

مقدمه:

امروزه علم کامپیوتر پیشرفت شگرفی در کلیه علوم و صنایع به وجود آورده است و با پیشرفت این علوم مسائل مربوط به کامپیوتر نیز پیشرفت شگرفی داشته است از جمله می توان به پیشرفت در بانک های اطلاعاتی اشاره نمود که از جدیدترین و مهمترین آنها می توان به Visual basic اشاره نمود. در اینجا سعی شده است مطالبی در جهت معرفی این نرم افزار تقدیم گردد.

فصل اول

نرم افزار ویژوال بیسیک

برنامه نویسی با Visual Basic (آخرین ویرایش Visual Basic) می تواند کار پرزحمت و خسته کننده برنامه نویسی را به یک تفریح تبدیل کند.

محبوبیت Visual Basic، علاوه بر گرافیکی و ساده بودن، از آنجا ناشی می شود که زبانی اینترپرتری (interpreter) و کامپایلری (Compiler) است. دید که زبانهای اینترپرتری

از نظر سهولت تولید و توسعه برنامه چقدر به برنامه نویسی کمک می کنند و از طرف دیگر

برنامه های کامپایل شده بنحو چشمگیری سریعتر از برنامه های اینترپرتری هستند. با ترکیب

این دو ویژگی Visual Basic را روانه بازار کرد، بسیاری از منتقدان شکست آن و مرگ

خاندان بیسیک را پیش بینی می کردند. این مخالفان می گفتند که بیسیک اساساً یک زبان جدی

نیست و ساختار آن (برخلاف C و پاسکال) به درد محیط ویندوز نمی خورد اما

میکروسافت با ارائه Visual Basic درسهای زیادی به جامعه برنامه نویسان داد:

- زبانی مانند بیسیک می تواند قابل فهم و قوی باشد.
- Visual Basic بسته به نیاز برنامه نویس می تواند کامپایلری یا اینترپرتری باشد.
- Visual Basic نه تنها فراموش نشد بلکه تبدیل به محبوبترین زبان برنامه نویسی دنیا

شد.

- Visual Basic در چندین مدل مختلف عرضه می شود.
- مدل فوق العاده : محیط برنامه نویسی مناسب کارهای شبکه و برنامه های توزیع شده است.
- مدل آموزشی: مجموعه ای از ابزارهای استاندارد و هر آنچه برای شروع برنامه نویسی بدان نیاز دارید.
- مدل استودیوی ویژوال در این مدل Visual Basic به همراه چند زبان ویژوال دیگر (مانند ویژوال ++ و (ویژوال ++) در یک بسته نرم افزاری گرد آمده اند تا آموختن آنها متضمن کمترین زحمت برای برنامه نویسان باشد.

ویژوال بیسیک و فرآیند برنامه نویسی

برنامه نویسی یک کار حرفه ای (یا حداقل نیمه حرفه ای) است که قواعد خاص خود را دارد. البته کاربران کامپیوتر کمتر درگیر مشکلات برنامه نویسی می شوند ولی اگر فضای روزگار شمارا وارد این وادی کرده است، سعی کنیم اصول کلی ذیل را همواره مد نظر داشته باشیم:

- ۱- درباره برنامه تصمیم بگیرید و یک طرح کلی بریزید.
- ۲- قسمت های ظاهری برنامه (پنجره ها، منوها و هر آنچه را که کاربر می بیند) را بوجود آوریم.

۳- با نوشتن کد عناصر برنامه را به هم مرتبط کنیم.

۴- برنامه را تست کرده و اشکالات آن (که در قاموس برنامه نویسی به آنها باگ گفته می شود) را یافته و برطرف کنیم.

۵- برنامه تست شده را کامپایل کنیم.

خصلت بصری برنامه نویسی Visual Basic مراحل طراحی برنامه را بسیار ساده و شیرین کرده است چون جای قلم و کاغذ می توانیم مراحل اول و دوم را روی صفحه کامپیوتر انجام دهید.

اولین برنامه:

اگر با سایر برنامه های ویندوز کار کرده باشیم احتمالاً با جادوگر (wizard) ها آشنائید، جادوگر برنامه ای است که قدم به قدم شما را در فرآیند طراحی همراهی کرده و در پایان الگویی شایسته از آنچه در ذهن دارید برایتان خواهد ساخت. Visual Basic هم برای ایجاد برنامه ها دارای جادوگر برنامه (application wizard) است. جادوگر برنامه در پایان اسکلت یک برنامه کامل و عملیاتی را برایتان می سازد اما اکثر برنامه نویسان ترجیح می دهند خودشان برنامه را از صفر شروع کنند، بسیاری نیز از کپی برنامه های قدیمیشان استفاده کرده و با تغییر دادن در آن برنامه جدیدی می سازند. بعد از آن که کمی Visual Basic تجربه کسب کردید احتمالاً ترجیح خواهیم داد از جادوگر برنامه استفاده نکنیم.

برای شروع کار در این قسمت برنامه‌ای با استفاده از جادوگر برنامه ایجاد خواهیم کرد. این برنامه کار چندانی انجام نمی‌دهد و فقط یک پوسته (Shell) است. بلافاصله بعد از شروع Visual Basic جادوگر برنامه آماده انجام وظیفه است، گفتگوی پروژه جدید

(New project) اولین چیزی است که بعد از اجرای Visual Basic خواهیم دید.

- New اجازه می‌دهد تا جادوگرهای مختلفی را به کمک احضار کرده یا برنامه را از صفر شروع کنیم.

- Existing اجازه می‌دهد تا پروژه‌های موجود را باز کنیم.

- Recent لیستی از آخرین پروژه‌های باز شده یا ایجاد شده را نشان می‌دهد.

برای بازکردن گفتگوی پروژه جدید از File New Project هم می‌توانیم استفاده کنیم، اما در این حالت برگه‌های Existing و Recent را نخواهیم دید.

پروژه (Project) عبارتست از مجموعه فایل‌های یک برنامه هر برنامه می‌تواند چندین فایل داشته باشید. برخی از این فایل‌ها حاوی کد برنامه، برخی دیگر حاوی مشخصات ظاهری برنامه و برخی دیگر (احتمالاً) شامل کدهای ارتباط برنامه با سیستم عامل یا برنامه‌های دیگر هستند. اگر میل ندارید هر بار در شروع Visual Basic گفتگوی پروژه جدید را ببینید، می‌توانیم جعبه یک Don't show this dialog box in the future را علامت بزنید.

اگر روی آیکون Visual Basic Application Wizard (در برگه New) کلیک کنیم جادوگر برنامه کارش را شروع خواهد کرد. صفحه اول جادوگر فقط یک خوشامد گویی و اعلام شروع بکار جادوگر است. مانند تمام جادوگرها وقتی کارتان در یک صفحه تمام شد، باید با کلیک کردن دکمه Next به صفحه بعد بروید.

گزینه‌های این صفحه عبارتند از:

- واسطه چندی سندی (MID): این نوع واسطه اجازه می‌دهد تا در آن واحد چندین

سند (document) را در برنامه باز کنیم در حقیقت باین واسطه می‌توانیم با چندین

مجموعه داده در پنجره‌های جداگانه کار کنیم. به هر پنجره سند پنجره فرزند

(Child window) گفته می‌شود.

- واسطه تک سندی (SDI) : با این واسطه در هر لحظه فقط یک پنجره سند می‌تواند باز

باشد. احتمالاً اکثر برنامه‌هایی که خواهیم نوشت از این نوعند.

- سبک کاوشگر: با این واسطه می‌توانیم برنامه‌هایی شبیه کاوشگر ویندوز

(windows Explorer) بوجود آوریم که در آن سرفصل مطالب در قاب سمت چپ و

توضیحات سرفصل انتخاب شده در قاب سمت راست نمایش داده می‌شوند.

با انتخاب هر گزینه نمونه کوچکی از ظاهر برنامه در قسمت چپ - بالای صفحه جادوگر

ظاهر خواهد شد. برنامه اول ما از واسطه SDI استفاده می‌کند. در همین صفحه نام پروژه را

هم می‌توانیم انتخاب کنیم از آنجایی که نام پیش فرض، Project، چندان با مسمای نیست، آن را به First App تغییر داده (توجه داشته باشیم که فاصله در نام پروژه مجاز نیست) و Next را کلیک کنیم تا صفحه بعدی جادوگر ظاهر شود.

جادوگر برنامه منوهای را این صفحه انتخاب کنیم به برنامه اضافه خواهد کرد. این منوها در میان اغلب برنامه‌های ویندوز یکسان هستند، منوهای مانند File و Edit.

آیتم‌های موجود در هر منو هم قابل انتخاب است، مثلاً، منوی File میتواند دارای آیتم New و Open و Close باشد. علامت & در کنار هر حرف نشان دهنده کلید دسترسی

سریع آن آیتم است، به عبارت دیگر، &New در هنگام اجرای برنامه به شکل New ظاهر خواهد شد و کاربر می‌تواند با زدن همزمان کلیدهای Alt + N آن آیتم را فعال کند. اما

اگر بخواهیم خود علامت & را در منو داشته باشیم، چه باید بکنیم؟ ساده است، & را دوبار پشت سر هم بنویسید، مثلاً، A&&B هنگام اجرای برنامه به صورت A&B ظاهر خواهد

شد. برای مثال ما، منوها را همان گونه که هست رها کرده و با یک کلیک کردن Next به صفحه بعد جادوگر بروید.

صفحه بعد محل انتخاب میله‌های ابزار (toolbars) برنامه است. همانطور که می‌بینید، جادوگر برنامه در این زمینه هم سنگ تمام گذاشته و تقریباً هر آنچه را که احتمالاً نیاز

دارید در اختیارتان گذاشته است. قاب سمت چپ حاوی تمام ابزارهای موجود و قاب

سمت راست حاوی ابزارهای انتخاب شده برای برنامه هستند. با کلیک کردن Next این صفحه را هم پشت سر بگذارید.

صفحه بعد جادوگر محل انتخاب منابع برنامه (مانند فایل های چند زبانی، برای بین المللی کردن برنامه ها) است. در این برنامه به هیچ منبع خارجی نیاز نداریم بنابراین گزینه No را علامت زده و به صفحه بعد بروید. صفحه بعد صفحه اتصال به اینترنت است. اگر در این قسمت گزینه Yes را انتخاب کنیم، جادوگر برنامه یک کاوشگر اینترنت تمام عیار به برنامه تان اضافه خواهد کرد. این کاوشگر درست مانند سایر کاوشگرها می تواند شما را از دورن برنامه به هر صفحه وبی که مایلند متصل کند. بدون انتخاب گزینه اتصال به اینترنت به صفحه بعد بروید.

در صفحه بعد می توانیم گزینه هایی ذیل را به برنامه اضافه کنیم:

- صفحه معرفی برنامه (Splash screen): صفحه ای که در ابتدای شروع برنامه برای مدتی کوتاه ظاهر شده و آن را معرفی می کند.

- گفتگوی ورود (Login dialog): محلی برای گرفتن نام کاربر، کلمه عبور و سایر اقدامات امنیتی.

- گفتگوی گزینه ها (Options dialog): صفحه ای چند برگه ای که کاربر می تواند برنامه را از آنجا تنظیم کند.

- پنجره About : گفتگویی که با کلیک کردن Help About ظاهر شده و توضیحاتی درباره برنامه خواهد داد.

در قسمت From Template ، چندین فرم (که در قاموس Visual Basic همان پنجره است) دیگر ، از جمله ODBC login (برای ارتباط با پایگاه‌های داده پیشرفته) و Tip of the Day (نکته روز)، وجود دارد که می‌توانیم از آنها استفاده کنیم.

به این برنامه فقط پنجره About اضافه کرده و Next را کلیک کنیم. در صفحه ارتباط با پایگاه‌های داده بدون هیچ اقدامی فقط Next را کلیک کنیم در پایان هم با کلیک کردن Finish ایجاد برنامه را به پایان برسانید، با کلیک کردن دکمه View Report می‌توانیم خلاصه‌ای از پروژه ایجاد شده را مشاهده کنیم.

تا این جا بدون آنکه چیزی از برنامه‌نویسی Visual Basic بدانید، یک برنامه تمام و کمال و عملیاتی بوجود آورده‌اید که حتی در همین حالت هم می‌توانیم آن را اجرا کنیم، با کلیک کردن آیتم Run Start (یا معادل آن، F5) برنامه را اجرا کنیم.

این برنامه که با کمک جادوگر برنامه آن را ساختید کار چندانی انجام نمی‌دهد و فقط یک پوسته است.

- برنامه یک پنجره استاندارد دارد که اندازه آن را می‌توان تغییر داد، نام پروژه، Firt App، در میله عنوان (title bar) برنامه دیده می‌شود.

- برنامه یک میله وضعیت (Status bar) دارد که تاریخ و وقت را نشان می دهد. این میله را می توان از منوی View ظاهر ساخت و یا پنهان کرد.
- منوهای برنامه نیاز به کدنویسی دارند، ولی برخی از آنها (مانند Help About) بدون هیچگونه کوششی سیستم (از قبیل مقدار حافظه و منابع آزاد سیستم) ارائه خواهد کرد.
- برنامه یک میله ابزار استاندارد دارد، که از منوی View می توان آن را ظاهر یا ضنهان کرد.
- جادوگر برنامه یک الگوی کامل برای شما بوجود آورده که می توانیم آن را مطابق نیاز خود تغییر دهید. فعلاً با کلیک کردن File Exit برنامه را ببندید، در پاسخ به ذخیره کردن پروژه پاسخ NO بدهیم، چون هر وقت خواستید جادوگر برنامه دوباره آنرا برایتان خواهد ساخت.

برنامه نویسی رویدادگر

این یک برنامه ویندوز است، این پنجره دارای انواع متعددی از کنترل های ویندوز است، کنترلهایی از قبلی دکمه فرمان (Command button)، جعبه چک (Check box) و میله لغزنده (Scroll bar) ویندوز دارای کنترل های متعددی است که می توانیم در محیط برنامه نویسی Visual Basic از آنها استفاده کرده و به برنامه اضافه کنیم.

برنامه‌های ویندوز ذاتاً به چنین کنترل‌هایی نیاز دارند چون برخلاف برنامه‌های قدیمی باید به رویداد (Event) ها پاسخ دهند. رویداد می‌تواند از این کنترل‌ها یا هر جای دیگر (کی‌برد یا ماوس یا ساعت سیستم) منشاء بگیرد. رویدادها می‌توانند بدون هیچ ترتیبی اتفاق افتند، چون از قبل نمی‌توان پیش‌بینی کرد که کاربر چگونه با کنترل‌ها کار خواهد کرد. برای مقابله با چنین وضعیتی باید خود را برای برنامه‌نویسی رویدادگرا (event driven) آماده کنیم. اساساً هر عملی که در حین اجرای برنامه روی دهد یک رویداد است. ویندوز یک سیستم عامل چند وظیفگی (multitasking) است یعنی در آن واحد چندین برنامه می‌توانند در کنار هم اجرا شوند و در این حالت ویندوز باید رویدادهای هر برنامه را به همان برنامه بفرستد. هنگام برنامه‌نویسی، تعدادی از کنترل‌های Visual Basic را روی یک فرم (پنجره) قرار می‌دهید و خاص ظاهری آن را مطابق نیاز خود تنظیم می‌کنیم. هر کنترل دارای ویژگی‌های ظاهری و عملکردی متعددی است که خاصیت (Property) نامیده می‌شود. در حقیقت این خواص هستند که کنترل‌ها را از یکدیگر متمایز می‌کند. تمام این کنترل‌ها از نوع دکمه فرمان هستند که فقط خواص متفاوتی برای آنهاست شده است. هر کنترل به رویدادهای خاص خود پاسخ می‌دهد، مثلاً دکمه فرمان دارای رویداد کلیک شدن است که هرگاه کاربر روی دکمه کلیک کند (یا Enter را بزند) روی خواهد داد و برنامه‌نویس باید برای این رویداد کد لازم را بنویسد. برنامه‌های رویدادگرا برخلاف

برنامه‌های سابق که لیست بلند بالایی از کد بودند، از قطعات کوچک کد که هر قطعه مجری یک رویداد خاص است، تشکیل می‌شوند. این قطعات کد در حالت عادی هیچ کاری نمی‌کنند و منتظر بروز رویداد مربوط به خود هستند تا وارد صحنه شوند و وظیفه‌شان را انجام دهند.

پنجره پروژه جدید

هرگاه Visual Basic شروع می‌شود پنجره‌ای بنام New Project (پروژه جدید) ظاهر می‌شود و به شما امکان می‌دهد تا نوع برنامه خود را انتخاب کنیم.

اغلب برنامه‌های که خواهیم نوشت از نوع Standard EXE خواهند بود. می‌گویید که اگر برنامه را کامپایل کنیم یک فایل exe تولید خواهد شد.

برنامه‌های Standard exe را به صورت اینترنت‌تری هم می‌توان اجرا کرد.

میله ابزار

میله ابزار Visual Basic درست زیر منوی آن قرار دارد و خود از چهار میله ابزار کوچکتر تشکیل شده است.

- دیباگ. این میله ابزار هنگام دیباگ کردن برنامه‌ها فعال خواهد شد.
- ادیت. از این میله ابزار برای ادیت کردن کدهای Visual Basic استفاده خواهیم کرد.
- ادیتورفرم. با این میله ابزار می‌توانیم اشیاء روی فرم را کنترل کنیم.

- استاندارد. میله ابزار پیش فرض Visual Basic که در ذیل منو دیده می شود. برای ظاهر یا پنهان کردن این میله ها می توانیم از View Toolbars استفاده کنیم. هر میله ابزار دارای دکمه های متعددی است که هر یک راه میانبری برای اجرای فرمانهای Visual Basic هستند و زحمت جستجو در منوهای Visual Basic را از دوش شما برمی دارند. در تمام این میله های ابزار چند دکمه وجود دارد که بطور مرتب مورد استفاده قرار می گیرند و بسیار مفید هستند. البته هیچ نیازی نیست که میله های ابزار Visual Basic می توانند چسبیده (docked) یا شناور (float) باشند، یعنی هر میله ابزار می تواند بطور ثابت به سایر قسمتهای محیط Visual Basic بچسبد یا بصورت شناور (در هر قسمت که برنامه نویس میل داشته باشد) قرار گیرد.

- جعبه ابزار

- پنجره جعبه ابزار (Toolbox) با میله ابزار تفاوت دارد. این پنجره که به آن مختصراً جعبه ابزار گفته می شود، انباره کنترل های Visual Basic است. مسئله جاب در مورد توجه جعبه ابزار این است که این جعبه هرگز خالی نمی شود. یعنی هر چقدر که از ابزارهای آن استفاده کنیم باز هم این جعبه ابزار می تواند ابزارهای دیگری در اختیاران بگذارد. در حقیقت جعبه ابزارها را خلق کرده و روی فرم قرار می دهد.

پنجره فرم

پنجره فرم جاییست که کنترل‌های برنامه روی آن قرار می‌گیرند. البته یک برنامه می‌تواند چندین فرم داشته باشد (MDI) که می‌توانیم آنها را در محیط برنامه‌نویسی Visual Basic همزمان مشاهده کنیم البته در هر لحظه فقط یکی از فرمها فعالی است که رنگ میله عنوان این فرم با سایر فرمها تفاوت خواهد داشت.

پنجره طراحی فرم

پنجره طراحی فرم پنجره جالبی است که ارتباط نزدیکی با فرم‌های برنامه دارد و ظاهر و موقعیت فرم‌های برنامه را هنگام اجرای آن نشان می‌دهد. برای جابجا کردن فرمها روی صفحه کامپیوتر می‌توانیم از پنجره طراحی فرم استفاده کنیم. اگر برنامه چند فرم داشته باشد، تمام آنها را می‌توانیم در این پنجره مشاهده کرده و موقعیت نهایی آن را تعیین کنیم. (بعدها خواهیم دید که محل و موقعیت فرمها روی صفحه مانیتور را از طریق کدنویسی هم می‌توان کنترل کرد).

پنجره پروژه

پنجره پروژه محل کنترل اجزا برنامه است. برنامه می‌تواند دارای چندین فایل باشد و تعداد این فایل‌ها می‌تواند به سرعت بالا رود. پنجره پروژه برای مدیریت این فایل‌هاست و

می‌توانیم به کمک آن هر فایل را به ناحیه کاری Visual Basic آورده و روی آن کار کنیم.
(به پنجره پروژه، کاوشگر پروژه هم می‌گویند.)

پنجره پروژه برای نمایش اجزا برنامه و روابط آنها از ساختار درختی استفاده می‌کند. در Visual Basic به هر شی (از قبیل فرم یا مدول) می‌توان نامی داد و هر آیتم با نامی دیگر روی دیسک ذخیره خواهد شد. نام فایل با نام پروژه متفاوت است (نام پروژه، بر خلاف نام فایل، پسوند ندارد) و در پراپرتیز نمایش داده می‌شود بدین ترتیب در پنجره پروژه علاوه برنام پروژه سه دکمه دیده می‌شود:

- دکمه Visual Basic: که با آن می‌توانیم پنجره کد آیتم انتخاب شده را مشاهده کنیم.
 - دکمه Visual object: که با آن می‌توانیم ساختار نمایش پنجره پروژه را تغییر دهید.
 - دکمه Visual folders: که با آن می‌توانیم ساختار نمایش پنجره پروژه را تغییر دهید.
- در پنجره پروژه اشیا مختلفی می‌توانند ظاهر شوند:
- پروژه . یک برنامه می‌تواند دارای چندین پروژه باشد (مانند کنترل‌های Activex)
 - پسوند نام فایل پروژه همیشه VBP است.
 - فرم. هر پروژه می‌تواند فرم‌های متعددی داشته باشد. پسوند نام فایل فرم‌ها همیشه FRM است.

- مدول. کدها و روتین های عمومی برنامه، که به هیچ کنترل خاصی مربوط نیستند. باید در مدول (module) ها قرار داده شوند. از یک مدول به دلیل خصلت عمومی آنها می توان در برنامه های متعددی استفاده کرد. پسوند نام فایل مدول ها BAS است.
- مدول کلاس. محل تعریف کدهایی که اشیا پروژه را تعریف می کنند پسوند نام فایل مدولهای کلاس (Class module) همیشه CLS است.
- کنترل کاربر. کنترل های اکتیوایکس اضافه شده به پروژه فایل های اکتیوایکس دارای پسوند OCX هستند.
- سند کاربر. سند کاربر سندی است که مشخصات پروژه را توضیح دهید. پسوند نام فایل سند کاربر (User document) همیشه DOB است.
- صفحه خواص. صفحات خواص برای تعریف کنترل های پروژه بکار می روند (مانند صفحاتی که در گفتگوهای برگه دار دیده می شود). پسوند نام فایل صفحه خواص (Property page) همیشه ح ش ل است.
- پنجره پروژه می تواند دارای اشیا دیگری از قبیل فایل های منبع (Resource file) و سندهای دیگر باشد.

پنجره خواص

یک فرم می تواند دارای تعداد زیادی کنترل باشد. وقتی روی یک فرم کنترلی را انتخاب می کنیم، پنجره خواص تمام خواص آن را نشان خواهد داد.

خواص هر کنترل هنگامی که روی فرم قرار داده می شود دارای مقادیر پیش فرض است، اما می توان آنها را به مقادیر دلخواه ست کرد.

چون هر خصایت دارای یک نام است، مقدار آن را از طریق کد هم می توان دستکاری کرد.

برای مثال Visual Basic به اولین دکمه فرمان که روی فرم برنامه قرار دهید نام

Command 1 خواهد داد. بنابراین، مقدار خاصیت Name این کنترل Command 1 است.

باید سعی کنیم همیشه به کنترل ها نامی متناسب با عملکردشان بدهیم روش استاندارد اضافه

کردن یک پیشوند سه حرفی (برای تعیین نوع کنترل) به اول نام آن است. با این روش هر

کس با دیدن نام کنترل می تواند نوع و عملکرد آن را تشخیص دهد. در جدول صفحه بعد

پیشوندهای استاندارد را مشاهده می کنیم.

کمک

Visual Basic دارای انواع مختلفی از کمک در سطوح مختلف است که می توانیم هر لحظه

از آنها استفاده کنیم.

پیشوندهای استاندارد برای اشیای Visual Basic

پیشوند	نوع شی	پیشوند	نوع شی
Cho	جعبه ترکیبی	Lst	جعبه لیست
Chk	جعبه چک	Mnu	منو
Cmd	دکمه فرمان	Mod	مدول
Dir	جعبه درایو	Ole	OLE
Drv	جعبه درایو	Opt	دکمه گزینه
Fil	جعبه لیست فایل	Pic	جعبه تصویر
Fra	قاب	Res	منبع
Grd	شبکه	Shp	شکل
Hsd	مليه لغزشی افقی	Tmr	تایمر
Img	تصویر	Txt	جعبه متن
lb1	برچسب	Tyb	نوع داده تعریف شده توسط کاربر
Lin	خط	Vsb	مليه لغزشی عمودی

کمک محلی

محیط Visual Basic در هر مرحله و سطحی می تواند کمک های قابل توجهی در اختیارتان قرار دهد، گزینه Help Contents یک سیستم کمک HTML ارائه می کند. قاب سمت چپ این پنجره دارای مقدار زیادی کتاب کمک در جا (on – line) help است و قاب سمت راست سرفصل های کمک دکتر Gui را نشان خواهد داد.

قسمت اعظم کتابهای کمک در جا در CD شبکه توسعه دهندگان میکروسافت (MSDN) قرار دارند. MSDN مجموعه ای از مقالات متنوع درباره جنبه های مختلف محصولات میکروسافت است که Visual Basic هم یکی از آنهاست.

گفتگوی کمک Visual Basic دارای گزینه های ذیل است:

- Conternts: با این گزینه می توانیم به کتابهای طبقه بندی شده Visual Basic دسترسی پیدا کنیم.

- Index: با این گزینه می توانیم در میان تعداد زیادی لغت اندیش شده به دنبال مطلب مورد نظر خود بگردید.

- Search: با این گزینه می توانیم مقالات حاوی متن خاصی را پیدا کنیم.

- Favorites: با این گزینه می توانیم مقالات حاوی متن خاصی را پیدا کنیم.

- Search: با این گزینه می توانیم مقالات حاوی متن خاصی را پیدا کنیم.

- Favorites: با این گزینه می توانیم مقالات مورد علاقه تان را ذخیره و دسته بندی کنیم.
- با این تشکیلات مفصل کمک Visual Basic دیگر نیازی به کتابهای راهنمای قطور ندارید سیستم کمک Visual Basic خود چند کتاب قطور و مفصل درباره موضوعات مختلف و متنوع است.
- منوی Help دارای یک آیتم دیگر است: About Microsoft Visual Basic با این گزینه می توانیم اطلاعاتی درباره خود نرم افزار Visual Basic بدست آوریم. این گفتگو دارای یک دکمه بنام System Info است که به کمک آن می توانیم اطلاعات جامعی درباره سیستم خود به دست آوریم.

پشتیبانی

اگر درباره موضوعی در کمک محلی Visual Basic مطلبی نیافتید، یابه اشکالی در خود محیط Visual Basic برخورد کردید می توانیم با خدمات پشتیبانی میکروسافت یا کمک اینترنتی آن (که در سایت میکروسافت www.microsoft.com قرار داد) تماس برقرار کنیم. البته این قبیل تماس ها مجانی و بدون هزینه نیستند.

با محیط ویژوال بیسیک مانوس شوید

می توان به کمک جادوگر برنامه اقدام به ساختن برنامه های Visual Basic کرد. قبل از آن که در این راه بیشتر پیش برویم، اجازه دهید با اجرای یکی از برنامه های نمونه Visual

Basic کمی بیشتر با محیط آن آشنا شویم و با آن انس بگیریم، برای نوشتن برنامه‌های خوب، احساس راحتی با ابزار کار یکی از عوامل مهم است.

۱- Visual Basic را شروع کنیم.

۲- CD نصب Visual Basic را در درایو CD قرار دهید.

۳- در برگه Existing به پوشه Samples بروید و پروژه Controls را باز کنیم.

(اگر پسوند فایل‌ها را نمی‌بینید، با استفاده از آیت View Option file در کاوشگر ویندوز-

گزینه نمایش پسوند فایل‌ها را علامت بزنید.)

۴- اگر پیام آیا میل دارید این پروژه به Source safe یکی از ابزارهای ویژوال استودیو

برای حفظ ویرایش‌های مختلف کد یک پروژه است.)

۵- توجه کنیم که با باز کردن یک پروژه، به خودی خود اجرا نخواهد شد مگر آن که آیت

Run start را انتخاب کنیم.

توجه کنیم که چون این برنامه اینترپرتی است در همان محیط Visual Basic اجرا

می‌شود.

۶- برنامه Controls تعداد زیادی از کنترل‌های ویندوز را در خود دارد. روی یکی از این

کنترل‌ها کلیک کنیم تا گزینه‌های آن ظاهر شود.

۷- کمی با برنامه و کنترل‌های آن کار کرده و سپس با کلیک کردن دکمه Exit آن را ببندید، حال دوباره در محیط Visual Basic هستیم و دوباره پروژه Controls را در آن خواهیم دید. وقت آن است که کمی درباره پنجره‌های مختلف Visual Basic کنجکاوری کنیم.

۸- به پنجره پروژه نگاه کنیم. توجه کنیم که این پروژه فقط چند فرم دارد. با اینکه این برنامه کد هم دارای کدهای آن در همان فایل‌های فرم قرار دارند. اگر دکمه View Code را کلیک کنیم، کدهای فرم انتخاب شده را در پنجره کد خواهیم دید.

۹- روی یکی از فرم‌ها در پنجره پروژه دو - کلیک کنیم، پنجره آن فرم تقریباً به همان شکلی که هنگام اجرای برنامه دیدید، ظاهر خواهد شد. اگر به پنجره طراحی فرم نگاه کنیم موقعیت آن را در هنگام اجرای برنامه خواهیم دید.

۱۰- در پنجره طراحی فرم، با کشیدن مستطیل کوچک نماینده فرم فعال آن را جابجا کرده و سپس برنامه را دوباره اجرا کنیم، خواهیم دید که فرم مورد نظر در محل جدید ظاهر خواهد شد.

۱۱- به پنجره خواص نگاه کنیم. دقت کنیم که در این پنجره در هر لحظه خواص یکی از کنترل‌های فرم فعال را مشاهده خواهیم کرد.

۱۲- خواص کنترل انتخاب شده را بررسی کنیم.

۱۳- کنترل‌های دیگر را انتخاب و خواص آنها را هم بررسی کنیم. با استفاده از اهرم‌های تغییر اندازه (Sizing handle) یکی از کنترل‌ها را بزرگ یا کوچک کرده و خواص آن را در حالت جدید بررسی کنیم. حال که با محیط Visual Basic آشنا تر شدید، وقت آن است که خلاقیت خود را محک بزنید.

ساختن فرم

اولین برنامه ما فقط یک دکمه فرمان دارد و تصویری را نمایش می دهد و وقتی دکمه را کلیک کنیم برنامه این تصویر را عوض خواهد کرد. برای ساختن برنامه مراحل ذیل را دنبال کنیم:

۱- آیتم File/ New project را کلیک کنیم تا گفتگوی پروژه جدید باز شود. اگر برنامه باز قسمت قبل را هنوز نبسته‌اید، Visual Basic سؤال می کند که آیا میل دارید آن را ذخیره کنیم، جواب منفی بدهیم.

۲- آیکون Standard EXE را انتخاب کنیم. یک فرم بنام Form در ناحیه کاری محیط Visual Basic ظاهر خواهد شد (به میله عنوان آن دقت کنیم).

۳- با کلیک کردن دکمه Maximize ناحیه کاری Visual Basic، فضای آن را به حداکثر برسانید، دقت کنیم که فرم 1 هم دارای اهرم‌های تغییر اندازه است و پنجره

خواص مقدار خواص آن را نشان می دهد، فرم هم یک شی است مانند سایر اشیا
Visual Basic.

۴- اهرم سمت راست- پایین فرم را گرفته و آنرا به راست و پائین بکشید. توجه کنیم که با
این عمل اعداد موجود در قسمت راست میله ابزار (که عرض و ارتفاع فرم فعال را
نشان می دهند) تغییر خواهند کرد. ابعاد فرم را به ۷۴۰۰×۵۲۰۰ ست کنیم.

شاید از خود پرسیده باشیم واحد اندازه گیری ابعاد فرم (۷۴۰۰×۵۲۰۰) چیست؟ واحد
اندازه گیری پیش فرض Visual Basic واحدیست بنام twip.twip یک واحد اندازه گیری
مستقل از وضوح مانیتور است و معادل $۱/۱۴۴۰$ اینچ می باشد.

۵- فرم را در وسط پنجره طراحی فرم قرار دهید. به تغییرات مختصات فرم (در سمت
چپ ابعاد فرم، در میانه Visual Basic ندارد و فقط هنگام اجرای برنامه خود را نشان
خواهد داد. حتماً توجه کرده اید که روی فرم نقاطی وجود دارند که تشکیل یک شبکه
را می دهند، این نقاط صرفاً برای تعیین محل دقیق کنترل ها روی فرم هستند هنگام
اجرای برنامه دیده نخواهند شد، اگر میل ندارید این نقاط را (حتی در مرحله طراحی
برنامه) ببینید، می توانیم در برگه General گفتگوی View Options آیتم Show Grid
را غیر فعال کنیم.

۶- پنجره طراحی فرم را ببندید تا جای بیشتری باز شود.

۷- برای فرم نام بهتری انتخاب کنیم. برای این کار، در قسمت (Name) پنجره خواص نام مناسبتی به فرم بدهیم. وجود () در اطراف Name باعث می شود تا این خاصیت همیشه در ابتدای پنجره خواص ظاهر شود.

۸- نام فرم را از Form 1 به Frm My First تغییر دهید. به تغییر میله عنوان Visual Basic هم توجه کنیم.

۹- آنچه که در میله عنوان فرم برنامه دیده خواهد شد، مقدار خاصیت Captions فرم عبارت "Happy Appliction" را وارد کنیم به میله عنوان فرم دقت کنیم.

۱۰- قبل از ادامه کار، پروژه را ذخیره کنیم، برای این منظور آیتم / project File save را انتخاب کنیم، با این کار تمام فایل های پروژه ذخیره خواهند شد. پروژه ما فعلاً فقط دو فایل دارد: فایل فرم و فایل پروژه Visual Basic بطور پیش فرض از نام فرم به عنوان نام فایل آن استفاده خواهد کرد (البته پسوند FRM را هم به آن اضافه می کند). بعد از ذخیره شدن فایل های برنامه، در پایان نوبت ذخیره کردن فایل پروژه (VBP) می رسد به این پروژه نام Happy App بدهیم. در پاسخ به تقاضای Visual Basic مبنی بر اضافه کردن این پروژه به کتابخانه Source Safe جواب منفی بدهیم.

با نوشتن کد برنامه را کامل کنیم

برای کامل کردن برنامه ای که تا بدین جا نوشته ایم باید کمی کدنویسی کنیم:

۱- روی یک قسمت خالی پنجره فرم دو- کلیک کنیم، فرم ناپدید شده و پنجره کد با خطوط ذل ظاهر می شود:

private Sub Form – Load ()

End Sub

فرم ما به چهار خط کد احتیاج دارد که Visual Basic دو خط آن را برایمان نوشته است. پنجره کد درست مثل یک واژه پرداز کوچک کاست که در آن می توانیم چیزهایی نوشته، پاک و یا جابجا کنیم. هر کدی که می نویسد باید در یک روال (procedure) قرار داشته باشد. (هر روال یک نقطه شروع و یک نقطه پایان دارد. شروع و پایان اکثر روال ها را Visual Basic خود بوجود می آورد).

روال مجموعه ای است از دستورات Visual Basic که کار خاصی انجام می دهد.

۲- دستورات ذیل را بین دو خط بالا بنویسید:

$$\text{frm My First.Left} = (\text{Screen.Width} - \text{frm Ny First.Width}) / 2$$

$$\text{frm My First.Top} = (\text{Screen.Height} - \text{frm My First.Height}) / 2$$

قبل از هر خط سه بار کلید Space را بزنید تا ابتدای دستورات از لبه صفحه فاصله داشته باشند. این کار به وضوح کدها کمک زیادی خواهد کرد.

هنگام نوشتن دستورات بالا، همین که اولین (نقطه) را وارد کردید، Visual Basic یک لیست ظاهر خواهد کرد. در اینجا Visual Basic احساس کرده است که شما قصد دارید کاری درباره مقدار یک خاصیت انجام دهید و Visual Basic لیستی از گزینه های موجود را در اختیارتان می گذارد تا از میان آنها انتخاب کنیم. بعد از انتخاب خاصیت مورد نظر، کلید Top را بزنید تا کار ادامه پیدا کند.

۳- روی دکمه View Object در پنجره پروژه کلیک کنیم تا به پنجره فرم برگردید.
۴- روی دکمه فرمان دو- کلیک کنیم تا پنجره کد مجدداً باز شود، کدی که قبلاً نوشته اید همانجاست و روال جدیدی هم اضافه شده است (البته فقط ابتدا و انتهای آن) کد ذیل را در روال مزبور بنویسید:

```
img Happy. Picture= Load picture ( " / program files/ Microsoft Visual  
Studio /Common/ Graphics/ Bitmaps/ Assorted/ Happy. Bmp")
```

البته همین که پرانتز Load picture را باز کنیم باز هم Visual Basic با لیست خود به کمک شما خواهد آمد. اکثر توابع Visual Basic معمولاً بیش از یک مقدار می گیرند و این لیست کمک بزرگی در نوشتن این قبیل دستورات است. باز هم در اینبار صحبت خواهم کرد.

۵- برنامه را اجرا کرده و روی دکمه فرمان کلیک کنیم.

۶- برنامه را ببندید و قبل از خارج شدن از Visual Basic پروژه را ذخیره کنیم.

مروری بر کنترل‌ها

جعبه ابزار مجموعه‌ایست از کنترل‌ها، که می‌توان از آنها در برنامه استفاده کرد. استفاده از کنترل‌های جعبه ابزار هیچ محدودیتی ندارد، به عبارت دیگر هر چقدر از جعبه ابزار استفاده کنیم هرگز خالی نخواهد شد. کنترل‌های موجود در جعبه ابزار به نیازهای برنامه بستگی دارد. از یک کنترل تا زمانی که در جعبه ابزار نباشد. نمی‌توان استفاده کرد (سهولت کار با جدوگر برنامه در آن است که کنترل‌های مورد نیاز را به خود به پروژه اضافه خواهد کرد). اگر بخواهیم از کنترلی استفاده کنیم که در جعبه ابزار نباشد، طرز اضافه کردن آن به جعبه ابزار را نشان خواهیم داد.

که جعبه ابزار چقدر می‌تواند پییده شود. البته می‌توانیم هر قدر که بخواهیم جعبه ابزار را کوچک کنیم ولی به یاد داشته باشیم که جعبه ابزار میله لغزشی ندارد و برای دیدن مجدد کنترل‌ها باید آن را دوباره بزرگ کنیم. کنترل‌های Visual Basic را می‌توان به سه دسته اساسی تقسیم کرد. کنترل‌های ذاتی، آنهایی که جز استاندارد Visual Basic هستند، کنترل‌های اکتیوایکس، فایل‌هایی با پسوند OCX که می‌توان آنها را به یک Visual Basic قرار می‌دهند.

کنترل‌های دارای خواص مشترک بسیاری هستند، وجود این خواص مشترک ناشی از آن است که اکثر آنها دارای اندازه، مکان، رنگ و از این قبیل هستند.

بیاد داشته باشیم که تمام خواص یک کنترل از قبل مقادیری دارند، به محض آن که Visual Basic کنترلی بوجود می‌آورد به تمام خواص آن مقادیر پیش فرض می‌دهد (که اغلب هم با آنها کاری نخواهیم داشت).

خواص یک کنترل را (علاوه بر پنجره خواص) از طریق کد هم می‌توان دستکاری کرد. مثلاً اگر به دلایلی یک کنترل (موقتاً) قابل استفاده نباشد، می‌توان خاصیت Enabled آن را در کد برنامه به False ست کرد. خواصی وجود دارند که فقط از طریق کد برنامه قابل دستکاری هستند، این قبیل خواص هرگز در پنجره خواص دیده نمی‌شوند).

خواص مشترک کنترل‌های Visual Basic

توضیح	خاصیت
تراز متن کنترل را تعیین می‌کند.	Alignment
رنگ زمینه کنترل را تعیین می‌کند.	Back Color
تعیین می‌کند که آیا کنترل حاشیه دارد یا خیر	Border Style
متنی که روی کنترل دیده خواهد شد.	Caption
تعیین می‌کند که آیا کاربر قادر به کار با کنترل هست یا خیر	(True/False) Enabled
ویژگی‌های مختلف فونت کنترل را تنظیم می‌کند	Font
رنگ قلم (پیش زمینه) کنترل را تنظیم می‌کند.	Fore Color
ارتفاع کنترل را مشخص می‌کند (برچسب twip)	Height
فاصله لبه چپ کنترل را با پنجره فرم تعیین میکند (Twip)	Left
شکل کرسر ماوس را وقتی که روی کنترل است، تعیین می‌کند،	Mouse pointer
نام کنترل: نمی‌کند در برنامه به آن ارجاع می‌شود.	Name
وقتی کرسر ماوس لحظه‌ای روی کنترل توقف کند، این متن ظاهر خواهد شد.	Tool Tip Text
فاصله لبه بالایی کنترل با پنجره فرم (Twip)	Top
(True/ False) تعیین می‌کند که آیا کنترل مریی است یا نامریی	Visible
عرض (پهنای) کنترل (Twip)	Width

فصل دوم

بانک اطلاعاتی چیست؟

از بانک اطلاعاتی تعاریف گوناگونی شده است که تعریف زیر از همه جامعتر به نظر می‌رسد: بانک اطلاعاتی مجموعه‌ای سازمان یافته از اطلاعات و داده‌های مرتبط به هم است. داده‌ها عبارتند از: حقایق و ارقام یک موضوع خاص و اطلاعات عبارتند از تباعی که از ترکیب داده‌ها حاصل می‌گردند. مؤسسات و سازمانها معمولاً به دو صورت سیستمهای اطلاعاتی خود را تشکیل داده از آنها استفاده می‌کنند.

استفاده از بانکهای اطلاعاتی:

در این روش، داده‌های موجود به صورت مجتمع یا بانک مورد استفاده قرار می‌گیرند. در چنین سیستمی کاربر می‌تواند بدون سردرگمی و با صرف وقت اندکی، اطلاعات مورد نیاز خود را از داده‌های موجود در مجتمع اخذ کند. امروزه اکثر مؤسسات و سازمانها سعی می‌کنند از این سیستم اطلاعاتی استفاده کنند.

اجزای تشکیل دهنده بانک اطلاعاتی:

- ۱- کاربران
- ۲- سیستم بانکهای اطلاعاتی
- ۳- داده‌های موجود در بانکهای اطلاعاتی

بانک اطلاعاتی ویژوال بیسیک

در ویژوال بیسیک می توان برنامه هایی نوشت که بانک اطلاعاتی را ایجاد و ویرایش کنند، سه نوع بانک اطلاعاتی در ویژوال بیسیک قابل استفاده اند.

۱- بانکهای اطلاعاتی access : این نوع بانکهای اطلاعاتی را می توان از طریق

Microsoft Access یا Visual basic ایجاد و مدیریت نمود. (که ما برای فرهنگ لغت

نیز از طریق نرم افزار Visual basic به ایجاد بانکهای اطلاعاتی access می پردازیم)

۲- بانکهای اطلاعاتی خارجی: این نوع بانکهای اطلاعاتی شامل بانکهای اطلاعاتی

پارادوکس، Base III و Base IV و Faxpro است. (این نوع بانکهای اطلاعاتی را

می توان در ویژوال بیسیک ایجاد و مدیریت کنید:

۳- بانکهای اطلاعاتی خارجی ODBC : پیکر بندی این نوع بانکهای اطلاعاتی، مشتری /

کارگزار است و

شامل اوراکل و SQL Server است.

ساختار بانک اطلاعاتی:

رکورد: به اطلاعات مربوط به هر فرد، شی، یا هر چیز دیگری که نه بانک اطلاعاتی وارد

شود، رکورد گویند.

فیلد: به هر یک از اجزاء یک رکورد، فیلد گفته می شود. که خود دارای مشخصاتی به شرح زیر می باشد که بیانگر محتویات آن فیلد باشد.

نوع فیلد: هر فیلد دارای نوعی است. مانند Text، یا صحیح و یا اعشاری و غیره...

اندازه فیلد: اندازه فیلد مشخص می کند که فیلد چند بایت از فضای حافظه را باید اشغال کند.

جدول در بانک اطلاعاتی:

اطلاعات موجود در بانک اطلاعاتی به صورت جدول ذخیره می شوند و هر جدول دارای تعدادی سطر و ستون است.

ایجاد یک بانک اطلاعاتی:

پس از آشنایی مختصر با مفاهیم بانک اطلاعاتی، به نظر می رسد که بهترین کار، ایجاد یک بانک اطلاعاتی می باشد. برای ایجاد یک بانک اطلاعاتی در محیط ویژوال بیسیک مراحل زیر را به ترتیب انجام می دهیم.

۱- گزینه Visual Data manager را از منوی Add- ins را انتخاب می کنیم تا صفحه

visData ظاهر شود شکل ۱-۲

۲- گزینه File/ new/ Microsoft Access/ Version 7.0 MB را انتخاب کنید. اکنون،

پنجره ای ظاهر می شود و نام فایل بانک اطلاعاتی را از شما درخواست می کند. پس از

اینکه نام بانک اطلاعاتی خود را وارد کردید دکمه Save کلیک کرده اکنون صفحه‌ای با دو پنجره ظاهر می‌شود. یک پنجره به نام Database Windows و دیگری به نام SQL Statement است. شکل ۲-۲

۳- در پنجره Data base windows دکمه راست ماوس را کلیک کنید تا منویی با دو گزینه ظاهر شود.

برای ایجاد جدول جدید گزینه Newtable را انتخاب کنید. اکنون شکل ۳-۲ ظاهر می‌شود که می‌توانید ساختار جدول خود را تنظیم کنید. نام این پنجره Table Structure است. در بخش Tablename نام جدول خود را وارد کنید.

۴- برای افزودن یک فیلد به جدول گزینه Addfield را کلیک کنید. تا شکل ۴-۲ ظاهر شود

این شکل دارای بخشهای مختلفی است که به شرح آن می‌پردازیم:

۵- بخش NAME: در این بخش باید نام فیلد را وارد کنید.

۶- بخش Type: در این بخش نوع فیلد را تعیین کنید.

بخشهای Variable field, Fixed field: این بخش‌ها برای فیلدهای معین بکار می‌رود.

گزینه Fixed field مشخص می‌کند که طول فیلد قابل تغییر نباشد ولی گزینه variable

field مشخص می‌کند که طول فیلد قابل تغییر باشد. (شما باید گزینه variablefield را

انتخاب کنید).

بخش allow zero length : اگر این گزینه فعال نباشد (علامت در آن نباشد) ، محتویات فیلد متنی هنگام ورود اطلاعاتی می تواند خالی باشد ولی اگر این گزینه را فعال کنید حداقل یک کاراکتر باید در آن قرار دهید. شما این گزینه را فعال کنید.

بخش Required : اگر این بخش برابر با مقدار True باشد، فیلد مورد نظر نباید تهی باشد(Nall)

بخش Ordinal position : موقعیت عادی فیلد را در جدول مشخص می کند.

بخش Validation Rule : در این بخش می توانید شرطی را وارد کنید که پس از ورود داده است. این فیلد، ارزیابی شود و چنانچه شرط نادرست بود، خطایی صادر می گردد، و نمی توان از آن عبور کرد این بخش معمولاً برای فیلدهای عددی مورد استفاده قرار می گیرد تا بازه آنها را کنترل کند.

بخش Validation text : می توانید متنی را در این بخش تایپ کنید تا در صورتی که خطایی در Validation text رخ داد این متن به عنوان پیام خطا ظاهر شود.

بخش Defaultvalue : در این بخش مقداری را باید وارد کنید که به عنوان مقدار اولیه محسوب می شود. در آخر بعد از تنظیم همه بخشها و وارد کردن همه فیلدها دکمه OK را فشار دهید و در آخر Close را فشار دهید. سپس در پنجره ای که در آن هستند دکمه Build table را کلیک کنید:

۵- از صفحه Vis Data خارج شوید. در ادامه می توانید اطلاعاتی را وارد بانک اطلاعاتی کنید.

چاپ ساختار بانک اطلاعاتی :

برای چاپ ساختار بانک اطلاعاتی ، در صفحه table structure که بخشی از آن در شکل ۲-۳ آمده است Print structure را کلیک کنید.

ورود اطلاعات به بانک:

اکنون که ساختار یک بانک اطلاعاتی را ایجاد کردید، باید بتوانید اطلاعاتی را وارد آن کنید. برای این کار مراحل زیر را انجام دهید.

۱- گزینه Addins / Visual Data Manager را انتخاب کنید تا صفحه VisData ظاهر شود.

۲- در صفحه VisData ، گزینه File/ open Deta Base/ Microsoft access را انتخاب کنید.

اکنون لیست فایل های بانک اطلاعاتی ظاهر می شود. از این لیست فایل بانک اطلاعاتی مورد نظر را دو بار کلیک کنید. اکنون نام جدول های بانک اطلاعاتی ظاهر می شوند. یکی از آنها به نام Properties است که خود سیستم تشکیل می دهد تا ویژگی های بانک اطلاعاتی را نگهداری کند و دیگری خود جدول ماست که قبلاً آن را ایجاد کردیم. اکنون می خواهیم

جدول را باز کنیم. برای این کار روی آیکن بالای صفحه Vis Data که مربوط به Table است کلیک کرده اکنون برای بازشدن جدول بر روی اسم جدول دو بار کلیک کنید. تا شکل ۲-۵ باز شود. برای ورود یک رکورد به جدول، دکمه Add را کلیک کنید. (پنجره ای با دو دکمه Update و Cancel ظاهر می شود). حال رکورد مورد نظر خود را وارد کنید و دکمه Update را فشار دهید و در پنجره ای که ظاهر می شود دکمه Add را کلیک کنید. اکنون آماده هستید رکورد دوم را وارد کنید. این روند را تا وارد کردن کل رکوردها به فایل ادامه دهید.

۴- در نوار لغزانی که در پایین شکل ۲-۵ می بینید می توانید تمام رکوردهای فایل را ببینید. تعیین رکورد جاری :

توسط دکمه Move می توانید رکورد جاری را تعیین کنید.

مرتب سازی رکوردها:

برای مرتب سازی رکوردهای جدول پس از باز کردن جدول، دکمه sort را انتخاب کنید کادری ظاهر می شود و سؤال می کند که مرتب سازی بر اساس کدام ستون (فیلد) صورت می گیرد.

ویرایش محتویات رکورد

برای ویرایش محتویات رکورد، پس از باز کردن جدول دکمه Edit را کلیک کنید. اکنون می توانید رکورد جاری را تغییر دهید پس از تغییر رکورد جاری دکمه Update را در صفحه کلیک کنید.

حذف رکورد

حذف رکورد توسط دکمه Delet صورت می گیرد. پس از کلیک کردن این دکمه پیامی صادر می شود و سؤال می کند که آیا رکورد جاری حذف شود یا خیر. بازدن دکمه yes، رکورد مورد نظر حذف خواهد شد.

جستجوی رکورد

برای اینکه رکوردی را جستجو کنید در صفحه ای که ظاهر می شود دکمه Find را کلیک کرد. و در بخش Field نام فیلدی که می خواهید در آن عمل جستجو صورت گیرد را تعیین کنید.

ولی برای اجرای طراحی فرم، نیاز به گزینه Data form wizard است که در این منو موجود ندارد. اگر در سیستم شما، این گزینه وجود دارد، مرحله 2 به بعد را انجام دهید.

۲- گزینه Add-In Manager را در منوی فوق کلیک کنید تا پنجره آن ظاهر شود. در این شکل گزینه VB6 Data Form Wizard را دو بار کلیک کنید تا کلمه Loaded در مقابل آن تایپ شود. اکنون این پنجره را با دکمه OK ببندید.

۳- گزینه Add ins / Data form wizard را انتخاب کنید. پنجره ای ظاهر می شود و اجرای برنامه ویزارد فرم را اعلان می کند. دکمه next را کلیک کنید تا به مرحله بعدی بروید اکنون شکلی برای شما ظاهر می شود و از شما می خواهد که مشخص کنید با چه نوع بانک اطلاعاتی می خواهید کار کنید. شما بانک اطلاعاتی Access را برگزینید و دکمه next را کلیک کنید.

۴- در کادری که ظاهر می شود، نام بانک اطلاعاتی مورد نظرتان را انتخاب کنید. و سپس دکمه next را کلیک کنید.

۵- اکنون شکل دیگری ظاهر می شود و از شما می خواهد طرح فرم خود را مشخص کنید. شما Grid (Data sheet) را انتخاب کنید و بخش Binding type را به عهده خود سیستم واگذار کنید.

۵- اکنون شکل گیری ظاهر می شود که در این شکل، در بخش Record Source بر روی ▼ کلیک کنید تا لیست جدولهای این بانک اطلاعاتی ظاهر شود. جدول مورد نظر را انتخاب کنید که در فرم قرار گیرند. برای اینکه فیلدی را برای این منظور انتخاب کنید، مکان

نمای ماوس را به آن فیلد ببرید و دکمه ▶ را کلیک کنید. اکنون این فیلد در بخش Selected field قرار می گیرد. برای اینکه رکوردها برحسب فیلد خاصی (طول فرهنگ لغت) مرتب باشند در بخش Column to Sort By نام آن فیلد را انتخاب کنید. سپس دکمه next را کلیک کنید تا به مرحله بعدی بروید.

۷- شکل در مرحله بعد می توانید دکمه هایی را برای قرار گرفتن در فرم انتخاب کنید. سپس دکمه next را کلیک کنید.

۸- در صفحه ای که ظاهر می شود دکمه Finish را کلیک کنید تا کار طراحی فرم به پایان برسد. صفحه دیگری ظاهر می شود و به شما اطلاع می دهد که فرم ایجاد شد و به پروژه اضافه گردید. در این صفحه دکمه OK را کلیک کنید. هنوز کار تمام نشده است.

۹- گزینه Project / Prajeet1 Praperties را انتخاب کرده، در این شکل در بخش Startup object نام Form را مشاهده می کنید. بر روی دکمه ▼ کلیک کنید و فرم خود را انتخاب کرده تا به عنوان اولین فرم پروژه انتخاب گردد. سپس OK را کلیک کنید. سپس فرم Form را که در هنگام راه اندازی پروژه ایجاد شد ببندید.

۱۰- با کلید F5 پروژه را اجرا کنید.

مفهوم ایندکس

عمل ایندکس، کار جستجو و یافتن اطلاعات را ساده می کند.

ما در دیکشنری یا فرهنگ لغت از عمل ایندکس سازی بسیار بهره می بریم به این صورت که اگر ما میخواستیم یک لغت را در فرهنگ لغت از بین هزاران لغت پیدا کنیم. شدیداً به مشکل برمی خوردیم. عمل ایندکس این امکان را به ما می دهد تا لغات را بر اساس حروف الفبا از A تا Z مرتب کنیم. مثلاً برای یافتن معنای لغتی مثل Computing، به جایی از دیکشنری می روید که لغات آن با حرف C شروع می شوند. مرتب سازی این 26 حرف زبان انگلیسی در یک دیکشنری انگلیسی به فارسی منجر به سهولت دستیابی به لغات می شود. حال چگونه می توانیم ایندکس را در بانک اطلاعاتی اعمال کنیم.

و برای اینکار ویژوال بیسیک را راه اندازی کرده گزینه Add-ins/ Visual Data manager را انتخاب کنید تا صفحه Visual data manager باز شود. در این صفحه گزینه File / Open Data Base / Microsoft access را کلیک کنید و سپس نام فایل بانک اطلاعاتی خود را دو بار کلیک کنید تا باز شود. سپس بر روی جدول Mytable دکمه سمت راست ماوس را کلیک کنید تا منویی ظاهر شود. در این منو، گزینه Design را انتخاب کنید. در این شکل می توانید فیلد یا فیلدهایی را به عنوان ایندکس انتخاب کنید. سپس گزینه Add index وارد کنید سپس نام فیلد را در بخش Available fields کلیک کنید تا به عنوان فیلد کلید

این دکس معرفی شود. دکمه OK را کلیک کنید. این پنجره و پنجره دیگری که ظاهر می شود را ببندید. در آخر روی جدول Mytable دو بار کلیک کنید. اکنون جدول برحسب فیلد مورد نظر به طور صعودی مرتب است.

برنامه نویسی و بانک اطلاعاتی

برای برنامه نویسی در بانک اطلاعاتی باید با اشیای بانک اطلاعاتی آشنا شویم. در برنامه نویسی به دو طریق می توان با بانک اطلاعاتی کار کرد

۱- از طریق شی Data

۲- از طریق شی Database

اشیای بانک اطلاعاتی:

اشیای بانک اطلاعاتی، ابزارهایی برای کار کردن با بانک اطلاعاتی هستند.

از طریق این اشیاء می توان اعمالی مثل ایجاد بانک اطلاعاتی، تغییر، حذف و اضافه رکوردها

به بانک اطلاعاتی، حذف و اضافه فیلدهای بانک اطلاعاتی، مرتب سازی و ایندکس را انجام

داد.

شی Data :

عنصر Data ، ابزاری برای برقراری ارتباط بایانک اطلاعاتی است و دارای خواص است که عملکرد آن را مشخص می کند. بعضی از خواص این شی مثل Font , Name , Caption و غیره، همانند عناصر دیگر است.

اما این شی دارای خواصی است که برای بانک اطلاعاتی مورد استفاده قرار می گیرد.

جدول مربوط به شی Data طبق جدول ۱-۲ می باشد.

شی Data Base :

Data Base مجموعه ای از بانک اطلاعاتی است که در یک فضای کاری باز می شود.

جدول ۱-۲: بعضی از خواص مهم شیء Data در بانک اطلاعاتی

عملکرد	خاصیت
این خاصیت مشخص می کند که عنصر Data باید با چه نوع بانک اطلاعاتی ارتباط برقرار کند، مثل Paradox, Excel, Access و غیره. برای مشاهده انواع بانکهای اطلاعاتی، خاصیت Connect را کلیک کنید تا علامت ▼ ظاهر شود. این علامت را کلیک کنید تا انواع بانکهای اطلاعاتی را که عنصر Data می تواند با آنها ارتباط برقرار کند مشاهده کنید و نوع مورد نظرتان را انتخاب کنید.	Connect
این خاصیت، نام بانک اطلاعاتی را مشخص می کند که عنصر Data باید با آن ارتباط برقرار کند	DataBaseName
این خاصیت مشخص می کند که چنانچه یک بانک اطلاعاتی در حال استفاده باشد، کاربر دیگر نیز می تواند از آن استفاده کند یا خیر. مقدار False موجب می شود کس دیگری استفاده نکند ولی مقدار True موجب می شود تا توسط چند نفر قابل استفاده باشد.	Exclusive
این خاصیت مشخص می کند که آیا بانک اطلاعاتی که Data با آن ارتباط دارد قابل تغییر باشد یا خیر. مقدار False مشخص می کند که قابل تغییر باشد و مقدار True مشخص می کند که بانک اطلاعاتی فقط قابل خواندن باشد.	ReadOnly
این خاصیت مشخص می کند که عنصر Data با چه منبع داده ای باید ارتباط برقرار کند. این منابع داده ها، همان جدولهای بانک اطلاعاتی هستند که نام آن در خاصیت DataBaseName قرار می گیرد.	RecordSource

جدول ۲-۲: خواص شیء DataBase

خاصیت	عملکرد
CollatingOrder	زبان مربوط به بانک اطلاعاتی را برای مقایسه فیلدها مشخص می کند که مقدار فرضی آن انگلیسی است.
Connect	نوع بانک اطلاعاتی را مشخص می کند و در یک متغیر رشته ای قرار می دهد. برای بانک اطلاعاتی اکسس، این رشته تهی است.
Name	محل وجود بانک اطلاعاتی را به همراه نام آن در یک متغیر رشته ای قرار می دهد.
Transactions	مشخص می کند که آیا بانک اطلاعاتی امکان عقبگرد داشته باشد یا خیر.
Updateable	مشخص می کند که بانک اطلاعاتی به صورت خواندن و نوشتن باشد یا به صورت فقط خواندنی. مقدار True به معنای خواندن و نوشتن و مقدار False به معنای فقط خواندنی است.
Version	گونه یا نسخه بانک اطلاعاتی را مشخص می کند که قابل تغییر نیست.
ReplicaID	اگر مقدار این خاصیت True باشد، بانک اطلاعاتی قابل تکثیر ایجاد می شود که هر نسخه ای از آن را می توان در اختیار کاربران قرار داد.
QueryTimeOut	مدت زمانی را برحسب ثانیه مشخص می کند که یک تقاضا در بانکهای اطلاعاتی مشتری/کارگزار، صرف می کند. مقدار اولیه آن ۶۰ ثانیه است.

جدول ۳-۲: متدهای شیء DataBase

نام متد	عملکرد
CreateProperty	با استفاده از این متد می توان خاصیتی را ایجاد کرد.
CreateRelation	با استفاده از دستور Set بین فیلدهای جدول ارتباط برقرار می کند.
MakeReplica	نسخه ای از بانک اطلاعاتی ایجاد می کند و در صورتی قابل اجراست که بانک اطلاعاتی ویژگی تکثیر شدن را دارا باشد.
Close	شیء بانک اطلاعاتی (DataBase) را می بندد.
Refresh	اشیای موجود در بانک اطلاعاتی را به روز می کند.

بازکردن بانک اطلاعاتی:

برای بازکردن بانک اطلاعاتی در محیط برنامه نویسی، از تابع Open data Base استفاده می شود.

این تابع به صورت زیر به کار می رود.

Open Data Base (Data Base neme [, options [, readonly [, Conne c4]])

بازکردن Recordset :

Recordset دارای مجموعه ای از فیلدها و ایندکسها است.

برای باز کردن Recordset از متد Open Recordset استفاده می شود که به صورت زیر به

کار می رود ([Type] , Table Name) Open Record

- اگر Recordset از نوع Dynaset باشد، جستجو از طریق Find امکان پذیر است و می توان در بانک اطلاعاتی تغییراتی ایجاد کرد.

- اگر Recordset از نوع Snapshot باشد، فقط قابل خواندن است.

اگر به دستورات صفحه بعد نگاه کنید دستورات (۱) و (۲) به ترتیب اشیای db و rs را از

نوع Data Base و Recordset تعریف می کنند. دستور (۳) بانک اطلاعاتی و دستور (۴)

Recordset را باز می کند. دستور (۵) ایندکس « StnolIndex » مرتب شود.

حذف ایندکس:

برای حذف index از مجموعه ایندکسها از delete استفاده می شود. به عنوان مثال دستور

زیرایندکس Stnolindex را از مجموعه ایندکسها حذف می کند.

Db. Table Defs ("My table") . Indexes . Delet "StnolIndex"

نمایش محتویات بانک اطلاعاتی:

در این جاشی دیگری به نام DataGrid , ADODC را مورد بررسی قرار می دهیم این دو

عنصر در لیست ابزارهای ویژوال بیسیک وجود ندارد. برای اینکه آنها را به فرم اضافه کنیم

مراحل زیر را انجام می دهیم:

ابتدا پروژه جدیدی را ایجاد کنید و گزینه Project / Components را انتخاب کنید تا صفحه Component (OLEDB) ظاهر شود. در این صفحه دو گزینه

☒ Microsoft Data Grid Control 6.0 (OLEDB) ,

☒ Microsoft ADO Data Control 6.0 (OLEDB)

را انتخاب کنید. سپس دکمه OK را کلیک کرده اکنون دو عنصر به ابزارهای ویژوال بیسیک اضافه می شوند که نام یکی از آنها Data Grid و نام دیگری Adodc است. حال اگر بخواهیم به عنوان مثال محتویات یک جدول از بانک اطلاعاتی را ببینیم ابتدا: یک عنصر Data Grid و یک عنصر Adodc به فرم اضافه کرده سپس بر روی عنصر Adodc کلیک راست کرده و در منویی که ظاهر می شود گزینه AdODC Properties را انتخاب کنید تا صفحه Property Pages ظاهر شود. در این صفحه در بخش Source of Connection گزینه Use Connection String را انتخاب کرده دکمه Build را در کنار کادر این گزینه . کلیک کنید تا صفحه Data link Properties باز شود. در این صفحه گزینه Microsoft Jet 3.51 OLEDB Provider را انتخاب کنید و سپس گزینه Connection را در همین صفحه انتخاب کنید و در قسمت ۱ نام بانک اطلاعاتی موردنظرتان را وارد کنید. می توانید دکمه ... را در این کادر کلیک کنید تا نام بانکهای اطلاعاتی ظاهر شوند و از میان آنها ، نام مورد نظر را انتخاب کنید. در همین صفحه دکمه Test Connection را انتخاب

کنید، چنانچه اتصال به خوبی برقرار شده باشد، پیامی صادر می گردد و صحت ارتباط را اعلان می دارد. اکنون دکمه Ok را کلیک کنید تا پیام از صفحه محو شود و دکمه Ok را کلیک کنید تا به صفحه Property. Pages بروید. حال در این صفحه، گزینه Record Source را کلیک کنید و در بخش Command type گزینه Odcmd table-2 را برگزینید. در بخش Table or Stared Procedure Name، علامت ▼ را کلیک کنید تا جدولهای بانک اطلاعاتی ظاهر شوند جدول My table را انتخاب کنید و سپس دکمه Ok را کلیک کنید تا به فرم برنامه برگردید. سپس بر روی عنصر Data grid کلیک کنید و سپس کلید F4 را فشار دهید خاصیت Data Source آنرا برابر با Adobe1 قرار دهید. اکنون همه چیز برای نمایش محتویات بانک اطلاعاتی آماده است.

افزودن ایندکس به جدول:

افزودن ایندکس به جدول موجب دسترسی سریع به اطلاعات می شود. هر جدول حداکثر می تواند ۱۵ ایندکس مختلف داشته باشد و هر ایندکس نیز می تواند شامل یک یا چند فیلد باشد. جزئیات هر ایندکس در مجموعه ایندکسها نگهداری می شود. با استفاده از متد Append می توان ایندکسی را به مجموعه ایندکسها و همچنین به جدول بانک اطلاعاتی اضافه کرد. به عنوان مثال به کمک عملیات زیر ما می توانیم یک ایندکسی را تعریف کنیم و آن را به جدول Mytable اضافه کنیم.

- 1- Dim db As Data Base .
- 2- Dim NewIndex As New Index.
- 3- Set db = Open Data base ("Mybank . mdb") ؛
- 4- New Index . name = "StnoIndex"
- 5- New Index. Fields = "Stno"
- 6- New Index. Unique = False
- 7- Db. TableDefs ("My table") . Indexes. Append newIndex
- 8- Db. Close

فعال کردن یک فایل ایندکس در جدول:

در قسمت بالا ما توانستیم یک ایندکس را تعریف کرده و آنرا به جدول Mytable اضافه کنیم. اما این کار یک بار صورت می گیرد، یعنی آن برنامه را فقط می توانید یکبار اجرا کنید، زیرا اگر باردیگر سعی در اجرای آن داشته باشید، سیستم پیغام می دهد که این ایندکس قبلاً ساخته شده و در جدول وجود دارد. بارهای بعدی فقط باید ایندکس مورد نظرتان را فعال کنید و در جدول از آن بهره ببرید. برای فعال کردن یک فایل ایندکس در جدول، از خاصیت ایندکس استفاده می شود.

1-Dim db As Data Base

2-Dim rs As Recordset

3-Sed db = Open Datat Base ("My bank. Mdb") ؛

4- Set rs = Open Recordset ("My table")

5- R.s index = "Stnolnex"

منابع:

گام به گام ویژوال بیسیک : تألیف مهندس عین ا... جعفرنژاد قمی - مهندس رمضان عباس

نژاد

آموزش ویژوال بیسیک ۶ : تألیف شیمون مورزینسکی و لوئل موئر، مترجم: مهندس جمشید

گیلانی پور

آموزش ویژوال بیسیک ۶: تألیف دکتر هاروی دیتل - پل دیتل - تم نیتو، مترجم : مهندس

بهرام پاشایی

ویژوال بیسیک ۶ : تألیف گریک پری، مترجم: علیرضا زارع پور