

<<C#图解教程>>

图书基本信息

书名：<<C#图解教程>>

13位ISBN编号：9787115320902

10位ISBN编号：711532090X

出版时间：2013-6

出版单位：人民邮电出版社

作者：[美] Daniel M. Solis

译者：姚琪琳,苏 林,朱 晔

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<C#图解教程>>

内容概要

本书是广受赞誉的C# 图解教程的最新版本。

作者在本书中创造了一种全新的可视化叙述方式，以图文并茂的形式、朴实简洁的文字，并辅以大量表格和代码示例，全面、直观地阐述了C# 语言的各种特性。

新版本除了精心修订旧版内容外，还全面涵盖了C# 5.0 的新增特性，比如异步编程、调用者信息、case 表达式、带参数的泛型构造函数、支持null 类型运算等。

通过本书，读者能够快速、深入理解C#，为自己的编程生涯打下良好的基础。

本书是C# 入门的经典好书，适合对C# 感兴趣的所有读者。

作者简介

Daniel M. Solis 拥有20余年软件开发经验，微软、IBM等公司技术顾问，深受读者爱戴和推崇的技术图书作家。

他同时还是一位享誉世界的技术培训专家，在美国和欧洲各地传授编程语言、Windows程序设计和Unix底层技术的经验和心得。

多年的教学实践启发他创造了一种可视化的教学方法，并总结了一套代码与图示相结合的方法，为千千万万的入门读者扫除了步入编程职业生涯的障碍。

Illustrated WPF是他另一本畅销技术书。

书籍目录

第1章 C#和.NET框架 1 1.1 在.NET之前 1 1.1.1 20世纪90年代末的Windows编程 1 1.1.2 下一代平台服务的目标 2 1.2 .NET时代 2 1.2.1 .NET框架的组成 2 1.2.2 大大改进的编程环境 3 1.3 编译成CIL 5 1.4 编译成本机代码并执行 6 1.5 CLR 7 1.6 CLI 8 1.7 各种缩写 9 1.8 C#的演化 9 第2章 C#编程概述 10 2.1 一个简单的C#程序 10 2.2 标识符 12 2.3 关键字 12 2.4 Main：程序的起始点 13 2.5 空白 13 2.6 语句 14 2.7 从程序中输出文本 15 2.7.1 Write 15 2.7.2 WriteLine 15 2.7.3 格式字符串 16 2.7.4 多重标记和值 16 2.7.5 格式化数字字符串 17 2.8 注释：为代码添加注解 20 2.8.1 关于注释的补充 21 2.8.2 文档注释 21 2.8.3 注释类型总结 22 第3章 类型、存储和变量 23 3.1 C#程序是一组类型声明 23 3.2 类型是一种模板 24 3.3 实例化类型 24 3.4 数据成员和函数成员 25 3.5 预定义类型 26 3.6 用户定义类型 27 3.7 栈和堆 28 3.7.1 栈 28 3.7.2 堆 29 3.8 值类型和引用类型 30 3.8.1 存储引用类型对象的成员 31 3.8.2 C#类型的分类 31 3.9 变量 32 3.9.1 变量声明 32 3.9.2 多变量声明 34 3.9.3 使用变量的值 34 3.10 静态类型和dynamic关键字 34 3.11 可空类型 35 第4章 类的基本概念 36 4.1 类的概述 36 4.2 程序和类：一个快速示例 37 4.3 声明类 38 4.4 类成员 38 4.4.1 字段 38 4.4.2 方法 40 4.5 创建变量和类的实例 41 4.6 为数据分配内存 41 4.7 实例成员 42 4.8 访问修饰符 43 4.9 从类的内部访问成员 45 4.10 从类的外部访问成员 46 4.11 综合应用 47 第5章 方法 49 5.1 方法的结构 49 5.2 方法体内部的代码执行 50 5.3 本地变量 51 5.3.1 类型推断和var关键字 52 5.3.2 嵌套块中的本地变量 52 5.4 本地常量 53 5.5 控制流 54 5.6 方法调用 55 5.7 返回值 56 5.8 返回语句和void方法 57 5.9 参数 59 5.9.1 形参 59 5.9.2 实参 59 5.10 值参数 61 5.11 引用参数 63 5.12 引用类型作为值参数和引用参数 65 5.13 输出参数 68 5.14 参数数组 70 5.14.1 方法调用 71 5.14.2 用数组作为实参 73 5.15 参数类型总结 74 5.16 方法重载 74 5.17 命名参数 75 5.18 可选参数 76 5.19 栈帧 79 5.20 递归 81 第6章 深入理解类 83 6.1 类成员 83 6.2 成员修饰符的顺序 84 6.3 实例类成员 85 6.4 静态字段 86 6.5 从类的外部访问静态成员 86 6.5.1 静态字段示例 87 6.5.2 静态成员的生存期 87 6.6 静态函数成员 88 6.7 其他静态类成员类型 89 6.8 成员常量 90 6.9 常量与静态量 90 6.10 属性 91 6.10.1 属性声明和访问器 92 6.10.2 属性示例 93 6.10.3 使用属性 94 6.10.4 属性和关联字段 94 6.10.5 执行其他计算 96 6.10.6 只读和只写属性 96 6.10.7 属性与公共字段 97 6.10.8 计算只读属性示例 97 6.10.9 自动实现属性 98 6.10.10 静态属性 99 6.11 实例构造函数 100 6.11.1 带参数的构造函数 101 6.11.2 默认构造函数 102 6.12 静态构造函数 102 6.13 对象初始化语句 104 6.14 析构函数 105 6.15 readonly修饰符 105 6.16 this关键字 106 6.17 索引器 107 6.17.1 什么是索引器 108 6.17.2 索引器和属性 108 6.17.3 声明索引器 109 6.17.4 索引器的set访问器 110 6.17.5 索引器的get访问器 110 6.17.6 关于索引器的补充 111 6.17.7 为Employee示例声明索引器 111 6.17.8 另一个索引器的示例 112 6.17.9 索引器重载 113 6.18 访问器的访问修饰符 114 6.19 分部类和分部类型 115 6.20 分部方法 116 第7章 类和继承 118 7.1 类继承 118 7.2 访问继承的成员 119 7.3 所有类都派生自object类 120 7.4 屏蔽基类的成员 121 7.5 基类访问 123 7.6 使用基类的引用 124 7.6.1 虚方法和覆写方法 125 7.6.2 覆写标记为override的方法 127 7.6.3 覆盖其他成员类型 130 7.7 构造函数的执行 130 7.7.1 构造函数初始化语句 132 7.7.2 类访问修饰符 134 7.8 程序集间的继承 134 7.9 成员访问修饰符 136 7.9.1 访问成员的区域 137 7.9.2 公有成员的可访问性 138 7.9.3 私有成员的可访问性 138 7.9.4 受保护成员的可访问性 138 7.9.5 内部成员的可访问性 139 7.9.6 受保护内部成员的可访问性 139 7.9.7 成员访问修饰符小结 140 7.10 抽象成员 141 7.11 抽象类 142 7.11.1 抽象类和抽象方法的示例 142 7.11.2 抽象类的另一个例子 143 7.12 密封类 144 7.13 静态类 144 7.14 扩展方法 145 7.15 命名约定 148 第8章 表达式和运算符 150 8.1 表达式 150 8.2 字面量 151 8.2.1 整数字面量 152 8.2.2 实数字面量 153 8.2.3 字符字面量 153 8.2.4 字符串字面量 154 8.3 求值顺序 156 8.3.1 优先级 156 8.3.2 结合性 157 8.4 简单算术运算符 157 8.5 求余运算符 158 8.6 关系比较运算符和相等比较运算符 159 8.7 递增运算符和递减运算符 160 8.8 条件逻辑运算符 162 8.9 逻辑运算符 163 8.10 移位运算符 164 8.11 赋值运算符 165 8.12 条件运算符 167 8.13 一元算术运算符 168 8.14 用户定义的类型转换 169 8.15 运算符重载 172 8.15.1 运算符重载的限制 172 8.15.2 运算符重载的示例 173 8.16 typeof运算符 174 8.17 其他运算符 176 第9章 语句 177 9.1 什么是语句 177 9.2 表达式语句 178 9.3 控制流语句 179 9.4 if语句 179 9.5 if...else语句 180 9.6 while循环 181 9.7 do循环 182 9.8 for循环 183 9.8.1 for语句中变量的作用域 185 9.8.2 初始化和迭代表达式中的多表达式 185 9.9 switch语句 186 9.9.1 分支示例 187 9.9.2 switch语句的补充 188 9.9.3 分支标签 189 9.10 跳转语句 189 9.11 break语句 190 9.12 continue语句 190 9.13 标签语句 191 9.13.1 标签 191 9.13.2 标签语句的作用域 192 9.14 goto语句 192 9.15 using语句 193

<<C#图解教程>>

9.15.1 资源的包装使用 194 9.15.2 using语句的示例 195 9.15.3 多个资源和嵌套 196 9.15.4 using语句的另一种形式 197 9.16 其他语句 197 第10章 结构 198 10.1 什么是结构 198 10.2 结构是值类型 199 10.3 对结构赋值 200 10.4 构造函数和析构函数 201 10.4.1 实例构造函数 201 10.4.2 静态构造函数 202 10.4.3 构造函数和析构函数小结 203 10.5 字段初始化语句是不允许的 203 10.6 结构是密封的 203 10.7 装箱和拆箱 203 10.8 结构作为返回值和参数 204 10.9 关于结构的其他信息 204 第11章 枚举 205 11.1 枚举 205 11.1.1 设置底层类型和显式值 206 11.1.2 隐式成员编号 207 11.2 位标志 208 11.2.1 Flags特性 210 11.2.2 使用位标志的示例 212 11.3 关于枚举的补充 213 第12章 数组 216 12.1 数组 216 12.1.1 定义 216 12.1.2 重要细节 217 12.2 数组的类型 217 12.3 数组是对象 218 12.4 一维数组和矩形数组 219 12.5 实例化一维数组或矩形数组 220 12.6 访问数组元素 220 12.7 初始化数组 221 12.7.1 显式初始化一维数组 222 12.7.2 显式初始化矩形数组 222 12.7.3 初始化矩形数组的语法点 223 12.7.4 快捷语法 223 12.7.5 隐式类型数组 223 12.7.6 综合内容 224 12.8 交错数组 224 12.8.1 声明交错数组 225 12.8.2 快捷实例化 225 12.8.3 实例化交错数组 226 12.8.4 交错数组中的子数组 227 12.9 比较矩形数组和交错数组 228 12.10 foreach语句 229 12.10.1 迭代变量是只读的 230 12.10.2 foreach语句和多维数组 231 12.11 数组协变 232 12.12 数组继承的有用成员 233 12.13 比较数组类型 236 第13章 委托 237 13.1 什么是委托 237 13.2 委托概述 239 13.3 声明委托类型 240 13.4 创建委托对象 241 13.5 给委托赋值 242 13.6 组合委托 243 13.7 为委托添加方法 243 13.8 从委托移除方法 244 13.9 调用委托 244 13.10 委托的示例 245 13.11 调用带返回值的委托 246 13.12 调用带引用参数的委托 247 13.13 匿名方法 248 13.13.1 使用匿名方法 249 13.13.2 匿名方法的语法 249 13.13.3 变量和参数的作用域 250 13.14 Lambda表达式 252 第14章 事件 255 14.1 发布者和订阅者 255 14.2 源代码组件概览 257 14.3 声明事件 257 14.4 订阅事件 258 14.5 触发事件 259 14.6 标准事件的用法 261 14.6.1 通过扩展EventArgs来传递数据 262 14.6.2 移除事件处理程序 264 14.7 事件访问器 265 第15章 接口 267 15.1 什么是接口 267 15.2 声明接口 272 15.3 实现接口 273 15.4 接口是引用类型 275 15.5 接口和as运算符 276 15.6 实现多个接口 276 15.7 实现具有重复成员的接口 277 15.8 多个接口的引用 279 15.9 派生成员作为实现 280 15.10 显式接口成员实现 281 15.11 接口可以继承接口 283 15.12 不同类实现一个接口的示例 284 第16章 转换 286 16.1 什么是转换 286 16.2 隐式转换 287 16.3 显式转换和强制转换 288 16.4 转换的类型 289 16.5 数字的转换 290 16.5.1 隐式数字转换 290 16.5.2 溢出检测上下文 291 16.5.3 显式数字转换 292 16.6 引用转换 295 16.6.1 隐式引用转换 296 16.6.2 显式引用转换 297 16.6.3 有效显式引用转换 298 16.7 装箱转换 299 16.7.1 装箱是创建副本 300 16.7.2 装箱转换 300 16.8 拆箱转换 301 16.9 用户自定义转换 302 16.9.1 用户自定义转换的约束 302 16.9.2 用户自定义转换的示例 302 16.9.3 评估用户自定义转换 304 16.9.4 多步用户自定义转换的示例 304 16.10 is运算符 305 16.11 as运算符 30 第17章 泛型 308 17.1 什么是泛型 308 17.2 C#中的泛型 310 17.3 泛型类 311 17.4 声明泛型类 312 17.5 创建构造类型 312 17.6 创建变量和实例 313 17.6.1 使用泛型的栈的示例 315 17.6.2 比较泛型和非泛型栈 316 17.7 类型参数的约束 317 17.7.1 Where子句 317 17.7.2 约束类型和次序 318 17.8 泛型方法 319 17.8.1 声明泛型方法 319 17.8.2 调用泛型方法 320 17.8.3 泛型方法的示例 321 17.9 扩展方法和泛型类 322 17.10 泛型结构 323 17.11 泛型委托 323 17.12 泛型接口 325 17.12.1 使用泛型接口的示例 326 17.12.2 泛型接口的实现必须唯一 327 17.13 协变 328 17.14 逆变 330 17.14.1 接口的协变和逆变 332 17.14.2 有关可变性的更多内容 333 第18章 枚举器和迭代器 335 18.1 枚举器和可枚举类型 335 18.2 IEnumerator接口 337 18.3 泛型枚举接口 341 18.4 迭代器 343 18.4.1 迭代器块 343 18.4.2 使用迭代器来创建枚举器 344 18.4.3 使用迭代器来创建可枚举类型 346 18.5 常见迭代器模式 347 18.6 产生多个可枚举类型 348 18.7 将迭代器作为属性 349 18.8 迭代器实质 350 第19章 LINQ 352 19.1 什么是LINQ 352 19.2 LINQ提供程序 353 19.3 方法语法和查询语法 355 19.4 查询变量 356 19.5 查询表达式的结构 357 19.5.1 from子句 358 19.5.2 join子句 359 19.5.3 什么是联结 360 19.5.4 查询主体中的from let where片段 362 19.5.5 orderby子句 365 19.5.6 select group子句 366 19.5.7 查询中的匿名类型 367 19.5.8 group子句 368 19.5.9 查询延续: into子句 369 19.6 标准查询运算符 370 19.6.1 标准查询运算符的签名 373 19.6.2 查询表达式和标准查询运算符 374 19.6.3 将委托作为参数 375 19.6.4 LINQ预定义的委托类型 376 19.6.5 使用委托参数的示例 377 19.6.6 使用Lambda表达式参数的示例 378 19.7 LINQtoXML 379 19.7.1 标记语言 379 19.7.2 XML基础 379 19.7.3 XML类 381 19.7.4 使用XML特性 387 19.7.5 节点的其他类型 389 19.7.6 使用LINQtoXML的LINQ查询 390 第20章 异步编程 393 20.1 什么是异步 393 20.2 async/await特性的结构 398 20.3 什么是异步方法 399 20.3.1 异步方法的控制流 402 20.3.2 await表达式 404 20.3.3 取消一个异步操作 407 20.3.4 异常处理和await表达式 409 20.3.5 在调用方法中同步

地等待任务 410 20.3.6 在异步方法中异步地等待任务 413 20.3.7 Task.Delay方法 415 20.4 在GUI程序中执行异步操作 416 20.5 使用异步Lambda表达式 420 20.6 完整的GUI程序 421 20.7 BackgroundWorker类 423 20.8 并行循环 428 20.9 其他异步编程模式 430 20.10 Begin Invoke和End Invoke 431 20.10.1 等待一直到结束模式 432 20.10.2 Async Result类 433 20.10.3 轮询模式 434 20.10.4 回调模式 435 20.11 计时器 438 第21章 命名空间和程序集 440 21.1 引用其他程序集 440 21.2 命名空间 444 21.2.1 命名空间名称 447 21.2.2 命名空间的补充 447 21.2.3 命名空间跨文件伸展 448 21.2.4 嵌套命名空间 449 21.3 using指令 450 21.3.1 using命名空间指令 450 21.3.2 using别名指令 451 21.4 程序集的结构 451 21.5 程序集标识符 453 21.6 强命名程序集 453 21.7 程序集的私有方式部署 455 21.8 共享程序集和GAC 455 21.8.1 把程序集安装到GAC 455 21.8.2 GAC内的并肩执行 456 21.9 配置文件 457 21.10 延迟签名 457 第22章 异常 459 22.1 什么是异常 459 22.2 try语句 460 22.3 异常类 461 22.4 catch子句 462 22.5 使用特定catch子句的示例 462 22.6 catch子句段 463 22.7 finally块 464 22.8 为异常寻找处理程序 465 22.9 更进一步搜索 466 22.9.1 一般法则 466 22.9.2 搜索调用栈的示例 467 22.10 抛出异常 469 22.11 不带异常对象的抛出 470 第23章 预处理指令 472 23.1 什么是预处理指令 472 23.2 基本规则 472 23.3 #define和#undef指令 473 23.4 条件编译 474 23.5 条件编译结构 475 23.6 诊断指令 476 23.7 行号指令 477 23.8 区域指令 478 23.9 #pragmawarning指令 479 第24章 反射和特性 480 24.1 元数据和反射 480 24.2 Type类 480 24.3 获取Type对象 482 24.4 什么是特性 484 24.5 应用特性 485 24.6 预定义的保留的特性 485 24.6.1 Obsolete特性 485 24.6.2 Conditional特性 486 24.6.3 调用者信息特性 488 24.6.4 Debugger StepThrough特性 488 24.6.5 其他预定义特性 489 24.7 有关应用特性的更多内容 490 24.7.1 多个特性 490 24.7.2 其他类型的目标 490 24.7.3 全局特性 491 24.8 自定义特性 491 24.8.1 声明自定义特性 492 24.8.2 使用特性的构造函数 492 24.8.3 指定构造函数 492 24.8.4 使用构造函数 493 24.8.5 构造函数中的位置参数和命名参数 493 24.8.6 限制特性的使用 494 24.8.7 自定义特性的最佳实践 495 24.9 访问特性 496 24.9.1 使用Is Defined方法 496 24.9.2 使用Get Custom Attributes方法 497 第25章 其他主题 499 25.1 概述 499 25.2 字符串 499 25.3 使用String Builder类 501 25.4 把字符串解析为数据值 502 25.5 关于可空类型的更多内容 503 25.5.1 为可空类型赋值 505 25.5.2 使用空接合运算符 505 25.5.3 使用可空用户自定义类型 506 25.6 Main方法 508 25.7 文档注释 509 25.7.1 插入文档注释 510 25.7.2 使用其他XML标签 510 25.8 嵌套类型 511 25.8.1 嵌套类的示例 512 25.8.2 可见性和嵌套类型 513 25.9 析构函数和dispose模式 514 25.9.1 标准dispose模式 515 25.9.2 比较构造函数和析构函数 517 25.10 和COM的互操作 518 索引 521

<<C#图解教程>>

章节摘录

版权页：插图：7.6使用基类的引用 派生类的实例由基类的实例加上派生类新增的成员组成。派生类的引用指向整个类对象，包括基类部分。

如果有一个派生类对象的引用，就可以获取该对象基类部分的引用（使用类型转换运算符把该引用转换为基类类型）。

类型转换运算符放置在对象引用的前面，由圆括号括起的是要被转换成的类名组成。

类型转换将在第16章阐述。

接下来的几节将阐述使用对象的基类部分的引用来访问对象。

我们从观察下面两行代码开始，它们声明了对象的引用。

图7.6阐明了代码，并展示了不同变量所看到的对象部分。

第一行声明并初始化了变量dezived，它包含一个MyDezivedClass类型对象的引用。

第二行声明了一个基类类型JMyBaseClass的变量，并把dezived中的引用转换为该类型，给出对象的基类部分的引用。

基类部分的引用被储存在变量mybc中，在赋值运算符的左边。

其他部分的引用不能“看到”派生类对象的其余部分，因为它通过基类类型的引用“看”这个对象。

```
HyDeriVedclass dezived=new MyDerivedclass ( ) ; // 创建一个对象 MyBaseClass mybc = ( MyBaseClass ) derived ; // 转换引用 下面的代码展示了两个类的声明和使用。
```

图7—7阐明了内存中的对象和引用。

Main创建了一个MyDezivedClass类型的对象，并把它的引用储存到变量derived中。

Main还创建了一个MyBaseClass类型的变量，并用它储存对象基类部分的引用。

当对每个引用调用Print方法时，调用的是该引用所能看到的方法的实现，并产生不同的输出字符串。

```
class MyBaseClass { public void Pzint ( ) { Console.WriteLine ( “ This is the base class. ” ) ; } 2.捕获变量的生命周期的扩展 要捕获方法还是委托的一部分，即使变量已经离开了作用域，捕获的外部变量也会一直有效。
```

例如，图13—15中的代码演示了被捕获变量的生命周期的扩展。

局部变量x在块中声明和初始化。

然后，委托mDel用匿名方法初始化，该匿名方法捕获了外部变量X。

块关闭时，x超出了作用域。

如果取消块关闭之后的WriteLine语句的注释，就会产生编译器错误。

因为它引用的x现在已经离开了作用域。

<<C#图解教程>>

媒体关注与评论

“ C#终极好书！

这是迄今为止最容易看懂的一本C#书，仅仅是把引用类型和值类型讲清楚的那一章就值这个价了，更不要说还有漂亮的图示。

写得非常好，非常深入。

要学习C#，从这本书开始，必须的。

”——Jon Hermiz如果这本书的厚度让你望而却步，那请放心。

这本书与其他书最大的区别，就是你可以看得很快，因为书中到处都是图示。

而且，作者的表达风格也很提神，他的话永远简洁明了、一语中的。

如果你想彻底学会C#，这本书一定要看！

保你不会失望。

”——Hugh

<<C#图解教程>>

编辑推荐

大家时间都很宝贵，三分钟就能理解的技术细节，干嘛要花三个小时？

！

C#图解教程(第4版)被誉为C#入门图书中的“神书”，是高效、深入掌握C#的不二之选！

“想通过学习C#为你的编程生涯打下坚实的基础？

这本书是我发自内心推荐给你的。

”为什么那么多C#学习者对《C#图解教程》有口皆碑，令这本书一版再版，长销不衰？

因为作者创造了一种全新的可视化叙述方式。

他在书中通过把大量的图、表和代码片段有机结合，把很多同类书需要好几页才能讲清楚的技术细节，形象、直观、一目了然地呈现在读者眼前。

不管你信不信，很多技术细节只有通过图示才能讲得这么清晰、透彻！

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>