

### تولید بسته‌های نرم‌افزاری

هدف‌های رفتاری: پس از آموزش این فصل، هنرجو می‌تواند:

- پروژه‌های خود را اجرایی کند؛
- برای پروژه‌های خود، یک پرونده قابل نصب (Setup) ایجاد کند.

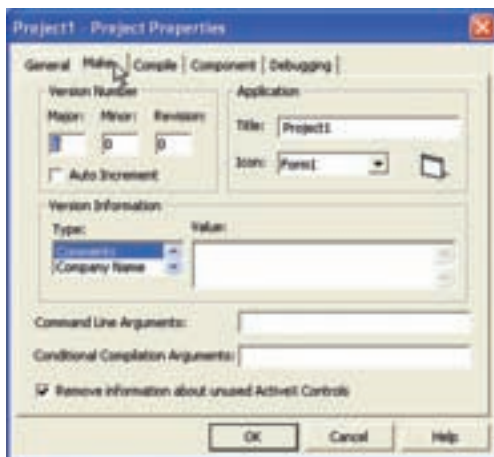
#### ۸-۱- کارکردن با اطلاعات نسخه

یکی از اولین موارد حرفه‌ای که می‌توان به برنامه کاربردی اضافه کرد، درج اطلاعات عمومی متداول برنامه کاربردی، شامل نام شرکت، شماره نسخه و سایر اطلاعات مشابه است. ویژگی بیسیک امکان ذخیره تمام این اطلاعات را با استفاده از شیء App (یک شیء تعریف شده در ویژگی بیسیک که نیاز به ایجاد آن برای برنامه کاربردی ندارید) فراهم می‌کند. اغلب مشخصه‌های شیء App برای ارائه اطلاعات عمومی برنامه کاربردی، مورد استفاده قرار می‌گیرند. جدول ۸-۱ مشخصه‌های متداول را نشان می‌دهد.

## جدول ۱-۸- مشخصه‌های متداول شیء App

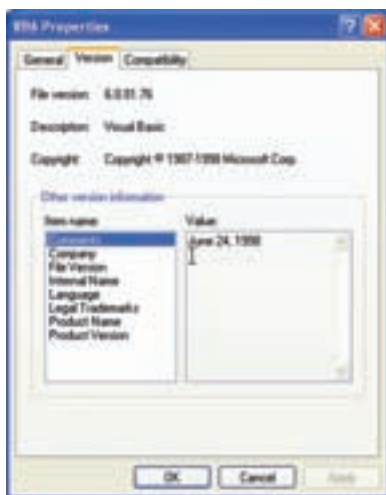
شرح	مشخصه
رشته‌ای را برمی‌گرداند که شامل توضیحاتی درباره برنامه کاربردی است. در زمان اجرا فقط خواندنی است.	Comments
نام شرکت یا تولیدکننده برنامه کاربردی را برمی‌گرداند. در زمان اجرا فقط خواندنی است.	Company Name
نام پرونده اجرایی را بدون پسوند برمی‌گرداند (فقط خواندنی).	EXENAME
رشته‌ای که به طور خلاصه هدف برنامه کاربردی را شرح می‌دهد. در زمان اجرا فقط خواندنی است.	FileDescription
فایل راهنمای مربوط به برنامه کاربردی را تعیین می‌کند. در زمان اجرا خواندنی/نوشتنی است.	HelpFile
رشته حق کپی‌رایت را برمی‌گرداند. از Character Map برای افزودن نمادهای ویژه در این کادر استفاده کنید. در زمان اجرا فقط خواندنی است.	Legal Copyright
اطلاعات علامت تجاری را در صورت نیاز برمی‌گرداند. از Character Map برای اضافه کردن نمادهای ویژه استفاده کنید. در زمان اجرا فقط خواندنی است.	Legal Trademarks
شماره اصلی نسخه را برمی‌گرداند (به عنوان مثال، ۴ در ۳، ۴). در زمان اجرا فقط خواندنی است.	Major
شماره فرعی نسخه را برمی‌گرداند (به عنوان مثال، ۳ در ۳، ۴). در زمان اجرا فقط خواندنی است.	Minor
فهرستی را که برنامه کاربردی در آن قرار دارد برمی‌گرداند. در زمان اجرا فقط خواندنی است.	Path
اگر نمونه‌ای از برنامه کاربردی در حال اجرا باشد، مقداری را برمی‌گرداند. در زمان اجرا فقط خواندنی است.	PrevInstance
نام محصول برنامه کاربردی را برمی‌گرداند. در زمان اجرا فقط خواندنی است.	Product Name
شماره بازنگری برنامه کاربردی را برمی‌گرداند. در زمان اجرا فقط خواندنی است.	Revision

می‌توان از این مشخصه‌ها برای ارائه اطلاعات مهم درباره برنامه کاربردی مورد نظر، استفاده کرد. این مشخصه‌ها در کادر محاوره‌ای Project Properties تعیین می‌شوند (شکل ۸-۱).



شکل ۸-۱- برای دسترسی به کادر محاوره‌ای Project Properties، گزینه Project Name Properties (نام پروژه) را از منوی Project انتخاب کنید.

می‌توان مقادیر این مشخصه‌ها را در زمان اجرا از داخل کد خواند. همچنین می‌توان مقداری را برای مشخصه‌های اطلاعات نسخه شیء App تعیین کرد که انجام این کار، با کلیک راست روی پرونده اجرایی و انتخاب Properties ممکن است (شکل ۸-۲).



شکل ۸-۲- برای مشاهده مشخصه‌های نسخه، زبانه Version را از Properties انتخاب کنید.

مقادیر مشخصه‌های اطلاعات نسخه‌ی شیء App درون قالب دودویی پرونده اجرایی اضافه می‌شوند و در زمان اجرا نمی‌توان آن‌ها را تغییر داد.

```
Private Sub cmdCopyright_Click()
    lblMain.Caption = App.LegalCopyright
End Sub
```

```
Private Sub cmdPath_Click()
    lblMain.Caption = App.Path
End Sub
```

```
Private Sub cmdProductName_Click()
    lblMain.Caption = App.ProductName
End Sub
```

```
Private Sub cmdVersionNum_Click()
    Dim strVerNum
    str VerNum = CStr(App.Major) & "." & CStr(App.Minor) & "." &
    & CStr(App.Revision)
    lblMain.Caption = strVerNum
End Sub
```

## ۸-۲- کامپایل کردن پروژه

بعد از تعیین مقادیر مشخصه‌های شیء App پروژه، می‌توان کد را کامپایل کرد. پروژه‌هایی که تا این مرحله از درس ایجاد کرده‌اید، دارای متن و پرونده‌های گرافیکی بودند که با IDE ویژوال بیسیک به‌وجود می‌آمدند. اکنون زمان آن است که این پرونده‌ها را به یک پرونده اجرایی تبدیل کنید که به‌طور مستقل از

IDE اجرا خواهد شد. این فرایند را کامپایل کردن کد یا ساخت پرونده اجرایی می‌نامند. ویژوال بیسیک از دو قالب برای کامپایل کد پشتیبانی می‌کند؛ P-code یا native code. هنگامی که کد را به P-code کامپایل می‌کنید، پرونده اجرایی به‌وجود آمده به صورت کد مفسری اجرا می‌شود.

## نتیجه

اندازهٔ پرونده اجرایی *native code* نسبت به *P\_code* بزرگ‌تر است. بنابراین، اگر می‌خواهید کوچک‌ترین پرونده اجرایی را ارایه دهید، باید از *P-code* استفاده کنید. ولی اگر می‌خواهید که سریع‌ترین را ارایه دهید، باید *native code* را به وجود آورید.

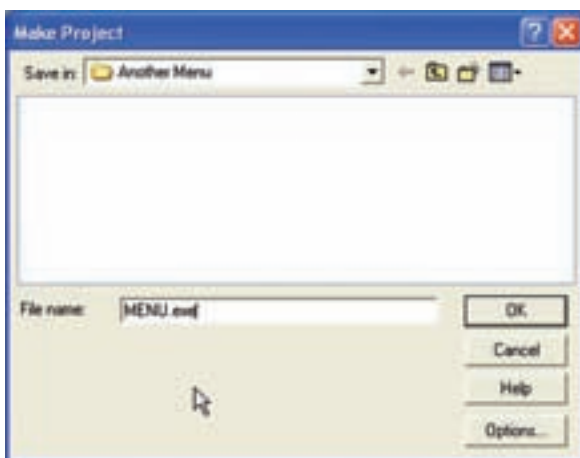
اگر کد را به صورت *native code* کامپایل کنید، پرونده‌های پروژه به کد دودویی کارآمدتری تبدیل می‌شوند که از تمام قابلیت‌های پردازنده استفاده می‌کند. این کد سریع‌تر اجرا خواهد شد ولی *native code* هنوز هم به DLL‌های زمان اجرا نیاز دارد (تنها تفاوت این است که DLL‌ها به وسیلهٔ EXE به طور متفاوت مورد دسترسی و استفاده واقع می‌شوند).

### ۸-۲-۱ کامپایل کد به Standard EXE

۱- پروژه‌ای را که می‌خواهید کامپایل کنید باز کنید.

۲- از منوی File گزینهٔ Make Project Name.exe را انتخاب کنید. کادر محاوره‌ای Make

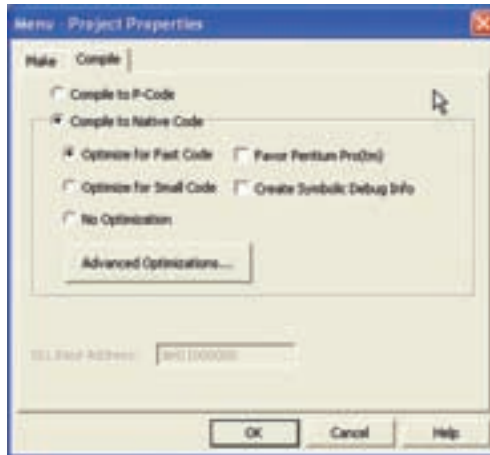
Project ظاهر می‌شود (شکل ۸-۳).



شکل ۸-۳ نام پرونده اجرایی را در این کادر محاوره‌ای وارد کنید. برای تغییر بعضی از مشخصه‌های شیء App روی دکمهٔ Options کلیک کنید.

۳- در صورتی که می‌خواهید، نام پرونده اجرایی را در کادر متن File Name تغییر دهید، روی دکمه Options کلیک کنید تا کادر محاوره‌ای Properties باز شود. در زبانه Compile یکی از دو روش کامپایل P-code یا Native code را انتخاب کنید.

۴- در کادرهای محاوره‌ای Project Properties و Make Project روی Ok کلیک کنید تا کد کامپایل شود.

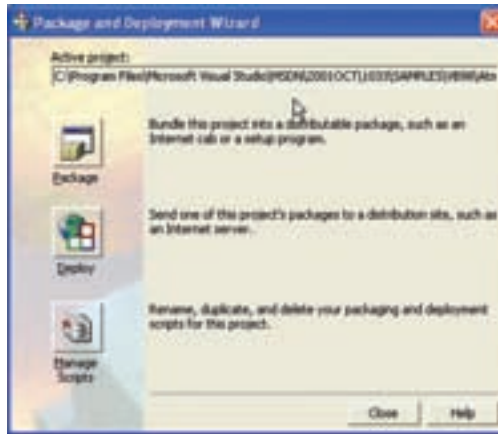


شکل ۸-۴- می‌توان چندین گزینه برای کامپایل Native code انتخاب کرد. به‌طور کلی، انتخاب کد سریع‌تر، پرونده اجرایی بزرگ‌تری را ایجاد می‌کند.

کامل کردن این فرایند، یک پرونده اجرایی را تولید می‌کند که می‌توان خارج از IDE ویرژوال بیسیک اجرا کرد. با این وجود برنامه هنوز برای ارایه آماده نیست. برای تحویل برنامه کاربردی، نیاز به اجرای Application Setup Wizard برای پرونده اجرایی دارید که امکان اجرا روی سیستمی که ویرژوال بیسیک نصب نشده است را فراهم می‌کند.

### ۸-۳- کاربرد Package and Deployment Wizard

از این ابزار می‌توان برای ایجاد بسته‌های نرم‌افزاری که نصب می‌شوند (Setup) از هر نوع برنامه کاربردی در ویرژوال بیسیک استفاده کرد. برای انجام این کار، Package and Deployment Wizard را شروع کنید (شکل ۸-۵). این ابزار باید در زیر منوی Visual Basic از منوی Start لیست شده باشد، زیرا به‌طور پیش‌فرض، نصب می‌شود. در صورتی که نیست، باید آن را روی سیستم نصب کنید.



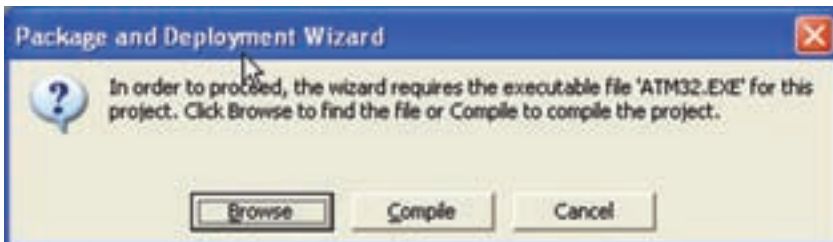
شکل ۸-۵- ابزار Package and Deployment Wizard

مراحل ایجاد یک بسته نرم‌افزاری که نصب می‌شود، به صورت زیر است :  
 ۱- پرونده پروژه را در بالای کادر محاوره‌ای انتخاب کنید.

## نتیجه

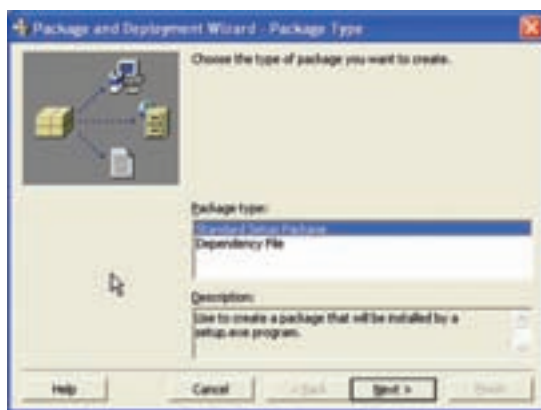
برای این مراحل، می‌توانید از هر پروژه‌ای که در ویژوال بیسیک دارید یا از نمونه‌های خود ویژوال استفاده کنید. ما از پروژه ATM موجود در ویژوال بیسیک استفاده کرده‌ایم.

- ۲- برای ساخت یک برنامه نصب (setup)، روی دکمه Package کلیک کنید.
- ۳- در صورتی که برنامه کاربردی را کامپایل نکرده باشید، ویزارد از شما کامپایل برنامه را سؤال می‌کند (شکل ۸-۶). برای ادامه روی دکمه Compile کلیک کنید.



شکل ۸-۶- ویزارد نیاز به پرونده اجرایی در داخل برنامه Setup دارد، بنابراین به طور خودکار پرونده پروژه برنامه کاربردی را کامپایل می‌کند.

۴- بعد از کامپایل برنامه کاربردی، از شما خواهد پرسید که چه نوع بسته نرم‌افزاری می‌خواهید (شکل ۷-۸). گزینه Standard Setup Package را انتخاب کرده و روی OK کلیک کنید.

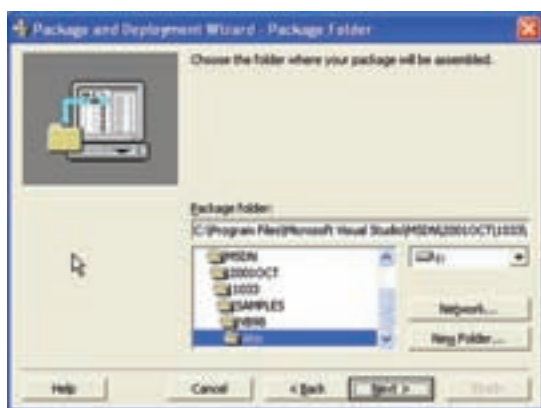


شکل ۷-۸- نوع بسته نرم‌افزاری را که می‌خواهید با ویزارد ایجاد کنید انتخاب کنید.

## توجه

پرونده‌های فلاپی دیسک یا CD-ROM را در این مرحله برای ویزارد ارایه نکنید، زیرا قبل از کامل شدن فرایند، ویزارد چندین بار پرونده‌ها را تغییر می‌دهد. پرونده‌ها را روی هارد دیسک ذخیره کنید و بعد از پایان فرایند آن‌ها را روی CD کپی کنید.

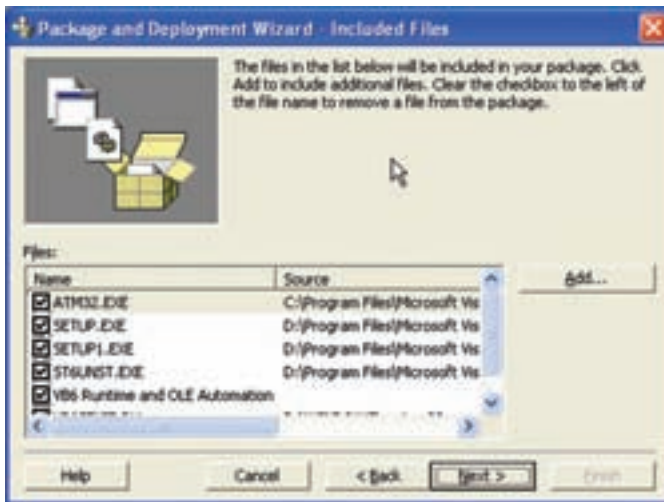
۵- محل پرونده‌های نصب را، که ویزارد ایجاد خواهد کرد، تعیین کنید. این پرونده‌ها به رسانه مقصد که ممکن است دیسک یا CD باشد، کپی شوند. فهرستی را انتخاب کرده (شکل ۸-۸) و برای ادامه روی Next کلیک کنید.



شکل ۸-۸- ویزارد نیاز به محلی برای نگهداری موقتی پرونده‌های موردنظر برای نصب دارد.



۶- همان‌طور که قبلاً نیز بیان شد، ویژوال بیسیک پرونده‌های زیادی به غیر از پرونده اجرایی دارد. کادر محاوره‌ای بعدی (شکل ۹-۸)، پرونده‌های مورد نیاز برای نصب را به همراه پرونده اجرایی لیست می‌کند. اگر پرونده‌های دیگری (مثل پرونده‌های راهنما) دارید که باید نصب شوند، می‌توانید آن‌ها را در این مرحله اضافه کنید. روی دکمه next کلیک کنید.



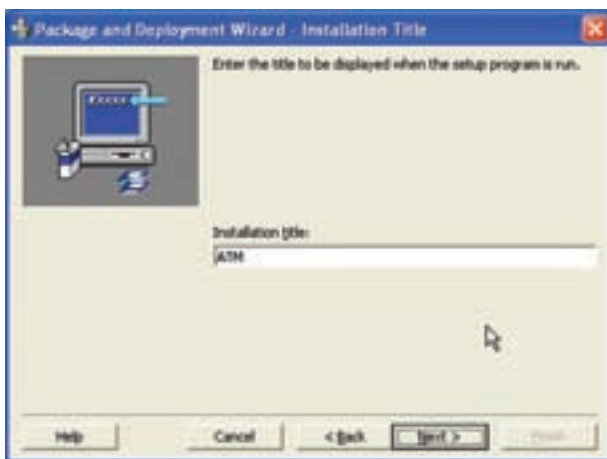
شکل ۹-۸- این کادر محاوره‌ای، لیست کاملی از پرونده‌های مورد نیاز برای ایجاد برنامه کاربردی را که روی کامپیوتر دیگری اجرا می‌شود نشان می‌دهد.

## نتیجه

پرونده‌هایی را که ویزارد تولید خواهد کرد پرونده‌های CAB (کابینت) می‌نامند. این پرونده‌ها نوع خاصی از پرونده‌های آرشیوی هستند که به وسیلهٔ مایکروسافت طراحی شده و خیلی شبیه به پرونده Zip هستند. اگر از نرم‌افزار Winzip نسخهٔ ۷.۰ یا بالاتر استفاده می‌کنید، می‌توانید محتوای پرونده‌های CAB را مشاهده کنید.

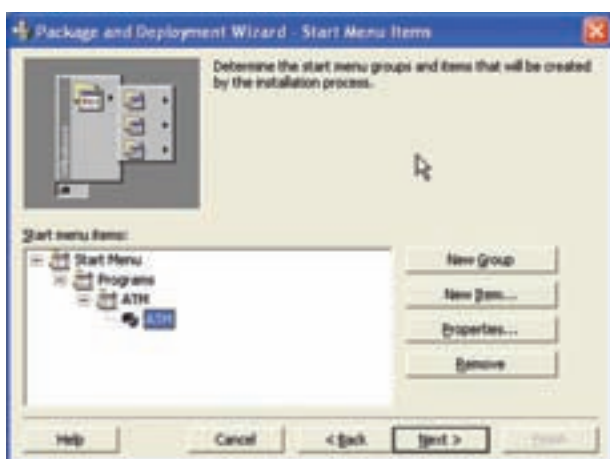
۷- مرحلهٔ بعدی در ویزارد تعیین اندازهٔ رسانهٔ مقصد است. اگر قصد دارید برنامهٔ کاربردی را روی فلاپی دیسک‌ها قرار دهید، بزرگ‌ترین پرونده‌ای که ویزارد می‌تواند تولید کند به اندازهٔ دیسک خواهد بود. برای کپی پرونده‌ها روی فلاپی دیسک، Single Cab را انتخاب کرده و روی Next کلیک کنید تا ادامه یابد.

۸- عنوان برنامه کاربردی را تعیین کنید (شکل ۸-۱۰). این عنوان در طول نصب نمایش داده خواهد شد. عنوان مناسبی را وارد کرده و برای ادامه روی Next کلیک کنید.



شکل ۸-۱۰- عنوان برنامه کاربردی را در این کادر محاوره‌ای وارد کنید.

۹- برای برنامه کاربردی که استفاده می‌کنید، نیاز دارید نشانه‌ای در منوی Start داشته باشید. مرحله بعدی ویزارد (شکل ۸-۱۱) یک روش منحصر به فرد برای تعیین گروه‌هایی از نشانه‌هاست. تنظیمات پیش فرض، ایجاد گروه با نام برنامه کاربردی و سپس ایجاد نشانه‌ای برای شروع برنامه است.



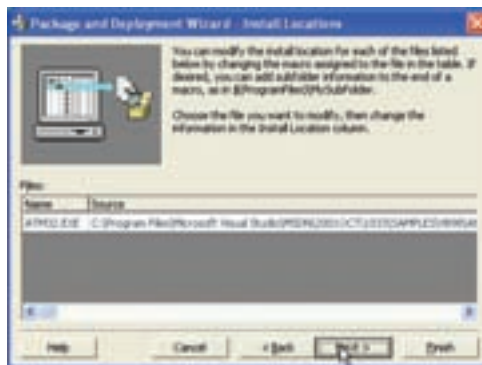
شکل ۸-۱۱- نشانه‌ها و گروه‌های موردنظر را برای برنامه کاربردی انتخاب کنید.

به دلیل این که برنامه کاربردی فقط دارای یک آیکن است، به طور استاندارد نشانه آن تحت All Programs ایجاد می شود. ابتدا روی گروه ATM و بعد روی دکمه Remove کلیک کنید. سپس روی دکمه New Item کلیک کرده و نام برنامه کاربردی را وارد کنید، آن گاه در کادر محاوره ای که ظاهر می شود، روی Ok کلیک کنید (شکل ۸-۱۲). بعد از پایان اضافه کردن گروه ها و نشانه ها روی Next کلیک کنید.



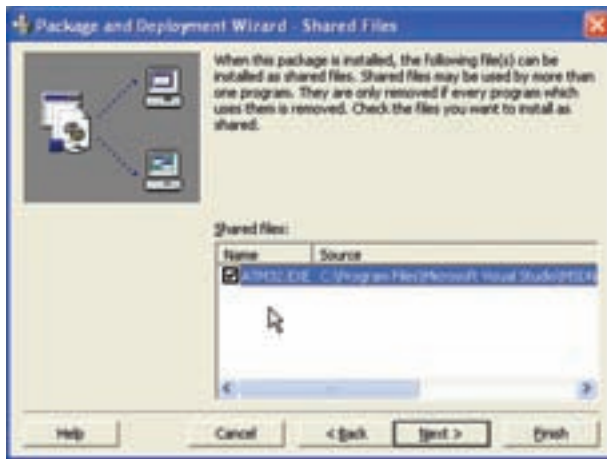
شکل ۸-۱۲- گزینه های اصلی را برای هر نشانه ای که می خواهید ایجاد کنید تعیین کنید.

۱۰- کادر محاوره ای بعدی (شکل ۸-۱۳) امکان تغییر محل نصب هر پرونده ای را که به وسیله سیستم مورد نیاز نیست فراهم می کند. تمام پرونده های سیستمی به طور خودکار در فهرست windows\system نصب می شوند، مگر این که محل دیگری را تعیین کنید. به دلیل این که فهرست تعیین شده برای این مثال صحیح است، روی Next کلیک کنید.

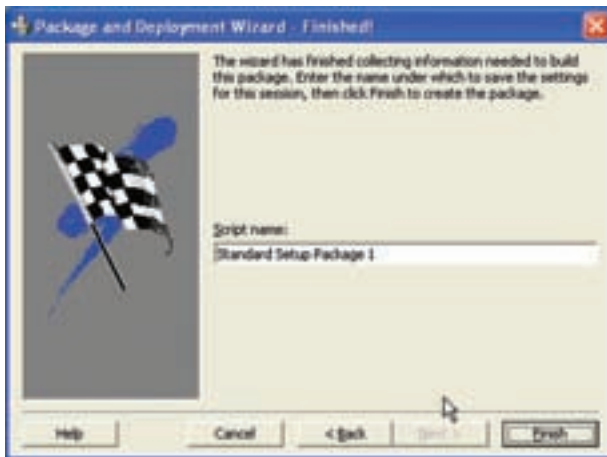


شکل ۸-۱۳- اگر نیاز به تغییر فهرست نصب برای پرونده هایی که اضافه کرده اید دارید، این کادر محاوره ای امکان انجام این کار را فراهم می کند.

۱۱- پرونده‌های اصلی (مثل DLLها و OCXها)، پرونده‌های اشتراکی هستند. اگر هر کدام از این پرونده‌ها را به عنوان بخشی از نصب اضافه کنید، آن‌ها به صورت اشتراکی نشانه‌گذاری می‌شوند. بنابراین، هنگامی که کاربران برنامه کاربردی را حذف (Uninstal) کنند، پرونده‌های به اشتراک گذاشته شده قبل از حذف، اصلاح می‌شوند. کادر محاوره‌ای نشان داده شده در شکل ۱۴-۸ امکان نشانه‌گذاری همه پرونده‌های اشتراکی را فراهم می‌کند. روی Next کلیک کنید.



شکل ۱۴-۸- پرونده‌های اشتراکی را در این کادر محاوره‌ای نشانه‌گذاری کنید.



شکل ۱۵-۸- آخرین مرحله، ویزارد امکان ارایه نامی برای کاربرد بعدی را فراهم می‌کند.

۱۲- نامی را تعیین کنید (شکل ۱۵-۸) و روی دکمه Finish کلیک کنید تا بسته نرم‌افزاری نصب ایجاد شود. ۱۳- بعد از این که ویزارد ایجاد بسته نرم‌افزاری نصب را خاتمه داد، گزارشی را با چند پیام مهم تولید می‌کند. گزارش را بخوانید و سپس در گزارش و ویزارد روی Close کلیک کنید.

۱- پروژه‌ای را به صورت Setup شده به هنرآموز خود تحویل دهید.

۲- مقایسه‌ای بین یکی از برنامه‌های

Package and deployment, Setup factory, Install sheild, Wyse installer

از نظر امکانات ساخت برنامه نصب انجام دهید.

۳- برنامه نصب پروژه پایانی خود را با استفاده از یکی از سه برنامه معرفی شده در خودآزمایی

۲ به صورت setup درآورید.

## پروژه ۱: طراحی یک مسابقه

در این مسابقه ۸ سؤال به همراه پاسخ‌هایشان روی فرم وجود دارد. ابتدا سؤال موردنظر را انتخاب می‌کنیم و سپس پاسخ مناسب آن را. اگر پاسخ درست انتخاب شده باشد، سؤال و پاسخ هر دو از روی فرم حذف می‌شود ولی اگر پاسخ اشتباه انتخاب شده باشد، سؤال و پاسخ به همان صورت اولیه باقی می‌ماند. یک زمان‌سنج در پایین فرم وجود دارد تا زمان باقی‌مانده را اعلام کند. اگر به تمامی سؤالات در زمان موردنظر پاسخ درست دهید پیام «آفرین...» می‌دهد و اگر زمان به صفر برسد و هنوز سؤال بدون پاسخ مانده باشد پیام «متأسفم...» می‌دهد.

\* ما این مسابقه را با ۸ سؤال طراحی کرده‌ایم ولی می‌توانید آن را به مسابقه‌ای با سؤالات

بیش‌تر تعمیم دهید.

### کنترل‌های موردنیاز:

۱- دو کنترل label برای عنوان‌های «سؤالات» و «نتایج»

۲- ۸ کنترل Command برای سؤالات و ۸ کنترل Command برای پاسخ آن‌ها

۳- یک کنترل Timer برای نگهداری زمان روی فرم قرار می‌دهیم.

۴- یک کنترل label در پایین صفحه برای نمایش زمان باقیمانده.

### تحلیل برنامه

وقتی تعدادی کنترل یکسان داریم که مثل هم کار می‌کنند بهتر است آن‌ها را به صورت آرایه کنترلی استفاده کنیم. به این ترتیب آرایه کنترلی برای سؤالات و آرایه کنترلی دیگری برای پاسخ‌ها خواهیم داشت و برای حل بهتر این برنامه، Command‌های با اندیس یکسان را برای سؤال و پاسخ مرتبط با هم قرار

می‌دهیم. فقط دقت کنید چیدمان Command‌ها طوری باشد که سؤال و جواب مرتبط با هم مقابل یکدیگر نباشند.

### طراحی برنامه

یک کنترل Command را روی فرم آورده، ظاهر آن را تنظیم کنید و مشخصه Name آن را cmdq می‌نویسیم و سپس ۷ کپی از آن می‌گیریم و اندیس این آرایه کنترلی صفر تا هفت خواهد بود. (سؤالات) یک کنترل Command دیگر روی فرم



آورده و مثل قبلی عمل می‌کنیم فقط مشخصه Name آن را CmdA قرار می‌دهیم. (پاسخ‌ها)  
یک کنترل Timer را روی فرم آورده و مشخصه Interval آن را 500 می‌دهیم.  
یک کنترل label به نام Label1 در پایین فرم قرار داده و مشخصه Caption آن را 60 می‌دهیم.

### کدنویسی

۱- سؤالات و پاسخ آن‌ها را طرح کرده و روی Commandها قرار می‌دهیم. این عمل را در Form-load انجام می‌دهیم.

```
Private sub Form-load()
```

```
k=0
```

```
Cmdq(0).Caption = "کنترلی که قادر است تصویر را به اندازه خود درآورد"
```

```
Cmdq(1).Caption = "پسوند فایل پروژه چیست"
```

```
Cmdq(2).Caption = "کدام کنترل هنگام اجرا دیده نمی‌شود"
```

```
Cmdq(3).Caption = "کدام خصوصیت در تمام کنترل‌ها وجود دارد"
```

```
Cmdq(4).Caption = "کدام کنترل رویداد ندارد"
```

```
Cmdq(5).Caption = "کدام کنترل برای گروه‌بندی کنترل‌هاست"
```

```
Cmdq(6).Caption = "تابع چیست"
```

```
Cmdq(7).Caption = "یک متد گرافیکی"
```

```
CmdA(0).Caption = "ImageImag"
```

```
CmdA(1).Caption = "vbp"
```

```
CmdA(2).Caption = "Timer"
```

```
CmdA(3).Caption = "name"
```

```
CmdA(4).Caption = "Shape"
```

```
CmdA(5).Caption = "Frame"
```

```
CmdA(6).Caption = "روالی که مقداری را بر می‌گرداند"
```

```
CmdA(7).Caption = "pset"
```

```
end sub
```

۲- می‌خواهیم هرگاه روی سؤالی کلیک شد اندیس آن را در جایی نگهداری کنیم تا بتوانیم آن را با اندیس پاسخ مقایسه کنیم.

```
Private sub cmdq-click(index as integer)
```

```
    i = index
```

```
end sub
```

```
private Sub cmdA_Click (index as integer)
```

```
    if i = index then
```

```
        cmdq(i).visible = false
```

```
        cmdA(index).visible = false
```

```
        k = k+1
```

```
    end if
```

```
end sub
```

۳- در رویداد CmdA-Click متغیری به نام k می‌بینیم که تعداد پاسخ‌هایی که خصوصیت visible آن False است را می‌شمارد تا هرگاه تعداد آن به ۸ رسید یعنی به همه سؤالات، پاسخ درست داده شده است و این شرط اعلام پیام برنده شدن در مسابقه است که در رویداد Timer از Timer1 قرار می‌دهیم.

```
Private sub Timer1-timer()
```

```
    If k = 8 Then
```

```
        Timer1 . enabled = False
```

```
        msgbox (« آفرین ... »)
```

```
    end
```

```
end if
```

```
    If label1 . Caption>0 then
```

```
        label1 . Caption = label1 . Caption-1
```

```
    else
```

```
        Timer1 . enabled = False
```

```
        msgbox (« متأسفم ... »)
```

```
    end
```

```
end if
```



end sub

۴- در این برنامه متغیرهای  $i$  و  $k$  استفاده شده پس باید این متغیرها را تعریف کنیم برای این کار در قسمت general می نویسیم:

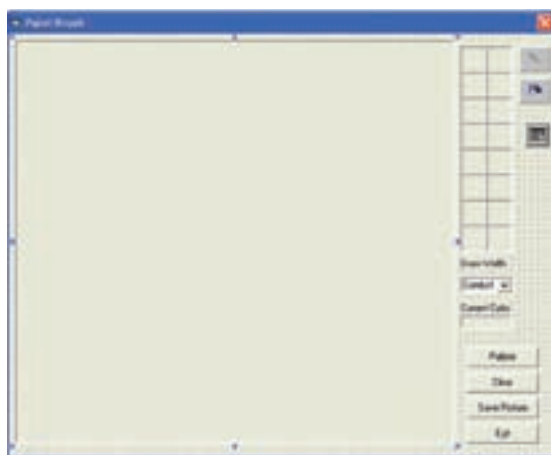
Dim I as integer

Dim K as integer



## پروژه ۲: PaintBrush

فرمی به شکل زیر ایجاد کنید



کد این پروژه به صورت زیر خواهد بود :

```
Dim IntOldX As Integer, IntOldY As Integer, LngColor As Long
```

```
Dim IntLineSize As Byte
```

```
Private Sub CboLineSize_Click()
```

```
IntLineSize = Val(CboLineSize.Text)
```

```
Pic1.DrawWidth = IntLineSize
```

```
End Sub
```

```
Private Sub CmdClear_Click()
```

```
    Pic1.Cls
```

```
End Sub
```

```
Private Sub CmdExit_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub CmdPallete_Click()
```

```
    CommonDialog1.ShowColor
```

```
    LngColor = CommonDialog1.Color
```

```
    LBLCurrentColor.BackColor = LngColor
```

```
End Sub
```

```
Private Sub CmdSave_Click()
```

```
    Dim StrFn As String
```

```
    With CommonDialog1
```

```
        .Filter = "BitMap Format|*.bmp"
```

```

.ShowSave
StrFn = UCase(.FileName)
If StrFn <> "" Then
SavePicture Pic1. Image, StrFn
End If
End With
End Sub

```

```

Private Sub Form_Load()
Dim IntI As Integer
Pic1.AutoRedraw = True
Pic1.BackColor = QBColor(15)
Pic1.ScaleMode = vbPixels
Pic1.MousePointer = vbCrosshair
'Set Color Box
For IntI = 0 To 15
    LBLColor(IntI).BackColor = QBColor(IntI)
Next
'Set CurrentColor
LngColor = 0
LBLCurrentColor.BackColor = QBColor (LngColor)
'Fill Draw Width 1 to 5 in ComboBox
For IntI = 1 To 5
    CboLineSize.AddItem IntI
Next
'Set DrawWidth to 1
CboLineSize.ListIndex = 0

```

```

IntLineSize = Val(CboLineSize.Text)
Pic1.DrawWidth = IntLineSize
'Set Drawing Line Option
OptLine.Value = True
End Sub

```

```

Private Sub LBLColor_Click(Index As Integer)
    LngColor = LBLColor(Index).BackColor
    LBLCurrentColor.BackColor=LngColor
End Sub

```

```

Private Sub Pic1_MouseDown(Button As Integer, Shift As Integer,X As Single,
Y As Single)
    IntOldX = X
    IntOldY= Y
End Sub

```

```

Private Sub Pic1_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
Dim IntI As Integer, IntR As Integer, IntA As Integer
Dim SngPI As Single
    SngPI = 4* Atn(1)
    Randomize Timer
    If Button = 1 Then
        If OptLine.Value = True Then
            Pic1.Line(IntOldX,IntOldY)-(X,Y),LngColor
            IntOldX = X
            IntOldY = Y

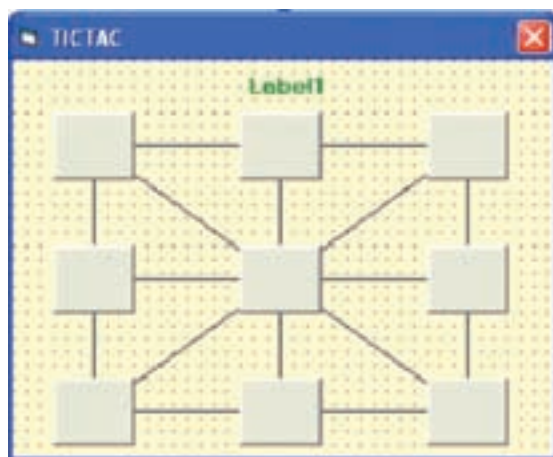
```

```

ElseIf OptSpray.Value = True Then
    For IntI= 1 To 10
        IntA = Rnd*2*SngPI
        IntR = Rnd*15
        Pic1.PSet (X+IntR*Cos(IntA), Y+IntR*Sin(IntA)),LngColor
    Next
End If
End If
End Sub

```

## پروژه ۳: بازی XO



```
Dim StrPlayer As String
```

```
Private Sub CLEAR()
```

```
Dim Inti As Variant
```

```
For Each Inti In Controls
```

```
    If TypeOf Inti Is CommandButton Then
```

```

        Inti.Caption = ""
    End If
    If TypeOf Inti Is Line Then
        Inti.BorderWidth = 1
        Inti.BorderColor = RGB(0, 0, 0)
    End If
Next Inti
End Sub

```

```

Private Sub CHECK()

```

```

    a = cmd(0).Caption + cmd(1).Caption + cmd(2).Caption

```

```

    If a = "XXX" Then Line1.Border Width = 3: Line1.BorderColor = RGB(255, 0,
0):

```

```

    MsgBox "Playaer X Win": CLEAR

```

```

    If a = "OOO" Then Line1.Border Width = 3: Line1.BorderColor = RGB(0, 255,
0):

```

```

    MsgBox "Playaer O Win": CLEAR

```

```

    a = cmd(3).Caption + cmd(4).Caption + cmd(5).Caption

```

```

    If a = "XXX" Then Line2.Border Width = 3: Line2.BorderColor = RGB(255, 0,
0):

```

```

    MsgBox "Playaer X Win": CLEAR

```

```

    If a = "OOO" Then Line2.Border Width = 3: Line2.BorderColor = RGB(0, 255,
0):

```

```

    MsgBox "Playaer O Win": CLEAR

```

```

    a = cmd(6).Caption + cmd(7).Caption + cmd(8).Caption

```

If a = "XXX" Then Line3.Border Width = 3: Line3.BorderColor = RGB(255, 0, 0):

MsgBox "Playaer X Win": CLEAR

If a = "OOO" Then Line3.Border Width = 3: Line3.BorderColor = RGB(0, 255, 0):

MsgBox "Playaer O Win": CLEAR

a = cmd(0).Caption + cmd(3).Caption + cmd(6).Caption

If a = "XXX" Then Line4.BorderWidth = 3: Line4.BorderColor = RGB(255, 0, 0):

MsgBox "Playaer X Win": CLEAR

If a = "OOO" Then Line4.BorderWidth = 3: Line4.BorderColor = RGB(0, 255, 0):

MsgBox "Playaer O Win": CLEAR

a = cmd(1).Caption + cmd(4).Caption + cmd(7).Caption

If a = "XXX" Then Line5.Border Width = 3: Line5.BorderColor = RGB(255, 0, 0):

MsgBox "Playaer X Win": CLEAR

If a = "OOO" Then Line5.BorderWidth = 3: Line5.BorderColor = RGB(0, 255, 0):

MsgBox "Playaer O Win": CLEAR

a = cmd(2).Caption+cmd(5).Caption + cmd(8).Caption

If a = "XXX" Then Line6.BorderWidth = 3: Line6.BorderColor = RGB(255, 0, 0):

MsgBox "Playaer X Win": CLEAR

If a = "OOO" Then Line6.BorderWidth = 3: Line6.BorderColor = RGB(0, 255, 0):

0):

```
MsgBox "Playaer O Win": CLEAR
```

```
a = cmd(0).Caption + cmd(4).Caption + cmd(8).Caption
```

```
If a = "XXX" Then Line7.BorderWidth = 3: Line7.BorderColor = RGB(255, 0,  
0):
```

```
MsgBox "Playaer X Win": CLEAR
```

```
If a = "OOO" Then Line7.BorderWidth = 3: Line7.BorderColor = RGB(0, 255,  
0):
```

```
MsgBox "Playaer O Win": CLEAR
```

```
a = cmd(2).Caption + cmd(4).Caption + cmd(6).Caption
```

```
If a = "XXX" Then Line8.BorderWidth = 3: Line8.BorderColor = RGB(255, 0,  
0):
```

```
MsgBox "Playaer X Win": CLEAR
```

```
If a = "OOO" Then Line8.BorderWidth = 3: Line8.BorderColor = RGB(0, 255,  
0):
```

```
MsgBox "Playaer O Win": CLEAR
```

```
End Sub
```

```
Private Sub cmd_Click(Index As Integer)
```

```
If cmd(Index).Caption = "" Then
```

```
    cmd(Index).Caption = StrPlayer
```

```
    If StrPlayer = "X" Then
```

```
        StrPlayer = "O"
```

```
        LBL1.Caption = "PLAYER2 O"
```

```
    Else
```



```
StrPlayer = "X"  
LBL1.Caption = "PLAYER1 X"  
End If  
CHECK  
End If  
End Sub  
  
Private Sub Form_Load()  
StrPlayer = "X"  
LBL1.Caption = "PLAYER1 X"  
CLEAR  
End Sub
```

توضیحات این پروژه را به عهده هنرجویان واگذار می‌کنیم.

## پروژه‌های پیشنهادی

- ۱- با کمک و راهنمایی هنرآموز درس پروژه PaintBrush را کامل کنید.
- ۲- بازی منچ را شبیه‌سازی کنید.
- ۳- یک پازل تصویری طراحی و امکان مرتب نمودن آن را فراهم کنید.

