

# 湖 北 大 学

2024 -- 2025 学年度

第 1 学期

## 学 生 实 验 报 告 册

学 院： 网络空间安全学院

学生姓名： 汪应松

班 级： 信息安全 2304 班

学 号： 202331120011118

课程名称： Python 程序设计

任课老师： 胡钊

## 学生实验守则

- 1、学生在规定的时间内进行实验，不得无故缺席或迟到。
- 2、学生在每次实验前对排定要做的实验应进行预习，并按要求作好预习报告。
- 3、每次实验前，必须交上次实验报告和本次实验预习报告，并经指导教师提问、检查同意后，才可进行本次实验。
- 4、学生进入实验室指定位置后，首先根据仪器清单核对自己使用的仪器是否有缺少或损坏，发现问题及时向指导教师报告，严禁擅自动用别组仪器。
- 5、实验时必须有实事求是、严肃认真的科学态度，严格遵守仪器操作规程和注意事项。
- 6、实验完毕应将实验数据交给指导教师检查，合格后，整理复原好仪器设备，方可离开实验室。
- 7、保持实验室肃静和整洁，不得大声喧哗，乱丢垃圾和吃东西。
- 8、学生在实验过程中，由于不遵守操作规程或未经许可，擅自进行实验而造成事故、损坏仪器设备，应及时报告，并填写损坏清单，按院有关规定进行赔偿。

# 实验报告单

实验名称: Python 程序设计实验

同组人:

实验室: 双创大楼 701

时间: 2024/11/27

## 实验目标和实验内容:

(包含实验目的、实验器材、实验原理、实验性质、实验步骤、数据记录与处理及结果讨论等内容)

实验目的:

1. 掌握类的继承, 理解类的多态。
2. 熟悉 random、timeit、time 模块。
2. 熟悉 numpy 模块, 掌握利用 numpy 模块处理高维数组和矩阵。

实验题目:

1. 设计一个图形类 Shape, 具有以下成员和方法:

成员: name: 图形的名称 ; color: 图形的颜色;

方法: area(): 返回图形的面积, 具体实现留空(使用 pass 语句)。

从 Shape 类派生出两个子类: Circle 和 Rectangle。

Circle 类有新添加的实例成员: radius, 表示圆的半径, 并重写 area() 方法以计算圆的面积。

Rectangle 类有新添加实例成员: length, 表示长方形的长度; width, 表示长方形的宽度, 并重写 area() 方法以计算长方形的面积。

设计并实现上述类, 然后创建一些 Circle 和 Rectangle 对象, 调用它们的 area() 方法, 并显示结果。

源代码:

```
import math
class Shape:
    def __init__(self, name, color):
        self.name=name
        self.color=color
    def area(self):
        pass

class Circle(Shape):
    def __init__(self, name, color, radius):
```

```

        Shape.__int__(self, name, color)
        self.radius=radius
def area(self):
    return self.radius*self.radius*math.pi

class Rectangle(Shape):
    def __init__(self, name, color, length, width):
        Shape.__int__(self, name, color)
        self.length=length
        self.width=width
    def area(self):
        return self.length*self.width
if __name__=="__main__":
    s1=Circle('圆形','红色',2)
    areal=s1.area()
    print("%s 的%s 的面积为: "%(s1.color, s1.name), areal)
    s2=Rectangle('长方形','白色',5,8)
    area2=s2.area()
    print("%s 的%s 的面积为: " % (s2.color, s2.name), area2)

```

实验结果:

```

C:\Users\27356\AppData\Local\Programs\Python\Python39\python.exe
红色的圆形的面积为:  12.566370614359172
白色的长方形的面积为:  40

进程已结束，退出代码为 0

```

2. 随机生成 10 个 1-100 范围内的整数,构成一个列表 A。随机生成 10 个 1-100 范围内的偶数,构成一个列表 B。删除其中的重复元素后得到新的列表 A1 和 B1,输出结果。

源代码:

```
import random
A=[]
B=[]
for i in range(1,11):
    A.append(random.randint(1,100))
print("列表 A 为: ",A)
for j in range(1,11):
    B.append(random.randrange(0,101,2))
print("列表 B 为: ",B)
A1=[]
for a in A:
    if a not in A1:
        A1.append(a)
print("去重后列表 A1 为: ",A1)
B1=[]
for b in B:
    if b not in B1:
        B1.append(b)
print("去重后列表 B1 为: ",B1)
```

实验结果:

```
C:\Users\27356\AppData\Local\Programs\Python\Python
列表A为:  [86, 86, 2, 52, 59, 3, 58, 72, 22, 2]
列表B为:  [24, 36, 6, 46, 68, 2, 4, 100, 46, 64]
去重后列表A1为:  [86, 2, 52, 59, 3, 58, 72, 22]
去重后列表B1为:  [24, 36, 6, 46, 68, 2, 4, 100, 64]

进程已结束,退出代码为 0
```

3. 编写一个计算正整数 a 和 b 最大公因子的函数 gcd(a,b)。使用 time 模块的 time() 函数计算 gcd(1658, 654) 的运行时间,再使用 timeit 模块的 timeit() 函数计算 gcd(1658, 654) 运行 100 次所需时间,并输出所需时间。最后利用 time.ctime() 函数输出当前日期和时间。

源代码:

```
import time
import timeit
def gcd(a,b):
    if b==0:
        return a
    else:
        return gcd(b,a%b)
if __name__=="__main__":
    start=time.time()
    print("最大公因子为: ",gcd(1658,654))
    stop=time.time()
    print("运行时间为: ",stop-start)

    t1=timeit.timeit(stmt='gcd(1685,654)',globals=globals(),number=100)
    print("运行 100 次时间为: ",t1)

    print("当前日期时间",time.ctime())
```

实验结果:

```
C:\Users\27356\AppData\Local\Programs\Python
最大公因子为:  2
运行时间为:  0.0
运行100次时间为:  3.9000005926936865e-05
当前日期时间 Mon Dec  2 18:27:20 2024

进程已结束,退出代码为 0
```

4. 由列表[5, 6, 3, 12, 41, 25, 33, 51, 16] 和 [5, 6, 8, 10, 15, 3, 25, 29, 66]分别生成两个矩阵 3\*3 的 A 和 B。输出 A 和 B，求 A 的转置并输出，求 A 的逆并输出，求矩阵 A 的行列式并输出，求 A 和 B 矩阵相乘的结果。

源代码：

```
import numpy as np
s1=[5, 6, 3, 12, 41, 25, 33, 51, 16]
s2=[5, 6, 8, 10, 15, 3, 25, 29, 66]
A=np.asmatrix(s1).reshape(3, 3)
print("矩阵 A 为：")
print(A)
B=np.array(s2).reshape(3, 3)
print("矩阵 B 为：")
print(B)
print("A 的转置为：")
print(A.T)
print("A 的逆为：")
print(A.I)
print("A 的行列式为：")
print(np.linalg.det(A))
print("AB 的乘积为：")
print(np.matmul(A, B))
```

实验结果：

```
C:\Users\27356\AppData\Local\Programs\Python\Pyth
矩阵A为：
[[ 5  6  3]
 [12 41 25]
 [33 51 16]]
矩阵B为：
[[ 5  6  8]
 [10 15  3]
 [25 29 66]]
```

A的转置为:

```
[[ 5 12 33]
 [ 6 41 51]
 [ 3 25 16]]
```

A的逆为:

```
[[ 0.40723684 -0.0375 -0.01776316]
 [-0.41644737 0.0125 0.05855263]
 [ 0.4875      0.0375 -0.0875    ]]
```

A的行列式为:

-1520.0000000000005

AB的乘积为:

```
[[ 160 207 256]
 [1095 1412 1869]
 [1075 1427 1473]]
```

进程已结束,退出代码为 0

成绩:

批阅教师: \_\_\_\_\_

日期: \_\_\_\_\_