水利水电出版社-Python 程序设计与应用 课后习题参考答案

练习一 参考答案

一、 选择题

题号	1	2	3	4	5	6
答案	A	C	A	C	D	C

二、 填空题

题号	1	2	3	4
答案	控制器，存储器	解释	matplotlib，Pandas	使用方便，清晰易读，功能丰富

练习二 参考答案

一、判断题：

题号	1	2	3	4	5	6	7	8	9	10
答案	对	错	对	错	错					

二 选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	A	B	C	A	D	D	B	B	B	A

题号	11	12	13	14	15	16	17	18		
答案	B									

三 操作题

略

练习三 参考答案

一、选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	B	B	D	D	B	B	A	D	A	D

题号	11	12	13	14	15	16	16	17	18	19
答案	A	A	C	C	C	D	B	A	A	B

题号	20	21	22	23
答案	A	C	C	题目有误，忽略

二、填空题

（1）False

（2）True False

（3）2.5 2

（4）9

（5）回车

（6）3；6；9

（7）缩进

（8）d de
（9）字符串
（10）max()
三、设计题
（1）
```
x=5234
y=6885
print('%2.1f%%'%((y-x)*100/x))
```
（2）
```
a=int(input('a='))
b=int(input('b='))
c=int(input('c='))
d=int(input('d='))
print(a+b-c*d)
```
（3）
```
F=float(input('请输入华氏度:'))
C=5/9*(F-32)
print('摄氏度为：%.1f'%C)
```
（4）
```
r=float(input('请输入半径 r:'))
L=2*3.14*r
S=3.14*r*r
print('圆周长为%.2f,面积为%.2f'%(L,S))
```

练习四 参考答案

一、选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	C	D	C	C	D	B	A	C	B	B

题号	11	12	13
答案	C	D	D

二、填空题
（1）顺序结构、选择结构、循环结构
（2）break
（3）根据条件表达式的不同结果执行不同语句块
（4）结束当次循环，继续下次循环
（5）1：3：5：
（6）3：5：7：
（7）for while
（8）选择
三、设计题
（1）
```
n=eval(input('输入一个数：'))
```

```
if n%3==0 and n%5==0 :
    print(n,'可以同时被 3 和 5 整除')
else:
    print(n,'不可以同时被 3 和 5 整除')
```
（2）
```
year=eval(input('输入一个年份：'))
if year % 400 !=0 :
    if year % 4==0 and year %100 !=0:
        print(year,'是闰年')
    else:
        print(year,'不是闰年')
else:
     print(year,'是闰年')
```
（3）
```
import math
weight=eval(input('请输入快件重量：'))
if weight<=1 :
    pay=12
else:
    pay=12+math.ceil(weight-1)*1.3
print('快递费是：%.2f 元'%pay)
```
（4）
```
x=eval(input('输入 x:'))
if x==0:
    y=1
elif x>0:
    y=x+1
else:
    y=x*x+1
print(y)
```
（5）
```
s=0
i=100
n=1
while n<=10:
    s=s+i
    i=i/2
    if n==10: break
    s=s+i
    n+=1
print('s=%f,i=%f'%(s,i))
```
（6）
```
num = int(input("请输入一个数字: "))
factorial = 1
```

```
if num < 0:
    print("抱歉，负数没有阶乘")
elif num == 0:
    print("0 的阶乘为 1")
else:
    for i in range(1,num + 1):
        factorial = factorial*i
    print("%d 的阶乘为 %d" %(num,factorial))
```

（7）

```
sign=-1
s=0
for i in range(2,101):
    sign=-sign
    s=s+i*sign
print (s)
```

（8）

```
x=237600
r=12.45/100
y=x
n=1
while y<=x*2:
    n+=1
    y=x*(1+r)**n
print("%d 年后可以翻番"%n)
```

（9）

```
for n in range(100,1000):
    i = n / 100
    j = n / 10 % 10
    k = n % 10
    if n == i ** 3 + j ** 3 + k ** 3:
        print n
```

（10）

```
s=1
f=1
n=eval(input('输入 n：'))
for i in range(1,n+1):
    f=f*i
    s=s+1/f
print('e=%f'%s)
```

练习五 参考答案

一、 选择题

题号	1	2	3	4	5	6	7	8	9	10

答案	D	A	C	A	D	B	C D	B	C	A

题号	11	12	13	14	15	16	17	18		
答案	D	A	B	D	C	A	A	B		

二、设计题

1．输入一个班某门课的成绩（实数），用列表求最高成绩、最低成绩、不及格的成绩列表、不及格的人数，并将所有的成绩取整。

参考源程序：

```
#输入该门课的每个学生的成绩
score_list = []        #成绩列表初始化
print('请逐个输入学生成绩，用空格分隔，以回车结束：')
score_str = input()
temp = score_str.split(' ')     #以空格为分隔符切分输入的字符串为每个成绩一个子串
for str in temp:
    score_list.append(float(str))     #将成绩字符串转换为 float 型并添加到成绩列表
print(score_list)

#求最高、最低成绩
print('最高分',max(score_list),'最低分',min(score_list))

#求不及格的成绩、人数
score_fail = []     #不及格成绩列表初始化
for score in score_list:
        if score<60:
            score_fail.append(score)
print('不及格的成绩列表：',score_fail,'不及格的人数：',len(score_fail))

#将成绩取整
i = 0
for score in score_list:
    score_list[i] = int(score)
    i+=1
print(score_list)        #输出取整后的成绩列表
```

参考运行结果：

```
请逐个输入学生成绩，用空格分隔，以回车结束：
90 89.5 78 88 60.5 45.5 30
[90.0, 89.5, 78.0, 88.0, 60.5, 45.5, 30.0]
最高分 90.0 最低分 30.0
不及格的成绩列表： [45.5, 30.0] 不及格的人数： 2
[90, 89, 78, 88, 60, 45, 30]
```

2．用元组定义游戏菜单，提示用户输入菜单选择，提示信息为“请输入菜单项对应的数字，1. 游戏设置 2. 选择游戏级别 3. 我的装备 4. 我的积分 0. 退出”，当用户输入数字后，输出相应的菜单项名称，若输入 0，则显示“谢谢使用”，然后退出游戏。

参考源程序：

```
prompt = '请输入菜单项对应的数字，1. 游戏设置 2. 选择游戏级别 3. 我的装备 4. 我的积分 0. 退出'
menuTuple = ('游戏设置','选择游戏级别','我的装备','我的积分','谢谢使用')    #菜单项元组
print(prompt)    #显示提示信息
numStr = input()   #输入选择项对应的数字
if numStr>='0' and numStr<='4':
    print(menuTuple[int(numStr)-1])   #以（数字-1）为下标，输出对应的菜单项
else:
    print('输入错误，超出 0~4.')
```

3．输入一行字符，统计每个字符的出现次数，并将结果存放在字典中。

参考源程序：

```
#输入一行字符
strLine = input('输入一行字符')
#将字符串转换为字符列表
strList = list(strLine)
print(strList)

#统计  ,结果放在字典 statistics 中,每个元素为(字符：次数)
statistics = {}
for i in strList:
    if i in statistics.keys():      # 如果字典中的 key 已包含字符 i，则字符 i 的出现次数加 1
        statistics[i] += 1
    else:                                    # 如果字典中不包含 i 代表的 key，说明该元素还未出现过,设为 1
        statistics[i] = 1

 # 遍历统计结果字典：先用 items()把字典转换为列表，然后输出列表中的键值对。
for ele, count in statistics.items():
    print("%s 的出现次数为:%d" % (ele, count))
```

练习六 参考答案

一、 选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	A	D	C	D	D	B	A	C	B	A

二、填空题

1．def

2．global

3．None
4．15
5．8
6．['abc', 'acd', 'ade']
7．10
8．6
9．6
10．5

三、简答题

1．写出 Python 函数的基本格式。

Python 函数的基本结构，概括起来有如下几条：

（1）函数代码块以 def 关键词开头，后接函数标识符名称和圆括号()。

（2）任何传入参数和自变量必须放在圆括号中间，圆括号之间可以用于定义参数。

（3）函数的第一行语句可以选择性地使用文档字符串——用于存放函数说明。

（4）函数内容以冒号起始，并且缩进。

（5）return [表达式]结束函数，选择性地返回一个值给调用方。不带表达式的 return 相当于返回 None。

2．简述 return 语句的使用方法。

见实例：

【例 6.4】return 语句的用法。

```
def sum( arg1, arg2 ):
# 返回 2 个参数的和.
total = arg1 + arg2
print ("函数内：", total)
return total
# 调用 sum 函数
total = sum( 10, 20 )
print ("函数外：", total)
```

例 6.4 运行结果如下：

```
函数内：30
函数外：30
```

例 6.4 展示了 return 基本的使用方法，当使用 return 后，将 return 后的值返回到函数体并且记录下来，因此，第一个 print 函数 total 输出值是 30；由于 return 函数返回并记忆了返回值，因此，第二个 print 函数 total 输出值也是 30。

3．什么是匿名函数？在什么情况下使用匿名函数。

Python 允许不适用标准样式 def 语句定义一个函数，叫做匿名函数，Python 中使用 lambda 来创建匿名函数。

lambda 函数的用法。

（1）将 lambda 函数赋值给一个变量，通过这个变量间接调用该 lambda 函数。例如，执行语句 add=lambda x, y: x+y，定义了加法函数 lambda x, y: x+y，并将其赋值给变量 add，这样变量 add 便成为具有加法功能的函数。例如，执行 add(1,2)，输出

为 3。

（2）将 lambda 函数赋值给其他函数，从而将其他函数用该 lambda 函数替换。例如，为了把标准库 time 中的函数 sleep 的功能屏蔽（Mock），我们可以在程序初始化时调用 time.sleep=lambda x:None。这样，在后续代码中调用 time 库的 sleep 函数将不会执行原有的功能。例如，执行 time.sleep(3)时，程序不会休眠 3 秒钟，而是什么都不做。

（3）将 lambda 函数作为其他函数的返回值，返回给调用者。函数的返回值也可以是函数，例如 return lambda x, y: x+y 返回一个加法函数。这时，lambda 函数实际上是定义在某个函数内部的函数，称之为嵌套函数，或者内部函数。对应地，将包含嵌套函数的函数称之为外部函数。

（4）将 lambda 函数作为参数传递给其他函数。部分 Python 内置函数接收函数作为参数。典型的此类内置函数有如下几种：

1）filter 函数。此时 lambda 函数用于指定过滤列表元素的条件。例如 filter(lambda x: x％3== 0, [1, 2, 3])指定将列表[1,2,3]中能够被 3 整除的元素过滤出来，其结果是[3]。

2）sorted 函数。此时 lambda 函数用于指定对列表中所有元素进行排序的准则。例如 sorted([1, 2, 3, 4, 5, 6, 7, 8, 9], key=lambda x: abs(5-x))将列表[1, 2, 3, 4, 5, 6, 7, 8, 9]按照元素与 5 距离从小到大进行排序，其结果是[5, 4, 6, 3, 7, 2, 8, 1, 9]。

3）map 函数。此时 lambda 函数用于指定对列表中每一个元素的共同操作。例如 map(lambdax: x+1, [1, 2,3])将列表[1, 2, 3]中的元素分别加 1，其结果[2, 3, 4]。

4）reduce 函数。此时 lambda 函数用于指定列表中两两相邻元素的结合条件。例如 reduce(lambda a, b: '{}, {}'.format(a, b), [1, 2, 3, 4, 5, 6, 7, 8, 9])将列表[1, 2, 3, 4, 5, 6, 7, 8, 9]中的元素从左往右两两以逗号分隔的字符的形式依次结合起来，其结果是'1, 2, 3, 4, 5, 6, 7, 8, 9'。

4．什么是变量的作用域？变量的作用域有哪几类？

变量的作用域决定了在哪一部分程序可以访问哪个特定的变量名称。Python 的作用域一共有 4 种，分别是：（1）L（Local）局部作用域（2）E（Enclosing）闭包函数外的函数中（3）G（Global）全局作用域（4）B（Built-in）内建作用域。

5．简述 global 和 nonlocal 关键字的异同点。

相同点：都能修改变量的作用域。

区别：（1）两者的功能不同。global 关键字修饰变量后标识该变量是全局变量，对该变量的修改就是修改全局变量，而 nonlocal 关键字修饰变量后标识该变量是上一级函数中的局部变量，如果上一级函数中不存在该局部变量，nonlocal 位置会发生错误（最上层的函数使用 nonlocal 修饰变量必定会报错）。

（2）两者使用的范围不同。global 关键字可以用在任何地方，包括最上层函数中和嵌套函数中，及时之前未定义该变量，global 修饰后也可以直接使用，而 nonlocal 关键字只能用于嵌套函数中，并且外层函数中定义了相应的局部变量，否则会发生错误。

四、程序设计题

1．输入三个整数 x,y,z，请把这三个数有小到大输出。

参考源程序：

```
# 输入三个整数 x，y，z，请把这三个数由小到大输出。
x = int(input('请输入 x：'))
y = int(input('请输入 y：'))
```

```
z = int(input('请输入 z：'))

list = [x,y,z]    #直接给 list 赋值
print('排序前：',list)
list.sort()        #sort()是内置的排序函数，可以排序字符类型。 默认为升序
print('升序：',list)
```

参考运行结果：

```
请输入 x：4

请输入 y：3

请输入 z：4
排序前：  [4, 3, 4]
升序：  [3, 4, 4]
```

2．判断 101~200 之间有多少个素数，并输出所有素数。

参考源程序：

```
for i in range(101,201):
    a=2
    while a<i: #
        if i%a==0:break
        else:a=a+1
    if a==i:
        print(i)
```

参考运行结果：

```
101
103
107
109
113
127
131
137
139
149
151
157
163
167
173
179
181
191
```

193
197
199

3．输入一行字符，分别统计出其中英文字母、空格、数字和其他字符的个数。。
参考源程序 1：

```
s =input('请输入字符串：')
dic={'letter':0,'space':0,'integer':0,'other':0}
for i in s:
    if i >'a' and i<'z' or i>'A' and i<'Z' :
        dic['letter'] +=1
    elif i ==' ':
        dic['space'] +=1
    elif i in '0123456789':
        dic['integer'] +=1
    else:
        dic['other'] +=1
print('统计字符串：',s)
print(dic)
```

参考源程序 2：

```
#输入字符串
tmpStr = input('请输入字符串：')
alphaNum=0
spaceNum=0
numbers=0
otherNum=0
for i in tmpStr:
    if i.isalpha():
        alphaNum +=1
    elif i.isspace():
        spaceNum +=1
    elif i.isnumeric():
        numbers +=1
    else:
        otherNum +=1
print('字母=%d'%alphaNum)
print('空格=%d'%spaceNum)
print('数字=%d'%numbers)
print('其他=%d'%otherNum)
```

4．用 lambda 函数实现两个数相乘。
参考源程序：

```
mul=lambda x,y:x*y
a=eval(input('请输入 a:'))
```

```
b=eval(input('请输入 b:'))
print(mul(a,b))     #或者：print('{:.3f}*{:.3f}={:.3f}'.format(a,b,mul(a,b)))
```

5．有 n 个人围成一圈，顺序排号。从第一个人开始报数（从 1 到 3 报数），凡报到 3 的人退出圈子，问最后留下来的是原来第几号？

参考源程序 1：

```
n=int(input('输入数字：'))#输入数字
a=list(range(1,n+1))#建立一个列表，存放的是号码数
count=0;#构造一个全局变量，使得其储存每一位报的数
while len(a)>1:#循环直到列表只剩一个元素
    b=a[:]#复制列表，为下一步删除做准备
    for i in range(0,len(b)):#在 len(b)的次数中，计数，并去除数
        count+=1
        if count%3==0:#如果报三，则去除 a 中的这一位
            a.remove(b[i])
print(a[0])
```

参考源程序 2：

```
def who_remain(n):
    lst = [i for i in range(1, n+1)]#构造一个列表
    while n > 2:
        lst.pop(2)#弹出第三个元素
        before = lst[:2]#弹出后把前两个元素移到最后
        lst = lst[2:]
        lst.extend(before)
        n = len(lst)#下一次循环就从原来的 4 号（即现在的 1 号开始）
    print(lst[1])
n=int(input('输入数字：'))#输入数字
who_remain(n)
```

参考运行结果：输入 12 输出 10

练习七 参考答案

一、 填空题

1、（1）['a', 'b', 'c']　　（2）['3', '1']

2、（1）['1', '2', '3', '4']　　（2）'a bb'

二、简答题

1．为什么要引入模块，如何定义一个模块？

答：使用模块可以提高代码的可维护性，同时也使得编写代码不必从零开始。当一个模块编写完毕，就可以被其他地方引用。我们在编写程序的时候，也经常引用其他模块，包括 Python 内置的模块和来自第三方的模块。所有的.py 文件都被视为是一个模块

2．模块的导入方法有哪些？

答：Python 3 中提供了三种导入方式，即直接使用 import 导入模块、通过 sys 模块导入自定义模块、通过 pth 文件找到自定义模块。

3．使用 import、from…import、from…import *导入模块有什么异同点？

答：import 语句、from…import 语句和 from…import*语句都能实现模块的导入。各自特点如下：

（1）import 模块：导入一个模块，相当于导入的是一个文件夹，是个相对路径。

（2）from…import：导入了一个模块中的一个函数，相当于导入的是一个文件夹中的文件，是个绝对路径。

（3）from…import *：是把一个模块中所有函数都导入进来，相当于导入的是一个文件夹中所有文件，所有函数都是绝对路径。

总结起来，from…import*语句与 import 区别在于：import 导入模块，每次使用模块中的函数都要确定是哪个模块；from…import*导入模块，每次使用模块中的函数，直接使用函数就可以了，因为已经知道该函数是哪个模块中的了。

4．导入自定义模块方法有哪些？

答：

（1）直接 import：这里有个大前提，就是你的 py 执行文件和模块同属于同个目录（父级目录）；

（2）通过 sys 模块导入自定义模块的路径 path：如果执行文件和模块不在同一目录，这时候直接 import 是找不到自定义模块的。

在环境变量中找到自定义模块

（3）通过 pth 文件找到自定义模块：通过 pth 文件找到自定义模块利用了系统变量，Python 会扫描 path 变量的路径来导入模块，可以在系统 path 里面添加。此处可以采用 pth 文件添加。

5．如何在 Python 使用第三方库？

答：

在第三方模块安装好后，其搜索路径就已经设置好了，我们可以尝试直接 import numpy 等已安装的第三方模块。当我们试图加载一个模块时，Python 会在指定的路径下搜索对应的.py 文件，如果找不到，就会报错。默认情况下，Python 解释器会搜索当前目录、所有已安装的内置模块和第三方模块，搜索路径存放在 sys 模块的 path 变量中。

如果我们要添加自己的搜索目录，有两种方法。一是直接修改 sys.path，添加要搜索的目录。第二种方法是设置环境变量 PYTHONPATH，该环境变量的内容会被自动添加到模块搜索路径中。设置方式与设置 Path 环境变量类似。注意只需要添加你自己的搜索路径，Python 自己本身的搜索路径不受影响。

6．包和模块之间的关系是什么，如何导入和使用包？

答：

在模块使用过程中，可能会遇到下面的问题，那就是如果不同的人编写的模块名相同怎么办？为了避免模块名冲突，Python 又引入了按目录来组织模块的方法，称为包（Package）。

（1）import：导入一个包，相当于导入的是一个文件夹，是个相对路径。

（2）from…import：导入了一个包中的一个函数，相当于导入的是一个文件夹中的文件，是个绝对路径。

（3）from…import *：是把一个包中所有函数都导入进来，相当于导入的是一个文件夹中所有文件，所有函数都是绝对路径

三、程序设计题

1．设计一个简单程序，调用系统模块 time。

Filename: test.py

```
# 导入模块
import time
# 现在可以调用模块里包含的函数了
```

time.time()

2．编写一个模块并调用。

```
# Filename: support.py
def print_func( par ):
print ("Hello : ", par)
return
```

test.py 引入 support 模块：

```
# Filename: test.py
# 导入模块
import support
# 现在可以调用模块里包含的函数了
support.print_func("成功导入模块！")
```

练习八 参考答案

一、选择题

题号	1	2	3	4
答案	D	A	D	B

二、设计题

1．完善例 8.2 中的 Student 类，增加构造函数（带默认初始值）。基于 Student 类定义 UniversityStudent 类，增加新的属性“学生专业”，并重载相关的方法。然后定义派生类的 2 个对象（不带参数的、带参数的），设置对象的值、显示对象的值。

```
#定义 Student 类
class Student:
    stu_count=0 #类数据成员
    #构造方法
    def __init__(self,no_1='',name_1='',sex_1=''):
        self.no=no_1            #3 个对象数据成员
        self.name=name_1
        self.sex=sex_1
        Student.stu_count+=1 #每设置一个学生信息，学生数量加 1
    #设置学生信息
    def set(self,no_1,name_1,sex_1):
        self.no=no_1            #3 个对象数据成员
        self.name=name_1
        self.sex=sex_1

    #显示学生信息
    def display(self):
        print('No:',self.no,'Name:',self.name,'Sex:',self.sex)
```

```
#定义派生类 UniversityStudent
class UniversityStudent(Student):
    #重载构造方法
    def __init__(self,no_1='',name_1='',sex_1='',speciality_1=''):
        Student.__init__(self,no_1,name_1,sex_1);
        self.speciality=speciality_1;
    #重载 set()方法
    def set(self,no_1,name_1,sex_1,speciality_1):
        Student.set(self,no_1,name_1, sex_1)
        self.speciality = speciality_1
    #重载 display()方法
    def display(self):
        Student.display(self);
        print('speciality:',self.speciality)


#定义学生类的对象 stu1 和 stu2，并调用方法实现 2 个学生对象的设置和显示
stu1=UniversityStudent()
stu1.set('180805210','feng cheng','M','Computer')
stu1.display()

stu2=UniversityStudent('180805211','wang haimeng','FM','English')
stu2.display()

#分别通过类名和对象名访问 stu_count,输出学生数量
print('学生对象数量：',Student.stu_count)
print('学生对象数量：',stu1.stu_count)
print('学生对象数量：',stu2.stu_count)
```

运行结果：

```
No: 180805210 Name: feng cheng Sex: M
speciality: Computer
No: 180805211 Name: wang haimeng Sex: FM
speciality: English
学生对象数量： 2
学生对象数量： 2
学生对象数量： 2
```

2．完善例 8.5，实现两个向量对象的减法（v1-v2）和乘法（v1*v2）

提示：设两个向量 a=(x1,y1),b=(x2,y2),则

a+b=(x1+x2,y1+y2) a-b=(x1-x2,y1-y2) a·b=x1x2+y1y2

参考源程序：

```
#定义 Vector 类
class Vector:
```

```
    #定义构造方法
    def __init__(self, newa, newb):
        self.a = newa
        self.b = newb
    #定义方法，输出向量
    def print_vector(self):
        print('Vector (%d, %d)' % (self.a, self.b))

    #重载运算符 “+”
    def __add__(self,other):
        return Vector(self.a + other.a, self.b + other.b)

    #重载运算符 “-”
    def __sub__(self,other):
        return Vector(self.a - other.a, self.b - other.b)

    #重载运算符 “*”
    def __mul__(self,other):
        return (self.a * other.a +self.b*other.b)


# 定义对象并调用方法
v1 = Vector(4,10)
v2 = Vector(2,-3)
print('v1:',end=' ')
v1.print_vector()
print('v2:',end=' ')
v2.print_vector()
#调用重载的加法方法__add__()，实现两个对象直接相加
v3=v1+v2
print('v1+v2:',end=' ')
v3.print_vector()
#调用重载的加法方法__sub__()，实现两个对象直接相减
v4= v1-v2
print('v1-v2:',end=' ')
v4.print_vector()
#调用重载的加法方法__mul__()，实现两个对象直接相乘
v5=v1*v2
print('v1*v2：',v5)
```

练习九 参考答案

一、选择题

题号	1	2	3	4	5	6	7	8
答案	B	A	A	D	C	C	A	B

二、填空题

（1）open()

（2）write()

（3）[‘北京’, ’上海’ , ’广州\n 济南’ , ’青岛’ , ’烟台’]

三、设计题

（1）

答：r 以只读方式打开文件，文件的指针将会放在文件的开头，此模式为默认模式。r+模式是打开一个文件用于读写。文件指针将会放在文件的开头。

（2）

```
f=open('student.txt','w')
stop=False
while( not stop):
    Xuehao=input('请输入学号:')
    Xingming=input('请输入姓名:')
    Xingbie=input('请输入性别:')
    Banji=input('请输入班级:')
    f.write(' '.join([Xuehao, Xingming, Xingbie, Banji])+'\n')
    sp=input('是否继续?(y/n):')
    if sp=='n':
        stop=True
f.close()
```

（3）

```
f=open('student.txt','r')
ls=[]
lines=f.readlines()
for line in lines:
    ls.append(list(line.replace('\n','').split(' ')))
f.close()
```

（4）

```
f = open("article.txt", "r")
words = []
for line in f:
    for word in line.replace('\n', '').split(" "):
        words.append(word)

map = {}
for word in words:
    map[word] = map[word] + 1 if word in map.keys() else 1

for key in map:
    print(key + ": " + str(map[key]))
```

练习十 参考答案

一、选择题

题号	1	2	3
答案	D	A	B

二、问答题

1、答：可以。因为 try 语句块中可能出现多类异常，利用多个 except 语句可以分别捕获并处理我们感兴趣的异常。

2、答：利用一个 try 语句对应多个 except 语句，可以对多个异常进行分别处理。

3、（1）语法错误 SyntaxError

（2）异常，索引数超出范围 IndexError

（3）异常，属性错误 AttributeError

（4）异常，找不到关键字 KeyError

（5）语法错误 SyntaxError

三、设计题

1.

```
try:
    f = open("d:/f9-1.txt", "w")
    f.write( "学习 Python 棒极了！  " )
except OSError:
    print('文件出错啦！ ')
finally:
    f.close()
```

2.

```
#猜数游戏 1~100.py
import random
Guess_number = random.randint(0,100)
print('请输入整数:')
i = 0
while True:
    try:
        i = i + 1
        a = int(input())
        if a < Guess_number:
            print('太小了')
        elif a > Guess_number:
            print('太大了')
        else:
            print('预测{}次，猜中了'.format(i))
            break
    except ValueError:
```

```
        print('输入错误，请输入整数：')
```