer(Len(Ilist(0)) * = Set IBuffer
D3DPOOL_DEFAULT)x ,36, 0, D3DFMT_INDEX16

```
,D3DIndexBuffer8SetData IBuffer, 0, Len(Ilist(0)) * 36, 0
Ilist(0)x
```

تابع Render : برای رندر کردن این مکعب دو روش وجود دارد :

۱ – استفاده از تابع DrawIndexedPrimitive : در این روش از VBuffer و IBUffer و آرایه vertex ها استفاده می شود :

```
Public Sub Render()x
D3DCLEAR_TARGET Or  ,D3DDevice.Clear 0, ByVal 0
۰ ,D3DCLEAR_ZBUFFER, 0, 1#
D3DDevice.BeginScene
Len(Vlist(0))x ,D3DDevice.SetStreamSource 0, VBuffer
۰ ,D3DDevice.SetIndices IBuffer
D3DDevice.DrawIndexedPrimitive
۱۲ ,D3DPT_TRIANGLELIST, 0, 36, 0
D3DDevice.EndScene
۰ D3DDevice.Present ByVal 0, ByVal 0, 0, ByVal
End Sub
```

۲ – استفاده از تابع DrawIndexedPrimitiveUP : در این روش از آرایه های vertex و index استفاده می شود :

```
Public Sub Render()x
D3DCLEAR_TARGET Or  ,D3DDevice.Clear 0, ByVal 0
۰ ,D3DCLEAR_ZBUFFER, 0, 1#
D3DDevice.BeginScene
D3DDevice.DrawIndexedPrimitiveUP
D3DPT_TRIANGLELIST, 0, 8, 12, Ilist(0),
```

```
Len(Vlist(0))x ,(D3DFMT_INDEX16, Vlist(0
D3DDevice.EndScene
ByVal 0 ,۰ ,D3DDevice.Present ByVal 0, ByVal 0
End Sub
```

نکته : برای دریافت برنامه نمونه و یا اطلاعات بیشتر در مورد توابعی که در این درس استفاده شد با من تماس بگیرید .

مباحث پیشرفته Direct3D - درس چهارم

موضوع : Vertex/Mesh Animation

در این درس در مورد روشهای ساخت انیمیشن در Direct3D صحبت خواهیم کرد . انیمیشن در فضای سه بعدی در حالتهای مختلفی می تواند ایجاد شود که بسته به engine گرافیکی شما و ابزارهایی که ایجاد کرده اید ، دارد . سه روش اصلی ساخت انیمیشن وجود دارد که عبارتند از :

- **Tween سازی دستی / درون یابی خطی (manual tweening/linear interpolation)**
- **درون بابی برداری (vector interpolation)**

- درون یابی بر اساس فریم کلیدی (interpolation keyframe)

۱ – روش اول یکی از ساده ترین راههای ساخت انیمیشن است . این روش در زمانیکه با مدلهای پیچیده سر و کار دارید مناسب نیست – و یا مدلهایی با تعداد زیادی vertex – این روش نوعی tween کردن است که از مزیت index buffer ها استفاده می کند .

درون یابی ، چگونگی تغییرات شیی در طول یک زمان مشخص می باشد . در درسهای قبلی شما درون یابی رنگ را روی یک شی دیدید که در آن یک رنگ بطور ملایم به رنگ دیگری تبدیل می شد (fadeشدن (. درون یابی خطی نیز مشابه آن است . برای درون یابی خطی از موقعیت A به موقعیت B از فرمول زیر استفاده می شود :

(B*V)+A*(1-V)

که A و B مختصاتهای مبدا و مقصد هستند و V ضریب درون یابی است که عددی بین صفر و یک می باشد . این فرمول مختصات نقطه tween را در هر لحظه مشخص می کند .

همانطور که می بینید بکار بردن این فرمول برای یک شی با تعداد زیادی vertex بسیار وقت گیر بوده و fram rate را پایین می آورد .

تابع زیر دو vertex و یک مقدار ضریب درون یابی را می گیرد تا نقطه tween را محاسبه کند :

```
,Private Function TweenVertices(Source As LITVERTEX
Dest As LITVERTEX, TweenAmount As Single) As
LITVERTEX
Dest.X * TweenAmount) + Source.X * ) = TweenVertices.X
(1# - TweenAmount)x
Dest.Y * TweenAmount) + Source.Y * ) = TweenVertices.Y
(1# - TweenAmount)x
```

```
Dest.Z * TweenAmount) + Source.Z * ) = TweenVertices.Z
(1# - TweenAmount)x
Source.color = TweenVertices.color
End Function
```

اگر شما از vertex های UNLIT استفاده کنید – vertex هایی با بردار نرمال – در اینصورت باید کد فوق را تغییر دهید و باید tween را از نرمال مبدا به نرمال مقصد نیز انجام دهید .

همانطور که می بینید رنگ tween vertex نیز تنظیم شده است . در یک تابع tweening مناسبتر می توانید رنگها ، مختصات بافت و مقادیر specular را نیز tween کنید .

محدودیتی که این روش دارد اینست که خطی است و برای مدل کردن حرکتهای غیر خطی درست کار نمی کند .

حال می خواهیم از تابع tween استفاده کنیم تا یک مکعب را در یک انیمیشن به یک هرم تبدیل کنیم . ابتدا سه شی را بصورت زیر تعریف می کنیم :

در ابتدای انیمیشن ، شی current cube همان source cube است'

```
HFF0000, & ,CubeVertices(0) = CreateLitVertex(-1, -1, -1
x(۰ ,0, 0
,۰ ,&HFF00& ,CubeVertices(1) = CreateLitVertex(-1, 1, -1
x(۰ ,۰
,۰ ,۰ ,&HFF& ,CubeVertices(2) = CreateLitVertex(1, -1, -1
x(۰
HFF00FF, 0, & ,CubeVertices(3) = CreateLitVertex(1, 1, -1
x(۰ ,0
```

```
HFFFF00, & ,CubeVertices(4) = CreateLitVertex(-1, -1, 1
x(۰ ,0, 0
,HFFFF, 0, 0& ,CubeVertices(5) = CreateLitVertex(-1, 1, 1
x(۰
HFFCC00, 0, & ,CubeVertices(6) = CreateLitVertex(1, -1, 1
x(۰ ,0
HFFFFFF, 0, & ,CubeVertices(7) = CreateLitVertex(1, 1, 1
0, 0)x
'مکعب اولیه
,CubeVerticesSource(0) = CreateLitVertex(-1, -1, -1
HFF0000, 0, 0, 0)x&
,CubeVerticesSource(1) = CreateLitVertex(-1, 1, -1
x(۰ ,۰ ,۰ ,&HFF00&
,CubeVerticesSource(2) = CreateLitVertex(1, -1, -1
x(۰ ,۰ ,۰ ,&HFF&
,CubeVerticesSource(3) = CreateLitVertex(1, 1, -1
HFF00FF, 0, 0, 0)x&
,CubeVerticesSource(4) = CreateLitVertex(-1, -1, 1
HFFFF00, 0, 0, 0)x&
,CubeVerticesSource(5) = CreateLitVertex(-1, 1, 1
HFFFF, 0, 0, 0)x&
,CubeVerticesSource(6) = CreateLitVertex(1, -1, 1
HFFCC00, 0, 0, 0)x&
,CubeVerticesSource(7) = CreateLitVertex(1, 1, 1
HFFFFFF, 0, 0, 0)x&
'هرم مقصد
,CreateLitVertex(-1, -1, -1 = (CubeVerticesDest(0
HFF0000, 0, 0, 0)x&
```

```
,CreateLitVertex(-0.1, 1, -0.1 = (CubeVerticesDest(1
x(۰ ,۰ ,۰ ,&HFF00&
,&HFF& ,CreateLitVertex(1, -1, -1 = (CubeVerticesDest(2
x(۰ ,۰ ,۰
,CreateLitVertex(0.1, 1, -0.1 = (CubeVerticesDest(3
HFF00FF, 0, 0, 0)x&
,CreateLitVertex(-1, -1, 1 = (CubeVerticesDest(4
HFFFF00, 0, 0, 0)x&
,CreateLitVertex(-0.1, 1, 0.1 = (CubeVerticesDest(5
HFFFF, 0, 0, 0)x&
,CreateLitVertex(1, -1, 1 = (CubeVerticesDest(6
HFFCC00, 0, 0, 0)x&
,CreateLitVertex(0.1, 1, 0.1 = (CubeVerticesDest(7
HFFFFFF, 0, 0, 0)x&
```

حال باید در یک حلقه با استفاده از تابع twen پیکسلهای CubeVertices را update کنیم :

```
Private Sub UpdateAnimation()x
Integer Dim I As
'به روز کردن پارامترهای زمان و جهت
Then If AnimTweenDir = True
AnimTweenFactor = AnimTweenFactor +
(#۱*(۱۰۰۰ / ((((GetTickCount() - LastTimeTweened
LastTimeTweened = GetTickCount
Then #۱ =< If AnimTweenFactor
AnimTweenFactor = 1#
AnimTweenDir = False
If End
Else
```

```
AnimTweenFactor = AnimTweenFactor -
(LastTimeTweened) / 1000)*1# – ()(((GetTickCount
LastTimeTweened = GetTickCount
Then #۰ => AnimTweenFactor If
AnimTweenFactor = 0#
True = AnimTweenDir
End If
End If
' ها vertex به روز کردن اطلاعات
۷ For I = 0 To
CubeVertices(I) = TweenVertices(CubeVerticesSource(I),
AnimTweenFactor)x ,(CubeVerticesDest(I
Next I
’vertex به روز کردن بافر
D3DVertexBuffer8SetData(VBuffer, 0,  If
CubeVertices(0)) =  ,Len(CubeVertices(0)) * 8, 0
:D3DERR_INVALIDCALL Then GoTo Error
Sub Exit
:Error
Debug.Print “Error occured whilst updating the
animation…”x
End Sub
```

زمان پایه انیمیشن توسط عبارت زیر تنظیم می شود :

(GetTickCount() - LastTimeTweened) / 1000) * 1#)))

همانطور که می دانید دو نوع انیمیشن وجود دارد : انیمیشن بر مبنای frame و انیمیشن بر مبنای زمان . در انیمیشن بر مبنای frame شماره فریم با یک مقدار ثابت در زمان افزایش می یابد اما اگر اینکار باعث می شود کیفیت انیمیشن در کامپیوترهای با سرعت متفاوت تغییر کند .

بنابراین انیمیشن را بر مبنای زمان تولید کرده ایم . انیمیشن های بر مبنای زمان بجای " ۱ فریم در هر سیکل " ، " ۳۰ فریم در هر ثانیه " هستند .

۲ – روش دوم از توابع کتابخانه D3DX برای انجام عمل tweening استفاده می کند و بنابراین بهبودی در سرعت انیمیشن نسبت به روش بالا حاصل می شود . با استفاده از کتابخانه D3DX می توانیم عمل درون یابی خطی را برای تمام اجزا اصلی یک vertex انجام دهیم . لیست زیر توابعی را برای اینکار نشان می دهد :

– تابع D3DXVec3Lerp : انجام درون یابی برای موقعیت و نرمال :

D3DXVec3Lerp(VOut as D3DVECTOR, V1 as as D3DVECTOR, S as Single)x D3DVECTOR, V2
VOut = The result of the interpolation –

The source coordinates = V1 –

V2 = The destination coordinates –

interpolation amount - between, but not limited S = The –

the source and 1 is the to, 0.0 - 1.0 scale; where 0 is destination

– تابع D3DXColorLerp : انجام درون یابی برای رنگهای vertex :

D3DXColorLerp(COut as D3DCOLORVALUE, C1 as D3DCOLORVALUE, C2 as D3DCOLORVALUE, S as Single)x
colour COut = The resulting –

C1 = The source colour –

C2 = The destination colour –

,interpolant S = The –

on a 0.0 to 1.0 scale

– تابع D3DXVec2Lerp : انجام درون یابی برای مختصاتهای دوبعدی :

VOut = The result of this interpolation –

coordinates V1 = The source –

V2 = The destination coordinates –

to 1.0 scale ۰,۰ S = The interpolant on a –

– تابع D3DXVec3Hermite : تولید یک مسیر منحنی که از دو نقطه کنترل عبور می کند :

D3DXVec3Hermite(VOut as D3DVECTOR, V1 as
T1 as D3DVECTOR, V2 as D3DVECTOR, ,D3DVECTOR
T2 as D3DVECTOR, S as Single)x
Result VOut = The –

V1 = The Source Coordinate –

coordinate, this is the T1 = The Tangent at the Source –

.point direction and speed the line will leave the source

V2 = The Destination Coordinate –

Destination coordinate, this is T2 = The Tangent at the –

destination the direction and speed the line will enter the .point

S = The Interpolant Value –

برای اینکه بتوانیم از کتابخانه D3DX استفاده کنیم باید توصیف vertex هایمان را تغییر دهیم و بایستی یکسری مقادیر ARGB اضافی را به ساختار vertex اضافه کنیم :

```
Private Type LITVERTEX
X As Single
Single Y As
Z As Single
color As Long
specular As Long
tu As Single
As Single tv
ColorEx As D3DCOLORVALUE
End Type
```

حال تابع tween را بصورت زیر می نویسیم :

```
,Private Function TweenVertices(Source As LITVERTEX
Dest As LITVERTEX, TweenAmount As Single) As
LITVERTEX
D3DVECTOR Dim vResult As
Dim vResult2 As D3DVECTOR2
' Tween کردن موقعیت vertex ها
D3DXVec3Lerp vResult, MakeVector(Source.X, Source.Y,
MakeVector(Dest.X, Dest.Y, Dest.Z),  ,(Source.Z
TweenAmount
vResult.X = TweenVertices.X
TweenVertices.Y = vResult.Y
TweenVertices.Z = vResult.Z
' Tween کردن اطلاعات texture
,D3DXVec2Lerp vResult2, MakeVector2D(Source.tu
Source.tv), MakeVector2D(Dest.tu, Dest.tv),
```

```
TweenAmount
vResult2.X = TweenVertices.tu
TweenVertices.tv = vResult2.Y
' Tween کردن اطلاعات رنگ
D3DXColorLerp TweenVertices.ColorEx,
TweenAmount ,Source.ColorEx, Dest.ColorEx
With TweenVertices.ColorEx
G * 255, .R * 255)x. ,TweenVertices.color = RGB(.B * 255
End With
End Function
```

نکته ای که باید به آن توجه کنید اینست که در تابع فوق برای اشاره به vertex ، یک بردار ساخته شده است (توسط تابع MakeVector) .

۳ – روش سوم پر استفاده ترین روش انیمیشن سازی است . اگر شما انیمیشن های پیچیده با تعداد زیادی شی در آن داشته باشید و اگر بخواهید تغییرات اشیا را در هر فریم ذخیره کنید ، به حجم بالایی از منابع ذخیره سازی نیاز است . بجای آن ما با استفاده از یکسری فریم کلیدی ، فریمهای میانی را پیش بینی می کنیم .

برای انجام درون یابی فریم کلیدی ، بایستی مقدار vertex را در هر فریم کلیدی بدانیم و نیز بدانیم هر فریم کلیدی در چه زمانی ظاهر می شود . بنابراین باید برای هر انیمیشن چند فایل را بعنوان فریم کلیدی ذخیره کنیم .

در این درس ما داده های کلیدی انیمیشن را از یکسری فایل load می کنیم بنابراین تمام ثابتهای زمان keyframe درون برنامه قرار داده می شود (شما می توانید خودتان یک ماژول بنویسید که انیمیشن های عمومی تر را نیز مدیریت کند . این ماژول باید قادر باشد که یک فرمت استاندارد فایل را import کند ، اشیا و texture های مربوطه را load

نماید و سپس خودش ساخت انیمیشن را بطور اتوماتیک انجام دهد و برنامه اصلی فقط روتین render و یا update را فراخوانی کند) . پس از جمع آوری اطلاعات فریم های کلیدی ، باید در هر زمان محاسبه کنیم که چه مدتی از شروع انیمیشن گذشته است و بنابراین انیمیشن در چه موقعیتی قرار دارد . سپس محاسبه می کنیم که فریم کلیدی قبلی و فریم کلیدی بعدی چیست همچنین حساب می کنیم در چه فاصله زمانی از ایندو قرار داریم . سرانجام یک درون یابی نرمال را انجام می دهیم تا اطلاعات فریم جاری بدست آید و این اطلاعات را درون یک شی Mesh می گذاریم و آنرا رندر می کنیم .

در درسهای قبلی در مورد load کردن اشیا از یک فایل X صحبت کردم اما در مورد چگونگی گرفتن اطلاعات vertex از یک شی Mesh صحبت نشد . کتابخانه D3DX برای اینکار دو تابع دارد :

- تابع D3DXMeshVertexBuffer8GetData : اطلاعات یک شی D3DXMesh را گرفته و در یک آرایه از D3DVERTEX ذخیره می کند :

```
D3DXMeshVertexBuffer8GetData( D3DXMeshobj As
Unknown, Offset As Long, Size As Long, Flags As Long,
Data As Any) As Long
D3DXMeshobj As Unknown = A D3DXMESH object -
.from that you want to extract the data
Offset As Long = How far into the vertex buffer we want -
reading, 0 is the beginning to start
will be  Size As Long = Size of the vertex buffer, this -
Len(D3DVERTEX) * Mesh.GetNumVertices
of the  Flags As Long = A combination -
.CONST_D3DLOCKFLAGS, leave as 0
```

the array that you Data As Any = The first element in –
want the data to be read into, should be an array of D3DVERTEX vertices
Return Code As Long = Returns D3D_OK for success, or –
either of D3DERR_INVALIDCALL or E_INVALIDARG for an error

– تابع D3DXMeshVertexBuffer8SetData : اطلاعات یک بافر vertex را در یک شی D3DXMesh قرار می دهد :

D3DXMeshVertexBuffer8SetData(D3DXMeshobj As Unknown, Offset As Long, Size As Long, Flags As Long, Data As Any) As Long
D3DXMeshobj As Unknown = The D3DXMESH object –
data to be placed that defines where you want the
Offset As Long = How far into the Destination vertex –
buffer you want to place the data
bytes, this will be Size As Long = The Size of the buffer in –
Len(D3DVERTEX) * Mesh.GetNumVertices
Combination of the Flags As Long = A –
CONST_D3DLOCKFLAGS, leave as 0
element in the array of data you Data As Any = The first –
want placed in the mesh's vertex buffer
Return Code As Long = D3D_OK for success or –
for D3DERR_INVALIDCALL or E_INVALIDARG failure

عملیات انجام انیمیشن فریم کلیدی بصورت زیر است :

– load کردن اشیا از فایلهای X به درون شی D3DXMesh

- استخراج اطلاعات vertex از این شی
- انجام درون یابی بین فریمهای کلیدی
- قرار دادن اطلاعات vertex های درون یابی در یک شی D3DXMesh

فرض می کنیم که انیمیشن ما همیشه از زمان صفر تا زمان n باشد – برحیب میلی ثانیه – بنابراین می توانیم از GetTickCount برای توابع زمانی خود استفاده کنیم . همچنین یک ساختار را برای هر فریم کلیدی بصورت زیر تعریف می کنیم :

```
Private Type KeyFrame
Mesh As D3DXMesh ' شی load شده از یک فایل
MatList() As D3DMATERIAL8 ' آرایه material برای هر شی
Texture' TexList() As Direct3DTexture8 آرایه
nMaterials ' تعداد material ها و texture هایی که استفاده می کنیم
As Long
VertexList() As ' داده های vertex برای این فریم کلیدی
D3DVERTEX
As Long TimeIndex ' موقعیت این فریم کلیدی در انیمیشن
End Type
```

حال باید تابعی بنویسیم که اطلاعات را از یک فایل X استخراج کرده و درون فریم کلیدی قرار دهد :

```
Private Function CreateKeyFrameFromFile(Filename As
String, TexturePrefix As String, Time As Long) As
KeyFrame
' Filename :نام فایل X برای شی سه بعدی
پوشه ای که اطلاعات texture این شی در آن قرار دارد :
' TexturePrefix
```

```
' Time : اندیس زمان برای این فریم کلیدی
Long Dim I As
Dim XBuffer As D3DXBuffer
Dim TextureFile As String
Long Dim hResult As
'خواندن اطلاعات از فایل ورودی به حافظه
CreateKeyFrameFromFile.Mesh = Set
D3DX.LoadMeshFromX(Filename,
D3DDevice, Nothing, XBuffer, ,D3DXMESH_MANAGED
CreateKeyFrameFromFile.nMaterials)x
' تولید material ها و texture ها
ReDim
CreateKeyFrameFromFile.MatList(CreateKeyFrameFro
D3DMATERIAL8 mFile.nMaterials) As
ReDim
CreateKeyFrameFromFile.TexList(CreateKeyFrameFrom
Direct3DTexture8 File.nMaterials) As
For I = 0 To CreateKeyFrameFromFile.nMaterials - 1
,D3DX.BufferGetMaterial XBuffer, I
CreateKeyFrameFromFile.MatList(I)x
= CreateKeyFrameFromFile.MatList(I).Ambient
CreateKeyFrameFromFile.MatList (I).diffuse
D3DX.BufferGetTextureName(XBuffer, I)x = TextureFile
Then "" <> If TextureFile
= (Set CreateKeyFrameFromFile.TexList(I
D3DX.CreateTextureFromFileEx(D3DDevice,
D3DX_DEFAULT, ,TextureFile & TexturePrefix
D3DX_DEFAULT, D3DX_DEFAULT, 0,
,D3DFMT_UNKNOWN, D3DPOOL_MANAGED
,۰ ,D3DX_FILTER_LINEAR, D3DX_FILTER_LINEAR
ByVal 0, ByVal 0)x
If End
```

```
Next I
'استخراج داده های vertex
ReDim
CreateKeyFrameFromFile.VertexList(CreateKeyFrameFr
D3DVERTEX omFile.Mesh.GetNumVertices) As
hResult =
D3DXMeshVertexBuffer8GetData(CreateKeyFrameFrom
((Len(CreateKeyFrameFromFile.VertexList(0 ,۰ ,File.Mesh
,reateKeyFrameFromFile.Mesh.GetNumVertices, 0 *
((CreateKeyFrameFromFile.VertexList(0
CreateKeyFrameFromFile.TimeIndex = Time
End Function
```

در تابع Initialize خطوط زیر را برای ساخت فریم های کلیدی اضافه می کنیم :

```
nKeyFrames = 4
kfAnimLength = ۲۵۰۰
AnimLastStartAt = GetTickCount()x
KeyFrame ReDim kfAnim(nKeyFrames - 1) As
& kfAnim(0) = CreateKeyFrameFromFile(App.Path
x(۰ ,"\" & App.Path ,"frame0.x\"
& kfAnim(1) = CreateKeyFrameFromFile(App.Path
kfAnimLength * (1 / 3))x ,"\" & frame1.x", App.Path\"
& CreateKeyFrameFromFile(App.Path = (kfAnim(2
kfAnimLength * (2 / 3))x ,"\" & frame2.x", App.Path\"
& kfAnim(3) = CreateKeyFrameFromFile(App.Path
kfAnimLength)x ,"\" & frame3.x", App.Path\"
& CreateKeyFrameFromFile(App.Path = kfCurrent
(۰ ,"\" & frame0.x", App.Path\"
```

دقت کنید که از یک اندیس زمان برای ساخت فریم های کلیدی استفاده شده است .

حال باید کدی برای نمایش دادن انیمیشن بنویسیم . ابتدا باید به روشی تغییرات فریمها را کنترل کنیم :

```
For I = 0 To nKeyFrames - 2
kfAnim(I).TimeIndex Then =< If CurrentTimeIndex
PrevFrame = I
NextFrame = I + 1
If End
Next I
```

سپس باید با توجه به زمان index دو فریم کلیدی و زمان جاری ، پارامتر درون یابی را محاسبه کنیم :

```
sTime = kfAnim(PrevFrame).TimeIndex
kfAnim(NextFrame).TimeIndex = eTime
cTime = CurrentTimeIndex
sTime – eTime = eTime
cTime = cTime - sTime
sTime = sTime - sTime
cTime / eTime = InterpolateAmount
```

سپس باید بر اساس این پارامتر عمل درون یابی را روی داده های vertex انجام دهیم :

```
For I = 0 To kfCurrent.Mesh.GetNumVertices
'درون یابی مختصاتها
```

```
,D3DXVec3Lerp vTemp3D
MakeVector(kfAnim(PrevFrame).VertexList(I).X,
_ ,kfAnim(PrevFrame).VertexList(I).Y
,(kfAnim(PrevFrame).VertexList(I).Z
MakeVector(kfAnim(NextFrame).VertexList(I).X,
_ ,kfAnim(NextFrame).VertexList(I).Y
InterpolateAmount ,(kfAnim(NextFrame).VertexList(I).Z
vTemp3D.X = kfCurrent.VertexList(I).X
kfCurrent.VertexList(I).Y = vTemp3D.Y
vTemp3D.Z = kfCurrent.VertexList(I).Z

'درون یابی نرمالها
,D3DXVec3Lerp vTemp3D
,MakeVector(kfAnim(PrevFrame).VertexList(I).nx
_ ,kfAnim(PrevFrame).VertexList(I).ny
,(kfAnim(PrevFrame).VertexList(I).nz
,MakeVector(kfAnim(NextFrame).VertexList(I).nx
_ ,kfAnim(NextFrame).VertexList(I).ny
InterpolateAmount ,(kfAnim(NextFrame).VertexList(I).nz
vTemp3D.X = kfCurrent.VertexList(I).nx
vTemp3D.Y = kfCurrent.VertexList(I).ny
kfCurrent.VertexList(I).nz = vTemp3D.Z

'درون یابی اطلاعات بافت
D3DXVec2Lerp vTemp2D,
,MakeVector2D(kfAnim(PrevFrame).VertexList(I).tu
_ ,(kfAnim(PrevFrame).VertexList(I).tv
```

```
,MakeVector2D(kfAnim(NextFrame).VertexList(I).tu
InterpolateAmount ,(kfAnim(NextFrame).VertexList(I).tv
vTemp2D.X = kfCurrent.VertexList(I).tu
kfCurrent.VertexList(I).tv = vTemp2D.Y
Next I
```

حال باید داده تولید شده را به فرمت Mesh برگردانیم :

```
= hResult
D3DXMeshVertexBuffer8SetData(kfCurrent.Mesh, 0,
* ((Len(kfCurrent.VertexList(0
kfCurrent.Mesh.GetNumVertices, 0,
kfCurrent.VertexList(0))x
```

با استفاده از روش فوق می توانید هر تعداد فریم کلیدی را به انیمیشنتان اضافه کنید . اشکالی که روش فوق دارد اینست که اطلاعات texture برای تمام فریمهای کلیدی جداگانه ذخیره شده است در حالیکه texture در تمام فریمها ثابت است . در درسهای بعدی از روشی بنام texture pooling استفاده می کنیم تا تنها یک کپی از texture ها نگهداری کنیم .

آموزش DirectXAudio - بخش اول

آموزش DirectXAudio - بخش اول

موضوع : پخش افکتهای صوتی در برنامه های مالتی مدیا

مقدمه : در سلسله مباحث DirectXAudio شما تکنیکهای لازم برای اضافه کردن موزیک و افکتهای صوتی سریع و دینامیک را به بازیها و برنامه های مالتی مدیا خواهید آموخت . DirectXAudio جایگزینی برای بخشهای DirectSound، DirectSound3D و DirectMusic موجود در DirectX 7 می باشد و دارای امکانات بهتر و سریعتری بوده و برنامه نویسی آن نیز ساده تر است .
در اولین درس از DirectXAudio چگونگی پخش افکتهای صوتی را در برنامه هایتان خواهید آموخت .

Initial کردن DirectSound :
DirectSound اولین مبحثی است که آنرا توضیح خواهم داد . گرچه DirectXAudio یک نام عمومی برای امکانات صوتی DirectX8 می باشد اما بین Sound و Music تفاوت وجود دارد .
DirectSound با پخش افکتهای صوتی ارتباط دارد . DirectSound همانند Direct3D از یکسری device سخت افزاری و نرم افزاری استفاده می کند و افکتهای صوتی در یکسری بافر ذخیره می شوند .
اولین قدم برای برپاسازی DirectSound ، اضافه کردن کتابخانه DirectX8 به پروژه تان می باشد . قدم بعدی تعریف متغیرها و object های موردنیاز است . برای استفاده از DirectSound به متغیرهای زیر نیاز داریم :

```
Private DX As DirectX8
DirectSound8 Private DS As
Private DSBuffer As DirectSoundSecondaryBuffer8
```

```
DSEnum As DirectSoundEnum8 Private
Private bLoaded As Boolean
```

DirectX شی کنترل کننده مرکزی است . DirectSound8 واسط مراقب برای تمام interface های پخش صدا است .
DirectSoundSecondaryBuffer8 داده audio واقعی را برای پخش ذخیره می کند . DirectSoundEnum8 اجازه می دهد که اطلاعاتی را در مورد device های سخت افزاری/نرم افزاری استخراج کنید و متغیر bLoaded یک flag وضعیت می باشد .
حال در برنامه باید لیست تمام device های در دسترس را مشخص کنیم . (این امر کاملاً امکان پذیر است که یک کامپیوتر بیش از یک device برای DirectSound داشته باشد) :

```
Private Sub Form_Load()x
bLoaded = False
As Long Dim I
Set DX = New DirectX8
Set DSEnum = DX.GetDSEnum
DSEnum.GetCount For I = 1 To
MsgBox(DSEnum.GetDescription(I))x
Next I
Sub End
```

فرض کنیم که یکی از device های شناخته شده را انتخاب کردیم . حال بایستی device را واقعاً برپا کنیم :

```
If bLoaded Then
Set DSBuffer = Nothing
Nothing = Set DS
Set DX = Nothing
```

```
End If
Dim DSBDesc As DSBUFFERDESC
New DirectX8 = Set DX
= Set DS
DX.DirectSoundCreate(DSEnum.GetGuid(devicenumber)
)x
frmMain.hWnd,  DS.SetCooperativeLevel
DSSCL_NORMAL
```

متغیر devicenumber شماره device ای است که شما می خواهید با آن کار کنید . DSBDesc فایل صوتی شما را توصیف می کند .

آموزش DirectXAudio – بخش دوم

موضوع : پخش افکتهای صوتی در برنامه های مالتی مدیا

ساخت بافر و play کردن آن : تاکنون ما توانستیم DirectSound را initial کنیم . همانطور که می دانید در تمام component های DirectX داده ها در یکسری بافر ذخیره می شوند . در مورد DirectSound نیز ما یک بافر با نام DirectSoundSecondaryBuffer8 می سازیم و داده های صوتی را در آن قرار می دهیم . برخی پارامتر ها هستند که باید برای بافر تنظیم شوند مثل : stereo یا mono بودن بافر ، ۸ بیتی یا ۱۶ بیتی بودن بافر ، فرکانس صوتی (۲۲ khz، 44khz و غیره) . اگر این پارامترها را مشخص نکنیم DirectSound از اطلاعات فایل صوتی استفاده می کند .

در یک کاربرد ساده ، ما تنها یک بافر صوتی از یک فایل ایجاد می کنیم اما امکان ایجاد چندین بافر بطور همزمان و نیز پخش چندین صدا بطور همزمان نیز وجود دارد :

```
DSBDesc.lFlags = DSBCAPS_CTRLFREQUENCY Or
DSBCAPS_CTRLPAN Or DSBCAPS_CTRLVOLUME
DS.CreateSoundBufferFromFile(App.Path = Set DSBuffer
Sample.wav", DSBDesc)x\" &
SOUND BUFFER CREATED:"x" MsgBox
bytes" & DSBDesc.lBufferBytes & " :MsgBox "Buffer Size
kb)"x" & (Round(DSBDesc.lBufferBytes / 1024, 3 & ")
& ":MsgBox "Buffer Channel Count
DSBDesc.fxFormat.nChannelsIIf(DSBDesc.fxFormat.nCh
Stereo)")x) " ,"(annels = 1, " (Mono
& " :MsgBox "Buffer Bits per channel
bits"x " & DSBDesc.fxFormat.nBitsPerSample
```

در بالا یک بافر صوتی ایجاد شده و اطلاعات صدا از فایل به بافر load شده است .

حال بایستی داده صوتی موجود در بافر را play کنیم :

دستور لازم برای Play کردن بافر بصورت loop :

```
DSBuffer.Play DSBPLAY_LOOPING
```

دستور لازم برای Play کردن بافر بدون loop :

```
DSBuffer.Play DSBPLAY_DEFAULT
```

دستورات لازم برای Stop کردن بافر :

DSBuffer.Stop
۰ DSBuffer.SetCurrentPosition

دستور لازم برای Pause کردن بافر :

DSBuffer.Stop

تنظیم خصوصیات بافر : سه خصوصیت وجود دارد که در مورد بافر تنظیم می شود pannig، volume و frequency

محدوده مقادیر pannig بین اعداد زیر است :

DSBPAN_LEFT = -۱۰،۰۰۰
DSBPAN_CENTER = 0
DSBPAN_RIGHT = 10,000

توسط متد SetPan می توان pannig بافر را تنظیم کرد :

DSBuffer.SetPan yourValue

DirectSound صدا را تقویت نمی کند بلکه آنرا تضعیف می نماید بنابراین ماکزیمم volume عبارت است از volume ای که فایل صوتی با آن ضبط شده است . بعبارت دیگر محدود مقادیر volume بین اعداد زیر است :

DSBVOLUME_MAX = 0
DSBVOLUME_MIN = -10000

توسط متد SetVolume می توان volume بافر را تنظیم کرد :

DSBuffer.SetVolume yourValue

محدود فرکانسی DirectSound عبارت است از :

DSBFREQUENCY_MIN = 100 (hz)x

DSBFREQUENCY_MAX = ۱۰۰۰۰۰ x) 100khz = hz)

توسط متد SetFrequency می توان فرکانس بافر را تنظیم کرد :

DSBuffer.SetFrequency yourValue

آموزش DirectXAudio – بخش سوم

موضوع : پخش موزیک توسط DirectMusic

مقدمه :

در اولین درس از آموزش DirectXAudio با چگونگی پخش افکتهای صوتی آشنا شدید . اکنون این توانایی را دارید که یک engine ساده صوتی بنویسید . در این بخش مبانی پخش موزیک را فرا خواهید گرفت . پس از این درس شما می توانید یک ماژوال برای پخش موزیکهای پس زمینه و افکتهای صوتی برای برنامه هایتان ایجاد کنید .

Initil کردن DirectMusic8 :

قبل از هر کار بایستی ماژول DirectMusic8 را مقداردهی اولیه کنید .
اینکار بصورت زیر انجام می شود :

```
Option ExplicitImplements DirectXEvent8
oDX As DirectX8 Private
Private oDMPerf As DirectMusicPerformance8
oDMLoader As DirectMusicLoader8 Private
Private oDMSeg As DirectMusicSegment8

Dim dmParams As DMUS_AUDIOPARAMS
Set oDX = New DirectX8
oDMPerf = oDX.DirectMusicPerformanceCreate Set
oDX.DirectMusicLoaderCreate = Set oDMLoader
oDMPerf.InitAudio frmMain.hWnd,
dmParams, Nothing,  ,DMUS_AUDIOF_ALL
۱۲۸ ,DMUS_APATH_DYNAMIC_STEREO
oDMPerf.SetMasterAutoDownload True
```

شی DirectMusicLoader8 کمک می کند تا موزیک درون بافر load شود .
شی DirectMusicSegment8 ممموزیکی را که باید پخش شود ذخیره می کند .
کد فوق کافی است یکبار زمانیکه برنامه آغاز می شود ، اجرا گردد .
اکنون ما یک واسط مقدار دهی شده از DirectMusic داریم اما قبل از اینکه موزیک را Load کرده و پخش کنیم چگونگی terminate کردن DirectMusic را در زیر می بینید :

```
If ObjPtr(oDMSeg)Then Set oDMSeg = Nothing
ObjPtr(oDMLoader)Then Set oDMLoader = Nothing If
Then (If Not (oDMPerf Is Nothing
```

```
oDMPerf.CloseDown
Set oDMPerf = Nothing
End If
Then Set oDX = Nothing (If ObjPtr(oDX
```

پیغامها :

در برخی از component های DirectX8 مثل Input , Sound , Music و Play برنامه شما بایستی یک سیستم messaging را برپا کند تا DirectX زمان وقوع برخی رخدادهای خاص را بشما گزارش دهد . این مطلب بخصوص زمانیکه یک موزیک را پخش می کنید مفید است برای مثال می تواند زمان خاتمه یافتن موزیک را به شما اطلاع دهد و آنگاه شما می توانید قطعه موزیک بعدی را پخش کنید .
پیغامها توسط یک سیستم callback انجام می شوند . کد زیر را در تابع InitDMusic تان پس از initial کردن DirectMusic8 قرار دهید :

```
oDMPerf.AddNotificationType
DMUS_NOTIFY_ON_SEGMENT
hEvent = oDX.CreateEvent(Me)x
oDMPerf.SetNotificationHandle hEvent
```

اولین سطر به DirectMusic می گوید چه نوع پیغامهایی را می خواهید به برنامه تان بفرستد . چندین نوع پیغام وجود دارد :
DMUS_NOTIFY_ON_SEGMENT = اطلاعات موزیک فعلی (شروع پخش ، پایان پخش و غیره)
DMUS_NOTIFY_ON_CHORD = اطلاعات تغییر chord موزیک

DMUS_NOTIFY_ON_COMMAND = زمانیکه یک event فرمانی صدا زده شود .

DMUS_NOTIFY_ON_MEASUREANDBEAT = اطلاعات beat/measure مربوط به موزیک فعلی

DMUS_NOTIFY_ON_PERFORMANCE = که event مربوط به سطح performance می باشد .

DMUS_NOTIFY_ON_RECOMPOSE = که recomposition event می باشد .

آخرین بخش از پیغام دهی ، تابع اصلی آن می باشد . همانطور که در بخش Initial کردن DirectMusic دیدید یک توصیف بصورت Implements DirectXEvent8 داشتیم . بخش اصلی تابع callback مربوط به DirectXEvent8 ، شامل یک select case است که بین پیغامهای مختلف سوئیچ می کند :

```
As  Private Sub DirectXEvent8_DXCallback(ByVal eventid
Long)x
If eventid = hEvent Then
Dim dmMSG As DMUS_NOTIFICATION_PMSG
If Not oDMPerf.GetNotificationPMSG(dmMSG) Then
Else
dmMSG.lNotificationOption Select Case
Case DMUS_NOTIFICATION_SEGABORT
DMUS_NOTIFICATION_SEGALMOSTEND Case
Case DMUS_NOTIFICATION_SEGEND
DMUS_NOTIFICATION_SEGLOOP Case
Case DMUS_NOTIFICATION_SEGSTART
Case Else
End Select
End If
End If
End Sub
```

پخش موزیک / متوقف کردن موزیک :

برای پخش یک موزیک ابتدا بایستی آنرا load کنید . اینکار توسط کد زیر انجام می شود :

```
x"\" & oDMLoader.SetSearchDirectory App.Path
& Set oDMSeg = oDMLoader.LoadSegment(App.Path
FILENAME)oDMSeg.SetStandardMidiFile
```

DirectMusic تنها چهار نوع فرمت صوتی را می پذیرد : WAV، MID ، RMI و SEG .

برای پخش فایلهای MP3 بایستی از DirectXShow استفاده کنید که آنرا در درسهای بعدی خواهید دید .

اکنون که داده های فایل صوتی درون بافر load شد می توانید آنرا پخش کنید :

```
oDMSeg.SetRepeats 0
oDMSeg,  oDMPerf.PlaySegmentEx
DMUS_SEGF_DEFAULT, 0
```

تعداد پخش شدن فایل را با متد SetRepets تنظیم کنید . اگر این مقدار صفر باشد ، آهنگ تنها یکبار پخش می شود و اگر ۱- باشد بطور ممتد پخش خواهد شد .

برای متوقف کردن موزیک از کد زیر استفاده کنید :

```
DMUS_SEGF_DEFAULT ,oDMPerf.StopEx oDMSeg, 0
```

برای تنظیم میزان صدا از متد SetMasterVolume استفاده کنید :

oDMPerf.SetMasterVolume yourvalue

رنج صدا بین ۲۰+ دسی بل تا ۲۰۰- دسی بل است .
برای تنظیم Tempo از متد SetMasterTempo استفاده کنید :

oDMPerf.SetMasterTempo yourvalue/ 100

بطور نرمال tempo برابر ۱ می باشد . عدد ۲ سرعت را دو برابر می کند و عدد ۰ موزیک را قطع می کند

آموزش DirectXAudio - بخش چهارم

موضوع : ایجاد صدای سه بعدی توسط DirectSound3D

مقدمه

تاکنون با چگونگی پخش افکتهای صوتی و موسیقی پس زمینه توسط DirectXAudiuo آشنا شدید . این مطالب برای کاربردهای ساده مناسبند اما اینکه فقط ما صدای استریو داشته باشیم کافی نیست و در کاربردهاب حرفه ای بایستی از صداهای کاملاً سه بعدی استفاده کنیم . با استفاده از افکتهای صوتی سه بعدی می توانیم صدا را در تمام جهتها برای کاربر شبیه سازی کنیم اما با همه مزایای صدای سه بعدی ، دو اشکال برای آن وجود دارد : اول اینکه پخش صدای سه بعدی پیچیده تر

از پخش صدای عادی است و تنها کارت های سخت افزاری جدید بطور کاملاً واقعی از آن پشتیبانی می کنند و دوم اینکه صدای سه بعدی با ۴ بلندگو یا بیشتر حاصل می شود – کیفیت حالت ۲ بلندگو بد نیست اما در مقایسه با حالت ۴ بلندگو ، بسیار کیفیت صدای سه بعدی پایین است .

برپاسازی DirectSound3D

برپاسازی صدای سه بعدی چندان پیچیده نیست اما هر بافر صوتی که برای یک صدای سه بعدی می سازید ، یک overhead را به سیستم تان اضافه می کند . همچنین برخی درایورها هستند که تنها اجازه ایجاد تعداد محدودی بافر سه بعدی را در یک لحظه می دهند و نیز اغلب درایورها تعداد بافرهای سه بعدی که می توان در یک لحظه پخش کرد را محدود می کنند (معمولاً ۸ تا ۱۶ بافر) .
اولین قدم در استفاده از صدای سه بعدی تعریف متغیرها و اشیا زیر است :

```
Dim DSBuffer As DirectSoundSecondaryBuffer8
DSBuffer3D As DirectSound3DBuffer8 Dim
DirectSound3DListener8 Dim DSBListener As
```

تنها دو شی آخر برای شما جدید هستند . شی DirectSound3dBuffer8 یک ارائه سه بعدی از بافرهای عادی است . ما همچنان از DirectSoundSecondaryBuffer8 برای نگهداری داده صوتی استفاده می کنیم و از DirectSound3Dbuffer8 برای نگهداری پارامترهای سه بعدی و تنظیمات سه بعدی استفاده می کنیم . شی

DirectSound3Dlistener8 نیز یک listener است و برای تنظیم کردن سرعت و جهت صدا و برخی پارامترهای دیگر استفاده می شود .

مرحله دوم ، ساخت بافر صوتی است . این کار در دو بخش انجام می شود . اول ما یک بافر صوتی نرمال می سازیم و سپس یک واسط بافر صوتی سه بعدی را از آن بدست می آوریم :

```
DSBuffer.Stop If Not (DSBuffer Is Nothing) Then
Set DSBuffer = Nothing
DSBDesc.lFlags = DSBCAPS_CTRL3D Or
DSBCAPS_CTRLVOLUME
Set DSBuffer = DS.CreateSoundBufferFromFile(App.Path
blip.wav", DSBDesc)x\" &
Then ١ < If DSBDesc.fxFormat.nChannels
You can only use mono (1 channel) sounds with " MsgBox
DirectSound3D"x
End If
optLow.Value Then DSBDesc.guid3DAlgorithm =  If
GUID_DS3DALG_NO_VIRTUALIZATION
optMedium.Value Then DSBDesc.guid3DAlgorithm =  If
GUID_DS3DALG_HRTF_LIGHT
optHigh.Value Then DSBDesc.guid3DAlgorithm =  If
GUID_DS3DALG_HRTF_FULL
DSBuffer = DS.CreateSoundBufferFromFile(App.Path Set
DSBDesc)x ,"blip.wav\" &
Set DSBuffer3D = DSBuffer.GetDirectSound3DBuffer()x
```

سه نکته است که باید به آن دقت شود :

۱ – اضافه کردن DSBCAPS_CTRL3D بسیار مهم است . شما اگر این پارامتر را بکار نبرید ، قادر نخواهید بود که واسط سه بعدی را بدست آورید .

۲ – ما بایستی تنها از افکتهای صوتی Mono (تک کاناله) استفاده کنیم

زیرا افکت صوتی استریو در صدای سه بعدی معنا ندارد زیرا صدا از یک نقطه در فضای سه بعدی می آید .

۳ – سطح الگوریتم سه بعدی – که در پارامتر DSBDesc.guid3Dalgorhthm آمده . حالت NO VIRTULIZATION تنها از CPU استفاده می کند و روی تمام سیستم ها کار می کند اما افکتها مینیمم هستند . حالت HRTF LIGHT هم از CPU و هم سخت افزار کارت صوتی استفاده می کند و کیفیت بهتری را نسبت به خالت اول ارائه می دهد . حالت FULL HRTF بهترین حالت است اما در صورتی درست کار می کند که یک سخت افزار سه بعدی داشته باشید .

آخرین پارامتری که باید تنظیم کنیم شی listener است :

```
DSBDesc_2.lFlags = DSBCAPS_CTRL3D Or
DSBCAPS_PRIMARYBUFFER
Set DSBPrimary =
x (DS.CreatePrimarySoundBuffer(DSBDesc_2
DSBPrimary.GetDirectSound3Dlistener = Set DSBListener
,#۰ ,#۱ ,#۰ ,DSBListener.SetOrientation 0#, 0#, 1#
DS3D_IMMEDIATE
```

تا اینجا صدای سه بعدی ما آماده است و می توانیم برخی پخش بافر را مشابه درسهای قبلی شروع کنید .

پارامترهای اختیاری :

چند پارامتر وجود دارد که می توان آنها را تغییر داد :

۱ – Volume : عدد ۰ بیشترین میزان صدا و عدد ۳۰۰۰ – کمترین میزان صدا را دارد :

```
Sub If DSBuffer Is Nothing Then Exit
DSBuffer.SetVolume scrlVolume.Value
```

۲ – Position : تنظیم محل listener :

```
,DSBuffer3D.SetPosition Src_X, 0, Src_Y
DS3D_IMMEDIATE
DSBListener.SetPosition Src_X, 0, Src_Y,
DS3D_IMMEDIATE
```

۳ – Velocity : تنظیم سرعت و جهت منبع صدا :

```
DS3D_IMMEDIATE ,DSBuffer3D.SetVelocity X, Y, Z
DSBListener.SetVelocity X, Y, Z, DS3D_IMMEDIATE
```

۴ – Dppler Effect : انحراف صدا از مسیری که می پیماید انحراف سرعت حرکت صدا :

```
CSng(scrlDoppler.Value),  DSBListener.SetDopplerFactor
DS3D_IMMEDIATE
```

۵ – rolloff : Rolloff Effect چگونگی تضعیف صدا با تغییر فاصله است .

```
CSng(scrlRolloff.Value),  DSBListener.SetRolloffFactor
DS3D_IMMEDIATE
```

۶ – Distance : ماکزیمم فاصله ای که یک صدا می تواند شنیده شود :

DS3D_IMMEDIATE ,DSBuffer3D.SetMaxDistance 250
DSBuffer3D.SetMinDistance 0.01, DS3D_IMMEDIATE

آموزش DirectX Input – مقدمه

آموزش DirectX Input – مقدمه

مقدمه

در بخش مباحث برنامه نویسی DirectX Input شما می آموزید که چگونه اطلاعات ورودی را از کاربر بگیرید . این اطلاعات می تواند از طریق کیبرد ، ماوس ، جوی استیک یا گیم پد باشد .
بنابراین در پایان این سلسله مباحث قادر خواهید بود که یک بازی بنویسید و یا یک برنامه مالتی مدیا که بتواند توسط کاربر کنترل شود .
DirectInput ساده ترین و سریع ترین روش برای گرفتن داده ها از هر نوع ابزار ورودی می باشد .

مباحث این بخش

بخش اول : چگونگی دریافت اطلاعات از کیبرد

بخش دوم : چگونگی دریافت اطلاعات از ماوس
بخش سوم : راهنمای استفاده از Action Mapping

آموزش DirectX Input - بخش اول

گرفتن اطلاعات ورودی از کیبرد - ۱

مقدمه

Direct Input ۸ همانطور که از نامش مشخص است به شما اجازه می دهد که بتوانید برنامه هایی بنویسید که توسط هر نوع دستگاه ورودی کنترل شود .
Direct Input 8 دارای چندین مزیت نسبت به استفاده از کنترلهای ورودی خود ویژوال بیسیک دارد – کنترلهایی مثل Form_KeyUp, Form_KeyDown, Form_MouseMove - و همچنین قابلیت کنترل بیشتری نسبت به توابع استاندارد Win32 از قبیل GetCursorPos, GetKeyState دارد .
Direct Input 8 سریعتر ، کاراتر و قدرتمند تر بوده و برای ساخت بازیها طراحی شده بنابراین باعث کندی برنامه ها نخواهد شد .

چگونگی کار با Direct Input 8 برای گرفتن ورودی از کیبرد

دو روش برای استفاده از کیبرد در DirectX8 وجود دارد : روش polling و روش event-based که هر دو دارای مزایا و معایبی هستند .
بطور کلی در اغلب طراحیها از روش event-based استفاده می شود زیرا کار با آن راحت تر اسن . در این روش هر پیغام فرستاده شده

ازطرف دستگاه ورودی log می شود و برنامه نیازی به هیچگونه پردازشی بمنظور منتظر ماندن برای یک پیغام از طرف ورودی ندارد ، بنابر این کاراتر است . در روش polling کنترل کمی دقیقتر و راحتر است .

اگر در مورد برنامه نویسی بر مبنای polling و بر مبنای event اطلاعات کافی ندارید می توانید از منابع موجود در سایتهایی چون Gamasutra و GameDev استفاده کنید .

روش Polling

مراحل این روش عبارتند از :

۱ – تعریفات Declarations : یک فرم ایجاد کرده و یک TextBox به نام txtOutput با خصوصیات Multiline، Locked و Vertical Scroll Bar در آن قرار دهید . کدهای زیر را در بخش کدنویسی این فرم بنویسید :

```
True = Private Const UsePollingMethod As Boolean
Private Const UseEventMethod As Boolean = False
'نکته مهم اینست که تنها یکی از دو ثابت فوق بایستی True باشد .
Boolean Private bRunning As
'این متغیر برای polling استفاده می شود
DirectX8 Private DX As
Private DI As DirectInput8
'تعریف شی اصلی DirectX و شی DirectInput
Private DIDevice As DirectInputDevice8
DIKEYBOARDSTATE Private DIState As
'این دو شی برای دسترسی به دستگاه ورودی ( کیبرد ) استفاده می
```

شوند

```
Private KeyState(0 To 255) As Boolean
'آرایه ای برای تشخیص فشرده شدن کلید
Private Const BufferSize As Long = 10
' سایز بافر نگهدارنده event ها . در روش event-based این مقدار
برابر یک و در روش polling برابر ۱۰ تا ۲۰ است ( بسته به سرعت
حلقه بازی )
Private Declare Sub Sleep Lib "kernel32" (ByVal
dwMilliseconds As Long)x
' تابع Sleep برای متوقف کردن حلقه polling در صورت بالا بودن نرخ
ورودی
```

۲- مقدار دهی اولیه Initialisation : این بخش سه مرحله دارد :

در مرحله اول اشیا و Device ها ساخته می شوند .

در مرحله دوم تنظیمات مربوط به Device انجام می شود .

در مرحله سوم به Device می گوئیم که می خواهیم شروع به استفاده از آن کنیم .

در Form_Load کدهای زیر را بنویسید :

```
Me.Show
Dim I As Long
DIPROPLONG Dim DevProp As
Dim DevInfo As DirectInputDeviceInstance8
BufferSize) As  Dim pBuffer(0 To
DIDEVICEOBJECTDATA
Then If UsePollingMethod And UseEventMethod
MsgBox "You must select only one of the constants before
```

```
running"x
Unload Me
End
End If

txtOutput.Text = "Using  If UsePollingMethod Then
vbCrLf & "Polling Method
txtOutput.Text = "Using Event  If UseEventMethod Then
vbCrLf & "Based Method

'مقداردهی اولیه روش انتخاب شده
Set DX = New DirectX8
Set DI = DX.DirectInputCreate
DI.CreateDevice("GUID_SysKeyboard")x = DIDevice Set

DIDevice.SetCommonDataFormat
DIFORMAT_KEYBOARD
DIDevice.SetCooperativeLevel frmMain.hWnd,
Or ISCL_NONEXCLUSIVE DISCL_BACKGROUND

'برپاسازی بافر
DIPH_DEVICE = DevProp.lHow
DevProp.lData = BufferSize
DIPROP_BUFFERSIZE, DevProp DIDevice.SetProperty

' به دایرکت ایکس می گوئیم که می خواهیم از دستگاه ورودی استفاده
کنیم
DIDevice.Acquire

'استخراج اطلاعاتی در مورد دستگاه ورودی
Set DevInfo = DIDevice.GetDeviceInfo()x
```

```
& " :Product Name" & txtOutput.Text = txtOutput.Text
vbCrLf & DevInfo.GetProductName
& " :Device Type" & txtOutput.Text = txtOutput.Text
vbCrLf & DevInfo.GetDevType
& " :GUID" & txtOutput.Text = txtOutput.Text
vbCrLf & DevInfo.GetGuidInstance
```

```
'در صورتی که بخواهیم به برنامه خاتمه بدهیم کدهای زیر را می
نویسیم
DIDevice.Unacquire
Nothing = Set DIDevice
Set DI = Nothing
Set DX = Nothing
Me Unload
End
```

۳ – گرفتن ورودی از کیبرد : در این بخش فرض کنید بخواهیم یک بازی را در یک حلقه Do-Loop شبیه سازی کنیم . در این حلقه هر بار فشرده شدن کلیدهای کیبرد را چک می کنیم :

```
If Not Err.Number Then bRunning = True

While bRunning Do

'دریافت اطلاعات شامل خواندن وضعیت کیبرد ، خواندن اطلاعات بافر و
سپس خطا
DIState DIDevice.GetDeviceStateKeyboard
```

```
DIDevice.GetDeviceData pBuffer, DIGDD_DEFAULT
DI_BUFFEROVERFLOW Then = If Err.Number
Msgbox("BUFFER OVERFLOW (Compensating)...")x
:ENDOFLOOP GoTo
End If
'بررسی فشرده شدن کلیدها
For I = 0 To 255
DIState.Key(I) = 128 And (Not KeyState(I) = True)  If
Then
& " { DOWN }" & txtOutput.Text = txtOutput.Text
vbCrLf &((KeyNames(CInt(I
txtOutput.SelStart = Len(txtOutput.Text)x
KeyState(I) = True
If End
Next I

'بررسی رها شدن کلید
For I = 0 To BufferSize
KeyState(pBuffer(I).lOfs) = True And pBuffer(I).lData  If
Then = 0
KeyState(pBuffer(I).lOfs) = False
& " { UP }" & txtOutput.Text = txtOutput.Text
vbCrLf & ((KeyNames(CInt(pBuffer(I).lOfs
txtOutput.SelStart = Len(txtOutput.Text)x
End If
I Next

Sleep (50)x
DoEvents
:ENDOFLOOP
Loop
```

در کد فوق یک تابع KeyName وجود دارد که نام کلید فشارداده شده را بر می گرداند . بخشی از این تابع را در زیر می بینید :

```
String Function KeyNames(iNum As Integer) As

Dim aKeys(0 To 255) As String

"DIK_ESCAPE" = (aKeys(1
aKeys(2) = "DIK_1 On main keyboard"x
main keyboard"x aKeys(3) = "DIK_2 On
aKeys(4) = "DIK_3 On main keyboard"x
main keyboard"x aKeys(5) = "DIK_4 On
aKeys(6) = "DIK_5 On main keyboard"x
main keyboard"x aKeys(7) = "DIK_6 On
aKeys(8) = "DIK_7 On main keyboard"x
main keyboard"x aKeys(9) = "DIK_8 On
aKeys(10) = "DIK_9 On main keyboard"x
On main keyboard"x aKeys(11) = "DIK_0
aKeys(12) = "DIK_MINUS On main keyboard"x
DIK_EQUALS On main keyboard"x" = (aKeys(13
aKeys(14) = "DIK_BACK BACKSPACE"x
DIK_TAB"x" = (aKeys(15
aKeys(16) = "DIK_Q"x
aKeys(17) = "DIK_W"x
DIK_E"x" = (aKeys(18
aKeys(19) = "DIK_R"x
aKeys(20) = "DIK_T"x
.
.
.
KeyNames = aKeys(iNum)x
```

End Function

آموزش DirectX Input - بخش دوم

موضوع : کنترل کیبرد با روش Event-Based

مقداردهی اولیه و مفاهیم اصلی در روش Event-Based مشابه روش Polling است و تنها بایستی ساختار بخش جمع آوری داده و حلقه پردازشی را تغییر دهیم . مراحل کار با روش Event-Based بصورت زیر می باشد :

۱ - تعاریف و مقداردهی اولیه : در بخش تعاریف دو تعریف جدید بصورت زیر داریم :

```
Dim hEvent As Long
DirectXEvent8 Implements
```

hEvent یک پارامتر هندل برای یک می باشد .
نکته : زمانی که کلیدی فشرده یا رها می شود ، DirectX این امر با فراخوانی تابعی به اسم DirectXEvent8_DXCallback به برنامه شما اطلاع می دهد . (این نوع توابع را Back Function Call گویند) . این تابع به برنامه شما می گوید که یک رویداد اتفق افتاده است و بایستی بافرها را چک کند .

تنها تغییری که در بخش مقداردهی اولیه نیاز است ، برپاسازی یک event می باشد :

```
If UseEventMethod Then
DX.CreateEvent(frmMain)x = hEvent
DIDevice.SetEventNotification hEvent
If End
```

در انتهای برنامه نیز کد زیر را برای از بین بردن event اضافه کنید :

```
hEvent Then DX.DestroyEvent ٠ <> If hEvent
```

۲ – استفاده از event : برای این بخش کدهایی را در داخل تابع DirectXEvent8_DXCallback می نویسیم :

```
As  Private Sub DirectXEvent8_DXCallback(ByVal eventid
Long)x
'متغیرهای موردنیاز
Dim I As Long
As  (Dim pBuffer(0 To BufferSize
DIDEVICEOBJECTDATA
If eventid = hEvent Then
Exit Sub If DIDevice Is Nothing Then
'درصورت رخ دادن event داده را از کیبرد می گیریم
DIDevice.GetDeviceStateKeyboard DIState
DIGDD_DEFAULT ,DIDevice.GetDeviceData pBuffer
'چک کردن تمام کلیدها برای اینکه متوجه شویم چه اتفاقی افتاده است
For I = 0 To 255
```

```
'عدد ۱۲۸ نشان دهنده key_down event است .
DIState.Key(I) = 128 Then If
If pBuffer(0).lData = 128 Then
& " { DOWN }" & txtOutput.Text = txtOutput.Text
vbCrLf & ((KeyNames(CInt(I
If End
End If
'کد فوق برای بررسی فشرده شدن یک کلید بود . کد زیر رها شدن کلید
را بررسی می کند
Then (If (pBuffer(0).lData = 0 And pBuffer(0).lOfs = I
& "{ UP }" & txtOutput.Text = txtOutput.Text
vbCrLf & ((KeyNames(CInt(I
End If

txtOutput.SelStart = Len(txtOutput.Text)x
I Next
End If
End Sub
```

آموزش DirectX Input – بخش سوم

موضوع : کنترل ماوس با DirectX Input

مقدمه :

برای استفاده از ماوس در برنامه های مالتی مدیا و بازیها همانند کی برد

می توانیم از امکانات دایرکت ایکس استفاده کنیم . روش کنترل ماوس توسط DirectX Input بسیار ساده بوده و مشابه کنترل کیبرد می باشد بنابراین درصورتی که دو درس گذشته را نخوانده این پیشنهاد می کنم ابتدا آنها را مطالعه کنید .

برپاسازی Device :

علاوه بر متغیرهایی که در بخش کنترل کیبرد تعریف شد بایستی متغیرهای جدید زیر را نیز در ابتدای برنامه تان تعریف کنید :

```
Private Const mSpeed As Single = 2
BufferSize As Long = 10 Private Const
Private mPosition As Point
```

mSpeed مقدار سرعت حرکت کرسر ماوس را مشخص می کند .
BufferSize سایز بافر DI می باشد .
mPosition موقعیت جاری کرسر ماوس را نشان می دهد .

در مرحله بعدی بایستی مقداردهی های اولیه لازم را انجام دهید :

```
DI.CreateDevice("guid_SysMouse")x = Set DIDevice
Call
DIDevice.SetCommonDataFormat(DIFORMAT_MOUSE)
x
DIDevice.SetCooperativeLevel(frmMain.hWnd,  Call
DISCL_EXCLUSIVE)x DISCL_FOREGROUND Or
```

تفاوت عمده کدهای فوق با کدهای مقداردهی اولیه در بخش کی برد

آنست که cooperativelevel تغییر کرده است . در اینجا گفته شده که ما می خواهیم از ماوس بصورت انحصاری در برنامه استفاده کنیم . این حالت برای برنامه های window-base مناسب نیست و بهترست از آن در بازیهایی که بصورت full screan هستند استفاده کنید .

خواندن ورودی از ماوس :

در این بخش می توانید هم از روش polling و هم event-based استفاده کنید . نکته مهمی که در اینجا وجود دارد آنست که Direct Input فقط حرکت داده شدن ماوس و کلیک شدن یک دکمه را به شما اطلاع می دهد و برای تشخیص حالتهای double click و single click خودتان بایستی کد بنویسید برای مثال اگر فاصله زمانی بین دو کلیک کمتر از ۴۰ میلی ثانیه باشد آنگاه این یک double click بوده است . کد زیر حرکت داده شدن ماوس و کلیک یکی از سه دکمه آنرا اطلاع می دهد :

```
Dim DevData(1 To BufferSize) As
DIDEVICEOBJECTDATA
Dim nEvents As Long
Dim I As Long
DIDevice.GetDeviceData(DevData,  = nEvents
DIGDD_DEFAULT)x
nEvents For I = 1 To
Select Case DevData(I).lOfs
Case DIMOFS_X
mPosition.x + (DevData(I).lData * mSpeed)x = mPosition.x
mPosition.x = 0 Then ۰ > If mPosition.x
= frmMain.ScaleWidth Then mPosition.x < If mPosition.x
frmMain.ScaleWidth
```

```
imgCursor.Top = mPosition.y
mPosition.x = imgCursor.Left
mPosition.x & "] :lablel(1).Caption = "Mouse Coordinates
x"[" & mPosition.y & " ," &
Case DIMOFS_Y
mPosition.y + (DevData(I).lData * mSpeed)x = mPosition.y
mPosition.y = 0 Then ، > If mPosition.y
= frmMain.ScaleHeight Then mPosition.y < If mPosition.y
frmMain.ScaleHeight
imgCursor.Top = mPosition.y
mPosition.x = imgCursor.Left
mPosition.x & "] :lablel(1).Caption = "Mouse Coordinates
x"[" & mPosition.y & " ," &
DIMOFS_BUTTON0 Case
& " :label(2).Caption = "Button 0 State
IIf(DevData(I).lData = 0, "Up", "Down")x
DIMOFS_BUTTON1 Case
& " :label(3).Caption = "Button 1 State
IIf(DevData(I).lData = 0, "Up", "Down")x
DIMOFS_BUTTON2 Case
& " :label(4).Caption = "Button 2 State
IIf(DevData(I).lData = 0, "Up", "Down")x
DIMOFS_BUTTON3 Case
& " :label(5).Caption = "Button 3 State
IIf(DevData(I).lData = 0, "Up", "Down")x
End Select
Next I
```

برای استفاده از کد فوق در روش Polling ، بایستی آنرا در یک حلقه Do

while-Loop قرار دهید .

برای استفاده از کد فوق در روش Event-Based ، بایستی آنرا درون روتین DirectXEvent8_DXCallback قرار دهید .

ترفندها

تشخیص فشرده شدن کلیدهای کیبرد

یکی از دوستان سوال کرده بودند که چگونه می توان کلیدهای کیبرد را حتی وقتی فوکوس روی برنامه ما نیست تشخیص داد مانند دیکشنری ها که مثلاً با CTRL+F12 فعال می شوند و یا Keylogger ها که کلیدهای فشرده شده را ثبت می کنند .

من دو روش زیر را برای اینکار پیشنهاد می کنم :

۱ – استفاده از یک تابع کتابخانه ای به اسم GetAsyncKeyState موجود در کتابخانه user32.dll . این تابع ، فشرده شدن یا رها شدن یک کلید را تشخیص می دهد . نحوه declare کردن این تابع بصورت زیر است :

```
Private Declare Function GetAsyncKeyState Lib "user32"
(ByVal vKey As Long) As Integer
```

حال در برنامه تان یک timer قرار داده و در event آن کد زیر را قرار دهید :

```
For i = 1 To 255
   results = 0
```

```
    results = GetAsyncKeyState(i)
    If results <> 0 Then
      Msgbox(Chr(i))
    End If
  Next
```

برای مشاهده یک برنامه نمونه به این آدرس مراجعه کنید .

۲ – استفاده از قلاب یا Hook : قلاب ، یک ابزار در مکانیزم مدیریت پیغام سیستم ویندوز است که توسط آن برنامه ها می توانند یک روتین را برای مدیریت و پردازش پیغامهای خاصی قبل از اینکه آن پیغامها به برنامه مقصد برسند نصب نمایند . قلابها باعث کندی سیستم می شوند زیرا حجم پردازشی سیستم روی هر پیغام را افزایش می دهند بنابراین بایستی زمانیکه واقعاً به قلاب نیاز دارید آنرا نصب نموده و هر چه زودتر آنرا حذف نمایید . سیستم ویندوز از انواع زیادی از قلابها پشتیبانی می کند که هر کدام امکان دستیابی به پیغامهای خاصی را مهیا می نمایند برای مثال یک برنامه کاربردی می تواند با استفاده از قلاب کیبرد برای مدیریت و پردازش پیغامهای مربوط به آن (مثل فشرده شدن یک کلید خاص یا رها شدن آن) استفاده کند .

برای نصب یک قلاب در برنامه از یک تابع کتابخانه ای به اسم SetWindowsHookEx استفاده می شود . این تابع یک قلاب را به زنجیره قلابهای سیستم اضافه می کند . نحوه declare کردن این تابع بصورت زیر است :

```
Declare Function SetWindowsHookEx Lib "user32" Alias
"SetWindowsHookExA" (ByVal idHook As Long, ByVal
lpfn As Long, ByVal hmod As Long, ByVal dwThreadId
As Long) As Long
```

همچنین برای آزاد کردن یک قلاب و حذف آن از زنجیره قلابها از تابع کتابخانه ای UnhookWindowsHookEx استفاده می گردد . نحوه declare کردن این تابع بصورت زیر است :

```
Declare Function UnhookWindowsHookEx Lib "user32"
(ByVal hHook As Long) As Long
```

برای ایجاد قلاب کیبرد همچنین نیاز به تعریف یک ثابت است که شماره قلاب کیبرد در آن قرار دارد :

```
Public Const WH_KEYBOARD = 2
```

حال بایستی یک تابع پس زمینه یا Callback Function نوشت که به ازای فشرده شدن کیبرد اجرا شود و آدرس آنرا (با استفاده از کلمه کلیدی Address Of) بهمراه ثابت فوق به تابع SetWindowsHookEx فرستاد .

برای اطلاعات بیشتر و مشاهده یک نمونه برنامه به این آدرس مراجعه کنید

استخراج مشخصات سخت افزاری یک سیستم در وی بی

در این بخش یک کنترل Ocx معرفی می شود که بوسیله آن می توانید مشخصات سخت افزاری سیستم خود را استحراج کنید .
این کنترل را که Hardware Info نام دارد می توانید از اینجا دانلود نمایید .

پس از باز نمودن فایل zip دانلود شده مشاهده خواهید کرد که دو فایل dll و یک فایل ocx در آن وجود دارد . همچنین یگ فایل راهنما نیز بهمراه

آنها وجود دارد که طریقه استفاده از کنترل را نشان می دهد . برای استفاده از کنترل فوق وارد محیط ویژال بیسیک شده و سپس وارد منوی Components شوید . در آنجا روی دکمه Browse کلیک کنید . وارد پوشه ای که فایل zip را در آنجا باز کرده اید شده و فایل HWInfo.ocx را انتخاب کنید تا این کنترل به لیست کنترلهای نوار ابزار شما اضافه شود . حال می توانید از کنترل را روی فرم خود قرار دهید و از امکانات آن استفاده کنید .

این کنترل دارای خصوصیات زیر است :

BaseBoardManufacturer : مشخصات سازنده مادربورد

BaseBoardProduct : نوع چیپ ست مادربورد

BiosVendor : سازنده بایوس

BiosReleaseDate : تاریخ انتشار بایوس

BiosVersion : ورژن بایوس

BiosROMSize : سایز حافظه رام بایوس

SocketDesignation : نوع سوکت پردازنده

ProcessorType : نوع پردازنده

ProcessorManufactor : سازنده پردازنده

ProcessorID : شماره ID پردازنده

ProcessorSerialNumber : شماره سریال پردازنده

با استفاده از این کنترل همچنین می توان اطلاعات هر چهار هارد دیسک IDE سیستم را استخراج نمود برای مثال اگر بخواهید اطلاعات Primary Hard (شماره یک) را بدست آورید از خصوصیات زیر استفاده کنید :

HardDisk1ModelNumber : شماره مدل هارددیسک

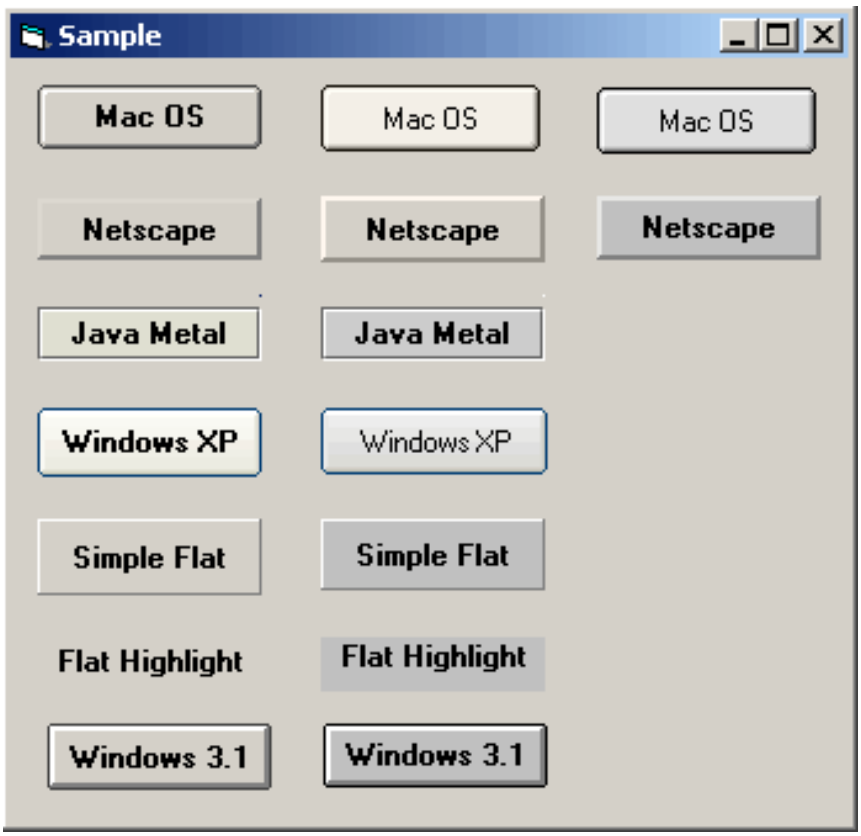
HardDisk1SerialNumber : شماره سریال هارد دیسک (شماره

سریال کارخانه)

خصوصیات دیگری نیز در این کنترل وجود دارد که برای اطلاعات بیشتر به راهنمای آن مراجعه کنید

یک کنترل خوب برای ویژوال بیسیک

استفاده از یک کنترل خوب در ویژوال بیسیکاین کنترل انواع مختلفی از کلید رو به شکل کلیدهای ویندوز XP و ویندوز ۳,۱ و مکینتاش و نت اسکپ و ... را در اختیار مان می گذارد که نمونه هایی از این رو تو عکس زیر می بینیم :



در اینجا لازم می بینم که نحوه استفاده از کنترلها تو VB رو براتون بگم .

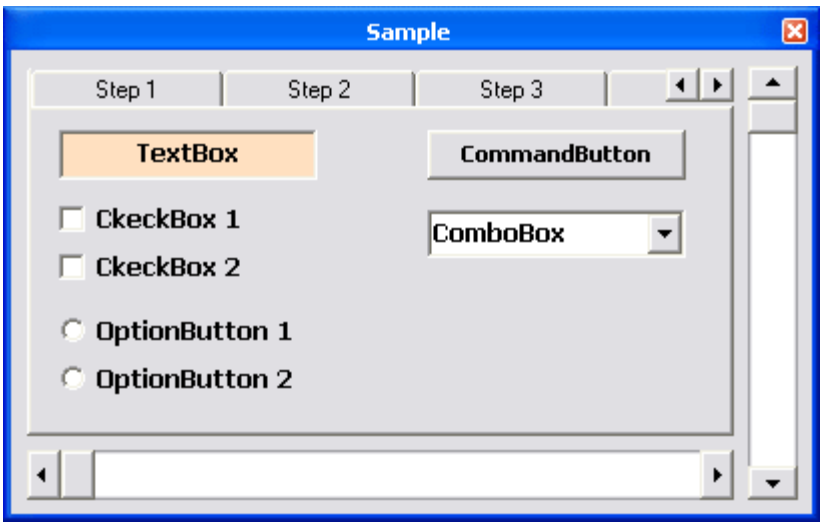
ابتدا فایل فایل کنترل که با پسوند ctl رو در محلی که پروژه مون قرار دارد کپی میکنیم و بعد از منوی Project آیتم Add user control رو انتخاب

مکنیم می بینیم که پنجره ای ظاهر میشود سربرگ Exiting را انتخاب کرده و بعد فایل کنترل مورد نظر را انتخاب می کنیم می بینیم که این فایل در پنجره Project Explorer به پروژه مون اضافه میشود و از طرفی آیکنی نیز در به **Toolbox** مون اضافه میشود ولی ممکن است که گاهی این آیکن غیر فعال باشد برای فعال کردن آن کافیست تمام پنجره ها را در VB بسته و دوباره آنرا از Project Explorer باز کنیم می بینیم که این آیکن فعال می شود اکنون می توانید از آن کنترل در فرمهایتان در Visual Basic استفاده کنید .

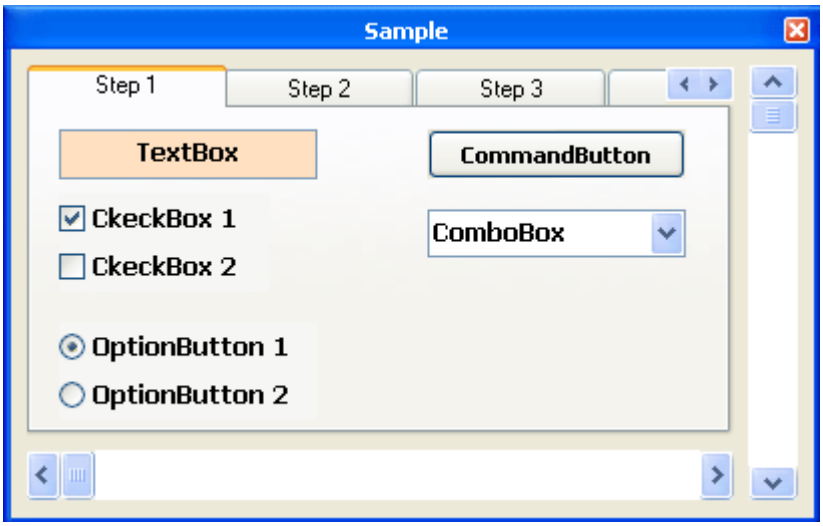
برای دریافت این کنترل اینجا کلیک کنید .

::استفاده از Windows XP Themes در ویژوال بیسیک

در ابتدا برای اینکه این مطلب جا بیفتد نظری به شکلهای زیر بیاندازند



شکل ۱ – حالت عادی فرم



شکل ۲ – حالت فرم با استفاده از **WinXP Themes**

برای این کار باید از **API** ها استفاده کنیم . **API** برگرفته شده از سر واژه های **Application Programming Interface** (رابط برنامه نویسی کاربردی) است . **API** ها توابعی از پیش ساخته شده در سیستم عامل ویندوز هستند که می توانیم از انها در برنامه نویسی در **Visual Basic** استفاده کنیم . این توابع در چندین فایل **Dll** موجودند که از جمله این فایلها متوان **KERNEL32** , **GDI32** , **USER32** ,

WINMM و .... ما برای استفاده از Themes WinXP باید از تابع InitCommonControls که در کتابخانه ی که در و یندوز XP موجود است استفاده می کنیم . برای فراخوانی (Declare) یک تابع از متد زیر استفاده میکنیم :

```
rivate Declare Function <نام تابع> Lib " Dll کتابخانه نام " ()
s Long
```

پس برای فراخوانی تابع InitCommonControls با توجه به متد بالا در قسمت General فرم در ویژوال بیسیک کد زیر را می نویسیم :

```
Lib InitCommonControls Private Declare Function
As Long () "comctl32.dll"
```

اکنون برای form در رویداد Initialize کد زیر را بنویسید (رویداد Initialize وقتی فراخوانی می شود که فرم ایجاد شود) :

```
rivate Sub Form_Initialize()
  InitCommonControls
d Sub
```

تا اینجا کارمان با فرم تمام شد و اکنون می توانید چند CommandButton , CheckBox , OptionButton , . . . را به فرم اضافه کنید . و پس از اتمام کار بر روی فرم آنرا Compile کرده و بصورت Exe می سازیم .

یک Document New Text را در همان پوشه ای که فایل exe وجود دارد می سازیم و کدهای زیر را در آن کپی می کنیم و بعد آنراذخیره می کنیم . حالا باید نام و پسوند فایل را تغییر دهیم و مطابق الگوی مقابل عمل کنیم .EXE.MANIFEST نا فایل exe

برای اینکه نام و پسوند فایل را عوض کنیم نیاز به این داریم که نام و پسوند فایل را به طور کامل ببینیم برای این کار وارد Control Panel شده و Folder Option را

اجرا می کنیم بعد به قسمت View شده و تیک Hide Extentions For Known File Type را بر می داریم حالا همه فایلها را با نام و پسوند کامل مشاهده می کنیم و میتوانیم با استفاده از Rename (کلید F2) نام فایل را با توجه به متد بالا تغییر دهیم مثلا اگر نام فایل exe بصورت Sample.exe باشد نام فایل که با پسوند txt است ر به Sample.exe.MANIFEST تغییر می دهیم .

-=-

کد مربوط به فایل EXE.MANIFEST.نام فایل exe

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-
com:asm.v1" manifestVersion="1.0">
<assemblyIdentity type="win32"
processorArchitecture="*" version="6.0.0.0"
name="mash"/>
<description>Enter your Description
Here</description>
<dependency>
<dependentAssembly>
<assemblyIdentity
type="win32"
name="Microsoft.Windows.Common-Controls"
version="6.0.0.0"
language="*"
processorArchitecture="*"
publicKeyToken="6595b64144ccf1df"
/>
</dependentAssembly>
</dependency>
</assembly>
```

::چطور تو Visual Basic کیبرد فارسی بشه

میخواهم براتون بگم که چطور می شه تو **VB** وقتی برنامه مون اجرا شد بشه برای ورود اطلاعات از زبان فارسی استفاده کرد

برای اینکار اول باید از یه **API** استفاده کرد . تابع **LoadKeyboardLayout** رو از کتابخانه **User32** به صورت زیر فراخوانی کنیم :

```
Public Declare Function LoadKeyboardLayout Lib "user32" Alias "LoadKeyboardLayoutA" (ByVal pwszKLID As String, ByVal flags As Long) As Long
```

بعد از برای فراخوانی تابع باید تابع را در رویداد مورد نظر فراخوانی کرد مثلا من تابع رو وقتی که فرم برنامه **load** شده فراخونی کردم :

```
LoadKeyboardLayout "00000429", 1 ' 00000429 :::::> For Farsi Keyboard
```

برای اینکه بعد از **Unload** شدن فرم برنامه زبان ویندوز به انگلیسی برگردد کد زیر را میتوانیم در رویداد **unload** بنویسیم

```
LoadKeyboardLayout "00000409", 1 ' ۰۰۰۰۰۴۲۹ :::::>
```

For English Keyboard

:: در ضمن اگر مایل به دیدن کد سایر زبانها هستید یه سری به آدرس زیر در رجیستری ویندوز بزنید .

```
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Keyboard Layouts
```

یک کد کاربردی دیگه استفاده از تایع(sendkeys)

خوب اینجا می خوام یک کد کاربردی دیگه رو بهتون بگم . این کد باعث می شه که وقتی شما رویداد خاصی رو اجرا مکنید , کلید خاصی از کیبرد اجرا شود یعنی مثلاً اگر شما روی یک **Textbox** هستید و کلید **Enter** را فشردید عملی معادل فشردن کلید ..... , **Tab**, **Delete,Pagedown** , **F1 ,F2** روی دهد :

```
Private Sub TextBox_KeyPress(KeyAscii As
Integer)
    If KeyAscii = 13 Then
       SendKeys "{tab}"
    End If
End Sub
```

باز هم این کد را در رویداد **keypress** پیشنهاد می کنم .

در اینجا کدهایی که می توان از طریق این تابع با آنها کار کرد را می بینیم :

Key	Code
BACKSPACE	{BACKSPACE}, {BS}, or {BKSP}
BREAK	{BREAK}
CAPS LOCK	{CAPSLOCK}
DEL or DELETE	{DELETE} or {DEL}
DOWN ARROW	{DOWN}
END	{END}
ENTER	{ENTER} or ~
ESC	{ESC}
HELP	{HELP}
HOME	{HOME}
INS or INSERT	{INSERT} or {INS}
LEFT ARROW	{LEFT}
NUM LOCK	{NUMLOCK}

PAGE DOWN	{PGDN}
PAGE UP	{PGUP}
PRINT SCREEN	{PRTSC}
RIGHT ARROW	{RIGHT}
SCROLL LOCK	{SCROLLLOCK}
TAB	{TAB}
UP ARROW	{UP}
F1	{F1}
F2	{F2}
F3	{F3}
F4	{F4}
F5	{F5}
F6	{F6}
F7	{F7}
F8	{F8}
F9	{F9}
F10	{F10}
F11	{F11}
F12	{F12}
F13	{F13}
F14	{F14}
F15	{F15}
F16	{F16}

آموزش کار با توابع مربوط به زمان و تاریخ

در این برنامه میخوام درباره استفاده از توابع مربوط به زمان و تاریخ بهتون یه چیزایی بگم. برای تفهیم بیشتر این مطالب همه ی این توابع رو

در یک مثال کاربردی براتون بیان می کنم که امیدورم براتون مفید واقع بشه .

:: برای دریافت این برنامه اینجا کلیک کنید .

برای این مثال بعد از ایجاد فرم در ویژوال بیسیک چند **Label** و یک **Timer** به فرم اضافه کنید , ابتدا با تابع **Format** شروع می کنیم . این تابع برای فرمت بندی خروجی به کار می رود و ما در اینجا از آن برای فرمت بندی زمان و تاریخ استفاده می کنیم . شکل کلی این تابع بصورت زیر است :

Format(*expression*[, *format*[, *firstdayofweek*[, *firstweekofyear*]]])

:: برای فرمت بندی زمان و تاریخ در تابع بالا می توان ار فرمت های زیر استفاده کرد :

مقادیری پارامتر format	
Format	**Description**
General	نمایش زمان و تاریخ با هم
Long Date	نمایش تاریخ به طور کامل
Medium Date	نمایش تاریخ بصورت خلاصه
Short Date	نمایش تاریخ
Long Time	نمایش زمان بصورت ساعت , دقیقه ,

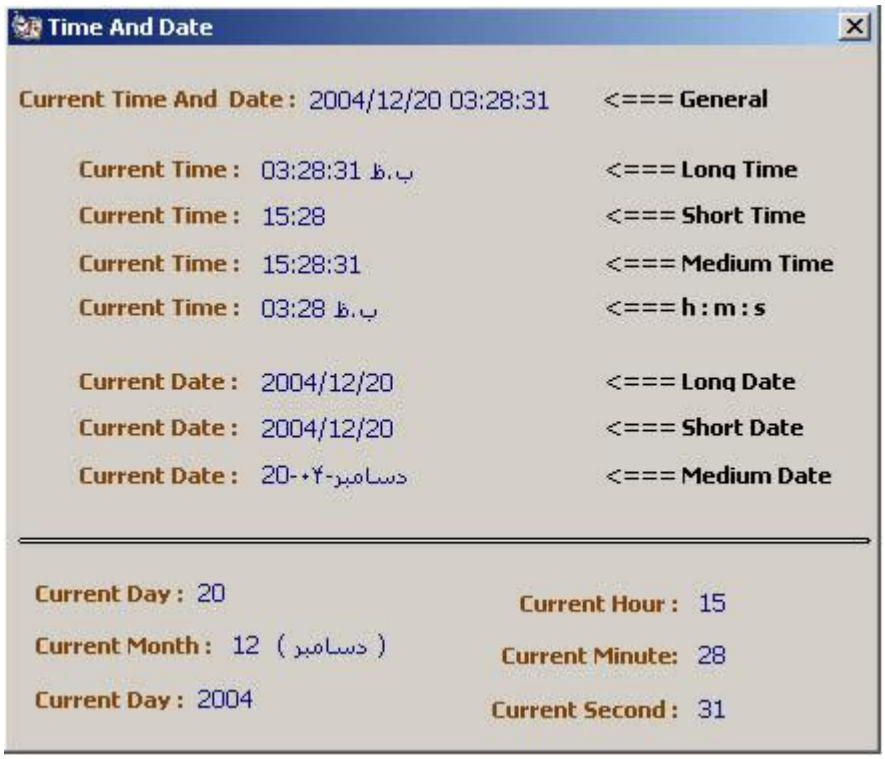
	ثانیه و AM/PM
Medium Time	نمایش زمان بصورت ساعت , دقیقه و ثانیه
Short Time	نمایش زمان بصورت ساعت و دقیقه

به کد زیر توجه کنید و نتایج آنرا در نمای برنامه ببینید :

```
Private Sub Timer1_Timer()

LblTime.Caption = Format(Now, "long time")
'Not Diffrence With ===> LblTime.Caption =
Format(Time, "long time")

LblTime2.Caption = Format(Now, "short
time")
'Not Diffrence With ===> LblTime2.Caption =
Format(Time, "short time")

LblTime3.Caption = Format(Now, "h:m:s")
'Not Diffrence With ===> LblTime3.Caption =
Format(Time, "h:m:s")

LblTime4.Caption = Format(Now, "Medium
Time")
'Not Diffrence With ===> LblTime4.Caption =
Format(Time, "Medium Time")



LblDate.Caption = Format(Now, "long date")
'Not Diffrence With ===> LblDate.Caption =
Format(Date, "long date")
```

```
LblDate2.Caption = Format(Now, "short
Date")
'Not Diffrence With ===> LblDate2.Caption =
Format(Date, "short Date")

LblDate3.Caption = Format(Now, "Medium
Date")
'Not Diffrence With ===> LblDate3.Caption =
Format(Date, "Medium Date")

LblTimeDate.Caption = Format(Now, General)



LblDay.Caption = Day(Date)
' Not Diffrence With ====> LblDay.Caption =
Day(Now)

LblMonth.Caption = Month(Date) & " ( " +
MonthName(Month(Date)) + " ) "
' Not Diffrence With ====> LblMonth.Caption
= Month(Now) & " ( " +
MonthName(Month(Date)) + " ) "

LblYear.Caption = Year(Date)
' Not Diffrence With ====> LblDay.Caption =
Year(Now)



LblHour.Caption = Hour(Time)
' Not Diffrence With ====> LblHour.Caption
= Hour(Now)
LblMinute.Caption = Minute(Time)
' Not Diffrence With ====>
LblMinute.Caption = Minute(Now)
LblSecond.Caption = Second(Time)
' Not Diffrence With ====>
```

```
LblSecond.Caption = Second(Now)

End Sub
```



خوب حالا لازمه که در مورد توابع به کار رفته در مثال بالا توضیحات بدم :

۱. تابع **Format** : در بالا به این تابع اشاره شد .

۲. تایع **Day** : این تابع بخش روز از یک تاریخ را بر می گرداند و بصورت مقابل به کار می رود :

Day(Now) کار میرود و یا به اینصورت نیز به **Day(Date)**

۳. تابع **Month** : این تابع بخش ماه را از یک تاریخ بر می گرداند و بصورت مقابل به کار می رود :

Month(Date) کار میرود و یا به اینصورت نیز به **Month(Now)**

۴. تابع **Year** : این تابع بخش سال از یک تاریخ را بر می گرداند و به صورت مقابل به کار میرود :

Year(Now) کار میرود و یا به اینصورت نیز به **Year(Date)**

۵ . تابع **MonthName** : این تابع بخش نام ماه را از یک تاریخ بر می گرداند و بصورت مقابل به کار می رود :

MonthName(Date) به کار میرود و یا به اینصورت نیز
MonthName(Now)

۶. تابع **Hour** : این تابع بخش ساعت را از یک زمان بر می گرداند و بصورت مقابل به کار می رود :

Hour(Now) کار میرود و یا به اینصورت نیز به **Hour(Time)**

۷. تابع **Minute** : این تابع بخش دقیقه را از یک زمان بر می گرداند و بصورت مقابل به کار می رود :

Minute(Time) کار میرود و یا به اینصورت نیز به
Minute(Now)

۸ . تابع **Second** : این تابع بخش ثانیه را از یک زمان بر می گرداند و بصورت مقابل به کار می رود :

Second(Time) کار میرود و یا به اینصورت نیز به
Second(Now)

:: در اینجا می خواهم چندتا تابع دیگه برای کار با تاریخ و زمان بهتون بگم چون فکر مکنم که بکارتون میاد :

تابع **DataPart** : این تابع بخش خاصی از یک تاریخ را بر می گرداند و بصورت زیر می باشد (اگر یادتون باشه قبلا گفته بودم که مواردی که در دستور العمل استفاده از تابع در داخل [ ] قرار دارند استفاده از آنها اجباری نیست) :

DatePart(*interval , Date*)

مواردی که می توان برای پارامتر **Interval** از آنها استفاده کرد به قرار زیر است :

مقادیر پارامتر interval	
Interval	**Description**
"d"	روز
"y"	روز از سال
"m"	ماه
"q"	فصلی از سال
"yyyy"	سال
"ww"	هفته
"w"	روز هفته
"s"	ثانیه
"n"	دقیقه
"h"	ساعت

فکر کنم با این مثال حساب کار دستون بیاد (اگر تاریخ جاری ۲۰/۱۲/۲۰۰۴ باشد) :

DatePart("d" , now) ==========> 20

توابع TimeSerial و DateSerial : این توابع به ترتیب سه عدد صحیح را به نوع زمان و تاریخ تبدیل می کنند و شکل کلی آنها بصورت زیر می باشد :

TimeSerial (*Hour , Minute , Second*)

DateSerial (*Year , Month , Day*)

فکر کنم دیگر نیازی به تو ضیح بشتر نباشه .

تابع DateDiff : این تابع اختلاف بین دو تاریخ را بر می گرداند و بصورت زیر استفاده می شود :

DateDiff(*Interval , Date1 , Date2*)

لازم به ذکر است که مقادیری که پارامتر Interval می پذیرد همان مقادیری است که در جدول بالا برای تابع DatePart به آنها اشاره شد .
در زیر با یک مثال کار با این تایع را به پایان می بریم :

DateDiff (*"d" , #20/11/2004# , #20/12/2004#*)
======Output=====> 30

تابع CVDate : این تابع یک رشته را به تاریخ تبدیل می کند .

استفاده کاربردی از تابع Instr

سلام دوستان , امروز می خوام یه مطلب کاربردی تو **VB** رو براتون بگم که شاید خیلی به کارتون بیاد این کد شاید براتون تو برنامه هایی که باید از ورودی مقدار خاصی رو بگیرید مثلا می خواهید از ورودی فقط یک مقدار عددی رو بگیرید و بعد اون رو تو محاسبات استفاده کنید و اگر کاربر مثلا در ورودی ۷۸۸در۱۲ را تایپ کنه برنامه **Error** میده خوب برای رفع این مشکل می شه از ورودی فقط عدد گفت یعنی در صورتی که کاربر فقط اعداد ۰ تا ۹ رو تایپ کنه در ورودی نمایش داده می شه این هم حلال این مشکل .

خوب این کد رو بهتر که در رویداد **Keypress** کنترل مورد نظر خود (از جمله **Textbox** , **Rich Inputbox , Textbox** و ...) بنویسید که به محض فشرده شدن هر کلید از سوی کاربر این قسمت چک می شود .

```
Private Sub TextBox_KeyPress(KeyAscii As
Integer)
'========================================
===
   Dim StrValid As String
   StrValid = "0123456789"
   If InStr(StrValid, Chr(KeyAscii)) = 0
Then
      KeyAscii = 0
   End If
End Sub
```

بد نیست یه اشاره ای هم به توابع به کار رفته تو این کنم :

۱. تابع **Instr** : یک زیررشته را در یک رشته دیگر جستجو میکند و موقعیت آنرا در رشته بر می گرداند شکل کلی این تابع به صورت زیر است :

InStr([*start*,]*string1*, *string2*[, *compare*])

قسمت	توضیحات
start	مکانی از رشته که جستجو باید از آنجا آغاز شود
string1	رشته اول
string2	رشته دوم
compare	مشخص کردن نوع مقایسه که بین دو رشته انجام میشود که خود بر سه نوع **vbBinaryCompare** و **vbDatabaseCompare** و **vbTextCompare** می باشد

به این مثال توجه کنید :

```
instr("visual basic Language","Language")
==============> عدد ۱۴ برگردانده میشود
```

:: یک نکته و آن اینکه موا**V**دی که در شکل کلی این تابع در داخل [ ] قرار دارند اختیاری می باشند .

۲. تابع **Chr** : این تابع کاراکتر معادل کد اسکی یک مقدار را بر می گرداند

KeyAscii : آرگومانی است که کد اسکی کلیدی از صفحه کلید را که فشرده شده بر می گرداند

اضافه کردن آیکون به منو

برای اضافه آیکون به منوهای موجود در یک برنامه visual basic بایستی از توابع زیر که موجود در کتابخانه User32 هستند استفاده کنید :

۱ – GetMenu

۲ – GetSubMenu

۳ – GetMenuItemID

۴ – SetMenuIcon

ابتدا یک ماژول ایجاد کنید و توابع فوق را در آن declare کنید :

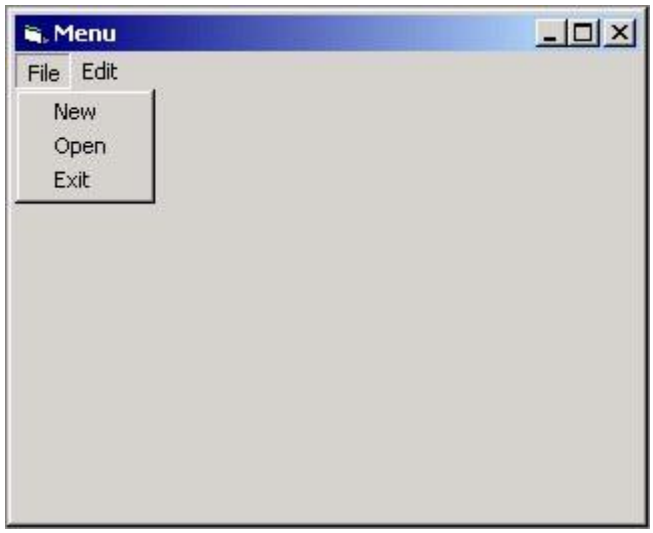
```
ByVal ) "Public Declare Function GetMenu Lib "user32
hwnd As Long) As Long

user32" " Public Declare Function GetSubMenu Lib
(ByVal hMenu As Long, ByVal nPos As Long) As Long

Function GetMenuItemID Lib "user32"  Public Declare
Long (ByVal hMenu As Long, ByVal nPos As Long) As

Public Declare Function SetMenuItemBitmaps Lib
As Long, ByVal nPosition As  "user32" (ByVal hMenu
Long, ByVal wFlags As Long, ByVal hBitmapUnchecked
As Long, ByVal hBitmapChecked As Long) As Long
```

برای قرار دادن یک آیکون در کنار یکی از آیتمهای منو نیاز به handle فرم ، شماره منو ، شماره آیتم مورد نظر و نیز یک picture داریم :

```
,Public Function SetMenuIcon(FrmHwnd As Long
MainMenuNumber As Long, MenuItemNumber As Long,
BitmapUncheckedHandle As Long,  ,Flags As Long
BitmapCheckedHandle As Long)x
Long Dim lngMenu As
Dim lngSubMenu As Long
Dim lngMenuItemID As Long
GetMenu(FrmHwnd)x = lngMenu
,lngSubMenu = GetSubMenu(lngMenu
MainMenuNumber)x
,lngMenuItemID = GetMenuItemID(lngSubMenu
MenuItemNumber)x
SetMenuIcon = SetMenuItemBitmaps(lngMenu, ,
Flags ,lngMenuItemID
BitmapUncheckedHandle, BitmapCheckedHandle)x
End Function
```

image های مورد نظر خود را با ابعادی حدود ۱۶*۱۶ پیکسل و بصورت PictureBox در فرم خود قرار دهید و خاصیت Visible مربوط به PictureBox ها را False کنید .

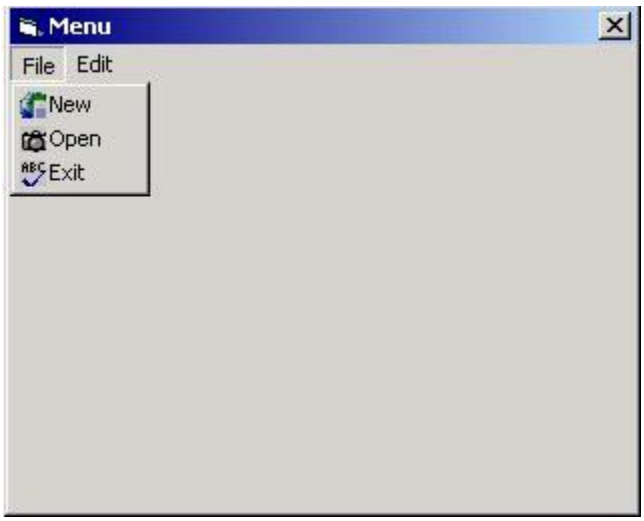
سپس منوهای خود را توسط Menu Editor طراحی کنید . فرض کنید در فرمتان منوهایی بصورت زیر دارید :



File و Edit منوهای اصلی هستند . پارامتر MainMenuNumber در تابع فوق شماره منوی اصلی است که برای File برابر صفر و برای Edit برابر یک می باشد . پارامتر MenuItemNumber شماره هر آیتم در یک منو است که این پارامتر نیز از صفر شروع می شود .

اکنون برای اضافه کردن سه آیکون به سه آیتم منوی File کدهای زیر را در Form_Load بنویسید :

```
Private Sub Form_Load()x
pic1.Picture, pic1.Picture ,۰ ,۰ ,۰ ,SetMenuIcon Me.hwnd
pic2.Picture, pic2.Picture ,SetMenuIcon Me.hwnd, 0, 1, 0
pic3.Picture ,SetMenuIcon Me.hwnd, 0, 2, 0, pic3.Picture
End Sub
```



طراحی فرمهای دلخواه برای برنامه های ویژوال بیسیک

فرض کنید یک image غیر مستطیلی دارید که می خواهید از آن بعنوان فرم برنامه تان استفاده کنید

۱ – تصویر فرم مورد نظرتان را طراحی کرده و با فرمت bmp ذخیره کنید . دقت نمایید که بایستی image خود را درون یک کادر مستطیلی قرار دهید که با یک رنگ با RGB مشخص رنگ آمیزی شده است :

۲ – یک فرم ویژوال بیسیک ایجاد کنید و خاصیت BorderStyle آنرا صفر نمایید .

۳ – در متد Form Load بایستی image مورد نظر را به فرمتان assign کنید :

```
Me.picture=loadpicture(yourimagename)x
Me.width=Me.picture.width
Me.height=Me.picture.height
```

۴ – سپس بایستی یک ناحیه از این image بسازید که نسبت به رنگ RGB ای که در بالا به آن اشاره کردم transparent باشد . اگر فرض کنیم این رنگ ، رنگ سیاه باشد (r=0 , g=0, b=0) :

```
LRegion=MakeRgn(yourimagename,0,0,0)x
```

۵ – حال بایستی ناحیه مشخص شده را بعنوان فرم برنامه تان قرار دهید :

```
call SetWindowRgn(Me.hwnd,LRegion,True)x
```

۶ – یک ماژوال ایجاد کنید و خطوط زیر را در آن بنویسید :

```
user32" " Public Declare Function SetWindowRgn Lib
(ByVal hwnd As Long, ByVal hRgn As Long, ByVal
Long bRedraw As Boolean) As
Public Declare Function MakeRgn Lib "Region.dll"
String, ByVal R As Integer, ByVal g  (ByVal FileName As
```

```
Long As Integer, ByVal b As Integer) As
Public Declare Function DeleteRgn Lib "Region.dll"
Long)x (ByVal Region As

Global lRegion As Long
```

۷ – در متد Form Unload عبارت زیر را قرار دهید :

```
Call DeleteRgn(LRegion)x
```

نکته ۱: بوسیله روتین زیر می توانید فرم خود را در وسط صفحه قرار دهید :

```
(Sub CenterForm(frm As Form
Screen.Width - frm.Width) / 2) = frm.Left
۲ / (frm.Top = (Screen.Height - frm.Height
End Sub
```

چگونگی عبور از پروکسی

احتمالاً اخیراً دیده اید که اکثر ISP ها برخی سایتها (از جمله سایت http://i.hoder.com/) را با استفاده از پروکسی بسته اند . یکی از روشهای عبور از پروکسی و دسترسی به سایتها استفاده از سایت Anonymization می باشد . نمونه ای از چگونگی استفاده از این سایت

بصورت زیر است :
http://www.anonymization.net/http://i.hoder.com

اما چگونه می توانیم خودمان با استفاده از ویژوال بیسیک یک Anonymizer بنویسیم ؟
اینکار دارای یکسری مراحل است که چون برخی از آنها را در این وبلاگ هنوز آموزش نداده ام فعلاً فقط کلیات مطلب را بیان می کنم :

۱ – با استفاده از RAS API یک مدخل برای دسترسی به اتصالات Dial Up ایجاد می کنیم تا نرم افزار بتواند با آن به یک سرویس دهنده اینترنت متصل شود .

۲ – مرورگر IE خود را طوری تنظیم می کنیم که بجای استفاده از اتصالات Dial Up درخواستهای خود را به یک پروکسی سرور بفرستد .
(برای اینکار در IE به منوی Tools/Internet Option/Connections/LAN Settings بروید و در آنجا در قسمت Server Proxy آدرس ۱۲۷,۰,۰,۱ و شمار پورت ۱۰۰۰۰ را وارد کنید)

۳ – با استفاده از WinSock Control تقاضاهایی را که به پورت ۱۰۰۰۰ می آیند گرفته و آدرس URL آنها را استخراج کنید . (این تقاضاها بصورت GET url هستند)

۴ – با استفاده از Internet Transfer Control این درخواست را به وب سرور مربوطه بفرستید و نتیجه را که سورس آن صفحه مورد نظر است بگیرید .

۵ – با استفاده از WinSock Control اطلاعات گرفته شده را به پورت ۱۰۰۰۰ بفرستید .

در واقع نرم افزار شما بعنوان یک Proxy Server کار می کند و درخواستهای IE را گرفته و با استفاده از پروتکل HTTP آنها را انجام داده و نتیجه را به IE باز می گرداند . بنابراین شما بایستی دانش کاملی در مورد پروتکل HTTP داشته و همچنین با RAS API آشنایی داشته باشید

قرار دادن آیکون برنامه در کنار ساعت ویندوز ۲

برای قرار دادن آیکون برنامه در system tray ابتدا یک ماژول تعریف کرده و اطلاعات زیر را در آن قرار دهید :

ابتدا تعریف constant های مورد نیاز :

```
H205& = Public Const WM_RBUTTONUP
H200& = Const WM_MOUSEMOVE Global
Global Const NIM_ADD = 0
NIM_DELETE = 2 Global Const
Global Const NIM_MODIFY = 1
۲ = Global Const NIF_ICON
Global Const NIF_MESSAGE = 1
H5& = Global Const ABM_GETTASKBARPOS
```

سپس تعریف یک type با نام RECT برای نشان دادن یک مستطیل :

```
Type RECT
Left As Long
Long Top As
Right As Long
```

```
Bottom As Long
End Type
```

سپس تعریف یک type با نام NOTIFYICONDATA برای توصیف آیکون :

```
Type NOTIFYICONDATA
cbSize As Long
Long hwnd As
uID As Long
uFlags As Long
uCallbackMessage As Long
Long hIcon As
szTip As String * 64
End Type
```

حال تعریف یک type با نام APPBARDATA برای توصیف اطلاعات application bar :

```
Type APPBARDATA
cbSize As Long
Long hwnd As
uCallbackMessage As Long
uEdge As Long
rc As RECT
Long lParam As
End Type
```

دو متغیر را بصورت زیر تعریف می کنیم :

```
Global Notify As NOTIFYICONDATA
BarData As APPBARDATA Global
```

حال نیاز به declare کردن توابع Shell_NotifyIcon و
SHAppBarMessage از کتابخانه shell32 داریم :

```
Private Declare Function Shell_NotifyIcon Lib
shell32.dll" Alias "Shell_NotifyIconA" (ByVal "
NOTIFYICONDATA) As  dwMessage As Long, lpData As
Long
Private Declare Function SHAppBarMessage Lib
shell32.dll" (ByVal dwMessage As Long, pData As "
APPBARDATA) As Long
```

روتین قراردادن آیکون بصورت زیر است :

```
As  Sub AddIcon(Form1 As Form, IconID As Long, Icon
Object, ToolTip As String)x
Dim Result As Long
&۳۶ = BarData.cbSize
,Result = SHAppBarMessage(ABM_GETTASKBARPOS
BarData)x
&Notify.cbSize = 88
Notify.hwnd = Form1.hwnd
IconID = Notify.uID
Notify.uFlags = NIF_ICON Or NIF_MESSAGE Or
NIF_TIP
Notify.uCallbackMessage = WM_MOUSEMOVE
```

```
Icon = Notify.hIcon
Chr$(0)x & Notify.szTip = ToolTip
Shell_NotifyIcon(NIM_ADD, Notify)x = Result
End Sub
```

روتین حذف آیکون بصورا زیر است :

```
(Sub delIcon(IconID As Long
Long Dim Result As
Notify.uID = IconID
(Notify ,Result = Shell_NotifyIcon(NIM_DELETE
End Sub
```

در فرم مورد نظرتان ابتدا یک متغیر از نوع object تعریف کنید :

```
Public IconObject As Object
```

در Form load عبارات زیر را بنویسید :

```
Set IconObject = Form.Icon
IconObject.Handle, IconObject,  ,AddIcon Form
"TrayIcon"x
```

در Form unload عبارات زیر را بنویسید :

```
delIcon IconObject.Handle
Form.Icon.Handle delIcon
```

فرض کنید یک منو با نام popmenu در فرم دارید و می خواهید با کلیک راست روی آیکون برنامه در system tray ، آن منو باز شود . ابتدا visible این منو را false کنید و سپس متد زیر را برای mousemove بنویسید :

```
Shift  ,Private Sub Form_MouseMove(Button As Integer
(As Integer, X As Single, Y As Single
Static Message As Long
X / Screen.TwipsPerPixelX = Message
Select Case Message
:WM_RBUTTONUP Case
Me.PopupMenu Popup
End Select
End Sub
```

امروز یک کلک خیلی جالب براتون میزارم. تنها با دو خط کد ناقابل میتونید جلوه ای رو بوجود بیارید که فکرشم نمی کردید. تصور کنید بتونید یک فرم رو توی یک فرم دیگه جابدید. استفاده های زیادی میشه ازش کرد. مثلا ساخت نوار ابزارهایی مثل اونی که فتوشاپ داره. راجع

بهش فکر کنید. این هم کدش، در ضمن یه مثال عملی رو هم میتونید انتهای این پست دانلود کنید

```
_ ) "Lib "user32 Private Declare Function SetParent
ByVal hWndChild As Long, ByVal hWndNewParent As
Long Long) As

()Private Sub Form_Load
hWnd ,SetParent Form2.hWnd
Form2.Show
End Sub
```

امروز میخوام در ادامه درسای قبلی شیوه ساخت فرمهایی که بصورت اشکال و تصاویر هستن رو بهتون نشون بدم. به سادگی مراحل قبل نیست اما با صرف چند دقیقه وقت میشه این کار رو کرد.

البته من قبلا این کار رو کردم و شما هم میتونید در پایان این پست اون رو دانلود کنید و حالشو ببرید!! بریم سراصل مطلب.

اول اجازه بدید شیوه کلی عملکرد این کد رو براتون توضیح بدم. یادتون هست که یا استفاده از تابع **CombineRgn** میتونستیم دو **Region** رو با متدهای مختلفی با هم مخلوط کنیم. مثلا میتونستیم **Region** ی بدست بیاریم که حاصل از تفاضل یا جمع دو **Region** مختلف باشه. حالا توی این روش هم از همین قابلیت استفاده میکنیم. روش کار به این صورته که ابتدا یک عکس خاص رو به فرمون اختصاص میدیم. این همون عکسیه که این فرم در نهایت به شکل اون درمیاد. یعنی ما یک رنگ خاص و

برجسته رو مثلا رنگ بنفش **rgb(255,0,255)** رو در عکسمون پیدامیکنیم، سپس به ازای هر پیکسلی از عکس که معادل اون رنگه فرممون رو در اون قسمت شفاف میکنیم. حالا چطوری میگم. ابتدا یک ناحیه به اندازه کل مساحت فعلی فرممون درست میکنیم. حالا توی یک حلقه تک تک نقاط فرم رو چک میکنیم، در صورتی که اون نقطه به رنگ مورد نظر ما برای شفاف شدن باشه، یک **Region** مربعی شکل به طول و عرض یک پیکسل توی اون نقطه تشکیل میدم و با استفاده از متد **RGN_XOR** این **Region** یا بهتر بگم نقطه ایجاد شده رو از **Region** ی که قبلا به اندازه کل فرم ایجاد کردیم کم میکنیم. ببینید.

```
lWidth = ScaleWidth
lHeight = ScaleHeight
m_hRgn = CreateRectRgn(0, 0, lWidth,
lHeight)
For X = 0 To (lWidth - 1)
    For Y = 0 To lHeight - 1
                                    . . . .
```

البته این روش میتونه کمی کند باشده اما میشه کلکی سوار کرد که مقدار قابل توجهی سرعت رو بالا میبره. یعنی لازم نیست ما به ازای هر نقطه ای که داریم یک **Region** تشکیل بدیم. تصور کنید که ما یک فرم به طول ۱۰۰ و عرض ۲۰۰ داریم. به عبارتی در حالت نرمال ما باید ۱۰۰*۲۰۰ یعنی ۲۰۰۰۰ **Region** جداگونه ایجاد و با هم **Merge** کنیم. با یک تکنیک ساده من تونستم این مقدار رو به نزدیک به ۹۹۰۰ **Region** برسونم!!!!!!!!. خیلی تفاوت داره نه!! البته بسته به عکسی که استفاده میکنید این مقدار متغیره. اما به هر حال میتونه بسیار مفید باشه. اجازه بدید در موردش کمی توضیح بدم. فرض کنید شما تعداد ۱۰۰ پیکسل پشت سر هم دارید که همگی همرنگ هستند. به جای اینکه صد **Region** مختلف ایجاد و با

Region اصلی ترکیب کنیم میتونیم یک **Region** مربعی شکل برای کل این پیکسلهای ممتد ایجاد کنیم. نمیدونم تونستم منظورمو برسونم یا نه. اما اگر هم شما نگرفتید خوب مشکل خودتونه!!!!!!!!. من توی کدم دو متغیر بنامهای **bStart** و **yStart** تعریف کردم هر وقت که برنامه به یک پیکسل برنگی که ما تعیین کرده ایم برخورد کرد در صورتی که این متغیر مقدار **False** رو توی خودش داشت متغیر **yStart** رو برابر با مقدار **y** فعلی قرار میده. این متغیر بعدا نقطه شروع رسم **Region** مستطیل شکل ما هست. حالا اگه برنامه به پیکسلی غیر همرنگ با رنگ تعیین شده توسط ما برخورد مشخص میشه که باید **Region** رو همینجا رسم کنه. از مختصات **y** شروع **yStart** به مختصات **y** فعلی. (خودم هم نفهمیدم که چی گفتم). به هر حال خودتون اگه کد رو ببینید خیلی بهتر از اینهمه چرت و پرتیه که من نوشتم!!!!!!!. کد برنامه رو اینجا میتونید ببینید.

```
Private Declare Function CreateRectRgn Lib
"gdi32.dll" ( _
                        ByVal X1 As Long,
ByVal Y1 As Long, _
                        ByVal X2 As Long,
ByVal Y2 As Long) _

             As Long
Private Declare Function SetWindowRgn Lib
"user32.dll" ( _
                        ByVal hWnd As
Long, ByVal hRgn As Long, _
                        ByVal bRedraw As
Boolean) As Long
Private Declare Function CombineRgn Lib
"gdi32.dll" ( _
                        ByVal hDestRgn As
Long, _
                        ByVal hSrcRgn1 As
```

```
Long, _
                              ByVal hSrcRgn2 As
Long, _
                              ByVal nCombineMode
As Long) As Long
Private Declare Function DeleteObject Lib
"gdi32.dll" ( _
                              ByVal hObject As
Long) As Long


Private Const RGN_XOR As Long = 3

Dim m_hRgn As Long

Private Sub Form_Click()

Dim X As Long, Y As Long
Dim lX As Long
Dim yStart As Long
Dim bStart As Boolean
Dim hRgnTmp As Long
Dim lWidth As Long, lHeight As Long

'in rangii hast ke besoorate Transparent
Dar miayad
'Va inja barabar ba range avvalin pixel
rooye form gharar dadim
    lTransColor = Point(0, 0)


    lWidth = ScaleWidth
    lHeight = ScaleHeight
    m_hRgn = CreateRectRgn(0, 0, lWidth,
lHeight)
    For X = 0 To (lWidth - 1)
        For Y = 0 To lHeight - 1
            If Me.Point(X, Y) = lTransColor
```

```
Then
                    If Not bStart Then
                        yStart = Y
                        bStart = True
                    End If
                Else
                    If bStart Then
                        hRgnTmp =
CreateRectRgn(lX, yStart, lX + 1, Y)
                        CombineRgn m_hRgn,
hRgnTmp, m_hRgn, RGN_XOR
                        DeleteObject hRgnTmp
                        bStart = False
                    End If
                End If
            Next Y
            If bStart Then
                hRgnTmp = CreateRectRgn(lX,
yStart, lX + 1, Y)
                CombineRgn m_hRgn, hRgnTmp,
m_hRgn, RGN_XOR
                DeleteObject hRgnTmp
                bStart = False
            End If
            lX = lX + 1
        Next X

        SetWindowRgn hWnd, m_hRgn, True

End Sub
```

نکته ای که باید این میون براتون توضیح بدم در مورد تابع **DeleteObject** هست. تا حالا در موردش صحبت نکرده بودیم. اولا اینکه کاملا اشتباه کرده بودیم!! و دوما اینکه چون میخواستیم زیاد درگیر توابع مختلف نشیم. کار این تابع اینه که **Region** موقتی رو که توی حافظه ایجاد میکنیم از روی حافظه پاک میکنه. تاحالا همیشه یکی دوتا

Region بیشتر تولید نمیکردیم اما حالا که تعدادش سر به فلک میزنه اگه از این تابع استفاده نکنیم. اونوقته که یاور ویندوزتون استاد میشه و به من فحش میدید. اما شاید با خودتون بگید که چرا مثلا از `Set hRgnTmp=Nothing` استفاده نمیکنیم و من هم اینجا میگم که فکرای **plz**.

توی پست بعدی یه تکنیک خیلی ساده اما بسیار مفید و باورنکردنی یادتون میدم. فقط هم با دو خط کد. سر بزنید.

امیدوارم درسای قبلی رو خونده باشید. قبلا دیدیم چسوری میشه پنجره هایی با شکل دلخواه ساخت. اما پا رو هنوز هم از این میشه فراتر گذاشت. امروز میخوایم فرمهایی بسازیم که مثلا به شکل یک نوشته باشند. روشی که امروز بهتون آموزش میدم به شما اجازه میده که فرمهایی با اشکال پیچیده تر بوجود بیارید. توی توابع **API** مفهومی به نام **Path** وجود داره. مجموعه ای از توابع ترسیمی هستند که برای ایجاد **Path**ها بکار میرند. برای ایجاد یک **Path** از دستور **BeginPath** استفاده می کنیم. بعد از توابع معمول **API** مثل:

Rectangle,Ellipse,RoundRect,TextOut و ... برای رسم اشکال مورد نظر استفاده میکنیم. پس از اتمام توابع ترسیمی از دستور **EndPath** برای اعلام پایان ترسیم استفاده میکنیم. حالا میتونیم این **Path** ایجاد شده رو مورد استفاده سایر توابع قرار بدیم. توابعی مثل **StrokePath** که برای کشیدن خط حاشیه یا همون **Border** برای **Path** ایجاد شده بکار میرن و یا **FillPath** برای پرکردن **Path** ایجاد شده از **FillColor** فرم مورد نظر. همچنین تابع **StrokeAndFillPath** که کار این دو تابع رو یکجا انجام میده.

چیزی که مورد نظر ماست درآوردن فرممون به شکل این **Path** هست. البه اینکار بطور مستقیم قابل اجرا نیست بلکه راهی که موجوده اینه که از تابع **PathToRegion** برای تبدیل این **Path** به یک **Region** استفاده کنیم. حالا میتونیم این **Region** رو به فرممون اعمال کنیم.

اینجا کد یک برنامه نمونه که من نوشتم وجود داره. البته شما میتونید به سلیقه خودتون و وقتی که کمی بیشتر با عملکرد این توابع آشنا شدید فرمهایی با اشکال دیگه و یا حتی پیچیده تر ایجاد کنید.

در ضمن من توی این برنامه روش حرکت دادن فرمهایی که عنوان ندارند رو نشون دادم. با چند خط کد ساده میتونید اینکار رو انجام بدید. البته با استفاده از توابع **API** هم میشه اینکار رو انجام داد که در آینده به شما آموزش خواهم داد.

```
Private Declare Function RoundRect Lib
"gdi32" ( _
                ByVal hdc As Long, ByVal X1
As Long, _
                ByVal Y1 As Long, ByVal X2
As Long, _
                ByVal Y2 As Long, ByVal X3
As Long, _
                ByVal Y3 As Long) As Long
Private Declare Function Rectangle Lib
"gdi32" ( _
                ByVal hdc As Long, ByVal X1
As Long, _
                ByVal Y1 As Long, ByVal X2
As Long, _
                ByVal Y2 As Long) As Long
Private Declare Function Ellipse Lib
"gdi32" ( _
                ByVal hdc As Long, ByVal X1
As Long, _
```

```
                    ByVal Y1 As Long, ByVal X2
As Long, _
                    ByVal Y2 As Long) As Long
Private Declare Function LineTo Lib "gdi32"
( _
                    ByVal hdc As Long, ByVal X
As Long, _
                    ByVal Y As Long) As Long

Private Declare Function TextOut Lib
"gdi32" Alias "TextOutA" ( _
                    ByVal hdc As Long, ByVal X
As Long, _
                    ByVal Y As Long, ByVal
lpString As String, _
                    ByVal nCount As Long) As
Long

Private Declare Function SetWindowRgn Lib
"user32" ( _
                    ByVal hWnd As Long, ByVal
hRgn As Long, _
                    ByVal bRedraw As Boolean)
As Long
Private Declare Function PathToRegion Lib
"gdi32" ( _
                    ByVal hdc As Long) As Long

Private Declare Function BeginPath Lib
"gdi32" ( _
                    ByVal hdc As Long) As Long
Private Declare Function EndPath Lib
"gdi32" ( _
                    ByVal hdc As Long) As Long
Private Declare Function StrokeAndFillPath
Lib "gdi32" ( _
                    ByVal hdc As Long) As Long
```

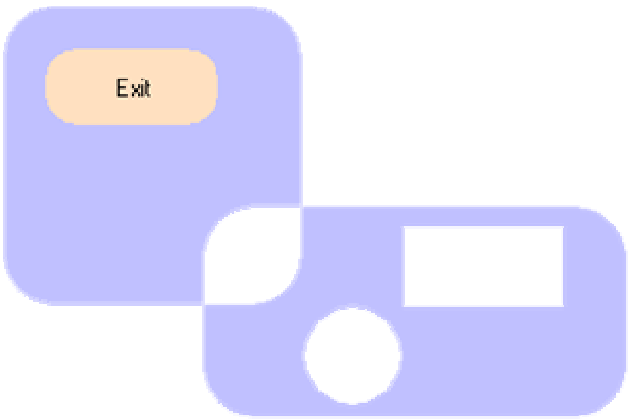
```
Dim m_Drag As Boolean
Dim pX As Long
Dim pY As Long

Private Sub Form_Load()

    Me.ScaleMode = vbPixels

    vba_DrawPath
    StrokeAndFillPath hdc

    vba_DrawPath
    rgn = PathToRegion(hdc)

    SetWindowRgn hWnd, rgn, True

End Sub

Sub vba_DrawPath()

    Me.DrawWidth = 2
    Me.BackColor = vbRed
    Me.FillStyle = vbDiagonalCross
    Me.FillColor = RGB(240, 190, 100)

    Me.Font.Name = "Arial Black"
    Me.Font.Italic = False
    Me.Font.Bold = False

    BeginPath hdc

        Me.Font.Size = 30
        TextOut hdc, 0, 0, "vbadvanced", 10

        Me.Font.Size = 55
        TextOut hdc, 40, 25, "@", 1

        Me.Font.Size = 30
```

```
        TextOut hdc, 100, 35,
"persianblog.com", 15
        RoundRect hdc, 30, 48, 470, 91, 10,
10

        Me.Font.Italic = True
        Me.Font.Size = 10
        TextOut hdc, 100, 90, _
            "R I G H T C L I C K T O C L O
S E M E . . .", 47

    EndPath hdc

End Sub

Private Sub Form_MouseDown(Button As
Integer, _
                        Shift As Integer, X
As Single, Y As Single)
    Select Case Button
        Case vbRightButton
            Unload Me
        Case vbLeftButton
            m_Drag = True
            pX = X
            pY = Y
    End Select
End Sub

Private Sub Form_MouseMove(Button As
Integer, _
                        Shift As Integer, X
As Single, Y As Single)
    If Not m_Drag Then Exit Sub
    Me.Move Me.Left + (X - pX), _
    Me.Top + (Y - pY)
End Sub
```

```
Private Sub Form_MouseUp(Button As Integer,
_
                         Shift As Integer, X
As Single, Y As Single)
    m_Drag = False
End Sub
```

توی درس بعدی نحوه ایجاد فرمهایی که شکل یک عکس رو به خودشون میگیرن رو بهتون یادمیدم. مثلا فرمهایی که به شکل یک سیب هستند. یا هر عکس دیگه ای.

اگه درس قبلی رو خوندید دیدید که چطوری میتونیم فرمهای استاندارد رو از اون حالت مربعی شکل خود در بیاریم. اما از این هم میشه فراتر رفت. یعنی میشه ترکیبی از اشکال مختلف رو هم برای شکلمون بوجود بیاریم. بعنوان مثال ترکیبی از یک مستطیل با لبه های گرد با یک بیضی. به شکل زیر توجه کنید...



جالبه نه!! این کار توسط تابعی با عنوان **CombineRgn** انجام میشه. وظیفه این تابع همونطور که از اسمش پیداست ترکیب کردن دو **Region** با هم هست. این تابع متدهای مختلفی رو برای الحاق دو **Region** با هم

داره. متدهای مثل **XOR,OR,AND,DIFF** و ... این متدها تعیین میکنند که دو **Region** چطوری با هم ترکیب بشن.

```
Declare Function CombineRgn Lib "gdi32" ( _
                    ByVal hDestRgn As Long,
ByVal hSrcRgn1 As Long, _
                    ByVal hSrcRgn2 As Long,
ByVal nCombineMode As Long) As Long
```

این تابع چهار آرگومان مختلف داره. **hDestRgn** که متغیری است که **Region** خروجی تابع در اون ذخیره میشه و hSrcRgn1 و hSrcRgn2 هم اشاره به دو **Region** ی که می خوان با همدیگه ترکیب بشن دارن. چهارمین آرگومان مربوط به نحوه ترکیب دو **Region** میشه. مثلا وقتی که مقدار **RGN_AND** برابر با ۱ رو به این تابع میدیم نتیجه مجموعی از دو **Region** ورودی است. همچنین یک **Region** هم به **Command1** اختصاص داده شده. این به این معنیه که **Region** ها رو فقط به فرمها نمیشه اعمال کرد بلکه این کار رو با هر کنترلی که خاصیت **hWnd** داره میشه انجام داد.

توی درس بعدی که بزودی براتون میزارم میخوام شیوه ایجاد فرمهایی بصورت نوشته و یا اشکال پیچیده تر رو بهتون آموزش بدم. منتظر نظرات و پیشنهاداتتون هستم

مقدمه ای در مورد توابع API

امیدوارم با توابع API آشنایی داشته باشید. اگه ندارید باید یگم که توابع API توابع استاندارد ویندوز هستند که ویندوز کارهای معمول خودش مثل ایجاد و حذف پنجره ها مدیریت پرینتر و کارهای چاپی و بطور کلی همه کارهاشو با فراخوانی اونا انجام میده. این توابع میتونه به شما

قابلیت انجام خیلی کارا رو بده. خیلی کارهایی که با توابع و امکانات معمول **Visual Basic** قابل انجام نیستند. یکی دیگه از مزایای این توابع سرعت بسیار بالای اونا هستند. یعنی در بعضی موارد دهها بار از توابع استاندارد معادل در ویژوال بیسیک سریعتر هستند. لیست کاملی از این توابع رو میتونید با استفاده از برنامه **API Text Viewer** که یکی از برنامه های همراه ویژوال بیسیک هست ببینید.

توابع مورد استفاده در برنامه:

توی این قسمت از دو تابع **API** استفاده میشه.

۱-اولین تابع با عنوان **CreateRoundRectRgn** :

```
Declare Function CreateRoundRectRgn Lib
"gdi32" ( _
                  ByVal X1 As Long, ByVal Y1
As Long, _
                  ByVal X2 As Long, ByVal Y2
As Long, _
                  ByVal X3 As Long, ByVal Y3
As Long)As Long
```

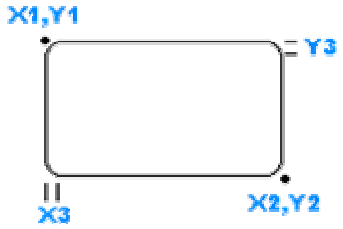
کاری که این تابع و سایر توابع از این دسته انجام میدن (بعدا به چند مورد دیگه اشاره میکنم) اینه که ناحیه ای رو به اشکال مختلف تعریف میکنن که میشه در توابع دیگه از اونا استفاده کرد. این توابع معمولا دارای کلمه **Rgn** هستند. در مورد تابع **CreateRoundRectRgn** این تابع ناحیه ای مستطیلی شکل با لبه های گرد ایجاد میکنه. میزانه گردی لبه ها و اندازه کلی شکل رو شما تعیین میکنید.

آرگومانهای تابع:

X1 و **Y1** مختصات نقطه منتهی الیه بالا و چپ شکل هستند.

X۲ و Y۲ مختصات نقطه منتهی الیه پایین و راست شکل هستند.

X۳ و Y۳ اندازه شعاع گردی لبه های مربع در مختصات X و Y است.



۲- دومین تابع با عنوان SetWindowRgn :

```
Declare Function SetWindowRgn Lib "user32"
( _
                    ByVal hWnd As Long, ByVal
hRgn As Long, _
                    ByVal bRedraw As Boolean)
As Long
```

این تابع Region ی که با استفاده از تابع بالا ایجاد کردیم رو به پنجره مورد نظر ما اختصاص میده. یعنی در حقیقت این پنجره رو به این شکل در میاره. این تابع سه تا آرگومان داره. اولی یعنی hWnd اشاره به پنجره ای دارید که میخواید عملیات رو روی اون انجام بدید. مثلا Form1.hWnd در ویندوز هر فرم با استفاده از یک Handle شناخته میشه و اکثر توابع API از همین خاصیت برای کار بر روی پنجره ها استفاده میکنن. دومین آرگومان hRgn اشاره به Regionی که قبل با استفاده از توابعی مثل تابع بالا ایجاد کردیه ایم داره. سومین آرگومان bRedraw که مقداری Boolean هست میتونه مقادیر True یا False رو قبیل کنه و تعیین میکنه که آیا پنجره بعد از اجرای این تابع دومرتبه از اول رسم یا به عبارتی Refresh بشه یا نه.

حالا که با تئوری و عملکرد این برنامه آشنا شدید اجازه بدید مراحل ساخت اونرو ادامه بدیم.

۱- پروژه فعلی خودتون رو ببندید و یک پروژه جدید ایجاد کنید.

۲- ویژوال بیسیک بطور اتوماتیک **Form1** رو براتون ایجاد میکنه. روی فرم دابل کلیک کنید. پنجره کد این فرم دیده میشه. همه کدها رو پاک کرده و کدهای زیر رو توی اون **Copy** و **Paste** کنید.

```
'************************
Private Declare Function CreateRoundRectRgn
Lib "gdi32" ( _
                    ByVal X1 As Long, ByVal Y1
As Long, _
                    ByVal X2 As Long, ByVal Y2
As Long, _
                    ByVal X3 As Long, ByVal Y3
As Long)As Long

Declare Function SetWindowRgn Lib "user32"
( _
                    ByVal hWnd As Long, ByVal
hRgn As Long, _
                    ByVal bRedraw As Boolean)
As Long

Private Sub Form_Load()
Dim rgn As Long

    Me.ScaleMode = vbPixels

    rgn = CreateRoundRectRgn(0, 0,
ScaleWidth, ScaleHeight, 50, 50)
    SetWindowRgn hWnd, rgn, True

End Sub
```

'**********

کلید F5 رو فشار بدید و نتیجه رو ببینید!!!!!!!!

.

ذخیره فایلهای صوتی real

تا حالا شده یه آهنگ خیلی باحال رو از اینترنت با player real گوش بدید و افسوس بخورید که چرا نمی تونید اونو save کنید ؟
پس بهتره این برنامه رو download کرده و به جون من دعا کنید :

۳,۲ Total Recorder

برای دریافت username و password با من تماس بگیرید

تذکر : مطلب فوق به برنامه نویسی ویژوال بیسیک ربطی نداره ولی اگه بهتون نمی گفتم دق می کردم

ساخت نرم افزارهای فارسی توسط زبان ساز شتاب

+ برخی دوستان در مورد چگونگی ساخت برنامه های فارسی توسط ویژوال بیسیک سوال کرده بودند . یکی از ابزارهایی که می توان نرم افزارهای فارسی را ایجاد نمود زبان ساز شتاب می باشد . مطلب زیر را که در مورد این زبان ساز است از وبلاگ فارس تک برای شما نقل می کنم

" زبان ساز شتاب یک ابزار قدرتمند است که به برنامه نویسان اجازه میدهد نرم افزارهای فارسی خود را در ویندوز فارسی، XP یا ۲۰۰۰ بنویسند و در ویندوزهای انگلیسی نیز اجرا کنند. اگر چه زبان ساز شتاب یک نرم افزار قوی است، اما استفاده از آن بسیار ساده میباشد و هیچ نیازی به نوشتن برنامه و یا دانش برنامه نویسی ندارد. این نرم افزار تنها فارسی سازی است که با استاندارد مایکروسافت کار می کند. همچنین بهترین انتخاب برای فارسی سازی برنامه های تهیه شده با Director Macromedia میباشد و قابلیت ورود اطلاعات و نمایش صحیح را در تمامی ویندوز ها به این نرم افزارها می دهد و مشکلات فارسی آنها را از قبیل جستجو و نمایش حتی در ویندوز های XP و ۲۰۰۰ بر طرف می کند. به دلیل اینکه این نرم افزار از روش استاندارد مایکروسافت استفاده می کند، در مقایسه با فارسی سازهای دیگر (که معمولاً کنترلهای فارسی و یا فونتهای غیر استاندارد را به کار می برند) قابلیتهای مهمی به نرم افزار شما می دهد که به چند مورد از آنها عبارت اند از :

- عدم نیاز به هرگونه برنامه نویسی و کد گزاری حتی یک خط
- قابلیت استفاده از علامتهایی مانند تنوین، تشدید، فتحه، کسره و ...
- امکان اجرای نرم افزار های ترجمه شده در همه ویندوزها
- امکان استفاده در تمامی محیط های برنامه سازی
- امکان جستجو و مرتب سازی استاندارد

- امکان استفاده بدون اشکال از سیستمهای ذخیره سازی (مانند MDB)
- امکان Copy/Paste استاندارد به برنامه های دیگر
- پشتیبانی سایر زبانهای غیر فارسی
- امکان استفاده از فونتهای استاندارد ویندوز
- امکان بهره گیری از سیستم استاندارد ویندوز
- قابلیت فارسی نمودن کلیه کنترلهای ویندوز و هزاران کنترل آماده ای که در اینترنت وجود دارند. "

دانلود نسخه نمایشی (با سایز ۴,۹ مگابایت)

اطلاعات بیشتر

پاسخ به سوالات

پاسخ به سوالات – ۱

۱ – چگونه می توان یک فرم شفاف (Transparent) ساخت بطوریکه اطلاعات پشت فرم مشخص باشد ؟

پاسخ : در دات نت به راحتی می توانید توسط خاصیت Opacity اینکار را انجام دهید . اما در ویژوال بیسیک ۶ برای ساخت یک فرم شفاف به این آدرس مراجعه کنید .

۲ – چگونه می توان برنامه ای برای ارسال ایمیل نوشت ؟

پاسخ : استفاده ازیک کامپوننت به اسم MAPI و یا پیاده سازی پروتکل SMTP در برنامه . برای روش اول به این صفحه و برای روش دوم به این صفحه مراجعه کنید .

۳ – چگونه می توان در سی دی رام را در ویژوال بیسیک باز و بسته کرد .

پاسخ : با استفاده از یک API به اسم mciSendString بصورت زیر :

```
Function mciSendString Lib  Private Declare
_ ""winmm.dll
mciSendStringA" (ByVal lpstrCommand As " Alias
_ ,String
lpstrReturnString As String, ByVal  ByVal
_ ,uReturnLength As Long
hwndCallback As Long) As Long ByVal

()OpenCDDoor Sub
Set CDAudio Door Open Wait", " mciSendString
&۰ ,&۰ ,&0
Sub End

()CloseCDDoor Sub
```

Set CDAudio Door Closed Wait", " mciSendString &۰ ,&۰ ,&0

End Sub

۴ – چگونه می توان از اطلاعات جداول یک بانک اطلاعاتی یک Report تهیه کرده و به کاربر نشان داد ؟

پاسخ : چند روش برای اینکار وجود دارد :

- **استفاده از CrystalReport که ابزاری مجزا برای ساخت گزارش ها از بانک های اطلاعاتی است و در وی بی می توان از آن استفاده نمود .**
- **استفاده از DataReport موجود در خود وی بی (Project/Add Data Report)**

برای اطلاعات بیشتر در این زمنیه به دو کتاب زیر مراجعه کنید :

- مباحث برنامه نویسی پیشرفته در ویژوال بیسیک ۶

- برنامه نویسی بانک های اطلاعاتی در ویژوال بیسیک ۶

هر دو کتاب را انتشارات نص چاپ کرده است .

۵ – چگونگی تایپ فارسی در ویژوال بیسیک ۶ و بطور کلی ساخت برنامه های فارسی در وی بی ۶ ؟

پاسخ : یک راه استفاده از یکسری ActiveX است که توسط برخی شرکتهای ایرانی ایجاد شده . دوستان اگر راه حل دیگری می دانند جواب بدهند .

۶ – چگونه می توان کاری کرد که اگر کاربر در یک کادر متنی (Textbox) یک دستور وی بی (مثلاً msgbox"OK") نوشت ، این دستور اجرا شود ؟

پاسخ : من راه حلی به ذهنم نرسید . دوستان اگر راه حلی دارند لطفاً جواب بدهند .

۷ – چگونه می توان یک **CommandButton** ساخت که یک تصویر در سمت چپ و یک متن در سمت راست آن وجود داشته باشد ؟

پاسخ : بنظر من بهتر است دکمه **CommandButton** را بطور **Custom** خودمان ایجاد کنیم . بدین منظور مراحل زیر را دنبال کنید :

– دو تصویر برای دکمه مورد نظر خود طراحی کنید : یکی برای حالت نرمال و یکی برای حالتی که روی دکمه کلیک می کنید و دکمه فرو می رود . این کار را می توانید توسط هر نرم افزار گرافیکی انجام دهید .

– یک **PictureBox** بر روی فرم خود قرار دهید و نام آنرا **PicButton** بگذارید .

– کد زیر را در **Form_Load** قرار دهید :

```
PicButton.Picture = LoadPicture(normal_file)
```

که **normal_file** نام و مسیر فایل حالت نرمال می باشد .

– کد زیر را برای **PicButton_MouseDown** بنویسید :

```
PicButton.Picture =
LoadPicture(mousedown_file)
  PicButtonRefresh
```

که **mousedown_file** نام و مسیر فایل حالت کلیک می باشد .

– کد زیر را برای **PicButton_MouseUp** بنویسید :

```
PicButton.Picture = LoadPicture(normal_file)
PicButtonRefresh
```

۸ – چگونه می توان برای دکمه مربوط به **DataGrid** منوی **DropDown** ایجاد کرد ؟

پاسخ : من راه حلی به ذهنم نرسید . دوستان اگر راه حلی دارند لطفاً جواب بدهند .

۹ – چگونگی ساخت **WebBrowser** در وی بی ؟

پاسخ : به این آدرس مراجعه کنید .

۱۰ – چگونگی ساخت یک **MailServer** با وی بی ؟

پاسخ : به این آدرس مراجعه کنید .

۱۱ – چگونگی طراحی سخت افزار با آی سی های **8255 & 8254 & 8212** و برنامه نویسی آنها ؟

پاسخ : به جلد دوم کتاب ۸۰ X86 IBM PC and Compatible Computers نوشته آقای علی مزیدی مراجعه کنید . هم خود کتاب و هم ترجمه آن در بازار موجود می باشد .

۱۲ - چگونه می توان فرمهای به شکل یک متن طراحی کرد که در هنگام نمایش فقط نوشته نمایش یابد و از لابلای آن دسکتاب دیده شود ؟

پاسخ : به مقاله ای که قبلاً در این وبلاگ در مورد طراحی فرمهای **Custom** نوشته ام مراجعه کنید . لینک به مقاله

پاسخ به سوالات-۲

۱ – زمانیکه از کنترل Winsock در وی بی استفاده می کنم خطای رجیستر نبودن می گیرم ؟

پاسخ : سرویس پک ۵ مربوط به ویژوال استدیو ۶ را از این آدرس دانلود کرده و نصب کنید تا نسخه کنترل Winsock شما ارتقا یابد . در صورتیکه مشکلتان حل نشد می توانید از WinSock API نیز استفاده کنید . این API را بعداً شرح خواهم داد اما فعلاً از این آدرس نیز می توانید کمک بگیرید .

۲ – چگونه می توان تصاویر را در بانک اطلاعاتی ذخیره کرد ؟

پاسخ : برای اینکار اولاً بایستی فیلدی را در جدول بانک اطلاعاتی خود از نوع باینری ایجاد کنید . سپس بایستی فایل مورد نظر خود را باز کرده و آنرا درون یک آرایه از نوع بایت کپی کنید . حال می توانید این آرایه را بعنوان مقدار فیلد در جدول مورد نظر قرار دهید :

روش دیگر استفاده از ADO Stream است .

در این آدرس و این آدرس و این آدرس می توانید اطلاعات بیشتر همراه با برنامه های نمونه بدست آورید .

۳ – آیا می توان از Microsoft Form 2 Control بدون نصب برنامه Word استفاده کرد ؟

پاسخ : ایندو تا ربطی بهم ندارند .

۴ – چگونه می توان برنامه ای شبیه Yahoo Messenger نوشت ؟

پاسخ : برای اینکار شما بایستی با مبحث Network Programming

با استفاده از Winsock آشنا باشید . برای اطلاع بیشتر به آرشیو موضوعی وبلاگ مراجعه کنید .

۵ – نحوه کار با کامپوننتهای OXC را در وی بی توضیح دهید .
پاسخ : اگر می خواهید خودتان کنترل OCX بنویسید می توانید از کتاب Visual Basic 6 Internet Programming in که انتشارات نص نیز آنرا ترجمه کرده کمک بگیرید . اما اگر می خواهید از کنترلهای OCX موجود در خود وی بی استفاده کنید از منوی Project وارد Component شده و کامپوننت مورد نظرتان را انتخاب کنید تا به نوار ابزار اضافه شود . توضیحات هر کامپوننت را می توانید در سایت MSDN یا سی دی های MSDN بدست آورید .

۶ – چگونه می توان دو دستور را بطور همزمان در وی بی اجرا کرد ؟
پاسخ : برای اینکار بایستی از امکان Threading ویندوز استفاده کنید . برای این منظور تعدادی API وجود دارد که می توانید از آنها کمک بگیرید . برای اطلاعات بیشتر به نرم افزار API Guide و بخش Thread آن مراجعه کنید .

۷ – تفاوت Module و Class Module را در وی بی توضیح دهید .
پاسخ : Class Module برای نوشتن کلاس (برنامه نویسی شی گرا OOP) در وی بی استفاده می شود . البته وی بی ۶ تمام امکانات برنامه نویسی شی گرا را ندارد مثلاً بطور کامل از وراثت پشتیبانی نمی کند یا امکان ایجاد کلاسهای Abstract را ندارد و نیز دارای امکان تعریف متدها و متغیرهای اشتراکی نیست . در صورتیکه بخواهید از تمام امکانات شی گرایی استفاده کنید سراغ VB.NET بروید .
اما Module برای تعریف یکسری ثابت ، متغیر ، تابع و یا روتین بکار

می رود که در تمام پروژه استفاده می شوند . برای مثال Declare های توابع API را در ماژول قرار می دهیم .

۸ – من با کدی که قبلاً برای باز و بسته کردن در سی دی رام گفته بودید مشکل دارم .

پاسخ : ابتدا یک ماژول به پروژه تان اضافه کنید و خط زیر را در آن قرار دهید :

```
Public Declare Function mciSendString Lib “winmm.dll" Alias "mciSendStringA” (ByVal lpstrCommand As String,ByVal lpstrReturnString As String, ByVal uReturnLength As Long, ByVal hwndCallback As Long) As Long
```

حال می توانید توابع OpenCDDoor و CloseCDDoor را در کد فرمتان قرار داده و آنها را صدا بزنید :

```
Sub OpenCDDoor()

  mciSendString “Set CDAudio Door Open Wait”, 0&, 0&, 0&

End Sub


Sub CloseCDDoor()

  mciSendString “Set CDAudio Door Closed Wait”, 0&, 0&, 0&

End Sub
```

۹ – چگونه می توان در وی بی فرمهای غیر مستطیلی به شکلهای زیبا درست کرد ؟

پاسخ : این مطلب قبلاً در این وبلاگ توضیح داده شده است . به اینجا مراجعه کنید .

۱۰ – چگونه می توان کاری کرد که در حین اجرای برنامه کاربر بتواند دستوری را در یک TextBox تایپ کند و با زدن یک دکمه آن دستور اجرا شود ؟

پاسخ : برای اینکار بایستی از Microsoft Script Control استفاده کنید . ابتدا وارد بخش References شده و این مورد را به پروژه تان اضافه کنید . حال می توانید از کد زیر استفاده کنید :

```
Dim script As ScriptControl
Set script = New ScriptControl
script.Language = "VBSCript"
script.ExecuteStatement (Text1.Text)
```

۱۱ – چگونه می توان از Microsoft Speech Library در وی بی استفاده کرد ؟

پاسخ : در آینده این مطلب را توضیح خواهم داد . اما فعلاً می توانید از این آدرس و این آدرس و این آدرس استفاده کنید .

۱۲ – چگونه می توان ActiveX های ایجاد شده توسط وی بی را در وب استفاده کرد ؟

پاسخ : برای اینکار ابتدا پروژه ای از نوع ActiveX Document ایجاد کنید . کد مورد نظرتان را در پروژه بنویسید و سپس پروژه را کامپایل کنید . نهایتاً توسط ابزاری به اسم Deployment & Package Wizard که در ویژوال استدیو موجود است می توانید پروژه خود را قابل استفاده در وب کنید . برای اطلاعات بیشتر به از کتاب Internet Programming in Visual Basic 6 که انتشارات نص نیز آنرا ترجمه کرده مراجعه کنید .

پاسخ به سوالات-۳

– چگونه می توان در ویژوال بیسیک برنامه ای نوشت که بتوان با آن رجیستری را دستکاری کرد ؟

پاسخ : به آرشیو موضوعی وبلاگ بخش کار با رجیستری مراجعه کنید .

۲ – چگونه می توان فایلهای صوتی را به MID تبدیل کرد ؟

پاسخ : اگر منظورتان در وی بی است که باید دنبال یک OCX یا ActiveX مناسب برای اینکار بگردید . در غیراینصورت باید از نرم افزارهای تبدیل کننده فایلهای صوتی استفاده کنید .

۳ – چگونه می توان در بانک های اطلاعاتی از تاریخ شمسی استفاده کرد ؟

پاسخ : تاریخ را بصورت String در بانک اطلاهاتی ذخیره کنید و در اینصورت بایستی خودتان روتینهای مربوط به کار با تاریخ را بنویسید .

۴ – اگه ما بخواهیم یه HTML بوسیله DHTML Application در ویژوال بیسیک بسازیم هنگام ساختن فایل hTML یه فایل DLL دیگر هم Save میشود که در صورت نبود آن ، برنامه ما کار نمیکد و اگر هم بوسیله ویژوال بیسیک کاری کنیم که همه چیزها فقط در فایل html ذخیره بشود کد هایی که در قسمت کد نوشتیم در فایل ما ذخیره نمیشود به همین خاطر ما نمی توانیم از html ی که در ویژوال بیسیک ساختیم در یک سرور وب استفاده کنیم چون html موجود در سرور ، دیگر به dll ای که ما در کامپیوتر خودمان ساختیم دسرسی ندارد و اگه ما dll را جابجا کنیم هم دیگر html کار نمیکند . چگونه می توان این مشکل را حل کرد ؟

پاسخ : این فایل DLL ، در واقع کلاسها و کدهای کامپایل شده مورد نیاز در صفحات DHTML شما می باشد . شما بایستی این فایل dll را در کنار صفحات html تان در سرور مربوطه قرار دهید . همچنین این فایل dll بایستی در سرور مربوطه رجیستر شود .

۵ – – من با وی بی کار می کنم و تازه سر برنامه نویسی پورتها رفته ام . من دیتا(&H378) وآدرس(مثلا۲۵۵) را می خواست به من گفتن باید مبنا ها را یاد بگیری من آنهارا یاد گرفتم (مثلا مبنای ۲، ۱۲۸) اما چه ربطی داشت؟ من الان می خواهم با وی بی وسایل را کنترل کنم اما شماره پورت و طریقه روشن نگه داشتن چند تای آنها باهم در یک دیتا مانند مثال دیتا بالایی را نمی دانم .

پاسخ : برای آشنایی با مبناها ، شماره پورتها و برنامه نویسی آنها به کتاب PC۸۰ IBM X86 Compatible Computers and که هم خود آن و هم ترجمه اش در بازار ایران موجود است مراجعه کنید . برای آشنایی با چگونکی برنامه نویسی سخت افزار در ویندوز و نیز برنامه نویسی پورت پرینتر به آرشیو موضوعی وبلاگ مراجعه کنید .

۶ – چه جوری میشه از داخل برنامه ای که توسط وی بی نوشته ام با فشار بر روی یک CommandButton یا از طریق منویی که ساخته ام بر نامه ای دیگر مثلا نصب یک برنامه یا ماشین حساب ویندوز را اجرا کرد؟

پاسخ : با استفاده از یک API به اسم ShellExecute . فرمت کلی استفاده از این تابع بصورت زیر است :

```
Const SW_SHOW = 1
Const SW_SHOWMAXIMIZED = 3

Public Declare Function ShellExecute Lib "Shell32.dll"
Alias "ShellExecuteA" _
  (ByVal hwnd As Long, _
   ByVal lpOperation As String, _
   ByVal lpFile As String, _
   ByVal lpParameters As String, _
   ByVal lpDirectory As String, _
   ByVal nShowCmd As Long) As Long

Sub RunYourProgram()
  Dim RetVal As Long
  On Error Resume Next
  RetVal = ShellExecute(0, "open", "<full path to
program>", "<arguments>", "<run in folder>",
SW_SHOWMAXIMIZED)
End Sub
```

۷ – چگونه می توان از پورت پرینتر یکسری اطلاعات بصورت ورودی گرفت ؟

پاسخ : به آرشیو موضوعی و بخش برنامه نویسی پورت موازی مراجعه کنید .

۸ – چگونه می توان فلاپی دیسک یا یکی از درایوهای هارددیسک را توسط ویب بی فرمت کرد ؟

پاسخ : ابتدا دو API زیر را در برنامه تان Declare کنید :

```
Private Declare Function SHFormatDrive Lib "shell32" _
   (ByVal hwnd As Long, ByVal Drive As Long, ByVal
fmtID As Long, _
   ByVal options As Long) As Long


Private Declare Function GetDriveType Lib "kernel32"
Alias _
   "GetDriveTypeA" (ByVal nDrive As String) As Long
```

سپس کد زیر را برای فرمت درایو بنویسید :

```
   Dim DriveLetter$, DriveNumber&, DriveType&
   Dim RetVal&, RetFromMsg%
   DriveLetter = UCase(Drive1.Drive)
   DriveNumber = (Asc(DriveLetter) - 65)
   DriveType = GetDriveType(DriveLetter)

   If DriveType = 2 Then
      RetVal = SHFormatDrive(Me.hwnd, DriveNumber,
0&, 0&)
   Else
RetFromMsg = MsgBox("This drive is Not a removeable"
& vbCrLf & _
      "drive! Format this drive?", 276, "SHFormatDrive
Example")


      Select Case RetFromMsg
         Case 6'Yes
         RetVal = SHFormatDrive(Me.hwnd, DriveNumber,
0&, 0&)
         Case 7'No
```

```
    ' Do nothing
  End Select

End If
```

۹ – چگونه می توان به یک IP اطلاعات فرستاد ؟

پاسخ : می توانید از کنترل WinSock استفاده کنید . برای اطلاعات بیشتر به آرشیو موضوعی مراجعه کنید .

۱۰ – چگونه می توان ipMACHINENAME هر سیستمی را پیدا کرد ؟

پاسخ : می توانید از کنترل WinSock استفاده کنید . برای اطلاعات بیشتر به آرشیو موضوعی مراجعه کنید .

۱۱ – چطور میشه یه برنامه رو تو بک گرا ند اجرا کرد در حالی که آیکون برنامه در سینی سیستم باشه. مثلا مثل آنتی ویروس ها ؟

پاسخ : به آرشیو موضوعی وبلاگ و بخش " قرار دادن آیکون برنامه در کنار ساعت ویندوز " مراجعه کنید .

۱۲ – در مورد تابع DoEvents توضیح دهید .

پاسخ : این تابع به سیستم عامل اجازه می دهد سایر event ها را پردازش کند . سینتکس این تابع بصورت زیر است :

```
DoEvents( )
```

این تابع زمانیکه بخواهیم به کاربر اجازه دهیم یک فرایند را در وسط کارش کنسل کند مفید می باشد . برای مثال فرض کنید برنامه ای نوشته اید که با زدن یک دکمه اطلاعاتی را در یک فایل می نویسد و با زدن دکمه دیگر از برنامه خارج می شوید . اگر شما دکمه بستن برنامه را

در وسط کار نوشتن اطلاعات در فیل فشار دهید برنامه تا زمانیکه کار نوشتن انجام نشود بسته نخواهد شد اما اگر دستور DoEvents را جایی در وسط کد مربوط به دکمه نوشتن قرار دهید اجرا هر زمان به آنجا برسد سیستم event مربوط به کلیک شدن دکمه close را اجرا می کند .

۱۳ – با ویژوال بیسیک چگونه می توان با پورت سریال موازی و یواس بی ارتباط برقرار کرد ؟
پاسخ : برای اینکار ابتدا باید شماره پورت آنها را بدانید و سپس با استفاده از dll ای که در بخش برنامه نوسی سخت افزار در آرشیو موضوعی آورده شده اطلاعات را به این شماره پورتها بفرستید یا از آنها بخوانید .

۱۴ – چگونه می توانم توسط وی بی دایرکتوری ایجاد کنم یا فایل کپی کنم؟
پاسخ : به آرشیو موضوعی وبلاگ بخش کار با فایلها مراجعه کنید .

پاسخ به سوالات-۴

– نحوه دسترسی به پورت سریال را بیان کنید .
پاسخ : با چندین روش می توان به پورت سریال در ویژوال بیسیک دسترسی داشت :

– استفاده از کنترل MSCOMM موجود در خود ویژوال بیسیک . برای استفاده از این کنترل وارد منوی Project شده و از بخش Component مورد Microsoft Comm Control را انتخاب کنید . سپس می توانید تنظیمات مورد نظر خود را از قبل bitrate ، وجود parity و تعداد stop bit و غیره انجام دهید . پارامترهایی مانند RThreshold، InputMode و Settings مربوط به این کنترل را می

توانید تنظیم کنید .
برای اطلاعات بیشتر به این آدرس مراجعه کنید .

- استفاده از کنترلهای پیشرفته تر : با جستجو در اینترنت می توانید کنترلهای پیشرفته تری پیدا کنید که البته اکثر آنها رایگان نیستند . مثل این آدرس یا این آدرس

- دسترسی مستقیم به پورت سریال : برای اطلاعات بیشتر به کتاب زیر مراجعه کنید :

Interfacing of & Design : Compatible Computers & X86 IBM PC۸۰
Compatible Computers, Volume II & IBM PC, PS
By Muhammad Ali Mazidi, Janice Mazidi

در این آدرس و این آدرس در مورد چگونگی دسترسی به پورت سریال در .NET توضیح داده شده است .

این آدرس دارای لینکهای بسیار خوبی در زمینه مفاهیم پورت سریال ، کاربردهای آن ، نحوه برنامه نویسی آن و معرفی کتابهای مرتبط می باشد .

۲ – چگونه می توان با پورت USB در ویژوال بیسیک کار کرد ؟
پاسخ : به این آدرس و این آدرس مراجعه کنید .

۳ - چگونه می توان بین ویژوال بیسیک و اکسس ارتباط برقرار کرد .
پاسخ : به کتاب برنامه نویسی بانکهای اطلاعاتی در ویژوال بیسیک ۶ چاپ شده توسط انتشارات نص مراجعه کنید .
در ضمن قبلاً در همین وبلاگ بطور خلاصه ، کار با بانکهای اطلاعاتی با استفاده از ADO را توضیح داده ام .

۴ – دوست عزیزی سوالهای زیر را در مورد Reporting در ویژوال بیسیک پرسیده اند :
- چگونه می توان در زمان اجرا در Data Environment ، Command جدید

ساخت ؟

- چگونه می توان در Data Report ، گزارشهای دینامیک ساخت ؟

- چگونه می توان Data Source یک Data Report را بدون استفاده از Data Environment تنظیم کرد ؟

پاسخ : : به کتاب برنامه نویسی بانکهای اطلاعاتی در ویژوال بیسیک ۶ چاپ شده توسط انتشارات نص مراجعه کنید . در ضمن دوستان اگر نظری دارند بدهند .

۵ – چگونه می توان کامپوننتهایی در ویژوال بیسیک را که خاصیت رنگ ندارند مثل یک scrollbar را تغییر رنگ داد . البته اینکار با استفاده از API های setsyscolor و getsyscolor امکان پذیر است اما این API ها باعث تغییر رنگ کل scrollbar ها می شوند که این درست نیست ؟

پاسخ : دوستان اگر نظری دارند بدهند .

۶ – فایلهای RES چه فایلهایی هستند ؟

پاسخ : فایلهای RES فایلهای منبع یا resource می باشند و همانطور که از نامشان پیداست برای ذخیره برخی منابع برنامه مثل آیکون ، کرسر و یا برخی تصاویر بیت مپ مورد استفاده در برنامه که نمی خواهیم فایل آنها بطور جداگانه در کنار برنامه باشد استفاده می شوند .

برای ایجاد فایلهای res در ویژوال بیسیک ابتدا از منوی Add-Ins مورد Add-In Manager را انتخاب کرده و در آنجا VB6 Resource Editor را انتخاب کرده و گزینه Loaded آنرا علامت بزنید و OKکنید . حال می توان با استفاده از این برنامه Res های مورد نظر خود را ایجاد کرده و منابعتان را به آن اضافه کنید . پس از ایجاد فایل Res ، با استفاده از منوی Project و انتخاب Add File می توانید Res مورد نظر را به پروژه اضافه نموده و از منابع موجود در آن استفاده کنید .

برای مثال اگر آیکونی به اسم MyImage در فایل Res قرار داده باشید با دستور زیر می توانید آنرا در PictureBox مورد نظرتان Load کنید :

```
Picture1.Picture=LoadResPicture("MyImage",vbResIcon)
```

برای اطلاعات بیشتر به این آدرس مراجعه کنید .

۷ – در مورد نحوه استفاده از تابع WNetEnumCachedPasswords در ویژوال بیسیک توضیح دهید .
پاسخ : به این آدرس مراجعه کنید .

۸ – چگونه می توان یک dll نوشته شده توسط C++ را به refrence های ویژوال بیسیک اضافه کرد ؟
پاسخ : بایستی علاوه بر آن dll ، فایلهای منبعی را که آن dll به آنها نیاز دارد را در کنار آن قرار دهید . توسط ابزارهای Dependency Checker مانند نرم افزار PE Explorer می توانید نام این dll ها استخراج کنید .
در مورد نرم افزار PE Explorer بزودی توضیح خواهم داد .

۹ – لطفا سایتی برای دانلود کردن ebook های انگلیسی به صورت مجانی معرفی کنید که در زمینه برنامه نویسی باشند ؟
پاسخ : در سایتهای FTP می توانید بدنبال EBook های مورد نظرتان بگردید .
سایت
http://persianblog.com/pbe34r2/www.ftpsearchengines.com آرشیو بسیار خوبی از لیست سایتهای FTP در خود دارد .

۱۰ – در مورد خاصیت راست به چپ که بعد از کامپایل شدن برنامه بر روی سیستم بدون ویژوال جواب نمی‌دهد ، راهنمایی ام کنید .
پاسخ : دوستان اگر نظری دارند بدهند . در ضمن مشکل خود را واضحتر بیان کنید .

۱۱ – در VB.Net چطور میتونم فایل exe بسازم؟آیا فایل Exe ی من همونیه که در شاخه/Bin جاییه که پروژه رو ذخیره کردم؟اگه همونه من میتونم اونو بدون نیاز به اکتیو ایکس دیگه ای به کسی بدم(اگه از اکتیوایکس خاصی در برنامم استفاده نکرده باشم)
پاسخ : فایل EXE مورد نظر شما در زمان کامپایل کردن پروژه در دایرکتوری Bin

ساخته می شود اما برای Publish کردن برنامه تان بایستی سایر فایلهای منبع برنامه به آن افزوده شده و یک فایل Setup ساخته شوند . برای اینکار شما بایستی از ابزارهای ساخت Setup استفاده کنید . برای مثال در visual Studio .Net پروژه های از نوع Setup and Deployment اینکار را برای شما انجام می دهند .

۱۲ – در صورت امکان به مبحث کار با شی پرینتر در وی بی بپردازید .
پاسخ : بزودی در این زمینه مطالبی خواهم نوشت .

۱۳ – چگونه می توان فرمهای بدونControl box را با کد نویسیMinimize کرد؟
پاسخ : با استفاده از خاصیت WindowsState . برای مثال اگر نام فرم شما Form1 باشد کد زیر فرم شما را مینیمم می کند :

```
Form1.WindowState = ۱
```

و کد زیر فرم شما را ماکزیمم می کند :

```
Form1.WindowState = 2
```

۱۴ – چه طوری میشه فونتtooltiptex t را عوض کرد ؟ مثلا فونت اونو فارسی کرد؟
پاسخ : ؟؟؟؟

پاسخ به سوالات-۵

– چگونه می توان فایل Pdf ای را درون فرم ویژوال بیسیک نشان داد ؟
پاسخ : دو راه برای نمایش فایلهای Pdf در ویژوال بیسیک وجود دارد :
– استفاده از دستور ShellExecute برای نمایش فایل توسط Acrobat Reader . برای مثال :

```
ShellExecute hwnd, "open",
"C:\acrobat5\reader\acrobat.pdf", vbNullString, "C:\", 1
```

در این روش نرم افزار Adobe Acrobat حتماً بایستی روی هر کامپیوتری که پروژه تان را روی آن اجرا می کنید وجود داشته باشد .

- استفاده از کتابخانه Adobe Acrobat Type Library:

در این روش بایستی نرم افزار Acrobat Adobe روی کامپیوتری که پروژه تان را روی آن Develop می کنید وجود داشته باشد . ابتدا کتابخانه فوق را از بخش references موجود در منوی Project به پروژه تان اضافه کنید . سپس با استفاده از اشیای کلاس Acrobat می توانید برنامه مورد نظرتان را بنویسید . برای مثال کد زیر عنوان فایل Pdf را استخراج می کند :

```
Dim opdf As Acrobat.CAcroPDDoc
Set opdf = CreateObject("AcroExch.PDDoc")
opdf.Open (x)
Dim y As String
y = opdf.GetInfo("Title")
```

کد زیر مشابه روش بالا است اما احتیاجی به اضافه کردن کتابخانه مذکور به references نیست :

```
Dim opdf As Object
Set opdf = CreateObject("AcroExch.PDDoc")
opdf.Open (x)
Dim y As String
y = opdf.GetInfo("Title")
```

۲ – چگونه برای MP3 Player ساخت خودم در ویژوال بیسیک یک رقص نور مانند Winamp درست کنم ؟

پاسخ : بایستی از ترکیبی از الگوریتم های ریاضی و گرافیکی استفاده

کنید . این روشها بسیار متنوع بوده است . اگر در اینترنت کمی بگردید به جواب خواهید رسید .

۳ – لطفاً در مورد hwnd یا همان هندل فرمها توضیح کامل بدهید .
پاسخ : هر فرم در یک برنامه کاربردی در ویندوز با استفاده از الحاق یک دستگیره یا هندل به آن مشخص می شود . این هندل را با hWnd (یا HWindow) نیز نشان می دهیم . برخی توابع کتابخانه ای ویندوز به خاصین هندل فرم جاری بعنوان یک آرگومان نیاز دارند تا عملی را بر روی آن فرم انجام دهند . بعبارت دیگر توسط هندل یک فرم می توان به مشخصات و خصوصیات آن فرم دسترسی داشت . خاصیت هندل فرم در ویژوال بیسیک خاصیتی فقط خواندنی است .
باید توجه داشت که هندل یک فرم با هندل زمینه دستگاه آن متفات است . زمینه دستگاه یا device context یکی دیگر از مشخصات یک فرم می باشد . در ویندوز هر سطحی که عمل رسم بر روی آن انجام می شود یک زمینه دستگاه نام دارد . برای دسترسی به زمینه دستگاه هر فرم از هندلی به اسم hDC استفاده می شود .

۴ – چگونه می توان در ویژوال بیسیک از کتابخانه openGL استفاده کرد ؟
پاسخ : قبلاً در این وبلاگ مطلبی در این زمینه نوشته ام . به این آدرس مراجعه کنید :

۵ – در مورد نصب ویژوال بیسیک دات نت منبعی را معرفی کنید .
پاسخ : به کتابهای.NET مراجعه کنید . فقط این نکته را ذکر می کنم که در زمان نصب ویژوال استدیو دات نت ابتدا بایستی FrontPage Server Extension را روی سیستمتان نصب کرده و فعال کنید . سپس بایستی توسط سی دی Prerequisites موجود در مجموعه سی

دی های نصب ، موارد مورد نیاز برای نصب دات نت را روی سیستمتان نصب کنید (برای مثال .Net Framework) سپس بسراع نصب خود دات نت بروید .

۶ – چگونه می توان از دستورات داس در VB استفاده کرد برای مثال dir ؟

پاسخ : بابا بیخیال !!!

۷ – چگونه می توان skin هایی را که با استفاده از وی بی ایجاد می کنیم به سیستممان اضافه کنیم ؟

پاسخ : دوست عزیزی جواب این سوال را داده اند : " شما میتونید از برنامه زیبا و توانمند ActiveSkin استفاده کنی که ورژن ۴,۳ اونرو میتونی از ZDnet دانلود کنی اگر کرکش رو پیدا نکردی خبر بده برات بفرستم "

۸ – چگونه می توان دو فایل را بهم bind کرد (بهم چسباند) مثلاً دو فایل اجرایی (EXE) را بهم چسباند .

پاسخ : اگر هدف تنها اینست که دو فایل بهم بچسبند می توان محتوای فایل اول و سپس محتوای فایل دوم را خواند و آنها را در یک فایل جدید ریخت . در مورد چگونگی خواندن محتوای فایلهای باینری قبلاً در این وبلاگ مطالبی نوشته ام . اما اگر هدف الحاق دو فایل بهم است بطوریکه فایلها قابل دسترسی باشند (مثلاً الحاق یک ویروس که به فایلی اجرایی می چسبد) بایستی با ساختار فایلهای گوناگون آشنا باشید .

۹ – چگونه می توان یک فرم را در حالت Always on Top قرار داد ؟

پاسخ : حالت Always on Top حالتی است که در آن همیشه فرم برنامه شما قابل مشاهده در صفحه ویندوز باشد . (حتی اگر برنامه

دیگری انتخاب شده و فعال باشد) . برای قرار دادن فرم در این حالت از یک API موجود در کتابخانه user32 با نام SetWindowPos استفاده می شود . چگونگی declare کردن این تابع بصورت زیر است (این declare را در بالای کدهای مربوط به فرمتان قرار دهید) :

```
Private Declare Function SetWindowPos Lib "user32"
(ByVal hwnd As Long, ByVal hWndInsertAfter As Long,
ByVal x As Long, ByVal y As Long, ByVal cx As Long,
ByVal cy As Long, ByVal wFlags As Long) As Long
```

همچنین ثابتهای زیر را در بالای کدتان تعریف کنید :

```
Const SWP_NOMOVE = 2
Const SWP_NOSIZE = 1
Const FLAGS = SWP_NOMOVE Or SWP_NOSIZE
Const HWND_TOPMOST = -1
Const HWND_NOTOPMOST = -2
```

یک تایمر با Interval ای برابر ۱ در فرمتان قرار دهید و کد زیر را برای متد Timer آن بنویسید تا فرم در این حالت قرار بگیرد :

```
Dim result As Long
result = SetWindowPos(Form1.hwnd,
HWND_TOPMOST, 0, 0, 0, 0, FLAGS)
```

برای غیر فعال کردن این حالت کد زیر را در برنامه تان بنویسید :

```
Timer1.Enabled = False
```

```
Dim result As Long
result = SetWindowPos(Form1.hwnd,
HWND_NOTOPMOST, 0, 0, 0, 0, FLAGS)
```

برای فعال کردن مجدد این حالت کافیست خاصیت Enabled تایمر را True کنید .

۱۰ – من الان دارم با jro(jet replica) با vb6 کار میکنم. مشکلم اینجاست که نمیتونم با یک فایل از یک ftp ارتباط بر قرار کنم هتی یک فولدر با ای پی ولید درست کردم ولی نشده .
پاسخ : در مورد مشکلتان واضحتر توضیح بدهید .

۱۱ – چگونه می توان از طریق ویژوال بیسیک با اسکنر ارتباط برقرار نموده و عکس را از آن گرفت و در بانک اطلاعاتی ذخیره نمود .
پاسخ : در مورد بخش آخر سوال که ذخیره عکس در بانک اطلاعاتی می باشد قبلاً مطالبی در این وبلاگ نوشته ام . اما در مورد قسمت اول بایستی از یکسری OCX برای اینکار استفاده کنید مانند Twain Scanning ocx و یا Kodak Image Control ocx .

۱۲ – من از datagrid استفاده میکنم و هر تغییر در رکوردهای آن صورت بگیرد مستقیماً در دیتابیس اعمال میشود. من میخواهم وقتی کاربر هر کاری در برنامه کرد و در آخر از برنامه خواست خارج بشه و از کل تغییراتی که کرده بود صرف نظر کنه یه پیغام بهش بده و اگه خواست تغییراتی که انجام داده صرف نظر کنه همه تغییرات برگرده به حالت اول .
پاسخ : من راه حلی بنظرم نرسید . دوستان اگر نظری دارند بدهند . در ضمن به کتاب برنامه نویسی بانکهای اطلاعاتی در ویژوال بیسیک چاپ شده توسط انتشارات نص نیز مراجعه کنید .

۱۳ – در مورد کار با ADO و چگونگی برقراری ارتباط با فایلهای mdb توضیح دهید .

پاتسخ : قبلاً به اختصار در این زمینه مطالبی نوشته ام . در ضمن به کتاب برنامه نویسی بانکهای اطلاعاتی در ویژوال بیسیک چاپ شده توسط انتشارات نص نیز مراجعه کنید .

۱۴ – چگونه می توان یک صفحه A4 را در یک فرم ایجاد کرد وروی اون مطالب و یا جدول و ... را برای چاپ آماده کرد مثل لیست حقوقی و یا صورت هزینه و این جور چیزها .

پاسخ : شما اطلاعات خود را بهر شکلی که می خواهید روی فرم قرار دهید . در زمان چاپ فرم می توانید سایز صفحه را با استفاده از خاصیت Papersize شی Printer مشخص کنید . برخی مقادیر که می توان برای خاصیت PaperSize تنظیم نمود عبارتند از :

۱ : Letter, 8 1/2 x 11 in

۸ : mm A3, 297 x 420

۹ : A4, 210 x 297 mm

۱۱ : A5, 148 x 210 mm

برای اطلاعات بیشتر به مطلبی که در مورد شی پرینتر نوشته ام مراجعه کنید .

نکته : در دات نت نیز شی پرینتر در کلاس System.Drawing.Printing موجود است .

۱۵ – وقتی که یک فرم جهت ورود اطلاعات ساخته می شود باستی از طریق دکمه Tab به فیلدهای بعدی رفت . چگونه می توان کاری کرد که با زدن کلید Enter در هر فیلد به فیلد بعدی رفت ؟

پاسخ : یک روش اینست که در متد KeyPress هر کادر متنی (یا فیلد

ورود اطلاعات) کدی بنویسید که تشخیص دهد اگر دکمه Enter فشرده شده فوکوس را به فیلد بعدی مورد نظر شما منتقل کند . برای مثال اگر فرض کنید دو کادر متنی با نامهای Text1 و Text2 در فرمتان دارید کد زیر را برای متد KeyPress کادر متنی Text1 بنویسید :

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
  If KeyAscii = 13 Then Text2.SetFocus
End Sub
```

۱۶ – لطفاً در مورد StatusBar در WebBrowser توضیح بدهید .

لطفاً در مورد نمایش History در WebBrowser توضیح دهید .

پاسخ : سوال خود را دقیقتر مطرح کنید .

۱۷ – چگونه می توان فعالیت کلیدهای Alt+Ctrl+Del و Alt+Esc و Alt+Tab را در ویژوال بیسیک غیر فعال کرد ؟

پاسخ : برای اینکار از تابع SystemParametersInfo موجود در کتابخانه user32 استفاده می شود . برای declare کردن این API بصورت زیر عمل کنید :

```
Private Declare Function SystemParametersInfo Lib
"user32" Alias "SystemParametersInfoA" (ByVal uAction
As Long, ByVal uParam As Long, lpvParam As Any,
ByVal fuWinIni As Long) As Long
```

همچنین ثابت زیر را نیز تعریف کنید :

```
Private Const SPI_SCREENSAVERRUNNING = 97
```

حال توسط کد زیر می توانید ترکیب Alt+Ctrl+Del را غیر فعال کنید .

```
Dim ret As Integer
Dim pOld As Boolean
ret =
SystemParametersInfo(SPI_SCREENSAVERRUNNING,
True, pOld, 0)
```

برای اطلاعات بیشتر و نیز چگونگی غیر فعال کردن دو ترکیب دیگر به این آدرس مراجعه کنید :

۱۸ – چگونه می توان API های WinInet را به API ها اضافه کرد ؟
پاسخ : در بخش WinInet این مطلب را توضیح داده ام .

۱۹ – چگونه می توان یک فایل را بصورت مخفیانه به یک آدریس ایمیل در یاهو ارسال کرد ؟
پاسخ : ؟؟؟

۲۰ – یک برنامه تحت ویژوال بیسیک معرفی کنید که بتوان با آن انتگرال دو گانه گرفت .
پاسخ : اگر منظورتان محاسبه مقدار عددی انتگرال دو گانه یک تابع دو متغیره در یک بازه است که باید از الگوریتم های محاسبات عددی استفاده نمائید . برای اطلاعات بیشتر به کتابهای محاسبات عددی پیشرفته مراجعه کنید . اما اگر منظورتان محاسبه انتگرال دوگانه یک تابع است که اینکار به این راحتی ها امکان پذیر نیست و حتی شاید غیر ممکن باشد .

۲۱ – یک Data Type به اسم PRAS_PORT_0 در Msdn وجود دارد که بصورت زیر تعریف شده است :

```
typedef struct _RAS_PORT_0 {
  WCHAR
wszPortName[RASSAPI_MAX_PORT_NAME];
  WCHAR
wszDeviceType[RASSAPI_MAX_DEVICETYPE_NAME]
;
  WCHAR
wszDeviceName[RASSAPI_MAX_DEVICE_NAME];
  WCHAR
wszMediaName[RASSAPI_MAX_MEDIA_NAME];
  DWORD reserved;
  DWORD Flags;
  WCHAR wszUserName[UNLEN + 1];
  WCHAR wszComputer[NETBIOS_NAME_LEN];
  DWORD dwStartSessionTime;
  WCHAR wszLogonDomain[DNLEN + 1];
  BOOL fAdvancedServer;
} RAS_PORT_0, *PRAS_PORT_0;
```

چگونه می توان این data type را در ویژوال بیسیک تعریف نمود ؟ کلاً اگر بخواهیم data type ها را در ویژوال بیسیک تعریف کنیم چه باید بکنیم ؟

پاسخ : برای تعریف data type در ویژوال بیسیک از ساختار Type- End Type استفاده می شود . برای مثال :

```
Private Type SampleType
  mem1 As Integer
```

```
  mem2 As String
End Type
```

در مثال فوق یک data type از نوع Private با نام SampleType تعریف شده که دارای دو عضو به نامهای mem1 از نوع Integer و mem2 از نوع String است .

در مورد تغییر انواع داده ای نیز بصورت زیر عمل کنید :

Integer <-- Int

Boolean <-- Bool

Long <-- DWORD

array of Byte <-- WCHAR

۲۲ - Platform SDK که در سایت Msdn از آن اسم برده می شود چیست ؟ آیا همان سی دی های Msdn است ؟

پاسخ : SDK یا همان kit source development ، شامل یکسری مطلب آموزشی و نمونه کد است که در مورد یک زمینه برنامه نویسی خاص توسط مایکروسافت منتشر می شود . برخی از این SDK ها قابل دانلود از سایت مایکروسافت (مثلاً DirectX SDK) و برخی دیگر فروشی هستند (Windows Driver Model SDK) .

پاسخ به سوالات-٦

۱ – اگر بر روی نسخه ای از ویژوال استدیو کار کنیم که سرویس پک بر روی آن نصب شده باشد و بعد پروژه را بر روی سیستمی منتقل کنیم که فاقد سرویس پک باشد با چه مشکلی مواجه می شویم ؟

پاسخ : سرویس پک باعث ارتقا برخی از منابع و کامپوننتهای وی بی می

شود که اگر شما از آن کامپوننتها و منابع در برنامه تان استفاده کرده باشید برنامه شما در سیستم جدید قابل کامپایل نخواهد بود .

۲ – من تو برنامه ام از DataReport استفاده کردم. می خوام به یه پرینتر که تو شبکه است, پرینت بفرستم. از کامپیوتر خودم میفرسته. ولی برنامه رو package میکنم و نصب می کنم روی یه کامپیوتر دیگه, موقع اجرا, وقتی آیکون پرینتر رو می زنم, هیچ عملی انجام نمیده و پنجره انتخاب پرینتر رو نمیاره. فکر می کنید مشکل از کجاست؟
پاسخ : من چیزی به ذهنم نرسید .

۳ – درباره توابعی که با ماوس کار می کنند بیشتر توضیح دهید . برنامه
پاسخ : به زودی مطلبی در این مورد خواهم نوشت .

٤ – چگونه می شود از دست فایلهای Runtime ویژوال بیسیک مثال msvbvm60.dll خلاص شد و کاری کرد که فایل EXE بدون این فایل ها کار کند ؟
پاسخ : متاسفانه هیچ راهی وجود ندارد زیرا برنامه های EXE ساخته شده توسط وی بی برای اجرا به این فایلها نیاز دارند . تنها راه اینست که یک برنامه Setup از پروژه تان بسازید . نرم افزارهای ساخت Setup نیز در واقع فایلهای Runtime را درون فایل Setup قرار می دهند .

٥ – اگر بخواهیم یک درایو را توسط ویژوال بسیک و بدون منوی فرمت ویندوز فرمت کنیم چگونه باید عمل کنیم ؟
پاسخ : قبلاً در بخش پرسش و پاسخها به این سوال جواب داده ام .

٦ – چگونه می توان دکمه هایی مانند ئکمه های XP در برنامه قرار داد که گوشه آنها حالت خمیدگی داشته باشد ؟
پاسخ : قبلاً در بخش پرسش و پاسخها به این سوال جواب داده ام .

۷ – چند کتاب راجع به Api و استفاده از دیتا بیس در ویژوال بیسیک معرفی کنید .
پاسخ : قبلاً در بخش پرسش و پاسخها به این سوال جواب داده ام .

۸ – در یک برنامه پرینتر چطوری سریع فعال بشه؟
پاسخ : ؟؟؟؟

۹ – ما هر جا رفتیم این لینک اکتیو اسکین خراب بود حتی خود سایتش هم خراب بود اگه میشه لینکشو برامون بزار
پاسخ : لینکی که من هم در اختیار دارم خراب است . در صورتی که لینک درستی پیدا کردم حتماً اعلام می کنم .

۱۰ – چگونه می توان در وی بی شماره تلفن گرفت ؟
پاسخ : به مبحث TAPI مراجعه کنید .

۱۱ – چگونه میشود کلید های alt , ctrl ,delete بر روی کیبرد را از کار انداخت ؟
پاسخ : قبلاً در بخش پرسش و پاسخها به این سوال جواب داده ام .

پاسخ به سوالات-۷

– در مورد Crystal Report و گزارشهای شرطی توضیح دهید .
پاسخ : به کتاب برنامه نویسی بانکهای اطلاعاتی در ویژوال بیسیک مراجعه کنید . در ضمن اگر فرصت کردم در این زمینه مطالبی خواهم نوشت .

۲ – چگونه می توان توسط وی بی یک فایل را از روی کامپیوتر به روی دایرکتوری های یک سایت آپ لود کرد ؟

پاسخ : در آرشیو سایت به بخش آشنایی با کنترل Internet Transfer مراجعه کنید .

۳ – چگونه می توان با VB Script Control کار کرد یعنی مثلاً در یک TextBox بنویسیم sin(x) و در کد برنامه sin(x) قرار داده شود ؟

پاسخ : به پرسش و پاسخهای قبلی مراجعه کنید .

۴ – چگونه می توان در وی بی برنامه ای نوشت که نام و مسیر یک فایل را از کاربر بگیرد و در مسیر انتخاب شده شورتکاتی از فایل گرفته شده ایجاد کند ؟

پاسخ : ابتدا متغیرهای زیر را که از نوع Object هستند تعریف کنید :

```
Dim wsh As Object
Dim Shortcut As Object
```

سپس بایستی شی wsh را ایجاد نمائید :

```
Set wsh = CreateObject("wscript.shell")
```

سپس بایستی شی Shortcut را ایجاد نمائید :

```
Set Shortcut =
wsh.CreateShortcut("c:\YourProgram.lnk")
```

مسیری که در کد فوق داده شده محل ساخت شورتکات می باشد .

سپس بایستی پارامترهای شی Shortcut را تنظیم کنید :

```
Shortcut.TargetPath = "C:\Program
Files\Test\YourProgram.exe"
Shortcut.IconLocation = "C:\Program
Files\Test\YourIcon.ico"
Shortcut.WorkingDirectory = "C:\Program Files\Test\"
Shortcut.Description = "Your Description"
```

پارامتر TargetPath برنامه مرتبط با شورتکات را نشان می دهد .
در پایان بایستی شورتکات را ذخیره کنید :

```
Shortcut.Save
```

نکته : برای ایجاد شورتکات در دسکتاپ یا در Startup ، بایستی متغیری از نوع SpecialForders ایجاد نمائید :

```
Dim sf As Object
Set sf = wsh.SpecialFolders
```

سپس محل ساخت شورتکات را بصورت ("sf("AllUsersDesktop & YourPath و یا ("sf("AllUsersStartup & YourPath بدهید .

۵ – چگونه می توان در وی بی مسیر دایرکتوری استارت آپ را پیدا کرد ؟

پاسخ : با استفاده از روش زیر می توانید مسیر کلیه دایرکتوریهای سیستمی را پیدا کنید :
ابتدا دو تابع زیر را از کتابخانه Shell32 تعریف کنید :

```
Private Declare Function SHGetSpecialFolderLocation
Lib "shell32" (ByVal hwnd As Long, ByVal nFolder As
Long, Pidl As Long) As Long
Private Declare Function SHGetPathFromIDList Lib
"shell32" (Pidl As Long, ByVal FolderPath As String) As
Long
```

حال متغیر SystemFolder از نوع Enum را بصورت زیر تعریف کنید :

```
Public Enum SystemFolder
Desktop = 0
```

```
StartMenu_Programs = 2
My_Documents = 5
Favorites = 6
Startup = 7
Recent = 8
SendTo = 9
Start_Menu = 11
Windows_Desktop = 16
Network_Neighborhood = 19
Fonts = 20
ShellNew = 21
AllUsers_Desktop = 25
ApplicationData = 26
Printhood = 27
TemporaryInternetFiles = 32
Cookies = 33
History = 34
End Enum
```

تابع پیدا کردن دایرکتوریهای سیستمی بصورت زیر خواهد بود :

```
Public Function FindSystemFolder(ByVal lngNum As
SystemFolder) As String
  Dim lpStartupPath As String * MAX_PATH
  Dim Pidl As Long
  Dim hResult As Long
  hResult = SHGetSpecialFolderLocation(0, lngNum, Pidl)
  If hResult = 0 Then 'there is a result
    hResult = SHGetPathFromIDList(ByVal Pidl,
lpStartupPath)
    If hResult = 1 Then
      lpStartupPath = Left$(Trim$(lpStartupPath),
InStr(lpStartupPath, Chr(0)) - 1)
      FindSystemFolder = Trim$(lpStartupPath)
    End If
  End If
End Function
```

۶ – چگونه می توان از دستورات Command Prompt مثلاً دستورات Net Command در وی بی استفاده کرد ؟

پاسخ : فکر نمی کنم روشی برای استفاده از نتیجه این دستورات در وی بی وجود داشته باشد اما برای دسترسی به Prompt Command در وی بی بایستی از روش Shell Programming استفاده کنید . برا ی اطلاعات بیشتر به این کتاب مراجعه کنید .

۷ – در مورد وارد کردن فایلهای Pdf در وی بی با اینکه نرم افزار Adobe Acrobat روی سیستم من نصب است اما در بخش References نتوانستم نام Adobe Acrobat Type Library را پیدا کنم همچنین با این روش نمی توان به عکسهای داخل فایل Pdf دسترسی پیدا کرد .

پاسخ : این reference در موقع نصب Adobe Acrobat 5 (کل Adobe Acrobat و نه فقط Reader آن) در دایرکتوری Adobe\Acrobat\Acrobat\۵,۰ قرار داده می شود . در مورد عکس هم من راه حلی به ذهنم نرسید و بنظر من بهترست از همان Ocx موجود استفاده کنید .

۸ – من یک سخت افزار ساخته ام که با ویندوز ۹۸ و با پورت سریال و تحت وی بی کار می کند . مشکلی که من دارم اینست که وقتی کامپیوتر روشن می شود و ویندوز در حال بوت شدن است سیستم عامل پورت سریال را چک می کند و این باعث یک فرمان ناخواسته در سخت افزار می شود . آیا راهی وجود دارد که ویندوز ۹۸ در هنگام بوت شدن پورتهای سریال را چک نکند ؟

پاسخ : یک روش اینست که یک مدار محافظت کننده را به سخت افزارتان

اضافه کنید که در هنگام بوت شدن فرمان ناخواسته ای اجرا نشود . در مورد غیر فعال کردن گزینه چک پورت سریال من روشی پیدا نکردم .

۹ – چگونه می توان با کلیک کردن روی یکی از مقادیر درون یک کمبو باکس (استیل ۲ : دراپ داون لیست) از فرم شامل کمبو باکس خارج و فرم دیگری را ظاهر نمود ؟

پاسخ : با استفاده از رویداد Click مربوط به کمبوباکس و متد Hide و Show مربوط به فرم .

برای مثال فرض کنید یک کمبوباکس با نام Combo1 با یک آیتم با مقدار Test1 داشته باشیم . کد زیر را برای رویداد Click آن بنویسید :

```
If Combo1.Text = "Test1" Then
  Form1.Hide
  Form2.Show
End If
```

۱۰ – چگونه می توان درایو سی دی رام را در وی بی تشخیص داد ؟

پاسخ : ابتدا مورد Microsoft Scripting Runtime را از بخش references پروژه خود اضافه کنید . سپس کد زیر را برای مشخص نمودن سی دی درایو در برنامه تان بنویسید :

```
Dim fso As New Scripting.FileSystemObject
Set fso = New Scripting.FileSystemObject
Dim drv As Drive
For Each drv In fso.Drives
   If drv.DriveType = CDRom Then
      CDPath = drv.Path
      Exit For
   End If
Next drv
MsgBox (CDPath)
```

```
Set drv = Nothing
Set fso = Nothing
```

پاسخ به سوالات-۸

۱ – چگونه می توان Calender Control را فارسی کرد ؟

پاسخ : فکر می کنم خودتان باید یک کنترل جدید طراحی کنید .

۲ – من دارم سعی می کنم یک MsgBox با دکمه های فارسی طراحی کنم . یک کنترل ActiveX نیز برای آن نوشته ام اما هم کامل نیست و هم محدودیتهایی دارد چون خود جعبه پیغام یک تابع است . نظر شما در این مورد چیست ؟

پاسخ : لینک های زیر درباره چگونگی ساخت MsgBox های سفارشی در وی بی است :

A Custom MsgBox

Custom Message Box

A non Standrad MsgBox

Custom MsgBox Control

Create your own custom MsgBox function

A cool custom (form) msgbox using API to get icon

برای یافتن نمونه های بیشتر به سایت

http://www.planetsourcecode.com/ رفته و عبارت custom msgbox را در بخش ویژوال بیسیک ۶ آن جستجو کنید .

۳ – چگونه می توان در وی بی اطلاعات را روی یک فایل ذخیره کرد . برای مثال اطلاعات روی Database را در فایل backup گرفت و بتوان از این اطلاعات بر اساس یک فیلد یا گزینه خاص گزارش گیری کرد ؟
پاسخ : برای ایجاد و کار با فایلها قبلاً مطالبی در وبلاگ نوشته ام . به آرشیو موضوعی وبلاگ مراجعه کنید . در مورد جستجو و گزارش گیری از اطلاعات فایل خودتان بایستی روتین های جستجو در فایل را بنویسید بعبارت دیگر بایستی اطلاعات فایل را load کرده و روی آنها عمل جستجو را انجام دهید .

۴ – چگونه می توان فرم فعال را تشخصی داد ؟ آیا می توان متغیری با خاصیت فرم تعریف کرد و آن را به فرم فعالمان اختصاص داد ؟
پاسخ : در مورد تشخیص فرم فعال من روشی پیدا نکردم اما فکر می کنم بایستی از دستگیره فرم یا Hook استفاده کنید . در مورد تعریف متغیر نیز وی بی دارای شیی به نام Form است که می توانید در برنامه تان از آن استفاده کنید برای مثال اگر فرمی به اسم Form1 در برنامه تان داشته باشید آنگاه :

```
Dim F As Form
Set F = New Form1
```

۵ – چگونه می توان محتوای چند فایل را گرفته و درون یک فایل Exe که از قبل وجود دارد ریخت . این فایل exe یک برنامه است و می خواهم این فایلها را با دستوراتی کنترل کنم یعنی کلاً چند تا فایل را داخل یک فایل Exe ریخت (مثل برنامه های فشرده ساز) ؟
پاسخ : قبلاً به سوال مشابه این سوال پاسخ داده ام . شما بایستی با

ساختار فایلهای Exe آشنا باشید که مطلب ساده ای نیست و فایلهای مورد نظر خود را در ناحیه مشخصی از فایل Exe درج کنید . به این لینک مراجعه کنید :

۶ – من از سورسی که آیکون برنامه را کنار ساعت ویندوز قرار می دهد استفاده کردم اما نمی دانم چگونه برای رویدادهای کلیک و دابل کلیک روی آیکون برنامه بنویسم ؟
پاسخ : همانطور که در مقاله هم گفته شده بایستی از متد Form_MouseMove استفاده کنید و در آن نوع کلیک شدن ماوس را تشخیص داده و بر اساس آن عملی را انجام دهید .

۸ – چگونه می توان از برنامه های وی بی در AutoCad استفاده کرد .
بعنوان مثال من یک برنامه نوشتم که وزن و قیمت انواع پروفیلها را در یک جدول در اختیار کاربر قرار می دهد . من می خواهم این جدول را در AutoCad وارد کنم با توجه به اینکه هم وی بی و هم آتوکد محیط گرافیکی دارند ؟
پاسخ : استفاده از ویژوال بیسیک بهمراه آتوکد توسط یک سیستم ابداعی شرکت AutoDesk به اسم ADS – AutoCAD Develpoment System – امکان پذیر است . این سیستم ارتباطی به برنامه نویسان اجازه می دهد تا برنامه های کاربردیی بنویسند که بتواند آتوکد را کنترل کرده و داده را بین آتوکد و برنامه خارجی رد و بدل کند .
پشتیبانی از ویژوال بیسیک در ADS از آتوکد ۱۳ آغاز شد . برای اطلاعات بیشتر به این فایل pdf مراجعه کنید :
همچنین در بخش reference های ویژوال بیسیک (در صورتیکه آتوکد روی سیستمتان نصب باشد) گزینه ای به اسم AutoCAD Type

Library وجود دارد که می توانید از آن در برنامه تان استفاده کنید

برای مثال :

```
Private autocadApp As AcadApplication
Set autocadApp =
CreateObject("AutoCAD.Application.15")
Dim doc As AcadDocument
Set doc =
autocadApp.Application.Documents.Open("c:\drawing.dw
g")
```

همچنین از کتاب Using Visual Basic with AutoCAD نیز می توانید استفاده کنید .

۹ – چگونه می توان تمام اطلاعات مربوط به کانکت جاری سیستم را بدست آورد مانند سرعت اتصال و اطلاعات ارسال و دریافت ؟

پاسخ : از WinInet Api استفاده کنید . قبلاً در وبلاگ مطالبی در این زمینه نوشته ام .

+ در ضمن یکی از دوستان زحمت کشیدند و مطالبی در مورد سوال استفاده از دستورات command promt در وی بی را که در پست قبلی مطرح شده بود عنوان کرده اند :

" با سلام راجع به سوال اون دوست عزیزی که پرسیده بود آیا میشه دستورات command prompt رو اجرا کرد باید بگم میشه.من این روش رو تو xp آزمایش کردم احتمالا تو ویندوز های دیگه هم جواب میده. اگه دستور Command /k رو تو runوارد کنیم و جلوش دستور رو بنویسیم دستور ما اجرا میشه و Command prompt بسته می شه. و اگه بنویسیم command /c پنجره command prompt باز

میمونه. حالا تو vb از دستور shell استفاده میکنیم. مثلا مینویسیم:
"shell "command /k dir یا "shell "command /c dir که نتیجه هردو dir گرفتن از فهرست جاریه با این تفاوت که دومی پنجره Command رو پس از اجرای دستور میبنده. واسه اطلاعات بیشتر میتونید تو command prompt بنویسید command /? در xp هم به جای command میتونید بنویسید cmd در ضمن، دستور NET خودش یه فایل اجرایی داره به همین نام که اگه تو درایو ویندوز بگردی پیداش میکنی. میتونی اونو مستقیما با دستور Shell اجرا کنی . "

پاسخ به سوالات-۹

- برنامه Multimedia Builder از چه کامپایلری برای ساخت برنامه های اجراییش استفاده می کند ؟ آیا می توان از امکانات آن در برنامه نویسی وی بی استفاده کرد ؟

پاسخ : در مورد این برنامه و کامپایلر آن اطلاعات زیادی ندارم اما شاید بتوان reference های آنرا در ویژوال بیسیک استفاده نمود .

۲ – چگونه می توان یک بانک اطلاعاتی اکسس را به فرمت یونیکد تبدیل کرد ؟

پاسخ : یک برنامه بنویسید که اطلاعات را از جداول بانک طلاعاتی بخواند ، معادل یونیکد آنها را تولید کرده و در جداول قرار دهد .

۳ - چگونه می توان در وی بی برای یک بانک Backup درست کرد ؟

پاسخ : اگر منظورتان از بانک ، بانک اطلاعاتی است می توانید برنامه ای بنویسید که اطلاعات جداول بانک اطلاعاتی مورد نظرتان را بخواند و در یک فایل قرار دهد .

۴ – چگونه می توان صدا را توسط TCP/IP انتقال داد ؟

پاسخ : شما بایستی اطلاعات صدا را بصورت stream بگیرید و توسط WinSocket که قبلاً در مورد آن نوشته ام انتقال دهید و در مقصد نیز بصورت stream دریافت کنید و پخش نمایید . برای کار با stream های صدا می توانید از یک Dll به نام WavStream استفاده نمایید . برای اطلاعات بیشتر به این آدرس مراجعه کنید .

۵ – من یک مشکل در گزارش گیری دارم . می خواهم در گزارش روی پرینتر چند آیتم بصورتی از بقیه آیتمها متمایز باشند مثلاً رنگ زمینه آنها فرق کند و غیره ؟

پاسخ : در زمان ساخت گزارشتان باید آیتم های مورد نظرتان را با background رنگی قرار دهید . (برای مثال در زمان استفاده از Date Report یا Crystal Report) .

۶ – چگونه می توان با استفاده از وی بی دیسکت یا درایو های هادر را فرمت کرد ؟

پاسخ : قبلاً به این سوال پاسخ داده ام . به اینجا (سوال شماره ۸) مراجعه کنید .

۷ – چگونه می توان پارتیشنی که ویندوز در آن قرار دارد را توسط وی بی شناسایی کرد ؟

پاسخ : از دستور زیر استفاده کنید :

```
Environ$("windir")
```

این دستور دایرکتوری ویندوز را به شما می دهد . با گرفتن اولین کاراکتر آن درایو ویندوز مشخص می شود .

۸ – وقتی از نرم افزار اکتیو اسکین استفاده می کنم گزینه راست به چپ برای آن تنظیم نمی شود . آیا می توان گزینه راست به چپ را برای آن تنظیم کرد ؟

پاسخ : من زیاد با این نرم افزار کار نکرده ام . اگر دوستان راه حل را پیدا کردند جواب بدهند .

۹ – برنامه من در مانتیور ۱۷ اینچ خوب نشان داده ی شود اما در مانیتور ۱۵ اینچ نه . علت چیست ؟

پاسخ : سایز فرم را بزرگ گرفته اید .

۱۰ – چگونه در وی بی می توان برنامه ای ساخت که توسط آن ایمیل فرستاد ؟

پاسخ : قبلاً به این سوال پاسخ داده ام . به آرشیو ماهیانه مراجعه کنید .

۱۱ – چند کتاب پیشرفته در مورد Dll و اکتیو اکس در وی بی معرفی کنید .

پاسخ : کتابهای انگلیسی زیادی در این زمینه وجود دارد که در سایت amazon با یک جستجوی ساده می توانید آنها را بیابید . کتاب فارسی در زمینه dll ندیده ام اما در مورد اکتیوایکس به کتاب برنامه نویسی اینترنت در ویژوال بیسیک ۶ انتشارات نص مراجعه کنید .

۱۲ – چگونه می توان توسط تابعهای FindWindow و FindWindowEx یا با هر روش دیگر username و password اکانت ها را در آورد ؟

پاسخ : ؟؟؟

۱۳ – در مورد کار با تلفن در وی بی توشیح دهید ؟ می می خواهم در برنامه ام شماره تماس گیرنده را داشته باشم و بتوانم با او از طریق برنامه ام صحبت کنم .
پاسخ : به آرشیو موضوعی سایت و بخش کار با TAPI مراجعه کنید .

۱۴ – چگونه می توان Scrollbar را در FlexGrid به پایین ترین جای ممکن آورد برای مثال هر بار که با استفاده از AddItem یک ردیف به جدول اضافه می شود Scrollbar به پایین ترین جلی ممکن برود .
پاسخ : من راه حلی به نظرم نرسید .

۱۵ – چگونه می توان شماره سریال هارد را بدست آورد ؟
پاسخ : از تابع GetVolumeInformation موجود در کتابخانه Kernel32 استفاده کنید . برای declare کردن این تابع بصورت زیر عمل کنید :

```
Declare Function GetVolumeInformation Lib "kernel32"
Alias
"GetVolumeInformationA" (ByVal lpRootPathName As
String, ByVal
lpVolumeNameBuffer As String, ByVal
nVolumeNameSize As Long,
lpVolumeSerialNumber As Long,
lpMaximumComponentLength As Long,
lpFileSystemFlags As Long, ByVal
lpFileSystemNameBuffer As String, ByVal
nFileSystemNameSize As Long) As Long
```

حال کد زیر را برنامه تان قرار دهید :

```
Dim lngNumSerial As Long
Dim strRotulo As String
Dim strTipoVolume As String
```

```
Dim lngVal As Long
strRotulo = Space(255)
strTipoVolume = Space(255)
lngVal = GetVolumeInformation("C:\", strRotulo,
Len(strRotulo), lngNumSerial, 0, 0, strTipoVolume,
Len(strTipoVolume))
MsgBox (lngNumSerial)
```

البته شماره سریالی که کد فوق بر می گرداند شماره سریالی است که سیستم عامل به هارد می دهد و زمانیکه هارد مجدداً پارتیشن بندی شود این شماره تغییر خواهد کرد . برای بدست آوردن شماره سریالی که سازنده روی هارد قرار داده از تابع DeviceIoControl استفاده کنید . برای اطلاعات بیشتر به آدرسهای زیر مراجعه کنید :

شماره سریال هارد-۱

شماره سریال هارد-۲

شماره سریال هارد-۳

شماره سریال هارد-۴

شماره سریال هارد-۵

۱۶ – در مورد نحوه رایت بر روی سی دی توسط وی بی توضیح دهید ؟

پاسخ : بایستی از یکسری ActiveX هایی که برای این منظور ایجاد شده استفاده کنید . یکی آنها mcdburner نام دارد که می توانید اطلاعات بیشتر در مورد را در اینجا ببینید .

۱۷ – برنامه ای برای ساخت برنامه های نصبی معرفی کنید زیرا برنامه Deployment Wizard & Package اصلاً امکانات مناسبی در این زمینه ندارد ؟

در مورد EXE کردن برنامه هایی که می نویسیم به طوری که در کامپیوتری که VB نداشته باشد اجرا شود توضیح دهید .

پاسخ : دو برنامه مشهور برای اینکار وجود دارد :

InstallShield

Installer Wise

توسط لینکهای فوق می توانید وارد سایت رسمی آنها شده و نسخه های آزمایشی این برنامه ها را دریافت کنید .

در اینجا نیز می توانید مقایسه بین این دو محصول را ببینید . لازم بذکر است که هر دو محصول Net Support. می باشند .

سعی می کنم در مورد یکی از ایندو برنامه یک دوره آموزشی را در وبلاگ بنویسم .

۱۸ – چگونه می توان از کامپوننت هایی که در ویندوز XP استفاده شده استفاده کرد (منظورم دکمه ها و فریم ها و غیره است که در وی بی بصورت کلاسیک هستند) ؟

پاسخ : راه حلی به نظر من نرسید .

۱۹ – چگونه می توان یک ماشین حساب را در وی بی طراحی کرد ؟

چگونه می توان یک متغیر رشته ای را بصورت فرمول به برنامه داد تا آنرا محاسبه کند ؟

پاسخ : بایستی برای پردازش عبارت محاسباتی ابتدا عبارت مورد نظر

را بصورت Postfix یا Prefix در آورید . سپس آنرا خوانده و در Stack بریزید و آنگاه Stack را پردازش نمایید .
برای اطلاعات بیشتر به کتابهای ساختمان داده (Data Structure) مراجعه کنید .

۲۰ – در مورد ارتباط یک دستگاه دیجیتال با کامپیوتر (مثلاً پورت موازی) توضیح دهید .
لطفا در مورد کنترل وسایل از طریق پورت موازی یا USB توضیح بدید (آیا نیازی به مدار جانبی دارد یا نه) .
پاسخ : سایتها و کتابهای زیادی را تاکنون دربخش مقالات و نیز پرسش و پاسخهای وبلاگ در این زمینه معرفی کرده ام . به آرشیو ماهیانه مراجعه کنید .

۲۱ – من می خواهم اطلاعاتی را از طریق یک صفحه وب در بانک اطلاعاتی SQL Server 2000 به صورت یونیکد ذخیره کنم. اولاً هنگام درست کردن بانک اطلاعاتی و جداول آن چه نوع کاراکتر ستی را برای آنها در SQLServer انتخاب کنم؟ ثانیاً آیا فقط تنظیم CodePage صفحه وب به یونیکد کافی است یا باید تغییرات دیگری نیز در نحوه ذخیره اطلاعات اعمال نمایم؟
پاسخ : کاراکتر ست صفحات html خود را windows-1252 قرار داده و کدپیج Session را در برنامه تان نیز ۱۲۵۲ قرار دهید .

۲۲ – چگونه می توان برای یک فایل خروجی از برنامه آیکون تعیین کرد ؟
پاسخ : برای فرم برنامه تان یک آیکون قرار دهید . برای اینکار از بخش خصوصیات فرم مورد Icon را زده و یک فایل ico یا cur به آن اختصاص دهید .

۲۳ – من قبلا یک برنامه answering machine نوشته بودم که با مودم Rockwell جواب میداد حالا که آن را روی مودم zoltrix اجرا میکنم جواب نمی دهد . در ضمن برنامه را با استفاده از کنترلMscomm نوشته بودم چطور میشه برنامه ایی نوشت که روی همه مودم ها جواب بده و وابسته به دستورات خاص نباشه ؟

پاسخ : بجای استفاده از MsComm از TAPI استفاده کنید . برای این منظور به آرشیو موضوعی یا ماهیانه وبلاگ مراجعه کنید .

۲۴ – سوال من در مورد D3DVertex است . زمانیکه ما یک Vertex را با P(x,y,z),N(x,y,z),u,v معرفی میکنیم منظور از N وu-v چیست ؟

پاسخ : دایرکت ایکس و سایر برنامه های مدلسازی سطوح سه بعدی (مثل DMax۳) برای مدل کردن اشیا سه بعدی و تقریب سطوح انحنا دار این اشیا از یک تقریب به اسم تقریب مش چند ضلعی استفاده می کنند . در این تقریب سطح سه بعدی را توسط یکسری چند ضلعی مسطح (معمولاً مثلث یا مربع) تخمین می زنیم . وظیفه برنامه مدلساز اینست که اطلاعات این چند ضلعی ها را بگیرد و بر اساس آن سطح شی سه بعدی را رندر کرده و با در نظر گرفتن اطلاعات نورپردازی ، ماده و بافت ، تصویر نهایی از آن شی را تولید کند . هر چند ضلعی توسط پارامترهای زیر مشخص می شود :

– مختصات رووس چند ضلعی : شامل مختصات x-y-z نقاط راس چند ضلعی

– بردار نرمال سطح : بردار نرمال سطح یک چند ضلعی ، از ضرب خارجی یا Cross Product بین دو بردار ساخته شده از اتصال سه راس چند ضلعی ایجاد می شود . از بردار نرمال سطح در الگوریتم های نورپردازی و حذف سطوح مخفی استفاده می شود .

– مختصاتهای نگاشت بافت (مختصاتهای u-v) رووس چند ضلعی :

برای اینکه یک تصویر دو بعدی را بر روی سطح یک شی سه بعدی بعنوان بافت قرار دهیم بایستی مشخص کنیم هر نقطه از سطح سه بعدی متناظر با کدام نقطه روی تصویر دو بعدی است . برای اینکار از مختصات نگاشت بافت استفاده می شود .

در صورت تمایل و بدست آوردن اطلاعات بیشتر در این مورد می توانید توسط ایمیل با من مکاتبه کنید .

۲۵ – من می خواهم برنامه ای بسازم که در ویندوز بصورت مخفی باشد و هر کلیدی که زده می شود را ثبت کند اما برنامه نمی تواند رویدادهای کیبورد را تشخیص دهد ؟
پاسخ : اگر شما از رویدادهای کیبورد برای گرفتن کلیدهای زده شده در ویندوز استفاده کنید اینکار تا زمانی امکان پذیر است که برنامه شما بعنوان برنامه فعال سیستم باشد اما اگر بخواهید حتی وقتی برنامه دیگری بعنوان برنامه فعال سیستم است بتوانید از کیبورد Log بگیرید بایستی از Api های کار با کیبورد استفاده کنید . در مورد این توابع در آینده مطالبی خواهم نوشت .
به این آدرس نیز سر بزنید .

۲۶ – ویژوال بیسیکی که اینجانب نصب کرده ام مربوط به ۲۰۰۳ و نسخه ۶ است اما پس از اینکه با استفاده از deployment and package wizard مجموعه نصبی مورد نظر را می سازم پس از اجرای فایل نصب دایماً با پیغام "فایلهای که نصب خواهد شد ازفایلهای موجود درکامپیوتر شما قدیمیتر است وآیا می خواهید آنها را نگه دارید مواجه می شوم. ضمناً ویندوز من XP است. با توجه به اینکه هیچ update برای تولید فایل نصبی نیافته ام خواهشمندم با توضیحات راهگشای

خودتان راه حلی برای این معضل ارایه دهید.

پاسخ : سرویس پک ۵ ویژال استدیو را از روی سایت مایکروسافت دریافت کرده و نصب کنید .

۲۷ – من برنامه ای نوشته ام که توسط CommonDialog تعداد بیش از یک فایل را بتوان باز کرد (توسط CommonDialogFlag=&H200&) و نام فایلها را درون یک لیست اضافه کند . مشکل اینجاست که در این برنامه اگر تعداد فایلهای انتخاب شده توسط کاربر بیش از ۱۷ مورد باشد وی بی خطا می دهد و احتمالاً بافر FileName پر می شود . آیا راه حلی وجود دارد که در انتخاب فایلها محدودیت نباشد .

پاسخ : ابتدا ثابت زیر را در برنامه تان تعریف کنید :

```
Private Const CD_FLAGS = cdlOFNAllowMultiselect +
cdlOFNExplorer + cdlOFNLongNames
```

سپس خاصیت Flags مربوط به CommonDialog را برابر ثابت فوق بگذارید .

آنگاه بایستی خاصیت MaxFileSize را برابر ۳۲۰۰۰ قرار دهید .

کد زیر را برای نمونه مطالعه کنید (توسط این کد می توان نام فایلهای انتخاب شده را نیز بدست آورد) :

```
   Dim i As Integer
   Dim myFiles() As String
   Dim myPath As String

   With CommonDialog1
      .MaxFileSize = 32000 'this will max out the buffer for
the filenames array for large        selections.
      .CancelError = True
      .Filter = "All Files *.*/*.*"
```

```
        .Flags = CD_FLAGS 'this is where we tell it to use
multiselect
        .ShowOpen

        myFiles = Split(.FileName, vbNullChar) 'the Filename
returned is delimeted by a null character because we
selected the cdlOFNLongNames flag

        Select Case UBound(myFiles)
           Case 0 'if only one was selected we are done
              List1.AddItem myFiles(0)
           Case Is > 0 'if more than one, we need to loop
through it and append the root directory
              For i = 1 To UBound(myFiles)
                 myPath = myFiles(0) & IIf(Right(myFiles(0),
1) <> "\", "\", "") & myFiles(i)
                 List1.AddItem myPath
              Next i
        End Select

    End With
```

برای اطلاعات بیشتر در مورد این کنترل به اینجا و اینجا مراجعه کنید .

در اینجا می توانید یک CommonControl قدرتمند تر را ببینید .

۲۸ – من می خواستم در وی بی فرمی داشته باشم که همیشه زیر تمام فرمهایم باشد . از MDIForm نیز نمی توانم استفاده کنم زیرا می خواهم روی آن فرم کنترل قرار دهم ؟

پاسخ : ابتدا سه تابع زیر را از کتابخانه user32 تعریف کنید :

```
Public Declare Function FindWindow Lib "user32" Alias
"FindWindowA" (ByVal lpClassName As String, ByVal
lpWindowName As String) As Long
Public Declare Function SetParent Lib "user32" (ByVal
hWndChild As Long, ByVal hWndNewParent As Long)
```

```
As Long
Public Declare Function SetWindowPos Lib "user32"
(ByVal hWnd As Long, ByVal hWndInsertAfter As Long,
ByVal X As Long, ByVal Y As Long, ByVal Cx As Long,
ByVal Cy As Long, ByVal wFlags As Long) As Long
```

سپس ثوابت زیر را تعریف کنید :

```
Public Const SWP_NOMOVE = &H2
Public Const SWP_NOSIZE = &H1
Public Const HWND_TOPMOST = -1
Public Const Flags = SWP_NOMOVE Or SWP_NOSIZE
```

همچنین تابع کمکی زیر را نیز در برنامه تان قرار دهید :

```
Public Function GethWndByWinTitle(winTitle As String)
As Long
   Dim retval As Long
   GethWndByWinTitle = FindWindow(vbNullString,
winTitle)
End Function
```

تابع قرار دادن فرم در حالت Bottom (زیر همه فرمها) بصورت زیر است :

```
Sub FormOnBottom(Frm As Form)
Dim DeskH As Long
DeskH = GethWndByWinTitle("Program Manager")
Call SetParent(Frm.hWnd, DeskH)
End Sub
```

تابع قرار دادن فرم در حالت Top (روی همه فرمها) بصورت زیر است :

```
Sub FormOnTop(Frm As Form)
Call SetWindowPos(Frm.hWnd, HWND_TOPMOST, 0&,
0&, 0&, 0&, Flags)
End Sub
```

تابع قرار دادن فرم در حالت نرمال بصورت زیر است :

```
Sub FormNormal(Frm As Form)
Dim DeskH As Long
DeskH = GethWndByWinTitle("Form1")
Call SetParent(Frm.hWnd, DeskH)
End Sub
```

۲۹ – سیستم عاملهای لینوکس با چه زبانی نوشته شده اند و با چه زبانی می توان آنها را تغییر داد یا لینوکس جدیدی نوشت (منظور اینکه سورس آنها به چه زبانی است) ؟

پاسخ : معمولاً اکثر سیستم عاملها را با زبان سی ایجاد می کنند .

همچنین یکی از کامپایلرهایی که در محیط سیستم عامل لینوکس برای نوشتن برنامه های جدید وجود دارد Gcc نام دارد . برای اطلاعات بیشتر در مورد این کامپایلر به اینجا مراجعه کنید .

۳۰ – شما چه زبانی را برای ساختن بازیها پیشنهاد می کنید که امکانات بیشتر و قویتری داشته باشد ؟

پاسخ : ویژوال سی

پاسخ به سوالات-۱۰

با عرض معذرت از همه دوستان بابت این هم تاخیر که در پاسخ دهی به سوالات داشتم و باز هم بخاطر این همه لطف و محبتی که شما نسبت به من دارید شرمنده ام . از امروز سعی می کنم هر دو یا سه روز یکبار به چند تا از سوالات مطرح شده بترتیب تاریخ ارسال پاسخ بدهم .

بنابراین اگر سوال خود را تازه مطرح کرده اید ممکنست چند روز طول

بکشد تا به آن جواب داده شود . امیدوارم این جوابها برای شما مفید باشد . در ضمن دوستان عزیز نیز اگر نظری در مورد یک سوال و جواب آن دارند حتماً آنرا عنوان نمایند .

+ آرشیو موضوعی وبلاگ رو هم به درخواست تعداد زیادی از دوستان بزودی موضوع بندی و مرتب خواهم کرد .

۱ – چگونه می توان پورت ماوس PS/2 را پیدا کرد و تغییرات بیتها روی پورت را استخراج نمود ؟
پاسخ : اگر ماوس شما از نوع سریال بوده و به پورت COM1 متصل باشد آدرس آن x3F8۰ می باشد . اما اگر از نوع PS/2 باشد آدرس آن x60۰ یا x64۰ میباشد . با استفاده از این شماتره پورتها و روشهای خواندن پورت که قبلاً به آنها اشاره کرده ام می توانید تغییرات پورت ماوس را بخوانید .
اطلاعات بیشتر در مورد پورت PS/2 را در اینجا بخوانید .

۲ – من می خواستم که با وی بی برنامه ای بنویسم که بتوان با آن تلفن زد و یک فایل صوتی را برای مقصد پخش کرد . چگونه اینکار را انجام دهم ؟
پاسخ : در آخرین بخش از سری مطالب آموزشی TAPI چند لینک معرفی کرده ام که برای شما مفید خواهد بود . همچنین لینکهای زیر نیز مفید می باشند :

لینک – ۱

لینک – ۲

لینک – ۳

لینک - ۴

۳ – چگونه می توان اشیایی را که در برنامه Studio Max D۳ طراحی کرده ام را به وی بی منتقل کرد ؟
پاسخ : به اولین درس از مباحث پیشرفته DirectX مراجعه کنید . بطور خلاصه باید بگویم شما ابتدا باید شی خود را با فرمت .X ذخیره کرده و سپس با استفاده از Direct3D در وی بی آنرا لود کنید .

۴ – چگونه می توان شماره کسی را که تلفن می زندتوسط وی بی مشاهده کرد ؟
پاسخ : به سوال شماره ۲ مراجعه کنید .

۵ – چگونه می توان برنامه ای ساخت که توسط آن ایمیل فرستاد ؟
پاسخ : در وی بی ۶ از امکانات MAPI (Messaging Application Interface Programming) استفاده کنید . برای استفاده از آن به بخش Components رفته و مورد Microsoft MAPI Control را به پروژه تان اضافه نمایید . این کار باعث می شود دو کنترل به ابزارهایتان اضافه شود : MAPISession و MAPIMessages . برای دیدن چگونگی استفاده از این دو کنترل به این آدرس مراجعه کنید .
در VB.Net نیز می توانید از کتابخانه System.Web.Mail استفاده نمایید . این کتابخانه دارای سه کلاس MailMessage ، MailAttachment و SmtpMail می باشد . کلاس MailMessage برای ساخت ایمیل و تنظیم اطلاعات فرستنده ، گیرنده ، موضوع و بدنه ایمیل بکار می رود . کلاس MailAttachment برای افزودن فایل الصاقی به ایمیل استفاده می شود و کلاس SmtpMail برای ارسال ایمیل استفاده می شود . برای اطلاعات بیشتر به Msdn مراجعه نمایید .

+ در پرسش و پاسخ شماره ۹ سوالی در مورد flexGrid مطرح شده بود . یکی از دوستان زحمت کشیندد و جواب زیر را داده اند :
"در مورد فلکس گرید میتونید هر بار که Item جدیدی اضافه کردید بالا ترین ردیف رو معادل آخرین ردیف قرار بدید. Grid.topRow = Grid.Rows "

یکی دیگر از دوستان نیز جواب زیر را داده اند :
" در مورد سوال ۱۴ بخش پاسخ به سوالات ۹ که چگونه می توان Scrollbar را در FlexGrid به پایین ترین جای ممکن آورد باید بگویم که با خاصیت TopRow. می توان این کار را انجام داد. به اینصورت که این خاصیت بالاترین ردیف FlexGrid را تعیین می کند و با یکی دو خط کد می توان به مقصود رسید. "

+ در پرسش و پاسخ شماره ۹ همچنین سوالی در مورد گرفتن Log از کیبرد مطرح شده بود که باز هم یکی دیگر از دوستان عنوان کرده اند که در وبلاگشان به این سوال پاسخ داده اند . برای اطلاعات بیشتر به این آدرس مراجعه کنید :

پاسخ به سوالات-۱۱

۱- چگونه می توان فایلهای انیمیشن Gif را در برنامه های وی بی قرار

داد ؟

پاسخ : می توانید از کنترلهای Ocx و یا ActiveX هایی که برای اینکار طراحی شده اند استفاده کنید .

در این لینک می توانید یک ActiveX پخش کننده فایلهای gif را بهمراه سورس کامل آن به VB6 دریافت نمایید .

این لینک نیز شامل چند Ocx و ActiveX برای پخش فایلهای انیمیشن Gif است .

با جستجو در سایتهای برنامه نویسی وی بی که برخی از آنها را در وبلاگ نیز معرفی کرده ام می توانید کنترلهای دیگری نیز پیدا کنید .

۲ – چگونه می توان در وی بی یک فایل با هر پسوندی را در یکی از درایوها اجرا کرد ؟

پاسخ : همانطور که قبلاً گفته ام برای Run کردن یک فایل اجرایی در وی بی بایستی از تابع ShellExecute استفاده نمایید . نحوه declare کردن آن بصورت زیر است :

```
Private Declare Function ShellExecute Lib "shell32.dll"
Alias "ShellExecuteA" (ByVal hwnd As Long, ByVal
lpOperation As String, ByVal lpFile As String, ByVal
lpParameters As String, ByVal lpDirectory As String,
ByVal nShowCmd As Long) As Long
```

حال فرض کنید می خواهید در مسیر D:\TestShell\ فایلی به اسم Test.exe را توسط برنامه تان اجرا کنید . کد زیر بدین منظور نوشته شده است :

```
Call ShellExecute(Me.hwnd, vbNullString,
"D:\TestShell\Test.exe", "", "", SW_SHOWNORMAL)
```

۳ – چگونه می توان هندل (Handle) یک Textbox را در یک پنجره بدست آورد ؟

پاسخ : برای بدست آوردن هندل پنجره برنامه ای که هم اکنون باز است از تابع FindWindow استفاده کنید . نحوه declare کردن آن بصورت زیر است :

```
Private Declare Function FindWindow Lib "user32"  Alias "FindWindowA"   (ByVal lpClassName As String, ByVal lpWindowName   As String) As Long
```

فرض کنید caption فرم برنامه مورد نظرتان در متغیر strWindowName قرار داشته باشد . با دستور زیر می توانید هندل پنجره آنرا بدست آورید :

```
Dim hwndFound As Long
hwndFound = FindWindow(vbNullString, strWindowName)
```

نکته : برای پیدا کردن هندل پنجره برنامه ای که caption آنرا بطور دقیق نمی دانید می توانید از تابع FindWindowLike استفاده کنید .

حال که هندل پنجره مورد نظرتان را استخراج کردید می توانید با استفاده از تابع FindWindowEx هندل اشیا موجود در آن پنجره را بدست آورید . نحوه declare کردن این تابع بصورت زیر است :

```
Private Declare Function FindWindowEx Lib "user32" Alias "FindWindowExA" (ByVal hWnd1 As Long, ByVal hWnd2 As Long, ByVal lpsz1 As String, ByVal lpsz2 As String) As Long
```

این تابع را بصورت زیر استفاده کنید :

```
htextbox = FindWindowEx(hwndFound, ByVal 0&,
"ThunderRT6TextBox", vbNullString)
```

که ThunderRT6Textbox نام کلاس Rich Textbox ها در ویژوال بیسیک ۶ است . دستور فوق هندل اولین Textbox موجود در پنجره را به شما بر می گرداند . برای بدست آوردن هندل سایر Textbox ها از حلقه زیر استفاده کنید :

```
Dim lChild As Long
Dim lLast As Long

Do
    lLast = lChild
    lChild = FindWindowEx(lParent, lChild,
"ThunderRT6Textbox", vbNullString)
Loop While lChild
```

نکته : توسط تابع GetClassName می توانید نام کلاس سایر اشیا موجود در وی بی را بدست آورید .

۴ – چگونه می توان در وی بی عکسی از نوع BMP را مستقیماً بصورت JPG ذخیره کرد ؟

پاسخ : روشهای مختلفی برای اینکار وجود دارد که من برخی از آنها را لیست می کنم :

– استفاده از یک OCX/DLL به اسم PicFormat32 : با استفاده از این ابزار می توانید یک فرمت تصاویر را بهم تبدیل کنید . برای اطلاعات بیشتر به اینجا مراجع کنید .

– استفاده از کتابخانه Intel JPEG : برای اطلاعات بیشتر به اینجا و اینجا مراجعه کنید .

– استفاده از کتابخانه vic32.dll : برای اطلاعات بیشتر به اینجا مراجعه کنید .

با جستجو در سایت Google می توانید روشهای دیگری نیز پیدا کنید .

۵ – من قصد دارم یک برنامه دیکشنری بنویسم. از چه کنترلی استفاده کنم که بتوانم داده های مربوط به چند زبان را در آن فعال کنم و در همه ویندوز ها قابل رویت باشند. در ضمن میخواستم بدانم برنامه فاین ریدر قابلیت خواندن زبان فارسی را هم دارد یا نه. اگر نه آیا برنامه مشابهی که این قابلیت رو داشته باشد وجود دارد؟
پاسخ : در مورد ساخت برنامه های فارسی قبلاً مطلب نوشته ام اما در مورد سایر زبانها مطلبی پیدا نکردم . در مورد برنامه FineReader من تاکنون با این برنامه کار نکرده ام اما یک OCR فارسی به اسم واژه شناس در نمایشگاه کتاب امسال توسط شرکت هوش مصنوعی رایورز عرضه شده بود . برای اطلاعات بیشتر به سایت این شرکت به آدرس http://www.stonicasoft.com/ مراجعه کنید .

۶ – چطور میتوانم بین کامپیوتر و یک سخت افزار دیگر ارتباط برقرار کنم . مثلا یکLED را روشن و خاموش کنم ؟
پاسخ : برای اتصال کامپیوتر به یک سخت افزار می توانید از پورتهای سریال ، موازی و یا USB استفاده کنید . برای برنامه نویسی این پورتها نیز قبلاً مطالبی در وبلاگ نوشته ام که با مراجعه به آرشیو موضوعی می توانید آنها را بخوانید .

+ در مورد Log گرفتن از کیبرد به این آدرس مراجعه کنید .

پاسخ به سوالات -۱۲

– چگونه می توان شماره سریالی را که کارخانه روی هارد قرار می دهد را بدست آورد (یعنی شماره سریالی که با پارتیشن بندی مجدد هارد تغییز نکند و یک مشخصه منحصر بفرد برای هارد باشد) ؟
پاسخ : با دریافت این کد یک OCX خواهید داشت که با استفاده از آن می توانید شماره سریال هارد را که کارخانه سازنده روی آن قرار داده بدست آورید .
همچنین در این آدرس نیز یک برنامه دیگر بهمراه سورس کد زبان C برای اینکار وجود دارد .

۲– نحوه بدست آوردن سریال CDROM را توضیح دهید .
پاسخ : برای این منظور بایستی از تابع GetVolumeInformation استفاده کنید که قبلاً در بخش پرسش و پاسخها در مورد این تابع توضیح داده ام . برای اطلاعات بیشتر به این آدرس نیز مراجعه کنید .

۳ – چگونه می توان در وی بی تاریخ را بصورت شمسی از کاربر گرفت ؟
پاسخ : منظور شما از گرفتن تاریخ بصورت شمسی چیست ؟ شما می توانید یک texbox بگذارید تا کاربر تاریخ را بصورت یک String در آن وارد کند . اما اگر منظورتان تبدیل تاریخ میلادی به شمسی و یا تبدیل تاریخ شمسی به میلادی است (این مورد را یکی دیگر از دوستان نیز سوال کرده بودند) است به اینجا مراجعه کنید .

۴ – چگونه میتوان فرمی را کوچک کرد یعنی به حداقل رسانید و به جای اینکه به منوی Taskbar برود آیکون آن در کنار ساعت ظاهر شود و با کلیک رو آیکون آن منوی مورد نظر باز شود ؟
پاسخ : قبلاً در مورد قرار دادن آیکون برنامه در کنار ساعت ویندوز نوشته ام . به آرشیو موضوعی مراجعه کنید .

۵ – چگونه می توان از طریق وی بی روی یک فولدر یا فایل اجرایی قفل گذاشت که آن فولدر فقط با پسورد باز شود یا اینکه قابل کپی نباشد ؟
پاسخ : این کار امکان پذیر است چون برنامه هایی در این زمینه وجود دارد اما نیاز به دانش بسیار قویی در مورد Api Programming دارد . به سوال شماره ۹ مراجعه کنید .

۶ – چگونه می توان برنامه ای نوشت که محتویات فایل index.dat را که حاول آدرس سایتهای رفته شده است نشان دهد ؟
پاسخ : ابتدا بایستی ساختار و فرمت این فایل را بدانید . در این صورت با استفاده از ابزارهای کار با فایل در وی بی می توانید محتوای آنرا بخوانید .

۷ – چگونه می توان صدا را بعنوان ورودی گرفت و آنرا با یک کلمه مقایسه کرد ؟
پاسخ : اگر منظورتان تشخصی صحبت یا Speech Recognition است بایستی از موتور تشخیص صدای مایکروسافت که قابل دانلود از این آدرس می باشد استفاده نمایید . پس از نصب از موتور دو بخش به component های وی بی شما اضافه می شود : Microsoft Direct Speech Recognition و Microsft Direct Text-to-Speech . برای اطلاعات بیشتر در مورد این دو component به این آدرس مراجعه کنید

.

در این آدرس نیز اطلاعات مفیدی در مورد تشخصی صحبت وجود دارد .

۸ – چگونه می توان یکسری اطلاعات فارسی را از طریق صفحات ASP در پایگاه داده SQL Server ذخیره کرد ؟ آیا بایستی Collation دیتا بیس را تغییر دهم ؟

پاسخ : اولاً codepage صفحات Html خود را windows-1252 قرار دهید . ثانیاً codepage شی Session را نیز ۱۲۵۲ بگذارید . در اینصورت اطلاعات بصورت یونیکد در جدوال دیتا بیس ذخیره می شوند . همچنین با قرار دادن Collation بصورت arabic این کار امکان پذیر است .

۹ – آیا کتاب فارسیی در مورد برنامه نویسی API وجود دارد ؟

پاسخ : اخیراً دو کتاب فارسی در این زمینه چاپ شده است :

کتاب اول توسط انتشارات نص چاپ شده که به نظر من کتاب مفیدی است . کتاب دوم برنامه نویسی Api نیز توسط انتشارات ناقوس چاپ شده است .

پاسخ به سوالات-۱۳

– چگونه می توان برنامه ای نوشت که وقتی روی یکی از سیستمهای یک شبکه محلی LAN اجرا شد عملیات زیر را انجام دهد : – Domain فعال شبکه و کامپیوترهای آنرا پیدا کند ؟ – یک برنامه را در تمام کامپیوترهای شبکه کپی کند – فایلهای کپی شده را اجرا کند و اینکار بدون استفاده از winsock انجام شود ؟

پاسخ : بخش دوم و سوم مربوط بیشتر به بحث Hacking و exploit کردن سیستم ها بطوریکه بتوان فایلی را روی آنها اجرا کرد مربوط می

شود . این زمینه بهتر است به سایت http://www.tur2.com/ مراجعه کنید . اما در مورد قسمت اول سوال برای بدست آوردن domain شبکه و کامپیوترهای موجود در آن از توابع یک کتابخانه به اسم mpr.dll استفاده می شود . توابع مورد استفاده از این کتابخانه عبارتند از WNetOpenEnum، WNetEnumResource و WNetCloseEnum . برای اطلاعات بیشتر و دریافت یک نمونه برنامه به این آدرس مراجعه کنید .

۲ – چگونه می توان با کارت ویدیوی یا همان Capture توسط وی بی ارتباط برقرار کرد ؟ چگونه می توان با webCam توسط وی بی ارتباط برقرار کرد ؟
پاسخ : یکی از راهها استفاده از کنترلی به اسم ezVidCap می باشد . این کنترل را می توانید از این آدرس دریافت کنید .

در اینجا می توانید راهنمای استفاده از این کنترل را بخوانید .

در اینجا و اینجا نیز برنامه هایی نمونه برای کار با Video Capture وجود دارد .

در این آدرس نیز روش دیگری برای ارتباط با کارت ویدیویی بیان شده است .

۳ – چگونه می توان درایو (یا درایوهای) سی دی را توسط وی بی تشخیص داد ؟
پاسخ : برای اینکار از دو تابع از کتابخانه kernel32 به نامهای

GetLogicalDriveStrings و GetDriveType استفاده می شود . این دو تابع را بصورت زیر declare کنید :

```
Private Declare Function GetLogicalDriveStrings Lib
"kernel32" Alias "GetLogicalDriveStringsA" (ByVal
nBufferLength As Long, ByVal lpBuffer As String) As
Long

Private Declare Function GetDriveType Lib "kernel32"
Alias "GetDriveTypeA" (ByVal nDrive As String) As
Long
```

سپس ابتدا متغیر allDrives را که رشته ای ۶۴ کاراکتری از space است بصورت زیر تعریف کنید :

```
allDrives$ = Space$(64)
```

حال با استفاده از تابع GetLogicalDriveStrings لیست کلیه درایوهای سیستم را استخراج می کنیم :

```
ret& = GetLogicalDriveStrings(Len(allDrives$),
allDrives$)
allDrives$ = Left$(allDrives$, ret&)
```

حال با استفاده از یک حلقه و چک کردن درایوها با استفاده از تابع GetDriveType می توانیم تشخیص دهیم این درایو مربوط به سی دی است یا نه . برای اینکار اگر مقدار بازگشتی تابع به ازای یک درایو برابر عدد ۵ باشد آن درایو سی دی است .

```
Do
  pos% = InStr(allDrives$, Chr$(0))
   If pos% Then
   JustOneDrive$ = Left$(allDrives$, pos% - 1)
```

```
    allDrives$ = Mid$(allDrives$, pos% + 1,
Len(allDrives$))

    DriveType& = GetDriveType(JustOneDrive$)
        If DriveType& = 5 Then
          MsgBox UCase$(JustOneDrive$) & " is a CD
Drive"
        End If
    End If
Loop Until allDrives$ = ""
```

۴ – بجای استفاده از winsock32.ocx از چه روش دیگری می توان در برنامه نویسی شبکه استفاده کرد ؟

پاسخ : شما می توانید از کتابخانه Winsock.dll استفاده کنید و در اینصورت نیاز به فایل کمکی خاصی نیز ندارید . قبلاً در بخش پرسش و پاسخها ، لینکهایی در مورد این کتابخانه معرفی کرده ام و در صورتیکه فرصت کردم در مورد این کتابخانه توضیحاتی خواهم داد .

۵ – چگونه می توان مانند تروجان های معمول ، username و password یک اکانت را بدست آورید ؟

پاسخ : من تست نکردم اما به احتمال زیاد با استفاده از توابع کتابخانه RasApi قادر به اینکار خواهید بود . برای آشنایی با توابع این کتابخانه به این آدرس مراجعه کنید .

۶ – منظور از درایور دیتا بیس چیست ؟ آیا همان برنامه ای است که دیتابیس را با آن می نویسیم (مثلاً اکسس) ؟ چگونه می توان از یک فایل Text بجای بانک اطلاعاتی استفاده کرد ؟

پاسخ : درایور یک دیتابییس شامل یکسری توابع واسط برای اتصال و کار با آن دیتابیس است برای مثال با استفاده از داریور OLE JET DB می توان به دیتابیسهای اکسس متصل شد . زمانیکه شما یک برنامه

در وی بی می سازید که به یک دیتابیس متصل می شود برای اینکه بتوانید آنرا در کامپیوتر دیگری نیز اجرا کنید بایستی فایلهای درایور مربوطه نیز بهمراه آن کپی شوند . با استفاده از ابزار Package and Deployment Wizard می توانید فایلهای مورد نیاز برنامه را پیدا کنید و همچنین یک فایل Setup برای برنامه تان بسازید . در مورد فایلهای text نیز بله می توان اطلاعات جدول خود را در آن قرار دهید اما باید فرمت قرارگیری را خودتان تعریف کنید .

۷ – فایلهای اجرایی من حتی اگر بدون یک اکتیو ایکس هم باشد در کامپیوتری که فاقد ویژال بیسیک است اجرا نمی شود . چرا ؟
پاسخ : ساده ترین برنامه های وی بی نیز برای اجرا شدن احتیاج به کتابخانه Automation VB Runtime and OLE دارند . همانطور که در سوال قبل نیز توضیح داده ام می توانید با استفاده از برنامه های ساخت Setup مشکل خود را حل کنید .

۸ – من در حال نوشتن برنامه ای هستم که در آن امکان شماره گیری نیز می باشد. می خواستم بپرسم چگونه می توان اشغال بودن خط را تشخیص داد تا بتواند به طور اتوماتیک قطع و دوباره شماره گیری بنماید.؟
پاسخ : برای نوشتن برنامه تان از Ras Api استفاده کنید . این کتابخانه توابعی که مشکل شما را حل کند در خود دارد .

۹ – چگونه می توان برنامه ای نوشت که بتوان با آن سورس یک صفحه وب را با دادن آدرس آن صفحه مشاهده کرد ؟
پاسخ : برای اینکار component ای در وی بی به اسم Internet Transfer که قبلاً در مورد آن مطالبی نوشته ام وجود دارد . به آرشیو موضوعی سایت و بخش برنامه نویسی شبکه مراجعه کنید .

۱۰ - در مورد برنامه Multimedia Builder سایت یا کتاب معرفی کنید .

پاسخ : من با این برنامه زیاد کار نکرده ام اما با جستجو در Google می توانید سایتهای بسیار زیادی پیدا کنید . یک کتاب فارسی نیز در مورد این نرم افزار وجود دارد .

۱۱ – در برنامه های نمونه Msdn به ITLegacyCallMediaControl برخوردم . در مورد آن توضیح دهید .

پاسخ : این واسط یک از واسطهای TAPI می باشد که از برنامه هایی که بایستی بطور مستقیم با یک دستگاه ارتباط داشته باشند پشتیبانی می کند . برای اطلاعات بیشتر به اینجا مراجعه کنید .

۱۲ – چگونه می توان فایل مربوط به یک برنامه را به ویندوز شناساند بطوریکه وقتی روی آن فایل کلیک می شود ویندوز برنامه مربوط به آنرا اجرا کند ؟

پاسخ : اینکار با استفاده از نوشتن کلیدهای خاصی در رجیستری امکان پذیر است . قبلاً در بخش آموزش کار با رجیستری به این مطلب اشاره کرده و کلیدهای مربوط به آنرا معرفی کردم . به آرشیو موضوعی و بخش کار با رجیستری مراجعه کنید

پاسخ به سوالات-۱۴

+ قبل از مطرح کردن سوال خود ابتدا پرسش و پاسخهای قبلی را بخوانید . ممکن است به سوال شما و یا مشابه آن قبلاً پاسخ داده شده باشم .

+ به سوالات تکراری پاسخ داده نخواهد شد .

+ سوالات خود را یا فارسی و یا انگلیسی بفرستید و از نوشتن پینگلیش خودداری کنید .

۱ – من چند تا جدول SQL دارم که باید با علامت خوردن هر دکمه رادیویی یکی از جدولها به فرم من متصل شود . در حقیقت نیاز به سوئیچ کردن میان کانتنت ها دارم . از نظر تئوری کار انجام شده و در هنگام اجرا نیز خطایی گرفته نمی شود اما در عمل هیچ داده ای وارد نمی شود مشکل کجاست ؟
پاسخ : در مورد روند برنامه خود بیشتر توضیح دهید و در صورت امکان سورس برنامه تان را برای من بفرستید .

۲ – برنامه هایی مثل WinRar را چگونه فارسی می کنند ؟ چگونه می توان برنامه هایی نوشت که مستقل از ویندوز فارسی باشند ؟
پاسخ : برنامه هایی وجود دارند که اجازه می دهند با ایجاد یکسری فایل Setting برای آنها بتوان یک زبان جدید را به آن اضافه کرد . بعنوان مثال شما معادل هر یک از منوها یا دکمه های برنامه را به زبان مورد نظرتان در یک فایل تنظیمی قرار می دهید و آنگاه در بخش تنظیمات می توانید زبان مورد نظر را انتخاب کنید . در مورد ساخت برنامه های فارسی مستقل از سیستم عامل قبلاً مطالبی نوشته ام . به آرشیو موضوعی مراجعه کنید .

۳ – آیا می توان در بانک های اطلاعاتی اکسس توسط ویژوال بیسیک عکس اضافه کرد ؟

پاسخ : بله این امکان وجود دارد . در این مورد در بخش پرسش و پاسخهای قبلی توضیح داده ام .

۴ – از دو قسمتی که در مورد WinSock نوشته بودید برای نوشتن یک شبه پراکسی استفاده کردم اما در اولین قدم که اتصال IE به سرور بود ناموفق ماندم . آیا می توان با این کنترل یک پراکسی سرور درست کرد که در یک شبکه اطلاعات را از یک IP بگیرد و به یک IP دیگر رد کند ؟
پاسخ : بله این امکان وجود دارد . شما ابتدا بایستی با دستورات پروتکل HTTP آشنا باشید زیرا IE در خواست های خود را به پروکسی سرورها توسط این پروتکل می فرستد . شما بایستی برنامه ای بنویسید که روی یک پورت خاص به تقاضاهای HTTP گوش داده و پاسخ دهد . از طرف دیگر بایستی در IE آدرس IP و شماره پورت پروکسی را بدهید تا IE درخواست های خود را به آن بفرستد .
در این آدرس یک نمونه برنامه پروکسی سرور که به زبان وی بی نوشته شده وجود دارد .

۵ – من می خواهم یک گزارش ایجاد کنم که اطلاعات آن از دو جدول بایستی استخراج شود . چگونه این کار را انجام دهم ؟
پاسخ : بایستی یک Query برای Select کردن اطلاعات مربوطه بنویسید که این Query را برای تولید گزارش استفاده کنید .

۶ – چگونه می توان در وی بی یک متغیر را از یک فرم به فرم دیگر فرستاد ؟
پاسخ : منظورتان از فرستادن متغیر چیست ؟ شما با تعریف متغیرها بصورت Public ، در هر فرمی می توانید به آنها دسترسی داشته بشید برای مثال اگر متغیر Test را در فرمی به اسم Form1 تعریف کرده باشید با دستور Form1.Test در هر فرمی به آن دسترسی دارید .

۷ – چگونه می توان توسط وی بی ایمیلی فرستاد که به آن عکس نیز اضافه شده باشد ؟

پاسخ : شما بایستی از کامپوننت هایی که برای اینکار طراحی شده اند استفاده کنید . یک نمونه از این کامپوننت ها را می توانید از اینجا دریافت کنید .

۸ – چگونه می توان در VB.Net با استفاده از XML یک فرم را بصورت Runtime ایجاد کرد ؟

پاسخ : برای اینکار شما ابتدا بایستی اطلاعات فرم را در یک فایل XML قرار دهید . سپس اطلاعات چگونگی نمایش فرم را در یک فایل XSL قرار دهید . حال با استفاده از شی XML که در نوار ابزار IDE وجود دارد می تونايد با دادن این دو فایل ، فرم مورد نظرتان را ایجاد نمایید . همچنین می توانید کدی بنویسید که بتوانید با آن اطلاعات فایل XML مربوطه را کم و زیاد کنید .

۹ – چگونه می توان آیکون یک فایل EXE را با استفاده از برنامه نویسی عوض کرد ؟

پاسخ : باید با استفاده از برنامه نویسی رجیستری اینکار را انجام دهید . برای اطلاعات بیشتر به اینجا مراجعه کنید .

۱۰ – من با استفاده از یک برنامه ساخت فایل Help ، یک فایل chm ساخته ام . با توجه به اینکه با تابع Shell در برنامه وی بی این فایل راهنما باز نمی شود چگونه می توانم آنرا با منوی help ای که در برنامه ام ایجاد کردم ام باز کنم ؟

پاسخ : قبل از پاسخ دادن به این سوال باید بگویم که فایلهای help را می توانید با برنامه ای مثل Microsoft Help Workshop ایجاد کنید .

پس از آن برای نظیر کردن یک برنامه به یک فایل راهنما می توانید از خاصیت HelpFile شی App بصورت زیر استفاده کنید :

```
App.HelpFile = App.Path & "\test.chm"
```

در آنصورت در زمان اجرای برنامه کاربر با زدن دکمه F1 می تواند فایل help شما را ببیند .

برای اختصاص دادن یک فایل راهنما به منویتان بایستی از خاصیت HelpContextID استفاده نمائید . برای مثال برای اختصاص دادن یک فایل راهنما که ContextID آن برابر ۲۳۴۵۶ است به منویی به اسم MenuHelp از دستور زیر استفاده کنید :

```
MenuHelp.HelpContextID = 23456
```

برای اطلاعات بیشتر به اینجا مراجعه کنید .

معرفی سایت

معرفی سایت

سایت راهنمای ساخت بازیهای کامپیوتری : ورود به سایت

برخی از مطالب این سایت :

ایده های ساخت بازیهای کامپیوتری

پیاده سازی ایده ها

تکنیکهای برنامه نویسی

چند مثال

ame Programing

معرفی کتاب

Hack Proofing Your Netowrk ویرایش دوم

نویسنده : Ryan Russell

یکی از بهترین کتابها در زمینه Security و هک
در این کتاب مباحثی چون مبانی و اصول هک ، دسته بندی روشهای هک ، Buffer Overflow، Sniffing ، Session Hijacking، Spoofing، Tunneling ، رمز نگاری و چند مبحث جالب دیگر ذکر شده است .
فایل PDF کل کتاب را می توانید از آدرس زیر دریافت کنید (سایز فایل ۷,۴۵MB)
Proofing Your Network SE Hack

معرفی برنامه

+ دوست عزیزم رضا زحمت کشیدند و لینک نرم افزار ActiveSkin را بهمراه لینک کرک آن برای من فرستادند . توسط لینکهای زیر می توانید آنها را دانلود کنید :

دریافت نرم افزار (۲,۳۲ مگابایت)

دریافت کرک برنامه (۵,۸ کیلوبایت)

- **سایت VB4ir ، انجمن اختصاصی برنامه نویسان ویژوال بیسیک .**

معرفی سایت

سایت پرشیا پکت : یک سایت آموزشی واطلاع رسانی خصوصا در زمینه شبکه و راهبری آن .

مطالب این سایت عبارتند از :

اخبار کامپیوتری : جدید ترین اخبار تخصصی مربوط به اینترنت و شبکه

های کامپیوتری، جدیدترین نرم افزارهای کاربردی، اخبار جدید در مورد ویروس ها، آخرین استاندارد فارسی در اینترنت و...

بخش تخصصی Cisco و شبکه : آخرین اطلاعات در زمینه تکنولوژی مدرن و محصولات جدید شرکت های بزرگ شبکه ای دنیا، بخصوص (CISCO) را میتوانید در این بخش بیابید. همچنین میتوانید در این بخش از آخرین اطلاعات و تغییرات مربوط به آزمونهای مختلف مورد تائید شرکت Cisco مطلع شوید. معرفی مقالات منتشره و همچنین سایتهای تخصصی در زمینه شبکه از دیگر قسمتهای این بخش می باشند

کتابخانه تخصصی پرشیاپکت : مقالات تخصصی و نظری در زمینه های مختلف شبکه (طراحی، پیاده سازی و مدیریت شبکه) و مباحث آموزشی Cisco از بخشهای خواندنی این کتابخانه میباشند .

معرفی سایت

MazSoft.com
Share the Knowledge

سایت برنامه نویسی حرفه ای ویژوال بیسیک : در این سایت برخی از نکات آموزشی MSDN بصورت ترجمه شده به فارسی و با فرمت PDF قرار داده شده است .

برخی از مباحث این سایت :

۱ – برنامه نویسی API در ویژوال بیسیک
۲ – کار با فایلها در ویژوال بیسیک
۳ – کار با پنجره در ویژوال بیسیک
۴ – کار با منو در ویژوال بیسیک
۵ – مالتی مدیا
۶ – چاپ
۷ – ساخت برنامه های کاربردی
۸ – ساخت کنترلهای ActiveX
۹ – مباحث پیشرفته در مورد کنترل TextBox
۱۰ – استفاده از امکانات C در ویژوال بیسیک

معرفی سایت

سایت vbAccelerator دارای نکات و مطالب آموزشی برای برنامه نویسان حرفه ای ویژوال بیسیک می باشد .

معرفی سایت

سایت vbip یکی از بهترین سایتهای آموزش برنامه نویسی شبکه و اینترنت در ویژوال بیسیک می باشد . از جمله مباحث آموزشی این سایت عبارتند از :

۱ – آموزش کار با Microsoft Winsock ActiveX Control

۲ – آموزش کار با Windows Sockets API

۳ – آموزش کار با Win32 Internet API

۴ – آموزش کار با Internet Protocol Helper API

۵ – آموزش کار با Internet Transfer Control Microsoft

۶ – آموزش کار TAPI و RASAPI

معرفی سایت

شامل نکات آموزشی ، tutorial ها ، اخبار و کدهای مربوط به سی ++ ، سی # ، VB.NET و ویژوال بیسیک

معرفی سایت

بهترین منبع برای بدست آوردن اطلاعات در مورد برنامه نویسی API توسط ویژوال بیسیک . شامل قسمتهای :

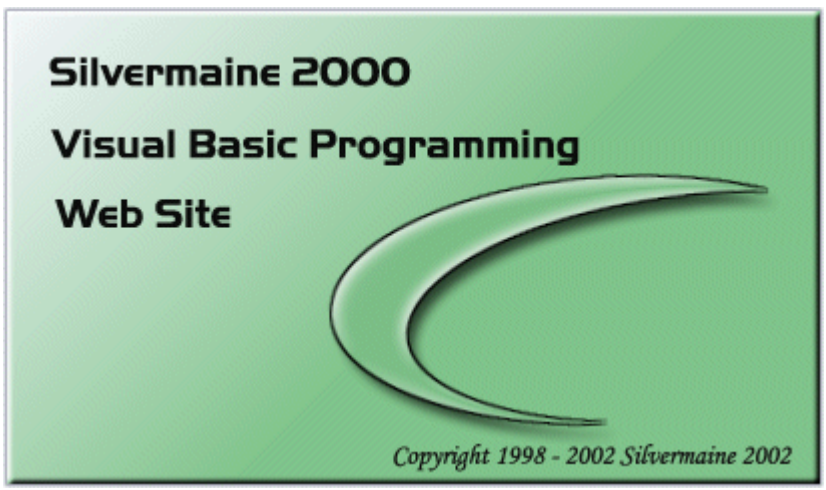
۱ – راهنمای API

۲ – لیست API ها

۳ – Tutorial های مربوط به API Programming

۴ – مثالهایی از Programming API در ویژوال بیسیک

معرفی سایت



سایت SilverMain یکی از سایتهای خوب آموزشی برای ویژوال بیسیک بوده و شامل source code ، معرفی سایتهای برنامه نویسی و برنامه های کامل نوشته شده با ویژوال بیسیک می باشد

معرفی سایت

سایتی با ۲۳۷۵ نمونه کد و مطلب آموزشی ویژوال بیسیک که هر روز update می شود . شامل بخشهای برنامه نویسی در زمینه های Asp ، بانکهای اطلاعاتی ، مالتی مدیا و موزیک ، بازی ، شبکه و اینترنت ، گرافیک ، برنامه نویسی سیستم و API و VB.NET .

+ ایجاد یک FTP Client با استفاده از ویژوال بیسیک

.لینک ...

+ ایجاد یک اسکنر پورت با WinSock . لینک ...

معرفی سایت

http://sheidaian.persianblog.com

http://sheidaian.persianblog.com

http://www.freevbcode.com

http://www.allapi.net

http://www.codeproject.com

http://www.silvermaine.co.uk

http://www.vbcode.com

http://max3d.3dluvr.com

http://www.3dcafe.com

http://www.vbip.com

http://www.vbaccelerator.com

http://www.coderoom.com
http://www.vbaccelerator.com

http://www.vbcity.com

http://www.iranvig.com

http://vb.blogfa.com

آشنایی با نرم افزار PE Explorer

بررسی کلی

PE Explorer یک مجموعه کامل از ابزارهای چند منظوره برای کاربا

فایلهای PE می باشد . این مجموعه دارای بخشهایی چون مشاهده گر سرآیند PE ، مشاهده گر API های import و export شده ، جستجوگر سینتکس توابع API ، مشاهده گر/ تغییر دهنده منابع ، اسکنر وابستگی و دی اسمبلر است .
در بالای همه این امکانات ، این برنامه دارای یکی از بهترین بازکننده (Unpacker) های UPX است . با کمک این مجموعه شما قادر خواهید بود تا ساختار درونی فایلهای PE را مشاهده ، ویرایش و تعمیر کنید . این مجموعه به شما امکان می دهد تا ساختار درونی فایلهای PE را آنالیز نموده و تغییر دهید ، checksum ها را تصحیح نمائید ، منابع آسیب دیده را تعمیر کنید ، کدهای مخرب در برنامه ها را تشخیص دهید ، بتوانید header و جداول موجود در فایلها را تغییر دهید ، مهندسی معکوس را برای سورس کدهای گمشده انجام دهید ، import ها و export های DLL های استاندارد را بیابید ، منابع اطلاعاتی درونی فایلها را تغییر دهید و ...

فایل PE چیست ؟

فایلهای (PE (portable executable ، فایلهای باینری اجرایی در ویندوزهای ۳۲ بیتی می باشد (مانند DLL ها ، درایور ها و برنامه های اجرایی) . مجموعه PE Explorer قادراست تا انواع گوناگونی از فایلهای PE را مدیریت کند مانند : EXE، DLL، SYS، DRV، CPL ، OCX ، SCR و غیره .

برخی از ابزارهای موجود در PE Explorer

۱ - ویرایشگر منبع *Resource Editor*

Explorer PE دارای یکی از بهترین و کاراترین ویرایشگرهای منبع می باشد . شما توسط این بخش براحتی می توانید منابع فایلهای اجرایی را از درون آنها بدون نوشتن هیچگونه اسکریپتی جستجو کرده و تغییر دهید .

۲ - *Function Syntax Lookup* جستجوگر سینتکس تابع

در زمان مشادهد توابع موجود در یک فایل PE با کلیک روی آن تابع می توان سینتکس آن تابع را مشاهده کرد . پارامترهای تابع و مقدیر بازگشتی آن نمایش داده می شوند . همچنین می توان با انتخاب یک تابع ، می توان کامنتهایی را به تابع اضافه کرده و یا جزئیات را تغییر داد .

۳ - *Dependency Scanner* اسکنر وابستگی

این ابزار به شما اجازه می دهد تا بطور بازگشتی ، تمام ماژوال های لینک شده به یک فایل PE را اسکن کنید . اسکنر وابستگی همچنین وابستگی های delay-load را تشخیص می دهد . این نوع از وابستگی در Visual C++ 6.0 معرفی شده است .

۴ - *DisAssembler* دی اسمبلر

این ابزار قادراست سورس اسمبلی اکثر برنامه های نوشته شده با کامپایلرهای مختلف را استخراج نماید .

دریافت نرم افزار

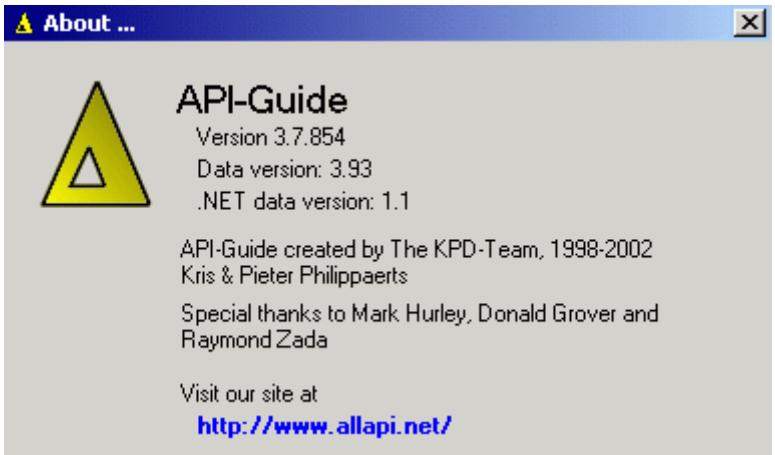
آخرین نسخه این مجموعه (ورژن ۱,۹۴) را می توانید از اینجا دریافت کنید .

کرک این برنامه را از اینجا می توانید دریافت کنید .

معرفی سایت

بزرگترین بانک اطلاعاتی کدهای برنامه نویسی در اینترنت با ۵۸۰۹۰۴۰ خط کد ، مقاله و مطلب آموزشی در زبانهای ویژوال بیسیک ، ویژوال سی ، دلفی ، Asp، SQL، Java، Perl، PHP

معرفی برنامه



برنامه Api-Guide جزو یکی از بهترین منابع و برنامه های آموزشی برای برنامه نویسی با API های ویندوز در ویژوال بیسیک می باشد . این برنامه دارای یک API-Database است که شامل حدود ۹۰۰ تابع API و همراه با مثالهای کامل می باشد . شما می توانید از لیست الفبایی و یا از لیست موضوعی ، توابع API مورد نظر خود را انتخاب کنید . در نسخه جدید این برنامه مثالهایی برای VB.NET نیز قرارداده شده است . همچنین database برنامه را می توانید از طریق اینترنت Update نمائید .

این برنامه را می توانید از محل زیر دریافت کنید :

API-Guide

معرفی سایت

سایت vbcode دارای کدهای متنوعی در زمینه های مختلف برنامه نویسی است مانند :

Internet، Game، Registery، Database، Call API،

Music/Sound، Graphic، ActiveX و غیره

معرفی سایت

سایت MAX3D : آرشیوی از tutorial ها و plugin های نرم افزار ۳ D Studio Max

سایت DCafe۳ : آرشیوی از tutorial ها ، article ها ، texture ها ، plugin ها و اطلاعات خبری برای نرم افزار های مدلسازی سه بعدی

به نام اهورای زیبایی

باسلام خدمت دوستان گرامی این کتاب الکترونیک برگرفته از سایت http:// sheidaian.persianblog.com می باشد. که به علت نداشتن زمان کافی ویرایشی در آن صورت نگرفته است . از تمامی علاقه مندان در خواست میشود برای که مرا در این امر یاری گردانند تا این کتاب به صورت اولین کتاب الکترونیک فارسی درزمینه ویژوال بیسیک به صورت رایگان در اختیار علاقه مندان قرار گیرد. برای راهنمایی بیشتر بااین این ایمیل تماس بگیرید saeed.vb@Gmail.com با تشکر ازتمامی دوستان و همراهان.