به نام يزدان بخشاينده

ساخت ماشين حساب كامپيوتري

با استفاده از VB 6

مسعود كايدخورده

Masoud.stu@Gmail.com

انجمن :

www.irankorea.com/masoudarticle

مقدمه :

همان طور كه مي دانيد ، ويژوال بيسيك (Visual Basic) زبان برنامه نويسي قدرتمندي تحت ويندوز مي باشد كه محصولي از شركت معتبر Microsoft است و با استفاده از آن مي توان برنامه هاي تحت ويندوز را به آساني و به سرعت تهيه نمود . در اين مقاله ما قصد داريم يك ماشين حساب كار آمد را با استفاده از ويژوال بيسيك در قالب آموزشي ، به صورت راهنماي گام به گام بسازيم و شما عزيزان را توانا در ساخت ماشين حساب هاي مربوط به تخصصتان كنيم (همانطور كه مي دانيد ماشين حساب ويندوز بسياري از امكانات مهم را دارا نيست .) اين قطعا تمرين مناسبي براي علاقه مندان به اين زبان برنامه نويسي است ، ضمن اينكه نتيجه و محصول كار هم ، نرم افزاري قابل استفاده و جالب است . اين را نيز بايد متذكر شويم كه هرچند در اين مقاله سعي شده با زباني ساده به جزئيات پرداخته شود اما با اين وجود هم نياز به آشنايي مختصر شما با ويژوال بيسيك و اصول برنامه نويسي دارد .

مرحله اول :

طراحي ماشين حساب

طراحي اولين قدم در برنامه نويسي و بخصوص برنامه نويسي تحت ويندوز است و اهميتي ويژه در جذب كاربر دارد . نرم افزاري كه از سوي يك برنامه نويس ارائه مي شود ، اگر طراحي كاربر پسندي نداشته باشد نمي تواند جايگاه قابل قبولي در بين مردم داشته باشد و خيلي زود از دور خارج مي شود ، هر چند نرم افزاري كارا و قدرتمند باشد .

اولين قدم تنظيم اندازهٔ سايز فرمي است كه VB در اختيار ما قرار مي دهد . كه ما بايد به گونه اي آن را تنظيم كنيم كه نه جاي خالي بيش از حد داشته باشيم و نه فضاي كم باشد .

براي ايجاد كليدهاي ماشين حساب از شي Command Button استفاده كنيد و ميزان استفاده هم به اندازه اي است كه مي خواهيد ماشين حسابتان گسترده باشد . خيلي ها دوست دارند با دادن يك معادله و زدن يك كليد ريشه يا ريشه هاي معادله شان بدست آيد و خيلي ها هم دوست دارند تابع آرك تانژانت را به صورت آماده روي ماشين حسابشان باشد . به اندازه خدماتي كه از ماشين حساب مي خواهيم بايد براي آن كليد ايجاد كنيم .

همچنين ميتوانيم از شي هاي Label يا TEXT BOX براي ساخت مانيتور ماشين حساب بهره ببريم .دانستن اصول طراحي برگ برنده اي براي هر برنامه نويس است . كتاب هاي بسياري در اين زمينه به چاپ رسيده اند و ما هم مطالبي را به شما عرضه مي داريم :

1. اطلاعات روي صفحه نمايش به صورت قرينه و تراز شده به نظر آيد
2. طراحي بايد افقي باشد ، نه عمودي و اين بدان معناست چند ستوني باشد و نبايد همه فضا ، به كار گرفته شود بلكه بايد قسمتي از آن خالي بماند .
3. كليد هاي ماشين حساب نبايد به صورت فشرده و چسبيده به هم قرار گيرد ، بلكه بين آنها مقداري فضاي خالي منظور گردد .
4. كليد هاي به صورت دسته بندي از هم جدا باشند ، مثلا كليد هاي عددي و كليد هاي عملگر از هم به صورت مجموعه جدا باشند .
5. بهتراست عنوان هر كليد روي خود كليد قرار گيرد و كليد هاي هم خانواده به يك رنگ باشند .

6. سايز كليد ها و مانيتور ماشين حساب اندازه باشد ، يعني نه آنقدر بزرگ و زننده و نه آنقدر كوچك و آزار دهنده .
7. نوع فونت و اندازه عنوان كليد ها متناسب با سايز آن درنظر گرفته شود .
8. رنگ زمينه و قلم در مانيتور ماشين حساب متمايز باشند .
9. بهتر است رنگ كليد ها هم تاحدي از رنگ زمينه ماشين حساب متفاوت باشند .
10. توجه به رنگ آميزي بسيار مهم است ،از يك طرف انتخاب رنگ درست باعث جلب توجه كاربر مي شود و از طرف ديگر استفاده از رنگ هاي نامناسب از خوانايي مي كاهد و باعث خستگي چشم مي شود .
11. از مخلوط رنگ هاي تيره خودداري شود و هرگز رنگ آبي تيره با پيش زمينه قرمز و يا برعكس تركيب نگردد .
12. استفاده از رنگ قرمز به عنوان رنگ فونت (قلم) كليد هاي عملگر بسيار مناسب است و در مقابل استفاده از رنگ هاي آبي و سبز براي نمايش كليد هاي عددي قابل قبول است .
13. الگو برداري نسبي نه كپي برداري عيني از ماشين حساب ويندوز و ساير ماشين حساب هاي معتبر براي پيشرفت ما بسيار مفيد است .

پس از طراحي نوبت به كد نويسي كه اصلي تزين بخش در ساخت نرم افزار است مي رسد .

مرحله دوم :

كد نويسي ماشين حساب

ما در اين قسمت شروع به برنامه نويسي ماشين حساب مي كنيم ، به تعبيري به عناصر روي ماشين حساب جان مي بخشيم . تمام سعي ما در اين مرحله نوشتن برنامه اي است كه به خوبي كار كند و كاربر را با پيغام ERROR روبرو نكند .

همانطور كه گفتيم ما مي توانيم براي ايجاد مانيتور از 2 شي LABEL يا TEXT BOX به دلخواه استفاده كنيم (اگر از LABEL استفاده كنيم ، براي نمايش اعداد ، خاصيت

Caption اين شي را تغيير مي دهيم و اگر از TEXT BOX بايد با خاصيت TEXT سر وكله بزنيم .)

ما در اينجا از شي Label استفاده مي كنيم و نام آن را Lb1 مي گذاريم .

نقش متغير هايي كه به صورت سراسري (PUBLIC) تعريف مي كنيم بسيار مهم است ، چونكه نگهدارنده موقت اعداد ما هستند .

متغير ها را در قسمت General پنجره كد نويسي به شكل زير تعريف مي كنيم :

```
Public a,b,Memory,Fact as Double
Public S as string
```

تعداد متغير ها در اين برنامه به 4 عدد مي رسد و دليل آنكه آنها را از نوع اعشاري با دقت بالا اين بود كه اولاً ممكن است با اعداد اعشاري را جمع و تفريق كنيم ثانياً ممكن است رقم اعداد بالا باشد .

استفاده از متغير Memory جهت استفاده در حافظه ماشين حساب بوده ، و كاربرد متغير Fact در محاسبه فاكتوريل است ، كاربرد ساير متغير ها را در ادامه بررسي مي كنيم .

كد نويسي كليد هاي عددي :.

براي انجام اعمالي بين دو عدد نياز به رابطه بين كليد هاي اعداد ، عملگر ، مساوي و مانيتور ماشين حساب داريم .

براي نمايش اعداد روي مانيتور مثلا عدد 1 از دستور زير استفاده مي كنيم(با دوبار كليك كردن روي هر شي مي توان به محل مخصوص به كد نويسي آن دسترسي پيدا كرد .)

```
Lb1.Caption=Lb1.Caption+"1"
```

نكته : بايد قبل از شروع به كد نويسي و در مرحله طراحي از طريق پنجره خصوصيات ، Caption اين شي را تهي كنيم .

براي نوشتن كد هاي سايركليد هاي عددي فقط رشته "1" را تغيير مي دهيم . مثلا براي كليد 2 كد را به اين شكل مي نويسيم :

```
Lb1.Caption=Lb1.Caption+"2"
```

همچنين با تغيير "2" به" ." براي كليد مربوط اعشار ، اعداد اعشاري را نيز وارد محاسبه كنيم .

```
Lb1.Caption=Lb1.Caption+"."
```

همانطور كه از اين دستور پيدا ست كه اعداد به صورت رشته در شي Label قرار مي گيرند و به همين دليل از نماد " استفاده كرديم . دليل نوشتن Lb1.Caption بعد از مساوي اين بود كه با زدن هر كليد ، اعداد قبلي از بين نروند و عدد جديد بعد از آنها قرار بگيرد .

كد نويسي كليدهاي جمع،ضرب، تقسيم ، تفريق :

با دوبار كليك كردن روي شي و رفتن به تابع مخصوص اين كليد و نوشتن اين سه دستور مي توانيم اولين عدد را ذخيره كرد و منتظر عدد دوم ماند .

```
a=Val(Lb1.Caption)
Lb1.caption=""
S="+"
```

در خط اول اين دستور با استفاده از تابع VAL ارزش عددي رشته موجود در شي Label را در متغير a ذخيره كرديم و در خط دوم مقدار مانيتور را خالي كرديم تا آماده براي گرفتن عدد جديد باشد . خط سوم هم براي اين است كه كليد مساوي بفهمد كه بايد دو عدد را جمع كند .

نكته : متناسب با هر عملگر از ميان جمع و تفريق و ضرب و تقسيم مقدار اين متغير رشته اي تغيير مي كند .

مثلا براي تفريق و در قسمت مربوط به آن اين دستورها را مي نويسيم .

```
a=Val(Lb1.Caption)
Lb1.caption=""
S="-"
```

كد نويسي كليد مساوي :

با نوشتن دستورات زير در منطقه مخصوص به كليد مساوي كه با Private شروع و با End Sub خاتمه مي يابد مي توانيم حاصل را مشاهده كرد :

```
b=Val(Lb1.Caption)
select case S
     case "+" : lb1.caption=a+b
     case "*" : lb1.caption=a*b
    case "-" : lb1.caption=a-b
End select
if (S="*") And (b>0) then
  lb1.caption=a/b
Else
  lb1.caption=" Cannot Division By Zero "
End if
 a=0
b=0
s=""
```

خط اول اين دستورات ، هر آنچه كه در شي Label است ، تبديل به عدد كرده (با استفاده از تابع Val) و در متغير b قرار مي دهد . در ادامه بنا به عملگري كه از قبل تعيين شده حاصل دو عدد a و b را محاسبه مي كند . دستور Select Case يك دستور شرطي است كه بنا به موجوديت متغير S حاصل را در مانيتور به نمايش در مي آورد . دستور IF هم بررسي مي كند كه اگر محتواي متغير S برابر تقسيم بوده و متغير b عددي غير از صفر باشد (چونكه تقسيم هر عدد بر صفر تعريف نشده است و اگر اينجا كنترل نشود Error مي دهد) عمل تقسيم را انجام دهد و الا پيغام مربوطه به معناي نمي توانم تقسيم بر صفر كنم روي مانيتور ماشين حساب به نمايش در مي آيد .

كد نويسي كليد -/+ :

كد مربوط به اين عمل بسيار ساده و قابل هضم است . با قرار دادن ضرب عملگر – در عدد مي توان مثبت يا منفي يك عدد را بدست آورد ، چونكه اگر عدد مثبت باشد آن را قرينه و به منفي تبديل مي كند و اگر منفي به مثبت .

كد زير اين كار را مي كند :

```
Lb1.Caption=-(Val(Lb1.Caption))
```

كد نويسي كليد فاكتور يل :

بعضي مواقع نياز است كه فاكتوريل يك عدد را محاسبه كنيم ، اين كار براي اعداد كوچك به صورت دستي قابل اجراست ، اما براي اعداد بزرگ خير .

جهت ايجاد كليد فاكتوريل بايد مراحل زير را طي كنيم :

1.ساخت يك Command Button

2.كد نويسي به صورت زير :

```
Fact=1 ;
For i=1 To Val(Lb1.Caption)
Fact=Fact* i
Next i
Lb1.Caption=Fact
```

كد نويسي كليد جذر وبه توان 2 :

براي گرفتن جذر يك عدد از تابع SQR استفاده مي شود و ما مي توانيم با نوشتن اين كد براي كليد مربوطه اين مهم را ميسر كنيم :

```
Lb1.Caption= Sqr(Val(Lb1.Caption))
```

اين دستور ابتدا رشته هاي موجود در شي Label را با استفاده از تابع Val تبديل به عدد مي كند و سپس با بهره گيري از تابع Sqr جذر اين عددرا گرفته و درمانيتور به نمايش مي گذارد .

براي محاسبه به توان 2 هم مي توان اين رشته كد را براي كلد مربوطه اش نوشت :

```
Lb1.Caption= Val(Lb1.Caption)^2
```

علامت ^ به معناي " به توان " است و ما مي توانيم با استفاده از اين علامت توان عدد را بدست آوريم .

کد نويسي کليد هاي Sin , Cos , Tan , Cot :

براي اين کار کافي است که از تابع هاي آماده در VB استفاده کنيم و کد هاي زير را براي کليدهايشان بنويسيم :

محاسبه سينوس :

```
Lb1.Caption=Sin(Val(Lb1.Caption))
```

محاسبه کسينوس :

```
Lb1.Caption=Cos(Val(Lb1.Caption))
```

محاسبه تانژانت :

```
Lb1.Caption=Tan(Val(Lb1.Caption))
```

محاسبه کتانژانت :

```
Lb1.Caption=1/(Tan(Val(Lb1.Caption)))
```

از آنجا که تابع کتانژانت در VB موجود نيست از رابطه 1/Tan(x) کتانژانت را به دست مي آوريم

نمونه اي از ماشين حساب نوشته شده با ويژوال بيسيك با الگو برداري از ماشين حساب ويندوز 98

مرحله سوم :آشنايي با توابع مهم VB و ساخت توابع پيچيده (توابعي كه در VB وجود ندارند)

برای نوشتن برنامه‌های مهندسی، محاسباتی، گرافیکی و آماری نیاز دارید تا از برخی توابع ریاضی استفاده نمائید. ویژوال بیسیک 6 دارای مجموعه‌ای از توابع است که برای انجام محاسبات عددی پیش بینی شده اند. در ادامه 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.

بررسي توابع موجود در ويژوال بيسيك :

تابع قدرمطلق: (Abs)

مقدار بدون مثبت یک عدد را برمی‌گرداند .

تابع آرک تانژانت: (Atn)

خروجی این تابع عددی از نوع double است که برابر زاویه‌ای است که تانژانت آن عدد ورودی تابع است .

تابع توان نمانی:(Exp)

خروجی این تابع عددی از نوع double است که برابر e به توان ورودی تابع است .

تابع کف یا تابع جزء صحیح:(Int)

نزدیکترین عدد صحیح مساوی یا کوچکتر نسبت به عدد ورودی را برمی‌گرداند .

تابع لگاریتم :(Log)

خروجی این تابع عددی از نوع double است که برابر لگاریم طبیعی عدد ورودی است (لگاریتم بر مبنای عدد e یا همان Ln

تابع گرد کردن(Round)

خروجی این تابع عددی از نوع double است که برابر نزدیکترین عدد صحیح به مقدار عدد ورودی است .

تابع علامت :(Sgn)

خروجی این تابع عددی از نوع صحیح است که نشان دهنده علامت عدد ورودی است . اگر خروجي بزرگتر از صفر عدد 1 را برمي گرداند ، اگر برابر صفر عدد 0 و اگر کوچکتر از صفر باشد محصول تابع عدد 1- است .

ایجاد سایر توابع ریاضی که در ویژوال بیسیک 6 وجود ندارند :

جدول زیر چگونگی محاسبه سایر توابع ریاضی که در ویژوال بیسیک 6 وجود ندارند را نشان می‌دهد :

تابع	فرمول
سکانت	Sec(X) = 1 / Cos(X)
کسکانت	Cosec(X) = 1 / Sin(X)
کتانژانت	Cotan(X) = 1 / Tan(X)
آرک سینوس	Arcsin(X) = Atn(X / Sqr(1-X * X))
آرک کسینوس	Arccos(X) = Atn(-X / Sqr(1-X * X)) + 2 * Atn(1)
آرک سکانت	Arcsec(X) = Atn(X / Sqr(X * X - 1)) + Sgn((X) -1) * (2 * Atn(1))
آرک کسکانت	Arccosec(X) = Atn(X / Sqr(X * X - 1)) + (Sgn(X) - 1) * (2 * Atn(1))
آرک کتانژانت	Arccotan(X) = Atn(X) + 2 * Atn(1)
سینوس هیپربولیک	HSin(X) = (Exp(X) - Exp(-X)) / 2
کسینوس هیپربولیک	HCos(X) = (Exp(X) + Exp(-X)) / 2
تانژانت هیپربولیک	HTan(X) = (Exp(X) - Exp(-X)) / (Exp(X) + Exp(-X))
سکانت هیپربولیک	HSec(X) = 2 / (Exp(X) + Exp(-X))
کسکانت هیپربولیک	HCosec(X) = 2 / (Exp(X) - Exp(-X))
کتانژانت هیپربولیک	HCotan(X) = (Exp(X) + Exp(-X)) / (Exp(X) - Exp(-X))
آرک سینوس هیپربولیک	HArcsin(X) = Log(X + Sqr(X * X + 1))
آرک کسینوس هیپربولیک	HArccos(X) = Log(X + Sqr(X * X - 1))
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 مبنایN	LogN(X) = Log(X) / Log(N)

اعداد p و n

برای استفاده از اعداد پی و e در برنامه‌های خود، ثوابت زیر را تعریف نمائید :

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

وبراي نمايش آن مي توان اين كد را براي كليد مربوطه نوشت :

```
La1.Caption=Pi
```

همچنین عدد پی را می‌توان به صورت زیر تعریف کرد

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

تبدیل رادیان به درجه و برعکس

چون اکثر توابع مثلثاتی بر حسب رادیان کار می کنند گاهی اوقات نیاز داریم تا زوایا را از در جه به رادیان و بالعکس تبدیل کنیم. برای تبدیل یک زاویه بر حسب رادیان به درجه، آنرا در 180 ضرب کرده و سپس بر عدد پی تقسیم می‌کنیم :

```
Darage = x * 180 / Pi
```

برای تبدیل یک زاویه بر حسب درجه به رادیان، آنرا در عدد پی ضرب کرده و سپس بر 180 تقسیم می‌کنیم :

```
Radian= x * Pi / 180
```

در پايان :

انشاالله كه از اين مطالب استفاده كامل كرده باشيد ، اما اگر بخواهيد در زمينه برنامه نويسي پيشرفت كنيد ، بايد ازكتب مرجع استفاده كنيد و هرگز در برنامه نويسي ناميد و مايوس نشويد .

منابع :

تجزيه و تحليل و طراحي سيستم هاي كامپيوتري – دكتر مونس علي فرمنش

ويژوال بيسيك –مير محمد رضا علوي ميلاني

همكاري سايت iDevCenter.com در زمينه توابع رياضي

تجربيات خودم

مسعود كايدخورده

Masoud.stu@Gmail.com

www.masoudarticle.mihanblog.com

www.irankorea.com/masoudarticle