普通高等学校规划教材

C#面向对象程序设计

主　编　朱兴亮　庄　致
副主编　魏庆琦　王定军　胡　勇　冯运义　梁永宏

C# MIANXIANG DUIXIANG CHENGXU SHEJI

人民交通出版社股份有限公司
China Communications Press Co.,Ltd.

普通高等学校规划教材

C#面向对象程序设计

主　编　朱兴亮　庄　致
副主编　魏庆琦　王定军
　　胡　勇　冯运义
　　梁永宏

人民交通出版社股份有限公司
China Communications Press Co.,Ltd.

内 容 提 要

C#语言是微软公司开发的一种新的面向对象编程语言，它吸收了 C、C + + 和 Java 语言的优点，语法简洁、功能强大，开发效率极高。本书介绍了 C#语言的基础知识，深入剖析了面向对象的编程思想，阐述了 C#语言的常用开发技术。不同于一般介绍 C#语言的书籍，本书深入浅出地阐述了面向对象程序设计的基本概念和理论体系，精心设计了大量案例帮助读者理解面向对象的编程思想。同时，作者根据多年的教学经验和项目开发经验，针对每章内容精编了大量的编程练习题，读者可以通过这些练习题迅速提高编程能力。

全书叙述简洁、通俗易懂、实用性强，书中所有源程序均在 Visual Studio 2012 平台下调试通过。本书可作为高等院校计算机及相关专业的教材，也可作为初、中级程序员的参考用书。

图书在版编目(CIP)数据

C#面向对象程序设计 / 朱兴亮，庄致主编. —北京：人民交通出版社股份有限公司，2015.8

ISBN 978-7-114-12425-9

Ⅰ.①C… Ⅱ.①朱…②庄… Ⅲ.①C 语言—程序设计 Ⅳ.①TP312

中国版本图书馆 CIP 数据核字(2015)第 179865 号

普通高等学校规化教材

书　　名：C#面向对象程序设计

著 作 者：朱兴亮　庄　致

责任编辑：刘永芬

出版发行：人民交通出版社股份有限公司

地　　址：(100011)北京市朝阳区安定门外外馆斜街 3 号

网　　址：http://www.ccpress.com.cn

销售电话：(010)59757973

总 经 销：人民交通出版社股份有限公司发行部

经　　销：各地新华书店

印　　刷：北京鑫正大印刷有限公司

开　　本：787 × 1092　1/16

印　　张：16.5

字　　数：380 千

版　　次：2015 年 8 月　第 1 版

印　　次：2015 年 8 月　第 1 次印刷

书　　号：ISBN 978-7-114-12425-9

定　　价：32.00 元

(如有印刷、装订质量问题的图书由本公司负责调换)

目　录

第1章　C#语言概述

1.1　计算机和程序

计算机是一种能执行计算和做出逻辑判断的设备，它的计算速度要比人快上几百万倍甚至几十亿倍。例如，今天很多的个人计算机都能够在一秒内执行几十亿次的加法运算。尽管物理外观上不同，实际上每台计算机都可以认为由5个逻辑单元组成：

(1)输入单元：它是计算机从各种输入设备获得信息(数据和计算机程序)的“接收”部分，然后再把信息交由其他单元进行处理。现在，大部分用户通过键盘和鼠标设备输入信息。其他的输入设备包括麦克风(用来向计算机输入语音)、扫描仪(用来扫描图像)和数码相机(用来照相和制作视频)。

(2)输出单元：它是计算机的“运送部分”，即将计算机处理过的信息交给各种输出设备，以便在计算机之外利用信息。计算机可以通过各种方式输出信息，包括在屏幕上显示输出，在音频、视频设备上播放输出，在打印机上打印文字和图像等。

(3)存储单元：它是计算机中可以高速访问、容量相对较小的“仓库”，用于数据临时存储。存储单元存有输入单元输入的信息，使信息能够被迅速有效处理。除此之外，存储单元还存有处理过的信息，直到将这些信息传送到输出设备。

(4)中央处理器(CPU)：它是由控制器和运算器(ALU)构成。运算器负责加、减、乘、除等计算的执行；控制器是计算机中的“管理者”部分，是计算机中的协调员，负责监督计算机其他部分的执行。当用户通过输入设备输入数据时，控制器会将数据传送到存储单元；当系统需要输出信息时，控制器也会将存储单元的数据传送到输出设备；当需要计算时，控制器会将存储单元的数据传送到运算器。

(5)二级存储单元：它是计算机中可长期保存数据且容量大的“仓库”。二级存储设备(例如硬盘和磁盘)通常保存其他单元不常用到的程序或者数据。在需要这些信息时，计算机能从二级存储单元获得这些信息。访问二级存储单元需要的时间比访问主存需要的时间长，但二级存储单元的每单位价格远远低于主存每单位的价格。二级存储单元是非易失性的，这意味着即使计算机关机，二级存储单元仍然存储着信息。

由上述描述可以看出，计算机中最核心的部件是控制器，而控制器依赖于存储单元中程序完成各项“管理”工作。所谓程序，是用来指挥计算机运行的各项指令的序列，是为了解决特定数据处理任务由程序设计者编写的“步骤”序列。运行在计算机上的程序也称为软件。近年来硬件的成本呈现出下降趋势，但是由于软件开发技术并没有明显的改善，因此软件开发的费用却是稳步上升的。目前普遍认为，本书将介绍的面向对象编程技术是一个重大的突破，能极大地提高程序员的开发效率。

1.2 机器语言、汇编语言和高级语言

程序员可以用各种编程语言(包含一套完整的语法和语义规则)编写指令,有的指令计算机可以直接理解,而另一些则需要中间的"翻译"步骤。尽管现在有数百种计算机语言在使用,但这些不同的语言可以分成机器语言、汇编语言、高级语言3大类。

计算机都只能直接理解自己的"机器语言"。作为每种特定的计算机的"天生的语言",机器语言在计算机硬件设计时定义。机器语言通常由数字串(最终简化为0和1)组成,这些数字指示计算机去执行最基本的操作。机器语言是依赖于机器的,这意味着一种特定的机器语言只能用在这种类型的计算机上。下面这是一段机器语言代码,功能是将"加班工资"和"基本工资"相加,并将结果存入"工资总额"。在每行数字串中,加黑显示的数字串表示操作符,其后是操作数在存储单元的地址。由这个简单的例子可以看出机器语言对人来说是难以理解的。

1300042774

1400593419

1200274027

随着计算机的普及,机器语言编程愈发显得效率低下,并且枯燥和易于出错。程序员开始使用类似英语的缩写代表计算机的基本操作,而不是使用计算机可以直接理解的数字串。这些缩写构成了"汇编语言"的基础。一种"翻译程序"(称为汇编程序)将汇编语言程序转化为机器语言程序。下面是一段汇编语言代码,也是将"加班工资"和"基本工资"相加,并将结果存入"工资总额",显然比机器语言更易于理解。

```
LOAD    BASEPAY
ADD     OVERPAY
STORE   GROSSPAY
```

这样的代码计算机不能理解,还要转化为机器语言才行。

尽管随着汇编语言的出现,计算机的使用迅速增加,但是即使要完成一件最简单的任务,汇编语言仍然需要很多条指令。为了加快编程的过程,又产生了高级语言,可以用简单的语句完成大量的任务。称为"编译器"的翻译程序将高级语言转化为机器语言。程序员使用高级语言写出的指令和日常英语非常相似,并且还包含常见的数学符号。上面的工资处理程序如果用高级语言书写,仅仅需要如下的一条语句:

```
grosspay = basepay + overpay;
```

显然,与机器语言或汇编语言相比,高级语言更受程序员欢迎。

1.3 C#语言概述

C#(读作C Sharp)语言是微软公司于2000年7月发布的一种新的面向对象的编程语言,它扎根于C、C++和Java,吸取了每种语言的优点并增加了自己的特点。由于C#建立在已经广泛使用和良好发展的语言基础之上,程序员会发现学习C#是件容易并且有趣的事情。C#语

言有如下特点：

(1)简洁的语法；

(2)精心的面向对象设计；

(3)与 Web 的紧密结合；

(4)较高的安全性与错误处理能力；

(5)完善的版本处理技术；

(6)灵活性和兼容性好。

C#语言是微软为. NET 平台专门设计的语言。. NET 平台是一个用来建立、开发、运行和发布基于因特网的服务和应用程序的平台，允许以不同的语言创建的应用程序相互通信。. NET 平台的构成如图 1-1 所示。

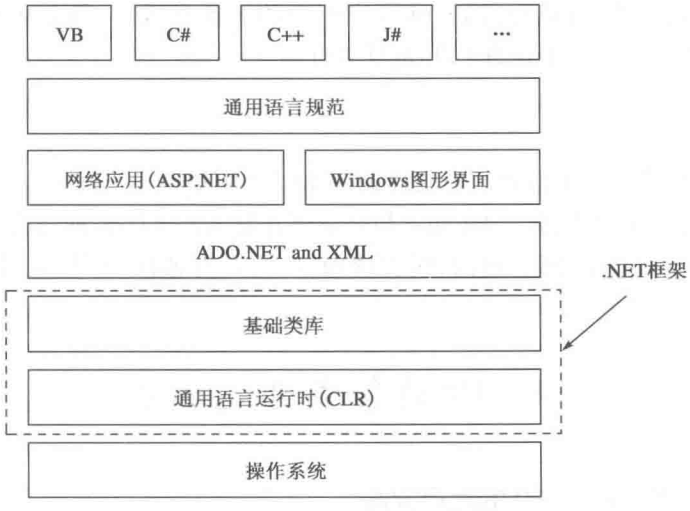

图 1-1 NET 平台的构成

. NET 平台的核心是. NET 框架，它是生成、运行. NET 应用程序和 Web Service 的组件库，包括两个主要组件，一个是公共语言运行时(Common Language Runtime，又称为运行库)，另一个是类库。运行库提供. NET 应用程序所需要的核心服务，类库为开发和运行. NET 应用程序提供了各种支持。使用. NET 框架开发的应用程序，不论使用的是哪种高级语言，均必须在安装了. NET 框架上的计算机才能运行。这种架构与 Java 应用程序必须由 Java 虚拟机支持相似。

1.3.1 公共语言运行时

公共语言运行时(CLR)提供程序的执行环境，和 Jvav 的虚拟机相似，处理程序的装入、编译、连接和管理程序的执行，并且提供内存管理、线程管理、远程处理、跨语言集成、跨语言异常处理和良好的安全性服务。

使用. NET 框架提供的编译器可以直接将源程序编译为. EXE 或. DLL 文件。但是需要注意的是，此时编译出来的程序代码并不是 CPU 能直接执行的机器代码，而是一种中间语言 IL (Microsoft Defined Intermediate Language)代码。

使用中间语言代码的优点有两点：一是可以是实现平台无关性，即与特定的 CPU 无关；二是只要能将某种语言编译为 IL 代码，就可以实现这些语言之间的交互操作。

在代码被调用执行时，CLR 会将需要的 IL 代码调入内存，然后再通过 JIT 编译器（Just-In-Time）将其编译成所用平台的 CPU 可直接执行的机器代码。但是要注意，JIT 编译器进行即时编译时，并不是一次把整个应用程序全部编译完，而是只编译它调用的那部分代码所在的模块。一旦代码经过 JIT 编译，得到的 CPU 可直接执行的代码就保存在内存中，直到退出该应用程序为止。这样再次执行这部分代码时，就不需要重新编译了。

为什么要采用即时编译呢？这是因为 JIT 编译器可以有效地提高系统的性能。由于即时编译是在执行程序的过程中进行的，JIT 编译器自然知道程序运行在什么类型的 CPU 上，因此它可以根据现有的 CPU 性能生成更加优化的可执行代码。而. NET 之前的编译器一开始就将源程序编译为特定的 CPU 可执行代码，这样当用户的机器升级后，除非再针对该种类型的 CPU 重新编译，否则就无法利用现有 CPU 的优秀性能。

1.3.2 类库

类库是一个与公共语言运行时紧密集成的可重用类的集合。. NET Framework 4 版的类库由 4000 多个类组成，这些类提供了 Internet 和企业级开发所需要的各种功能，支持基本的输入输出、字符串操作、安全管理、网络通信、线程管理等，为开发各种. NET 应用程序提供了很大的方便。

1.4 C#的集成开发环境

1.4.1 Visual Studio 2010 的运行界面

C#语言的集成开发环境是 Visual Studio，目前最新的版本是 Visual Studio 2010，用来创建、运行和调试由各种. NET 编程语言编写的程序。Visual Studio 2010 运行界面如图 1-2 所示。

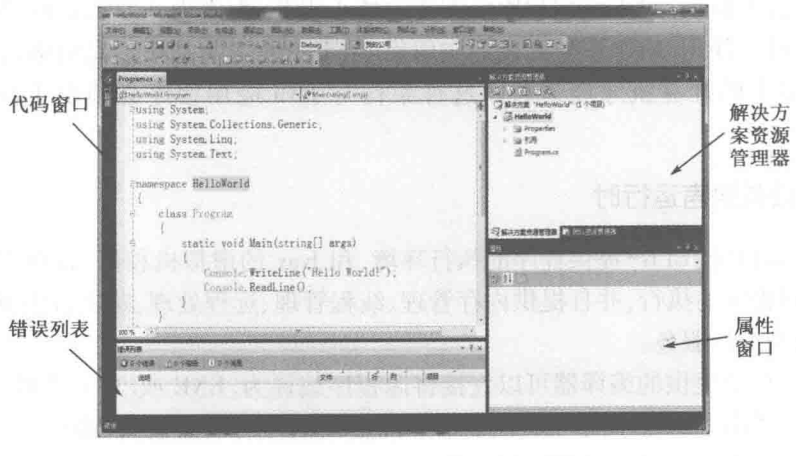

图 1-2 Visual Studio 2010 的运行界面

由图 1-2 可以看出，Visual Studio 2010 运行界面主要由：代码窗口、解决方案资源管理器窗口、属性窗口、错误列表窗口四个子窗口构成。代码窗口用来显示程序的代码，一般显示一个文件的内容（当显示多个文件的内容时，需通过代码窗口上方的选项卡在不同文件之间切换）。错误列表窗口用来显示编译当前程序所产生的错误信息。属性窗口主要用于图形用户界面设计时显示对象的各个属性值。解决方案资源管理器窗口用来显示项目的文件结构。

需要说明的是，Visual Studio 2010 是按照解决方案组织和管理各种程序和文件的。一个解决方案可以包含多个项目，不同的项目完成不同的功能，项目之间相对独立。大多数情况下，一个解决方案只包括一个项目。一个项目完成一个独立的功能，可以包含多个文件和子目录，子目录下面又可以包含多个文件。

C#源程序文件扩展名为 . CS，如 Program. CS，一个文件可以包含一个类，也可以包含多个类。

值得注意的是，上述窗口布局是在默认配置情况下的显示界面。有些初学者由于不熟悉 Visual Studio 2010 的开发环境而操作错误，导致显示界面和默认配置下的窗口布局不一样，如属性窗口被关闭、错误列表窗口被关闭等，造成操作上的不便。此时，可以使用【窗口】菜单→【重置窗口布局】命令恢复默认情况下的窗口布局。

1.4.2 Visual Studio 2010 应用程序的创建

Visual Studio 2010 开发平台提供了很多应用程序模板，常用的有以下几种。

(1)控制台应用程序

控制台应用程序在命令行方式运行，用于交互性操作不多、主要偏重于内部功能的应用程序。控制台应用程序十分适合 C#的初学者学习基本 C#语句和语法。

(2) Windows 应用程序

Windows 应用程序实现 Windows 窗体形式的操作界面，主要用于交互性操作较多的场合，如大型网络游戏、复杂的办公软件、大量网络信息传递以及其他高端的网络开发与应用设计等。

(3) ASP. NET Web 应用程序

ASP. NET Web 应用程序通过 Internet 传递可以被客户浏览的页面，如目前流行的各类网站以及基于 Web 的网络办公系统。

(4) ASP. NET Web 服务应用程序

Web 服务应用程序主要用于在服务器端通过 Internet 提供给 Windows 应用程序和 Web 应用程序调用的功能模块。这些模块既可以提供给服务器端应用程序调用，也可以提供给客户端应用程序调用。

(5)安装和部署应用程序

Visual Studio 2010 开发平台为安装和部署应用程序提供了模板。在完成了某个项目的开发之后，开发人员可以利用 Visual Studio 2010 开发平台生成该项目的安装和部署应用程序。

下面以控制台应用程序为例，说明在 Visual Studio 2010 中创建和运行应用程序的方法。

【例 1-1】 编写一个应用程序，显示“Hello, World”。

(1)运行 Visual Studio 2010，单击【文件】菜单，选【新建】命令，再选【项目】子命令，在弹出

的窗体中选择【控制台应用程序】模板，输入项目名 HelloWorld，如图 1－3 所示。

（2）单击【确定】按钮，Visual Studio 2010 将创建一个新的解决方案（和项目名同名）和一个项目，并在指定位置（在图 1-3 中是 F:\vsproject）创建一个新的文件夹 HelloWorld（同解决方案名）存储解决方案中的所有文件，文件夹结构如图 1-4 所示。

在 Visual Studio 2010 运行界面的代码窗口中，在自动生成的程序中添加一条输出语句。

```
using System;
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args){
            Console. WriteLine("Hello,World!");
        }
    }
}
```

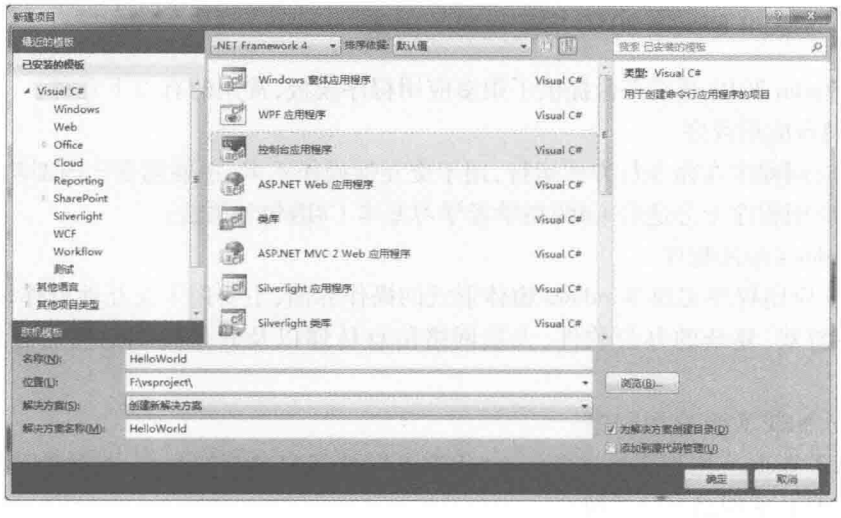

图 1-3　控制台应用程序的创建

图 1-4　解决方案文件夹的结构

(3)在 Visual Studio 2010 中运行项目,分为调试运行和不调试运行两类。无论是哪一类,系统都会先编译整个项目,然后自动运行。一般情况下,编译生成的可执行文件(.exe 文件)默认保存在项目文件夹下的 bin\debug 子目录下,编译上述项目将在该目录中生成 HelloWorld.exe 文件。如果不需要调试程序,一般选不调试运行。如调试运行项目选【调试】菜单→【启动调试】命令(或按 <F5> 键),不调试运行项目可以选【调试菜单】→【开始执行(不调试)】命令(或按 Ctrl + F5 键)。上述程序按不调试运行的结果如图 1-5 所示。

图 1-5　例 1-1 的运行结果

习　题

1. 什么是程序?
2. 机器语言、汇编语言和高级语言有什么区别和联系?
3. .NET 框架中,通用语言运行时的功能是什么?
4. 在 Visual Studio 2010 开发环境中,什么是解决方案和项目?

第2章　C#数据类型

2.1　常量和变量

为了让计算机处理数据，首先必须在内存中存储数据，这就要对内存单元进行操作。由于内存单元只有内存地址，不便于在程序中使用，因此需要为内存单元命名。这种命名的内存单元就是常量和变量，在程序运行中存储单元的值不能改变的内存单元称为常量，反之，可以改变的内存单元称为变量。在C#中，变量必须与数据类型相配合，系统按照相应的数据类型分配内存单元的数量。这些已定义数据类型的变量在程序执行中不能随意改变可以存储的数据类型，这个特性称为程序语言的静态类型系统。变量和常量都必须先声明，后使用。

2.1.1　常量

在C#中，使用const关键字和如下语法声明一个符号常量：

const 数据类型 常量名 = 值；

常量必须在声明时初始化，而且一经初始化，就不能改变了。习惯上，常量采用大写字母命名，最多不超过255个字符。例如：

```
const int MAXSIZE = 32768;
```

2.1.2　变量

变量在声明时就可以初始化，也可以在任何时候给变量赋新值，改变原值。用“=”运算符给变量赋值。例如：

```
int age = 30;
```

基本的变量命名规则如下：

(1)变量名的第一个字符必须是字符、下划线或@(但若第一个字符是@的话，第二个字符不能是数字)；

(2)其余的字符可以是字母、下划线或数字，字符总数不超过255。

例如，正确的变量名有：

```
x1
_test
@test
```

不正确的变量名有：

```
@123
88abc
```

注意：同C语言一样，C#是大小写敏感的，变量Test和test并不一样。另外，变量必须先声明，后使用；在对变量进行计算之前必须赋值。例如：

```
int x;
double y;
z = 10;                    //错误,因为变量 z 没有声明。
x = 2.3;                   //错误,数据类型不匹配。
y = y + 2;                 //错误,y 没有事先赋值。
x = 6;
y = x + 8;
```

2.2 数 据 类 型

由于计算机的中央处理器只能按照机器指令处理内存中的数据,所以在指示计算机处理数据之前,必须先在内存中存储数据。任何一个指令都包含了执行代码和数据存储两部分内容。由于各种数据在内存中占用的内存空间不同,不同类的数据运算方式不同,对其运算的机器指令也不同,因此大多数的程序设计语言对不同种类的数据都进行分别处理。有了数据类型之后,可以带来如下好处:

(1)指示计算机为数据分配适当的内存空间;

(2)可以检查程序对各种数据的运算是否合法;

(3)数据的各种运算能够容易地转换为机器指令;

(4)可以指示计算机方便地解释内存中的数据。

如图 2-1 内存的数据,按整数解释,值为 1311586011;按单精度浮点数则解释,值为 726513344;按字符串解释,为"中国"(按 Unicode 编码解释)。

01001110	00101101	010101110	11111101

图 2-1 内存数据的解释

必须牢记 C#是强类型语言,对数据是分类型存储的,因此对每一个常量和变量都要声明数据类型,编译器会检查变量的赋值是否正确。C#中类型可分为值类型和引用类型两类,两者主要的区别是在内存中存储的方式不同。值类型变量存储数据,引用类型变量存储对实际数据的引用(即地址)。

2.2.1 值类型

值类型又可以分为简单类型、枚举类型、结构类型。此节我们只介绍简单类型,枚举类型和结构类型在后面的章节介绍。

1. 整数类型

C#语言提供了 8 种整数类型,其表现形式及取值范围如表 2-1 所示。

给整数类型的变量赋值时,可采用十进制或十六进制的数值常数。如果是十六进制的常数,在书写时要加前缀"0x",如:

```
long  x = 0x12ab;
```