

21世纪高等学校计算机类课程创新规划教材 · 微课版



500分钟  
视频讲解

# Python 程序设计 与算法基础教程 (第2版)

微课版

◎ 江红 余青松 编著

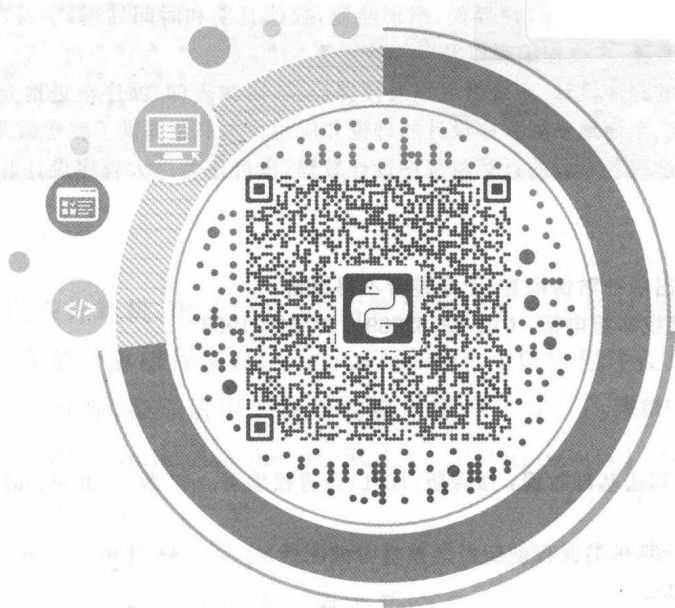
- ▷ 教材 + 练习册 + 上机指导 + 微课视频
- ▷ 基于Windows 10+Python 3.7
- ▷ 700多个实例, 431道复习题, 563个实践操作任务, 37个综合应用案例

《教学课件》 《教学大纲》 《电子教案》 《程序源码》 《期末试卷》 《习题答案》

清华大学出版社



21世纪高等学校计算机类课程创新规划教材 · 微课版



# Python 程序设计 与算法基础教程 (第2版)

微课版

© 江红 余青松 编著

清华大学出版社  
北京

## 内 容 简 介

本书集教材、练习册、上机指导于一体,基于 Windows 10 和 Python 3.7 构建 Python 开发平台,阐述 Python 语言的基础知识,以及使用 Python 语言的开发应用实例,具体内容包括 Python 概述,Python 语言基础,程序流程控制,常用内置数据类型,序列数据类型,输入和输出,错误和异常处理,函数、类和对象,模块和客户端,算法与数据结构基础,图形用户界面,图形绘制,数值日期和时间处理,字符串和文本处理,文件、数据库访问,网络和 Web 编程,多线程编程以及系统管理等。

本书编者结合多年的程序设计、系统开发以及授课经验,由浅入深、循序渐进地介绍 Python 程序设计语言,让读者能够较为系统、全面地掌握程序设计的理论和应用。本书还提供了教学微课视频。

本书可以作为高等学校各专业的计算机程序设计教程,也可作为广大程序设计开发者、爱好者的自学参考书。

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

Python 程序设计与算法基础教程:微课版/江红,余青松编著.—2版.—北京:清华大学出版社,2019  
(2019.8重印)

(21世纪高等学校计算机类课程创新规划教材·微课版)

ISBN 978-7-302-52483-0

I. ①P… II. ①江… ②余… III. ①软件工具—程序设计—高等学校—教材 IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2019)第 043701 号

策划编辑:魏江江

责任编辑:王冰飞

封面设计:刘 键

责任校对:李建庄

责任印制:杨 艳

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>,010-62795954

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:26.25

字 数:674千字

版 次:2017年7月第1版 2019年7月第2版

印 次:2019年8月第3次印刷

印 数:25001~28000

定 价:59.00元

产品编号:081892-01



# 前言

程序设计是大专院校计算机、电子信息、工商管理等相关专业的必修课程。Python 语言是一种解释型、面向对象的计算机程序设计语言,广泛用于计算机程序设计教学语言、系统管理脚本语言、科学计算等,特别适用于快速的应用程序开发。Python 编程语言广受开发者的喜爱,并被列入 LAMP(Linux、Apache、MySQL 以及 Python/Perl/PHP),已经成为最受欢迎的程序设计语言之一。

本书集教材、练习册、上机指导于一体,基于 Windows 10 和 Python 3.7 构建 Python 开发平台,通过大量的实例由浅入深、循序渐进地阐述 Python 语言的基础知识,以及使用 Python 语言的开发应用实例,具体内容包括 Python 概述,Python 语言基础,程序流程控制,常用内置数据类型,序列数据类型,输入和输出,错误和异常处理,函数、类和对象,模块和客户端,算法与数据结构基础,图形用户界面,图形绘制,数值日期和时间处理,字符串和文本处理,文件、数据库访问,网络和 Web 编程,多线程编程以及系统管理等。

本书是第 1 版的升级和完善。

在第 1 版的基础上,在每个章节中增加了“蒙特卡洛模拟:赌徒破产命运”“基于字典的通信录”“使用随机数估值圆周率”“去除列表中的重复项生成器函数”“文本统计”“基因预测”“字符串加密和解密”“病毒扫描”“遍历并输出文件目录结构”等实用小案例。

本书的每个章节末还增加了“网络爬虫案例”“百度音乐批量下载器”“使用 pandas 进行数据分析和处理”“猜单词游戏”“井字棋(Tic Tac Toe)游戏”“21 点扑克牌游戏”“简易图形用户界面计算器”“基于 turtle 的汉诺塔问题求解动画的设计和实现”“基于模块的库存管理系统”“基于数据库和 GUI 的教务管理系统”“文本相似度比较分析”“文本统计并行处理”“科学计算和数据分析”“使用嵌套循环实现图像处理算法”“NLTK 与自然语言处理”等大的实用案例研究。实用案例研究作为本书的电子资源,采用二维码的方式印在书上,作为开源的补充阅读和学习资源,并且随着 Python 程序的需求和演变将不断增补和更新。

教程还提供教学微课视频,方便学生反复观看和学习课程相关内容,扫描书中的二维码,可以在线观看视频讲解。

为了更好地帮助读者理解和掌握知识点及应用技能,本书提供了 700 多个大大小小的实例、431 道复习题(选择题、填空题和思考题)、563 个实践操作任务、37 个综合应用案例。本书

配套的教学课件、教学大纲、电子教案、期末试卷、习题答案可以通过扫描封底课件二维码下载。

本书由华东师范大学江红和余青松共同编写,衷心感谢清华大学出版社的编辑,敬佩他们的睿智和敬业。由于时间和编者学识有限,书中不足之处在所难免,敬请诸位同行、专家和读者指正。

编者



2019年5月



# 目 录







源码下载





<b>第 1 章 Python 概述</b>	<b>1</b>
1.1 初识 Python 语言	1
1.1.1 Python 语言简介	1
1.1.2 Python 语言的特点	1
1.1.3 Python 语言的应用范围	2
1.2 Python 语言版本和开发环境	2
1.2.1 Python 语言的版本	2
1.2.2 Python 语言的实现	2
1.2.3 Python 语言的集成开发环境	3
1.3 下载和安装 Python	3
1.3.1 下载 Python	3
1.3.2 安装 Python	4
1.3.3 安装和管理 Python 扩展包	4
1.4 使用 Python 解释器解释执行 Python 程序	6
1.4.1 运行 Python 解释器	6
1.4.2 运行 Python 集成开发环境	7
1.5 使用文本编辑器和命令行编写和执行 Python 源文件程序	8
1.5.1 编写输出“Hello, World!”的程序	8
1.5.2 输出“Hello, World!”程序的源代码分析	9
1.5.3 运行 Python 源代码程序	9
1.5.4 命令行参数	10
1.6 使用集成开发环境 IDLE 编写和执行 Python 源文件程序	11
1.6.1 使用 IDLE 编写程序	11
1.6.2 使用 IDLE 编辑程序	12
1.7 在线帮助和相关资源	12
1.7.1 Python 交互式帮助系统	12
1.7.2 Python 文档	14



1.7.3	Python 官网	15
1.7.4	Python 扩展库索引	15
1.8	复习题	16
1.9	上机实践	17
1.10	案例研究: 安装和使用其他 Python 环境 	17
<b>第2章</b>	<b>Python 语言基础 </b>	<b>18</b>
2.1	Python 程序概述	18
2.1.1	引例	18
2.1.2	Python 程序的构成	18
2.2	Python 对象和引用	19
2.2.1	Python 对象概述	19
2.2.2	使用字面量创建实例对象	19
2.2.3	使用类对象创建实例对象	20
2.2.4	数据类型	20
2.2.5	变量和对象的引用	20
2.2.6	Python 是动态类型语言	21
2.2.7	Python 是强类型语言	21
2.2.8	对象内存示意图	22
2.2.9	对象的值比较和引用判别	22
2.2.10	不可变对象和可变对象	23
2.3	标识符及其命名规则	23
2.3.1	标识符	23
2.3.2	保留关键字	24
2.3.3	Python 预定义标识符	24
2.3.4	命名规则	25
2.4	变量和赋值语句	25
2.4.1	变量的声明和赋值	25
2.4.2	链式赋值语句	25
2.4.3	复合赋值语句	26
2.4.4	删除变量	26
2.4.5	序列解包赋值	26
2.4.6	常量	27
2.5	表达式和运算符	27
2.5.1	表达式的组成	27
2.5.2	表达式的书写规则	28
2.5.3	运算符概述	28
2.5.4	Python 运算符	28
2.6	语句	29
2.6.1	Python 语句	29
2.6.2	Python 语句的书写规则	29


2.6.3	复合语句及其缩进书写规则 .....	30
2.6.4	注释语句 .....	30
2.6.5	空语句 .....	31
2.7	函数和模块 .....	31
2.7.1	函数的创建和调用 .....	31
2.7.2	内置函数 .....	32
2.7.3	模块函数 .....	32
2.7.4	函数 API .....	32
2.8	类和对象 .....	33
2.8.1	创建类对象 .....	33
2.8.2	实例对象的创建和调用 .....	33
2.9	模块和包 .....	33
2.10	复习题 .....	34
2.11	上机实践 .....	36
2.12	案例研究：使用 Pillow 库处理图像文件  .....	36
<b>第 3 章</b>	<b>程序流程控制 </b> .....	<b>38</b>
3.1	顺序结构 .....	38
3.2	选择结构 .....	38
3.2.1	分支结构的形式 .....	39
3.2.2	单分支结构 .....	39
3.2.3	双分支结构 .....	40
3.2.4	多分支结构 .....	40
3.2.5	if 语句的嵌套 .....	42
3.2.6	if 语句的典型示例代码 .....	43
3.2.7	选择结构综合举例 .....	43
3.3	循环结构 .....	44
3.3.1	可迭代对象 .....	44
3.3.2	range 对象 .....	45
3.3.3	for 循环 .....	45
3.3.4	while 循环 .....	45
3.3.5	循环的嵌套 .....	47
3.3.6	break 语句 .....	47
3.3.7	continue 语句 .....	48
3.3.8	死循环 .....	49
3.3.9	else 子句 .....	49
3.3.10	enumerate() 函数和循环 .....	50
3.3.11	zip() 函数和循环 .....	50
3.3.12	map() 函数和循环 .....	51
3.3.13	循环语句的典型示例代码 .....	51
3.3.14	循环结构综合举例 .....	52


3.4	复习题 .....	53
3.5	上机实践 .....	55
3.6	案例研究:使用嵌套循环实现图像处理算法  .....	58
<b>第4章</b>	<b>常用内置数据类型 </b> .....	<b>59</b>
4.1	Python 内置数据类型概述 .....	59
4.1.1	数值数据类型 .....	59
4.1.2	序列数据类型 .....	59
4.1.3	集合数据类型 .....	60
4.1.4	字典数据类型 .....	60
4.1.5	NoneType、NotImplementedType 和 EllipsisType .....	60
4.1.6	其他数据类型 .....	60
4.2	int 类型 .....	60
4.2.1	整型字面量 .....	61
4.2.2	int 对象 .....	61
4.2.3	int 对象的方法 .....	61
4.2.4	整数的运算 .....	62
4.3	float 类型 .....	62
4.3.1	浮点类型字面量 .....	63
4.3.2	float 对象 .....	63
4.3.3	float 对象的方法 .....	63
4.3.4	浮点数的运算 .....	64
4.4	complex 类型 .....	64
4.4.1	复数类型字面量 .....	64
4.4.2	complex 对象 .....	65
4.4.3	complex 对象的属性和方法 .....	65
4.4.4	复数的运算 .....	65
4.5	bool 类型 .....	66
4.5.1	布尔值字面量 .....	66
4.5.2	bool 对象 .....	66
4.5.3	逻辑运算符 .....	66
4.6	str 类型 .....	67
4.6.1	字符串字面量 .....	67
4.6.2	字符串编码 .....	67
4.6.3	转义字符 .....	68
4.6.4	str 对象 .....	68
4.6.5	str 对象的属性和方法 .....	69
4.6.6	字符串的运算 .....	69
4.6.7	对象转换为字符串 .....	69
4.6.8	字符串的格式化 .....	69
4.6.9	格式化字符串变量 .....	70



4.7	比较关系运算和条件表达式 .....	70
4.7.1	条件表达式 .....	70
4.7.2	关系和测试运算符 .....	71
4.8	算术运算符和位运算符 .....	72
4.8.1	算术运算符 .....	72
4.8.2	位运算符 .....	72
4.9	混合运算和数值类型转换 .....	73
4.9.1	隐式转换 .....	73
4.9.2	显式转换 .....	73
4.10	内置标准数学函数 .....	74
4.10.1	内置数学运算函数 .....	74
4.10.2	数制转换函数 .....	74
4.11	复习题 .....	74
4.12	上机实践 .....	77
4.13	案例研究：科学计算和数据分析  .....	81
<b>第5章</b>	<b>序列数据类型 </b> .....	<b>82</b>
5.1	Python 序列数据概述 .....	82
5.1.1	数组 .....	82
5.1.2	Python 内置的序列数据类型 .....	82
5.2	序列数据的基本操作 .....	83
5.2.1	序列的长度、最大值、最小值、求和 .....	83
5.2.2	序列的索引访问操作 .....	83
5.2.3	序列的切片操作 .....	84
5.2.4	序列的连接和重复操作 .....	85
5.2.5	序列的成员关系操作 .....	85
5.2.6	序列的比较运算操作 .....	86
5.2.7	序列的排序操作 .....	86
5.2.8	内置函数 all() 和 any() .....	86
5.2.9	序列的拆分 .....	87
5.3	元组 .....	87
5.3.1	使用元组字面量创建元组实例对象 .....	87
5.3.2	使用 tuple 对象创建元组实例对象 .....	88
5.3.3	元组的序列操作 .....	88
5.4	列表 .....	88
5.4.1	使用列表字面量创建列表实例对象 .....	88
5.4.2	使用 list 对象创建列表实例对象 .....	88
5.4.3	列表的序列操作 .....	89
5.4.4	list 对象的方法 .....	89
5.4.5	列表解析表达式 .....	90
5.5	字符串 .....	90
5.5.1	字符串的序列操作 .....	90



5.5.2	字符串编码 .....	90
5.5.3	字符串的格式化 .....	91
5.6	字节序列 .....	93
5.6.1	bytes 常量 .....	93
5.6.2	创建 bytes 对象 .....	94
5.6.3	创建 bytearray 对象 .....	94
5.6.4	bytes 和 bytearray 的序列操作 .....	95
5.6.5	字节编码和解码 .....	95
5.7	复习题 .....	95
5.8	上机实践 .....	98
5.9	案例研究:猜单词游戏  .....	98
<b>第 6 章</b>	<b>输入和输出 </b> .....	<b>100</b>
6.1	输入和输出概述 .....	100
6.2	命令行参数 .....	100
6.2.1	sys.argv 与命令行参数 .....	100
6.2.2	argparse 模块和命令行参数解析 .....	101
6.3	标准输入和标准输出函数 .....	102
6.3.1	输入和输出函数 .....	102
6.3.2	交互式用户输入 .....	103
6.3.3	运行时提示输入密码 .....	103
6.4	文件和文件对象 .....	104
6.4.1	文件对象和 open() 函数 .....	104
6.4.2	文件的打开、写入、读取和关闭 .....	104
6.4.3	with 语句和上下文管理协议 .....	105
6.5	标准输入、输出和错误流 .....	106
6.5.1	标准输入、输出和错误流文件对象 .....	106
6.5.2	读取任意长度的输入流 .....	106
6.5.3	标准输入、输出和错误流重定向 .....	107
6.6	重定向和管道 .....	107
6.6.1	重定向标准输出到一个文件 .....	108
6.6.2	重定向文件到标准输入 .....	108
6.6.3	管道 .....	109
6.6.4	过滤器 .....	110
6.7	复习题 .....	111
6.8	上机实践 .....	112
6.9	案例研究:21 点扑克牌游戏  .....	112
<b>第 7 章</b>	<b>错误和异常处理 </b> .....	<b>113</b>
7.1	程序的错误 .....	113
7.1.1	语法错误 .....	113
7.1.2	运行时错误 .....	113

7.1.3	逻辑错误	114
7.2	异常处理	115
7.2.1	异常处理概述	115
7.2.2	内置的异常类	115
7.2.3	引发异常	116
7.2.4	捕获处理异常机制概述	117
7.2.5	Python 虚拟机捕获处理异常	117
7.2.6	使用 try...except...else...finally 语句捕获处理异常	117
7.2.7	捕获异常的顺序	118
7.2.8	finally 块和发生异常后的处理	118
7.2.9	自定义异常类	119
7.3	断言处理	119
7.3.1	断言处理概述	119
7.3.2	assert 语句和 AssertionError 类	120
7.3.3	启用/禁用断言	120
7.4	程序的基本调试方法	121
7.4.1	语法错误的调试	121
7.4.2	运行时错误的调试	121
7.4.3	逻辑错误的调试	121
7.5	使用 logging 模块输入日志	122
7.5.1	logging 模块概述	122
7.5.2	logging 的配置和使用	123
7.6	复习题	125
7.7	上机实践	126
7.8	案例研究：使用调试器调试 Python 程序 	127
<b>第 8 章</b>	<b>函数和函数式编程 </b>	<b>128</b>
8.1	函数概述	128
8.1.1	函数的基本概念	128
8.1.2	函数的功能	128
8.1.3	Python 中函数的分类	128
8.2	函数的声明和调用	129
8.2.1	函数对象的创建	129
8.2.2	函数的调用	129
8.2.3	函数的副作用	130
8.3	参数的传递	131
8.3.1	形式参数和实际参数	131
8.3.2	形式参数变量和对象引用传递	132
8.3.3	传递不可变对象的引用	132
8.3.4	传递可变对象的引用	132
8.3.5	可选参数	133

8.3.6	位置参数和命名参数	133
8.3.7	可变参数	134
8.3.8	强制命名参数	135
8.3.9	参数类型检查	135
8.4	函数的返回值	136
8.4.1	return 语句和函数返回值	136
8.4.2	多条 return 语句	136
8.4.3	返回多个值	137
8.5	变量的作用域	137
8.5.1	全局变量	137
8.5.2	局部变量	138
8.5.3	全局语句 global	138
8.5.4	非局部语句 nonlocal	139
8.5.5	类成员变量	140
8.5.6	输出局部变量和全局变量	140
8.6	递归函数	140
8.6.1	递归函数的定义	140
8.6.2	递归函数的原理	141
8.6.3	编写递归函数时需要注意的问题	142
8.6.4	递归函数的应用:最大公约数	142
8.6.5	递归函数的应用:汉诺塔	143
8.7	内置函数的使用	144
8.7.1	内置函数一览	144
8.7.2	eval()函数	144
8.7.3	exec()函数	145
8.7.4	compile()函数	145
8.8	Python 函数式编程基础	145
8.8.1	作为对象的函数	145
8.8.2	高阶函数	145
8.8.3	map()函数	146
8.8.4	filter()函数	146
8.8.5	Lambda 表达式和匿名函数	146
8.8.6	operator 模块和操作符函数	147
8.8.7	functools.reduce()函数	148
8.8.8	偏函数	148
8.8.9	sorted()函数	148
8.8.10	函数装饰器	149
8.9	复习题	150
8.10	上机实践	152
8.11	案例研究:井字棋游戏 	152

第 9 章 面向对象的程序设计 	153
9.1 面向对象概念	153
9.1.1 对象的定义	153
9.1.2 封装	153
9.1.3 继承	153
9.1.4 多态性	153
9.2 类对象和实例对象	154
9.2.1 类对象	154
9.2.2 实例对象	154
9.3 属性	155
9.3.1 实例对象属性	155
9.3.2 类对象属性	156
9.3.3 私有属性和公有属性	156
9.3.4 @property 装饰器	157
9.3.5 特殊属性	158
9.3.6 自定义属性	159
9.4 方法	159
9.4.1 对象实例方法	159
9.4.2 静态方法	160
9.4.3 类方法	161
9.4.4 __init__()方法和__new__()方法	162
9.4.5 __del__()方法	162
9.4.6 私有方法与公有方法	163
9.4.7 方法的重载	164
9.5 继承	165
9.5.1 派生类	165
9.5.2 查看继承的层次关系	165
9.5.3 类成员的继承和重写	166
9.6 对象的特殊方法	166
9.6.1 对象的特殊方法概述	166
9.6.2 运算符重载与对象的特殊方法	167
9.6.3 @functools.total_ordering 装饰器	169
9.6.4 __call__()方法和可调用对象	169
9.7 对象的引用、浅拷贝和深拷贝	170
9.7.1 对象的引用	170
9.7.2 对象的浅拷贝	170
9.7.3 对象的深拷贝	170
9.8 可迭代对象：迭代器和生成器	171
9.8.1 可迭代对象	171
9.8.2 迭代器	171

9.8.3	迭代器协议	171
9.8.4	可迭代对象的迭代: iter()函数和 next()函数	172
9.8.5	可迭代对象的迭代: for 语句	172
9.8.6	自定义可迭代对象和迭代器	172
9.8.7	生成器函数	173
9.8.8	反向迭代: reversed 迭代器	174
9.8.9	生成器表达式	174
9.8.10	range 可迭代对象	175
9.8.11	map 迭代器和 itertools.starmap 迭代器	175
9.8.12	filter 迭代器和 itertools.filterfalse 迭代器	175
9.8.13	zip 迭代器和 itertools.zip_longest 迭代器	176
9.8.14	enumerate 迭代器	176
9.8.15	无穷序列迭代器 itertools.count、cycle 和 repeat	177
9.8.16	累计迭代器 itertools.accumulate	177
9.8.17	级联迭代器 itertools.chain	177
9.8.18	选择压缩迭代器 itertools.compress	178
9.8.19	截取迭代器 itertools.dropwhile 和 takewhile	178
9.8.20	切片迭代器 itertools.islice	178
9.8.21	分组迭代器 itertools.groupby	178
9.8.22	返回多个迭代器 itertools.tee	179
9.8.23	组合迭代器 itertools.combinations 和 combinations_with_replacement	179
9.8.24	排列迭代器 itertools.permutations	179
9.8.25	笛卡儿积迭代器 itertools.product	180
9.9	自定义类应用举例	180
9.9.1	Color 类	180
9.9.2	Histogram 类	181
9.10	复习题	183
9.11	上机实践	184
9.12	案例研究: 文本相似度比较分析 	184
<b>第 10 章</b>	<b>模块和客户端 </b>	<b>185</b>
10.1	模块化程序设计的概念	185
10.1.1	模块化程序设计	185
10.1.2	模块的 API	185
10.1.3	模块的实现	186
10.1.4	模块的客户端	187
10.1.5	模块化程序设计的优越性	187
10.2	模块的设计和实现	187
10.2.1	模块设计的一般原则	187
10.2.2	API 设计	188
10.2.3	创建模块	188

10.2.4	模块的私有函数 .....	189
10.2.5	模块的测试代码 .....	189
10.2.6	编写模块文档字符串 .....	190
10.2.7	按字节编译的 .pyc 文件 .....	191
10.3	模块的导入和使用 .....	191
10.3.1	导入模块和使用模块 .....	191
10.3.2	导入模块中的成员 .....	191
10.3.3	重新加载模块 .....	192
10.3.4	动态导入模块 .....	192
10.4	包 .....	192
10.4.1	包的概念 .....	192
10.4.2	创建包 .....	193
10.4.3	包的导入和使用 .....	193
10.5	模块的导入顺序 .....	194
10.5.1	导入模块时的搜索顺序 .....	194
10.5.2	模块搜索路径 .....	195
10.5.3	dir()函数 .....	195
10.6	名称空间与名称查找顺序 .....	196
10.6.1	名称空间概述 .....	196
10.6.2	名称查找顺序 .....	196
10.6.3	顶层模块和__name__变量 .....	196
10.6.4	Python 解释器 .....	197
10.6.5	全局名称空间 .....	197
10.6.6	局部名称空间 .....	198
10.6.7	类和对象名称空间 .....	199
10.7	复习题 .....	199
10.8	上机实践 .....	199
10.9	案例研究：基于模块的库存管理系统  .....	200
<b>第 11 章</b>	<b>算法与数据结构基础 </b> .....	<b>201</b>
11.1	算法及其性能分析 .....	201
11.1.1	算法概述 .....	201
11.1.2	算法的时间复杂度分析 .....	201
11.1.3	增长量级 .....	202
11.1.4	算法的空间复杂度分析 .....	203
11.2	查找算法 .....	203
11.2.1	顺序查找法 .....	203
11.2.2	二分查找法 .....	204
11.2.3	Python 语言提供的查找算法 .....	205
11.3	排序算法 .....	206
11.3.1	冒泡排序法 .....	206

11.3.2	选择排序法	207
11.3.3	插入排序法	208
11.3.4	归并排序法	209
11.3.5	快速排序法	210
11.3.6	Python 语言提供的排序算法	211
11.4	常用数据结构	211
11.4.1	数据结构概述	211
11.4.2	常用数据结构概述	212
11.4.3	Python 中的 collections 模块	212
11.5	数组	212
11.5.1	列表和数组	212
11.5.2	array.array 对象和数组	213
11.6	栈和队列	214
11.6.1	栈的实现:使用列表	214
11.6.2	deque 对象	214
11.6.3	deque 作为栈	215
11.6.4	deque 作为队列	215
11.7	集合	216
11.7.1	集合的定义	216
11.7.2	集合解析表达式	216
11.7.3	判断集合元素是否存在	217
11.7.4	集合的运算:并集、交集、差集和对称差集	217
11.7.5	集合的比较运算:相等、子集和超集	218
11.7.6	集合的长度、最大值、最小值、元素和	218
11.7.7	可变集合的方法	218
11.8	字典	219
11.8.1	对象的哈希值	219
11.8.2	字典的定义	219
11.8.3	字典的访问操作	220
11.8.4	字典的视图对象	220
11.8.5	字典的遍历	220
11.8.6	字典解析表达式	221
11.8.7	判断字典键是否存在	221
11.8.8	字典对象的长度和比较	221
11.8.9	字典对象的方法	222
11.8.10	defaultdict 对象	222
11.8.11	OrderedDict 对象	223
11.8.12	ChainMap 对象	223
11.8.13	Counter 对象	224
11.9	collections 模块的其他数据结构	225
11.9.1	namedtuple 对象	225