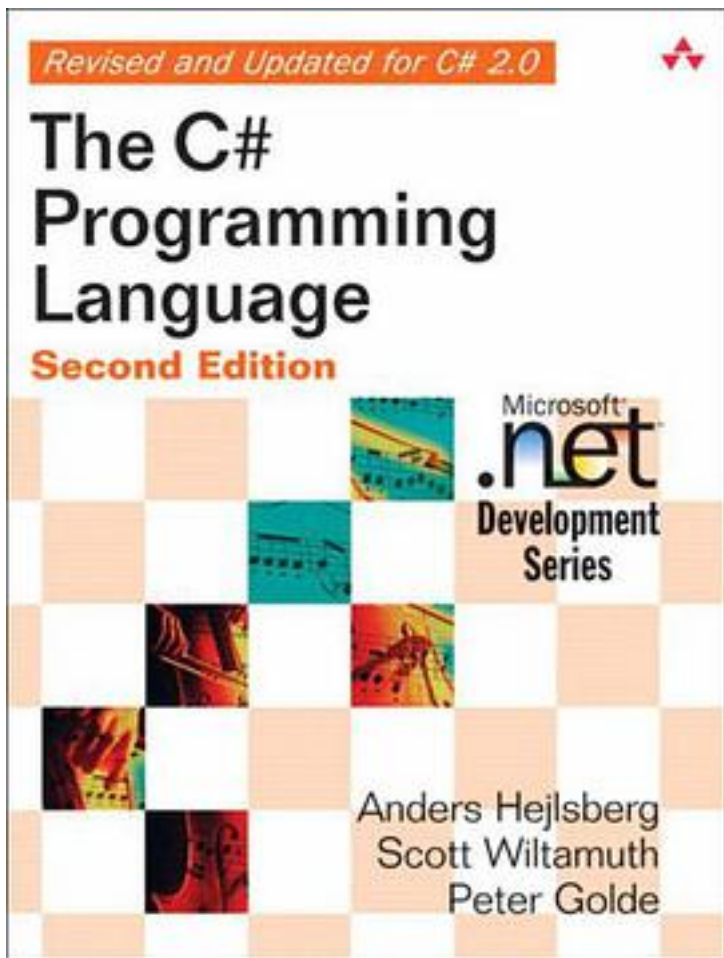


C# Programming Language, The (2nd Edition) (Microsoft .NET Development Series)



[C# Programming Language, The \(2nd Edition\) \(Microsoft .NET Development Series\) 下载链接1](#)

著者:Anders Hejlsberg

出版者:Addison-Wesley Professional

出版时间:2006-06-09

装帧:Hardcover

isbn:9780321334435

C# is a simple, modern, object-oriented, and type-safe programming language that combines the high productivity of rapid application development languages with the raw power of C and C++. Written by the language's architect, Anders Hejlsberg, and design team members, and now updated for C# 2.0, *The C# Programming Language, Second Edition*, is the definitive technical reference for C#. The book provides the complete specification of the language, along with descriptions, reference materials, and code samples from the C# design team. The first part of the book opens with an introduction to the language to bring readers quickly up-to-speed on the concepts of C#. Next follows a detailed and complete technical specification of the C# 1.0 language, as delivered in Visual Studio .NET 2002 and 2003. Topics covered include Lexical Structure, Types, Variables, Conversions, Expressions, Statements, Namespaces, Exceptions, Attributes, and Unsafe Code. The second part of the book describes the many new features of C# 2.0, including Generics, Anonymous Methods, Iterators, Partial Types, and Nullable Types. This second edition describes C# 2.0 as actually released in Visual Studio .NET 2005, with many additions and improvements over the design presented in the first edition. Reference tabs and an exhaustive index allow readers to easily navigate the text and quickly find the topics that interest them most. *The C# Programming Language, Second Edition*, is the definitive reference for programmers who want to acquire an in-depth knowledge of C#.

作者介绍:

译者序

序言

前言

作者简介

第1章 介绍 1

1.1 Hello, World 2

1.2 程序结构 3

1.3 类型和变量 5

1.4 表达式 9

1.5 语句 10

1.6 类和对象 14

1.6.1 成员 15

1.6.2 访问控制 16

1.6.3 类型参数 16

1.6.4 基类 17

1.6.5 字段 18

1.6.6 方法	19
1.6.7 其他函数成员	28
1.7 结构	35
1.8 数组	37
1.9 接口	39
1.10 枚举	40
1.11 委托	42
1.12 特性	43
第2章 词法结构	45
2.1 程序	45
2.2 文法	45
2.2.1 文法表示法	45
2.2.2 词法文法	46
2.2.3 语法规文法	46
2.3 词法分析	46
2.3.1 行终结符	47
2.3.2 注释	47
2.3.3 空白符	48
2.4 标记	49
2.4.1 Unicode字符转义序列	49
2.4.2 标识符	50
2.4.3 关键字	51
2.4.4 字面量	52
2.4.5 操作符和标点符号	57
2.5 预处理指令	58
2.5.1 条件编译符号	59
2.5.2 预处理表达式	59

2.5.3 声明指令 60

2.5.4 条件编译指令 61

2.5.5 诊断指令 63

2.5.6 区域指令 64

2.5.7 行指令 64

2.5.8 编译指示指令 65

第3章 基本概念 67

3.1 应用程序起始 67

3.2 应用程序终止 68

3.3 声明 68

3.4 成员 70

3.4.1 命名空间成员 71

3.4.2 结构成员 71

3.4.3 枚举成员 71

3.4.4 类成员 71

3.4.5 接口成员 72

3.4.6 数组成员 72

3.4.7 委托成员 72

3.5 成员访问 72

3.5.1 声明可访问性 72

3.5.2 可访问域 74

3.5.3 实例成员的保护访问 76

3.5.4 访问限制 78

3.6 签名和重载 79

3.7 作用域 80

3.7.1 名字隐藏 83

3.8 命名空间和类型名 85

3.8.1 完全限定名 87

3.9 自动化内存管理 88

3.10 执行顺序 92

第4章 类型 93

4.1 值类型 93

4.1.1 System.ValueType类型 94

4.1.2 默认构造函数 94

4.1.3 构造类型 95

4.1.4 简单类型 95

4.1.5 整数类型 96

4.1.6 浮点数类型 98

4.1.7 decimal类型 99

4.1.8 bool类型 100

4.1.9 枚举类型 101

4.1.10 可空值类型 101

4.2 引用类型 101

4.2.1 类类型 102

4.2.2 object类型 102

4.2.3 dynamic类型 103

4.2.4 string类型 103

4.2.5 接口类型 103

4.2.6 数组类型 103

4.2.7 委托类型 103

4.3 装箱和拆箱 103

4.3.1 装箱转换 104

4.3.2 拆箱转换 105

4.4 构造类型 107

4.4.1 类型实参 107

4.4.2 开放式和封闭式类型 108

4.4.3 绑定和未绑定类型 108

4.4.4 满足限制 108

4.5 类型参数 109

4.6 表达式树类型 110

4.7 dynamic类型 110

第5章 变量 112

5.1 变量类别 112

5.1.1 静态变量 113

5.1.2 实例变量 113

5.1.3 数组元素 113

5.1.4 值参数 113

5.1.5 引用参数 114

5.1.6 输出参数 114

5.1.7 局部变量 115

5.2 默认值 116

5.3 明确赋值 116

5.3.1 初始赋值的变量 117

5.3.2 未赋初值的变量 117

5.3.3 确定明确赋值的精确规则 117

5.4 变量引用 126

5.5 变量引用的原子性 127

第6章 转换 128

6.1 隐式转换 128

6.1.1 标识转换 129

6.1.2 隐式数字转换 129

- 6.1.3 隐式枚举转换 129
- 6.1.4 隐式可空值转换 130
- 6.1.5 null字面量转换 130
- 6.1.6 隐式引用转换 130
- 6.1.7 装箱转换 132
- 6.1.8 隐式动态转换 132
- 6.1.9 隐式常量表达式转换 133
- 6.1.10 带类型参数的隐式转换 133
- 6.1.11 自定义隐式转换 134
- 6.1.12 匿名函数转换和方法组转换 134
- 6.2 显式转换 134
 - 6.2.1 显式数字转换 134
 - 6.2.2 显式枚举转换 136
 - 6.2.3 显式可空值转换 136
 - 6.2.4 显式引用转换 136
 - 6.2.5 拆箱转换 138
 - 6.2.6 显式动态转换 138
 - 6.2.7 带类型参数的显式转换 138
 - 6.2.8 自定义显式转换 140
- 6.3 标准转换 140
 - 6.3.1 标准隐式转换 140
 - 6.3.2 标准显式转换 140
- 6.4 自定义转换 140
 - 6.4.1 允许的自定义转换 141
 - 6.4.2 提升转换操作符 141
 - 6.4.3 自定义转换的计算 141
 - 6.4.4 自定义隐式转换 142

6.4.5 自定义显式转换	143
6.5 匿名函数转换	144
6.5.1 匿名函数到委托类型转换的计算	145
6.5.2 匿名函数到表达式树类型转换的计算	146
6.5.3 实现举例	146
6.6 方法组转换	149
第7章 表达式	152
7.1 表达式分类	152
7.1.1 表达式的值	153
7.2 静态和动态绑定	153
7.2.1 绑定期	154
7.2.2 动态绑定	155
7.2.3 组成表达式的类型	155
7.3 操作符	156
7.3.1 操作符优先级和结合性	156
7.3.2 操作符重载	157
7.3.3 一元操作符重载决策	159
7.3.4 二元操作符重载决策	159
7.3.5 候选自定义操作符	159
7.3.6 数字提升	160
7.3.7 提升操作符	161
7.4 成员查找	162
7.4.1 基础类型	163
7.5 函数成员	164
7.5.1 参数列表	165
7.5.2 类型推导	168
7.5.3 重载决策	175

7.5.4 动态重载决策的编译期检查 179

7.5.5 函数成员调用 179

7.6 基础表达式 180

7.6.1 字面量 181

7.6.2 简单名字 181

7.6.3 括号表达式 183

7.6.4 成员访问 183

7.6.5 调用表达式 187

7.6.6 元素访问 193

7.6.7 this访问 195

7.6.8 base访问 196

7.6.9 后缀递增和递减操作符 196

7.6.10 new操作符 197

7.6.11 typeof操作符 207

7.6.12 checked和unchecked操作符 209

7.6.13 默认值表达式 211

7.6.14 匿名方法表达式 212

7.7 一元操作符 212

7.7.1 一元加号操作符 212

7.7.2 一元减号操作符 213

7.7.3 逻辑否操作符 213

7.7.4 按位求补操作符 213

7.7.5 前缀递增和递减操作符 214

7.7.6 转换表达式 215

7.8 算术操作符 216

7.8.1 乘法操作符 216

7.8.2 除法操作符 217

- 7.8.3 求余操作符 218
- 7.8.4 加法操作符 220
- 7.8.5 减法操作符 221
- 7.9 移位操作符 223
- 7.10 关系和类型测试操作符 224
 - 7.10.1 整数比较操作符 225
 - 7.10.2 浮点数比较操作符 226
 - 7.10.3 小数比较操作符 227
 - 7.10.4 布尔值相等操作符 227
 - 7.10.5 枚举比较操作符 227
 - 7.10.6 引用类型相等操作符 227
 - 7.10.7 字符串相等操作符 229
 - 7.10.8 委托相等操作符 229
 - 7.10.9 相等操作符和null 230
 - 7.10.10 is操作符 230
 - 7.10.11 as操作符 230
- 7.11 逻辑操作符 232
 - 7.11.1 整数逻辑操作符 233
 - 7.11.2 枚举逻辑操作符 233
 - 7.11.3 布尔逻辑操作符 233
 - 7.11.4 可空值布尔逻辑操作符 233
- 7.12 条件逻辑操作符 234
 - 7.12.1 布尔条件逻辑操作符 234
 - 7.12.2 用户自定义条件逻辑操作符 235
- 7.13 Null拼接操作符 235
- 7.14 条件操作符 236
- 7.15 匿名函数表达式 238

7.15.1 匿名函数签名 239

7.15.2 匿名函数主体 240

7.15.3 重载决策 240

7.15.4 匿名函数和动态绑定 241

7.15.5 外部变量 241

7.15.6 匿名函数表达式的计算 244

7.16 查询表达式 245

7.16.1 查询表达式里的歧义 246

7.16.2 查询表达式翻译 247

7.16.3 查询表达式模式 254

7.17 赋值操作符 256

7.17.1 简单赋值 256

7.17.2 组合赋值 259

7.17.3 事件赋值 260

7.18 表达式 260

7.19 常量表达式 260

7.20 布尔表达式 261

第8章 语句 263

8.1 终点和可及性 263

8.2 块 265

8.2.1 语句列表 265

8.3 空语句 266

8.4 标签语句 267

8.5 声明语句 268

8.5.1 局部变量声明 268

8.5.2 局部常量声明 271

8.6 表达式语句 271

8.7 选择语句 272

8.7.1 if语句 272

8.7.2 switch语句 273

8.8 迭代语句 276

8.8.1 while语句 277

8.8.2 do语句 277

8.8.3 for语句 278

8.8.4 foreach语句 279

8.9 跳转语句 283

8.9.1 break语句 284

8.9.2 continue语句 284

8.9.3 goto语句 285

8.9.4 return语句 286

8.9.5 throw语句 287

8.10 try语句 288

8.11 checked和unchecked语句 291

8.12 lock语句 292

8.13 using语句 293

8.14 yield语句 296

第9章 命名空间 299

9.1 编译单元 299

9.2 命名空间声明 300

9.3 Extern别名 301

9.4 using指令 301

9.4.1 using别名指令 302

9.4.2 using命名空间指令 305

9.5 命名空间成员 306

9.6 类型声明 307

9.7 命名空间别名限定符 307

9.7.1 别名的惟一性 308

第10章 类 310

10.1 类声明 310

10.1.1 类修饰符 310

10.1.2 partial修饰符 312

10.1.3 类型形参 313

10.1.4 类基础规范 313

10.1.5 类型形参限制 315

10.1.6 类主体 320

10.2 局部类型 320

10.2.1 特性 321

10.2.2 修饰符 321

10.2.3 类型形参和限制 321

10.2.4 基类 322

10.2.5 基础接口 322

10.2.6 成员 322

10.2.7 局部方法 323

10.2.8 名字绑定 326

10.3 类成员 327

10.3.1 实例类型 328

10.3.2 构造类型的成员 328

10.3.3 继承 330

10.3.4 new修饰符 331

10.3.5 访问修饰符 331

10.3.6 组成类型 331

10.3.7 静态成员和实例成员 331

10.3.8 嵌套类型 332

10.3.9 保留成员名 337

10.4 常量 338

10.5 字段 340

10.5.1 静态字段和实例字段 341

10.5.2 只读字段 342

10.5.3 易失字段 344

10.5.4 字段初始化 345

10.5.5 变量初始化语句 346

10.6 方法 348

10.6.1 方法形参 350

10.6.2 静态方法和实例方法 356

10.6.3 虚拟方法 356

10.6.4 重写方法 359

10.6.5 密封方法 361

10.6.6 抽象方法 362

10.6.7 外部方法 363

10.6.8 局部方法 364

10.6.9 扩展方法 364

10.6.10 方法主体 365

10.6.11 方法重载 366

10.7 属性 366

10.7.1 静态属性和实例属性 367

10.7.2 访问器 367

10.7.3 自动实现的属性 372

10.7.4 可访问性 373

10.7.5 虚拟、密封、重写和抽象访问器 375

10.8 事件 376

10.8.1 类似字段的事件 378

10.8.2 事件访问器 380

10.8.3 静态事件和实例事件 381

10.8.4 虚拟、密封、重写和抽象访问器 381

10.9 索引 381

10.9.1 索引重载 385

10.10 操作符 385

10.10.1 一元操作符 386

10.10.2 二元操作符 387

10.10.3 转换操作符 388

10.11 实例构造函数 390

10.11.1 构造函数初始化语句 391

10.11.2 实例变量初始化语句 392

10.11.3 构造函数的执行 392

10.11.4 默认构造函数 394

10.11.5 私有构造函数 395

10.11.6 可选的实例构造函数参数 395

10.12 静态构造函数 396

10.13 析构函数 398

10.14 迭代器 400

10.14.1 枚举器接口 400

10.14.2 可枚举接口 400

10.14.3 yield类型 400

10.14.4 计数对象 400

10.14.5 枚举对象 402

10.14.6 实现举例 403

第11章 结构 410

11.1 结构声明 410

11.1.1 结构修饰符 411

11.1.2 partial修饰符 411

11.1.3 结构接口 411

11.1.4 结构主体 411

11.2 结构成员 411

11.3 类和结构的区别 412

11.3.1 值语义 412

11.3.2 继承 413

11.3.3 赋值 413

11.3.4 默认值 414

11.3.5 装箱和拆箱 414

11.3.6 this的含义 416

11.3.7 字段初始化语句 417

11.3.8 构造函数 417

11.3.9 析构函数 418

11.3.10 静态构造函数 418

11.4 结构举例 419

11.4.1 数据库整数类型 419

11.4.2 数据库布尔类型 421

第12章 数组 424

12.1 数组类型 424

12.1.1 System.Array类型 425

12.1.2 数组和泛型IList接口 425

12.2 数组创建 426

12.3 数组元素访问 426

12.4 数组成员 426

12.5 数组协变 426

12.6 数组初始化语句 427

第13章 接口 430

13.1 接口声明 430

13.1.1 接口修饰符 430

13.1.2 partial修饰符 431

13.1.3 可变类型形参列表 431

13.1.4 基础接口 432

13.1.5 接口主体 434

13.2 接口成员 434

13.2.1 接口方法 435

13.2.2 接口属性 435

13.2.3 接口事件 436

13.2.4 接口索引 436

13.2.5 接口成员访问 436

13.3 完全限定接口成员名 438

13.4 接口实现 439

13.4.1 显式接口成员实现 440

13.4.2 实现接口的惟一性 442

13.4.3 泛型方法的实现 443

13.4.4 接口映射 444

13.4.5 接口实现继承 447

13.4.6 重新实现接口 448

13.4.7 抽象类和接口 450

第14章 枚举 451

14.1 枚举声明 451

14.2 枚举修饰符 452

14.3 枚举成员 452

14.4 System.Enum类型 454

14.5 枚举值和操作 454

第15章 委托 456

15.1 委托声明 456

15.2 委托兼容性 459

15.3 委托实例化 459

15.4 委托调用 460

第16章 异常 463

16.1 异常产生的原因 464

16.2 System.Exception类 464

16.3 异常是如何处理的 464

16.4 常见的异常类 465

第17章 特性 467

17.1 特性类 467

17.1.1 特性的用法 467

17.1.2 位置和命名参数 469

17.1.3 特性参数类型 470

17.2 特性规范 470

17.3 特性实例 475

17.3.1 特性的编译 475

17.3.2 在运行时获取一个特性实例 475

17.4 保留特性 476

17.4.1 AttributeUsage特性 476

17.4.2 Conditional特性 477

17.4.3 Obsolete特性	480
17.5 用于互操作的特性	481
17.5.1 与COM以及Win32组件互操作	481
17.5.2 与其他.NET语言互操作	482
第18章 不安全的代码	483
18.1 不安全的上下文	483
18.2 指针类型	485
18.3 固定变量和可移动变量	488
18.4 指针转换	488
18.4.1 指针数组	490
18.5 表达式里的指针	490
18.5.1 指针间接寻址	491
18.5.2 指针成员访问	491
18.5.3 指针元素访问	492
18.5.4 取地址操作符	493
18.5.5 指针递增和递减	494
18.5.6 指针算术	494
18.5.7 指针比较	495
18.5.8 sizeof操作符	495
18.6 fixed语句	496
18.7 定长缓冲区	500
18.7.1 定长缓冲区声明	500
18.7.2 表达式里的定长缓冲区	501
18.7.3 明确赋值检查	502
18.8 栈分配	502
18.9 动态内存分配	503
附录A 文档注释	506

附录B 文法 526

附录C 参考 560

目录:

[C# Programming Language, The \(2nd Edition\) \(Microsoft .NET Development Series\) 下载链接1](#)

标签

C

#编程语言

Programming

评论

很全面的C#书，包括了C# 的方方面面以及2.0新特性。不过不包括.net库的

[C# Programming Language, The \(2nd Edition\) \(Microsoft .NET Development Series\) 下载链接1](#)

书评

C#语言结合了快速应用开发语言的高效和C/C++语言的强大。现在C# 4.0又加入了动态绑定、命名和可选参数以及协变和逆变泛型等新特性，这些特性把C#语言提升到了一个新的层次上，使程序的编写更简洁，不再依赖于静态类型定义。这使得C#与动态编程语言（例如Python、Rudy和JavaS...

买了书的前辈，都怎么看的呀？我实在看得头痛呀！刚才有个同学说他在猎豹网校上学的，象看电影一样，可容易了。有空要去看一下了。否则我会累倒在这里了。

试读过几章，我还有一些C语言基础，也学过c#。但是，读起来很不顺。许多语法格式。感觉，像是枯燥的 语法规范。而不是讲解语言。估计想深入研究c#编译机理的更有用。这还短？ 这还短？ 这还短？ 这还短？ 这还短？ 这还短？ 这还短？ 这还短？ 这还短？ ...

[C# Programming Language, The \(2nd Edition\) \(Microsoft .NET Development Series\)_下载链接1](#)