

软件工程师培养丛书

# C#编程技术基础

武汉厚溥教育科技有限公司 编著

“理论→总结→上机→习题”四阶段教学模式

- ★ 理论结合实践，注重动手能力培养
- ★ 任务驱动讲解，有效激发学习兴趣
- ★ 典型项目案例，扎实培养专业素质
- ★ 教学做一体化，极大提高教学效率



清华大学出版社

软件工程师培养丛书

# C#编程技术基础

武汉厚溥教育科技有限公司 编著

清华大学出版社

北 京

## 内 容 简 介

本书按照高等院校、高职高专计算机课程基本要求,以案例驱动的形式来组织内容,突出计算机课程的实践性特点。本书共包括 12 章:深入了解.NET Framework、C#语法基础、类和对象(一)、类和对象(二)、C# OOP 深入、C#事件处理、继承和多态、抽象类和接口、常用类、集合和泛型、调试和异常处理以及 C#中的文件处理。

本书附赠 PPT 教学课件和案例源文件,这些教学资源可通过<http://www.tupwk.com.cn/downpage> 下载。

本书内容安排合理,层次清楚,通俗易懂,实例丰富,突出理论和实践的结合,可作为各类高等院校、高职高专及培训机构的教材,也可供广大程序设计人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

## 图书在版编目(CIP)数据

C#编程技术基础 / 武汉厚溥教育科技有限公司 编著. —北京:清华大学出版社, 2014  
(软件工程师培养丛书)

ISBN 978-7-302-34747-7

I. ①C… II. ①武… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 295213 号

责任编辑:刘金喜 蔡 娟

封面设计:崔东方

版式设计:妙思品位

责任校对:邱晓玉

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课 件 下 载: <http://www.tup.com.cn>, 010-62794504

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:20.25

字 数:332 千字

版 次:2014 年 1 月第 1 版

印 次:2014 年 1 月第 1 次印刷

印 数:1~3000

定 价:36.00 元

产品编号:057154-01

# 编委会

## 主任:

翁高飞

## 副主任:

|     |     |     |     |
|-----|-----|-----|-----|
| 王 鹏 | 余晓刚 | 刘 伟 | 曹 静 |
| 方风波 | 邹治伟 | 李建利 | 管胜波 |

## 委员:

|     |     |     |     |
|-----|-----|-----|-----|
| 罗 炜 | 耿 杰 | 谢日星 | 吴金秀 |
| 夏超群 | 陈 琴 | 夏 晶 | 彭 莉 |
| 徐 霞 | 明素华 | 王 敏 | 严 滔 |
| 胡智方 |     |     |     |

# 前言

C#(C Sharp)是微软(Microsoft)公司为.NET Framework 量身订做的一种面向对象编程语言,于2000年6月发布。C#拥有C/C++的强大功能和 Visual Basic 简单易用的特性,是第一个面向组件(component-oriented)的程序语言,与C++和Java一样也是面向对象(object-oriented)程序语言,但是C#程序只能在Windows下运行。C#是微软公司研究员 Anders Hejlsberg 的研究成果。

本书隶属于“软件工程师培养丛书”中的一本专业基础教材,该丛书是由武汉厚溥教育科技有限公司开发,以培养符合企业需求的软件工程师应用开发、实施为目标的IT职业教育丛书。在开发该丛书之前,我们对IT行业的岗位序列做了充分的调研,包括研究从业人员技术方向、项目经验和职业素质等方面的需求,通过对面向的学生特点、行业需求的现状以及实施等方面的详细分析,结合“厚溥”对软件人才培养模式的认知,按照软件专业总体定位要求,进行软件专业产品课程体系设计。该丛书集应用软件知识和多领域的实践项目于一体,着重培养学生的熟练度、规范性、集成和项目能力,从而达到预定的培养目标。

本书共包括12章:深入了解.NET Framework、C#语法基础、类和对象(一)、类和对象(二)、C# OOP 深入、C#事件处理、继承和多态、抽象类和接口、常用类、集合和泛型、调试和异常处理以及C#中的文件处理。

我们对本书的编写体系做了精心的设计,按照“理论学习—知识总结—上机操作—课后习题”这一思路进行编排。“理论学习”部分描述通过本案例要达到的学习目的与涉及的相关知识点,使学习目标更加明确;“知识总结”部分概括案例所



涉及的知识点，使知识点完整系统地呈现；“上机操作”部分对案例进行了详尽分析，通过完整的步骤帮助读者快速掌握该案例的操作方法；“课后习题”部分帮助读者理解章节的知识点。本书在内容编写方面，力求细致全面；在文字叙述方面，注意言简意赅、重点突出；在案例选取方面，强调案例的针对性和实用性。

本书凝聚了编者多年来的教学经验和成果，可作为各类高等院校、高职高专及培训机构的教材，也可供广大程序设计人员参考。

本书PPT教学课件和案例源文件可通过<http://www.tupwk.com.cn/downpage>下载。

本书由武汉厚溥教育科技有限公司编著，由翁高飞、王鹏等多名企业实战项目经理编写。本书编者长期从事项目开发和教学实施，并且对当前高校的教学情况非常熟悉，在编写过程中充分考虑到不同学生的特点和需求，加强了项目实战方面的教学。本书编写过程中，得到了武汉厚溥教育科技有限公司各级领导的大力支持，在此对他们表示衷心的感谢。

参与本书编写的人员还有：武汉商学院曹静老师、荆州职业技术学院方风波老师、武汉工程职业技术学院邹治伟和彭莉老师、湖北三峡职业技术学院李建利老师、武汉软件工程职业学院谢日星老师、黄冈职业技术学院吴金秀老师、湖北国土资源职业学院徐霞和明素华老师等。

限于编写时间和编者的水平，书中难免存在不足之处，希望广大读者批评指正。

服务邮箱：[wkservice@163.com](mailto:wkservice@163.com)。

编 者

2013年10月

# 目 录

## 第1章 深入了解.NET Framework .....1

### 1.1 .NET Framework 回顾.....2

### 1.2 .NET Framework 的体系结构.....2

### 1.3 .NET Framework 的组件 .....3

#### 1.3.1 公共语言运行时 .....3

#### 1.3.2 通用类型系统.....4

#### 1.3.3 公共语言规范.....5

### 1.4 .NET 程序编译原理.....5

### 1.5 .NET 框架类库 .....6

#### 1.5.1 框架类库.....6

#### 1.5.2 框架类库中的命名空间.....6

### 1.6 示例.....7

### 【小结】 .....8

### 【自测题】 .....8

## 第2章 C#语法基础.....9

### 2.1 预定义数据类型.....10

#### 2.1.1 为什么需要区分数据类型.....10

#### 2.1.2 主要预定义数据类型.....10

### 2.2 程序中的变量和常量.....11

#### 2.2.1 变量的概念和作用.....11

#### 2.2.2 变量的定义.....12

#### 2.2.3 变量的赋值.....13

#### 2.2.4 常量.....13

#### 2.2.5 使用 var 创建隐型局部变量 .....13

### 2.3 表达式和常用运算符.....16

#### 2.3.1 表达式以及运算符的分类的 定义.....16

#### 2.3.2 算术运算符.....16

#### 2.3.3 比较运算符.....16

#### 2.3.4 逻辑运算符.....17

#### 2.3.5 快捷运算符.....17

#### 2.3.6 三元运算符.....18

#### 2.3.7 运算符优先级 .....18

### 2.4 C#中的程序控制结构.....19

#### 2.4.1 条件判断结构.....19

#### 2.4.2 循环结构.....23

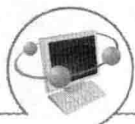
### 2.5 数组.....26

#### 2.5.1 数组的定义.....26

#### 2.5.2 数组的赋值和取值.....27

#### 2.5.3 使用 var 和数组初始化器 创建隐型数组.....28

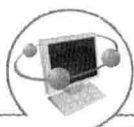
#### 2.5.4 数组结合循环.....29



|                               |     |                               |     |
|-------------------------------|-----|-------------------------------|-----|
| 2.6 枚举.....                   | 31  | 3.6.1 数据类型的分类: 值类型和引用类型.....  | 74  |
| 2.7 简单类型转换.....               | 35  | 3.6.2 值类型和引用类型的转换: 装箱和拆箱..... | 78  |
| 2.7.1 数值类型转换成字符串.....         | 35  | 【小结】.....                     | 79  |
| 2.7.2 字符串转换成数值类型.....         | 36  | 【自测题】.....                    | 80  |
| 【小结】.....                     | 36  | 【上机部分】.....                   | 80  |
| 【自测题】.....                    | 37  | 【课后作业】.....                   | 88  |
| 【上机部分】.....                   | 38  | 第3章 类和对象(一).....              | 47  |
| 【课后作业】.....                   | 46  | 3.1 C#——面向对象语言.....           | 48  |
| 第3章 类和对象(一).....              | 47  | 3.1.1 面向对象语言的诞生和特点.....       | 48  |
| 3.1 C#——面向对象语言.....           | 48  | 3.1.2 面向对象语言基础——类的概念和定义.....  | 49  |
| 3.1.1 面向对象语言的诞生和特点.....       | 48  | 3.1.3 对象.....                 | 50  |
| 3.1.2 面向对象语言基础——类的概念和定义.....  | 49  | 3.1.4 创建匿名类的对象.....           | 53  |
| 3.1.3 对象.....                 | 50  | 3.2 成员方法.....                 | 54  |
| 3.1.4 创建匿名类的对象.....           | 53  | 3.2.1 成员方法的定义.....            | 54  |
| 3.2 成员方法.....                 | 54  | 3.2.2 方法调用.....               | 55  |
| 3.2.1 成员方法的定义.....            | 54  | 3.3 构造方法.....                 | 56  |
| 3.2.2 方法调用.....               | 55  | 3.3.1 为什么需要构造方法.....          | 56  |
| 3.3 构造方法.....                 | 56  | 3.3.2 this 关键字.....           | 62  |
| 3.3.1 为什么需要构造方法.....          | 56  | 3.4 命名空间.....                 | 64  |
| 3.3.2 this 关键字.....           | 62  | 3.5 面向对象语言特点和访问修饰符.....       | 67  |
| 3.4 命名空间.....                 | 64  | 3.5.1 面向对象语言的特点.....          | 67  |
| 3.5 面向对象语言特点和访问修饰符.....       | 67  | 3.5.2 访问修饰符.....              | 68  |
| 3.5.1 面向对象语言的特点.....          | 67  | 3.6 值类型和引用类型.....             | 74  |
| 3.5.2 访问修饰符.....              | 68  | 3.6.1 数据类型的分类: 值类型和引用类型.....  | 74  |
| 3.6 值类型和引用类型.....             | 74  | 3.6.2 值类型和引用类型的转换: 装箱和拆箱..... | 78  |
| 3.6.1 数据类型的分类: 值类型和引用类型.....  | 74  | 【小结】.....                     | 79  |
| 3.6.2 值类型和引用类型的转换: 装箱和拆箱..... | 78  | 【自测题】.....                    | 80  |
| 【小结】.....                     | 79  | 【上机部分】.....                   | 80  |
| 【自测题】.....                    | 80  | 【课后作业】.....                   | 88  |
| 【上机部分】.....                   | 80  | 第4章 类和对象(二).....              | 89  |
| 【课后作业】.....                   | 88  | 4.1 类的静态成员.....               | 90  |
| 第4章 类和对象(二).....              | 89  | 4.1.1 静态方法和static 关键字.....    | 90  |
| 4.1 类的静态成员.....               | 90  | 4.1.2 静态成员变量.....             | 93  |
| 4.1.1 静态方法和static 关键字.....    | 90  | 4.2 ref 关键字和 out 关键字.....     | 94  |
| 4.1.2 静态成员变量.....             | 93  | 4.2.1 ref 关键字.....            | 94  |
| 4.2 ref 关键字和 out 关键字.....     | 94  | 4.2.2 out 关键字.....            | 96  |
| 4.2.1 ref 关键字.....            | 94  | 4.3 成员方法的重载.....              | 97  |
| 4.2.2 out 关键字.....            | 96  | 【小结】.....                     | 101 |
| 4.3 成员方法的重载.....              | 97  | 【自测题】.....                    | 101 |
| 【小结】.....                     | 101 | 【上机部分】.....                   | 102 |
| 【自测题】.....                    | 101 | 【课后作业】.....                   | 106 |
| 【上机部分】.....                   | 102 | 第5章 C# OOP 深入.....            | 107 |
| 【课后作业】.....                   | 106 | 5.1 类似于类的数据类型——结构体.....       | 108 |
| 第5章 C# OOP 深入.....            | 107 | 5.2 访问私有成员的利器——属性.....        | 110 |
| 5.1 类似于类的数据类型——结构体.....       | 108 | 5.2.1 如何定义和使用属性.....          | 111 |
| 5.2 访问私有成员的利器——属性.....        | 110 | 5.2.2 自动属性.....               | 114 |
| 5.2.1 如何定义和使用属性.....          | 111 |                               |     |
| 5.2.2 自动属性.....               | 114 |                               |     |



|                           |            |                                         |            |
|---------------------------|------------|-----------------------------------------|------------|
| 5.3 索引器.....              | 114        | 7.1.3 base 关键字和 protected<br>访问修饰符..... | 159        |
| 5.4 静态类.....              | 118        | 7.2 密封类.....                            | 163        |
| 5.5 使用类图查看类的构造.....       | 119        | 7.3 多态.....                             | 164        |
| 【小结】.....                 | 120        | 【小结】.....                               | 168        |
| 【自测题】.....                | 121        | 【自测题】.....                              | 168        |
| 【上机部分】.....               | 121        | 【上机部分】.....                             | 170        |
| 【课后作业】.....               | 128        | 【课后作业】.....                             | 176        |
| <b>第 6 章 C#事件处理</b> ..... | <b>129</b> | <b>第 8 章 抽象类和接口</b> .....               | <b>177</b> |
| 6.1 委托.....               | 130        | 8.1 抽象类.....                            | 178        |
| 6.1.1 定义委托.....           | 131        | 8.2 接口.....                             | 182        |
| 6.1.2 实例化委托.....          | 131        | 8.2.1 使用接口.....                         | 183        |
| 6.1.3 调用委托.....           | 133        | 8.2.2 继承基类并实现接口.....                    | 184        |
| 6.1.4 匿名方法.....           | 133        | 8.2.3 多重接口实现.....                       | 186        |
| 6.2 事件.....               | 135        | 8.2.4 is 和 as 关键字.....                  | 187        |
| 6.2.1 定义事件.....           | 136        | 8.2.5 接口绑定.....                         | 188        |
| 6.2.2 订阅事件.....           | 136        | 8.2.6 接口作为参数的意义.....                    | 189        |
| 6.2.3 引发事件.....           | 137        | 8.2.7 接口小结.....                         | 189        |
| 6.3 自定义事件完整实例.....        | 137        | 8.3 接口和抽象类的区别.....                      | 189        |
| 6.4 含参数事件完整实例.....        | 139        | 【小结】.....                               | 190        |
| 【小结】.....                 | 143        | 【自测题】.....                              | 190        |
| 【自测题】.....                | 143        | 【上机部分】.....                             | 191        |
| 【上机部分】.....               | 144        | 【课后作业】.....                             | 193        |
| 【课后作业】.....               | 150        | <b>第 9 章 常用类</b> .....                  | <b>195</b> |
| <b>第 7 章 继承和多态</b> .....  | <b>151</b> | 9.1 Math 类.....                         | 196        |
| 7.1 继承.....               | 152        | 9.2 Random 类.....                       | 197        |
| 7.1.1 继承 C#中的类.....       | 152        | 9.3 DateTime 结构.....                    | 200        |
| 7.1.2 继承中的构造方法.....       | 156        |                                         |            |



|                                      |     |                              |     |
|--------------------------------------|-----|------------------------------|-----|
| 9.4 System.String 类.....             | 202 | 10.5 Lambda 表达式与语句.....      | 238 |
| 9.5 StringBuilder 类.....             | 204 | 10.5.1 Lambda 表达式.....       | 238 |
| 9.6 正则表达式.....                       | 205 | 10.5.2 Lambda 语句.....        | 239 |
| 9.6.1 System.Text.RegularExpressions |     | 10.5.3 带有标准查询运算符的            |     |
| 命名空间.....                            | 206 | Lambda.....                  | 239 |
| 9.6.2 Regex 类.....                   | 206 | 10.6 扩展方法.....               | 240 |
| 9.6.3 Match 类和                       |     | 10.6.1 扩展方法一:                |     |
| MatchCollection 类.....               | 211 | FirstOrDefault().....        | 241 |
| 9.6.4 模糊匹配.....                      | 212 | 10.6.2 扩展方法二: Max().....     | 243 |
| 【小结】.....                            | 214 | 10.6.3 扩展方法三:                |     |
| 【自测题】.....                           | 214 | SingleOrDefault().....       | 244 |
| 【上机部分】.....                          | 215 | 10.6.4 扩展方法四: Where().....   | 245 |
| 【课后作业】.....                          | 218 | 10.6.5 扩展方法五: Select().....  | 246 |
| 第 10 章 集合和泛型.....                    | 219 | 10.7 IComparable 接口实现排序..... | 247 |
| 10.1 System.Array 概述.....            | 220 | 10.8 泛型接口.....               | 252 |
| 10.2 System.Array 的属性和               |     | 10.8.1 IComparable<T>接口..... | 252 |
| 方法.....                              | 221 | 10.8.2 IComparer<T>接口.....   | 253 |
| 10.3 System.Collections 命名           |     | 【小结】.....                    | 256 |
| 空间.....                              | 223 | 【自测题】.....                   | 257 |
| 10.3.1 ArrayList 类.....              | 223 | 【上机部分】.....                  | 257 |
| 10.3.2 Hashtable 类.....              | 226 | 【课后作业】.....                  | 264 |
| 10.4 泛型集合.....                       | 229 | 第 11 章 调试和异常处理.....          | 265 |
| 10.4.1 System.Collections.Generic    |     | 11.1 调试.....                 | 266 |
| 命名空间.....                            | 231 | 11.1.1 调试的必要性.....           | 266 |
| 10.4.2 List<T>类.....                 | 231 | 11.1.2 调试过程.....             | 267 |
| 10.4.3 Dictionary<TKey,TValue>类..... | 234 | 11.1.3 Visual Studio 2008 中的 |     |
| 10.4.4 对象与集合初始化器.....                | 237 | 调试工具.....                    | 270 |



|                              |     |                       |     |
|------------------------------|-----|-----------------------|-----|
| 11.2 异常.....                 | 272 | 12.2 File 类.....      | 290 |
| 11.2.1 System.Exception..... | 273 | 12.3 Directory 类..... | 291 |
| 11.2.2 try 和 catch 块.....    | 275 | 12.4 对文本文件的读写操作.....  | 292 |
| 11.2.3 使用 throw 引发异常.....    | 276 | 12.4.1 文件流.....       | 293 |
| 11.2.4 使用 finally.....       | 276 | 12.4.2 流读写对象.....     | 294 |
| 11.2.5 多重 catch 块.....       | 277 | 12.5 二进制文件的读写.....    | 297 |
| 11.2.6 自定义异常类.....           | 277 | 12.6 序列化和反序列化.....    | 301 |
| 11.3 应用程序示例.....             | 278 | 【小结】.....             | 304 |
| 【小结】.....                    | 280 | 【自测题】.....            | 304 |
| 【自测题】.....                   | 281 | 【上机部分】.....           | 305 |
| 【上机部分】.....                  | 281 | 【课后作业】.....           | 309 |
| 【课后作业】.....                  | 288 | 参考文献.....             | 311 |
| 第 12 章 C# 中的文件处理.....        | 289 |                       |     |
| 12.1 System.IO 命名空间.....     | 290 |                       |     |

# 第1章

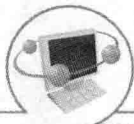
# 深入了解

# .NET Framework



## 课程目标

- ▶ 回顾.NET Framework
- ▶ .NET Framework的体系结构
- ▶ .NET Framework的组件
- ▶ .NET 程序编译原理
- ▶ 框架类库简介和简单使用体验



## 简介

.NET Framework 是一个框架，是一门新技术，是一个非常大的代码库。目前最新版本是 .NET Framework 4.5，它包含很多概念，刚接触时不是很容易理解，但是快速浏览这些基础知识对于理解如何利用 C# 进行编程是非常重要的，所以先学习 .NET Framework 是不可避免的。

C# 是一门新的面向对象的编程语言，我们不能孤立地使用 C# 语言，而必须与 .NET Framework 一起考虑。C# 是专门为与 Microsoft 的 .NET Framework 一起使用而设计的。

### 1.1 .NET Framework 回顾

.NET Framework 是 Microsoft 为开发应用程序而创建的一个富有革命性的新平台，它定义了应用程序的开发和运行环境。.NET Framework 可以创建 Windows 应用程序、Web 应用程序、Web 服务和其他各种类型的应用程序。

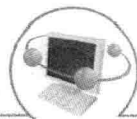
.NET Framework 的设计方式保证它可以用于各种语言，包括本书要介绍的 C# 语言，以及 C++、Visual Basic、JScript，甚至一些旧的语言，如 COBOL。目前还在不断推出更多的 .NET 版本的语言。

.NET Framework 包括最早期的 1.0 版，目前经历了 1.1、2.0、3.0、3.5、4.0 和最新的 4.5 版。

### 1.2 .NET Framework 的体系结构

.NET Framework 是一个创建、部署和运行应用程序的多语言多平台环境。.NET Framework 包含以下两个主要组件。

- 公共语言运行时(Common Language Runtime, CLR)



- .NET Framework 类库(Framework Class Library, FCL)

重用代码, 避免重复开发并缩短时间, 这一直是软件开发人员的目标。.NET Framework 提供了许多开发人员可重用的基础类, 包括线程、文件、数据库支持、XML 分析和数据结构等各个方面, 并且这些类库可用于所有支持.NET Framework 的编程语言。通过 CLR 支持, 任何.NET 语言都可以使用.NET 类库中的所有类。

除 CLR 和 FCL 外, .NET Framework 还对应着两种编程模型: ASP.NET(Web Forms 及 Web Services)和 WinForms。每种编程模型又对应着很多不同的编程语言, 如图 1-1 所示。

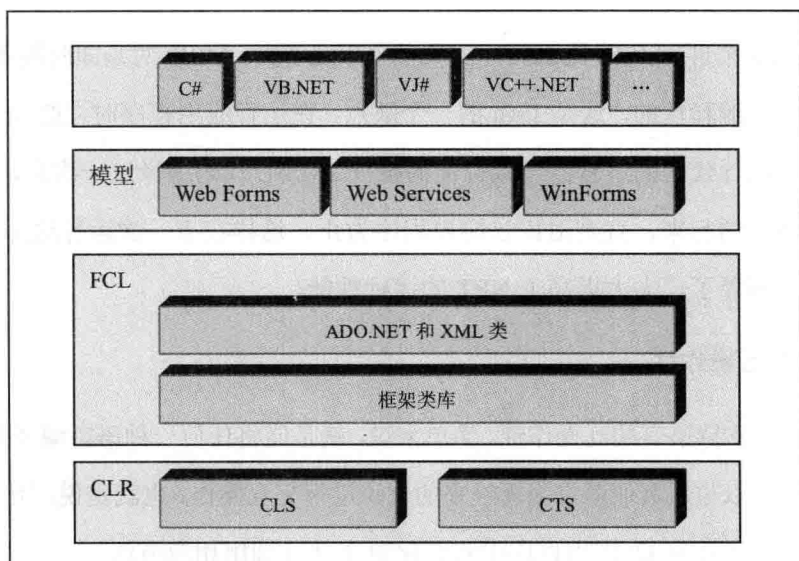


图 1-1

## 1.3 .NET Framework 的组件

### 1.3.1 公共语言运行时

.NET Framework 的核心是其运行库的执行环境, 称为公共语言运行时(CLR)或.NET 运行库。通常将在 CLR 的控制下运行的代码称为托管代码。

但是, 在 CLR 执行开发的源代码之前, 需要编译它们(语言本身自己的编译器)。在.NET 中, 编译分为如下两个阶段。



- 把源代码编译为 Microsoft 中间语言(Microsoft Intermediate Language, MSIL)。
- CLR 把 MSIL 编译为平台专用的机器代码。

CLR 从某种意义上理解相当于 Java 中的 Java 虚拟机(JVM), 而 MSIL 相当于 Java 中的字节码(.class 文件)。MSIL 总是即时编译(称为 JIT 编译)为相应平台的机器代码, 这一点与 Java 也很相似。

.NET 代码托管有如下优点。

### 1. 提高性能

相对于 Java 来讲, MSIL 比 Java 字节码的作用还要大。MSIL 总是即时编译的, 而 Java 字节代码常常是解释性的, 这是 Java 的一个缺点, 在运行应用程序时, 把 Java 字节代码转换为内部可执行代码的过程会导致性能的损失。MSIL 代码编译过一次后, 得到的内部可执行代码就存储起来, 直到退出该应用程序为止。这样在下次运行这部分代码时, 就不需要重新编译了, 大大提高了 .NET 的运行性能。

### 2. 语言的互操作性

使用 MSIL 支持语言的互操作性。简单来说, 就是能将任何一种语言编译为中间代码, 编译好的代码可以与从其他语言编译过来的代码进行交互操作。也就是说, 用 Visual Basic 语言声明的类, 可以在 C# 应用程序中实例化对象并且调用相应方法。

无论用 Visual Basic 编写代码, 还是用 C# 或者 C++ 编写代码, CLR 都可以把源代码编译成 MSIL, 也就是中间语言, 这样才能实现语言互操作。那么, CLR 是如何做到这一点的呢? CLR 包含两个组成部分: CTS(Common Type System, 通用类型系统)和 CLS(Common Language Specification, 公共语言规范)。CLR 用这两个组件来完成语言互操作的功能。

## 1.3.2 通用类型系统

CTS 定义了一套可以在中间语言中使用的预定义数据类型, 所有面向 .NET Framework 的语言都可以生成最终基于这些类型的编译代码。也就是说, 通用类型系统用于解决不同编程语言的数据类型不同的问题。这也为跨语言的实现做出了重大贡献。例如, 在 Visual Basic 中定义整型变量, 是关键字 `integer`, 而在 C# 中用的是关键字 `int`, 但无论 Visual



Basic 还是 C#, 经编译后都映射为 `System.Int32`, 所以 CTS 实现了不同语言数据类型的最终统一。

### 1.3.3 公共语言规范

编程语言之间不仅仅是数据类型的不同, 语法也有非常大的区别。所以需要定义公共语言规范, CLR 定义了所有编程语言必须遵守的共同标准。公共语言规范和通用类型系统一起确保语言的互操作性。CLS 是一个最低标准集, 所有面向 .NET 的编译器都必须支持它。只要遵守这个标准编写的程序, 就可以在 .NET 框架下实现互相操作。例如, 在 C# 中变量名是区分大小写的, 而 VB.NET 不区分大小写, 这时, CLS 就规定编译后的 MSIL 必须除了大小写以外有其他的区别。

## 1.4 .NET 程序编译原理

编译 C#.NET 应用程序操作步骤如下。

(1) 使用 C# 语言编写应用程序代码。

(2) 把 C# 源程序编译为 MSIL, 以程序集的形式存在, CPU 不执行程序集, 如图 1-2 所示。

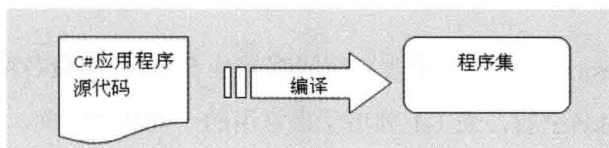


图 1-2

(3) 在执行代码时, 必须使用 JIT 编译器将程序集编译为本机代码, 如图 1-3 所示。



图 1-3

(4) 在托管的 CLR 环境下运行本机代码, 程序执行结果显示出来, 如图 1-4 所示。

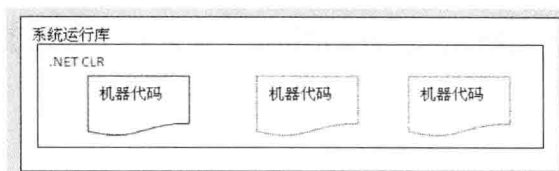


图 1-4

## 1.5 .NET 框架类库

### 1.5.1 框架类库

在前面讲解 .NET 框架结构时，了解到框架类库(FCL)有非常多的命名空间和类，这些类使用非常方便，且功能强大。框架类库提供了实现基本功能的类，如输入/输出、字符串的操作、网络通信等。

在 .NET 开发中，应用程序实现的很多功能不需要我们编写大量代码，只需要直接调用框架类库中相应的类就可以实现。那么，如何使用这些类呢？我们知道，命名空间用来将具有相关功能的一些类在结构上进行分组和管理，所以要使用框架类库中的类就必须了解 .NET 框架中的常用命名空间。框架类库中包含了 170 多个命名空间和上千个类，下面就来学习类库中的一些主要命名空间。

### 1.5.2 框架类库中的命名空间

在 .NET Framework 中，所有的命名空间都是从 System 的命名空间形成的。System 命名空间又称为根命名空间，表 1-1 列出了最常用的一些命名空间。

表 1-1

| 命名空间                       | 描述                          |
|----------------------------|-----------------------------|
| System.Drawing             | 处理图形和绘图                     |
| System.Data                | 处理数据存取和管理，在 ADO.NET 中扮演重要角色 |
| System.IO                  | 管理对文件和流的操作                  |
| System.Threading           | 包含用于多线程编程的类                 |
| System.Collections.Generic | 包含定义泛型集合的接口和类               |
| System.Collections         | 包含定义各种对象集合的接口和类             |
| System.Windows.Forms       | 包含用于开发 Windows 窗体应用程序的控件和类  |
| System.Net                 | 对网络协议进行编程                   |