

# Python 语言程序设计

## 模拟试卷 3

### 一、单项选择题

1. 在面向对象方法中，一个对象请求另一对象为其服务的方式是通过发送 B  
A. 命令      B. 口令      C. 消息      D. 调用语句
2. 下面不属于软件需求分析阶段主要工作的是 C  
A. 需求评审      B. 需求获取      C. 需求变更申请      D. 需求分析
3. 下面不属于软件测试实施步骤的是 C  
A. 确认测试      B. 单元测试      C. 回归测试      D. 集成测试
4. 结构化程序设计中，下面对 goto 语句使用正确的是 B  
A. 禁止使用 goto 语句  
B. 应避免滥用 goto 语句  
C. goto 语句最好用，不容易造成结构体程序混乱  
D. 使用 goto 语句程序效率高
5. 数据库应用系统中的核心问题是 D  
A. 数据库管理员培训      B. 数据库维护  
C. 数据库系统设计      D. 数据库设计
6. 在 E-R 图中，用来表示实体联系的图形是 A  
A. 三角形      B. 菱形  
C. 椭圆形      D. 矩形
7. 在数据库设计中，将 E-R 图转换为关系数据模型的过程属于 D  
A. 物理设计阶段      B. 需求分析阶段      C. 概念设计阶段      D. 逻辑设计阶段

8. 一个栈的初始状态为空。现将元素 1、2、3、4、5、A、B、C、D、E 依次入栈，然后再依次出栈，则元素出栈的顺序是 B

A. 12345ABCDE B. EDCBA54321 C. 54321EDBCA D. ABCDE12345

9. 下列排序方法中，最坏情况下比较次数最少的是 D

A. 堆排序 B. 直接插入排序

C. 冒泡排序 D. 简单选择排序

10. 支持子程序调用的数据结构是 D

A. 队列 B. 二叉树 C. 树 D. 栈

11. 定义  $x=2.6$ ，表达式 `int(x)` 的结果是：D

A. 3 B. 2.6 C. 2.0 D. 2

12. 以下 Python 语句运行结果异常的选项是

A. `PI, r = 3.14, 4`

B. `a = 1`

`b = a = a + 1`

C. `x = True`

`int(x)`

D. `a`

13. 以下对 Python 程序设计风格描述错误的选项是 D

A. Python 中允许把多条语句写在同一行

B. Python 语句中，增加缩进表示语句块的开始，减少缩进表示语句块的退出

C. Python 可以将一条长语句分成多行显示，使用续行符“\”

D. Python 中不允许把多条语句写在同一行

14. 下列表达式的运算结果是 D

```
a = 100
```

```
b = False
```

```
a * b > -1
```

A. False      B. 1      C. 0      D. True

15. 运行以下程序，输出结果的是 D

```
str1 = "Nanjing University"
```

```
str2 = str1[:7] + "Normal" + str1[-10:]
```

```
print(str2)
```

A. Normal U      B. Nanjing Normal  
C. Normal University      D. Nanjing Normal University

16. 运行以下程序，输出结果的是：D

```
print("love".join(["Everday", "Yourself", "Python"]))
```

A. Everyday love Yourself      B. Everyday love Python  
C. love Yourself love Python      D. Everyday love Yourself love Python

17. 哪个选项是使用 PyInstaller 库对 Python 源文件打包的基本使用方法 B

A. pip -h

B. pip install <拟下载库名>

C. pip download <拟下载库名>

D. pyinstall 需要在命令行运行：pyinstaller <Python 原程序文件名>

18. 以下程序的不可能输出结果是

```
from random import *
```

```
print(round(random(),2))
```

A. 0.47      B. 0.54      C. 0.27      **D.1.87**

19. 以下程序的输出结果是 D

```
astr = '0\n'
```

```
bstr = 'A\ta\n'
```

```
print("{} {}".format(astr,bstr))
```

A. 0              B. 0

        a a              A A

C. A a              D. 0

        A    a

20.以下关于异常处理的描述，正确的是

A. try 语句中有 except 子句就不能有 finally 子句

B. Python 中，可以用异常处理捕获程序中的所有错误

C. 引发一个不存在索引的列表元素会引发 NameError 错误

**D. Python 中允许利用 raise 语句由程序主动引发异常**

21. 以下语句执行后 a、b、c 的值是 D

```
a = "watermelon"
```

```
b = "strawberry"
```

```
c = "cherry"
```

```
if a > b:
```

```
    c = a
```

```
    a = b
```

b = c

- A. watermelon strawberry cherry      B. watermelon cherry strawberry  
C. strawberry cherry watermelon      D. strawberry watermelon watermelon

22. 以下关于 Python 的控制结构，错误的是 D

- A. 每个 if 条件后要使用冒号 ( : )  
B. 在 Python 中，没有 switch-case 语句  
C. Python 中的 pass 是空语句，一般用作占位语句

D. elif 可以单独使用

23. 以下代码段，不会输出 A，B，C 的选项是

- |                          |                          |
|--------------------------|--------------------------|
| A. for i in range(3):    | B. for i in [0,1,2]:     |
| print(chr(65+i),end=",") | print(chr(65+i),end=",") |
| C. i = 0                 | D. i = 0                 |
| while i < 3:             | while i < 3:             |
| print(chr(i+65),end=",") | print(chr(i+65),end=",") |
| i += 1                   | break                    |
| continue                 | i += 1                   |

24. 设 x = 10; y = 20, 下列语句能正确运行结束的是:

- A. max = x > y ? x : y  
B. if(x > y)  
    print(x)  
C. while True:  
    pass

D. `min = x if x < y else y`

25. 以下程序的输出结果是

```
Da = {"北美洲": "北极兔", "南美洲": "托哥巨嘴鸟", "亚洲": "大熊猫", "非洲": "单峰驼", "南极洲": "帝企鹅"}
```

```
Da["非洲"] = "大猩猩"
```

```
print(Da)
```

A. ('北美洲': '北极兔', '南美洲': '托哥巨嘴鸟', '亚洲': '大熊猫', '非洲': '大猩猩', '南极洲': '帝企鹅')

B. ['北美洲': '北极兔', '南美洲': '托哥巨嘴鸟', '亚洲': '大熊猫', '非洲': '大猩猩', '南极洲': '帝企鹅']

C. {'北美洲': '北极兔', '南美洲': '托哥巨嘴鸟', '亚洲': '大熊猫', '非洲': '单峰驼', '南极洲': '帝企鹅'}

D. {'北美洲': '北极兔', '南美洲': '托哥巨嘴鸟', '亚洲': '大熊猫', '非洲': '大猩猩', '南极洲': '帝企鹅'}

26. 以下关于列表操作的描述，错误的是

A. 通过 `append` 方法可以向列表添加元素

B. 通过 `extend` 方法可以将另一个列表中的元素逐一添加到列表中

C. 通过 `insert(index,object)`方法在指定位置 `index` 前插入元素 `object`

D. 通过 `add` 方法可以向列表添加元素

27. 以下关于字典操作的描述，错误的是

A. `del` 用于删除字典或者元素

B. `clear` 用于清空字典中的数据

C. `len` 方法可以计算字典中键值对的个数

D. `keys` 方法可以获取字典的值视图

28. 以下程序的输出结果是

```
L1 = ['abc',['123','456']]
```

```
L2 = ['1','2','3']
```

```
print(L1 > L2)
```

A. False

B. TypeError:'>'not supported between instances of 'list' and 'str'

C. 1

D. True

29. 以下属于 Python 脚本程序转变为可执行程序的第三方库的是

A. requests      B. scrapy

C. numpy      D. pyinstaller30. 以下属于 Python 中文分词方向第三方库的是

A. pandas      B. beautifulsoup4

C. python-docx      D. jieba

31. 以下生成词云的 Python 第三方库的是

A. matplotlib      B. TVTK      C. mayavi      D. wordcloud

32. Python 函数中不包括

A. 标准函数      B. 第三库函数

C. 内建函数      D. 参数函数

33. Python 中，函数定义可以不包括以下：

A. 函数名      B. 关键字 def

C. 一对圆括号      D. 可选参数列表

34. 以下程序的输出结果是：

```
def func(num):
```

```
    num *= 2
```

```
x = 20
```

```
func(x)
```

```
print(x)
```

A. 40      B. 出错      C. 无输出      D. 20

35. 以下程序的输出结果是

```
def func(a,*b):
```

```
    for item in b:
```

```
        a += item
```

```
    return a
```

```
m = 0
```

```
print(func(m,1,1,2,3,5,7,12,21,33))
```

A. 33      B. 0      C. 7      D. 85

36. 以下程序的输出结果是

```
a = ["a","b","c"]
```

```
b = a[::-1]
```

```
print(b)
```

A. ['a','b','c']

B. 'c','b','a'

C. 'a','b','c'

D. ['c','b','a']

37. Python 文件只读打开模式是



A. w                      B. x

C. b                      D. r

38. Python 文件读取方法 `read(size)` 的含义是

A. 从头到尾读取文件所有内容

B. 从文件中读取一行数据

C. 从文件中读取多行数据

D. 从文件中读取指定 `size` 大小的数据，如果 `size` 为负数或者空，则读取到文件结束

39. 关于数据组织的维度描述正确的是

A. 二维数据由对等关系的有序或无序数据构成

B. 高维数据由关联关系数据构成

C. CSV 是一维数据

D. 一维数据采用线性方式存储

40. 同时去掉字符串左边和右边空格的函数是

A. `center()`              B. `count()`              C. `fomat()`              D. `strip()`

## 二、基本编程题

1. 考生文件夹下存在一个文件 `PY101.py`，请写代码替换横线，不修改其他代码，实现以下功能：

键盘输入正整数 `n`，按要求把 `n` 输出到屏幕，格式要求：宽度为 20 个字符，减号字符- 填充，右对齐，带千位分隔符。如果输入正整数超过 20 位，则按照真实长度输出。例如：键

盘输入正整数 `n` 为 1234，屏幕输出：

-----1,2348

```
n = eval(input("请输入正整数: "))
print("{:->20,}".format(n))
```

2. 考生文件夹下存在一个文件 PY102.py，请写代码替换横线，不修改其他代码，实现以下功能：

a 和 b 是两个列表变量，列表 a 为[3,6,9]已给定，键盘输入列表 b，计算 a 中元素与 b 中对应元素乘积的累加和。

例如：键盘输入列表 b 为 [1,2,3]，累加和为  $1 * 3 + 2 * 6 + 3 * 9 = 42$ ，因此，屏幕输出计算结果为 42

```
a = [3, 6, 9]
b = eval(input()) #例如: [1,2,3]
s = 0
for i in range(3):
    s += a[i] * b[i]
print(s)
```

3. 考生文件夹下存在一个文件 PY103.py，请写代码替换横线，不修改其他代码，实现以下功能：

以 123 为随机数种子，随机生成 10 个在 1（含）到 999（含）之间的随机数，每个随机数后跟随一个逗号进行分隔，屏幕输出这 10 个随机数。

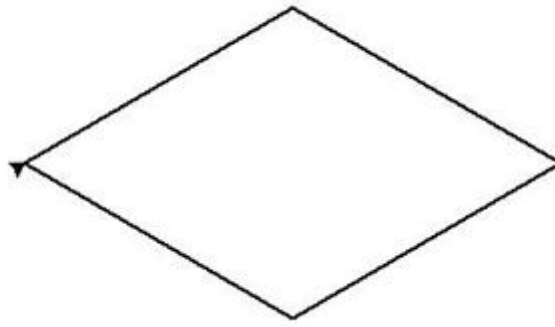
```
import random
random.seed(123)
for i in range(10):
    print(random.randint(1,1000), end=",")
```

### 三、简单应用题

1. 考生文件夹下存在一个文件 PY201.py，请写代码替换横线，不修改其他代码，实现以下功能：

使用 turtle 库的 turtle.right()函数和 turtle.fd()函数绘制一个菱形，边长为 200 像素，

4 个内角度数为 2 个 60 度和 2 个 120 度，效果如图所示。



```
import turtle
turtle.right(-30)
turtle.fd(200)
turtle.right(60)
turtle.fd(200)
turtle.right(120)
turtle.fd(200)
turtle.right(60)
turtle.fd(200)
turtle.right(120)
```

2. 考生文件夹下存在一个文件 PY202.py，该文件是本题目的代码提示框架在该文件中编写代码，以实现如下功能：

键盘输入一组人员的姓名、性别、年龄等信息，信息间采用空格分隔，每人一行，空行回车结束录入；示例格式如下：

张三 男 23

李四 女 21

王五 男 18

计算并输出这组人员的平均年龄（保留 2 位小数）和其中男性人数，格式如下：平均年龄

是 20.67 男性人数是 2

```
data = input() # 姓名 性别 年龄
total_age = 0
man_count = 0
i = 0
while data:
    i += 1
    ls = data.split(" ") #["张三","男","23"]
    sex = ls[1]
```



理斯卫 ... （后略,共 10 个字符）

```
file = open("命运.txt",'r')
lines = file.read()
d = {}
for word in lines:
    if word not in "\n":
        d[word] = d.get(word,0) + 1
file.close()
ls = list(d.items())
ls.sort(key= lambda x:x[1],reverse=True)
for i in range(10):
    print(ls[i][0],end="")
```

问题 3: 在 PY301\_3.py 文件中修改代码,对“命运.txt”文件进行字符频次统计,将所有字符按照频次从高到低排序,字符包括中文、标点、英文等符号,但不包含空格和回车。将排序后的字符及频次输出到考生文件夹下,文件名为“命运-频次排序.txt”。字符与频次之间采用英文冒号” : ”分隔,各字符之间采用英文逗号’ , ’分隔,参考 CSV 格式,最后无逗号,文件内部示例格式如下:

理:224,斯:120,卫:100

```
file = open("命运.txt",'r')
lines = file.read()
d = {}
fuhao = [" ", "\n", "\u3000"]
for word in lines:
    if word not in fuhao:
        d[word] = d.get(word,0) + 1
file.close()
ls = list(d.items())
ls.sort(key= lambda x:x[1],reverse=True)
newfile = open("命运-频次排序.txt","w")
new_ls = []
for a in ls:
    new_ls.append(a[0] + ":" + str(a[1]))
newfile.write(",".join(new_ls))
newfile.close()
```

