普通高等教育“十三五”规划教材

C#程序设计教程

周洪建　主编

科学出版社

普通高等教育“十三五”规划教材

C#程序设计教程

周洪建　主　编
蔡桂艳　张俊妍　毕志升　副主编

科学出版社
北　京

内 容 简 介

本书以 Visual Studio 2013 为平台，以程序设计为主线，通过对典型示例的分析与实验，将 C#程序设计语言的基本概念、可视化编程技术和面向对象程序设计方法融入示例中，在讲解基本理论和算法的同时，为应用系统的开发与设计及数据分析、图像处理在各领域的应用提供思路。全书共 11 章，内容包括 C#概述、C#程序设计入门、C#语法基础、结构化程序设计、数组、面向对象程序设计基础、Windows 应用程序开发基础、文件、数据库应用开发、C#多线程技术、图形图像编程基础。

本书适合作为本科院校相关专业的教材，也可作为高职院校、培训学校相关课程的教材，还可作为编程爱好者的参考用书。

图书在版编目（CIP）数据

C#程序设计教程/周洪建主编．—北京：科学出版社，2017.12

普通高等教育“十三五”规划教材

ISBN 978-7-03-055743-8

Ⅰ. ①C… Ⅱ. ①周… Ⅲ. ①C 语言-程序设计-高等学校-教材

Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2017）第 294094 号

责任编辑：吕燕新 王 惠 / 责任校对：陶丽荣

责任印制：吕春珉 / 封面设计：东方人华平面设计部

科学出版社出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

三河市骏杰印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2017 年 12 月第 一 版 开本：787×1092 1/16

2017 年 12 月第一次印刷 印张：20 1/2

字数：472 000

定价：49.00 元

（如有印装质量问题，我社负责调换<骏杰>）

销售部电话 010-62136230 编辑部电话 010-62135397-2052

前 言

C#语言是在继承C++和Java等语言优点的基础上演变而来的一种基于.NET的完全面向对象的编程语言。它避免了C语言中复杂的指针和多继承，简单易学且功能强大，能较好地满足软件工程的需要，目前已经成为开发基于.NET 的企业级应用程序的首选语言。

本书根据编者多年的教学实践经验编写而成，在内容上结合初学者的特点，从程序设计的基本思想起步，着重建立学生在计算机方面的知识体系，使学生在掌握计算机程序设计基本知识的同时，提高逻辑思维能力和计算机应用能力，了解可视化编程及面向对象程序设计原理，并运用这些原理和方法，提高处理信息、数据、图像的能力。

本书共11章，主要内容如下。

第1章介绍.NET框架、C#语言特点、C#集成开发环境等。

第2章介绍C#代码的编写基础、C#项目组织结构、C#程序的编译和运行、输入/输出操作等。

第3章介绍C#语法基础，包括C#程序结构、数据类型、变量与常量、运算符与表达式、类型转换、装箱和拆箱等。

第4章介绍流程控制语句，包括结构化程序设计的概念、顺序结构、选择结构、循环结构等。

第5章介绍数组，包括一维数组、二维数组、Array类及应用等。

第6章介绍C#面向对象程序设计基础知识，包括类和对象基本概念、C#的常用类、继承与接口、C#命名空间等。

第7章介绍Windows应用程序设计入门知识和控件的基本用法。

第8章介绍目录和文件管理，以及文件的读写等相关操作。

第9章介绍数据库应用开发，包括ADO.NET的基础知识，以及数据检索、处理、更新和显示等。

第10章介绍C#多线程技术，包括线程的创建、中断、暂停、同步等相关知识。

第11章介绍图形图像编程基础，包括GDI+绘图基础、C#图像处理基础，以及简单的图像处理技术。

本书由周洪建担任主编，由蔡桂艳、张俊妍、毕志升担任副主编。具体编写分工如

下：第1～4章由蔡桂艳编写，第5～7章由张俊妍编写，第8～10章由毕志升编写，第11章由周洪建编写，周洪建对全书进行了规划、组稿、修改和定稿。

为配合教学需要，本书提供了配套的PPT教学课件、书中所有源程序代码及全部习题参考答案，读者可从http://www.abook.cn下载。

由于编者水平有限，书中难免存在不足和疏漏之处，恳请广大读者提出宝贵意见。

编　者

2017年11月

目 录

第1章 C#概述

1.1 程序设计概述

什么是程序设计？有一个著名的公式：程序设计=数据结构+算法。其中，程序设计即编程；数据结构是非数值计算的程序设计问题中计算机的操作对象，也是它们之间的关系和操作；算法是对特定问题求解步骤的一种描述，是对指令的有序排列。简单地说，程序设计就像盖房子，数据结构是砖、瓦，算法就是设计图纸。若想盖房子，首先必须有原料（数据结构），然后按照设计图纸（算法）的说明，一砖一瓦地砌出来。程序设计也一样，编译工具（如 Visual Studio、Eclipse、Visual Basic、Free Pascal 等）中有各种功能语句（如 C#、Java、BASIC、Pascal 等）或基本结构（如 Read、Write、Real、Boolean 等），它们不会自动排列成需要的程序代码。我们需要按照程序规定的语法编写程序，而程序的功能是实现目标，是算法的具体体现。所以，按照特定的规则，把特定的功能语句和基本结构按照特定的顺序排列起来，形成一个有特定功能的程序，这就是程序设计=数据结构+算法。在程序设计中，数据结构就像物质，算法就是意识。这就如同哲学上所说，意识是依赖于物质而存在的，物质是由于意识而发展的，双方相互依存、缺一不可。

数据结构内容不多，仅有一些基本数据类型（如整型、实型、布尔型、字符型等）和用户自定义的数据结构（如数组、集合、文件、指针等。可是算法却不同，它的种类多样，可使数据以任何符合语法和功能要求的方式排列。这就是算法的灵活性、不固定性。

程序设计有以下两种描述方式。

1）程序设计是使用一些指令来实现某种功能的过程。计算机只能识别一种语言——机器语言，它是由 0 和 1 组成的指令。由其他语言编写的程序最终要转换成二进制的机器语言才能被计算机执行，这个过程是通过编译器或解释器实现的。

2）程序设计是给出解决特定问题程序的过程，是软件构造活动中的重要组成部分。程序设计往往以某种程序设计语言（本书为 C#）为工具，给出该语言的程序。程序设计过程应包括分析、设计、编码、测试、排错等阶段。

1.2 语 言 概 述

计算机只能识别机器语言，但是为什么能够使用各种语言编写程序呢？这是因为现

在使用的计算机采用冯·诺依曼存储程序式体系，包括控制器、运算器、存储器、输入设备和输出设备。

冯·诺依曼式计算机具体的工作流程：在控制器的指挥下，①运算器从存储器取出指令；②运算器分析指令，得到计算命令和待操作数，传送到存储器；③运算器从存储器取出待操作数；④运算器计算出结果；⑤运算器将结果输出到存储器，输出设备。

存储器分为寄存器（位于 CPU 内部，用于存放指令、待操作数和结果）、高速缓存（通常位于 CPU 内部，用作数据缓冲区）、内存储器、外存储器。而指令是预先定义的由 CPU 执行的命令，即 CPU 的指令集。具体的指令以二进制码表示，包含一个或多个字节，由指令码（具体命令）和操作数（要操作的数或地址）构成。在具体的执行中，首先要把宏观层次的命令转换为满足指令集要求的二进制代码，然后才能在计算机上运行。

程序的执行就是在以上基础进行的，开始使用机器语言时，具体的命令形式如10010101010100110001111O。虽然这种底层命令能够直接与计算机进行交互，但是一般人难以接受。于是就出现了用助记符代替机器指令的汇编语言，如 add 2,3，相对来说比较友好；接着出现了高级语言，其更加抽象，但接近一般人的思维习惯，如 d=a*b+c。高级语言将很多细节封装起来，人们直接使用即可，不用掌握其具体的转换方法，即具体的编译方法。否则，一条程序语句有可能转换为多条指令，其执行的次序和次数、各种内存地址和数据的调用就足够使人望而却步。程序设计语言的各个发展阶段如图 1-1 所示。

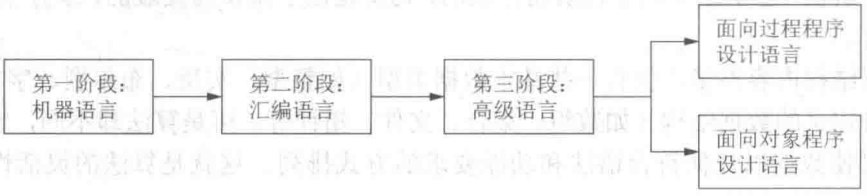

图 1-1　程序设计语言的各个发展阶段

1.3　.NET 与 C#

1.3.1　.NET 框架介绍

.NET 的全称是 Microsoft .NET，是一个新的开发平台，.NET Framework（框架）是其核心部分。.NET Framework 是集成在 Windows 系统中的组件，它支持生成和运行下一代应用程序与 XML Web Services，开发思路是敏捷软件开发（agile software development，ASD）、快速应用开发（rapid application development，RAD）、平台无关性和网络透明化。

.NET Framework 包括应用程序开发技术、.NET Framework 类库、基类库和公共语言运行时（common language runtime，CLR）4 部分。

1. 应用程序开发技术

应用程序开发技术包括 WinForms 技术、Web 应用程序技术等高级编程技术，它位于.NET 架构的顶端。这些应用程序均可使用 Visual C# .NET、Visual C++ .NET、Visual Basic .NET 等来编写。

2. .NET Framework 类库

.NET Framework 类库提供了用于应用程序开发的通用接口，它是支持开发人员进行系统开发的功能性接口的集合。开发人员可以使用这些功能接口更方便地开发出各种形式的应用程序。.NET Framework 的主要类库包括数据库访问类库（ADO.NET 等）、文件管理类库、网络编程类库、正则表达式和消息支持等。

3. 基类库

基类库为开发人员提供了访问底层的一系列接口，通过这些接口，开发人员不需要了解具体的底层实现就可以完成输入/输出（I/O）操作、多线程支持等底层功能。

4. 公共语言运行时

公共语言运行时是.NET Framework 的核心内容。所有用.NET 开发的应用程序都是在公共语言运行时上运行的，它为应用程序提供了许多核心服务，如内存管理、线程管理、文件管理及远程处理等。

1.3.2 C#语言特点

C#是微软公司专门为.NET 平台开发的一种语言。使用 C#开发的应用程序是基于.NET 应用程序的，具有良好的安全性和跨平台性。利用 Visual Studio .NET 的“所见即所得”功能，可以使整个开发过程更为简洁、明快。

自微软公司开发 C#语言以来，C#语言就受到了广泛的关注，从系统开发到网站设计，各处都能应用到 C#的知识。

C#是由 C、C++语言发展而来的，它在继承 C、C++语言强大功能的同时抛弃了它们的一些复杂特性，从而变得相对简单。C#中没有宏，没有模板，不允许多重继承，不再强调使用指针，抛弃了大多数令读者头疼的特性。C#在语法、思维方面与 Java 有着很大的相似性。总体来讲，C#具有以下优点。

1. 简洁的语法

默认情况下，C#的代码在.NET 框架提供的“可操纵”环境下运行，不允许直接进行内存操作。它所具有的最大特色是没有指针。与此相关的是，那些在 C++中使用的操

作符（如“::”“→”）不再出现。

2. 精心地面向对象设计

C#语言是完全按照面向对象的思想来设计的，因此它具有面向对象所应有的一切特性：封装、继承和多态性。C#语言只允许单继承，即一个类不会有多个基类，从而避免了类型定义的混乱。在C#语言中，每种类型都是一个对象，因此不存在全局函数、全局变量和全局常数等概念。所有常量、变量、属性、方法、索引和事件等都必须封装在类中，从而使代码具有更好的可读性，也减少了命名冲突的可能。

在C#的类型系统中，每种类型都可以看作一个对象。C#提供了一种装箱（boxing）与拆箱（unboxing）的机制来完成这种操作，而不给使用者带来麻烦。

3. 与Web紧密结合

.NET中新的应用程序开发模型意味着越来越多的解决方案需要与Web标准相统一，如超文本标记语言（hyper text markup language，HTML）和可扩展标记语言（extensible markup language，XML）。对于开发人员来说，有了Web服务框架的帮助，网络服务就如同C#的本地对象。开发人员能够利用已有的面向对象知识与技巧开发Web服务。仅需要使用简单的C#语言结构，即可驱动C#组件为Web服务，并允许它们通过Internet被运行在任何操作系统上的任何语言所调用。

4. 完整的安全性与错误处理

安全性和错误处理能力是衡量一种语言是否优秀的重要依据，C#语言可以避免很多软件开发中的常见错误，并提供了包括类型安全在内的完整的安全性能。

默认情况下，从Internet和Intranet（企业内部网）下载的代码都不允许访问任何本地文件和资源；C#语言不允许使用未初始化的变量，并提供了边界检查与溢出检查等功能。其在内存管理中提供的垃圾回收机制也大大减轻了开发人员的内存管理负担。

5. 版本处理技术

C#语言内置了版本控制功能，如对函数重载和接口的处理方式及特性支持等，从而保证方便地开发和升级复杂的软件。

6. 灵活性与兼容性

灵活性是指在托管状态下，C#语言不使用指针，而是用委托（delegate）来模拟指针的功能。兼容性是指C#语言允许与具有C或C++语言风格的需要传递指针型参数的应用程序编程接口（application programming interface，API）进行交互操作，允许C#语言组件与其他语言组件间进行交互操作等。

在微软公司提供的.NET框架中，可以用C#开发C/S应用和Web应用，并且可以在

一个项目中混合使用C#和VB（Visual Basic）语言。从某种意义上讲，.NET框架和Java虚拟机有很多相似之处。C#的语法规则和C++非常相似，有C语言基础的开发者比较容易入手。

1.4 C#集成开发环境

2002年，微软公司推出第一款基于.NET架构的开发工具Visual Studio .NET。该架构将强大功能与新技术结合起来，用于构建具有良好视觉体验的应用程序，实现跨技术边界的无缝通信，并且支持各种业务流程。另外，后续版本的Visual Studio都继承了这种架构。版本简表如表1-1所示（更新至2017年）。

表1-1 版本简表

名称	内部版本	支持.NET Framework版本	发布日期	备注
Visual Studio .NET 2002	7.0	1.0	2002-02-13	—
Visual Studio .NET 2003	7.1	1.1	2003-04-24	—
Visual Studio 2005	8.0	2.0	2005-11-07	微软公司将“.NET”字样从产品名称中移除
Visual Studio 2008	9.0	2.0、3.0、3.5	2007-11-19	移除 Visual J#
Visual Studio 2010	10.0	2.0、3.0、3.5、4.0	2010-04-12	加入 Visual F#
Visual Studio 2012	11.0	2.0、3.0、3.5、4.0、4.5	2012-09-12	支持开发 Windows UI 应用程序
Visual Studio 2013	12.0	2.0、3.0、3.5、4.0、4.5、4.5.1、4.5.2、4.6、4.6.1、4.6.2	2013-10-17	对 Windows 8.1 提供支持
Visual Studio 2015	14.0	2.0、3.0、3.5、4.0、4.5、4.5.1、4.5.2、4.6、4.6.1、4.6.2	2014-11-10	—
Visual Studio 2017	15.0	2.0、3.0、3.5、4.0、4.5、4.5.1、4.5.2、4.6、4.6.1、4.6.2	2017-03-08	—

微软公司打破了Visual Studio两年升级一次的传统，Visual Studio 2012发布后相隔一年，微软公司就发布了Visual Studio 2013。与Visual Studio 2012相比，Visual Studio 2013新增了代码信息指示（code information indicators）、团队工作室（team room）、身份识别、.NET内存转储分析仪、敏捷开发项目模板、Git支持及更强大的单元测试支持。

本书操作均以Visual Studio 2013为平台，下面对Visual Studio 2013的几个主要功能进行简单介绍。

1. 支持 Windows 8.1 APP 开发

Visual Studio（以下简称“VS”）2013 提供的工具集非常适合生成在 Windows 8.1 平台上运行的新式应用程序，同时在所有 Microsoft 平台上支持设备和服务；支持在 Windows 8.1 预览版中开发 Windows 应用商店应用程序，具体表现在对工具、控件和模板进行了许多更新，对于可扩展应用程序标记语言（extensible application markup language，XAML），应用程序支持新近提出的编码 UI 测试、用于 XAML 和 HTML 应用程序的 UI 响应能力分析器与能耗探查器，增强了用于 HTML 应用程序的内存探查工具，以及改进了与 Windows 应用商店的集成。

2. 敏捷项目管理

Visual Studio 2013 提供敏捷项目组合管理，提高团队协作。Team Foundation Server（以下简称“TFS”）2012 已经引入了敏捷项目管理功能，在 TFS 2013 中该功能得到进一步完善（如需求列表 backlog 的应用）。TFS 擅长处理流程分解，为不同层级的人员提供不同级别的需求列表，同时支持多个 Scrum 团队分开管理各自的需求列表，最后汇总到更高级的需求列表。

3. 应用生命周期管理

在得到有效应用的情况下，应用生命周期管理（application lifecycle management，ALM）实践方法可以消除团队之间的壁垒，使企业能够克服挑战，更快速地提供高质量的软件，还可以减少浪费、缩短时间周期和提高业务灵活性。

4. 版本控制

VS 一直在改进自身的版本控制功能，包括 Team Explorer 新增的 Connect 功能，可以帮助用户同时关注多个团队项目。新的 Team Explorer 主页也更简洁、明确，各任务之间切换也更加方便。同时，由于众多用户反馈，VS 2013 已恢复更改挂起（pending changes）功能。如果对 VS、TFS 有建议或者意见，用户可以向 VS 开发团队反馈。

5. 轻量代码注释

轻量代码注释（lightweight code commenting）与 VS 高级版中的代码审查功能类似，可以通过网络进行简单的注释。

6. 编程过程

在编程过程中，VS 2013 增强了代码提示功能，能在编码的同时帮助开发人员检查错误，并通过多种指示器进行提示。此外，VS 2013 中还增加了内存诊断功能，能对潜在的内存泄漏问题进行提示。