

# Python 语言程序设计

## 模拟试卷 4

### 一、单项选择题

1. 下面不属于软件工程的 3 个要素的是 ( )  
A. 过程      B. 方法      C. 环境      D. 工具
2. 下面不属于软件设计原则是 ( )  
A. 模块化      B. 自底向上      C. 信息隐蔽      D. 抽象
3. 在关系数据库中, 用来表示实体之间联系的是 ( )  
A. 网结构      B. 二维表      C. 线性表      D. 树结构
4. 一般情况下, 当对关系 R 和 S 进行自然连接时, 要求 R 和 S 含有一个或者多个共有的 ( )  
A. 属性      B. 元组      C. 记录      D. 行
5. 有表示公司和职员及工作的三张表, 职员可在多家公司兼职。其中公司表 c(公司号, 公司名, 地址, 注册资本, 法人代表, 员工数), 职员表 s(职员号, 姓名, 性别, 年龄, 学历), 工作表 w(公司号, 职员号, 工资), 则 w 表的键(码)为 ( )  
A. 公司号, 职员号, 工资      B. 职员号      C. 职员号, 工资      D. 公司号, 职员号
6. 下列叙述中正确的是 ( )  
A. 栈是一种先进先出的线性表  
B. 栈与队列都是非线性结构  
C. 队列是一种后进先出的线性表  
D. 栈与队列都是线性结构
7. 下列叙述中正确的是 ( )  
A. 线性表的链式存储结构所需要的存储空间一般要少于顺序存储结构

B. 线性表的链式存储结构所需要的存储空间一般要多于顺序存储结构

C. 线性表的链式存储结构与顺序存储结构所需要的存储方式是相同的

D. 线性表的链式存储结构与顺序存储结构所需要的存储空间是相同的

8. 算法的有穷性是 ( )

A. 算法只能被有限的用户使用

B. 算法程序所处理的数据量是有限的

C. 算法程序的运行时间是有限的

D. 算法程序的长度是有限的

9. 软件设计中划分模块的一个准则是 ( )

A. 高内聚高耦合      B. 低内聚高耦合      C. 高内聚低耦合      D. 低内聚低耦合

10. 耦合性和内聚性是对模块独立性度量的两个标准。下面叙述中正确的是 ( )

A. 降低耦合性提高内聚性有利于提高模块的独立性

B. 耦合性是提一个模块内部各元素间彼此结合的紧密程度

C. 提高耦合性降低内聚性有利于提高模块的独立性

D. 内聚性是模块间相互连接的紧密程度

11. 每个程序都具有的统一的运算模式是 ( )

A. 顺序计算模式      B. 输入输出模式      C. IPO 模式      D. 函数调用模式

12. 以下不是 Python 语言关键字的选项是 ( )

A. None B. function      C. raise      D. if

13. 以下关于同步赋值语句描述错误的选项是 ( )

A. 同步赋值能够使得赋值过程变得更简洁

B. 判断多个单一赋值语句是否相关的方法是看其功能上是否相关或者相同

C. 设  $x, y$  表示一个点的坐标, 则  $x=a; y=b$  两条语句可以用  $x, y=a, b$  一条语句来赋值

D. 多个无关的单一赋值语句组合成同步赋值语句, 会提高程序可读性

14. 以下代码的输出结果是 ( )

```
print('{:*^10.4}'.format('Flower'))
```

`{:*^10.4}` 表示将字符串居中对齐, 总宽度为 10, 截取前 4 个字符, 并用星号填充两侧。

A. Flow      B. Flower      C. \*\*\*Flow\*\*\*      D. Flower

15. 表达式 `print(float(complex(10+5j).imag))` 的结果是 ( )

A. 10.5      B. 5      C. 5.0      D. 10.0

16. 表达式 `print("{:.2f}".format(20-2**3+10/3**2*5))` 的结果是 ( )

17.55      B. 67.56      C. 12.22      D. 17.56

17. 如果  $p = \text{ord}('a')$ , 表达式 `print(p, chr((p+3)%26+ord('a')))` 的结果是 ( )

A. 97 d      B. 97 w      C. 97 x      D. 97 c

18. 以下程序的输出结果是 ( )

```
chs = "|'\-'"
```

```
for i in range(6):
```

```
    for ch in chs[i]:
```

```
        print(ch, end='')
```

A. |' \'-'      B. |\-|      C. “|’ -’ |”      D. |"-|

19. 以下关于随机运算函数库的描述, 错误的是 ( )

A. `random` 库里提供的不同类型的随机数函数是基于 `random.random()` 函数扩展的

B. 伪随机数是计算机按一定算法产生的, 可预见的数, 所以“伪”随机

C. `uniform(a,b)` 产生一个  $a$  到  $b$  之间的随机整数

D. Python 内置的 `random` 库主要用于产生各种伪随机数序列

20. 以下关于分支和循环结构的描述，错误的是（ ）

A. 二分支结构的 **if else** 形式，适合用来控制程序分支

B. 分支结构中的代码块是用冒号来标记的

C. While 循环如果设计不小心会出现死循环

D. **continue** 在分支和循环语句里都可以使用

21. 以下程序的输出结果是（ ）

```
for i in "CHINA":  
  
    for k in range(2):  
  
        print(i,end="")  
  
        if i=='N':  
  
            break
```

A. CCHHIINNAA B. CCHHIIAA C. CCHHIINAA D. CCHHIAA

22. 以下程序的输出结果是（ ）

```
x=10  
  
while x:  
  
    x -= 1  
  
    if not x % 2:  
  
        print(x,end="")  
  
print(x)
```

A. 86420 B. 975311 C.97531 D. 864200

23. 用户输入整数的时候不合规导致程序出错，为了不让程序异常中断，需要用到的语句是（ ）

A. try-except 语句    B. eval 语句    C. 循环语句    D. if 语句

24. 以下关于 Python 内置函数的描述，错误的是（ ）

A. id() 返回一个变量的一个编号，是其在内存中的地址

B. all(ls) 返回 True，如果 ls 的每个元素都是 true

C. type() 返回一个对象的类型

D. sorted() 对一个序列类型数据进行排序，将排序后的结果写回到该变量

25. 以下关于函数的描述，正确的是（ ）

A. 函数的全局变量是列表类型的时候，函数内部不可以直接引用该全局变量

B. 函数的简单数据类型全局变量在函数内部使用的时候，需要在显式声明为全局变量

C. Python 的函数里引用一个组合数据类型变量，就会创建一个该类型对象

D. 如果函数内部定义了跟外部的全局变量同名的组合数据类型的变量，则函数内部引用的变量不确定

26. 以下程序的输出结果是（ ）

```
s1 = "QQ"
```

```
s2 = "Wechat"
```

```
print("{:*10}".format(s1,s2))
```

A. \*\*\*\*\*QQWechat=====

B. QQWechat

C. QQ\*\*\*\*\*=====Wechat

D. Invalid format specifier

27. 关于以下程序输出的两个值的描述正确的是（ ）

```
da = [1,2,3]
```

```
print(id(da))
```

```
def getda(st):
```

```
    fa = da.copy()
```

```
    print(id(fa))
```

```
getda(da)
```

A. 两个值不相等      B. 每次执行的结果不确定    C. 首次不相等      D. 两个值相等

28. 以下不是组合数据类型的是（            ）

集合类型      B. 引用类型    C. 映射类型    D. 序列类型

29. 以下关于组合类型的描述，错误的是（            ）

A. 空字典和空集合都可以用大括号来创建

B. 嵌套的字典数据类型可以用来表达高维数据

C. 字典的 pop 函数可以返回一个键对应的值，并删除该键值对

D. 可以用大括号创建字典，用中括号增加新元素

30. 以下程序的输出结果是（            ）

```
x = ['90', '87', '90']
```

```
n = 90
```

```
print(x.count(n))
```

A. 1      B. 2      C. None      D. 0

31. 以下程序的输出结果是（            ）

```
dict = {'Name':'baby','Age':7}
```

```
print(dict.items())
```

A. [('Age',7),('Name','baby')]

B. ('Age',7),('Name','baby')

C. dict\_items([('Age',7),('Name','baby')])

D. 'Age':7,'Name':baby

32. 以下程序的输出结果是（ ）

```
dat = ['1','2','3','0','0','0']
```

```
for item in dat:
```

```
    if item == '0':
```

```
        dat.remove(item)
```

```
print(dat)
```

A. [ '1' , ' 2' , ' 3' ]

B. [ '1' , ' 2' , ' 3' , ' 0' , ' 0' ]

C. [ '1' , ' 2' , ' 3' , ' 0' , ' 0' , ' 0' ]

D. [ '1' , ' 2' , ' 3' , ' 0' ]

33. 以下程序的输出结果是（ ）

```
l2 = [[1,2,3,4],[5,6,7,8]]
```

```
l2.sort(reverse = True)
```

```
print(l2)
```

A. [[5,6,7,8],[1,2,3,4]]

B. [[8,7,6,5],[4,3,2,1]]

C. [8,7,6,5],[4,3,2,1]

D. [5,6,7,8],[1,2,3,4]

34. 关于数据维度的描述，错误的是（ ）

- A. 一维数据采用线性方式组织，对应于数组概念
- B. 一维数据由键值对类型的数据组成，采用对象方式组织
- C. 一维数据是由对等关系的有序数据构成，无序数据不是一维数据
- D. 高维数据由键值对类型的数据组成，采用对象方式组织

35. 以下程序的输出结果是（ ）

```
fo = open("text.txt",'w+')  
  
x,y = 'this is a test','hello'  
  
fo.write('{}+{}\n'.format(x,y))  
  
print(fo.read())  
  
fo.close()
```

- A. this is a test hello
- B. this is a test+hello
- C. this is a test,hello
- D. this is a test

36. 文件 dat.txt 里的内容如下

QQWechat

Google Baidu

以下程序的输出结果是（ ）

```
fo = open("tet.txt",'r')  
  
fo.seek(2)  
  
print(fo.read(8))  
  
fo.close()
```



A. Wechat B. Wechat G C. Wechat Go D. Wecha

37. 以下属于 Python HTML 和 XML 解析的第三方库的是 ( )

A. Django B. BeautifulSoup C. Requests D. Networkx

38. 以下属于 Python 打包源文件为可执行文件的第三方库的是 ( )

A. PIL B. Matplotlib C. Sklearn D. Pyinstaller

39. 以下属于 Python Web 开发框架第三方库的是 ( )

A. WeRoBot

B. Django

C. Pygame

D. Wheel

40. 以下关于 Python 内置库、标准库和第三方库的描述，正确的是 ( )

A. 第三方库需要单独安装才能使用

B. 内置库里的函数不需要 import 就可以调用

C. 第三方库有三种安装方式，最常用的是 pip 工具

D. 标准库跟第三方库发布方法不一样，是跟 Python 安装包一起发布的

## 二、基本编程题

1. 考生文件夹下存在一个文件 PY101.py，参照代码模板完善代码，实现下述功能。从键盘输入一个整数和一个字符，以逗号隔开，在屏幕上显示输出一条信息。示例如下：

输入：

10,@

输出：

@@@@@@@@@10@@@@@@@@@

输入：

8,#

输出：

#####8#####

```
a, x = input("请输入一个整数和一个符号，逗号隔开：").split(',')
a = int(a) # 将输入的整数部分转换为整数类型
print(x * a + str(a) + x * a)
```

2. 考生文件夹下存在一个文件 PY102.py，参照代码模板完善代码，实现下述功能。从键盘输入一个由 1 和 0 组成的二进制字符串 s，转换为八进制数输出显示在屏幕上，示例如下：

输入

1100

输出：

转换成八进制数是：14

```
s = input() # 请输入一个由 1 和 0 组成的二进制数字串
decimal_value = int(s, 2)
octal_value = oct(decimal_value)[2:]
print("转换成八进制数是：{}".format(octal_value))
```

3. 考生文件夹下存在一个文件 PY103.py，参照代码模板完善代码，实现下述功能。文件 data.txt 文件中有多行数据，打开文件，读取数据，并将其转化为列表。统计读取的数据，计算每一行的总和、平均值，在屏幕上输出结果。

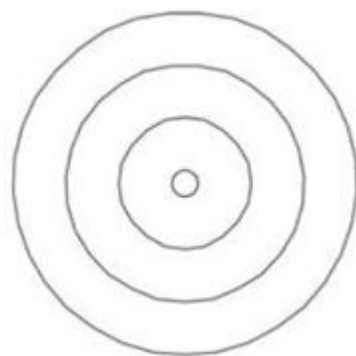
屏幕输出结果示例如下：

总和是：511.0，平均值是：85.17

```
with open('data.txt', 'r') as fi:
    s = 0 # 总和
    n = 0 # 行数
    for l in fi:
        numbers = list(map(float, l.split())) # 将每一行的数据转换为浮点数列表
        s += sum(numbers) # 计算每一行的总和并累加到 s
        n += len(numbers) # 计算每一行的元素数量并累加到 n
    print("总和是：{}, 平均值是：{:.2f}".format(s, s/n))
```

### 三、简单应用题

1. 考生文件夹下存在一个文件 PY201.py，参照代码模板完善代码，实现下述功能，不得修改其他代码。使用 turtle 库的 turtle.seth()函数绘制同心圆套圈，最小的圆圈半径为 10 像素，不同圆圈之间的半径差是 40 像素，效果如下图所示。



```

import turtle

r = 10
dr = 40
head = 90

for i in range(5): # 设置循环次数为 5
    turtle.seth(head) # 设置乌龟的方向为 head
    turtle.circle(r)

    r += dr

    turtle.seth(-head) # 设置乌龟的方向为 -head
    turtle.fd(dr)

    turtle.seth(0) # 重置乌龟的方向为 0 度
turtle.done()

```

2.参照代码模板完善代码，实现下述功能，不得修改其他代码。从键盘输入一个中文字符串变量 **s**，内部包含中文标点符号。

问题 1：考生文件夹下存在一个文件 PY202-1.py，用 **jieba** 分词，计算字符串 **s** 中的中文词汇个数，不包括中文标点符号。显示输出分词后的结果，用“/”分隔，以及中文词汇个数。示例如下：

输入：

“工业互联网实施的方式是通过通信、控制和计算技术的交叉应用，建造一个信息物理系统，促进物理系统和数字系统的融合。”

输出：

工业/互联网/实施/的方式/是/通过/通信/控制/和/计算技术/的/交叉/应用/建造/一个/信息/物理/系统/促进/物理/系统/和/数字/系统/的/融合

中文词语数是：27

```

import jieba

s = input("请输入一段中文，包含逗号和句号：")
words = jieba.lcut(s) # 使用 jieba 进行分词
# 去除标点符号
words = [word for word in words if word.strip() and not all('\u4e00' <= char <= '\u9fff' for char in word)]
# 输出分词结果
print("/".join(words))
print("\n 中文词语数是: {}".format(len(words)))
all_words = {}
max_count = 0
high_words = ''

```

```

# 统计每个词出现的次数
for i in words:
    if i in all_words:
        all_words[i] += 1
    else:
        all_words[i] = 1
# 更新出现次数最多的词
if all_words[i] > max_count:
    max_count = all_words[i]
    high_words = i
# 输出每个词及其出现次数
for key in all_words:
    print("{}:{}".format(key, all_words[key]))
print("出现最多的词是({}): {}次".format(high_words, max_count))

```

问题 2：考生文件夹下存在一个文件 PY202-2.py，在问题 1 的基础上，统计分词后的词汇出现的次数，用字典结构保存。显示输出每个词汇出现的次数，以及出现次数最多的词汇。如果有多个词汇出现次数一样多，都要显示出来。示例如下：

继续输出：

控制: 1

物理: 2

通信: 1

交叉: 1

互联网: 1

...

出现最多的词是（的 系统）: 3 次

```

# 以下代码为提示框架
# 请在...处使用一行或多行代码替换
# 请在_____处使用一行代码替换
#
# 注意：提示框架代码可以任意修改，以完成程序功能为准

# 问题 2：考生文件夹下存在一个文件 PY202-2.py，在问题 1 的基础上，统计分词后的词汇出现的次数，用字典结构保存。显示输出
每个词汇出现的次数，以及出现次数最多的词汇。如果有多个词汇出现次数一样多，都要显示出来。示例如下：
# 继续输出：
# 控制: 1

```

```
# 物理: 2
# 通信: 1
# 交叉: 1
# 互联网: 1
# ...
# 出现最多的词是（的 系统）: 3 次
```

```
# 假设分词后的词汇列表为 word_list
word_list = [...] # 这里应该是从问题 1 中得到的分词结果
```

```
# 1. 使用字典统计每个词汇出现的次数
```

```
word_count = {}
for word in word_list:
    if word in word_count:
        word_count[word] += 1
    else:
        word_count[word] = 1
```

```
# 2. 输出每个词汇出现的次数
```

```
for word, count in word_count.items():
    print(f"{word}: {count}")
```

```
# 3. 找出出现次数最多的词汇
```

```
max_count = max(word_count.values())
most_frequent_words = [word for word, count in word_count.items() if count == max_count]
```

```
# 4. 输出出现次数最多的词汇
```

```
most_frequent_words_str = " ".join(most_frequent_words)
print(f"出现最多的词是（{most_frequent_words_str}）: {max_count}次")
```

## 四、综合编程题

1. 一个人脸识别研究小组对若干名学生做了人脸识别的测试，将测试结果与被测试者的现场照片组合成文件名，写到了一个文件 `dir_100.txt` 中，每行是一个文件名的信息，示例如下：

[ ‘1709020621’ ,’ 0’ ]\_116.jpg

[ ‘1709020621’ ]\_115.jpg

[ ‘1770603107’ ,’ 1770603105’ ,’ 0’ ,’ 0’ ]\_1273.jpg

文件名各部分含义如下：

[‘识别出学号 1’，‘识别出学号 2’，‘0’ 表示检测到人脸但未识别出人]照片的顺序

编号.jpg

测试过程中，一个学生可能被抓拍到多张照片中，所以会在多个文件名中被识别，学号出现在多个文件名中；一张照片中，可能有多个人脸，但有些分辨率不够而识别不出来，文件名位置用‘0’代替学号。使用字典和列表型变量进行数据分析，最终获取实际参加测试的学生人数和人均被测次数。

考生文件夹下存在一个文件 PY301.py

(1) 读入 dir\_300.txt 文件的内容，处理每一行文件名信息。将文件名的学号内容以列表形式保存，丢掉‘0’的字符串；照片的顺序编号作为字典的关键字，学号列表作为字典的值。转换后，显示字典中的每行信息，示例如下：

116: 1709020621

115: 1709020621

117: 1709020621

1273: 1770603107, 1770603105

(2) 将该字典中的学号提取出来，构造另一个字典，以学号作为字典的关键字，累计值作为字典的值。格式示例如下：

1709020621:3

1770603107:1

1770603105:1

(3) 累计字典中关键字的个数，即为实际参加测试的学生人数；累加每个关键字对应的值，即为所有学号测试次数；与实际测试人数之比，即为人均被测次数。将实际参加测试人数和人均被测次数显示输出在屏幕上，示例如下：

实际参加测试的人数是:1024

人均被测次数是: 2.7

```
#问题一
```

```
# 读取文件内容
```

```
with open('dir_300.txt', 'r') as file:

    lines = file.readlines()
```

```
# 处理文件名信息

photo_dict = {}

for line in lines:

    # 去除换行符并分割文件名

    filename = line.strip()

    parts = filename.split('_')

    photo_number = parts[1].split('.')[0]

    student_ids = parts[0].strip('[]').split(',')

    student_ids = [id.strip().strip('"') for id in student_ids if id.strip().strip('"') != '0']

    # 构造字典

    photo_dict[photo_number] = student_ids
```

```
# 显示字典中的每行信息

for photo_number, student_ids in photo_dict.items():

    print(f"{photo_number}: {' '.join(student_ids)}")
```

#问题二

```
# 构造以学号为关键字的字典

student_count_dict = {}

for student_ids in photo_dict.values():

    for student_id in student_ids:

        if student_id in student_count_dict:

            student_count_dict[student_id] += 1

        else:

            student_count_dict[student_id] = 1
```

```
# 显示字典中的每行信息

for student_id, count in student_count_dict.items():

    print(f"{student_id}:{count}")
```

#问题三

```
# 计算实际参加测试的学生人数和人均被测次数

num_students = len(student_count_dict)

total_tests = sum(student_count_dict.values())

average_tests_per_student = total_tests / num_students if num_students > 0 else 0
```

```
# 显示结果

print(f"实际参加测试的人数是: {num_students}")
```

```
print(f"人均被测次数是: {average_tests_per_student:.1f}")
```