21世纪高等院校计算机系列教材

影印版

C#程序设计

C# For Students

[英] Douglas Bell　Mike Parr　著

中国水利水电出版社
www.waterpub.com.cn

DOUGLAS BELL
MIKE PARR

C#

FOR

Students

中国水利水电出版社
www.waterpub.com.cn

图书在版编目（CIP）数据

C#程序设计/（英）贝尔（Bell, D.），（英）帕尔（Parr, M.）著. —影印本. —北京：中国水利水电出版社，2006
（21世纪高等院校计算机系列教材）
书名原文：C# FOR Students
ISBN 7-5084-4107-9

Ⅰ.C… Ⅱ.①贝…②帕… Ⅲ.C语言—程序设计—高等学校—教材—英文 Ⅳ.TP312
中国版本图书馆CIP数据核字（2006）第116324号

书　　名	C#程序设计
作　　者	[英] Douglas Bell Mike Parr 著
出版发行	中国水利水电出版社（北京市三里河路6号 100044）
	网址：www.waterpub.com.cn
	E-mail：mchannel@263.net（万水）
	sales@waterpub.com.cn
	电话：（010）63202266（总机）、68331835（营销中心）、82562819（万水）
经　　售	全国各地新华书店和相关出版物销售网点
排　　版	北京万水电子信息有限公司
印　　刷	北京市天竺颖华印刷厂
规　　格	787mm × 1092mm　16开本　24.25印张　537千字
版　　次	2006年10月第1版　2006年10月第1次印刷
印　　数	0001—3000册
定　　价	36.00元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

Preface

This book is for novices

If you have never done any programming before – if you are a complete novice – this book is for you. This book assumes no prior knowledge of programming. It starts from scratch. It is written in a simple, direct style for maximum clarity. It is aimed at first level students at universities and colleges, but it is also suitable for novices studying alone.

Why C#?

C# is arguably one of the best programming languages to learn and use in the 21st century because:

- C# continues the tradition of the family of languages that includes C, C++ and Java.
- Object-oriented languages are the latest and most successful approach to programming. C# is completely object-oriented from the ground up.
- C# is a completely general-purpose language – anything that Visual Basic, C++ and Java can do, so can C#.
- C# gains most of its functionality from a library of components provided by the .NET framework.

You will need . . .

To learn to program in C# you need a PC running Windows 2000, NT, XP or above and the software that allows you to prepare and run C# programs in a convenient way. There are two versions of the software provided by Microsoft – Visual C# .NET (for C# alone) and Visual Studio .NET (which supports both C# and other languages). This book comes with CDs containing a full version of Visual Studio .NET.

The approach of this book

We explain how to use objects early in this book. Our approach is to start with the ideas of variables, assignment and methods, then use objects created from library classes. Next we explain how to use control structures for selection and looping. Then comes the treatment of how to

write your own classes.

We wanted to make sure that the fun element of programming was paramount, so we use graphics right from the start. We think graphics is fun, interesting and clearly demonstrates all the important principles of programming. But we haven't ignored programs that input and output text – they are also included.

The programs we present use many of the features of graphical user interfaces (GUIs), such as buttons and text boxes. But we also explain how to write console programs in C#.

We introduce new ideas carefully, one at a time rather than all at once. So, for example, there is a single chapter on writing methods. We introduce simple ideas early and more sophisticated ideas later on.

What's included?

This book explains the fundamental ideas of programming:

- variables;
- assignment;
- input and output using a graphical user interface (GUI);
- calculation;
- repetition;
- selection between alternatives.

It explains how to use numbers and character strings. Arrays are also described. These are all topics that are fundamental, whatever kind of programming you do. This book also thoroughly explains the object-oriented aspects of programming – using objects, writing classes, methods and properties, and using library classes. We also look at some of the more sophisticated aspects of object-oriented programming including inherit-ance, polymorphism and interfaces.

What's not included?

This book confines itself to the essentials of C#. It does not explain all the bits and pieces, the bells and whistles. Thus the reader is freed from unnecessary detail and can concentrate on mastering C# and programming in general.

UML

The Unified Modeling Language (UML) is the current mainstream notation for describing programs. We use elements of UML selectively, where appropriate, throughout this book.

Applications

Computers are used in many diaerent applications and this book uses examples from all areas including:

- games;
- information processing;

- scientific calculations.

We have also included a few exercises which look at the exciting idea of artificial life.

The reader can choose to concentrate on those application areas of interest and ignore other areas.

Exercises are good for you

If you were to read this book time and again until you could recite it backwards, you still wouldn't be able to write programs. The practical work of writing programs is vital to becoming fluent and confident at programming.

There are exercises for the reader at the end of each chapter. Please do some of them to enhance your ability to program.

There are also short self-test questions throughout the text, so that you can check you have understood things properly. The answers are given at the end of each chapter.

Have fun

Programming is creative and interesting, particularly in C#. Please have fun!

Visit our website

The website includes:

- the text of all the programs in this book;
- a discussion forum for students;
- a bonus chapter covering the use of C# with databases;
- additional resources for instructors.

Our website can be reached via the Pearson Education website at:
http://www.mikeparr.info/csharplst/csabout.html
http://www.pearsoned.co.uk/HigherEducation/Booksby/BellParr/.

Contents

Detailed contents

The background to C#

This chapter explains:

- how and why C# came into being;
- Microsoft's .NET framework;
- the introductory concepts of programming.

The history of C#

A computer program is a series of instructions that are obeyed by a computer. The point of the instructions is to carry out a task – e.g. play a game, send an e-mail, etc. The instructions are written in a particular style: they must conform to the rules of the programming language we choose. There are hundreds of programming languages, but only a few have made an impact and become widely used. The history of programming languages is a form of evolution, and here we will look at the roots of C# ('C Sharp'). The names of the older languages are not important, but we provide them for completeness.

Around 1960, a programming language named Algol 60 was created. ('Algol' from the term 'algorithm' – a series of steps that can be performed to solve a problem.) This was popular in academic circles, but its ideas persisted longer than its use. At this time, other languages were more popular: COBOL for data processing, and Fortran for scientific work. In the UK, an extended version of Algol 60 was created (CPL – combined programming language), which was soon simplified into basic CPL, or BCPL.

We then move to Bell Laboratories USA, where Dennis Ritchie and others transformed BCPL into a language named B, which was then enhanced to become C, around 1970. C was tremendously popular. It was used to write the UNIX operating system, and much later, Linus Torvalds used it to write a version of UNIX – named LINUX – for PCs.

The next step came when C++ ('C plus-plus') was created around 1980 by Stroustrup, also at Bell Labs. This made possible the creation and reuse of separate sections of code, in a style known as 'object-oriented programming'. (In C, you could use ++ to add one to an item – hence C++ is one up from C.)

C++ is still popular, but hard to use; it takes a lot of study. Around 1995, Sun Microsystems