

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

مسعود بداخانیان

نویسنده :

آدرس وبلاگ :

MasoudLearning.Blogfa.ir

masoud_lr@yahoo.com

پست الکترونیک :

masoud.bedakhanian@gmail.com

وبلاگ آموزشی با هر موضوعی که فکرشو می کنی بخصوص کامپیوتر و نرم افزار برای دانلود و...

و البته هر مطلب، برنامه یا موضوعی که شما بخواهید.

فصل اول

آشنایی با محیط ویژوال بیسیک

شیء (object) : اشیاء یک برنامه ویژوال بیسیک را تشکیل می دهند هر شیء دارای سه جنبه زیر است :

۱: خاصیت ها (properties):

خاصیت ها حاوی مشخصات شیء هستند مثلا رنگ زمینه پنجره در خاصیت `backcolor` و عنوان پنجره در خاصیت `caption` قرار دارد که با تغییر مقدار این خاصیت ها مشخصات شیء هم تغییر می کند .

۲: رویدادها (events) :

(keypress) : حوادثی هستند که معمولا توسط کاربر رخ میدهند مانند فشردن کلیدی از صفحه کلید .

برای هر رویداد شیء زیر برنامه ای وجود دارد مثلا برای رویداد `click` روی کلید `command1` زیر برنامه ای به شکل زیر وجود دارد :

```
Sub command1_click()
```

```
end sub
```

با کلیک روی `command1` زیر برنامه فوق اجرا می شود و اگر دستوراتی درون آن تایپ شود آن دستورات اجرا میشود

که باید کد ها را در بین کدهای فوق قرار داد .

۳: متدها (methods) : باعث انجام رفتاری در شیء می شود

مثلا متد `move` باعث حرکت در شیء و متد `newpage` باعث تعویض کاغذ در چاپگر می شود

مراحل ایجاد یک برنامه در **basic visual** :

۱: مرحله طراحی : در این مرحله اشیاء مورد نیاز را از جعبه ابزار

انتخاب کرده درون فرم قرار می‌دهیم سپس در پنجره **properties** خاصیت های آن شیء (مشخصات) را تغییر می‌دهیم

۲: مرحله کد نویسی : در این مرحله رفتار شیء را تعیین میکنیم

برای این کار روی شیء مورد نظر دابل کلیک کرده تا ناحیه کدنویسی ظاهر شود

درون رویداد مورد نظر دستورات لازم را تایپ میکنیم

مثال : در برنامه زیر پس از تایپ عبارتی درون شیء **text1** و با فشردن کلید **command1** عبارت تایپ شده در عنوان فرم قرار می‌گیرد:

سمت چپ برنامه ویژوال بیسیک پنجره ای وجود دارد که حاوی چندین ابزار است به آن **toolbox** میگویند , در آن میتوانید **textbox** را انتخاب کنید و روی فرم رسم کنید

(لازم به ذکر است که ابزار های درون **toolbox** را وقتی بر روی فرم رسم می‌کنیم به آن شیء کنترل میگویند)

ابتدا اشیاء را درون فرم قرار می‌دهیم(فرم همان صفحه طراحی است)

سپس **command1** را انتخاب کرده و در جدول **properties** خاصیت **caption** عبارت "نمایش" را تایپ میکنیم

شیء **text1** را انتخاب کرده و خاصیت **text** آن را پاک میکنیم

در ناحیه کدنویسی در رویداد **command1_click** دستور زیر را تایپ میکنیم

```
form1.caption=text1.text
```

مثلا میخواهیم برنامه ای بنویسیم که اگر کاربر روی کلید کلیک کند از برنامه خارج شود

اول کامندی در فرم ایجاد میکنیم و بعد کپشن آن را به خروج تبدیل میکنیم و بعد روی کلید دوبار کلیک میکنیم تا ناحیه کد نویسی آن پیدا شود در ناحیه کد نویسی فقط **end** را تایپ می‌کنیم و بعد

کلید **f5** را فشار میدهیم تا برنامه اجرا شود میبینیم که اگر روی کلید خروج کلیک کنیم برنامه از حالت اجرا خارج خواهد شد.

اجرای برنامه با کلید **f5** یا از منوی **run/start**

فصل دوم

روش های تغییر یک خاصیت

۱ : هنگام طراحی برنامه پس از انتخاب شی در پنجره **properties** خاصیت مورد نظر را یافته مقدار آن را تغییر میدهیم

مثال : تغییر خاصیت **caption** فرم به **iran** :

```
Form1.caption="iran"
```

نکته : بعضی از خاصیت ها فقط هنگام طراحی و بعضی دیگر فقط در حین اجرای برنامه قابل تغییرند.

خاصیت **name** (برای همه اشیاء) : نام شی در این خاصیت قرار میگیرد , این خاصیت در حین اجرای برنامه قابل تغییر نیست

خاصیت **caption** (برای اکثر اشیاء) :عنوان شی در این خاصیت قرار میگیرد.

رویداد **click** (برای همه اشیاء) : با کلیک روی شی این رویداد رخ می دهد

رویداد **dblclick** (برای فرم) : دابل کلیک روی فرم باعث رخ دادن این رویداد می شود

مثال : در برنامه زیر با کلیک روی فرم عنوان آن با کلیک روی فرم عنوان آن به **hello** و دابل کلیک روی فر عنوان آن به **bye** تغییر میکند:

Form1_click :

```
Form1.caption="hello"
```

Form1_dbclick :

```
Form1.caption="bye"
```

خاصیت **backcolor** (برای فرم) : رنگ زمینه فرم در این خاصیت نگهداری می شود .

شیء **text** : از این شیء برای دریافت یک عبارت از کاربر یا نمایش مقداری به کاربر استفاده می شود .

خاصیت **text** (برای **text**) : متن درون شیء **text** در این خاصیت نگهداری می شود

مثال: در برنامه زیر با فشردن کلید **command1** متن تایپ شده درون شیء **text1** در عنوان فرم ظاهر شده و متن درون **text** پاک می شود :

ابتدا خواصیت های زیر را تغییر می دهیم :

برای شیء **text1** :

```
text1=""
```

برای شیء **command1** :

```
Caption= نمایش
```

در رویداد **command1_click** :

```
Form1.caption=text1.text
```

```
Text1.text=""
```

رویداد **change** (برای **text**): با هر گونه تغییر در محتویات **text** این رویداد رخ می دهد .

مثال : در برنامه زیر پس از تایپ دو عدد درون **text1** و **text2** و فشردن **command1** مجموع آن دو عدد درون **text3** ظاهر می شود:

در رویداد **command1_click** :

```
Text3.text=val(text1.text)+val(text2.text)
```

خاصیت **fontsize** (برای **text**): اندازه قلم متن درون **text1** در این خاصیت قرار دارد .

مثال: در برنامه زیر فشردن **command1** باعث بزرگ شدن اندازه قلم **text1** و فشردن **command2** باعث کوچک شدن اندازه قلم **text1** میشود :

در رویداد **command1_click** :

```
text1.fontzise =text1.fontzise+2
```

در رویداد **command2_click** :

```
text1.fontzise =text1.fontzise-2
```

خاصیت **Enabled** (برای همه اشیاء) : اگر این خاصیت برابر با **false** قرار داده شود شی غیرفعال می شود .

خاصیت **tooltiptext** (برای همه اشیاء): متنی هنگام مکت روی شی ظاهر می شود .

خاصیت **locked** (برای همه اشیاء): اگر این خاصیت را برابر با **true** قرار دهیم کادر **text** قفل شده و کاربر نمی تواند محتویات آن را تغییر دهد

خاصیت **maxlenght** (برای **text**): حداکثر تعداد کاراکترهایی که میتوانند درون **text** تایپ شوند در این خاصیت قرار می گیرند .

شیء **option** (دکمه رادیویی): برای انتخاب یک گزینه از میان چند گزینه به کار می رود .

خاصیت **value** (برای **option**): اگر شیء **option** انتخاب شود خاصیت فوق برای **true** می شود

مثال: در برنامه زیر به کمک سه شیء **option** میتوان اندازه قلم **text1** را تغییر داد :

در رویداد **option1_click** :

```
if option1.value=true then
```

```
text1.fontSize=8
```

```
end if
```

در رویداد **option2_click** :

```
if option2.value=true then
```

```
text1.fontSize=16
```

```
end if
```

به همین ترتیب شما میتوانید شیء **option** را زیادتر کنید و شماره فونت بیشتری به آن بدهید .

شیء **frame**: برای دسته بندی اشیاء روی فرم از این شیء استفاده می کنیم و در صورتی که **frame**

را غیرفعال کنیم اشیاء درون آن نیز غیر فعال میشوند .

شیء **check**: از این شیء برای انتخاب یا لغو انتخاب یک گزینه استفاده می شود.

خاصیت **value** (برای **check**): مقدار این خاصیت وضعیت انتخاب شیء **check** را مشخص می کند

که میتواند یکی از سه مقادیر زیر باشد

1: **checked**: شیء **check** انتخاب شده است

۲: unchecked : شیء check انتخاب نشده است.

۳: grayed : شیء check در وضعیت بین انتخاب و عدم انتخاب قرار دارد .

خاصیت **TabStop** (برای همه اشیاء) : اگر مقدار این خاصیت برابر با **false** قرار دهیم در حین اجرای برنامه به کمک کلید **tab** شیء مزبور فعال نمی شود .

خاصیت **TabIndex** (برای همه اشیاء): عدد درون این خاصیت ترتیب جابجا شدن بین اشیاء برنامه با کلید **tab** را معین میکند

اولین شیء دارای مقدار صفر و سومین شیء دارای مقدار دو برای خاصیت فوق می باشد .

اشیاء **image** و **picture** : از این دو شیء برای نمایش فایل های تصویری در برنامه استفاده می شود .

خاصیت **picture** (برای **image** و **picture**) : نام و مسیر فایل تصویری مورد نظر در این خاصیت قرار می گیرد .

خاصیت **loadpicture** (برای **image**): از این خاصیت برای تعویض عکس در موقع اجرای برنامه استفاده میشود .

خاصیت **stretch** (برای **image**) : اگر این خاصیت برابر با **true** قرار داده شود تصویر انتخاب شده به اندازه آن در می آید

خاصیت **visible** (برای همه اشیاء) : اگر این خاصیت را برابر با **false** قرار دهیم شیء پنهان میشود

مثال: در برنامه زیر با هر با فشردن کلید **command1** تصویر درون **image** ظاهر و پنهان می شود :

به خاصیت **picture** برای **image** تصویر مورد نظر را اختصاص می دهیم

در رویداد **command1_click** :

If image1.vizable=true then


```
Image1.vizable=false
```

```
Else
```

```
Image1.vizable=true
```

```
End if
```

شیء **timer** : از این شیء برای زمان بندی حوادث برنامه استفاده می شود.

خاصیت **interval** (برای **timer**) : زمان کار **timer** بر حسب میلی ثانیه در این خاصیت قرار داده می شود.

رویداد **timer** (برای شیء **timer**) : با اتمام زمان کار **timer** این رویداد رخ می دهد .

مثال : در برنامه زیر در هر ثانیه تصویر درون **image1** ظاهر و پنهان می شود:

در خاصیت **picture** برای **image1** تصویر مورد نظر را قرار داده در خاصیت **interval** برای **timer1** عدد ۱۰۰۰ را وارد میکنیم

```
در رویداد timer1_timer :
```

```
If image1.vizable=true then
```

```
image1.vizable=false
```

```
else
```

```
image1.vizable=true
```

```
end if
```

مثال ۲ : نشان دادن ساعت در عنوان فرم :

ابتدا خاصیت **interval** تایمر را ۱۰۰۰ اختصاص می دهیم و بعد در رویداد **timer1_timer** :

```
Form1.caption=time
```

توجه برای تعریف متغیر سراسری در بالای ناحیه کدنویسی (general) از تابع `dim` استفاده می کنیم

و تا بخواهیم می توانیم تعریف کنیم:

شیوه ساخت متغیر سراسری :

نوع `as` نام متغیر `Dim`

مثال `Dim y as Byte`

برای تعریف متغیر سراسری می توان به جای `dim` از `public` استفاده نمود این دو از نظر کارکرد هیچ تفاوتی با هم ندارند.

تابع `msgbox` : به کمک این تابع می توان پیغامی را به کاربر نمایش داده و پاسخ وی را دریافت کرد.

(عنوان پنجره , عبارت تعیین کننده نوع و تعداد کلید , "پیام") `msgbox` = متغیر

عبارت تعیین کننده نوع و تعداد کلید ها از مقادیر زیر تشکیل می دهد :

(گروه اول)

مفهوم	مقدارها	ثابتها
نمایش کلید <code>ok</code>	۰	<code>Vbokonly</code>
نمایش کلید های <code>ok</code> و <code>cancel</code>	۱	<code>Vbokcancel</code>
نمایش کلیدهای <code>abort</code> و <code>retry</code> و <code>ignore</code>	۲	<code>Vbabortretryignore</code>
نمایش کلیدهای <code>yes</code> و <code>no</code> و <code>cancel</code>	۳	<code>Vbyesnocancel</code>
نمایش کلیدهای <code>yes</code> و <code>no</code>	۴	<code>Vbyesno</code>
نمایش کلیدهای <code>cancel</code> و <code>retry</code>	۵	<code>vbretrycancel</code>

(گروه دوم)

مفهوم	مقدارها	ثابتها
نمایش علامت حالت بحرانی	۱۶	Vbcritical
نمایش علامت سوال	۳۲	vbqustion
نمایش علامت هشدار	۴۸	vbexclanation
نمایش علامت اطلاعات	۶۴	vbinformation

(گروه سوم)

مفهوم	مقدارها	ثابتها
کلید اول پیش فرض	۰	Vbdefaultbutton1
کلید دوم پیش فرض	۲۵۶	Vbdefaultbutton2
کلید سوم پیش فرض	۵۱۲	Vbdefaultbutton3
کلید چهارم پیش فرض	۷۶۸	Vbdefaultbutton4

عبارت تعیین کننده نوع و مقدار کلید ها :

از مجموع یکی از مقادیر گروه اول با یکی از مقادیر گروه دوم با یکی از مقادیر گروه سوم تشکیل می شود

A=MsgBox("you want exit program.?", Vbdefaultbutton2 , vbqustion , Vbyesno)

یا

A=MsgBox(" you want exit program.?", ۲۵۶+۳۲+۴)

و بالاخره مقادیری که توسط تابع برگردانیده می شوند طبق جدول زیر می باشد :

مفهوم	مقدارها	ثابتها
فشاردن کلید ok	۱	Vbok
فشاردن کلید cancel	۲	Vbcancel
فشاردن کلید abort	۳	Vbabort
فشاردن کلید retry	۴	Vbretry
فشاردن کلید ignore	۵	vbignore
فشاردن کلید yes	۶	vbyes
فشاردن کلید no	۷	vbno

مثال: در برنامه زیر با فشاردن کلید **command1** پنجره پیام در صورت تایید کاربر برنامه خاتمه می یابد :

در رویداد **command1_click**

```
a=msgbox(" you want exit program ",4+32+256)
```

```
if a=6 then end
```

تابع **inputbox** : از این تابع برای دریافت اطلاعات از کاربر استفاده می شود .

([جواب پیش فرض],[عنوان پنجره],[پیام] = **inputbox** = متغیر رشته ای)

مثال: در برنامه زیر پس از فشاردن **command1** پنجره ای باز شده و عنوانی از کاربر دریافت می کند سپس عنوان دریافتی در عنوان پنجره قرار می گیرد :

اول یک متغیر سراسری به نام **a** درست می کنیم

در رویداد **command1_click**

```
a=inputbox(" type")
```

```
form1.caption=a
```

اشیاء آرایه ای : همانند متغیرهای معمولی میتوان برای اشیاء نیز آرایه ایجاد کرد

و اشیاء آرایه ای شامل تعدادی شیء هممنوع و هم نام می باشند که با یک عدد یا یک اندیس از هم متمایز می شوند.

برای ایجاد اشیاء آرایه ای اولین شیء را بروی فرم قرار داده خاصیت های لازم را مقدار دهی کرده آنگاه آن را با `copy` و `paste` به مقدار مورد نیاز تکرار می کنیم.

(اندیس اولین شیء آرایه برابر با صفر می باشد) اشیاء آرایه ای دارای رویدادهای مشترک می باشند.

در این حالت رویدادها دارای پارامتری به نام `index` می باشد که شماره شیء ای که رویداد برای آن رخ داده است را در خود نگهداری می کند

به عنوان مثال اگر آرایه ای از کلیدها به عنوان `cmd` و به مقدار ۱۰ کلید ایجاد کنیم اعضای آرایه عبارتند از `cmd(9)` . . . تا `(cmd(1) , cmd(0)` و رویداد `click` برای همه آنها به شکل
`sub cmd_click(integer)` روبه رو است :

مثال در برنامه زیر که حاوی یک بازی ساده می باشد پنج تصویر در نقاط مختلف فرم ظاهر و پنهان می شود:

اگر کاربر بتواند هر یک از آنها کلیک کند امتیازی به وی تعلق میگیرد:

درون فرم آرایه ای از `image` ها با نام `img` و به مقدار ۶ عنصر ایجاد میکنیم و درون هر کدام تصویر متفاوتی قرار میدهیم
مقدار دهی خاصیت ها :

برای همه `img` ها بجز (`img0`)

`visible=false`

برای همه `img` ها :

`stretch=true`

برای `timer1` :

`interval=300`

`enabled=false`

در بالای ناحیه کد نویسی (`general`) متغیر سراسری رو به رو را تعریف می کنیم : `Dim a`

در رویداد `command1_click` :

`If timer1.enebeled=false then`

`Timer1.enebeled=true`

`Form1.caption= 0`

`a=0`

```
command1.caption stop = ""
```

```
else
```

```
timer1.enebled=false
```

```
command1.caption="start "
```

```
end if
```

```
: timer1.timer در رویداد
```

```
Img(a).vizable=false
```

```
If a=5 then a=0 else a=a+1
```

```
Img(a).vizable=true
```

```
: img_click در رویداد
```

```
Form1.caption=val(form1.caption)+1
```

فصل سوم

ساخت منو

استفاده بهینه از منو ها در برنامه های ویژوال بیسیک علاوه بر کمک به عملکرد بهتر برنامه می تواند نقش مهمی در زیبایی برنامه نیز داشته باشد و از شلوغی برنامه به اندازه ای کاست برای اینکه برنامه شما از نظر کاربران جذاب باشد باید منو برای آن طراحی کرد و ساخت، اگر چه از نظر ما مثلا کلید خروج در فرم باشد ساده تر است اما چون کاربران همیشه با هر برنامه ای که کار می کنند آن برنامه دارای منو است و داخل آن منو کلید خروج گذاشته شده است، همیشه کاربران عادت دارند با منو ها کار کنند نه با کلیدها!

منوها: برای قراردادن و طراحی منو برای فرم از منوی `Tools\ Menu Editor` را انتخاب می کنیم.

برای افزودن گزینه ای جدید به منو عنوان آن را در کادر `caption` و نام آن را در کادر `name` تایپ می کنیم.

برای آنکه گزینه فعلی زیر منوی گزینه قبلی شود کلید جهت راست را فشرده و برای لفو آن از کلید جهت سمت چپ استفاده میکنیم.

برای ترتیب گزینه ها در منو از فلش بالا و پایین استفاده می کنیم و برای آنکه کنار گزینه علامت تیک گذاشته شود از کادر `checked` برای فعال کردن گزینه از کادر `enabled` و برای قابل رویت بودن گزینه از کادر `visible` استفاده می شود .

برای استفاده کردن گزینه ای بین گزینه های دیگر از `insert` و برای حذف یک گزینه از `delete` استفاده می شود .

برای آنکه بین گزینه های منو خطی قرار گیرد که مثلا بعضی گزینه ها را از بعضی دیگر جدا کند ، گزینه ای با `caption` خط فاصله ایجاد می کنیم .

و برای قرار دادن خط زیر برای یکی از حروف گزینه به منظور انتخاب گزینه به کمک `Alt` قبل از حرف مورد نظر در `caption` کاراکتر `&` را قرار می دهیم .

مثال: &File <= File

برنامه نویسی برای منو : هر گزینه ای از منو در واقع یک شیء است و دارای رویداد **click** می باشد .

خاصیت **enabled** : در حین اجرای برنامه تغییر این خاصیت به **true** و **false** باعث غیر فعال شدن گزینه می شود .

مثال : در برنامه رو به رو گزینه **open** را غیر فعال می کند : **mnuopen.enabled=false** که در اینجا **mnuopen** نام گزینه است.

خاصیت **visible** : اختصاص مقدار **false** به این خاصیت گزینه را پنهان می کند .

خاصیت **checked** : خاصیت مقدار **true** به این خاصیت باعث می شود تا کنار گزینه علامت تیک ظاهر شود .

مثال:در برنامه زیر انتخاب گزینه "**show**" باعث نمایش کلمه **iran** در عنوان فرم و انتخاب مجدد آن باعث پاک شدن عنوان فرم می شود :

(توجه : در پایین ما نام **mnudisplay** را انتخاب کردیم ولی این نام همیشه هر چه می تواند باشد)

در رویداد **click_mnudisplay** :

```
IF mnudisplay.checked=true Then
    mnudisplay.checked=false
    form1.caption=""
else
    mnudisplay.checked=true
    form1.caption=" iran"
end if
```

دستور ظاهر کردن منو : `popupmenu`

مثال : دستور رو به رو باعث ظاهر شدن منوهای `mnufile` میشود : `popupmenu mnufile`

رویدادهای `mousedown` و `mouseup` (برای فرم) :

در صورت فشردن هر کلیدی از ماوس رویداد `mousedown` و با رها کردن هر کلید آن رویداد `mouseup` رخ می دهد.

رویداد فوق چهار پارامتر زیر را در اختیار قرار می دهند :

`form_mouseup(button,shift,x,y Sub)`

`button` : اگر کلید سمت چپ فشرده شود این پارامتر برابر با یک و اگر کلید سمت راست فشرده شود برابر با دو می شود .

`Shift` : این پارامتر وضعیت فشرده شدن کلیدهای `shift` و `ctrl` و `alt` را نشان می دهد پارامتر این مقدار معادل مبنای ۱۰ یک عدد ۳ رقمی مبنای ۲ است .

مثال : در برنامه زیر با راست کلیک روی فرم منو ظاهر می شود :

در رویداد `form_mousedown` :

`If button=2 then popupmenu mnufile`

البته منوهای چند ستونی را میتوان با برنامه ویژوال بیسیک ساخت که آموزش اون رو در آپدیت بعدی در اختیارتون میگذارم.

فصل چهارم

گرافیک

استفاده از گرافیک باعث زیباتر و جذابتر شدن برنامه می شود .

شیء **screen** : به کمک این شیء میتوان مشخصات صفحه نمایش را بدست آورد بعضی از خاصیت های آن عبارتند از :

توضیح	خاصیت
شماره شیء فعال در برنامه	Active control
شماره فرم فعال	Active from
تعداد قلم های صفحه	Font count
آرایه ای حاوی نام قلم های صفحه	Font
ارتفاع صفحه بر حسب twips	height
شکلک ماوس	Mouseicon
شکل اشاره گر ماوس	Mousepointer
تعداد twips ها در هر پیکسل (افقی)	twipsperpixelX
تعداد twips ها در هر پیکسل (عمودی)	twipsperpixelY
پهنای صفحه بر حسب twips	width

مثال:در برنامه زیر با فشردن کلید **command1** وضوح صفحه در **text1** و **text2** نمایش داده میشود :

در رویداد **comman1_click** :

```
Text1.text=screen.width / screen.twipsperpixel X
```

```
Text2.text=screen.height / screen.twipsperpixel Y
```

خاصیت **top** (برای همه اشیاء) : این خاصیت فاصله لبه بالایی شیء تا بالای فرم را در خود نگهداری می کند .

خاصیت **left** (برای همه اشیاء) : این خاصیت فاصله لبه سمت چپ فرم را در خود نگهداری می کند .

خاصیت **height** : ارتفاع شیء در خاصیت نگهداری می شود و **width** که همان پهنای شیء است (هر دو برای همه اشیاء)

خاصیت **scalmode** (برای همه اشیاء) : واحد اندازه گیری مورد استفاده در شیء را تعیین می کند .

مثال : در برنامه زیر فشردن کلید **command1** باعث می شود تا پهنای فرم زیاد و کم شود :

خاصیت **scalmode** را برای **form1** را برابر با پیکسل قرار می دهیم و **interval** شیء **timer1** را برابر با ۱۰۰ قرار می دهیم .

در ناحیه **general** : **dim c**

در رویداد **command1_click** :

Timer1.enabled=timer

در رویداد **timer1_timer** :

Form1.width=form1.width+c

If Form1.width >=400 then c=-10

If Form1.width <=400 then c=10

قبل از اجرای برنامه خاصیت **enabled** برای **timer1** را **false** می کنیم .

متد **move** (برای همه اشیاء) : از این متد برای جابجایی یا تغییر اندازه شیء استفاده می شود .

مثال:دستور زیر باعث جابجایی فرم به مختصات (۲۰۰ و ۳۰۰) می شود :

form1.move 300,200

روشهای اضافه کردن گرافیک به برنامه :

۱: بارگذاری تصویر ۲: استفاده از کنترل های گرافیکی ۳: استفاده از متدهای گرافیکی

۱: (بارگذاری تصویر) بسیاری از اشیاء مانند **form** و **command** دارای خاصیتی به نام **picture** می باشد.

که در صورت اختصاص تصویری به آن تصویر یاد شده روی شیء قرار می گیرد .

۲: (کنترل های گرافیکی): کنترل های مانند **image,picture,shape,line** امکان قرار دادن تصاویر و طرح های ساده در برنامه فراهم میکند .

۳: (متدهای گرافیکی): متدهای گرافیکی مانند **line,circle** امکان طرح های ساده روی اشیایی مانند **form,picture** و چاپگر و غیر ممکن می کند .

کنترل **line**: به کمک این کنترل می توان خطوط مستقیم درون برنامه قرار داد .

کنترل **shape**: این کنترل امکان رسم دایره و بیضی و مربع و مستطیل را فراهم می کند .

خاصیت **fillstyle** (برای همه اشیاء): نحوه پر شدن درون شیء توسط این خاصیت تعیین می شود .

خاصیت **shape** طرح درون شیء **shape** توسط این خاصیت تعیین می شود .

خاصیت **fillcolor**: رنگ درون شیء توسط این خاصیت تعیین می شود .

رسم تصویر به کمک متدهای گرافیکی :

۱: متد **pset**: نام فرم با این پارامتر میتوان یک نقطه روی فرم رسم کرد .

مثال : در برنامه زیر با هر بار کلیک نقطه ای از محل کلیک رسم می شود :

در رویداد `Form_MouseDown` : `Form1.PSet (X, Y)`

۲:متد `line` : شکل فرمول : `form1.line(x1,y1)-(x2,y2)[,color[,bf or b]]`

پارامتر `b` باعث رسم یک مستطیل تو خالی و `bf` باعث رسم یک مستطیل تو پر می گردد .

۳: متد `circle` : شکل فرمول `[color,]شعاع,circle(x,y)`، نام فرم (این متد دایره رسم می کند)

۴: متد `cls` : شکل فرمول `form1.cls` (این متد ترسیمات روی صفحه را پاک می کند)

۵: متد `print` : به کمک این متد میتوان عبارت ها و یا مقدار متغیرها را روی فرم نمایش داد .

مثال در برنامه زیر فشردن `command1` باعث نمایش `hello` روی فرم می شود :

در رویداد `click_command1` : `form1.print"hello"`

خاصیت `currentX` و `currentY` (برای فرم) : این خاصیت ها مختصات مکان نمای گرافیکی را تعیین می کند

مثال : نمایش کلمه `iran` در مختصات (۵۰۰،۳۰۰) :

توجه برای طراحی این مثال باید فرم را در حالت `activate` قرار داشته باشد توضیح بیشتر `Activate` در فصل شش.

```
Form1.currentX=۵۰۰
```

```
Form1.currentY=۳۰۰
```

```
Form1.print "IRAN"
```

متد `textwidth` , `textheight` : این متد پهنا و ارتفاع یک متن را برمی گرداند.

شکل فرمول نویسی : (عبارت مورد نظر) `textwidth`. نام فرم = متغیر

متد `print` (برای فرم) : این متد رنگ یک نقطه را برمی گرداند. شکل فرمول نویسی :
`print(x,y)`. نام متغیر

تابع `qcolor` : امکان انتخاب یکی از رنگهای ۱۶ گانه `qbasic` را فراهم می کند.

مثال رسم دایره ای به رنگ زرد :

```
form1.circle(300,250),200,qbcolor(14)
```

تابع `RGB` : با این تابع می توان رنگی از میان ۱۶۰۰۰۰۰۰ انتخاب کرد نحوه استفاده در زیر نشان داده شده است :

مثال رسم یک خط : `form1.line(10,250)-(580,670),RGB(38,140,48)`

متد `printform` (برای فرم) : این متد باعث چاپ محتویات فرم توسط چاپگر می شود روش کار :
`printform`. نام فرم

شیء `printer` : به کمک این شیء امکان کار با چاپگر و انجام تنظیمات آن فراهم می شود .

متد `print` (برای `printer`): با این متد میتوان متنی را برای چاپ تعیین کرد. مثال تعیین `iran` برای چاپ `printer.print"iran"`

خاصیت `page` (برای `printer`) : شماره کاغذ در این خاصیت نگهداری می شود .

متد `newpage` (برای `printer`) : این متد باعث تعویض کاغذ می شود .

متد `enddoc` (برای `printer`) : این متد باعث آغاز به کار چاپگر و چاپ داده های ارسالی به آن می شود .

مثال : در برنامه زیر فشردن کلید **command1** باعث چاپ **iran** در کادر اول و چاپ **Masoud** در کادر دوم می شود :

در رویداد **command1_click** :

```
Printer.print"+ شماره صفحه" + printer.page  
  
Printer.print"iran"  
Printer.newpage  
Printer.print"+ شماره صفحه" + printer.page  
  
Printer.fontsize=40  
Printer.fontname="Test"  
Printer.print"Masoud"  
Printer.enddoc
```

نکته : متدهای **circle** , **pset** , **line** , **textheight** , **textwidth** برای شیء **printer** نیز وجود دارند .

مثال ۲ : در برنامه زیر بعد از فشردن **command1** باعث چاپ یک دایره و چاپ محتویات **text1** در کاغذ دوم می شود :

در رویداد **command1_click** :

```
Printer.circle(300,200),100  
  
Printer.newpage  
Printer.print text1.text  
Printer.enddoc
```


فصل پنجم

آشنایی با کنترل ها

کنترل های **list** و **combo** : این کنترل گزینه هایی را به کاربر نمایش می دهد و وی بتواند یکی از گزینه ها را انتخاب کند .

اضافه کردن گزینه به **list** :

۱: **combo** : استفاده از خاصیت **list** در حین طراحی

۲ : استفاده از متد **additem** در حین اجرا

خاصیت **list** (برای **list** و **combo**) : گزینه های **list** و **combo** در آن نگهداری می شود.

برای اضافه کردن گزینه به آن پس از تایپ یک گزینه کلید های **ctrl+enter** را فشار می دهیم .

متد **additem** (برای **list** و **combo**): از این متد برای اضافه کردن گزینه به **list** و **combo** در حین اجرای برنامه استفاده می شود.

گزینه مورد نظر **Additem**. نام **combo** یا **list**

مثال: در برنامه زیر فشردن **command1** باعث می شود که متن درون **text1** به صورت گزینه ای به **list1** اضافه شود :

در رویداد **command1_click** :

```
If len(text1.text)<>0 then
```

```
List1.additem text1.text
```

```
Text1.text=""
```

```
End if
```

خاصیت **listcount** (برای **list** , **combo**) : تعداد گزینه های **list** و **combo** در این خاصیت قرار دارد .

خاصیت **listindex** (برای **list** و **combo**) : شماره گزینه انتخاب شده درون **list** و **combo** در این خاصیت نگهداری می شود و شماره اولین گزینه صفر است .

مثال : در برنامه زیر با یک بار کلیک روی گزینه ای از **list** شماره آن در عنوان فرم و با **dbclick** روی یک گزینه در عنوان فرم ظاهر می شود

در رویداد **list_click** : `form1.caption=list1.listindex`

در رویداد **list1_dbclick** : `form1.caption=list1.list(list1.listindex)`

متد **removeitem** (برای **list** و **combo**) : به کمک این متد میتوان گزینه ای از **list** یا **combo** را حذف کرد :

شکل فرمول : شماره مورد نظر **additem** نام **combo** یا **list**

مثال در برنامه زیر با فشار **command1** گزینه انتخاب شده از **list1** حذف و به **combo1** اضافه می شود :

ابتدا در خاصیت **list** از **list1** گزینه های رو به رو را وارد می کنیم : **one , two , three , four**

در رویداد **command1_click** :

`list1.list(list1.listindex Combo1.additem)`

`List1.removeitem(list1.listindex)`

`If list1.listcount=0 then command1.enabled=false`

متد **clear** (برای **list** و **combo**) : از این متد همه گزینه های **list** و **combo** را حذف می کند .

خاصیت **multiselect** (برای **list**): این خاصیت امکان انتخاب چندین گزینه از لیست را فراهم می کند و یکی از مقادر زیر است :

none : انتخاب فقط یک گزینه .

Simple : با هر بار کلیک روی یک گزینه آن گزینه انتخاب شده و با کلیک مجدد از انتخاب در می آید .

Extended : به کمک کلیدهای **ctrl** و **shift** می توان چندین گزینه را انتخاب کرد .

خاصیت **selected** (برای **list**) : در حالتی که امکان انتخاب چند گزینه از لیست وجود داشته باشد اگر **t** مین گزینه انتخاب شده باشد (**true** برابر **selected t**) می باشد .

مثال: در برنامه زیر پس از انتخاب چند گزینه از **list1** و فشردن **command1** گزینه های انتخاب شده به **combo1** اضافه می شود :

ابتدا چند گزینه به **list1** اضافه می کنیم و خاصیت **multiselect** را برابر با **simple** یا **extended** قرار می دهیم

در رویداد **command1_click** :

```
For i=0 to list1.listcount-1
```

```
If List1.Selected(i) = True Then Combo1.AddItem List1.lis(i)
```

```
Next i
```

Drag کردن اشیاء : در برنامه امکان دراگ کردن اشیاء وجود دارد و رویدادهای مورد نیاز عبارتند از :

خاصیت **dragmode** (برای همه اشیاء): اگر این خاصیت را برابر با **automatic** قرار دهیم دراگ کردن شیء فراهم می شود .

رویداد **dragover** (برای همه اشیاء): در حین دراگ کردن یک شیء این رویداد دائماً رخ می دهد این رویداد دو پارامتر زیر را دارد :

source: شیء ای که دراگ می شود .

X,Y: مختصات اشاره گر ماوس

مثال : در برنامه زیر در حین دراگ کردن مختصات فعلی آن در عنوان فرم نمایش داده می شود :

خاصیت **dragmode** شیء **command1** را برابر با **automatic** قرار می دهیم

در رویداد **form1_dragover** :

Form1.caption=str(x)+","+str(y)

رویداد **dragdrop** (برای همه اشیاء): در پایان دراگ کردن یک شیء این رویداد اتفاق می افتد پارامتر های آن مشابه با **dragover** است.

مثال: در برنامه زیر در پایان دراگ کردن **command1** در صورت رها کردن کلید ماوس **command1** در محل جدید خود قرار می گیرد :

خاصیت **dragmode** را **automatic** گذاشته و در رویداد **form1_dragdrop** :

command1.top=y

command1.left=x

متد **setfocus** : برای اجرای این متد برای یک شیء کنترل برنامه به آن شیء منتقل شده و آن شیء انتخاب شود

مثال : در برنامه زیر فشردن **command1** باعث انتخاب شدن **Commnad2** می شود :

در رویداد **command1_click** : **command2.setfocus**

رویداد **keypress** (برای همه اشیاء): وقتی که یک شیء فعال است فشردن هر کلیدی از صفحه کلید باعث رخ دادن رویداد فوق می شود

این رویداد کد اسکی کلید فشرده شده را از طریق پارامتر **keyascii** برمی گرداند.

```
sub text_keypress(keyascii)
```

مثال در برنامه زیر هنگام تایپ در **text1** فشردن کلید **enter** باعث فعال شدن **command1** و فشردن **esc** باعث خروج از برنامه می شود :

در رویداد **text1_keypress** :

```
If keyascii=13 then
```

```
command1.setfocus
```

```
If keyascii=27 then
```

```
End
```

```
End if
```

رویداد **Gotfocus** : هنگامی که کنترل برنامه به یک شیء منتقل می شود این رویداد و برای آن رخ می دهد .

مثال در برنامه زیر با فعال کردن هر **text** توضیحی برای آن در عنوان فرم ظاهر می شود :

در رویداد **text1_Gotfocus** :

```
Form1.caption=" در این کادر نام را وارد کنید "
```

در رویداد **text2_Gotfocus** :

```
Form1.caption=" در این کادر نام خانوادگی را وارد کنید "
```

در رویداد `text3_Gotfocus` :

```
Form1.caption = "در این کادر سن را وارد کنید"
```

رویداد `lostfocus` (برای همه اشیاء) : هنگامی که کنترل برنامه از یک شیء خارج می شود این رویداد برای آن شیء رخ می دهد.

مثال در برنامه زیر کاربر باید تاریخ تولد روز و ماه و سال خود را در `text3, text2, text3` وارد کند هنگام خروج از هر `text`ها معتبر بودن چک می شود :

در رویداد `text1_lostfocus` :

```
a=val(text1.text)
```

```
If a < 1 Or a > 31 Then MsgBox ("روز نا معتبر است")
```

```
Text1.setfocus
```

```
End if
```

در رویداد `text2_lostfocus` :

```
a=val(text2.text)
```

```
If a < 1 Or a > 12 Then MsgBox("ماه نا معتبر است")
```

```
Text2.setfocus
```

```
End if
```

در رویداد `lostfocus_text3` :

```
a=val(text3.text)
```

If a < 1300 Or a > 1384 Then MsgBox ("سال نا معتبر است")

Text3.setfocus

End if

رویدادهای **keydown** و **keyup** : هنگامی که یک شیء فعال است فشردن هر کلیدی از صفحه کلید باعث **keydown** و رها کردن آن باعث رخ دادن **keyup** می شود

این رویدادها دارای پارامترهای زیر می باشند :

KeyCode : کد پویس کلید فشرده شده در این پارامتر نگهداری می شود .

Shift : این پارامتر وضعیت فشرده شدن کلیدهای **alt** , **ctrl** , **shift** را در خود نگهداری می کند .

قرار دادن تصویر روی کلیدها :

خاصیت **style** کلید **command** را برابر **graphical** قرار می دهیم و بعد فایل تصویری را درون خاصیت **picture** قرار می دهیم.

خاصیت **downpicture** (برای **command**) : تصویری که به این خاصیت نسبت داده شود در هنگام فشردن کلید دیده می شود .

خاصیت **disablepicture** (برای **command**): تصویری که به این خاصیت نسبت داده شود هنگام غیر فعال بودن کلید روی آن دیده می شود

کنترل های **vscroll** و **hscroll** : در این دو کنترل هنگام انتخاب مقداری بین دو مقدار حداقل و حداکثر را فراهم می کند.

خاصیت **min** : (برای **vscroll** و **hscroll**) : حداقل مقدار این دو کنترل در خاصیت **min** قرار می گیرد (مقدار پیش فرض صفر است) .

خاصیت **max** : (برای **vscroll** و **hscroll**) : بیشترین مقدار این دو کنترل است (مقدار پیش فرض ۳۲۷۶۷ است) .

خاصیت **value** (برای **vscroll** و **hscroll**) : مقدار انتخاب شده توسط کاربر در این خاصیت قرار می گیرد .

رویداد (**scroll** و **hscroll**) : با لغزاندن کلید وسط این دو کنترل این رویداد رخ می دهد .

رویداد (**change** و **hscroll**) : با فشردن کلیدهای دو طرف این دو کنترل این رویداد رخ می دهد .

مثال در برنامه زیر به کمک **hscroll** می توان یکی از رنگهای ۱۶ گانه **qbasic** برای زمینه فرم انتخاب کرد :

خاصیت **min** را برابر ۰ و **max** را برابر ۱۵ قرار داده

در رویداد **hscroll_scroll** :

```
Form1.backcolor=qbcolor(hscroll1.value)
```

در رویداد **hscroll_change** :

```
Hscroll1_scroll
```

کنترل های **drive** , **dir** , **file** : این سه کنترل امکان دسترسی درایوها و لیست پوشه ها و فایل ها را فراهم می کند .

خاصیت **Drive** (برای **drive**) : درایو انتخاب شده درون کنترل **drivr** در این خاصیت نگهداری می شود .

خاصیت **path** (برای **dir** , **file**) : مسیر کنترل های **dir** و **file** در این خاصیت نگهداری می شود .

رویداد **change** (برای **drive** , **dir**) : با تغییر درایو و پوشه درون کنترل های **drive** و **dir** این رویداد رخ می دهد.

خاصیت **filename** (برای **file**) : نام فایل انتخاب شده در این خاصیت قرار دارد .

مثال در برنامه زیر با انتخاب یک فایل نام آن در عنوان فرم نمایش داده می شود :

در رویداد **drive1_change** :

```
Dir1.path=drive1.drive
```

در رویداد **dir1_change**:

```
File1.path=dir1.path
```

در رویداد **file1_click**:

```
Form1.caption=file1.filename
```

نکته : کنترل **file** نیز مانند کنترل **list** دارای خاصیت های **listindex** , **listcount** هستند.

خاصیت **pattern** (برای **File**) : به کمک این خاصیت میتوان الگویی برای گزینش فایل های درون **File** تعیین کرد مثلا اختصاص ***.txt** که این خاصیت باعث نمایش فایل هایی با پسوند **txt** درون **File** می شود و مقدار ***.jpg;*.bmp;*.gif** باعث نمایش فایل تصویری درون فایل می شود.

مثال ۲ : در برنامه زیر با انتخاب یک فایل تصویری تصویر درون **image1** دیده می شود :

مقدار دهی خاصیت ها :

برای **file1** :

```
pattern=*.jpg;*.bmp;*.gif
```

برای **image1** :

```
stretch=true
```

در رویداد :drive1_change

Dir1.path=drive1.drive

در رویداد : dir1_change

File1.path=dir1.path

در رویداد :file1_click

If len(file1.path)=3 then

Iamge1.picture=loadpicture(file1.path+file1.filename)

Else

image1.picture=loadpicture(file1.path+"\\"+file1.filename)

end if

مثال ۳: در برنامه فوق به کمک یک تایمر فایل های تصویری به صورت خودکار هر یک ثانیه عوض شوند :

به مثال بالا فقط تایمر اضافه شده است interval آن را برابر ۱۰۰۰ قرار داده

در رویداد : timer1_timer

If file1.listcount > ۰ Then

If file1.listindex=file1.listcount-1 then

File1.listindex=0

Else

File1.listindex=file1.listindex+1

End if

File1_click

End if

کنترل **commdialog** : به کمک این کنترل امکان استفاده از پنجره های محاوره ای رایج در ویندوز در برنامه **vb** فراهم می شود به طور پیش فرض این کنترل در **toolbox** قرار ندارد برای اضافه کردن آن از منوی **project** گزینه **components** را انتخاب کرده در پنجره باز شده گزینه **Microsoft common dialog control** را تیک می زنیم و پنجره را تایید می کنیم.

خاصیت **filename** (برای **commdialog**) : نام و مسیر فایل انتخاب شده در این خاصیت قرار می گیرد .

خاصیت **dialogtitle** (برای **commdialog**) : عنوان پنجره **commdialog** در این خاصیت جا داده می شود .

خاصیت **filter** (برای **commdialog**) : نوع فایل هایی که درون پنجره های **commdialog** دیده می شوند توسط این خاصیت قابل انتخاب است .

نحوه انتخاب آن به شکل زیر است :

... نوع دوم|عنوان دوم|نوع اول|عنوان اول = **filter**

مثال : **Filter=textfile|*.txt|word documents|*.doc|all file|*.***

مثال : در برنامه زیر با انتخاب یک فایل تصویری تصویر آن درون **image1** نشان داده می شود :

ابتدا منوی رو به رو را طراحی می کنیم :

File (mnufile)

open (mnuopen)

خاصیت **filter** برای **commdialog** را برابر با عبارت زیر قرار می دهیم :

`bitmap|*.bmp|jpegs|*.jpg|all grapghic|*.bmp;*.jpg`

خاصیت `stretch` را برای `image1` را `true` می کنیم .

در رویداد `mnuopen_click` :

`Commondialog1.showopen`

`Image1.picture=loadpicture(commondialog1.filename)`

متد `showsave (commondialog)` : این متد باعث باز شدن پنجره `save` می شود.

متد `showcolor (برای commondialog)` : این متد باعث باز شدن پنجره جعبه رنگ ویندوز می شود.

خاصیت `(color commondialog)` : رنگ انتخاب شده در جعبه رنگ در این خاصیت قرار دارد.

مثال:در برنامه زیر با فشردن کلید `command1` امکان تغییر رنگ زمینه فرم با استفاده از متد `showcolor` فراهم می گردد :

در رویداد `command1_click` :

`Commondialog1.showcolor`

`Form1.backcolor=commondialog1.color`

فصل ششم

کار با توابع و فرم ها

افزافه کردن فرم به برنامه : برای افزودن فرم های دیگر به برنامه از منوی `project` گزینه `add form` را انتخاب می کنیم.

متد `show` (برای `form`) : این متد باعث نمایش فرم می شود . `show` نام فرم

رویداد `load` (برای `form`) : در اولین فراخوانی یک فرم (با متد `show`) این رویداد رخ می دهد و معمولا از آن برای مقدار دهی اولیه به متغیر ها استفاده می شود.

رویداد `activate` (برای `form`) : این رویداد در سه حالت زیر رخ می دهد :

۱: در اولین فراخوانی فرم

۲: اگر فرم قبلا فراخوانی شده باشد در فراخوانی مجدد نیز این رویداد رخ می دهد

۳: اگر فرم فراخوانی شده باشد

ولی با فراخوانی آن فرم به روی دیگر پنجره ها آمده این رویداد برای آن رخ می دهد.

متد `hide` (برای `form`) : این متد باعث بسته شدن و ناپدید شدن فرم می شود `name form.hide`

رویداد `deactivate` (برای `form`) : در صورت بسته شدن یک فرم این رویداد برای آن رخ می دهد.

مثال : در برنامه زیر فشردن کلید `command1` در `form1` باعث باز شدن `form2` و فشردن کلید `command1` در `form2` باعث بسته شدن `form2` می شود :

در رویداد `command1` برای `form1` :

`Form2.show`

در رویداد **command1** برای **form2** :

Form2.hide

فرم مدال (**modal**) : به فرمی گفته می شود که تا این فرم باز است نمی توان به دیگر فرم های برنامه دسترسی پیدا کرد .

برای باز کردن فرم مدال به شکل رو به رو عمل می کنیم **name form.show vbmodal**

تعیین شیء راه انداز برنامه : نقطه شروع یک برنامه می تواند یک فرم و یا یک روال باشد.

برای تغییر شیء راه انداز از منوی **project** گزینه **properties** را انتخاب می کنیم در کادر **startup object** شیء مورد نظر را تعیین می کنیم .

انواع رابط های گرافیکی :

۱: رابط های گرافیکی تک سندی (**SDI**) : در این نوع رابط در هر لحظه می توان تنها با یک سند (متن و تصویر) کار کرد

مانند **notepad , paint**

۲: رابط گرافیکی چند سندی (**MDI**) : در این نوع رابط در هر لحظه می توان با چندین سند کار کرد
مانند **word , photoshop**

فرم **mdi** : با انتخاب منوی **project** و برگزیدن **add mdi form** میتوان یک فرم **mdi** به برنامه اضافه کرد. باید توجه داشت که هر برنامه میتواند حاوی یک فرم **mdi** باشد.

از بسیاری از کنترل های نمیتوان در فرم **mdi** استفاده کرد

معمولا برای فراخوانی دیگر فرم های برنامه از طریق فرم **mdi** از منوها و نوارهای ابزار استفاده می شود.

فرم فرزند : به فرمی از برنامه گفته می شود که نمی توان آن را از فرم mdi خارج کرد. ضمن آنکه اگر فرم فرزند حاوی منو باشد در صورت باز کردن آن منوی آن در بالای فرم mdi دیده می شود.

خاصیت mdichild (برای فرم) : اگر این خاصیت را برابر با true قرار دهیم فرم به فرم فرزند برای فرم mdi تبدیل می شود.

نکته : فرم mdi می تواند دارای فرم غیر فرزند نیز باشد.

ماژول ها (modules) : نواحی برای کد نویسی و ایجاد زیر برنامه ها و توابع می باشند

برای اضافه کردن یک ماژول به برنامه گزینه از منوی project گزینه addmodules را برمی
گزینیم

فصل هفتم

روش دستیابی به فایل ها

روش دستیابی به فایل ها :

۱: دسترسی ترتیبی ۲: دستیابی تصادفی ۳: دسترسی دودویی

*دسترسی ترتیبی : در این روش اطلاعات در فایل به صورت سطر به سطر خوانده یا نوشته می شوند

و برای دستیابی به یک سطر خاص لازم است تا از اطلاعات قبل آن عبور کرده به اطلاعات مورد نظر برسیم . خواندن و نوشتن اطلاعات در این روش ممکن نیست

مثال : فایل های متنی

*دستیابی تصادفی : در این روش اطلاعات به صورت رکورد به رکورد خوانده و نوشته می شوند.

و با دانستن شماره هر رکورد میتوان مستقیماً به اطلاعات آن رکورد دسترسی پیدا کرد

خواندن و نوشتن همزمان رکوردها امکان پذیر است . مثال فایل های پایگاه داده

*دسترسی دودویی : در این روش اطلاعات به صورت بایت به بایت خوانده یا نوشته می شوند

برای دستیابی به یک بایت دانستن شماره آن بایت کافی است به طور همزمان می توان عملیات خواندن و نوشتن را روی اطلاعات انجام داد

مثال فایل های صوتی و تصویری .

دستور باز کردن فایل : قبل از استفاده از یک فایل ابتدا باید با دستور زیر آن را باز کرد:

شماره فایل As عملیات مورد نظر For نام و مسیر فایل open

عملیات مورد نظر می تواند یکی از موارد زیر باشد :

Input : باز کردن فایل به صورت ترتیبی برای خواندن اطلاعات آن.

Output : باز کردن فایل به صورت ترتیبی برای نوشتن اطلاعات در آن در صورتی که فایل از قبل موجود باشد اطلاعات قبلی آن پاک می شوند .

Append : باز کردن فایل به صورت ترتیبی برای اضافه کردن اطلاعات به انتهای آن.

Random : باز کردن فایل به صورت تصادفی برای نوشتن و خواندن اطلاعات.

Binary : باز کردن فایل به صورت دودویی برای نوشتن و خواندن اطلاعات آن . شماره فایل باید عددی غیر تکراری باشد مثال :

Open"c:\persian.txt" for input as 3

Open"e:\vb.dat" for random as 5

دستورات فوق فایل c:\persian.txt را برای خواندن اطلاعات آن به صورت ترتیبی باز کرده و شماره 3 را به آن اختصاص می دهد.

دستور دوم فایل e:\vb.dat را به صورت تصادفی برای نوشتن و خواندن رکورد های آن باز کرده و شماره 5 را به آن اختصاص می دهد.

تابع یافتن شماره غیر تکراری برای فایل :

freeware() = نام فایل

مثال :

A=freefile()

Open"Persian.txt"for append as a

دستور بستن فایل : در پایان کار با یک فایل آن را با دستور زیر می بندیم . شماره فایل Cloce#

دستور نوشتن در فایل ترتیبی :

اطلاعات مورد نظر , شمار فایل Print#

مثال 1: در برنامه زیر عبارت فشردن کلید command1 باعث ایجاد فایل به نام persian.txt در درایو d: شده و 10 بار عبارت iran را درون آن می نویسد:

در رویداد : command1_click

```
Open"d:\persian.txt"for output as 9
```

```
For i=1 to 10
```

```
print#9,"iran"
```

```
Next
```

```
Close#8
```

مثال 2: در برنامه زیر فشردن کلید save نام فایلی درخواست شده و اطلاعات درون text1 با آن نام ذخیره می شود:

اول خاصیت multiline برای text1 را برابر با true قرار می دهیم.

در رویداد : command1_click

```
Commondialog1.showsave
```

```
Open commondialog1.filename+".txt"for output as 6
```

```
Print#6,text1.text
```

```
Close#6
```

تابع تشخیص انتهای فایل شماره فایل (eof)

در صورتی که به انتهای فایل رسیده باشیم تابع و مقدار فوق true می شود.

مثال : در برنامه زیر با فشردن کلید command1 محتویات فایل d:\persian.txt درون text1 نمایش داده می شود:

اول خاصیت multiline برای text1 را برابر با true قرار می دهیم

در رویداد command1_click

```
open"d:\persian.txt" for input as 8
```

```
do until eof(8)
```

```
input#8,a$
```

```
text1.text=text1.text+a$
```

```
loop
```

```
close
```

دستور نوشتن در فایل تصادفی :

مقدار مورد نظر, شماره رکورد و بایت, شماره فایل put#

دستور خواندن در فایل تصادفی :

متغیر, شماره رکورد و بایت, شماره فایل get#

مثال :در برنامه زیر پس از انتخاب در file و فشردن command1 فایل یاد دشه به file2 کپی می شود:

در رویداد drive1_change :

Drive1.path=dir1.path

در رویداد dir1_change :

File1.path=dir1.path

در رویداد drive2_change :

Dir1.path=drive2.drive

در رویداد Dir2_change :

File2.path=dir2.path

در رویداد command1_click (همان دکمه کپی):

Open file1.path+"\\"+file1.filename for binary as 1

Open file2.path+"\\"+file1.filename for binary as 2

I=0

Form2.show

Form2.label1.caption="copy file"+file1.filename

Do until eof(1)

Get#1,I,a\$

Put#2,I,a\$

I=i+1

Loop

Form2.hide

close

در مثال بالا فرم 2 نشانه صبر و نشان دادن فایل های که دارند کپی می شود.

فصل هشتم

کار با فایل ها

curdir دستور تغییر نام پوشه

curdir دستور تغییر نام پوشه

chdir دستور تغییر فهرست جاری

chdrive دستور تغییر درایو جاری

mmdir دستور ایجاد پوشه

مثال ساخت پوشه class در درایو e:\

```
Chdrive("e:")
```

```
Chdir("\\")
```

```
Mmdir("class")
```

rmdir دستور حذف پوشه خالی

دستور حذف فایل kill

مثال :حذف فایل c:/dos/os2.txt

```
Kill("c:\dos\os2.txt")
```

دستور جستجوی فایل dir

مثال:در برنامه زیر فایل readme.txt جستجو شده و پیغام با جستجو به نمایش در می آید:

```
A=dir(readme.txt)
```

```
If a=" " then
```

```
Msgbox("no find file")
```

```
Else
```

```
Msgbox("find file")
```

```
End if
```

مثال در برنامه زیر فشردن command1 باعث چاپ فایل های با پسوند txt روی کاغذ می شود:

در رویداد : command1_click

```
a=dir("*.txt")
```

```
do while a<>""
```

```
printer.print a
```

```
a=dir()
```

```
loop
```

```
printer.enddoc
```