

فصل اول

پروگرام نویسی شی گرا

Object-Oriented Programming

پروگرام های کمپیوتر روز به روز وسعت پیدا کرده و جهت تهیه آن به هزارها سطر، طبق قوانین املائی و منطقی پروگرام نویسی ضرورت است. اینکار حجم پروگرام هارا به شکل مبهم کننده افزایش داده و کار پروگرام نویسی را مشکل ساخته است. برای حل این مشکل از روش های مختلف استفاده گردیده است. یکی از این روش ها استفاده از لسانهای پروگرام نویسی شی گرا (Object-oriented) است که در اوائل دهه 1990 معرفی گردیده و تا به حال بهترین روش برای خورد ساختن حجم پروگرام ها، و سهولت در پروگرام نویسی شناخته شده است. پروگرام نویسی شی گرا یک روش موثر، و یک راه جدید جهت ترتیب و تنظیم یک پروگرام است که کار پروگرام نویسی را نهایت آسان ساخته است. این لسان ها از آبجکتها یا آبجکت های از قبل تهیه شده استفاده میکند، و به همین سبب بنام لسان های پروگرام نویسی شی گرا (Object-oriented Languages) یاد میشود. اساس پروگرام نویسی شی گرا استفاده دوباره (Reuseability) است. در این روش پروگرام به صورت مجموعه از اشیاء (Objects) در نظر گرفته میشود. در حالی که این اشیاء، متحولین (Variables) و پروگرامهای فرعی (Methods) را در داخل خود جابه داده و به کمک همین متحولین و پروگرامهای فرعی با محیط خارج و آبجکتهای دیگر ارتباط برقرار مینماید.

لسان های پروگرام نویسی ویژول بسیک (Visual Basic)، ++C، و ویژول سی (Visual c) ، سی شارپ (#C)، SmallTalk و جاوا (Java) نمونه لسان های پروگرام نویسی شی گرا میباشد. قابل یادآوری است که لسانهای جاوا و SmallTalk از ابتدا و اساس خود شی گرا (Object-oriented) بوده، و لسان ویژول بسیک مرحله به مرحله شی گرا گردیده است. هر نمونه جدید ویژول بسیک به شی گرا نزدیکتر شده است که نمونه ویژول بسیک دات نت (Visual Basic .NET) کاملاً شی گرا میباشد. این لسانها روش مرئی (Visual) برای پروگرام نویسی دارد یعنی در این لسانها به عوض کلمات میتواند با اشکال کار شود.

در پروگرام نویسی شی گرا، به عوض اینکه یک پروگرام بحر از متحولین و پروگرام های فرعی باشد، همین متحولین و پروگرام های فرعی در گروپ ها باهم یکجا گردیده و در اثنای پروگرام نویسی از همین گروپ ها استفاده میشود. این گروپ ها را بنام کلاسها (Classes) یاد گردیده و آبجکتها (Objects) نمونه کلاسها میباشد. یعنی آبجکت عبارت از چیزی است که هم شامل توابع / پروگرامهای فرعی (Methods) وهم شامل data یعنی متحولین (Variables) است. پروگرامهای فرعی کارهای که آبجکت همان کلاس انجام داده میتواند، و متحولین خصوصیات آبجکت ها را تعیین میکند. در ابتدا باید نقشه هر آبجکت تعیین گردد. نقشه که از روی آن آبجکتها تعریف میگردد بنام کلاس (Class) یاد میشود. به عباره دیگر آبجکتها توسط کلاسها ساخته میشود. مشخصات و وظایف جدید به آسانی به آبجکتها اضافه شده میتواند. در پروگرام نویسی شی گرا Data یک چیز مهم و اساسی تلقی میشود و باید به احتیاط کار شود و به این ترتیب حرکت آن و تغییرات در آن نباید آزادانه باشد. در این روش Data به شکل محکم به توابع که آنرا پراسس میکند بسته میشود و دیزاین کلاس ها اینرا تعیین میکند که آبجکتی (Object) مربوط آن، کدام معلومات (Data) و کدام میتود ها را استفاده کرده میتواند. همچنین در دیزاین کلاس، تعیین میگردد که آبجکتها به چه شکل از Data و میتود های مخفی شده استفاده کرده میتواند. متحولین (Data) خصوصیات (Properties)، و میتودها عملکرد آبجکتها همان کلاس را تعیین میکند.

لسانهای مانند پاسکال، بسیک، و کوبول، شی گرا نبوده بلکه لسانهای عملگرا (Action-oriented) بوده و بنام لسانهای وظیفوی (Procedural Languages) نیز یاد میشود. در این لسان ها، Data و

تابع از هم جدا و مستقل از یکدیگر اند. در این لسانها هر قدم باید پروگرام شود. کد Client و Server از هم مستقل نبوده بلکه به یکدیگر بسته میباشد. در صورت تغییر در انترفیس، کد هم باید تغییر داده شود. برای پروگرام نویسی در این لسانها، پروگرام نویس باید به ساختن الگوریتم و ترکیب معلومات اولیه (Data Structures) بلدیت داشته باشد. در این لسانها سخن Niklaus Wirth، دیزاین کننده لسان پاسکال (اولین لسان ترکیبی و مشهور ترین لسان در سالهای 1970 و اوائل دهه 1980) به خوبی صدق میکند. Niklaus Wirth گفته بود:

پروگرام = الگوریتم + ساختار معلومات اولیه

نوت: باید دقت نمائیم که چند سطر فوق هیچگاه به معنی آن نیست که امروز از اهمیت الگوریتم (Algorithm) و ترکیب معلومات اولیه (Data Structures) کاسته شده است. الگوریتم و ترکیب معلومات قسمت عمده و مهم در پروگرام نویسی بوده، و در اثنای دیزاین لسانهای شی گرا نیز از الگوریتم و ساختار معلومات اولیه استفاده میشود. ترکیب معلومات زمینه وسیع برای تحقیقات علمی در ساحه کمپیوتر ساینس را دارا میباشد.

در لسانهای شی گرا مانند ویژول بسیک، پروگرام نویس با آبجکتها (Objects) سرو کار دارد. برای مثال اگر در پروگرام اکسل از ویژول بسیک استفاده مینمائیم، فایل اکسل (Work Book) یک شی یا آبجکت است، صفحه (Sheet) یک آبجکت است، حجره (Cell) یک آبجکت است، و به همین ترتیب ده ها آبجکت دیگر در اکسل موجود است. همچنین در خود ویژول بسیک، فورم (Form) یک آبجکت است، باکس متن (Text Box) یک آبجکت است، دکمه امر (Command Button) یک آبجکت است. لسانهای پروگرام نویسی شی گرا میتواند از آبجکت های خارجی، یعنی آبجکت های پروگرامهای دیگر نیز استفاده نماید. آبجکت ها بسیاری موارد پروگرام نویسی را از پروگرام نویس مخفی ساخته، یا به عبارت دیگر، جدا ساخته و به این ترتیب کار پروگرام نویسی را آسان مینماید. قابل یادآوری است که یک لسان را وقتی واقعا شی گرا گفته میتوانیم که هم از آبجکت ها استفاده نمایند و هم توانایی ایجاد آبجکت های جدید را داشته باشد. و به این ترتیب هر پروگرام آن در داخل یک کلاس تعریف گردد تا از همین کلاسها آبجکتها ایجاد شده بتواند. لسان جاوا (Java) و لسان ویژول بسیک دات نت لسانهای کاملا شی گرا است که هر پروگرام آن در داخل کلاس تعریف میشود.

آبجکت ها دارای خصوصیات (Properties) است. خصوصیات طرز نمایش آبجکت ها را اداره میکند. یا به عباره دیگر متحولین مخفی شده را بنام خصوصیات (Properties) یاد مینمایند. همچنین می توانیم به این خصوصیات مانند متحولین قیمت دهیم. در اثر تغییر قیمت متحولین، خصوصیت آبجکت و طرز نمایش آنرا تغییر داده میتوانیم.

آبجکت ها دارای میتودها (Methods) هم است. میتود چیزی را گویند که آبجکت آنرا انجام داده می تواند. یا به عباره دیگر توابع و پروگرام های که در داخل کلاس تعریف گردیده میتود ها برای آبجکت های همان کلاس میباشد. میتود ها در حقیقت اعمالی است که توسط آبجکت ها انجام شده میتواند.

متحولین (Properties) هسته آبجکت ها را تشکیل میدهد. میتودها هسته را احاطه نموده و آنرا از آبجکت های دیگر مخفی نگاه میکند. پاکت کردن (Packaging) متحولین در بین میتود ها را بنام محفوظ سازی (Encapsulation) یاد میکند. دلایل زیادی موجود است که به اساس آن متحولین محفوظ شده باید به شکل مستقیم مورد دسترسی قرار نگیرند بلکه از طریق میتودهای آبجکت های مربوطه مورد دسترسی قرار گیرد. این دلایل را در صفحات آینده مورد بحث قرار میدهیم. البته ممکن است تا این متحولین به شکل مستقیم و بدون میتود ها هم مورد دسترسی قرار گیرد. این امر مربوط به این میشود که این متحولین به شکل خصوصی (Private) تعریف گردیده و یا به شکل عمومی (Public) تعریف گردیده است.

آبجکت ها دارای حادثات (Events) است. حادثه چیزی است که بالای آبجکت صورت میگیرد. یا به عباره دیگر، حادثه عبارت از جواب یا عکس العمل آبجکت ها در مقابل عملیات است که بالای آنها اجرا میشود. مثلا عکس العمل آبجکت در مقابل فشار دادن دکمه طرف راست و یا دکمه طرف چپ موس (Click)، عکس العمل آبجکت در مقابل فشار دادن یک کلید، عکس العمل آبجکت در اثنای باز شدن، عکس العمل آبجکت در اثنای بسته شدن، و یا عکس العمل یک آبجکت در اثر تغیر خوردن محتوای همان آبجکت.

در زنده گی روزمره تلفون موبایل شما یک آبجکت است. این آبجکت دارای خصوصیات مانند رنگ، وزن، تعداد دکمه ها، و غیره میباشد. همچنین این آبجکت دارای میتود ها است مانند زنگ زدن، ذخیره کردن یک شماره، پالیدن شماره یک شخص، و غیره. موبایل دارای حادثات هم است مانند آمدن زنگ و یا پیغام، و غیره. شما از موبایل تان استفاده میکنید بدون اینکه تفصیلات طرز کار داخلی موبایل را بفهمید. در لسان های شی گرا به عین ترتیب میتوانیم از آبجکت ها استفاده نماییم بدون اینکه از تفصیلات طرز کار آن باخبر باشیم. در حقیقت مخفی ساختن تفصیلات یک سیستم، استفاده سیستم را آسانتر میسازد.

اصطلاحات پروگرام نویسی شی گرا

محفوظ سازی (Encapsulation)

طوریکه قبلا گفته شد، آبجکتها را به کمک کلاسها (Classes) ساخته میتوانیم. آبجکتها (Objects) در داخل خود، متحولین (Variables) و پروگرامهای فرعی (Methods) را جابجاده و آنرا مخفی میسازد. در اثنای استفاده این آبجکت ها، به کمک متحولین مخفی شده نمایش آبجکت ها را تعیین کرده میتوانیم. و به کمک پروگرامهای فرعی مخفی شده، آبجکت ها کار های مختلف را انجام داده میتوانند. درحقیقت پروگرامهای مخفی شده که بنام میتود ها یاد میشود، راه های ارتباط (Interfaces) به آبجکت ها میباشند. هر آبجکت دارای انترفیس های عمومی است که آبجکت های دیگر از طریق آن به این آبجکت ارتباط میگیرد. معلومات که در داخل آبجکت است تنها از طریق انترفیسها مورد دسترسی قرار گرفته میتواند. شما موبایل تانرا از طریق فشار دادن دکمه ها استفاده میکنید. پس این دکمه ها راه ارتباط (Interface) میان شما و موبایل تان میباشد. روش کار موبایل در داخل خود موبایل محفوظ بوده و شما از آن خبر ندارید. این محفوظ سازی معلومات (Encapsulation) سیستم استفاده از موبایل را برای هر کس خیلی آسان ساخته است.

استفاده دوباره (Reusability)

استفاده دوباره عبارت از قابلیت است که توسط آن میتوانیم یک پروگرام فرعی و یا قسمت از آنرا در پروگرامهای دیگر، و به تکرار (چندین بار) استفاده نماییم. و استفاده دوباره در حقیقت یکی از فواید اساسی پروگرام نویسی شی گرا است. وقتیکه شما یک کلاس را ایجاد میکنید، شما همان کلاس را در پروژه های مختلف استفاده کرده میتوانید. در پروگرام نویسی شی گرا ما میتوانیم از روی یک کلاس تعریف شده از قبل، ده ها و صدها و هزارها و یا بهتر است به یک عبارت دیگر بیان کنیم که هر چند آبجکت که خواسته باشیم، آنرا ایجاد کرده میتوانیم. ایجاد آبجکت ها از کلاس های آن را بنام نمونه سازی (Instantiation) یاد میکند. استفاده دوباره یک اصل مهم در انجینری سافت ویرها (Software Engineering) میباشد.

اشکال مختلف یا تغیر شکل (Polymorphism)

در پروگرام نویسی شی گرا از وراثت و تغیر شکل برای تهیه (Develop) پروگرامهای پیچیده استفاده میشود. به کمک تغیر شکل پروگرامهای را ساخته میتوانیم که از کلاسهای زیاد استفاده کرده میتواند و کلاس های جدید را نیز به آسانی به پروگرام اضافه شده میتواند. به کمک تغیر شکل سیستم را دیزاین کرده میتوانیم که خوبتر و به آسانی قابل توسعه باشد. در اینصورت پروگرامها میتواند آبجکتهای هر کلاس را استفاده کند. کلاس های جدید به آسانی و با تغیر کم به پروگرام اضافه شده

میتواند به شرط که این کلاسهای جدید جز تسلسل وراثت باشد. در صورت اضافه شدن کلاسهای جدید صرف همان اجزای پروگرام تغییر داده میشود که به فهم مستقیم کلاسهای جدید ضرورت دارد. یک شی (Object) میتواند در اوقات مختلف، اشکال و انواع مختلف را بخود اخذ نماید. یا به عبارت دیگر، کلاس های مختلف از آبجکت ها، که دارای عین نام اند، میتواند به اشکال مختلف تطبیق گردد.

بخش بندی (Modularity)

کود یک آبجکت میتواند مستقل از کود آبجکت دیگر باشد. همچنین آبجکت به آسانی میتواند در هر بخش سیستم استفاده شود. شما میتوانید موتر خود را به کسی دیگر بدهید و باز هم موتر کار میکند.

کلاس (Class)

هر آبجکت نمونه (Instance) یک کلاس است. مثلاً موتر نمونه یک کلاس است که آنرا بنام کلاس یا مدل موتر یاد کرده میتوانیم. برای توضیح بیشتر گفته میتوانیم که موتر آبجکت و مدل موتر عبارت از کلاس آن میباشد. کلاس در حقیقت مانند نقشه و قالب برای آبجکت ها میباشد که از روی همین نقشه، قالب و یا مدل ده ها و صد ها آبجکت دیگر، که مربوط کلاس خود میباشد و تمام خصوصیات کلاس خود را به ارث میبرد، ایجاد کرده میتوانیم. هر فورم جدید که شما آنرا به یک پروژه ویژول بسیک اضافه میکنید نمونه کلاس فورم است. هر صفحه (Sheet) جدید که شما آنرا در یک فایل اکسل اضافه میکنید نمونه کلاس صفحه (Sheet) است. فورم ویژول بسیک و یا صفحه اکسل تمام خصوصیات را از کلاس خود به ارث برده و شما هم میتوانید خصوصیات آنرا تغییر دهید. قابل تذکر است که شما میتوانید آبجکت های یک پروگرام را در پروگرام دیگر استفاده کنید.

کلاس عبارت از Blueprint است که متحولین و میتود های مشترک تمام آبجکت های یک نوع را تعریف میکند. کلاس برای موتر دربرگیرنده متحولین است که حالت موتر را حفظ میکند. همچنین کلاس موتر طرز تطبیق میتود ها را تعریف میکند. بعد از ایجاد نمودن کلاس موتر، شما میتوانید به هر تعداد آبجکت همان کلاس (موتر ها) را ایجاد نمایید.

علاوه بر متحولین نمونه (Instance Variables) کلاس میتواند متحولین کلاس (Class Variables) را نیز تعریف نماید. متحول کلاس دارای معلومات است که بین تمام نمونه های کلاس مشترک است. برای مثال فرض نمائید که تمام موتر ها دارای عین تعداد تایر ها است. در این حالت تعریف متحول نمونه (Instance Variable) که تعداد تایر ها را حفظ کند خوب نیست. بهتر اینست که در این حالت متحول کلاس (Class Variable) تعریف شود که تعداد تایر ها را حفظ میکند. این متحول میان تمام نمونه ها شریک است. اگر یک آبجکت قیمت آنرا تغییر دهد برای تمام نمونه های دیگر تغییر مینماید.

به طور خلاصه گفته میتوانیم که فایده آبجکت ها، مخفی ساختن معلومات (Information Hiding) و بخش بندی (Modularity) است. و فایده Class ها استفاده دوباره (Reusability) است.

وراثت (Inheritance)

وراثت یک شکل از استفاده دوباره است. به کمک وراثت میتوانیم از Data و میتودهای کلاس های جدید در کلاس های دیگر (نو تشکیل) استفاده نماییم. وراثت رابطه "Is-a" را تمثیل میدهد. در رابطه "Is-a" یک آبجکت کلاس فرعی (Sub-class) میتواند مانند آبجکت کلاس اصلی (Super-class) استفاده شود. مثلاً موتر یک عراده است. رابطه "Has-a" توسط Composition تمثیل میشود. میتودها و Data عمومی (Public) کلاس اصلی در هر کلاس دیگر و به شمول کلاسهای فرعی قابل استفاده است. این میتود ها و Data خصوصی هم توسط کلاس فرعی به ارث برده میشود. به کمک Protection میتوانیم میتودها و Data خصوصی کلاسهای اصلی را توسط کلاس فرعی مورد دسترسی قرار دهیم ولی در عین وقت توسط آبجکتهای کلاس دیگر استفاده شده نمیتواند.

آبجکتها جدید را از کلاس هاي موجود ايجاد کرده ميتوانيم. آبجکتهاي جديد تمام خصوصيات کلاس خود را دارا ميباشد. آبجکتها (Objects) و کلاس ها (Classes) داراي تسلسل يا سلسله مراتب بوده و هر آبجکت و هر کلاس، متحولين و ميتود ها را از کلاس که در سلسله مراتب قبل از آن قرار دارد (کلاس پدر) به ارث ميبرد. به عنوان مثال هر صفحه (Work Sheet) جديد در مايکروسافت اکسل خصوصيات کلاس خود (کلاس صفحه يا Work Sheet) را دارا ميباشد. و هر فورم جديد در ویژول بسیک داراي خصوصيات کلاس فورم را دارا ميباشد. ارث زمينه استفاده دوباره را مساعد نموده است. به کمک صفت وراثت، کلاسهاي جديد را از کلاسهاي موجود ساخته ميتوانيم.

به صورت عمومي آبجکتها از روي کلاس آنها تعريف ميشود. ما ميتوانيم راجع به آبجکت معلومات داشته باشيم اگر ما کلاس آنرا بفهميم. کلاس ها را ميتوانيم از روي ديگر کلاس ها نیز تعريف کنيم. در اصطلاح پروگرام نويسي شي گرا اينوع کلاس ها بنام کلاس هاي فرعي (Subclasses) ياد ميشود. و کلاس اصلي آن بنام کلاس اصلي (Superclass) ياد ميشود. هر کلاس فرعي خصوصيات وميتودها را از کلاس اصلي خود به ميراث ميبرد. البته خصوصيات و ميتود هاي کلاس هاي فرعي تنها محدود به کلاس اصلي آن نيست بلکه ميتواند خصوصيات و ميتود هاي ديگر نیز داشته باشد. وراثت (Inheritance) داراي فوائد ذيل ميباشد:

- با استفاده از وراثت پروگرامر ميتوانند کود که در کلاس اصلي است آنرا چندين بار استفاده نمايد.
- پروگرامر ميتواند کلاس اصلي را استفاده و تطبيق کند که خصوصيات مشترک را تعريف ميکند. اينوع کلاس ها را بنام کلاس هاي مجرد (Abstract Classes) ياد ميکند.

راه ارتباط (Interface)

به صورت عمومي انترفس ها عبارت از راه هاي است که از طريق آن وسائل غير وصل شده (وسائل خارجي) به سيستم راه پيدا ميکند و همراي سيستم کار ميکند. به اساس اين تعريف ريموت کنترول تلفزيون یک انترفس است که از طريق آن شما با تلفزيون رابطه برقرار ميکنيد. لسان هم یک انترفس است میان دوشخص. ميتود ها هم انترفس است بين کلاس ها و آبجکت هاي آن.

فصل دوم

لسان پروگرام نویسی ویژول بسیک دات نت

Visual BASIC.NET

لسان پروگرام نویسی ویژول بسیک از جمله لسانهای پروگرام نویسی شی گرا بوده و نسبت به استفاده وسیع آن از اهمیت زیاد برخوردار است. این لسان به شکل مستقل و همچنین در پروگرام های تطبیقی مانند مایکروسافت ورد (MS-Word)، مایکروسافت اکسل (MS-Excel)، مایکروسافت اکسس (MS-Access)، فایل های خطی (Script Files)، دیزاین صفحات وب، و غیره موارد استفاده میشود. ویژول بسیک مشهورترین لسان برای توسعه (Development) پروگرامها میباشد. اگر VB (Visual Basic)، VBA، VB Script (Visual Basic For Application) و CCE (Control Creation Edition) را در نظر بگیریم، در آنصورت بسیاری توسعه دهنده گان (Developers) از لسان ویژول بسیک استفاده میکنند.

نوت: CCE سافت ویر Development و مربوط VB است که توسط آن کنترولهای ActiveX ایجاد شده میتواند. کنترولهای ActiveX برای استفاده در دیزاین وب خیلی مفید است. کنترولهای ActiveX قابلیت استفاده دوباره را دارد و توسط Developer ها استفاده میشود.

نمونه اول ویژول بسیک در سال 1991 توسط شرکت مایکروسافت معرفی گردید. از همان تاریخ تا به امروز در ویژول بسیک تغییرات زیاد به میان آمده و نمونه های جدید به آن معرفی گردیده است. در سال 1998 نمونه ویژول بسیک 6 توسط شرکت مایکروسافت معرفی گردید که باعث استفاده وسیع ویژول بسیک شد. ویژول بسیک دات نت (Visual Basic.NET) که بعد از نمونه ششم ویژول بسیک معرفی گردیده کاملاً شی گرا بوده و برای پروگرام نویسی صفحات وب استفاده شده میتواند. هر نمونه جدید ویژول بسیک دارای خصوصیات و توانایی های بیشتر میباشد. با استفاده از محیط و ابزار ویژول بسیک، شما به سرعت و آسانی پروگرامهای تطبیقی را تهیه کرد ه میتوانید. محیط کاری ویژول بسیک که بنام IDE (Integrated Development Environment) یاد میشود سهل الاستفاده بوده و دارای ویندوزها، مینوها و ابزار (Tools) متنوع میباشد. لسان ویژول بسیک می تواند کار پرزحمت و خسته کننده پروگرام نویسی را برای شما دلچسپ ساخته و آنرا به یک فن تفریحی برای تان تبدیل نماید به شرط آنکه از لحاظ تیوری به اساسات این لسان آشنای داشته باشید. ویژول بسیک دات نت نمونه جدید ویژول بسیک بوده و به اساس ستراتیژی دات نت مایکروسافت کار میکند.

در ویژول بسیک دات نت میتوانیم کتابخانههای کلاس (Class Library) و سروسهای وب XML، و پروگرامهای که میتواند در Web Browser، تلفنونهای موبایل، و PDA استفاده شود تهیه نماییم. همچنین به کمک ویژول بسیک دات نت میتوانید add-ins را برای پروگرامهای مایکروسافت تهیه نمایید.

چوکات دات نت (.NET Framework) یک طبقه میان لسان و سیستم عامل کمپیوتر است. در حقیقت چوکات دات نت جایگزین Win32 میباشد.

محیط کاری ویژول بسیک دات نت

برای کار در ویژول بسیک دات نت یک پروژه (Project) را باز کنید. پروژه عبارت از مجموعه فایل های یک پروگرام میباشد. هر پروگرام میتواند دارای چندین فایل باشد. بعضی از این فایل ها حاوی کد (Code) پروگرام، و برخی دیگر آن حاوی اشکال ظاهری پروگرام میباشد.

ویژول بسیک دات نت بعد از اجرای هر پروگرام برای همان پروگرام چند فولدر را ایجاد مینماید:

- یک فولدر بنام bin که در آن پروگرام اجرا شده و اجزای آن را حفظ مینماید.
- فولدر obj که منابع که به Application تخصیص گردیده آنرا حفظ مینماید.
- فولدر Myproject که فایلهاي عيارسازي را حفظ میکند.

در پروژه کار بالاي فورم (Form) میباشد. در یک پروژه چندین فورم می تواند وجود داشته باشد. و فورم یک آبجکت است. پس فورم دارای خصوصیات (Properties) است. فورم دارای میتودها (Methods) است. و فورم دارای حادثات (Events) است. پروژه و فورمهاي مربوط همان پروژه باید در دیسک ذخیره (Save) شود. در ویژول بسیک دات نت فورم را بنام فورم ویندوز (Windows Form) یاد میکند.

در وسط صفحه فورم است. در قسمت بالاي صفحه مینو ها (Menus) و ابزار (Tools) قرار دارد. به طرف چپ صفحه کنترلها و یا آبجکت هاي اند که در اثنای کار در پروژه از آن استفاده میشود. هر کدام از این آبجکت ها دارای خصوصیات، میتودها و حادثات است. در طرف راست صفحه، خصوصیات (Properties) آبجکت انتخاب شده نشان داده شده است. و شما میتوانید با تعیین قیمت برای هر یک از این خصوصیات نمایش آبجکت را تغییر دهید.

ارتباط گرافیکی استفاده کننده

Graphical User Interface (GUI)

GUI عبارت از راه گرافیکی ارتباط است که به کمک آن استفاده کننده کمپیوتر به شکل گرافیکی و مرئی با پروگرام ها ارتباط برقرار میکند. GUI دارای اجزای مختلف از قبیل مینوها (Menus)، دکمه ها (Buttons)، باکس هاي متن (Text Boxes)، لیبل ها (Labels)، اسکول بارها (Scrolls Bars)، دکمه هاي اختیاری (Option Buttons)، کامبو باکس (Combo Boxes)، لیست باکس ها (List Boxes) و غیره میباشد. اجزای GUI در حقیقت آبجکتهای است که از طریق آن و به کمک موس یا صفحه کلید استفاده کننده همراي پروگرام ارتباط برقرار میکند. این آبجکتها را بنام کنترل ها هم یاد میکند. برای ایجاد کردن GUI از فورم (Form) استفاده میشود. فورم هم یک کنترل یا آبجکت است که کنترلها و آبجکتهای دیگر بالاي آن قرار داده میشود. در ویژول بسیک دات نت ده ها آبجکت وجود دارد و پروگرامر هم میتواند آبجکتهای جدید را ایجاد نماید. در این قسمت جهت آشنائی با محیط کاری ویژول بسیک دات نت بعضی از آبجکتهای که استفاده بیشتر دارد مورد بحث قرار میدهم.

آبجکتهای ویژول بسیک

فورم (Form)

فورم یک آبجکت مهم ویژول بسیک است. در اثنای کار در پروژه هاي ویژول بسیک فورم مانند زمین برای آبجکت هاي دیگری که در پروژه استفاده میشود، میباشد. آبجکت ها بالاي فورم قرار داده میشود.

طوریکه قبلا نیز یادآور شدیم که فورم یک آبجکت است، پس فورم دارای خصوصیات (Properties)، میتودها (Methods)، و حادثات (Events) است. در این قسمت جهت آشنائی با محیط کاری ویژول بسیک دات نت بعضی خصوصیات (Properties)، میتودها (Methods)، و حادثات (Events) فورم را با چند مثال مطالعه می نمائیم:

خصوصیات فورم: (Form Properties)

خصوصیات طرز نمایش آبجکتها را تعیین میکند. فورم دارای خصوصیات بوده که طرز نمایش فورم را تعیین میکند. خصوصیات فورم قیمت هاي مختلف را به خود گرفته میتوانند. در اثر این قیمت هاي

متفاوت نمایش فورم تغییر میکند. قیمت خصوصیات فورم را هم از حالت دیزاین فورم و هم از کود تغییر داده میتوانیم. قابل تذکر است که قیمت بعضی خصوصیات را تنها از کود تغییر داده میتوانیم.

خصوصیات مهم فورم قرار ذیل است:

خاصیت Name: توسط خاصیت Name، نام برای فورم تعیین کرده میتوانیم. به کمک همین نام فورم در ویژول بسیک شناخته میشود. از نام فورم بیشتر در کود ویژول بسیک استفاده میشود. مثلاً اگر خواسته باشیم رنگ زمین یک فورم را تغییر دهیم، از نام فورم باید استفاده نمائیم:

```
Form1.BackColor = Color.Red
```

دستور فوق رنگ زمین (BackColor) برای Form1 را به رنگ سرخ تبدیل میکند. در این دستور خود فورم که بنام Form1 است آبجکت است، BackColor خاصیت فورم است و Color.Red قیمت است که به این خاصیت (متحول) داده شده است. اگر خواسته باشیم تا نام فورم را تغییر دهیم، به خاصیت Name آن قیمت میدهیم. مثلاً میخواهیم نام فورم را به frmCSF تبدیل نمائیم، پس به خاصیت Name فورم قیمت frmCSF میدهیم.

```
Form1.Name = "frmCSF"
```

دستور فوق نام فورم را از Form1 به frmCSF تبدیل میکند.

نوت: در ویژول بسیک دات نت اگر از داخل خود فورم قیمت یک خاصیت را تغییر میدهیم باید از Alias فورم (Me) استفاده کنیم. برای تغییر دادن رنگ آن باید طور ذیل تایپ نمائیم:

```
Me.Name = "frmCSF"
```

خاصیت Backcolor: این خاصیت رنگ زمین فورم را تعیین میکند.

```
Form1.BackColor = Color.Red
```

خاصیت BackgroundImage: توسط این خاصیت برای صفحه فورم یک تصویر را انتخاب کرده میتوانیم.

خاصیت Forecolor: این خاصیت رنگ متن فورم را تعیین میکند.

```
Form1.forecolor = Color.Blue
```

خاصیت ControlBox: این خاصیت دارای قیمت درست (True) و نادرست (False) میباشد. اگر قیمت آن درست باشد کنترولهایی که جهت بسته نمودن فورم (Close)، Restore کردن فورم و خورد ساختن فورم (Minimize) است نشان داده میشود. و اگر قیمت آن نادرست (False) باشد این کنترولها نشان داده نمیشود.

خاصیت Cursor: این خاصیت شکل موس را بالای آبجکت تعیین میکند.

خاصیت FormBorderStyle: این خاصیت طرز نمایش چوکات فورم را تعیین میکند.

خاصیت MinimizeBox: این خاصیت میتواند قیمت درست و نادرست داشته باشد. اگر قیمت آن درست باشد فورم را بعد از اجرای پروگرام هم میتوانیم خورد نمائیم. و اگر قیمت آن نادرست باشد در آنصورت فورم را بعد از اجرای پروگرام خورد ساخته نمیتوانیم.

خاصیت Text: این خاصیت عنوان فورم را تعیین میکند. قیمت این خاصیت، که معمولاً یک متن میباشد، عنوان فورم میشود.

خاصیت MaximizeBox: این خاصیت نیز میتواند قیمت درست و نادرست داشته باشد. اگر قیمت آن درست باشد فورم را بعد از اجرای پروگرام هم میتوانیم بزرگ نمائیم. و اگر قیمت آن نادرست باشد در آنصورت فورم را بعد از اجرای پروگرام بزرگ ساخته نمیتوانیم.

خاصیت Icon: توسط این خاصیت برای فورم Icon را تعیین کرده میتوانیم.

خاصیت TopMost: اگر قیمت این خاصیت درست باشد فورم هر وقت بالای آبجکتهای دیگر قرار میگیرد.

خاصیت ShowIcon: این خاصیت میتواند دارای قیمت درست و یا نادرست باشد. در صورتیکه قیمت آن درست باشد Icon فورم نشان داده میشود و اگر نادرست باشد Icon فورم نشان داده نمیشود.

خاصیت RightToLeft: این خاصیت هم میتواند قیمت True و یا False را داشته باشد. اگر قیمت آن True باشد در صفحه فورم از راست به چپ نوشته میشود (برای متن عربی، دری و یا پشتو)، و اگر قیمت آن False باشد از چپ به طرف راست نوشته میگردد (برای متن انگلیسی).

`Form1.RightToLeft = True`

خاصیت ShowInTaskbar: اگر قیمت این خاصیت درست باشد، فورم در اثنای اجرا در Taskbar ویندوز نشان داده میشود. و اگر نادرست باشد نشان داده نمیشود.

خاصیت StartPosition: توسط این خاصیت محل فورم را در اثنای اجرای پروگرام تعیین کرده میتوانیم.

خاصیت AcceptButton: این خاصیت دکمه که در اثر فشار دادن کلید Enter صفحه کلید اجرا شود را تعیین میکند.

خاصیت AutoScroll: قیمت منطقی (Boolean) که نشان دادن و یا عدم نشان دادن ScrollBars را تعیین میکند.

خاصیت CancelButton: این خاصیت دکمه که در اثر فشار دادن کلید Enter صفحه کلید اجرا شود را تعیین میکند.
مثال:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Me.BackColor = Color.Coral
        Me.Text = "Kabul University"
        Me.ControlBox = False
        Me.ForeColor = Color.BlueViolet
        Me.Cursor = Cursors.Hand
        Me.AutoScroll = True
    End Sub
End Class
```

میتود های فورم (Form Methods)

فورم داراي میتود ها است. میتود ها عبارت از پروگرام هاي فرعي از قبل تهیه شده است که در داخل کلاس تعریف گردیده و آبجکت هاي همان کلاس آنرا استفاده میکند. میتودها در حقیقت کار هاي را تعیین میکنند که یک آبجکت آنرا انجام داده میتواند.

بعضي میتود هاي مهم فورم قرار ذیل میباشد:

میتود Show: این میتود فورم را در صفحه نمایش نشان میدهد. به پروگرام فرعي ذیل توجه کنید:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles Button1.Click
        Form2.Show()
    End Sub
End Class
```

میتود Hide: این میتود فورم را از روی صفحه نمایش مخفی می سازد. به مثال ذیل توجه نمائید:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles Button1.Click
        Form2.hide()
    End Sub
End Class
```

میتود Load: این میتود با دو بار فشار دادن دکمه موس بالاي فورم پروگرام شده میتواند. پروگرام ذیل میتود Load را نشان میدهد:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
    Me.ControlBox = False
End Sub
```

میتود Close: این میتود فورم را بسته میکند. پروگرام ذیل میتود Load را نشان میدهد:

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles MyBase.Load
        Me.Close()
    End Sub
End Class
```

حادثات فورم (Form Events)

در محیط هاي شي گرا، براي نوشتن پروگرام، روش هاي متعدد وجود دارد. يكي از اين روشها استفاده از حادثات (Events) است. حادثه چه است؟ حادثه يا رویداد عبارت از یک عمل میباشد که بالاي یک شي يعني آبجکت اجرا میشود. اگر دکمه موس را بالاي دکمه امر (Command Button) فشار دهيم حادثه فشار دادن دکمه موس (Click) رخ میدهد. همراي هر حادثه میتوانيم دستور هاي مشخص را اجرا نمائيم. از اينکه فورم یک آبجکت است پس فورم داراي حادثات (Events) است.

حادثات عبارت از کارهاي است که بالاي فورم انجام شده ميتواند. اکثر حادثات فورم توسط موس و يا صفحه کلید انجام ميشود. در اينجا بعضي حادثات را با مثال ميبينيم:

حادثه Click: اين حادثه وقتي رخ ميدهد که بالای فورم یکبار توسط دکمه موس فشار وارد شود. به مثال ذیل توجه کنید:

```
Public Class Form1
Private Sub Form1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Click
    Me.BackColor = Color.FromArgb(Rnd() * 255, Rnd() * 255, Rnd() * 255)
End Sub
End Class
```

حادثه Dbclick: اين حادثه وقتي رخ ميدهد که بالای فورم دوبار توسط دکمه موس فشار وارد شود.

```
Private Sub Form1_DoubleClick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.DoubleClick
    Me.Text = "You clicked twice"
    MessageBox.Show("Please click on Ok")
End Sub
```

علاوه بر فورم آجکت های دیگر نیز در ویژول بسیک وجود دارد. اين آجکت ها بنام کنترلها نیز یاد ميشوند. بعضی از اين آجکت ها قرار ذیل ميباشد:

عنوان يا نشاني (Label): اين کنترل جهت نشان دادن متن در روی فورم به کار برده ميشود. معمولاً عناوين در روی فورم توسط همين کنترل نشان داده ميشود. کنترل Label نیز یک شي يعني آجکت بوده فلذا داراي خواص است. بعضي خواص آن مانند فورم بوده و قسمتي ديگر از خواص (Properties) آن قرار ذیل است:

- خاصيت Text: عنوان يا محتوي Label را حفظ ميكند.
- خاصيت Font: فونت Label را تعين ميكند.
- خاصيت BorderStyle: نمايش چوكات Label را تعين ميكند.
- خاصيت Location: محل قرار گرفتن Label بالاي فورم را تعين ميكند.
- خاصيت TextAlign: متن Label را تعين ميكند. متن ميتواند از طرف راست Align باشد و يا چپ و يا هم در وسط يك کنترل قرار داشته باشد.
- خاصيت Visible: اين خاصيت ميتواند قيمت درست (True) و يا نادرست (False) باشد. اگر قيمت آن درست باشد Label در اثنای اجرای پروگرام در بالاي فورم نشان داده ميشود. و اگر نادرست باشد نشان داده نميشود.

باکس متن (Text box): اين کنترل ميتواند جهت نشان دادن متن و همچنين اجرای فورمول ها به کار برده شود. متنی که توسط اين کنترل نشان داده ميشود، ميتواند توسط استفاده کننده تغيير داده شود. جهت اجرای محاسبات هم می توانيم از همين کنترل استفاده كنيم.

دکمه امر (Button): از اين کنترل در روی فورم استفاده وسيع صورت ميگيرد. معمولاً جهت اجرای Code ها از همين کنترل استفاده ميگردد.

استفاده از کنترلها

اکنون میخواهیم طرز استفاده این کنترلها را با ارائه مثالهایی آن مورد بحث قرار دهیم. چون هر کنترل یک آبجکت است پس هر یک از کنترل های فوق دارای خصوصیات (Properties)، میتود ها (Methods)، و حادثات (Events) است.

مثال 1: پروگرام فرعی ذیل مجموع اعداد که در TextBox1، TextBox2، و TextBox3 است در TextBox4 نشان میدهد:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    TextBox4.Text = Val(TextBox1.Text) + Val(TextBox2.Text) +
    Val(TextBox3.Text)
End Sub
```

مثال 2: پروگرام فرعی ذیل اوسط اعداد که در TextBox1، TextBox2، و TextBox3 است در TextBox4 نشان میدهد:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    TextBox5.Text = (Val(TextBox1.Text) + Val(TextBox2.Text) +
    Val(TextBox3.Text)) / 3
End Sub
```

مثال 3: پروگرام فرعی ذیل جذر m عدد که در TextBox2، است یافته و آنرا در TextBox3 نشان میدهد:

```
Private Sub btntotal_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btntotal.Click
    Dim m As Integer
    m = Val(TextBox1.Text)
    TextBox3.Text = Val(TextBox2.Text) ^ (1 / m)
End Sub
```

پروگرام نویسی در ویژول بسیک دات نت

طوری که میدانیم که از لسانهای پروگرام نویسی برای نوشتن پروگرامها استفاده میشود. هر پروگرام که شما در کمپیوتر تان میبینید و یا از آن استفاده میکنید به کمک لسانهای پروگرام نویسی تهیه میشود. بدون وجود لسانهای پروگرام نویسی پروگرامهای کمپیوتری را تهیه کرده نمیتوانیم و در اثنای کار با کمپیوتر باید از لسان خود کمپیوتر که عبارت از صفر و یک است استفاده نمائیم.

هر لسان پروگرام نویسی دارای کلمات کلیدی (Key-words)، دستورها، و همچنین قواعد املایی (Syntax Rules) برای نوشتن پروگرامها است. جهت استفاده از یک لسان برای ساختن پروگرامها، آموختن دستورها و قواعد املایی آن ضروری میباشد. خوشبختانه که منطق ساختن پروگرامها در لسانهای مختلف پروگرام نویسی یکسان بوده و فهم یک لسان در آموختن لسان دیگر کمک میکند. به نظر من اگر ما یک لسان پروگرام نویسی را بهتر یاد بگیریم بعد از آن خودما میتوانیم لسانهای دیگر پروگرام نویسی را بیاموزیم.

در اینجا ما با فرض اینکه محصلین به یک لسان پروگرام نویسی از قبل آشنائی دارد (مانند جاوا)، به شکل مختصر قواعد لسان پروگرام نویسی ویژول بسیک دات نت را مورد بحث قرار میدهیم.

متحول (Variable)

قسمتي از حافظه کمپیوتر است که به یک نام مسمی گردیده (نام متحول) و قیمت ها را در اثنای اجرای پروگرام ذخیره میکند، عبارت از متحول میباشد. برای اینکه یک متحول را استفاده کرده بتوانیم اول باید آنرا تعریف کنیم. در ویژول بسیک متحول را توسط دستور Dim تعریف کرده میتوانیم. شکل عمومی دستور Dim قرار ذیل است:

Dim نام متحول As نوع متحول

نام متحول در ویژول بسیک میتواند از یک تا 255 حرف باشد. کلمات کلیدی (Key Words) ویژول بسیک را نمی توانیم به حیث نام متحول استفاده نماییم.

نوع متحول میتواند عددی، حرفی، منطقی، تاریخ و یا تعریف شده توسط استفاده کننده باشد. در حقیقت نوع متحول قیمت های را که متحول گرفته میتواند و طرز ذخیره آنرا تعیین میکند. متحول عددی در ویژول بسیک به پنج نوع ذیل میباشد:

Byte: این نوع متحول قیمت های بدون علامه از صفر تا 255 را گرفته و یک بایت از حافظه کمپیوتر را اشغال میکند.

Integer: این نوع متحول چهار بایت از حافظه کمپیوتر را اشغال میکند.

Long: این نوع متحول هشت بایت از حافظه کمپیوتر را اشغال میکند.

Single: این نوع متحول نیز چهار بایت از حافظه کمپیوتر را اشغال میکند.

Double: این نوع متحول نیز هشت بایت از حافظه کمپیوتر را اشغال میکند.

علاوه بر انواع عددی Data یک متحول میتواند یکی از انواع ذیل نیز باشد:

متحول حرفی: برای تعریف متحول حرفی، در قسمت نوع متحول کلمه کلیدی String را مینویسیم. به مثال ذیل توجه کنید:

Dim stname As string

متحول حرفی به تعداد حروف سلسله جمع 10 بایت دیگر حافظه را اشغال میکند. مثلا اگر سلسله 15 حرفی را ذخیره مینمائیم 25 بایت حافظه گرفته میشود.

متحول منطقی یا بولی (Boolean): این نوع متحول قیمت های درست (True) و یا نادرست (False) را ذخیره میکند. این نوع متحول قیمت درست را توسط صفر و قیمت نادرست را توسط 1- نشان میدهد. ویژول بسیک دو بایت (البته این نقطه مربوط به سیستم عامل هم است و از یک سیستم عامل تا سیستم عامل دیگر فرق میکند) را برای متحول منطقی در نظر میگیرد. متحول منطقی را طور ذیل تعریف کرده میتوانیم:

DIM نام متحول AS BOOLEAN

Example:

DIM pass AS BOOLEAN

نوت: همراي متحول منطقي از عملگرهاي منطقي مانند NOT، OR و AND هم استفاده کرده میتوانیم. مثلا عملگر NOT قیمت متحول منطقی را به قیمت عکس آن تغییر میدهد:

Pass = NOT pass

در حقیقت دستور فوق جای چندیدن دستور ذیل را میگیرد:

IF pass = TRUE THEN

Pass = FALSE

ELSE

Pass = TRUE

END IF

متحول Date : اینوع متحول قیمت های که به شکل تاریخ و وقت باشد گرفته میتواند. تاریخ و وقت در داخل کمپیوتر به یک فارمت خاص که بنام Double-precision یاد میشود، ذخیره میگردد. قسمت تام این عدد تاریخ و قسمت اعشاری آن وقت را تمثیل میکند. متحول Date هشت بایت حافظه را اشغال میکند. این متحول میتواند قیمت های از 00:00:00 اول جنوری سال 1 تا به 11:59:59 ماه دسامبر سال 9999 را ذخیره نماید.

متحول Date را طور ذیل تعریف کرده میتوانیم:

AS DATE نام متحول DIM

Example:

```
Public Class Form1
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
Dim sdate As Date
```

```
sdate = Date.Now
```

```
TextBox1.Text = sdate.Date
```

```
TextBox2.Text = sdate.DayOfWeek
```

```
TextBox3.Text = sdate.DayOfYear
```

```
TextBox4.Text = sdate.Kind
```

```
TextBox5.Text = sdate.Month
```

```
TextBox6.Text = sdate.Hour
```

```
TextBox7.Text = sdate.Year
```

```
TextBox8.Text = sdate.Ticks
```

```
End Sub
```

```
End Class
```

متحول آبجکت:

اینوع متحول میتواند آبجکت های VB را ذخیره نماید. مقدار حافظه که توسط متحول آبجکت گرفته میشود نظر به سیستم عامل فرق میکند. در سیستم عامل 32 بیتی چهار بایت و در سیستم عامل 64 بیتی 8 بایت حافظه را اشغال میکند. این نوع متحول ریفرنس هر نوع (مانند String، Array، Class و غیره) را ذخیره کرده میتواند. وقتیکه نوع Data در اثنای اجرای پروگرام برای ما معلوم نباشد از این نوع متحول استفاده میکنیم.

```
Dim myObject As Object
```

```
Dim datTyp As Integer
```

```
Dim v As Object
```

```
v = "17"
```

```
v = v - 15
```

```
v = "H" & v
```

جدول ذیل انواع معلومات اولیه و مقدار حافظه مورد ضرورت برای هر نوع را نشان میدهد:

Type	Size (in bit)
Boolean	16
Char	16
Byte	8

Date	64
Decimal	128
Short	16
Integer	32
Long	64
Single	32
Double	64
Object	32

ثابت (Constant)

ثابت عبارت از نوع Data است که قیمت آن در اثنای اجرای پروگرام تغیر نمیکنند. در ویژول بسیک ثابت را طور ذیل تعریف کرده میتوانیم:

```
Const pi AS DOUBLE = 3.1415926
```

ثابت نظر به متحول دارای خوبی های ذیل میباشد:

- 1- قیمت ثابت بعد از تعریف آن تغیر نمیکنند.
- 2- با تغیر دادن قیمت آن در تمام قسمت های پروگرام، قیمت آن تغیر گردیده و اینکار باعث سهولت در تغیر قیمت ها میشود.
- 3- ثابت نظر به متحول به سرعت و آسانی توسط کمپیوتر پراسس میشود.

امتحان نمودن نوع متحول : برای معلوم نمودن نوع متحول از دو تابع ذیل استفاده کرده می توانیم:

VarType(): جواب عددی که همان عدد نظربه نوع متحول فرق میکند.

TypeName(): جواب string که همان string نوع متحول را نشان میدهد.

Vbinteger	2
Vblong	3
Vbsingle	4
Vbdate	7
Vbstring	8

برای تبدیل کردن **Text** به **Value** از تابع **Parse** هم استفاده کرده میتوانیم:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    TextBox3.Text = Integer.Parse(TextBox1.Text) +
Integer.Parse(TextBox2.Text)
End Sub
End Class
```

و یا هم میتوانیم از تابع **CType** استفاده کنیم:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    TextBox3.Text = CType(TextBox1.Text, Integer) +
CType(TextBox2.Text, Integer)
End Sub
```

End Class

دستور های شرطی (Conditional Statements)

برای اجرای مشروط یک یا چند سطر پروگرام، از دستور های ذیل استفاده میشود:

```
IF .....THEN
IF.....THEN.....ELSE
SELECT.....CASE
```

شکل عمومی IF....THEN قرار ذیل است:

```
IF Condition THEN Statement
```

همچنین این دستور را طور ذیل نیز استفاده کرده میتوانیم:

```
IF Condition THEN
    Statement(s)
END IF
```

شکل عمومی IF....THEN.....ELSE قرار ذیل است:

```
IF Condition-1 THEN
    Statement(s)
ELSEIF Condition-2 THEN
    Statement(s)
```

```
.....
.....
```

```
ELSE
    Statement(s)
END IF
```

مثالهای دستور IF ... THEN ELSE

مثال 1: پروگرام فرعی ذیل زکات پول نقد را محاسبه میکند:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
    Dim money, zakat As Integer
    money = InputBox("Enter the amount of money: ")
    If money < 200 Then
        MessageBox.Show("No Zakat")
    Else
        zakat = (money - 200) \ 40 + 5
        MessageBox.Show("Zakat is: " & zakat)
    End If
```

```
End Sub
```

مثال 2: پروگرام فرعی ذیل مالیه معاشات را محاسبه میکند:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
    If Val(TextBox1.Text) < 12000 Then
        TextBox2.Text = "No Tax"
```



```

ElseIf Val(TextBox1.Text) < 20000 Then
    TextBox2.Text = Val(TextBox1.Text) * 10 / 100
ElseIf Val(TextBox1.Text) < 30000 Then
    TextBox2.Text = Val(TextBox1.Text) * 20 / 100
Else
    TextBox2.Text = Val(TextBox1.Text) * 30 / 100
End If
End Sub

```

مثال 3: پروگرام فرعی ذیل درجه محصل را نظر به اوسط نمرات وی معلوم میکند:

```

Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    avg.Text = (Val(t1.Text) + Val(t2.Text)) / 2
    If Val(avg.Text) >= 90 Then
        grade.Text = "Excellent"
        grade.BackColor = Color.Green
    ElseIf Val(avg.Text) >= 80 Then
        grade.Text = "Very Good"
        grade.BackColor = Color.Blue
    ElseIf Val(avg.Text) >= 70 Then
        grade.Text = "Good"
        grade.BackColor = Color.Cyan
    ElseIf Val(avg.Text) >= 60 Then
        grade.Text = "Pass"
        grade.BackColor = Color.Yellow
    Else
        grade.Text = "Fail"
        grade.BackColor = Color.Red
    End If
    T1.Enabled = False
    t2.Enabled = False
End Sub

```

مثال 4: پروگرام فرعی ذیل جذر معادله $x+3x^2-10=0$ را محاسبه مینماید:

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    TextBox1.Text = "X+3x^2-10=0"
    Dim x As Single
    x = 0
again:
    If (x + 3 * x ^ 2 - 10) - 0 >= 0.00001 Then
        MessageBox.Show("The value of x is: " & x)
        GoTo endprog

```

```

Else
    x = x + 0.00001
    GoTo again
End If
endprog:
End Sub

```

مثال 5: پروگرام فرعی ذیل جذرهای معادله درجه دوم را محاسبه میکند:

```

' x1, x2 = (-b+|(b^2-4ac)^0.5)/2a & ' x2 = (-b- (b^2-4ac)^0.5)/2a
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim a, b, c, r, x1, x2 As Single
    a = TextBox1.Text
    b = TextBox2.Text
    c = TextBox3.Text
    r = (b ^ 2 - 4 * a * c)
    If r < 0 Then
        GoTo halt
    Else
        r = r ^ 0.5
    End If
    x1 = (-b + r) / (2 * a)
    x2 = (-b - r) / (2 * a)
    MessageBox.Show("The value of x1 & x2 are: " & x1 & ", " & x2)
    GoTo endprog
halt:
    MessageBox.Show("No root for this equation! ")
endprog:
End Sub

```

مثال 6: پروگرام فرعی ذیل پاسورد را ارزیابی مینماید:

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim c As Byte
    c = 1
    Dim prog As Object
abc:
    c = c + 1
    prog = InputBox("Enter you Password")
    If prog = "ab@_6s" Then
        MessageBox.Show("Pssword accepted")
    Else
        MessageBox.Show("Wrong Password")
    If c < 4 Then

```

```

    GoTo abc
  Else
    Me.Enabled = False
  End If
End If
End Sub

```

مثال 7: پروگرام فرعی ذیل نیز پاسورد را ارزیابی مینماید:

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
  If pass.Text = "abc123" Then
    T1.Enabled = True
    t2.Enabled = True
  Else
    MessageBox.Show("پاسورد شما درست نیست")
  End If
End Sub

```

مثال 8: پروگرام فرعی ذیل پیغام ("Good marning") و یا ("Good Afternoon") را نظر به وقت روز میدهد:

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
  score.Text = DateAndTime.TimeOfDay
  If Hour(DateAndTime.TimeOfDay) < 12 Then
    MessageBox.Show("Good marning")
  Else
    MessageBox.Show("Good Afternoon")
  End If
End Sub

```

مثال 9: پروگرام فرعی ذیل نیز پاسورد را ارزیابی میکند:

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
  Dim pass As String
  TextBox1.PasswordChar = "*"
  pass = TextBox1.Text
  If pass = passstore Then
    Me.Close()
  Else
    MessageBox.Show("پاسورد شما درست نیست")
  End If
End Sub

```

دستور :SELECT CASE

وظیفه دستور SELECT CASE مشابه به دستور IF ... THEN ... ELSE میباشد و برای تست نمودن چند شرط استفاده میشود. در صورت که تعداد حالات زیاد باشد بهتر است از دستور SELECT ... CASE استفاده شود.
شکل عمومی دستور SELECT.....CASE قرار ذیل است:

```
SELECT CASE Expression
CASE Value-1
      Statement(s)
CASE Value-2
      Statement(s)
.....
.....
CASE ELSE
      Statement(s)
END SELECT
Example:
SELECT CASE Weekday(date)
CASE 1
      DayName = "Monday"
CASE 2
      DayName = "Saturday"
.....
.....
END SELECT
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
  Select Case TextBox1.Text
    Case 1 To 299
      TextBox2.Text = TextBox1.Text * 1.5
    Case 300 To 599
      TextBox2.Text = TextBox1.Text * 3
    Case Else
      TextBox2.Text = TextBox1.Text * 4
  End Select
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
  Select Case TextBox1.Text
    Case 1 To 299
      TextBox2.Text = TextBox1.Text * 1.5
    Case 300 To 599
      TextBox2.Text = 299 * 1.5 + (TextBox1.Text - 299) * 3
    Case Else
```

```

    TextBox2.Text = 299 * 1.5 + 599 * 3 + (TextBox1.Text - 599) * 4
End Select
End Sub

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

```

```

    Dim mydate As Date
    mydate = TextBox1.Text
    Dim month As Byte
    month = mydate.Month
    Select Case month
        Case 1, 3, 5, 7, 8, 10, 12
            MessageBox.Show("این ماه 31 روز است")
        Case 4, 6, 9, 11
            MessageBox.Show("این ماه 30 روز است")
        Case 2
            If mydate.Year Mod 4 = 0 Then
                MessageBox.Show("این ماه 29 روز است")
            Else
                MessageBox.Show("این ماه 28 روز است")
            End If
    End Select
End Sub

```

دستورهای تکرار (Loop Statements)

از دستور های تکرار جهت اجرای تکراری یک یا چند سطر پروگرام استفاده می شود. در ویژول بسیک دستور های ذیل به این منظور استفاده میشود:

```

FOR.....NEXT
DO.....LOOP

```

حلقه ForNext: شکل عمومی آن قرار ذیل است:

```

For variable_name = Initial_value To final_value [Step i]
    Statement(s)
Next

```

حلقه تکراری DO....LOOP

این حلقه دارای دو شکل عمومی ذیل میباشد:

```

DO WHILE Condition
    Statement(s)
LOOP

```

```

DO UNTIL Condition
    Statement(s)
LOOP

```

نوت: اگرخواستہ باشیم تا قبل از تست کردن شرط، جملات داخل حلقه اجرا شود، حلقه DO...LOOP را طور ذیل می نویسیم:

```
DO
    Statement(s)
LOOP WHILE Condition
DO
    Statement(s)
LOOP UNTIL Condition
```

پروگرام ذیل فکتوریل یک عدد را محاسبه میکند:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim number As Integer
    Dim factorial As Double
    number = InputBox("Enter a number: ")
    factorial = 1
    For i = 1 To number
        factorial = factorial * i
    Next
    MessageBox.Show("Factorial of " & number & " is:" & vbCrLf & factorial)
End Sub
```

نوت: بعضی اوقات ضرورت می افتد تا در صورت تکمیل شدن یک شرط قبل از تکمیل حلقه For ...Next ویا Do ... While خارج شویم. در اینصورت از دستور Exit استفاده کرده میتوانیم.

حلقه For EachNext:

این حلقه برای دوران در عناصر غیر عددی استفاده میشود.

For Each/Next Repetition Structure

```
For each grade in gradeArray
    If grade < lowgrade then
        Lowgrade = grade
    End if
Next
```

```
Dim abc As TextBox
    For Each abc In Me.Controls
        abc.BackColor = Color.Blue
    Next abc
End Sub
```

عملگر منطقی ANDALSO

این عملگر مانند عملگر AND بوده و تنها فرق این عملگر با عملگر ANDALSO اینست که عملگر ANDALSO بعضی اوقات تنها به تست کردن یکی از شرط ها اکتفا نموده و نتیجه را پیدا میکند. این در صورت است که تست کردن یک شرط منجر به نتیجه نهایی شود. از همین سبب این عملگر را بنام عملگر منطقی Short-circuiting نیز یاد میکند. شکل عمومی آن قرار ذیل میباشد:

Result= expression-1 ANDALSO expression-2

متحول result نتیجه را ذخیره میکند و از نوع Boolean میباشد.

تابع Inputbox

این تابع پیغام معین (تعیین شده از قبل) را به فارمت یک باکس مکالماتی (Dialog Box) در روی صفحه نمایش نشان داده و برای ورودی اطلاعات توسط user انتظار میکشد. اطلاعات داخل شده میتواند قیمت متحول باشد.

شکل عمومی این تابع در ویژول بسیک قرار ذیل است:

```
Public Function InputBox ( ByVal Prompt As String, _
    Optional ByVal Title As String = "", _
    Optional ByVal DefaultResponse As String = "", _
    Optional ByVal Xpos As Integer = -1, _
    Optional ByVal YPos As Integer = -1 ) As String
```

Prompt: این پارامتر اجباری بوده و عبارت از جمله است که در باکس نشان داده میشود. طول این جمله میتواند تا 1024 حرف باشد. برای نوشتن چندین جمله از توابع chr(10) و chr(13) نیز استفاده کرده میتوانیم.

Title: این پارامتر اختیاری بوده و عبارت از جمله میباشد که در قسمت عنوان (Title Bar) باکس نشان داده میشود.

DefaultResponse: این پارامتر نیز اختیاری بوده و قیمت از قبل تعیین شده (Default) را تعیین میکند.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
    Dim sum, number, numcount As Integer
```

again:

```
    number = InputBox("Enter a number: ")
```

```
    numcount = numcount + 1
```

```
    sum = sum + number
```

```
    If numcount < 5 Then GoTo again
```

```
    TextBox1.Text = sum
```

```
End Sub
```

مثال:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
    Dim message, title, defaultValue As String
```

```
    Dim myValue As Object
```

```

message = "Enter a value between 1 and 3"
title = "InputBox Demo"
defaultValue = "1"
myValue = InputBox(message, title, defaultValue)
' If user has clicked Cancel, set myValue to defaultValue
If myValue Is "" Then myValue = defaultValue
myValue = InputBox(message, title, defaultValue, 100, 100)
If myValue Is "" Then myValue = defaultValue
End Sub
End Class

Public Sub CreateMyForm()
    Dim mydialog As New Form()
    Dim button1 As New Button()
    Dim button2 As New Button()
    Dim lbldisplay As New Label
    button1.Text = "بلی"
    button1.Location = New Point(10, 50)
    button2.Text = "Cancel"
    button2.Location = New Point(button1.Right + 60, button1.Top)
    button1.DialogResult = DialogResult.OK
    button2.DialogResult = DialogResult.Cancel
    lbldisplay.Text = "Do you want to close? "
    lbldisplay.Location = New Point(10, 10)
    lbldisplay.Size = New Size(120, 20)
    mydialog.Text = "My Dialog Box"
    mydialog.FormBorderStyle = FormBorderStyle.FixedDialog
    mydialog.AcceptButton = button1
    mydialog.CancelButton = button2
    mydialog.StartPosition = FormStartPosition.CenterScreen
    mydialog.Controls.Add(button1)
    mydialog.Controls.Add(button2)
    mydialog.Controls.Add(lbldisplay)
    mydialog.ShowDialog()
    If mydialog.DialogResult = DialogResult.OK Then
        Me.Close()
    Else
        MessageBox.Show("The Cancel button on the form was clicked.")
    End If
End Sub

```

تابع **MessageBox.Show()**

این تابع نیز پیغام را روی صفحهء نمایش نشان داده و به کلیک کردن یک دکمه انتظار میکشد. این تابع عدد تامی را برگشت میدهد که همان عدد نشان دهندهء دکمهء کلیک شده است.

```
Public Class Form1
```



```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
    MessageBox.Show("This method is used to display" & Chr(13) &
    "messages on the screen")
```

```
End Sub
```

```
End Class
```

```
Public Class Form1
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
        Dim answer As DialogResult
```

```
        answer = MessageBox.Show("Do you want to close ?", "Dialog Box",
```

```
MessageBoxButtons.YesNo)
```

```
        If answer = System.Windows.Forms.DialogResult.Yes Then
```

```
            Me.Close()
```

```
        End If
```

```
    End Sub
```

```
End Class
```

```
Private Sub checkstudentname()
```

```
    If txtname.Text.Length = 0 Then
```

```
        Dim Message As String = "You did not enter a student name. Cancel
this operation?"
```

```
        Dim Caption As String = "No student Name Specified"
```

```
        Dim Buttons As Integer = MessageBoxButtons.YesNo
```

```
        Dim Result As DialogResult
```

```
        Result = MessageBox.Show(Me, Message, Caption,
```

```
MessageBoxButtons.YesNo)
```

```
        If Result = System.Windows.Forms.DialogResult.Yes Then
```

```
            Me.Close()
```

```
        End If
```

```
    End If
```

```
End Sub
```

نوت: در مثال ذیل طرز کار توابع InputBox و MessageBos.show را مشاهده نمائید.

```
Public class form1
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
    Dim prog As String
```

```
    prog = InputBox("Select program to run (Excel / Word / Access /
Calculator)")
```

```
    If prog = "Excel" Then
```

```
        Shell("c:\program files\microsoft office\office10\excel.exe",
```

```
AppWinStyle.MaximizedFocu)
```

```

ElseIf prog = "Word" Then
    Shell("c:\program files\microsoft office\office10\winword.exe")
ElseIf prog = "Access" Then
    Shell("c:\program files\microsoft office\office10\msaccess.exe")
ElseIf prog = "calc" Then
    Shell("c:\windows\system32\calc.exe")
End If
End Sub
End Class

```

در مثال ذیل از حادثه Load استفاده گردیده تا یک فورم بدون Password باز نگردد:

```

Public class form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim prog As String
again:
    prog = InputBox("Enter your password!")
    If prog = "er@bt-835" Then
        MessageBox.Show("Password accepted")
    Else
        MessageBox.Show("Wrong password")
        GoTo again
    End If
End Sub
End Class

```

```

Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim c As Byte
    c = 1
    Dim prog As Object
abc:
    c = c + 1
    prog = InputBox("Enter you Password")
    If prog = "ab@_6s" Then
        MessageBox.Show("Pssword accepted")
    Else
        MessageBox.Show("Wrong Password")
        If c < 4 Then
            GoTo abc
        Else
            Me.Enabled = False

```

```

End If
End If
End Sub
End Class

```

حادثه Close:

```

Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    If textBox1.Text = "" Then
        MessageBox.Show("You must enter a name.", "Name Entry Error",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
    End If
End Sub

```

حادثات صفحه کلید

حادثات صفحه کلید وقتی رخ میدهد که یک کلید ان فشار داده شود. هر آبجکت میتواند از حادثات صفحه کلید استفاده نماید و آنرا اجرا کند.

حادثه KeyPress

این حادثه در اثر فشار دادن کلید رخ میدهد که دارای کد اسکی تا 128 را داشته باشد. این حادثه نمیتواند حالت کلیدهای تغییر دهنده (Modifier keys) مانند Alt، Ctrl، و Shift را تعیین نماید. در صورت محکم گرفتن یک کلید این حادثه به تکرار واقع میشود و به سرعت که توسط سیستم عامل تعیین میگردد.

حادثه KeyDown

این حادثه وقتی رخ میدهد که کلید در اول فشار داده شود. این حادثه حالت کلیدهای Alt، Ctrl، و Shift را تعیین کرده میتواند.

حادثه KeyUp

این حادثه وقتی رخ میدهد که یک کلید بعد از فشار دادن رها شود. این حادثه حالت کلیدهای Alt، Ctrl، و Shift را تعیین کرده میتواند. نوت: برای اینکه حادثات صفحه کلید را به فورم محدود نمائیم و کنترولهای دیگر آنرا دریافت ننماید خاصیت KeyPressEventArgs.Handled حادثه KeyPress فورم را True نمائید.

حادثات صفحه کلید دارای خصوصیات ذیل میباشد:

خاصیت KeyChar

این خاصیت حرف اسکی کلید فشار داده شده را میدهد.

خاصیتهای Alt، Control، و Shift

این خصوصیات حالت کلید های Alt، Ctrl، و Shift را میدهد.

خاصیت KeyCode

این خاصیت کد اسکی کلید فشار داده شده را بدون معلومات راجع به کلیدهای تغییر دهنده را میدهد.

خاصیت KeyData

این خاصیت کود اسکی کلید فشار داده شده همراي معلومات راجع به کلید هاي تغیر دهنده را میدهد.

خاصیت KeyValue

مثال هاي ذیل حادثات صفحه کلید را نشان میدهد:

```
Public Class Form1
Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    TextBox2.Text = e.KeyCode
    TextBox3.Text = e.KeyData
    TextBox4.Text = e.KeyValue
    TextBox5.Text = Chr(e.KeyValue)
    TextBox6.Text = e.Shift
    TextBox7.Text = e.Control
    TextBox8.Text = e.Alt
    TextBox9.Text = e.GetType.FullName
End Sub
End Class
```

```
Public Class Form4
Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    If e.KeyCode = Keys.F2 Then
        TextBox2.Text = DateAndTime.DateValue(Now)
    ElseIf e.KeyCode = Keys.F5 Then
        TextBox2.Text = DateAndTime.TimeValue(Now)
    End If
End Sub
End Class
```

```
Public Class Form4
Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    TextBox2.Text = DateTime.Today
    TextBox3.Text = DateTime.Now
    TextBox4.Text = DateAndTime.Weekday(Today)
    TextBox5.Text = DateAndTime.WeekdayName(5)
    TextBox6.Text = DateAndTime.WeekdayName(Weekday(Today))
End Sub
End Class
```

```
Public Class Form4
```

```
Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    TextBox2.Text = Date.DaysInMonth(2000, 8)
    TextBox3.Text = Date.Now
    TextBox4.Text = Date.UtcNow
    TextBox5.Text = Date.MaxValue
    TextBox6.Text = Date.Today
End Sub
End Class
```

مثال: در مثال ذیل، هر حرف که از صفحه کلید فشار داده شود، در Label که بالای همین فورم است نشان داده میشود:

```
Public Class Form1
Private Sub Form1_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles Me.KeyPress
    Label1.Text = "Key pressed: " & e.KeyChar
End Sub
End Class
```

مثال ذیل کود اسکریپت کلید فشار داده شده را نشان میدهد:

```
Public Class Form1
Private Sub Form1_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles Me.KeyPress
    Label1.Text = "Key pressed: " & Asc(e.KeyChar)
End Sub
End Class
```

```
Public Class Form2
Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
    TextBox2.Text = e.KeyChar
    TextBox3.Text = e.ToString
    TextBox4.Text = e.GetHashCode
    TextBox5.Text = Asc(e.KeyChar)
End Sub
End Class
```

حادثه Keyup: این حادثه در اثر رها شدن یک کلید رخ میدهد.

```
Public Class Form1
Private Sub TextBox1_KeyUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyUp
    If e.KeyCode = Keys.F1 Then
        Help.ShowPopup(TextBox1, "Enter your first name", New
Point(TextBox1.Right, Me.TextBox1.Bottom))
    End If
End Sub
End Class
```

```

End If
End Sub
End Class

```

```

Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    TextBox2.Text = e.KeyCode
    TextBox3.Text = e.KeyData.ToString
    TextBox4.Text = e.KeyValue.ToString
End Sub

```

```

Private nonNumberEntered As Boolean = False
Private Sub textBox1_KeyDown(sender As Object, e As
System.Windows.Forms.KeyEventArgs) Handles textBox1.KeyDown
    nonNumberEntered = False
    If e.KeyCode < Keys.D0 OrElse e.KeyCode > Keys.D9 Then
        If e.KeyCode < Keys.NumPad0 OrElse e.KeyCode > Keys.NumPad9 Then
            If e.KeyCode <> Keys.Back Then
                nonNumberEntered = True
            End If
        End If
    End If
End Sub

```

```

Private Sub textBox1_KeyPress(sender As Object, e As
System.Windows.Forms.KeyPressEventArgs) Handles textBox1.KeyPress
    If nonNumberEntered = True Then
        e.Handled = True
    End If
End Sub
End Class

```

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Button2.PerformClick()
End Sub
End Class

```

پا استفاده از خاصیت KeyChar میتوانیم از کلید فشار داده شده در اثنای اجرای پروگرام نمونه بگیریم و همچنین فشار دادن کلید در شرایط خاص تغییر دهیم. مثلا میتوانیم توسط این خاصیت فشار دادن کلید های غیر عددی را در اثنای داخل نمودن کود ZIP غیر فعال سازیم یا مثلا تمام حروف را به کلان تبدیل کنیم.

مثال: در مثال ذیل به کمک خاصیت KeyChar دیده میشود که آیا کلید Enter فشار داده شده است و یا خیر؟ قیمت خاصیت Handled به True مساوی گردیده است.

```
Imports System
Imports System.Windows.Forms
Public Class Form1
    Inherits Form
    Public Sub New()
        Dim tb As New TextBox()
        Me.Controls.Add(tb)
        AddHandler tb.KeyPress, AddressOf keypressed
    End Sub
    Private Sub keypressed(ByVal o As [Object], ByVal e As
KeyPressEventArgs)
        If e.KeyChar = Microsoft.VisualBasic.ChrW(Keys.Return) Then
            e.Handled = True
        End If
    End Sub
    Public Shared Sub Main()
        Application.Run(New Form1())
    End Sub
End Class
```

حادثات موس (Mouse Events)

حادثات موس هم در اثر فشار دادن دکمه های موس رخ میدهد. هر آبجکت میتواند این حادثات را استفاده نماید. کلاس MouseEventArgs معلومات راجع به حادثه موس را دارد. این معلومات از قبیل کوردینات x و y موشر موس، کدام دکمه فشار داده شده است، چند بار دکمه فشار داده شده است، ویا چند بار چرخ موس دوران داده شده است (دکمه سومی موس). بعضی حادثات موس قرار ذیل است:

حادثه MouseEnter

این حادثه وقتی رخ میدهد که موشر موس به ساحه یک آبجکت داخل شود.

حادثه MouseEventArgs

حادثه MouseLeave

این حادثه وقتی رخ میدهد که موشر موس از ساحه آبجکت خارج شود.

حادثه MouseDown

حادثه MouseWheel

حادثه MouseUp

حادثه MouseHover

این حادثه وقتی رخ میدهد که موشر موس درحالت شنا بالای یک ابجکت باشد. از این حادثه برای نمایش Tool tip در اثنای شنای بالای کنترول استفاده میشود.

پروگرامو ذیل با استفاده از حادثات موس بالای فورم رسامی میکند:

```
Public Class FrmPainter
    Inherits System.Windows.Forms.Form
    Dim shouldPaint As Boolean = False
    Private Sub FrmPainter_MouseMove( _
        ByVal sender As System.Object, _
        ByVal e As System.Windows.Forms.MouseEventArgs) _
        Handles MyBase.MouseMove
        If shouldPaint Then
            Dim graphic As Graphics = CreateGraphics()
            graphic.FillEllipse _
                (New SolidBrush(Color.BlueViolet), e.X, e.Y, 4, 4)
        End If
    End Sub
    Private Sub FrmPainter_MouseDown(ByVal sender As Object, _
        ByVal e As System.Windows.Forms.MouseEventArgs) _
        Handles MyBase.MouseDown
        shouldPaint = True
    End Sub
    Private Sub FrmPainter_MouseUp(ByVal sender As Object, _
        ByVal e As System.Windows.Forms.MouseEventArgs) _
        Handles MyBase.MouseUp
        shouldPaint = False
    End Sub
End Class
```


فصل سوم

مفهوم انترفس گرافيكي

کنترول هاي مهم ویژول بسیک

بعد از آشنائي به محیط ویژول بسیک و قسمتي از مفاهيم مهم پروگرام نویسي، در این قسمت میخواهیم به کنترولهایی مهم ویژول بسیک 2010، و طرز استفاده آن آشنائي بیشتر حاصل نماییم. پروژه های ویژول بسیک یا پروگرامهای ویژول بسیک از همین کنترولها به حیث آبجکتها یا وسائل کار استفاده میکند. در آشنائي استفاده باید به خصوصیات (Properties)، میتودها (Methods) و حادثات (Events) این آبجکتها بلدیت داشته باشیم. در پهلوي این سه چیز، باید از فن پروگرام نویسي خود استفاده اعظمي نماییم. هر یک از این کنترولها را در موارد مختلف استفاده کرده میتوانیم. در بعضي موارد صرف از کنترولهای مشخص استفاده کرده میتوانیم. پس درک مورد استفاده هر کنترول نیز اهمیت دارد.

کنترول های ListBox و ComboBox

کنترول لست باکس جهت حفظ کردن لست از اختیارات به کار میرود. این اختیارات میتواند نام ها، ادرس ها، نمبر شناخت و غیره باشد. معمولاً از لست باکس طوری استفاده می شود که استفاده کننده (User) یکی از اختیارات آنرا انتخاب، و مطابق آن محتوایی چند کنترول دیگر تغییر نماید. مثلاً ممکن است خواسته باشیم تا با انتخاب نام یک شخص از لست، مقدار معاش و آدرس همان شخص در دو باکس متن نشان داده شود. یا ممکن است یک آدرس را از لست باکس انتخاب نمایم تا تعداد اشخاص همان آدرس را مشاهده نماییم.

عناصر را میتوانیم در آشنائي دیزاین، از طریق خاصیت Items به ListBox اضافه نماییم. همچنین میتوانیم با استفاده از میتود Add از طریق Code عناصر را به لست اضافه نماییم.

خصوصیات کنترول لست باکس (ListBox Properties)

خاصیت Text: این خاصیت محتوای عنصر انتخاب شده یا به عباره دیگر متن عنصر انتخاب شده را میدهد.

خاصیت SelectedItem: این خاصیت نیز محتوای عنصر انتخاب شده یا به عباره دیگر متن عنصر انتخاب شده را میدهد.

خاصیت SelectedIndex: این خاصیت اندکس عنصر انتخاب شده را می دهد. اگر چندین عنصر انتخاب گردد پده باشد در آن صورت اندکس آخرین عنصر انتخاب شده را میدهد. و اگر هیچ عنصر انتخاب نگردیده باشد قیمت آن 1- می باشد.

خاصیت Sorted: این خاصیت میتواند قیمت True و یا False را داشته باشد. اگر این قیمت True باشد عناصر لست خود بخود مرتب (Sort) میگردد. و اگر False باشد در آن صورت عناصر لست به ترتیبی که در لست داخل میگردد میباشند.

خاصیت Items: این خاصیت عناصر لست را به شکل یک لست (Array) حفظ مینماید. هر عنصر نظر به اندکس آن در لست شناسائی می گردد. اندکس اولین عنصر صفر، اندکس عنصر دومی 1 و اندکس آخرین عنصر مساوی به تعداد عناصر در لست منفی یک (n-1) میباشند.

خاصیت DataSource: توسط این خاصیت منبع گرفتن قیمتهای لست باکس را تعیین کرده میتوانیم. منبع قیمت ها میتواند لست باکس، Array و بانک معلومات باشد.

خاصیت TopIndex: توسط این خاصیت میتوانیم اولین عنصر لست را اخذ نمائیم و یا اولین عنصر لست را تعیین کنیم.

خاصیت MultiColumn: اگر قیمت این خاصیت True باشد لست میتواند چندین ستون داشته باشد. و اگر قیمت آن False باشد، لست صرف یک ستون میتواند داشته باشد.

خاصیت SetSelected: این خاصیت جهت انتخاب نمودن یک عنصر لست استفاده میشود.

```
ListBox1.SetSelected(1, True)
```

میتود های ListBox و ComboBox

میتود Add: این میتود جهت اضافه نمودن عناصر به لست، مورد استفاده قرار میگیرد. شکل عمومی آن قرار ذیل است:

```
ListBox1.Items.Add (item)
```

مثال ذیل اعداد از 1 تا 100 را به لست باکس اضافه میکند:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim i As Integer
    For i = 1 To 100
        ListBox1.Items.Add(i)
    Next i
End Sub
End Class
```

پروگرام فرعی ذیل، بعد از فشار دادن کلید Enter نام که در TextBox1 است آنرا به لست باکس انتقال میدهد:

```
Public Class Form1
Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    If e.KeyCode = Keys.Enter Then
        ListBox1.Items.Add(TextBox1.Text)
        TextBox1.Text = " "
    End If
End Sub
End Class
```

در پروگرام ذیل درجه حرارت از ComboBox انتخاب گردیده و معادل آن با فارنهایت در TextBox نشان داده می شود.

```
Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim i As Integer
    For i = 0 To 100
        ComboBox1.Items.Add(i)
    Next
End Sub
```

```
Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged
    TextBox1.Text = ComboBox1.SelectedItem * 9 / 5 + 32
End Sub
End Class
```

پروگرام ذیل جدول ضرب عدد انتخاب شده از ListBox را در TextBox نشان می دهد.

```
Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim i As Integer
    For i = 1 To 10
        ComboBox1.Items.Add(i)
    Next
End Sub
```

```
Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged
    Dim i, k As Integer
    TextBox1.Text = " "
    For i = 1 To 10
        k = ComboBox1.SelectedItem * i
        TextBox1.Text = TextBox1.Text & " " & k
    Next
End Sub
End Class
```

در مثال ذیل عناصر لست باکس اول به لست باکس دوم به کمک خاصیت DataSource انتقال مینماید:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim i As Integer
    For i = 1 To 25
        ListBox1.Items.Add(i)
    Next
    ListBox2.DataSource = ListBox1.Items
End Sub
End Class
```

در مثال ذیل عناصر Array به لست باکس انتقال مینماید:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim newarray As String() = New String(4) {"Computer Science", "Natural
Science", "Agriculture", "Engineering", "Economy"}
    ListBox1.DataSource = newarray
End Sub
```

میتود AddRange : این میتود جهت اضافه نمودن عناصر از یک Array به ListBox و یا
ComboBox استفاده میشود. شکل عمومی آن قرار ذیل است:

```
Listbox1.Items.AddRange (Array-name)
```

مثال:

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    ComboBox1.Items.AddRange(New Object() {"None", "Red", "Yellow",
"Green", "Blue", "Purple"})
End Sub
```

مثال:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim dep() As String = {"CS", "Math", "Phys", "Bio", "Chem"}
    ListBox1.Items.AddRange(dep)
End Sub
```

پروگرام ذیل سایز فونت یک TextBox را به کمک یک ListBox تعیین میکند:

```
Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    For i = 8 To 72 Step 2
        ListBox1.Items.Add(i)
    Next
End Sub
```

```
Private Sub ListBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ListBox1.SelectedIndexChanged
    TextBox1.Font = New Font("arial", ListBox1.SelectedItem)
End Sub
End Class
```

کنترول تایمر (Timer)

این کنترول جهت اجرای یک حادثه در وقفه که توسط استفاده کننده تعیین میگردد استفاده میگردد. مثلا
میخواهیم بعد از هر یک دقیقه محتوای یک باکس متن تغییر شود. یا میخواهیم مقدار وقت استفاده شده
توسط User را محاسبه و از User قیمت آن درخواست شود (مثلا در کلب انترنت).
خصوصیات کنترول تایمر:

خاصیت Interval: این خاصیت انتروال یا وقفه زمانی که بعد از گذشت آن حادثه اجرا شود، را تعیین میکند. قیمت این صفت به ملي ثانیه بوده فلذا اگر قیمت آن 1000 باشد حادثه در هر ثانیه یک بار اجرا میشود.
مثال:

```
Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Timer1.Start()
    Timer1.Interval = 1000
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    TextBox1.Text = TimeOfDay
End Sub
End Class
```

به مثال ذیل توجه نمائید:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Label1.Text = TimeOfDay()
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Label2.Text = TimeOfDay()
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Dim num1 As Integer
    num1 = Val(Mid(Label2.Text, 8, 2)) - Val(Mid(Label1.Text, 8, 2))
    Label3.Text = num1 * 10
End Sub
End Class
```

میتود Tick: با استفاده از میتود Tick کنترل تایمر میتوانیم بدون مداخله User دستورهای مشخص را در وقفه زمانی که در خاصیت انتروال است اجرا نمائیم. به مثال ذیل توجه نمائید:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Timer1.Start()
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    Label1.Text = TimeOfDay()
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Timer1.Stop()
End Sub
End Class
```

```
Public Class Form1
    Private Sub showtime()
        TextBox1.Text = TimeOfDay
    End Sub
    Private Sub changecolor()
        TextBox1.BackColor = Color.FromArgb(Rnd() * 255, Rnd() * 255, Rnd()
* 255)
    End Sub
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Timer1.Interval = 1000
    AddHandler Timer1.Tick, AddressOf showtime
    AddHandler Timer1.Tick, AddressOf changecolor
End Sub
End Class
```

خاصیت Enabled: اگر قیمت این خاصیت True باشد در آنصورت میتود Tick دستورهای مورد نظر را بدون اجرای دستور Timer.Start() اجرا میکند. و به این ترتیب اجرای پروگرام کاملاً بدون دخالت User صورت میگیرد.

خاصیت Timer:

خاصیت Timer تعداد ثانیه ها و ملی ثانیه ها را بعد از نصف شب نشان میدهد. ثانیه ها در قسمت تام و ملی ثانیه ها در قسمت اعشاری آن میباشد.

```
Public Class Form1
Public Sub waitFiveSeconds()
    If TimeOfDay >= #11:59:55 PM# Then
        MsgBox("The current time is within 5 seconds of midnight" & _
vbCrLf & "The timer returns to 0.0 at midnight")
    Return
End If
```

```

Dim start, finish, totalTime As Double
If (MsgBox("Press Yes to pause for 5 seconds", MsgBoxStyle.YesNo)) _
    = MsgBoxResult.Yes Then
    start = Microsoft.VisualBasic.DateAndTime.Timer
    finish = start + 5.0
    Do While Microsoft.VisualBasic.DateAndTime.Timer < finish
    Loop
    totalTime = Microsoft.VisualBasic.DateAndTime.Timer - start
    MsgBox("Paused for " & totalTime & " seconds")
End If
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Call waitFiveSeconds()
End Sub
End Class

```

کنترول CheckBox

این کنترول حالت On و Off بودن یک شرط را نشان میدهد. هر Check Box مستقل از یکدیگر بوده و به این ترتیب شما میتوانید چندین Check Box را همزمان انتخاب کنید. ازین کنترول زمانی استفاده میشود که بیشتر از یک شرط همزمان درست باشد.

خصوصیات کنترول CheckBox

خاصیت Checked: این خاصیت از نوع Boolean بوده و میتواند قیمت های True و False را اختیار نماید. این خاصیت تعیین میکند که آیا Check Box انتخاب شده یا نه و مهمترین مشخصه Check Box است.

خاصیت Appearance

این خاصیت طرز نمایش کنترول را تعیین میکند و میتواند قیمت Button و یا Normal را داشته باشد.

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    CheckBox1.Appearance = Appearance.Button
    CheckBox2.Appearance = Appearance.Normal
End Sub

```

خاصیت AutoChecked: این خاصیت نیز از نوع Boolean بوده و میتواند قیمت های True و False را اختیار نماید و هنگام Click نمودن، Check Box حالت انتخاب را اختیار نموده و یا از حالت انتخاب خارج میشود. اگر قیمت این کنترول False باشد در آنصورت با فشار دادن دکمه موس حالت انتخاب و یا غیر انتخاب آن تغییر نمیکند.

```

CheckBox2.AutoCheck = False

```



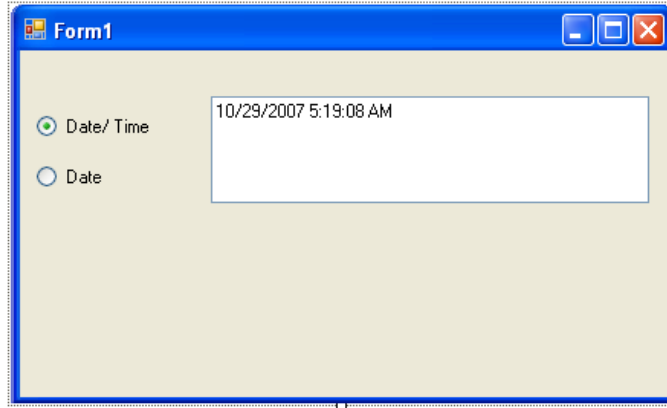
```
Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged
    If CheckBox1.Checked Then
        If RichTextBox1.Font.Bold Then
            RichTextBox1.Font = New Font(Me.Font, FontStyle.Regular)
        Else
            RichTextBox1.Font = New Font(Me.Font, FontStyle.Bold)
        End If
    End If
End Sub
```

```
Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox2.CheckedChanged
    If CheckBox2.Checked Then
        If RichTextBox1.Font.Italic Then
            RichTextBox1.Font = New Font(Me.Font, FontStyle.Regular)
        Else
            RichTextBox1.Font = New Font(Me.Font, FontStyle.Italic)
        End If
    End If
End Sub
```

```
Private Sub CheckBox3_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox3.CheckedChanged
    If CheckBox3.Checked Then
        If RichTextBox1.Font.Underline Then
            RichTextBox1.Font = New Font(Me.Font, FontStyle.Regular)
        Else
            RichTextBox1.Font = New Font(Me.Font, FontStyle.Underline)
        End If
    End If
End Sub
```


کنترول Radio Button

این کنترول برای انتخاب یک حالت و یا یک اختیار از میان چندین حالت و یا اختیار بکار میرود. فرق این کنترول با کنترول Check Box در این است که از مجموعه این کنترول ها در یک زمان فقط یکی آن انتخاب شده میتواند.



```
Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged
    Textbox1.Text = Date.Now
End Sub
```

```
Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton2.CheckedChanged
    Textbox1.Text = Date.Today
End Sub
```

و یا میتوانید برای فهم بیشتر مثال ذیل را در نظر بگیرید:

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged
    TextBox1.Text = Oct(Val(TextBox1.Text))
End Sub
```

```
Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton2.CheckedChanged
    TextBox1.Text = Hex(Val(TextBox1.Text))
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    If CheckBox1.Checked Then
        If RichTextBox1.Font.Bold Then
            RichTextBox1.Font = New Font(Me.Font, FontStyle.Regular)
        Else
            RichTextBox1.Font = New Font(Me.Font, FontStyle.Bold)
        End If
    End If
```

```

    End If
End If
If CheckBox2.Checked Then
    If RichTextBox1.Font.Italic Then
        RichTextBox1.Font = New Font(Me.Font, FontStyle.Regular)
    Else
        RichTextBox1.Font = New Font(Me.Font, FontStyle.Italic)
    End If
End If
If CheckBox3.Checked Then
    If RichTextBox1.Font.Underline Then
        RichTextBox1.Font = New Font(Me.Font, FontStyle.Regular)
    Else
        RichTextBox1.Font = New Font(Me.Font, FontStyle.Underline)
    End If
End If
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    If (ListBox1.Text = "Item1") And RadioButton2.Checked Then
        RadioButton1.PerformClick()
    End If
End Sub

Public Class Form1
Private Sub CheckBox3_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chktime.CheckedChanged
    Call timedate()
End Sub

Sub timedate()
    If chktime.Checked And chkdate.Checked Then
        TextBox1.Text = TimeOfDay & vbCrLf & Date.Today
    ElseIf chktime.Checked Then
        TextBox1.Text = TimeOfDay
    ElseIf chkdate.Checked Then
        TextBox1.Text = Date.Today
    Else
        TextBox1.Text = " "
    End If
End Sub

```

```
Private Sub chkdate_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles chkdate.CheckedChanged
    Call timedate()
End Sub
End Class
```

کنترول **CheckedListBox**

این کنترول درحقیقت کنترول لست باکس را توسعه میدهد. همراي هر عنصر **ListBox** یک **CheckBox** را نیز اضافه میکند. به کمک اینکار استفاده کننده میتواند چندین عنصر را یکبار انتخاب نماید. بعضي خصوصیات این کنترول قرار ذیل است:

خاصیت **CheckedItems**

لست از عناصر را دارد که انتخاب گردیده است.

خاصیت **CheckedIndices**

اندکس هاي تمام عناصر انتخاب شده را میآهد.

خاصیت **SelectionMode**

این خاصیت تعیین میکند که چند عنصر انتخاب شده میتواند. میتواند هاي این کنترول قرار ذیل است:

میتود **SelectionMode**

این میتود تعداد عناصر که انتخاب شده میتواند آنرا تعیین میکند. اگر قیمت آن یک باشد چندین عنصر انتخاب شده میتواند. و اگر قیمت آن **None** باشد هیچ عنصر انتخاب شده نمیتواند.

میتود **GetItemChecked**

یک اندکس را میگیرد و قیمت **True** را میدهد اگر عنصر آن انتخاب شده باشد.

حادثه **ItemCheck**

این حادثه وقتی رخ میدهد که عنصر انتخاب شود. به مثالي ذیل توجه شود:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim dep() As String = {"CS", "Math", "Phys", "Bio", "Chem"}
    CheckedListBox1.Items.AddRange(dep)
    CheckedListBox1.CheckOnClick = True
End Sub
```

درمثال ذیل اگر متن **Textbox1** در اختیارات **CheckedListBox** نباشد در انصورت همان متن **TextBox** به **CheckedListBox** اضافه میشود.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    If TextBox1.Text <> "" Then
        If CheckedListBox1.CheckedItems.Contains(TextBox1.Text) = False Then
            CheckedListBox1.Items.Add(TextBox1.Text, CheckState.Checked)
        End If
        TextBox1.Text = ""
    End If
End Sub
```

مثال ذیل عناصر انتخاب شده CheckedListBox را به listBox انتقال میدهد:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim i As Integer
    For i = 0 To CheckedListBox1.CheckedItems.Count - 1
        ListBox1.Items.Add(CheckedListBox1.CheckedItems(i))
    Next i
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim myEnumerator As IEnumerator
    myEnumerator = CheckedListBox1.CheckedIndices.GetEnumerator()
    Dim y As Integer
    While myEnumerator.MoveNext() <> False
        y = CInt(myEnumerator.Current)
        CheckedListBox1.SetItemChecked(y, False)
    End While
End Sub
```

نوت: از Enumerator در خواندن Data از Collection استفاده میشود. در ابتدا Enumerator قبل از اولین عنصر Collection قرار میگیرد و هر وقت که میتود Reset را Call نمایم هم Enumerator را به همین موقعیت می آورد. در این حالت استفاده از خاصیت Current باعث غلطی میشود. فلذا باید میتود MoveNext را صدا نمایم تا Enumerator به اولین عنصر Collection برود قبل از خواندن قیمت Current. به عوض Enumerator از حلقه For Each هم به شکل ذیل استفاده کرده میتوانیم:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    For Each item In CheckedListBox1.CheckedIndices
        CheckedListBox1.SetItemChecked(item, False)
    Next
End Sub
```

کنترول ProgressBar

این کنترول پیشرفت یک عملیه را نشان میدهد. در صورت تکمیل شدن عملیه ProgressBar هم تکمیل (پر) میشود. درحقیقت ProgressBar به User مفکوره راجع به پیشرفت کار یک عملیه میدهد. در اثنای انستال کردن پروگرام های مانند ویندوز و آفس ProgressBar مقدار پیشرفت کار ویندوز را نشان میدهد. این کنترول را صرف میتوانیم به شکل افقی روی فورم قرار دهیم.

خصوصیات کنترول ProgressBar:

- خاصیت Minimum:** این خاصیت قیمت شروع کنترول ProgressBar را نشان میدهد.
- خاصیت Maximum:** این خاصیت قیمت نهایی کنترول ProgressBar را تعیین میکند.
- خاصیت Value:** این خاصیت قیمت حالی ProgressBar را نشان میدهد.
- خاصیت Step:** خاصیت Step مقدار تغیر و حرکت ProgressBar را تعیین میکند.

خاصیت Style: این خاصیت شکل که پیشرفت ProgressBar را تعیین میکند نشان میدهد. قیمت این خاصیت میتواند Continuous، Blocks ویا Marquee باشد.

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    ProgressBar1.Value += 1
End Sub
End Class
```

```
Public Class Form1
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    ProgressBar1.Value += 1
End Sub
End Class
```

مثال دوم:

```
Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ProgressBar1.Visible = True
    ProgressBar1.Style = ProgressBarStyle.Continuous
    ProgressBar1.Minimum = 1
    ProgressBar1.Maximum = 100
    ProgressBar1.Enabled = True
```

```
End Sub
```

```
Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    If e.KeyCode = Keys.Enter Then
        ProgressBar1.PerformStep()
        TextBox1.Text = " "
    End If
End Sub
End Class
```

مثال سوم:

```
Public Class Form1
Private time As New Timer()
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    time.Interval = 250
    AddHandler time.Tick, AddressOf IncreaseProgressBar
    time.Start()
End Sub
```

```

Private Sub IncreaseProgressBar(ByVal sender As Object, ByVal e As
EventArgs)
    ProgressBar1.Increment(1)
    TextBox1.Text = ProgressBar1.Value.ToString() + "% Completed"
    If ProgressBar1.Value = ProgressBar1.Maximum Then
        time.Stop()
    End If
End Sub
End Class

```

نوټ: دستور AddHandler یک حادثه (Event) را همراي یک Event Handler در اثناي اجراي پروگرام یکجا میسازد.

```

Public Class Form1
    Sub TestEvents()
        Dim Obj As New Class1
        AddHandler Obj.Ev_Event, AddressOf EventHandler
        Obj.CauseSomeEvent()
        RemoveHandler Obj.Ev_Event, AddressOf EventHandler
        Obj.CauseSomeEvent()
    End Sub

    Sub EventHandler()
        MsgBox("EventHandler caught event.")
    End Sub

Public Class Class1
    Public Event Ev_Event()
    Sub CauseSomeEvent()
        RaiseEvent Ev_Event()
    End Sub
End Class

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Call TestEvents()
End Sub
End Class

```

```

Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ProgressBar1.Minimum = 0
    ProgressBar1.Maximum = 25

```

```

    ProgressBar1.Step = 1
End Sub

```

```

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    If ProgressBar1.Value < 25 Then
        ProgressBar1.Value += 1
    Else
        Timer1.Enabled = False
        MessageBox.Show("The process is completed")
        ProgressBar1.Value = 0
    End If
End Sub
End Class

```

```

Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    If e.KeyCode = Keys.Enter Then
        ProgressBar1.Value += 1
        TextBox1.Text = " "
        TextBox2.Text = Val(TextBox2.Text) + 1
    End If
End Sub

```

```

Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        ProgressBar1.Minimum = 0
        ProgressBar1.Maximum = 20
        ProgressBar1.Style = ProgressBarStyle.Blocks
    End Sub

```

```

Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    If e.KeyCode = Keys.Enter Then
        ProgressBar1.Value += 1
        TextBox1.Text = " "
        TextBox2.Text = ProgressBar1.Value
        Label1.Text = ProgressBar1.Value * 100 / ProgressBar1.Maximum & "
" & "% Completed"
    End If
End Sub
End Class

```

```

Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Time.Interval = 250
    AddHandler Time.Tick, AddressOf changeprogress
    AddHandler Time.Tick, AddressOf changeprogress2
    Time.Start()
End Sub

Private Sub changeprogress()
    ProgressBar1.Increment(1)
    TextBox1.Text = ProgressBar1.Value.ToString() + "% Completed"
    If ProgressBar1.Value = ProgressBar1.Maximum Then
        Time.Stop()
        MessageBox.Show("completed")
        Me.Close()
    End If
End Sub

Private Sub changeprogress2()
    Me.BackColor = Color.FromArgb(Rnd() * 255, Rnd() * 255, Rnd() * 255)
End Sub
End Class

```

```

Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ProgressBar1.Minimum = 0
    ProgressBar1.Maximum = 50
    ProgressBar1.Style = ProgressBarStyle.Blocks
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    If ProgressBar1.Value < ProgressBar1.Maximum Then
        ProgressBar1.Value += 1
        Label1.Text = ProgressBar1.Value * 100 / ProgressBar1.Maximum & "
" & "% Completed"
    Else
        Timer1.Enabled = False
        MessageBox.Show("The process is completed")
    End If
End Sub

```



```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim result As DialogResult = MessageBox.Show("Do you to cancel? ",
"Stopping Dialog box", MessageBoxButtons.OKCancel)
    If result = Windows.Forms.DialogResult.OK Then
        Timer1.Enabled = False
        Me.Close()
    End If
End Sub
End Class

```

کنترول TrackBar

از این کنترول جهت تعیین سرعت و یا جستجو در معلومات زیاد استفاده میشود. این کنترول را بنام Slider هم یاد میکند. خصوصیات مهم این کنترول عبارت اند از:

خاصیت Value: این خاصیت قیمت و موقعیت فعلی کنترول را تعیین میکند.

خاصیت Minimum: کمترین قیمت که کنترول آنرا میگیرد.

خاصیت Maximum: بلندترین قیمت که کنترول آنرا میگیرد.

خاصیت TickFrequency: مسافه زمانی بین دو Tick را تعیین میکند.

خاصیت SmallChange: اندازه حرکت یا تغییر در قیمت (خاصیت Value) کنترول در اثر فشار دادن دکمه موس و یا کلید های Left Arrow و یا Right Arrow بالای خود Slide کنترول.

خاصیت LargeChange: اندازه حرکت یا تغییر در قیمت (خاصیت Value) کنترول در اثر فشار دادن دکمه موس و یا کلید های Left Arrow و یا Right Arrow بالای دو طرف کنترول.

به مثال ذیل توجه کنید:

```

Public Class Form1
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    If ProgressBar1.Value + TrackBar1.Value < ProgressBar1.Maximum Then
        ProgressBar1.Value += TrackBar1.Value
    Else
        ProgressBar1.Value += ProgressBar1.Maximum - ProgressBar1.Value
    End If
End Sub
End Class

```

```

Public Class Form1
Private Sub TrackBar1_Scroll(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TrackBar1.Scroll
    TextBox1.Text = TrackBar1.Value * 2
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    TrackBar1.Minimum = 0

```

```

        TrackBar1.Maximum = 50
        TrackBar1.TickFrequency = 5
    End Sub
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        ProgressBar1.Value += TrackBar1.Value
    End Sub
End Class

Private Sub TrackBar1_Scroll(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TrackBar1.Scroll
    TextBox1.Text = TrackBar1.Value * 2
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    TrackBar1.Minimum = 0
    TrackBar1.Maximum = 50
    TrackBar1.TickFrequency = 5
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    ProgressBar1.Value += TrackBar1.Value
End Sub

Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        TrackBar1.Minimum = 0
        TrackBar1.Maximum = 100
    End Sub
    Private Sub TrackBar1_Scroll(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TrackBar1.Scroll
        TextBox1.Text = TrackBar1.Value
        TextBox2.Text = TrackBar1.Value * 9 / 5 + 32
    End Sub

End Class

Public Class Form1
    Private Sub trblue_Scroll(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles trblue.Scroll

```

```

    Me.BackColor = Color.FromArgb(trred.Value, trgreen.Value,
trblue.Value)
End Sub
Private Sub trgreen_Scroll(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles trgreen.Scroll
    Me.BackColor = Color.FromArgb(trred.Value, trgreen.Value,
trblue.Value)
End Sub
Private Sub trred_Scroll(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles trred.Scroll
    Me.BackColor = Color.FromArgb(trred.Value, trgreen.Value,
trblue.Value)
End Sub
End Class

```

کنترول DomainUpDown

این کنترول مشابه به کنترول ComboBox بوده و میتواند جهت نشان دادن اوست از قیمت ها استفاده شود. به مثال ذیل توجه شود:

```

Public Class Form1
Protected domainUpDown1 As DomainUpDown
Dim mycounter As Integer
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    domainUpDown1 = New System.Windows.Forms.DomainUpDown()
    Controls.Add(domainUpDown1)
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    domainUpDown1.Items.Add((TextBox1.Text.Trim() & " - " &
mycounter))
    mycounter = mycounter + 1
    TextBox1.Text = ""
End Sub

Private Sub CheckBox1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CheckBox1.Click
    If domainUpDown1.Sorted Then
        domainUpDown1.Sorted = False
    Else
        domainUpDown1.Sorted = True
    End If
End Sub

```

```
Private Sub domainUpDown1_SelectedItemChanged _
    (ByVal sender As System.Object, ByVal e As System.EventArgs)
    MessageBox.Show(("SelectedIndex: " &
    domainUpDown1.SelectedIndex.ToString() & _
    ControlChars.Cr & "SelectedItem: " &
    domainUpDown1.SelectedItem.ToString()))
End Sub
End Class
```

کنترول DateTimePicker

توسط این کنترول میتوانیم تاریخ و وقت را انتخاب و آنرا به فارمت مورد نظر خود نشان دهیم. با استفاده از خصوصیات این کنترول میتوانیم مواصفات مختلف این کنترول را تغییر دهیم. برای تعیین نمودن شروع و ختم تاریخ و وقت از خصوصیات MinData و MinTime این کنترول استفاده میکنیم. برای تغییر قسمت تقویم آن از خصوصیات CalendarFont، CalendarForeColor، و غیره خصوصیات آن استفاده میکنیم. خاصیت Format آن فارمت آنرا تعیین میکند. فارمت تعیین شده از قبل (Default) عبارت از DateTimePickerFormat.Long است. فارمت Default را میتوانیم با تغییر دادن قیمت این خاصیت تغییر دهیم. اگر قیمت آنرا به Custom تغییر دهیم در آنصورت میتوانیم فارمت مورد نظر خود را تعیین نمائیم. مثلا اگر خواسته باشیم تا تاریخ را به شکل "September 03, 2007 – Monday" نشان دهیم قیمت CustomFormat را به "MMMM dd, yyyy – dddd" تغییر میدهیم. به مثال ذیل توجه نمائید:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    TextBox1.Text = DateTimePicker1.Value
End Sub
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    DateTimePicker1.MinDate = "2007, 7, 20"
    DateTimePicker1.MaxDate = DateTime.Today
    DateTimePicker1.CustomFormat = "MMMM dd, yyyy - dddd"
    DateTimePicker1.Format = DateTimePickerFormat.Custom
    DateTimePicker1.ShowCheckBox = True
End Sub
End Class
```

مثال دوم:

```
Public Class frmDate
Dim dateinterval As Integer = 7
Private Sub DateChooser_ValueChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DateChooser.ValueChanged
    If DateNumber.Text.Length > 0 Then
        DateInterval = CInt(DateNumber.Text)
    End If
```

```

NextWeek.Text =
DateChooser.Value.Add(TimeSpan.FromDays(dateinterval)).ToString()
End Sub
End Class

```

نوت: آبجکت TimeSpan یک مقطع (Interval) وقت را تمثيل میکند. واحد این مقطع میتواند روز، ساعت، دقیقه، ثانیه، و یا جز از ثانیه باشد. البته قیمت این آبجکت میتواند مثبت و یا منفي باشد. قیمت حقيقي آن به تیک (Ticks) است. یک تیک مساوي است به 100 نانو ثانیه. آبجکت TimeSpan مربوط به کلاس System بوده و از همین سبب دستور Import System ممکن ضرور باشد.

کنترول MonthCalendar

این کنترول جنتری مکمل را نشان میدهد. البته تاریخ را به کمک صفات MinDate و MaxDate محدود هم ساخته میتوانیم.

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    MonthCalendar1.FirstDayOfWeek =
System.Windows.Forms.Day.Monday
    MonthCalendar1.MaxDate = New System.DateTime(2010, 12, 31, 0, 0,
0, 0)
    MonthCalendar1.MinDate = New System.DateTime(1999, 1, 1, 0, 0, 0, 0)
    MonthCalendar1.MaxSelectionCount = 21
    MonthCalendar1.ScrollChange = 1
    MonthCalendar1.ShowToday = True
    MonthCalendar1.ShowTodayCircle = True
    MonthCalendar1.ShowWeekNumbers = True
    ClientSize = New System.Drawing.Size(920, 566)
    Controls.AddRange(New System.Windows.Forms.Control()
{Me.TextBox1, Me.MonthCalendar1})
    Text = "Month Calendar Example"
End Sub

```

کنترولهایی GroupBox و panel

از این دو کنترول جهت مرتب ساختن کنترولهایی دیگر استفاده میشود. مثلاً میتوانیم دکمه های که وظائف مشابه را انجام میدهد در این کنترولهایی جابجا نمائیم. برای استفاده از این کنترولهایی اول آنرا به فورم آورده و بعد از آن کنترولهایی دیگر را در آن جابجا میکنیم. این کنترولهایی دارای خصوصیات BorderStyle، AutoScroll، و Controls میباشد.

کنترول ToolTip

به کمک این کنترول شرح مختصر راجع به یک کنترول نشان داده میشود. این شرح در اکثر وقت مخفی است و هنگامی که توسط موس به همان کنترول اشاره میشود، به استفاده کننده نشان داده میشود. ToolTip همراه هر کنترول استفاده شده میتواند. میتود های کلیدی این کنترول عبارت از SetToolTip و GetToolTip میباشد. برای اینکه ToolTip یک کنترول را نشان دهیم از میتود

SetToolTip استفاده میکنیم. برای این کار اول باید کنترل ToolTip را بالای فورم قرار داده شود.

```
Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ToolTip1.SetToolTip(Button1, "For more information")
    ToolTip1.SetToolTip(TextBox1, "Enter student name")
End Sub
End Class
```

خصوصیات کنترل ToolTip:

خاصیت AutomaticDelay: این خاصیت وقت که میان اشاره توسط و نشان دادن ToolTip میباشد.

خاصیت AutoPopDelay: این خاصیت مدت که در آن ToolTip نشان داده میشود، تعیین میکند.

خاصیت ReShowDelay: وقفه که بعد از گذشت آن ToolTip دوباره نشان داده شود.

خاصیت ShowAlways: اگر قیمت این خاصیت True باشد ToolTip دائمی نشان داده میشود. در مثال ذیل از همین خصوصیات کنترل ToolTip استفاده گردیده است.

```
Private Sub Form1_Load(sender As Object, e As System.EventArgs) Handles
MyBase.Load
    Dim toolTip1 As New ToolTip()
    toolTip1.AutoPopDelay = 5000
    toolTip1.InitialDelay = 1000
    toolTip1.ReshowDelay = 500
    toolTip1.ShowAlways = True
End Sub
```

استفاده از Help:

```
Public Class Form1
Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    If e.KeyCode = Keys.F1 Then
        Help.ShowPopup(TextBox1, "Enter your name.", New
Point(TextBox1.Bottom, TextBox1.Right))
    End If
End Sub
End Class
```

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Help.ShowHelp(Button1, "d:\asd.txt")
End Sub
End Class
```

مثال:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Help.ShowHelp(Label1, "C:\abc.htm")
End Sub
End Class
```

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Help.ShowHelp(Nothing, "C:\abc.htm")
End Sub
End Class
```

کنترول HelpProvider

در ویژول بسیک میتوانیم فایل‌های HTML را نمایش دهید. این فایلها به کمک کلاس Help صورت میگیرد. این کلاس دارای دو میتود میباشد: میتود ShowHelp و میتود ShowHelpIndex. از میتود ShowHelp جهت نشان دادن فایل Help برای یک کنترول معین استفاده میشود. URL فایل Help میتواند به شکل C:\Myhelp (در صورت که فایل محلی باشد) و یا به شکل http:\\ www.startvbdotnet.com/help.htm در صورت وب باشد.

```
Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    HelpProvider1.SetHelpString(Button1, "Enter this button to continue")
    HelpProvider1.SetShowHelp(Button1, True)

    HelpProvider1.SetHelpString(CheckBox1, "select this check box to show
the help")
    HelpProvider1.SetShowHelp(CheckBox1, True)
    Me.MaximizeBox = False
    Me.MinimizeBox = False
    Me.HelpButton = True
End Sub
End Class
```

```
Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    HelpLabel.Text = "Click the Help button in the title bar, then click a
control " & _
    "to see a Help tooltip for the control. Click on a control and press F1 to
invoke " & _
```

```

    "the Help system with a sample Help file."
    HelpProvider1.SetHelpString(addressTextBox, "Enter the street address in
this text box.")
    HelpProvider1.SetShowHelp(addressTextBox, True)
    HelpProvider1.SetHelpString(cityTextBox, "Enter the city here.")
    HelpProvider1.SetShowHelp(cityTextBox, True)
    HelpProvider1.SetHelpString(stateTextBox, "Enter the state in this text
box.")
    HelpProvider1.SetShowHelp(stateTextBox, True)
    HelpProvider1.SetHelpString(zipTextBox, "Enter the zip code here.")
    HelpProvider1.SetShowHelp(zipTextBox, True)
    Me.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedDialog
    Me.HelpButton = True
    Me.MaximizeBox = False
    Me.MinimizeBox = False
    Me.ClientSize = New System.Drawing.Size(292, 160)
    Me.Text = "Help Provider Demonstration"
End Sub
End Class

```

جز HelpProvider

به کمک HelpProvider میتوانیم Help را برای یک کنترل با فشار دادن دکمه F1 تهیه نماییم. یک خاصیت مهم Helpprovider عبارت از HelpNameSpace میباشد که URL فایل کمک را تعیین میکند. Helpprovider برای هر کنترل روی فورم دارای سه خاصیت ذیل هم میباشد:

خاصیت HelpString

این خاصیت سلسله مربوط به کنترل را تعیین میکند.

خاصیت HelpkeyWord

این خاصیت کلمه کلیدی help مربوط به کنترل را تعیین میکند.

خاصیت HelpNavigator

این خاصیت نوع Help مربوط به کنترل را تعیین میکند. این خاصیت میتواند یکی از شش قیمت ذیل را به خود بگردد: AssociateIndex، Topic، Index، Find، TableOfContents، و KeywordIndex.

سه خاصیت فوق برای هر کنترل بوجود می آید در صورت که HelpProvider را به فورم اضافه کنیم.
مثال:

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    HelpProvider1.SetHelpString(Button1, "I am supported by
HelpProvider1")
End Sub

```

```

Me.helpProvider1 = New System.Windows.Forms.HelpProvider()

```



```
' Tell the HelpProvider what controls to provide Help for, and
' what the Help string is.
Me.helpProvider1.SetHelpString(Me.addressTextBox, "Enter the street
address in this text box.")
Me.helpProvider1.SetShowHelp(Me.addressTextBox, True)

Me.helpProvider1.SetHelpString(Me.cityTextBox, "Enter the city here.")
Me.helpProvider1.SetShowHelp(Me.cityTextBox, True)

Me.helpProvider1.SetHelpString(Me.stateTextBox, "Enter the state in this
text box.")
Me.helpProvider1.SetShowHelp(Me.stateTextBox, True)

Me.helpProvider1.SetHelpString(Me.zipTextBox, "Enter the zip code
here.")
Me.helpProvider1.SetShowHelp(Me.zipTextBox, True)

' Sets what the Help file will be for the HelpProvider.
Me.helpProvider1.HelpNamespace = "mspaint.chm"
میتود ShowHelp محتوایی فایل Help را نشان میدهد. به مثال ذیل توجه نمایید:
Private Sub Button1_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button1.Click

    Help.ShowHelp(TextBox1, "file://c:\chormap.chm")
End Sub
```

کنترول PictureBox

این کنترول میتواند تصویر را نشان دهد. تصویر میتواند از نوع BMP، GIF، JPEG و غیره باشد. خصوصیت Image این کنترول تصویر مورد نظر را حفظ میکند.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    PictureBox1.Image = Image.FromFile("c:\picture1.png")
    PictureBox1.SizeMode = PictureBoxSizeMode.StretchImage
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Me.BackgroundImage = PictureBox1.Image
End Sub
```

کنترول ImageList

توسط این کنترول مجموعه از آبجکتهای Image را اداره کرده میتوانیم. به مثال ذیل توجه نمایید.

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```

Static i As Integer
If i < 2 Then
    PictureBox1.Image = ImageList1.Images(i)
    i += 1
Else
    Timer1.Enabled = False
    MessageBox.Show("Completed")
End If
End Sub
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ImageList1.ImageSize = New Size(112, 112)
    ImageList1.Images.Add(Image.FromFile("c:\Picture1.png"))
    ImageList1.Images.Add(Image.FromFile("C:\adelyar.bmp"))
End Sub
End Class

```

کنترول ImageList معمولاً توسط کنتروله‌های دیگر استفاده می‌شود. شما می‌توانید تصاویر را به این کنترول اضافه نمایید و کنتروله‌های دیگر می‌تواند آنرا استفاده کند.

کنترول LinkLabel

این کنترول جهت رابطه به منابع دیگر از قبیل فایلها، صفحه وب و یا به عباره دیگر از این کنترول جهت ساختن HyperLink استفاده کرده می‌توانیم. بعضی خصوصیات این کنترول قرار ذیل است:

خاصیت LinkArea

این خصوصیت قسمت از متن LinkLabel که در Link نشان داده شود آنرا تعیین میکند.

خاصیت LinkBehavior

طرز نمایش Link را تعیین میکند. مثلاً وقتی که موس بالای Link آورده شود چه قسم نشان داده شود.

خاصیت LinkColor

رنگ متن Link را تعیین میکند.

خاصیت Link

لست از link های که این کنترول آنرا حفظ میکند نشان میدهد.

خاصیت LinkVisited

اگر قیمت این خصوصیت True باشد در آنصورت link های دیده شده نشانی می‌شود.

خاصیت VisitedLinkColor

رنگ link های دیده شده را نشان میدهد.

نوت: خاصیت LinkArea همان HyperLink که در متن (Text) این کنترول نشان داده می‌شود، تعیین مینماید. و آبجکت LinkArea خصوصیات را تهیه میکند که به کمک آن شروع هر Link در متن این کنترول تعیین می‌شود و همچنین طول متن برای همان HyperLink. وقتی که یک HyperLink به کمک خاصیت LinkArea تعیین می‌شود همان HyperLink به LinkCollection کنترول اضافه می‌شود. همچنین خاصیت LinkArea آبجکت LinkArea که مربوط آن است به آبجکت LinkLabel.Link تبدیل میکند که در Collection ذخیره است.

برای اینکه چندین HyperLink را به متن کنترل اضافه نماییم از خاصیت Link استفاده میکنیم. خاصیت Link به شما اجازه میدهد تا خصوصیات و میتود های LinkLabel.LinkCollection را استفاده کنید. این میتود ها Link های که برای کنترل است تعیین میکند. این روش اضافه نمودن Link ها به LinkLabel همچنین به شما اجازه میدهد تا Data که در LinkData است آنرا تعیین کند. به کمک قیمت خاصیت LinkData میتوانید موقعیت فایل ویا آدرس وبسایت را ذخیره کرده میتوانید.

```
Public Class Form1
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    LinkLabel1.DisabledLinkColor = System.Drawing.Color.Red
```

```
    LinkLabel1.VisitedLinkColor = System.Drawing.Color.Blue
```

```
    LinkLabel1.LinkBehavior =
```

```
System.Windows.Forms.LinkBehavior.HoverUnderline
```

```
    LinkLabel1.LinkColor = System.Drawing.Color.Navy
```

```
    LinkLabel1.LinkArea = New System.Windows.Forms.LinkArea(0, 8)
```

```
    LinkLabel1.Text = "Microsoft Word."
```

```
    LinkLabel1.Links(0).LinkData = "c:\program files\microsoft office\office11\winword.exe"
```

```
    LinkLabel2.DisabledLinkColor = System.Drawing.Color.Red
```

```
    LinkLabel2.VisitedLinkColor = System.Drawing.Color.Blue
```

```
    LinkLabel2.LinkBehavior =
```

```
System.Windows.Forms.LinkBehavior.HoverUnderline
```

```
    LinkLabel2.LinkColor = System.Drawing.Color.Navy
```

```
    LinkLabel2.LinkArea = New System.Windows.Forms.LinkArea(0, 8)
```

```
    LinkLabel2.Text = "Calculator"
```

```
    LinkLabel2.Links(0).LinkData = "c:\windows\system32\calc.exe"
```

```
End Sub
```

```
Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked
```

```
    LinkLabel1.Links(LinkLabel1.Links.IndexOf(e.Link)).Visited = True
```

```
    Dim target As String = CType(e.Link.LinkData, String)
```

```
    Shell(target)
```

```
End Sub
```

```
Private Sub LinkLabel2_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel2.LinkClicked
```

```
    LinkLabel2.Links(LinkLabel2.Links.IndexOf(e.Link)).Visited = True
```

```
    Dim target As String = CType(e.Link.LinkData, String)
```

```
    Shell(target)
```

```
End Sub
```

End Class

مثال دوم:

```

Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    LinkLabel1.DisabledLinkColor = System.Drawing.Color.Red
    LinkLabel1.VisitedLinkColor = System.Drawing.Color.Blue
    LinkLabel1.LinkBehavior =
System.Windows.Forms.LinkBehavior.HoverUnderline
    LinkLabel1.LinkColor = System.Drawing.Color.Navy
    LinkLabel1.LinkArea = New System.Windows.Forms.LinkArea(0, 8)
    LinkLabel1.Links(0).Visited = True
    LinkLabel1.Text = "BBC News. Visit Microsoft. Visit MSN."
    If LinkLabel1.Text.Length >= 45 Then
        LinkLabel1.Links.Add(0, 8, "www.BBC.com")
        LinkLabel1.Links.Add(24, 9, "www.microsoft.com")
        LinkLabel1.Links.Add(42, 3, "www.msn.com")
        LinkLabel1.Links(1).Enabled = False
    End If
End Sub

Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel1.LinkClicked
    LinkLabel1.Links(LinkLabel1.Links.IndexOf(e.Link)).Visited = True
    Dim target As String = CType(e.Link.LinkData, String)
    If (target IsNot Nothing) AndAlso (target.StartsWith("www")) Then
        System.Diagnostics.Process.Start(target)
    Else
        MessageBox.Show(("Item clicked: " + target))
    End If
End Sub
End Class

```

کنترول RichTextBox

این کنترول توانایی کامل برای یک Editor متن را دارد. بر خلاف کنترول TextBox، صفحه Editor می تواند دارای فارمت های مختلف باشد. پرگرام Wordpad که در ویندوز است به اساس همین کنترول ساخته شده است.

خصوصیات کنترول RichTextBox برای پراسس متن:

بسیاری خصوصیات این کنترول مانند کنترول TextBox میباشد. در پهلوئی آن خصوصیات ذیل نیز بیشتر استفاده میگردد. خصوصاً در اثنائی استفاده این کنترول در دیزاین Editor متن:

،SelectionBackColor ،SelectionAlignment ،SelectionColor ،SelectedText
،Selectionlength ،SelectionFont و SelectionStart زیاد استفاده میگردد.
به مثال هاي ذیل توجه نمائید:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
RichTextBox1.SelectionBackColor = Color.Red
RichTextBox1.SelectionColor = Color.Blue
RichTextBox1.SelectionFont = New Font("Arial", 16)
RichTextBox1.SelectionAlignment = HorizontalAlignment.Center
RichTextBox1.SelectionBullet = True
RichTextBox1.SelectionCharOffset = -12
RichTextBox1.SelectionHangingIndent = 6
End Sub
End Class

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
RichTextBox1.SelectionStart = 0
RichTextBox1.SelectionLength = Len(RichTextBox1.Text)
RichTextBox1.Select()
End Sub

Private Sub RichTextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles RichTextBox1.KeyDown
If e.KeyCode = Keys.Control + Keys.A Then
RichTextBox1.SelectionStart = 0
RichTextBox1.SelectionLength = Len(RichTextBox1.Text)
RichTextBox1.Select()
End If
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
RichTextBox1.SelectAll()
RichTextBox1.Select()
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
ColorDialog1.ShowDialog()
RichTextBox1.ForeColor = ColorDialog1.Color
End Sub
```

و یا کود ذیل عین وظیفه را انجام میدهد:

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    FontDialog1.ShowDialog()
    RichTextBox1.Font = FontDialog1.Font
End Sub
```

پروگرام فرعي ذیل، متن انتخاب شده را به فونت "Arial" و سایز 16 تبدیل میکند:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    RichTextBox1.SelectionFont = New Font("Arial", 16)
End Sub
End Class
```

پروگرام فرعي ذیل، متن انتخاب شده را به رنگ سرخ تبدیل میکند:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    RichTextBox1.SelectionColor = Color.Red
End Sub
End Class
```

پروگرام فرعي ذیل، تمام متن داخل RichTextBox را Bold میسازد:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    RichTextBox1.Font = New Font(Me.Font, FontStyle.Bold)
End Sub
End Class
```

پروگرام فرعي ذیل، متن انتخاب شده را Bold میسازد:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    RichTextBox1.SelectionFont = New Font(Me.Font, FontStyle.Bold)
End Sub
End Class
```

پروگرام فرعي ذیل، متن انتخاب شده را Italic میسازد:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    RichTextBox1.SelectionFont = New Font(Me.Font, FontStyle.Italic)
End Sub
End Class
```

پروگرام فرعي ذیل، متن انتخاب شده را UnderLine میسازد:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    RichTextBox1.SelectionFont = New Font(Me.Font, FontStyle.Underline)
End Sub
End Class
```

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    RichTextBox2.Focus()
End Sub
End Class
```

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    RichTextBox1.Clear()
    RichTextBox1.SelectionFont = New Font("Arial", 24)
    RichTextBox1.SelectionBullet = True
    RichTextBox1.SelectionColor = Color.Red
    RichTextBox1.LoadFile("C:\abc.rtf")
    RichTextBox1.Find("Text")
    RichTextBox1.SaveFile("C:\abc.rtf", RichTextBoxStreamType.RichText)
End Sub
End Class
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    RichTextBox1.Find("abc")
    RichTextBox1.Select()
```

```
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    RichTextBox1.Find(TextBox1.Text)
    RichTextBox1.Select()
End Sub
```

کنترول TreeView

این کنترول عناصر را در یک درخت به شکل تسلسلی نشان میدهد. عناصر که نشان داده میشود نمونه های از کلاس بنام TreeNode اند. هر TreeNode دارای مجموعه از عناصر (Nodes) میباشد که دارای لست از TreeNode میباشد. بعضی خصوصیات این کنترول قرار ذیل است:

خاصیت CheckBox

خاصیت ImageList

این خاصیت icon های عنصر را حفظ میکند.

خاصیت Node

این خاصیت مجموعه TreeNodes را در کنترل نشان میدهد.

خاصیت SelectedNode**حادثه AfterSelect**

پروگرام ذیل با استفاده از کنترل TreeView فایل‌هایی که در کمپیوتر شما است آنرا نشان میدهد:

```
Imports System.Windows.Forms
Imports System.IO
Public Class FrmTreeViewDirectory
    Inherits Form
    Public Sub PopulateTreeView(ByVal directoryValue As String, _
        ByVal parentNode As TreeNode)
        Try
            Dim directoryArray As String() = _
                Directory.GetDirectories(directoryValue)
            If directoryArray.Length <> 0 Then
                Dim currentDirectory As String
                For Each currentDirectory In directoryArray
                    Dim myNode As TreeNode = _
                        New TreeNode(currentDirectory)
                    parentNode.Nodes.Add(myNode)
                    PopulateTreeView(currentDirectory, myNode)
                Next
            End If
            Catch unauthorized As UnauthorizedAccessException
                parentNode.Nodes.Add("Access Denied")
            End Try
        End Sub
        Private Sub FrmTreeViewDirectory_Load(ByVal sender As Object, _
            ByVal e As System.EventArgs) Handles MyBase.Load
            treDirectory.Nodes.Add("C:")
            PopulateTreeView("C:\", treDirectory.Nodes(0))
        End Sub
    End Class
```

کنترل WebBrowser

از این کنترل جهت نشان دادن صفحات وب در فورم ویژول بسیک استفاده میشود. در ختم استفاده از این کنترل باید میتود Dispose() را صدا نماییم تا منابع را رها کنیم. این کنترل دارای دارای میتودهای ذیل است:

```
Url
Navigate
GoBack
GoForward
```


GoHome

GoSearch

توسط این کنترلر محتوای که شما در ویژول بسیک ایجاد نمودید، یا از یک فایل است انرا نشان دهید. برای این کار خصوصیات DocumentText، و DocumentStream را استفاده کنید.

```
Private Sub toolStripTextBox1_KeyDown(ByVal sender As Object, ByVal e As KeyEventArgs) _
```

```
Handles toolStripTextBox1.KeyDown
```

```
If (e.KeyCode = Keys.Enter) Then
```

```
    Navigate(toolStripTextBox1.Text)
```

```
End If
```

```
End Sub
```

```
Private Sub goButton_Click(ByVal sender As Object, ByVal e As EventArgs)
```

```
Handles goButton.Click
```

```
    Navigate(toolStripTextBox1.Text)
```

```
End Sub
```

```
Private Sub Navigate(ByVal address As String)
```

```
    If String.IsNullOrEmpty(address) Then Return
```

```
    If address.Equals("about:blank") Then Return
```

```
    If Not address.StartsWith("http://") And _
```

```
        Not address.StartsWith("https://") Then
```

```
        address = "http://" & address
```

```
    End If
```

```
    Try
```

```
        webBrowser1.Navigate(New Uri(address))
```

```
    Catch ex As System.UriFormatException
```

```
        Return
```

```
    End Try
```

```
End Sub
```

```
Private Sub webBrowser1_Navigated(ByVal sender As Object, _
```

```
    ByVal e As WebBrowserNavigatedEventArgs) _
```

```
Handles webBrowser1.Navigated
```

```
    toolStripTextBox1.Text = webBrowser1.Url.ToString()
```

```
End Sub
```

کنترولهایی باکس های مکالماتی ویندوز

اکثر پروگرامها برای باز کردن و یا ذخیره نمودن فایل ها از باکس های مکالماتی استفاده میکند که از طریق آن میتوانیم نام، محل، فارمت و غیره خصوصیات فایل را تعیین نمائیم. سیستم عامل ویندوز این باکس های مکالماتی را دارا بوده و پروگرامهای آفس که در سیستم عامل ویندوز کار میکند مانند مایکروسافت اکسل (MS-Excel)، مایکروسافت ورد (MS-Word)، مایکروسافت پاورپاینت (MS-PowerPoint) و غیره از همین باکسهای مکالماتی ویندوز استفاده میکند. پروگرامهایی که شما

در ویژول بسیک ایجاد میکنید هم میتواند از همین باکس ها استفاده نمایند. و این کار در حقیقت مفهوم استفاده دوباره (Reusability) است که در آن آبجکت های از قبل ایجاد شده به تکرار استفاده شده میتواند. هیچ پروگرام ضرورت ندارد تا این آبجکت ها را در خود داشته باشد. بلکه میتواند در اثنای ضرورت آنرا استفاده نماید. و در حقیقت استفاده دوباره هدف اساسی پروگرام نویسی شی گرا است.

در ویژول بسیک 6 از کنترول CommonDialog جهت آوردن با کس های مکالماتی (Dialog Box) باز کردن فایل، ذخیره کردن فایل، چاپ نمودن، فونت، رنگ و کمک (Help) استفاده میشود. در ویژول بسیک 2005 چندین کنترول دیگر جایگزین این کنترول گردیده است. این کنترولها عبارت اند از:

PrintDialog، OpenFileDialog، SaveFileDialog، ColorDialog، FontDialog، و PrintDialog میباشد.

باکس مکالماتی Save

AddExtension: خاصیت منطقی بوده که اضافه شدن و یا عدم اضافه شدن Extension به نام فایل را تعیین میکند. البته در صورتیکه User انرا تایپ نکند.

CheckExist: این خاصیت یک خاصیت منطقی بوده و اگر قیمت آن True باشد، وجود فایل را به عین نام تست میکند و از User میخواهد تا فایل جدید به جای فایل قبلی

FileName: نام فایل که باید باز شود.

InitialDirectory: فولدر که هنگام استفاده از باکس مکالماتی از قبل انتخاب باشد.

DefaultExt: توسعه از قبل تعیین شده (Default) برای فایل.

Filter: فلتر که توسط آن توسعه های دلخوا خود را تعیین کرده میتوانیم.

FilterIndex

ReadOnlyChecked: خاصیت منطقی (Boolean) بوده که انتخاب و یا عدم انتخاب حالت Read Only فایل را نشان میدهد.

ShowReadOnly: این خاصیت هم منطقی بوده و وجود و یا عدم وجود اختیار انتخاب Read Only را تعیین میکند.

Reset: این میتود تمام حالات کنترول را به حالت Default می آورد.

ShowDialog: این میتود باکس مکالماتی را در روی صفحه نشام میدهد.

Title: عنوان را برای باکس مکالماتی تعیین میکند.

ShowHelp: خاصیت منطقی بوده و وجود و یا عدم وجود دکمه Help را تعیین میکند.

ValidateName: خاصیت منطقی بوده و جهت تصحیح و یا عدم تصحیح نام فایل را تعیین میکند. مثال ذیل استفاده از باکس مکالماتی Save را نشان میدهد:

```
Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
    SaveFileDialog1.ShowDialog()
    RichTextBox1.SaveFile(SaveFileDialog1.FileName)
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    SaveFileDialog1.Filter = "txt files (*.txt)|*.txt|MS Word file (*.doc)|*.doc |
All files (*.*)|*.*"
    SaveFileDialog1.ShowDialog()
    RichTextBox1.SaveFile(SaveFileDialog1.FileName)
End Sub
```

```

Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    RichTextBox1.SelectionBackColor = Color.Red
    RichTextBox1.SelectionColor = Color.Blue
    RichTextBox1.SelectionFont = New Font("Arial", 16)
    RichTextBox1.SelectionAlignment = HorizontalAlignment.Center
    RichTextBox1.SelectionBullet = True
    RichTextBox1.SelectionCharOffset = -12
    RichTextBox1.SelectionHangingIndent = 6
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    ColorDialog1.ShowDialog()
    RichTextBox1.ForeColor = ColorDialog1.Color
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    FontDialog1.ShowDialog()
    RichTextBox1.Font = FontDialog1.Font
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    OpenFileDialog1.ShowDialog()
    RichTextBox1.LoadFile(OpenFileDialog1.FileName)
End Sub

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
    SaveFileDialog1.ShowDialog()
    RichTextBox1.SaveFile(SaveFileDialog1.FileName)
End Sub
End Class

```

مثال ذیل استفاده از SaveFileDialog را نشان میدهد:

```

Imports System.Windows.Forms
Public Class Form1
    Inherits System.Windows.Forms.Form
    Dim userInput As New System.IO.MemoryStream
Private Sub Button1_Click(ByVal sender As System.Object, _

```

```

ByVal e As System.EventArgs) Handles Button1.Click
RichTextBox1.SaveFile(userInput, RichTextBoxStreamType.PlainText)
userInput.WriteByte(13)
userInput.Position = 0
RichTextBox2.LoadFile(userInput, RichTextBoxStreamType.PlainText)
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button2.Click
SaveFileDialog1.CreatePrompt = True
SaveFileDialog1.OverwritePrompt = True
SaveFileDialog1.FileName = "myText"
SaveFileDialog1.DefaultExt = ".txt"
SaveFileDialog1.Filter = "Text files (*.txt)|*.txt"
SaveFileDialog1.InitialDirectory = "c:\"
Dim result As DialogResult = SaveFileDialog1.ShowDialog()
Dim fileStream As System.IO.Stream
If (result = DialogResult.OK) Then
fileStream = SaveFileDialog1.OpenFile()
userInput.Position = 0
userInput.WriteTo(fileStream)
fileStream.Close()
End If
End Sub
End Class

```

دستور هاي ذيل براي نشان دادن باکس مکالماتي Save استفاده ميشود:

```

SaveFileDialog1.InitialDirectory = My.Application.Info.DirectoryPath
SaveFileDialog1.ShowDialog()

```

دستور هاي ذيل براي نشان دادن باکس مکالماتي Print استفاده ميشود:

```

PrintDocument1.DocumentName = My.Application.Info.DirectoryPath _
& "MyFile.txt"
PrintDialog1.Document = PrintDocument1
PrintDialog1.ShowDialog()

```

خاصيت Filter:

اين خاصيت سلسله که نوع فایل را در اثنای ذخيره و بازکردن مشخص میکند تعين مينمايد.

```

Imports System.IO
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim myStream As Stream
Dim openFileDialog1 As New OpenFileDialog()
openFileDialog1.InitialDirectory = "c:\"

```

```

openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
openFileDialog1.FilterIndex = 2
openFileDialog1.RestoreDirectory = True
If openFileDialog1.ShowDialog() = DialogResult.OK Then
    myStream = openFileDialog1.OpenFile()
    If Not (myStream Is Nothing) Then
        myStream.Close()
    End If
End If
RichTextBox1.Text = openFileDialog1.FileName
End Sub
End Class

```

باکس مکالماتی Open

FileName: نام فایل که باید باز شود.
InitialDirectory: فولدر که هنگام استفاده از باکس مکالماتی از قبل انتخاب باشد.
DefaultExt: توسعه از قبل تعیین شده (Default) برای فایل.
Filter: فیلتر که توسط آن توسعه های دلخوا خود را تعیین کرده می‌توانیم.
FilterIndex:
ReadOnlyChecked: خاصیت منطقی (Boolean) بوده که انتخاب و یا عدم انتخاب حالت Read Only فایل را نشان می‌دهد.
ShowReadOnly: این خاصیت هم منطقی بوده و وجود و یا عدم وجود اختیار انتخاب Read Only را تعیین می‌کند.
Reset: این می‌تود تمام حالات کنترل را به حالت Default می‌آورد.
ShowDialog: این می‌تود باکس مکالماتی را در روی صفحه نشام می‌دهد.
Title: عنوان را برای باکس مکالماتی تعیین می‌کند.
ShowHelp: خاصیت منطقی بوده و وجود و یا عدم وجود دکمه Help را تعیین می‌کند.
ValidateName: خاصیت منطقی بوده و جهت تصحیح و یا عدم تصحیح نام فایل را تعیین می‌کند.

مثال ذیل پروگرام مکمل جهت استفاده از باکس مکالماتی Open را نشان می‌دهد:

```

Private Sub btnOpen_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOpen.Click
    OpenFileDialog1.ShowDialog()
    RichTextBox1.LoadFile(OpenFileDialog1.FileName)
End Sub

```

کنترل ColorDialog

این کنترل جهت نشان دادن و استفاده از رنگها به کار برده میشود. مثال ذیل استفاده از این کنترل را نشان می‌دهد:

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim MyDialog As New ColorDialog()

```

```

MyDialog.AllowFullOpen = False
MyDialog.ShowHelp = True
MyDialog.Color = TextBox1.ForeColor
If (MyDialog.ShowDialog() = Windows.Forms.DialogResult.OK) Then
    TextBox1.ForeColor = MyDialog.Color
End If
End Sub

```

دیزاین مینوها

مینوها جزء اساسی پروگرام های ویندوز (windows) بوده که از طریق آن استفاده کننده (user) با پروگرام ها ارتباط میگیرد. در یک پروگرام اختیارات مشابه (باهم مرتبط) در یک مینو باهم جمع گردیده و مورد استفاده قرار میگیرد. در ویژول بسیک اول باید چوکات مینو ها (مینو Bar و مینوها و اختیارات آن) را دیزاین نموده و بعد از آن اختیارات آنرا پروگرام نمائیم . البته کار مهم ، پروگرام نمودن اختیارات مینوها میباشد. چوکات مینوها را به کمک کنترل MenuStrip طور ذیل دیزاین کرده می توانیم :

1. یک فورم جدید را باز نمائید.
 2. از Tool Bars کنترل MenuStrip را بالای فورم قرار دهید. بعد از آن میتوانیم مینوها و اختیارات آنرا دیزاین نمائیم.
- نوت: بعد از تکمیل نمودن مینوها نیز می توانیم اختیارات جدید را به مینو ها اضافه نمائیم .
خصوصیات و حادثات کنترل مینو قرار ذیل است:

خصوصیت Checked

این خصوصیت علامه Tick را در پهلوی اختیار مینو اضافه میکند در صورت که این خصوصیت انتخاب شود.

خصوصیت Index

ترتیب اختیار را در لست مینو نشان میدهد. مثلا اگر قیمت آن صفر شود آن اختیار در اول مینو قرار میگیرد.

خصوصیت MenuItem

خصوصیت RadioCheck

خصوصیت Shortcut

خصوصیت ShowShortcut

خصوصیت Text

پروگرام نمودن اختیارات مینوها

جهت پروگرام نمودن اختیارات مینوها، بالای اختیار مورد نظر دو بارکلیک نموده و بعد از آن کد مناسب برای همان اختیار را مینویسیم . در اینجا جهت فهم بهتر مینو ها، پروژه دیزاین یک Editor متن را در نظر می گیریم. این Editor دارای سه مینو بنام های File، Edit و Format بوده و هر مینو دارای چندین اختیار میباشد. کار اساسی پروگرام نمودن این اختیارات است.

پروگرام فرعي ذیل فورم را بسته میکند. این پروگرام فرعي میتواند به اختیار Exit مینوی فایل بسته گردد:

```
Public Class Form1
Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click
    Me.Close()
End Sub
End Class
```

پروگرام فرعي ذیل عملیات Cut، Copy و Paste را انجام میدهد:

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Clipboard.SetText(rtb.SelectedText)
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Clipboard.SetText(rtb.SelectedText)
    rtb.SelectedText = " "
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    rtb.SelectedText = Clipboard.GetText
End Sub
End Class
```

```
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    If RichTextBox1.SelectionFont.Bold Then
        RichTextBox1.SelectionFont = New Font(Me.Font,
FontStyle.Regular)
    Else
        RichTextBox1.SelectionFont = New Font(Me.Font, FontStyle.Bold)
    End If
End Sub
End Class
```

اضافه نمودن مینو های قرینه (Context Menu)

عبارت از مینوهای است که در اثر فشار دادن کلید طرف راست موس (Right Click) ظاهر میگردد. قابل تذکر است که این مینو میتواند با فشار دادن دکمه چپ موس فعال شود و این مربوط به عیار سازی موس است که در صفحه کنترل (Control Panel) ویندوز عیار میگردد.

برای اضافه ساختن مینوی قرینه کنترول ContextMenuStrip را به فورم اضافه کنید. بعد از آن اختیارات آنرا تعریف و پروگرام نمائید. توسط خاصیت ContextMenuStrip کنترولها میتوانند مینوی قرینه را به یک کنترول مربوط سازید تا در صورت فشار دادن دکمه طرف راست موس بالای همان کنترول مینوی قرینه ظاهر شود. دستور ذیل از طریق کود یک مینوی قرینه را به کنترول فورم مربوط میسازد:

```
Public class form1
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Me.ContextMenuStrip = ContextMenuStrip1
End Sub
End class
```