ویژوال بیسیک
در
N+1
روز

نوشته : سید امیر احسانی

مقدمه

چند سال بود که قصد داشتم تجربیات خودم در ویژوال بیسیک را بنویسم تا وقتی میخواهم مشتاقی تازه کار را راهنمایی کنم؛ راهنمایی هایم منسجم تر باشند و باعث سردرگمی او نشوند.

این مستند بمنظور استفاده در ارائه شفاهی / کتبی (ارائه شفاهی با جزوه) نوشته شده است و نه برای ارائه کتبی.

سعی کرده ام از روده درازی بپرهیزم و جمله های کوتاه استفاده کنم تا دانش آموز مجبور نشود شش خط یکی جزوه را بخواند.

مهمترین فصل جزوه فصل اول است و هرچه به انتها میرویم از اهمیت فصل های کاسته میشود ، هرچند که حرفه ای تر میشوند. در حقیقت فصل های اول جزوه بعنوان پیشنیاز برای فصل های بعد هستند و صرف کردن وقت زیاد برای آنها کاملا معقول است.

از دانش آموز خواهش میکنم که برای حل کردن کارگاه ها وقت صرف کنند چون این بخش را جهت مستقل کردن شما از این جزوه نوشته ام و فکر میکنم مفید ترین بخش است . مشروط بر اینکه از اول جواب آن را نگاه نکنید و حتی سعی کنید از راهنمایی ها هم استفاده نکنید.

تا وقتی کتابها و جزوه ها را نبندید چیزی یاد نمی گیرید. یادگیری از تجربه حاصل میشود نه از کتاب.

تقدیم به آموزگارم

که من سرکش را
رام کرد، دوست
داشتن و زندگی
کردن زندگی خود را
به من آموخت.
و آن هنگام که به او
وابسته شدم
استقلالم را به من
بازگرداند ،
استقلالی که هرگز
نداشتم...

خدایا، گرامیدارش

برنامه نویسی شی گرا(OOP[1]) چیست؟
برنامه نویسی شی گرا بر اساس همان مفاهیمی است که در کودکی یاد گرفتیم. مفاهیمی از جمله اشیا، صفات، دسته ها و ... استفاده از این مفاهیم برنامه را بیشتر به دنیای واقعی شبیه میکند.
به طور خلاصه میتوان گفت که هر شیء دارای صفات و رفتارهایی است و هر شیء جزء یک دسته از اشیا است. برای مثال صندلی یک شیء است و صفاتی از قبیل رنگ ، جنس و غیره دارد. صندلی ها دسته بندی متنوعی دارند مثلا یک صندلی از دستة ،صندلی های دسته دار است ، یکی از دستة صندلی های بدون دسته است و ... و در نهایت تمام آنها از دستة صندلی ها هستند.
شاید در مورد دسته ها(کلاسها) مثال حیوانات بهتر باشد ؛ هرچند که از دید عمومی یک گربه به عنوان شیء کمی غلط به نظر میرسد اما از نظر برنامه نویسی شی گرا ایرادی ندارد. گربه از دسته گربه سانان است که آن نیز از دسته پستانداران است و ...

فصل اول :بسیار مختصر از محیط ویژوال بیسیک 6

پنجره New Project
این پنجره برای انتخاب نوع پروژه ای که میخواهیم انجام بدهیم است کار ما در اینجا تقریبا به standard exe محدود میشود و فقط در فصل آخر کتاب به ActiveX Control میپردازیم. پس خودتان را با این گزینه های متعدد گیج نکنید!
زبانه دوم برای باز کردن یک پروژه ایجاد شده است و سومی لیست پروژه هایی که اخیرا باز شده است را نشان میدهد(Figure 1-1)
برای شروع کار در زبانه New گزینه Standard Exe را انتخاب کنید و روی دکمه open کلیک کنید.

[1] Object-Oriented Programming

Figure 1-1

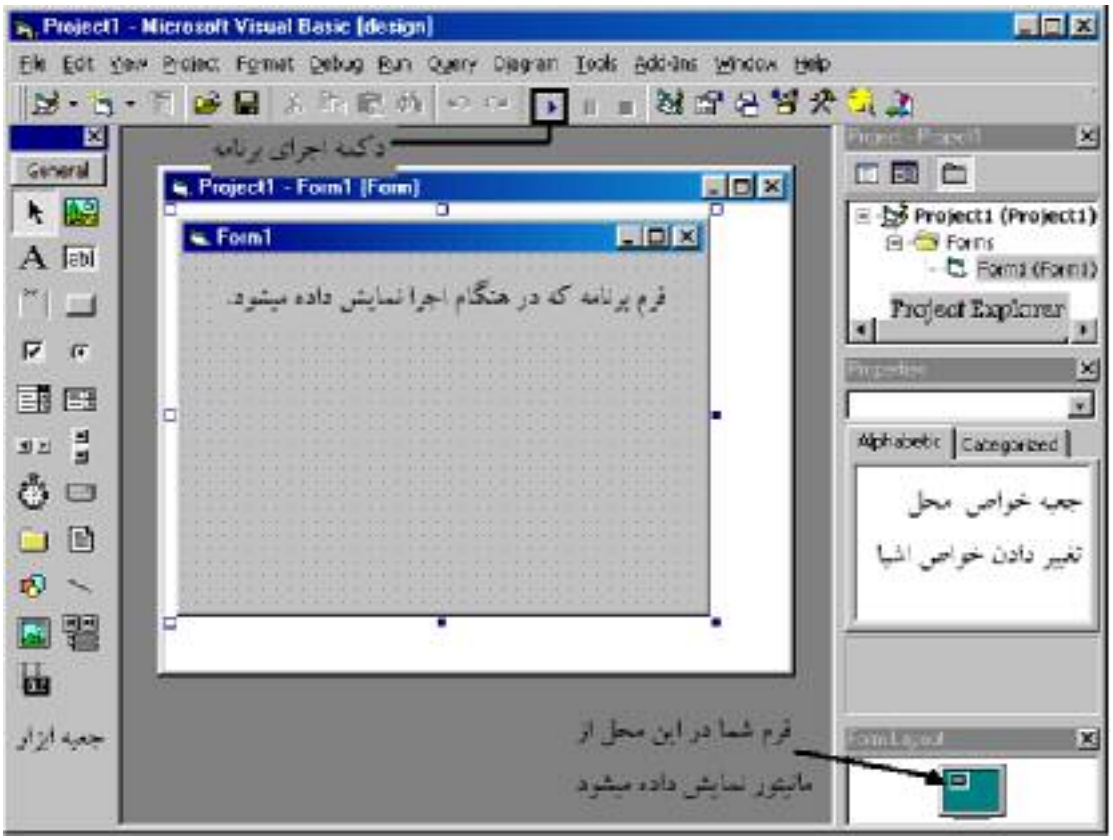


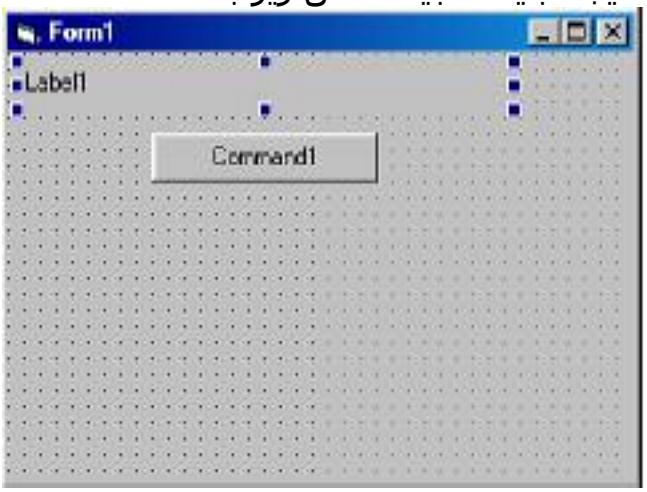
Figure 1-2

فصل دوم : کار با کنترول ها

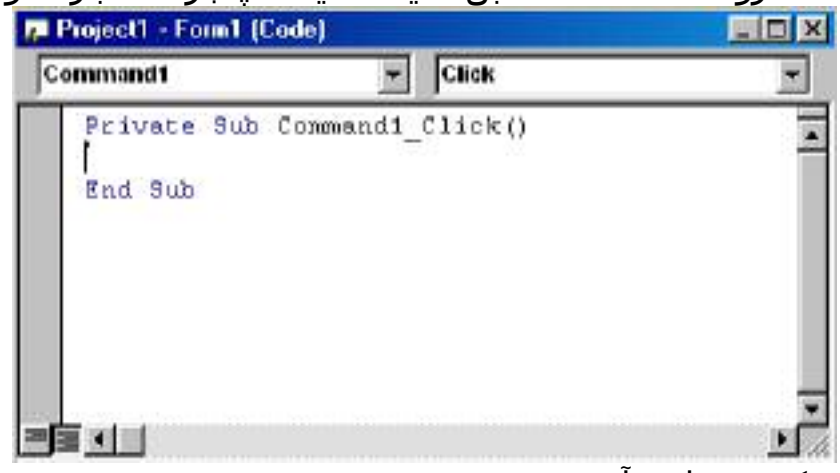
2-2:برنامه کلاسیک Hello World

کارمان را با مثال کلاسیک Hello World شروع میکنیم. از جعبه ابزار، ابزار های ,CommandButton Label را پیدا کنید و از هرکدام یکی روی صفحه قرار دهید.

نتیجه باید شبیه شکل زیر باشد



حالا روی Command1 دبل کلیک کنید تا پنجره کد باز شود

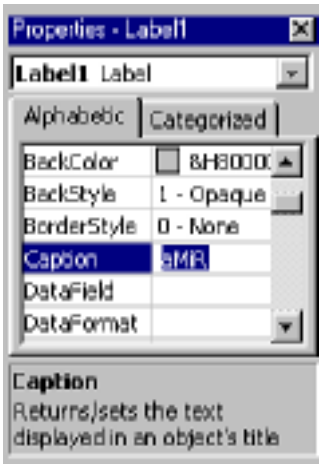


و کد زیر را در آن بنویسید :

```
Private Sub Command1_Click()
   Label1.Caption = "hello world"
End Sub
```

برنامه را اجرا کنید و پس از اجرای آن روی دکمة Command1 کلیک کنید.

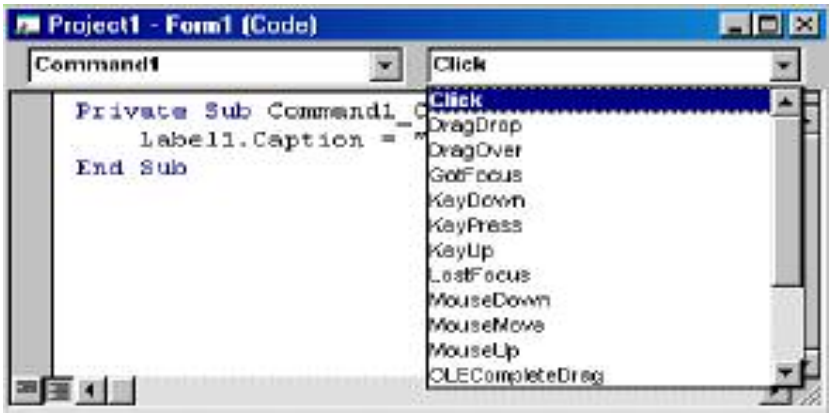
اگر برنامه اجرا نشد یا آنچه میخواستیم رخ نداد بار دیگر کد نوشته شده را به دقت تایپ کنید.
اگر به خاطر داشته باشید گفتیم که هر شی مجموعه ای از خواص دارد ؛ یکی از خواص شی Label ، Caption آن است . Caption همان متنی است که در Label نشان داده میشود.
اگر هنوز برنامه در حال اجرا است آن را ببندید تا به محیط طراحی بازگردید . حالا همان Label را سلکت کنید و در پنجره خواص (Properties) به دنبال Caption بگردید ومقدار آن را به نام خودتان تغییر دهید. حال اگر برنامه را اجرا کنید میبینید که در ابتدا نام شما نوشته شده است و میتوانید با زدن دکمه Command1 آن را به Hello Word تغییر دهید.



اما معنی کدی که نوشتیم چه بود. برای این منظور باید ابتدا به مفهوم رویداد (event) بپردازیم.رویداددر برنامه نویسی دقیقا به همان مفهومی استفاده میشود که در دنیای واقعی استفاده میشود.
وقتی روی یک دکمه کلید میکنیم ، میگوییم روی دکمه کلیک رخداد یا رویداد کلیک برای دکمه اتفاق افتاد.
در ویژوال بیسیک قطعه برنامه ای به نام Event Handler مسوول این است که رویداد ها را کنترل کند. این قطعه برنامه برای هر رویداد کارهای خاصی را انجام میدهد و از جمله این کارها صدا کردن یک تابع (زیربرنامه) نوشته شده توسط ما میباشد.
حالا مراحل کلیک شدن روی دکمه Command1 را با هم مرور میکنیم

1. کاربر روی دکمه کلیک میکند.
2. سیستم عامل پیغامی به برنامه ما میفرستد و ما را از رخدادن کلیک بر روی دکمه با خبر میکند.
3. Event Handler پیغام سیستم عامل را دریافت میکند و اگر ما زیر برنامه ای برای این رخداد نوشته باشیم آن را صدا میکند.
4. زیر برنامه ما عنوان Label تغییر میدهد.

شاید بپرسید آیا رویداد های دیگری هم برای یک دکمه اتفاق می افتد؟ بله؛ برای اینکه لیست رویدادهایی که قابل برنامه نویسی هستند ببینید روی click بالای پنجره کلیک کنید.



حالا که تابع ما اجرا شده است کد آن تابع به صورت ترتیبی اجرا میشود. میخواهیم خاصیت caption را برای شی Label1 تغییر دهیم ، برای اینکه شی و خاصیت آن را به هم ارتباط دهیم از '.' استفاده می کنیم پس Label1.Caption به معنی خاصیت Caption از Label1 است. پس مفهوم خط کدی که نوشتیم این میشود :
خاصیت Caption از شی Label1 را برابر مقدار رشته ای "Hello world" قرار بده.

تمرین :
1. برنامه Hello World را تغییر دهید تا متن Hello Visual Basic World را نمایش دهد.
2. برنامه را تغییر دهید تا وقتی روی Label1 دبل کلیک میکنیم نوشته درون آن پاک شود.

کارگاه :
1. یک جعبه متن (TextBox) و یک دکمه به برنامه اضافه کنید و برنامه را طوری تغییر دهید که با کلیک کردن روی دکمه جدید متن Label به متنی که در TextBox نوشته شده تغییر کند. توجه داشته باشیدکه اگر در زمان اجرا متن TextBox تغییر کند و سپس دکمه زده شود متن جدید در Label ریخته شود.
2. برنامه قبل را طوری تغییر دهید که در هر وقت متن TextBox تغییر کرد متن Label نیز تغییر کند.

راهنمایی :
0. وقتی نام یک شی را می نویسید و نقطه میزنید لیستی از خواص و متد های آن شی نشان داده می شود.
1. چه خاصیتی از TextBox حاوی متن(Text) است؟
2. آیا وقتی متن TextBoxتغییر(Change) میکند رویدادی برای آن پیش آمده ؟

پاسخ کارگاه :

1.
```
Private Sub Command2_Click()
   Label1.Caption = Text1.Text
End Sub
```
2.
```
Private Sub Text1_Change()
   Label1.Caption = Text1.Text
End Sub
```

2-3: بازی حدس زدن عدد

خوب! کمی هم تفریح؛ شما بازی دوست دارید؟ چطور است که بازی خودمان را خودمان بنویسیم البته این بازی به پای بازی هایی مثل Quake نمیرسد! ولی بازی جذابی است.

شرح برنامه : برنامه یک عدد تصادفی انتخاب میکند و کاربر باید با توجه به راهنمایی های برنامه آن عدد را به دست بیاورد.
حالا چشمهایتان را ببندید و کمی فکر کنید!
اولین قدم برای حل یک مساله تجزیه کردن آن مساله به مساله های کوچکتر است.

برای مثال مساله ما از سه بخش تشکیل شده:
1. انتخاب یک عدد تصادفی
2. وارد کردن عدد توسط کاربر
3. راهنمایی کردن کاربر

پس از حل کردن بخش های کوچکتر همه را به هم متصل میکنیم .

حل 1 :
این کار خیلی راحت تر از آن است که به نظر میرسد. تابع rnd یک عدد تصادفی بین صفر ویک ایجاد میکند. و تابع int بخش صحیح یک عدد را برمیگرداند. پس برای انتخاب یک عدد تصادفی بین یک تا ده میتوان از کد زیر استفاده کرد:

```
Int(Rnd * 10) + 1
```

برای اینکه نتیجه این خط را ببینید فعلا صورت مساله اصلی را فراموش کنید و یک پروژه جدید باز کنید (File\new project\ standard exe) یک Label روی صفحه قرار دهید و خط زیر را در رویداد dblclick آن بنویسید.

```
Private Sub Label1_DblClick()
    Label1.Caption = Int(Rnd * 10) + 1
End Sub
```

آن را اجرا کنید و چند بار روی نوشته آن دبل کلیک کنید و عدد ها را بخاطر بسپارید. برنامه را ببندید و دوباره اجرا کنید. وای! شما که نمیخواهید هر بار که برنامه شما اجرا میشود همان عدد قبلی را انتخاب کند؟ پس باید قبل از انتخاب کردن عدد خط زیر را اضافه کنید

```
Randomize Timer
```

دستور Randomize تعیین میکند که عدد تصادفی برحسب چه عددی انتخاب شود. و Timer زمان سیستم است. پس عدد تصادفی که انتخاب میشود بر حسب زمان کامپیوتر است و دیگر با اجرای برنامه تکراری نمیشود.

متغیر ها در ویژوال بیسیک

تعریف متغیر ها : ویژوال بیسیک بر خلاف زبانهایی از جمله پاسکال و سی کاربر را مجبور به تعریف متغیر نمیکند. این مساله شاید برای برنامه نویسان تازه کار به معنی رهایی از دردسر های تعریف متغیر باشد ولی در عمل مشکلات زیادی پیش می آورد.(اگر نام متغیر را اشتباه تایپ کنید کامپایلر پیام خطایی به شما نمیدهد و آن را یک متغیر جدید در نظر میگیرد.) برای اینکه به ویژوال بیسیک بگویید که شما همه متغیر ها را تعریف میکنید دستور زیر را بالای همه دستور ها بنویسید.

```
Option Explicit
```

```
Private Sub Label1_DblClick()
  Randomize Timer
  …
```

برای اینکه ویژوال بیسیک این دستور را به طور خودکار به همه بخشهای برنامه شما اضافه کند به منوی Tools\Option بروید و گزینه Requare Variable Declaration را انتخاب کنید.

برای تعریف کردن متغیر از دستور Dim به صورت زیر استفاده میشود:

```
Dim VarName as VarType
```

برای تعریف متغیر RndNum از نوع صحیح(integer) :

```
Dim RndNum as Integer
```

لیست انواع متغیر ها در ادامه آمده است.

دید و مدت عمر متغیر :

مقدار هر متغیر در یک محدوده خاص از برنامه (احتمالا همه برنامه) قابل دیدن است که به آن محدوده ،محدوده دید متغیر گویند. تعیین محدوده بیش از هر چیز به محل تعریف متغیر بستگی دارد. متغیرهایی که داخل یک تابع تعریف میشوند فقط داخل تابع دیده میشوند. اگر بخواهیم یک متغیر در همه تابع های فرم دیده شود باید آن را در قسمت Declarations تعریف کنیم (محلی را که Option Explicit را نوشتید بخاطر دارید؟)

دو کلمه کلیدی public و private نیز در تعیین کردن محدوده دید موثرند. این کلمه های کلیدی به جای dim و فقط در بخش Declarations استفاده میشوند. Public محدوده دید متغیر را به تمام برنامه گسترش میدهد و Private آن را به فرم محدود میکند(مانند dim).

مدت زمانی را که یک متغیر مقدار فعلی خود را حفظ میکند مدت عمر متغیر گویند. متغیر هایی که در بخش Declarations تعریف میکنند طول عمری برابر با فرم دارند.یعنی با ایجاد فرم ، ایجاد میشوند و با ازبین رفتن فرم نابود میشوند(حافظه را به سیستم برمیگردانند). متغیر هایی که داخل یک تابع تعریف میشوند طول عمری برابر با طول عمر تابع دارند. یعنی با از بین رفتن تابع از بین میرند. به مثال زیر توجه کنید (تایپ کنید):

```
Private Sub Command1_Click()
  Dim x As Integer
  x = x + 1
  Label1.Caption = x
End Sub
```

انتظار دارم که از تعرف تابع Private Sub Command1_Click() و خط Label1.Caption متوجه شده باشید که به یک دکمه و یک Label احتیاج داریم. برنامه را اجرا کنید و روی دکمه یک بار کلیک کنید.

متغیر x تعریف میشود و به طور پیش فرض به آن مقدار صفر داده میشود. مقدار آن برابر با یکی بیشتر از مقدار قبلی میشود(x=x+1) . و در Label نمایش داده میشود. نتیجه همان است که میخواستیم. عدد 1 نشان داده میشود. اما اگر یک بار دیگر هم کلیک کنیم عدد چند میشود؟ تغییری نمیکند! چرا؟

به این دلیل که پس از رسیدن به خط End Sub متغیر x به پایان عمر خود میرسد و حافظه ای که گرفته بود به سیستم عامل بازمیگرداند. وقتی بار

دوم کلیک میکنیم متغیر دوباره تعریف میشود و مقدار اولیه صفر میگیرد و ...
حالا کلمه dim را با کلمه کلیدی static جایگزین کنید. و برنامه را اجرا کنید:

```
Private Sub Command1_Click()
  Static x As Integer
  …
```

و اجرا کنید. معجزه شده! عددی که تغییری نمیکرد حالا اضافه مشود.
کلمه کلیدی static به کامپایلر میگوید که ما این متغیر را چقدر دوست داریم و نمیخواهیم مرگ آن را ببینیم! دید متغیر های static مانند dim و طول عمر آنها برابر فرم است. به این ترتیب با خروج از تابع متغیر از بین نمیرود و با بازگشت بعدی به تابع مقدار قبلی خود را حفظ کرده است.
انواع متغیر ها :

Data type	Storage size	Range
Byte	1 byte	0 to 255
Boolean	2 bytes	True or False
Integer	2 bytes	-32,768 to 32,767
Long (long integer)	4 bytes	-2,147,483,648 to 2,147,483,647
Single (single-precision floating-point)	4 bytes	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values
Double (double-precision floating-point)	8 bytes	-1.79769313486232E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values
Currency (scaled integer)	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Decimal	14 bytes	+/-79,228,162,514,264,337,593,543,950,335 with no decimal point; +/-7.9228162514264337593543950335 with 28 places to the right of the decimal; smallest non-zero number is +/-0.0000000000000000000000000001
Date	8 bytes	January 1, 100 to December 31, 9999
Object	4 bytes	Any Object reference
String (variable-length)	10 bytes + string length	0 to approximately 2 billion
String (fixed-length)	Length of string	1 to approximately 65,400
Variant (with numbers)	16 bytes	Any numeric value up to the range of a Double
Variant (with characters)	22 bytes + string length	Same range as for variable-length String

حل 2 :

این بخش، بخش رابط کاربر[2] است و به نظر خیلی از برنامه نویسان مهمترین بخش برنامه است. چون کاربر کد برنامه را نمیبیند. و حقیقت این است که کاربران برنامه ای را که رابط کاربر خوبی نداشته باشد خیلی زود پاک میکنند. و به اینکه خوب کار میکند یا بد اهمیتی نمیدهند. و برعکس برنامه ای را که رابط کاربر خوبی دارد اما کمی کند است نگه میدارند.
معمولا وقتی صحبت از رابط کاربر میشود همه به جنبه زیبایی برنامه توجه میکنند و راحتی کار با برنامه را فراموش میکنند یا فدای زیبایی میکنند.
باید توجه کنید که ظاهر برنامه باید متناسب با کار برنامه باشد مثلا تصور کنید اگر ویژوال بیسیک دارای شکل و شمایلی مانند windows media player بود چه کسی می توانست با آن کار کند!

نکات زیر را هنگام طراحی رابط کاربر در نظر بگیرید

1. عادت کاربر را تغییر ندهید.(استاندارد ها را رعایت کنید) مثلا کاربر عادت دارد که exit آخرین گزینه از اولین منو سمت چپ (منوی file) است اگر شما آن را جابجا کنید ممکن است باعث عصبانیت بعضی از کاربرانتان شود.
2. طیف کاربران خود را بشناسید و رابط کاربر را برای آنها طراحی کنید.معمولا کاربران مبتدی علاقه به سرک کشیدن در برنامه و دیدن همه طرف آن دارند در حالی که کاربران حرفه ای میخواهند هرچه زودتر کارشان با برنامه تمام شود.
3. از سوال کم کن و بر undo افزای!
4. کاربر احمق نیست . (وقتی میگوید : فلان فایل را پاک کن مسلما منظورش این است که آن را پاک کن!)
5. رابط کاربر باید قابل تنظیم باشد. کاربر ها سلیقه های مختلفی دارند . باید بتوانند برنامه شما را برحسب سلیقه خودشان تنظیم کنند اما پیش فرض باید بهترین تنظیم باشد تا کاربر مجبور نشود در اولین برخورد با برنامه به منوی setting یا option برود.
6. "سواره از پیاده خبر نداره." برای اینکه یک طراح واسط کاربر خوب باشید اول باید یک کاربر باشید. پس اگر کاربر خوبی برای windows نیستید همین الان ویژوال بیسیک را ببندید و مدتی را به کار کردن با ویندوز بپردازید.
7. حافظه برنامه خود را تقویت کنید. برنامه شما باید بخاطر داشته باشد که آخرین بار کاربر چه تنظیماتی را انتخاب کرده بوده.

حل 3 :
می خواهیم به کاربر بگوییم که عددی که انتخاب کرده بزرگتر از عدد انتخابی ما است یا کوچکتر.
برای تصمیم گیری از دستور شرطی if[3] استفاده میکنیم :

```
If Text1.Text > RndNum Then
   Label1.Caption = "random# < " + Text1.Text
ElseIf Text1.Text < RndNum Then
```

[2] UI (User Interface)
[3] برای اطلاعات بیشتر در مورد دستورات بیسیک به ضمیمه مراجعه کنید.

```
   Label1.Caption = "random# > " + Text1.Text
Else
   Label1.Caption = "u r winner"
End If
```

اگر عبارت منطقی بعد از if برابر با true (صحیح) باشد. دستورات پس از then اجرا میشوند و اگر false(نادرست) باشد به اولین خط پس از دستورات شرط میرود که در اینجا ElseIf است. اگر شرط آن نیز برقرار نباشد else اجرا میشود. از اولین then تا elseif بخش دستورات if است که اگر شرط برقرار باشد اجرا میشوند. و به آن یک بلوک میگوییم و برای وضوح بیشتر با تورفتگی آن را مشخص میکنیم.
احتمالا اگر تا بحال با C یا پاسکال برنامه نوشته باشید برایتان این سوال پیش آمده که چگونه یک رشته را با یک عدد صحیح مقایسه کرده ایم. ویژوال بیسیک در این مورد نیز مانند تعریف متغیر سختگیری نمیکند. یک تبدیل نوع خودکار text1.text را به عدد صحیح تبدیل می کند. اما بهتر است از تبدیل نوع دستی استفاده کنید مگر در مواردی که تبدیل نوع بسیار واضح است. برای اینکه به مشکلات تبدیل نوع خودکار پی ببرید به منوی view\imidiate window بروید. پنجره کوچکی پایین صفحه باز میشود. در این پنجره میتوانید دستورات خود را تایپ کنید و نتیجه آن را خط به خط ببینید.
دستور زیر را تایپ کنید :

```
Print 2 + "3"
```

نتیجه عدد 5 است. ولی برای این دستور چطور؟

```
Print "2" + "3"
```

نتیجه 23 میشود. یعنی بجای جمع کردن دو عدد صحیح دو رشته را به هم الحاق کرده است.
برای تبدیل به انواع مختلف از تابع های زیر استفاده میکنیم

CBool(expression)	تبدیل به نوع boolean
CByte(expression)	تبدیل به نوع Byte
CDate(expression)	تبدیل به نوع Date
CDbl(expression)	تبدیل به نوع Double
CDec(expression)	تبدیل به نوع Decimal
CInt(expression)	تبدیل به نوع Integer
CCur(expression)	تبدیل به نوع Currency
CSng(expression)	تبدیل به نوع Single
CStr(expression)	تبدیل به نوع String
CVar(expression)	تبدیل به نوع Variant
CLng(expression)	تبدیل به نوع Long

نتیجه

```
Print Cint("2") + "3"
```

را ببینید.

سه... دو... یک... شروع میکنیم!

اگر هنوز فرم بازی را طراحی نکرده اید بهتر است زودتر این کار را بکنید. فرمی که من طراحی کردم همان است که اول این فصل دیدید. حتما فرم

شما خیلی زیباتر از فرم ساده ایی که من طراحی کرده ام شده است.
میخواهیم کاربر علاوه بر اینکه میتواند با کیبرد عدد مورد نظرش را وارد کند ، با موس هم بتواند این کار را بکند برای همین یک کیبرد کوچک روی فرم طراحی کرده ایم دکمه سمت چپ پایین (clr) برای پاک کردن متن textbox است که نوشتن کد آن را به عهده خودتان میگذارم.
دکمه (cp) هم کار مقایسه و راهنمایی کاربر را انجام میدهد که شرط های آن را قبلا نوشتیم. و اما دکمه های عددی، میتوانیم در هر یک از این دکمه ها کدی شبیه زیر بنویسیم :

```
text1.Text = text1.Text + "1"
```

البته نتیجه بدست آمده مطلوب است ولی باید این کد را برای تک تک این ده دکمه بنویسیم و فقط یک عدد را در آن تغییر دهیم. این کار شدنی است اما خوشایند نیست. بخاطر داشته باشید که هرگاه بیش از آنکه کدی را تایپ کنید، copy و paste میکردید حتما راه بهتری برای نوشتن آن کد وجود دارد!

آرایه ای از کنترول ها

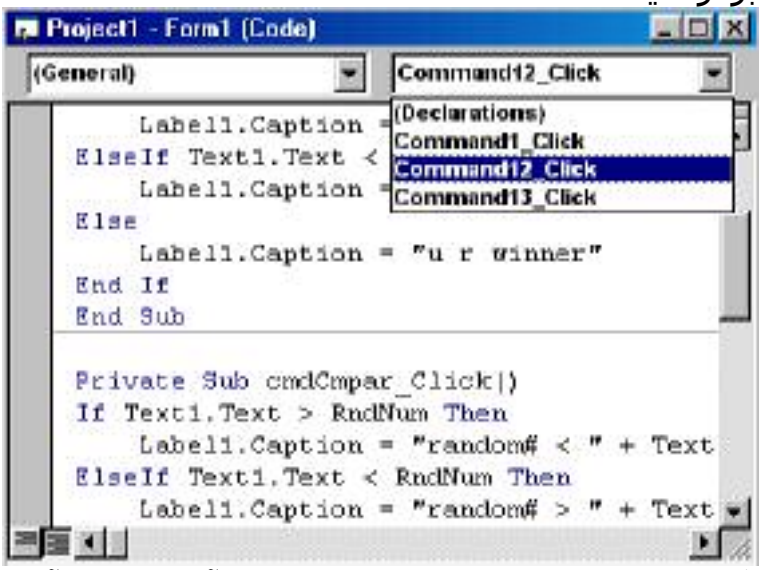
در خواص دکمه یک (Name) را پیدا کنید و مقدار آن را به cmdKeyboard تغییر دهید. تابحال ما نام اشیا را عوض نمیکردیم. اما این نام چیست؟ نام Label1 ، Label1.Caption=1 در کد ، مشخصه تفکیک اشیا از یکدیگر است. به برنامه ای که طراحی کرده اید نام شی و Caption خاصیت آن است. اگر نام اشیا را عوض نکرده باشید این همه دکمه با نامهای command1 ta 12
باعث سردرگمی میشود. شاید بهتر باشد نام دکمه cp را هم به cmdCmpare تغییر دهیم ، این قابل درک تر است. بهتر است سه کاراکتر اول نام اشیا طوری باشد که بتوان به راحتی نوع آن را تشخیص داد برای مثال cmd برای command و lbl برای Label و... جدول زیر پیشوند های توصیه شده مایکروسافت برای نامگزاری است :

Control type	prefix	Example
Check box	chk	chkReadOnly
Combo box, drop-down list box	cbo	cboEnglish
Command button	cmd	cmdExit
Common dialog	dlg	dlgFileOpen
Directory list box	dir	dirSource
Drive list box	drv	drvTarget
File list box	fil	filSource
Form	frm	frmEntry
Frame	fra	fraLanguage
Image	img	imgIcon

ImageList	ils	ilsAllIcons
Label	lbl	lblHelpMessage
List box	lst	lstPolicyCodes
Menu	mnu	mnuFileOpen
Option button	opt	optGender
Picture box	pic	picVGA
ProgressBar	prg	prgLoadFile
RichTextBox	rtf	rtfReport
StatusBar	sta	staDateTime
Text box	txt	txtLastName
Timer	tmr	tmrAlarm

اگر نام بعضی از این کنترول ها برایتان مفهومی ندارد خودتان را ناراحت نکنید. فقط شماره این صفحه را برای مراجعه های بعدی بخاطر داشته باشید[4].

نکته : اگر برای رویداد های یک شی کدی نوشته باشید و سپس نام آن شی را عوض کنید دیگر کد قبلی در رویداد آن کنترول نیست. اما تابع آن در بخش General نگه داری میشود و میتوانید کد را با copy و paste کردن برگردانید.



اگر نام دو یا چند کنترول یکی باشد آنها با هم آرایه میشوند یعنی همه آنها یک نام دارند و هر کدام اندیس منحصر به فرد دارد. حالا برای این کار نام(Name) دکمه دو را هم به cmdKeyboard تغییر دهید و به سوال "آیا میخواهید یک آرایه از کنترول ها درست کنید؟" جواب مثبت بدهید.

[4] این جدول را از MSDN آورده ام و عنوان صفحه آن در MSDN، Object Naming Conventions است.

حالا خاصیت اندیس(index) در این کنترل ها را نگاه کنید. باید اندیس دکمه یک ، صفر باشد و دکمه دو ، یک. البته بهتر بود از دکمه صفر شروع میکردیم تا اندیس هرکدام برابر با caption آن شود. اگر شما هم مانند من captionها را 1 2 3 ... گذاشته اید مساله اندیس زیاد مهم نیست اما اگر captionهای شما چیز دیگری است زودتر اندیس ها را به ترتیب کنید. مانند نام میتوان اندیس را هم تغییر داد.
وقتی از چند کنترل آرایه درست میکنید برای رویداد های تمام آنها یک تابع صدا میشود.

```
Private Sub cmdKeyboard_Click(Index As Integer)

End Sub
```

پس باید راهی برای آنکه بفهمیم رویداد برای کدام کنترل اتفاق افتاده است وجود داشته باشد. متغیر اندیس در اعلان تابع در بر دارنده اندیس کنترلی است که رویداد برای آن اتفاق افتاده.
و کدی برای تمام دکمه ها :

```
Private Sub cmdKeyboard_Click(Index As Integer)
   Text1.Text = Text1.Text + cmdKeyboard(Index).Caption
End Sub
```

روش دستری به خواص کنترول های آرایه همانطور که می بینید با ControlName(ControlIndex).Property است. وقتی روی دکمه 2 کلیک میکنید این تابع با index=1 فراخوانی میشود که caption آن برابر 2 است.
اگر شما caption دکمه ها را چیزی غیر از عدد گذاشته اید یا مثلا عدد فارسی نوشته اید و میخواهید عدد انگلیسی در textbox بنویسد . بهتر است اندیس ها را آنطور که گفته شد به ترتیب کنید و از خط زیر استفاده کنید :

```
   Text1.Text = Text1.Text + CStr(Index)
```

برای شروع بازی فقط باید یک عدد انتخاب کنیم
برای اینکه به محض اجرای برنامه عدد تصادفی انتخاب شده باشد کد مربوط به انتخاب عدد را در form_load مینویسیم. این رویداد زمانی که فرم در حافظه بار میشود اجرا میشود و محلی مناسب برای مقدار دهی های اولیه است.

```
Private Sub Form_Load()
   Form1.Caption = "Find Number"
   Text1.Text = ""
   Randomize Timer
   RndNum = Int(Rnd * 10) + 1
End Sub
```

برای اینکه بعد از هر بار مقایسه textbox خالی شود و آماده دریافت عدد بعدی باشد. میتوانید پس از مقایسه خودتان تابع مربوط به کلیک شدن دکمه clr را فراخوانی کنید. برای این کار کافی است نام این تابع را بنویسید . تابع کلیک کنترول clr من بصورت زیر است :

```
Private Sub Command11_Click()
   Text1.Text = ""
```

```
End Sub
```

پس باید تابع cp را به

```
…
End If
Command11_Click
End Sub
```

تغییر دهم.
وقتی که کاربر کاری را با کیبرد انجام میدهد ترجیح میدهد دستش را از روی کیبرد بلند نکند. بهتر است که بتواند با زدن کلید enter کار مقایسه را انجام دهد و با زدن Esc کار پاک کردن را.
دکمه ها یک خاصیت default دارند که اگر آن را true کنید دور دکمه یک حاشیه مشکی میکشد. وقتی کاربر کلید enter را بزند رویداد کلیک برای این دکمه فراخوانی میشود. و خاصیت cancel اگر برای دکمه ای true باشد وقتی کلید Esc زده می شود رویداد کلیک برای آن دکمه فراخوانی میشود. در هر فرم فقط یک دکمه default وفقط یک دکمه cancel میتواند باشد، چرا؟
حالا شما میدانید و دکمه ها! کدام را default و کدام را cancel میکنید؟

تمرین :

1. ابتدای کار در Label ، عبارت Label1 نوشته شده است . آن را حذف کنید.
2. بازی ما یک مشکل دارد. فقط یک بار میتوان آن را بازی کرد! اگر بخواهید یک دست دیگر هم بازی کنید باید از بازی خارج شوید و آن را دوباره اجرا کنید. کاری کنید که برای انتخاب عدد جدید نیازی به اجرای مجدد نباشد. (توجه داشته باشید که همه چیز باید مانند این باشد که برنامه تازه اجرا شده است.)
3. در پیغام مربوط به برنده شدن کاربر به او اطلاع دهید که با چند بار حدس زدن برنده شده است.
4. کاربر میتواند با موس عددی مانند 0435 تایپ کند، این امکان را از او بگیرید.
5. راهی برای اینکه زود تر برنده شوید پیدا کنید (;

کارگاه :

1. وقتی کاربر برنده میشود هنوز هم میتواند عدد جدیدی وارد کند و مقایسه انجام دهد. این امکان را از او بگیرید.
2. دکمه ایی برای اینکه کاربر بتواند رقم آخری که وارد کرده پاک کند ایجاد کنید.
3. کاری کنید که وقتی عمل مقایسه انجام میشود textbox خالی نشود و در عوض زمانی که کاربر اولین رقم را بعد از مقایسه وارد میکند textbox خالی شود.
4. این امکان را به کاربر بدهید که محدوده عدد تصادفی را تعیین کند. و برای حدس های خارج از محدوده پیغام مناسب نشان داده شود.

5. اگر کاربر درtextbox چیزی غیر از عدد وارد کند برنامه با دچار run time error میشود این مشکل را حل کنید.(عددهای اعشاری و اعدادی که با صفرشروع میشوند نیز نامطلوبند.)

راهنمایی :

0. از تغییر دادن خواصی که نمیشناسید نترسید! شما نمیتوانید چیزی را خراب کنید. هرچه می بینید تغییر بدهید و نتیجه آن را ببینید.
1. تمام کنترول ها خاصیتی به نام enable دارند که مقداری بولیین([5] boolean) میگیرد
2. اگر عددی را به 10 تقصیم کنیم چه میشود؟

1-3. به Selstart و SelLength درخواص textbox سری بزنید. این خواص در Properties Window نشان داده نمی شوند و تنها زمان کد نویسی (درستتر این است که بگوییم زمان اجرا run time) دردسترس هستند.

2-3. اگر بدانیم قبل از زدن کلید عددی ، مقایسه انجام شده یا خیر میتوانیم در باره اینکه متن را پاک کنیم یا نه تصمیم بگیریم. با یک متغیر سراسری (همان که در Declaration تعریف میشود) موافقید؟

4. شاید دو textbox و کمی تغییر در خط انتخاب عدد تصادفی مساله را حل کند.

1-5. میتوانید از کاربر خواهش کنید که همچین کاری نکند!

2-5. قبل از تبدیل کردن نوشته کاربر به عدد و مقایسه آن و مکانهای دیگری که احتمال خطا وجود دارد ببینید که آنچه کاربر وارد کرده واقعا عدد است یا خیر(سری به تابعIsNumeric در help ویژوال بیسیک بزنید.) اگر عدد نبود شدیدا به کاربر اخطار بدهید که دیگر جرات همچین جسارتی نکند!(این روش کاملا ضد کاربر است و بهتر است از آن پرهیز کنید)

3-5. کاربر نتواند غیر از عدد وارد کند. پس وقتی کاربر مشغول وارد کردن عدد است (دکمه ها را فشار(keypress) میدهد) باید کاری انجام داد.

پاسخ کارگاه :

1.

```
Else
   Label1.Caption = "u r winner"
   Text1.Enabled = False
   Dim i As Integer
   For i = 0 To 9
      cmdKeyboard(i).Enabled = False
   Next
   cmdCmpar.Enabled = False
End If
```

[5] متغیري متطقي است که فقط true یا false میگیرد.

اگر شما راهی برای شروع مجدد بازی تعبیه کرده اید باید زمانی که بازی را دوباره آغاز میکنید تمام این کنترل ها را enable کنید(enabled=true)

2.
```
Text1.Text = CDbl(Text1.Text) \ 10
```
برای تقسیم صحیح از \ و برای تقصیم اعشاری از / و برای باقیمانده تقسیم از mod استفاده میکنیم.

3-1.
```
Private Sub cmdCmpar_Click()
…
End If
Text1.SelStart = 0
Text1.SelLength = Len(Text1.Text)
End Sub
Private Sub cmdKeyboard_Click(Index As Integer)
   Text1.SelText = cmdKeyboard(Index).Caption
End Sub
```

3-2.
```
Dim cmpred As Boolean

Private Sub cmdCmpar_Click()
…
End If
cmpred = True
End Sub
Private Sub cmdKeyboard_Click(Index As Integer)
   If cmpred Then Text1.Text = "": cmpred = False
  …
End Sub
```

4.
```
Int(Rnd * (CDbl(txtEnd.Text) - CDbl(txtbegin.Text))) + CDbl(txtbegin.Text)
```
توجه داشته باشید که این textbox ها فقط باید شامل عدد باشند.

5-1.
```
Private Sub Form_Load()
…
Label1.Caption = "جووون من فقط عدد تایپ کن"
End Sub
```

5-2.
```
Private Sub cmdCmpar_Click()
   If Not IsNumeric(Text1.Text) Then Label1.Caption = "just integer": Exit Sub
End Sub
```
البته این کد مشکل runtime error ما را حل میکند اما همه خواسته ما را بر آورده نمی کند چون کاربر میتواند عدد اعشاری و عددی که با صفر شروع میشود وارد کند.

5-3.
```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
    If Not (Chr(KeyAscii) >= 0 And Chr(KeyAscii) <= 9 Or KeyAscii = 8) _
        Then KeyAscii = 0
End Sub
```

کمی توضیح : تمام کاراکتر های کیبرد دارای کدی هشت بیتی به نام کد اسکی هستند . با دستور chr() میتوان این کد را به خود کاراکتر تبدیل کرد .(شاید بهتر باشد print chr(x) را در پنجره immidiate آزمایش کنید. بجای x هر بار یک عدد قرار دهید.) چون تغییر دادن متن(text) بعد از خروج از تابع keypress است اگر ما کد اسکی را تغییر دهیم کاراکتری که ما کد اسکی آن را داده ایم نمایش داده میشود. در اینجا برای هر کاراکتری که بین صفر تا 9 یا کد اسکی آن 8 (کلید بک اسپیس) نباشد؛ کد اسکی برابر با صفر میشود. که هیچ کاراکتری ندارد (کاراکتر پوچ)

پروژه برنامه نویسی :

برنامه ای بنویسید که شما عددی در یک بازه مشخص انتخاب کنید و کامپیوتر آن عدد را پیدا کند. برنامه شما با چند بار حدس زدن عدد را پیدا میکند؟

3-3 : برنامه نمایش تصویر

در این برنامه تقریبا تمام ابزارهای استاندارد را استفاده می کنیم.

شرح برنامه : یک برنامه برای نمایش دادن تصاویر گرافیکی.

شرح بیشتری لازم نیست. مقصود ما نمایش دادن تصاویر گرافیکی است و همه چیز بستگی به سلیقه شخصی و حرفه ای شما دارد. جواب این سوال میتواند برنامه ای با یک textbox برای گرفتن آدرس فایل، یک دکمه و یک Image برای نمایش دادن تصویر باشد یا شاید برنامه عمومی و مشهوری مانندACDSee .

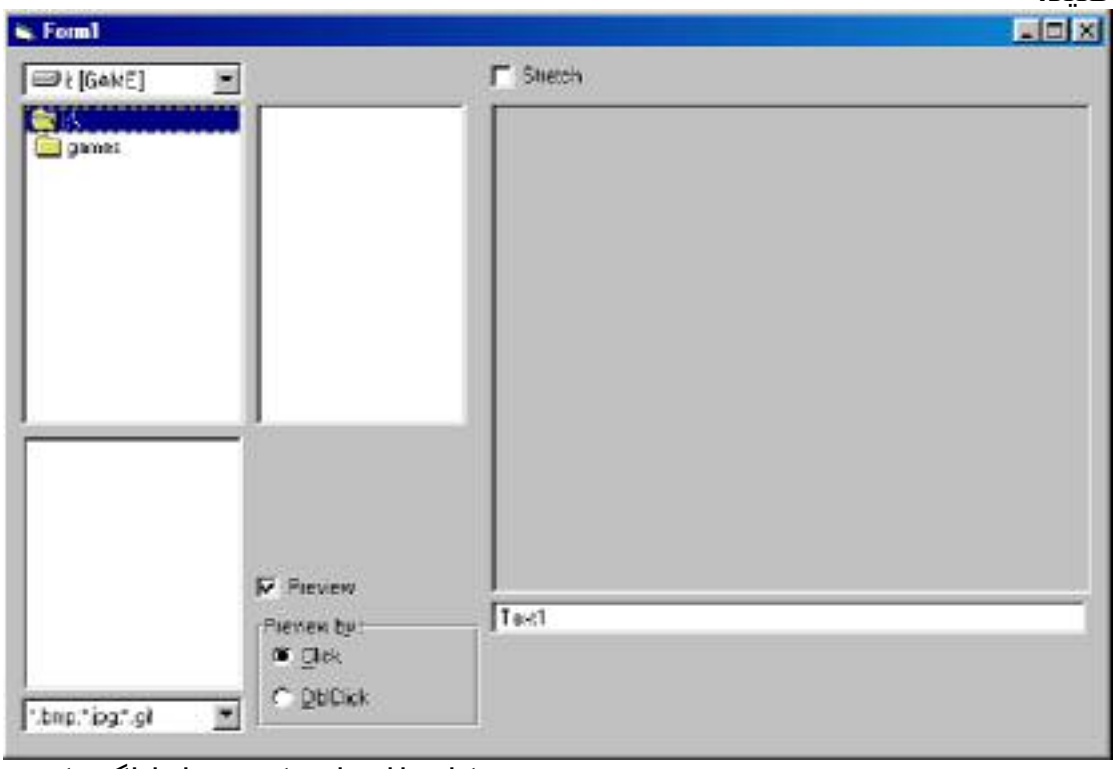
زمان بستن کتاب رسیده! خودتان را از شر این نوشته ها خلاص کنید و چند ساعتی را به فکر کردن و بازی کردن با ابزارهای ویژوال بیسیک بپردازید. کشف کردن همیشه از خواندن موثرتر است. و بخاطر داشته باشید که تا وقتی کتاب باز است چیزی یاد نمیگیرید.(MSDN(help-e VBرا فراموش نکنید.)

ابزارهایی که من استفاده کردم عبارتند از :

کار شی	نام شی	نوع شی
نمایش تصویر	imgPreview	Image
نمایش لیست درایو های موجود	drvImage	DriveListBox
نمایش دایرکتوری های موجود در درایو	dirImage	DirListBox
نمایش فایلهای موجود در دایرکتوری	filImage	FileListBox
نگه داشتن لیست فایلهایی که قبلا باز شده. برای دسترسی سریع	lstHistory	ListBox
برای محدود کردن فایل ها به پسوند های خاص	cboFilter	ComboBox
گزینش روش نمایش	optPreview(2)	OptionButton
عکس نشان داده شود یا خیر	chkPreview	CheckBox
نمایش آدرس کامل فایل	txtAddress	TextBox

Frame	fraPreview	دربردارنده OptionButtonها

فعلا برنامه ای که خودتان طراحی کرده اید ذخیره کنید تا وقتی درس تمام شد همان برنامه ای که خودتان طراحی کرده اید را کامل کنید.
این ابزارها را به سلیقه خودتان روی صفحه قرار دهید و برنامه را اجرا کنید.



DriveListBox و DirListBox و FileListBox همه کار ظاهرا میکنند . اما اگر یکی را تغییر دهید متوجه میشوید که به هم متصل نیستند.

```
Private Sub dirImage_Change()
   filImage.Path = dirImage.Path
End Sub

Private Sub drvImage_Change()
   dirImage.Path = drvImage.Drive
End Sub
```

با این دو خط هر سه به هم متصل میشوند و شما میتوانید تمام فایل های کامپیوتر خود را ببینید. این دو رویداد به ترتیب وقتی که dirImage و drvImage تغییر کنند اجرا میشوند.
FileListBox.Path : آدرس محلی که FileListBox باید فایل های آن را نمایش دهد.
DirListBox.Path : آدرس شاخه ای که DirListBox باید زیرشاخه های آن را نشان دهد.
DriveListBox.Drive : آدرس درایوی که DriveListBox نشان میدهد.

FileListBox.Pattern : تعیین کننده فایلهایی که در FileListBox نشان داده میشود. مقدار پیش فرض "*.*" است .برای اینکه فایلهای exe و dll را نشان دهد باید آن را به"*.exe;*.dll" تغییر داد.
ListBox ابزاری برای نگه داری و نمایش یک لیست از رشته ها.
چگونه میتوان به ListBox درج کرد؟
برای آزمایش کلیدی روی فرم قرار دهید و کد زیر را در آن بنویسید.

```
Private Sub Command1_Click()
   lstHistory.AddItem "1st"
   lstHistory.AddItem "2nd"
End Sub
```

هر دو آیتم به ترتیب در لیست درج میشوند. البته ما هنگام درج میتوانیم تعیین کنیم که آیتم جدید در چه خانه ای درج شود و اگر انتخاب نکنیم در انتها درج میشود. مثال زیر آیتم دوم را قبل از آیتم اول درج میکند.

```
Private Sub Command1_Click()
   lstHistory.AddItem "1st"
   lstHistory.AddItem "2nd", 0
End Sub
```

برای حذف کردن آیتم دوم از لیست :

```
   lstHistory.RemoveItem 1
```

برای گرفتن اندیس آیتم انتخاب شده :

```
   lstHistory.ListIndex
```

برای حذف کردن تمام لیست :

```
   lstHistory.Clear
```

برای گرفتن مقدار آیتم اول :

```
   Var = lstHistory.List(0)
```

برای گرفتن مقدار آیتم انتخاب شده؟
کمی تمرین کنید و سپس آن دکمه را حذف کنید.

ComboBox ترکیبی از ListBox و TextBox است. که سه شکل(Style) مختلف دارد
0.Dropdown Combo 1.Simple Combo 2.Dropdown List
اولی یک TextBox و ListBox بازشونده ، در دومی هر دو زیر هم قراردارند و سومی ListBox باز شونده با TextBox فقط خواندنی است. که برای محدود کردن حق انتخاب به مواردی خواص است.در اینجا ما از Dropdown List استفاده میکنیم چون نمیخواهیم کسی فرمت فایلی غیر از آنها که ما تعیین کرده ایم انتخاب کند.
کار با combo تا حد بسیار زیادی مشابه لیست است. فقط اینکه متن combo در خاصیتtext است.

```
Private Sub Form_Load()
   cboFilter.AddItem "*.bmp;*.jpg;*.gif"
   cboFilter.AddItem "*.bmp"
   cboFilter.AddItem "*.jpg"
   cboFilter.AddItem "*.gif"
   cboFilter.Text = cboFilter.List(0)
```

```
End Sub
Private Sub cboFilter_Click()
  filImage.Pattern = cboFilter.Text
End Sub
```

خط آخر در form_load (تابعی که قبل از نشان داده شدن فرم فراخوانی میشود) بمنظور انتخاب کردن آیتم اول بطور پیش فرض است.
رویداد کلیک در combo زمانی اجرا میشود که یک آیتم انتخاب شود. و وقتی که آیتم انتخاب شد متن آن در FileListBox.pattern قرار میگیرد تا پسوند های فایل را به آنها محدود کند.

OptionButton : ابزای برای انتخاب یک (و فقط یک) گزینه از میان گزینه هایی ثابت. به نکات زیر توجه داشته باشید :

1. سعی کنید از OptionButton ها به صورت آرایه ای استفاده کنید.
2. همیشه خودتان قبل از کاربر یکی از گزینه ها را انتخاب کنید.

OptionButton.Value : اگر true باشد به معنی انتخاب شده و اگر false باشد به معنی انتخاب نشده است. برای انتخاب کردن هم میتوان به value مقدار true داد.

```
Private Sub Form_Load()
  …
  optPreview(0).Value = True
End Sub
```

آزمایشگاه : یک optionbutton دیگر در frame اضافه کنید و دوتا بیرون از frame برنامه را اجرا کنید و روی آنها کلیک کنید. Optionbutton های داخل یک frame از یک گروهند و در یک لحظه فقط یکی از آنها روشن است. (هرچه زودتر هرچه از این آزمایش هسته ای به جا مانده نابود کنید!)
CheckBox : ابزاری برای گرفتن جواب های بله / خیر از کاربر. که اگر value آن 1 باشد، جواب مثبت است و اگر 0 باشد جواب منفی است.

```
Private Sub Form_Load()
  …
  chkPreview.Value = 1
End Sub
Private Sub filImage_Click()
  If chkPreview.Value = 1 And optPreview(0).Value Then
    Dim pth As String
    pth = filImage.Path
    If Right(pth, 1) <> "\" Then pth = pth + "\"
    imgPreview.Picture = LoadPicture(pth + filImage.FileName)
  End If
End Sub
```

FilImage_click زمانی اجرا میشود که روی یکی از آیتم های آن کلیک شود. و اگر checkbox تیک خورده بود و optPreview هم بر روی کلیک بود تصویر را در image نشان میدهد. اما چگونه؟ دستور loadpicture این دستور اسم و

آدرس یک فایل گرافیکی را میگیرد و تصویر آن را برمیگرداند. بنظر می رسد که می توان از

```
imgPreview.Picture = LoadPicture(filImage.Path + filImage.FileName)
```

استفاده کرد. اما مشکل اینجاست که Path زمانی که آدرس یک درایو باشد به صورت "c:\" یا "d:\" است و اگر آدرس یک دایرکتوری باشد به صورت "c:\windows" یا "d:\program files" است. همانطور که می بینید یکی "\" دارد و دیگری ندارد و اگر یکی از آدرس دایرکتوری ها به نام فایل الحاق شود نتیجه شبیه "c:\windowstest.bmp" و این فایلی نیست که ما میخواستیم.

برای این منظور باید اگر کاراکتر سمت راست "\" نیست این کاراکتر را اضافه کنیم.

دستور right یک رشته و یک عدد n میگیرد و n کاراکتر سمت راست رشته را برمیگرداند.(دستور left وmid را هم در help ببینید.)

از آنجا که باید مانند همین کد(دستورات داخل شرط) را برای dblClick هم بنویسیم از همین حالا یک زیر برنامه برای این کار می نویسیم تا هر دو رویداد بتوانند آن را فراخوانی کنند.

محل تعریف تابع و زیربرنامه نیز مانند متغیر های سراسری در Declaration (بالای همه کد ها) است.

```
Private Sub Preview()
   Dim pth As String
   pth = filImage.Path
   If Right(pth, 1) <> "\" Then pth = pth + "\"
   imgPreview.Picture = LoadPicture(pth + filImage.FileName)
End Sub
Private Sub filImage_Click()
   If chkPreview.Value = 1 And optPreview(0).Value Then Preview
End Sub
```

برای حالتی که کاربر میخواهد با دبل کلیک عکس نمایش داده شود در صورتی عکس با دبل کلیک نشان داده میشود که optionbutton مربوط به آن (1) انتخاب شده باشد.:

```
Private Sub filImage_DblClick()
   If chkPreview.Value = 1 And optPreview(1).Value Then Preview
End Sub
```

اگر این برنامه را اجرا کنید متوجه میشوید که پس از نمایش image به اندازه تصویر بزرگ میشود. از برنامه خارج شوید و stretch را برای image برابر true کنید.(یا کد مربوط به آن را در form_load بنویسید.)

```
Private Sub Form_Load()
   …
   imgPreview.Stretch = True
End Sub
```

با این خاصیت ، عکس اندازه image میشود نه image اندازه عکس.

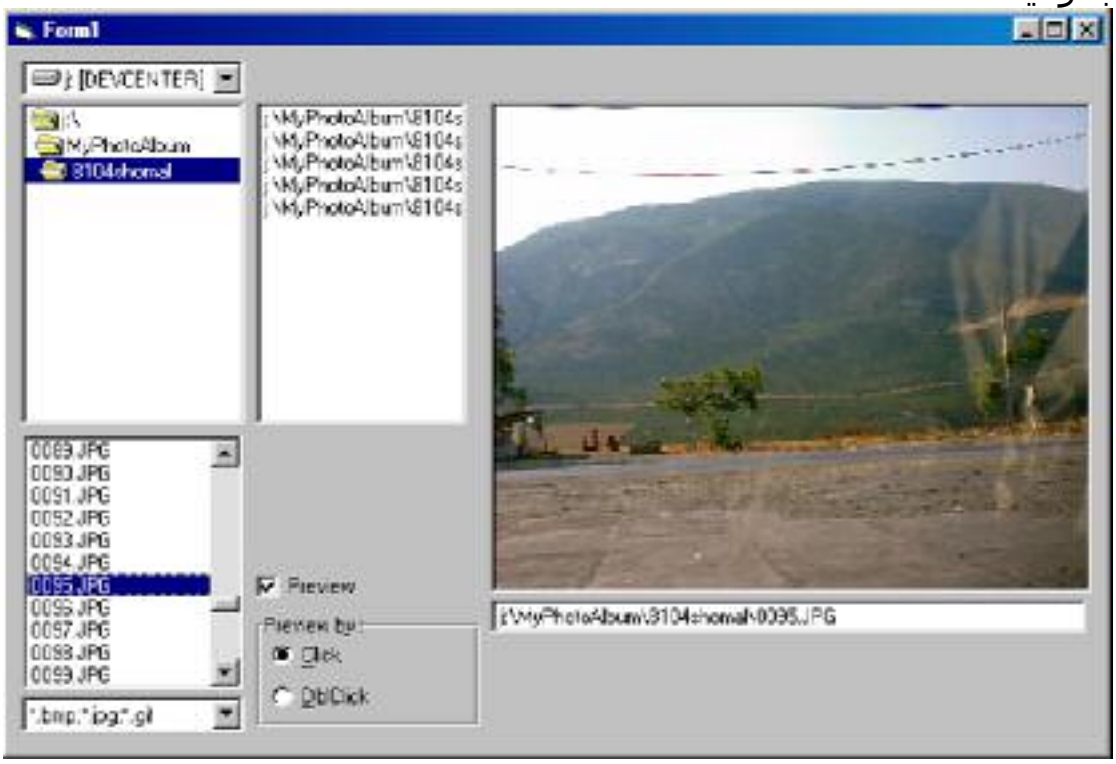
```
Private Sub Preview()
   Dim pth As String
```

```
    pth = filImage.Path
    If Right(pth, 1) <> "\" Then pth = pth + "\"
    pth = pth + filImage.FileName
    txtAddress.Text = pth
    lstHistory.AddItem pth
    imgPreview.Picture = LoadPicture(pth)
End Sub
```

نکته: زمانی که از تابعی که فراخوانی کرده ایم مقدار برگشتی میخواهیم باید در فراخوانی آن از پرانتز استفاده کنیم و زمانی که مقدار برگشتی را در متغیری نمی ریزیم نباید از پرانتز هم استفاده کنیم.

```
Private Sub lstHistory_Click()
    imgPreview.Picture = LoadPicture(lstHistory.List(lstHistory.ListIndex))
End Sub
```

اگر از این خط سر در نمی آورید به توضیحات لیست برگردید و دوباره بخوانید.



تمرین :

1. بعضی از ابزار ها مقداردهی اولیه مناسبی ندارند آنها را مقداردهی اولیه کنید.
2. کاری کنید که عکس ها به ترتیب از عکسی که تازه دیده شده به عکس قدیمی تر در lstHistory قرار بگیرند . الان به چه ترتیبی است؟
3. کاری کنید که بتوان فایل های wmf و ico را نیز دید.
4. وقتی preview چک نخوردهoptPreview ها و frame را غیر فعال کنید.
5. به کاربرامکان حذف از lstHistory را بدهید.
6. وقتی عکس را از lstHistory نشان میدهیم آدرس آن در textbox نشان داده نمی شود. آن را نشان دهید.
7. وقتی preview چک نخورده lstHistory نیز عکس را نمایش ندهد.

کارگاه :

1. کاری کنید که فایل های مخفی و سیستمی نیز نشان داده شوند.
2. کدی در یک dropdown combo بنویسید که پس از زدن دکمه enter متن combo به لیست آن اضافه شود.
3. به کارگاه 2 قابلیت حدس زدن آنچه کاربر میخواهد تایپ کند را بدهید (با توجه به آنچه قبلا تایپ کرده)

راهنمایی :

0. با زدن دکمه F8 میتوانید برنامه را خط به خط اجرا کنید و در این حالت اگر با موس روی متغیر ها و خواص چند لحظه مکس کنید مقدار آن متغیر یا خاصه را به شما میگوید.
1. مخفی (hidden) و سیستمی (system)
2. وقتی enter(13) فشار(press) داده میشود...
3. در حقیقت باید یک جستجو روی آن انجام دهید و اگر کاراکتر های اول آیتم لیست با متن برابر بود متن را برابر آن کنید و کاراکتر هایی که اضافه کرده اید select کنید. اگر کد را در رویداد change می نویسید بخاطر داشته باشید که خودتان هم دارید متن را تغییر می دهید.

پاسخ کارگاه :

1.

```
filPreview.Hidden = true
filPreview.System = true
```

2.

```
Private Sub Combo1_KeyPress(KeyAscii As Integer)
  If KeyAscii = 13 Then Combo1.AddItem Combo1.Text
End Sub
```

3.

```
Private Sub Combo1_Change()
Dim i As Integer, selstrt As Integer
For i = 0 To Combo1.ListCount - 1
  If Left(Combo1.List(i), Len(Combo1.Text)) = Combo1.Text Then
    Combo1.SelStart = Len(Combo1.Text)
    selstrt = Combo1.SelStart
    Combo1.Text = Combo1.List(i)
    Combo1.SelStart = selstrt
    Combo1.SelLength = Len(Combo1.Text) - Combo1.SelStart
  End If
Next
End Sub
```

پروژه برنامه نویسی :

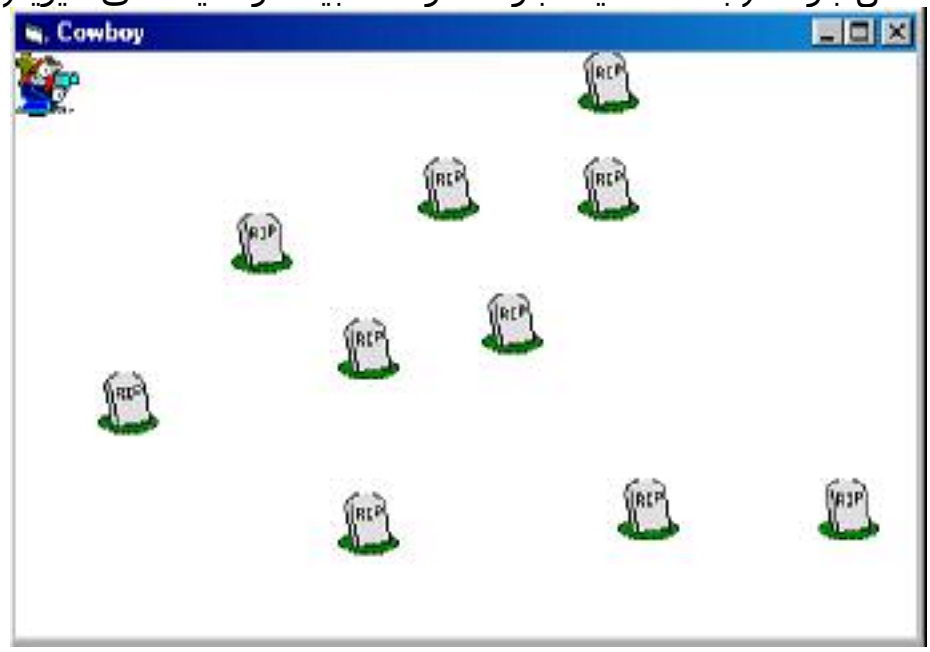
بازی حافظه : این بازی دارای یک صفحه با چندین مربع است.وقتی روی یکی از مربع ها که کلیک کنید یک عکس نشان میدهد و وقتی روی دومین مربع کلیک میکنید آن هم عکسی نشان میدهد. اگر این دو عکس

یکی بودند این دو مربع حذف می شوند و اگر یکی نبودند به همان شکلی میشوند که از اول بودند. بخاطر داشته باشید که هر عکس در محل ثابت خودش می آید یعنی اگر یک باز در گوشه سمت چپ بالا عکس شیر باشد تا آخر بازی آن نقطه عکس شیر است.

فصل سوم : موس و کی بورد و مختصات

3-1: کابوی

فصل قبل تا حدودی با رویدادهای موس و کی بورد آشنا شدیم . در این فصل باز هم به لطف یک بازی کارهای بیشتری یاد می گیریم.



من صفحه دوبعدی سفید را برای این برنامه ترجیح میدهم نظر شما را نمیدانم!

فرمی که می بینید appearance اش برابر 0-flat و نام آن frmMain است.

شی دیگری که بجز فرم لازم داریم یکPictureBox است که من appearance آن را نیز مانند فرم به 0-flat ست کردم. PictureBox تقریبا یک فرم کامل است! و اکثر خواص یک فرم را دارد. اگر در یک برنامه میخواهید یک عکس را فقط نشان دهید ترجیحا از imagebox استفاده کنید چون ابزار سبک تری است. PictureBox خاصیت Stertch ندارد و به طور پیش فرض اگر به خاصیت Picture آن مقدار بدهید اندازه اش تغییر نمیکند. اگر autosize را true کنید هم اندازهٔ تصویر می شود.

شرح برنامه : این برنامه یک بازی دو نفره است به این صورت که یک نفر با کی بورد کابوی را فراری میدهد و دیگری با موس آن را میزند. (نبرد نابرابری است!)

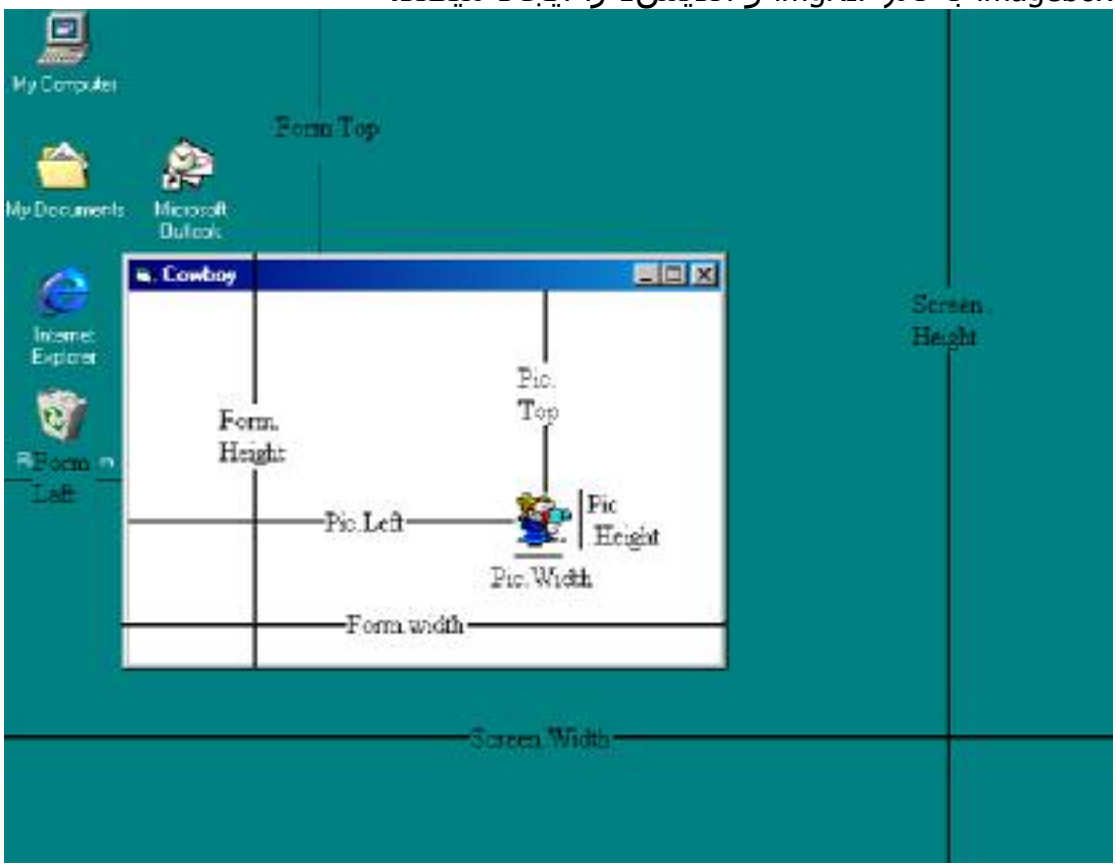
اول : کشتن کابوی بی دفاع

میخواهیم وقتی کابوی تیر میخورد (موس کلیک) تصویر یک سنگ قبر در آن نقطه به جا بماند و بازی ادامه پیدا کند. یعنی اگر 10 بار کابوی ما تیر خورد 10 سنگ قبر درصفحه نشان داده شود. برای نشان دادن سنگ قبرها از imagebox استفاده میکنیم اما این همه imagebox از کجا بیاوریم؟ جواب این است که با هر بار تیر خوردن کابوی یک imagebox جدید درست می کنیم. پس یک imagebox با نامimgRIP روی فرم قرار دهید ، اندیس آن

را 0 کنید و visible آن را false و یک تصویر برای آن لود کنید.(برای اینکه در زمان اجرا نشان داده نشود).
اما چگونه یک ImageBox در زمان اجرا به برنامه اضافه کنیم؟
راه راحت برای این کار استفاده از آرایه کنترل ها و دستور load است.

```
Private Sub picCowboy_Click()
   Load imgRIP(1)
   imgRIP(1).Left = 0
   imgRIP(1).Top = 0
   imgRIP(1).Visible = True
End Sub
```

خط load imgRIP(1) دقیقا همان کاری را میکند که ما میخواهیم. یعنی imagebox با نام imgRIP و اندیس1 را ایجاد میکند.



ImgRIP(1).left فاصله این شی از سمت چپ فرم است . در حقیقت left فاصله شی از سمت چپ با شیی است که آن را در بر دارد. مثلا در مورد فرم ، فرم داخل شی screen است و left فرم فاصله فرم از سمت چپ صفحه است.
واحد مختصات پیش فرض ویژوال بیسیک pixel[6] نیست. بلکه twip (برابر با 1/1440 اینچ) است.
در خط دوم و سوم تصویر را به نقطه (0،0) منتقل کرده ایم و در خط چهارم آن را نشان داده ایم.

[6] به هر نقطه نوراي صفحه يك pixel مي گويند.

مسلما این کد فقط یک تصویر ایجاد میکند. برای ایجاد کردن تصاویر بیشتر باید اندیس های بعدی را لود کنیم. برای اینکه بفهمیم اندیس بعدی باید چند باشد نیازی به متغیر تعریف کردن نیست به کد زیر توجه کنید:

```
Private Sub picCowboy_Click()
   Load imgRIP(imgRIP.Count)
   imgRIP(imgRIP.Count - 1).Left = picCowboy.Left
   imgRIP(imgRIP.Count - 1).Top = picCowboy.Top
   imgRIP(imgRIP.Count - 1).Visible = True
   picCowboy.Move 0, 0
End Sub
```

ImgRIP.count تعداد imgRIP هایی که داریم را برمیگرداند. چون از اندیس صفر شروع کرده ایم پس count یکی بیشتر از آخرین اندیس است .
(وقتی فقط اندیس صفر داریم برابر یک و ...)
اما از خط دوم با count-1 کار کرده ایم چون تعداد imgRIP ها در خط اول یکی اضافه شد . و در آخر picCowboy را به مختصات (0،0) فرستادیم تا بازی را از آنجا شروع کند.

دوم : حرکت دادن کابوی
برای حرکت دادن کابوی میخواهیم از کی بورد استفاده کنیم . قبلا با رویداد keypress آشنا شدیم . دو رویداد دیگر کی بورد عبارتند از : keydown و keyup . keydown به محض پایین رفتن یکی از کلید های کی بورد اجرا میشود و keyup به محض بالا آمدن آن کلید. و keypress پس از هردو آنها فراخوانی میشود. تفاوت عمده keydown و keyup با keypress در مقادیری است که به عنوان آرگمان دارند. درkeypress این مقدار keyascii است که شماره اسکی کلید است. شماره اسکی در حقیقت کد کاراکتر است. اما همه کلیدهای کی بورد کاراکتری نیستند برای مثال کلید های جهتی.
برای کنترل کردن رویدادهایی که برای این کلیدها اتفاق می افتد بسته به زمان مورد نظر برای اجرا می توان ازkeydown یاkeyup استفاده کرد.
توجه : رویداد های کی بورد در یک لحظه فقط به یک کنترل ارسال میشوند، کنترلی که فوکوس focus دارد، در اینجا فوکوس همیشه در اختیار picturebox است ولی اگر یک دکمه یا textbox داشتیم این دو هم می توانستند فوکوس را به خود اختصاص دهند. هنگام کار با رویداد های کی بورد همیشه به این نکته دقت کنید.
KeyCode: کد کلید، ویژوال بیسیک برای کد اکثر کلید ها مقادیر ثابت(constant) هایی دارد که با vbkey شروع میشوند، مثلا vbKeyLeft برای کلید جهتی چپ.
Shift : نشان دهنده وضعیت کلیدهای shift,alt,ctrl است.

dec	bin
1	001
2	010
3	011
4	100
5	101
6	110
7	111

این مورد نیاز به توضیح بیشتری دارد. اگر هیچ کدام فشارداده نشده باشند مقدار صفر است برای shift این مقدار برابر 1 برای ctrl برابر 2 و برای alt برابر با 4 است. تا اینجا مشکلی نیست اما برای حالت های ترکیبی مقدارها با هم جمع میشوند یعنی اگر هر 3 کلید فشارداده شده باشند این مقدار برابر با 7 میشود. ولی این عدد ها از کجا آمده؟
به باینری (عدد در مبنای 2) این اعداد دقت کنید:
هرکدام که رقم سمت راست باینری آن 1 است ترکیبی دارای shift است، دومی ctrl و سومی alt است .
با استفاده از AND باینری میتوان این را تشخیص داد(اگر از باینری و AND باینری چیزی نمیدانید فقط کد زیر را جایی نگه دارید.)

```
Dim msg As String
If (Shift And vbShiftMask) = vbShiftMask Then msg = msg + " shift"
If (Shift And vbCtrlMask) = vbCtrlMask Then msg = msg + " cntrl"
If (Shift And vbAltMask) = vbAltMask Then msg = msg + " alt"
```

حالا برای حرکت دادن کابوی کافی است که left یا top آن را تغییر دهیم. مقدار حرکت بستگی به میل شما دارد من آن را 100 در نظر گرفتم که البته در مقدار shift+1 ضرب میشود. یعنی اگر هر سه کلیدalt,ctrl,shift را نگه دارید طول پرش های کابوی 800 (متر؟) میشود.

```
Private Sub picCowboy_KeyDown(KeyCode As Integer, Shift As Integer)
   If KeyCode = vbKeyLeft Then picCowboy.Left = picCowboy.Left - 100 * (1 + Shift)
   If KeyCode = vbKeyDown Then picCowboy.Top = picCowboy.Top + 100 * (1 + Shift)
   If KeyCode = vbKeyRight Then picCowboy.Left = picCowboy.Left + 100 * (1 + Shift)
   If KeyCode = vbKeyUp Then picCowboy.Top = picCowboy.Top - 100 * (1 + Shift)
End Sub
```

سوم : تغییر دادن شکل موس
برای این کار فقط خاصیت MousePointer را تغییر دهید. من برای این بازی از 2-cross استفاده کردم. شما میتوانید 99-custom استفاده کنید و هر آیکونی که می خواهید را به MouseIcon بدهید تا نشان داده شود.(کار بیشتر به عهده خودتان)

چهارم : محدود کردن کابوی به محیط فرم
تجربه های قبلی آموزش زبانهای ویژوال که داشتم دیدم اکثرا در این بخش مشکل دارند. اما در حقیقت این بخش بسیار ساده است فقط نیاز به کمی تفکر و دقت دارد. اگر تصور ابعاد هنوز برایتان مشکل است عکس مشخصات صفحه را نگاه کنید.
میخواهیم اگر کابوی از سمت راست صفحه خارج شد او را به دیواره سمت راست بچسبانیم.
به آن شکل دقت کنید و شرط مناسب را پیدا کنید

```
If picCowboy.Left + picCowboy.Width > frmMain.Width Then _
    picCowboy.Left = frmMain.Width - picCowboy.Width
```

من فکرمیکنم هر توضیحی درباره این بخش باعث گیج شدن میشود . پس لطف میکنم و توضیح بیشتری نمی دهم و در عوض نوشتن 3 شرط دیگر (برای سمت چپ، بالا و پایین) را بعهده خودتان میگذارم. فقط این توضیح که بدلیل حاشیه های فرم (مخصوصا نوار عنوان در بالا) باید از یک مقدار تقریبی برای دقیق کردن کارتان استفاده کنید یا اینکه borderstyle را برابر 0-none کنید تا تمام حاشیه فرم محو شود.

پنجم : شروع مجدد بازی

تنها کاری که میکنیم حذف کردن تمام سنگ قبر ها است . برای این کار آنها را unload میکنیم. اما بخاطر داشته باشید که فقط اشیایی را میتوان unload کرد که با load درست شده باشند.(در این مورد فرم ها استثنااند unload frmMain را آزمایش کنید.)

```
If KeyCode = vbKeyF2 Then
   Dim i As Integer
   For i = 1 To imgRIP.Count - 1
      Unload imgRIP(i)
   Next
End If
```

بازدن کلید F2 تمام سنگ قبر ها محو میشوند.

تمرین :

1. تعداد گلوله های قاتل کابوی را محدود کنید تا این نبرد کمی عادلانه شود.
2. کاری کنید که وقتی گلوله به کابوی نمیخورد جای آن روی صفحه بماند .
3. آمار تعداد گلوله های به هدف خورده و نخورده را نشان دهید و وقتی گلوله ها تمام شد بر حسب این آمار برنده را تعیین کنید.

2-3: رسم خط

برای آنکه با رویداد های موس آشنا شویم برنامه ای برای رسم کردن خط مینویسیم که میتوان آن را به یک برنامه نقاشی تبدیل کرد.

یادآوری میکنم که برنامه نویسی کار بسیار ساده ای است و اگر کسی به شما گفته برنامه نویسان افرادی با ضریب هوشی بالا و پشتکار بسیار زیاد ودرآمد های خیلی زیاد و علاقه مند به قهوه یا چای هستند فقط آخری را راست گفته!

مراحل کار :

1. رسم کردن یک خط بین دو نقطه مشخص : دستور line که از بیسیک باقی مانده است با گرفتن مختصات نقطه شروع و پایان خط آن را رسم میکند. مانند همیشه ابتدا یک برنامه آزمایشی می نویسیم . پس یک picturebox و یک دکمه روی صفحه قرار دهید کد زیر را بنویسید :

```
Private Sub Command1_Click()
   Picture1.Line (0, 0)-(Picture1.Width, Picture1.Height)
End Sub
```

در این دستور (که به نظر من کمی از مد افتاده وبد قیافه است.) پرانتز اول مختصات نقطه شروع و دومی مختصات نقطه پایان است و در هر پرانتز عدد اولی x و عدد دومی y است.
برای تغییر دادن رنگ خط میتوانید از خاصیت forecolor شی picture استفاده کنید.

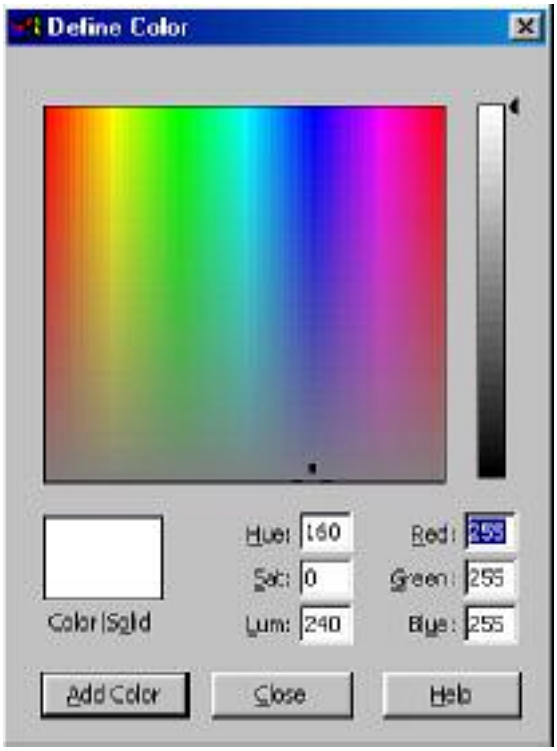
```
Private Sub Command1_Click()
   Picture1.ForeColor = RGB(0, 100, 200)
   Picture1.Line (0, 0)-(Picture1.Width, Picture1.Height)
End Sub
```

رنگهای بیسیک جوابگوی محیط ویندوز نبودند و در ویندوز از یک عدد long برای تعیین رنگ استفاده میشود که بین رنگ سفید برابر 16777215 و مشکی برابر صفر است. اما کار کردن با این اعداد هشت رقمی و بخاطر سپردن آنها کار ساده ای نیست . به همین دلیل ما بیشتر از تابع RGB[7] استفاده میکنیم این تابع سه عدد بین صفر تا 255 به عنوان ورودی می گیرد که اولی قرمز(Red) دومی سبز(Green) و آخری آبی(Blue) است. با تغییر دادن میزان این رنگ ها و ترکیب آنها میتوان 16777215 رنگ بدست آورد.

رنگ	RGB
قرمز	255,0,0
سبز	0,255,0
آبی	0,0,255
بنفش	255,0,255
زرد	255,255,0
نارنجی	255,100,0
سفید	255,255,255
مشکی	0,0,0
خاکستری	192,192,192

خودتان میتوانید رنگ دلخواهتان را با ترکیب این سه رنگ درست کنید و اگر این کار برای شما هم مانند من سخت است میتوانید از پنجره زیر که همه جا پیدا میشود استفاده کنید.

[7] Red Green Blue



آیا تابع RGB معجزه میکند؟ در مورد آن عدد long عجیب غریب و ایجاد کردن آن از روی RGB عزیز! هیچ معجزه ای در کار نیست.
فرمول این تبدیل در زیر آمده است:

```
r + g * 256 + b * 256 ^ 2
```

"^"نماد توان است.
آن خط قشنگ را که فراموش نکرده اید؟برنامه را اجرا کنید و دکمه را بزنید. وقتی خط رسم شد یک پنجره دیگر را روی آن حرکت دهید. پاک می شود!
برای اینکه خط پاک نشود دو راه هست اول اینکه هر وقت پاک شد آن را دوباره رسم کنیم و دوم اینکه رسم دوباره آن را به خود ویژوال بیسیک بسپاریم.
دوم: فقط کافی است که خاصیت AutoRedraw را برای picture به true ست کنید.
اول: وقتی که بخشی از برنامه پاک میشود یا به هر دلیلی نیاز به رسم مجدد است رویداد paint برای آن شی فراخوانی میشود کد زیر را آزمایش کنید:

```
Private Sub Picture1_Paint()
   Picture1.ForeColor = RGB(0, 100, 200)
   Picture1.Line (0, 0)-(Picture1.Width, Picture1.Height)
End
```

رنگ خط را به یک روش دیگر هم میتوان تعیین کرد :

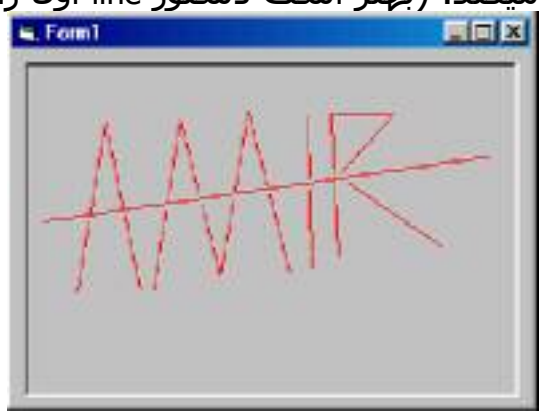
```
   Picture1.Line (0, 0)-(Picture1.Width, Picture1.Height), RGB(0, 100, 200)
```

2. گرفتن مختصات خط از کاربر

رویدادMouseDown این رویداد وقتی فراخوانی میشود که یکی از دکمه های موس فشار داده شود (MouseUp هم برعکس آن است) برای اینکه نقطه ابتدای خط را از کاربر بگیریم از این رویداد استفاده میکنیم. متغیر های x و y محل فعلی موس و button دکمه ای را که فشار داده شده است را نشان میدهند. Shift هم مانند shift در keydown است.
رویداد MouseMove این رویداد وقتی اجرا میشود که موس روی فرم حرکت کند (یا فرم زیر موس) مقدار پارامتر های آن مانند mousedown است.
واین هم کد کامل برنامه با AutoRedraw = true

```
Dim X1 As Integer, Y1 As Integer, X2 As Integer, Y2 As Integer
Dim oldX As Integer, oldY As Integer

Private Sub Form_Load()
X1 = 0: Y1 = 0: oldX = 0: oldY = 0
End Sub

Private Sub Picture1_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)
X1 = X
Y1 = Y
End Sub

Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
If Button = vbLeftButton Then
   X2 = X
   Y2 = Y
   Picture1.Line (X1, Y1)-(oldX, oldY), Picture1.BackColor
   Picture1.Line (X1, Y1)-(X2, Y2), RGB(255, 0, 0)
   oldX = X2: oldY = Y2
End If
End Sub
```

اولین دستور line خط قبلی را پاک میکند و دومین lineخط جدید را رسم میکند. (بهتر است دستور line اول را پاک کنید و نتیجه را ببینید)



تمرین :

1. به این برنامه امکان رسم مستطیل (آخر دستور line یک B, اضافه کنید) و دایره(با دستور circle) را اضافه کنید.
2. امکان تعیین رنگ را به کاربر بدهید.
3. قابلیت پاک کردن تمام تصویر را به کاربر بدهید.

کارگاه :

1. امکان save و load کردن تصویر را به برنامه اضافه کنید.
2. تابعی برای تبدیل کردن رنگ به سه عدد RGB بنویسید (برعکس کار تابع RGB)

راهنمایی :

1. بهتر است picture1.image,LoadPicture,SavePicture را ببینید.
2. کاری شبیه به تبدیل کردن به مبنای 256 است.

پاسخ کارگاه :

1.
```
Private Sub cmdLoad_Click()
   Picture1.Picture = LoadPicture("c:\image.bmp")
End Sub

Private Sub cmdSave_Click()
   SavePicture Picture1.Image, "c:\image.bmp"
End Sub
```

2.
```
SubPrivate Type cRGB
   r As Byte
   g As Byte
   b As Byte
End Type
Private Function RGBDecoder(ByVal c As Long) As cRGB
   If c < 0 Then Exit Function
   RGBDecoder.r = c Mod 256
   c = c \ 256
   RGBDecoder.g = c Mod 256
   RGBDecoder.b = c \ 256
End Function
```

فصل چهارم : ابزار Timer

ابزار تایمر یکی از جذاب ترین ابزارهای ویندوز است. با آن به راحتی میتوانید تصاویر متحرک یا ساعت زنگدار درست کنید . در اینجا هم یک برنامه با تایمر می نویسیم. توجه کنید که این ابزار زمان اجرا دیده نمی شود.
و بخاطر داشته باشید که منابع تایمر ویندوز محدود است . بیش از 3 یا 4 تایمر در برنامه خودتان استفاده نکنید تا از به خطر افتادن منابع ویندوز و برنامه های دیگر جلوگیری کنید. در کل استفاده از تایمر زیاد توصیه نمیشود.

تصاویر متحرک

تصاویر متحرک تصاویری هستند که در یک مدت زمان مشخص به تصویر بعدی تبدیل میشوند. در اینجا دو چیز لازم داریم 1. تصویر بعدی 2. فهمیدن اینکه چه وقت باید تصویر عوض شود.
برای فهمیدن تصویر بعدی نباید مشکلی داشته باشید. اول چند تصویر (مثلا راه رفتن آدم) تهیه کنید. سپس به تعداد تصاویر imagebox(imgPicLoad) روی صفحه قرار دهید و از آنها آرایه درست کنید.یک imagebox(imgShow) جدا از آرایه و یک دکمه هم روی فرم قرار دهید. و کد زیر را در دکمه بنویسید.

```
Dim i As Integer
i = i + 1
imgShow.Picture = imgLoadPic(i).Picture
```

برنامه را اجرا کنید تا نتیجه را ببینید. اما فقط یکی از تصویر ها می آید آن هم تصویر دوم! اینکه چرا تصویر دوم می آید به عهده شما. اما اینکه چرا تغییر نمی کند به دلیل اینکه ما متغیر I را با dim تعریف کرده ایم و اگر بخاطر داشته باشید عمر این نوع متغیر با عمر تابع است پس وقتی برای بار دوم تابع را اجرا کنیم متغیر دوباره تعریف میشود و مقدار صفر میگیرد . برنامه را کمی تصحیح میکنیم :

```
Static i As Integer
imgShow.Picture = imgLoadPic(i).Picture
i = i + 1
```

مشکلات قبلی حل شده و مشکل جدیدی ایجاد شده. با هر بار کلیک کردن تصویر بعدی نمایش داده میشود ولی وقتی همه تصاویر یک بار نشان داده میشوند برنامه دچار RunTimeError میشود. بدلیل اینکه I از بزرگترین اندیس ما یکی بیشتر شده و چنین اندیسی وجود ندارد میتوانیم برای برطرف کردن این مشکل از یک شرط استفاده کنیم اما بهتر است اگر مشکلی بدون شرط حل میشود از شرط استفاده نکنیم (کدی که درد نمیکنه if نمی بندن)

```
Static i As Integer
imgShow.Picture = imgLoadPic(i).Picture
i = (i + 1) Mod imgLoadPic.Count
```

دستور mod باقیمانده تقسیم پارامتر سمت چپ به پارامتر سمت راست را برمیگرداند و مسلما باقیمانده تقسیم از مقسوم علیه کمتر است . پس حاصل همیشه عددی کوچکتر از imgLoadPic.Count است.
یک ابزار تایمر به برنامه اضافه کنید. تایمر فقط یک رویداد دارد که نام آن هم تایمر است. اما کار آن چیست ؟ تایمر یک خاصیت به نام interval دارد که عددی بر حسب میلی ثانیه است. رویداد تایمر هر interval هزارم ثانیه یک بار فراخوانی میشود. البته این در صورتی است که کاری که در فراخوانی قبلی شروع کرده بود تمام شده باشد و منابع سیستم هم آزاد باشد در غیر این صورت تایمر قربانی سایر برنامه ها میشود و فراخوانی آن به تاخیر می افتد.
Interval را با 100 مقدار دهی کنید و کدی که در دکمه نوشته بودیم در تایمر کپی کنید. و برنامه را اجرا کنید. وقتی برای تایمر کد می نویسید بخاطر داشته باشید که تایمر مانند دکمه ای است که مدام کلیک میشود.
یک دکمه دیگر به تایمر اضافه کنید و caption آن را "&start" کنید. و enabled را برای تایمر false کنید.
میخواهیم وقتی روی این دکمه کلیک میشود عنوان آن به "&stop" تغییر کند و تایمر فعال شود و اگر بار دیگر روی دکمه کلیک شد به وضع اولیه بازگردد.

```
Private Sub Command2_Click()
   Command2.Caption = IIf(Command2.Caption = "&Start", "&Stop", "&Start")
   Timer1.Enabled = Not Timer1.Enabled
End Sub
```

دستور iif یک if then else کوچک است . پارامتر اول شرط است ، اگر برقرار باشد پارامتر دوم برگردانده میشود و اگر بر قرار نباشد پارامتر سوم برگردانده میشود یعنی اگر عنوان دکمه ما "start" باشد "stop" میشود و اگر "start" نباشد ، "start" میشود.

var	Not
true	*false*
false	*true*

خط بعدی مقدار enabled را برابر not enabled میکند . به عبارت دیگر آن را قرینه میکند اگر true بود false میشود و اگر false بود true میشود .

تمرین :

1. برنامه را طوری تغییر دهید که تصویر در صفحه هم حرکت کند. و وقتی به یکی از دیوارهای فرم برخورد کرد با همان زاویه برگردد (مانند وقتی نور به آینه تخت برخورد میکند)
2. یک label روی فرم قرار دهید که نام شما کاراکتر به کاراکتر در آن نوشته شود، پاک شود و دوباره نوشته شود.

کارگاه :

ساعت : برنامه ساعتی بنویسید که در یک ساعت خاص زنگ بزند. قابلیت روشن خاموش کردن زنگ هم داشته باشد.
راهنمایی :

دستور time دستور Format ویک TextBox و...
پاسخ :

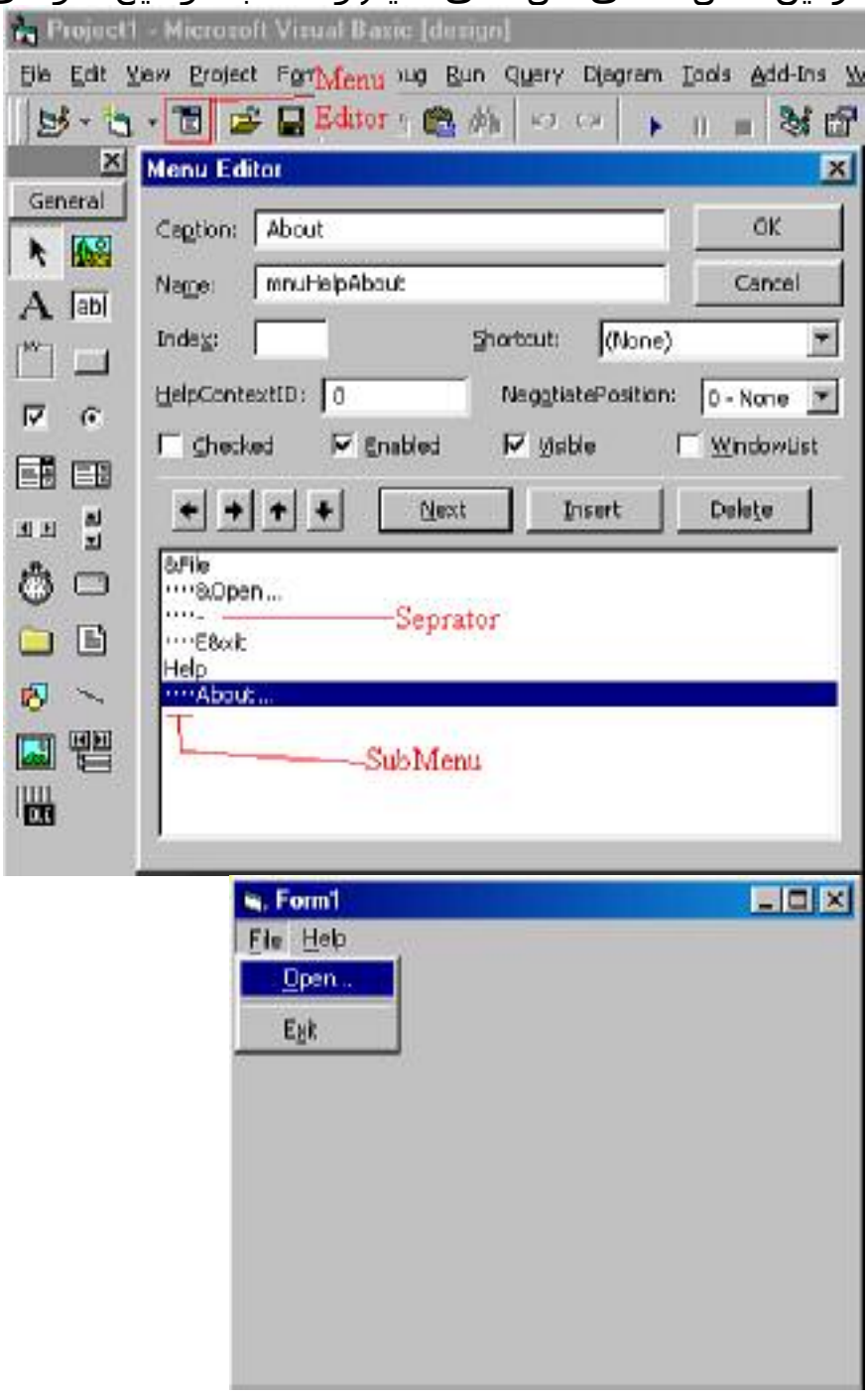
```
Private Sub Form_Load()
   Timer1_Timer
End Sub

Private Sub Timer1_Timer()
Label1.Caption = Format(Time, "hh:mm:ss ampm")
If (CDate(Label1.Caption) >= CDate(Text1.Text)) And (CBool(Check1.Value)) Then
   Beep
   Beep
End If
End Sub
```

در form_load، Timer1.Timer را صدا کرده ایم تا Label1 از ابتدا مقدار داشته باشد.

پروژه برنامه نویسی :
برنامه کابوی را یک نفره کنید. یعنی کابوی خودش حرکت کند و کاربر آن را بزند. (برای سخت تر شدن بازی میتوان کابوی را هر لحظه به یک نقطه تصادفی فرستاد)

فصل پنجم : منوها

یکی از مهمترین ابزارهای رابط کاربر منوها هستند. این ابزار بقدری مورد استفاده است که تصور یک برنامه کامل بدون منو تقریبا غیر ممکن است. در این فصل مثالی حل نمی کنیم و فقط به توضیح منو می پردازیم.



کار با منو ادیتور بسیار ساده است . منو ادیتور را باز کنید . caption و name را پر کنید. در حقیقت کار تمام شده! میتوانید با OK پنجره را ببندید و منوی زیبای خودتان را ببینید. اما کار ما با منو ادیتور تمام نشده. پس دکمه next را بزنید تا منوی بعدی را طراحی کنید. مانند قبلی caption و name را

تکمیل کنید. اگر ok کنید می بینید که هر دو منو در کنار هم هستند . حالا به منو ادیتور باز گردید و منوی دومی را select کنید و سپس دکمه <- را بزنید تا قبل از نوشته آن```` اضافه شود. حال اگر به فرم برگردید می بینید که منوی دومی داخل منوی اولی است.
تنها مساله ای که در طراحی منو باقی میماند خط های جدا کننده است. برای اینکه چند گزینه در یک منو را که از نظر مفهومی خیلی ارتباطی با هم ندارند از هم جدا کنیم از این خط ها استفاده میکنیم . برای تعریف یک خط جداکننده کافی است که caption آن منو را (حتما باید زیر منو باشد) برابر با – (منها) کنید. و نام دلخواهی مانند mnuSep1 به آن بدهید. کار تمام شده است.

کلید میانبر :

برای تنظیم کردن کلید میانبر یک زیر منو، در منو ادیتور آن زیر منو را انتخاب کنید و از لیست (combobox) کلیدهای میانبر(shortcut) یکی را برای آن انتخاب کنید. حالا اگر آن کلید میانبر را بزنید مانند کلیک کردن روی این منو است.

Checked :

منو میتواند شبیه یک checkbox عمل کند. برای این منظور باید خاصیت checked آن را true یا false کرد. برای اینکه بطور پیش فزض این خاصیت true باشد میتوان checked را در منو ادیتور true کرد.

منوهای جهنده (منوهای حساس به محتوا) PopupMenu

منوهای جهنده تفاوتی در طراحی با منوی اصلی ندارند فقط معمولا visible نیستند. برای آنکه روی یک شی منوی جهنده بیاید باید از form.popupmenu استفاده کنید. که آرگمانهای آن بترتیب نام منویی که میخواهید بیاید(top level menu) ، setting های خاص مانند اینکه منو از راست باز شود یا چپ یا وسط ، x نقطه ای که منو آنجا باز میشود، y نقطه ای که منو آنجا باز میشود(مختصات پیش فرض مختصات موس است.) و آخرین آرگمان اینکه کدام آیتم از منو باید پررنگ نوشته شود.(معمولا آیتمی که پررنگ (bold) نوشته میشود با دبل کلیک کردن در آن نقطه هم قابل دسترسی است.)

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Button = vbRightButton Then _
        Form1.PopupMenu mnuFile, defaultmenu:=mnuFileOpen
End Sub
```

توضیح غیر ضروری : برای این به این منو ها منوی محتوی میگویند که میتوانند بسته به جایی که باز میشوند متفاوت باشند مثلا اگر برای یک picturebox باز میشود یک منو باشد و اگر برای listbox باز میشود یک منوی دیگر باشد. کافی است برای هر کدام یک منوی جداگانه درست کنید و در mousedown ابزار مربوط به آن صدایش کنید.

نوشتن کد برای منو :

اگر روی منو در فرم کلیک کنید زیر برنامه رویداد click منو باز می شود سایر کارها همانطور است که میداند.

استاندارد ها :

در طراحی منو ها توجه به نکاتی ریز بسیار مهم است.

1. اگر یک منو قرار است پنجره دیگری را باز کند در انتهای نام آن یک "..." قرار میدهیم تا این مساله را به کاربر اطلاع داده باشیم.
2. برای نام منوهای اصلی از یک کلمه استفاده کنید (space ممنوع) چون نامهای دو کلمه ای کاربر را به اشتباه می اندازد.
3. اولین منو از سمت چپ همیشه File است (مگر در بازی ها که میتواند Game باشد) و منوی exit همیشه آخرین آیتم از این منو است و با خط از سایر آیتم ها جدا میشود.
4. منوی help آخرین منو است و about آخرین‌آیتم آن است که اطلاعاتی درباره برنامه نویسان و کپی رایت و ... را به کاربر نشان میدهد.
5. منوهای استاندارد و میانبر های استاندارد را همیشه رعایت کنید. (کاری که همه میکنند درست است . چون کاربر به آن عادت کرده است.)

تمرین :

برای برنامه هایی که تابحال نوشته ایم منو طراحی کنید.

فصل ششم : پنجره ها برای که باز می شوند؟

در این فصل برنامه های قبلی خود دا تغییر میدهیم و پنجره های دیگری به آنها اضافه میکنیم.

1-6 : پنجره پیغام

پنجره پیغام یکی از راه های ساده برای ارتباط با کاربر است. که میتواند برای پرسیدن یک سوال،گرفتن تایید و اطلاع دادن وضعیت به کاربر مورد استفاد ه قرار گیرد. اما همیشه توجه داشته باشید که پنجره پیام بدلیل اینکه کارهای کاربر را متوقف میکند تا کاربر به آن پاسخ دهد فقط در موارد اضطراری و مهم مطلوب است. پیغام های زیاد و طولانی کاربر را ناراحت وحتی عصبی میکنند.

روش استفاده :

سری به برنامه پیدا کردن عدد بزنید . می خواهیم همیشه زود تر از برادرمان که با 3 حدس عدد را پیدا میکند برنده شویم! بالاخره ما برنامه را نوشته ایم اگر در بازی خودمان بازنده شویم آبروریزی میشود!

```
Private Sub Label1_DblClick()
   MsgBox "RndNum is " + CStr(RndNum)
End Sub
```

پارامتر های msgbox :

Prompt : متنی که در جعبه پیغام نمایش داده میشود.

Buttons : دکمه(ها)ی پنجره پیغام که میتواندیکی یا مجموعی از مقادیر زیر باشد :

Constant	Value	توضیح
vbOKOnly	0	فقط دکمه OK را نمایش میدهد.
vbOKCancel	1	دکمه های OK و Cancel را نشان میدهد.
vbAbortRetryIgnore	2	دکمه های abort ، Retry و Ignore را نشان میدهد.
vbYesNoCancel	3	دکمه های Yes، No و Cancel را نشان میدهد.
vbYesNo	4	دکمه های Yes و No را نمایش میدهد.
vbRetryCancel	5	دکمه های Retry و Cancel را نشان میدهد.
vbCritical	16	شکل ضربدر سفید در دایره قرمز(بحرانی) را نشان میدهد.
vbQuestion	32	علامت سوال را نمایش میدهد.
vbExclamation	48	علامت اخطار را نشان میدهد.
vbInformation	64	علامت اطلاع رسانی را نمایش میدهد.

vbDefaultButton1~4	0	تعیین دکمه پیشفرض از دکمه اول تا چهارم
vbApplicationModal	0	کاربر باید پیش از هر کاری با این برنامه به پیغام پاسخ دهد.
vbSystemModal	4096	کاربر باید به این پیغام پاسخ دهد در غیر این صورت نمیتواند با هیچ برنامه ای کار کند.

در مورد استفاده از شکل های کنار پیغام دقت کنید. این شکل ها برای قشنگی برنامه نیستند!هرکدام از آنها دقیقا به همان منظوری طراحی شده اند که گفته شده. مخصوصا vbCritical که فقط برای وضعیت های واقعا بحرانی است و استفاده زیاد از آن حساسیت کاربر را نسبت به آن کم میکند که این تنها محدود به برنامه شما هم نمی شود.
Title : عنوان پنجره پیغام که اگر وارد نشود برابر با نام پروژه است.

پنجره پیغام با کدام دکمه بسته شده؟
برنامه نمایش تصویر را بیاورید. میخواهیم از کاربر برای خروج از آن تایید بگیریم (این کار را فقط بمنظور آموزش انجام میدهیم برنامه ای که کاربر سندی در آن ویرایش نکرده نیازی به پرسش ندارد.)

```
Private Sub Form_Unload(Cancel As Integer)
  If MsgBox("Are you sure to exit?", _
    vbQuestion + vbYesNo) = vbNo Then Cancel = True
End Sub
```

همانطور که اشاره کردیم میتوان بعضی از مشخصه های buttons را با هم جمع کرد در مثال بالا به معنی YesNo با علامت سوال است. می توانستیم آن را با یکی از **vbDefaultButton1~4** ها نیز جمع کنیم.

Constant	Value	Description
vbOK	1	**OK**
vbCancel	2	**Cancel**
vbAbort	3	**Abort**
vbRetry	4	**Retry**
vbIgnore	5	**Ignore**
vbYes	6	**Yes**
vbNo	7	**No**

مقدار برگشتی پیغام :
هر کدام از دکمه ها که زده شود یک مقدار برمیگرداند.
در اینجا چون فقط دو دکمه روی فرم بود ، نیازی به ذخیره کردن مقدار برگشتی در متغییر نبود اما اگر نیاز به این کار پیدا کردید میتوانید آن را در یک متغیر integre یا VbMsgBoxResult ذخیره کنید.

```
Private Sub Form_Unload(Cancel As Integer)
  Dim ret As VbMsgBoxResult
  ret = MsgBox("Are you sure to exit?", vbQuestion + vbYesNo)
  If ret = vbNo Then Cancel = True
End Sub
```

رویداد unload این رویداد زمانی فراخوانی میشود که فرم در حال بسته شدن است و اگر متغیر cancel را true کند بستن فرم لغو میشود. این متغیر بطور پیش فرض false است.
طرح تمرین برای این بخش بعهده خودتان است. هرچه بیشتر بهتر! اینجا به یکی بسنده میکنیم.
تمرین :
در بازی کابوی قبل ازnew game از کاربر تایید بگیرید.

2-6 : پنجره ورودی

این پنجره برای وقتی که می خواهید کمی هم تنبلی کنید مفید است! مثلا میتوانید در بازی کابوی یک نفره (که لابد نوشته اید!) با زدن یکی از منو ها (که لابد طراحی کرده اید) سرعت کابوی را با یکی از این پنجره ها از کاربر بگیرید(که لابد میگیرید!).

```
Dim ret As String
ret = InputBox("enter ur name", "cowboy", "10")
If IsNumeric(ret) Then
   If ret < 2 ^ 16 And ret > 0 Then Timer1.Interval = CInt(ret)
End If
```

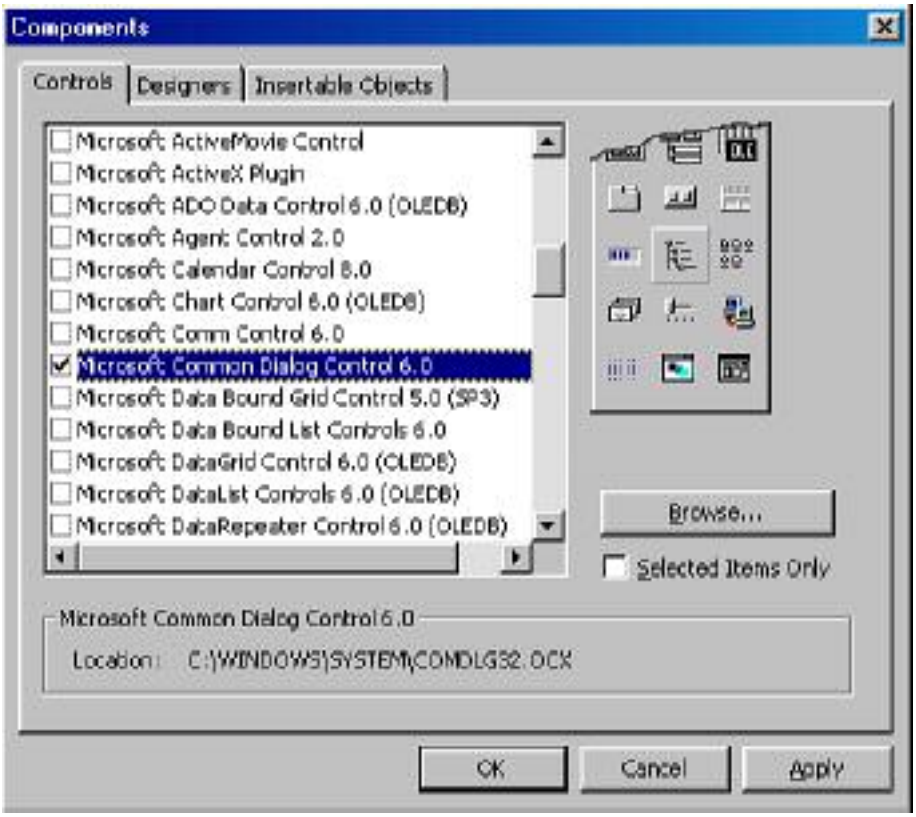
آرگمان های آن شبیه msgbox است فقط آرگمان defualt در msgbox وجود ندارد. این آرگمان مقدار پیشفرضی است که در پنجره نوشته شده است.شما میتوانید آن را برابر با interval فعلی تایمر قرار دهید.شرط اول برای این است که اگر کاربر بجای عدد اسم خودش را وارد کرد برنامه ما دچار خطای زمان اجرا(runtime error) نشود.
به این دلیل گفتم برای تنبلی خوب است که نه ظاهر خوبی دارد نه ما کنترل مناسبی که باید میتوانیم روی آن داشته باشیم ، اما شاید در بعضی موارد بهترین انتخاب باشد.

3-6 : دیالوگ های استاندارد

حتما پنجره هایی که برنامه های مختلف برای باز یا ذخیره کردن فایل ، انتخاب فونت و یا رنگ استفاده میکنند دیده اید. اگر دقت کرده باشید اینها همه مانند هم هستند. دلیلش چیست؟ این دیالوگ ها جزو دیالوگ های استاندارد ویندوز هستند و ما نیز میتوانیم از آنها استفاده کنیم .
روی جعبه ابزار کلید راست را بزنید و ...components را انتخاب کنید(ctrl+T) در لیستی که باز میشود Microsoft Common Dialog Control 6.0 را انتخاب کنید.
این ابزار به جعبه ابزار اضافه می شود ، یکی روی صفحه قرار دهید .
این ابزار مانند تایمر در زمان اجرا نامریی است.

1-3-6 : **open dialog**

یک دکمه و یک label روی صفحه قرار دهید تا این ابزار جدید را آزمایش کنیم.
در دکمه کد زیر را بنویسید :




```
Private Sub Command1_Click()
   CommonDialog1.Filter = "All Picture Files|*.jpg;*.bmp;*.gif|JPG only|*.jpg"
   CommonDialog1.ShowOpen
   Label1.Caption = CommonDialog1.FileName
End Sub
```

Filter : تعیین کننده فایل هایی که باید نمایش داده شوند. قبل از | (pipeline) اول نامی که برای این فرمت فایل نشان داده میشود و بعد از آن فایلهایی که باید نمایش داده شوند که با ; از هم جدا میشوند و باز pipeline و ...

Showopen : پنجره باز کردن فایل را نشان میدهد و برنامه در این خط صبر میکند تا پنجره بسته شود.

FileName : در بر دارنده نام فایل انتخاب شده توسط کاربر.

Flags : میتوانید به دلخواه تغییراتی در شکل و روش کار این ابزار ایجاد کنید. برای مثال اگر

```
CommonDialog1.Flags = cdlOFNHideReadOnly
```

را قبل از showopen بنویسید دکمه readonly حذف میشود.

save نیز دقیقا مانند open است.

6-3-2: **FontDialog**

پنجره انتخاب قلم از دیالگ های پر کاربرد است. در مورد این پنجره یکی از سه فلگ زیر باید ست شود :

1. cdlCFBoth برای نمایش فونت های چاپگر و صفحه نمایش.
2. cdlCFPrinterFonts برای نمایش فونت های چاپگر.
3. cdlCFScreenFonts برای نمایش فونت های صفحه نمایش.

کد نمونه در زیر آمده است. این کد مشخصات فونت را برای label تنظیم میکند.

```
Private Sub Command2_Click()
   CommonDialog1.Flags = cdlCFScreenFonts
   CommonDialog1.ShowFont
   If Trim(CommonDialog1.FontName) = "" Then Exit Sub
   Label1.FontName = CommonDialog1.FontName
   Label1.FontSize = CommonDialog1.FontSize
   Label1.FontBold = CommonDialog1.FontBold
   Label1.FontItalic = CommonDialog1.FontItalic
End Sub
```

شرط استفاده شده برای این است که اگر کاربر cancel کرد برنامه دچار خطا نشود (در صورت cancel کردن مقدار fontname برابر با " " میشود.) و دستور trim فاصله های خالی دوطرف رشته را حذف میکند.

3-3-6 : **color dialog**

انتخاب رنگ ساده تر از همه است . توضیح بیشتر لازم نیست! فقط یک نمونه.

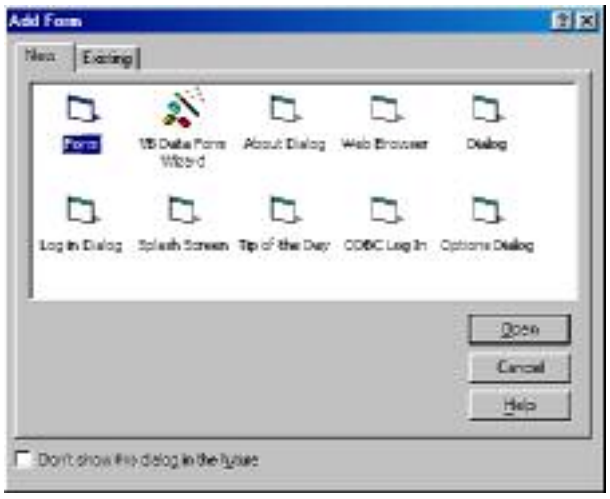
```
Private Sub Command3_Click()
   CommonDialog1.ShowColor
   Label1.ForeColor = CommonDialog1.Color
End Sub
```

تمرین :

به کاربر برنامه cowboy این امکان را بدهید که شکل کابوی ، شکل کرسر موس ، سنگ قبر و رنگ زمینه را خودش انتخاب کند.

4-6 : طراحی فرمهای دیگر

هرچقدر هم که فرم های استاندارد ویندوز خوب و کامل باشند نمیتوانند تمام نیازهای ما را بر آورده کنند و لازم میشود که خودمان آستین بالا بزنیم. پس از همین حالا آستین هایتان را بالا بزنید!



برنامه پیدا کردن عدد را باز کنید ، به منوی project بروید و add form را انتخاب کنید. از پنجره ای که باز میشود گزینه Form را انتخاب کنید .

فرم جدید قرار است فرم تنظیمات برنامه باشد. مثلا در آن محدوده عدد تصادفی یا تعداد دفعات مجاز برای حدس زدن را تنظیم کنیم. نباید در طراحی این فرم مشکلی داشته باشید. فقط به چند سوال شایع پاسخ میدهیم :

1. چگونه فرم دیگری را نمایش دهیم ؟

برای نمایش یک فرم دیگر از متد show استفاده می کنیم.

Form2.Show

اگر این متد بدون پارامتر اجرا شود شما هم میتوانید با فرم جدید کار کنید هم با فرم قدیمی اما اگر با پارامتر vbModal صدا کنید فقط فرمی که تازه باز شده کار میکند و فرمی که آن را صدا کرده صبر میکند تا کار آن فرم تمام شود.

2. چگونه یک فرم را ببندیم؟

اگر میخواهید یک فرم بطور موقت مخفی شود از Hide استفاده کنید. این دستور فرم را محو میکند اما باعث نمیشود که عمر متغیر های عمومی آن (مانند متن یک جعبه متن) به پایان برسد.

Form2.Hide

اگر می خواهید یک فرم کاملا بسته شود و متغیر های آن از بین بروند از unload استفاده کنید

Unload Form2

یا

Unload Me

شی me نام دیگر فرمی است که شما الان در آن هستید.

دکمه X بالای فرم ، فرم را unload میکند.

3. چگونه به مقداری را از یک فرم به فرم دیگر منتقل کنیم؟

میتوان با ذکر کردن نام فرم به ابزارهای و متغییر های عمومی آن دسترسی پیدا کرد

Form2.Text1.Text

Form1.MinReng

برای روشن تر شدن مساله کد قسمتی از تنظیمات cowboy را می نویسیم

Form 1 :

قسمت declaration

```
public MinReng as integer
```

Form2 :

```
Private Sub Command1_Click()
   Form1.MinReng = CInt(Me.Text1.Text)
   Unload Me
End Sub
```

در اینجا قبل از اینکه فرم دوم از حافظه خارج شود و متغییرهای آن نابود شوند مقدارهایی که فرم اول لازم دارد را در متغیری در همان فرم ذخیره میکند.

Form 1 :

```
private MinReng as integer
Private Sub Command1_Click()
   Form2.Show   VBModal
   MinReng = Cint(Form2.Text1.Text)
End Sub
```

Form2 :

```
Private Sub Command1_Click()
```

```
    Me.Hide
End Sub
```

در این روش فرم را مخفی کرده ایم و از متغیر های آن استفاده کرده ایم . مشخصا اگر فرم با دکمه X بالای فرم بسته شود دچار مشکل می شویم.

تمرین :
تنظیمات را کامل کنید. و برای cowboyهم فرمی برای تنظیمات درست کنید که از آن inputbox بهتر باشد!

کارگاه :
در هنگام اجرا از روی یک فرم، فرم جدیدی درست کنید.
راهنمایی :
می خواهید یک فرم جدید(new) تعریف کنید.
پاسخ کارگاه :

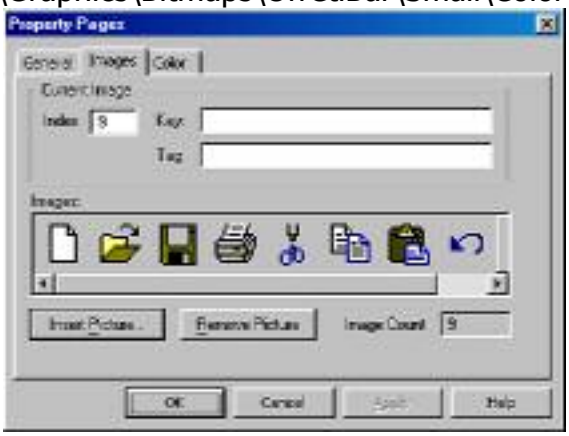
```
Dim x As New Form2, y As New Form2
y.Show
x.Show
```

فصل هفتم : یک برنامه کاربردی

در فصل های گذشته آنقدر ویژوال بیسیک یاد گرفته ایم که یک برنامه کاربردی بنویسیم.
البته در این برنامه از ابزارهای جدیدی هم استفاده می کنیم که همه شما در ویندوز با آنها کار کرده اید. و حتما میخواهید در برنامه خودتان هم استفاده کنید. ابزارهایی مانند toolbar و status bar و richtext و ...
از این فصل سرعت کارمان بیشتر میشود و بهتر است قبل از مطالعه این فصل در مطالب فصل های قبلی حرفه ای شده باشید.

MiniWordPad

ابتدا Microsft Windows Common Controls وMicrosoft Rich TextBox و Microsoft Common Dialogs را به components پروژه خودتان اضافه کنید.
یک ImageList روی صفحه قرار دهید و روی (custom) در خواص آن دبل کلیک کنید. و عکس های لازم برای toolbar یک برنامه ویرایش متن را به آن اضافه کنید (بد نیست سری به <VisualStduioDir>\Common\Graphics\Bitmaps\OffCtlBar\Small\Color بزنید.)



سپس یک toolbar به فرم اضافه کنید و ImageList آن را به ImageList1 ست کنید و برای هرکدام از عکس ها یک دکمه اضافه کنید و Image آن را به اندیس تصویر در imagelist اضافه کنید.یک combo هم به در toolbar قرار دهید. و مکان زیر آن را با placeholder به آن اختصاص دهید.

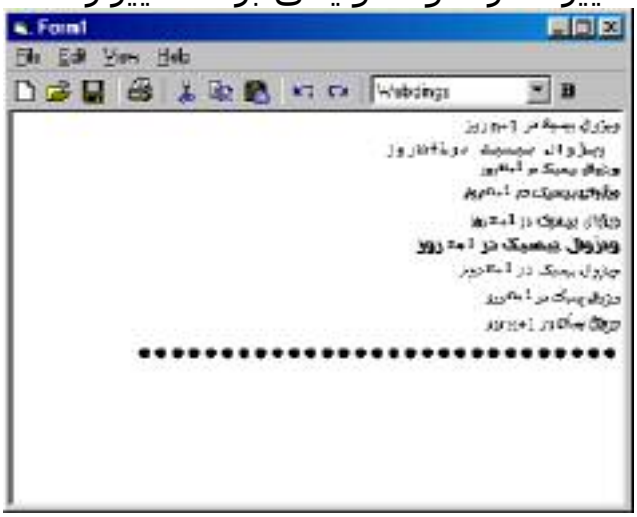
یک richtextbox هم اضافه کنید و منو های لازم را هم طراحی کنید.
در آن combo میخواهیم لیست فونت های سیستم را نمایش دهیم

```
Private Sub Form_Load()
Dim i As Integer
For i = 0 To Screen.FontCount - 1
   Combo1.AddItem Screen.Fonts(i)
Next
End Sub
```

و اگر یکی از آنها انتخاب شد فونت richtext به آن تغییر کند.

```
Private Sub Combo1_Click()
RichTextBox1.SelFontName = Combo1.Text
End Sub
```

RichTextBox جعبه متنی است که میتوان در آن قطعه های مختلف متن را با فونت های مختلف نوشت.
شما دکمه های italicbold و underline را بکار بیندازید و یک combo برای تغییر اندازه فونت و یکی برای تغییر رنگ آن اضافه کنید.



وقتی اندازه فرم تغییر میکند باید اندازه richtext نیز تغییر کند:

```
Private Sub Form_Resize()
   RichTextBox1.Width = Me.ScaleWidth
```

```
   RichTextBox1.Height = Me.ScaleHeight - 300
End Sub
```

عدد 300 را بدلیل اختلاف ارتفاع فرم و ارتفاع محیط کاری آن نوشته ایم.
در فصل بعد این فاصله را اندازه گیری خواهیم کرد.
حال شما کاری کنید که اندازه فرم از حد مشخصی کوچکتر نشود.
و در منوی view یک گزینه برای toolbar قرار دهید که checked آن true باشد .

```
Private Sub mnuViewToolbar_Click()
   mnuViewToolbar.Checked = Not mnuViewToolbar.Checked
   Toolbar1.Visible = mnuViewToolbar.Checked
   RichTextBox1.Top = IIf(mnuViewToolbar.Checked, Toolbar1.Height, 0)
End Sub
```

با کلیک کردن این منو toolbar ظاهر یا غیب میشود.
منوی Edit
این منو شامل cut,copy و paste است. البته میتوانیم AutoVerbMenu را برای RichTextBox برابر با True کنیم تاوقتی روی این ابزار کلید راست میزنیم منویی شبیه آنچه ما میخواهیم باز شود میتوانید خودتان این منو را درست کنید. البته کلید های میانبر richtext برایتان دردسر درست میکنند. ولی در ابزارهایی که خودشان از این کلید ها استفاده نمی کنند مشکلی ندارید.
Windows clipboard در حقیقت یک جعبه که هر برنامه ای میتواند چیزی را بطور موقت در آن قرار دهد. تا خودش یا سایر برنامه ها از آن استفاده کنند.
در ویژوال بیسیک Clipboard نیز یک شی است.

```
Private Sub mnuEditCopy_Click()
   Clipboard.SetText RichTextBox1.SelText, vbCFRTF
End Sub

Private Sub mnuEditPaste_Click()
   RichTextBox1.SelText = Clipboard.GetText(vbCFRTF)
End Sub
```

فکر میکنم خودتان از پس cut بر بیایید.
توضیح غیر ضروری : خواصی که با sel شروع میشوند معمولا درباره متن انتخاب (select) شده هستند.
Clipboard :

1. SetText

بخش متنی clipboard را برابر با پارامتر اولی می کند که می گیرد. پارامتر دوم نوع متن را تعیین میکند که از نوع متن عادی باشد یا RichTextFormat (این فرمت دارای مشخصات فونت و غیره است که متن عادی این مشخصات را ندارد.)

2. GetText

بخش متن clipboard را برمی گرداند. پارامتر آن مانند پارامتر دوم settext است.

3. GetData

بخش تصویر clipboard را برمیگرداند . فرمت های آن : VBCFBitmap و VBCFDIB و VBCFMetaFile است.

4. SetData

بخش تصویر را برابر با پارامتر اولی که میگیرد میکند.

5. Clear

برای خالی کردن clipboard

اگر کسی در یک تایمر clipboard.clear را اجرا کند چه می شود؟

منوی File

```
Private Sub mnuFileNew_Click()
Dim ret As VbMsgBoxResult
ret = MsgBox("Do u want to save the changes?", vbYesNoCancel, "MiniWordPas")
If ret = vbNo Then
   RichTextBox1.Text = ""
ElseIf ret = vbYes Then
   mnuFileSave_Click
End If
End Sub
Private Sub saveit(Optional fname As String = "")
   CommonDialog1.Filter = "Rich Text Format|*.rtf|Text File|*.txt"
   If fname = "" Then
      CommonDialog1.ShowSave
      sFileName = CommonDialog1.FileName
   Else
      sFileName = fname
   End If

   If Trim(sFileName) <> "" Then
      If Right(sFileName, 3) = "rtf" Then
         RichTextBox1.SaveFile sFileName
      Else
         RichTextBox1.SaveFile sFileName, rtfText
      End If
   End If
End Sub

Private Sub mnuFileSave_Click()
   saveit sFileName
End Sub

Private Sub mnuFileSaveas_Click()
   saveit
End Sub

Private Sub mnuFileOpen_Click()
   CommonDialog1.Filter = "Rich Text Format|*.rtf|Text File|*.txt"
   CommonDialog1.ShowOpen
```

```
   sFileName = CommonDialog1.FileName
   If Trim(sFileName) <> "" Then
      If Right(sFileName, 3) = "rtf" Then
         RichTextBox1.LoadFile sFileName
      Else
         RichTextBox1.LoadFile sFileName, rtfText
      End If
   End If
   RichTextBox1.Refresh
End Sub

Private Sub mnuFilePrint_Click()
   Printer.Print RichTextBox1.Text
   Printer.EndDoc
End Sub
```

در مورد Printer این نکته را بگویم که Printer هم در ویژوال بیسیک یک شی است و میتوان روی آن نوشت. پس از enddoc متن نوشته شده در printer چاپ میشود.

و اما toolbar

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
   Case 1:
      mnuFileNew_Click
   Case 2:
      mnuFileOpen_Click
   Case 3:
      mnuFileSave_Click
   Case 5:
      mnuFilePrint_Click
End Select
End Sub
```

اینجا کار زیادی نداریم فقط باید برای هر اندیس دکمه رویداد کلیک منوی مربوط به آن را فراخوانی کنیم یا اینکه کد مخصوص خودش را بنویسیم (اگر منویی مربوط به آن نیست.)

تمرین :

1. منویی اضافه کنید که FontDialog را برای تنظیم فونت باز کند.
2. روشی برای تنظیم فونت در برنامه قرار دهید که فونت همه متن را عوض کند نه فقط بخش انتخاب شده.

پروژه برنامه نویسی :

یک برنامه نقاشی با منو و toolbar و...

فصل هشتم : پدر windows ، برادر C

وقتی در منزلتان یک ابزار را نیاز دارید چه میکنید؟ اول جعبه ابزار خودتان را میگردید و اگر نداشتید از آشنایانتان میخواهید که آن را به شما قرض بدهند.
آشنایان ویژوال بیسیک windows و C هستند.
ویندوز دارای توابعی است که با زبان C نوشته شده اند. این توابع در DLL[8] های ویندوز قرار دارند و به آنها Windows API[9] می گویند. برای استفاده از این توابع باید ابتدا تعریف آنها را در برنامه خودمان بنویسیم این تعریف شامل نام تابع ، آرگمانهای آن و نامDLL ای که تابع در آن قرار دارد است. برای مثال تعریف یکی از API های ویندوز که نام آن SndPlaySound است و درwinmm.dll قرار دارد در زیر آمده است:

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

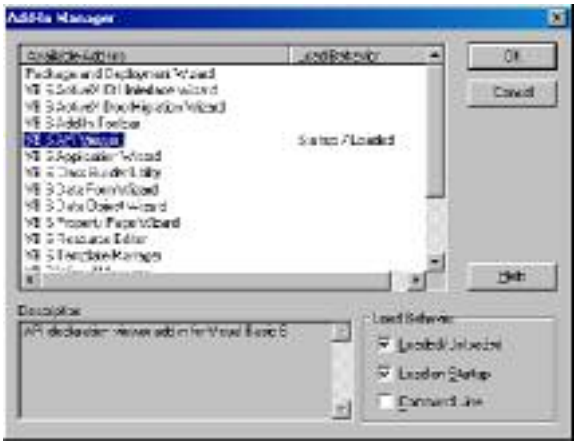
اگر فکر میکنید طولانی و گیج کننده است به BitBlt نگاه کنید (فلفل نبین چه ریزه!)

```
Public 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, ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal
dwRop As Long) As Long
```

یا این یکی :

```
Public Declare Function StretchBlt Lib "gdi32" Alias "StretchBlt" (ByVal hdc As
Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight
As Long, ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long,
ByVal nSrcWidth As Long, ByVal nSrcHeight As Long, ByVal dwRop As Long) As
Long
```

خوشبختانه نیازی به حفظ کردن اینها نیست.به منوی Add-ins\Add in ... manager بروید و مانند شکل روبرو برای VB6 API Viewer گزینه های Loaded و LoadOnStartup را انتخاب کنید. به همان منو برگردید. یک گزینه اضافه شده API Viewer آن را اجرا کنید.و در آن فایل Win32API.txt را load کنید. حالا در combo عبارت Declares را انتخاب کنید و روی نام یکی از API ها که در زیر آمده دبل کلیک کنید تا تعریف آن را ببینید . بجر تعریف ها در API Viewer بخش های Constants و Types نیز وجود دارد که constant ها و type های استفاده شده در api های ویندوز هستند.



[8] Dynamic Link Library
[9] Windows Application Programming Interface

مقدمه چینی کافی است تعریف sndplaysound و ثابت SND_ASYNC را به بخش declares در فرم خودتان اضافه کنید. (دقت کنید که تعریف تابع در فرم باید Private باشد.)

```
Private Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA"
(ByVal lpszSoundName As String, ByVal uFlags As Long) As Long
Private Const SND_ASYNC = &H1        ' play asynchronously

Private Sub Command1_Click()
   sndPlaySound "c:\windows\media\chimes.wav", SND_ASYNC
End Sub
```

اگر آدرس فایل درست باشد فایل پخش می شود. برای پایان دادن به آن میتوانید همین تابع را با نام فایل خالی صدا کنید. در MSDN بدنبال این تابع بگردید و راهنمای آن را بخوانید ، متاسفانه اکثر راهنماهای API ها به زبان C است. و این برای برنامه نویسان VB کمی سخت است.

Restart کردن windows

هر کاری که خود ویندوز میتواند انجام دهد شما هم میتوانید با API ها انجام دهید. یکی هم restart کردن.

```
Private Declare Function ExitWindowsEx Lib "user32" (ByVal uFlags As Long,
ByVal dwReserved As Long) As Long
Private Const EWX_LOGOFF = 0
Private Const EWX_SHUTDOWN = 1
Private Const EWX_REBOOT = 2
Private Const EWX_FORCE = 4

Private Sub Command1_Click()
   ExitWindowsEx EWX_REBOOT + EWX_FORCE, 0
End Sub
```

پارامتر اول کاری است که باید انجام شود. Shutdown,Reboot(restart) یا logoff و هر کدام از آنها که با ewx_force جمع شود به نوع اجباری آن عمل تبدیل میشود. در نوع عادی هر کدام از این ها برنامه ها میتوانند عمل خارج شدن از ویندوز را لغو کنند اما نوع اجباری قابل لغو کردن نیست.

HWnd : هندل، هر شی در ویندوز با یک عدد منحصر به فرد شناخته میشود. این عدد مانند نام آن شی است و برای صدا کردن شی بوسیله APIها به این عدد نیاز داریم. اگر شی در برنامه خودتان باشد میتوانید با object.hwnd هندل آن را بگیرید.

```
Private Declare Function EnableWindow Lib "user32" (ByVal hwnd As Long,
ByVal fEnable As Long) As Long
Private Sub Command1_Click()
   EnableWindow Text1.hwnd, False
End Sub
```

با EnableWindow میتوان یک window را فعال یا غیر فعال کرد. اگر هنوز با شنیدن نام window یاد فرم می افتید باید بگویم که در windows همه چیز یک window است حتی یک textbox.

اما فرق این API با اینکه از Enabled در Text1 استفاده کنیم چیست؟ تفاوت اصلی در این است که این API به wnd اهمیت میدهد نه به TextBox یعنی اگر شما هندل هر window دیگری را هم به این api بدهید کار میکند حتا اگر آن هندل در برنامه شما نباشد!
برای بدست آوردن هندل یک window که در برنامه ما نیست چند راه وجود دارد یکی از آنها در زیر آمده است :

```
Private Declare Function EnableWindow Lib "user32" (ByVal hwnd As Long,
ByVal fEnable As Long) As Long
Private Declare Function WindowFromPoint Lib "user32" (ByVal xPoint As Long,
ByVal yPoint As Long) As Long
Private Sub Command1_Click()
   Dim ret As Long
   ret = WindowFromPoint(0, 0)
   EnableWindow ret, False
End Sub
```

این API (WindowFromPoint) یک مختصات می گیرد و هندل پنجره ای را که الان در آن مختصات است بر می گرداند.(شاید نتیجه آزمایش کمی برایتان دردسر درست کند!)

گرفتن مختصات موس در صفحه :

```
Private Declare Function GetCursorPos Lib "user32" (lpPoint As POINTAPI) As
Long
Private Type POINTAPI
      x As Long
      y As Long
End Type

Private Sub Timer1_Timer()
   Dim pt As POINTAPI
   GetCursorPos pt
   Label1.Caption = pt.x
   Label2.Caption = pt.y
End Sub
```

گرفتن متن یک window :

این کار یکی از هیجان انگیزترین کارهایی است که میتوان در windows انجام داد.

```
Private Declare Function GetCursorPos Lib "user32" (lpPoint As POINTAPI) As
Long
Private Declare Function WindowFromPoint Lib "user32" (ByVal xPoint As Long,
ByVal yPoint As Long) As Long
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal
hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, lParam As Any) As
Long
Private Declare Function SendMessageString Lib "user32" Alias "SendMessageA"
(ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal
lParam As String) As Long
Private Const WM_GETTEXT = &HD
```

```
Private Const WM_GETTEXTLENGTH = &HE
Private Type POINTAPI
      x As Long
      y As Long
End Type

Private Function GetText(wnd As Long) As String
   Dim buf As String, l As Long
   l = SendMessage(wnd, WM_GETTEXTLENGTH, 0, 0)
   buf = Space(l)
   SendMessageString wnd, WM_GETTEXT, l + 1, buf
   GetText = buf
End Function

Private Sub Command1_Click()
   Timer1.Enabled = Not Timer1.Enabled
End Sub
Private Sub Timer1_Timer()
   Dim pt As POINTAPI, wnd As Long
   GetCursorPos pt
   wnd = WindowFromPoint(pt.x, pt.y)
   Label1.Caption = GetText(wnd)
End Sub
```

شاید بد نباشد که روی پنجره DialUp Networking خودتان این برنامه را آزمایش کنید . مخصوصا اگر windows9x دارید.

توضیح تابع GetText :

اولین sendmessage طول متن این هندل را برمیگرداند.

دستور space یک آرگمان میگیرد و رشته ای با آن تعداد space برمیگرداند.

دومین sendmessage که در حقیقت با نام دیگری از همان تابع است اما آرگمان آخر آن رشته است، متن را میگیرد و در متغیر buf قرار میدهد.

آرگمان سوم آن طول رشته است و به این دلیل یکی بیشتر از طول واقعی است که در C کاراکتر آخر رشته کارکتر کد اسکی صفر(NULL) است. و ما نمی خواهیم این کاراکتر در رشته ما باشد.

HDC : در حقیقت بوم نقاشی در ویندوز است. اگر جایی گرافیک هست ، dc هم هست.برای گرفتن hdc از تابع GetDC(hwnd) استفاده میکنیم تا dc یک wnd را بگیریم.

برنامه عکس گرفتن از صفحه:

یک picturebox در مختصات 0,0 از فرم خودتان قرار دهید، BorderStyle فرم و pictureرا برابر با صفر کنید و windowstate در فرم را برابر با Maximized کنید.

```
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As
Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal
hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop As
Long) As Long
Private Declare Function GetDesktopWindow Lib "user32" () As Long
Private Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long
```

```
Private Const SRCCOPY = &HCC0020 ' (DWORD) dest = source
Private Sub Form_Load()
   Dim wnd As Long, dc As Long
   Picture1.AutoRedraw = True
   wnd = GetDesktopWindow
   dc = GetDC(wnd)
   Form_Resize
   BitBlt Picture1.hdc, 0, 0, Screen.Width / 15, Screen.Height / 15, dc, 0, 0,
SRCCOPY
End Sub

Private Sub Form_Resize()
   Picture1.Move 0, 0, Screen.Width, Screen.Height
End Sub

Private Sub Picture1_Click()
   SavePicture Picture1.Image, "c:\test.bmp"
   Unload Me
End Sub
```

تابع bitblt تصویر روی یک dc را روی یک dc دیگر رسم میکند.

تغییر دادن شکل فرم :
اگر از شکل مستطیل فرم خسته شده اید میتوانید آن را تغییر دهید!


```
Private Declare Function CreateEllipticRgn Lib "gdi32" (ByVal X1 As Long, ByVal
Y1 As Long, ByVal X2 As Long, ByVal Y2 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 Sub Command2_Click()
Unload Me
End Sub

Private Sub Form_Load()
   Dim rgn As Long
   rgn = CreateEllipticRgn(0, 0, Me.Width / 15, Me.Height / 15)
   SetWindowRgn Me.hWnd, rgn, True
End Sub
```

تابع CreateEllipticRgn یک محدودة بیضی شکل تعریف میکند و
SetWindowRgn آن محدوده را در یک window تاثیر میدهد. درباره api های
زیر تحقیق کنید.

CreatePolygonRgn

CreateRoundRectRgn
CombineRgn

فایل های INI :

فایل های ini نوع خاصی از فایلهای متنی هستند که بیشتر برای ذخیره کردن تنظیمات برنامه بکار میروند. و برای تقویت کردن حافظه برنامه بسیار موثرند.

```
Private Declare Function WritePrivateProfileString Lib "kernel32" Alias
"WritePrivateProfileStringA" (ByVal lpApplicationName As String, ByVal
lpKeyName As Any, ByVal lpString As Any, ByVal lpFileName As String) As Long
Private Declare Function GetPrivateProfileString Lib "kernel32" Alias
"GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName
As Any, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize
As Long, ByVal lpFileName As String) As Long
Private iLeft As Integer, iTop As Integer

Private Sub Command2_Click()
Unload Me
End Sub

Private Sub Form_Load()
   Dim buf As String
   buf = Space(255)
   GetPrivateProfileString "myAPP", "Left", "10", buf, 255, "testini.ini"
   iLeft = CInt(Left(Trim(buf), Len(Trim(buf)) - 1))
   buf = Space(255)
   GetPrivateProfileString "myAPP", "Top", "10", buf, 255, "testini.ini"
   iTop = CInt(Left(Trim(buf), Len(Trim(buf)) - 1))

   Me.Move iLeft, iTop
End Sub

Private Sub Form_Unload(Cancel As Integer)
   WritePrivateProfileString "myAPP", "Left", CStr(Me.Left), "testini.ini"
   WritePrivateProfileString "myAPP", "Top", CStr(Me.Top), "testini.ini"
End Sub
```

این هم عکسی از فایل ini برنامه ما در notepad :

اگر در قسمت filename آدرس را تعیین نکنید فایل ini در دایرکتوری windows ذخیره میشود.

هنگامی که ini ورافتاد!

از وقتی که windows95 وارد بازار شد فایل های ini کمرنگ تر و کمرنگ تر شدند و جای خود را به رجیستری دادند. رجیستری از یک نظر شبیه یک فایل ini بزرگ است که تمام برنامه ها تنظیمات خود را در آن می نویسند. برای کار با رجیستری میتوانید از دستورات vb یا از api ها استفاده کنید. در اینجا به دستورات vb اکتفا میکنیم

```
Private Sub Form_Load()
   Me.Left = CInt(GetSetting("myAPP", "Position", "Left", 10))
   Me.Top = CInt(GetSetting("myAPP", "Position", "Top", 10))
End Sub

Private Sub Form_Unload(Cancel As Integer)
   SaveSetting "myAPP", "Position", "Left", Me.Left
   SaveSetting "myAPP", "Position", "Top", Me.Top
End Sub
```

کار با این دستورات بسیار راحت است. اما با آنها نمیتوان همه جای رجیستری نوشت . در عوض کارکردن با API های رجیستری سخت تر است و من همه کارهای سخت را به شما واگذار میکنم!

برای مطالعه بیشتر در مورد API ها به MSDN یا کتابهایی که در این مورد هست مراجعه کنید.
سایت http://www.allapi.net/ نیز یک برنامه مفید دارد که در آن برای اکثر توابع ویندوز مثال و توضیح دارد.

تمرین :

1. برنامه ای که هرگاه پنجره Connect To باز شد username و password و شماره تلفن را در یک فایل ذخیره کند.
2. تصویر صفحه را کوچک کنید تا اندازه picturebox شود.(stretch)

فصل نهم : ارتباط با دیگران(winsock.ocx)

تا اینجا برنامه های ما فقط روی یک کامپیوتر بودند ولی در این فصل برنامه هایی مینویسیم که روی دو یا چند کامپیوتر مجزا کار میکنند. برای آزمایش کردن این برنامه ها نیازی به دو کامپیوتر ندارید فقط کافی است که دو ویژوال بیسیک باز کنید و دو برنامه را مجزا بنویسید.
در حقیقت اینجا دو یا چند برنامه (نه الزاما متفاوت) داریم که میخواهند با هم ارتباط برقرار کنند. هر ارتباطی نیاز به ابزار ارتباطی دارد ما این ارتباط را با سیستم تلفن مثال میزنیم. در این ارتباط ابزار winsock تلفن و IP(Internet Protocol) شماره تلفن و Port شماره داخلی ما است.
دو ویژوال بیسیک باز کنید و به هر دو آنها ابزار Microsoft Winsock Control را اضافه کنید.
نام فرم یکی را frmClient و دیگری را frmServer کنید.
در فرم frmClient دو دکمه و یک جعبه متن و یک winsock قرار دهید. و در سرور نیز یک جعبه متن و یک winsock قرار دهید.
به این مدل برنامه، برنامهٔ مشتری / خدمتگزار(Client/Server) میگویند . در این سیستم ابتدا مشتری از خدمتگزار درخواست ارتباط میکند؛ خدمتگزار در صورت موافقت درخواست را تایید میکند و ارتباط برقرار میشود. از این لحظه تا زمان قطع کردن ارتباط همه چیز بر اساس فرستادن و دریافت کردن پیغام است. قطع ارتباط میتواند از هر طرف هر کدام از مشتری یا خدمتگزار باشد.

 چگونه خدمتگزار منتظر درخواست شود؟

```
Private Sub Form_Load()
   Winsock1.LocalPort = 10300
   Winsock1.Listen
End Sub
```

مثال تلفن را بخاطر دارید؟ ما باید تلفن(winsock) خودمان را به یکی از شماره های داخلی(port) وصل کنیم خط اول شماره داخلی 10300 را انتخاب میکند. باید دقت کنیم که از خط سایر تلفن ها استفاده نکنیم. خط های بالای 1024 معمولا خالی هستند . اما پورتی مانند 80 به احتمال زیاد مشغول است (این پورت مربوط به webserver است)
خط دوم تلفن ما را به خط داخلی انتخاب شده وصل میکند. و گوش به زنگ تلفن می ماند.

- چگونه مشتری درخواست ارتباط کند؟

```
Private Sub Command1_Click()
   Winsock1.Connect "127.0.0.1", 10300
End Sub
```

یا

```
Private Sub Command1_Click()
   Winsock1.RemoteHost = "127.0.0.1"
```

```
Winsock1.RemotePort = 10300
Winsock1.Connect
End Sub
```

در اینجا خط اول شماره تلفن (IP address) کامپوتری است که میخواهید به آن متصل شوید. ما هر دو برنامه را روی یک کامپیوتر اجرا کرده ایم پس این شماره آدرس IP همان کامپیوتر است. اگر مانند من به شبکه ای وصل نباشید احتمالا IP شما هم همین عدد است ولی اگر این عدد نیست میتوانید آن را بدست بیاورید ، Winsock1.LocalIP آدرس IP همین کامپیوتر را برمیگرداند.
بجای IP میتوانید از نام سرور نیز استفاده کنید. که میتواند یک نام دامنه مانند www.yahoo.com باشد. HostName کامپیوتر خودتان را میتوانید با winsock1.LocalHostName بدست بیاورید.
Remoteport شماره port ای است که برنامه ای که شما میخواهید به آن متصل شوید آنجا گوش به زنگ است.
Connect درخواست شما را به آدرس گفته شده می فرستد.

- چگونه سرور درخواست دریافت شده را تایید کند؟

```
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
If Winsock1.State <> sckConnected Then Winsock1.Close
Winsock1.Accept requestID
End Sub
```

وقتی یک درخواست به سرور میرسد رویداد ConnectionRequest اتفاق می افتد.
State نشان دهنده وضعیت فعلی winsock است تعدادی از مقادیر آن را در پایین می بینید:

Constant	Value	توضیح
sckClosed	0	پیش فرض، ارتباطی برقرار نیست.بسته
sckListening	2	گوش به زنگ، منتظر درخواست.
sckConnecting	6	در حال برقراری ارتباط
sckConnected	7	ارتباط برقرار شده
sckClosing	8	در حال قطع کردن ارتباط
sckError	9	خطا پیش آمده.

Close ارتباط را قطع میکند.
Accept درخواست را قبول میکند . مقداری که میگیرد مشخص میکند که کدام شماره درخواست تایید شده است.

- چگونه اطلاعات (پیغام) بفرستیم؟

```
Private Sub Command2_Click()
Winsock1.SendData Text1.Text
```

```
End Sub
```

دستور senddata یک پارامتر میگیرد که اطلاعاتی است که باید فرستاده شود. و آن را به طرف مقابل در ارتباط میفرستد.(در مورد مشتری و خدمتگزار فرقی نمیکند.)

- چگونه پیغام را دریافت کنیم ؟

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
   Dim mydata As String
   Winsock1.GetData mydata, vbString
   Text1.Text = mydata
End Sub
```

وقتی اطلاعاتی میرسد این رویداد فراخوانی می شود.
Bytetotal کل تعداد بایت های رسیده است.
GetData اطلاعات را میگیرد و در متغیر پارامتر اول قرار میدهد. نوع اطلاعات را پارامتر دوم تعین میکند و پارامتر سوم طول اطلاعات است که ما اینجا از آن استفاده نکرده ایم.

- چگونه ارتباط را قطع کنیم؟

برای این کار از متد close استفاده میکنیم

دقت کنید که موقع فرستادن اطلاعات باید ارتباط برقرار شده باشد.
وقتی ارتباط قطع میشود اگر میخواهید سرور بتواند باز هم ارتباط جدیدی را قبول کند باید دوباره آن را گوش به زنگ کنید.

```
Private Sub Winsock1_Close()
   Winsock1.Close
   Winsock1.Listen
End Sub
```

بیشتر از نمایش پیغام

میتوان برنامه آزمایشی که نوشتیم را به یک چت تبدیل کرد. و حتا میتوان کارهای بسیار هیجان انگیزتری هم انجام داد!
گرفتن آدرس دایرکتوری ویندوز یک کامپیوتر دیگر:
کد مشتری :

```
Private Sub Command1_Click()
   Winsock1.Connect "127.0.0.1", 10300
End Sub

Private Sub Command2_Click()
   Winsock1.SendData "WinDir"
End Sub

Private Sub Command3_Click()
   Winsock1.Close
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
   Dim sData As String
```

```
   Winsock1.GetData sData, vbString
   If Left(sData, 6) = "WinDir" Then Text1.Text = _
      Right(sData, Len(sData) - 6)
End Sub
```

بخش خدمتگزار :

```
Private Declare Function GetWindowsDirectory Lib "kernel32" Alias
"GetWindowsDirectoryA" (ByVal lpBuffer As String, ByVal nSize As Long) As
Long
Private Sub Form_Load()
   Winsock1.LocalPort = 10300
   Winsock1.Listen
End Sub

Private Sub Winsock1_Close()
   Winsock1.Close
   Winsock1.Listen
End Sub

Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
   If Winsock1.State <> sckConnected Then Winsock1.Close
   Winsock1.Accept requestID
End Sub

Private Function WindowsDir()
   Dim buf As String
   buf = Space(255)
   GetWindowsDirectory buf, 255
   WindowsDir = Left(Trim(buf), Len(Trim(buf)) - 1)
End Function

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
   Dim mydata As String
   Winsock1.GetData mydata, vbString
   If mydata = "WinDir" Then Winsock1.SendData mydata + WindowsDir
End Sub
```

این کار خیلی به همان برنامه قبلی شبیه است فقط یک پیغام مشخص فرستاده میشود و طرف دیگر اگر آن پیغام را دریافت کند بجای نشان دادن کار خاصی (مانند پیدا کردن آدرس ویندوز) انجام میدهد و نتیجه را با یک سرایند مناسب (مثلا همان پیغامی که دریافت کرده) بر میگرداند.

پروژه برنامه نویسی :

برنامه ای بنویسید که با کامپیوتر مشتری بتوان کامپیوتر خدمتگزار را کنترل کرد. هرچه کنترل بیش ، برنامه به!

ضمیمه : آشنایی با دستورات بیسیک

دستور شرطی IF

ساختار دستور :

if condition *then* statement *else* statment

به عنوان یک مثال از زبان محاوره‌ای میتوان گفت "*اگر* تلاش کنی *آنگاه* میتوانی فلان کار را انجام دهی *در غیر این صورت* نمیتوانی"

در بخش شرط هر عبارتی که نتیجه بولی(یا قابل تبدیل به آن) داشته باشد میتواند قرار بگیرد، عبارتهای مقایسه‌ای مانند a<b یا یک تابع که مقدار برگشتی بولی دارد. و در بخش دستور(statement) یک دستور بیسیک قرار میگیرد.

درصورتی که بخواهیم در بخش دستور بیستر از یک دستور بنویسیم از ساختار بلوکی استفاده میکنیم

```
if condition then
   statements
else
   statements
end if
```

مثال :

میخواهیم ماکسیمم a و b را در متغیر c کپی کنیم.

```
if a<b then c=b else c=a
```

حلقه For

این حلقه برای تکرار کردن یک عمل به تعداد دفعات مشخص است.

```
for var = start_index to  end_index [step step_count]
   statements
next [var]
```

متغیر var را ابتدا برابر start_index قرار میدهد و سپس مقدار آن را هر بار به اندازه step_count اضافه میکند تا به end_index برسد. پس از هر بار تغییر دادن مقدار var یک بار دستورات اجرا میشوند مگر اینکه مقدار var از end_index بیشتر شده باشد.

با دستور exit for میتوان قبل از موعد از حلقه for خارج شد.

در صورتی که کار حلقه بصورت عادی (بدون استفاده از exit for) خاتمه یابد مقدار var پس از خروج از حلقه برابر با end_index+step_count خواهد بود.

مقدار پیشفرض برای step_count برابر با یک است، این مقدار میتواند منفی نیز باشد.

مثال :

محاسبه مجموع اعداد یک تا صد

```
for i=1 to 100
   sum=sum+i
next
```

حلقه While

اجرای دستورات تا زمانی که یک شرط برقرار است. شرط حلقه قبل از ورود به حلقه چک میشود. و اگر درست نباشد حلقه هرگز اجرا نمیشود.

ساختار :

```
while (condition)
   statements
wend
```

حلقه Do

```
Do [{While | Until} condition]
   statements
Loop
```

در اینجا شرط در ابتدای حلقه چک میشود.اگر while انتخاب شود دقیقا مانند حلقه while عمل میکند. اگر until انتخاب شود دستورات تا زمانی اجرا میشوند که شرط برقرار نباشد.
با exit do میتوان از حلقه do خارج شد.

```
Do
   statements
Loop [{While | Until} condition]
```

درصورتی که بخش شرطی پس از loop نوشته شود، شرط پس از اجرای دستورات چک میشود پس دستورات حداقل یک بار اجرا میشوند. کاربرد کلمات کلیدی while و until مانند قسمت قبلی است.