# 第1章 开启Python学习之旅

1. **填空题**
2. Python是一种面向 语言。
3. Python编写的程序可以在任何平台中执行，这体现了Python的 特点。
4. **判断题**
5. Python具有丰富的第三方库。（ ）
6. Python2中的异常与Python3中的异常使用方式相同。（ ）
7. PyCharm是一个完全免费的IDE工具。（ ）
8. **选择题**
9. 下列选项中，不属于Python特点的是（ ）。
10. 简单易学
11. 免费开源
12. 面向对象
13. 编译型语言
14. 下列关于Python2与Python3的说法中，错误的是（ ）。
15. Python3默认使用UTF-8编码
16. Python2与Python3中的print语句的格式没有变化
17. Python2默认使用ASCII编码
18. Python2与Python3中运算符“//”使用方式一致
19. 下列关于Python命名规范的说法中，错误的是（ ）。
20. 模块名、包名应简短且全为小写
21. 类名首字母一般使用大写
22. 常量通常使用全大写字母命名
23. 函数名中不可使用下画线
24. 下列选项中，（ ）是不符合规范的变量名。
25. \_text
26. 2cd
27. ITCAST
28. hei\_ma
29. 下列关于input()函数与print()函数的说法中，错误的是（ ）。
30. input()函数可以接收由键盘输入的数据
31. input()函数会返回一个字符串类型数据
32. print()函数可以输出任何类型的数据
33. print()函数输出的数据不支持换行操作
34. **简答题**
35. 请简述Python的特点。
36. 请简述Python2与Python3中的区别。

# 第2章 数字类型与字符串

1. **填空题**
2. Python的数字类型包含整型、浮点型、 、 。
3. 布尔类型是一种特殊的 。
4. Python中的复数是由 和 组成。
5. **判断题**
6. Python中的整型可以使用二进制、八进制、十进制、十六进制表示。（ ）
7. 浮点型不可与复数类型的数据进行计算。（ ）
8. 使用切片操作字符串，切片的步长只能是正整数。（ ）
9. 在Python中的运算符"<>"用于判断两个操作数是否相等。（ ）
10. **选择题**
11. 下列函数中，可以将数值类型转换为字符串的是（ ）。
12. complex()
13. int()
14. float()
15. str()
16. 下列关于Python字符串的说法中，错误的是（ ）。
17. 字符串是用来表示文本的数据类型
18. Python中可以使用单引号、双引号、三引号定义字符串
19. 单引号定义的字符串不能够包含双引号字符
20. 使用三引号定义的字符串可以包含换行符
21. 已知变量name="张昊"、age=18，下列字符串格式化输出，错误的是（ ）。
22. print ('我叫%s，今年我%d岁了' % (age, name))
23. print ('我叫%s，今年我%d岁了' % (name,age))
24. print('我叫{}，今年我{}岁了'.format(name, age))
25. print(f'我叫{name}，今年我{age}岁了')
26. 下列关于字符串操作的说法中，正确的是（ ）。（多选）
27. 字符串支持加减乘除操作
28. 字符串可以使用“+”符号进行拼接
29. 字符串可以使用split()方法替换子串
30. 字符串可以使用strip()方法去除两侧多余空格
31. 已知a=3，b=5，下列表达式的计算结果错误的是（ ）。
32. a+=b的值为8
33. a<<b的值为96
34. a and b的值为5
35. a//b的值为0.6
36. **编程题**
37. 请使用print()函数输出I'm from China.。
38. 已知变量string = ' python是一种解释型语言 '，请将字符串string两侧空格去除，并将小写字母p替换为大写字母P。
39. 请将字符串'tsacti'逆序输出。

# 第3章 流程控制语句

1. **填空题**
2. Python中的循环语句有 循环和 循环。
3. Python中使用关键字 表示条件语句。
4. 只有if条件表达式为 时才会执行满足条件的语句。
5. **判断题**
6. Python中break和continue语句可以单独使用。（ ）
7. if…else语句可以处理多个分支条件。（ ）
8. for循环嵌套可理解为for循环中包含for循环语句。（ ）
9. if语句不支持嵌套使用。（ ）
10. **选择题**
11. 请阅读下面的程序：

a = 1

b = 2

c = 3

if b < c:

c -= a

a += b

b \*= a

print(a, b, c)

运行程序，程序的输出结果为（ ）。

1. 1，2，3
2. 3，6，2
3. 2，6，3
4. 3，2，1
5. 请阅读下面的程序：

x = 0

for x in range(5):

x += 1

if x == 3:

break

print(x)

运行程序，程序的输出结果为（ ）。

1. 1 2
2. 1 2 3
3. 1 2 3 4 5
4. 0
5. 请阅读下面的程序：

i = 3

j = 5

while True:

if i < 5:

i += i

print(i)

break

elif j < 1:

j -= j

print(j)

运行程序，程序的输出结果为（ ）。

1. 8
2. 2
3. 6
4. 0
5. 下列关于for循环的说法中，描述正确的是（ ）。
6. for循环可以遍历可迭代对象
7. for循环不能使用循环嵌套
8. for循环不可以与if语句一起使用
9. for循环能控制循环执行的次数
10. 下列语句中，可以跳出循环体的是（ ）。
11. continue
12. break
13. if
14. while
15. **编程题**
16. 使用for循环输出1+2+3+…+100的结果。
17. 使用while循环输出100以内的偶数。
18. 使用for循环或while循环输出100以内的素数。

# 第4章 列表与元组

1. **填空题**
2. 使用内置的 函数可创建一个列表。
3. Python中列表的元素可通过索引或 两种方式去访问。
4. 使用内置的 函数可创建一个元组。
5. **判断题**
6. 列表只能存储同一类型的数据。（ ）
7. 元组支持增加、删除和修改元素的操作。（ ）
8. 列表的索引是从1开始。（ ）
9. 如果元组中只有1个元素，那么该元素的后面必须有逗号。（ ）
10. **选择题**
11. 下列方法中，可以对列表元素排序的是（ ）。
12. sort()
13. reverse()
14. max()
15. list()
16. 阅读下面程序：

li\_one = [2, 1, 5, 6]

print(sorted(li\_one[:2]))

运行程序，输出结果正确的是（ ）。

1. [1 ,2]
2. [2 ,1]
3. [1 ,2 ,5 ,6]
4. [6 ,5 ,2 ,1]
5. 下列方法中，默认删除列表最后一个元素的是（ ）。
6. del
7. remove()
8. pop()
9. extend()
10. 阅读下面的程序：

li\_one = ['p', 'c', 'q', 'h']

li\_two = ['o']

li\_one.extend(li\_two)

li\_one.insert(2, 'n')

print(li\_one)

运行程序，输出结果正确的是（ ）。

1. ['p', 'c', 'n', 'q', 'h', 'o']
2. ['p', 'c', 'h', 'q', 'n', 'o']
3. ['o', 'p', 'c', 'n', 'q', 'h']
4. ['o', 'p', 'n', 'c', 'q', 'h']
5. 下列创建元组的语句中，正确的是（ ）。
6. tu\_one = tuple('1', '2')
7. tu\_two = ('q')
8. tu\_three = ('on',)
9. tu\_four = (3,5)
10. **编程题**
11. 已知列表li\_num1 = [4, 5, 2, 7]和li\_num2 = [3, 6]，请将两个列表合并为一个列表，并将合并后列表中的元素按照从大到小的排序排列。
12. 已知元组tu\_num1 = ('p', 'y', 't', ['o', 'n'])，请向元组的最后一个列表中添加新元素“h”。

# 第5章 字典和集合

1. **填空题**
2. 字典元素由 和 组成。
3. 字典中的键具有 性。
4. 通过Python的内置方法 可以查看字典键的集合。
5. 调用items()方法可以查看字典中的所有 。
6. Python中可变集合和不可变集合的共同特点是 和 。
7. **判断题**
8. 字典中的键是唯一的。（ ）
9. 集合中的元素是无序的。（ ）
10. 字典中的元素可通过索引方式访问。（ ）
11. 集合中元素可以重复。（ ）
12. **选择题**
13. 下列方法中，可以获取字典中所有键的是（ ）。
14. keys()
15. value()
16. list()
17. values()
18. 阅读下面程序：

lan\_info = {'01': 'Python', '02': 'Java', '03': 'PHP'}

lan\_info.update({'03': 'C++'})

print(lan\_info)

运行程序，输出结果是（ ）。

1. {'01': 'Python', '02': 'Java', '03': 'PHP'}
2. {'01': 'Python', '02': 'Java', '03': 'C++'}
3. {'03': 'C++','01': 'Python', '02': 'Java'}
4. {'01': 'Python', '02': 'Java'}
5. 下列方法中，不能删除字典中元素的是（ ）。
6. clear()
7. remove()
8. pop()
9. popitem()
10. 阅读下面程序：

set\_01 = {'a', 'c', 'b', 'a'}

set\_01.add('d')

print(len(set\_01))

运行程序，以下输出结果正确的是（ ）。

1. 5
2. 3
3. 4
4. 2
5. 下列语句中，可以正确创建字典的是（ ）。
6. test\_one = ()
7. test\_two = {'a': 'A'}
8. test\_three = dict('a')
9. test\_four = dict{'a': 'A'}
10. **编程题**
11. 已知字符串str= 'skdaskerkjsalkj'，请统计该字符串中各字母出现的次数。
12. 已知列表li\_one = [1,2,1,2,3,5,4,3,5,7,4,7,8]，编写程序实现删除列表li\_one中重复数据的功能。

# 第6章 函数

1. **填空题**
2. Python中使用关键字 声明一个函数。
3. 匿名函数使用关键字 声明。
4. 在函数内部对全局变量进行修改，需要先使用 关键字声明。
5. **判断题**
6. 函数可以提高代码的复用性。（ ）
7. 全局变量在所有的函数中都可以访问。（ ）
8. 函数的位置参数有严格的位置关系。（ ）
9. 函数中的默认参数不能传递实际参数。（ ）
10. 函数执行结束后，其内部的局部变量会被回收。（ ）
11. **选择题**
12. 下列关于函数参数的说法中，错误的是（ ）。
13. 若无法确定需要传入函数的参数个数，可以为函数设置不定长参数
14. 当使用关键字参数传递实参时，需要为实参关联形参
15. 定义函数时可以为参数设置默认值
16. 不定长参数\*args可传递不定数量的关联形参名的实参
17. 下列关于Python函数的说法中，错误的是（ ）。
18. 递归函数就是在函数体中调用了自身的函数
19. 匿名函数没有函数名
20. 匿名函数与使用关键字def定义的函数没有区别
21. 匿名函数中可以使用if语句
22. 阅读下面程序：

num\_one = 12

def sum(num\_two):

global num\_one

num\_one = 90

return num\_one + num\_two

print(sum(10))

运行代码，输出结果是（ ）。

1. 102
2. 100
3. 22
4. 12
5. 阅读下面程序：

def many\_param(num\_one, num\_two, \*args):

print(args)

many\_param(11, 22, 33, 44, 55)

运行代码，输出结果是（ ）。

1. (11,22,33)
2. (22,33,44)
3. (33,44,55)
4. (11,22)
5. 阅读下面程序：

def fact(num):

if num == 1:

return 1

else:

return num + fact(num - 1)

print(fact(5))

运行代码，输出结果是（ ）。

1. 21
2. 15
3. 3
4. 1
5. **简答题**
6. 简述匿名函数的特点。
7. 简述位置参数、关键字参数、不定长参数的使用方法。
8. **编程题**
9. 编写函数，输出1~100中偶数之和。
10. 编写函数，计算20×19×18×…×3的结果。

# 第7章 面向对象程序设计

1. **填空题**
2. Python中使用关键字 声明一个类。
3. 在\_\_init\_\_()方法中第一个参数永远是 。
4. 子类中使用 函数可以调用父类中的方法。
5. Python中通过在属性名前添加 方式设置私有属性。
6. **判断题**
7. 一个类只能创建一个实例化对象。（ ）
8. 构造方法会在创建对象的时候自动调用。（ ）
9. 类方法可以使用类名进行访问。（ ）
10. 对象的引用计数器的值为0时会调用析构方法。（ ）
11. **选择题**
12. 下列关于类的说法，错误的是（ ）。
13. 在类中可以定义私有方法和属性
14. 类方法的第一个参数是cls
15. 实例方法的第一个参数是self
16. 类的实例无法访问类属性
17. 下列关于继承的说法中，错误的是（ ）。
18. Python不支持多继承
19. 如果一个类有多个父类，该类会继承这些父类的成员
20. 子类会自动拥有父类的属性和方法
21. 私有属性和私有方法是不能被继承的
22. 下列方法中，用于初始化属性的方法是（ ）。
23. \_\_del\_\_()
24. \_\_init\_\_()
25. \_\_init()
26. \_\_add\_\_()
27. 阅读下面程序：

class Test:

count = 21

def print\_num(self):

count = 20

self.count+=20

print(count)

test= Test()

test.print\_num()

运行程序，输出结果是（ ）。

1. 20
2. 40
3. 21
4. 41
5. 阅读下面程序：

class Init:

def \_\_init\_\_(self, addr, tel):

self.\_\_addr = addr

self.tel = tel

def show\_info(self):

print(f"地址：{self.\_\_addr}")

print(f"手机号：{self.tel}")

init = Init('北京', '12345')

init.show\_info()

运行程序，输出结果是（ ）。

1. 程序无法运行
2. 手机号：12345
3. 地址：北京

手机号：12345

1. 地址：北京
2. **简答题**
3. 简述构造方法与析构方法的特点。
4. 简述类方法与静态方法的区别。
5. 简述Python中的继承机制。
6. **编程题**

设计一个Circle（圆）类，该类中包括属性radius（半径），还包括\_\_init\_\_()、get\_perimeter()（求周长）和get\_area()（求面积）等方法。设计完成后，创建Circle类的对象并测试求周长和面积的功能。

# 第8章 模块

1. **填空题**
2. Python中模块分为内置模块、 和 。
3. 通过 和 可导入模块。
4. Python中的包是一个包含 目录。
5. **判断题**
6. 使用第三方模块时需要提前安装。（ ）
7. 一个py文件就是一个模块（ ）
8. 包结构中的\_\_init\_\_.py文件不能为空。（ ）
9. os模块提供系统级别的操作。（ ）
10. random模块中random()函数只能生成随机整数。（ ）
11. **选择题**
12. 下列关于Python中模块的说法，正确的是（ ）。
13. 程序中只能使用Python内置的标准模块
14. 只有标准模块才支持import导入
15. 使用import语句只能导入一个模块
16. 只有导入模块后，才可以使用模块中的变量、方法和类
17. 下列关于标准模块的说法中，错误的是（ ）。
18. 标准模块无须导入就可以使用
19. random模块属于标准模块
20. 标准模块可通过import进行导入
21. 标准模块也是一个.py文件
22. 下列导入模块的方式中，错误的是（ ）。
23. import random
24. from random import random
25. from random import \*
26. from random
27. 下列选项中，能够随机生成指定范围的整数的是（ ）。
28. random.random()
29. random.randint()
30. random.choice()
31. random.uniform()
32. 下列关于包的说法中，错误的是（ ）：
33. 包可以使用import语句导入
34. 包中必须含有\_\_init\_\_.py文件
35. 功能相近的模块可以放在同一包中
36. 包不能使用from..import…方式导入
37. **简答题**
38. 简述\_\_init\_\_.py文件的作用。
39. 简述\_\_name\_\_属性的用法。

# 第9章 文件与文件路径操作

1. **填空题**
2. 使用 方法可以关闭一个文件对象。
3. os.path模块中使用 函数拼接路径。
4. os模块中使用 函数可以获取文件列表。
5. **判断题**
6. 文件打开后不需要关闭。（ ）
7. 文件默认访问方式为可读。（ ）
8. 使用a+模式打开文件，文件不存在则会创建一个新文件。（ ）
9. read()方法可以设置读取的字符长度。（ ）
10. readlines()方法可以读取文件中的所有内容。（ ）
11. **选择题**
12. 下列关于文件读取的说法，错误的是（ ）。
13. read()方法可以一次读取文件中所有的内容
14. readline()方法一次只能读取一行内容
15. readlines()以元组形式返回读取的数据
16. readlines()一次可以读取文件中所有内容
17. 下列关于文件写入的说法，正确的是（ ）。
18. 如果向一个已有文件写数据，在写入之前会先清空文件数据
19. 每执行一次write()方法，写入的内容都会追加到文件末尾
20. writelines()方法用于向文件写入多行数据
21. 文件写入时不能使用“r”模式
22. 下列关于文件操作的说法，错误的是（ ）。
23. os模块中的mkdir()函数可创建目录
24. shutil模块中rmtree()函数可删除目录
25. os模块中的getcwd()函数获取的是相对路径
26. rename()函数可修改文件名
27. 下列选项中，用于获取当前读/写位置的是（ ）。
28. open()
29. close()
30. tell()
31. seek()
32. 已知“txt\_demo.txt”文件中的内容为“live with smile,we will have harvest!”执行下面程序：

txt\_file = open('txt\_demo.txt', 'r',encoding='utf-8')

txt\_file.seek(16, 0)

print(txt\_file.read())

输出结果是（ ）。

1. we will have harvest!
2. live with smile,we will have harvest!
3. live with smile
4. live with smile,we
5. **简答题**
6. 简述什么是相对路径与绝对路径。
7. 简述文件读/写位置的作用。
8. **编程题**

打开一个文本文件words\_file.txt，读取该文件中的所有内容，将这些文件内容中的英文字母按照一定的方法加密后写入到一个新文件new\_file.txt中。加密的方法是：将A变成B，B变成C，……，Y变成Z，Z变成A；a变成b，b变成c，……，y变成z，z变成a，其他字符不变化。

# 第10章 错误和异常

1. **填空题**
2. Python中所有异常的父类是 。
3. 无论try…except…finally语句是否捕获异常， 子句中的代码一定会执行。
4. with语句通过 技术管理Python中的资源。
5. **判断题**
6. 访问超出列表范围的索引时会出现IndexError异常。（ ）
7. 使用raise语句可以引发特定异常。（ ）
8. assert语句用于判定一个表达式是否为真。（ ）
9. with语句可以自动关闭资源。（ ）
10. **选择题**
11. 下列关于异常的说法，正确的是（ ）。
12. 程序中一旦遇到异常便会终止
13. 语法格式正确的代码不会出现异常
14. try-except语句可以捕获异常
15. 如果except子句没有指明任何异常，可以捕获所有的异常。
16. 下列关于try…except的说法，错误的是（ ）。
17. try子句中若没有发生异常，则忽略except子句中的代码
18. 程序捕获到异常会先执行except语句，再执行try语句
19. 若在执行try子句中的代码时引发异常，则会执行except子句中的代码
20. except可以指定错误的异常类型
21. 阅读下面代码：

num\_one = 9

num\_two = 0

print(num\_one/num\_two)

运行代码，Python解释器抛出的异常是（ ）

1. ZeroDivisionError
2. SyntaxError
3. FloatingPointError
4. OverflowError
5. 下列语句中，不能捕获和处理异常的是（ ）

try:

9/0

try:

9/0

except:

print("除数不能为0")

try:

9/0

except Exception as e:

print(e)

try:

9/0

except ZeroDivisionError as e:

print(e)

1. 下列描述中，错误的是（ ）
2. 一条try子句只能对应一个except语句
3. 一条except子句可以处理捕获的多个异常
4. 使用关键字as可以获取异常的具体信息
5. 程序发生异常后默认返回的信息包括异常类、原因和异常发生的行号
6. **简单题**
7. 简述try…except的用法与作用。
8. 简述with语句如何实现资源的释放。

# 第11章 正则表达式

1. **填空题**
2. Python中 模块为正则表达式模块。
3. 在Python正则模块中 和 用于替换目标文本中的匹配项。
4. 正则表达式中有两种匹配方式，分别是贪婪匹配和 匹配。
5. **判断题**
6. 在Python正则表达式中，\d等价于[0-9]。（ ）
7. 使用complie()函数进行预编译会后生成一个Pattern对象。（ ）
8. 贪婪匹配会尽量多地进行匹配。（ ）
9. match()函数会将所有符合匹配模式的结果返回。（ ）
10. split()函数分割的子项会保存到元组中。（ ）
11. **选择题**
12. 下列关于正则表达式的说法，错误的是（ ）。
13. 正则表达式由丰富的符号组成
14. re模块中的compile()函数会返回一个Pattern对象
15. 预编译可以减少编译正则表达式的资源开销
16. 只有通过预编译的字符串才能使用正则表达式
17. 下列关于元字符功能的说法，错误的是（ ）。
18. “.”字符可以匹配任何一个字符，除换行符外
19. “^”字符可以匹配字符串的开始
20. “?”字符表示匹配0次或多次
21. “\*”字符表示匹配1次或多次
22. 下列选项中，说法错误的是（ ）
23. match()函数从字符串开始位置检测
24. search()函数从字符串任意位置检测
25. findall()函数会以列表形式将匹配结果返回
26. finditer()函数会以列表形式将匹配结果返回
27. 阅读下面代码：

import re

str\_data = '90python \_-2'

reg = r'[A-Za-z\_](\w|\_)\*'

obj\_pattern = re.compile(reg)

match\_res= re.search(obj\_pattern, str\_data)

print(match\_res.group())

运行代码，正则表达式的匹配结果是（ ）

1. python
2. 90python
3. 90python \_-2
4. \_-2
5. 下列函数中，用于文本分割的是（ ）
6. split()
7. sub()
8. subn()
9. compile()
10. **编程题**
11. 请编写用于匹配URL的正则表达式。
12. 请编写用于匹配电子邮箱的正则表达式。

# 第12章 图形用户界面编程

1. **填空题**
2. tkinter模块中使用 组件显示错误信息或提供警告。
3. Label组件中的 属性可提供文本显示。
4. 使用tkinter中 可以创建文本框。
5. tkinter中使用 创建菜单。
6. **判断题**
7. label组件中只能显示文本信息。（ ）
8. tkinter中的组件可以独立存在。（ ）
9. 在tkinter中同一个父窗口只能使用一种几何管理器。（ ）
10. tkinter中可变类型数据的值可通过get()方法获取。（ ）
11. command可以为具体的事件进行绑定处理方法。（ ）
12. **选择题**
13. 下列选项中，可以创建一个窗口的是（ ）。
14. root = Tk()
15. root = Window()
16. root = Tkinter()
17. root = Frame()
18. 下列组件中，用于创建文本域的是（ ）。
19. Listbox
20. Text
21. Button
22. Lable
23. 已知data = StringVar()，下列选项中可以将data设置为Python的是（ ）。
24. data.set('Python')
25. data = 'Python'
26. data.value('Python')
27. data.setvalue('Python')
28. 下列关于几何布局管理器的使用，说法错误的是（ ）。
29. 在同一个父窗口中可以使用多个几何管理器
30. pack可视为一个容器
31. grid管理器可以将父组件分隔为一个二维表格
32. place布局管理器分为绝对布局和相对布局
33. 下列选项中，用于实现弹出菜单的方法是（ ）。
34. post()
35. alert()
36. add\_cascade()
37. jump()

# 第13章 进程和线程

1. **填空题**
2. 操作系统调度并执行程序，这个“执行中的程序”称为\_\_\_\_\_\_\_\_。
3. 若当前执行的进程是子进程，fork()函数返回\_\_\_\_\_\_\_\_。
4. 线程可与同属一个进程的其他线程\_\_\_\_\_\_\_\_该进程所拥有的全部资源。
5. 通过Thread()方法创建的线程默认是\_\_\_\_\_\_\_\_线程。
6. 互斥锁是最简单的加锁技术，它有\_\_\_\_\_\_\_\_和非锁定两种状态。
7. **判断题**
8. 每个应用程序都有一个自己的进程。（ ）
9. 每个进程中所拥有的数据都可以与其他进程共享。（ ）
10. 线程包含在进程之中，是进程的实际运作单位。（ ）
11. 线程在创建完成后即可启动。（ ）
12. 若没有外力作用，处于死锁的线程会一直阻塞下去。（ ）
13. **选择题**
14. 当运行中的进程因等待用户输入而无法继续运行时，会进入（ ）。
15. 新建态
16. 就绪态
17. 阻塞态
18. 挂起态
19. 下列选项中，（ ）可以批量创建进程。
20. fork()
21. Pool()
22. Process()
23. Thread()
24. （ ）是独立于程序且不会因程序终止而执行结束的。
25. 主线程
26. 子线程
27. 前台线程
28. 后台线程
29. 下列方法中，可以将运行态的线程转换成阻塞态的是（ ）。
30. join()
31. run()
32. start()
33. current\_thread()
34. 下列方法中，用于向队列中添加元素的是（ ）。
35. qsize()
36. get()
37. full()
38. put()
39. **简答题**
40. 简述主线程的作用。
41. 什么是死锁？
42. **编程题**

自定义两个线程类PrintNum和PrintWord，一个线程（PrintNum类对象）负责打印1~52，另一个线程（PrintWord类对象）打印A~Z，打印顺序是12A34B...5152Z。

# 第14章 网络编程

1. **填空题**
2. TCP、UDP协议应用在 层。
3. 计算机中进程的端口号区间是 。
4. 使用socket模块中的 方法可以创建一个socket对象。
5. A、B、C类IP地址每个网络号中的可用IP地址数量是 。
6. socket通信中服务器可使用 和 方法接收数据 。
7. **判断题**
8. TCP协议是一种面向连接的、可靠的传输协议。（ ）
9. IP地址和端口号可以标记网络中的进程。（ ）
10. 当客户端数据发送完毕之后，服务器自动关闭连接。（ ）
11. 在Python3中，使用socket模块进行数据传输时，数据类型应为字符串类型。（ ）
12. 在计算机中一个处理器某个时刻只能处理一个程序。（ ）
13. **选择题**
14. IPV4地址可分为几类（ ）。
15. 3
16. 4
17. 5
18. 6
19. 下列选项中不属于TCP/IP模型的是（ ）。
20. 应用层
21. 传输层
22. 网际层
23. 物理层
24. 下列选项中，用于绑定IP地址与端口号的方法是（ ）。
25. listen()
26. bind()
27. accept ()
28. connect ()
29. 下列关于TCP与UDP特点，说法错误的是（ ）。
30. TCP是面向连接、可靠的传输协议
31. UDP是面向无连接的传输协议
32. UDP传输数据比TCP更高效
33. TCP是基于数据包模式传输
34. 下列关于TCP并发服务器，说法错误的是（ ）。
35. 单进程非阻塞服务器通过解阻塞方式实现
36. 多进程并发服务器只能接收一个请求连接
37. 在连接数量相同时多线程并发服务器比多进行并发服务器节省资源
38. 在单进程非阻塞式服务器中只能使用非阻塞模式处理数据
39. **简答题**

分别画出基于TCP协议和基于UDP协议进行网络通信时，客户端和服务器端的通信流程图。

1. **编程题**

编写C/S模式的程序，实现客户端与服务器端的通信。要求服务器端可接收客户端发送的数据，对数据进行计算并将计算结果返回到客户端；客户端可接收服务器返回的计算结果并输出到终端。

# 第15章 数据库编程

1. **填空题**
2. 数据库是按照 来组织、存储和管理数据的仓库。
3. 关系型数据库采用 形式组织数据，由数据表和数据表之间的关系组成。
4. 文档型存储数据库采用 结构存储数据。
5. pymysql库中提供的Connection对象用于建立与MySQL数据库的 。
6. **判断题**
7. 非关系型数据库没有固定的结构，无须事先为要存储的数据建立字段。（ ）
8. MongoDB采用二维表格结构存储数据。（ ）
9. Redis数据库中的数据都是键值对，键为不能重复字符串类型。（ ）
10. Python程序无法操作任何数据库中存储的数据。（ ）
11. **选择题**
12. 下列常见的数据库中，属于关系型数据库的是（ ）。
13. Hbase
14. MongoDB
15. MySQL
16. Redis
17. MySQL是典型的关系型数据库，它主要包含的核心元素有（ ）。（多选）
18. 数据行
19. 数据列
20. 数据表
21. 数据库
22. 下列选项中，不属于MongoDB的核心元素的是（ ）。
23. 数据表
24. 文档
25. 集合
26. 数据库
27. 下列方法中，用于执行SQL语句的是（ ）。
28. commit()
29. execute()
30. fetchall()
31. cursor()
32. 下列选项中，表示包含多条匹配文档的查询对象的是（ ）。
33. MongoClient对象
34. DataBase对象
35. Collection对象
36. Cursor对象
37. **简答题**
38. 什么是数据库？
39. 简述关系型与非关系型数据库的区别。
40. **编程题**

请按照下列要求编写程序：

1. 通过pymysql与MySQL数据库建立连接。
2. 创建数据库db\_student。
3. 设计数据表students，该表格中包含学号、姓名、性别、年龄共4个字段。其中，字段“学号”为主键。
4. 向students表中插入5条记录，插入后的数据表如图15-46所示。



1. 数据表students
2. 将students表中“学号”为“5”的记录中字段“年龄”对应的值改为“21”。
3. 查询students表中字段“性别”为“女”的所有记录。
4. 删除students表的最后一条记录。

# 第16章 Djnago框架介绍

1. **填空题**
2. HTML、CSS、JavaScript文件都属于 。
3. Python Web编程中常用的Web应用统一接口为 。
4. Django创建项目的命令是 。
5. Django创建应用的命令是 。
6. 启动Django开发服务器的命令是 。
7. **判断题**
8. URL由服务器地址、资源路径组成。（ ）
9. 视图函数会返回一个HttpResponse对象。（ ）
10. 视图函数中的第一个的参数必须是HttpRequest对象。（ ）
11. Django通过路由系统设置url。（ ）
12. Django中一个模型类对应着数据库中的一张数据表。（ ）
13. **选择题**
14. 下列关于HTML协议，说法错误的是（ ）。
15. HTTPS协议比HTTP协议更安全
16. HTTP协议使用TCP连接
17. HTTPS协议建立在传输层
18. HTTP协议默认使用80端口
19. 下列选项中，不属于静态文件的是（ ）
20. test.py
21. test.js
22. test.css
23. test.html
24. 下列状态码中，表示资源请求成功的是（ ）。
25. 500
26. 400
27. 200
28. 303
29. 下列关于模板系统与路由系统，说法错误的是（ ）
30. 模板文件就是HTML文件
31. 模板文件使用“{{字典键名}}”方式传入变量
32. 每个URL都对应着一个视图函数
33. 视图中的每个函数都有对应的URL
34. 下列关于Django，说法错误的是（ ）
35. Django是一个开源的Web框架
36. Django是由Java语言编写
37. Django内置ORM框架
38. Django可以与多种数据库进行连接
39. **简答题**
40. 简述什么是WSGI。
41. 简述MVT流程。