

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

برنامه نویسی و ویژوال بیسیک پیشرفته

(جلد دوم)

شاخه: کاردانش

زمینه: خدمات

گروه تحصیلی: کامپیوتر

زیرگروه: کامپیوتر

رشته مهارتی: برنامه نویسی پایگاه داده

شماره رشته مهارتی: ۳۱۵-۱۰۲-۱۷-۳

کد رایانه ای رشته مهارتی: ۶۱۴۰

نام استاندارد مهارت مبنا: برنامه نویسی، (VB، DELPHI)

کد استاندارد متولی: ۰-۸۴/۸۰/۱/۳/۳

عملی: ۰۴۹۵

شماره درس: نظری: ۰۴۹۴

عنوان و نام پدیدآور: برنامه نویسی و ویژوال بیسیک پیشرفته [کتابهای درسی] ... کد استاندارد متولی ۰-۸۴/۸۰/۱/۳/۳ - مؤلف منصور ولی نژاد؛ [برای] وزارت آموزش و پرورش، سازمان پژوهش و برنامه ریزی آموزشی؛ برنامه ریزی محتوا و نظارت بر تألیف، دفتر برنامه ریزی و تألیف آموزش های فنی و حرفه ای و کاردانش.

مشخصات نشر: مؤسسه فرهنگی هنری دیباگران تهران، ۱۳۸۹.

مشخصات ظاهری: ج۲: مصور، جدول.

شابک دوره: 978-600-124-054-6 ؛ ج۱: 978-600-124-053-9 ؛ ج۲: 978-600-124-055-3

وضعیت فهرست نویسی: فیبا

یادداشت: ج۱ (چاپ اول: ۱۳۸۹) (فیبا).

یادداشت: ج۱ (چاپ دوم: ۱۳۹۰) (فیبا).

موضوع: ویژوال بیسیک (زبان برنامه نویسی کامپیوتر)

موضوع: بیسیک (زبان برنامه نویسی کامپیوتر)

شناسه افزوده: ولی نژاد، منصور، ۱۳۴۵-

شناسه افزوده: سازمان پژوهش و برنامه ریزی آموزشی. دفتر برنامه ریزی درسی آموزش های فنی و حرفه ای و کاردانش.

شناسه افزوده: سازمان پژوهش و برنامه ریزی آموزشی

رده بندی کنگره: ۱۳۸۹ ۹۴۶/۷۳/۹ Q476

رده بندی دیویی: ۳۷۳/۳/۳/۱۳۳/۸۰-۸۴/۸۰

شماره کتابشناسی ملی: ۲۰۷۰۵۶۲

سازمان پژوهش و برنامه‌ریزی آموزشی

همکاران محترم و دانش‌آموزان عزیز:

پیشنهادها و نظرهای خود را درباره محتوای این کتاب به نشانی: تهران- صندوق پستی شماره ۴۸۷۴/۱۵ دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کار دانش، ارسال فرمایند.

پایان‌نگار (ایمیل) tvoccd@roshd.ir وب‌گاه (وب‌سایت) www.tvoccd.edu.ir

برنامه‌ریزی محتوا و نظارت بر تألیف: دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کار دانش

عنوان و شماره کتاب: برنامه‌نویسی ویژوال بیسیک پیشرفته (جلد دوم) - ۶۱۲/۹

مجری: مؤسسه فرهنگی هنری دیباگران تهران

شماره درس: ۰۴۹۴ ، ۰۴۹۵

مؤلف: مهندس منصور ولی‌نژاد

ویراستار ادبی: شیوا غمگسار

صفحه‌آرا: احمد یوسفی دلچه، مهسا کوراوی

طراح جلد: بیتا اشرفی مقدم

محتوای این کتاب در نوزدهمین جلسه مورخ ۸۹/۳/۱۸ کمیسیون تخصصی رشته کامپیوتر دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کار دانش با عضویت: بتول عطاران، محمدرضا شکرریز، سید حمیدرضا ضیایی، افشین اکبری، فرنگیس شاکری و حسن رحیمی مقدم تأیید شده است.

نوبت و سال چاپ: چاپ سوم ۱۳۹۱

لیتوگرافی / چاپ و صحافی: شرکت چاپ و نشر کتابهای درسی ایران ((سهامی خاص))

ناشر: مؤسسه فرهنگی هنری دیباگران تهران

نشانی ناشر: تهران- سعادت‌آباد- میدان کاج- خیابان سرو شرقی- روبه‌روی خیابان علامه- پلاک ۴۹

آدرس الکترونیکی: www.dibagaran.mft.info

صندوق پستی: ۱۴۶۵۵/۴۶۶

دورنگار: ۲۲۰۹۸۴۴۸

تلفن: ۲۲۰۹۸۴۴۶-۷

نظارت بر چاپ و توزیع: اداره کل چاپ و توزیع کتاب‌های درسی، سازمان پژوهش و برنامه‌ریزی آموزشی

تهران: خیابان ایرانشهر شمالی- ساختمان شماره ۴ آموزش و پرورش (شهید موسوی)

کد پستی: ۱۵۸۴۷۴۷۳۵۹

دورنگار: ۸۸۳۰۹۲۶۶

تلفن: ۸۸۸۳۱۱۶۱-۹

آدرس الکترونیکی: www.chap.sch.ir

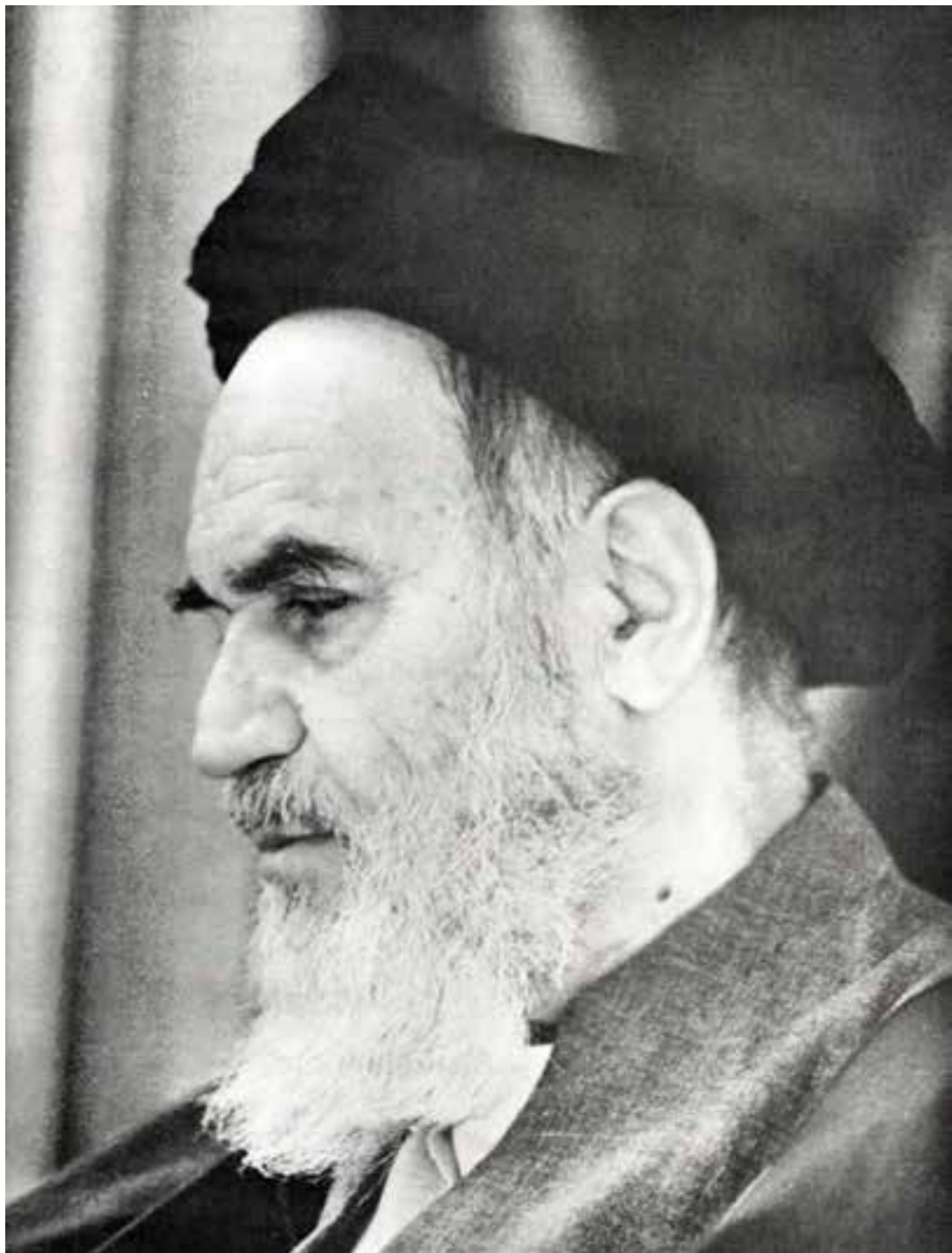
شابک: ۹۷۸-۶۰۰-۱۲۴-۰۵۵-۳

شابک دوره: ۹۷۸-۶۰۰-۱۲۴-۰۵۴-۶

نشانی دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کار دانش صندوق پستی شماره: ۱۵۸۷۵/۴۸۷۴

کلیه حقوق مربوط به تألیف، نشر و چاپ این اثر متعلق به ناشر و سازمان پژوهش و برنامه‌ریزی آموزشی است.

(حق طبع محفوظ است)



بدانید مادام که در احتیاجات صنایع پیشرفته، دست خود را پیش دیگران دراز کنید و به درِ یوزگی عمر را بگذرانید، قدرت ابتکار و پیشرفت در اختراعات در شما شکوفا نخواهد شد.

امام خمینی «قدّس سره الشّریف»

مجموعه کتاب‌های درسی رشته کامپیوتر شاخه کاردانش

(استاندارد وزارت فرهنگ و ارشاد اسلامی)

رشته تصویرسازی	رشته طراحی صفحات وب	رشته تولید چندرسانه‌ای
مفاهیم پایه فناوری اطلاعات	مفاهیم پایه فناوری اطلاعات	مفاهیم پایه فناوری اطلاعات
سیستم‌عامل مقدماتی	سیستم‌عامل مقدماتی	سیستم‌عامل مقدماتی
اطلاعات و ارتباطات	اطلاعات و ارتباطات	اطلاعات و ارتباطات
واژه‌پرداز Word 2007	واژه‌پرداز Word 2007	واژه‌پرداز Word 2007
صفحه گسترده Excel 2007	صفحه گسترده Excel 2007	صفحه گسترده Excel 2007
ارایه مطالب PowerPoint 2007	ارایه مطالب PowerPoint 2007	ارایه مطالب PowerPoint 2007
نرم‌افزارهای اداری تکمیلی	نرم‌افزارهای اداری تکمیلی	نرم‌افزارهای اداری تکمیلی
بانک اطلاعاتی Access 2007	بانک اطلاعاتی Access 2007	بانک اطلاعاتی Access 2007
سیستم‌عامل پیشرفته	سیستم‌عامل پیشرفته	سیستم‌عامل پیشرفته
برنامه‌نویسی مقدماتی	برنامه‌نویسی مقدماتی	برنامه‌نویسی مقدماتی
طراحی امور گرافیکی با رایانه	طراحی امور گرافیکی با رایانه	طراحی امور گرافیکی با رایانه
کاربر FreeHand	کاربر Flash	کاربر Flash
کاربر CorelDraw	طراحی صفحات وب مقدماتی	کاربر Director
	طراحی صفحات وب پیشرفته	میکس رایانه‌ای

مجموعه کتاب‌های درسی رشته کامپیوتر شاخه کار دانش

(استاندارد وزارت کار و امور اجتماعی)

رشته تصویرسازی	رشته طراحی صفحات وب	رشته تولید چندرسانه‌ای	رشته برنامه‌نویسی پایگاه داده
مفاهیم پایه فناوری اطلاعات	مفاهیم پایه فناوری اطلاعات	مفاهیم پایه فناوری اطلاعات	مفاهیم پایه فناوری اطلاعات
سیستم‌عامل مقدماتی	سیستم‌عامل مقدماتی	سیستم‌عامل مقدماتی	سیستم‌عامل مقدماتی
اطلاعات و ارتباطات	اطلاعات و ارتباطات	اطلاعات و ارتباطات	اطلاعات و ارتباطات
سیستم‌عامل پیشرفته	سیستم‌عامل پیشرفته	سیستم‌عامل پیشرفته	سیستم‌عامل پیشرفته
واژه‌پرداز Word 2007	واژه‌پرداز Word 2007	واژه‌پرداز Word 2007	واژه‌پرداز Word 2007
صفحه‌گسترده Excel 2007	صفحه‌گسترده Excel 2007	صفحه‌گسترده Excel 2007	صفحه‌گسترده Excel 2007
ارایه مطالب PowerPoint 2007	ارایه مطالب PowerPoint 2007	ارایه مطالب PowerPoint 2007	ارایه مطالب PowerPoint 2007
نرم‌افزارهای اداری تکمیلی	نرم‌افزارهای اداری تکمیلی	نرم‌افزارهای اداری تکمیلی	نرم‌افزارهای اداری تکمیلی
طراح گرافیک رایانه‌ای	طراح گرافیک رایانه‌ای	طراح گرافیک رایانه‌ای	برنامه‌نویسی مقدماتی
شهروند الکترونیکی	نرم‌افزار گرافیکی Flash Mx	شهروند الکترونیکی	برنامه‌نویسی ویزوال بیسیک پیشرفته (جلد اول)
نرم‌افزار گرافیکی FreeHand	طراحی مقدماتی صفحات وب	نرم‌افزار گرافیکی Director	برنامه‌نویسی ویزوال بیسیک پیشرفته (جلد دوم)
نرم‌افزار گرافیکی CorelDraw	رایانه کار Interdev	تدوین فیلم و صدا SSP	مدیریت پایگاه داده
نرم‌افزار گرافیکی Flash Mx	رایانه کار DreamWeaver	نرم‌افزار گرافیکی Flash Mx	مهارت عمومی برنامه‌نویسی
	رایانه کار CIW	نرم‌افزار گرافیکی Authorware	

فهرست مطالب

۸ مقدمه ناشر

۹ مقدمه

واحدکار ششم: توانایی ارتباط و مدیریت با پایگاه داده‌ها و تهیه انواع گزارش

۱۱ کلیات

۱۲ ۶-۱ زبان SQL (Structured Query Language)

۱۵ ۶-۲ مدیریت پایگاه داده‌ها در ویژوال بیسیک

۹۸ واژه‌نامه

۹۹ خلاصه مطالب

۱۰۲ آزمون نظری

۱۰۴ آزمون عملی

واحدکار هفتم: توانایی ارتباط با نرم‌افزارهای Office رویه‌های API و تهیه برنامه نصب

۱۰۷ کلیات

۱۰۷ ۷-۱ ارتباط ویژوال بیسیک با برنامه‌های Office

۱۱۲ ۷-۲ نحوه استفاده از رویه‌های API

۱۱۸ ۷-۳ نحوه ایجاد برنامه نصب (Setup)

۱۲۸ واژه‌نامه

۱۲۸ خلاصه مطالب

۱۳۰ آزمون نظری

۱۳۱ آزمون عملی

واحدکار هشتم: توانایی استفاده از محیط .Net Framework

۱۳۳ کلیات

۱۳۴ ۸-۱ نحوه ایجاد پروژه در ویژوال بیسیک Net 2008

۱۴۰ ۸-۲ کلاس فرم (Form)

۱۴۳ ۸-۳ کنترل‌ها در ویژوال بیسیک Net

۱۴۸	۸-۴ نحوه ایجاد و اضافه کردن فرم‌های جدید به پروژه
۱۵۶	۸-۵ نحوه خطایابی، رفع خطا و مدیریت خطاها در ویژوال بیسیک Net
۱۵۹	۸-۶ نحوه ایجاد و استفاده از آرایه‌ها
۱۶۰	۸-۷ نحوه ایجاد و استفاده از ساختار یا ساختمان (Structure)
۱۶۱	۸-۸ نحوه ایجاد و استفاده از نوع داده شمارشی (Enumeration)
۱۶۷	۸-۹ نحوه ایجاد و استفاده از مجموعه‌ها (Collections)
۱۶۹	۸-۱۰ نحوه اتصال به پایگاه‌های داده و دسترسی به داده
۱۸۵	واژه‌نامه
۱۸۵	خلاصه مطالب
۱۸۸	آزمون نظری
۱۹۰	آزمون عملی

واحداکار نهم: توانایی برنامه‌نویسی با زبان برنامه‌نویسی دلفی

۱۹۲	کلیات
۱۹۲	۹-۱ نحوه اجرای زبان برنامه‌نویسی دلفی و ایجاد یک پروژه
۱۹۶	۹-۲ ساختار واحد (unit) در زبان دلفی
۱۹۹	۹-۳ انواع داده‌ها در زبان برنامه‌نویسی دلفی
۲۱۵	۹-۴ دستورات شرطی در زبان برنامه‌نویسی دلفی
۲۱۷	۹-۵ دستورات حلقه در زبان برنامه‌نویسی دلفی
۲۲۰	۹-۶ نحوه تعریف توابع در زبان برنامه‌نویسی دلفی
۲۲۳	۹-۷ نحوه تعریف رویه‌ها در زبان برنامه‌نویسی دلفی
۲۲۸	۹-۸ مدیریت خطا در زبان برنامه‌نویسی دلفی
۲۲۹	۹-۹ نحوه ایجاد رابط‌های گرافیکی SDI و MDI
۲۴۶	واژه‌نامه
۲۴۶	خلاصه مطالب
۲۴۸	آزمون نظری
۲۵۰	آزمون عملی
۲۵۱	ضمیمه ۱
۲۵۸	ضمیمه ۲
۲۶۴	پاسخنامه
۲۶۴	منابع

مقدمه ناشر

سپاس بیکران پروردگار را که به انسان قدرت اندیشیدن بخشید تا به یاری این موهبت راه ترقی و تعالی را بییماید و امید به این که عنایات الهی شامل حال ما باشد تا با بضاعت اندک علمی خود در خدمت جوانان و آینده‌سازان کشور عزیزمان باشیم.

یکی از بارزترین ویژگیهای عصر حاضر، حضور گسترده کامپیوتر در کلیه عرصه‌های فعالیت انسان است به گونه‌ای که انجام برخی از کارها، بدون استفاده از کامپیوتر قابل تصور نیست. کامپیوتر به عنوان ابزاری قدرتمند، سرعت و دقت کارها را فوق‌العاده افزایش داده و گذرگاه‌های صعب‌العبور علم را به شاهراه‌های هموار مبدل ساخته است. به همین دلیل در جهان کنونی، آموزش و یادگیری علوم کامپیوتر یک ضرورت اجتناب‌ناپذیر است.

در همین راستا دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش بر اساس موافقت‌نامه‌ای، تألیف کتاب برنامه‌نویسی و ویژوال بیسیک پیشرفته شاخه کاردانش را به مؤسسه فرهنگی هنری دیباگران تهران به‌عنوان یکی از ناشران برگزیده محول کرده که افتخاری بزرگ است. کتاب حاضر با همکاری جمعی از اساتید، متخصصان و مهندسان مجرب رشته کامپیوتر تألیف و محتوای آن در کمیسیون تخصصی برنامه‌ریزی و تألیف کتاب‌های درسی رشته کامپیوتر دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش مورد بررسی و تصویب قرار گرفته است.

طراحی کتاب‌ها براساس ساختار آموزش‌های پیمانهای (مادولار) انجام گرفته و ساختار آن بر اساس توانایی‌های مورد انتظار در استانداردهای مهارتی طراحی شده است. این کتاب‌ها حتی‌المقدور به صورت خودآموز و خود محتوای سازماندهی شده و تلاش بر این است که کتاب‌های آموزش گام به گام، به همراه مثال‌ها، تمرین‌های عملی و کاربردی برای کارهای آزمایشگاهی و کارگاهی، مجموعه منسجمی از هر پیمانانه را ارائه دهد به طوری که دانش‌آموزان پس از پایان هر پیمانانه، از مهارت کافی برای کار با موضوع پیمانانه برخوردار باشند.

در خاتمه ضمن سپاسگزاری از شما دانش‌پژوه گرامی درخواست می‌نماید با مراجعه به آدرس dibagaran.mft.info (ارتباط با مشتری) فرم نظرسنجی را برای کتابی که در دست دارید تکمیل و ارسال نموده، انتشارات دیباگران تهران را که جلب رضایت و وفاداری مشتریان را هدف خود می‌داند، یاری فرمایید.

امیدواریم همواره بهتر از گذشته خدمات و محصولات خود را تقدیم حضورتان نماییم.

مدیر انتشارات

مؤسسه فرهنگی هنری دیباگران تهران

publishing@mftmail.com

مقدمه

کتاب حاضر برای رفع نیازهای آموزشی پیمانان مهارتی برنامه‌نویسی پیشرفته، تألیف و ارائه شده است که لازم است پس از پیمانان مهارتی برنامه‌نویسی مقدماتی مطالعه شود. کتاب از ۹ واحد کار تشکیل شده است که واحد کار اول تا هفتم به بررسی مباحث پیشرفته در زبان برنامه‌نویسی ویژوال بیسیک ۶، واحد کار هشتم به زبان برنامه‌نویسی ویژوال بیسیک Net و واحد کار آخر به زبان برنامه‌نویسی دلفی اختصاص داده شده‌اند. علاوه بر تمرین‌های موجود در هر واحد کار، در پایان هر واحد کار نیز یک آزمون نظری و عملی ارائه شده است تا مطالب مجدداً مرور شوند. در پایان از کلیه همکاران و دانش‌آموزان تقاضا دارم اینجانب را از پیشنهادات و انتقادات ارزشمند خود بهره‌مند نمایند.

مؤلف

VBBOOK6@yahoo.com

توانایی ارتباط و مدیریت با پایگاه داده‌ها

و تهیه انواع گزارش

زمان (ساعت)	
عملی	نظری
۴۵	۱۵

هدف‌های رفتاری

پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:

- ۱- مفاهیم اولیه در پایگاه داده‌ها را بداند.
- ۲- مفهوم سیستم‌های مدیریت پایگاه داده‌ها را بداند و مزایای استفاده از آن‌ها را بیان کند.
- ۳- تفاوت تکنولوژی DAO و OLEDB را توضیح داده و تفاوت آن‌ها را بیان کند.
- ۴- کاربرد کنترل Data را بداند و بتواند از این کنترل و متدها و خاصیت‌های آن استفاده کند.
- ۵- کاربرد کنترل ADO را بداند و بتواند از این کنترل و متدها و خاصیت‌های آن استفاده کند.
- ۶- توانایی استفاده از شیء Recordset را داشته باشد.
- ۷- بتواند کنترل‌ها را به دو روش Bound و UnBound به منبع داده متصل کند.
- ۸- کاربرد متد Find و خاصیت Filter را بداند و تفاوت‌های آن‌ها را بیان کند.
- ۹- کاربرد کنترل‌های DataList و DataCombo را بداند و توانایی استفاده از آن‌ها را داشته باشد.
- ۱۰- کاربرد کنترل DataGrid را بیان کند و نحوه استفاده از آن را بداند.
- ۱۱- اجزای اصلی تشکیل‌دهنده کتابخانه OLEDB را بیان کند و بتواند هریک از آن‌ها را با کاربردشان توضیح دهد.
- ۱۲- بتواند یک Data Environment را ایجاد کند و کاربرد آن را توضیح دهد.
- ۱۳- مفهوم گزارش را توضیح داده و اجزای اصلی در یک گزارش را همراه با وظایف هریک بیان کند.
- ۱۴- بتواند یک گزارش را با Data Report ایجاد کرده و آن را چاپ کند یا نمایش دهد.
- ۱۵- بتواند یک گزارش را با Crystal Report ایجاد کرده و آن را چاپ کند یا نمایش دهد.

کلیات

فایل‌های متنی و باینری بیشتر در مواردی که حجم داده‌ها کم است و بخش عمده‌ای از اطلاعات ذخیره شده در فایل‌ها مورد استفاده قرار می‌گیرند، کاربرد دارند و بیشتر عملیات در رابطه با خواندن و نوشتن داده‌ها در فایل‌ها خلاصه می‌شود. اما در صورتی که مقدار داده‌ها زیاد باشد و نیاز به عملیات دیگری مانند استخراج بعضی از داده‌ها از بین تمام اطلاعات موجود در فایل‌ها، حذف موقت یا دائم بعضی از داده‌ها، ایجاد ارتباط بین داده‌ها در بین فایل‌های داده مختلف و مانند آن‌ها باشد، استفاده از روش‌های قبلی عملی نبوده یا عملکرد برنامه را دچار مشکل می‌کند.

نرم‌افزارها و سیستم‌های مدیریت پایگاه داده رابطه‌ای (Relational DataBase Management)، داده‌ها را در واحدهای مستقل و مرتبط به هم به نام جدول ذخیره و نگهداری می‌کنند تا علاوه بر خواندن و نوشتن سریع‌تر و مطمئن‌تر داده‌ها، امکان استخراج داده‌ها و اطلاعات از یک یا چند جدول و تهیه گزارش‌های متنوع، ساده یا پیچیده به راحتی امکان‌پذیر شود. به علاوه استفاده از این نوع سیستم‌ها مزایای دیگری را نیز به همراه دارند که بعضی از آن‌ها عبارتند از:

- مدیریت آسان‌تر و مطمئن‌تر داده‌ها
- ایجاد محدودیت‌های امنیتی
- به اشتراک‌گذاری داده‌ها و اطلاعات
- امکان استفاده از زبان پرس‌وجوی ساخت یافته SQL
- جستجو و مرتب‌سازی داده‌ها و اطلاعات

تاکنون نرم‌افزارهای سیستم‌های مدیریت پایگاه داده متعددی تولید و در اختیار برنامه‌نویسان و متخصصان پایگاه داده قرار گرفته‌اند که بعضی از آن‌ها دیگر مورد استفاده قرار نمی‌گیرند. در حال حاضر نرم‌افزارهای Access، SQLServer و Oracle از معروف‌ترین نرم‌افزارهای ایجاد و مدیریت پایگاه داده هستند. در این واحد کار از پایگاه‌های داده‌ای که با نرم‌افزار پایگاه داده Access به وجود آمده‌اند، استفاده می‌کنیم. در هر نرم‌افزار پایگاه داده رابطه‌ای مفاهیم مشترکی برای ذخیره‌سازی و نگهداری داده‌ها و اطلاعات به شکل مناسب وجود دارد که در این‌جا مورد بررسی قرار می‌گیرند.

ستون (Column) یا فیلد (Field)

به یک ویژگی مشخص در مورد یک موضوع خاص ستون یا فیلد می‌گویند؛ به عنوان مثال نام یک دانش‌آموز یک ویژگی معین از اطلاعات مربوط به آن دانش‌آموز را بیان می‌کند همین‌طور نام خانوادگی، شماره شناسنامه و مانند آن‌ها.

به عنوان مثالی دیگر کد کارمندی یک کارمند در یک شرکت یا سازمان به عنوان یک فیلد یا ستون مشخصات سازمانی وی را بیان می‌کند، همین‌طور نام، نام خانوادگی و حقوق وی هر یک به عنوان یک ستون

یا فیلد از اطلاعات وی به حساب می‌آیند.

رکورد (Record) یا سطر (Row)

به مجموعه‌ای از فیلدها در رابطه با یک موضوع خاص، یک سطر یا رکورد می‌گویند. به عنوان مثال مجموعه ستون‌های کد دانش‌آموزی، نام، نام خانوادگی و رشته تحصیلی یک دانش‌آموز معین، سطر یا رکورد اطلاعاتی وی را تشکیل می‌دهند.

همین‌طور ستون‌های کد کارمندی، نام، نام خانوادگی و حقوق هر کارمند در یک شرکت یا سازمان سطر یا رکورد اطلاعاتی وی را تشکیل می‌دهند.

جدول (Table)

به مجموعه‌ای از سطرها (رکوردهای) مربوط به هم، جدول یا Table می‌گویند. یک جدول در واقع مجموعه‌ای از چندین سطر و ستون است. به عنوان مثال برای نگهداری اطلاعات چندین دانش‌آموز می‌توان رکوردهای اطلاعاتی آن‌ها را در یک جدول با نام دانش‌آموزان در کنار یکدیگر قرار داد یا رکوردهای اطلاعاتی کارمندان یک شرکت را در یک جدول به صورت یک مجموعه واحد نگهداری نمود. به عبارت دیگر یک جدول امکان طبقه‌بندی منطقی مجموعه‌ای از رکوردهای مربوط به هم را فراهم می‌کند.

پایگاه داده‌ها (DataBase)

به مجموعه‌ای از جدول‌ها پایگاه داده می‌گویند.

نکته: نرم‌افزارهای مدیریت پایگاه داده در یک پایگاه داده با توجه به اهدافی که برای آن طراحی شده‌اند علاوه بر جدول ممکن است توانایی ایجاد و استفاده از اجزای دیگری را نیز فراهم کنند که با سایر نرم‌افزارها متفاوت باشد. به عنوان مثال در نرم‌افزار Access علاوه بر جدول می‌توان انواع Query, Form, Report و مانند آن‌ها را در پایگاه داده ایجاد کرد و مورد استفاده قرار داد.

۱-۶ زبان SQL (Structured Query Language)

SQL یک زبان برای اجرای انواع پرس‌وجوها و بازیابی داده و اطلاعات ذخیره شده در پایگاه‌های داده رابطه‌ای (Relational Database) است. زبان SQL امکان بازیابی، مرتب‌سازی و جداسازی داده‌ها و اطلاعات را نیز از پایگاه‌های داده فراهم می‌کند؛ به علاوه در این زبان امکان درج، حذف یا تغییر داده‌ها و اطلاعات در پایگاه داده وجود دارد.

۱-۱-۶ دستور Select

با استفاده از این دستور می‌توان یک یا مجموعه‌ای از سطرها را از پایگاه داده موردنظر استخراج نمود. دستور

Select از بخش‌های مختلفی تشکیل می‌شود که هر یک وظیفه‌ای را به عهده دارند؛ شکل کلی این دستور به این صورت است:

نام جدول From نام ستون(ها) Select

Where Order By

در این دستور پس از کلمه کلیدی Select نام یک یا چند ستون که لازم است بازیابی شوند قرار می‌گیرند و بین نام ستون‌ها کاراکتر کاما (,) قرار داده می‌شود. اگر لازم باشد کلیه ستون‌های سطرها بازیابی شوند می‌توان از کاراکتر ستاره (*) استفاده نمود.

به عنوان مثال این دستور اطلاعات کلیه کارمندان را از جدول employees بازیابی می‌کند.

Select * From employees

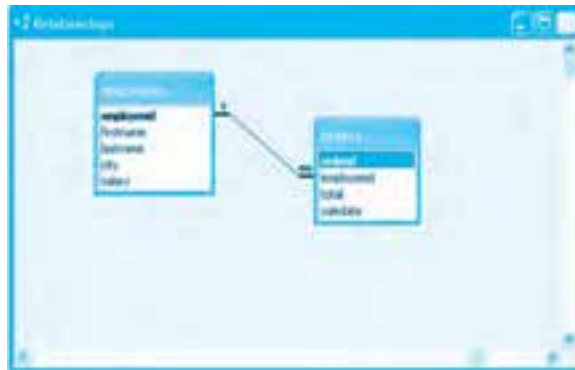
در بخش Where می‌توان شرط یا شرط‌هایی را که براساس آن‌ها سطرها فیلتر و جداسازی می‌شوند، مشخص کرد.

۲-۱-۶ شرط Where

گاهی اوقات لازم است تا فقط بعضی از رکوردها که شرایط خاصی دارند بازیابی شوند در این صورت می‌توان از بخش Where در دستور Select استفاده کرد شکل کلی نحوه استفاده از بخش Where به این صورت است.

نام جدول form نام ستون (ها) Select

Where شرط (ها)



شکل ۱-۶

توجه: در این واحد کار تصور بر این است که دانش‌آموز با نرم‌افزار Access آشنایی کافی دارد، بنابراین لازم است به منظور انجام مثال‌ها و تمرین‌ها یک پایگاه داده با نام company که شامل دو جدول با نام‌های employees و orders باشد روی سیستم خود ایجاد نمایید. این پایگاه داده اطلاعات کارمندی را که در یک

شرکت خرید و فروش کالا فعالیت دارند، مدیریت می‌کند. برای راهنمایی بیشتر به شکل ۱-۶ و جدول‌های ۱-۶ و ۲-۶ مراجعه کنید.

جدول ۱-۶ ساختار جدول employees

نام ستون	نوع داده	داده
employeeid	Long Integer	کد کارمند
firstname	(Text 25)	نام کارمند
lastname	(Text 30)	نام خانوادگی کارمند
salary	Currency	حقوق
City	(Text 20)	محل تولد

جدول ۲-۶ ساختار جدول Orders

نام ستون	نوع داده	داده
Orderid	Long Integer	شماره سفارش (فاکتور)
employeeid	Long Integer	کد کارمند
total	Currency	جمع فروش (فاکتور)
saledate	Date/Time	تاریخ فروش

به عنوان مثالی دیگر این دستور فقط ستون‌های نام، نام خانوادگی و حقوق کارمندان را بازیابی می‌کند.

```
Select firstname, lastname, salary
```

```
From employees
```

با استفاده از بخش Where در دستور Select می‌توان سطر را فیلتر نمود؛ به عنوان مثال این دستور مشخصات کارمندانی را که حقوق آن‌ها کمتر از ۶۰۰۰۰۰۰ ریال است بازیابی می‌کند.

```
Select firstname, lastname
```

```
From employees
```

```
Where salary < 6000000
```

در بخش Where می‌توانید از عملگر In برای انجام چند مقایسه به‌طور همزمان استفاده کنید؛ به عنوان مثال این دستور مشخصات کارمندانی را نشان می‌دهد که محل تولد آن‌ها تهران یا تبریز باشد.

Select firstname, lastname

From employees

Where city In ("tehran" , "tabriz")

۳-۱-۶ بخش Order By

با استفاده از بخش Order By در دستور Select می‌توان رکوردها را براساس ستون موردنظر مرتب نمود. شکل کلی نحوه استفاده از بخش Order By به این صورت است:

Select , نام ستون (ها) From جدول نام

Where شرط (ها)

Order By نام ستون (ها)

به عنوان مثال این دستور مشخصات کلیه کارمندان را به‌صورت مرتب شده براساس نام خانوادگی بازیابی می‌کند.

Select * From employees

Order By LastName

۲-۶ مدیریت پایگاه داده‌ها در ویژوال بیسیک

ویژوال بیسیک امکان استفاده از انواع پایگاه‌های داده رابطه‌ای رایج را به راحتی فراهم می‌کند و تکنولوژی‌ها و کنترل‌های متعددی را به منظور انجام عملیات بازیابی، ویرایش و ثبت داده و اطلاعات ارائه می‌کند که به بررسی هریک از آن‌ها می‌پردازیم.

مدل (Data Access Objects) DAO اولین تکنولوژی است که در ویژوال بیسیک به منظور دسترسی به داده‌ها و اطلاعات در پایگاه‌های داده محلی ارائه می‌شود.


در این روش از یک تکنولوژی با نام Microsoft Joint Engine Technology که به اختصار به آن Jet می‌گویند، استفاده می‌شود.

برای ارتباط با پایگاه‌های داده‌ای که به وسیله نرم‌افزارهایی مانند Access، Excel، Lotus، Foxpro و مانند آن‌ها ایجاد می‌شوند از تکنولوژی Jet استفاده می‌شود. کنترل Data در ویژوال بیسیک با استفاده از مدل DAO امکان اتصال به این نوع پایگاه‌های داده را فراهم می‌کند.





۱-۲-۶ کنترل Data


این کنترل جزء کنترل‌های ذاتی ویژوال بیسیک می‌باشد. برای استفاده از این کنترل می‌توان به پایگاه‌های

داده‌ای که به وسیله نرم‌افزارهایی مانند Access، Foxpro، Lotus، Excel یا DBase ایجاد شده‌اند متصل شد و از آن‌ها استفاده کرد.

 **نکته:** ممکن است کنترل داده نتواند با پایگاه‌های داده‌ای که با نگارش‌های جدید این نرم‌افزارها ایجاد شده‌اند ارتباط برقرار کنند، در این صورت یا باید این نوع پایگاه‌های داده را با قالب‌بندی نگارش‌های قبل از سال ۲۰۰۰ مثل نگارش Access 97 تبدیل کنید یا برای ایجاد پایگاه‌های داده جدید از امکانات داخلی ویژوال بیسیک استفاده نمایید. برای این کار می‌توانید از Visual Data Manager در منوی Add-Ins استفاده کنید. نحوه استفاده از این برنامه در ضمیمه کتاب توضیح داده شده است.

برای این کار می‌توانید به این ترتیب عمل کنید:


- ۱- یک کنترل Data روی فرم قرار دهید. برای این کار در جعبه ابزار ویژوال بیسیک روی آیکن  دو بار کلیک کنید.
- ۲- در پنجره خواص روی خاصیت Connect و سپس روی دکمه  کلیک کرده و از کادر لیستی که نمایش داده می‌شود نوع پایگاه داده خود را انتخاب کنید.
- ۳- اکنون در پنجره خواص روی خاصیت DatabaseName و سپس روی دکمه  کلیک کنید و در کادرمحاوره‌ای که نمایش داده می‌شود نام فایل پایگاه داده خود را پیدا کرده و آن را انتخاب کنید. این خاصیت نام و مسیر فایل پایگاه داده را مشخص می‌کند.
- ۴- در این مرحله باید نام یک جدول، یک دستور Select یا Query را از پایگاه داده تعیین شده انتخاب کرد. برای این کار در پنجره خواص ابتدا روی خاصیت RecordSource و سپس روی دکمه  روبه‌روی آن کلیک کنید و از لیستی که نمایش داده می‌شود جدول خود را انتخاب کنید. خاصیت RecordSource منبع داده را مشخص می‌کند.
- ۵- اکنون ارتباط کنترل Data با پایگاه داده برقرار شده است.

 **مثال ۱:** می‌خواهیم پروژه‌ای طراحی کنیم که به وسیله آن بتوان اطلاعات مربوط به کارمندان را در پایگاه داده company بازیابی و مشاهده نمود؛ برای انجام این مثال، عملیات بعدی را انجام دهید:

- ۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE همراه با یک فرم با نام frmemployee و عنوان Employees ایجاد کنید.
- ۲- یک کنترل Data با نام dcmemployee و چند کنترل کادر متن مانند شکل ۲-۶ روی فرم قرار دهید، سپس عنوان کنترل Data را روی مقدار employees تنظیم کنید تا شکل ظاهری آن مناسب‌تر باشد.



شکل ۲-۶

- ۳- در این مرحله روی کنترل داده کلیک کنید، سپس خاصیت Connect را روی مقدار Access و خاصیت DatabaseName را روی نام فایل پایگاه داده company.mdb تنظیم کنید.
- ۴- اکنون روی خاصیت RecordSource و سپس روی دکمه  روبه‌روی آن کلیک کنید و از کادر لیستی که باز می‌شود جدول employees را انتخاب کنید. در این مرحله ارتباط کنترل داده با پایگاه داده‌ها و جدول employees برقرار شده است.
- ۵- کنترل داده نمی‌تواند به تنهایی محتویات سطرها و ستون‌ها را نمایش دهد. برای این کار باید از کنترل‌های دیگر مانند کنترل کادر متن استفاده کرد، بنابراین اولین کادر متن از بالای فرم را انتخاب کرده و در پنجره خواص روی خاصیت DataSource آن کلیک کنید. این خاصیت منبع داده را برای کنترل کادر متن مشخص می‌کند. روی دکمه  روبه‌روی این خاصیت کلیک کرده و از کادر لیستی که باز می‌شود نام کنترل داده یعنی dcmemployee را انتخاب کنید.
- ۶- اکنون روی خاصیت DataField کنترل کادر متن txtid کلیک کنید، سپس روی دکمه  روبه‌روی خاصیت کلیک کرده و از کادر لیستی که نمایش داده می‌شود، نام ستون employeeid را انتخاب کنید؛ در این حالت کنترل کادر متن به ستون employeeid متصل می‌شود. به این عمل Bound کردن کنترل به منبع داده نیز می‌گویند.
- ۷- سایر کنترل‌های کادر متن را مانند کادر متن txtid به ستون‌های متناظر با آن‌ها متصل کنید.
- ۸- فرم را با نام employee و پروژه را با نام company ذخیره کنید.
- ۹- پروژه را اجرا کنید، در این لحظه مقدار ستون‌های سطر اول جدول employees در کنترل‌های کادر متن

مربوطه نمایش داده می‌شود (شکل ۳-۶).



شکل ۳-۶

۱۰- در کنترل داده چهار دکمه وجود دارد که با استفاده از آن‌ها می‌توان روی رکوردها حرکت کرد (جدول ۳-۶).

جدول ۳-۶ عملکرد دکمه‌های کنترل

عملکرد	دکمه
یک رکورد به جلو	
یک رکورد به عقب	
اولین رکورد	
آخرین رکورد	

۱۱- در این مرحله به رکورد اول بروید و شهر وی را به kashan تغییر دهید و به اجرای برنامه خاتمه داده و برنامه را دوباره اجرا کنید و به رکورد اول رفته و نام شهر کارمند اول را بررسی کنید. نام شهر از مقدار tehran به kashan تغییر کرده است، بنابراین با این روش می‌توان علاوه بر بازیابی داده‌ها، آن‌ها را ویرایش نمود.

۱۲- اجرای برنامه را متوقف کرده و به پنجره ویژوال بیسیک بازگردید.

تمرین: یک پروژه جدید طراحی کنید که به وسیله آن بتوان داده و اطلاعات جدول orders را

مشاهده نمود.

شیء Recordset

Recordset به مجموعه‌ای از چندین رکورد گفته می‌شود. با استفاده از ویژگی‌های این شیء می‌توان بعضی از عملیات را روی رکوردها و جدول‌ها انجام داد. هر کنترل داده دارای یک خاصیت از نوع Recordset است که به وسیله آن می‌توان روی رکوردها حرکت کرده یا سطرها را حذف، ویرایش یا درج کرد.

جدول ۴-۶ متدهای مربوط به شیء Recordset

نام متد	توضیح	مثال
MoveFirst	حرکت به اولین رکورد	Data1.Recordset Move Firest
MoveLast	حرکت به آخرین رکورد	Data1.Recordset Move Last
MoveNext	حرکت به رکورد بعدی	Data1.Recordset.Move Next
MovePrevious	حرکت به رکورد قبلی	Data1.Recordset.Move Previous
AddNew	درج یک سطر جدید	Data1.Recordset.Add New
Delete	حذف یک سطر	Data1.Recordset Delete
Update	ویرایش یک سطر	Data1.Recordset.Update

مثال ۲: می‌خواهیم پروژه company را به شکلی تنظیم کنیم که مانند شکل ۴-۶ بتوان با استفاده از دکمه فرمان روی رکوردها حرکت نمود؛ برای انجام این مثال، به ترتیبی که در ادامه می‌آید، عمل کنید:

۱- برنامه ویژوال بیسیک را اجرا کرده و پروژه company را باز کنید و خاصیت Visible کنترل داده dcemployee را روی مقدار False تنظیم کنید تا کنترل داده در زمان اجرا مشاهده نشود.

۲- چهار کنترل دکمه فرمان مانند شکل ۴-۶ روی فرم قرار داده و خاصیت Name و Caption آن‌ها را تنظیم کنید.

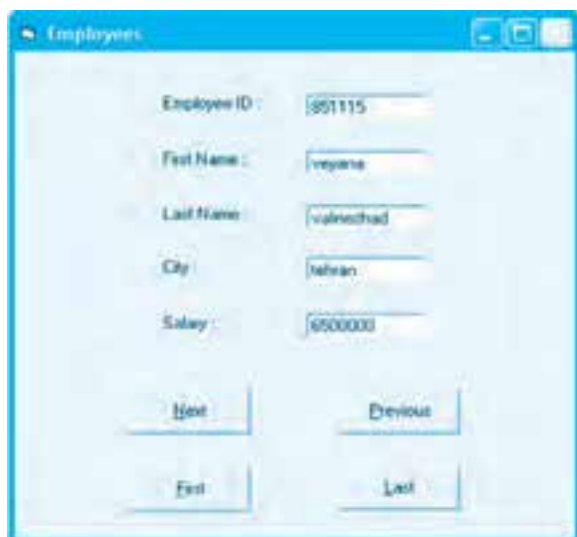
۳- رویداد Click دکمه فرمان cmdnext را به این صورت تنظیم کنید:

```
Private Sub cmdnext_Click()
```

```
    If Not dcemployee Recordset.EOF Then dcemployee.Recordset MoveNext
```

```
    If dcemployee Recordset.EOF Then dcemployee.Recordset MoveLast
```

```
End Sub
```



شکل ۴-۶

در این رویداد با استفاده از متد MoveNext شیء Recordset امکان حرکت به جلو روی سطرها به وجود می‌آید، اما اگر بعد از عبور از آخرین رکورد این متد اجرا شود، پیام خطایی مانند شکل ۵-۶ نمایش داده می‌شود.



شکل ۵-۶

برای جلوگیری از این خطا می‌توان رویداد کلیک دکمه فرمان Cmdnext را به این صورت تنظیم کرد و در این رویداد از دستور شرطی If همراه با خاصیت EOF شیء Recordset استفاده نمود.

```
Private Sub cmdnext_Click()
```

```
    If Not dcemployee.Recordset.EOF Then dcemployee.Recordset.MoveNext
```

```
    If dcemployee.Recordset.EOF Then dcemployee.Recordset.MoveLast
```

```
End Sub
```

در صورتی که از آخرین سطر عبور کنید خاصیت EOF روی مقدار True تنظیم می‌شود، در غیر این صورت یعنی وقتی

هنوز از آخرین سطر عبور نکرده‌اید، مقدار این خاصیت False می‌باشد. بنابراین متد MoveNext زمانی اجرا می‌شود که از آخرین سطر عبور نکرده‌اید زیرا از عملگر منطقی Not همراه با خاصیت EOF در دستور If استفاده شده است. دستور دوم در این رویداد نیز از یک دستور شرطی If و خاصیت EOF شیء Recordset کنترل داده استفاده کرده است. از این دستور به این دلیل استفاده شده است تا وقتی رکورد جاری در Recordset، آخرین رکورد باشد هنوز خاصیت EOF روی مقدار False تنظیم شده است، در نتیجه کلیک روی دکمه فرمان Next سبب اجرای دوباره متد MoveNext می‌شود که باعث می‌شود مقدار خاصیت EOF روی True تنظیم شود؛ اما در این لحظه کادرهای متن به صورت خالی نمایش داده می‌شوند. برای جلوگیری از این حالت دستور If دوم نوشته شده است تا با اجرای متد MoveLast رکورد جاری را دوباره به آخرین سطر Recordset بازگرداند تا رابط گرافیکی به شکل مناسبی کار کرده و داده‌ها را نمایش دهد.

۴- رویداد Click دکمه فرمان cmdprevious را به این صورت تنظیم کنید:

```
Private Sub cmdprevious_Click()
```

```
    If Not dcmmployee.Recordset.BOF Then dcmmployee.Recordset.MovePrevious
```

```
    If dcmmployee.Recordset.BOF Then dcmmployee.Recordset.MoveFirst
```

```
End Sub
```

در این رویداد نیز با استفاده از متد MovePrevious شیء Recordset رکورد جاری به عقب حرکت داده می‌شود اما با رسیدن به اولین سطر و عبور از آن برنامه متوقف شده و پیام خطایی مانند شکل ۵-۶ نمایش داده می‌شود. برای جلوگیری از این خطا می‌توان از خاصیت BOF شیء Recordset استفاده کرد تا در صورتی که سطر جاری از سطر اول عقب‌تر برود، تابع BOF روی مقدار True تنظیم شده و از اجرای متد MovePrevious جلوگیری شود و برای آن که وقتی از سطر اول عقب‌تر می‌روید کادرهای متن به صورت خالی نمایش داده نشوند، یک دستور If دیگر مقدار خاصیت BOF را بررسی می‌کند تا در صورتی که مقدار آن True باشد سطر جاری را با استفاده از MoveNext دوباره روی سطر اول تنظیم کند.

۵- در این مرحله نیز رویدادهای Click دکمه‌های فرمان First و Last را به این صورت تنظیم کنید:

```
Private Sub cmdfirst_Click()
```

```
    dcmmployee.Recordset.MoveFirst
```

```
End Sub
```

```
Private Sub cmdlast_Click()
```

```
    dcmmployee.Recordset.MoveLast
```

```
End Sub
```

در این رویدادها با استفاده از متدهای MoveFirst و MoveLast شیء Recordset امکان تغییر رکورد جاری به رکورد اول یا آخر فراهم می‌شود.

۶- تغییرات را ذخیره کرده و پروژه را اجرا، آزمایش و نتیجه را با حالت قبل مقایسه کنید.

۷- به اجرای برنامه خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.

نکته: خاصیت Recordset Type در کنترل داده می‌تواند عملکرد شیء Recordset را در کنترل مشخص کند. اگر این خاصیت روی مقدار Dynaset-1 تنظیم شود رکوردهای موجود در Recordset همگام با پایگاه داده تغییر می‌کنند و برعکس، یعنی اگر سطر یا ستونی در پایگاه داده ویرایش، اضافه یا حذف شود، این تغییرات بلافاصله روی رکوردهای موجود در Recordset اعمال می‌شود و اگر عملیاتی روی رکوردهای Recordset در کنترل داده صورت گیرد، مستقیماً روی پایگاه داده اعمال می‌شود. اما اگر این خاصیت روی مقدار Snapshot-2 تنظیم شود فقط یک کپی از رکوردهای پایگاه داده در اختیار کنترل داده قرار می‌گیرد و هرگونه تغییرات در پایگاه داده هیچ تأثیری روی Recordset نمی‌گذارد و اگر رکوردهای Recordset ویرایش، حذف یا اضافه شوند این تغییرات روی پایگاه داده اثری نخواهد گذاشت. اگر این خاصیت روی مقدار Table-0 تنظیم شود، Recordset عملکردی مشابه حالت Dynaset خواهد داشت. مقدار پیش فرض این خاصیت Dynaset می‌باشد.

تمرین: پروژه Company را به شکلی تنظیم کنید تا اطلاعات کارمندان فقط جنبه نمایشی داشته باشد و تغییر در آن‌ها سبب تغییر اطلاعات در جدول employees نشود.

مثال ۳: می‌خواهیم یک پروژه طراحی کنیم تا به وسیله آن بتوان عملیات درج، ویرایش و به‌روزرسانی سطرها را در جدول employees انجام داد. به این منظور عملیات بعدی را به ترتیب انجام دهید:

۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE همراه یک فرم با نام frmemployee و عنوان Employees مانند شکل ۶-۶ ایجاد کنید.



شکل ۶-۶

- ۲- یک کنترل داده با نام dcmemployee روی فرم قرار دهید و خاصیت DatabaseName آن را روی پایگاه داده company و خاصیت RecordSource آن را روی جدول employees تنظیم کنید.
- ۳- در این مرحله سایر کنترل کادر متن، برچسب و دکمه فرمان را مانند شکل ۶-۶ روی فرم قرار دهید و خاصیت‌های آن‌ها را تنظیم کنید.
- ۴- خاصیت DataSource کنترل‌های کادر متن را روی کنترل dcmemployee و خاصیت DataField هر یک را روی فیلد مربوطه تنظیم کنید.
- ۵- رویداد Click دکمه فرمان Add و Update را به این صورت تنظیم کنید:

```
Private Sub cmdAdd_Click()
```

```
    dcmemployee.Recordset.AddNew
```

```
End Sub
```

```
Private Sub cmdUpdate_Click()
```

```
    dcmemployee.Update
```

```
End Sub
```

در این رویدادها از متد Add شیء Recordset برای درج یک رکورد جدید و از متد UpdateRecord آن برای به‌روز رسانی رکوردها استفاده شده است.

- ۶- رویداد Click دکمه فرمان Delete را به این صورت تنظیم کنید:

```
Private Sub cmdDelete_Click()
```

```
    dcmemployee.Recordset.Delete
```

```
    dcmemployee.Refresh
```

```
End Sub
```

در این رویداد ابتدا رکورد جاری با استفاده از متد Delete شیء Recordset حذف می‌شود، سپس متد Refresh کنترل داده dcmemployee اجرا می‌شود؛ در این صورت محتویات شیء Recordset مجدداً با رکوردهای جدول پر می‌شود تا رکورد حذف شده پس از حذف در کنترل‌های کادر متن مشاهده نشود.

- ۷- از متد Refresh در رویداد Click دکمه فرمان Refresh نیز استفاده کنید تا هر وقت که کاربر می‌خواهد، بتواند داده‌های شیء Recordset را به‌روز کند، بنابراین رویداد دکمه فرمان Refresh را به این صورت تنظیم کنید:

```
Private Sub cmdRefresh_Click()
```

```
    dcmemployee.Refresh
```

```
End Sub
```

- ۸- در این مرحله رویداد Reposition کنترل dcmemployee را به این صورت تنظیم کنید:

Private Sub dcmemployee_Reposition()

dcmemployee.Caption = "Record: " & (dcmemployee.Recordset.AbsolutePosition + 1)

End Sub

این رویداد زمانی اجرا می‌شود که رکورد جاری در کنترل داده تغییر می‌کند. در این رویداد با استفاده از خاصیت AbsolutePosition از شیء Recordset شماره سطر جاری نمایش داده شده و در عنوان کنترل داده نمایش داده می‌شود، بنابراین هر زمان که رکورد جاری تغییر کند، شماره رکورد فعلی در کنترل داده نمایش داده می‌شود. البته چون شماره رکوردها از صفر شروع می‌شود همواره یک واحد به آن اضافه می‌شود تا رابط گرافیکی با کاربر هماهنگ باشد. برای آن که بلافاصله پس از اجرای برنامه شماره رکورد جاری نمایش داده شود.


۹- پروژه و فرم را با نام دلخواه ذخیره و سپس آن را اجرا کنید.

۱۰- به رکورد آخر بروید و روی دکمه فرمان Add کلیک کنید. محتویات کادرهای متن به‌طور خودکار خالی می‌شود، در این لحظه اطلاعات یک کارمند جدید را در کادرهای متن تایپ کنید، سپس روی رکوردها حرکت کرده و رکوردهای قبلی و رکورد جدید را بررسی کنید.

۱۱- به رکورد دوم بروید و مقدار حقوق این کارمند را به مقدار دلخواهی تغییر دهید، سپس روی دکمه Update کلیک کنید تا تغییرات در شیء Recordset و پایگاه داده اعمال شود. نتیجه را بررسی کنید.

۱۲- به رکورد جدیدی که درج کرده‌اید بروید و روی دکمه فرمان Delete کلیک کنید تا حذف شود سپس نتیجه را بررسی کنید. در این حالت دوباره رکورد اول نمایش داده می‌شود.

۱۳- اجرای پروژه را متوقف کرده و به پنجره ویژوال بیسیک بازگردید.

 **تمرین:** این مثال را به شکلی تنظیم کنید تا پس از حذف رکورد جاری، رکورد بعد از آن نمایش داده شود و در صورتی که آخرین رکورد حذف شود رکورد قبل از آن نمایش داده شود. به‌علاوه با رسیدن به آخرین رکورد یک کادر پیغام این موضوع را به کاربر اطلاع دهد.

ارتباط کنترل‌ها به‌صورت غیرمتصل با شیء Recordset

در این روش محتویات هریک از فیلدها در رکورد جاری به‌صورت غیرمستقیم در کنترل موردنظر قرار داده می‌شود، بنابراین تغییر داده‌ها در کنترل برخلاف روش bound سبب ایجاد تغییر در پایگاه داده‌ها نخواهد شد. برای ایجاد این نوع ارتباط که به آن روش unbound نیز می‌گویند، می‌توان از ویژگی‌های شیء Recordset برای دسترسی به فیلدهای رکورد جاری استفاده نمود. نحوه دسترسی به محتویات یک ستون از رکورد جاری در شیء Recordset به این صورت امکان‌پذیر است:

نام ستون! Recordset. نام کنترل Data

به عنوان مثال این دستور محتویات ستون کد کارمند را در کنترل کادر متن نمایش می‌دهد.

txtid.Text = dcmemployee.Recordset.employeeid

مثال ۴: می‌خواهیم پروژه company را به شکلی تنظیم کنیم که با استفاده از روش unbound داده‌ها و اطلاعات جدول employees را بازیابی نماید. برای این کار این عملیات را به ترتیب انجام دهید:

- ۱- برنامه ویژوال بیسیک را اجرا کرده و پروژه company را باز کنید.
- ۲- کنترل داده موجود را حذف کرده و یک کنترل داده جدید با نام dcmemployee روی فرم قرار دهید، سپس محتویات خاصیت DataField تمام کنترل‌های کادر متن را حذف کنید.
- ۳- در این مرحله رویداد Load فرم را به این صورت تنظیم کنید:

```
Private Sub Form_Load()
```

```
    dcmemployee.DatabaseName = "company.mdb"
```

```
    dcmemployee.RecordSource = "employees"
```

```
    dcmemployee.RecordsetType = vbRSTypeDynaset
```

```
End Sub
```

در این رویداد خاصیت‌های مربوط به کنترل داده با استفاده از کد تنظیم می‌شوند. خاصیت‌های DatabaseName و RecordSource از نوع رشته‌ای می‌باشند و با این روش پایگاه داده company.mdb و جدول employees برای این دو خاصیت در نظر گرفته شده‌اند، سپس خاصیت RecordsetType نیز روی حالت Dynaset تنظیم می‌شود تا کنترل داده به‌صورت پویا همراه با پایگاه داده تغییر کند.

- ۴- اکنون رویداد Reposition کنترل داده یعنی dcmemployee را به این صورت تنظیم کنید:

```
Private Sub dcmemployee_Reposition()
```

```
    If Not dcmemployee.Recordset.EOF And Not dcmemployee.Recordset.BOF Then
```

```
        txtid.Text = dcmemployee.Recordset ! employeeid
```

```
        txtfirstname.Text = dcmemployee.Recordset ! firstname
```

```
        txtlastname.Text = dcmemployee.Recordset ! lastname
```

```
        txtcity.Text = dcmemployee.Recordset ! city
```

```
        txtsalary.Text = dcmemployee.Recordset ! salary
```

```
    End If
```

```
End Sub
```

در این رویداد که هر بار با تغییر رکورد جاری اجرا می‌شود، فیلدهای رکورد جاری به‌صورت غیرمستقیم (unbound) در کنترل‌های کادر متن نمایش داده می‌شوند و برای جلوگیری از ایجاد خطا وقتی از رکورد آخر یا اول عبور می‌کنید از دستور شرطی If و خاصیت‌های EOF و BOF شیء RecordSet استفاده می‌شود تا رکوردها به درستی در کادرهای متن نمایش داده شوند. در این حالت با هر بار تغییر رکورد جاری که به وسیله دکمه‌های فرمان Next، Previous، Last یا First صورت می‌گیرد رویداد Reposition اجرا شده و کادرهای متن رکورد جاری را نمایش می‌دهند.

۵- تغییرات را ذخیره کرده و سپس آن را اجرا کنید و با استفاده از دکمه‌های فرمان رکوردهای جدول employees را مشاهده کنید و نتیجه را با حالت قبل بررسی کنید.

۶- به اجرای پروژه خاتمه داده و به پنجره ویژوال بیسیک بازگردید.

تمرین: پروژه‌های طراحی کنید که رکوردهای جدول سفارشات (orders) را به صورت unbound نمایش دهد.

مثال ۵: می‌خواهیم پروژه company را به شکلی تنظیم کنیم تا بتوان اطلاعات کارمندان شهر تهران را مشاهده نمود؛ برای این کار این عملیات را به ترتیب انجام دهید:

۱- برنامه ویژوال بیسیک را اجرا کرده و پروژه company را باز کنید.

۲- به رویداد Load فرم بروید و آن را به این صورت تنظیم کنید:

Private Sub Form_Load()

```
dcemployee DatabaseName = "company.mdb"
```

```
dcemployee RecordSource = "Select * from employees where city= 'tehran'"
```

```
dcemployee RecordsetType = vbRSTypeDynaset
```

End Sub

در این رویداد دستور اول و سوم هیچ تغییری نکرده‌اند اما دستور دوم که خاصیت RecordSource را تغییر می‌دهد، عوض شده است. این خاصیت می‌تواند علاوه بر دریافت نام یک جدول یک دستور Select را نیز بپذیرد با این روش می‌توان دستور Select و در نتیجه کنترل داده را مطابق نیاز تنظیم نمود. در دستور دوم یک دستور Select در خاصیت RecordSource قرار داده می‌شود تا مشخصات کارمندی را که متولد تهران می‌باشند، بازیابی کند.

۳- تغییرات را ذخیره کرده و پروژه را اجرا و نتیجه را مشاهده و بررسی کنید.

۴- به اجرای پروژه پایان دهید و به پنجره ویژوال بیسیک بازگردید.

تمرین: پروژه‌های طراحی کنید که مشخصات فاکتورهایی را نمایش دهد که مبلغ آن‌ها از یک مقدار مشخص بیشتر باشد.

تمرین: پروژه company را به شکلی تنظیم کنید که بتوان اسامی کارمندان متولد هر شهر دلخواه را مشاهده نمود.

۲-۲-۶ کنترل (ADO (ActiveX Data Object

ویژوال بیسیک در نگارش ۶، کنترل جدید دیگری به نام ADO را با قابلیت‌های ویژه‌ای ارائه می‌کند که امکان ارتباط با انواع پایگاه داده را فراهم می‌کند. این کنترل از تکنولوژی OLEDB استفاده می‌کند. OLEDB یک رابط برای دسترسی به منابع اطلاعاتی و پایگاه‌های داده می‌باشد؛ صرف‌نظر از این که در چه محلی قرار دارند یا از چه نوعی هستند. OLEDB این امکان را فراهم می‌کند تا برنامه‌ها به آسانی به منابع

داده و اطلاعات دسترسی پیدا کنند.

کنترل ADO با استفاده از تکنولوژی OLEDB امکان ارتباط با انواع پایگاه‌های داده محلی مانند Access یا پایگاه‌های داده روی کامپیوترهای سرویس‌دهنده مانند SQL Server و Oracle را فراهم می‌کند. از این کنترل می‌توان مستقیماً در برنامه استفاده کرد و با استفاده از دکمه‌های موجود در رابط گرافیکی آن روی داده‌ها و اطلاعات (رکوردها) حرکت نمود. این کنترل یک ارتباط سریع بین پایگاه داده و انواع کنترل‌ها مانند کادر متن، کادر علامت، کادر لیست، کادر لیست ترکیبی بازشو، کنترل برجسب و سایر کنترل‌های مشابه ایجاد می‌کند. برای اضافه کردن این کنترل به جعبه ابزار ویژوال بیسیک در کادرمحاوره Components گزینه Microsoft ADO Data Control 6.0 (OLEDB) را انتخاب کنید (شکل ۶-۷).



شکل ۶-۷

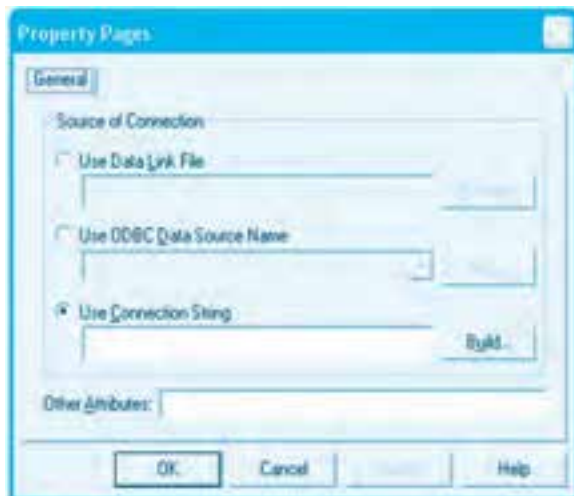
سپس روی دکمه فرمان OK کلیک نمایید تا کنترل ADO به جعبه ابزار ویژوال بیسیک اضافه شود (شکل ۶-۸).



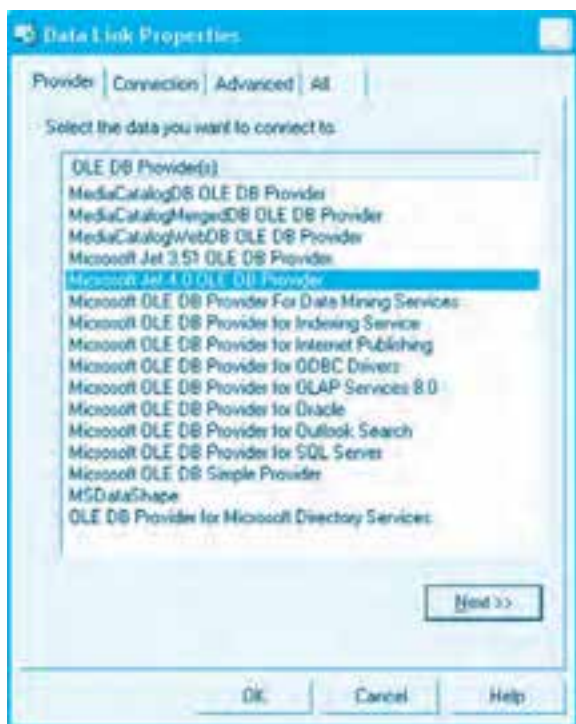
شکل ۶-۸

برای اتصال این کنترل به پایگاه‌های داده موردنظر می‌توانید در زمان طراحی به این ترتیب عمل کنید:
 ۱- در پنجره خواص ابتدا روی خاصیت `ConnectionString` و سپس روی دکمه `...` روبه‌روی آن کلیک

کنید تا کادرمحاوره Property Pages مانند شکل ۹-۶ نمایش داده شود.




شکل ۹-۶



شکل ۹-۱۰

۲- در این کادرمحاوره روی دکمه انتخاب Use Connection String کلیک کنید. در این بخش می‌توان اطلاعات لازم برای اتصال به پایگاه داده مانند نوع و نام پایگاه داده را انتخاب نمود. سپس روی دکمه فرمان ... Build کلیک کنید تا کادرمحاوره Data Link Properties مانند شکل ۱۰-۶ نمایش داده شود. در این کادرمحاوره و در زبانه Provider می‌توانید برنامه ارایه‌دهنده خدمات (Provider) برای اتصال به پایگاه داده خود را انتخاب کنید. برای اتصال به پایگاه‌های داده از نوع Access، می‌توانید مانند شکل ۱۰-۶ گزینه Microsoft Jet 4.0 OLE DB Provider را انتخاب و روی دکمه فرمان Next کلیک کنید.

۳- در این مرحله به زبانه Connection می‌روید (شکل ۱۱-۶). در این زبانه و در اولین کادر متن موجود در آن، مسیر و نام فایل پایگاه داده خود را تایپ کنید یا با کلیک روی دکمه  که روبه‌روی کادر متن قرار دارد کادرمحاوره Select Access Database را نمایش دهید و به وسیله آن پایگاه داده خود را انتخاب کنید.



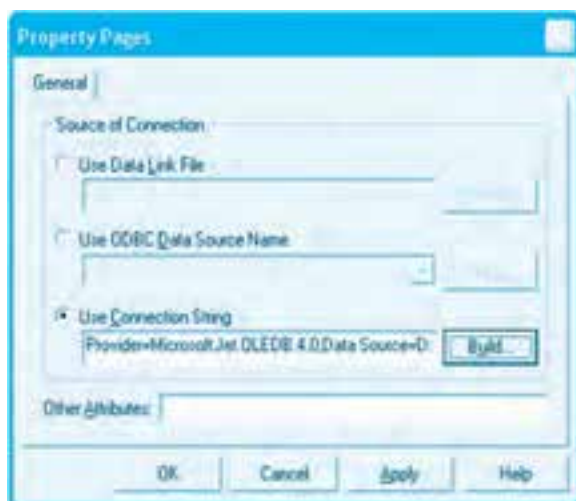
شکل ۱۱-۶

۴- در این مرحله می‌توانید با کلیک روی دکمه فرمان Test Connection اتصال با پایگاه داده را آزمایش کنید. اگر عملیات اتصال با موفقیت انجام شود کادرمحاوره‌ای مانند شکل ۱۲-۶ نمایش داده می‌شود.



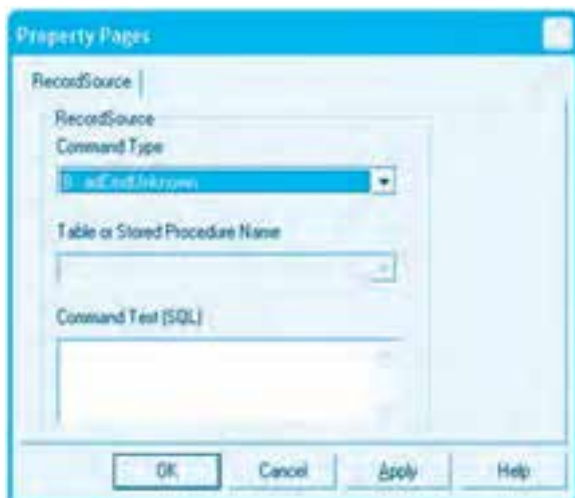
شکل ۱۲-۶

۵- در این مرحله روی دکمه فرمان OK کلیک کنید تا به کادرمحاوره اول یعنی Property Pages بازگردید. اکنون یک عبارت رشته‌ای به‌طور مناسب برای اتصال به پایگاه داده تعیین شده ایجاد شده است (شکل ۱۳-۶) و در کادر متن این کادرمحاوره نمایش داده می‌شود. در پایان روی دکمه فرمان OK کلیک کنید تا از این کادرمحاوره خارج شوید.



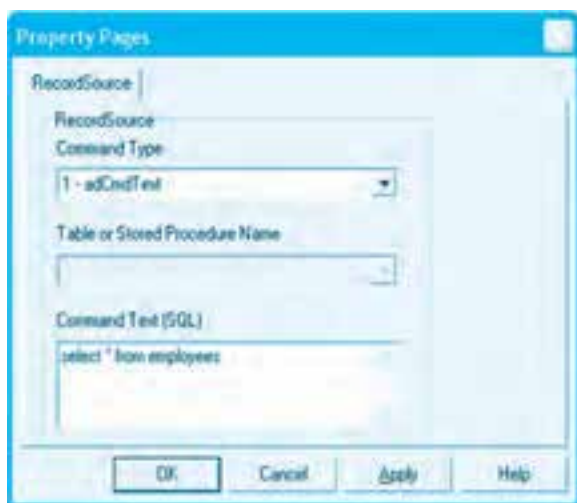
شکل ۱۳-۶

۶- اکنون لازم است تا خاصیت RecordSource را برای کنترل ADO انتخاب کنید تا کنترل بتواند رکوردهای موردنظر را از پایگاه داده بازیابی کند. پس در پنجره خواص روی دکمه ... روبه‌روی خاصیت RecordSource کلیک کنید تا کادرمحاوره Property Pages مانند شکل ۱۴-۶ نمایش داده شود.



شکل ۱۴-۶

در این کادرمحاوره می‌توانید با توجه به گزینه‌هایی که در کادر لیست ترکیبی باز شو Command Type وجود دارند یکی از آن‌ها را انتخاب کرده و با توجه به گزینه انتخاب شده کادر متن Command Text را نیز پر کنید. اگر در این کادرمحاوره و کادر لیست Command Type گزینه 1-adCmdText را انتخاب کنید، باید در کادر متن Command Text یک دستور Select بنویسید (شکل ۱۵-۶). در این حالت دستور Select رکوردهای موردنظر را بازیابی می‌کند.

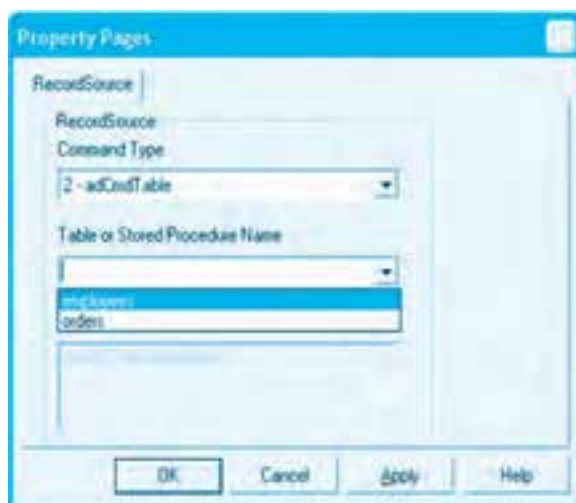


شکل ۱۵-۶

نکته: گزینه 8-adCmdUnknow در مواقعی انتخاب می‌شود که خاصیت CommandText مشخص نیست. این گزینه کاربرد زیادی ندارد.

نکته: گزینه 4-adCmdStoredProc در مواقعی انتخاب می‌شود که پایگاه داده از نوع SQL Server باشد در این حالت می‌توان نام رویه ذخیره شده را از کادر لیست دوم انتخاب نمود.

اگر گزینه 2-adCmdTable را انتخاب کنید، آنگاه باید در کادر لیست ترکیبی باز شو Table or Stored Procedure Name نام یک جدول از جدول‌های پایگاه داده را انتخاب کنید (شکل ۱۶-۶).



شکل ۱۶-۶

پس از انجام این عملیات کنترل ADO برای انجام عملیات بازیابی داده و اطلاعات آماده شده است و می‌توان با روش‌هایی که در کنترل Data توضیح داده شد با روش اتصال مستقیم (Bound) یا ارتباط غیرمستقیم (unBound) داده و اطلاعات را نمایش داد یا ویرایش نمود.


نکته: خاصیت‌های RecordSource و Command Text را می‌توانید مستقیماً از پنجره خواص نیز تنظیم نمایید.


مثال ۶: می‌خواهیم پروژه‌ای طراحی کنیم تا با استفاده از کنترل ADO داده‌ها و اطلاعات موجود در جدول employees و پایگاه داده company را بازیابی کنیم. برای این کار عملیات زیر را به ترتیب انجام دهید:

۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE مانند شکل ۱۷-۶ همراه با یک فرم و کنترل‌های آن ایجاد کنید.



شکل ۱۷-۶


- ۲- یک کنترل ADO روی فرم قرار دهید و نام آن را روی adoemployee و عنوان (Caption) آن را روی Employees تنظیم کنید.
- ۳- در این مرحله خاصیتConnectionString کنترل ADO را روی گزینه Microsoft Jet 4.0 OLE DB Provider و پایگاه داده Company تنظیم کنید.
- ۴- در پنجره خواص ابتدا روی خاصیت RecordSource و بعد روی دکمه  رویه‌روی آن کلیک کنید و در کادر لیست Command Type گزینه 1-adCmdText را انتخاب نمایید و بعد در کادر متن Command Text یک دستور Select مانند شکل ۱۵-۶ تایپ کنید و روی دکمه فرمان OK کلیک کنید.
- ۵- اکنون لازم است تا کنترل‌های کادر متن را به کنترل ADO متصل کنیم، بنابراین روی کادر متن txtid کلیک کرده و خاصیت DataSource آن را روی نام کنترل ADO یعنی adoemployee تنظیم کنید، سپس خاصیت DataField همین کادر متن را روی فیلد employeeid تنظیم کنید.
- ۶- سایر کنترل‌ها را نیز به همین روش به کنترل ADO متصل کنید.

- ۷- پروژه و فرم را با نام `adocompany` و `employees` ذخیره کنید.
- ۸- پروژه را اجرا کرده و با کلیک روی دکمه‌های کنترل ADO روی رکوردها حرکت کنید.
- ۹- به اجرای پروژه خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.
-  **تمرین:** پروژه‌ای طراحی کنید که با استفاده از کنترل ADO داده‌ها و اطلاعات جدول سفارشات (Orders) را بازیابی کند و نمایش دهد.
- در کنترل ADO نیز مانند کنترل Data می‌توان تنظیم خاصیت‌های کنترل‌ها را در زمان اجرا انجام داد. به علاوه می‌توان کنترل‌ها را در زمان اجرا نیز به کنترل ADO متصل نمود؛ هم‌چنین کنترل ADO دارای یک خاصیت از نوع Recordset است که مانند کنترل ADO می‌توان از خاصیت‌ها و متدهای آن استفاده کرد. به عنوان مثال این دستورات نحوه تنظیم خاصیت‌های کنترل ADO و متصل کردن کنترل کادر متن را به کنترل ADO نشان می‌دهد.

```
adoemployee.ConnectionString =  
    "Provider=Microsoft.Jet.OLEDB.4.0;Data Source" + _  
    "=C:\company.mdb;Persist Security Info=False"  
adoemployee.RecordSource = "select * from employees where salary>5000000"  
adoemployee.CommandType = adCmdText  
Set txtid.DataSource = adoemployee  
txtid.DataField = "employeeid"
```

در این دستورات ابتدا خاصیت `ConnectionString` تنظیم می‌شود. این خاصیت از نوع رشته‌ای است و اطلاعات لازم برای اتصال به پایگاه داده را دربرمی‌گیرد. این رشته را می‌توان در زمان تنظیم این خاصیت در پنجره خواص به‌دست آورد. سپس خاصیت `RecordSource` تنظیم می‌شود که این خاصیت نیز از نوع رشته‌ای است و در این جا روی رشته‌ای که شامل دستور `Select` است، تنظیم می‌شود. در مرحله بعد خاصیت `CommandType` با توجه به مقدار خاصیت `RecordSource` روی مقدار `adCmdText` تنظیم می‌شود.

برای اتصال کنترل `txtid` به ستون `employeeid` ابتدا باید خاصیت `DataSource` با استفاده از دستور `Set` روی کنترل ADO یعنی `adoemployee` تنظیم شود. سپس خاصیت `DataField` تنظیم می‌شود؛ این خاصیت از نوع رشته‌ای است و نام ستونی که کنترل به آن متصل می‌شود در این خاصیت قرار می‌گیرد.

 **تمرین:** پروژه‌ای طراحی کنید که به وسیله کنترل ADO و تنظیم خاصیت‌های آن در زمان اجرا بتوان محتویات جدول سفارشات (Orders) را مشاهده نمود و با استفاده از متدهای `AddNew`، `UpdateRecord`، `Delete` شیء `Recordset`، بتوان عملیات درج، به‌روز رسانی و حذف را انجام داد. از این متدها می‌توانید مانند معادل‌های آن‌ها در کنترل `Data` استفاده کنید.

خاصیت CursorType

شیء Recordset از این خاصیت استفاده می‌کند تا نحوه حرکت روی رکوردها و انجام عملیات به‌روز رسانی را بداند. این خاصیت می‌تواند روی یکی از این مقادیر تنظیم شود:

الف – 1-adOpenKeyset

در این حالت، برنامه نمی‌تواند رکوردهای درج شده به‌وسیله سایر برنامه‌ها یا کاربران را روی همان منبع داده‌ای که کنترل ADO به آن متصل است، مشاهده کند و از دسترسی به رکوردهایی که به وسیله سایر کاربران یا برنامه‌ها حذف می‌شود نیز جلوگیری به عمل می‌آید. به علاوه رکوردهایی که به وسیله سایر کاربران یا برنامه‌ها ویرایش می‌شوند به وسیله برنامه قابل دسترسی هستند.

ب – 2-adOpenDynamic

در این حالت برنامه می‌تواند درج، به‌روز رسانی و حذف رکوردها توسط سایر کاربران یا برنامه‌ها روی همان منبع داده‌ای را که کنترل ADO به آن متصل شده است، مشاهده کند.

ج – 3-adOpenStatic

در این حالت یک کپی مشابه از رکوردها در اختیار کنترل ADO قرار می‌گیرد و امکان هر نوع حرکتی روی رکوردها وجود دارد اما رکوردهایی که به وسیله سایر کاربران یا برنامه‌ها درج، ویرایش یا حذف می‌شوند به وسیله کنترل ADO قابل دسترسی نخواهد بود.

خاصیت LockType

از این خاصیت زمانی استفاده می‌شود که چندین کاربر به‌طور همزمان روی یک منبع داده مشترک عملیات درج، حذف و ویرایش را انجام می‌دهند. این خاصیت می‌تواند روی یکی از این مقادیر تنظیم شود، در نتیجه ممکن است:

الف – 1-adLockUnSpecified

در این حالت هیچ قفلی مورد استفاده قرار نمی‌گیرد.

ب – 1-adLockReadOnly

در این حالت کنترل ADO امکان تغییر در داده‌ها را از دست می‌دهد.

ج – 2-adLockPessimistic

در این حالت رکورد یا رکوردهایی که در حال به‌روز رسانی هستند به گونه‌ای در منبع داده قفل می‌شوند تا عملیات به‌روز رسانی آن‌ها حتماً با موفقیت انجام شود. به عبارت دیگر با شروع ویرایش رکورد یا رکوردهای مربوطه قفل می‌شوند.

د – 3-adLockOptimistic

در این حالت رکورد یا رکوردهایی که به‌روز رسانی می‌شوند وقتی قفل می‌شوند که کاربر عملیات و به‌روز رسانی را تأیید کند.

ه- 4-adLockBatchOptimistic

در این حالت ویرایش گروهی از رکوردها، قبل از آن که عملیات به‌روزرسانی آن‌ها روی پایگاه داده انجام شود، وجود نخواهد داشت.

خاصیت CursorLocation

این خاصیت می‌تواند روی یکی از مقادیر 2-adUseServer یا 3-adUseClient تنظیم شود. وقتی شیء Recordset و پایگاه داده در همان موقعیتی که برنامه اجرا می‌شود، قرار دارند باید روی مقدار adUseClient تنظیم شود.

خاصیت Mode

این خاصیت می‌تواند وضعیت اتصال را برای کنترل ADO مشخص کند. وقتی پایگاه داده در همان موقعیتی است که برنامه اجرا می‌شود، مقدار این خاصیت باید روی مقدار 0-adModeUnknown تنظیم شود. البته این خاصیت می‌تواند روی مقادیر مختلفی تنظیم شود که امکان بحث و بررسی روی این مقادیر از حوصله این کتاب خارج است.

مثال ۷: می‌خواهیم پروژه‌ای طراحی کنیم که به وسیله آن بتوان اطلاعات کارمندان را با توجه به نام خانوادگی آن‌ها جستجو و مشاهده نمود. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE همراه با یک فرم با نام frmemployee و عنوان Employee مانند شکل ۱۸-۶ ایجاد کنید.



شکل ۱۸-۶

۲- یک کنترل ADO روی فرم قرار دهید و نام آن را روی adoemployee و عنوان آن را روی عبارت Search Employees تنظیم کنید، سپس مانند شکل ۱۷-۶ کنترل‌های مورد نیاز را روی فرم قرار دهید و خاصیت‌های آن‌ها را تنظیم کنید. کنترل‌هایی که در کنترل قاب (Frame) قرار گرفته‌اند برای انجام عملیات جستجو و کنترل‌هایی که در زیر کنترل قاب قرار دارند، برای نمایش نتایج حاصل از جستجو استفاده می‌شوند.

۳- اکنون رویداد Load فرم را به این صورت تنظیم کنید تا کنترل ADO آماده اتصال و بازیابی شده و رکوردها از جدول employees در پایگاه داده company شود و کلیه رکوردها را بازیابی کند.

Private Sub Form_Load()

```
adoemployee.ConnectionString = _
    "Provider=Microsoft.Jet.OLEDB.4.0;Data Source" + _
    "=C:\company.mdb;Persist Security Info=False"
adoemployee.RecordSource = "select * from employees"
adoemployee.CommandType = adCmdText
Set txtid.DataSource = adoemployee
txtid DataField = "employeeid"
Set txtfirstname.DataSource = adoemployee
txtfirstname DataField = "firstname"
Set txtlastname DataSource = adoemployee
txtlastname DataField = "lastname"
Set txtcity.DataSource = adoemployee
txtcity.DataField = "city"
Set txtsalary.DataSource = adoemployee
txtsalary.DataField = "salary"
```

End Sub

به‌علاوه تمام کنترل‌های کادر متنی که خارج از کنترل قاب قرار دارند نیز به کنترل ADO متصل می‌شوند. با این کار در هنگام نمایش فرم می‌توان مشخصات کلیه کارمندان را مشاهده کرد. در ضمن در این رویداد خاصیت CursorType کنترل ADO نیز روی مقدار adOpenStatic تنظیم می‌شود تا فقط رکوردهایی که

جستجو می‌شوند قابل مشاهده باشند و از دسترسی به سطرهایی که به وسیله سایر برنامه‌ها یا کاربران درج، ویرایش یا حذف می‌شوند جلوگیری به عمل آید.

۴- در این مرحله رویداد Click دکمه فرمان Search را به این صورت تنظیم کنید:

```
Private Sub cmdsearch_Click()
```

```
    adoemployee.RecordSource = _
```

```
        "select * from employees where lastname=' ' + txtsearch.Text + ' ' "
```

```
    adoemployee.Refresh
```

```
    If adoemployee.Recordset.RecordCount = 0 Then MsgBox "The Employee Doesn't Exist!"
```

```
End Sub
```

در این رویداد با سفارشی کردن دستور Select خاصیت RecordSource کنترل ADO دوباره تنظیم می‌شود. برای این کار عبارتی که در کنترل کادر متن txtSearch به عنوان نام خانوادگی تایپ می‌شود به صورت یک مقدار رشته‌ای با استفاده از عملگر جمع به بقیه دستور Select که ثابت بوده و تغییر نمی‌کند، اضافه می‌شود در نتیجه با تغییر محتویات کنترل txtsearch عبارت مقایسه‌ای بخش Where در دستور Select نیز تغییر می‌کند. بنابراین وقتی این رویداد اجرا می‌شود عملیات بازبازی رکوردها با شرایط جدید انجام خواهد شد. پس از تنظیم دستور Select و قرار دادن آن در خاصیت RecordSource کنترل ADO، متد Refresh این کنترل اجرا می‌شود تا شیء Recordset کنترل ADO با داده‌های جدید پر شود.

آخرین دستور در این رویداد یک دستور If می‌باشد. این دستور با استفاده از خاصیت RecordCount شیء Recordset کنترل ADO را بررسی می‌کند. این خاصیت تعداد رکوردهای موجود در شیء Recordset را مشخص می‌کند، بنابراین اگر مقدار این خاصیت صفر باشد نشان‌دهنده آن است که کارمندی با نام خانوادگی تعیین شده وجود ندارد و در نتیجه یک کادر پیغام این مطلب را با نمایش یک پیام مناسب به کاربر اطلاع می‌دهد.

نکته: در هنگام سفارشی کردن دستور Select لازم است داده‌های از نوع رشته‌ای داخل دو کاراکتر ' و داده‌های از نوع تاریخ داخل دو کاراکتر # و داده‌های عدد به صورت عادی در دستور Select قرار داده شوند.

۵- پروژه و فرم را با نام search employee ذخیره کرده و سپس پروژه را اجرا کنید و رکوردهای موجود را با استفاده از دکمه‌های کنترل ADO مشاهده نمایید.

۶- نام خانوادگی را که در جدول employee وجود دارد، در کادر متنی که در کنترل قاب Search قرار دارد تایپ کرده و بعد روی دکمه فرمان Search کلیک کنید (شکل ۱۹-۶).



شکل ۱۹-۶

در این لحظه اطلاعات کارمندانی که نام خانوادگی آن‌ها با نام خانوادگی موردنظر یکسان باشند پیدا شده و اولین رکورد با این نام خانوادگی در کنترل‌های کادر متن نمایش داده می‌شوند. اگر چند کارمند با این نام خانوادگی در جدول employees وجود داشته باشند، می‌توانید با استفاده از دکمه‌های کنترل ADO روی رکوردها حرکت کرده و مشخصات آن‌ها را نیز مشاهده کنید.


۷- یک نام و نام خانوادگی را که در جدول employees وجود ندارد، در کادر متن مربوطه تایپ کرده و بعد روی دکمه فرمان Search کلیک کنید. در این صورت کادر پیغامی مانند شکل ۲۰-۶ نمایش داده می‌شود و سایر کنترل‌های کادر متن با مقدار خالی مشاهده می‌شوند (شکل ۲۱-۶).



شکل ۲۰-۶



شکل ۲۱-۶

۸- به اجرای پروژه خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.  **تمرین:** پروژه‌ای طراحی کنید که به وسیله کنترل ADO و تنظیم خاصیت‌های آن در زمان اجرا بتوان مشخصات کارمندانی را که حقوق آن‌ها از مقدار دلخواهی کمتر است، مشاهده نمود و سپس با استفاده از متدهای شیء Recordset روی رکوردها حرکت کرد.

متد Find

شیء Recordset در کنترل ADO دارای یک متد با نام Find برای جستجوی رکوردها با شرایط موردنظر می‌باشد. این متد می‌تواند عملیات جستجو را روی مجموعه رکوردهایی که در Recordset قرار دارند انجام دهد بدون آن که نیازی به تغییر خاصیت RecordSource کنترل ADO و اجرای متد Refresh در این کنترل باشد. نحوه استفاده از این متد به این صورت است:

"عبارت مقایسه‌ای" Recordset.Find نام کنترل ADO

به عنوان مثال اگر کلیه سطرهای جدول employees به وسیله کنترل adoemployee بازیابی شده باشند، این دستور رکورد جاری را در کنترل adoemployee روی اولین رکوردی که نام خانوادگی وی 'minae' باشد، تنظیم می‌کند.

```
adoemployee.Recordset Find "lastname='minae'"
```

پس از اجرای این دستور برای آن که بتوان سایر رکوردهای دارای این شرط را جستجو نمود باید متد Find

به این صورت مورد استفاده قرار گیرد. به عنوان مثال این دستور رکورد بعدی را پس از اجرای دستور قبل و پیدا کردن اولین رکورد، جستجو می‌نماید.

```
adoemployee.Recordset Find "lastname='minae'", 1
```

نکته: در صورتی که متد Find رکوردی را مطابق شرایط تعیین شده پیدا نکند خاصیت EOF شیء Recordset روی مقدار True تنظیم می‌شود.

تمرین: پروژه Searchemployee را به شکلی تنظیم کنید تا عملیات جستجو را بتوان با متد Find انجام داد.

خاصیت Filter

شیء Recordset در کنترل ADO دارای یک خاصیت با نام Filter می‌باشد، به وسیله این خاصیت می‌توان یک یا مجموعه‌ای از رکوردها را با شرایط موردنظر فیلتر کرد. نحوه استفاده از این خاصیت به این صورت است:

"عبارت مقایسه‌ای" = Recordset.Filter نام کنترل ADO

به عنوان مثال اگر کلیه سطرهای جدول employees به وسیله کنترل adoemployee بازبایی شده باشند این دستور رکوردهایی را که نام خانوادگی آن‌ها 'minae' باشد، فیلتر می‌کند؛ به این معنی که پس از اجرای این دستور فقط رکوردهایی را می‌توان مورد دسترسی قرار داد که نام خانوادگی آن‌ها 'minae' باشد.

```
adoemployee.Recordset Filter = "lastname='minae' "
```

به عنوان مثالی دیگر این دستور رکوردهایی را که نام آن‌ها 'Arash' و نام خانوادگی آن‌ها 'Sohrabi' باشد، فیلتر می‌کند.

```
adoemployee.Recordset Filter = "lastname='sohrabi' and firstname='arash' "
```

تمرین: پروژه searchemployee را به شکلی تنظیم کنید تا با استفاده از خاصیت Filter بتوان رکوردها را براساس نام، نام خانوادگی و نام شهر محل تولد پیدا کرد.

۳-۲-۶ کنترل DataList و DataCombo

با استفاده از این کنترل می‌توان محتویات یک ستون تمام سطرها را به‌طور همزمان مورد دسترسی قرار داد. کنترل‌هایی مانند کادر متن می‌توانند در هر لحظه مقدار یک ستون را در سطر جاری نمایش دهند. برای اضافه کردن این کنترل به جعبه ابزار ویژوال بیسیک در کادرمحاوره Components، گزینه Microsoft DataList Controls 6.0 (OLEDB) را انتخاب کنید (شکل ۲۲-۶)، سپس روی دکمه فرمان OK

کلیک کنید تا این کنترل‌ها به جعبه ابزار ویژوال بیسیک اضافه شوند (شکل ۶-۲۳).



شکل ۶-۲۲



شکل ۶-۲۳

کنترل DataList شبیه به کنترل کادر لیست (ListBox) و کنترل DataCombo شبیه به کنترل کادر لیست ترکیبی بازشو (ComboBox) می‌باشند. برای اتصال این کنترل‌ها به کنترل ADO ابتدا خاصیت‌های DataSource و RowSource آن‌ها را روی کنترل ADO تنظیم کنید، سپس خاصیت ListField آن‌ها را روی نام ستونی که می‌خواهید کنترل، محتویات آن را نمایش دهد تنظیم نمایید.

نکته: در این کنترل‌ها بعضی از خاصیت‌ها و رویدادها مانند کنترل‌های ListBox و ComboBox قابل دستیابی هستند.

مثال ۸: می‌خواهیم پروژه‌های طراحی کنیم که به وسیله کنترل DataCombo بتوان رکوردهای جدول employees را براساس نام شهر محل تولد جستجو کرد. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

- ۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE همراه با یک فرم ایجاد کنید.
- ۲- در این مرحله کنترل‌های مورد نیاز را همراه با یک کنترل ADO با نام adoemployee و خاصیت عنوان Employee روی فرم قرار دهید (شکل ۶-۲۴).



شکل ۶-۲۴

- ۳- کنترل adoemployee را روی پایگاه داده company و جدول employees به شکلی تنظیم کنید که کلیه رکوردها بازیابی شوند. سپس کنترل‌های کادر متن را به این کنترل و ستون‌های مربوط به هر یک متصل کنید.
- ۴- در این مرحله یک کنترل ADO دیگر روی فرم قرار داده و نام آن را روی adocity، خاصیت عنوان آن را روی City و خاصیت Visible آن را روی مقدار False تنظیم کنید.
- ۵- کنترل adocity را روی پایگاه داده company و جدول employees به شکلی تنظیم کنید که اسامی کلیه شهرها بازیابی شوند برای این کار می‌توانید دستور Select را به این صورت تنظیم کنید:
 Select distinct city From employees
 در این دستور از کلمه کلیدی distinct استفاده شده است تا رکوردهایی که نام شهر آن‌ها تکراری باشد بازیابی نشوند.
- ۶- کنترل DataCombo را به جعبه ابزار ویژوال بیسیک اضافه کنید و بعد مانند شکل ۶-۲۴ یک کنترل DataCombo با نام dbccity روی فرم قرار دهید.
- ۷- اکنون خاصیت‌های DataSource و RowSource کنترل dbccity را روی نام کنترل ADO یعنی adocity

تنظیم کنید. سپس خاصیت ListField آن را روی فیلد City تنظیم کنید.

۸- رویداد Load فرم را به این صورت تنظیم کنید تا در زمان اجرای برنامه نام اولین شهر به‌طور پیش‌فرض در کنترل کادر لیست ترکیبی بازشو نمایش داده شود.

```
Private Sub Form_Load()
```

```
dbccity.Text = adocity Recordset!city
```

```
End Sub
```

۹- در این مرحله رویداد Change کنترل کادر لیست ترکیبی بازشو dbccity را به این صورت تنظیم کنید:

```
Private Sub dbccity_Change()
```

```
adoemployee.Recordset Filter = "City=" + dbccity.Text + ""
```

```
End Sub
```

این رویداد زمانی اجرا می‌شود که یک گزینه در کادر لیست انتخاب شود یا به عبارت دیگر خاصیت Text کنترل تغییر کند.

در این رویداد با استفاده از خاصیت Filter شیء Recordset کنترل adoemployee رکوردهای این کنترل براساس نام شهری که در کنترل کادر لیست ترکیبی بازشو انتخاب شده است و به وسیله خاصیت Text به‌دست می‌آید فیلتر می‌شوند.

۱۰- پروژه و فرم را با نام CityFilter ذخیره کرده و پروژه را اجرا کنید.

۱۱- در کادر لیست ترکیبی بازشو نام یک شهر را انتخاب کنید؛ در این صورت مشخصات اولین کارمندی که متولد شهر انتخاب شده باشد در کنترل‌های کادر متن نمایش داده می‌شود. اگر کارمند دیگری در این شهر متولد شده باشد، می‌توانید با دکمه‌های کنترل adoemployee مشخصات آن‌ها را نیز مشاهده کنید.

۱۲- عمل جستجو را با تایپ نام یک شهر در کنترل کادر لیست ترکیبی بازشو نیز انجام داده و نتیجه را با حالت قبل مقایسه و بررسی کنید.

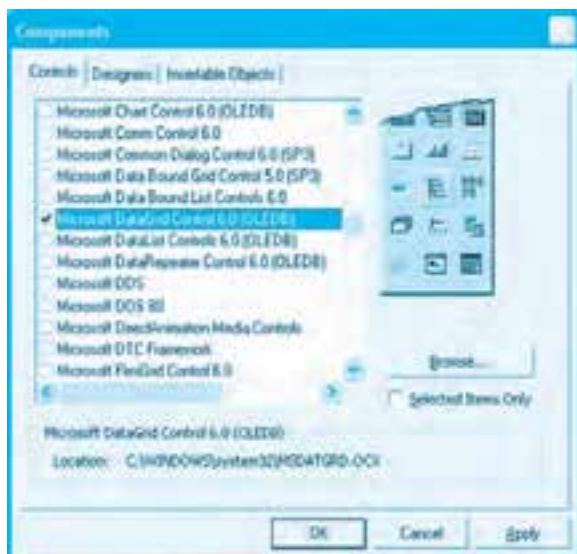
۱۳- به اجرای پروژه خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.

تمرین: پروژه cityfilter را با یک کنترل کادر لیست داده (DataList) انجام دهید.



۴-۲-۶ کنترل DataGrid

با استفاده از این کنترل می‌توان کلیه سطرها و ستون‌های موجود در شیء Recordset را به‌طور همزمان نمایش و مورد دسترسی قرار داد. به علاوه با استفاده از ماوس یا کلیدهای حرکت مکان‌نما روی رکوردها به جلو و عقب حرکت کرد. برای اضافه کردن این کنترل به جعبه ابزار ویژوال بیسیک در کادرمحاوره Components گزینه Microsoft DataGrid Control 6.0 (OLEDB) را انتخاب کنید (شکل ۲۵-۶). سپس روی دکمه فرمان OK کلیک کنید تا این کنترل به جعبه ابزار ویژوال بیسیک اضافه شود (شکل ۲۶-۶).



شکل ۲۵-۶



شکل ۲۶-۶

برای اتصال این کنترل به کنترل ADO کافی است خاصیت DataSource آن را روی یک کنترل ADO تنظیم کنید. شکل ظاهری این کنترل شبیه به کنترل FlexGrid می‌باشد و سطرها و ستون‌ها را با استفاده از خطوط به صورت مستقل از یکدیگر در سطرها و ستون‌های مجزا نمایش می‌دهد. این کنترل علاوه بر خاصیت‌های مشترکی که با سایر کنترل‌ها دارد دارای خاصیت‌ها، متدها و رویدادهای ویژه‌ای نیز هست که در این جا بررسی می‌شوند.

خاصیت AllowAddNew

این خاصیت از نوع منطقی است؛ اگر این خاصیت روی مقدار True تنظیم شود، به وسیله کنترل DataGrid می‌توان سطرها را در جدول درج کرد.

خاصیت AllowArrows

این خاصیت از نوع منطقی است؛ اگر این خاصیت روی مقدار True تنظیم شود در کنترل می‌توان با استفاده از کلیدهای حرکت مکان‌نما روی رکوردها به جلو و عقب حرکت کرد اما اگر این خاصیت روی مقدار False

تنظیم شود فقط با ماوس امکان تغییر رکورد جاری وجود دارد.

خاصیت AllowDelete

این خاصیت از نوع منطقی است؛ اگر این خاصیت روی مقدار True تنظیم شود امکان حذف رکوردها به وجود می‌آید.

خاصیت AllowUpdate

این خاصیت نیز از نوع منطقی است؛ اگر این خاصیت روی مقدار True تنظیم شود امکان ویرایش رکوردها به وجود می‌آید.

خاصیت AllowRowSizing

این خاصیت از نوع منطقی است؛ اگر این خاصیت روی مقدار True تنظیم شود امکان تغییر ارتفاع سطرها در کنترل به وجود می‌آید. این خاصیت در پنجره خواص قابل تنظیم نیست.

خاصیت ColumnHeaders

این خاصیت از نوع منطقی است؛ اگر این خاصیت روی مقدار True تنظیم شود عناوین ستون‌ها نمایش داده شده و در غیر این صورت ستون‌ها بدون عنوان نمایش داده می‌شوند.

خاصیت HeadFont

به وسیله این خاصیت می‌توان ویژگی‌های قلم را برای عنوان ستون‌ها تنظیم نمود.

خاصیت HeadLines

به وسیله این خاصیت می‌توان تعداد خطوطی را که برای نمایش متن در عنوان ستون‌ها استفاده می‌شود، تعیین کرد.

خاصیت RowDivideStyle

به وسیله این خاصیت می‌توان نوع خطوطی که رکوردها را از هم جدا می‌کند، تعیین نمود.

خاصیت RowHeight

به وسیله این خاصیت می‌توان ارتفاع سطرها را در کنترل تنظیم نمود.

خاصیت TabAction

این خاصیت عملکرد کلید Tab را وقتی مشخص می‌کند که کنترل DataGrid فوکوس را در اختیار می‌گیرد؛ اگر این خاصیت روی مقدار 0-dbgControlNavigation تنظیم شود، فشردن کلید Tab فوکوس بر اساس مقدار خاصیت TabIndex کنترل‌ها به کنترل بعدی منتقل می‌شود. اگر روی مقدار 1-dbgColumnNavigation تنظیم شود، با استفاده از کلید Tab می‌توان بین ستون‌ها حرکت کرد و با عبور از آخرین ستون، فوکوس به

کنترل بعدی می‌رسد. اگر روی مقدار 2-dbgGridNavigation تنظیم شود با استفاده از کلید Tab می‌توان فقط بین ستون‌ها حرکت کرد.

متد ReBind

این متد مجدداً کنترل را به منبع داده متصل می‌کند. به عبارت دیگر محتویات کنترل را از روی منبع داده به روز می‌نماید.

رویداد AfterDelete

این رویداد پس از حذف یک رکورد در کنترل اجرا می‌شود.

رویداد AfterInsert

این رویداد پس از درج یک رکورد در کنترل اجرا می‌شود.

رویداد AfterUpdate

این رویداد پس از ویرایش یک رکورد در کنترل اجرا می‌شود.

رویداد RowColChange

این رویداد زمانی اجرا می‌شود که ستون یا سطر جاری در کنترل تغییر کند. این رویداد دو آرگومان دارد؛ آرگومان LastRow که شماره سطر قبلی و آرگومان LastCol که شماره ستون قبلی را مشخص می‌کنند. شماره سطرها از یک و شماره ستون‌ها از صفر شروع می‌شوند.



شکل ۲۷-۶

مثال ۹: می‌خواهیم پروژه‌ای طراحی کنیم که مانند شکل ۲۷-۶ بتوان با انتخاب یک کارمند از لیست کارمندان مشخصات کلیه سفارش‌هایی را که تاکنون انجام داده است، مشاهده کرد.

برای انجام این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE ایجاد کنید.

۲- کنترل‌های ADO و DataGrid را به جعبه ابزار و ویژوال بیسیک اضافه کنید.

۳- دو کنترل ADO با نام‌های adoemployee و adoorder روی فرم قرار داده و کنترل adoemployee را به پایگاه داده company.mdb و جدول employees و کنترل adoorder را به همین پایگاه داده و جدول orders متصل کنید.

۴- خاصیت RecordSource کنترل adoemployee را به این صورت تنظیم کنید تا تمام رکوردهای جدول employees بازیابی شوند.

```
Select * From Employees
```

۵- اکنون خاصیت RecordSource کنترل adoorder را به این صورت تنظیم کنید تا تمام رکوردهای جدول orders بازیابی شوند.

```
Select * From orders
```

۶- در این مرحله دو کنترل DataGrid مانند شکل ۲۷-۶ روی فرم قرار دهید و خاصیت نام آن‌ها را روی dgdememployees و dgorders تنظیم کنید، سپس خاصیت DataSource کنترل dgdememployees را روی کنترل adoemployee و خاصیت DataSource کنترل dgorders را نیز روی کنترل adoorders تنظیم کنید.

۷- رویداد Load فرم را به این صورت تنظیم کنید:

```
Private Sub Form_Load()
```

```
adoorders.Recordset.Filter = "employeeid=" + _
```

```
Str(adoemployee.Recordset.employeeid)
```

```
End Sub
```

۸- در این رویداد با استفاده از خاصیت Filter رکوردها در کنترل adoorders بر اساس شماره کارمندی فیلتر می‌شوند. وقتی برنامه اجرا شود این رویداد نیز اجرا شده و شماره کارمند اولین رکورد در کنترل adoemployee در اختیار خاصیت Filter کنترل adoorder قرار می‌گیرد، در نتیجه وقتی فرم نمایش داده می‌شود مشخصات سفارش‌هایی که مربوط به اولین کارمند می‌باشد در کنترل DataGrid، مربوط به سفارشات نمایش داده خواهد شد. در این مرحله رویداد RowColChange کنترل dgorders را به این

صورت مانند رویداد Load تنظیم کنید. در این رویداد نیز با توجه به شماره کارمند رکورد جاری، رکوردهای کنترل adoorder فیلتر می‌شوند.

Private Sub dgdemployees_RowColChange(LastRow As Variant, ByVal LastCol As Integer)

adoorders.Recordset.Filter = "employeeid=" + _

Str(adoemployee.Recordset.employeeid)

End Sub

۹- پروژه و فرم را با نام searchempid ذخیره کرده سپس آن را اجرا کنید؛ فرم نمایش داده شده و مشخصات کارمندان در اولین کنترل DataGrid و مشخصات سفارش‌های اولین کارمند در کنترل DataGrid دوم نمایش داده می‌شوند.

۱۰- اکنون از کنترل DataGrid مربوط به کارمندان، کارمند دیگری را انتخاب کنید تا مشخصات سفارشات را که وی انجام داده است، در کنترل DataGrid مربوط به سفارشات نمایش داده شود.

۱۱- به اجرای پروژه خاتمه داده و به پنجره ویژوال بیسیک باز گردید.

۵-۲-۶ استفاده از کتابخانه شیء ADODB

کنترل ADO این امکان را فراهم می‌کند تا به‌طور غیرمستقیم و با استفاده از یک رابط گرافیکی که به صورت یک کنترل ارائه می‌شود از امکاناتی که در تکنولوژی OLEDB وجود دارد، استفاده کرد. علاوه بر این می‌توانید مستقیماً با استفاده از کتابخانه شیء ADODB با پایگاه‌های داده ارتباط برقرار کرده و داده‌ها را بازبایی، درج، ویرایش یا حذف کنید. کتابخانه شیء ADODB کلاس‌های مختلفی را برای انجام عملیات گوناگون ارائه می‌کنند. سه کلاس اصلی در این کتابخانه عبارتند از:

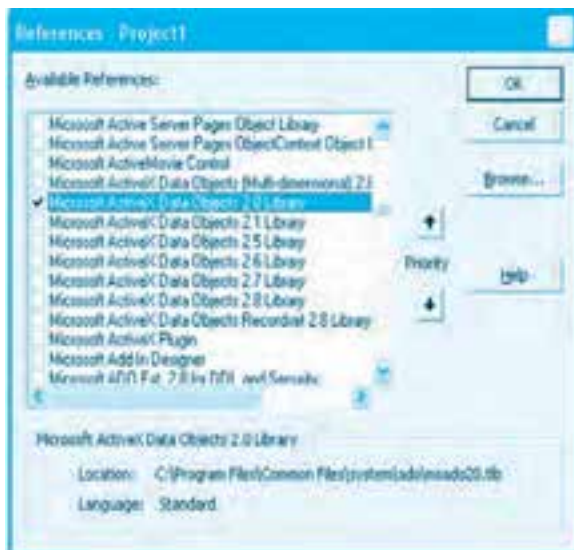
الف- **کلاس Connection**: این کلاس مجموعه‌ای از خاصیت‌ها، متدها و رویدادها را برای اتصال به انواع و مدل‌های مختلفی از پایگاه‌های داده مانند Access، SQL Server یا Oracle ارائه می‌کند.

ب- **کلاس Command**: این کلاس مجموعه‌ای از خاصیت‌ها، متدها و رویدادها را برای اجرای دستورات SQL روی پایگاه‌های داده و جدول‌های موجود در آن‌ها ارائه می‌کند.

ج- **کلاس Recordset**: این کلاس نیز مجموعه‌ای از خاصیت‌ها، متدها و رویدادها را برای دسترسی و مدیریت رکوردهایی که از پایگاه‌های داده بازبایی می‌شوند، ارائه می‌نماید

قبل از استفاده از کتابخانه شیء ADODB لازم است آن را به پروژه موردنظر اضافه کنید؛ برای این کار در نوار منوی برنامه ویژوال بیسیک ابتدا روی منوی Project و سپس گزینه References... کلیک کنید. در کادرمحاوره References گزینه Microsoft ActiveX Data Objects 2.0 Library یا نگارش‌های بالاتر آن را انتخاب کرده و بعد روی دکمه فرمان OK

کلیک کنید (شکل ۲۸-۶).



شکل ۲۸-۶

پس از اضافه کردن کتابخانه ADODB به پروژه، می‌توان از کلاس‌های موجود در این کتابخانه اشیای موردنظر را ساخت و عملیات لازم را روی پایگاه داده اجرا نمود. برای این کار می‌توانید به این ترتیب عمل کنید:

۱- برای انجام هرگونه عملیات روی پایگاه داده ابتدا لازم است به آن متصل شد، بنابراین باید یک شیء از کلاس Connection ایجاد نمود و با استفاده از خاصیتConnectionString پارامترهای لازم برای اتصال به پایگاه داده در آن مشخص شود و سپس با اجرای متد Open در این شیء اتصال برقرار شود.

به عنوان مثال اگر بخواهیم به پایگاه داده Company که در ریشه درایو C: قرار دارد و از نوع Access است، متصل شد از این دستورات استفاده می‌شود:

```
Dim cncompany As Connection
Set cncompany = New Connection
cncompany.ConnectionString = _
    "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\company.mdb"
cncompany.Open
```

۲- اکنون باید از کلاس Command یک شیء ایجاد و دستور SQL را برای اجرا روی پایگاه داده تعیین کرد. به عنوان مثال برای بازیابی رکوردهای جدول employees که در پایگاه داده Company قرار دارد می‌توان به این صورت عمل نمود:

```
Dim cmemployees As Command
Set cmemployees = New Command
cmemployees.ActiveConnection = cncompany
cmemployees.CommandText = "select * from employees"
cmemployees.CommandType = adCmdText
```

در این دستورات ابتدا شیء cncompany در خاصیت ActiveConnection شیء cmemployees ذخیره می‌شود تا این شیء بداند دستور Select را روی کدام پایگاه داده اجرا کند، سپس دستور Select در خاصیت CommandText قرار می‌گیرد. این خاصیت دستور موردنظر را برای اجرا روی پایگاه داده تعیین می‌کند و در مرحله بعد چون از یک دستور SQL استفاده شده است خاصیت CommandType روی مقدار adCmdText تنظیم می‌شود.

۳- در مرحله سوم از کلاس Recordset شیء ایجاد کنید تا هنگام اجرای شیء Command امکان ذخیره‌سازی، نگهداری رکوردها و دستیابی به آنها وجود داشته باشد به عنوان مثال با توجه به مرحله‌های قبل می‌توان این دستورات را اجرا نمود:

```
Dim rsemployees As Recordset
Set rsemployees = New Recordset
Set rsemployees = cmemployees.Execute
```

در این دستورات پس از ایجاد شیء rsemployees از کلاس Recordset، متد Execute دستور موجود در شیء cmemployees را روی پایگاه داده company اجرا کرده و رکوردهای بازیابی شده را در شیء rsemployees قرار می‌دهد. اکنون می‌توان پس از انجام عملیات، محتویات ستون‌های رکورد جاری در شیء rsemployees را مورد دستیابی قرار داد.

مثال ۱۰: می‌خواهیم پروژه‌ای طراحی کنیم که به وسیله کلاس‌های موجود در کتابخانه شیء ADODB محتویات جدول employees در پایگاه داده company را بازیابی کند؛ به علاوه امکان حرکت روی رکوردها نیز وجود داشته باشد برای انجام این مثال به این ترتیب عمل کنید:

۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE همراه یک فرم مانند شکل ۲۹-۶ همراه با کنترل‌های آن ایجاد کنید.



شکل ۲۹-۶

۲- کتابخانه شیء ADODB را به پروژه خود اضافه کرده و این دستورات را در بخش تعاریف فرم تایپ کنید تا بتوان از آنها در تمام رویدادها استفاده کرد.

```
Dim cncompany As Connection
```

```
Dim cmemployees As Command
```

```
Dim rsemployees As Recordset
```

۳- در این مرحله رویداد Load فرم را به این صورت تنظیم کنید:

```
Private Sub Form_Load()
```

```
Set cncompany = New Connection
```

```
Set cmemployees = New Command
```

```
Set rsemployees = New Recordset
```

```
cncompany.CursorLocation = adUseClient
```

```
rsemployees.CursorType = adOpenStatic
```

```
cncompany.ConnectionString = _
```

```
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\company.mdb"
```

```
cncompany.Open
```

```
cmemployees.ActiveConnection = cncompany
```

```
cmemployees.CommandText = "select * from employees"
```

```
cmemployees.CommandType = adCmdText
```

```
Set rsemployees = cmemployees.Execute
```

```
txtid.Text = rsemployees!employeeid
```

```
txtfirstname.Text = rsemployees!firstname
```

```
txtlastname.Text = rsemployees!lastname
```

```
txtcity.Text = rsemployees!city
```

```
txtsalary.Text = rsemployees!salary
```

```
End Sub
```

۴- در این رویداد پس از ایجاد اشیای موردنیاز و تنظیم خاصیت‌های آن‌ها، اتصال باز و دستور Select اجرا می‌شود و رکوردهای بازیابی شده در شیء rsrecordset قرار می‌گیرند و در پایان محتویات ستون‌های اولین رکورد در کادرهای متن مربوط به هر یک از آن‌ها نمایش داده می‌شود. اکنون رویداد دکمه‌های فرمان را به این صورت تنظیم کنید:

```
Private Sub cmdfirst_Click()
```

```
    rsemployees.MoveFirst
```

```
    txtid.Text = rsemployees! employeeid
```

```
    txtfirstname.Text = rsemployees ! firstname
```

```
    txtlastname.Text = rsemployees ! lastname
```

```
    txtcity.Text = rsemployees ! city
```

```
    txtsalary.Text = rsemployees ! salary
```

```
End Sub
```

```
Private Sub cmdlast_Click()
```

```
    rsemployees.MoveLast
```

```
    txtid.Text = rsemployees ! employeeid
```

```
    txtfirstname.Text = rsemployees ! firstname
```

```
    txtlastname.Text = rsemployees ! lastname
```

```
    txtcity.Text = rsemployees ! city
```

```
    txtsalary.Text = rsemployees ! salary
```

```
End Sub
```

```
Private Sub cmdnext_Click()
```

```
If Not rsemployees.EOF Then rsemployees.MoveNext
```

```
If rsemployees.EOF Then rsemployees.MoveLast
```

```
txtid.Text = rsemployees ! employeeid
```

```
txtfirstname.Text = rsemployees ! firstname
```

```
txtlastname.Text = rsemployees ! lastname
```

```
txtcity.Text = rsemployees ! city
```

```
txtsalary.Text = rsemployees ! salary
```

```
End Sub
```

```
Private Sub cmdprevious_Click()
```

```
If Not rsemployees.BOF Then rsemployees.MovePrevious
```

```
If rsemployees.BOF Then rsemployees.MoveFirst
```

```
txtid.Text = rsemployees ! employeeid
```

```
txtfirstname.Text = rsemployees ! firstname
```

```
txtlastname.Text = rsemployees ! lastname
```

```
txtcity.Text = rsemployees ! city
```

```
txtsalary.Text = rsemployees ! salary
```

```
End Sub
```

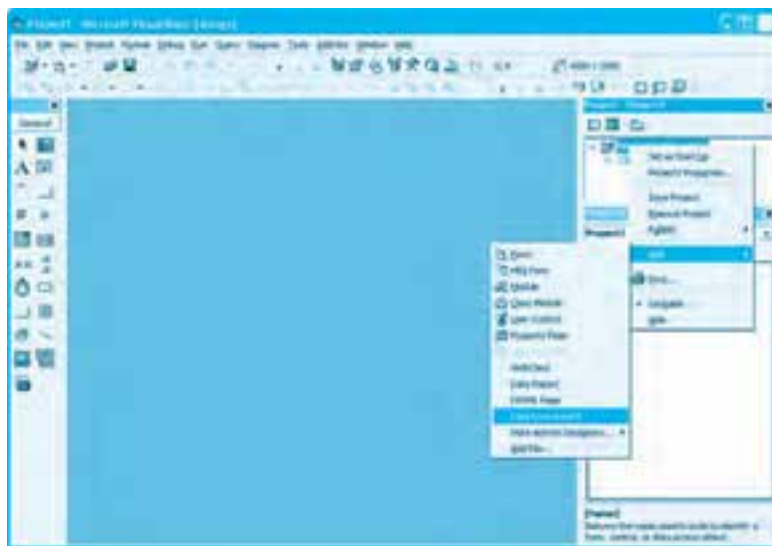
در این رویدادها با استفاده از متدهای شیء Recordset امکان حرکت روی رکوردها به وجود می‌آید و در هر رویداد محتویات ستون‌های رکورد جاری در کنترل‌های متن قرار می‌گیرد تا نمایش داده شوند. پروژه و فرم را با نام adobcompany ذخیره نموده سپس پروژه را اجرا کنید و نحوه عملکرد دکمه‌های فرمان را بررسی نمایید. به اجرای برنامه خاتمه داده و به پنجره ویژوال بیسیک بازگردید.

۶-۲-۶ نحوه استفاده از Data Environment

Data Environment (که از این به بعد به اختصار DE گفته می‌شود) یک محیط طراحی است که می‌توانید خاصیت‌های اشیای Command و Connection خود را در زمان طراحی پروژه مقداردهی کنید، برای رویدادهای ADO کد بنویسید، دستورات SQL را اجرا کرده و سایر عملیات را به راحتی انجام دهید. به علاوه با کمک محیط طراحی DE می‌توانید فرم‌ها و گزارشات خود را به آسانی با منابع پایگاه‌های داده متصل کرده و DE را جایگزین انواع دیگر روش‌های دسترسی به داده‌ها مثل کنترل‌های ADO، DataControl و مانند آن‌ها نمایید.

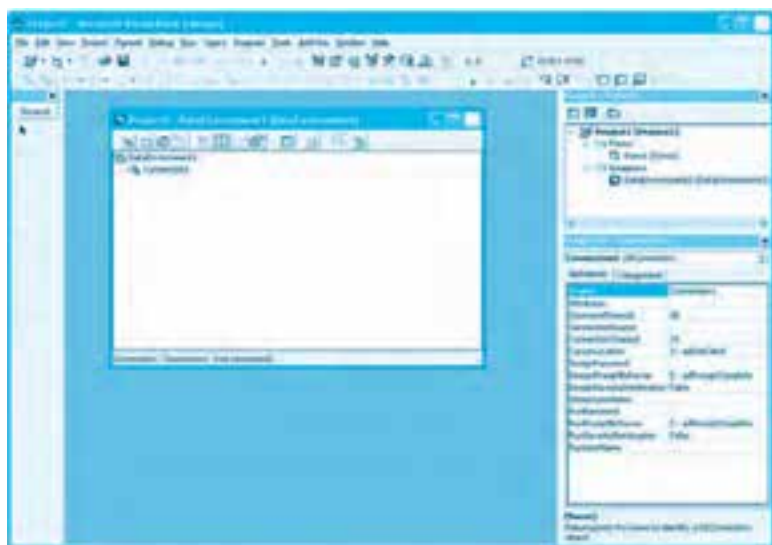
برای دسترسی به محیط طراحی DE و استفاده از امکانات آن می‌توان به این ترتیب عمل کرد:

۱- یک پروژه از نوع Standard EXE ایجاد کرده و در پنجره پروژه کلیک راست کنید. در منویی که نمایش داده می‌شود ابتدا روی گزینه Add و سپس روی گزینه Data Environment کلیک کنید (شکل ۶-۳۰).




شکل ۶-۳۰

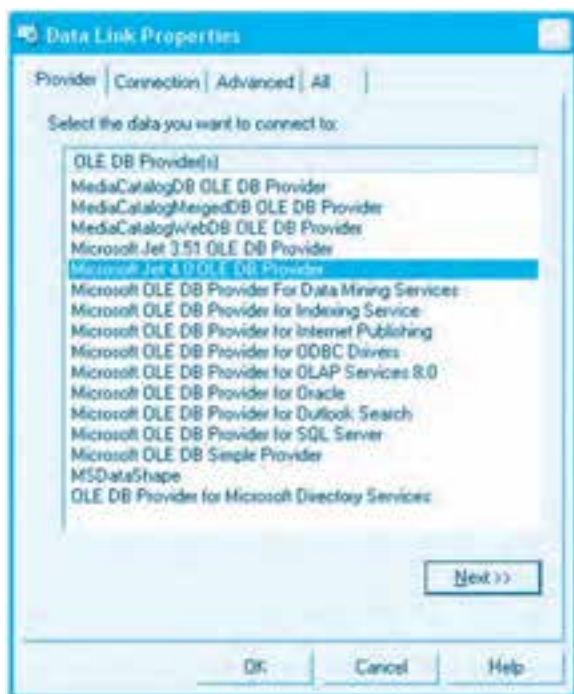
در این مرحله یک محیط طراحی DE جدید با نام Data Environment به پروژه اضافه شده و پنجره آن مانند شکل ۶-۳۱ داخل پنجره ویژوال نمایش داده می‌شود.



شکل ۶-۳۱


نکته: در صورتی که نرم‌افزار Crystal Report روی سیستم نصب باشد برای اضافه کردن یک DE جدید پس از کلیک روی گزینه Add روی گزینه More ActiveX Designer ... و بعد روی گزینه Data Environment کلیک کنید.

هنگام ایجاد یک محیط طراحی DE جدید به طور خودکار یک شیء Connection با نام Connection1 نیز ایجاد شده است. برای تنظیم این شیء Connection می‌توانید روی شیء Connection1 کلیک راست کرده و در منویی که نمایش داده می‌شود روی گزینه ... Properties کلیک کنید یا در نوار ابزار پنجره DE روی دکمه  کلیک کنید. در این صورت کادرمحاوره Data Link Properties مانند شکل ۳۲-۶ نمایش داده می‌شود.



شکل ۳۲-۶


در این کادرمحاوره و در زبانه Provider می‌توانید گزینه موردنظر خود را با توجه به نوع پایگاه داده انتخاب کنید (شکل ۳۲-۶)، سپس روی دکمه فرمان Next کلیک کنید. در این صورت وارد زبانه Connection

می‌شوید (شکل ۶-۳۳) در این زبانه می‌توانید مسیر و نام فایل پایگاه داده خود را (mdb) در اولین کادر متن تایپ کرده یا روی دکمه  کلیک کرده و پایگاه داده خود را به وسیله کادرمحاوره Select Access Database انتخاب کنید؛ سپس روی دکمه فرمان Test Connection کلیک کنید تا اتصال به پایگاه داده آزمایش شود.

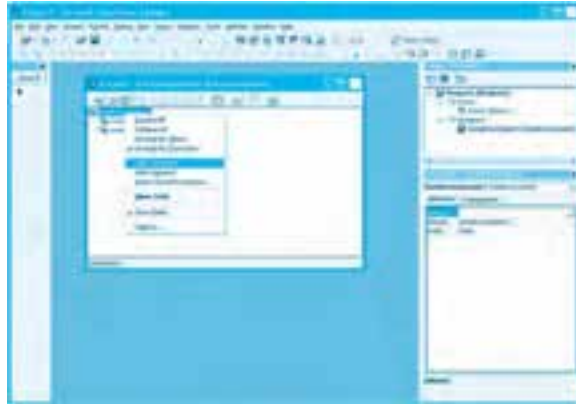
اکنون شیء Connection1 آماده تنظیم شده است. برای هماهنگی بیشتر نام شیء Connection1 را به نام cncompany تغییر دهید. برای این کار روی شیء Connection1 در پنجره DE کلیک راست کرده و در منویی که نمایش داده می‌شود، گزینه Rename را انتخاب نمایید؛ سپس نام جدید را تایپ کرده و کلید Enter را بفشارید.




شکل ۶-۳۳

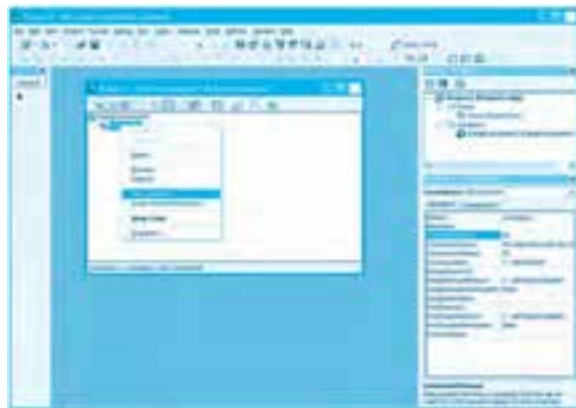
در محیط DE می‌توانید چندین شیء Connection بسازید. برای این کار می‌توانید در پنجره DE روی آیکن Data Environment کلیک راست کرده و در منویی که نمایش داده می‌شود روی گزینه Add Connection کلیک کنید (شکل ۶-۳۴) البته برای این کار می‌توانید در نوار ابزار پنجره DE روی دکمه  نیز کلیک کنید.

تمرین: یک پروژه از نوع Standard EXE ایجاد کرده و بعد یک محیط DE به آن اضافه کنید، سپس یک شیء Connection برای اتصال به پایگاه داده Company در آن ایجاد کنید.



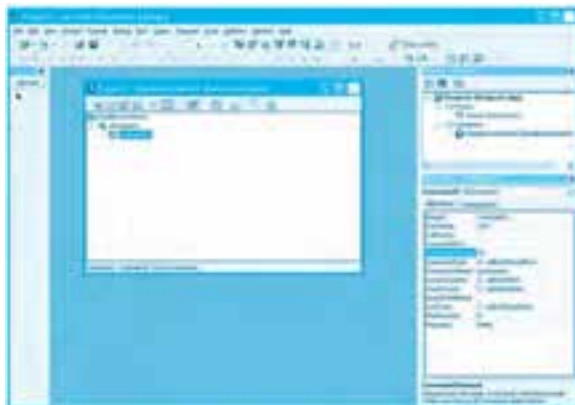
شکل ۳۴-۶

۲- پس از ایجاد شیء Connection باید یک شیء Command برای اجرای دستورات SQL ایجاد کنید. برای این کار در پنجره DE روی شیء cncompany کلیک راست کرده و در منویی که نمایش داده می‌شود روی گزینه Add Command کلیک کنید (شکل ۳۵-۶) یا در نوار ابزار پنجره DE روی دکمه  کلیک کنید.




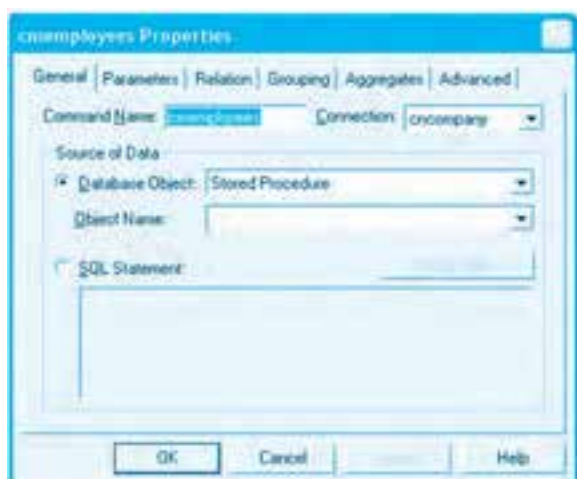
شکل ۳۵-۶

اکنون یک شیء Command جدید با نام Command1 ایجاد شده است (شکل ۳۶-۶).



شکل ۳۶-۶

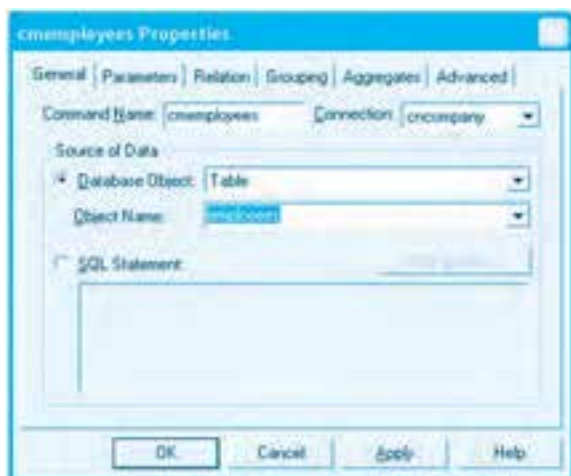
نام شیء Command1 را با همان روشی که نام شیء Connection را تغییر داده‌اید به cmemployees تغییر دهید. در این مرحله روی شیء cmemployees کلیک راست کرده و در منوی که نمایش داده می‌شود روی گزینه Properties کلیک کنید یا پس از کلیک روی نام شیء Command، روی دکمه  در نوار ابزار پنجره DE کلیک کنید. در این صورت کادرمحاوره‌ای مانند شکل ۳۷-۶ نمایش داده می‌شود.



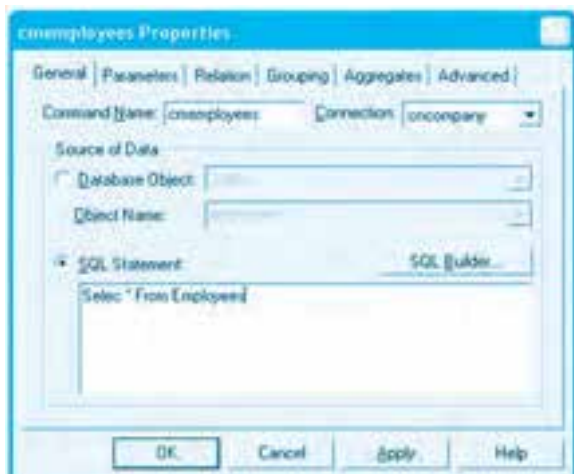
شکل ۳۷-۶

در این کادرمحاوره نام شیء Command و شیء Connection که روی آن ایجاد شده است نمایش داده می‌شود؛ در صورت لزوم می‌توان شیء Connection مربوط به شیء Command موردنظر را از کادر لیست ترکیبی بازشوی مربوطه انتخاب نمود و تغییر داد. اکنون در کادر لیست ترکیبی بازشوی Database Object می‌توانید روی گزینه Table برای انتخاب یک جدول یا گزینه View برای انتخاب


یک پرس‌وجوی (Query) ذخیره شده در پایگاه داده خود، کلیک کنید.
با توجه به گزینه انتخاب شده در این کادر لیست ترکیبی بازشو می‌توانید جدول یا پرس‌وجوی (Query) موردنظر را از کادر لیست ترکیبی بازشوی Object Name انتخاب کنید (شکل ۳۸-۶).



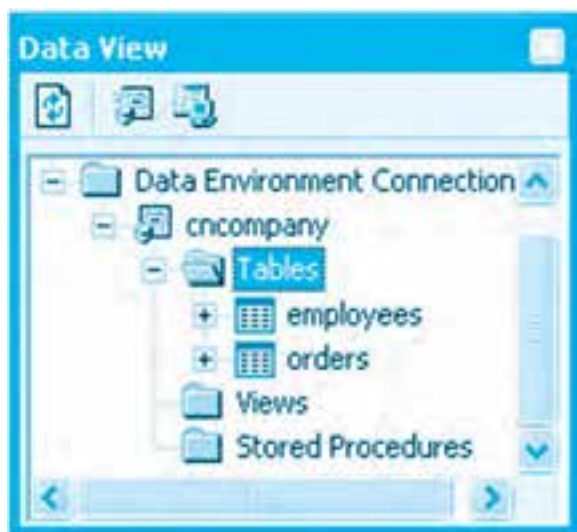
شکل ۳۸-۶



شکل ۳۹-۶

در این مرحله می‌توانید روی دکمه فرمان OK کلیک کنید تا تنظیم شیء Command کامل شده و خاتمه یابد. در این صورت آیکن شیء Command نیز در پنجره DE تغییر می‌کند و به شکل  درمی‌آید. به علاوه می‌توانید در کادرمحاوره cmemployees Properties به جای کلیک روی دکمه انتخاب Database Object روی دکمه انتخاب

SQL Statement کلیک کنید؛ در این صورت دستور SQL موردنظر را در کادر متن زیر این دکمه انتخاب تایپ کرده و روی دکمه فرمان OK کلیک کنید یا روی دکمه فرمان ... SQL Builder کلیک کنید (شکل ۶-۳۹). در این صورت دو پنجره دیگر برای طراحی پرس‌وجو نمایش داده می‌شود: پنجره Data View و پنجره Query Designer. پنجره Data View شیء Connection را نشان می‌دهد. اگر روی دکمه + در کنار نام آن کلیک کنید ساختار درختی پایگاه داده نمایش داده می‌شود (شکل ۶-۴۰).





شکل ۶-۴۰

پنجره Query Designer شباهت زیادی به پنجره طراحی Query در نرم‌افزارهای Access و SQL Server دارد. از این پنجره برای طراحی آسان‌تر Query می‌توان استفاده کرد (شکل ۶-۴۱).




شکل ۶-۴۱

پس از طراحی پرس‌وجو آن را ذخیره کنید و پنجره‌های Data View و Query Designer را ببینید تا به پنجره DE بازگردید.
در هر دو حالت آیکن شیء cmemployees به این شکل  تغییر می‌کند.
اکنون شیء Command نیز آماده شده است.

نکته: در صورتی که کار با پنجره Query Designer را نمی‌دانید به ضمیمه مربوطه یا به منابع مربوط به پایگاه‌های داده Access یا SQL Server مراجعه کنید. 

تمرین: در محیط DE پروژه خود یک شیء Command برای بازیابی تمام رکوردهای جدول employees و یک شیء Command برای بازیابی تمام رکوردهای جدول Orders ایجاد کنید.
۳- در این مرحله می‌توان از DE ایجاد شده برای طراحی فرم‌ها استفاده کرد و کنترل‌های خود را به ستون‌های موردنظر متصل کرد. وقتی برنامه اجرا می‌شود برای هر شیء Connection و Command که در محیط طراحی DE ایجاد شده است اشیای مستقل از نوع Connection و Command و RecordSet ایجاد خواهد شد که می‌توانید برای آن‌ها کدهای خود را در پنجره کدنویسی تایپ کنید. برای اتصال کنترل‌ها می‌توانید خاصیت DataSource را روی نام Data Environment، خاصیت DataMember را روی نام شیء Command که در محیط DE ایجاد شده است و خاصیت DataField را روی نام ستونی که می‌خواهید، تنظیم کنید.

نکته: اگر نام شیء Command در محیط DE cmemployees باشد، نام شیء Recordset آن rscmemployees خواهد بود. 

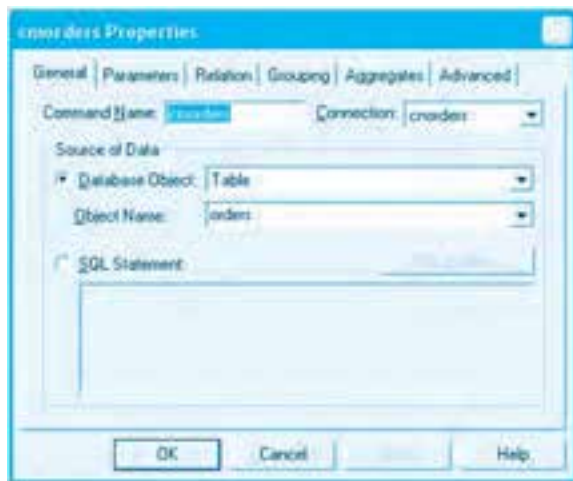
نکته: در صورتی که بخواهید یک شیء را در محیط DE حذف کنید پس از کلیک روی آیکن شیء، روی آن کلیک راست کرده و در منویی که نمایش داده می‌شود روی گزینه Delete کلیک کنید یا پس از انتخاب شیء روی دکمه  در نوار ابزار DE کلیک کنید.

مثال ۱۱: می‌خواهیم پروژه‌ای طراحی کنیم که با استفاده از محیط DE کلیه رکوردهای جدول orders employees را از پایگاه داده company بازیابی کند و در روی یک فرم نمایش دهد. برای انجام این مثال این عملیات را به ترتیب انجام دهید:
۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE همراه با یک فرم و کنترل‌های آن مانند شکل ۴۲-۶ ایجاد کنید.



شکل ۶-۴۲

- ۲- یک محیط طراحی DE جدید به پروژه اضافه کرده و نام آن را به decompany تغییر دهید. سپس یک شیء Connection در decompany با نام cnorders ایجاد کرده و آن را به پایگاه داده Company متصل کنید.
- ۳- در این مرحله روی شیء cnorders کلیک کرده و سپس یک شیء Command جدید با نام cmorders ایجاد کنید و خاصیت‌های آن را مانند شکل ۶-۴۳ تنظیم کنید.



شکل ۶-۴۳

- ۴- به پنجره طراحی فرم رفته و روی کادر متن txtid کلیک کنید، سپس در پنجره خواص خاصیت

DataSource آن را روی مقدار decompany، خاصیت DataMember آن را روی مقدار cmorders و خاصیت DataField آن را روی مقدار employeedid تنظیم کنید.

۵- سایر کنترل‌های کادر متن را نیز به همین شکل به ستون‌های متناظرشان متصل کنید.

۶- اکنون رویداد Click دکمه فرمان Next را به این صورت تنظیم کنید:

```
Private Sub cmdnext_Click()
```

```
    decompany.rscmorders.MoveNext
```

```
    If decompany.rscmorders.EOF Then decompany.rscmorders.MoveLast
```

```
End Sub
```

در این رویداد با توجه به نام شیء cmorders، نام شیء Recordset، rscmorders خواهد بود که با استفاده از متد MoveNext آن می‌توان روی رکوردها به سمت جلو حرکت کرد.

۷- رویداد Click سایر دکمه‌های فرمان را نیز به صورت مناسب تنظیم کنید.

۸- پروژه، فرم و محیط طراحی DE را به ترتیب با نام‌های newemployees، showemployees و decompany ذخیره کنید سپس پروژه را اجرا، آزمایش و بررسی کنید.

۹- به اجرای پروژه خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.

نکته: پسوند فایل‌های مربوط به **Data Environment, DSR و DCA** می‌باشد.



تمرین: پروژه‌های طراحی کنید که بتوان به وسیله آن و با استفاده از محیط طراحی DE اطلاعات مربوط به کارمندان یک شهر خاص را در یک کنترل DataGrid نمایش داد.

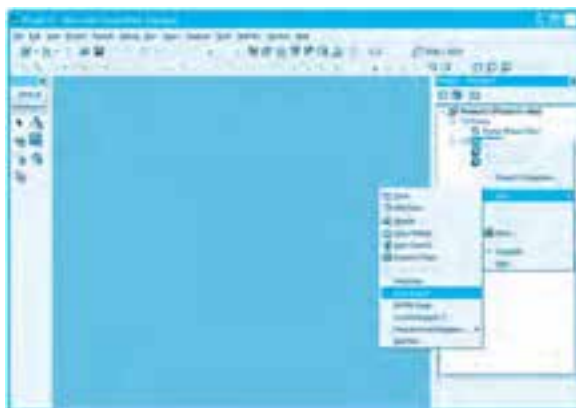


۷-۲-۶ نحوه ایجاد گزارش

یکی از ویژگی‌های مهم یک برنامه، گزارش‌هایی است که می‌تواند در اختیار کاربران قرار دهد. یکی از دلایل ذخیره‌سازی و نگهداری داده و اطلاعات این است که در زمان لازم بتوان از داده‌ها، اطلاعات و پردازش آن‌ها، به نتیجه‌گیری خاصی رسید به عبارت دیگر از جمع‌بندی و تحلیل داده‌ها و نتایج حاصل از پردازش آن‌ها استفاده مناسبی نمود. در این جاست که نقش طراحی و ارائه گزارش‌های مناسب در یک برنامه مشخص می‌شود. یک گزارش با ارائه داده‌ها و اطلاعات که به شکل مناسب نمایش داده شده یا روی کاغذ چاپ می‌شوند، می‌تواند به کاربران در رسیدن به اهدافشان کمک کند. گزارش‌ها با توجه به این که می‌توانند انواع و شکل‌های مختلفی را برای نمایش داده‌ها و اطلاعات و نتایج حاصل از محاسبات روی آن‌ها و ترتیب نمایش آن‌ها نسبت به یکدیگر ارائه کنند، نقش مهمی را در یک برنامه کاربردی ایفا می‌نمایند. ویژوال بیسیک در این زمینه نیز ابزار قدرتمندی را ارائه می‌کند که در این جا بررسی می‌شود.

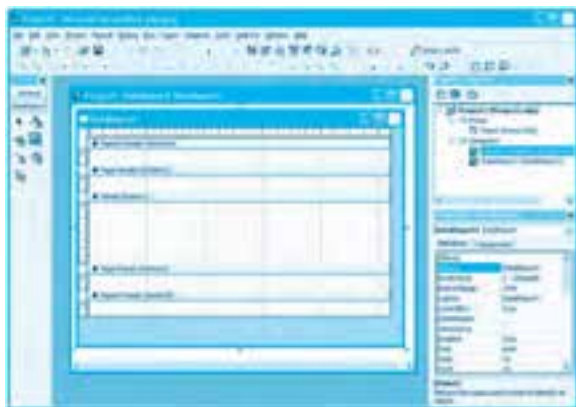
ایجاد گزارش با Data Report

برای ایجاد گزارش در ویژوال بیسیک پس از طراحی یک Data Environment مناسب می‌توانید به این صورت عمل کنید. در پنجره پروژه کلیک راست کرده و از منویی که نمایش داده می‌شود، ابتدا روی گزینه Add و سپس گزینه Data Report کلیک کنید (شکل ۶-۴۴).



شکل ۶-۴۴

در این صورت یک گزارش (Data Report) به پروژه اضافه و پنجره طراحی آن داخل پنجره ویژوال بیسیک نمایش داده می‌شود (شکل ۶-۴۵).



شکل ۶-۴۵

به‌طور کلی هر گزارش مانند شکل ۶-۴۵ به‌طور پیش‌فرض از پنج بخش تشکیل می‌شوند. این بخش‌ها عبارتند از:

الف- بخش سرصفحه گزارش یا Report Header

این بخش در قسمت بالای پنجره طراحی گزارش قرار دارد. در این بخش می‌توان اطلاعاتی را که لازم است یک‌بار در ابتدای گزارش نمایش داده شوند، قرار داد. به عنوان مثال نام گزارش یا نام شرکت یا سازمانی که گزارش برای آن‌ها تهیه شده است و موارد مشابه.

ب- بخش سرصفحه یا Page Header

این بخش در زیر بخش Report Header قرار دارد و می‌توان اطلاعاتی که در این قسمت قرار دارند در بالای هر صفحه از گزارش نمایش داد.

ج- بخش جزئیات یا Details

این بخش از مهم‌ترین بخش‌های یک گزارش است؛ در این بخش محتویات فیلدها و رکوردهای بازیابی شده نمایش داده می‌شوند.

د- بخش پاصفحه یا Page Footer

این بخش در زیر بخش جزئیات قرار دارد و محتویات این بخش در پایین هر صفحه از گزارش قرار داده می‌شوند.

ه- بخش پاصفحه گزارش یا Report Footer

این بخش آخرین قسمت در یک گزارش است و محتویات این بخش در پایان گزارش نمایش داده شده یا چاپ می‌شوند. وقتی پنجره طراحی گزارش باز باشد، یک جعبه ابزار مانند شکل ۴۶-۶ فعال می‌شود. با استفاده از کنترل‌هایی که در این جعبه ابزار وجود دارد، می‌توانید کنترل‌هایی مانند کنترل‌های استاندارد برچسب، کادر متن، شکل (Shape)، تصویر و خط را که در طراحی فرم‌ها استفاده می‌شوند، برای طراحی گزارش مورد استفاده قرار دهید.

جدول ۵-۶ کنترل‌های محیط طراحی گزارش

نام کنترل	توضیح
RptLabel	کنترل برچسب
RptTextBox	کنترل کادر متن
RptImage	کنترل تصویر
RptLine	کنترل خط
RptShape	کنترل شکل
RptFunction	کنترل تابع



شکل ۴۶-۶


کنترل RptFunction یک کنترل جدید در محیط طراحی گزارش است. با این کنترل می‌توان عملیات محاسباتی گوناگونی را روی داده‌ها انجام داد، به عنوان مثال محاسبه مجموع، تعداد، متوسط، بزرگ‌ترین مقدار و مانند آن‌ها. خاصیت FunctionType در این کنترل نوع عملیاتی را که کنترل انجام می‌دهد، مشخص می‌کند. این خاصیت می‌تواند روی یکی از مقادیر ارائه شده در جدول ۶-۶ تنظیم شود.

جدول ۶-۶ مقادیر مربوطه به خاصیت FunctionType

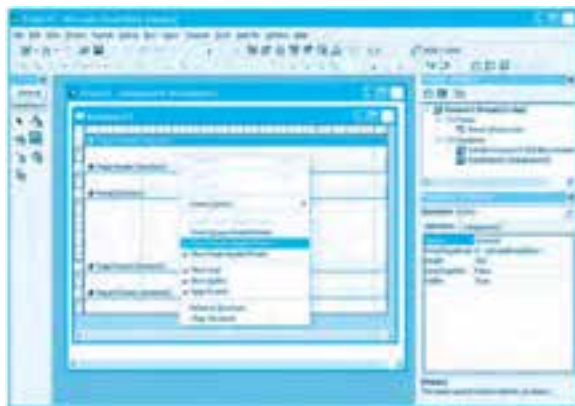
ثابت رشته‌ای	ثابت عددی	عملکرد
rptFuncSum	۰	محاسبه مجموع داده‌ها
rptFuncAve	۱	محاسبه متوسط داده‌ها
rptFuncMin	۲	محاسبه کوچک‌ترین مقدار داده در یک گروه
rptFuncMax	۳	محاسبه بزرگ‌ترین مقدار داده در یک گروه
rptFuncRcnt	۴	محاسبه تعداد رکوردها
rptFuncVcnt	۵	محاسبه تعداد فیلدهایی که دارای مقدار هستند (خالی نیستند).

برای نمایش داده و اتصال گزارش به رکوردها و ستون‌های داده می‌توانید از Data Environment استفاده کنید.

برای اتصال گزارش به یک Data Environment می‌توانید خاصیت DataSource را روی نام Data Environment و خاصیت DataMember آن را روی نام شیء Command که در Data Environment وجود دارد، تنظیم کنید.

نکته:  در صورتی که روی پنجره طراحی گزارش کلیک راست انجام دهید منویی مانند شکل

۴۷-۶ نمایش داده می‌شود. با استفاده از گزینه‌های موجود در این منو می‌توانید سرصفحه و پاصفحه گزارش (Show Report Header/Footer)، سرصفحه و پاصفحه (Show Page Header/Footer)، نقاط راهنما (Show Grid) و خط‌کش‌های افقی و عمودی (Show Rulers) را فعال یا غیرفعال کنید. با کلیک روی هر یک از این گزینه‌ها این بخش‌ها غیرفعال و دوباره فعال می‌شوند. به طور پیش‌فرض این بخش‌ها در پنجره گزارش فعال هستند.

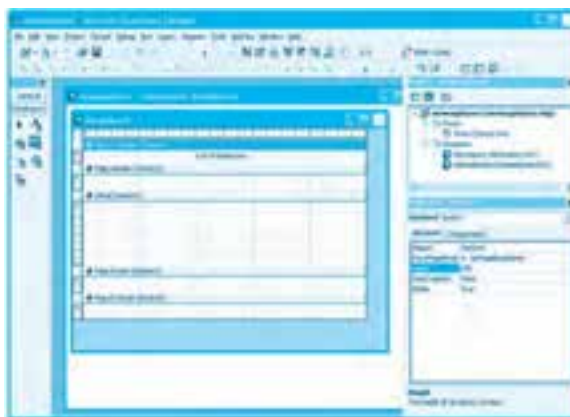


شکل ۴۷-۶

نکته: وقتی یک گزارش را ذخیره کنید اطلاعات آن در فایل با پسوند Dsr ذخیره می‌شود.



- مثال ۱۲:** می‌خواهیم پروژه newemployees را به شکلی تنظیم کنیم تا بتوان رکوردهای جدول employees را به صورت یک گزارش ساده مشاهده نمود. برای این کار این عملیات را به ترتیب انجام دهید:
- ۱- برنامه ویژوال بیسیک را اجرا کرده و پروژه newemployees را باز کنید.
 - ۲- در پنجره پروژه کلیک راست کنید و در منویی که نمایش داده می‌شود ابتدا روی گزینه Add و سپس روی گزینه Data Report کلیک کنید تا یک گزارش به پروژه اضافه شود، سپس نام آن را به rptemployees تغییر دهید.
 - ۳- به پنجره طراحی گزارش بروید و یک کنترل RptLabel روی بخش سرصفحه گزارش (Report Header) قرار دهید، سپس در پنجره خواص نام آن را به lblrh و عنوان آن را روی عبارت List Of Employees تنظیم کنید (شکل ۴۸-۶).




شکل ۴۸-۶

۴- در پنجره طراحی گزارش روی بخش سرصفحه (Page Header) کلیک راست کرده و در منویی که نمایش داده می‌شود ابتدا روی گزینه Insert Control و سپس Current Date (Short Format) کلیک کنید تا تاریخ جاری سیستم در زمان گزارش در سرصفحه نمایش داده شود، سپس موقعیت آن را مانند شکل ۶-۴۹ تنظیم کرده و یک کنترل برچسب در سمت چپ آن قرار دهید و عنوان کنترل برچسب را به: Date تغییر دهید.



شکل ۶-۴۹

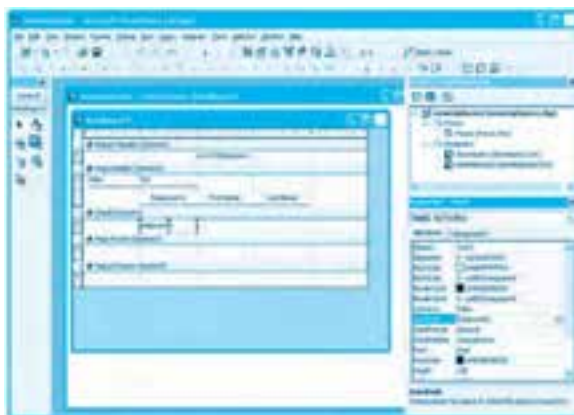
۵- در مرحله بعد یک کنترل خط در زیر کنترل‌های موجود در سرصفحه قرار دهید. (شکل ۶-۴۹)، سپس سه کنترل برچسب دیگر در زیر کنترل خط قرار داده و عنوان و موقعیت آن‌ها را مانند شکل ۶-۴۹ تنظیم کنید و بعد یک کنترل خط دیگر زیر آن‌ها قرار دهید.

نکته: در صورتی که بخواهید ارتفاع هر بخش را در گزارش تغییر دهید، روی بخش مربوطه کلیک کرده و خاصیت Height آن را در پنجره خواص تنظیم کنید یا اشاره‌گر را به قسمت بالای نوار مستطیل شکلی که نام بخش مربوطه در آن نمایش داده می‌شود منتقل کرده و پس از تغییر شکل اشاره‌گر ماوس به  عمل کشیدن ارتفاع بخش موردنظر را تنظیم کنید.

نکته: وقتی یک گزارش را ذخیره کنید اطلاعات آن در فایل با پسوند Dsr ذخیره می‌شود.

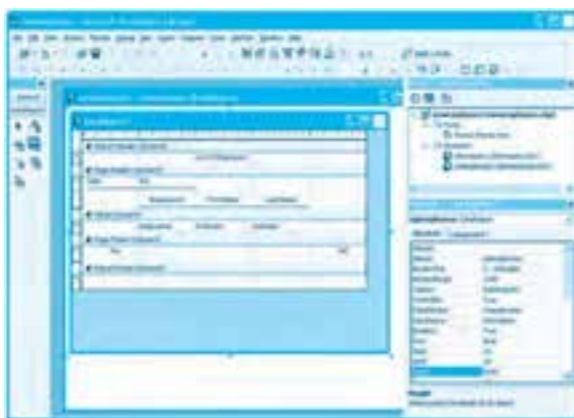
۶- در این مرحله می‌خواهیم رکوردهای جدول employees را نمایش دهیم. برای این کار ابتدا در پنجره پروژه روی نام گزارش کلیک کنید؛ سپس در پنجره خواص، خاصیت DataSource گزارش را روی کنترل decompany و خاصیت DataMember آن را روی موقعیت cmemployees تنظیم کنید.

۷- اکنون ارتفاع بخش جزئیات (Details) را تنظیم کرده و یک کنترل RptTextBox مانند شکل ۶-۵۰ داخل این بخش قرار دهید، سپس در پنجره خواص مقدار خاصیت DataMember آن را روی مقدار cmemployees و مقدار خاصیت DataField را روی مقدار employeeid تنظیم کنید تا این کنترل به ستون employeeid رکوردها متصل شود.



شکل ۶-۵۰

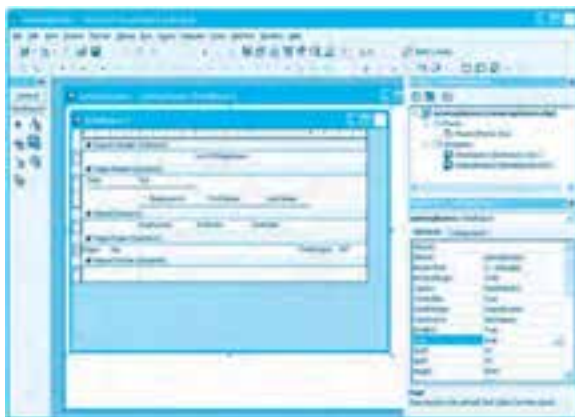
۸- در این مرحله نیز دو کنترل کادر متن دیگر داخل بخش جزئیات قرار داده و آن‌ها را به ستون‌های Firstname و Lastname متصل کنید (شکل ۶-۵۱).



شکل ۶-۵۱

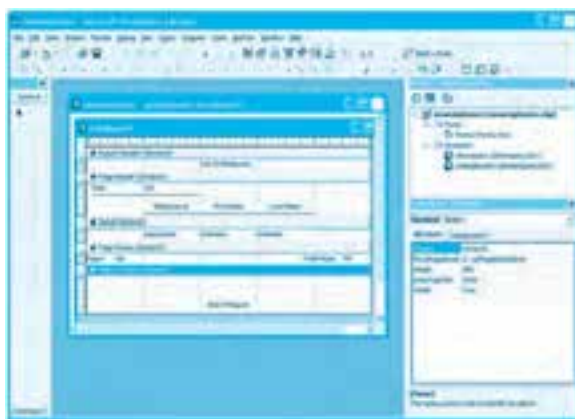
۹- اکنون در بخش پاصفحه (Page Footer) کلیک راست کنید و در منویی که نمایش داده می‌شود، ابتدا روی گزینه Insert Control و سپس روی گزینه Current Page Number کلیک کنید و کنترل برچسب

ایجاد شده را در سمت چپ این بخش قرار دهید. این کنترل شماره هر صفحه را در پایین هر صفحه نمایش می‌دهد و به همین شکل با گزینه Total Number of Pages یک کنترل برچسب برای نمایش تعداد کل صفحه‌ها در پایین هر صفحه ایجاد کرده و آنرا در سمت راست این بخش قرار دهید (شکل ۶-۵۱).
 ۱۰- سپس در سمت چپ کنترل‌های برچسبی که در مرحله ۹ ایجاد کرده‌اید دو کنترل برچسب با عنوان Total Pages: و pages: قرار دهید (شکل ۶-۵۲).



شکل ۶-۵۲

۱۱- اکنون به بخش پاصفحه گزارش (Report Footer) بروید و یک کنترل برچسب در این بخش قرار دهید و خاصیت عنوان آنرا روی عبارت End Of Report تنظیم کنید (شکل ۶-۵۳) و در پایان عنوان گزارش را نیز روی عبارت List of Employees تنظیم کنید.



شکل ۶-۵۳

۱۲- در این مرحله گزارش آماده شده است و می‌توان از آن استفاده نمود، بنابراین به پنجره طراحی فرم بروید و یک دکمه فرمان با نام cmdshow و عنوان Show Report روی فرم قرار دهید، سپس این دستور را در رویداد Click آن تایپ کنید:

```
Private Sub cmdshow_Click()  
    rptemployees.Show  
End Sub
```

در این رویداد با استفاده از متد Show گزارش rptemployees نمایش داده می‌شود.

۱۳- یک دکمه فرمان دیگر با نام cmdprint و عنوان PrintReport روی فرم قرار داده و رویداد Click آن را به این صورت تنظیم کنید:

```
Private Sub cmdprint_Click()  
    rptemployees.PrintReport  
End Sub
```

در این رویداد با استفاده از متد PrintReport گزارش به چاپگر پیش‌فرض ارسال می‌شود. اگر بخواهید قبل از چاپ، کادرمحاوره چاپ نمایش داده شود، می‌توانید از این متد استفاده کنید:

```
rptemployees.PrintReport True
```

۱۴- گزارش را با نام rptemployees ذخیره کنید، سپس تغییرات پروژه فرم را ذخیره کرده و پروژه را اجرا کنید و پس از نمایش فرم روی دکمه فرمان Show Report کلیک کنید. در این صورت گزارش در یک پنجره مستقل مانند شکل ۶-۵۴ و ۶-۵۵ نمایش داده می‌شود.



شکل ۶-۵۴



شکل ۵۵-۶

در بالای این پنجره یک نوار ابزار وجود دارد که به وسیله دکمه‌های آن می‌توانید گزارش را به چاپگر ارسال کنید یا گزارش را به صورت فایل با قالب‌های متنی یا HTML روی دیسک ذخیره نمایید و با استفاده از یک کادر لیست ترکیبی بازشو گزارش را روی صفحه نمایش کوچک یا بزرگ کنید. به علاوه در پایین پنجره گزارش دکمه‌هایی برای حرکت روی صفحه‌های گزارش وجود دارند. در زمانی که حجم گزارش بیش از یک صفحه باشد با این دکمه‌ها می‌توان روی صفحه‌های گزارش یک صفحه به عقب، جلو، صفحه اول یا آخر حرکت کرد.

۱۵- روی دکمه فرمان Print Report کلیک کنید تا گزارش به طور خودکار به چاپگر پیش‌فرض ارسال شود.

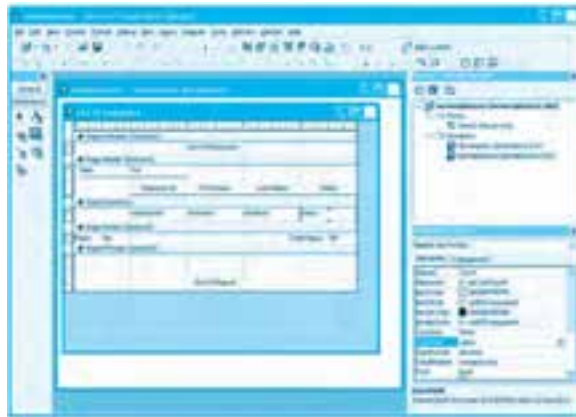
۱۶- به اجرای پروژه خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.

مثال ۱۳: می‌خواهیم گزارش rptemployees را در پروژه newemployees به شکلی تنظیم کنیم تا بتوان تعداد کارمندان و جمع حقوق آن‌ها را در گزارش مشاهده نمود. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

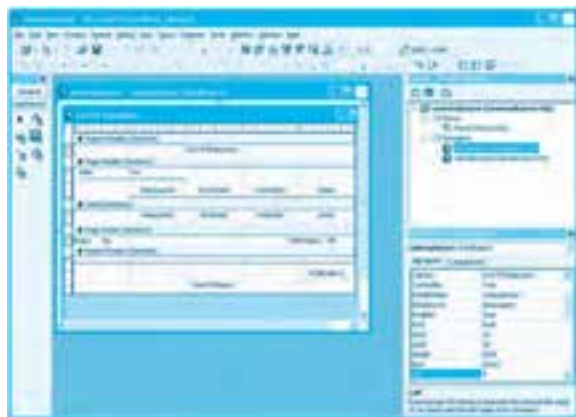
۱- برنامه ویژوال بیسیک را اجرا کرده و پروژه newemployees را باز کنید.

۲- به پنجره طراحی گزارش بروید و مانند شکل ۵۶-۶ یک کنترل برچسب در سرصفحه گزارش قرار داده و بعد در بخش جزئیات یک کنترل کادر متن قرار دهید و آن‌را به ستون Salary متصل کنید.

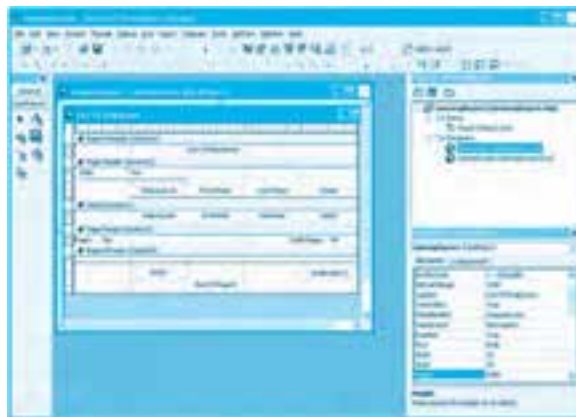
۳- در این مرحله در جعبه ابزار DataReport روی کنترل RptFunction کلیک کنید و با عمل کشیدن یک نمونه از این کنترل را در سمت راست بخش Report Footer قرار دهید (شکل ۵۷-۶)، سپس با همین روش یک کنترل RptFunction دیگر در سمت چپ این نوار بگذارید (شکل ۵۸-۶)، سپس نام این دو کنترل را به ترتیب به sumsalary و countemp تغییر دهید. اکنون خاصیت FunctionType کنترل counemp را نیز روی مقدار rptFuncRCnt تنظیم کنید تا این کنترل بتواند تعداد رکوردها را محاسبه کند. این خاصیت در کنترل sumsalary به طور خودکار روی مقدار rptFuncSum تنظیم شده است تا مجموع را محاسبه کند.



شکل ۵۶-۶

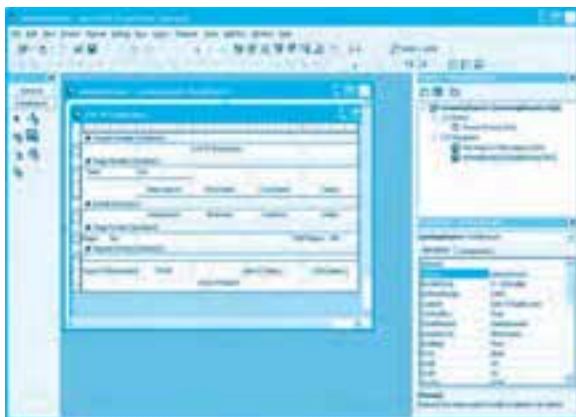


شکل ۵۷-۶



شکل ۵۸-۶

- ۴- در این مرحله خاصیت DataMember کنترل‌های sumsalary و countemp را روی مقدار cmemployees و خاصیت DataField آن‌ها را به ترتیب روی فیلدهای salary و employeeid تنظیم کنید.
- ۵- در این مرحله مانند شکل ۵۹-۶ دو کنترل برچسب با عنوان‌های Count of و Sum Of Salary: employees در سمت چپ کنترل‌های sumsalary و countemp قرار دهید.



شکل ۵۹-۶

- ۶- اکنون خاصیت Alignment تمام کنترل‌های موجود در پاصفحه گزارش را روی مقدار 2-rptJustifyCenter تنظیم کنید.
- ۷- تغییرات را ذخیره کنید و پروژه را اجرا نمایید، سپس روی دکمه فرمان Show Report کلیک کنید تا گزارش نمایش داده شود. این بار علاوه بر رکوردهای جدول کارمندان، مجموع حقوق و تعداد آن‌ها نیز در انتهای گزارش نمایش داده می‌شود (شکل ۶۰-۶).



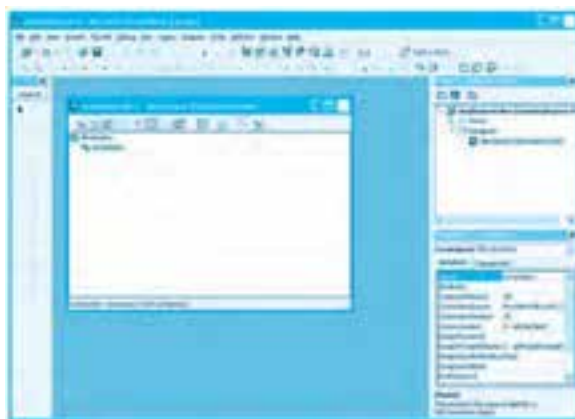
شکل ۶۰-۶

۸- به اجرای برنامه خاتمه داده و به پنجره ویژوال بیسیک بازگردید.

تمرین: پروژه‌ای طراحی کنید که به وسیله آن بتوان یک گزارش از کلیه سفارشات انجام شده در جدول orders تهیه کرد. به علاوه در انتهای گزارش مجموع، تعداد، بالاترین مبلغ و پایین‌ترین مبلغ سفارشات نیز نمایش داده شود. در ضمن امکان چاپ گزارش نیز وجود داشته باشد.

مثال ۱۴: می‌خواهیم پروژه‌ای طراحی کنیم که به وسیله آن بتوان به وسیله یک گزارش مشخصات کلیه سفارش‌ها را به همراه اطلاعات کارمندی که سفارش‌های مربوطه را انجام داده‌اند مشاهده کرد. برای این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE ایجاد کنید، سپس یک Data Environment جدید با نام decompany به پروژه اضافه کرده و شیء connection آن را برای اتصال به پایگاه داده company تنظیم کنید و نام آن را به cncompany تغییر دهید (شکل ۶-۶۱).

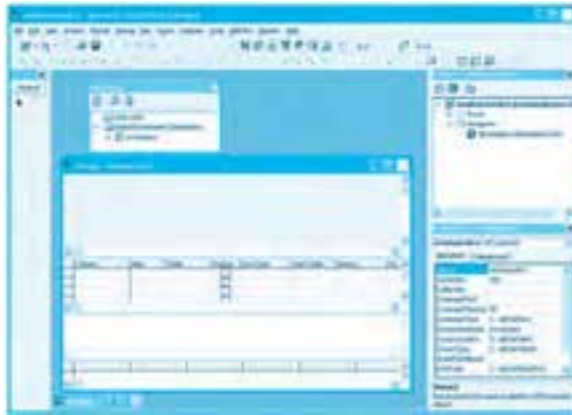


شکل ۶-۶۱

۲- اکنون روی شیء cncompany کلیک کرده و یک شیء command جدید در آن ایجاد کنید سپس نام آن را به cmemporders تغییر داده و روی آن کلیک راست کنید و بعد روی گزینه Properties کلیک کنید تا کادرمحاوره ویژگی‌های شیء cmemporders نمایش داده شود.

نکته: پنجره‌های Data View و Query Designer را می‌توانید در هر جایی که مایل هستید و با اندازه دلخواه قرار دهید.

۳- در این کادرمحاوره و در زبانه General روی دکمه انتخاب SQR Statement و بعد روی دکمه فرمان SQL Builder کلیک کنید تا پنجره‌های Data View و Query Designer نمایش داده شوند. با استفاده از این پنجره‌ها می‌توانید به سرعت و آسانی انواع پرس‌وجوهای ساده یا پیچیده بین جدول‌های رابطه‌ای را طراحی کنید (شکل ۶-۶۲).



شکل ۶-۶۲

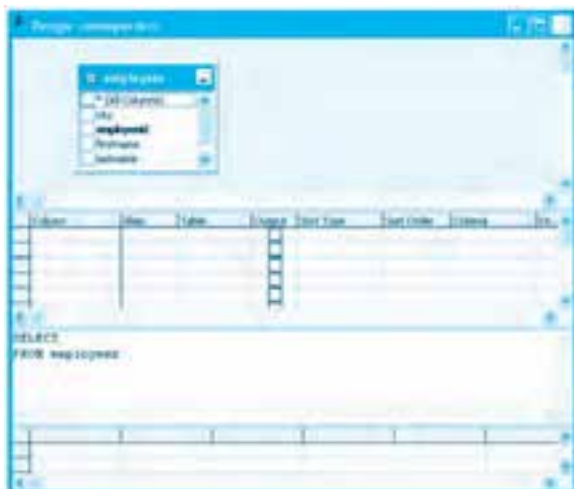
۴- در این مرحله به پنجره Data View بروید و ساختار درختی شیء cncompany را با کلیک روی علامت + باز کنید، سپس روی همین علامت در کنار شاخه Tables کلیک کنید تا اسامی جدول‌ها مانند شکل ۶-۶۳ نمایش داده شود.



شکل ۶-۶۳

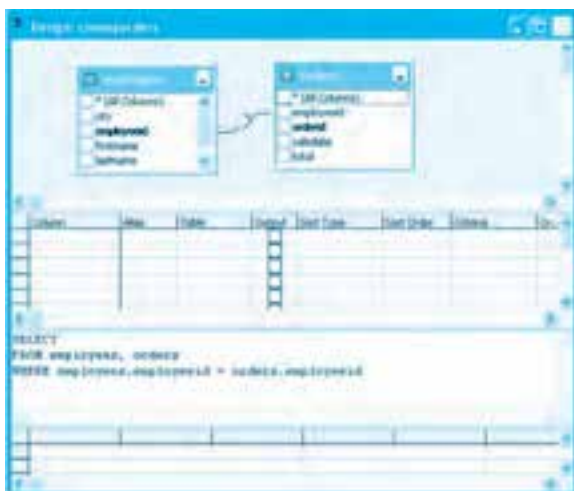
۵- به دلیل این‌که می‌خواهیم رکوردهای هر دو جدول بازیابی شوند، بنابراین روی نام جدول employees

در پنجره DataView کلیک کرده و سپس با عمل کشیدن آن را به بخش بالایی پنجره طراحی پرس‌وجو (Query Designer) انتقال دهید. در این صورت یک پنجره کوچک داخل پنجره طراحی پرس‌وجو با نام employees اضافه می‌شود و نام ستون‌های آن را نمایش می‌دهد (شکل ۶-۶۴). به علاوه یک دستور Select نیز در بخش دستور طراحی پرس‌وجو نمایش داده می‌شود.



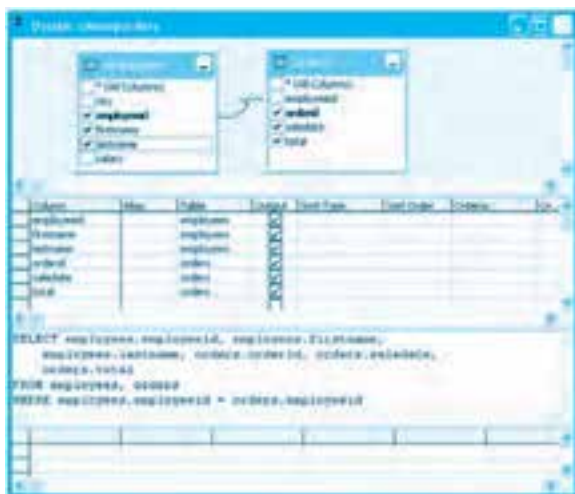
شکل ۶-۶۴

اکنون با همین روش جدول orders را به پنجره طراحی پرس‌وجو اضافه کنید. در این حالت رابطه بین دو جدول نیز با یک علامت گرافیکی نمایش داده می‌شود و دستور Select ایجاد شده و در بخش دستور در پنجره طراحی پرس‌وجو نمایش داده می‌شود (شکل ۶-۶۵).




شکل ۶-۶۵

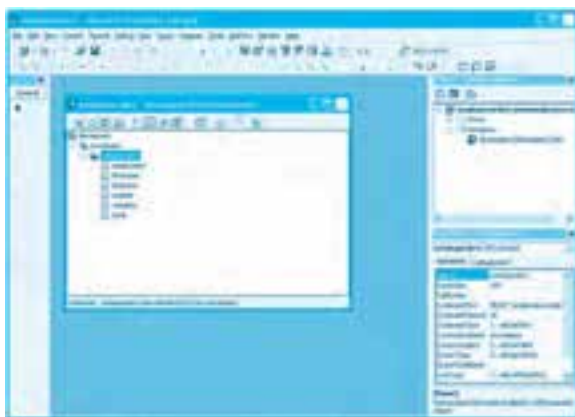
۶- در این مرحله در پنجره طراحی پرس‌وجو روی کادر علامت ستون‌های جدول‌های employess و orders مانند شکل ۶-۶۶ کلیک کنید تا این ستون‌ها به دستور Select اضافه شوند. این ستون‌ها هنگام اجرای شیء command ایجاد شده بازیابی می‌شوند.



شکل ۶-۶۶

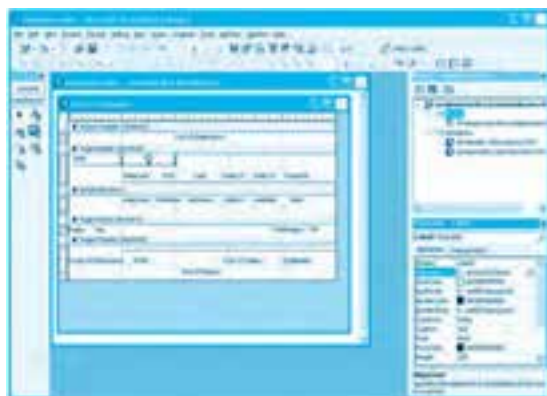
۷- اکنون در نوار ابزار استاندارد در پنجره ویژوال بیسیک روی دکمه  کلیک کنید تا پرس‌وجوی ایجاد شده ذخیره شود.

۸- در این مرحله به پنجره Data Environment بازگردید و روی علامت + کنار شیء cmemporders کلیک کنید؛ در این صورت نام ستون‌های انتخاب شده نمایش داده می‌شوند (شکل ۶-۶۷).



شکل ۶-۶۷

۹- اکنون یک گزارش (Data Report) جدید با نام rptemporders به پروژه اضافه کرده و بعد این گزارش را به decompany و شیء cmemporders متصل کنید. سپس گزارش را مانند شکل ۶-۶۸ طراحی کرده و کادرهای متن بخش جزئیات (Details) را به ستون‌های مربوط در شیء cmemporder متصل کنید و در بخش پاصفحه گزارش نیز یک کنترل rptFunction برای محاسبه مجموع ستون total و یک کنترل rptFunction دیگر برای شمارش تعداد رکوردها قرار دهید.



شکل ۶-۶۸

۱۰- در این مرحله گزارش آماده شده است. روی فرم یک دکمه فرمان با نام cmdshow و عنوان Show Report قرار دهید و رویداد آن‌را به این صورت تنظیم کنید:

```
Private Sub cmdshow_Click()  
    rptemployees.Show  
End Sub
```



شکل ۶-۶۹

۱۱- پروژه و فرم را به نام employeesorders ذخیره کنید و بعد پروژه را اجرا کرده و روی دکمه فرمان Show Report کلیک کنید. یک گزارش مانند شکل ۶۹-۶ نمایش داده می‌شود. در این گزارش می‌توان مشخصات کارمندی که هر سفارش فروش را انجام داده است، مشاهده کرد.

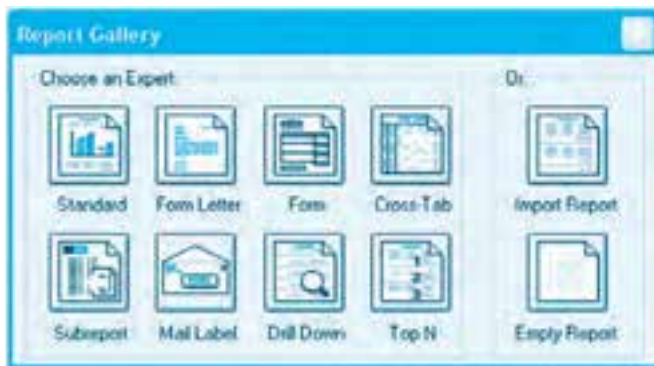
۱۲- به اجرای پروژه خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.

ایجاد گزارش با نرم‌افزار Crystal Report

برای ایجاد گزارش در ویژوال بیسیک، می‌توان علاوه بر Data Report از نرم‌افزار CrystalReport نیز استفاده کرد.

قبل از استفاده از این نرم‌افزار باید آن را به صورت جداگانه پس از نصب ویژوال استودیو روی سیستم نصب کرد.

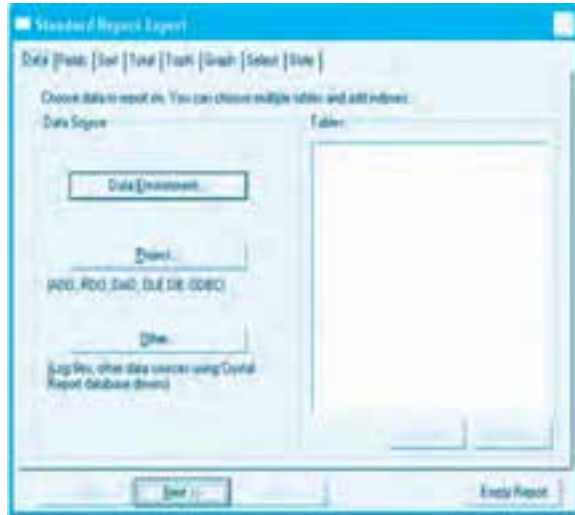
نصب این نرم‌افزار بسیار ساده است. پس از نصب، امکان استفاده از آن در محیط ویژوال بیسیک به آسانی امکان‌پذیر می‌باشد. برای ایجاد یک گزارش با Crystal Report در پروژه موردنظر در پنجره پروژه کلیک راست کنید و در منویی که نمایش داده می‌شود ابتدا روی گزینه Add و بعد روی گزینه Crystal Report کلیک کنید. در این مرحله کادرمحاوره Report Gallery نمایش داده می‌شود (شکل ۶-۷۰).



شکل ۶-۷۰

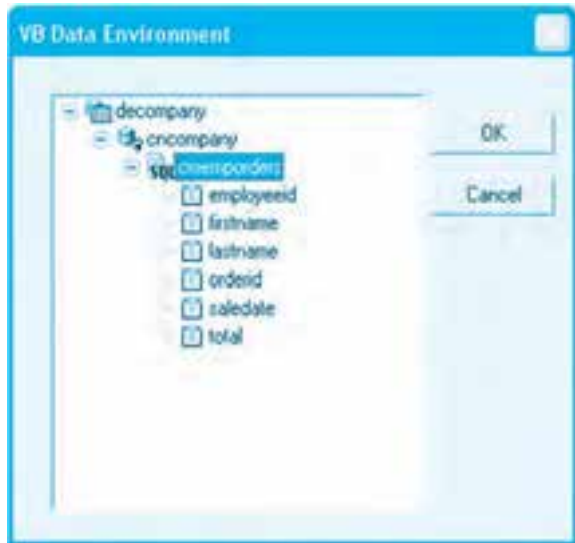
در این کادرمحاوره می‌توانید از امکانات آماده‌سازی یک گزارش با توجه به نوع موردنظر از مدل‌های موجود که به آن‌ها Wizard نیز می‌گویند، استفاده کنید.

روی دکمه Standard Report Expert کلیک کنید تا کادرمحاوره Standard Report Expert نمایش داده شود (شکل ۶-۷۱).



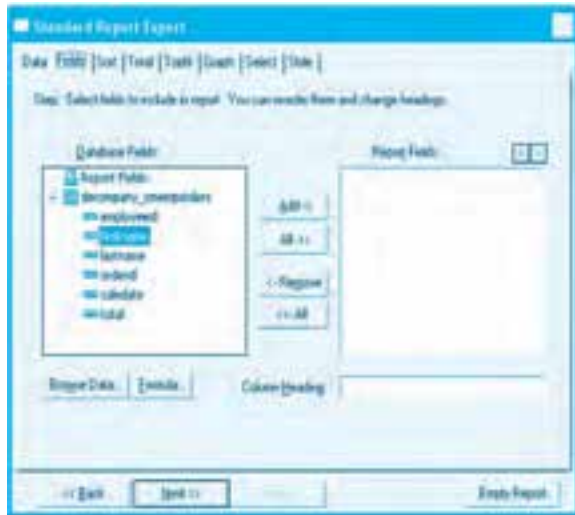
شکل ۶-۷۱

در این کادرمحاوره و در زبانه Data می‌توانید با استفاده از دکمه ... Data Environment یک DE از قبل طراحی شده را در پروژه انتخاب کنید تا از آن در طراحی گزارش برای دسترسی به رکوردها و فیلدهای آن‌ها استفاده کنید. پس از کلیک روی این دکمه فرمان، کادرمحاوره VB Data Environment نمایش داده می‌شود و ساختار درختی DE‌های موجود در پروژه را ارائه می‌کند (شکل ۶-۷۲).

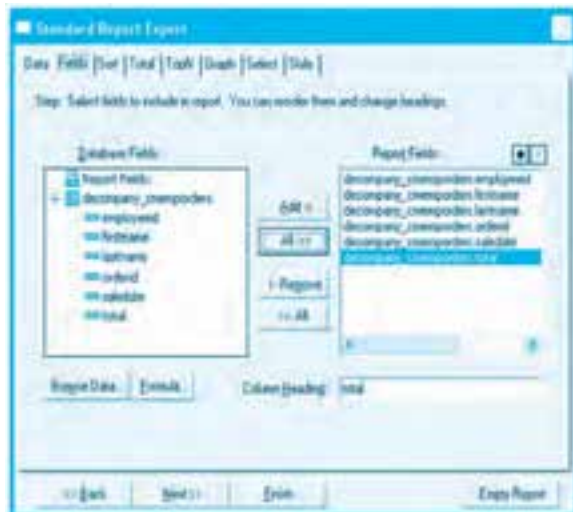


شکل ۶-۷۲

در این کادرمحاوره ساختار درختی DE و Connection را باز کرده و شیء Command موردنظر خود را انتخاب و بعد روی دکمه فرمان OK کلیک کنید. در این مرحله دوباره به کادرمحاوره Standard Report Expert باز می‌گردید. در این کادرمحاوره روی دکمه فرمان Next کلیک کنید (یا روی زبانه Fields کلیک کنید) تا به زبانه Fields بروید. (شکل ۶-۷۳).



شکل ۶-۷۳



شکل ۶-۷۴

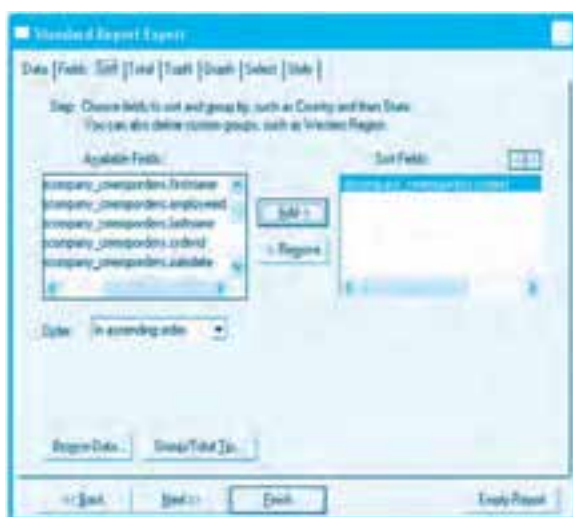
در این زبانه می‌توانید فیلدهایی را که می‌خواهید در گزارش از آن‌ها استفاده کنید، با دکمه‌های فرمان

→ Add یا →> All انتخاب کنید و از کادر لیست Database Fields در کادر لیست Report Fields قرار دهید یا با استفاده از دکمه‌های فرمان Remove- < و <<-Remove All آن‌ها را از حالت انتخاب خارج کرده و از کادر لیست Report Fields حذف کنید (شکل ۶-۷۴).

پس از انتخاب فیلدهای روی دکمه Next یا زبانه Sort کلیک کنید تا به مرحله بعد بروید. در زبانه Sort می‌توانید نحوه مرتب کردن رکوردها را در گزارش بر اساس یک یا چند ستون و به صورت صعودی یا نزولی تنظیم کنید (شکل ۶-۷۵).



شکل ۶-۷۵



شکل ۶-۷۶

برای این کار می‌توانید از کادر لیست Available Fields ستون خود را انتخاب کرده و با کلیک روی دکمه فرمان Add-> آنرا در کادر لیست Sort Fields قرار دهید (شکل ۶-۷۶).
 به طور پیش‌فرض مرتب‌سازی به صورت صعودی انجام می‌شود (شکل ۶-۷۶). اگر بخواهید به روش نزولی رکوردها مرتب شوند در این زبانه با استفاده از کادر لیست ترکیبی بازشو Order گزینه in descending order را انتخاب کنید پس از انجام تنظیم در این زبانه روی دکمه فرمان Next یا زبانه Total کلیک کنید؛ در این صورت زبانه Total مانند شکل ۶-۷۷ فعال می‌شود.

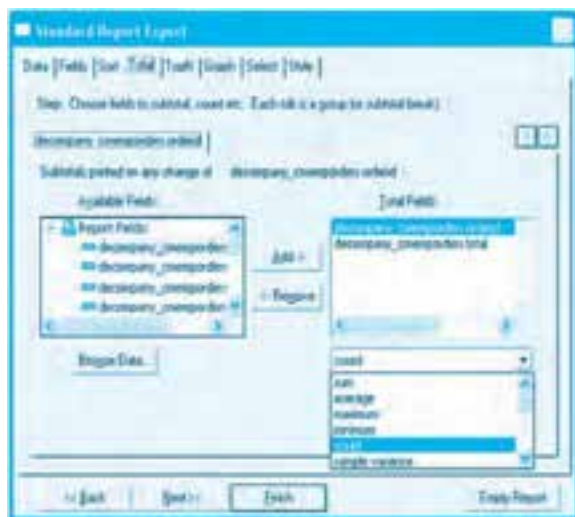


شکل ۶-۷۷

در این زبانه می‌توانید از توابعی مانند Sum، Count، Average و مانند آن‌ها که قبلاً در محیط Data Report بررسی شده‌اند استفاده کرده و فیلدهای محاسباتی خود را ایجاد کنید. به طور پیش‌فرض ستون‌هایی که امکان انجام عملیات محاسباتی روی آن‌ها وجود دارد در کادر لیست Total Field نمایش داده می‌شوند. در صورت نیاز می‌توانید سایر ستون‌ها را از کادر لیست Available Fields انتخاب کنید و با کلیک روی دکمه فرمان Add-> آنرا در کادر لیست Total Fields قرار دهید یا با استفاده از دکمه فرمان Remove-> ستون‌های انتخاب شده را از کادر لیست Total Fields خارج کنید. وقتی یک ستون در کادر لیست Total Fields قرار می‌گیرد، می‌توانید پس از انتخاب آن با استفاده از کادر لیست ترکیبی بازشویی که در زیر کادر لیست Total Fields قرار دارد نوع عملیاتی را که روی ستون انجام می‌شود، انتخاب کنید (شکل ۶-۷۸).

بررسی زبانه‌های بعدی در این کادرمحاوره از حوصله این کتاب خارج است، بنابراین در این مرحله یا هر مرحله‌ای که مایل هستید می‌توانید با کلیک روی دکمه فرمان Finish به عملیات خاتمه دهید یا با کلیک روی دکمه فرمان Back به مرحله قبل (زبانه قبلی) بازگردید. پس از کلیک روی دکمه

فرمان Finish کادرمحاوره دیگری با عنوان Crystal Report Expert مانند شکل ۶-۷۹ نمایش داده می‌شود. در این کادرمحاوره می‌توانید روی دکمه فرمان OK کلیک کنید تا یک فرم به طور خودکار آماده شود.

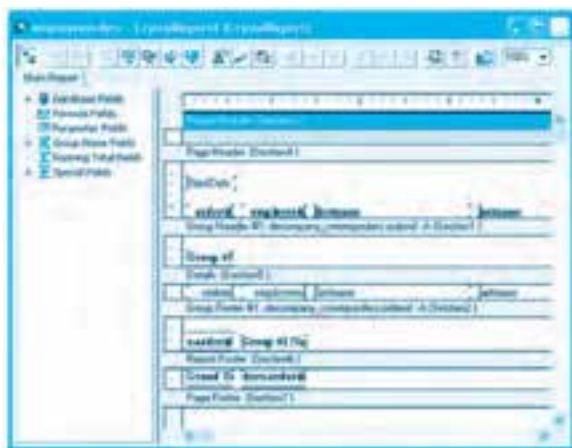


شکل ۶-۷۸



شکل ۶-۷۹

در این فرم یک کنترل از نوع CRViewer1 که توانایی نمایش گزارش‌های تولید شده با Crystal Report را دارد استفاده می‌شود و کدهای لازم نیز برای این فرم به طور خودکار تولید می‌شود. به علاوه فرم به عنوان فرم اصلی در هنگام اجرای برنامه نمایش داده شده و پنجره طراحی گزارش نیز مانند شکل ۸۰-۶ نمایش داده می‌شود.



شکل ۸۰-۶

اما اگر در کادرمحاوره Crystal Report Expert روی دکمه انتخاب No در اولین سؤال کلیک کرده و سپس روی دکمه فرمان OK کلیک کنید فقط پنجره طراحی گزارش مانند شکل ۸۰-۶ نمایش داده می‌شود. پس از خاتمه عملیات می‌توانید شکل ظاهر گزارش را به دلخواه خود تنظیم کنید و تغییر دهید.

نکته: نرم‌افزار Crystal Report از امکانات و قابلیت‌های بسیاری برخوردار است که متأسفانه امکان بررسی آن‌ها در این کتاب درسی وجود ندارد. علاقه‌مندان می‌توانند به کتاب‌هایی که به طور اختصاصی در این رابطه تهیه شده‌اند، مراجعه نمایند.

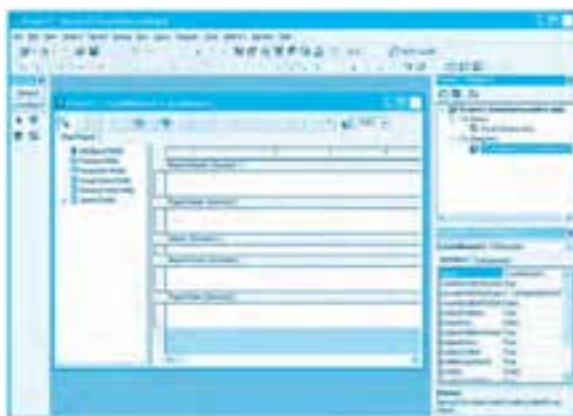
توجه داشته باشید که استفاده از Wizardهای ارائه شده همواره نمی‌توانند نیازها را برآورده کنند یا ممکن است از نظر شکل ظاهری، گزارش مناسبی تهیه نشود. در این صورت می‌توانید در اولین کادرمحاوره یعنی Report Gallery روی دکمه Empty Report کلیک کنید (شکل ۷۰-۶) تا مستقیماً کادرمحاوره Crystal Report Expert نمایش داده شود و با کلیک روی دکمه فرمان OK در این کادرمحاوره یک گزارش خالی در پنجره طراحی در اختیارتان قرار گیرد. در این حالت لازم است با داشتن مهارت کافی، یک گزارش مناسب را طراحی کنید. گزارش‌هایی که از نوع خالی ایجاد می‌شوند مانند گزارش‌های از نوع Data Report دارای پنج بخش می‌باشند که قبلاً بررسی شده‌اند و در این جا نیز دقیقاً همان کاربرد را دارند.

مثال ۱۵: می‌خواهیم پروژه‌ای طراحی کنیم تا بتوان اطلاعات کلیه سفارشات در جدول orders را



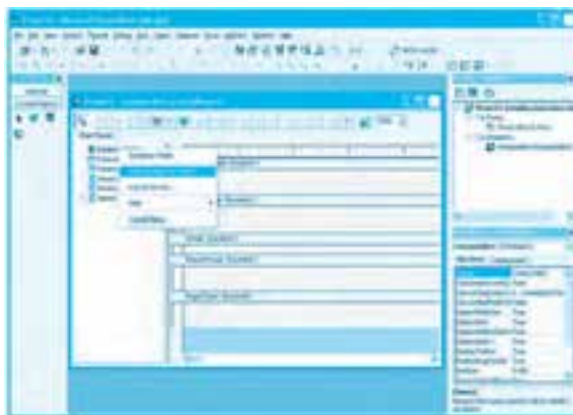
همراه با مشخصات کارمندی که هر سفارش را انجام داده است، به صورت یک گزارش مشاهده کرد. برای انجام این مثال این عملیات را انجام دهید:

- ۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE ایجاد کنید.
- ۲- در این مرحله یک گزارش از نوع Crystal Report به پروژه اضافه و در کادرمحاوره Report Gallery روی دکمه Empty Report کلیک کنید و بعد در کادرمحاوره Crystal Report Expert در اولین سؤال پاسخ No را انتخاب کنید. در این صورت یک گزارش جدید ایجاد شده و پنجره طراحی گزارش مانند شکل ۸۱-۶ نمایش داده می‌شود.



شکل ۸۱-۶

- ۳- اکنون نام گزارش را در پنجره خواص روی عبارت cremporders تنظیم و خاصیت Display Grid آن را روی مقدار True تنظیم کنید تا نقاط راهنما در پنجره طراحی گزارش نمایش داده شوند. این نقاط راهنما در زمان قرار دادن و تنظیم ابعاد اجزای گزارش مانند کنترل‌ها و ستون‌های داده و کنترل‌های نمایشی خط و کادر، عملیات را آسان و سریع می‌کند.



شکل ۸۲-۶

۴- در این مرحله در سمت چپ پنجره طراحی گزارش روی آیکن Database Fields کلیک راست کرده و بعد روی گزینه ... Add Database to Report کلیک کنید (شکل ۶-۸۲).



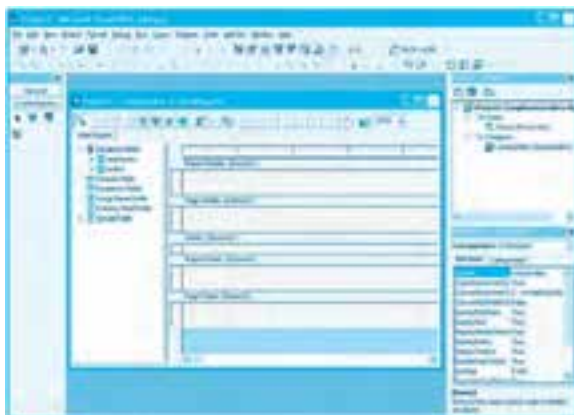
شکل ۶-۸۳

۵- در کادرمحاوره Choose Database File فایل پایگاه داده خود را (company.mdb) پیدا کرده و پس از انتخاب آن روی دکمه فرمان Open کلیک کنید تا کادرمحاوره Select Tables مانند شکل ۶-۸۳ نمایش داده شود. در این کادرمحاوره روی دکمه فرمان Select All کلیک کنید تا جدول‌های employees و orders هر دو انتخاب شوند، سپس روی دکمه فرمان OK کلیک کنید، اکنون پنجره Visual Linking Expert نمایش داده می‌شود. در این پنجره جدول‌های انتخاب شده در مرحله قبل همراه با رابطه‌ای که با یکدیگر دارند به صورت گرافیکی نمایش داده می‌شوند (شکل ۶-۸۴).



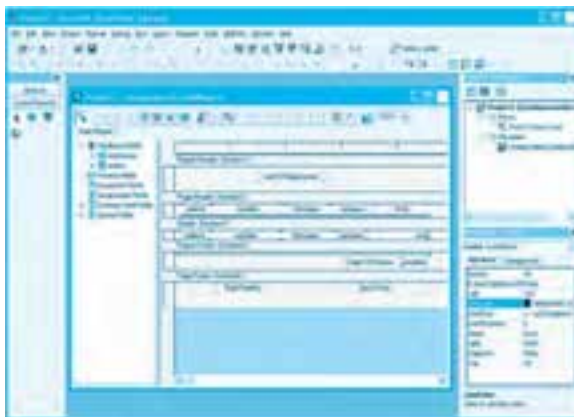
شکل ۶-۸۴

روی دکمه فرمان OK کلیک کنید تا جدول‌های انتخابی به ساختار درختی Database Fields در پنجره طراحی گزارش اضافه شوند (شکل ۸۵-۶).



شکل ۸۵-۶

۶- در این مرحله می‌توانید ساختار درختی جدول‌ها را در Database Fields باز کرده و ستون‌های موردنظر را با عمل کشیدن در بخش جزئیات گزارش قرار دهید و گزارش را مانند شکل ۸۶-۶ تنظیم کنید.




شکل ۸۶-۶

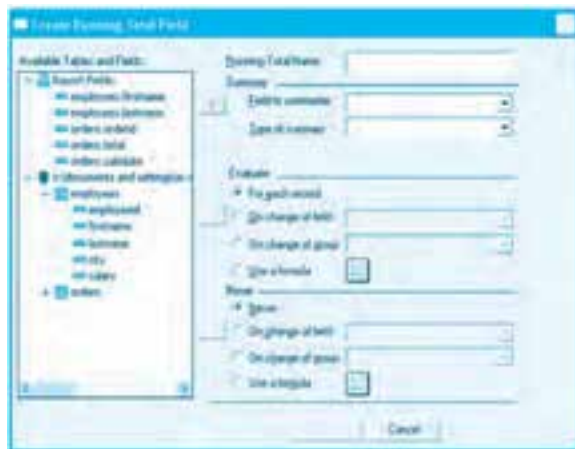
توجه داشته باشید که در هنگام قرار دادن هر ستون در بخش جزئیات به‌طور خودکار یک کنترل Text Object نیز برای عنوان آن ستون در سرصفحه گزارش بالای ستون مربوطه قرار می‌گیرد. وقتی در پنجره طراحی گزارش قرار دارید، در جعبه ابزار ویژوال بیسیک زبانه دیگری با نام Crystal Report قرار دارد که کنترل‌های موردنیاز برای طراحی گزارش را ارائه می‌کند؛ با این کنترل‌ها می‌توان انواع خطوط، کادر و متن

را روی گزارش قرار داد (شکل ۶-۸۷).



شکل ۶-۸۷

۷- در این مرحله می‌خواهیم یک ستون محاسباتی برای شمارش تعداد رکوردها ایجاد کنیم تا در پاصفحه گزارش نمایش داده شود. برای این کار در سمت چپ پنجره طراحی گزارش روی شاخه Running Total Fields کلیک راست کرده و در منویی که نمایش داده می‌شود روی گزینه New... کلیک کنید. کادرمحاوره Create Running Total Field مانند شکل ۶-۸۸ نمایش داده می‌شود. در این کادرمحاوره در کادر متن Running Total Name برای نام ستون محاسباتی عبارت countorders را تایپ کنید، سپس در ساختار درختی سمت چپ این کادرمحاوره و در شاخه Report Fields روی گزینه orders.orderid و بعد روی دکمه فرمان  کلیک کنید.



شکل ۶-۸۸

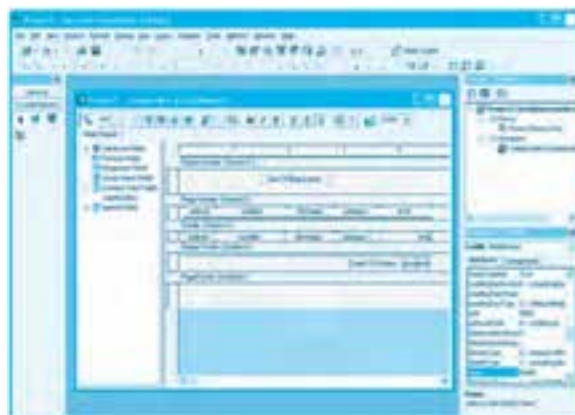
در مرحله بعد به کادر لیست ترکیبی بازشو Type of summery بروید و در آن تابع محاسباتی count را برای

شمارش تعداد رکوردها انتخاب کنید (شکل ۸۹-۶) در پایان روی دکمه فرمان OK کلیک کنید.



شکل ۸۹-۶

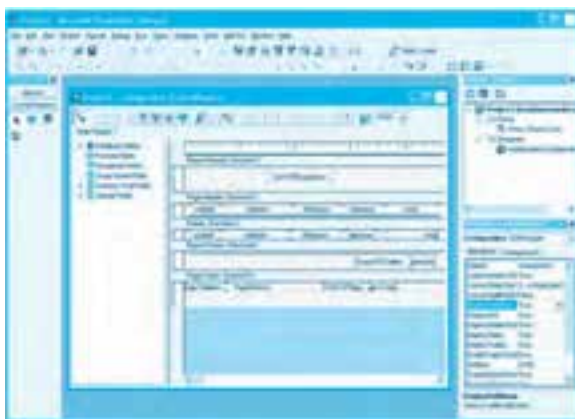
در این مرحله ستون محاسباتی در ساختار درختی Running Total Field در پنجره طراحی گزارش نمایش داده می‌شود. اکنون ستون محاسباتی countorders را با عمل کشیدن روی بخش پاصفحه گزارش و در پایین ستون total قرار دهید (شکل ۹۰-۶)، سپس یک کنترل متن (Text Object) نیز مانند شکل ۹۰-۶ در سمت چپ آن قرار دهید.



شکل ۹۰-۶

۸- در آخرین مرحله از طراحی گزارش در سمت چپ پنجره طراحی گزارش ساختار درختی شاخه Special Fields را باز کرده و در این ساختار درختی باز شده روی گزینه Page Number کلیک کنید و با عمل کشیدن آن را در بخش پاصفحه و در سمت چپ آن قرار دهید (شکل ۹۱-۶) تا شماره صفحه در پاصفحه هر صفحه از گزارش

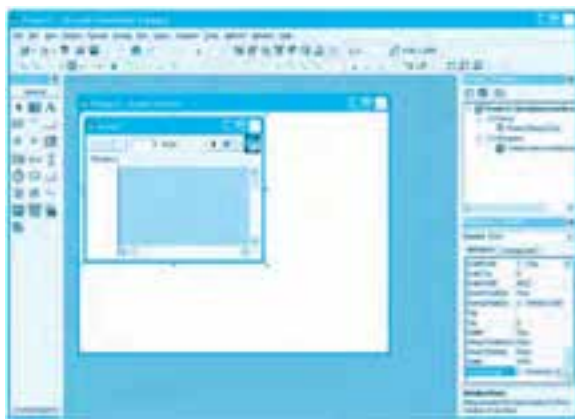
نمایش داده شود. سپس با همین روش یک نمونه از Total Page count روی پاصفحه و در سمت راست آن قرار دهید تا تعداد صفحه‌های گزارش را نمایش دهد و در پایان مانند شکل ۹۱-۶ دو کنترل متن دیگر با محتویات Total Pages: و Page Number: در سمت چپ ستون‌های شماره صفحه و تعداد صفحه‌ها قرار دهید.



شکل ۹۱-۶

نکته: برای آن‌که بتوانید در کنترل‌های متن (Text Object) عبارتی را تایپ نمایید پس از انتخاب آن در پنجره طراحی گزارش روی آن دو بار کلیک کنید.

- ۹- اکنون گزارش آماده شده است و می‌توانید از آن استفاده کنید. گزارش را با نام cremporders ذخیره کنید.
- ۱۰- به پنجره طراحی فرم بروید و کنترل CRViewer را از جعبه ابزار ویژوال بیسیک روی فرم قرار دهید (شکل ۹۲-۶).



شکل ۹۲-۶

۱۱- در این مرحله رویداد Load فرم را به این صورت تنظیم کنید:

```
Private Sub Form_Load()  
    cremporders.PaperSize = crPaperA5  
    crvemporders.ReportSource = cremporders  
    crvemporders.ViewReport  
End Sub
```

 **نکته:** اگر این کنترل به طور خودکار به جعبه ابزار اضافه نشده باشد به کادرمحاوره Components بروید و گزینه Crystal Report SmartViewer را انتخاب کرده و روی دکمه فرمان OK کلیک کنید تا کنترل مربوطه به جعبه ابزار اضافه شود.

در این رویداد نام گزارش تهیه شده با Crystal Report در خاصیت ReportSource کنترل crvemporders قرار داده می‌شود تا امکان نمایش گزارش به وجود آید، سپس متد ViewReport گزارش را در کنترل crvemporders نمایش می‌دهد. در ضمن اولین دستور، اندازه کاغذ را متناسب با ابعاد گزارش روی کاغذ A5 تنظیم می‌کند.

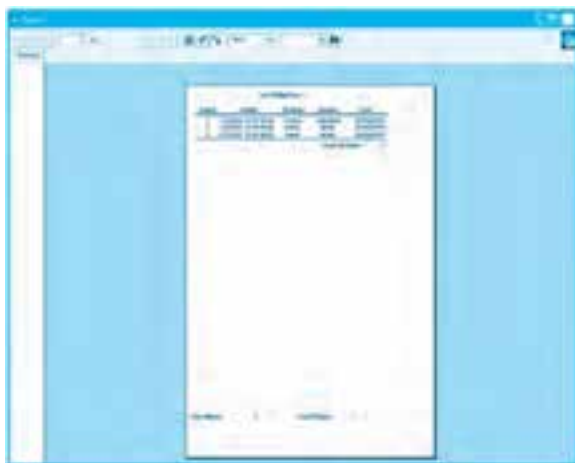
برای استفاده از ثابت crPaperAs لازم است کتابخانه Crystal Report7 ActiveX Designer Design Time Library را به وسیله گزینه References در منوی Project به پروژه اضافه کنید.

۱۲- برای آن‌که در زمان تغییر اندازه فرم، کنترل crvemporders نیز به طور مناسب تغییر کند رویداد Resize فرم را به این صورت تنظیم کنید:

```
Private Sub Form_Resize()  
    crvemporders.Top = 0  
    crvemporders.Left = 0  
    crvemporders.Height = Height  
    crvemporders.Width = Width
```

```
End Sub
```

۱۳- پروژه و فرم را با نام دلخواه ذخیره کنید. سپس پروژه را اجرا کرده و محتویات گزارش را مشاهده و بررسی کنید (شکل ۹۳-۶).



شکل ۹۳-۶

۱۴- به اجرای پروژه خاتمه داده و به پنجره ویژوال بیسیک بازگردید.

نکته: برای چاپ گزارش‌های از نوع Crystal Report می‌توانید از متد Printout شیء گزارش یا از دکمه چاپگر در نوار ابزار کنترل CRViewer استفاده کنید.

تمرین: پروژه مثال ۱۵ را به شکلی تنظیم کنید که مجموع مبلغ سفارش‌ها نیز در پاصفحه گزارش نمایش داده شود.

Learn in English

Building Data-Centric Applications with Visual Basic

In this release, Visual Basic offers a host of new and enhanced features that, together, make data access using Visual Basic easier than ever before. Visual Basic supports ActiveX Data Objects (ADO) 2.0, Microsoft's new, high-level interface to all kinds of data. You can create ADO objects at design-time using the updated Data Environment Designer. You can access the entire structure of a database on a connection through the Data View Window and the Visual Database Tools. Visual Basic also offers new and updated controls specifically for working with

data.

This section provides an overview of the following Visual Basic data access features.

- Support for ActiveX Data Objects (ADO) 2.0
- Data Environment designer
- Data View window
- Visual Database Tools integration
- New and enhanced controls

ADO 2.0

The new release of ADO features a simpler object model, better integration with Microsoft and non-Microsoft technologies, a common interface for both local and remote data access, remotable and disconnected recordsets, a user-accessible data binding interface, and hierarchical recordsets.

Data Environment Designer

The Data Environment designer helps you visually create and manipulate ADO connections and commands at design-time. In addition to supporting all of the functionality of the UserConnection designer, the Data Environment designer supports new features that make it easier to access your data, such as multiple Connection objects, support for OLE DB data sources, and OLE drag-and-drop support, so you can drag fields and tables from the Data Environment onto a form or the Data Report ActiveX designer.

Data View Window

The Data View window contains a list of all the database connections currently available to your project. You can see the entire structure of the database using the Data View window, which provides the means to use the Microsoft Visual Database Tools to browse and edit data from tables and views. In addition to inspecting the structure of your database, you can drag objects from the Data View onto the Data Environment for quickly creating ADO recordset commands.

Visual Database Tools Integration

The Microsoft Visual Database Tools (Database Designer and Query Designer) provide a visual interface to help you create and modify database structures (schemas) and queries. You can create Microsoft SQL Server™ and Oracle database tables, use drag-and-drop operations to create views, and automatically change column data types.

Visual Basic, Professional Edition includes a subset of the Visual Database Tools features. With Visual Basic, Professional Edition, you can create queries, edit queries, and execute stored procedures.

New and Enhanced Controls

The following controls are all designed to help you solve data access issues in your applications.

Control	Purpose
ADO Data Control	Uses ADO to create connections between data-bound controls and data providers.
DataCombo and DataList Controls	Provide a combo box and a list box that you can fill from a database field from the data control to which they are bound.
DataGrid Control	Views and edits recordsets using a spreadsheet-like interface.

واژه‌نامه

Bound	مقید کردن، محدود کردن
Connection	اتصال، وابستگی
DataBase	پایگاه داده
DataSource	منبع داده
Descending	نزولی
Designer	طراح
Disconnect	قطع کردن
Environment	محیط، اطراف
Feature	خصوصیت
Field	ستون، فیلد
Filter	تصفیه کردن، جداسازی
Functional	تابعی، وظیفه‌ای
Flexibility	انعطاف‌پذیر، قابل انعطاف
Functional	تابعی، وظیفه‌ای
Gallery	سالن، گالری
General	عمومی
Grid	نقطه راهنما
Hieranchical	سلسله مراتبی
Inspect	رسیدگی کردن
Lock	قفل
Management	مدیریت

Manipulate	اداره کردن، دستکاری کردن
Navigate	هدایت کردن، راهبری کردن
Navigation	راهبری، هدایت
Optimistic	خوش‌بینانه
Order	سفارش
Pessimistic	بدبینانه
Previous	قبل، قبلی
Provider	مهی‌اکننده، آماده‌کننده
Query	پرس‌وجو
Record	سطر، رکورد
Relational	رابطه‌ای
Remote	متحرک
Release	نگارش
Resemble	شبهت داشتن
Snapshot	تصویر فوری
Special	مخصوص، ویژه
Statement	دستور
Scheme	طرح، الگو
Structure	ساختمان، ساختار
Table	جدول
Unbound	بدون حد و مرز، آزاد
Useful	مفید

خلاصه مطالب

- به یک ویژگی مشخص در رابطه با یک موضوع خاص، فیلد می‌گویند.
- به مجموعه‌ای از چند فیلد، در رابطه با یک موضوع خاص رکورد یا سطر می‌گویند.
- یک جدول مجموعه‌ای از رکوردها و یک پایگاه داده مجموعه‌ای از جدول‌هاست.
- از کنترل Data برای انجام عملیات درج، به‌روزرسانی، حذف یا بازیابی داده‌ها و اطلاعات از پایگاه‌های داده استفاده می‌شود.
- SQL یک زبان برنامه‌نویسی است که دستورات لازم برای انجام عملیات بازیابی، درج، حذف و به‌روزرسانی داده را ارائه می‌کند.
- شکل کلی دستور Select به این صورت است:
نام جدول From نام ستون(ها) Select
- DAO اولین تکنولوژی در ویژوال بیسیک برای انجام عملیات روی پایگاه داده است.
- شیء Rcordset شامل مجموعه‌ای از رکوردها است که به وسیله کنترل‌های داده مانند ADO یا کنترل Data بازیابی می‌شوند.
- در شیء Recordset از متدهای MoveFirst, MoveLast, MoveNext, MovePrevious برای حرکت به اولین رکورد، آخرین رکورد، رکورد بعدی و رکورد قبلی استفاده می‌شود.
- در شیء Recordset از متدهای AddNew, Delete و Update برای درج، حذف و به‌روزرسانی رکوردها استفاده می‌شود.
- خاصیت‌های EOF و BOF در شیء Recordset به ترتیب برای تشخیص رسیدن به انتها و ابتدای رکوردها استفاده می‌شود.
- کنترل‌ها را می‌توان به دو روش Bound و UnBound به کنترل‌های داده متصل نمود.
- کنترل ADO یک کنترل پیشرفته در ویژوال بیسیک است که برای انجام عملیات روی پایگاه‌های داده استفاده می‌شود و از تکنولوژی OLE DB استفاده می‌کند.
- تکنولوژی OLEDB امکان اتصال و ارتباط با انواع پایگاه‌های محلی و راه دور با انواع مختلف را فراهم می‌کند.
- خاصیت ConnectionString پارامترهای اتصال به پایگاه داده را دریافت و نگهداری می‌کند.
- خاصیت RecordSource در کنترل ADO از نوع رشته‌ای است و دستور موردنظر را برای اجرا روی پایگاه داده و جدول‌های آن مشخص می‌کند.
- خاصیت CursorType در کنترل ADO نحوه حرکت روی رکوردها و انجام عملیات درج و به‌روزرسانی و حذف رکوردهای بازیابی شده را مشخص می‌کند.
- خاصیت LockType نحوه مدیریت رکوردها را در زمانی که چند کاربر به‌طور همزمان عملیاتی را روی پایگاه داده‌ها

انجام می‌دهند، مشخص می‌کند.

- خاصیت CursorLocation وقتی شیء Recordset پایگاه داده‌ها در همان محلی که برنامه اجرا می‌شود قرار دارند باید روی مقدار aduseClient تنظیم شود.
- خاصیت Mode وقتی اتصال به پایگاه داده‌ها در همان محلی است که برنامه اجرا می‌شود باید روی مقدار admodeUnknown تنظیم شود.
- از متد Find در شیء Recordset برای پیدا کردن رکوردها با شرایط مشخص استفاده می‌شود.
- از خاصیت Filter در شیء Recordset برای جداسازی رکوردها با شرایط مشخص استفاده می‌شود و از نوع رشته‌ای است.
- کنترل‌های DataList و DataCombo می‌توانند محتویات یک ستون از رکوردها را به طور همزمان نمایش دهند.
- برای اتصال کنترل‌های DataList و DataCombo از خاصیت‌های DataSource و RowSource استفاده می‌شود.
- برای اتصال کنترل‌های ساده مانند کادر متن از خاصیت‌های DataSource و DataField استفاده می‌شود.
- کنترل DataGrid می‌تواند مجموعه‌ای از رکوردها و فیلدها را به طور همزمان نمایش دهد.
- با استفاده از کنترل DataGrid می‌توان عملیات درج، حذف و به‌روزرسانی روی پایگاه‌های داده را انجام داد.
- کتابخانه شیء ADODB امکان استفاده مستقیم از تکنولوژی OLEDB را فراهم می‌کند.
- کتابخانه شیء ADODB سه کلاس اصلی Connection، Command و Recordset را برای اتصال و انجام عملیات روی پایگاه‌های داده ارائه می‌کند.
- کلاس Connection مجموعه‌ای از خاصیت‌ها، متدها و رویدادها را برای اتصال به انواع پایگاه داده ارائه می‌کند.
- کلاس Command مجموعه‌ای از خاصیت‌ها، متدها و رویدادها را برای اجرای دستورات SQL روی پایگاه‌های داده و جدول‌های موجود در آن‌ها ارائه می‌کند.
- کلاس Recordset مجموعه‌ای از خاصیت‌ها، متدها و رویدادها را برای دسترسی و مدیریت رکوردهای بازیابی شده ارائه می‌کند.
- Data Environment یک محیط گرافیکی برای ایجاد اشیای Connection و Command و تنظیم خاصیت‌های آن‌ها است.
- Data Report یک محیط گرافیکی برای طراحی و ساخت انواع گزارش‌هاست.
- هر گزارش از پنج بخش اصلی تشکیل می‌شود: سرصفحه و پاصفحه گزارش، سرصفحه و پاصفحه و بخش جزئیات.
- برای اتصال یک گزارش از نوع Data Report به یک Data Environment از خاصیت‌های DataSource و DataMember گزارش استفاده می‌شود.
- نرم‌افزار Crystal Report امکان ایجاد انواع گزارش‌های متفاوت را در محیط ویژوال بیسیک فراهم می‌کند.
- گزارش‌های از نوع DataReport را می‌توان با متدهای Show و PrintReport نمایش داد یا چاپ کرد.
- با استفاده از کنترل CRViewer می‌توان گزارش‌های از نوع Crystal Report را روی فرم‌ها نمایش داد.
- گزارش‌های از نوع Crystal Report را می‌توان با استفاده از متدهای ViewReport و Printout نمایش داد یا چاپ کرد.

آزمون نظری

۱- کدام خاصیت در کنترل داده امکان ایجاد ارتباط کنترل با جدول در یک پایگاه داده را فراهم می‌کند؟

الف - RecordSource ب - DataField

ج - DataSource د - DataMember

۲- کدام خاصیت در کنترل کادر متن می‌تواند بین کنترل و یک فیلد ارتباط برقرار کند؟

الف - RecordSource ب - DataField

ج - DataSource د - DataMember

۳- کوچک‌ترین واحد ذخیره‌سازی داده‌ها و اطلاعات در یک پایگاه داده است.

الف - فیلد ب - رکورد ج - سطر د - جدول

۴- متد در شیء DataReport می‌تواند گزارش را روی صفحه نمایش ارائه کند.

الف - View ب - ViewReport ج - Display د - Show

۵- کنترل ADO از کدام تکنولوژی برای ایجاد ارتباط با پایگاه‌های داده و انجام عملیات روی آن‌ها استفاده می‌کند؟

الف - DAO ب - RDO ج - OLE DB د - ADO DC

۶- کدام خاصیت در شیء Command می‌تواند دستور SQL را برای اجرا روی پایگاه داده مشخص کند؟

الف - CommandType ب - CommandText

ج - ConnectionString د - CursorType

۷- کدام خاصیت در Data Report می‌تواند برای استفاده از شیء Command موجود در محیط Data Environment به کار رود؟

الف - DataMember ب - DataSource ج - DataField د - Recordset

۸- کدام گزینه برای جستجوی داده‌ها در مجموعه رکوردهای بازیابی شده Recordset مناسب است؟

الف - Find ب - Filter ج - Select د - Search

۹- کدام خاصیت در کنترل ADO نحوه حرکت روی رکوردها را مشخص می‌کند؟

الف - CursorType ب - Mode ج - CursorLocation د - LockType

۱۰- کدام بخش در یک گزارش برای نمایش رکوردها مناسب‌تر است؟

ب- پاصفحه گزارش

الف- سرصفحه گزارش

د- سرصفحه و پاصفحه

ج- جزئیات

11- The designer Lets you Create and use ADO Connections and Command at design-time.

a- Data Environment

b- Data Combo

c- Data List

d- Data Grid

۱۲- مفاهیم فیلد، رکورد، جدول و پایگاه داده‌ها را توضیح دهید.

۱۳- دلایل استفاده از سیستم‌های مدیریت پایگاه داده رابطه‌ای را بیان کنید.

۱۴- دستور Select و نحوه استفاده از آن را با ذکر مثال توضیح دهید.

۱۵- تکنولوژی DAO را توضیح دهید.

۱۶- کاربرد کنترل Data را با ذکر مثال بیان کنید و خاصیت‌ها و متدهای آن را توضیح دهید.

۱۷- کاربرد کنترل ADO را همراه با متدها و خاصیت‌های آن توضیح دهید.

۱۸- تکنولوژی OLEDB را توضیح دهید و اجزای اصلی تشکیل دهنده آن را همراه با وظایف هر یک بیان کنید.

۱۹- شیء Recordset را توضیح دهید و کاربرد آن را به همراه خاصیت‌ها و متدهای مهم آن بیان کنید.

۲۰- نحوه اتصال به روش‌های Bound و UnBound را توضیح دهید و تفاوت آن‌ها را با یکدیگر بیان کنید.

۲۱- کاربرد کنترل‌های DataList و DataCombo را بیان کنید و نحوه اتصال آن‌ها را به کنترل‌های داده توضیح دهید.

۲۲- کاربرد یک Data Environment را بیان کرده و نحوه ایجاد و استفاده از آن را توضیح دهید.

۲۳- مفهوم گزارش و نحوه ایجاد آن را با Data Report با ذکر مثال توضیح دهید و روش نمایش و چاپ آن را بیان کنید.

۲۴- نحوه ایجاد گزارش با Crystal Report را به طور کامل توضیح دهید.

آزمون عملی

۱- یک پروژه به صورت MDI را همراه با یک پایگاه داده به شکلی ایجاد کنید که به وسیله آن بتوان اطلاعات یک کتابخانه را ذخیره، مدیریت و بازیابی نمود و همچنین قابلیت‌های زیر را داشته باشد:

الف- اطلاعات مربوط به کتاب‌ها و اعضا در جدول‌های مستقل نگهداری شود.

ب- اطلاعات مربوط به امانت دادن کتاب‌ها در یک جدول ذخیره شود.

ج- امکان جستجوی کتاب‌ها بر اساس موضوع، عنوان، مؤلف با ناشر وجود داشته باشد.

د- امکان جستجوی اعضای کتابخانه بر اساس کد عضویت یا نام و نام خانوادگی وی وجود داشته باشد.

ه- امکان مشاهده اطلاعات کلیه کتاب‌ها یا اعضا وجود داشته باشد.

و- بتوان یک گزارش از کتاب‌هایی که اعضا به امانت برده‌اند همراه با مشخصات عضوی که کتاب را امانت گرفته مشاهده نمود یا آن را چاپ کرد.

ز- بتوان گزارشی از مشخصات کلیه اعضای کتابخانه تهیه کرد به شکلی که در انتهای گزارش تعداد کل اعضا نیز محاسبه و نمایش داده شود.

۲- پروژه‌ای با این شرایط طراحی کنید که به وسیله یک پایگاه داده بتوان مکالمات تلفن‌های ثابت را برای هر مشترک ذخیره، نگهداری و بازیابی کرد.

الف- اطلاعات مشترکین در یک جدول ذخیره شود.

ب- اطلاعات مربوط به هر مکالمه شامل شماره تلفن مقصد، تاریخ و زمان شروع، تاریخ و زمان پایان مکالمه در یک جدول ذخیره شود.

ج- امکان جستجوی یک مشترک بر اساس شماره تلفن یا نام و نام خانوادگی وی وجود داشته باشد.

د- امکان جستجوی مشترکین با یک پیش شماره دلخواه، وجود داشته باشد.

ه- بتوان اطلاعات مشترکین را به صورت یک گزارش مشاهده یا چاپ کرد.

و- بتوان گزارشی از کلیه مکالمه‌ها همراه با مشخصات مشترکین هر یک را مشاهده یا چاپ کرد.

هدف جزئی

توانایی ارتباط با نرم افزارهای Office رویه های API و تهیه برنامه نصب

زمان (ساعت)	
عملی	نظری
۹	۴

هدف های رفتاری

پس از مطالعه این واحد کار از فراگیر انتظار می رود که:

- ۱- نحوه ارتباط با برنامه های Office را بداند و بتواند از این برنامه ها در پروژه های خود استفاده کند.
- ۲- مفهوم API را بداند و مزایای استفاده از آن را بیان کند.
- ۳- نحوه استفاده از رویه های API را بداند.
- ۴- نحوه ساخت و ایجاد برنامه نصب را بداند.

کلیات

زبان برنامه‌نویسی ویژوال بیسیک ویژگی‌های مختلفی برای استفاده از امکانات سایر نرم‌افزارها یا سیستم‌های عامل در پروژه‌هایی که به وسیله آن ایجاد می‌شود، دارد. برای استفاده از این ویژگی‌ها لازم است با مفاهیم موجود در نرم‌افزارهایی که در پروژه از آن‌ها استفاده می‌شود آشنایی کافی وجود داشته باشد. به علاوه در ویژوال بیسیک به راحتی می‌توان از پروژه‌های ایجاد شده برنامه نصب مناسب با ویژگی‌های پروژه تولید کرد. این واحد کار به بررسی این ویژگی‌ها در زبان برنامه‌نویسی ویژوال بیسیک می‌پردازد.

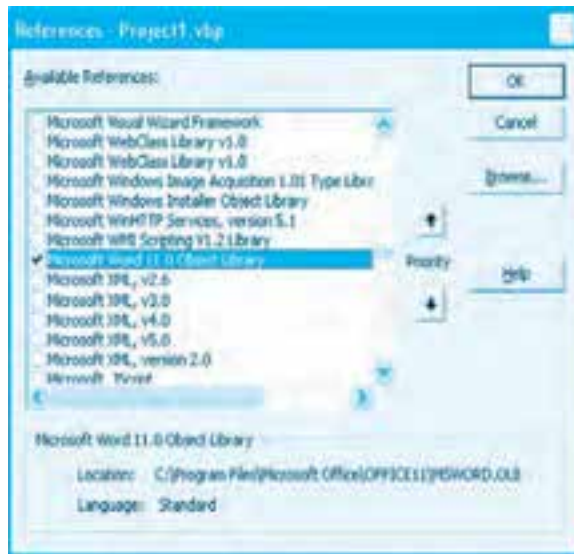
۱-۷ ارتباط ویژوال بیسیک با برنامه‌های Office

یکی دیگر از ویژگی‌ها در زبان ویژوال بیسیک استفاده از امکانات سایر برنامه‌ها در آن است. به عنوان مثال می‌توان به ایجاد یک صفحه گسترده مانند سند‌های Excel و انواع اشیایی که در آن استفاده می‌شوند مانند Workbook، Worksheet، Chart، Cell و غیره یا ایجاد یک سند Word و انواع اشیایی که در آن استفاده می‌شوند مانند پاراگراف، جمله و غیره اشاره کرد. برای ایجاد ارتباط با سایر برنامه‌ها از دو جزء استفاده می‌شود: یک جزء سرویس‌دهنده است که شامل اجزایی است که باید استفاده شوند و جزء دیگر سرویس‌گیرنده است که منظور همان بخشی است که برنامه با آن کار می‌کند. بعضی از برنامه‌ها مانند Access، Excel، Word و PowerPoint می‌توانند به صورت سرویس‌دهنده و سرویس‌گیرنده عمل کنند. در اولین مرحله لازم است از کلاس مربوطه یک شیء در حافظه ایجاد شود، سپس با استفاده از متد و خاصیت‌های شیء به وجود آمده عملیات مورد نظر انجام شود. باید توجه داشت که قبل از آن که بتوان شیء را از کلاس مربوطه ایجاد کرد فایل کتابخانه‌ای آن به پروژه اضافه شود.



شکل ۱-۷

نکته: قبل از ایجاد شیء، لازم است فایل کتابخانه‌ای مربوط به Excel و Word به وسیله گزینه References از منوی Project در پنجره ویژوال بیسیک به پروژه اضافه شود (شکل‌های ۷-۱ و ۷-۲).



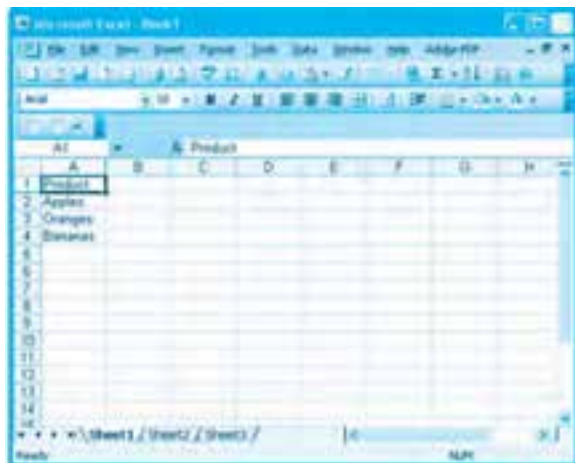
شکل ۷-۲

به عنوان مثال این دستورات یک سند از نوع Excel را باز می‌کند و پس از ایجاد یک Workbook عبارتهایی را داخل خانه‌های A1 تا A4 می‌نویسد.

```
Dim xlapp As Excel.Application
Set xlapp = New Excel.Application
With xlapp
    .Visible = True
    .Workbooks.Add
    .Range("A1").Value = "Product"
    .Range("A2").Value = "Apples"
    .Range("A3").Value = "Oranges"
    .Range("A4").Value = "Bananas"
End With
```

Set xlapp = Nothing

این دستورات ابتدا دو دستور اول یک شیء از کلاس Application که در کتابخانه Excel قرارداد ایجاد می‌کنند، سپس خاصیت Visible این شیء (xlapp) روی مقدار True تنظیم می‌شود تا بتوان پنجره برنامه Excel را در زمان اجرا مشاهده کرد. سپس با استفاده از متد Add در خاصیت Workbook یک کارپوشه ایجاد می‌شود. در دستورات بعدی با استفاده از خاصیت Range داده‌ها با توجه به آدرس آن‌ها که در خاصیت Range مشخص می‌شود، قرار می‌گیرند. در این دستورات برای آن‌که دسترسی به خاصیت‌های موردنظر آسان‌تر شود، از یک دستور With استفاده شده است. در پایان نیز با تنظیم مقدار شیء xlapp روی کلمه کلیدی Nothing شیء از حافظه خارج شده و منابعی که به وسیله آن استفاده شده‌اند، آزاد می‌شوند. اجرای این دستورات یک سند Excel مانند شکل ۳-۷ را ایجاد می‌کند.



شکل ۳-۷

Dim xlapp As Excel.Application

Set xlapp = New Excel.Application

With xlapp

.Visible = True

.Workbooks.Add

.Range("A1").Value = "Product"

.Range("A1").Font.Bold = True

.Range("A1").Font.Size = 10

.Range("A1").HorizontalAlignment = xlCenter

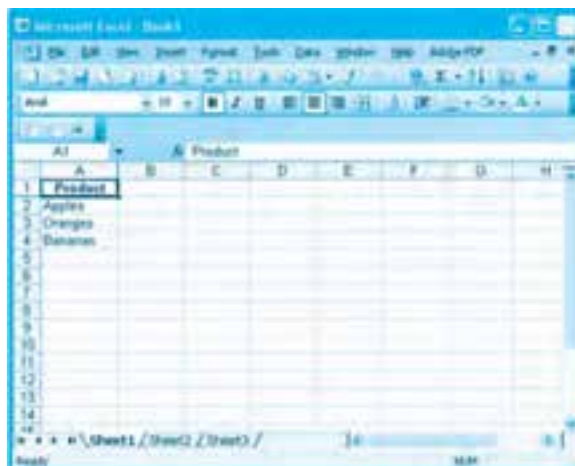
```

.Range("A1").ColumnWidth = 10
.Range("A2").Value = "Apples"
.Range("A3").Value = "Oranges"
.Range("A4").Value = "Bananas"
.ActiveSheet.PrintOut
.ActiveWorkbook.Close True, "c:\mytest"
.Quit
    
```

End With

Set xlapp = Nothing

به عنوان مثال این دستورات نیز جدولی مشابه جدول قبل در محیط Excel ایجاد می‌کنند با این تفاوت که عبارت Product جدول با اندازه ۱۰ و به صورت Bold در مرکز خانه A1 (از نظر افقی) قرار می‌گیرد. به علاوه عرض خانه A1 با استفاده از خاصیت ColumnWidth روی مقدار ۱۰ تنظیم می‌شود. در ضمن پس از درج داده‌ها، کاربرگ جاری (ActiveSheet) با استفاده از متد PrintOut برای چاپ به چاپگر ارسال می‌شود، سپس سند ایجاد شده با استفاده از متد Close بسته می‌شود و قبل از بسته شدن در فایل mytest.xls ذخیره می‌شود. اولین آرگومان در این متد از نوع منطقی است. اگر مقدار این آرگومان روی False تنظیم شود سند بدون ذخیره‌سازی بسته می‌شود و در پایان نیز با استفاده از متد Quit پنجره برنامه Excel بسته می‌شود. نتیجه اجرای این دستورات مانند شکل ۴-۷ است.

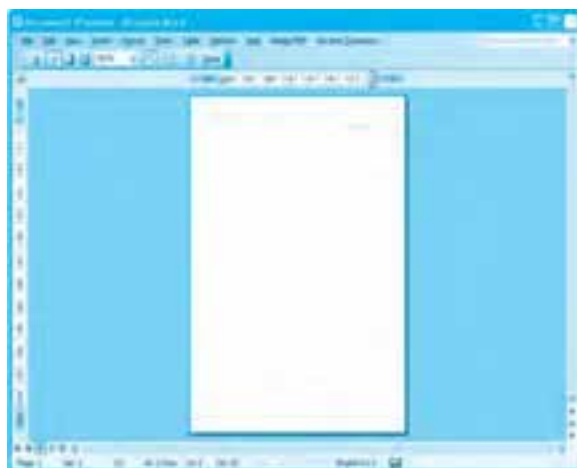


شکل ۴-۷

تمرین: پروژه‌ای طراحی کنید که نام و نام‌خانوادگی و معدل دانش‌آموزان یک کلاس را دریافت کرده و

در یک سند از نوع Excel ذخیره نموده و چاپ کند. به عنوان مثالی دیگر، این دستورات یک سند از نوع Word ایجاد کرده و سپس عبارت Visual Basic 6 را در آن می‌نویسد. دو دستور اول یک شیء از کلاس Application که در کتابخانه Word قرار دارد ایجاد می‌کنند. سپس خاصیت Visible این شیء (Wd doc) روی مقدار True تنظیم می‌شود تا بتوان پنجره برنامه Word را در زمان اجرا مشاهده کرد. سپس با استفاده از متد Add در خاصیت Documents یک سند ایجاد می‌شود و بعد با استفاده از متد TypeText عبارت Visual Basic 6 در یک سند ایجاد شده نوشته می‌شود و سپس با تنظیم خاصیت Print Preview سند روی مقدار True آن را در محیط پیش‌نمایش نشان داده و دستور بعد با متد Printout سند را چاپ می‌نماید و در پایان نیز با اجرای متد Quit پنجره برنامه Word بسته می‌شود. نتیجه اجرای این دستورات مانند شکل ۵-۷ می‌باشد.

```
Dim wddoc As Word.Application
Set wddoc = New Word.Application
With wddoc
    .Visible = True
    Documents.Add
    Selection.TypeText Text="Visual Basic 6"
    PrintPreview = True
    PrintOut
    .Quit
End With
```



شکل ۵-۷

تمرین: پروژه‌های طراحی کنید که یک متن را از کاربر دریافت کرده و در یک فایل از نوع Doc قرار دهد و چاپ کند.

۲-۷ نحوه استفاده از رویه‌های API

یکی از ویژگی‌های دیگر در زبان برنامه‌نویسی ویژوال بیسیک این است که می‌توانید عملیاتی را که به طور مستقیم در ویژوال بیسیک وجود ندارد، به وسیله توابع انجام دهید.

رویه‌های (Windows Application Programming Interface) API به صورت گروهی از فایل‌های کتابخانه‌ای (DLL) ارائه می‌شوند و فاقد هر گونه رابطه گرافیکی هستند. به طوری که شما نمی‌توانید توابع API را مانند برنامه‌های Word یا Excel مورد دسترسی قرار دهید و باید آن‌ها را به طور مستقیم فراخوانی کنید. این رویه‌ها امکان دسترسی مستقیم برنامه‌نویس را به توابع داخلی ویندوز فراهم می‌کنند. این کار مزایای متعددی دارد که عبارتند از:

- انجام اعمال و وظایفی که در ویژوال بیسیک غیرممکن است.
 - برای بهبود عملکرد برنامه احتیاجی به نوشتن کدهای اضافه و ایجاد بعضی از DLL‌های ارائه شده نخواهد بود، در نتیجه کار طراحی برنامه سریع‌تر انجام می‌شود و حجم کد را کم می‌کند.
 - به روزرسانی این بخش‌ها می‌تواند مستقل از برنامه انجام شود.
- ویندوز فایل‌های کتابخانه‌ای (DLL) زیادی ارائه می‌کند. سه DLL به عنوان کتابخانه‌های اصلی که بیشترین رویه‌های API را ارائه می‌کنند، عبارتند از User32 DLL، GDI32.DLL و Kernel32.DLL.

فایل کتابخانه‌ای User32 DLL امکان مدیریت منوها، کنترل‌ها، کادرهای محاوره و مانند آن‌ها را فراهم می‌کند. فایل کتابخانه‌ای GDI32 DLL امکان مدیریت خروجی گرافیکی، ترسیمات، قلم‌ها و مانند آن‌ها را فراهم می‌کند. فایل کتابخانه‌ای Kernel32 DLL وظایف سیستم‌عامل مانند مدیریت حافظه، منابع و سایر عملیات سیستمی پایین را ارائه می‌کند.

قبل از استفاده از فایل‌های کتابخانه‌ای و رویه‌های API موجود در آن، لازم است موقعیت و محل رویه مشخص شود و آرگومان‌های آن به شکلی که در رویه اصلی وجود دارد تعریف و مشخص شود. برای این کار نیز می‌توانید از ماژول کد استفاده کنید. نحوه تعریف یک رویه API به این صورت است:

"نام فایل کتابخانه‌ای (DLL)" Lib نام رویه Declare Function یا Private Public

برای دسترسی به تعریف یک رویه می‌توانید از برنامه API Text Viewer استفاده کنید. این برنامه اجازه می‌دهد. شما تعریف یک رویه را جستجو و پیدا کرده و آن را با عمل Copy و Paste در برنامه خود قرار دهید. برای استفاده از این برنامه به این ترتیب عمل کنید:

۱- در منوی شروع (Start) روی گزینه All Programs و بعد روی گزینه Microsoft Visual Studio 6.0

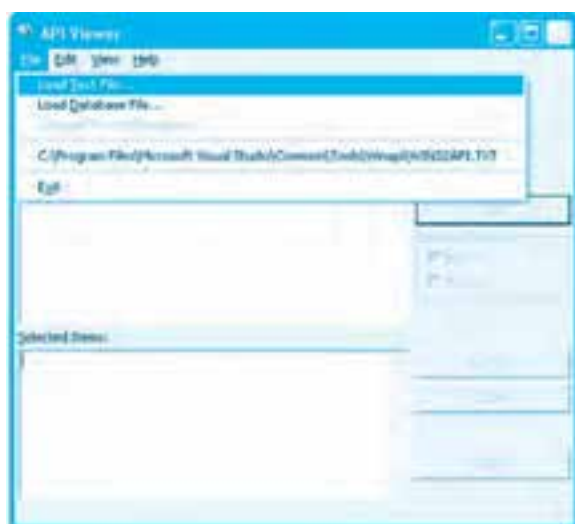
بعد گزینه Microsoft Visual Studio 6.0 Tools کلیک کنید.

۲- در منویی که نمایش داده می‌شود روی گزینه API Text Viewer کلیک کنید تا برنامه مانند شکل ۶-۷ نمایش داده شود.



شکل ۶-۷

۳- در منوی فایل برنامه API Viewer روی گزینه Load Text File... کلیک کنید (شکل ۷-۷).



شکل ۷-۷

۴- در کادرمحاوره‌ای که نمایش داده می‌شود (شکل ۸-۷) فایل متنی حاوی اطلاعات رویه‌های فایل کتابخانه‌ای موردنظر خود را روی سیستم پیدا کرده و پس از انتخاب آن روی دکمه فرمان Open کلیک کنید.

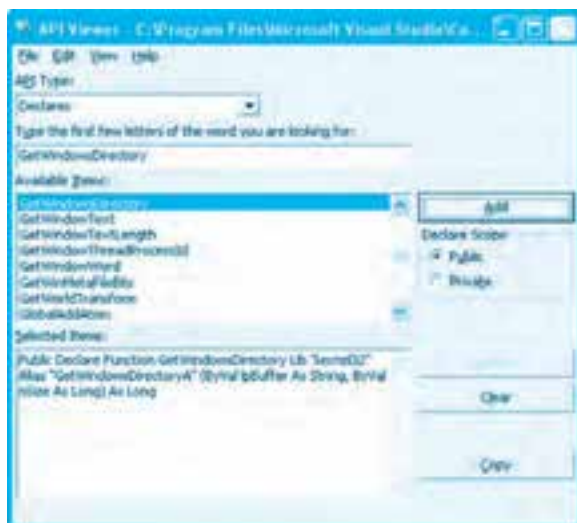


شکل ۸-۷

نکته: ویژوال بیسیک سه فایل متنی `APILOAD.TXT`، `MAPI32.TXT` و `WIN32API.TXT` را برای پیدا کردن رویه‌های API ارائه می‌کند.

نکته: تعریف اکثر رویه‌های API ویندوز در فایل `WIN32API.TXT` قرار دارد.

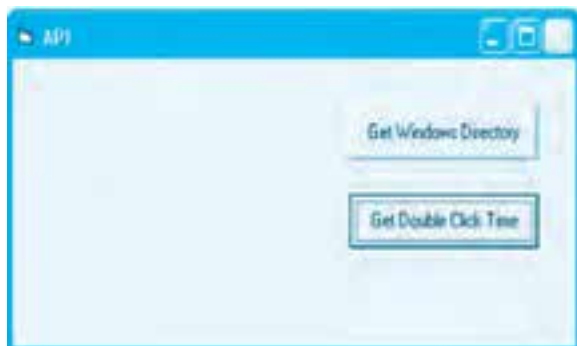
در این مرحله اسامی رویه‌های موجود در کتابخانه انتخاب شده در کادر لیست Available Items نمایش داده می‌شود. می‌توانید در این کادر لیست رویه خود را پیدا کنید یا نام آن را در کادر متن موجود در بالای این کادر لیست تایپ کرده تا رویه را سریع‌تر پیدا کنید. پس از پیدا کردن رویه خود، روی دکمه فرمان Add کلیک کنید تا تعریف رویه در کادر متن Selected Items نمایش داده شود (شکل ۹-۷). اکنون می‌توانید این تعریف را با عمل Copy (یا کلیک روی دکمه فرمان Copy در این پنجره) و Paste به برنامه خود منتقل کنید. به علاوه در پنجره برنامه API Viewer می‌توانید با دکمه‌های انتخاب موجود در کنترل قاب Declare Scope تعریف رویه را به صورت Public یا Private انتخاب کنید.



شکل ۹-۷

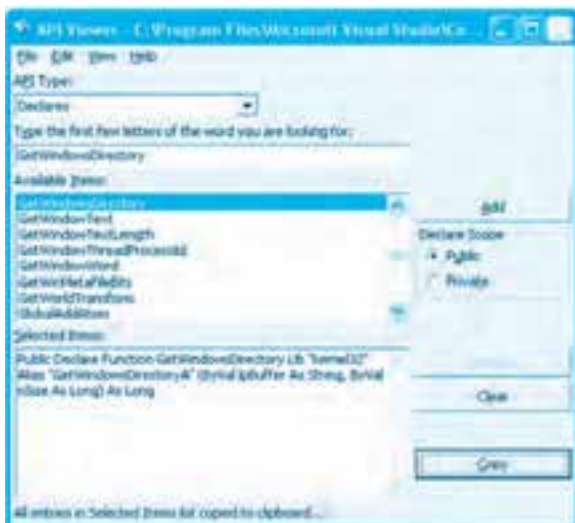
مثال ۱: می‌خواهیم برنامه‌ای داشته باشیم که به وسیله رویه‌های API بتوان مسیر نصب سیستم‌عامل ویندوز و سرعت دوبار کلیک ماوس را به دست آورد. برای انجام این مثال، عملیات بعد را به ترتیب انجام دهید:

۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE همراه با یک فرم مانند شکل ۱۰-۷ ایجاد کنید.



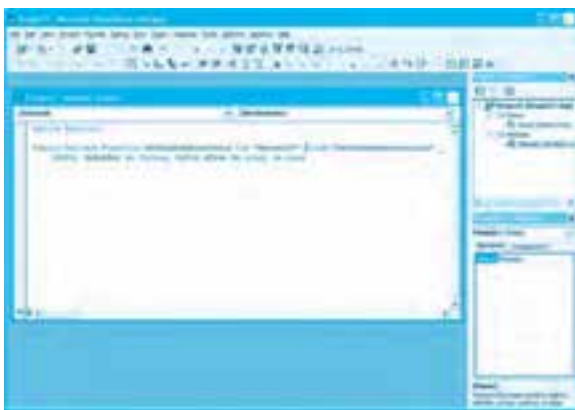
شکل ۱۰-۷

۲- به برنامه API Viewer بروید و فایل WIN32API.TXT را باز کنید، سپس عبارت GetWindowsDirectory را در کادر متن Type the first few letters of the word you are looking for تایپ کنید تا نام این رویه در کادر لیست Available Items پیدا شود. در این صورت روی دکمه فرمان Add کلیک کنید تا تعریف رویه در کادر متن Selected Items نمایش داده شود. سپس روی دکمه فرمان Copy کلیک کنید (شکل ۱۱-۷).



شکل ۷-۱۱

۳- به پنجره پروژه بروید و یک ماژول کد به پروژه اضافه کنید، سپس با عمل Paste تعریف رویه را در ماژول کد قرار دهید (شکل ۷-۱۲).



شکل ۷-۱۲

۴- در این مرحله رویداد Click دکمه فرمان Get Windows Directory را به این صورت تنظیم کنید:
 Private Sub cmdgetwindir_Click()

Dim strdir As String * 255

Dim lnglen As Long

lnglen = GetWindowsDirectory(strdir, 255)

Print Left(strdir, lnglen)

End Sub

در این رویداد ابتدا دو متغیر از نوع رشته‌ای و عدد صحیح تعریف می‌شوند. متغیر رشته‌ای برای ذخیره مسیر پوشه ویندوز و متغیر عددی برای ذخیره تعداد کاراکترهای متغیر رشته‌ای استفاده می‌شود. سپس تابع `GetWindowsDirectory` برای به دست آوردن مسیر و نام پوشه ویندوز فراخوانی می‌شود. این تابع دو آرگومان دارد؛ آرگومان اول یک متغیر رشته‌ای است که برای به دست آوردن مسیر و نام پوشه ویندوز به کار می‌رود و آرگومان دوم یک عدد صحیح است که طول رشته را مشخص می‌کند. وقتی این تابع فراخوانی شود طول رشته‌ای که مسیر و نام پوشه ویندوز را باز می‌گرداند، بازگشت می‌دهد. این مقدار نیز در متغیر `lnglen` ذخیره می‌شود.

۵- اکنون به همین صورت تعریف رویه `GetDoubleClickTime` را برای به دست آوردن سرعت انجام دو بار کلیک ماوس در ماژول کد قرار دهید و رویداد `Click` دکمه فرمان `GetDoubleClickTime` را به این صورت تنظیم کنید:

```
Private Sub cmdgetdbltime_Click()
```

```
Dim lngtime As Long
```

```
lngtime = GetDoubleClickTime()
```

```
Print lngtime
```

```
End Sub
```

در این رویداد نیز ابتدا یک متغیر از نوع عدد صحیح تعریف می‌شود، سپس تابع `GetDoubleClickTime` فراخوانی می‌شود. این تابع بدون آرگومان است و فقط سرعت دو بار کلیک ماوس را بازگشت می‌دهد.

۶- پروژه و فرم را با نام `APIfunction` ذخیره کنید، سپس آن را اجرا کرده و روی دکمه‌های فرمان کلیک کنید. مسیر نصب ویندوز و سرعت دو بار کلیک ماوس مانند شکل ۱۳-۷ نمایش داده می‌شود.



شکل ۱۳-۷

۷- به اجرای پروژه خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.

تمرین: پروژه‌ای طراحی کنید که به وسیله آن بتوان نام کامپیوتر (`Computer Name`) را با استفاده از رویه



GetComputerName به دست آورد و بلندگوی کامپیوتر را با رویه Beep به صدا درآورد.

مثال ۲: می‌خواهیم با استفاده از توابع API زبان جاری را برای تایپ تغییر دهیم. برای این کار لازم است از تابع ActivateKeyboardLayout که در کتابخانه user32.DLL قرار دارد، استفاده شود. این تابع دو آرگومان دریافت می‌کند؛ آرگومان اول که زبان بعدی را در مجموعه زبان‌های انتخاب شده روی سیستم فعال می‌کند و مقدارش باید روی مقدار ۱ تنظیم شود. آرگومان دوم نیز یک مقدار عددی است که باید روی مقدار صفر تنظیم شود. برای تعریف این تابع می‌توان به این صورت عمل کرد:

```
Public Declare Function ActivateKeyboardLayout _
```

```
Lib "user32" (ByVal HKL As Long, ByVal flags As Long) As Long
```

پس از قرار دادن تعریف این تابع در یک ماژول کد می‌توان آن را به این صورت فراخوانی کرد:

```
ActivateKeyboardLayout 1, 0
```

با هر بار اجرای این دستور زبان جاری به زبان بعدی تغییر می‌کند.

تمرین: یک پروژه برای تایپ متن طراحی کنید که در آن امکان تغییر زبان جاری از داخل برنامه وجود داشته باشد.

مثال ۳: می‌خواهیم با استفاده از توابع API امکان خروج کاربر از سیستم را فراهم کنیم. برای این کار از تابع ExitWindowEx که در کتابخانه user32.DLL قرار دارد، استفاده می‌شود. این تابع دو آرگومان عددی دارد که هر دو آرگومان باید روی مقدار صفر تنظیم شوند.

برای تعریف این تابع می‌توان به این صورت عمل نمود:

```
Public Declare Function ExitWindowsEx _
```

```
Lib "user32" (ByVal uFlags As Long, ByVal dwReserved As Long) As Long
```

پس از قرار دادن تعریف این تابع در یک ماژول کد می‌توان آن را به این صورت فراخوانی کرد:

```
ExitWindowsEx 0, 0
```

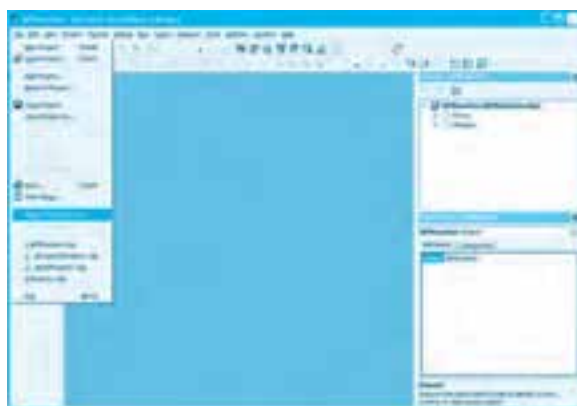
با اجرای این دستور سیستم به‌طور خودکار خاموش می‌شود.

تمرین: یکی از پروژه‌هایی را که تاکنون طراحی کرده‌اید، به گونه‌ای تنظیم کنید که از داخل برنامه امکان خروج کاربر از روی سیستم وجود داشته باشد.

۳-۷ نحوه ایجاد برنامه نصب (Setup)

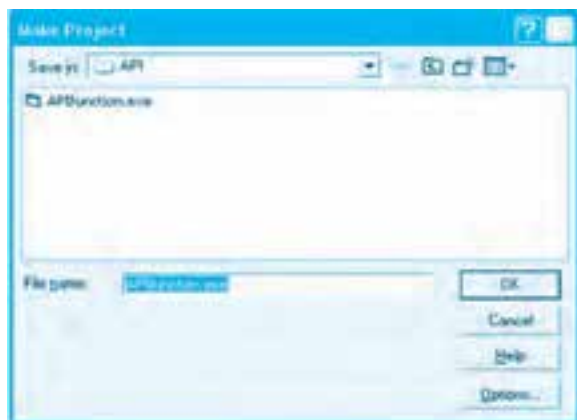
ویژگی مهم دیگری که در زبان برنامه‌نویسی ویژوال بیسیک وجود دارد، امکان ایجاد برنامه اجرای مستقل (EXE) از پروژه‌ها است. به علاوه با استفاده از امکان موجود در این زبان می‌توان به راحتی برنامه نصب یا Setup پروژه را تهیه کرد و در اختیار کاربران قرار داد تا بتوانند برنامه‌های تولید شده را به صورت استاندارد و حرفه‌ای روی سیستمی سریع و راحت نصب کنند. برای این کار می‌توانید این عملیات را به ترتیب انجام دهید:

- ۱- پس از طراحی و ساخت پروژه، برنامه ویژوال بیسیک را اجرا کرده و پروژه خود را باز کنید (در این جا از پروژه‌ای که در مثال ۱ ساخته شده است، استفاده می‌شود).
- ۲- در این مرحله لازم است فایل اجرایی (EXE) از پروژه ایجاد کنید، برای این کار در پنجره ویژوال بیسیک ابتدا روی منوی File و بعد روی گزینه (exe نام پروژه Make) کلیک کنید (شکل ۱۴-۷) تا کادرمحاوره Make Project نمایش داده شود. در این کادرمحاوره می‌توانید نام و مسیر فایل اجرایی را مشخص کنید. به طور پیش‌فرض از نام و مسیر پروژه استفاده شده است که در این صورت فایل مستقل اجرایی EXE ساخته می‌شود.



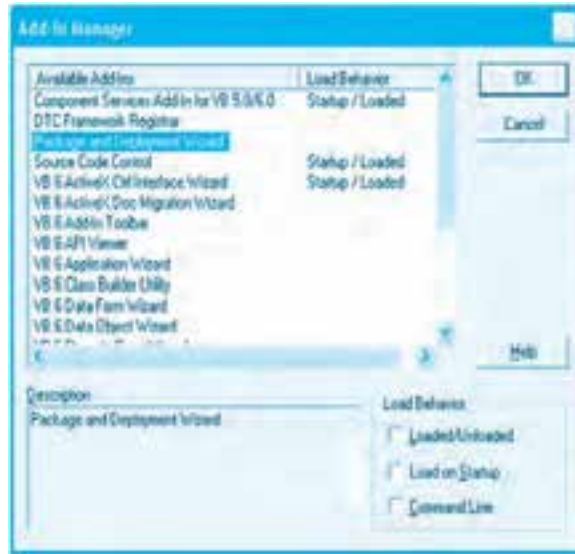
شکل ۱۴-۷

- پس از ایجاد فایل اجرایی می‌توانید به پوشه پروژه خود بروید و روی فایل اجرایی دو بار کلیک کنید. بدون آن‌که پروژه را در برنامه ویژوال بیسیک باز کنید برنامه اجرا می‌شود.



شکل ۱۵-۷

۳- اکنون لازم است ابزار ساخت برنامه نصب آماده استفاده شود. برای این کار باید ابتدا برنامه Package and Deployment Wizard and را به پنجره برنامه ویژوال بیسیک اضافه کرد. برای اضافه کردن این برنامه، ابتدا روی منوی Add-Ins و بعد روی گزینه Add-In Manager... کلیک کنید و در کادرمحاوره Add-In Manage روی گزینه Package and Deployment Wizard کلیک کنید(شکل ۷-۱۶).



شکل ۷-۱۶



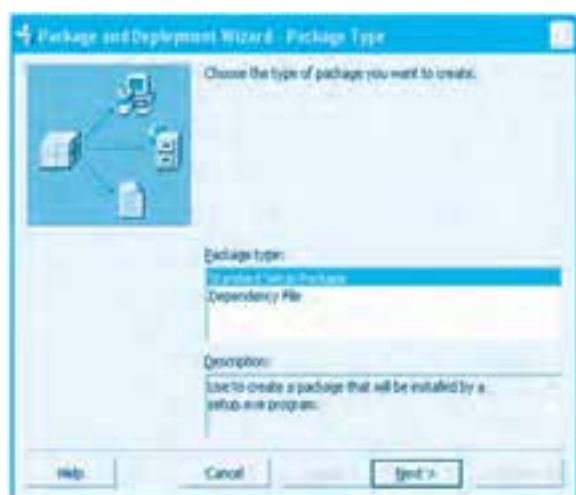
شکل ۷-۱۷

سپس روی کادرهای علامت Loaded/Unloaded و Load on Startup کلیک کنید و در پایان روی دکمه فرمان OK کلیک کنید. در این مرحله یک گزینه با عنوان ... Package and Deployment Wizard به منوی Add-Ins اضافه می‌شود. در پنجره ویژوال بیسیک ابتدا روی منوی Add-Ins و بعد روی گزینه Package and Deployment Wizard کلیک کنید تا اولین کادرمحاوره مانند شکل ۷-۱۷ نمایش داده شود. در این کادرمحاوره مسیر پروژه جاری نمایش داده می‌شود. روی اولین دکمه فرمان Package کلیک کنید تا به مرحله بعد بروید؛ اگر قبلاً برنامه را ترجمه کرده و فایل اجرایی را ایجاد کرده باشید، قبل از این که به مرحله بعد بروید کادر پیغامی مانند شکل ۷-۱۸ نمایش داده می‌شود که در رابطه با ترجمه و ایجاد دوباره فایل اجرایی کسب تکلیف می‌کند.



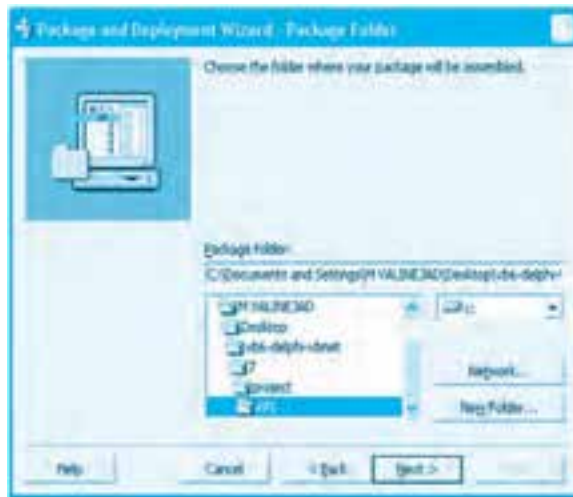
شکل ۷-۱۸

۴- در این مرحله کادرمحاوره‌ای مانند شکل ۷-۱۹ نمایش داده می‌شود. در این کادرمحاوره امکان انتخاب نوع بسته نرم‌افزاری وجود دارد که به طور پیش‌فرض گزینه Standard Setup Package انتخاب شده است. روی دکمه فرمان Next کلیک کنید تا به مرحله بعد بروید.



شکل ۷-۱۹

۵- در این مرحله کادرمحاوره‌ای مانند شکل ۷-۲۰ نمایش داده می‌شود. در این کادرمحاوره می‌توان محل قرارگیری فایل‌های برنامه نصب را انتخاب کرد. به طور پیش‌فرض فایل‌ها در یک پوشه با نام Package در داخل پوشه پروژه قرار می‌گیرند. پس از تعیین پوشه برنامه نصب روی دکمه فرمان Next کلیک کنید تا به مرحله بعد بروید. اگر پوشه تعیین شده وجود نداشته باشد، کادر پیغامی مانند شکل ۷-۲۱ مبنی بر ایجاد پوشه نمایش داده می‌شود. روی دکمه فرمان Yes کلیک کنید تا به مرحله بعد بروید.

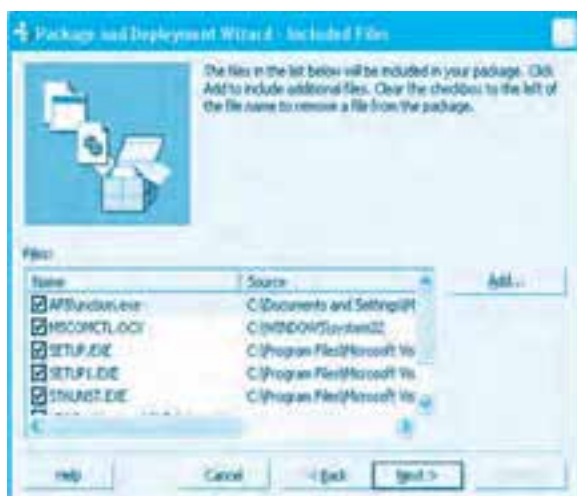


شکل ۷-۲۰



شکل ۷-۲۱

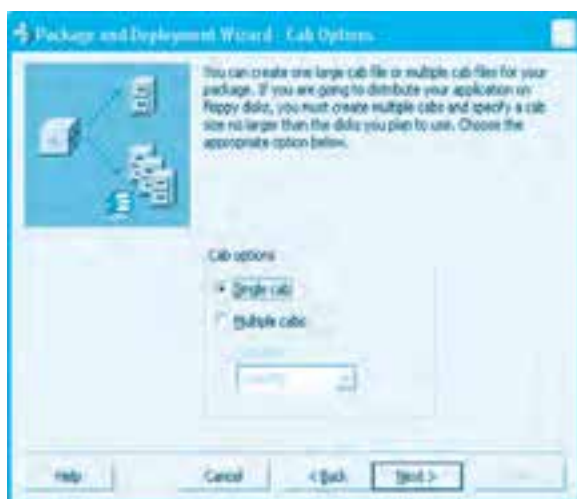
۶- در این مرحله کادرمحاوره‌ای مانند شکل ۷-۲۲ نمایش داده می‌شود. در این کادرمحاوره لیست فایل‌های برنامه نصب نمایش داده می‌شود. اگر پروژه به فایل‌های دیگری نیاز داشته باشد (مانند فایل پایگاه داده)، می‌توانید آن‌ها را با استفاده از دکمه فرمان Add به لیست فایل‌ها اضافه کنید. روی دکمه فرمان Next کلیک کنید تا به مرحله بعد بروید.



شکل ۲۲-۷

۷- در این مرحله کادرمحاوره دیگری مانند شکل ۲۳-۷ نمایش داده می‌شود که می‌توان در آن نحوه ایجاد فایل‌های فشرده را تعیین کرد. اگر می‌خواهید برنامه نصب را روی فلاپی دیسک توزیع کنید، از گزینه Multiple cabs و در غیر این صورت از گزینه پیش‌فرض یعنی Single cab استفاده کنید.

۸- در این مرحله کادرمحاوره‌ای مانند شکل ۲۴-۷ نمایش داده می‌شود. در این کادرمحاوره می‌توانید عنوان برنامه نصب را که هنگام اجرا نمایش داده می‌شود، تعیین کنید. به طور پیش‌فرض نام پروژه انتخاب شده است؛ پس از تعیین عنوان برنامه نصب روی دکمه فرمان Next کلیک کنید تا به مرحله بعد بروید.



شکل ۲۳-۷



شکل ۷-۲۴

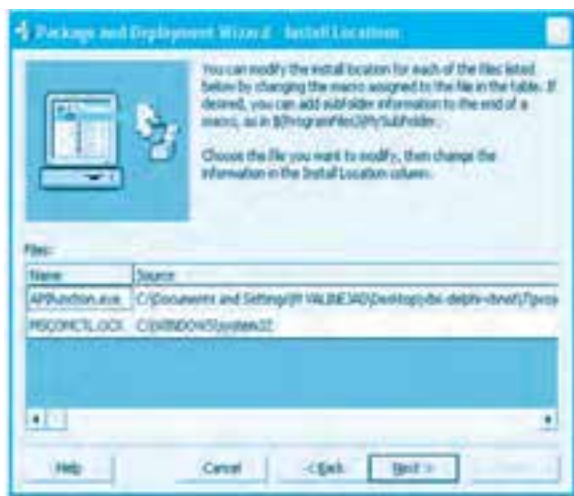
۹- در این مرحله نیز کادرمحاوره‌ای مانند شکل ۷-۲۵ نمایش داده می‌شود. در این کادرمحاوره می‌توانید نحوه ایجاد گروه‌ها و میانبرهای برنامه را در منوی شروع (Start) سیستمی که برنامه روی آن نصب می‌شود، مشخص کنید. با دکمه فرمان New Group می‌توانید یک گروه جدید ایجاد کنید.



شکل ۷-۲۵

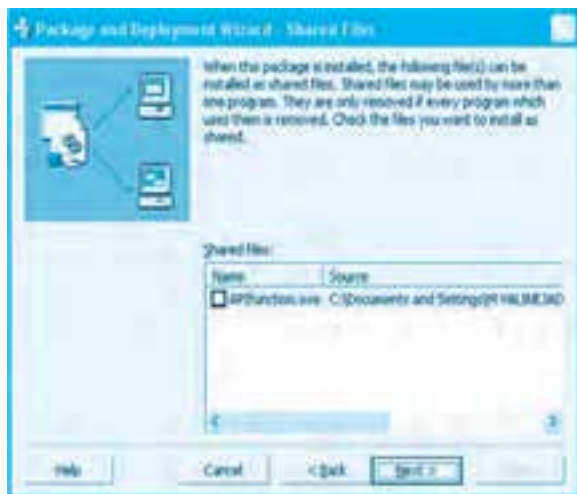
هم‌چنین با دکمه فرمان New Item می‌توانید به ساختار درختی برنامه خود برای نمایش در منوی شروع یک گزینه جدید اضافه کنید. با دکمه فرمان Remove نیز می‌توانید گزینه‌ها و گروه‌های موردنظر را حذف کنید. پس از تنظیم ساختار درختی برنامه در منوی شروع روی دکمه فرمان Next کلیک کنید.

۱۰- در این مرحله کادرمحاوره‌ای مانند شکل ۷-۲۶ نمایش داده می‌شود که در آن می‌توان محل نصب فایل‌های برنامه را هنگام اجرای برنامه نصب در سیستم مربوط مشخص کرد. روی دکمه فرمان Next کلیک کنید تا به مرحله بعد بروید.



شکل ۷-۲۶

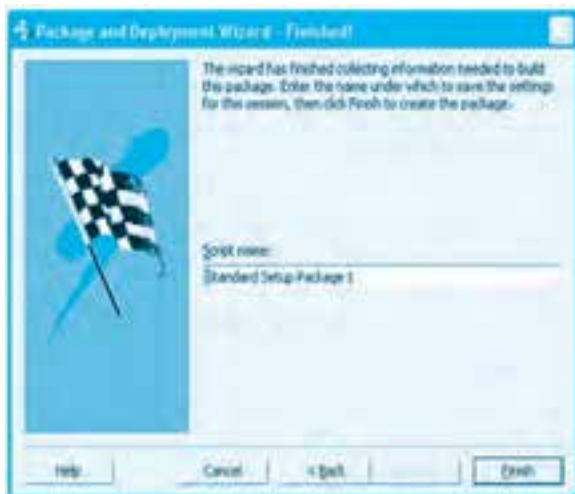
۱۱- در این مرحله کادرمحاوره‌ای مانند شکل ۷-۲۷ نمایش داده می‌شود که در آن می‌توان فایل‌های برنامه را در محیط شبکه به اشتراک گذاشت. بدون انجام هیچ تغییری، روی دکمه فرمان Next کلیک کنید تا به مرحله بعد بروید.



شکل ۷-۲۷

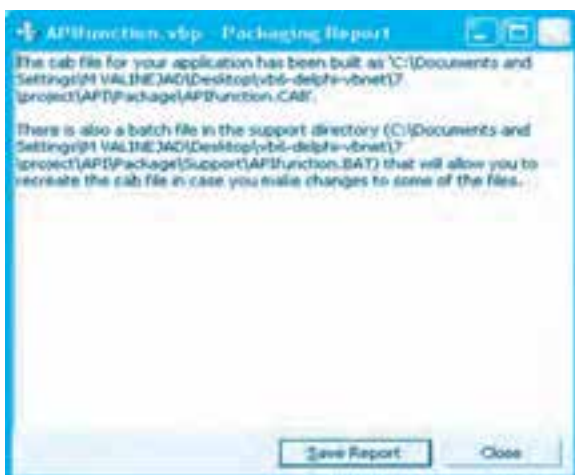
۱۲- در این مرحله، آخرین کادرمحاوره مانند شکل ۷-۲۸ نمایش داده می‌شود که در آن می‌توان نام فایلی

که اطلاعات لازم برای ساخت برنامه نصب را در خود نگهداری می‌کند، تعیین کرد.



شکل ۷-۲۸

روی دکمه فرمان Finish کلیک کنید تا عملیات ساخت برنامه نصب خاتمه یابد. پس از کلیک روی دکمه فرمان Finish مدتی طول می‌کشد تا عملیات ساخت برنامه نصب خاتمه یابد و فایل‌های موردنظر بسته‌بندی شوند. پس از این مرحله پنجره‌ای مانند شکل ۷-۲۹ نمایش داده می‌شود که شامل یک گزارش از ساخت برنامه نصب می‌باشد. در این پنجره می‌توانید گزارش را با دکمه فرمان Save Report ذخیره کنید یا روی دکمه فرمان Close کلیک کنید تا پنجره بسته شود.



شکل ۷-۲۹

نکته: وقتی عملیات نصب خاتمه می‌یابد اولین کادرمحاوره همچنان نمایش داده می‌شود. می‌توانید با کلیک روی دکمه فرمان Close آن را ببندید.

نکته: در کلیه مراحل ساخت برنامه نصب می‌توانید با کلیک روی دکمه فرمان Back به مرحله قبل بازگردید.

نکته: در کلیه مراحل می‌توانید با کلیک روی دکمه فرمان Finish مستقیماً به آخرین مرحله نصب بروید.

تمرین: برای یکی از پروژه‌های خود که در واحدکار قبل ایجاد کرده‌اید، برنامه نصب بسازید.

Learn in English

Accessing the Microsoft Windows API

You can gain access to the Windows API (or other outside DLLs) by declaring the external procedures within your Visual Basic application. After you declare a procedure, you can use it like any other language feature in the product.

The most commonly used set of external procedures are those that make up Microsoft Windows itself. The Windows API contains thousands of functions, subs, types, and constants that you can declare and use in your projects. These procedures are written in the C language, however, so they must be declared before you can use them with Visual Basic. The declarations for DLL procedures can become fairly complex. While you can translate these yourself, the easiest way to access the Windows API is by using the predefined declares included with Visual Basic.

The file Win32api.txt, located in the \Winapi subdirectory of the main Visual Basic directory, contains declarations for many of the Windows API procedures commonly used in Visual Basic. To use a function, type, or other feature from this file, simply copy it to your Visual Basic module. You can view and copy procedures from Win32api.txt by using the API Viewer application, or by loading the file in any text editor.

واژه‌نامه

Available	در دسترس، قابل استفاده
Cell	خانه
Chart	نمودار
Common	عمومی، مشترک
Deployment	گسترش دادن، توزیع کردن
Declare	اعلام کردن
External	خارجی، بیرونی
Gain	به‌دست آوردن، سودبردن
Package	بسته، بسته‌بندی کردن
Product	محصول
Predefined	از پیش تعریف شده
Translate	ترجمه کردن
Workbook	کارپوشه
WorkSheet	کاربرگ

خلاصه مطالب

- در ویژوال بیسیک می‌توان به امکانات برنامه‌هایی مانند Access، Excel یا سایر برنامه‌های مشابه دسترسی پیدا کرد.
- با استفاده از API می‌توان به توابع داخلی ویندوز دسترسی پیدا کرد.
- استفاده از توابع API سبب بهبود عملکرد برنامه و تولید سریع‌تر آن و انجام عملیات به‌روزرسانی خارج از برنامه می‌شوند.
- با استفاده از رویه‌های API می‌توان عملیاتی را که امکان انجام آن‌ها در ویژوال بیسیک وجود ندارد، انجام داد.
- کتابخانه User32.DLL مدیریت منوها، کنترل‌ها، کادرهای محاوره و مانند آن‌ها را به عهده دارد.
- کتابخانه GDI32.DLL مدیریت خروجی گرافیکی، ترسیمات، قلم‌ها و مانند آن‌ها را به عهده دارد.
- کتابخانه Kernel32.DLL وظایف مدیریت حافظه، منابع و سایر عملیات سیستمی سطح پایین را ارائه می‌کند.
- برای استفاده از توابع API لازم است آن‌ها را با دستور Declare تعریف کرد.
- نام فایل کتابخانه‌ای " Lib نام رویه Public Private Declare Function یا Private
- با استفاده از تابع API GetDoubleClick Time می‌توان سرعت دو بار کلیک ماوس را به دست آورد.
- با استفاده از تابع API GetWindowsDirectory می‌توان محل نصب ویندوز را به دست آورد.
- با استفاده از برنامه Package and Deployment Wizard می‌توان برنامه نصب برای پروژه‌ها تهیه نمود.

آزمون نظری

- ۱- با استفاده از کدام فایل کتابخانه‌ای می‌توان عملیاتی روی حافظه انجام داد؟
 الف- User32.DLL ب- Kernel32.DLL ج- GDI32.DLL د- API.DLL
 - ۲- با کدام خاصیت در یک شیء از نوع سند Excel می‌توان به یک خانه دسترسی پیدا کرد؟
 الف- Cell ب- Range ج- Cells د- Add
 - ۳- کدام برنامه برای مشاهده تعریف توابع API استفاده می‌شود؟
 الف- API Viewer ب- API Function ج- API Interface د- API Display
 - ۴- با کدام خاصیت در شیء از نوع سند Excel می‌توان محتویات سند را در یک فایل ذخیره کرد؟
 الف- Active Workbook ب- Active Worksheet
 ج- Save Workbook د- Save Worksheet
 - ۵- کدام گزینه برای تایپ یک متن در یک سند از نوع Word مناسب است؟
 الف- Document.Text ب- Document.TypeText
 ج- Selection.TypeText د- Selection.Text
- 6-The file, located in the \Winapi subdirectory of the main Visual Basic directory, contains declarations for many of the Windows API procedures commonly used in Visual Basic.
- a- User32API.txt b- Kernel32.txt c- GDI32API.txt d- Win32API.txt
- ۷- نحوه استفاده از برنامه Excel در برنامه‌های ویژوال بیسیک را با ذکر مثال توضیح دهید.
 - ۸- نحوه استفاده از برنامه Word در برنامه‌های ویژوال بیسیک را با ذکر مثال توضیح دهید.
 - ۹- توابع API را توضیح دهید و مهم‌ترین فایل‌های کتابخانه‌ای حاوی پرکاربردترین توابع را بیان کنید و وظایف هر یک را توضیح دهید.
 - ۱۰- نحوه استفاده از برنامه API Viewer را توضیح دهید.
 - ۱۱- نحوه ساخت و ایجاد برنامه نصب را توضیح دهید.

آزمون عملی

- ۱- پروژه‌های طراحی کنید که به وسیله آن بتوان برنامه کلاسی یک دانش‌آموز را دریافت کرد و در یک سند از نوع Excel قرار داد و روی دیسک ذخیره نمود.
- ۲- پروژه‌های طراحی کنید که به وسیله آن بتوان یک متن را از کاربر دریافت کرده و در یک سند از نوع Doc ذخیره نمود.
- ۳- پروژه‌های طراحی کنید که به وسیله توابع API بتوان سرعت دو بار کلیک ماوس را در سیستم تنظیم نمود.
- ۴- برای پروژه‌های موجود در آزمون‌های عملی ۱، ۲ و ۳ برنامه نصب تهیه کنید.

واحد کار هشتم

هدف جزئی

توانایی استفاده از محیط Net Framework.

زمان (ساعت)	
عملی	نظری
۳۰	۱۵

هدفهای رفتاری

- پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:
- ۱- NetFramework را بشناسد و اجزای آن را توضیح دهد.
 - ۲- تفاوت‌ها و نقاط مشترک زبان برنامه‌نویسی ویژوال بیسیک Net را با زبان برنامه‌نویسی ویژوال بیسیک توضیح دهد.
 - ۳- انواع داده‌ها را در NetFramework توضیح دهد.
 - ۴- نحوه ایجاد پروژه در ویژوال استودیو ۲۰۰۸ را بداند و اجزای تشکیل دهنده محیط ویژوال استودیو را توضیح دهد.
 - ۵- خاصیت‌ها، متدها و رویدادهای مهم و پر کاربرد کلاس فرم و کنترل‌ها را بداند.
 - ۶- نحوه استفاده از ویداده‌ها را در زبان برنامه‌نویسی ویژوال بیسیک Net بداند.
 - ۷- نحوه اضافه کردن یک فرم جدید و نمایش آن را بداند.
 - ۸- توانایی به‌کار بردن کنترل کادر محاوره باز کردن فایل OpenFileDialog را داشته باشد.
 - ۹- بتواند از کنترل‌های PictureBox, Button, Label, TextBox, RadioButton, CheckBox مانند آن‌ها استفاده کند.
 - ۱۰- خطاهای زمان اجرا را با دستور Try ...catch مدیریت کند.
 - ۱۱- بتواند آرایه‌ها، ساختارها، داده‌های شمارشی و کلکسیون را ایجاد و استفاده کند.
 - ۱۲- اجزای تشکیل دهنده ADO Net را توضیح داده و تفاوت‌های آن را با ADO بیان کند.
 - ۱۳- بتواند کنترل DataAdapter را پیکربندی کرده و داده‌ها را از پایگاه داده‌ها بازیابی کند.
 - ۱۴- بتواند از کلاس‌های Connection و Command جهت انجام عملیات درج، حذف و به‌روز رسانی استفاده کند و خاصیت‌ها و متدهای مهم آن‌ها را توضیح دهد.

کلیات

شرکت مایکروسافت پس از عرضه ویژوال استودیو در سال ۲۰۰۲ با عرضه سیستم‌عامل جدیدی به نام Windows XP با توجه به تغییرات عمده‌ای که در آن به وجود آمده بود بسته نرم‌افزاری جدیدی به نام Visual Studio Net را ارائه کرد. در این بسته نرم‌افزاری زبان‌های برنامه‌نویسی جدیدی مانند C# ارائه شده بودند و زبان برنامه‌نویسی مانند ویژوال بیسیک با تغییرات کمی و کیفی زیادی عرضه شد. یکی از ویژگی‌های جدید در این بسته نرم‌افزاری نزدیک شدن بعضی از زبان‌ها به یکدیگر بود مانند زبان‌های برنامه‌نویسی VB و C# و ویژگی دیگر استفاده از یک محیط یکپارچه برنامه‌نویسی و ترجمه و اجرای برنامه‌ها که به نام Net Framework معرفی شد. در واقع Net Framework یک نرم‌افزار با اجزا و ویژگی‌های خاص است که می‌تواند طراحی، ساخت و اجرای برنامه‌هایی را که به زبان‌های برنامه‌نویسی در محیط ویژوال استودیو Net انجام می‌شود به عهده بگیرد و مدیریت کند. به‌طور کلی Net Framework از بخش‌های مختلفی تشکیل می‌شود که مهم‌ترین بخش‌ها در این محیط عبارتند از:

کتابخانه کلاس‌های Net.

یکی از ویژگی‌های مهم در Net Framework ارائه مجموعه‌ای از انواع کلاس‌های متنوع می‌باشد که در قالب فایل‌های DLL ارائه می‌شوند. با این مجموعه کلاس‌ها برنامه‌نویسان به راحتی می‌توانند اجزای مختلف برنامه‌های خود را تولید نمایند. در Net برای دسترسی به این کلاس‌ها از یک شیوه جدید طبقه‌بندی به نام فضای نام یا Name Space استفاده می‌شود. فضای نام یک یا مجموعه‌ای از اسامی است که به‌صورت سلسله مراتبی نوشته می‌شوند و با نقطه (Dot) از یکدیگر جدا می‌شوند. به عنوان مثال برای دسترسی به کلاس Form از این فضای نام استفاده می‌شود:

System.Windows.Forms

یا برای دسترسی به کلاس Math از این فضای نام استفاده می‌شود:

System

زبان اجرایی مشترک (CLR) (Common Language Runtime)

بخش مهم دیگری در Net Framework وجود دارد که وظیفه اجرای برنامه‌هایی را که به زبان‌های برنامه‌نویسی Net نوشته می‌شوند، به عهده دارد. به‌علاوه هنگام اجرای برنامه، مدیریت حافظه از وظایف این بخش است. وقتی یک برنامه در محیط Net نوشته و اجرا می‌شود ابتدا به کد زبان میانی یا IL (Intermediate Language) ترجمه می‌شود و در این فرایند یک فایل اسمبلی به وجود می‌آید سپس کد حاصل از ترجمه برنامه در اختیار CLR قرار می‌گیرد تا اجرا شود.

زبان‌های برنامه‌نویسی Net.

از زمان ارائه اولین نگارش Net Framework تاکنون زبان‌های برنامه‌نویسی به‌صورت ثابت و بعضی

در نگارش‌های بعدی در این محیط عرضه شده‌اند. در واقع زبان‌های برنامه‌نویسی بخش مهمی از این نرم‌افزار محسوب می‌شوند. به عنوان مثال می‌توان به زبان‌های برنامه‌نویسی VB.Net، C#Net، ++Net VC اشاره نمود. در این واحد کار زبان برنامه‌نویسی ویژوال بیسیک (Net) مورد بررسی قرار می‌گیرد. زبان برنامه‌نویسی ویژوال بیسیک Net. نقاط مشترک زیادی با زبان برنامه‌نویسی ویژوال بیسیک ۶ دارد. به عنوان مثال، دستورات شرطی و حلقه در زبان برنامه‌نویسی ویژوال بیسیک ۶ به همان شکل در زبان برنامه‌نویسی ویژوال بیسیک Net. قابل استفاده است. هم‌چنین عملگرهای مقایسه‌ای، منطقی و ریاضی مانند قبل قابل استفاده هستند. البته ویژوال بیسیک Net دارای ویژگی‌های کاملاً جدیدی نیز می‌باشد که در نگارش قبلی وجود ندارد. به عنوان مثال ویژوال بیسیک Net. از تمام ویژگی‌های شیء‌گرایمانند ارث‌بری و چند ریختی^۱ کاملاً پشتیبانی می‌کند. به‌علاوه ADO.Net به شکل بسیار تکامل‌یافته‌تر با امکانات بیشتر نسبت به ADO قابل استفاده می‌باشد.

انواع داده‌ها

به‌طور کلی داده‌ها در Net. به دو گروه عمده تقسیم می‌شوند: داده‌ها از نوع مقدار (Value Type) و داده‌ها از نوع مرجع (Reference Type). داده‌ها از نوع مقدار شامل انواع داده‌های عددی، منطقی، کاراکتری، ساختارها (Structure) و انواع داده شمارشی (enum) می‌باشند. جدول ۱-۸ انواع داده‌ها از نوع مقدار را نشان می‌دهد و داده‌های از نوع مقدار شامل انواع کلاس‌ها، آرایه‌ها، رشته‌ها، Object و غیره می‌باشند. روش تعریف داده‌های از نوع مقدار و رشته‌ها و مقداردهی به آن‌ها مشابه ویژوال بیسیک ۶ می‌باشد. برای تعریف داده‌ها از نوع مرجع با توجه به نوع داده روش خاصی استفاده می‌شود که بعضی از آن‌ها در این پیمانه مهارتی بررسی می‌شوند. به منظور تعریف آرایه‌ها از داده‌های نوع مقدار نیز می‌توان از روشی مشابه ویژوال بیسیک ۶ استفاده کرد. برای تعریف ثابت‌ها نیز در زبان برنامه‌نویسی ویژوال بیسیک Net مشابه زبان برنامه‌نویسی ویژوال بیسیک عمل می‌شود. شکل کلی نحوه تعریف ثابت‌ها در زبان برنامه‌نویسی Net به این صورت است:

مقدار = نوع داده As نام ثابت Const [Public یا Private]

به عنوان مثال این دستورات سه ثابت از انواع مختلف را تعریف می‌کنند:

```
Const maximum As Long = 500
```

```
Private Const helpstring As String = "Help"
```

```
Public Const stortvalue As Integer = 5
```

۱-۸ نحوه ایجاد پروژه در ویژوال بیسیک 2008 .Net

پس از نصب برنامه ویژوال استودیو Net. نگارش ۲۰۰۸ در منوی شروع (Start) ویندوز، ابتدا روی گزینه All Programs و سپس گزینه Microsoft Visual Studio 2008 و بعد گزینه Microsoft Visual Studio 2008 کلیک کنید. در این مرحله کادر محاوره‌ای مانند شکل ۱-۸ نمایش داده می‌شود.

جدول ۸-۱ انواع داده‌ها از نوع مقدار

نوع داده	معادل .net	فضای اشغالی در حافظه	محدوده مقادیر
Boolean	Boolean	Depends on implementing platform	True or False
Byte	Byte	1 byte	0 through 255 (unsigned)
Char (single character)	Char	2 bytes	0 through 65535 (unsigned)
Date	DateTime	8 bytes	0:00:00 (midnight) on January 1, 0001 through 11:59:59 PM on December 31, 9999
Decimal	Decimal	16 bytes	0 through +/-79,228,162,514,264,337,593,543,950,335 (+/-7.9..E+28) † with no decimal point; 0 through +/-7.9228162514264337593543950335 with 28 places to the right of the decimal; .smallest nonzero number is +/- -00000000000000000000000000000001 (+/-1E-28)
Double (double-precision floating-point)	Double	8 bytes	-1.79769313486231570E+308 through -4.94065645841246544E-324 † for negative values; 4.94065645841246544E-324 through 1.79769313486231570E+308 † for POSitive values
Integer	Int32	4 bytes	-2,147,483,648 through 2,147,483,647 (signed)

Long (long integer)	Int64	8 bytes	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807 (9.2...E+18 †) (signed)
SByte	SByte	1 byte	-128 through 127 (signed)
Short (short integer)	Int16	2 bytes	-32,768 through 32,767 (signed)
Single (single-precision floating-point)	Single	4 bytes	-3.4028235E+38 through -1.401298E-45 † for negative values; 1.401298E-45 through 3.4028235E+38 † for positive values
UInteger	UInt32	4 bytes	0 through 4,294,967,295 (unsigned)
ULong	UInt64	8 bytes	0 through 18,446,744,073,709,551,615 (1.8...E+19 †) (unsigned)
UShort	UInt16	2 bytes	0 through 65,535 (unsigned)



شکل ۸-۱

در این کادر محاوره می‌توان نحوه نمایش پنجره ویژوال استودیو را مطابق میل خود تنظیم کنید. گزینه General Development Settings را انتخاب کرده و روی دکمه فرمان OK کلیک کنید. پس از این مرحله پنجره ویژوال استودیو مانند شکل ۸-۲ نمایش داده می‌شود.



شکل ۸-۲

در این پنجره می‌توانید در زبانه Start Page در بخش Recent Projects روی گزینه Create: Project ... کلیک کنید یا در نوار منوی پنجره ویژوال استودیو ابتدا روی منوی File و سپس روی گزینه New Project ... کلیک کنید (شکل ۸-۳).



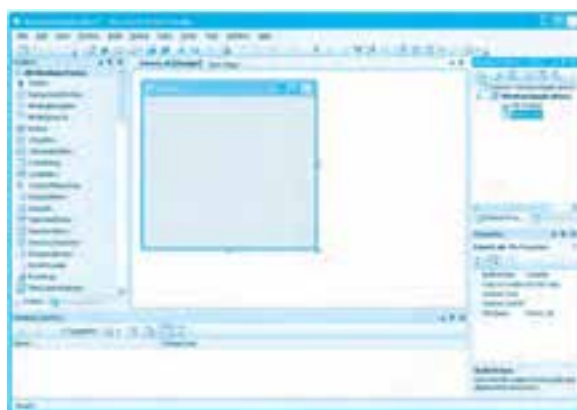
شکل ۸-۳

در هر صورت کادر محاوره New Project مانند شکل ۸-۴ نمایش داده می‌شود.



شکل ۴-۸

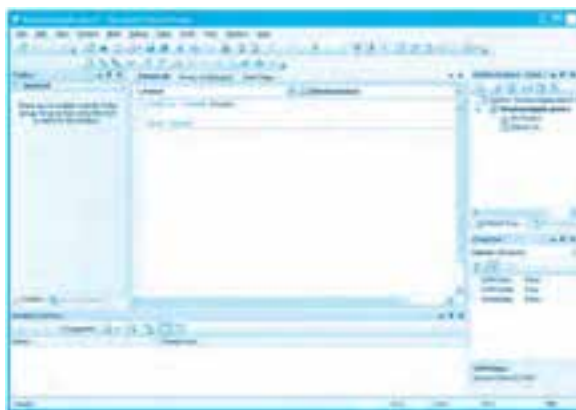
در این کادرمحاوره (شکل ۴-۸) در ساختار درختی Visual Basic و در شاخه Windows و در کادر لیست Templates روی آیکن Windows Forms Application کلیک کنید، سپس نام پروژه خود را در کادر متن Name تایپ کرده و روی دکمه فرمان OK کلیک کنید. در این مرحله پنجره ویژوال استودیو مانند شکل ۵-۸ نمایش داده می‌شود.



شکل ۵-۸

پنجره ویژوال استودیو علی‌رغم تغییرات ظاهری زیادی که نسبت به ویژوال استودیو ۶ کرده، اما تقریباً از همان بخش‌های اصلی تشکیل شده است؛ در قسمت بالای پنجره نوار منو و نوار ابزار قرار دارند. در سمت چپ پنجره ویژوال استودیو Net. جعبه ابزار قرار گرفته است که کنترل‌ها روی آن قرار دارند. در سمت راست به جای پنجره پروژه، مرورگر راه‌حل یا (SE) Solution Explorer قرار دارد. که ساختار پروژه و اجزای تشکیل دهنده آن را نشان می‌دهد


در این مرورگر ساختار درختی پروژه نمایش داده می‌شود. در واقع پروژه در Net. به عنوان یک Solution در نظر گرفته می‌شود. یک Solution خود می‌تواند از اجزای دیگری تشکیل شود که به عنوان مثال در این نوع پروژه که انتخاب شده است شامل یک Windows Application (WA) می‌باشد. هر WA نیز می‌تواند از اجزای دیگری مانند فرم‌ها یا سایر کلاس‌ها تشکیل شود که به‌طور پیش‌فرض یک فرم در WA قرار داده شده است. در پنجره SE می‌توانید مانند VB6 با استفاده از دکمه‌های View Designer و View Code به ترتیب شکل ظاهری و بخش کد را برای هر فرم نمایش دهید. در زیر پنجره SE، پنجره خواص قرار دارد که به وسیله آن می‌توان خاصیت‌های فرم‌ها و کنترل‌ها را در زمان اجرا تنظیم نمود. در قسمت مرکزی نیز می‌توان شکل ظاهری فرم‌ها را مشاهده و تنظیم کرد یا در پنجره کدنویسی، دستورالعمل‌های برنامه را تایپ نمود. اگر در پنجره SE، ابتدا روی فرم و بعد روی دکمه View Code کلیک کنید یک ماژول کلاس که مربوط به فرم می‌گردد نمایش داده می‌شود.



شکل ۶-۸

نکته: برای حرکت روی پنجره‌های باز شده می‌توانید روی زبانه پنجره موردنظر یا روی دکمه سمت راست پنجره طراحی کلیک کنید و از لیستی که نمایش داده می‌شود، پنجره موردنظر خود را انتخاب کنید.

نکته: برای بستن یک پنجره طراحی یا کدنویسی می‌توانید پس از کلیک روی زبانه پنجره موردنظر روی گزینه Close یا دکمه Close در سمت راست پنجره طراحی یا کدنویسی کلیک کنید.

نحوه ذخیره کردن پروژه در نگارش‌های مختلف Net. متفاوت است در نگارش ۲۰۰۸ می‌توانید روی دکمه  کلیک کنید. در این صورت کادرمحاوره‌ای مانند شکل ۷-۸ نمایش داده می‌شود. در این کادرمحاوره می‌توانید در کادر متن Name نام WA، در کادر متن Location درایو و مسیر ذخیره‌سازی فایل‌ها و در کادر متن Solution Name نام Solution یا به عبارت دیگر پروژه خود را تایپ کنید. اگر می‌خواهید یک پوشه اصلی برای کلیه پوشه‌ها و فایل‌های پروژه نیز ایجاد شود، روی کادر علامت Create directory for solution نیز کلیک کنید و در پایان روی دکمه فرمان Save کلیک کنید.



شکل ۷-۸

برای اجرای برنامه‌هایی که در ویژوال بیسیک Net نوشته می‌شوند می‌توانید در نوار ابزار پنجره ویژوال استودیو روی دکمه Start Debugging کلیک کنید یا در نوار منوی ویژوال استودیو ابتدا روی منوی Debug و سپس روی گزینه Start Debugging کلیک کنید. در این صورت برنامه به زبان IL ترجمه می‌شود و سپس CLR آن را اجرا خواهد کرد. وقتی برنامه اجرا می‌شود با توجه به دستوراتی که ویژوال استودیو به برنامه اضافه کرده است، از کلاس Form1 پروژه یک نمونه شیء ایجاد شده و نمایش داده می‌شود.

برای باز کردن پروژه‌ها نیز می‌توانید پس از اجرای برنامه ویژوال استودیو در زبانه Start Page در بخش Recent Projects روی نام پروژه موردنظر کلیک کنید تا باز شود. در این بخش آخرین پروژه‌هایی را که باز کرده‌اید، نمایش داده می‌شوند یا در پنجره ویژوال استودیو روی منوی File و بعد روی گزینه Open Project ... کلیک کنید و در کادرمحاوره Open Project پروژه خود را پیدا کرده و پس از انتخاب آن، روی دکمه فرمان Open کلیک کنید تا پروژه باز شود. در ویژوال بیسیک Net، پسوند فایل‌های پروژه Sln می‌باشد.

۲-۸ کلاس فرم (Form)

با توجه به این که بسیاری از خاصیت‌ها، متدها و رویدادهای فرم در ویژوال بیسیک ۶ مورد بررسی قرار گرفته‌اند و اعضای مشابه زیادی وجود دارد با این حال در این جا به‌طور خلاصه مهم‌ترین اعضای کلاس فرم بررسی می‌شود.

جدول ۲-۸ مهم‌ترین ویژگی‌های فرم

نام خاصیت	توضیح
Name	نام کلاس فرم را مشخص می‌کند.
BackColor	رنگ زمینه را تعیین می‌کند.
BackgroundImage	تصویر زمینه فرم را تعیین می‌کند.
BackgroundImage Layout	تصویر زمینه فرم را روی فرم به صورت کاشی، در مرکز یا با یک تصویر نمایش می‌دهد.
ControlBox	دکمه‌های کنترلی را فعال یا غیرفعال می‌کند.
Cursor	شکل اشاره‌گر ماوس را تعیین می‌کند.
Enabled	فرم را فعال و غیرفعال می‌کند.
Font	ویژگی‌های قلم را تعیین می‌کند.
ForeColor	رنگ قلم را روی فرم تعیین می‌کند.
FormBorderStyle	نوع فرم و ویژگی‌های کلی آن مانند قابلیت تغییر اندازه، کادر دور فرم، دکمه‌های کنترلی و غیره را مشخص می‌کند.
Icon	آیکن منوی کنترل را مشخص می‌کند.
MaximizeBox	دکمه به حداکثر رسانی را فعال و غیرفعال می‌کند.
MinimizeBox	دکمه به حداقل رسانی را فعال و غیرفعال می‌کند.
Opacity	شفافیت فرم را تعیین می‌کند.
RightToLeft	شکل ظاهری فرم را برای زبان‌های راست به چپ مثل فارسی تنظیم می‌کند. در این حالت عنوان فرم در سمت راست نوار عنوان نمایش داده می‌شود.
RightToLeft Layout	دکمه‌های کنترل را برای زبان‌های راست به چپ مثل فارسی تنظیم می‌کند.
ShowIcon	آیکن منوی کنترل را مرئی یا نامرئی می‌کند.
ShowInTaskbar	این خاصیت تعیین می‌کند که در زمان اجرای برنامه آیکن فرم در نوار وظیفه نمایش داده شود یا خیر.

Size	اندازه فرم را (عرض و ارتفاع) تعیین می‌کند.
Location	موقعیت فرم را (Left و Top) برای نمایش تعیین می‌کند.
Text	عنوان فرم را مشخص می‌کند.
StartPosition	موقعیت فرم را در زمان نمایش تعیین می‌کند. اگر روی مقدار Manual تنظیم شود با استفاده از خاصیت Location، اگر روی CenterScreen تنظیم شود در مرکز desktop و اگر روی مقدار WindowsDefaultLocation تنظیم شود، ویندوز موقعیت فرم را در زمان نمایش تعیین می‌کند.
WindowState	وضعیت فرم را در زمان نمایش نشان می‌دهد. اگر روی Normal تنظیم شود فرم با ابعادی که در خاصیت Size تعیین شده نمایش داده می‌شود، اگر روی Minimized تنظیم شود فرم با حداقل اندازه و اگر روی Maximized تنظیم شده باشد، فرم با حداکثر اندازه نمایش داده می‌شود.
MaximumSize	حداکثر اندازه فرم را تعیین می‌کند.
MinimumSize	حداقل اندازه فرم را تعیین می‌کند.

جدول ۳-۸ مهم‌ترین رویدادهای فرم

نام خاصیت	توضیح
Click	این رویداد وقتی روی فرم عمل کلیک انجام شود، اجرا می‌شود.
DoubleClick	این رویداد وقتی روی فرم دو بار کلیک شود، اجرا می‌شود.
Activated	این رویداد وقتی فرم فوکوس را کسب می‌کند، اجرا می‌شود.
Deactivated	این رویداد وقتی فرم فوکوس خود را از دست می‌دهد، اجرا می‌شود.
FormClosed	این رویداد وقتی فرم بسته شده و از حافظه خارج می‌شود، اجرا می‌شود.
Form Closing	این رویداد وقتی اجرا می‌شود که فرم در حال بسته شدن است.
KeyDown	وقتی یک کلید در صفحه کلید فشرده شود، اجرا می‌شود.
KeyPress	وقتی یک کلید در صفحه کلید فشرده شود، اجرا می‌شود.
KeyUp	این رویداد وقتی که کلید فشرده شده رها شود، اجرا می‌شود.

Load	این رویداد وقتی که فرم در حافظه بارگذاری شود، اجرا می‌شود.
MouseDown	این رویداد وقتی که کلیدهای ماوس فشرده شوند، اجرا می‌شود.
MouseUp	این رویداد وقتی که کلیدهای ماوس پس از فشرده شدن رها شوند، اجرا می‌شود.
MouseMove	این رویداد وقتی که اشاره‌گر ماوس روی فرم حرکت کند، اجرا می‌شود.
Resize	این رویداد وقتی ابعاد فرم تغییر کند، اجرا می‌شود.
Shown	این رویداد وقتی فرم روی desktop نمایش داده شود، اجرا می‌شود.

جدول ۴-۸ مهم‌ترین متدهای فرم

نام خاصیت	توضیح
Close	فرم را بسته و آن را از حافظه خارج می‌کند.
Focus	فوکوس را به فرم انتقال می‌دهد.
Show	فرم را روی desktop نمایش می‌دهد.
ShowDialog	فرم را روی desktop به صورت کادرمحاوره نمایش می‌دهد.
Hide	فرم را مخفی می‌کند.

۳-۸ کنترل‌ها در ویژوال بیسیک .Net

در این زبان برنامه‌نویسی کنترل‌های بسیار متنوعی در اختیار برنامه‌نویسان قرار می‌گیرد که بعضی از آن‌ها در ویژوال بیسیک نگارش ۶ بررسی شده‌اند. در این جا مهم‌ترین خاصیت‌ها، رویدادها و متدهای مشترک بین کنترل بررسی می‌شوند.

جدول ۵-۸ مهم‌ترین خاصیت‌های مشترک کنترل‌ها

نام خاصیت	توضیح
Name	نام کنترل را تعیین می‌کند.
AutoSize	در بعضی از کنترل‌ها، ابعاد کنترل را با توجه به خاصیت Text آن تنظیم می‌کند.
BackColor	رنگ زمینه را در کنترل تعیین می‌کند.

BackgroundImage	تصویر زمینه را در کنترل تعیین می‌کند.
Background Layout	تصویر زمینه را روی کنترل به صورت کاشی، در مرکز یا با یک تصویر نمایش می‌دهد.
Cursor	شکل اشاره‌گر ماوس را تعیین می‌کند.
Dock	با استفاده از این خاصیت می‌توان کنترل را به سمت چپ، بالا، پایین، راست یا تمام سطح فرم چسباند.
Font	ویژگی قلم را تعیین می‌کند.
ForeColor	رنگ قلم را تعیین می‌کند.
Image	تصویر روی کنترل را تعیین می‌کند.
Location	موقعیت کنترل را (Left و Top) روی فرم تعیین می‌کند.
MaximumSize	حداکثر اندازه کنترل را مشخص می‌کند.
MinimumSize	حداقل اندازه کنترل را مشخص می‌کند.
RightToLeft	شکل ظاهری کنترل را برای زبان‌های راست به چپ تنظیم می‌کند.
Size	اندازه کنترل را (عرض و ارتفاع) تعیین می‌کند.
TabIndex	ترتیب کسب فوکوس را تعیین می‌کند.
TabStop	اگر روی مقدار True تنظیم شود کنترل می‌تواند فوکوس را کسب کند.
Text	عنوان کنترل را تعیین می‌کند.
Visible	کنترل را مرئی یا مخفی می‌کند.

جدول ۶-۸ مهم‌ترین رویدادهای کنترل‌ها

نام خاصیت	توضیح
Click	این رویداد وقتی روی کنترل عمل کلیک انجام شد، اجرا می‌شود.
DoubleClick	این رویداد وقتی عمل دو بار کلیک انجام شد، اجرا می‌شود.
KeyDown	این رویداد وقتی کلیدی فشرده شود، اجرا می‌شود.

KeyPress	این رویداد وقتی کلیدی فشرده شود، اجرا می‌شود.
KeyUp	این رویداد وقتی که کلید فشرده شده رها شد، اجرا می‌شود.
MouseDown	این رویداد وقتی که کلیدهای ماوس روی کنترل فشرده شوند، اجرا می‌شود.
MouseUp	این رویداد وقتی که کلیدهای ماوس پس از فشردن شدن رها شدند، اجرا می‌شوند.
MouseMove	این رویداد وقتی که اشاره‌گر ماوس روی فرم حرکت کند، اجرا می‌شود.
Resize	این رویداد وقتی ابعاد کنترل تغییر کند، اجرا می‌شود.

جدول ۷-۸ مهم‌ترین متدهای کنترل‌ها

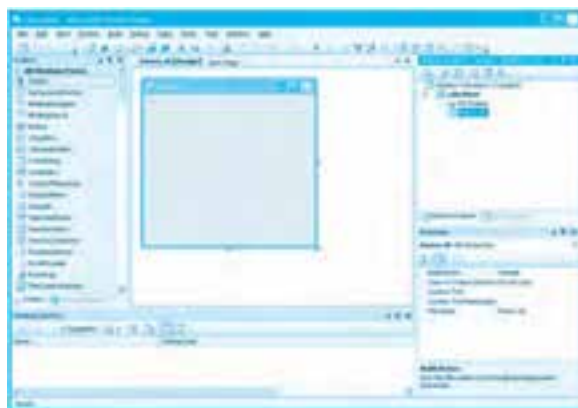
نام خاصیت	توضیح
Focus	فوکوس را به کنترل انتقال می‌دهد.
Show	کنترل مخفی شده را نمایش می‌دهد.
Hide	کنترل را مخفی می‌کند.

مثال ۱: می‌خواهیم یک ماشین حساب ساده مانند شکل ۸-۸ طراحی کنیم که بتواند مجموع دو عدد دلخواه را محاسبه کند و نمایش دهد. برای انجام این مثال عملیات بعد را به ترتیب انجام دهید:



شکل ۸-۸

۱- برنامه ویژوال استودیو ۲۰۰۸ را اجرا کنید و یک پروژه جدید ایجاد کنید و در کادرمحاوره New Project در ساختار درختی Visual Basic نوع پروژه را Windows Forms Application انتخاب نمایید، سپس در کادر متن Name نام پروژه را روی Calculator تنظیم کنید و بعد روی دکمه فرمان OK کلیک کنید تا پنجره ویژوال استودیو مانند شکل ۸-۹ نمایش داده شود.




شکل ۸-۹

۲- در پنجره مرورگر راه حل (SE) روی آیکن Form1 کلیک کنید، سپس روی نام فرم کلیک راست کنید و در منویی که نمایش داده می‌شود روی گزینه Rename کلیک کنید سپس نام فرم را به frmcalc.vb تغییر دهید تا علاوه بر نام فایلی که اطلاعات فرم را نگهداری می‌کند (Form1.vb) نام کلاس فرم نیز روی عبارت frmcalc تنظیم شود. فایل‌هایی که اطلاعات فرم را نگهداری می‌کند دارای پسوند vb می‌باشند.

۳- به پنجره خواص بروید و خاصیت Text فرم را روی عبارت Calculator تنظیم کنید، سپس مانند شکل ۸-۸ سه کنترل برچسب، سه کنترل کادر متن و یک کنترل دکمه (Button) روی فرم قرار دهید.

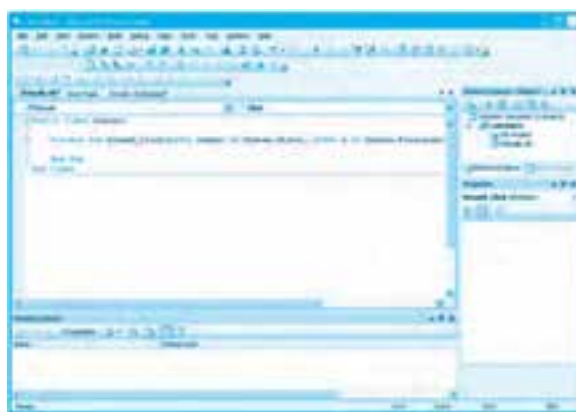
نکته: کنترل دکمه Button مشابه کنترل دکمه فرمان Command Button در ویژوال بیسیک

۶ می‌باشند.

۴- اکنون لازم است تا رویداد دکمه فرمان برای انجام عمل جمع تنظیم شود. برای این کار در پنجره طراحی فرم روی دکمه فرمان Add کلیک کنید، سپس به پنجره خواص بروید و روی دکمه رویداد  کلیک کنید تا لیستی از رویدادهای دکمه فرمان نمایش داده شود. رویداد Click را انتخاب کرده و روی آن دو بار کلیک

کنید؛ در این صورت پنجره کد و رویداد Click دکمه فرمان مانند شکل ۱۰-۸ داخل کلاس فرم ایجاد شده و نمایش داده می‌شود و مکان‌نما داخل رویداد قرار می‌گیرد.

نکته: می‌توانید از روش‌هایی که در ویژوال بیسیک ۶ برای دسترسی به رویدادها به کار می‌روند نیز استفاده کنید.



شکل ۱۰-۸

رویدادها در ویژوال بیسیک Net مانند ویژوال بیسیک ۶ به صورت رویه‌های فرعی (SUB) می‌باشند. اکنون رویداد Click دکمه Add را به این صورت تنظیم کنید:

```
Private Sub btnadd_Click _
```

```
    (ByVal sender As System.Object, ByVal e As System.EventArgs) _
```

```
    Handles btnadd.Click
```

```
    Dim no1 As Decimal
```

```
    Dim no2 As Decimal
```

```
    Dim result As Decimal
```

```
    no1 = Val(txtno1.Text)
```

```
    no2 = Val(txtno2.Text)
```

```
    result = no1 + no2
```

```
    txtresult.Text = Str(result)
```

```
End Sub
```

در این رویداد ابتدا متغیرهای لازم برای انجام عملیات جمع دو عدد از نوع داده اعشاری Decimal تعریف می‌شوند. سپس با استفاده از تابع Val مقادیری که در کادرهای متن وارد شده‌اند به عدد تبدیل شده و در متغیرهای no1 و no2 ذخیره می‌شوند. مجموع آن‌ها نیز محاسبه شده و در متغیر result ذخیره می‌شود و در پایان مقدار متغیر result به رشته تبدیل شده و در کادر متن سوم یعنی txtresult نمایش داده می‌شود.

۵- اکنون روی دکمه Save All کلیک کرده و پس از تعیین مسیر ذخیره‌سازی پروژه روی دکمه فرمان Save کلیک کنید.

۶- در این مرحله با فشردن کلید F5 برنامه را اجرا کنید. یک عدد در هر یک از کادرهای متن تایپ کرده و روی دکمه Add کلیک کنید، نتیجه در کادر متن txtresult نمایش داده می‌شود. حاصل جمع را بررسی کنید.

۷- به اجرای برنامه خاتمه داده و به پنجره ویژوال استودیو بازگردید.

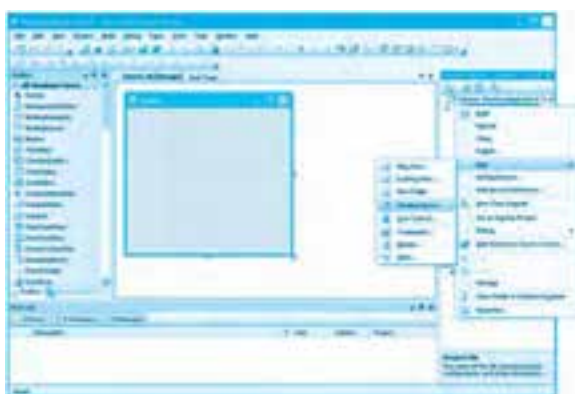
نکته: پروژه ماشین حساب را به شکلی تنظیم کنید که بتوان عملیات تفریق، ضرب و تقسیم را نیز انجام داد.



۴-۸ نحوه ایجاد و اضافه کردن فرم‌های جدید به پروژه

برای اضافه کردن فرم‌های جدید به پروژه‌ها می‌توانید به این صورت عمل کنید:

۱- در پنجره مرورگر راه‌حل روی آیکن Windows Application کلیک راست کنید و در منویی که نمایش داده می‌شود ابتدا روی گزینه Add و بعد روی گزینه Window Form ... کلیک کنید (شکل ۱۱-۸).



شکل ۱۱-۸

در این مرحله کادرمحاوره Add New Item مانند شکل ۱۲-۸ نمایش داده می‌شود.



شکل ۱۲-۸

نام فرم را در کادر متن Name تایپ کنید و بعد روی دکمه فرمان Add کلیک کنید. اکنون یک کلاس فرم جدید به پروژه اضافه شده و شکل ظاهری آن در پنجره طراحی نمایش داده می‌شود.

۲- در این مرحله می‌توانید مانند فرم اول، فرم جدید را طراحی و آماده نمایید، سپس فرم جدید را در یک محل و زمان مناسب نمایش دهید. برای این کار لازم است ابتدا از کلاس فرم جدید یک نمونه (شیء) ایجاد کرد، سپس با متد Show یا Show Dialog آن را نمایش داد. شکل کلی نحوه تعریف شیء از یک کلاس به این صورت است:

نام کلاس As New نام شیء Dim

به عنوان مثال برای ایجاد شیء از کلاس Form2 و نمایش آن می‌توان به این صورت عمل کرد:

```
Dim myform As New Form2
```

```
Myform.Show()
```

۱-۴-۸ نحوه ایجاد فرم‌های کادر محاوره

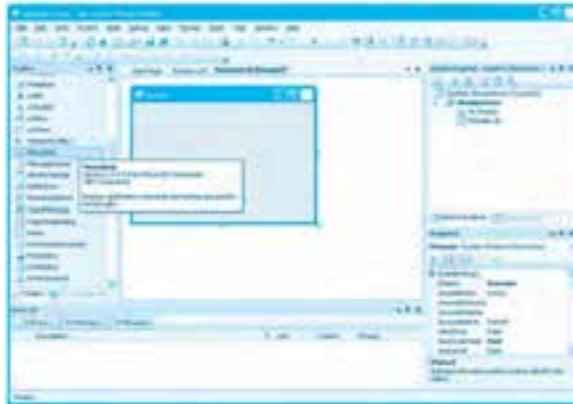
برای ایجاد کادرهای محاوره در پروژه‌ها می‌توان از کنترل‌های کادر محاوره مانند Open File Dialog, Save File Dialog و غیره استفاده کرد سپس با استفاده از متد Show Dialog آن‌ها را در زمان مناسب نمایش داده و استفاده نمود.

مثال ۲: می‌خواهیم پروژه‌ای طراحی کنیم که به وسیله آن بتوان تصاویر دلخواه را مشاهده کرد. برای این کار این عملیات را به ترتیب انجام دهید:

۱- برنامه ویژوال استودیو را اجرا کرده و یک پروژه از نوع Windows Forms Application با نام Show Pictures ایجاد کنید.

۲- پس از ایجاد پروژه نام فایل و کلاس Form1 را روی عبارت frmmain.vb و عنوان آن را روی Show Picture تنظیم کنید.

۳- در این مرحله می‌خواهیم یک نوار منو روی فرم برنامه قرار دهیم؛ برای این کار در جعبه ابزار پنجره ویژوال استودیو مانند شکل ۸-۱۳ روی کنترل MenuStrip دوبار کلیک کنید.



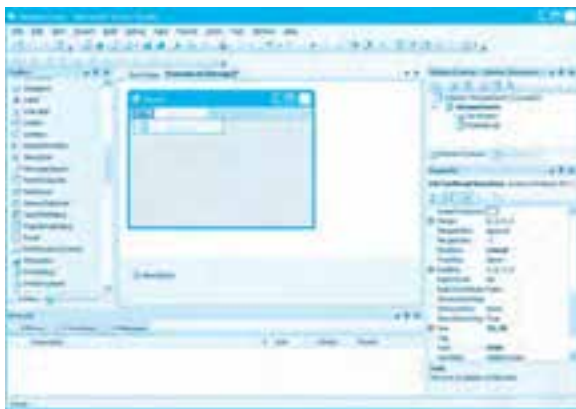
شکل ۸-۱۳

در این صورت کنترل منو، داخل نوار مستطیل شکلی در پایین پنجره طراحی فرم و نوار منو نیز در بالای فرم به صورت یک نوار خالی نمایش داده می‌شوند (شکل ۸-۱۴).



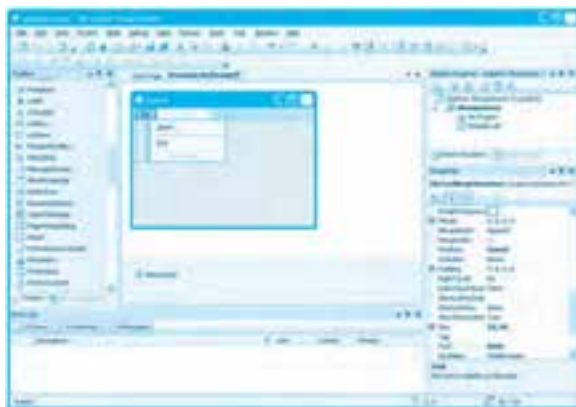
شکل ۸-۱۴

۴- در پنجره طراحی فرم روی نوار منو کلیک کنید تا در سمت چپ نوار منو عبارت Type Here نمایش داده شود. روی این عبارت کلیک کنید و عبارت &File را تایپ کنید تا اولین گزینه در نوار منو ایجاد شود اکنون روی عبارت Type Here که در زیر منوی File نمایش داده می‌شود (شکل ۸-۱۵) کلیک کنید و عبارت &Open ... را تایپ کنید.



شکل ۸-۱۵

سپس به همین روش یک گزینه با عبارت - برای ایجاد خط جداکننده و یک گزینه با عبارت E&xit ایجاد کنید تا نوار منو مانند شکل ۸-۱۶ کامل شود.



شکل ۸-۱۶

۵- در این مرحله روی منوی File در پنجره طراحی فرم کلیک کنید و بعد به پنجره خواص بروید و خاصیت Name آن را روی عبارت mnufile تنظیم کنید و به همین روش نام گزینه Open را روی mnuopen و نام گزینه Exit را روی mnuexit و نام گزینه‌ای که خط جداکننده را نمایش می‌دهد، روی mnusep تنظیم کنید.

۶- اکنون می‌خواهیم برای گزینه Open یک کلید میانبر یا ShortCutKey انتخاب کنیم. برای این کار پس از انتخاب گزینه Open در پنجره طراحی خاصیت ShortCutKeys را پیدا کرده و روی دکمه کلیک کنید. در این صورت یک کادر کوچک شامل چند کادر علامت و یک کادر لیست ترکیبی باز شو مانند شکل ۸-۱۷

نمایش داده می‌شود.




شکل ۱۷-۸

در این کادر روی کادر علامت Ctrl کلیک کرده و بعد گزینه O را از کادر لیست ترکیبی بازشو انتخاب کنید و سپس کلید Enter را بفشارید تا کلید ترکیبی Ctrl+O در روبه‌روی خاصیت ShortCutKeys نمایش داده شود اکنون کلید میانبر تعیین شده در پنجره طراحی نیز روبه‌روی گزینه Open نمایش داده می‌شود.

۷- اکنون روی کنترل منو در زیر پنجره طراحی فرم کلیک کرده و سپس نام آن را روی mnumenu تنظیم کنید.

۸- در این مرحله یک فرم جدید به پروژه اضافه کرده و نام آن را روی عبارت frmpicture تنظیم کنید. سپس یک کنترل کادر تصویر (Picture Box) روی فرم جدید و در مرکز آن قرار دهید و بعد نام آن را روی عبارت Picimage و خاصیت SizeMode آن را روی مقدار AutoSize تنظیم کنید تا ابعاد کنترل با تصویر نمایشی یکسان شود.

نکته: برای تعیین تصویر در کنترل کادر تصویر در زمان طراحی می‌توانید در پنجره خواص و روبه‌روی خاصیت Image روی دکمه  کلیک کنید.

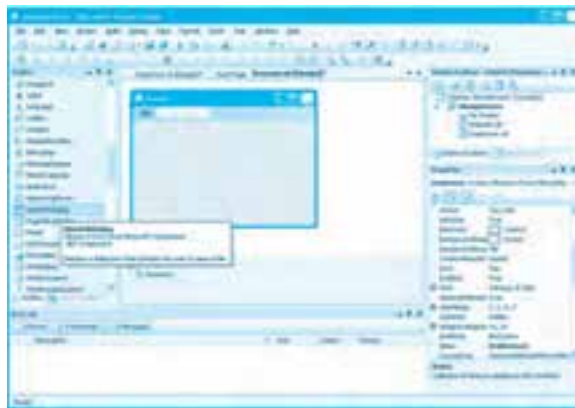
کادرمحاوره‌ای مانند شکل ۱۸-۸ نمایش داده می‌شود. در این کادرمحاوره روی دکمه Import کلیک کنید و تصویر خود را انتخاب نمایید و در پایان روی دکمه OK کلیک کنید تا تصویر در کنترل کادر تصویر نمایش داده شود.

۹- در این مرحله می‌خواهیم یک کنترل OpenFileDialog روی فرم قرار داده شود تا به وسیله آن بتوان فایل‌های تصویر را برای نمایش انتخاب کرد. برای این کار به پنجره طراحی فرم frmmain بروید و در جعبه ابزار ویژوال استودیو روی این کنترل دوبار کلیک کنید (شکل ۱۹-۸) تا کنترل روی فرم قرار بگیرد، سپس نام آن را روی ofdopen تنظیم کنید و خاصیت Filter آن را روی عبارت All File/*.*/Bitmap File/* Bmp نام آن را روی ofdopen تنظیم کنید تا امکان فیلتر کردن فایل‌ها در کادرمحاوره وجود داشته باشد و سپس خاصیت FileName آن

را خالی کنید.



شکل ۸-۱۸



شکل ۸-۱۹

۱۰- اکنون می‌خواهیم رویداد Click گزینه Open در منوی File را به شکلی تنظیم کنیم که وقتی روی این گزینه عمل کلیک انجام شود کادرمحاوره odfoopen نمایش داده شود و پس از انتخاب، فایل تصویر و بسته شدن کادرمحاوره تصویر انتخاب شده روی فرم دوم یعنی frmpicture نمایش داده شود. برای این کار رویداد Click گزینه Open را به این صورت تنظیم کنید.


```
Private Sub mnuopen_Click _
```

```
(ByVal sender As System.Object, ByVal e As System.EventArgs) _
```

```
Handles mnuopen.Click
```

```
Odfopen.FileName= "C:\*. bmp"
```

```
odfopen.ShowDialog()
```

```
If (odfopen.FileName <> " C:\*. bmp" )Then
```

```
Dim myform As New frmpicture
```

```
myform.picimage ImageLocation =Odfopen .Filename
```

```
myform.Show()
```

```
End If
```

```
End Sub
```

در این رویداد ابتدا کادرمحاوره Open با استفاده از متد ShowDialog در این کنترل نمایش داده می‌شود. اگر فایلی در این کادرمحاوره انتخاب شود و روی دکمه Open کلیک، عمل کلیک صورت گیرد نام و مسیر فایل انتخاب شده در خاصیت FileName کنترل odfoopen ذخیره می‌شود، سپس یک دستور If محتویات خاصیت FileName را کنترل می‌کند. اگر مقدار این خاصیت برابر با یک رشته خالی باشد به معنی این است که یا فایل انتخاب نشده است یا در کادرمحاوره روی دکمه Cancel کلیک شده است بنابراین نتیجه بررسی شرط نادرست خواهد بود و تصویری هم نمایش داده نمی‌شود. اما اگر فایلی انتخاب شود و روی دکمه Open کلیک شود نتیجه بررسی شرط درست خواهد بود. بنابراین دستورات موجود در دستور شرطی If اجرا می‌شوند. این دستورات ابتدا یک شیء از کلاس frmpicture ایجاد می‌کنند خاصیت Image Location نام و مسیر فایلی که تصویر انتخاب شده است در اختیار کنترل کادر تصویر قرار می‌گیرد و در پایان فرم با متد Show نمایش داده می‌شود.

نکته: برای نمایش تصویر در کنترل کادر تصویر می‌توان از متد Form File استفاده نمود. به‌عنوان مثال به جای خاصیت Image Location می‌توان به این صورت عمل کرد.

```
My Form. Picimage. Image= System. Drawing. Image. Form File(Odfopen. FileName)
```

۱۱- در این مرحله رویداد گزینه Exit را به این صورت تنظیم کنید:

```
Private Sub mnuexit_Click _
```


```
(ByVal sender As System.Object, ByVal e As System.EventArgs) _
```

```
Handles mnuexit.Click
```

```
Close()
```

End Sub


در این رویداد با استفاده از متد Close فرم بسته می‌شود.


۱۲- پروژه و فرم‌ها را با استفاده از دکمه  ذخیره کنید.

۱۳- پروژه را اجرا کرده و روی منوی File و بعد گزینه Open کلیک کنید. سپس یک فایل انتخاب کرده و روی دکمه Open کلیک کنید. در این صورت یک فرم جدید باز شده و تصویر را داخل آن نمایش می‌دهد.

۱۴- این کار را برای چند تصویر دیگر نیز انجام دهید و نتیجه را بررسی کنید.

۱۵- به اجرای پروژه خاتمه داده و به پنجره ویژوال استودیو بازگردید.

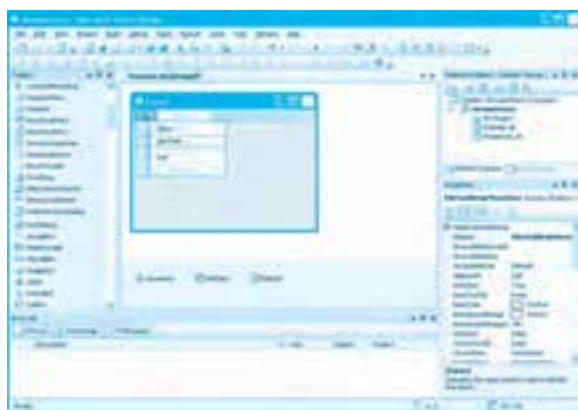
 **تمرین:** پروژه Showpictures را به شکلی تنظیم کنید تا نام و مسیر فایل تصویر در عنوان پنجره‌ای که تصویر را نمایش می‌دهد قابل مشاهده باشد.

 **مثال ۳:** می‌خواهیم پروژه Showpictures را به شکلی تنظیم کنیم تا کاربر بتواند یک مسیر را به عنوان مسیر پیش فرض برای کادرمحاوره Open انتخاب کند. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه ویژوال استودیو را اجرا کرده و پروژه Showpictures را باز کنید.

۲- به پنجره طراحی فرم frmmain بروید سپس در جعبه ابزار، کنترل FolderBrowserDialog را پیدا کنید و بعد با دوبر کلیک، آن را به فرم اضافه کنید و خاصیت نام آن را روی عبارت fbdpath تنظیم کنید. این کنترل می‌تواند منابع ذخیره‌سازی سیستم را مانند درایوها شناسایی کرده و آن‌ها را به صورت یک ساختار درختی نمایش دهد.

۳- در این مرحله به پنجره طراحی فرم رفته و روی نوار منو و گزینه File و سپس عبارت TypeHere در زیر گزینه Exit کلیک کنید و یک گزینه جدید با عنوان ... Set Path & و با نام mnusetpath اضافه کنید و پس از ایجاد آن گزینه جدید را با عمل کشیدن به زیر گزینه Open انتقال دهید تا منوی File مانند شکل ۲۰-۸ تنظیم شود.



شکل ۲۰-۸

۴- اکنون به ماژول کلاس فرم frmmain بروید و در بخش تعاریف کلاس فرم (بعد از تعریف کلاس فرم یعنی دستور Public Class frmmain) یک متغیر رشته‌ای با نام strpath تعریف کنید. این متغیر مسیر پوشه

انتخاب شده در کادرمحاوره Folder Browser را نگهداری می‌کند تا در رویداد Click منوی Open از آن استفاده شود، سپس رویداد Click گزینه Set Path را به این صورت تنظیم کنید:

```
Private Sub mnusetpath_Click _
```

```
(ByVal sender As System.Object, ByVal e As System.EventArgs) _
```

```
Handles mnusetpath.Click
```

```
fbdpath.ShowDialog()
```

```
If (fbdpath.SelectedPath <> "") Then strpath = fbdpath.SelectedPath
```

```
End Sub
```

در این رویداد ابتدا کادرمحاوره Folder Browser با استفاده از متد ShowDialog نمایش داده می‌شود. در این نوع از کادرمحاوره مسیری که از ساختار درختی انتخاب می‌شود در خاصیت Selected Path ذخیره و نگهداری می‌گردد. اگر پوشه‌ای انتخاب شده و روی دکمه OK کلیک شود، مسیر پوشه انتخاب شده در خاصیت Selected Path ذخیره می‌شود. در غیر این صورت یک رشته خالی در این خاصیت ذخیره می‌شود. پس از اجرای دستور اول و بسته شدن کادرمحاوره دستور شرطی If اجرا می‌شود. این دستور مقدار خاصیت SelectedPath را بررسی می‌کند. اگر مقدار این یک رشته خالی نباشد مقدار آن در خاصیت strpath ذخیره می‌شود و در غیر این صورت هیچ تغییری در مقدار خاصیت strpath ایجاد نمی‌شود. به این صورت مسیر پوشه انتخاب شده برای رویدادهای دیگر نیز قابل استفاده خواهد بود.

۵- رویداد Click گزینه Open را به این صورت تنظیم کنید.

```
Private Sub mnuopen_Click _
```

```
(ByVal sender As System.Object, ByVal e As System.EventArgs) _
```

```
Handles mnuopen.Click
```

```
odfopen.FileName = strpath + "\*.*"
```

```
odfopen.ShowDialog()
```

```
If (odfopen.FileName <> "") Then
```

```
Dim myform As New frmpicture
```

```
myform.picimage.Image = System.Drawing.Image.FromFile(odfopen.FileName)
```

```
myform.Show()
```

```
End If
```


```
End Sub
```

در این رویداد یک دستور نسبت به قبل اضافه شده است این دستور مسیر پوشه‌ای را که به وسیله کادرمحاوره Folder Browser انتخاب و در متغیر strpath ذخیره شده است، با رشته "*.*" ترکیب کرده و در خاصیت FileName کادرمحاوره Open قرار می‌دهد تا وقتی کادرمحاوره Open در دستور بعدی باز می‌شود محتویات

همان پوشه انتخاب شده در کادرمحاوره Folder Browser را نمایش دهد.

۶- تغییرات را ذخیره کرده و پروژه را اجرا کنید، سپس روی گزینه Set Path کلیک کرده و یک مسیر دلخواه را انتخاب کنید و بعد روی گزینه Open کلیک کنید. کادرمحاوره Open باز شده و محتویات پوشه‌ای که به وسیله گزینه Set Path انتخاب شده است نمایش داده می‌شود. این کار را چند بار با پوشه‌های دیگر انجام دهید.

۷- اجرای پروژه را خاتمه دهید و به پنجره ویژوال‌استودیو بازگردید.

 تمرین: پروژه Show Pictures را به شکلی تنظیم کنید که وقتی فقط یک درایو به وسیله گزینه Set Path انتخاب می‌شود اجرای دستورات در رویداد Click گزینه Open سبب ایجاد خطا نشوند.

۵-۸ نحوه خطایابی، رفع خطا و مدیریت خطاها در ویژوال بیسیک Net.

انواع خطاها و نحوه خطایابی و خطازدایی در ویژوال بیسیک در خطاهای نوشتاری، منطقی و زمان اجرا مشابه مواردی است که در ویژوال بیسیک ۶ به‌طور کامل بررسی شده‌اند. در ویژوال بیسیک Net نیز مانند ویژوال بیسیک ۶ خطاهای نوشتاری به شکل مناسبی در زمان تایپ دستور گزارش می‌شوند. به‌علاوه با استفاده از گزینه‌های موجود در منوی Debug می‌توان برنامه را در حالت توقف قرار داد و خطا به خط اجرا نمود یا از پنجره اجرای فوری و دستورات Watch استفاده کرد یا نقاط توقف در برنامه به وجود آورد و خطاهای منطقی را کشف و برطرف نمود.

نحوه مدیریت خطاهای زمان اجرا هم به همین شکل است با این تفاوت که برای این کار دو روش می‌توان استفاده کرد. یکی روش استفاده از دستور ON Error GoTo می‌باشد که مشابه ویژوال بیسیک ۶ می‌توان از آن استفاده کرد و روش دوم استفاده از دستور Try ... Catch می‌باشد که در این جا بررسی می‌شود. برای مدیریت خطاهای زمان اجرا در ویژوال بیسیک Net می‌توانید از دستور Try ... Catch به این صورت استفاده کنید:

Try



دستورات

Catch نوع خطا As نام شیء



دستورات مدیریت خطا

finally



دستورات

End Try

در این دستور، دستوراتی که ممکن است در زمان اجرا سبب ایجاد خطا شوند در بخش Try قرار می‌گیرند. اگر این دستورات سبب ایجاد خطا شوند به‌طور خودکار نوع خطا در اولین Catch بررسی می‌شود. اگر خطایی که به‌وجود آمده است از نوع خطایی باشد که در Catch ذکر شده است، دستورات موجود در همان Catch اجرا می‌شود، در غیر این صورت Catch بعدی بررسی می‌شود. استفاده از بخش Finally اختیاری است و دستورات آن همواره اجرا می‌شوند چه خطا رخ بدهد یا خطایی ایجاد نشود.

مثال ۴: می‌خواهیم پروژه showpictures را به شکلی تنظیم کنیم تا در صورت ایجاد هرگونه خطا در هنگام اجرای برنامه بدون آن که اجرای پروژه متوقف شود پیام مناسبی نشان داده شده و کاربر راهنمایی شود. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه ویژوال بیسیک را اجرا کرده و پروژه shopictures را باز کنید.

۲- رویداد Click گزینه Open را به این صورت تنظیم کنید:

```
Private Sub mnuopen_Click _
```

```
(ByVal sender As System.Object, ByVal e As System.EventArgs) _
```

```
Handles mnuopen.Click
```

```
Try
```

```
    odfopen.FileName = strpath + "\*.*"
```

```
    odfopen.ShowDialog()
```

```
    If (odfopen.FileName <> "") Then
```

```
        Dim myform As New frmpicture
```

```
        myform.picimage.Image = System.Drawing.Image.FromFile(odfopen.FileName)
```

```
        myform.Show()
```

```
    End If
```

```
Catch ex As Exception
```

```
    MsgBox(ex.Message, MsgBoxStyle.Critical, "Error")
```

```
End Try
```

```
End Sub
```

در این رویداد دستورات قبلی رویداد داخل دستور Try قرار گرفته‌اند، بنابراین اگر اجرای هر یک از این

دستورات سبب ایجاد خطا شود اجرای برنامه قطع نمی‌شود و به اولین Catch انتقال می‌یابد چون نوع خطایی که در Catch استفاده شده است از نوع Exception می‌باشد. بنابراین هر نوع خطایی که در دستورات بخش Try ایجاد شود با نوع خطایی که در این Catch اعلام شده برابر می‌باشد زیرا کلاس Exception یک کلاس پایه محسوب می‌شود و تمام کلاس‌های خطای دیگر از آن مشتق می‌شوند در نتیجه دستورات این Catch اجرا می‌شوند. در بخش Catch می‌توان با استفاده از نام شی که از کلاس خطا ایجاد می‌شود (ex) پیام خطا را با استفاده از خاصیت Message این شی به دست آورد. در این رویداد در بخش Catch از تابع MsgBox مانند ویژوال بیسیک ۶ برای نمایش پیام خطا استفاده شده است.

۳- تغییرات را ذخیره کرده و پروژه را اجرا کنید، سپس با استفاده از گزینه Open سعی کنید یک فایل متنی را در کنترل کادر تصویر نمایش دهید. در این صورت به جای آن که اجرای پروژه متوقف شود یک پیغام خطا مانند شکل ۲۱-۸ نمایش داده می‌شود. روی دکمه OK کلیک کنید تا اجرای برنامه ادامه یابد.



شکل ۲۱-۸

۴- به اجرای پروژه خاتمه دهید و به پنجره ویژوال استودیو بازگردید.

تمرین: برنامه ماشین حساب را به گونه‌ای تنظیم کنید تا در زمان انجام عمل تقسیم اگر تقسیم بر صفر رخ دهد به جای آن که برنامه متوقف شود پیام مناسبی نمایش داده شود.

۶-۸ نحوه ایجاد و استفاده از آرایه‌ها

نحوه ایجاد و استفاده از انواع آرایه‌های ثابت و پویا در زبان برنامه‌نویسی ویژوال بیسیک Net. مشابه زبان برنامه‌نویسی ویژوال بیسیک ۶ می‌باشد. با این تفاوت که در ویژوال بیسیک Net. اندیس آرایه‌ها از صفر آغاز می‌شود. به عنوان مثال این دستورات یک آرایه دو بعدی را با ۶ سطر و ۴ ستون تعریف و مقداردهی کرده سپس مجموع اعضای آن را محاسبه می‌نماید.

```
Dim no (5, 3) As Integer
```

```
Dim i, j As Integer
```

```
Dim sum As Long = 0
```

```
For i = 0 To 5
```

```
For j = 0 To 3
```

```
no(i, j) = i + 1
```

```
Next j
```

```
Next i
```

```
For i = 0 To 5
```

```
For j = 0 To 3
```

```
sum += no(i, j)
```

```
Next j
```

```
Next i
```

نکته: در زبان برنامه‌نویسی ویژوال بیسیک .Net می‌توانید آرایه‌ها را در زمان تعریف مقداردهی کنید. به عنوان مثال این دستور یک آرایه یک بعدی با چهار عضو را تعریف می‌کند.

```
Dim x () As Single = {1, 2, 4, 6}
```

۷-۸ نحوه ایجاد و استفاده از ساختار یا ساختمان (Structure)

ساختارها را می‌توان نوع ساده و اولیه‌ای از کلاس‌ها دانست که می‌توانند داده‌ها و اطلاعات مربوط به یک موجودیت خاص مثل یک شیء را به صورت یک مجموعه واحد دریافت و ذخیره کرده یا داده‌ها را از آن‌ها بازیابی کنند. البته ساختارها تمام ویژگی‌های کلاس‌ها را شامل نمی‌شوند و مفاهیمی مانند ارث‌بری و پلی‌مورفیسم و ویژگی‌هایی را که از ارث‌بری به وجود می‌آید، پشتیبانی نمی‌کند. ساختارها می‌توانند مانند کلاس شامل اعضای مثل فیلد، خاصیت و متد باشند. برای تعریف یک ساختار در زبان برنامه‌نویسی ویژوال بیسیک .Net می‌توان به این صورت عمل نمود:

نام ساختار Structure [Public یا Private]

```
⋮
```

End Structure.

به عنوان مثال این دستورات یک ساختار را با نام Person برای ذخیره و نگهداری اطلاعات یک شخص ایجاد می‌کنند:

```
Private Structure Person
```

Public FirstName As String

Public LastName As String

Public Age As Byte

End Structure

در این دستورات ساختار Person با سه فیلد داده تعریف شده است، فیلد نام FirstName، نام خانوادگی LastName و سن Age.

به منظور تعریف یک نمونه از ساختار می‌توان به این صورت عمل کرد:

نام ساختار As نام داده [Public یا Private یا Dim]

به عنوان مثال برای ایجاد یک داده از ساختار Person می‌توان به این صورت عمل نمود:

```
Dim myperson As Person
```

پس از تعریف داده از نوع ساختار می‌توان به این صورت به فیلدهای آن دسترسی پیدا کرد:

نام فیلد.نام داده

به عنوان مثال دستورات زیر فیلدهای نام، نام خانوادگی و سن را برای داده myperson مقداردهی می‌کنند.

```
Dim myperson As Person
```

```
myperson.FirstName = "kaveh"
```

```
myperson.LastName = "tehrani"
```

```
myperson.Age = 25
```

۸-۸ نحوه ایجاد و استفاده از نوع داده شمارشی (Enumeration)

گاهی اوقات ممکن است یک فیلد یا خاصیت فقط از دو یا چند مقدار مشخص، یک مقدار را در هر لحظه دریافت کند. در این صورت می‌توان برای آسانی و یکنواخت شدن عملیات مقداردهی به این نوع فیلدها و خاصیت‌ها، فیلد یا خاصیت را از یک نوع داده از نوع شمارشی تعریف کرد. یک نوع داده شمارشی در واقع مجموعه‌ای از دو یا چند عبارت می‌باشد که برای هر یک از آن‌ها یک عدد صحیح مثبت در نظر گرفته می‌شود به عنوان مثال در کلاس Form خاصیتی به نام WindowState وجود دارد. این خاصیت از نوع داده شمارشی به نام FormWindowState که در فضای نام System.Windows.Forms قرار دارد تعریف شده است بنابراین برای مقداردهی به این خاصیت در زمان طراحی یا زمان اجرا فقط می‌توان یکی از سه مقدار تعریف شده در این نوع داده شمارشی را به خاصیت WindowState فرم انتساب داد. به عنوان مثال‌های دیگر

می‌توان به خاصیت‌های `StartPosition` و `FormBorderStyle` در کلاس فرم نیز اشاره نمود.

برای تعریف یک نوع داده شمارشی جدید می‌توان به این صورت عمل نمود:

نوع داده شمارشی Enum [Public یا Private]

عدد صحیح مثبت اول = عبارت اول

عدد صحیح مثبت دوم = عبارت دوم

عدد صحیح مثبت سوم = عبارت سوم

....

....

....

End Enum

مقادیر عددی می‌توانند از هر مقدار عددی صحیح و مثبتی که مورد نظر است، باشند. به طور پیش فرض از مقادیر صفر و ۱ و ۲ و ... استفاده می‌شود.

برای تعریف داده‌ها از نوع داده شمارشی می‌توان به این صورت عمل نمود:

نام نوع داده شمارشی As نام داده [Public یا Private یا Dim]

به عنوان مثال اگر بخواهیم در ساختار `Person` یک فیلد برای تعیین جنسیت شخص ایجاد کنیم با توجه به این که یک شخص می‌تواند مرد یا زن باشد می‌توان یک داده شمارشی به این منظور ایجاد کرده و فیلد جنسیت را از این نوع داده معرفی کرد. برای این کار می‌توان یک نوع داده شمارشی با نام `Genders` به این صورت ایجاد نمود:

`Private Enum Genders`

`Male = 10`

`Female = 20`

`End Enum`

در این نوع داده شمارشی برای عبارت `Male` (مرد) مقدار ۱۰ و عبارت `Female` (زن) مقدار ۲۰ در نظر گرفته شده است. اکنون می‌توان از این نوع داده شمارشی برای تعریف فیلد `Gender` که جنسیت شخص را تعیین می‌کند، استفاده نمود. برای این کار در ساختار `Person` می‌توان یک فیلد جدید به این صورت اضافه کرد:

`Private Structure Person`

`Public FirstName As String`

`Public LastName As String`

Public Age As Byte

Public Gender As Genders

End Sub

در صورت ایجاد داده از ساختار Person فیلد Gender را می‌توان به این صورت مقداردهی نمود:

Dim myperson As Person

myperson.FirstName = "kaveh"

myperson.LastName = "tehrani"

myperson.Age = 25

myperson.Gender = Genders Male

مثال ۵: می‌خواهیم پروژه‌ای طراحی کنیم که به وسیله آن بتوان اطلاعات یک شخص را با استفاده از ساختار Person ذخیره نمود. برای این کار این عملیات را به ترتیب انجام دهید:

۱- برنامه ویژوال استودیو را اجرا کرده و یک پروژه از نوع Windows Forms Application با نام persons ایجاد کنید، سپس نام فرم را روی frmperson و عنوان آن را روی Save Person تنظیم کنید و کنترل‌های آن را مانند شکل ۲۲-۸ روی فرم قرار داده و خاصیت‌های Name و Text را تنظیم کنید.



شکل ۲۲-۸


۲- در این مرحله به ماژول کلاس فرم بروید و داخل کلاس فرم ساختار Person و نوع داده شمارشی Genders و یک داده از نوع ساختار Person با نام myperson را به این صورت تایپ کنید:

```
Public Class frmperson
    Private Enum Genders
        Male = 10
        Female = 20
    End Enum
    Private Structure Person
        Public FirstName As String
        Public LastName As String
        Public Age As Byte
        Public Gender As Genders
    End Structure
    Dim myperson As Person
End Class
```

۳- اکنون می‌خواهیم گزینه‌های Male و Female را داخل کادر لیست ترکیبی بازشو قرار دهیم.



شکل ۲۳-۸

برای این کار در پنجره طراحی فرم روی کنترل کادر لیست بازشو کلیک کنید و در پنجره خواص خاصیت Items را پیدا کرده و روی دکمه  که روبه‌روی آن قرار دارد، کلیک کنید. در این صورت کادر محاوره‌ای مانند شکل ۲۳-۸ نمایش داده می‌شود

سپس در این کادر محاوره عبارت‌های Female و Male را تایپ کنید.

۴- در این مرحله رویداد Load فرم را به این صورت تنظیم کنید تا هنگام نمایش فرم عبارت Female به طور پیش‌فرض در کادر لیست ترکیبی بازشو نمایش داده شود.

```
Private Sub frmperson_Load _
```

```
(ByVal sender As System.Object, ByVal e As System.EventArgs) _
```

```
Handles MyBase.Load
```

```
cmbgender.SelectedIndex = 0
```

```
End Sub
```

۵- در این مرحله رویداد Click دکمه Save را به این صورت تنظیم کنید:

```
Private Sub btnsave_Click _
```

```
(ByVal sender As System.Object, ByVal e As System.EventArgs) _
```

```
Handles btnsave.Click
```

```
myperson.FirstName = txtfirstname.Text
```

```
myperson.LastName = txtlastname.Text
```

```
    myperson Age = Convert.ToByte(txtage.Text)
```

```
    If (cmbgender.SelectedIndex = 0) Then
```

```
        myperson.Gender = Genders.Female
```

```
    Else
```

```
        myperson.Gender = Genders.Male
```

```
    End If
```

```
End Sub
```

در این رویداد محتویات کادرهای متن در فیلدهای مربوطه ذخیره می‌شوند و با توجه به گزینه انتخاب شده در کادر لیست ترکیبی بازشو فیلد Gender روی مقدار مناسب تنظیم می‌شود.

اگر اولین گزینه (Female) در کادر لیست ترکیبی بازشو انتخاب شود، خاصیت SelectedIndex آن مقدار صفر و اگر دومین گزینه (male) انتخاب شود، خاصیت SelectIndex مقدار یک را ارائه می‌کند.

بنابراین دستور شرطی If با توجه به مقدار این خاصیت فیلد Gender را روی مقدار Female یا Male تنظیم می‌کند. به علاوه برای تبدیل مقداری که در کادر متن txtage تایپ می‌شود از متد ToByte که به‌وسیله کلاس Convert ارائه می‌شود، استفاده شده است تا مقدار رشته‌ای را به نوع Byte تبدیل کند.

۶- یک فرم جدید با نام frmshow و عنوان Show Person به پروژه اضافه کنید و کنترل‌های آن را مانند شکل ۸-۲۴ روی فرم قرار دهید.



شکل ۸-۲۴

۷- در این مرحله به پنجره طراحی فرم frmperson بروید و یک دکمه با نام btnshow و عنوان Show روی فرم قرار دهید و رویداد Click آن را به این صورت تنظیم کنید:

Private Sub btnshow_Click _

(ByVal sender As System.Object, ByVal e As System.EventArgs) _

Handles btnshow.Click

Dim myform As New frmshow

myform.lblfirstname.Text = myperson.FirstName

myform.lbllastname.Text = myperson.LastName

myform.lblage.Text = Str(myperson.Age)

If myperson.Gender = Genders Male Then

myform.rbtmale.Checked = True

Else

myform.rbtfemale.Checked = True

End If


myform.Show()

End Sub

در این رویداد محتویات داده myperson یکی یکی خوانده شده و در کنترل‌های برجسب مربوط به خود نمایش داده می‌شوند. برای نمایش جنسیت نیز از دو دکمه انتخاب (مشابه دکمه‌های انتخاب در زبان برنامه‌نویسی ویژوال بیسیک ۶) استفاده شده است و ابتدا یک دستور شرطی If مقدار فیلد Gender را در شیء myperson بررسی می‌کند. اگر مقدار آن برابر با مقدار شمارشی male باشد، خاصیت checked کنترل دکمه انتخاب rbtmale و در غیر این صورت مقدار همین خاصیت در کنترل rbtfemale روی مقدار True تنظیم می‌شود.

۸- پروژه و فرم‌ها را ذخیره کنید و سپس آن‌را اجرا کنید. مشخصات یک شخص دلخواه را در کادرهای متنی وارد کرده و روی دکمه Save کلیک کنید، سپس روی دکمه Show کلیک کنید و درستی عملیات ذخیره‌سازی را بررسی نمایید.

۹- به اجرای پروژه خاتمه داده و به پنجره ویژوال استودیو بازگردید.

 **تمرین:** پروژه‌ای طراحی کنید که با استفاده از یک ساختار اطلاعات یک کتاب را دریافت و ذخیره نموده و در صورت نیاز نمایش دهد. به علاوه برای مقداردهی موضوع کتاب از یک نوع داده شمارشی جدید استفاده شود.

۸-۹ نحوه ایجاد و استفاده از مجموعه‌ها (Collections)

یکی از ویژگی‌های مهم در زبان برنامه‌نویسی NET ارائه مجموعه‌های مختلف و متنوع می‌باشد. با استفاده از مجموعه‌ها کار ذخیره‌سازی و مدیریت و دسترسی به اشیا سریع‌تر و آسان‌تر می‌شود. یکی از پرکاربردترین این مجموعه‌ها، ArrayList است. با استفاده از این مجموعه می‌توان انواع اشیا را از یک نوع یا انواع مختلف به تعداد دلخواه و نامحدود ذخیره و بازیابی کرد. برای دسترسی به کلاس ArrayList و سایر مجموعه‌ها از فضای نام System.Collections استفاده می‌شود. مجموعه‌هایی که از کلاس ArrayList ایجاد می‌شوند، اشیا خود را به صورت یک آرایه پویا ذخیره و مدیریت می‌کنند. برای اضافه کردن یک شیء به داخل این نوع از مجموعه‌ها از متد Add و برای حذف اشیا نیز از متد Remove یا RemoveAt استفاده می‌شود.

برای تعریف یک مجموعه از نوع ArrayList می‌توان به این صورت عمل نمود:

`As New System.Collections.ArrayList()` نام مجموعه [Dim یا Private یا Public]

به عنوان مثال این دستور یک مجموعه با نام colobjects ایجاد می‌کند.

`Dim colobjects As New System.Collections.ArrayList()`

مثال ۶: می‌خواهیم پروژه persons را به شکلی تنظیم کنیم تا مشخصات هر تعدادی از اشخاص را بدون محدودیت دریافت کند. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه ویژوال استودیو را اجرا کرده و پروژه persons را باز کنید. سپس این دستور را داخل کلاس فرم در بخش تعاریف کلاس (Declaration) قرار دهید. این دستور یک مجموعه از نوع ArrayList با نام colpersons ایجاد می‌کند.

`Private colpersons As New System.Collections.ArrayList`

۲- در رویداد Click دکمه Save این دستور را به عنوان آخرین دستور تایپ کنید تا وقتی مشخصات شیء myperson کامل می‌شود آن را به مجموعه colpersons اضافه کند. برای این کار از متد Add استفاده می‌شود:

`colpersons.Add(myperson)`

۳- اکنون رویداد Click دکمه Show را به این صورت تنظیم کنید:

`Private Sub btnshow_Click _`

`(ByVal sender As System.Object, ByVal e As System.EventArgs) _`

`Handles btnshow.Click`

Dim myform As New frmshow

myform.Show()

Dim item As Person

For Each item In colpersons

myform.lblfirstname.Text = item.FirstName

myform.lbllastname.Text = item.LastName

myform.lblage.Text = Str(item.Age)

If item.Gender = Genders.Male Then

myform.rbtmale.Checked = True

Else

myform.rbtfemale.Checked = True

End If

MsgBox("Next")

Next

End Sub

در این رویداد ابتدا یک نمونه از کلاس فرم frmshow نمایش داده می‌شود. سپس با استفاده از یک حلقه Foreach مجموعه پویش می‌شود و این حلقه به طور خودکار از اولین شیء در مجموعه شروع می‌کند و در اجرای دوباره دومین حلقه شیء را مورد دسترسی قرار می‌دهد تا تمام اشیا در مجموعه مورد دسترسی قرار گیرد. با هر بار اجرای حلقه For Each مشخصات یک شخص و بعد یک کادر پیغام نمایش داده می‌شود. اگر روی دکمه OK در کادر پیغام کلیک شود حلقه دوباره اجرا شده و مشخصات نفر بعد در مجموعه نمایش داده می‌شود و به همین ترتیب الی آخر.

۴- تغییرات را ذخیره کنید و پروژه را اجرا کرده و اطلاعات تعدادی از اشخاص را ثبت کنید سپس روی دکمه Show کلیک کنید و نتیجه را بررسی نمایید.

۵- به اجرای پروژه خاتمه دهید و به پنجره ویژوال استودیو بازگردید.

تمرین: پروژه کتابخانه را به شکلی تنظیم کنید تا بتوان تعدادی کتاب را ذخیره کرد. به علاوه



باید بتوان اطلاعات مربوط به یک کتاب را در مجموعه کتاب‌های ثبت شده جستجو نمود.

۱۰-۸ نحوه اتصال به پایگاه‌های داده و دسترسی به داده

زبان برنامه‌نویسی ویژوال بیسیک .Net امکان استفاده از ADO.Net برای اتصال و بازیابی داده و اطلاعات را فراهم می‌کند. ADO.Net با ساختار جدیدی که نسبت به ADO در زبان برنامه‌نویسی ویژوال بیسیک ۷ ارائه شده است، انجام هرگونه عملیات روی پایگاه‌های داده از انواع مختلف را عملی می‌کند.

ADO.Net شامل کلاس‌های مختلفی به منظور اتصال و انجام عملیات روی پایگاه‌های داده می‌باشد که عبارتند از:

- **کلاس Connection:** با استفاده از این کلاس می‌توان پارامترهای اتصال به پایگاه داده موردنظر را تنظیم کرد و به پایگاه داده متصل شد.
- **کلاس Command:** این کلاس امکان اجرای دستورات SQL را روی پایگاه‌های داده فراهم می‌کند.
- **کلاس DataAdapter:** این کلاس با ارائه خاصیت‌ها، متدها و رویدادهای مختلف امکان اجرای دستورات SQL و انجام عملیات را روی پایگاه داده‌ها فراهم می‌کند.
- **کلاس DataSet:** این کلاس نیز با ارائه خاصیت‌ها، متدها و رویدادهای مناسب امکان دسترسی به داده‌هایی را که بازیابی شده‌اند، فراهم می‌کند. این کلاس یک مجموعه از جدول‌ها به نام Tables را برای نگهداری داده‌های بازیابی شده ارائه می‌کند. این مجموعه می‌تواند یک یا چند شیء جدول را دریافت و نگهداری کند. هر جدول نیز می‌تواند نتایج حاصل از یک پرس‌وجو (Query) را در خود ذخیره و نگهداری نماید. در نتیجه امکان انتقال داده‌ها و اطلاعات از پایگاه داده به برنامه فراهم می‌شود.
- **کلاس DataTable:** این کلاس نیز امکان ذخیره‌سازی اطلاعات یک جدول را فراهم می‌کند. به عبارت دیگر این کلاس مجموعه‌ای برای ذخیره‌سازی سطرها و مجموعه دیگری برای ستون‌ها را شامل می‌شود. نام و سایر مشخصات ستون‌ها در مجموعه ستون‌ها و محتویات سطرها و داده و اطلاعات موجود در آن‌ها در مجموعه سطرها قرار می‌گیرد. اشیای جدول که در مجموعه Tables در کلاس DataSet قرار می‌گیرند از این کلاس ایجاد می‌شوند.

نکته: در ADO.Net برای اتصال به پایگاه‌های داده مختلف کلاس‌های مستقلی ارائه می‌شود. به عنوان مثال برای اتصال به یک پایگاه داده از نوع Access از کلاس **OLEDBConnection** و برای اتصال به یک پایگاه داده از نوع **SQLConnection** استفاده می‌شود. ارائه شده است.

۱-۱۰-۸ نحوه اتصال و بازیابی داده‌ها و اطلاعات با استفاده از کلاس **DataAdapter** در زبان برنامه‌نویسی ویژوال بیسیک **Net**. یک کنترل با نام **OleDbDataAdapter** ارائه شده است. به وسیله این کنترل می‌توان داده و اطلاعات را از پایگاه داده‌ها بازیابی نمود. برای این کار می‌توانید این کنترل را در جعبه ابزار پیدا کنید (شکل ۸-۲۵) و یک نمونه از آن را روی فرم قرار دهید. در این صورت یک کادر محاوره مانند شکل ۸-۲۶ نمایش داده می‌شود. در این شکل یک کادر لیست ترکیبی باز شو وجود دارد که به وسیله آن می‌توان از اتصال‌هایی که قبلاً ایجاد شده‌اند، استفاده کرد. به علاوه می‌توان با کلیک روی دکمه **New Connection...** یک اتصال جدید به وجود آورد. روی این دکمه کلیک کنید تا به کادر محاوره **Add Connection** بروید (شکل ۸-۲۷).



شکل ۸-۲۵



شکل ۸-۲۶



شکل ۸-۲۷

در این کادر محاوره می‌توان در کادر لیست Data source نوع پایگاه داده را انتخاب نمود. در این کادر محاوره به طور پیش‌فرض منبع داده از نوع SQL Server انتخاب شده است. روی دکمه ... Change کلیک کنید تا کادر محاوره دیگری مانند شکل ۸-۲۸ نمایش داده شود.



شکل ۸-۲۸

روی گزینه Microsoft Access Database File و بعد روی دکمه OK کلیک کنید تا به کادر محاوره Add Connection بازگردید. اکنون این کادر محاوره مانند شکل ۸-۲۹ نمایش داده می‌شود.



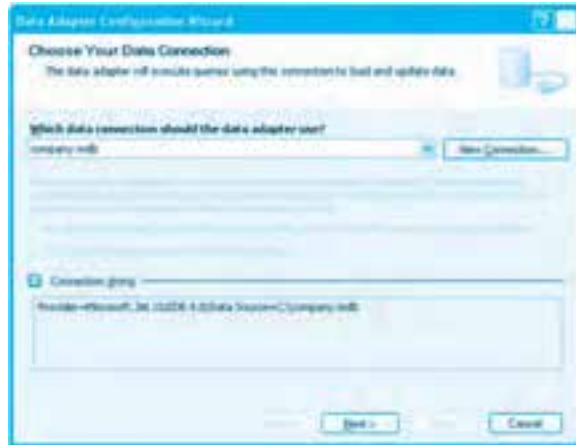
شکل ۲۹-۸

اکنون در این کادر محاوره در کادر متن Database file name می‌توانید نام و مسیر پایگاه داده خود را تایپ کنید یا با استفاده از دکمه Browse ... پایگاه داده موردنظر را پیدا کرده و آنرا انتخاب کنید. سپس با کلیک روی دکمه Test Connection اتصال ایجاد شده را بررسی کنید. در صورتی که یک کادر پیغام مانند شکل ۸-۳۰ نمایش داده شود، اتصال درست می‌باشد.



شکل ۳۰-۸

به عنوان آخرین عمل در این کادر محاوره روی دکمه OK کلیک کنید. در این مرحله به اولین کادر محاوره بازمی‌گردید. اکنون می‌توانید با کلیک روی علامت + در کنار عبارت Connection string رشته‌ای که پارامترهای اتصال به پایگاه داده را مشخص می‌کند، نمایش دهید (شکل ۸-۳۱) به علاوه نام پایگاه داده نیز در کادر متن نمایش داده می‌شود.



شکل ۸-۳۱

در این مرحله روی دکمه Next کلیک کنید، در این صورت کادر پیغامی مانند شکل ۸-۳۲ نمایش داده می‌شود.



شکل ۸-۳۲



شکل ۸-۳۳

در این کادر پیغام در رابطه با این که آیا تمایل دارید یک کپی از پایگاه داده تهیه شود یا از همان پایگاه داده

انتخاب شده در مسیر مربوط استفاده شود، سؤال پرسیده می‌شود. روی دکمه No کلیک کنید. در این صورت کادر محاوره بعدی مانند شکل ۸-۳۳ نمایش داده می‌شود. در این کادر محاوره فقط یک گزینه قابل انتخاب می‌باشد، بنابراین دوباره روی دکمه Next کلیک کنید تا به مرحله بعد بروید. اکنون کادر محاوره‌ای مانند شکل ۸-۳۴ نمایش داده می‌شود.



شکل ۸-۳۴

در کادر متن این کادر محاوره می‌توانید دستور Select مورد نظر خود را برای بازیابی داده‌ها تایپ نمایید. البته با کلیک روی دکمه ... Query Builder می‌توانید به پنجره طراحی پرس‌وجو (Query) بروید و با این رابط گرافیکی پرس‌وجوی خود را سریع و آسان طراحی کنید.



شکل ۸-۳۵

اگر در این کادر محاوره روی دکمه ... Advanced Options کلیک کنید، کادر محاوره‌ای مانند شکل ۸-۳۵

نمایش داده می‌شود. اگر اولین کادر علامت را در این کادر محاوره از حالت انتخاب خارج کنید، دستورات Insert، Update و Delete تولید نمی‌شوند. پس از تعیین دستور Select می‌توانید روی دکمه Next کلیک کنید تا به مرحله بعد بروید. در این صورت کادر محاوره‌ای مانند شکل ۸-۳۶ نمایش داده می‌شود.




شکل ۸-۳۶

اکنون عملیات پیکربندی و تنظیم کنترل OleDbDataAdapter خاتمه یافته است. روی دکمه Finish کلیک کنید تا به پنجره ویژوال استودیو بازگردید. در این صورت یک کنترل OleDbDataAdapter و یک کنترل OleDbConnection ایجاد شده و در زیر پنجره طراحی فرم نمایش داده می‌شوند (شکل ۸-۳۷).



شکل ۸-۳۷

در این مرحله می‌توان به وسیله کنترل OleDbDataAdapter یک کنترل DataSet برای نگهداری داده‌هایی که بازیابی می‌شوند، ایجاد نمود. برای این کار روی علامت  کلیک کنید و در کادر لیستی که نمایش داده می‌شود (شکل ۸-۳۸) روی گزینه Generate DataSet ... کلیک کنید.




شکل ۸-۳۸

کادر محاوره‌ای مانند شکل ۸-۳۹ نمایش داده می‌شود؛ در این کادر محاوره می‌توانید نام DataSet را در کادر متن New تایپ کرده و سپس روی دکمه OK کلیک کنید.



شکل ۸-۳۹

در این مرحله یک کنترل DataSet نیز ایجاد شده و در زیر پنجره طراحی فرم نمایش داده می‌شود. اکنون می‌توان کنترل‌های کادر متن، کادر لیست، کادر لیست ترکیبی بازشو یا کنترل DataGridView را روی فرم قرار داد و به منبع داده متصل نمود.

• برای ایجاد ارتباط کنترل‌های ساده مانند کنترل کادر متن یا کادر علامت با داده‌ها می‌توان از خاصیت DataBindings استفاده نمود. برای این کار روی علامت + در کنار این خاصیت کلیک کنید تا ساختار درختی مربوط به این خاصیت باز شود، سپس در این ساختار درختی روی دکمه  که روبه‌روی خاصیت Text است، کلیک کنید. یک ساختار درختی دیگر باز می‌شود (شکل ۸-۴۰).



شکل ۴۰-۸

روی علامت + در سمت چپ عبارت Other Data Source کلیک کنید و به همین صورت ساختار درختی Form1 List Instances و بعد نام DataSet11 و نام جدول را باز کرده و از لیست نام ستون‌ها یک ستون را انتخاب کنید تا کنترل به آن ستون از جدول انتخاب شده مرتبط شود.

- برای ایجاد ارتباط کنترل‌هایی مانند کادر لیست (ListBox) و کادر لیست ترکیبی بازشو (ComboBox) منبع داده با داده‌ها باید خاصیت‌های DataSource و DisplayMember را به ترتیب روی نام Dataset و نام ستون موردنظر تنظیم نمود.

- برای ایجاد ارتباط کنترل DataGridView با منبع داده باید خاصیت‌های DataSource و DataMember را به ترتیب روی نام DataSet و نام جدول موردنظر تنظیم کرد.

در آخرین مرحله باید با استفاده از متد Fill در کنترل OleDbDataAdapter داده‌ها را بازیابی کرد و به کنترل DataSet منتقل نمود. نحوه استفاده از این متد به این صورت است:

(نام Fill (Data set) . نام کنترل OleDbDataAdapter


مثال ۷: می‌خواهیم پروژه‌ای طراحی کنیم تا به‌وسیله آن بتوان اطلاعات کارمندان را در جدول employees از پایگاه داده Company مشاهده نمود.

برای انجام این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه ویژوال استودیو را اجرا کرده و یک پروژه از نوع Windows Forms Application با نام employees ایجاد کنید.

۲- یک کنترل OleDbDataAdapter با نام oldbemployees روی فرم قرار داده و آن را به پایگاه داده company متصل کنید و این دستور را برای بازیابی داده‌ها در بخش طراحی پرس‌وجو (Query) تایپ کنید.

```
Select * from employees
```

۳- پس از پیکربندی کنترل oldbemployees روی علامت  و بعد روی گزینه Generate DataSet کلیک کنید و یک DataSet با نام dsemployees ایجاد کنید.

۴- پس از ایجاد DataSet یک کنترل DataGridView با نام dgvemployees و کنترل دکمه با نام btnfill و عنوان Fill روی فرم قرار دهید (شکل ۴۱-۸)، سپس خاصیت DataSource و DataMember کنترل dgvemployees را به ترتیب روی dsemployees1 و جدول employees تنظیم کنید.



شکل ۴۱-۸

۵- در این مرحله رویداد Click دکمه Fill را به این صورت تنظیم کنید:

```
Private Sub btnfill_Click _
```

```
(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnfill.Click
```

```
oldbemployees.Fill(Dsemployees1)
```

```
End Sub
```

در این رویداد با استفاده از متد Fill کنترل oldbemployees دستور Select را اجرا کنید تا سطرهای بازبایی شده، داخل کنترل Dsemployees انتقال یابند و چون کنترل dgvemployees به کنترل DataSet و جدول employees مرتبط شده است، سطرها را نمایش می‌دهد.

۶- تغییرات را ذخیره کنید و پروژه را اجرا نمایید، سپس روی دکمه Fill کلیک کنید. در این صورت اطلاعات کارمندان مانند شکل ۴۲-۸ در کنترل DataGridView نمایش داده می‌شوند.

در صورتی که بخواهید سطرها را بر اساس شرط یا شرایط خاصی فیلتر کنید، می‌توانید از خاصیت SelectCommand کنترل OleDbDataAdapter استفاده کنید. به عنوان مثال می‌خواهیم پروژه employees را به شکلی تنظیم کنیم تا با توجه به نام شهر، محل تولد اسامی کارمندان نمایش داده شود. برای این کار در این پروژه روی فرم یک کنترل برچسب و کادر متن در کنار دکمه Fill قرار دهید و نام آن را روی txtcity تنظیم کنید، سپس رویداد Click دکمه Fill را به این صورت تنظیم کنید:



شکل ۴۲-۸

Private Sub btnfill_Click_

(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnfill.Click

oldbemployees.SelectCommand.CommandText = _

"select * from employees where city=" + txtcity.Text + ""


oldbemployees.Fill(Dsemployees1)

End Sub

در این رویداد با استفاده از خاصیت CommandText در خاصیت SelectCommand از کنترل OleDbDataAdapter دستور Select به شکلی تنظیم می‌شود تا نام شهری که در کادر متن txtcity تایپ می‌شود در شرط بخش Where این دستور قرار گیرد. بنابراین وقتی متد Fill اجرا می‌شود سطرهایی را بازیابی می‌کند که شرط بخش Where در دستور Select برای آن‌ها برقرار باشد. اگر تغییرات را ذخیره و پروژه را اجرا کنید و نام شهر دلخواهی را در کادر متن تایپ کرده و روی دکمه Fill کلیک کنید اسامی و مشخصات کارمندان متولد شهر موردنظر نمایش داده می‌شوند (شکل ۴۳-۸).



شکل ۴۳-۸

 **تمرین:** پروژه employees را به شکلی تنظیم کنید تا به‌وسیله آن بتوان اطلاعات کارمندان را بر اساس نام و نام خانوادگی آن‌ها جستجو کرد.

۲-۱۰-۸- حذف و به‌روزرسانی داده‌ها و اطلاعات

با استفاده از کلاس‌های موجود در Net می‌توان سطرها را در جدول‌های یک پایگاه داده درج، حذف و ویرایش کرد. برای این کار می‌توانید از کلاس‌های Connection و Command استفاده کنید. برای درج سطرها در یک جدول از دستور INSERT استفاده می‌شود. شکل کلی نحوه استفاده از این دستور به این صورت است:

INSERT INTO Values (نام ستون‌ها) نام جدول

به عنوان مثال این دستور یک سطر را در جدول employees پایگاه داده Company درج می‌کند.

```
INSERT INTO employees ( employeeid, firstname, lastname, city, salary )
```

```
VALUES (871123, 'arash', 'rostamee', 'tehran', 5000000)
```

برای ویرایش یک یا مجموعه‌ای از سطرها و محتویات ستون‌های آن می‌توان از دستور UPDATE استفاده کرد شکل کلی نحوه استفاده از این دستور به این صورت است:

... مقدار جدید = نام ستون SET نام جدول UPDATE

WHERE (ها) شرط

به عنوان مثال این دستور نام و نام خانوادگی کارمند با کد کارمندی ۸۷۴۲۱۳ را به alireza و tehrani تغییر می‌دهد.

```
UPDATE employees SET firstname='alireza', lastname='tehrani' where employeeid=874213
```

برای حذف یک یا مجموعه‌ای از سطرها می‌توان از دستور DELETE استفاده کرد. شکل کلی نحوه استفاده از این دستور به این صورت است:

 **نکته:** اگر از بخش WHERE در دستور UPDATE استفاده نشود تمام سطرها ویرایش و

به‌روز می‌شوند.

WHERE (ها) شرط DELETE * FROM نام جدول

به عنوان مثال این دستور کارمند با کد کارمندی ۸۷۱۰۲۱ را از جدول employees حذف می‌کند.

```
DELETE * FROM employees where employeeid=871021
```

 **نکته:** اگر بخش WHERE در دستور DELETE حذف شود، تمام سطرها حذف می‌شوند.

مثال ۸: می‌خواهیم پروژه‌ای طراحی و ایجاد کنیم که به وسیله آن بتوان داده‌ها و اطلاعات را در جدول employees درج کرد. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه ویژوال‌استودیو را اجرا کرده و یک پروژه از نوع Windows Forms Application با نام Sinsertemployee همراه با یک فرم مانند شکل ۸-۴۴ ایجاد کنید.

شکل ۸-۴۴

۲- رویداد Click دکمه Insert را به این صورت تنظیم کنید.

```
Private Sub btnsave_Click _
```

```
(ByVal sender As System.Object, ByVal e As System.EventArgs) _
```

```
Handles btnsave.Click
```

```
Dim mycommand As New System.Data.OleDb.OleDbCommand
```

```
Dim myconnection As New System.Data.OleDb.OleDbConnection
```

```
myconnection.ConnectionString= _
```

```
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\company.mdb"
```

```
mycommand.CommandText= _
```

```
"INSERT INTO employees(employeeID, firstname,lastname, city,salary)" + _
```

```
"VALUES (" + txtid.Text + "," + txtfirstname.Text + "," + _
```

```
txtlastname.Text + "," + txtcity.Text + "," + txtsalary.Text + ")"
```

```
mycommand.CommandType = CommandType.Text
```

```
mycommand.Connection = myconnection
```

```
myconnection.Open()
```

```
mycommand.ExecuteNonQuery()
```

```
myconnection.Close()
```

End Sub

در این دستورات ابتدا از کلاس‌های OleDbCommand و OleDbConnection دو شیء برای ایجاد اتصال و اجرای دستور INSERT ایجاد می‌شوند، سپس خاصیتConnectionString شیء myconnection تنظیم می‌شود تا این شیء بتواند به پایگاه داده company که از نوع Access می‌باشد متصل شود و بعد خاصیت CommandText شیء Command یعنی mycommand تنظیم می‌شود این خاصیت می‌تواند دستورات SQL را در برگیرد. در این جا برای آن که بتوان یک سطر را درج کرد از دستور INSERT استفاده شده است. چون خاصیت CommandText یک دستور SQL است که به صورت یک رشته تنظیم می‌شود بنابراین لازم است خاصیت CommandType شیء mycommand روی مقدار CommandType.Text تنظیم شود. در مرحله بعد شیء myconnection در خاصیت Connection شیء mycommand قرار داده می‌شود تا شیء mycommand بدانند باید روی کدام اتصال و پایگاه داده اجرا شود، سپس شیء myconnection با اجرای متد Open باز می‌شود و دستور INSERT به وسیله متد ExecuteNonQuery اجرا می‌شود و یک سطر در جدول employees درج می‌کند و در پایان با خاتمه عملیات، شیء myconnection با متد Close بسته می‌شود.

۳- تغییرات را ذخیره کنید و پروژه را اجرا کرده و اطلاعات یک کارمند جدید را در کادرهای متن تایپ کنید، سپس روی دکمه Insert کلیک کنید تا یک سطر در جدول employees درج شود و نتیجه را در جدول بررسی کنید.

۴- به اجرای پروژه خاتمه داده و به پنجره ویژوال استودیو بازگردید.

تمرین: پروژه‌های طراحی کنید که بتوان اطلاعات یک کارمند دلخواه را ویرایش کرد یا حذف نمود.



Learn in English

Create a Project in Visual Basic

For your first Visual Basic program, you will create a Web-browsing application that opens a Web page. Your first step in creating a Visual Basic program is to open Visual Studio and create a project. You will do this when you create any Visual Basic program.

To create a project for your program

1- From the Windows Start menu, click Visual Basic 2008.

The «Welcome to Visual Basic» screen appears. This is the interface for Visual Basic 2008, also known as the integrated development environment or IDE.

2- On the File menu, click New Project.

The New Project dialog box opens.

3- Select Windows Forms Application and click OK.

A new form displays in the IDE, and the necessary files for your project are added to the Solution Explorer window. If this is the first Windows Forms Application project that you have created, it is named «WindowsApplication1».

Create a User Interface

It is now time to start to build the user interface (the visible part that users with) by adding controls from the Toolbox to the form.

The Toolbox is on the left side of Visual Studio and consists of several tabs such as Data, Components, and All Windows Forms. Inside each tab is a set of entries that represent controls or components that you can add to your application. For example, the All Windows Forms tab has entries named TextBox, Button, and CheckBox that represent controls that you can add to your application by dragging them onto the form.

Add Visual Basic Code

In the previous paragraph, you used the Properties window to the properties of the controls on a form. In this paragraph, you will add the code that will control your program's functions.

To add code and functionality to your program

In the Form Designer, double-click on the control.

A new window called the Code Editor opens. This is where you add all the code for your program.

In the Code Editor, type the your code.

Run and Test Your Program

Now that your program is complete, it is time to run and test it. For complex programs, testing can be a long and difficult process. To run your program, On the Debug menu of the Visual

Basic IDE, click Start Debugging. This command runs your program.

واژه‌نامه

Collection	مجموعه
Common	مشترک
Configure	پیکربندی کردن
Complex	مجموعه پیچیده، مختلط
Declaration	اعلام، تعریف
Development	پیشرفت، توسعه
Enumeration	شمارشی
Exception	استثنا
Explorer	مرورگر
Framework	چهارچوب
Intermediate	میانی
Interact	متقابلاً عمل کردن
NameSpace	فضای نام
Necessary	واجب
Recent	اخیر
Represent	نمایش دادن
Structure	ساختار
Template	الگو، قالب

خلاصه مطالب

- Net Framework یک چارچوب و قالب است که برای طراحی، ساخت و اجرای برنامه‌ها به زبان‌های برنامه‌نویسی موجود در آن به کار می‌رود.
- Net Framework از بخش‌های مختلفی تشکیل می‌شود که مهم‌ترین آن‌ها عبارتند از: کتابخانه کلاس‌های Net، CLB و زبان‌های برنامه‌نویسی ASP.NET، VC++ Net، C# Net، VB.Net
- کتابخانه کلاس‌های Net، شامل مجموعه‌ای از کلاس‌های متنوع و مختلف می‌باشد که امکان ایجاد اجزای تشکیل دهنده یک برنامه را سریع و آسان می‌کند.
- زبان اجرایی مشترک بخشی از Net است که وظیفه مدیریت حافظه و اجرای برنامه‌ها را به عهده دارد.
- برنامه‌ها در Net پس از اجرا بلافاصله به کد زبانی میانی (IL) تبدیل می‌شوند و سپس توسط CLR اجرا می‌شوند.
- به‌طور کلی داده‌ها در Net به دو گروه داده‌ها از نوع مقدار و از نوع مرجع تقسیم می‌شوند.
- از کنترل OpenFileDialog برای ایجاد کادر محاوره باز کردن فایل می‌توان استفاده کرد.
- خاصیت Filter در کنترل OpenFileDialog امکان فیلتر کردن فایل‌ها بر اساس نوع آن‌ها را فراهم می‌کند.
- با استفاده از خاصیت FileName در کنترل OpenFileDialog می‌توان مسیر و نام فایل انتخاب شده را به‌دست آورد.
- از کنترل FolderBrowser برای تعیین و انتخاب یک پوشه استفاده می‌شود.
- با استفاده از خاصیت SelectedPath در کنترل FolderBrowser می‌توان مسیر پوشه انتخاب شده را به‌دست آورد.
- با استفاده از دستور Try...Catch می‌توان خطاهای زمان اجرا را مدیریت کرد.
- از کلاس Exception می‌توان برای شناسایی کلیه خطاها در دستور Try...Catch استفاده کرد.
- از ساختارها برای ایجاد ترتیبی از داده‌های ساده برای ذخیره‌سازی و نگهداری داده‌های مربوط به یک شیء مشخص استفاده می‌شود.
- از داده‌های شمارشی می‌توان مقادیری را از پیش آماده کرد تا در مقداردهی یک خاصیت یا متغیر مورد استفاده قرار گیرند.
- برای تعریف یک داده از نوع ساختار یا نوع داده شمارشی می‌توان از همان روشی که در داده‌های نوع مقدار به کار می‌رود، استفاده نمود.
- با استفاده از مجموعه‌ها، ذخیره‌سازی، مدیریت و دسترسی به اشیاء سریع‌تر و آسان‌تر می‌شود.
- با استفاده از کلاس ArrayList در Net، می‌توان مجموعه‌ای از انواع اشیاء را ایجاد و استفاده کرد.

- مجموعه‌هایی که با استفاده از کلاس ArrayList ایجاد می‌شوند، می‌توانند اشیا را به صورت یک آرایه پویا در خود ذخیره و مدیریت کنند.
- برای اضافه کردن اشیا به یک مجموعه از نوع ArrayList می‌توانید از متد Add استفاده کنید.
- ADO Net امکان اتصال و انجام عملیات بازیابی، درج، حذف و به‌روزرسانی را در پایگاه‌های داده از انواع مختلف فراهم می‌کند.
- ADO Net شامل کلاس‌های Connection، Command، DataAdapter، DataSet و DataTable می‌باشد.
- با استفاده از متد Fill در کنترل OleDbDataAdapter می‌توان یک Query را روی پایگاه داده موردنظر اجرا کرد و نتایج حاصل را در یک کنترل DataSet قرار داد.
- برای ایجاد ارتباط کنترل‌های ساده مانند کادر متن می‌توان از خاصیت Text که به‌وسیله خاصیت DataBindings ارائه می‌شود، استفاده نمود.
- برای ایجاد ارتباط کنترل‌هایی مانند کادر لیست و کادر لیست ترکیبی باز شو از خاصیت‌های DataSource و DataMember استفاده می‌شود.
- برای ایجاد ارتباط کنترل DataGridView می‌توان از خاصیت‌های DataSource و DataMember استفاده نمود.
- برای درج، حذف یا به‌روزرسانی سطرها می‌توانید از دستورات INSERT، DELETE یا UPDATE استفاده کنید.
- با ایجاد اشیا از کلاس‌های OleDbConnection و OleDbCommand می‌توان سطرها را در یک جدول درج، حذف یا ویرایش نمود.

آزمون نظری

۱- کدام خاصیت در کنترل کادر لیست می‌تواند ستونی را که کنترل با آن مرتبط می‌شود، مشخص کند؟

الف - DataSet ب - DataSource ج - Display Member د - DataMember

۲- با استفاده از کدام متد در کنترل OleDbDataAdapter می‌توان داده‌ها را از پایگاه داده‌ها بازیابی نمود؟

الف - Fill ب - Select ج - Open د - SqlConnectionString

۳- کدام خاصیت در کنترل DataGridView می‌تواند نام جدولی را که کنترل به آن مرتبط می‌شود، مشخص کند؟

الف - DataMember ب - DataSource

ج - DataField د - DisplayMember

۴- کدام کلاس در ADO.Net برای ذخیره‌سازی و نگهداری سطرهایی که از یک جدول بازیابی می‌شوند، مناسب‌تر است؟

الف - Connection ب - Command ج - Data Source د - Data Table

۵- کدام کلاس می‌تواند یک کلکسیون با تعداد اشیای نامحدود را ایجاد کند؟

الف - Array ب - ArrayList ج - Stack د - Tree

۶- با استفاده از کدام خاصیت در کنترل FolderBrowser می‌توان نام و مسیر پوشه انتخاب شده را به دست آورد؟

الف - Path ب - FileName ج - SelectedPath د - SelectedFolder

۷- کدام بخش در .Net وظیفه اجرای برنامه‌ها و مدیریت حافظه را به عهده دارد؟

الف - کتابخانه کلاس‌ها ب - CLR

ج - اسمبلی د - IL

۸- برای ایجاد نوع داده شمارشی از دستور استفاده می‌شود.

الف - enum ب - Structure ج - enumeration د - struct

۹- کلاس شامل یک کلکسیون از می‌باشد که برای نگهداری داده‌های بازیابی شده استفاده می‌شوند (از راست به چپ).

الف - Command, DataSet ب - Command, DataTable

ج - DataSet, DataTable د - DataTable, DataSet

۱۰- در .Net یک برنامه بلافاصله پس از اجرا (برای اولین بار)

- الف- در اختیار CLR قرار می‌گیرد تا دستورات آن را اجرا کند.
- ب- به کد زبان میانی (IL) تبدیل می‌شود.
- ج- اگر قبلاً ترجمه شده باشد، اجرا می‌شود.
- د- اگر قبلاً ترجمه شده باشد، در اختیار CLR قرار می‌گیرد.

11- To Run Program On The Debug menu Of the visual Basic IDE, Click ...

- a- Start b- Debug c- Start Debugging d- Debugging

- ۱۲- تکنولوژی NET. را توضیح دهید و اجزای آن را نام برده و هر یک را به طور مختصر بیان کنید.
- ۱۳- نحوه استفاده از کنترل OpenFileDialog و خاصیت‌های آن را توضیح دهید.
- ۱۴- کاربرد و نحوه استفاده از کنترل FolderBrowser را توضیح دهید.
- ۱۵- نحوه استفاده از دستور Try...Catch برای مدیریت خطاهای زمان اجرا را توضیح دهید.
- ۱۶- مفهوم ساختار را توضیح دهید و نحوه ایجاد و استفاده از آن را بیان کنید.
- ۱۷- مفهوم داده شمارشی (enum) را توضیح دهید و نحوه ایجاد و استفاده از آن را بیان کنید.
- ۱۸- مفهوم مجموعه را بیان کنید و نحوه ایجاد مجموعه از کلاس ArrayList را توضیح دهید.
- ۱۹- ADO.Net و تفاوت‌های آن با ADO را بیان کنید. به علاوه کلاس‌های ارائه شده در آن را همراه با وظایفشان توضیح دهید.
- ۲۰- نحوه استفاده و پیکربندی کنترل OleDbDataAdapter را توضیح دهید.
- ۲۱- نحوه استفاده از کلاس‌های Connection و Command را توضیح دهید.
- ۲۲- روش‌های ارتباط انواع کنترل‌ها به داده‌ها را بیان کنید.

آزمون عملی

- ۱- پروژه‌ای طراحی کنید که با استفاده از یک پایگاه داده بتواند اطلاعات مربوط به دانش‌آموزان یک مدرسه را ذخیره، نگهداری و بازیابی کند.
- ۲- پروژه‌ای طراحی کنید که با استفاده از مفاهیم ساختار، داده شمارشی و مجموعه بتواند اطلاعات مربوط به یک مجموعه از کالاهای یک فروشگاه را دریافت، ذخیره و بازیابی کند.
- ۳- پروژه‌ای طراحی کنید که با استفاده از یک پایگاه داده بتواند داده و اطلاعات داروها در یک داروخانه را ذخیره، مشاهده و مدیریت کند.

هدف جزئی

توانایی برنامه‌نویسی با زبان برنامه‌نویسی دلفی

زمان (ساعت)	
عملی	نظری
۳۰	۱۵

هدف‌های رفتاری

- پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:
- ۱- بتواند برنامه زبان برنامه‌نویسی دلفی را اجرا کرده و نحوه استفاده از آن را بداند.
 - ۲- مفهوم (VCL (Visual Component Library را توضیح دهد.
 - ۳- ساختار واحد پروژه و فرم را توضیح داده و بخش‌های موجود در هر یک را بیان کند.
 - ۴- انواع داده‌های ساده و پیچیده را توضیح داده و کاربرد هر یک را بیان کند.
 - ۵- نحوه ایجاد رابطه‌های گرافیکی MDI و SDI را بداند.
 - ۶- بتواند از کنترل‌های کادر محاوره، منو، منوهای محتوا و نوار ابزار استفاده کند.
 - ۷- بتواند رویه‌ها و توابع را تعریف کرده و فراخوانی نماید.
 - ۸- نحوه ارسال پارامترها با روش مقدار و مرجع را بداند و تفاوت‌های آن‌ها را با یکدیگر بیان کند.
 - ۹- نحوه استفاده از ساختارهای شرطی و حلقه را توضیح دهد.
 - ۱۰- نحوه مدیریت خطاهای زمان اجرا را بداند.
 - ۱۱- بتواند به پایگاه داده متصل شده و داده‌ها و اطلاعات را بازبازی کند.

کلیات

پس از عرضه سیستم‌های عامل ویندوز، نگارش ۹۵ و نگارش‌های بعدی شرکت بولند نیز زبان برنامه‌نویسی پاسکال را که برای سیستم‌های عامل DOS ارائه کرده بود ارتقا داده و با نام جدید Delphi به بازار عرضه نمود. اکنون بعد از سال‌ها و ارائه نگارش‌های مختلف از این زبان برنامه‌نویسی، نگارش‌های ۲۰۰۵ و ۲۰۱۰ این زبان برنامه‌نویسی تولید و عرضه شده‌اند که در این واحد کار نگارش ۲۰۱۰ بررسی می‌شود.

۹-۱ نحوه اجرای زبان برنامه‌نویسی دلفی و ایجاد یک پروژه

پس از نصب زبان برنامه‌نویسی دلفی نگارش ۲۰۱۰ برای اجرای آن، روی منوی شروع (Start) کلیک کنید. سپس ابتدا روی گزینه Embarcadero RAD Studio 2010 و بعد روی گزینه Delphi 2010 کلیک کنید. در این صورت برنامه دلفی ۲۰۱۰ اجرا شده و رابط گرافیکی آن مانند شکل ۹-۱ نمایش داده می‌شود.



شکل ۹-۱

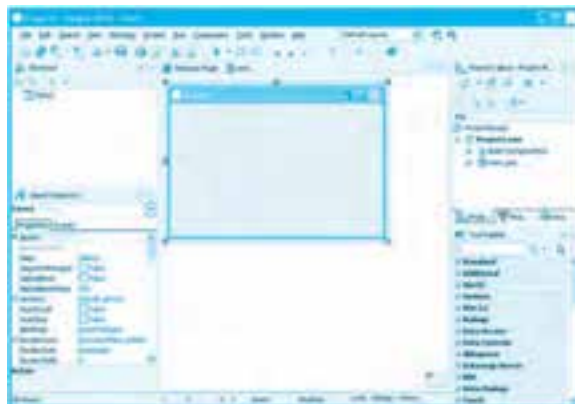
در این مرحله در زبانه Welcome Page و در بخش Recent Projects روی گزینه New Project ... کلیک کنید. در این صورت کادر محاوره New Items مانند شکل ۹-۲ نمایش داده می‌شود.

در این کادر محاوره می‌توانید انواع پروژه‌ها را از مدل‌های موجود انتخاب کرده و ایجاد نمایید. اکنون روی شاخه Delphi Projects و بعد روی آیکن VCL Forms Application روبه‌روی ساختار درختی کلیک کنید و سپس روی دکمه OK کلیک نمایید. در این صورت یک پروژه کاربردی برای استفاده در محیط

سیستم عامل ویندوز همراه با یک فرم ایجاد شده و پنجره برنامه دلفی مانند شکل ۹-۳ نمایش داده می‌شود.



شکل ۹-۲



شکل ۹-۳

نکته: برای ایجاد یک پروژه جدید می‌توانید روی منوی File کلیک کرده و بعد روی گزینه‌های New و VCL Form Application- Delphi کلیک کنید.

نکته: VCL مخفف کلمات **Visual Component Library** می‌باشد. در زبان برنامه‌نویسی دلفی با ابزار **VCL Designer** به راحتی می‌توان فرم‌ها و اجزای گرافیکی مانند کنترل‌ها را ایجاد، اضافه و تنظیم نمود.


در این پنجره بخش‌های مختلفی وجود دارد که در این جا بررسی می‌شوند.


در قسمت بالای پنجره نوار منو و نوارهای ابزار وجود دارند. در سمت چپ پنجره برنامه دلفی دو بخش وجود دارد بخش ساختار یا **Structure** و بخش بازرسی شیء یا **Object Inspector**.

بخش **Structure** ساختار هر یک از اجزای تشکیل دهنده پروژه را که در حال حاضر در پنجره طراحی فعال است نمایش می‌دهد و بازرسی شیء یا **Object Inspector** عملکردی مشابه پنجره خواص در **.Net** دارد و به وسیله آن می‌توان خاصیت‌ها و رویدادها را مورد دسترسی قرار داد. در سمت راست پنجره برنامه دلفی نیز دو بخش دیگر وجود دارد. بخش مدیر پروژه یا **Project Manager** و بخش پالت ابزار یا **Tool Palette**.

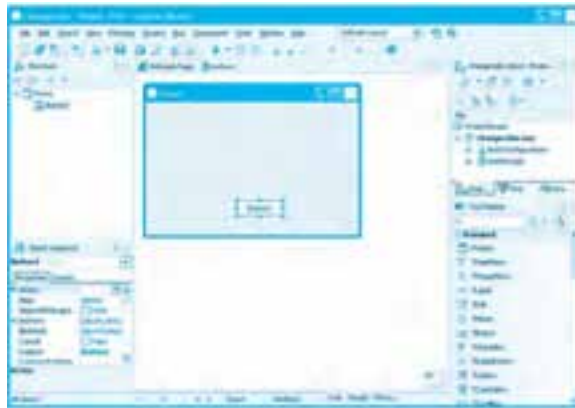
بخش مدیر پروژه عملکردی مانند **Solution Explorer** در **.Net** دارد و به وسیله آن ساختار پروژه و اجزای تشکیل دهنده آن نمایش داده می‌شود و بخش **Tool Palette** نیز عملکردی مشابه جعبه ابزار در ویژوال بیسیک ۶ و **.Net** به عهده دارد و انواع کنترل‌ها را برای طراحی برنامه در اختیار برنامه‌نویس قرار می‌دهد. بخش مدیر پروژه (**PM**) در بالاترین سطح گروه پروژه یا **ProjectGroup** قرار دارد. در واقع یک گروه پروژه (**PG**) شبیه به **Solution** در **.Net** است. یک گروه پروژه می‌تواند از بخش‌های کوچک‌تر مانند پروژه تشکیل شود که به طور پیش فرض با نام **Project1** و پسوند **exe** در مدیر پروژه نمایش داده می‌شود و روی دیسک فایل پروژه با نام **Project1** و پسوندهای **dproj** و **dpr** ذخیره می‌شوند.

هر پروژه از این نوع خود می‌تواند از اجزای دیگری مانند یک یا چند واحد (**unit**) تشکیل شود. در این نوع از پروژه‌ها به طور پیش فرض یک واحد با نام **Unit1.pas** وجود دارد. هر واحد (**unit**) می‌تواند یک یا مجموعه‌ای از دستورات عمل‌ها را شامل شود یا اجزای دیگری مانند یک فرم را برای برنامه معرفی و ایجاد کند که به آن واحد (**unit**) فرم می‌گویند. به طور پیش فرض در **unit1** پروژه، یک واحد فرم با نام **unit1** و پسوند **pas** معرفی می‌شود.

برای ذخیره کردن پروژه نیز می‌توانید از دکمه  در نوار ابزار پنجره برنامه دلفی استفاده کنید و پس از تعیین مسیر ذخیره‌سازی و نام فایل‌ها آن‌ها را روی دیسک ذخیره کنید.

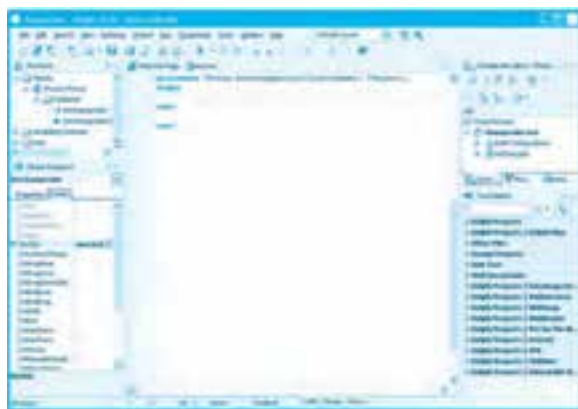
مثال ۱: می‌خواهیم پروژه‌ای طراحی کنیم که شامل یک فرم و یک دکمه باشد و با کلیک روی 

- دکمه، رنگ فرم تغییر کند. برای انجام این مثال این عملیات را به ترتیب انجام دهید:
- ۱- برنامه دلفی را اجرا کرده و یک پروژه از نوع VCL Forms Application همراه با یک فرم ایجاد کنید.
 - ۲- در پنجره مدیر پروژه روی آیکن پروژه کلیک کنید (Project1.exe). سپس به پنجره بازرس شیء (OI) بروید و در زبانه Properties روی خاصیت FileName کلیک کنید و نام پروژه را روی ChangeColor تنظیم کنید. البته در پنجره مدیر پروژه نیز می‌توانید روی آیکن پروژه کلیک راست کرده و در منویی که نمایش داده می‌شود روی گزینه Rename کلیک کرده و نام آن را تغییر دهید.
 - ۳- در این مرحله با همین روش نام واحد فرم یعنی unit1 را به myform تغییر دهید و بعد به پنجره طراحی فرم بروید و روی فرم کلیک کنید. سپس به پنجره بازرس شیء (OI) رفته و خاصیت‌های Name و Caption را به ترتیب روی مقادیر frmchangeColor و Change Color تنظیم کنید.
 - ۴- به پنجره طراحی فرم بروید و بعد در پنجره پالت ابزار روی دکمه Standard کلیک کنید. سپس کنترل TButton را پیدا کرده و روی آن دو بار کلیک کنید تا یک کنترل دکمه روی فرم قرار گیرد (شکل ۹-۴).



شکل ۹-۴

- ۵- در این مرحله به پنجره بازرس شیء بروید و خاصیت Name دکمه را روی btnchangeColor و خاصیت Caption آن را روی Change Color تنظیم کنید.
- ۶- اکنون در پنجره بازرس شیء روی زبانه Events کلیک کنید و رویداد OnClick دکمه btnchangeColor را پیدا کرده و روی آن کلیک کنید. سپس در کادر متن روبه‌روی این رویداد دو بار کلیک کنید تا مانند شکل ۹-۵ رویداد موردنظر در پنجره کدنویسی نمایش داده شود.




شکل ۵-۹

در این مرحله این دستور را در رویداد Click کنترل دکمه بنویسید:

```
frmchanecolor.Color := clgreen
```

این دستور با استفاده از خاصیت Color در فرم، رنگ زمینه فرم را به سبز تغییر می‌دهد. ثابت رشته‌ای ClGreen در زبان دلفی تعریف شده است. در ضمن برای انجام عمل انتساب در زبان دلفی به جای استفاده از عملگر = از := استفاده می‌شود.

۷- اکنون روی دکمه  در نوار ابزار پنجره برنامه دلفی کلیک کنید و در صورت لزوم مسیر ذخیره‌سازی پروژه و فرم را انتخاب نمایید.

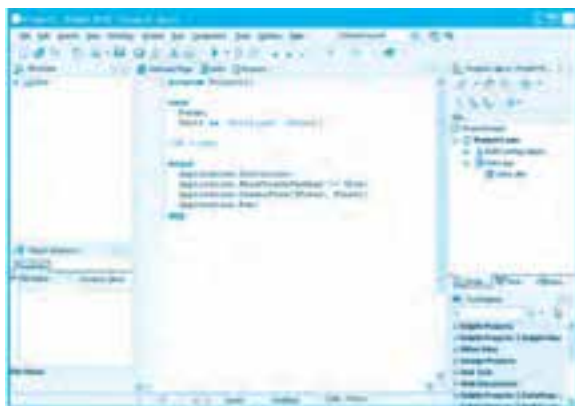
۸- در این مرحله می‌توانید با کلیک روی دکمه Run در نوار ابزار پنجره برنامه دلفی یا فشردن کلید F9 پروژه را اجرا کنید پس از نمایش فرم روی دکمه Change Color کلیک کنید. در این صورت رنگ زمینه فرم به رنگ سبز تغییر می‌کند.

۹- اجرای پروژه را خاتمه داده و به پنجره برنامه دلفی بازگردید.

۲-۹ ساختار واحد (unit) در زبان دلفی

به‌طور کلی یک پروژه در زبان دلفی به بخش‌های جداگانه به نام واحد یا unit تقسیم می‌شود. یک unit از قسمت‌های مختلفی تشکیل می‌شود. وقتی یک پروژه از نوع VCL Form Application ایجاد می‌کند به‌طور پیش‌فرض دو واحد ایجاد می‌شود. یک واحد که با نام پیش‌فرض Project1.exe در پنجره مدیر پروژه نشان داده می‌شود و یک واحد که مربوط به فرم پروژه می‌باشد و با نام unit1 نمایش داده می‌شود. برای مشاهده ساختار یک واحد از نوع پروژه می‌توانید روی آیکن پروژه Project1.exe در پنجره مدیر پروژه کلیک کنید.

سپس روی آیکن پروژه کلیک راست کرده و در منویی که نمایش داده می‌شود روی گزینه View Source کلیک کنید. در این صورت واحد پروژه مانند شکل ۹-۶ نمایش داده می‌شود.



شکل ۹-۶

واحد پروژه شباهت زیادی به برنامه‌هایی دارد که به زبان پاسکال نوشته می‌شود. در خط اول از این واحد، کلمه کلیدی Program قرار می‌گیرد که نام پروژه و همچنین واحد اصلی در پروژه را مشخص می‌کند. در بخش بعد کلمه کلیدی uses استفاده می‌شود. در این بخش نام واحدهایی که در این واحد مورد استفاده قرار می‌گیرند ذکر می‌شود. به عنوان مثال در شکل ۹-۶ واحد unit1 به عنوان واحدی که در پروژه از آن استفاده می‌شود قید شده است.

قسمت بعدی در این واحد با کلمه کلیدی begin شروع و با end خاتمه می‌یابد. در این بخش می‌توان دستورات موردنظر را برای اجرا قرار داد. به‌طور پیش‌فرض در این قسمت دستوراتی قرار می‌گیرند تا امکان نمایش فرم و اجرای پروژه امکان‌پذیر شود.

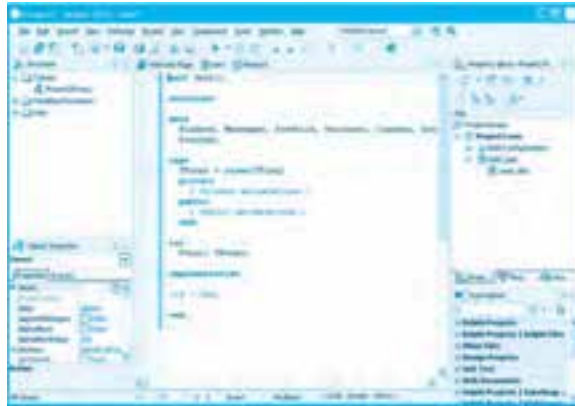
نکته: پسوند فایل‌هایی که واحد پروژه را نگهداری می‌کند dpr می‌باشد.



نکته: برای مشاهده واحد پروژه می‌توانید در نوار منوی پنجره برنامه دلفی ابتدا روی منوی Project و بعد گزینه View Source کلیک کنید یا از کلید ترکیبی Ctrl+V استفاده نمایید.



برای مشاهده ساختار واحد فرم در پنجره مدیر پروژه روی نام واحد unit1 دو بار کلیک کنید تا ساختار واحد در پنجره طراحی مانند شکل ۹-۷ نمایش داده شود.



شکل ۷-۹

واحد فرم با کلمه کلیدی unit شروع می‌شود و پس از آن نام واحد قرار می‌گیرد، سپس کلمه کلیدی interface قرار دارد. در این بخش عناصری مانند متغیرها، رویه‌ها، توابع قرار می‌گیرند که از آن‌ها می‌توان در سایر واحدهای پروژه استفاده کرد. این بخش با رسیدن به کلمه کلیدی بعدی خاتمه می‌یابد. می‌توان در بخش implementation بعضی از اعضا را که در بخش interface معرفی شده‌اند، پیاده‌سازی نمود. به عنوان مثال اگر یک رویه در بخش interface معرفی شود در بخش implementation تعریف و پیاده‌سازی می‌شود.

اگر در پروژه changecolor واحد فرم را باز کنید (شکل ۸-۹). بخش interface و قسمت type در این بخش قرار دارد. تعریف رویداد Click دکمه Change Color در قسمت Type وجود دارد و خود رویه رویداد در بخش implementation تعریف و پیاده‌سازی می‌شود.



شکل ۸-۹

نکته: برای آن که قسمت‌های مختلف در یک بخش، بهتر تشخیص داده شوند. در کنار کلمات کلیدی هر بخش یک علامت - وجود دارد که با کلیک روی آن، بخش مربوطه جمع شده و با کلیک روی علامت + جزئیات آن بخش قابل مشاهده خواهد بود.

در بخش Interface بخش دیگری به نام uses قرار دارد که عملکردی شبیه به همین بخش در واحد پروژه دارد و اسامی واحدهای دیگر که در واحد جاری مورد استفاده قرار می‌گیرند ذکر می‌شوند. پس از بخش uses، قسمت دیگری با کلمه کلیدی Type آغاز می‌شود. در این بخش می‌توان انواع جدیدی از داده‌ها را با توجه به انواع داده موجود، تعریف و در واحد جاری استفاده کرد. اما با توجه به این که واحد فرم در واقع یک کلاس است در این بخش یک کلاس فرم معرفی می‌شود (TForm1) و بخش Private و Public اعضای خصوصی و عمومی را در این کلاس معرفی می‌کنند. این بخش با کلمه کلیدی end; خاتمه می‌یابد. پس از بخش type، بخش var قرار دارد. در این بخش می‌توان متغیرهای موردنیاز را تعریف نمود. در خاتمه نیز واحد با کلمه کلیدی end. خاتمه می‌یابد.

نکته: برای نوشتن توضیحات در زبان دلفی می‌توان از دو کاراکتر // در ابتدای خط یا دو کاراکتر { } در دو طرف خط موردنظر استفاده نمود. مانند توضیحات موجود در بخش private یا public واحد فرم در شکل ۷-۹.

نکته: گاهی اوقات در واحدهای یک پروژه در زبان برنامه‌نویسی دلفی بعضی از خطوط داخل کاراکترهای { } قرار دارند و جزو توضیحات نیز محسوب نمی‌شوند. مثل خطی که بعد از کلمه کلیدی implementation در واحد فرم وجود دارد { \$R * .dfm }.

این خطوط دستوراتی هستند که در زمان ترجمه برنامه منظور خاصی را به مترجم زبان اعلام می‌کنند به این دستورات Compiler directive می‌گویند.

۳-۹ انواع داده‌ها در زبان برنامه‌نویسی دلفی

زبان دلفی انواع گوناگونی از داده‌ها را ارائه می‌کند که به وسیله آن‌ها می‌توان هرگونه اطلاعاتی را ذخیره، نگهداری و بازیابی نمود. در این جا انواع داده در زبان برنامه‌نویسی دلفی بررسی می‌شوند.

۱-۳-۹ داده‌های ساده (Simple Data)

این گروه از داده‌ها شامل داده‌های عددی صحیح، کاراکتری منطقی، نوع شمارشی، زیرمحدوده و اعشاری با ممیز شناور می‌باشند. جدول ۹-۱ انواع داده‌های عدد صحیح را در زبان دلفی معرفی می‌کند.

جدول ۹-۱ داده‌های عددی صحیح

نوع	دامنه	قالب
Integer	-2147483648..2147483647	۸ بیتی علامت دار
Cardinal	0..4294967295	۸ بیتی علامت دار
Shortint	-128..127	۸ بیتی علامت دار
Smallint	-32768..32767	۱۶ بیتی علامت دار
Longint	-2147483648..2147483647	۳۲ بیتی علامت دار
Int64	$-2^{63}..2^{63}-1$	۶۴ بیتی علامت دار
Byte	0..255	۸ بیتی بدون علامت
Word	0..65535	۱۶ بیتی بدون علامت
Longword	0..4294967295	۳۲ بیتی بدون علامت
UInt64	$0..2^{64}-1$	۶۴ بیتی بدون علامت

برای تعریف داده‌های از نوع کاراکتر، زبان برنامه‌نویسی دلفی دو نوع AnsiChar و WideChar را ارائه می‌کند. نوع داده AnsiChar برای ذخیره کاراکترهای اسکی به کار می‌رود و یک بایت را در حافظه اشغال می‌کند. نوع داده WideChar برای ذخیره کاراکترهای Unicode با ظرفیت دو بایت در حافظه به کار می‌رود. برای تعریف داده‌های از نوع منطقی، زبان برنامه‌نویسی دلفی چهار نوع داده منطقی را ارائه می‌کند که عبارتند از: Boolean، ByteBool، WordBool، LongBool و Boolean.

نوع داده Boolean و ByteBool یک بایت و WordBool دو بایت و LongBool چهار بایت را در حافظه اشغال می‌کنند. به علاوه برای ذخیره و نگهداری داده‌های اعشاری می‌توانید از انواع داده‌های ارائه شده در جدول ۹-۲ استفاده کنید.

جدول ۲-۹ داده‌های عددی اعشاری

نوع	دامنه	قالب
Real	$-5.0 \times 10^{-324} \dots 1.7 \times 10^{308}$	8
Real48	$2.9 \times 10^{-39} \dots 1.7 \times 10^{38}$	6
Currency	-922337203685477.5808.. 922337203685477.5807	8
Single	$1.5 \times 10^{-45} \dots 3.4 \times 10^{38}$	4
Double	$5.0 \times 10^{-324} \dots 1.7 \times 10^{308}$	8
Extended	$3.6 \times 10^{-4951} \dots 1.1 \times 10^{4932}$	10
Comp	$-2^{63+1} \dots 2^{63-1}$	8

برای ذخیره‌سازی رشته در زبان برنامه‌نویسی دلفی می‌توانید از انواع داده‌های رشته‌ای که در جدول ۳-۹ ارائه شده‌اند استفاده کنید.

جدول ۳-۹ انواع داده‌های رشته‌ای

نوع	حداکثر اندازه	فضای اشغالی	توضیح
ShortString	۲۵۵ کاراکتر	۲-۲۵۶ بایت	برای سازگاری با نسخه‌های قبلی
AnsiString	تقریباً ۳۱۸۲ کاراکتر	۴ بایت تا ۲ گیگا بایت	کاراکترهای اسکی و unicode
UnicodeString	تقریباً ۳۰۸۲ کاراکتر	۴ بایت تا ۲ گیگا بایت	کاراکترهای اسکی و unicode
WideString	تقریباً ۳۰۸۲ کاراکتر	۴ بایت تا ۲ گیگا بایت	کاراکترهای Unicode

به جای استفاده از نوع داده رشته‌ای ShortString می‌توانید از نوع داده String نیز استفاده کنید.

برای تعریف متغیرها از انواع داده‌های ساده می‌توان در یک واحد در بخش var به این صورت عمل نمود:

نوع داده : نام متغیر

به عنوان مثال این دستورات چند متغیر از انواع مختلف را تعریف می‌کنند:

var I: Integer;

J: Int64;

Letter: AnsiChar ;

OK: Boolean;

S: string;

MyString: string[100];

در این دستورات آخرین دستور یک رشته با طول ۱۰۰ کاراکتر را تعریف می‌کند.

برای مقداردهی به متغیرهایی که از نوع داده‌های ساده ایجاد می‌شوند می‌توان به این صورت عمل کرد:

مقدار = : نام متغیر

به عنوان مثال این دستورات مقادیر موردنظر را در متغیرهایی که تعریف شده‌اند ذخیره می‌کنند.

OK := True;

I:=10;

s :='tehran';

Letter :='a';

نوع داده شمارشی (Enumeration)

برای تعریف داده‌های شمارشی در زبان برنامه‌نویسی دلفی می‌توان در بخش var به این صورت عمل نمود:

(... , عبارت دوم , عبارت اول): نام نوع داده شمارشی

به عنوان مثال این دستورات دو نوع داده شمارشی با نام MyColor و Answer را تعریف می‌کنند:

var

MyColor : (mcRed, mcBlue, mcGreen, mcYellow, mcOrange);

Answer : (ansYes, ansNo, ansMaybe);

در این صورت دلفی به اولین عبارت مقدار صفر، به دومین عبارت مقدار یک و به همین شکل الی آخر اعداد صحیح را نسبت می‌دهد. در نوع داده شمارشی فقط می‌توان مقادیر تعیین شده را مورد استفاده قرار داد.

روش دیگر در تعریف انواع داده شمارشی، تعریف آن در بخش type می‌باشد. برای این کار می‌توان به این صورت عمل کرد:

(... , عبارت دوم , عبارت اول) = نام نوع داده شمارشی

به عنوان مثال:

type


```
TMyColor = (mcRed, mcBlue, mcGreen, mcYellow, mcOrange);
```

```
Answer = (ansYes, ansNo, ansMaybe)
```

پس از تعریف داده نوع شمارشی در بخش type می‌توان در بخش var از آن یک متغیر تعریف کرد. برای این کار به این صورت عمل کنید:

نوع داده شمارشی : نام متغیر

به عنوان مثال:

```
type
```

```
MyColor = (mcRed, mcBlue, mcGreen, mcYellow, mcOrange);
```

```
Answer = (ansYes, ansNo, ansMaybe)
```

```
var
```

```
testcolor: MyColor ;
```

```
testanswer: Answer ;
```

البته در صورت تمایل می‌توان در یک نوع داده شمارشی مقادیر عددی دلخواهی را به عبارت‌ها نیز نسبت داد. به عنوان مثال:

```
type
```

```
Size = (Small = 5, Medium = 10);
```

پس از تعریف متغیر از نوع داده شمارشی می‌توان آن را به این صورت مقداردهی نمود:


```
testcolor := MyColor.mcRed ;
```

```
testanswer := Answer.ansYes;
```

البته می‌توان متغیرهایی را که از نوع داده شمارشی تعریف می‌شوند به این صورت نیز مقداردهی کرد:

```
testcolor := mcRed ;
```

```
testanswer := ansYes;
```

 **تمرین:** یک نوع داده شمارشی تعریف کنید که به وسیله آن بتوان رشته‌های تحصیلی در یک دانشگاه را تعیین و مقداردهی نمود.

نوع داده زیرمحدوده (SubRange)

با استفاده از این نوع می‌توان محدوده‌ای از مقادیر را برای داده‌هایی که از آن تعریف می‌شوند مشخص نمود.

در این صورت داده‌های تعریف شده از نوع زیرمحدوده فقط می‌توانند مقادیر تعیین شده را دریافت کنند. برای تعریف یک داده از نوع زیرمحدوده می‌توان هم از بخش type یا از بخش var استفاده کرد. به عنوان مثال این دستورات چند نوع داده از نوع زیرمحدوده را تعریف می‌کنند:

type

```
Numbers = -128..127;
```


```
Caps = 'A'..'Z';
```

```
Percentile = 0..99 ;
```

var

```
SomeNum: 1..500;
```

برای مقداردهی به متغیرهایی که از نوع زیرمحدوده تعریف می‌شوند فقط می‌توان از مقادیری که در محدوده تعیین شده قرار دارند استفاده کرد. در غیر این صورت هنگام اجرای برنامه خطا رخ می‌دهد.

تمرین: یک داده از نوع زیرمحدوده تعریف کنید که به وسیله آن بتوان سن یک شخص را در محدوده ۱ تا ۱۵۰ سال تنظیم نمود. 

۲-۳-۹ انواع ثابت‌ها

به طور کلی دو نوع ثابت در زبان برنامه‌نویسی دلفی قابل استفاده است: True Constant و Type Constant. ثابت‌های واقعی (True Constant) داده‌هایی هستند که امکان تغییر مقدار آن‌ها در زمان اجرای برنامه وجود ندارد. برای تعریف این نوع ثابت‌ها به این صورت عمل کنید:

Const مقدار = نام ثابت;

به عنوان مثال این دستورات چند نوع ثابت واقعی را تعریف می‌کنند:

```
const
```

```
Min = 0;
```

```
Max = 100;
```

```
Message = 'Out of memory';
```

```
Ln10 = 2.302585092994045684;
```

نکته: برای تعریف ثابت‌ها در یک واحد (unit) از بخشی که با کلمه کلیدی Const شروع



می‌شود استفاده می‌کنند.

نکته: به ثابت‌های واقعی، ثابت‌های بدون نوع (Untype) هم می‌گویند.



گروه دوم ثابت‌ها، ثابت‌های نوع‌دار یا Type Constant می‌باشند. مقدار این نوع ثابت‌ها را می‌توان در زمان اجرا تغییر داد به شرط آن‌که راهنمای کامپایلر (Compiler directive) `{ $jt }` را قبل از کلمه کلیدی `Const` قرار داد. برای تعریف ثابت‌های نوع‌دار می‌توان به این صورت عمل کرد:

مقدار = نوع داده: نام ثابت Const

به عنوان مثال این دستورات ثابت‌های نوع‌دار را در یک واحد تعریف می‌کنند.

`const`

`Max: Integer = 100;`

مثال ۲: می‌خواهیم پروژه‌ای طراحی کنیم که میزان حافظه کامپیوتر را بر اساس گیگابایت دریافت کرده و مقدار آن‌را بر اساس واحدهای دیگر نمایش دهد. برای انجام این مثال به این ترتیب عمل کنید:

۱- برنامه دلفی را اجرا کرده و یک پروژه از نوع `VCL Forms Application` با نام `memory` ایجاد کنید و نام فرم و عنوان آن‌را روی عبارت `firmmemory` و `Memory` تنظیم کنید.

۲- در این مرحله دو کنترل `TButton` روی فرم قرار داده و خاصیت نام آن‌ها را به ترتیب روی `btnchange` و `btnexit` و عنوان آن‌ها را به شکل مناسب تنظیم کنید (شکل ۹-۹).



شکل ۹-۹

۳- در پنجره برنامه دلفی و نوار ابزار آن روی دکمه  کلیک کرده و پروژه و فرم را با نام changememory و unitmemory در مسیر دلخواه ذخیره کنید.

۴- اکنون به پنجره طراحی فرم بازگردید و یک کنترل TEdit و پنج کنترل TLabel مانند شکل ۹-۹ روی فرم قرار دهید. کنترل TEdit مانند کنترل TextBox و کنترل TLabel مانند کنترل Label در ویژوال بیسیک می‌باشد.

۵- به واحد فرم (Form unit) بروید و در بخش Var این دستورات را تایپ کنید:

```
frmmemory: Tfrmmemory;
```

```
lngmemory: Longint;
```

```
lngmb: Longint;
```

```
lngkb: Longint;
```

```
s : string;
```

```
test : integer ;
```

این دستورات سه متغیر را از نوع Longint برای ذخیره‌سازی مقدار حافظه بر اساس سه واحد اندازه‌گیری MB، GB و KB تعریف می‌نمایند. متغیرهای s و test برای انجام عملیات تبدیل داده‌ها به یکدیگر استفاده می‌شوند.

۶- در این مرحله رویداد Click دکمه btnchange را به این صورت تنظیم کنید:

```
procedure Tfrmmemory.btnchangeClick(Sender: TObject);
```

```
begin
```

```
val( teditmemory.Text ,lngmemory,test) ;
```

```
lngmb := lngmemory * 1024;
```

```
lngkb := lngmb * 1024;
```

```
str(lngmb,s);
```

```
lblmb.Caption := s;
```

```
str(lngkb,s) ;
```

```
lblkb.Caption := s;
```

```
end;
```

در این رویداد ابتدا با استفاده از رویه val عددی که در کادر متن teditmemory تایپ شده است به عدد تبدیل می‌شود. رویه val سه پارامتر دریافت می‌کند. پارامتر اول رشته‌ای است که به عدد تبدیل می‌شود. پارامتر دوم نام

یک متغیر عددی است که عدد به دست آمده از رشته پارامتر اول در آن ذخیره می‌شود و پارامتر سوم که یک متغیر از نوع Integer است که مشخص می‌کند عملیات تبدیل رشته به عدد موفقیت‌آمیز باشد مقدار صفر و در غیر این صورت مقدار ۱ در آن ذخیره می‌شود. سپس میزان حافظه به مقدار مگابایت تبدیل شده و در متغیر lngmb ذخیره می‌شود و بعد میزان حافظه به مقدار کیلوبایت تبدیل می‌شود و در متغیر lngkb ذخیره می‌شود. در مرحله بعد برای آن که بتوان مقادیر محاسبه شده را روی فرم و در کنترل‌های برجسب نمایش داد باید آن‌ها را به نوع رشته‌ای تبدیل کرد. برای این کار از رویه str استفاده می‌شود. این رویه دو پارامتر دریافت می‌کند پارامتر اول یک داده عددی است که به رشته تبدیل می‌شود و پارامتر دوم نام یک متغیر رشته‌ای است که رشته حاصل از تبدیل عدد در آن ذخیره می‌شود. پس از تبدیل هر مقدار محاسبه شده، به عدد، رشته به دست آمده در خاصیت Caption کنترل‌های برجسب نمایش داده می‌شوند.

۷- اکنون رویداد Click دکمه Exit را به این صورت تنظیم کنید:

```
procedure Tfrmmemory.btnexitClick(Sender: TObject);
```

```
begin
```

```
    frmmemory.Close;
```

```
end;
```

در این رویداد با استفاده از متد Close فرم frmmemory بسته می‌شود.

۸- تغییرات را ذخیره کرده و پروژه را اجرا کنید و عدد ۲ را در کادر متن تایپ کنید و بعد روی دکمه Change کلیک کنید. در این صورت فرم برنامه مانند شکل ۹-۱۰ نتایج محاسبات را نمایش می‌دهد.



شکل ۹-۱۰

۹- روی دکمه Exit کلیک کرده و به پنجره برنامه دلفی بازگردید.
تمرین: پروژه changmemory را به شکلی تنظیم کنید که به وسیله آن بتوان میزان حافظه را بر اساس واحدهای بایت و بیت نیز مشاهده کرد.

نکته: عملگرهای ریاضی، مقایسه‌ای و منطقی در زبان برنامه‌نویسی دلفی کاملاً مشابه همین عملگرها در زبان برنامه‌نویسی ویژوال بیسیک می‌باشند.

۳-۳-۹ داده‌های ساختار یافته (Structured Types)

این گروه از داده‌ها در زبان برنامه‌نویسی دلفی شامل انواع آرایه‌ها، رکوردها، اشاره‌گرها می‌باشند که در این جا بررسی می‌شوند.

آرایه

برای تعریف آرایه یک بعدی در زبان برنامه‌نویسی دلفی به این صورت عمل کنید:

نوع داده of [اندیس خاتمه .. اندیس شروع] Array: نام آرایه

به عنوان مثال این دستور یک آرایه یک بعدی را با پنج عضو از نوع عددی صحیح تعریف می‌کند:

```
no: Array[1..5] of Integer;
```

پس از تعریف آرایه در بخش var یک واحد (Unit) می‌توان اعضای آن را به این صورت مورد دسترسی قرار داد:

[اندیس عضو] نام آرایه

به عنوان مثال این دستور عدد ۱۲ را در دومین عضو از آرایه no ذخیره می‌کند:

```
no[2]:=12;
```

برای تعریف آرایه‌های دو بعدی نیز به این دو شکل عمل می‌شود:

نوع داده of [اندیس پایانی در ستون .. اندیس شروع در ستون] of Array [اندیس پایانی در سطر .. اندیس شروع در سطر] Array: نام آرایه

یا

نوع داده of [اندیس پایانی در ستون ... اندیس شروع در ستون، اندیس پایانی در سطر ... اندیس شروع در سطر] Array: نام آرایه

به عنوان مثال این دستور یک آرایه را با سه سطر و چهار ستون از نوع Byte تعریف می‌کند:

```
no: Array[1..3,1..4] of Byte
```

پس از تعریف آرایه در بخش var یک واحد (Unit) می‌توان اعضای آن را به این صورت مورد دسترسی قرار داد.

[اندیس ستون، اندیس سطر] نام آرایه

به عنوان مثال این دستور عدد ۱۴ را در عضو با اندیس سطر ۲ و اندیس ستون ۳ ذخیره می‌کند.

```
no[2,3]=14;
```

تمرین: پروژه‌ای طراحی کنید که ۲۰ عدد دلخواه را دریافت و در یک آرایه یک بعدی ذخیره نماید. سپس اعداد زوجی را که در آرایه وجود دارند نمایش دهد.

رکورد Record

یک رکورد مجموعه‌ای از دو یا چند داده از نوع مشابه یا انواع متفاوت را شامل می‌شود. در این حالت به هر یک از اجزای تشکیل دهنده رکورد فیلد می‌گویند. برای تعریف یک رکورد می‌توان در بخش type به این صورت عمل کرد:

```
record = نام رکورد
```

```
    نوع داده: نام فیلد اول
```

```
    نوع داده: نام فیلد دوم
```

```
    ⋮
```

```
end;
```

به عنوان مثال این دستورات یک رکورد با نام BirthDate را تعریف می‌کنند:

```
type
```

```
    BirthDate = record
```

```
        Year: Integer;
```

```
        Month: (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec);
```

```
        Day: 1..31;
```

```
end;
```

در این دستورات رکورد BirthDate با سه فیلد تعریف شده است. فیلد اول با نام Year و نوع داده Integer، فیلد Month به صورت نوع داده شماری و نوع Day از نوع زیرمحدوده تعریف شده‌اند.

پس از تعریف یک رکورد می‌توان از آن در بخش var به این صورت استفاده نمود:

```
type
```

```
    BirthDate = record
```

```
        Year: Integer;
```

```
        Month: (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec);
```

Day: 1..31;

end;

var

myrecord: BirthDate;

پس از تعریف متغیر از نوع داده رکورد می‌توان به این صورت به فیلدها دسترسی پیدا کرد:

نام فیلد . نام متغیر از نوع رکورد

به عنوان مثال:

myrecord.Year := 1904;

myrecord.Month := Jun;

myrecord.Day := 16;

برای خلاصه‌سازی دستورات می‌توان مانند زبان برنامه‌نویسی ویژوال بیسیک از دستور with در زبان دلفی استفاده نمود. به عنوان مثال:

with myrecord do

begin

Year := 1904;

Month := Jun;

Day := 16;

end;

اشاره‌گر Pointer

یک اشاره‌گر این امکان را فراهم می‌کند تا بتوان به مکان خاصی از حافظه با استفاده از آدرس آن دسترسی پیدا کرد. به عبارت دیگر یک اشاره‌گر، آدرس یک مکان در حافظه را در خود نگهداری می‌کند. نوع اشاره‌گرها با توجه به نوع داده‌ای که به آن اشاره می‌کنند مشخص می‌شود. برای تعریف اشاره‌گرها از کاراکتر \wedge استفاده می‌شود. به عنوان مثال این دستورات متغیر X را از نوع Integer و یک اشاره‌گر P که می‌خواهیم به آدرس متغیر X در حافظه اشاره کند تعریف شده‌اند.

var

X: Integer;

P: ^Integer

پس از تعریف اشاره‌گر می‌توان با استفاده از کاراکتر @ آدرس متغیر X را در اختیار اشاره‌گر قرار داد.

X := 17;

P := @X;

مثال ۳: می‌خواهیم پروژه‌ای طراحی کنیم که به وسیله آن اطلاعات کارمندان یک سازمان یا شرکت را ذخیره و بازیابی کنیم. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه دلفی را اجرا کرده و یک پروژه از نوع VCL Forms Application همراه با یک فرم و کنترل‌های آن مانند شکل ۹-۱۱ ایجاد کنید.



شکل ۹-۱۱

۲- اکنون خاصیت نام فرم را روی frmorg و عنوان آن را روی Save Information تنظیم کنید.

۳- در این مرحله به واحد فرم بروید و یک داده از نوع رکورد در زیر کلمه کلیدی Type و قبل از تعریف کلاس فرم `Tfrmorg=class(TForm)` به این صورت تعریف کنید:

```
personal = record
    fname: string[20];
    lname: string[30];
```

```
salary: real ;
```

```
end;
```

این رکورد از سه فیلد نام، نام خانوادگی و حقوق تشکیل شده است.

۴- در این مرحله به بخش var در واحد فرم بروید و این دستور را در زیر دستور موجود در این بخش تایپ کنید تا یک متغیر از نوع رکورد Personal با نام employee را برای ذخیره‌سازی اطلاعات یک کارمند ایجاد کند.

```
employee:personal
```

۵- به پنجره طراحی فرم بازگردید و از آن جا به رویداد Click دکمه Save بروید و آن را به این صورت تنظیم کنید:

```
procedure Tformorg.btnsaveClick(Sender: TObject);
```

```
var varsalary real ;
```

```
  i : integer ;
```

```
begin
```

```
  employee.fname := teditfname.Text ;
```

```
  employee.lname := teditlname.Text ;
```

```
  val (teditsalary.Text,varsalary,i) ;
```

```
  employee.salary := varsalary ;
```

```
end;
```

در این رویداد دو متغیر محلی برای تبدیل مقدار خاصیت Text کادر متن teditSalary به داده عددی از نوع Real تعریف شده‌اند. سپس اطلاعات کارمند موردنظر در فیلدهای متغیر employee که از نوع داده رکورد personal تعریف شده است ذخیره می‌شود.

۶- اکنون یک فرم جدید با نام frmdisplay و عنوان Display Information مانند شکل ۱۲-۹ همراه با کنترل‌های آن ایجاد کرده و خاصیت Enabled کادرهای متن را روی مقدار False تنظیم کنید.



شکل ۹-۱۲

سپس به رویداد Click دکمه Back بروید و در رویداد Click آن، این دستور را تایپ کنید:

```
frmdisplay.close;
```

۷- به پنجره طراحی فرم frmorg بازگردید و رویداد Click دکمه Display را به این صورت تنظیم کنید تا اطلاعات ذخیره شده در کنترل‌های کادر متن نمایش قرار گیرند. در پایان فرم frmdisplay با متد show نمایش داده می‌شود:

```
procedure Tfrmorg.btndisplayClick(Sender: TObject);
```

```
var vars : string ;
```

```
begin
```

```
    frmdisplay.teditfname.Text := employee fname ;
```

```
    frmdisplay.teditlname.Text := employee.lname ;
```

```
    str(employee.salary :12:3,vars);
```

```
    frmdisplay.teditsalary.Text := vars ;
```

```
    frmdisplay.Show ;
```

```
end;
```

۸- پروژه را با نام organization و فرم اول را با نام register و فرم دوم را با نام Display ذخیره کنید.

۹- اکنون در واحد فرم به بخش uses بروید و در ادامه لیست واحدها، نام واحد display را اضافه کنید و تغییرات را ذخیره نمایید.

۱۰- پروژه را اجرا کرده و اطلاعات یک شخص را وارد و روی دکمه Save کلیک کنید. سپس روی دکمه Display کلیک کنید تا اطلاعات ذخیره شده، نمایش داده شوند.

۱۱- اجرای پروژه را خاتمه داده و به پنجره برنامه دلفی بازگردید.

۴-۹ دستورات شرطی در زبان برنامه‌نویسی دلفی

زبان دلفی برای کنترل روند اجرای برنامه در شرایط مختلف دو دستور if و case را ارائه می‌کند. دستور if را می‌توان به این صورت مورد استفاده قرار داد:

then عبارت(های) مقایسه‌ای **if**

begin

⋮

دستورات

end

else

begin

⋮

دستورات

end;

استفاده از بخش else اختیاری است در ضمن اگر در بخش if یا else از یک دستور استفاده شود می‌توان کلمات کلیدی begin و end را در آن بخش حذف نمود. به عنوان مثال:


```
if grade >= 10 then
```

```
    lblresult.Caption:='pass';
```

```
else
```

```
    lblresult.Caption:='fail';
```

در این دستورات اگر مقدار متغیر grade بزرگ‌تر یا مساوی ده باشد عبارت pass و در غیر این صورت عبارت fail در کنترل برچسب نمایش داده می‌شود.

تمرین: پروژه Organization را به شکلی تنظیم کنید که هنگام اجرای برنامه از کاربر کلمه رمزی 

پرسیده شود. اگر کلمه رمز نامعتبر باشد اجرای برنامه خاتمه یابد. در غیر این صورت کاربر وارد برنامه شود.

دستور Case نیز شباهت زیادی به دستور Select Case در زبان برنامه‌نویسی ویژوال بیسیک دارد و شکل کلی آن به این صورت است:

Case مقایسه of عبارت مورد

مقدار اول:

```
begin
  ⋮
end;    دستورات
```

مقدار دوم:

```
begin
  ⋮
end;    دستورات
```

else

```
begin
  ⋮
end;    دستورات
```

end;

در این دستور اگر مقدار عبارت مورد مقایسه با اولین مقدار برابر باشد دستورات بخش اول اجرا می‌شود. در غیر این صورت مقدار دوم بررسی می‌شود و به همین شکل الی آخر. استفاده از بخش else اختیاری است و اگر نتیجه بررسی تمام مقادیر نادرست باشد دستورات بخش else اجرا می‌شوند. به عنوان مثال این دستورات رتبه یک دانش‌آموز را با توجه به مقدار معدل وی تعیین می‌کند.

case average of

```
18..20: lblresult.Caption := 'A';
```

```
16,17: lblresult.Caption := 'B';
```

```
14,15: lblresult.Caption := 'C';
```

else

lblresult.Caption := 'D' ;

end;

در این دستورات اگر معدل دانشجوی بین مقدار ۱۸ تا ۲۰ باشد رتبه وی A، اگر مقدار ۱۶ یا ۱۷ باشد رتبه وی B، اگر مقدار ۱۴ و ۱۵ باشد رتبه وی C و B اگر کمتر از ۱۴ باشد بخش else اجرا شده و رتبه وی D اعلام می‌شود.

نکته: در دستور Case می‌توانید مقدار انواع داده‌هایی را که قالب آن‌ها کوچک‌تر از ۳۲ بیتی باشد بررسی کنید. مانند داده‌های عددی از نوع Integer، زیرمحدوده، نوع داده شمارشی و مانند آن‌ها.

تمرین: پروژه‌ای طراحی کنید که میزان مصرف برق یک مشترک را به صورت ماهانه دریافت و در یک آرایه ذخیره نماید. سپس با توجه به متوسط مصرف برق وی در یک سال رتبه مصرف کننده را اگر از ۳۰۰ کیلووات بیشتر باشد پرمصرف، بین ۲۰۰ تا ۳۰۰ کیلووات عادی و از ۲۰۰ کیلووات کمتر را کم‌مصرف اعلام نماید.

۵-۹ دستورات حلقه در زبان برنامه‌نویسی دلفی

زبان برنامه‌نویسی دلفی برای تکرار اجرای دستورالعمل‌ها، انواع حلقه‌های for و repeat و While را ارائه می‌کند. عملکرد دستور for مانند همین دستور در زبان برنامه‌نویسی ویژوال بیسیک می‌باشد. نحوه استفاده از این دستور به این صورت است:

do مقدار خاتمه **to** مقدار شروع **:=** شمارنده حلقه **for**

begin

⋮
⋮
⋮

دستورات

end;

به عنوان مثال این دستورات مجموع اعداد ۱ تا ۱۰۰ را محاسبه می‌کنند.

sum := 0;

for k := 1 to 100 do

begin

sum := sum + k

end;

حلقه for را می‌توان به صورت کاهشی نیز مورد استفاده قرار داد. در این صورت مقدار شمارنده در هر بار

اجرای حلقه یک واحد کم می‌شود.

do مقدار خاتمه **downto** مقدار شروع =: شمارنده حلقه **for**

```
begin
  .
  .
  .
  .
  .
  .
end;
```

دستورات

به عنوان مثال حلقه قبل را می‌توان به این صورت تنظیم کرد:

```
sum := 0;
for k := 100 downto 1 do
begin
  sum := sum + k
end;
```

نکته: در صورتی که از یک دستور در حلقه **for** استفاده شود می‌توان کلمات کلیدی **begin** و **end** را حذف نمود.

تمرین: پروژه organization را به شکلی تنظیم کنید که بتوان اطلاعات ۱۰ کارمند را ذخیره و بازیابی نمود.

حلقه دیگری که می‌توان از آن در زبان دلفی برای تکرار دستورات استفاده نمود حلقه **While** می‌باشد. این حلقه عملکردی مشابه حلقه‌های **While ... Wend** و **Do While ... Loop** در زبان برنامه‌نویسی ویژوال بیسیک می‌باشند و تا زمانی که نتیجه بررسی شرط یا شرط‌های ذکر شده در آن‌ها درست (**True**) باشد دستورات حلقه اجرا می‌شوند. نحوه استفاده از این دستور به این صورت است:

do (شرط‌ها) **While**


```
begin
  .
  .
  .
  .
  .
  .
end;
```

دستورات

به عنوان مثال این دستورات مجموع اعداد فرد کوچک‌تر از ۱۰ را محاسبه می‌کنند.

```
k :=0;  
sum := 0;  
while K <=10 do  
begin  
  if Odd(K) then sum := sum + K;  
  K := K +1;  
end;
```

در این دستورات از تابع Odd برای شناسایی اعداد فرد استفاده شده است. اگر عددی که به این تابع ارسال می‌شود فرد باشد مقدار True را بازگشت می‌دهد.

نکته: اگر از یک دستور در حلقه While استفاده شود می‌توان کلمات کلیدی **end** و **begin** را استفاده نمود. 

نوع سوم حلقه‌ها در زبان دلفی حلقه repeat می‌باشد. عملکرد این حلقه مانند حلقه Do Until در زبان برنامه‌نویسی ویژوال بیسیک می‌باشد. یعنی دستورات حلقه تا زمانی اجرا می‌شود که نتیجه بررسی شرط حلقه نادرست است. نحوه استفاده از این حلقه به این صورت است:

```
repeat  
  ⋮  
  ⋮  
  ⋮  
until ((شرط(ها)));
```

دستورات

دستورات این حلقه حداقل یک بار اجرا می‌شوند چون شرط حلقه در انتهای حلقه بررسی می‌شود. به عنوان مثال این دستورات نیز مجموع اعداد فرد کوچک‌تر از ۱۰ را محاسبه می‌کنند:

```
k :=0;  
sum := 0;  
repeat  
  if Odd(K) then sum := sum + K;  
  K := K +1;  
until K >10 ;
```


۶-۹ نحوه تعریف توابع در زبان برنامه‌نویسی دلفی

در زبان برنامه‌نویسی دلفی می‌توانید توابع موردنظر خود را مانند سایر زبان‌های برنامه‌نویسی تعریف کنید تا در صورت نیاز مانند توابع تعریف شده در زبان برنامه‌نویسی دلفی از آن‌ها استفاده کنید. شکل کلی نحوه تعریف توابع به این صورت است:

نوع داده بازگشتی: (...; نوع داده : نام آرگومان) نام تابع **function**

```
var
:
:   تعریف متغیرهای محلی تابع
:
begin
:
:   دستورات
:
نام تابع
end;
```

به عنوان مثال این تابع طول و عرض یک مستطیل دلخواه را دریافت کرده و مساحت آن را محاسبه نموده و بازگشت می‌دهد.

```
function area (length : single ; width : single): single;
var
  s : single;
begin
  s := length * width ;
  area :=s;
end;
```

در این تابع دو آرگومان برای دریافت طول و عرض مستطیل با نام‌های Length و Width از نوع Single معرفی شده‌اند و چون حاصلضرب این دو یعنی مقدار مساحت نیز از نوع Single می‌باشد، بنابراین نوع داده بازگشتی تابع نیز به همین صورت Single اعلام می‌شود و برای محاسبه مساحت در بخش var تابع یک متغیر محلی با نام s معرفی می‌شود. سپس در بلوک begin...end تابع، دستورات محاسبه مساحت مستطیل اجرا می‌شوند و در پایان مقدار مساحت (s) به وسیله نام تابع به محل فراخوانی بازگشت می‌دهد.

پس از تعریف تابع لازم است قبل از فراخوانی آن را به واحد معرفی نمود. برای این کار لازم است فقط خط اول تعریف تابع را به همان شکل در بخش uses واحد مربوطه و قبل از کلمه کلیدی type قرار دهید.

اکنون برای فراخوانی این تابع نیز می‌توان به این صورت عمل نمود:

```
result :=area(4,3);
```

مثال ۴: پروژه‌ای طراحی کنید که به وسیله یک تابع بتوان تعداد ساعات کارکرد یک کارگر و میزان دستمزد هر ساعت کار وی را محاسبه نموده و نمایش داد. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه دلفی را اجرا کرده و یک پروژه از نوع VCL Forms Application همراه با یک فرم مانند شکل ۹-۱۳ با نام frmpayment و عنوان payment همراه با کنترل‌های آن ایجاد کنید.



شکل ۹-۱۳

۲- به واحد فرم بروید و قبل از کلمه کلیدی end. تابع را به این صورت تنظیم کنید:

```
function funcpayment (varhour : single ; varpay : single): single;
```

```
begin
```

```
    funcpayment := varhour * varpay ;
```

```
end;
```

در این دستورات تابع funcpayment با دو آرگومان varhour و varpay تعریف شده است تا تعداد ساعات کارکرد و دستمزد یک ساعت کارکرد را دریافت می‌کند. سپس این دو متغیر را در هم ضرب کرده و حاصل ضرب را که حقوق پرداختی کارگر است به محل فراخوانی بازمی‌گرداند.

۳- اکنون در واحد فرم به بخش uses بروید و در این بخش قبل از کلمه کلیدی type تابع را به این صورت

معرفی کنید:

function funcpayment (varhour : single ; varpay : single): single;

۴- در این مرحله رویداد Click دکمه Payment را به این صورت تنظیم کنید:

procedure Tfrmpayment.btnpaymentClick(Sender: TObject);

var varpayment ,varhour,vartotal : single;

 strpayment : string ;

 test : integer;

begin

 val(tedithour.Text , varhour ,test);

 val(teditpayment.Text , varpayment ,test);

 vartotal := funcpayment(varhour, varpayment);

 str(vartotal:10:3, strpayment);


 lblpayment.Caption := strpayment;

end;

در این رویداد پس از تبدیل داده‌های دریافتی به مقادیر عددی، تابع funcpayment فراخوانی می‌شود و نتیجه محاسبات پس از بازگشت در متغیر vartotal ذخیره می‌شود. سپس مقدار این متغیر در کنترل برجسب نمایش داده می‌شود.

۵- پروژه و فرم را با نام salary و payment ذخیره کنید. سپس پروژه را اجرا و برنامه را آزمایش و بررسی کنید.

۶- به اجرای پروژه خاتمه داده و به پنجره برنامه دلفی بازگردید.

 **تمرین:** پروژه‌های طراحی کنید که با استفاده از یک تابع، بخش‌پذیری دو عدد را بر یکدیگر بررسی نموده و با یک پیام مشخص کند دو عدد بر هم بخش‌پذیر هستند یا خیر.

۷-۹ نحوه تعریف رویه‌ها در زبان برنامه‌نویسی دلفی

در زبان برنامه‌نویسی دلفی می‌توان رویه‌های موردنظر خود را تعریف کرده و دستورات موردنظر را در آن‌ها قرار داد تا در صورت نیاز مانند رویه‌های تعریف شده زبان دلفی از آن‌ها استفاده کرد.

شکل کلی نحوه تعریف رویه‌ها به این صورت است:

Procedure (... , نوع داده : نام آرگومان) نام رویه

```
Var  
:  
:  
:  
تعریف متغیرهای محلی رویه  
begin  
:  
:  
:  
دستورات  
end;
```

به عنوان مثال این رویه یک پیام را به صورت رشته دریافت کرده و در یک کنترل برجسب روی یک فرم نمایش می‌دهد:

```
procedure displaymessage(vartext : string);  
  
begin  
    frmmessage.lblmessage.Caption := vartext;  
    frmmessage.ShowModal ;  
  
end;
```

در این رویه آرگومان vartext پیام موردنظر را به صورت یک رشته دریافت کرده و سپس آنرا در یک کنترل برجسب قرار داده و فرم frmmessage را با متد ShowModel نمایش می‌دهد. وقتی یک فرم با این متد نمایش داده می‌شود فرم به صورت کادرمحاوره نمایش داده شده و تا زمانی که فرم بسته نشود امکان ادامه برنامه وجود ندارد.

برای فراخوانی این رویه می‌توان به این صورت عمل کرد:

```
displaymessage('Delphi 2010');
```

نکته: رویه‌ها را نیز باید قبل از فراخوانی مانند توابع به واحد معرفی نمود.



مثال ۵: می‌خواهیم پروژه salary را به شکلی تنظیم کنیم تا در صورتی که مقادیر منفی در کادرهای متن وارد شود پیام مناسبی نمایش داده شود. برای انجام این مثال این عملیات را انجام دهید:

- ۱- برنامه دلفی را اجرا کرده و پروژه salary را باز کنید.
- ۲- به پنجره مدیر پروژه بروید و یک فرم جدید به پروژه اضافه کنید. سپس نام آنرا روی frmmessage و عنوان آنرا روی عبارت Error تنظیم کنید. به علاوه برای آن که بتوان این فرم را به صورت یک کادر پیغام

نمایش داد در پنجره بازرس شیء (OI) خاصیت `BorderStyle` را روی مقدار `bsDialog` تنظیم کنید سپس فرم جدید را با نام `message` ذخیره کنید.

۳- اکنون یک کنترل برچسب روی فرم جدید قرار داده و نام آن را به `lblmessage` تغییر دهید. سپس یک دکمه با نام `btnok` و عنوان `&OK` نیز روی آن قرار داده و دستور `frmmessage.close` را در رویداد `Click` آن تایپ کنید.

۴- به واحد فرم `frmpayment` بروید و این رویه را پس از تابع `funcpayment` تایپ کنید:

```
procedure displaymessage(vartext : string);
```

```
begin
```

```
    frmmessage.lblmessage.Caption := vartext;
```

```
    frmmessage.ShowModal ;
```

```
end;
```

۵- به بخش `uses` در واحد فرم `frmpayment` بروید و نام واحد `message` را به لیست واحدهای موجود اضافه کنید. سپس تعریف رویه را قبل از بخش `Type` و در زیر دستور معرفی تابع `funcpayment` به این صورت تایپ کنید:

```
procedure displaymessage(vartext : string);
```

۶- در این مرحله رویداد `Click` دکمه `Payment` را به این صورت تنظیم کنید:

```
procedure Tfrmpayment.btnpaymentClick(Sender: TObject);
```

```
var varpayment ,varhour,vartotal : single;
```

```
    strpayment : string ;
```

```
    test : integer;
```

```
begin
```

```
    val(tedithour.Text , varhour ,test);
```

```
    val(teditpayment.Text , varpayment ,test);
```

```
    if (varhour < 0) or (varpayment < 0 ) then
```

```
begin
```

```
    displaymessage('Data Entry Is Invalid!');
```

```
end
```

```
else  
begin  
    vartotal := funcpayment( varhour, varpayment);  
    str( vartotal:10:3, strpayment);  
    lblpayment.Caption := strpayment;  
end;  
end;
```

در این رویداد علاوه بر دستورات قبلی یک دستور شرطی if اضافه شده است تا مقدار متغیرهای varhour و varpayment را بررسی کند. اگر مقدار یکی از آنها یا هر دو کوچک‌تر از صفر باشد رویه displaymessage فراخوانی می‌شود و پیام 'Data Entry Is Invalid!' به آن ارسال می‌شود. با فراخوانی این رویه فرم frmmessage همراه با این پیام نمایش داده می‌شود.

۷- تغییرات را ذخیره کرده و پروژه را اجرا کنید و اعداد منفی در کادرهای متن مربوط به تعداد ساعات یا میزان دستمزد را تایپ کرده و روی دکمه Payment کلیک کنید. در این مرحله کادر پیغامی مانند شکل ۹-۱۴ نمایش داده می‌شود.



شکل ۹-۱۴

- ۸- به اجرای پروژه خاتمه داده و به پنجره برنامه دلفی بازگردید.
- تمرین: پروژه salary را به شکلی تنظیم کنید تا اگر میزان ساعات کارکرد از ۱۲ ساعت بیشتر شود پیام خطای مناسبی نمایش داده شود. ✓
- تمرین: رویه‌ای بنویسید که به وسیله آن بتوان میزان حقوق دریافتی را در پروژه salary روی یک فرم جداگانه نمایش داد. ✓

نکته: در توابع و رویه‌ها می‌توان برای آرگومان‌ها مقدار پیش‌فرض قرار داد. در این صورت در زمان فراخوانی، ارسال مقدار به آرگومان‌ها اختیاری خواهد بود. به عنوان مثال رویه `displaymessage` را می‌توان به این صورت تعریف کرد:

```
procedure displaymessage(vartext : string = 'ok');
begin
  frmmessage.lblmessage.Caption := vartext;
  frmmessage.ShowModal ;
end;
```

در این حالت این رویه را علاوه بر روش‌های قبلی می‌توان به این صورت نیز فراخوانی نمود:

```
displaymessage();
```

نکته: در زمان فراخوانی رویه‌ها و توابع می‌توان پارامترها را به دو صورت ارسال با مقدار یا ارسال با مرجع به آن‌ها انتقال داد. به طور پیش‌فرض پارامترهای ارسالی به صورت مقدار ارسال می‌شوند. برای ارسال پارامترها به صورت مرجع لازم است قبل از نام آرگومان در تعریف تابع یا رویه از کلمه کلیدی `var` استفاده شود.

۱-۷-۹ روش ارسال پارامترها در زمان فراخوانی رویه‌ها و توابع

به طور کلی در زمان فراخوانی رویه‌ها و توابع می‌توان پارامترهای ارسالی را به دو صورت ارسال با مقدار یا ارسال با مرجع مورد استفاده قرار داد. به طور پیش‌فرض پارامترهای ارسالی به صورت مقدار ارسال می‌شوند و تغییر در مقدار آرگومان‌ها در رویه یا تابع تأثیری روی پارامترهای ارسالی نمی‌گذارد. در روش ارسال با مرجع، هرگونه تغییر در مقدار آرگومان‌ها در رویه یا تابع سبب تغییر پارامترهای ارسالی می‌شود. برای ارسال پارامترها با روش ارسال با مرجع لازم است قبل از نام آرگومان در تعریف تابع یا رویه از کلمه کلیدی `var` استفاده شود. به عنوان مثال این رویه مساحت یک مرجع دلخواه را محاسبه می‌کند:

```
procedure proccarea (varlenght : single ;var vararea : single);
begin
  vararea := varlenght * varlenght ;
```

end;

در این رویه آرگومان varlength به صورت مقدار طول ضلع مربع را دریافت می‌کند و آرگومان vararea به صورت مرجع اعلام می‌شود تا به وسیله آن بتوان مقدار مساحت را به پارامتر متناظر آن در زمان فراخوانی انتقال داد. برای فراخوانی این رویه می‌توان به این صورت عمل کرد:

```
procedure TForm1.btnareaClick(Sender: TObject);
```

```
var result :single ;
```

```
strarea :string ;
```

```
begin
```

```
proccarea (2, result) ;
```

```
str( result:10:3, strarea);
```

```
lblarea.Caption := strarea;
```

```
end;
```

در این رویداد رویه proccarea با دو پارامتر فراخوانی می‌شود. پارامتر اول مقدار عدد ۲ است که طول ضلع مربع می‌باشد و متغیر result که به صورت مرجع ارسال می‌شود. بنابراین پس از خاتمه فراخوانی رویه، مقدار مساحت مربع در این متغیر به محل فراخوانی بازگشت داده می‌شود.

۸-۹ مدیریت خطا در زبان برنامه‌نویسی دلفی

زبان برنامه‌نویسی دلفی مانند زبان برنامه‌نویسی ویژوال بیسیک عملکرد مناسبی را در رابطه با انواع خطاها از خود نشان می‌دهد. در این زبان برنامه‌نویسی، خطاهای نوشتاری مانند سایر زبان‌های برنامه‌نویسی حال حاضر مانند VB6 و VB.Net بلافاصله در زمان طراحی برنامه گزارش داده می‌شوند. به علاوه برای شناسایی و برطرف کردن خطاهای منطقی مانند زبان‌های VB6، VB.Net امکانات لازم در منوی Run وجود دارد. گزینه‌های Step Over، Trace Into، Run to Cursor، Add Watch و غیره عملکردی مشابه همین گزینه‌ها در زبان‌های برنامه‌نویسی VB6 و VB.Net دارند. در ضمن با استفاده از کلید F5 یا کلیک ماوس روی دستور موردنظر می‌توان نقطه توقف ایجاد نمود.

برای مدیریت خطاهای زمان اجرا نیز مانند زبان برنامه‌نویسی ویژوال بیسیک Net می‌توانید از دستور Try استفاده کنید. شکل کلی این دستور به این صورت است:

```
try
```

```
⋮  
دستورات اصلی  
⋮
```



```
except
```

```

:
:   on   نوع   خطا   do
:
:
begin   begin
:
:       :
:       :   دستورات مدیریت خطا
:       :
end;    end;
```

در این دستور اگر دستورات موجود در بخش try سبب ایجاد خطا شوند، اجرای برنامه به طور خودکار به بخش except منتقل می‌شود و نوع خطا در اولین on بررسی می‌شود. اگر نوع خطای به وجود آمده با نوع خطا در اولین on یکسان باشد، دستورات این بخش اجرا می‌شود و اجرای برنامه به بعد از بلوک try یعنی end; منتقل می‌شود. اما اگر نوع خطای به وجود آمده با نوع خطا در اولین on یکسان نباشد در صورت وجود on دیگری نوع خطای به وجود آمده با نوع خطا در on دوم بررسی می‌شود و به همین شکل الی آخر.

به عنوان مثال این دستورات از ایجاد خطای تقسیم بر صفر جلوگیری به عمل آورده و به جای نمایش کادر پیغام دلفی مبنی بر ایجاد خطای تقسیم بر صفر، کادر پیغامی با استفاده از تابع showMessage نمایش داده می‌شود.

```
procedure TForm1.btndivisionClick(Sender: TObject);
```

```
var x,y,z :integer ;
```

```
s : string;
```

```
begin
```

```
try
```

```
  x := 12 ;
```

```
  y := 0;
```

```
  z := x div y ;
```

```
except
```

```
  on EDivByZero do   showMessage('DivByZero');
```

```
end;
```

```
str(z,s);
```

```
label1.Caption := s;
```

```
end;
```

۹- نحوه ایجاد رابط‌های گرافیکی SDI و MDI

در رابط‌های گرافیکی تک‌سندی یا SDI یک فرم به عنوان فرم اصلی و سایر فرم‌ها به صورت کادرمحاوره نمایش داده می‌شوند. نحوه ایجاد این نوع از فرم‌ها مورد بررسی قرار گرفته‌اند و برای ایجاد رابط‌های گرافیکی چند سندی یا MDI یک فرم به عنوان فرم والد یا اصلی و سایر فرم‌ها به صورت فرم‌های فرزند یا کادره‌های محاوره نمایش داده می‌شوند. برای ایجاد این نوع از رابط‌های گرافیکی می‌توانید به این ترتیب عمل کنید:

۱- برنامه دلفی را اجرا کرده و یک پروژه از نوع VCL Forms Application همراه با یک فرم ایجاد کنید.

۲- خاصیت FormStyle فرم را روی مقدار fsMDIForm و مقدار خاصیت Color آن را روی مقدار ClAppWorkSpace تنظیم کنید تا فرم MDI برنامه تنظیم شود.

۳- یک فرم جدید به پروژه اضافه کنید و خاصیت FormStyle آن را روی مقدار fsMDIChild تنظیم کنید تا فرم فرزند برای برنامه تنظیم شود.

۴- در فرم مادر با استفاده از متد Create فرم فرزند می‌توان فرم‌های فرزند را ایجاد نمود.

مثال ۶: می‌خواهیم پروژه‌ای از نوع MDI طراحی کنیم که به وسیله آن بتوان تصاویر موردنظر خود را روی فرم‌های جداگانه مشاهده نمود. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

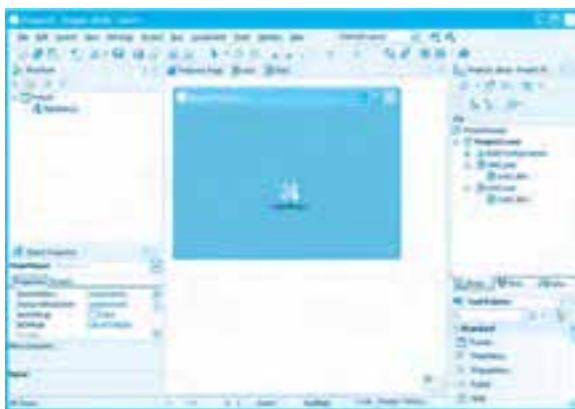
۱- برنامه دلفی را اجرا کرده و یک پروژه از نوع VCL Forms Application همراه با یک فرم ایجاد کنید. سپس خاصیت نام و عنوان آن را روی frmmdi و Showpictures تنظیم کنید و بعد خاصیت FormStyle آن را نیز روی مقدار fsMDIForm و خاصیت WindowState آن را روی مقدار wsMaximized تنظیم کنید، سپس خاصیت Color آن را روی مقدار clAppWorkSpace تنظیم نمایید.

۲- یک فرم جدید به پروژه اضافه کرده و نام آن را روی مقدار frmchild و عنوان آن را روی Child تنظیم کنید. سپس خاصیت FormStyle آن را روی مقدار fsMDIChild تنظیم کنید.

۳- در پالت Tool روی زبانه Additional کلیک کنید و یک کنترل تصویر TImage روی فرم frmchild قرار داده و خاصیت نام آن را روی impicture تنظیم کنید.

۴- در این مرحله می‌خواهیم یک نوار منو روی فرم اصلی برنامه ایجاد کنیم. برای این کار به پنجره طراحی فرم frmmdi بروید و در پالت Tool و در زبانه Standard روی کنترل TMainMenu دوبار کلیک کنید. در

این حالت یک کنترل منو روی فرم نمایش داده می‌شود (شکل ۹-۱۵).



شکل ۹-۱۵

۵- برای ایجاد گزینه‌های موردنظر در نوار منو در پنجره طراحی فرم، روی کنترل منو کلیک راست کنید و در منویی که نمایش داده می‌شود روی گزینه **Menu Designer ...** کلیک کنید. در این مرحله پنجره طراحی منو (Menu Designer) مانند شکل ۹-۱۶ نمایش داده می‌شود.



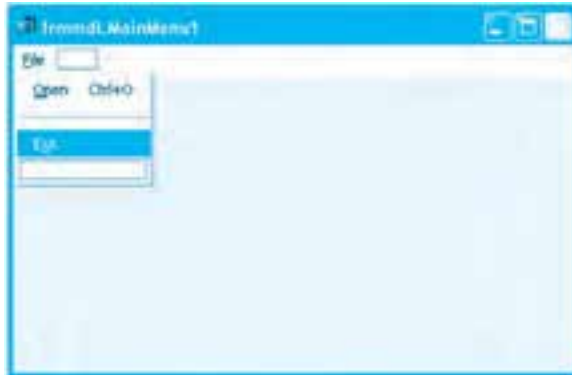
شکل ۹-۱۶

۶- اکنون به پنجره بازرس شیء (OI) بروید و خاصیت نام را روی مقدار `mnufile` و خاصیت `Caption` را روی مقدار `&File` تنظیم کنید تا گزینه اول در نوار منو ایجاد شود. در این صورت منوی `File` در پنجره طراحی منو نمایش داده می‌شود و امکان ایجاد یک گزینه در سمت راست و یک گزینه در زیر آن به وجود می‌آید (شکل ۹-۱۷).



شکل ۹-۱۷

۷- به همین صورت یک گزینه Open در منوی File ایجاد کرده و خاصیت Shortcut آن را روی مقدار Ctrl+O تنظیم کنید. سپس سایر گزینه‌ها را نیز مانند شکل ۹-۱۸ به منوی فایل اضافه کنید و پنجره طراحی منو را بسته و به پنجره طراحی فرم بازگردید.



شکل ۹-۱۸

۸- پروژه‌ها و فرم‌ها را به ترتیب با نام ShowPicture, mdi, و child ذخیره کنید.
۹- به واحد فرم frmmdl بروید و در بخش uses آن، نام واحد child را به انتهای اسامی واحدها اضافه کنید.
۱۰- برای جستجو و انتخاب تصویرها باید از یک کادرمحاوره استفاده کنیم. بنابراین به پنجره طراحی فرم frmmdl بروید و بعد در پالت Tool روی زبانه Dialogs کلیک کنید. در این زبانه انواع مختلفی از کادره‌های محاوره Open, Save Color, و غیره قابل استفاده می‌باشد. روی کنترل TOpenPictureDialog دو بار کلیک کنید تا کنترل روی فرم اضافه شود. سپس خاصیت نام آن را روی مقدار opdopen تنظیم کنید. به علاوه با

استفاده از خاصیت Filter در این کنترل می‌توان فایل‌های موردنظر را فیلتر نمود.
 ۱۱- اگر پروژه در این مرحله اجرا شود، یک فرم فرزند به طور خودکار ایجاد و نمایش داده می‌شود.



شکل ۹-۱۹

برای جلوگیری از این مسأله در منوی پنجره برنامه دلفی ابتدا روی منوی Project و بعد روی گزینه ... Options کلیک کنید تا کادرمحاوره‌ای مانند شکل ۹-۱۹ نمایش داده شود. سپس در این کادرمحاوره و ساختار درختی موجود در آن روی عبارت Forms کلیک کنید (شکل ۹-۲۰).



شکل ۹-۲۰

اکنون در کادر لیست Auto-createforms روی نام فرم frmchild و بعد روی دکمه > کلیک کنید تا فرم frmchild به کادر لیست Available forms منتقل شود. در این صورت هنگام اجرای پروژه فرم frmchild نمایش داده نمی‌شود (شکل ۹-۲۱).



شکل ۹-۲۱

در این رویداد ابتدا یک شیء از کلاس فرم فرزند Tfrmchild ایجاد می‌شود. سپس کادرمحاوره باز کردن تصویر با متد Execute اجرا می‌شود تا کادرمحاوره نمایش داده شود. در این صورت اگر روی دکمه Open در این کادرمحاوره کلیک شود، دستورات بخش then اجرا شده و با استفاده از متد Create کلاس فرم فرزند، فرم فرزند جدیدی ایجاد می‌شود.

بعد با استفاده از متد LoadFromFile خاصیت Picture کنترل تصویر، تصویری که نام و مسیر آن در خاصیت FileName کنترل کادرمحاوره باز کردن تصویر ذخیره شده است در کنترل تصویر نمایش داده می‌شود.

۱۲- به پنجره طراحی منو رفته و روی گزینه Open کلیک کنید، سپس در پنجره بازرس شیء (OI) روی زبانه Events کلیک کرده و رویداد OnClick آنرا به این صورت تنظیم کنید:

```
procedure Tfrmmdi mnuopenClick(Sender: TObject);
```

```
var mychild : Tfrmchild;
```

```
begin
```

```
if opdopen.Execute then
```

```
begin
```

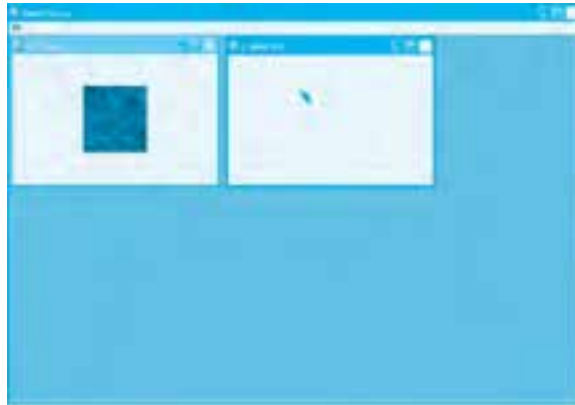
```
    mychild := Tfrmchild.Create(Self);
```

```
    mychild.imgpicture.Picture.LoadFromFile(opdopen.FileName );
```

```
    mychild.Caption := opdopen.FileName ;
```

```
end;
```

۱۳- در این مرحله تغییرات را ذخیره کرده و پروژه را اجرا کنید و با استفاده از گزینه Open در منوی File چند تصویر دلخواه را انتخاب کرده و مشاهده نمایید (شکل ۹-۲۲).

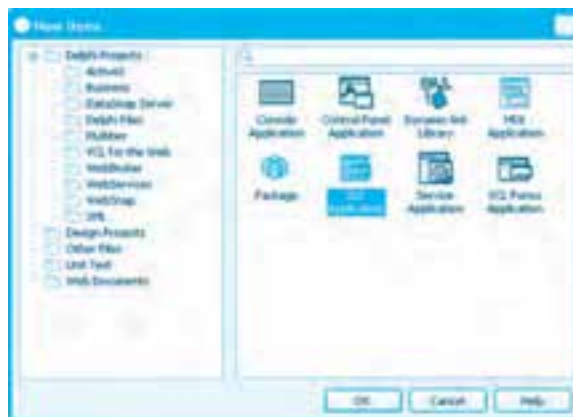


شکل ۹-۲۲

۱۴- به اجرای پروژه خاتمه داده و به پنجره برنامه دلفی بازگردید.

تمرین: پروژه showpicture را با استفاده از کادرمحاوره TopenDialog تنظیم و اجرا نمایید. ✓

نکته: ✓ برای ایجاد پروژه‌های SDI و MDI می‌توانید از نمونه‌های آماده‌ای که در کادرمحاوره New Item هنگام ایجاد پروژه جدید وجود دارد استفاده کنید. با استفاده از مدل آماده SDI Application و MDI Application در این کادرمحاوره، می‌توان برنامه‌های SDI و MDI را به سرعت و سهولت ایجاد نمود (شکل ۹-۲۳).



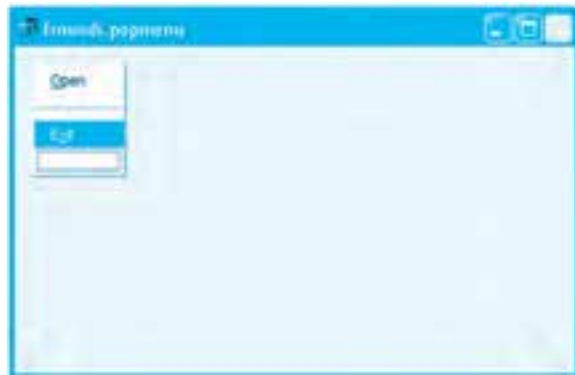
شکل ۹-۲۳

مثال ۷: می‌خواهیم پروژه showpicture را به شکلی تنظیم کنیم تا با کلیک راست روی فرم مادر نیز



بتوان تصاویر را انتخاب کرده و نمایش داد. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

- ۱- برنامه دلفی را اجرا کرده و پروژه showpicture را باز کنید.
- ۲- به پنجره طراحی فرم frmmdi بروید و سپس در پالت Tool روی زبانه Standard کلیک کنید. اکنون یک کنترل TPopupMenu روی فرم قرار دهید و نام آن را روی popmenu تنظیم کنید.
- ۳- در پنجره طراحی فرم frmmdi روی کنترل popmenu کلیک راست کرده و در منویی که نمایش داده می‌شود، روی گزینه ... Menu Designer کلیک کنید. سپس سه گزینه با نام‌های popopen و popexit و popsep مانند شکل ۹-۲۴ در آن ایجاد کنید.




شکل ۹-۲۴

- ۴- در پنجره طراحی منو Popmenu روی گزینه Open کلیک کنید، سپس رویداد آن را انتخاب کرده و به این صورت تنظیم کنید:

```
procedure Tfrmmdi.popopenClick(Sender: TObject);  
var mychild : Tfrmchild;  
begin  
  if opdopen.Execute then  
  begin  
    mychild := Tfrmchild.Create(Self);  
    mychild.imgpicture.Picture.LoadFromFile(opdopen.FileName );  
    mychild.Caption := opdopen.FileName ;  
  end;  
end;
```


end;

۵- اکنون می‌خواهیم منوی Popmenu را با فرم frmmdi مرتبط کنیم. بنابراین به پنجره طراحی فرم frmmdi بروید و در پنجره بازرس شی روی خاصیت Popmenu کلیک کنید. بعد روی دکمه  رویه‌روی این خاصیت کلیک کنید، سپس گزینه Popmenu را انتخاب کنید.

۶- تغییرات را ذخیره کرده و پروژه را اجرا کنید. سپس روی پنجره اصلی برنامه کلیک راست کنید. یک منوی مانند شکل ۹-۲۵ نمایش داده می‌شود. با استفاده از این گزینه، یک تصویر را انتخاب کرده و نمایش دهید.



شکل ۹-۲۵

۷- به اجرای پروژه خاتمه داده و به پنجره ویژوال بیسیک بازگردید.

مثال ۸: می‌خواهیم پروژه showpicture را به شکلی تنظیم کنیم تا با استفاده از یک نوار ابزار نیز بتوان تصاویر را انتخاب کرده و نمایش داد. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه دلفی را اجرا کرده و پروژه showpicture را باز کنید.

۲- در پنجره پالت Tool روی زبانه Win32 کلیک کرده و یک کنترل TImageList روی فرم اصلی برنامه (frmmdi) قرار دهید. این کنترل عملکردی مشابه کنترل ImageList در زبان برنامه‌نویسی ویژوال بیسیک دارد و به وسیله آن می‌توان یک یا مجموعه‌ای از تصاویر را جمع‌آوری کرده و در سایر کنترل‌ها مانند کنترل‌های منو و نوار ابزار استفاده نمود. برای قرار دادن تصاویر در این کنترل روی آن کلیک راست کرده و در منوی که نمایش داده می‌شود روی گزینه ImageList Editor... کلیک کنید. کادرمحاوره‌ای مانند شکل

۹-۲۶ نمایش داده می‌شود. در این کادرمحاوره می‌توان با کلیک روی دکمه Add تصاویر را انتخاب و به کنترل اضافه نمود. چند تصویر را انتخاب کرده و به کنترل اضافه کنید سپس روی دکمه OK کلیک کنید تا به پنجره طراحی فرم بازگردید. اکنون نام این کنترل را روی imglst1 تنظیم کنید.



شکل ۹-۲۶

۳- در این مرحله به پنجره پالت Tool بروید و در زبانه Win32 روی کنترل TToolBar دوبار کلیک کنید تا کنترل روی فرم frmmdi قرار داده شود، سپس نام آن را روی عبارت tbmdi تنظیم کنید.

۴- برای آن که بتوان از تصاویر موجود در کنترل imglst1 در کنترل نوار ابزار استفاده کرد، باید خاصیت Images را در پنجره بازرس شیء انتخاب کرده و نام کنترل imglst1 را از آن انتخاب نمود.

۵- در پنجره طراحی فرم frmmdi روی کنترل نوار ابزار کلیک راست کرده و در منویی که نمایش داده می‌شود، روی گزینه New Button کلیک کنید تا یک دکمه به نوار ابزار اضافه شود.

نام این دکمه را روی tbbtnopen تنظیم کنید. اکنون می‌توانید با استفاده از خاصیت ImageIndex تصویر موردنظر خود را از کنترل ImageList موردنظر انتخاب کنید. سپس رویداد Click این دکمه را به این صورت تنظیم کنید:

```
procedure Tfrmmdi.tbbtnopenClick(Sender: TObject);  
var mychild : Tfrmchild;  
begin  
if opdopen.Execute then  
begin  
mychild := Tfrmchild.Create(Self);  
mychild.imgpicture.Picture.LoadFromFile(opdopen.FileName );
```


در این صورت کادرمحاوره‌ای مانند شکل ۹-۲۸ نمایش داده می‌شود.



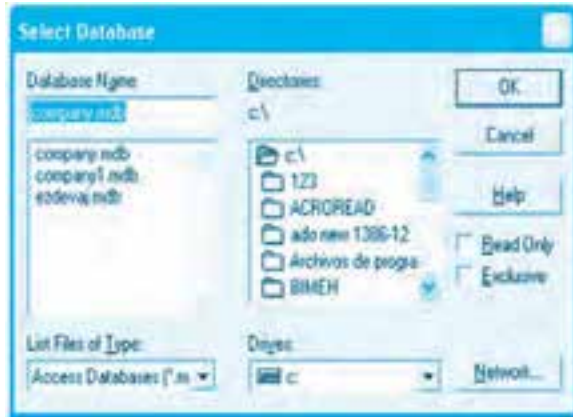
شکل ۹-۲۸

در این کادرمحاوره و در کادر لیست موجود در آن روی گزینه MS Access Database کلیک کنید و بعد روی دکمه ... Configure کلیک کنید تا کادرمحاوره دیگری مانند شکل ۹-۲۹ نمایش داده شود.

در این کادرمحاوره روی دکمه ... Select کلیک کنید. اکنون کادرمحاوره دیگری مانند شکل ۹-۳۰ نمایش داده می‌شود.

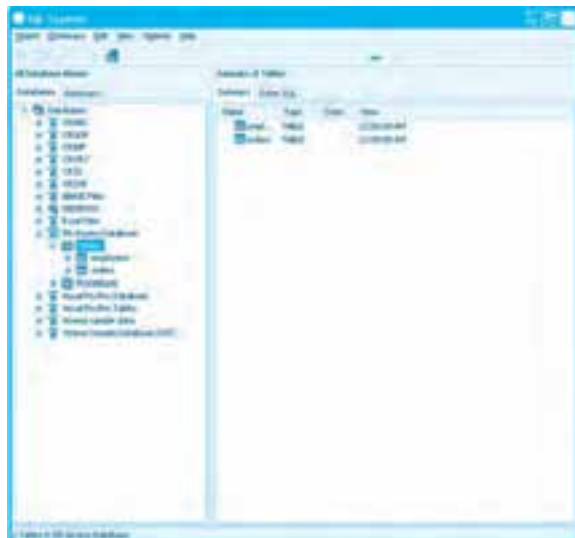


شکل ۹-۲۹



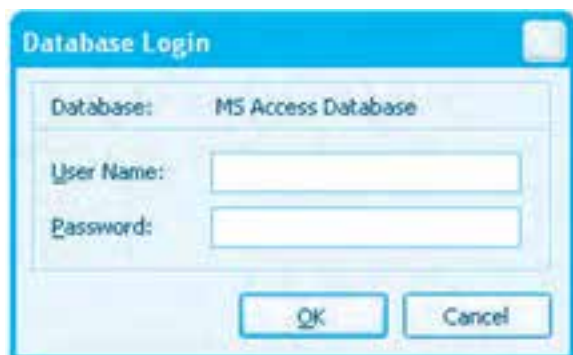
شکل ۹-۳۰

در این کادرمحاوره می‌توانید پایگاه داده خود را جستجو و انتخاب کنید. سپس روی دکمه OK کلیک کنید و به کادره‌ای محاوره قبلی بازگشته و به همین شکل با کلیک روی دکمه OK به پنجره SQL Explorer بازگردید. در این مرحله اگر روی شاخه ms Access Database کلیک کنید می‌توانید جدول‌های پایگاه داده‌ای را که به آن متصل شده‌اید مانند شکل ۹-۳۱ مشاهده کنید. اکنون می‌توانید پنجره SQL Explorer را بسته و از اتصال ایجاد شده در برنامه خود استفاده کنید.



شکل ۹-۳۱

نکته: اگر در پنجره SQL Explorer در زمان کلیک روی شاخه MS Access Database کادر
مجاوره‌ای مانند شکل ۹-۳۲ نمایش داده شود روی دکمه OK کلیک کنید تا بسته شود.



شکل ۹-۳۲

مثال ۹: می‌خواهیم پروژه‌ای طراحی کنیم که به وسیله آن بتوان اطلاعات جدول employees را در پایگاه داده Company بازبینی و مشاهده نمود. برای انجام این مثال این عملیات را به ترتیب انجام دهید:

۱- برنامه دلفی را اجرا کرده و یک پروژه از نوع VCL Forms Application همراه با یک فرم با نام frmemployees و عنوان Employees ایجاد کنید.

۲- به پالت Tool رفته و روی زبانه BDE کلیک کنید و یک کنترل از نوع TTable روی فرم قرار داده و آن را به پایگاه داده Company متصل کنید و نام آن را به tblemployees تغییر دهید.

۳- پس از اتصال کنترل tblemployees روی این کنترل کلیک کرده و خاصیت DatabaseName آن را روی مقدار MS Access Database تنظیم کنید. سپس خاصیت TableName آن را روی جدول employees تنظیم نمایید.

در این مرحله روی کنترل tblemployee کلیک راست کرده و در منویی که نمایش داده می‌شود، روی گزینه ... Fields Editor کلیک کنید تا کادرمجاوره‌ای مانند شکل ۹-۳۳ نمایش داده شود. در این کادرمجاوره کلیک راست کرده و در منویی که نمایش داده می‌شود روی گزینه Add all Fields کلیک کنید تا کلیه فیلدهای جدول employeeid انتخاب شوند (شکل ۹-۳۴). سپس این کادرمجاوره را ببندید

و به مرحله بعد بروید.



شکل ۹-۳۳



شکل ۹-۳۴

۴- اکنون به پالت Tool بروید و روی زبانه Data Access کلیک کنید. بعد روی کنترل TDataSource در این زبانه دو بار کلیک کنید و نام این کنترل را روی dsemployees و خاصیت DataSet آنرا روی نام کنترل tbemployees تنظیم کنید.

۵- در این مرحله دوباره به پالت Tool بروید و روی زبانه Data Controls کلیک کنید. سپس روی کنترل TDBGrid دوبار کلیک کنید تا روی فرم اضافه شود. این کنترل عملکردی مشابه کنترل DataGridView در ویژوال بیسیک Net دارد. اکنون خاصیت نام این کنترل را روی dbemployees تنظیم کنید و بعد خاصیت DataSource این کنترل را نیز روی dsemployees تنظیم کنید.

۶- اکنون یک کنترل TButton روی فرم قرار دهید و نام آن را روی benfill و عنوان آن را روی Fill تنظیم کنید، سپس رویداد Click آن را به این صورت تنظیم کنید:

```
procedure TForm1.btnfillClick(Sender: TObject);
```

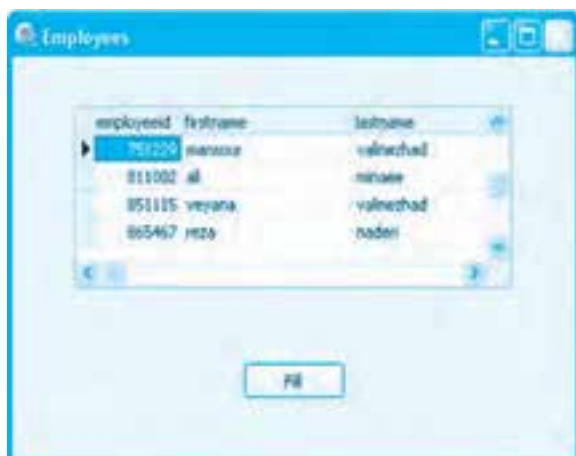
```
begin
```

```
tblemployees.Active := true;
```

```
end;
```

در این رویداد با تنظیم خاصیت Active کنترل tblemployees روی مقدار True داده‌ها و اطلاعات بازایی شده نمایش داده می‌شوند.

۷- پروژه و فرم را با نام Showinformation و employees ذخیره کنید. سپس پروژه را اجرا نموده و روی دکمه Fill کلیک کنید. در این صورت اطلاعات کارمندان روی فرم مانند شکل ۳۵-۹ نمایش داده می‌شوند.



شکل ۳۵-۹

۸- به اجرای پروژه خاتمه داده و به پنجره برنامه دلفی بازگردید.

تمرین: پروژه‌ای طراحی کنید که به وسیله آن بتوان اطلاعات دانش‌آموزان یک کلاس را از پایگاه



داده مربوطه بازیابی نموده و نمایش داد.



Learn in English

The RAD Studio

RAD Studio is an Integrated Development Environment (IDE) for building Win32 applications. The RAD Studio IDE provides a comprehensive set of tools that streamline and simplify the development life cycle. The tools available in the IDE depend on the version of RAD Studio you are using. The following sections briefly describe these tools.

واژه‌نامه

Component	مؤلفه، جزء
Directive	دستوردهنده
Implementation	اجرا، انجام
Inspector	بازرس
Interface	رابط، واسط
Organization	سازمان
Palette	پالت، تخته رنگ در نقاشی
Pointer	اشاره‌گر
Repeat	تکرار کردن
Subrange	زیرمحدوده
Unit	واحد

خلاصه مطالب

- زبان برنامه‌نویسی دلفی، نگارش ارتقا یافته زبان برنامه‌نویسی پاسکال است که برای سیستم‌های عامل ۳۲ بیتی تهیه شده است.
- ابزار امکان طراحی و ساخت فرم‌ها و اجزای گرافیکی آن‌ها را سریع و آسان می‌کند.
- برای دسترسی به کنترل‌ها می‌توان از پالت Tool استفاده کرد.
- از پنجره مدیر پروژه برای مدیریت پروژه و اجزای تشکیل دهنده آن استفاده می‌شود.
- برای تنظیم ویژگی‌های فرم‌ها و کنترل‌ها در زمان طراحی از پنجره بازرس شیء استفاده می‌شود.
- یک واحد پروژه به طور پیش‌فرض از بخش‌های Uses, Program, Begin ... end تشکیل می‌شود.
- یک واحد فرم به‌طور پیش‌فرض از بخش‌های Unit, Interface, Uses, Type و Implementation تشکیل می‌شود.

- داده‌ها به طور کلی به سه گروه داده‌های ساده، پیچیده و ثابت تقسیم می‌شوند.
- داده‌های ساده عبارتند از: داده‌های عدد، انواع شمارشی، رشته‌ای و زیرمحدوده.
- ثابت‌ها به دو گروه نوع‌دار و بدون نوع تقسیم می‌شوند.
- داده‌های ساخت یافته یا پیچیده عبارتند از: آرایه‌ها، رکوردها و اشاره‌گرها.
- دستورات شرطی در زبان دلفی عبارتند از: if و case.
- دستورات حلقه در زبان دلفی عبارتند از: for، repeat و while.
- برای تعریف توابع در زبان دلفی از دستور Function استفاده می‌شود.
- برای تعریف رویه‌ها در زبان دلفی از دستور Procedure استفاده می‌شود.
- داده‌ها را به دو روش می‌توان به توابع و رویه‌ها ارسال کرد: روش ارسال با مقدار، روش ارسال با مرجع.
- برای مدیریت خطاهای زمان اجرا در دلفی از دستور try ... except استفاده می‌شود.
- برای نمایش منوها در رابطه‌های گرافیکی می‌توان از کنترل‌های TMainMeny و TPopupMenu استفاده کرد.
- برای نمایش نوارهای ابزار می‌توان از کنترل TToolBar استفاده کرد.
- از کنترل TOpenPictureDialog می‌توان برای جستجو و انتخاب فایل‌های گرافیکی استفاده کرد.
- برای اتصال به پایگاه‌های داده و بازیابی داده و اطلاعات از کنترل‌های TTable، TDataSource و TDBGrid استفاده می‌شود.

آزمون نظری

۱- کدام خاصیت باید برای تعیین یک فرم به عنوان فرزند تنظیم شود؟

الف - BorderStyle ب - Style

ج - FormStyle د - FormsBorder

۲- کدام کنترل برای اتصال و بازیابی داده استفاده می‌شود؟

الف - TTable ب - TDataSource

ج - TDBGrid د - Table

۳- کدام ویژگی در زبان برنامه‌نویسی دلفی طراحی برنامه را سریع و آسان می‌کند؟

الف - Tool Palette ب - بازرس شی (OI)

ج - VCL د - پنجره Structure

۴- پسوند فایل‌هایی که واحد فرم را نگهداری می‌کنند می‌باشد.

الف - dfm ب - pas

ج - dpr د - dproj

۵- کدامیک از انواع داده‌های زیر برای نگهداری داده‌هایی که در یک محدوده مشخصی

قرار دارند مناسب است؟

الف - SubRange ب - Record

ج - String د - Array

۶- در زمان فراخوانی رویه‌ها و توابع، برای ارسال پارامترها به صورت مرجع از کلمه

کلیدی قبل از نام آرگومان استفاده می‌شود.

الف - Var ب - Ref

ج - By Var د - By Val

۷- کدامیک از حلقه‌های زیر تا زمانی که نتیجه بررسی شرط نادرست باشد اجرا

می‌شود؟

الف - For ب - While

ج - Repeat د - do while

۸- برای نمایش کادرهای محاوره از متد استفاده می‌شود.

الف - Show ب - ShowDialog

ج - Execute د - Exec

۹- کدام نوع از انواع داده‌های زیر برای نگهداری داده‌های منطقی مناسب

نیست؟

ب- ByteBool

الف- Boolean

د- Bool

ج- WordBool

۱۰- کدام یک از انواع داده‌ها می‌تواند به آدرس یک متغیر دیگر اشاره کند؟

ب- Word

الف- Record

د- Array

ج- Pointer

11- On the RAD Studio visual designer surface, you can create graphical user interfaces by dragging and dropping components from the to a form.

a- Tool Palette

b- Object Inspector

c- Project Manager

d- Structure Palette

۱۲- نحوه اجرا و اجزای تشکیل دهنده زبان برنامه‌نویسی دلفی را توضیح دهید.

۱۳- انواع داده‌ها در زبان برنامه‌نویسی دلفی را توضیح دهید.

۱۴- نحوه ایجاد درایوهای گرافیکی SDI و MDI را توضیح دهید.

۱۵- ساختار واحد فرم و واحد پروژه را توضیح داده و تفاوت‌های آن‌ها را بیان کنید.

۱۶- انواع دستورات شرطی را در زبان دلفی با ذکر مثال توضیح دهید.

۱۷- انواع دستورات حلقه را در زبان دلفی با ذکر مثال توضیح دهید.

۱۸- نحوه مدیریت خطاهای زمان اجرا را توضیح دهید.

۱۹- نحوه اتصال و بازیابی داده‌ها و اطلاعات را از پایگاه‌های داده توضیح دهید.

۲۰- نحوه ایجاد کنترل‌های منو، منوهای محتوا و نوار ابزار را توضیح دهید.

۲۱- نحوه تعریف رویه‌ها و توابع را در زبان دلفی توضیح دهید.

۲۲- انواع روش‌های ارسال پارامترها به توابع و رویه‌ها را با ذکر مثال توضیح دهید.

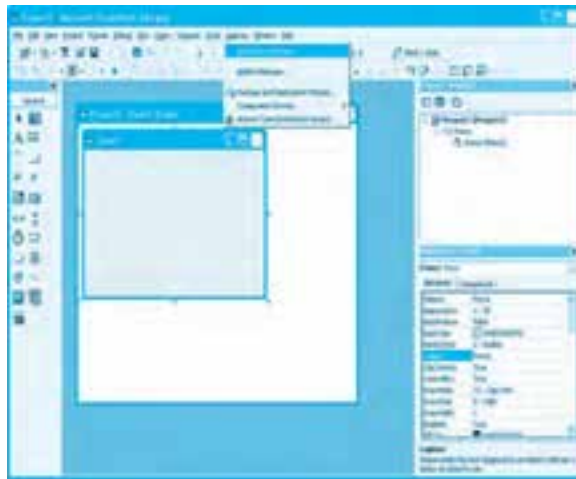
آزمون عملی

- ۱- پروژه‌ای طراحی کنید که به وسیله آن بتوان نام و نام خانوادگی و شماره تلفن اشخاص را با استفاده از یک پایگاه داده بازیابی نمود.
- ۲- تابعی بنویسید که ساعت ورود و خروج یک اتومبیل را در یک پارکینگ دریافت کرده و مدت زمان توقف وی را محاسبه نموده و بازگشت دهد.
- ۳- رویه‌ای بنویسید که به وسیله آن بتوان هر پیامی را همراه با یک آیکن دلخواه نمایش داد.
- ۴- پروژه‌ای طراحی کنید که به وسیله آن بتوان اطلاعات مربوط به کالاهای یک انبار (نام کالا، کد کالا و موجودی) را در یک آرایه ذخیره نموده و در صورت لزوم اطلاعات یک کالای دلخواه را جستجو نمود و در صورت عدم وجود کالا کادر پیغامی نیز نمایش داده شود.

ضمیمه ۱

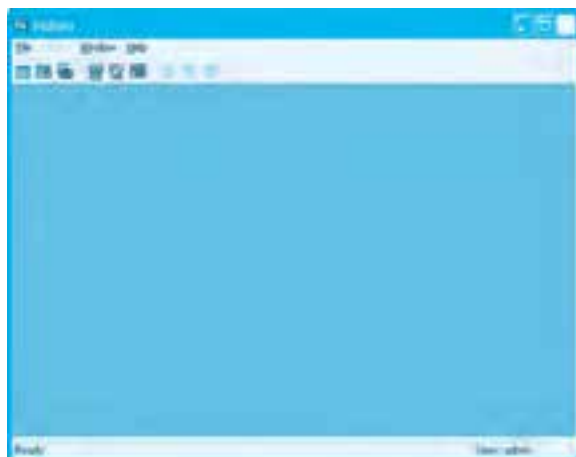
۱- نحوه ایجاد یک فایل پایگاه داده در ویژوال بیسیک

ویژوال بیسیک یک ابزار کارآمد برای ایجاد پایگاه‌های داده به نام Visual Data Manager دارد. برای فعال کردن پنجره برنامه Visual Data Manager که به اختصار آن را VisData می‌نامند، در پنجره برنامه ویژوال بیسیک و نوار منوی آن، روی منوی Add-Ins کلیک کنید و سپس گزینه ... Visual Data Manager را انتخاب کنید (شکل ۱).



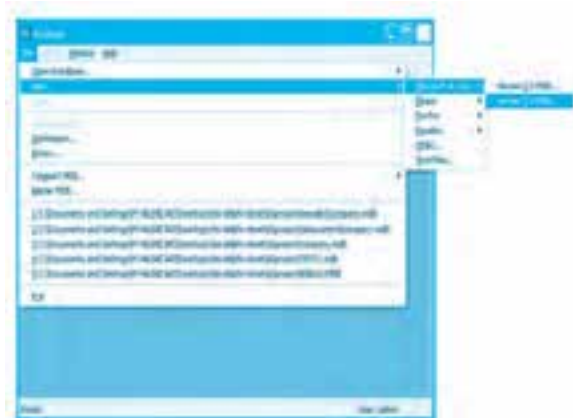
شکل ۱

پنجره برنامه VisData مانند شکل ۲ نمایش داده می‌شود.



شکل ۲

برای ایجاد یک فایل پایگاه داده به وسیله VisData، در پنجره این برنامه و در نوار منوی آن، روی منوی File کلیک کنید، سپس روی گزینه‌های New... و Microsoft Access و بعد روی گزینه Version 7.0 MDB ... کلیک کنید (شکل ۳).



شکل ۳

در این مرحله کادرمحاوره Select Microsoft Access Database to Create مطابق شکل ۴ نمایش داده می‌شود. در این کادرمحاوره نام و مسیر فایل پایگاه داده جدید را تنظیم کنید، سپس روی دکمه فرمان Save کلیک کنید.



شکل ۴

پس از ایجاد فایل پایگاه داده جدید پنجره VisData به صورت شکل ۵ نمایش داده می‌شود.



شکل ۵

همان‌طور که در شکل ۵ مشاهده می‌کنید، دو پنجره دیگر به نام‌های پنجره Database و پنجره دستورات SQL نمایش داده می‌شوند. در پنجره Database، اسامی جداول (Tables) که حاوی فیلدها و رکوردها هستند، نمایش داده می‌شوند و در پنجره دستورات SQL نیز دستوراتی که به زبان SQL نوشته می‌شوند، نمایش داده خواهند شد.

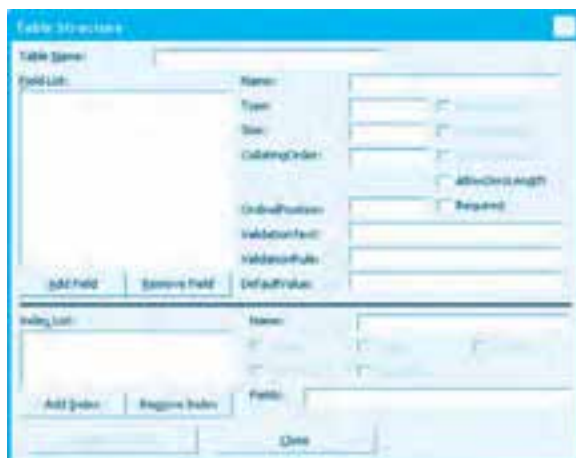
۲- نحوه ایجاد یک جدول (Table)

برای ایجاد یک جدول در پنجره VisData در فضای خالی آن کلیک راست کنید، سپس در منویی که ظاهر می‌شود روی گزینه NewTable کلیک کنید (شکل ۶).



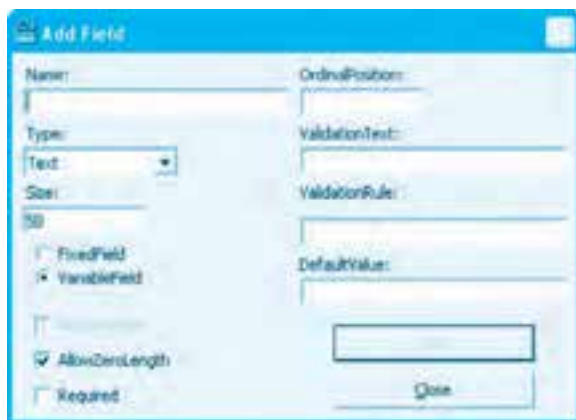
شکل ۶

پس از انتخاب گزینه NewTable کادرمحاوره Table Structure به منظور ایجاد جدول و تعریف فیلدهای موردنظر نمایش داده می‌شود (شکل ۷).



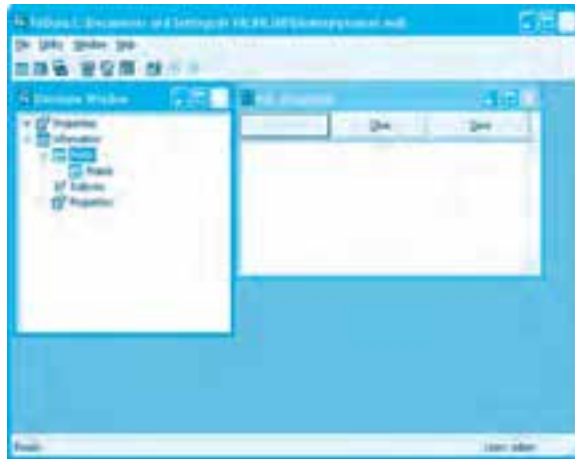
شکل ۷

همان‌طور که در شکل ۷ مشاهده می‌کنید، این کادرمحاوره از اجزای مختلفی تشکیل شده است. در بخش بالایی کادرمحاوره در کادر متن Table Name نام جدول را تایپ کنید. در سمت چپ نیز یک کادر لیست با نام Field List وجود دارد که اسامی فیلدهای جدول در آن نمایش داده می‌شوند. در زیر این کادر لیست نیز دو دکمه فرمان وجود دارد که دکمه فرمان Add Field برای ایجاد یک فیلد جدید و دکمه فرمان Remove Field به منظور حذف یک فیلد از جدول استفاده می‌شوند.



شکل ۸

در صورتی که بخواهید یک فیلد جدید تعریف کنید روی دکمه AddField کلیک کنید تا کادرمحاوره Add Field مطابق شکل ۸ نمایش داده می‌شوند. در این کادرمحاوره نام فیلد را در کادر متن Name تایپ کنید و نوع داده‌ای را که در فیلد ذخیره می‌شود از کادر لیست Type انتخاب کنید، سپس روی دکمه فرمان OK کلیک کنید. پس از تعریف تمام فیلدها کادرمحاوره Add Field را ببندید و در کادرمحاوره Table Structure روی دکمه فرمان Build the Table کلیک کنید تا جدول مربوطه ایجاد شود. پس از خاتمه عملیات ساخت جدول جدید می‌توانید نام جدول را در پنجره Database Window مشاهده کنید و در صورتی که ساختار درختی جدول را باز کنید، می‌توانید اطلاعات مربوط به اجزای تشکیل‌دهنده هر جدول را ببینید (شکل ۹).



شکل ۹



شکل ۱۰

در صورتی که در پنجره Database روی نام جدول کلیک راست کنید (شکل ۱۰) استفاده از گزینه‌های بعد برای انجام عملیات مختلف روی جدول پس از طراحی آن امکان‌پذیر خواهد بود.

جدول ۱

گزینه	عملکرد
Open	پنجره‌ای برای وارد کردن داده‌ها باز می‌شود.
Design	کادرمحاوره Table Structure را برای ویرایش، اضافه و حذف کردن فیلدها نمایش می‌دهد.
Rename	امکان تغییر نام جدول را فراهم می‌کند.
Delete	جدول را از فایل پایگاه داده حذف می‌کند.

۳- نحوه ورود، ویرایش و حذف داده‌ها در جدول

در صورتی که بخواهید مقادیر داده‌ها را وارد جدول کنید، در پنجره Database روی آیکن جدول موردنظر کلیک راست کنید و گزینه Open را انتخاب کنید، پنجره‌ای مانند شکل ۱۱ نمایش داده می‌شود و امکان ورود داده‌ها از طریق این پنجره میسر می‌شود.



شکل ۱۱

در این پنجره اسامی فیلدها براساس ساختار طراحی شده جدول نمایش داده می‌شود و می‌توان مقدار هر فیلد را در کادر متن مربوط به همان فیلد وارد نمود و با کلیک روی دکمه فرمان Update اطلاعات را به

جدول اضافه کرد. در مرحله بعد برای ورود رکوردهای جدید ابتدا باید روی دکمه Add و پس از ورود داده‌ها روی دکمه Update کلیک کرد تا تمام رکوردها وارد شوند. به‌علاوه دکمه‌های فرمان مختلفی برای مدیریت داده‌ها در این پنجره وجود دارند که عملکرد آن‌ها در جدول ۲ ارائه شده‌اند.

جدول ۲

نام دکمه فرمان	عملکرد
Add	اضافه کردن اطلاعات یک رکورد جدید
Update	ویرایش اطلاعات یک رکورد
Delete	حذف اطلاعات یک رکورد
Close	خروج از پنجره Dynaset
Find	پیدا کردن یک رکورد

به علاوه در این پنجره می‌توان با استفاده از دکمه‌های مثلثی که در پایین پنجره قرار دارند، روی رکوردها حرکت نمود.

ضمیمه ۲

نحوه استفاده از SQL Builder در طراحی پرس و جو (Query)

با استفاده از SQL Builder می‌توان سریع و آسان هر نوع پرس و جویی را طراحی نمود. پس از ورود به پنجره SQL Builder می‌توان چهار بخش را در آن مشاهده نمود (شکل ۱). در قسمت بالای پنجره که به آن بخش دیباگرام می‌گویند، جدول یا جدول‌هایی که در دستور Select از آن‌ها استفاده می‌شود همراه روابط بین جدول‌ها نمایش داده می‌شوند. به علاوه در این قسمت می‌توان ستون‌های مورد نظر را برای نمایش در نتایج پرس و جو انتخاب کرد.



شکل ۱

در ستون Output می‌توان مشخص کرد که آیا ستون جدول در نتایج حاصل از اجرای پرس و جو نمایش داده شود یا خیر.

برای این کار می‌توان روی کادر علامت موجود در ستون Output کلیک کرد تا انتخاب شود. در زیر این قسمت بخش Criteria قرار دارد. در این قسمت می‌توان نام ستون‌ها را برای نمایش در نتایج پرس و جو انتخاب کرد. به علاوه در ستون Alias می‌توان یک عبارت را برای ستون انتخاب شده تعیین کرد تا در نتایج پرس و جو به عنوان نام ستون از آن استفاده شود. در ستون Table می‌توان نام جدول را انتخاب کرد.

دو ستون بعدی نحوه مرتب کردن سطرها را در نتایج حاصل از اجرای پرس و جو مشخص می‌کنند. در ستون Sort Type می‌توان نحوه مرتب‌سازی را بر اساس یک ستون تعیین کرد. در این ستون مقدار Ascending برای مرتب‌سازی سطرها به روش صعودی، مقدار Descending برای مرتب‌سازی سطرها به روش نزولی و مقدار Unsorted برای عدم مرتب‌سازی سطرها قابل انتخاب هستند.

اگر بیش از یک ستون برای مرتب‌سازی استفاده شوند با استفاده از ستون Sort Order می‌توان اولویت مرتب‌سازی سطرها را به وسیله انتخاب مقادیر ۱، ۲ و ... برای هر ستون تعیین نمود. در ستون Criteria نیز می‌توان سطرها را بر اساس شرط تعیین شده فیلتر کرد و در ستون ... Or می‌توان شرط دیگری تعیین کرد تا با شرط موجود در ستون Criteria ترکیب منطقی Or را تشکیل دهد. در زیر بخش Criteria، بخش SQL قرار دارد؛ در این بخش دستور Select با توجه به نحوه عملکرد در بخش‌های قبلی به طور خودکار تنظیم شده و نمایش داده می‌شود. به علاوه در این بخش می‌توان دستور را متناسب با نیاز نیز تغییر داد. آخرین قسمت در پنجره SQL Builder بخش نتایج می‌باشد.

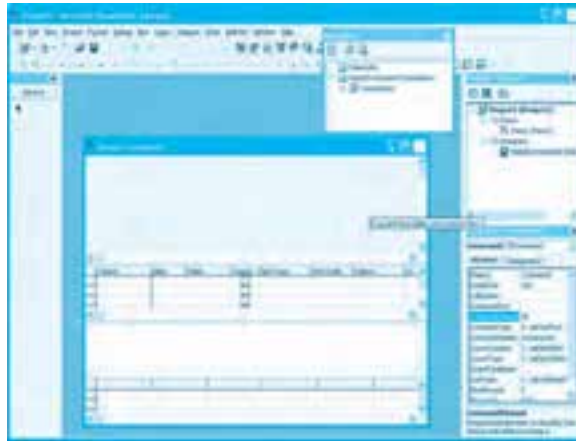
در این بخش می‌توان نتایج حاصل از اجرای پرس‌وجوی طراحی شده را مشاهده نمود. به عنوان مثال فرض کنید می‌خواهید در یک پروژه با استفاده از SQL Builder یک Query طراحی کنید. برای این کار این عملیات را به ترتیب انجام دهید:

- ۱- برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE ایجاد کنید.
- ۲- یک Data Environment به پروژه اضافه کنید. سپس در پنجره Data Environment ایجاد شده یک اتصال Connection برای دسترسی به پایگاه داده Company ایجاد کنید، سپس در اتصال ایجاد شده یک Command جدید ایجاد کنید.
- ۳- روی Command ایجاد شده کلیک راست کرده و در منویی که نمایش داده می‌شود روی گزینه Properties ... کلیک کنید تا کادر محاوره Command Properties مانند شکل ۲ نمایش داده شود.



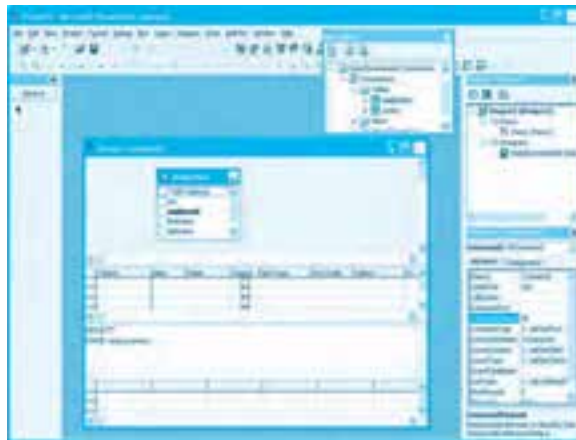
شکل ۲

در این کادر محاوره در زبانه General روی دکمه انتخاب SQL Statement کلیک کنید و بعد روی دکمه فرمان ... SQL Builder روی آن کلیک کنید تا پنجره SQL Builder و Data View نمایش داده شود. (شکل ۳).



شکل ۳

۴- در پنجره Data View ساختار درختی Connection ایجاد شده را باز کنید و در شاخه Tables روی جدول employees کلیک کنید و با عمل کشیدن و انداختن آن را در بخش دیاگرام در پنجره SQL Builder قرار دهید (شکل ۴)



شکل ۴

۵- با همین روش جدول Orders را نیز در پنجره SQL Builder قرار دهید (شکل ۵).



شکل ۵

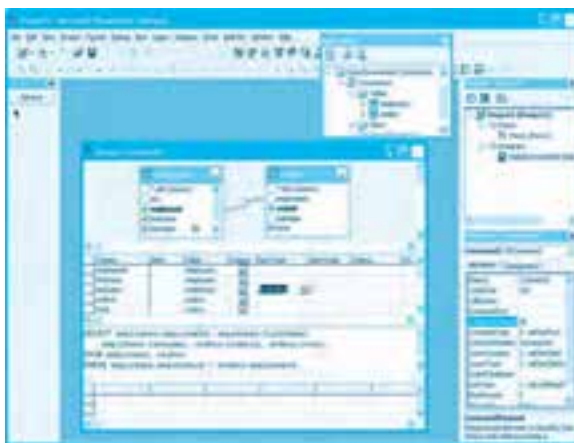
۶- در این مرحله به پنجره SQL Builder و بخش دیاگرام آن بروید. ستون‌های employeeid, firstname, lastname را از جدول employees و ستون‌های orderid و total را از جدول orders انتخاب کنید تا این ستون‌ها در نتایج حاصل از اجرای پرس‌وجو نمایش داده شوند (شکل ۶).



شکل ۶

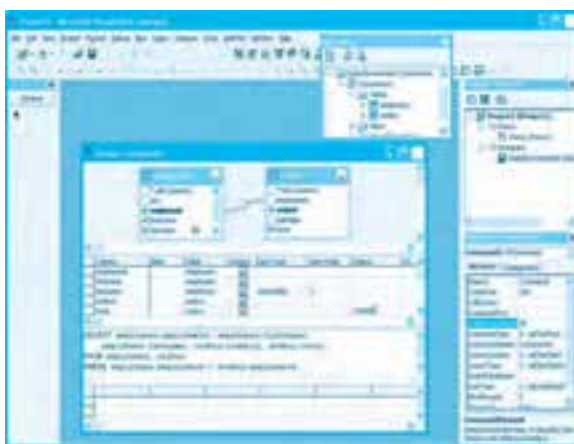
۷- اکنون در پنجره SQL Builder به بخش Criteria بروید و در سطر که ستون lastname قرار دارد در ستون Sort Type کلیک کنید و مقدار Ascending را انتخاب کنید تا سطرها در

نتایج حاصل از اجرای پرس‌وجو بر اساس نام خانوادگی کارمند مرتب شده و نمایش داده شوند (شکل ۷).



شکل ۷

۸- در این مرحله در پنجره SQL Builder و بخش Criteria و در سطر که ستون total قرار دارد در ستون Criteria کلیک کرده و عبارت $40000 >$ را تایپ کنید تا سفارش‌هایی که مبلغ آن‌ها بیشتر از ۴۰۰۰۰ ریال می‌باشد، نمایش داده شوند (شکل ۸).



شکل ۸

۹- اکنون می‌توانید پرس‌وجو را اجرا کرده و نتیجه را در بخش نتایج مشاهده کنید (شکل ۹).



شکل ۹

۱۰- در این مرحله می‌توانید از پنجره SQL Builder و Data Environment خارج شوید و از اجزای ایجاد شده در پروژه خود استفاده کنید.

پاسخنامه

واحدکار ششم

(د-۴)	(الف-۳)	(ب-۲)	(الف-۱)
(الف-۸)	(الف-۷)	(ب-۶)	(ج-۵)
	(a-۱۱)	(ج-۱۰)	(الف-۹)

واحدکار هفتم

(الف-۴)	(الف-۳)	(ب-۲)	(ب-۱)
		(b-۶)	(ج-۵)

واحدکار هشتم

(د-۴)	(د-۳)	(الف-۲)	(د-۱)
(الف-۸)	(ب-۷)	(ج-۶)	(ب-۵)
	(c-۱۱)	(ب-۱۰)	(د-۹)

واحدکار نهم

(الف-۴)	(ج-۳)	(الف-۲)	(ج-۱)
(ج-۸)	(ج-۷)	(الف-۶)	(الف-۵)
	(a-۱۱)	(ج-۱۰)	(د-۹)

منابع

۱- راهنمای MSDN

2- Desktop Applications for Microsoft Visual Basic 6.0 (MSPress)

