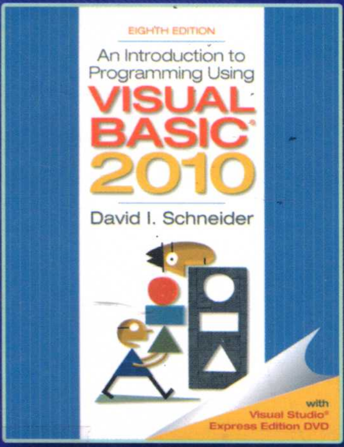
国外计算机科学教材系列

PEARSON

Visual Basic 2010 程序设计教程（第八版）

An Introduction to Programming Using Visual Basic 2010, Eighth Edition



EIGHTH EDITION

An Introduction to Programming Using VISUAL BASIC® 2010

David I. Schneider

with Visual Studio® Express Edition DVD

英文版

[美] David I. Schneider 著

罗 凌 改编

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

国外计算机科学教材系列

Visual Basic 2010
程序设计教程

（第八版）

（英文版）

An Introduction to Programming Using Visual Basic 2010

Eighth Edition

［美］ David I. Schneider 著

罗 凌 改编

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

David I. Schneider的Visual Basic系列教材自出版以来，深受读者欢迎，并被很多大学采用作为Visual Basic程序设计的教材。本书不是粗略地论及众多主题，而是针对重要问题进行深入分析。全书共分为10章，主要内容包括：Visual Basic简介、控件与事件，变量、输入和输出，分支结构，通用过程，循环结构，数组，其他控件和对象，面向对象编程，数据库，以及文本文件。全书给出了100多个示例和大量的习题，并且提供了实践性很强的程序设计项目，帮助读者掌握所学知识。

本书不仅可以作为Visual Basic程序设计课程的双语教材，也是广大计算机爱好者及程序开发人员学习Visual Basic的很好的参考用书。

图书在版编目（CIP）数据

Visual Basic 2010程序设计教程 = An Introduction to Programming Using Visual Basic 2010：第八版：英文 /（美）施奈德（Schneider, D. I.）著. 罗凌改编. — 北京：电子工业出版社，2012.9

国外计算机科学教材系列

ISBN 978-7-121-17722-4

Ⅰ. ①V… Ⅱ. ①施… ②罗… Ⅲ. ①BASIC语言－程序设计－高等学校－教材－英文 Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2012）第168906号

策划编辑：冯小贝
责任编辑：冯小贝
印　　刷：
装　　订：涿州市京南印刷厂
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路173信箱　邮编：100036
开　　本：787 × 1092　1/16　　印张：28.75　　字数：957千字
印　　次：2012年9月第1次印刷
定　　价：59.80元

凡所购买电子工业出版社的图书有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

导 读

Visual Basic是非常流行的编程语言，Visual Basic 2010是其最新版本，其功能比以前任何一个版本都强大，利用该语言能快速、高效地开发出各种类型的应用程序。“Visual Basic程序设计”课程也是计算机专业学生一门重要的专业选修课。

本书作者David I. Schneider任教于美国马里兰大学，是一位有着30多年教学经验的语言大师。作者从2004年开始至今出版了一系列Visual Basic教材，从Visual Basic 6.0、Visual Basic .NET、Visual Basic 2005、Visual Basic 2008到Visual Basic 2010，本书是该系列的最新版本。该系列教材具有以下特点：

1. 教材不是广泛地堆积众多的知识内容，而是针对重要知识点进行深入全面的讲解。
2. 知识讲解深入浅出，很多知识点不像多数国内教材直接先给出理论描述，而是直接从应用展开，这样更利于学生的理解。每个知识点都配有丰富的实例，实例来自于实际生活，使用的都是真实的数据，更易激发学生的学习兴趣。
3. 和国内大多数教材只是在每章结束后才配有习题不同，该系列教材中每个小节结束后就提供了大量的实践问题、习题等，习题总数超过1000道，而且基本上每章结束后都有多个实践性和综合性较强的程序设计项目，使得学生能通过大量的理论和实践的练习来深入掌握各个知识点。

自2005年，我开始采用该系列中的*An Introduction to Programming Using Visual Basic .NET*一书作为计算机科学与技术专业“Visual Basic程序设计”双语课程的教材，取得了较好的教学效果。但是在实际的使用过程中发现，由于教学课时有限，书中涉及的知识点和习题无法全部讲解和完成，而且部分章节的体系结构和课程教学大纲有一定出入。鉴于此，在改编本书时，本人将有关章节进行了相应的调整，并将部分内容和习题进行了删减，使得该书更符合国内课程的双语教学。

全书的主体结构如下：

- 第1章～第6章介绍VB 2010的基础知识，包括VB简介，VB控件和事件，变量，数据输入和输出，分支结构设计，通用过程设计，循环结构设计，以及数组。
- 第7章介绍VB中常用控件的重要属性、方法和事件，以及如何运用这些控件开发具有实用价值的程序。
- 第8章介绍面向对象编程的基本概念及如何创建面向对象应用程序。
- 第9章介绍数据库程序设计，包括如何通过数据库操纵语言实现对多种类型数据库的访问。
- 第10章讲解如何实现文本文件的操作。

在教学时，教师还可以根据实际情况对课程内容做进一步的调整。例如，教材中对数据类型、运算符及控件的介绍分别放在了不同的章节，有的学生在学习时会觉得知识内容分散，这

样学习、理解起来有一定的困难，因此可以进行适当的调整。对于数据库部分，教材中的讲解还不够深入详细，而这部分恰恰又是学习的重点，因此教师应做相应的补充。国内有关VB程序设计的英文版书籍不多，适合作为教材的更少，本书经过改编后，做到了结构合理、题量适中，非常适合作为“Visual Basic程序设计”双语课程的教材，也是其他Visual Basic程序设计爱好者的很好的参考书。

罗凌
重庆师范大学计算机与信息科学学院

改编说明

英文原版目录

影印改编版去除了原书第1章和第12章，并将原书第8章~第11章的章节顺序进行调整，具体请参见后面的目录。原书第1章和第12章的PDF文件请在华信教育资源网（www.hxedu.com.cn）下载。

PREFACE

Since its introduction in 1991, Visual Basic has become one of the world's most widely used programming languages. The latest incarnation, Visual Basic 2010, further refines the language and extends features such as Language Integrated Query. Visual Basic programmers are enthusiastically embracing VB 2010's powerful capabilities. Likewise, students learning their first programming language will find VB 2010 the ideal tool for understanding the development of computer programs.

My objectives in writing this text are as follows:

1. *To develop focused chapters*. Rather than covering many topics superficially, I concentrate on important ones and cover them thoroughly.
2. *To use examples and exercises that students can relate to, appreciate, and feel comfortable with.* I frequently use real data. Examples do not have so many embellishments that they distract students from the programming techniques illustrated.
3. *To produce compactly written text that students will find both readable and informative*. The main points of each topic are discussed first, and then peripheral details are presented as comments.
4. *To teach good programming practices that are in step with modern programming methodology*. Problem-solving techniques and structured programming are discussed early and used throughout the book. The style follows object-oriented programming principles.
5. *To provide insights into the major applications of computers.*

Unique and Distinguishing Features

Microsoft® Visual Studio® 2010 Express editions DVD.[①] The textbook comes with a Visual Studio 2010 Express Edition DVD. It contains several Microsoft products including Visual Basic 2010, SQL Server 2010, and Visual Web Developer.

VideoNotes. VideoNotes are step-by-step video tutorials specifically designed to enhance the programming concepts presented in the eighth edition. Students can view the entire problem-solving process outside the classroom, when they need help the most. A VideoNote icon in the margin of the textbook alerts the reader that a topic is discussed in a video. Also, a Guide to VideoNotes in the front of the book summarizes the different videos found in the text. Nearly 50 VideoNotes for this edition are available at www.pearsonhighered.com/schneider . See the Student Resources section later in the Preface for information on how to access VideoNotes.

Exercises for Most Sections. Each section that teaches programming has an exercise set. The exercises reinforce the understanding of the key ideas of the section, and they challenge the student to explore applications. Most of the exercise sets require the student to trace programs, find errors, and write programs.

[①] 此改编版未包含原书附带的光盘，相关软件可从Microsoft网站下载，或联系Te_service@phei.com.cn获取。

Practice Problems. Practice Problems are carefully selected exercises located at the end of a section, just before the exercise set. Complete solutions are given following the exercise set. The practice problems often focus on points that are potentially confusing or are best appreciated after the student has worked on them. Readers should seriously attempt the practice problems and study their solutions before moving on to the exercises.

Programming Projects. Beginning with Chapter 3, every chapter contains programming projects. These projects reflect the variety of ways that computers are used in the business community, and they also present some games and general-interest topics. The large number and range of difficulty of the programming projects provide flexibility to adapt the course to the students' interests and abilities. Some programming projects in later chapters can be assigned as end-of-the-semester projects.

Comments. To avoid breaking the flow of the presentation, extensions and fine points of new topics are deferred to the "Comments" portion at the end of each section.

Case Studies. Each of the three case studies focuses on an important programming application. The problems are analyzed and the programs are developed with top-down charts and pseudocode. The programs can be downloaded from the companion website at http://www.pearsonhighered.com/schneiderinternational.

Chapter Summaries. At the end of each of Chapters 1 through 10, the key concepts are stated and the important terms summarized`.

"How To" Appendix. Appendix B provides a compact, step-by-step reference on how to carry out standard tasks in the Visual Basic environment.

Appendix on Debugging. The discussion of Visual Basic's sophisticated debugger is located in Appendix D, allowing the instructor flexibility in deciding when to cover this topic.

How to Access Instructor and Student Resource Materials

Pearson offers many different products around the world to facilitate learning. In countries outside the United States, some products and services related to this textbook may not be available due to copyright and/or permissions restrictions. If you have questions, you can contact your local office by visiting www.pearsonhighered.com/international.

Instructor Resources [①]

The following protected instructor resource materials are available on the publisher's website at www.pearsonhighered.com/schneider. For username and password information, please contact your local Pearson Representative.

- Computerized Test Generator
- PowerPoint Lecture Slides
- Instructor Solutions Manual: A complete solutions manual is available in pdf format. The manual contains the code for every programming exercise along with screen captures of the output.
- All the programs in the book
- Links to online premium content
  - VideoNotes
  - Student Solutions Manual

[①] 选用本书作为教材的老师可获得相关教辅，具体申请方式请参见书后的“教学支持说明”。

Student Resources [1]

Access to the Premium Website and VideoNotes tutorials is located at www.pearsonhighered.com/schneider. Students must use the access card located in the front of the book to register and access the online material. If no access card is provided, students can pur chase access by going to www.pearsonhighered.com/schneiderinternational and selecting "pur chase access to premium content." Instructors must register on the site to access the material.

The following content is available through the Premium Web site:

- VideoNotes: Pearson's new visual tool designed for teaching key programming concepts
- Student Solutions Manual: All the answers to the odd-numbered exercises (along with screen captures) will be available for download in a solutions manual in pdf format.
- All programs in the book and all text files and databases needed for the exercises.

Notice: This book contains many screen captures. When you run one of the programs downloaded from the website, what you see on your monitor might not look exactly like the screen capture shown in the book. To make them appear the same, you must check that your monitor is set to display 96 DPI (Dots Per Inch). To determine and/or change the DPI setting for your monitor, see the first item under "Configuring the Windows Environment" in Appendix B. Also, there may be slight differences due to the version of Win dows being used.

What's New in the Eighth Edition

New Sections

1. A section on using radio buttons, check boxes, and list boxes for selection has been added to Chapter 3. (Much of this material was previously in Chapter 7. Now it appears alongside If and Select Case blocks.)
2. A section on using loops with list boxes has been added to Chapter 5. (This section presents many operations on lists, such as searching, summing, and finding maximum values.)
3. A section on XML has been added to Chapter 10.(LINQ techniques developed earlier are applied to XML files.)
4. A chapter on Web applications has been added. (The three sections in this chapter cover the use of Visual Web Developer to create Web programs. Topics include tables, hyperlinks, postbacks, validation controls, and databases. Data extracted from databases are displayed in both grids and the new-to-VB2010 Chart controls.)

New Concepts

1. Chapter 2: Implicit line continuation. (The underscore line-continuation character is rarely needed in VB 2010.)
2. Chapter 2: Date data type. (This data type enables us to create some interesting programs, such as a program that tells users whether they are eligible to run for president in 2012.
3. Chapter 2: Sending output to the printer. (This optional material demonstrates how to produce a major type of output with Visual Basic.)

[1] 其中的一些学生资源请登录 www.hxedu.com.cn 下载。

4. Chapter 6: The ReadAllLines method for filling an array with the contents of a text file. (This powerful method allows us to place the contents of a text file in an array without having to use repeated ReDim Preserve statements.)
5. Chapter 6: Language Integrated Query. (This recent addition to Visual Basic and other .NET languages provides a standardized way to specify queries for a variety of data sources. In this textbook, LINQ is used to sort, search, and filter information from arrays, text files, XML files, and databases. This approach enables students to write concise, higher-level code focused more on problem solving than on data-structure manipulation. Instructors wishing to teach a more elementary or traditional course can omit or reduce the use of LINQ. The textbook presents most of the standard operations on arrays and sequential files with and without LINQ.)
6. Chapter 10: The Set operators Concat, Union, Intersect, and Except are used to manage data from text files. (These operators allow us to perform tasks that previously required complicated algorithms.)
7. Chapter 7: ToolTip control. (This control gives us a capability possessed by nearly every commercial Windows application.)
8. Chapter 8: Auto-Implemented properties. (This new-to-VB-2010 concept simplifies the creation of classes.)

Other Changes

1. The version of Visual Basic has been upgraded from Visual Basic 2008 to Visual Basic 2010, and relevant new features of Visual Basic 2010 have been added.
2. The real-life data in the examples and exercises have been updated and revised.
3. Some new large collections of data have been added. For instance, data on every Supreme Court justice (past and present) are contained in both a text file and a database.
4. Named constants are introduced earlier (Chapter 2).
5. The use of input validation has been increased (Chapter 3 on).
6. Function procedures are presented before Sub procedures (Chapter 4). With this change, students begin learning about general procedures with a familiar and essential construct. Also, the instructor has the option of omitting the concept of passing by reference.
7. The use of text files for input has been postponed until Chapter 6.
8. Tables are displayed in DataGridView controls rather than in list boxes (Chapter 6 on).
9. Many new business applications have been added.
10. The OpenFileDialog control is discussed earlier (Chapter 10, Text Files).
11. The discussion of multiple-form programs in Chapter 7 has been expanded to an entire section.

12. Databases are connected to programs with the Visual Basic wizard rather than with code (Chapter 9).

New Materials for Instructors

1. Guide to Application Topics. (This section provides an index of programs that deal with various topics including Business, Mathematics, and Sports.)
2. A complete solution manual in pdf format. (The manual will contain the code for every programming exercise along with a screen capture of the output.)

New Materials for Students

1. Screen captures have been added to the programs in the answer section of the book.
2. All the answers to the odd-numbered exercises (along with screen captures) will be available for download in a solutions manual in pdf format.
3. Nearly 50 VideoNotes are available at www.pearsonhighered.com/schneider. VideoNotes are Pearson's new visual tool designed for teaching key programming concepts and techniques. A VideoNote icon in the margin of the textbook alerts the reader when a topic is discussed in a video. See the Student Resources section earlier in this Preface for information on how to access VideoNotes.

Acknowledgments

Many talented instructors and programmers provided helpful comments and constructive suggestions during the various editions of this text, and I am most grateful for their contributions. The current edition benefited greatly from the valuable comments of the following reviewers:

G.W. Willis, Baylor University
Jaygarl Hojun, Iowa State University
Teresa Peterman, Grand Valley State University
Joel Weinstein, Northeastern University
Daniel E. Turk, Colorado State University
Paul Norrod, Lorain County Community College
Cynthia Brown, Portland State University
Karen Arlien, Bismarck State College
Ron Conway, Bowling Green State University
Phil Larschan, Tulsa Community College
Carol Roberts, University of Maine at Orono
Markita Price, University of Missouri - Columbia
Josh Pauli, Dakota State University

Many people are involved in the successful publication of a book. I wish to thank the dedicated team at Pearson whose support and diligence made this textbook possible, especially Melinda Haggerty, Assistant Editor of Computer Science, and Scott Disanno, Senior Managing Editor.

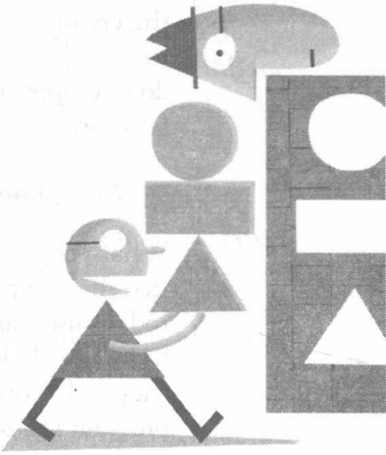
I also express my thanks to John Tarcza, a talented programmer, who helped with the development of the book and provided valuable insights and careful proofreading. I would like to thank Jeremy Schneider and Rob Teagarden for their helpful proofreading. Production editor Irwin Zucker did a fantastic job producing the book and keeping it on schedule. I am grateful to John Russo of the Wentworth Institute of Technology for producing the VideoNotes that accompany the book. The skill and graciousness of the team at Laserwords made for a pleasant production process. Copyeditor Bob Lentz contributed many suggestions for improving the book's quality.

I extend special thanks to my editor Tracy Dunkelberger. Her ideas and enthusiasm helped immensely with the preparation of the book.

David I. Schneider
dis@math.umd.edu

CONTENTS

1



Visual Basic, Controls, and Events

1.1 An Introduction to Visual Basic 2010

Visual Basic 2010 is the latest generation of Visual Basic, a language used by many software developers. Visual Basic was designed to make user-friendly programs easier to develop. Prior to the creation of Visual Basic, developing a friendly user interface usually required a programmer to use a language such as C or C++, often requiring hundreds of lines of code just to get a window to appear on the screen. Now the same program can be created in much less time with fewer instructions.

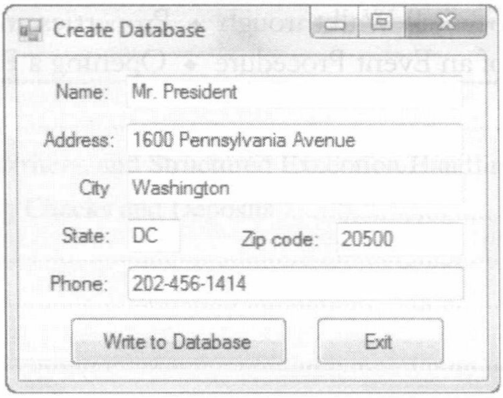
Why Windows and Why Visual Basic?

What people call **graphical user interfaces**, or GUIs (pronounced "gooies"), have revolutionized the computer industry. Instead of the confusing textual prompts that earlier users once saw, today's users are presented with such devices as icons, buttons, and drop-down lists that respond to mouse clicks. Accompanying the revolution in how programs look was a revolution in how they feel. Consider a program that requests information for a database. Figure 1.1 shows how a program written before the advent of GUIs got its information. The program requests the six pieces of data one at a time, with no opportunity to go back and alter previously entered information. Then the screen clears and the six inputs are again requested one at a time.

```
Enter name (Enter EOD to terminate): Mr. President
Enter Address: 1600 Pennsylvania Avenue
Enter City: Washington
Enter State: DC
Enter Zip code: 20500
Enter Phone Number: 202-456-1414
```

FIGURE 1.1 **Input screen of a pre-Visual Basic program to fill a database.**

Figure 1.2 shows how an equivalent Visual Basic program gets its information. The boxes may be filled in any order. When the user clicks on a box with the mouse, the cursor moves to that box. The user can either type in new information or edit the existing information. When satisfied that all the information is correct, the user clicks on the *Write to Database* button. The boxes will clear, and the data for another person can be entered. After all names have been entered, the user clicks on the Exit button. In Fig. 1.1, the program is in control; in Fig. 1.2, the user is in control!



Create Database

Name: Mr. President

Address: 1600 Pennsylvania Avenue

City Washington

State: DC Zip code: 20500

Phone: 202-456-1414

FIGURE 1.2 **Input screen of a Visual Basic program to fill a database.**

How You Develop a Visual Basic Program

A key element of planning a Visual Basic program is deciding what the user sees—in other words, designing the user interface. What data will he or she be entering? How large a window should the program use? Where will you place the buttons the user clicks on to activate actions in the program? Will the program have places to enter text (text boxes) and places to display output? What kind of warning boxes (message boxes) should the program use? In Visual Basic, the responsive objects a program designer places on windows are called *controls*. Two features make Visual Basic different from traditional programming tools:

1. You literally draw the user interface, much like using a paint program.
2. Perhaps more important, when you're done drawing the interface, the buttons, text boxes, and other objects that you have placed in a blank window will automatically recognize user actions such as mouse movements and button clicks. That is, the sequence of procedures executed in your program is controlled by "events" that the user initiates rather than by a predetermined sequence of procedures in your program.

In any case, only after you design the interface does anything like traditional programming occur. Objects in Visual Basic recognize events like mouse clicks; how the objects respond to them depends on the instructions you write. You always need to write instructions in order to make controls respond to events. This makes Visual Basic programming fundamentally different from traditional programming. Programs in traditional programming languages ran from the top down. For these programming languages, execution started from the first line and moved with the flow of the program to different parts as needed. A Visual Basic program works differently. Its core is a set of independent groups of instructions that are activated by the events they have been told to recognize. This event-driven methodology is a fundamental shift. The user decides the order in which things happen, not the programmer.

Most of the programming instructions in Visual Basic that tell your program how to respond to events like mouse clicks occur in what Visual Basic calls *event procedures*. Essentially, anything executable in a Visual Basic program either is in an event procedure or is used by an event procedure to help the procedure carry out its job. In fact, to stress that Visual Basic is fundamentally different from traditional programming languages, Microsoft uses the term *project* or *application*, rather than *program*, to refer to the combination of programming instructions and user interface that makes a Visual Basic program possible. Here is a summary of the steps you take to design a Visual Basic program:

1. Design the appearance of the window that the user sees.
2. Determine the events that the controls on the window should respond to.
3. Write the event procedures for those events.

Now here is what happens when the program is running:

1. Visual Basic monitors the controls in the window to detect any event that a control can recognize (mouse movements, clicks, keystrokes, and so on).
2. When Visual Basic detects an event, it examines the program to see if you've written an event procedure for that event.
3. If you have written an event procedure, Visual Basic executes the instructions that make up that event procedure and goes back to Step 1.
4. If you have not written an event procedure, Visual Basic ignores the event and goes back to Step 1.

These steps cycle continuously until the program ends. Usually, an event must happen before Visual Basic will do anything. Event-driven programs are reactive more than active—and that makes them more user friendly.

The Different Versions of Visual Basic

Visual Basic 1.0 first appeared in 1991. It was followed by version 2.0 in 1992, version 3.0 in 1993, version 4.0 in 1995, version 5.0 in 1997, and version 6.0 in 1998. VB.NET, initially released in February 2002, was not backward compatible with the earlier versions of Visual Basic. It incorporated many features requested by software developers, such as true inheritance. Visual Basic 2005, released in November 2005, Visual Basic 2008, released in November 2007, and Visual Basic 2010, released in April 2010 are significantly improved versions of VB.NET.

1.2 Visual Basic Controls

Visual Basic programs display a Windows-style screen (called a **form**) with boxes into which users type (and in which users edit) information and buttons that they click to initiate actions. The boxes and buttons are referred to as **controls**. In this section, we examine forms and four of the most useful Visual Basic controls.

Starting a New Visual Basic Program

Each program is saved (as several files and subfolders) in its own folder. Before writing your first program, you should use Windows Explorer to create a folder to hold your programs.

The process for invoking Visual Basic varies slightly with the edition of Visual Basic installed on the computer. To invoke Visual Basic from a computer that has Visual Basic Express installed, click the Windows Start button, hover over All Programs, and then click on Microsoft Visual Basic 2010 Express. With the other editions of Visual Basic, hover over All Programs, hover over Microsoft Visual Studio 2010, and then click on Microsoft Visual Studio 2010 in the short list that is revealed.

Figure 1.3 shows the top half of the screen after Visual Basic is invoked. A Menu bar and a Toolbar are at the very top of the screen. These two bars, with minor variations, are always present while you are working with Visual Basic. The remainder of the screen is called the **Start Page**. Some tasks can be initiated from the Menu bar, the Toolbar, and the Start Page. We will usually initiate them from the Menu bar or the Toolbar.

The first item on the Menu bar is *File*. Click on *File*, and then click on *New Project* to produce a New Project dialog box. Figure 1.4 shows the New Project dialog box produced by Visual Basic Express.

The Windows Forms Application item should be selected in the center list. If this is not the case, click on *Windows Forms Application* to select it. **Note:** The number of items in the list will vary depending on the edition of Visual Basic you are using.

The name of the program, initially set to WindowsApplication1, can be specified at this time. Since we will have a chance to change it later, let's just use the name WindowsApplication1 for now. Click on the *OK* button to invoke the Visual Basic programming environment. See Fig. 1.5. The Visual Basic programming environment is referred to as the **Integrated Development Environment** or **IDE**.