

深圳大学实验报告

课程名称： 现代程序设计

实验项目名称： 类与对象

学院： 电子与信息工程学院

专业：

指导教师： 邹文斌

报告人： 学号： 班级：

实验时间： 2023 年 11 月 20 日

实验报告提交时间： 2023 年 11 月 21 日

教务部制

一、 实验要求

- a) 类与对象
- b) 算法
- c) 需上交实验报告、py 源程序。

二、 实验环境

Python IDLE, Pycharm 等

三、 实验内容

1. Country 类

编写一个名为 Country 的类，该类具有一个类属性 continent，表示国家所在的大洲。此外，Country 类还具有一个实例属性 name，表示国家的名称。
编写一个方法 get_continent()，该方法用于返回国家所在的大洲。
编写一个方法 get_name()，该方法用于返回国家的名称。

2. 购物车

开发一个购物车程序。请设计一个 Cart 类，**至少具有以下功能**：

每个购物车有一个包含商品的列表属性。

实例方法 add_item(self, name, price) 用于向购物车添加商品，其中 name 为商品名称，price 为商品价格。

实例方法 remove_item(self, name) 用于从购物车移除指定名称的商品。

实例方法 calculate_total(self) 用于计算购物车中商品的总价格。

类方法 get_discounted_price(cls, price, discount) 用于返回折扣后的价格，其中 price 为原始价格，discount 为折扣率。

静态方法 convert_currency(price, exchange_rate) 用于将价格转换为其他货币的金额，其中 price 为原始价格，exchange_rate 为汇率。

要求：

calculate_total() 方法应返回购物车中商品的总价格。

get_discounted_price(price, discount) 方法应返回折扣后的价格，

计算公式为折扣后价格 = 原始价格 * (1 - 折扣率)。

convert_currency(price, exchange_rate) 方法应返回转换后的金额，计算公式为转换后金额 = 原始价格 * 汇率。

在 add_item(name, price) 方法中，如果输入的商品价格为负数，则输出错误信息。

在 remove_item(name) 方法中，如果购物车中不存在指定名称的商品，则输出错误信息。

3. 小马过河 (A+挑战题)

小慧要赶 N ($2 \leq N \leq 100$) 匹小马过河, N 匹小马过河都需要一定的时间(分钟), 小慧每次过河最多能赶两匹小马(骑一匹并赶一匹), 返回时需骑一匹, 每次过河的时间为走的小马花费的时间。请计算至少需要多长时间才能把 N 匹小马全部赶过河。

例如: $N=4$, 4 匹小马过河需要的时间分别为 1, 2, 3, 4(单位: 分钟)。

用时最少的一种过河方式:

第一次: 赶 1 分钟和 2 分钟的小马过河, 然后骑 1 分钟的小马返回, 共花费 3 分钟(过去花费 2 分钟, 回来花费 1 分钟);

第二次: 赶 3 分钟和 4 分钟的小马过河, 然后骑 2 分钟的小马返回, 共花费 6 分钟(过去花费 4 分钟, 回来花费 2 分钟);

第三次: 赶 1 分钟和 2 分钟的小马过河, 共花费 2 分钟(过去花费 2 分钟);

赶这 4 匹小马过河一共花费 11 分钟($11=3+6+2$)。

输入描述:

第一行输入一个正整数 N ($2 \leq N \leq 100$), 表示需要过河的小马数量

第二行输入 N 个正整数 ($1 \leq \text{正整数} \leq 100$), 表示每匹小马过河需要花费的时间(分钟), 正整数之间以一

个空格隔开

输出描述:

输出一个整数, 表示赶 N 匹小马全部过河至少需要花费的时间

样例输入:

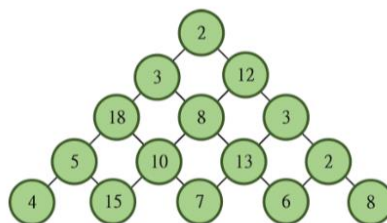
4
1 2 3 4

样例输出:

11

4. 数字塔 (A+挑战题)

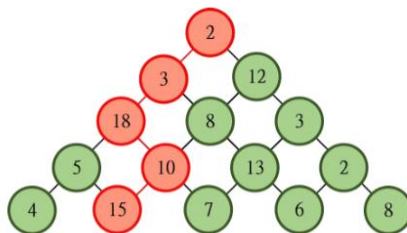
数字塔是由 N 行数堆积而成, 最顶层只有一个数, 次顶层两个数, 以此类推。相邻层之间的数用线连接, 下一层的每个数与它上一层左上方和右上方的数连接(左上方或右上方没有数则不需要连接)。



编程实现:

有一个 N 行 ($2 \leq N \leq 50$) 的数字塔, 小蓝想要从最顶层开始, 沿着线一层一层向下移动, 移动到底层。小蓝想找出一条移动路径, 使得路径上的数之和最大(包含顶层和底层的数), 请计算出最大的和是多少。

例如: $N=5$, 5 层的数字塔, 每层的数如下图所示:



从顶层数为 2 到底层数为 15 的路径上的数之和最大，最大和为 48。路径为：2→3→18→10→15。

输入描述：

第一行输入一个正整数 N ($2 \leq N \leq 50$), 表示数字塔的层数

接下来输入 N 行，其中第一行为一个正整数，接下来每行的正整数比上一行多一个，每行的正整数之间以一个英文逗号隔开 ($1 \leq \text{正整数} \leq 1000$)

输出描述：

输出一个整数，表示从数字塔最顶层移动到底层的路径上的数之和的最大值

样例输入：

```
5
2
3, 12
18, 8, 3
5, 10, 13, 2
4, 15, 7, 6, 8
```

样例输出：

```
48
```

四、实验过程

思路：

(涉及到算法实现的实验需阐述算法的逻辑关系)

一、Country 类

1. 编写一个名为 Country 的类，设置其类属性 continent，表示国家所在的大洲。
2. 设置实例属性 name，表示国家的名称。
3. 编写一个方法 get_continent()，该方法用于返回国家所在的大洲；编写一个方法 get_name()，该方法用于返回国家的名称。
4. 实例化对象，运行程序。

二、购物车

1. 编写一个名为 Cart 的类，设置一个列表作为实例属性，用于存储购物车商品信息。

2. 编写实例方法 `add_item(self, name, price)` 用于向购物车添加商品，其中 `name` 为商品名称，`price` 为商品价格，如果输入的商品价格为负数，则输出错误信息。
3. 编写实例方法 `remove_item(self, name)`，用于从购物车移除指定名称的商品，如果购物车中不存在指定名称的商品，则输出错误信息。
4. 编写实例方法 `calculate_total(self)`，用于计算购物车中商品的总价格。
5. 编写类方法 `get_discounted_price(cls, price, discount)` 用于返回折扣后的价格，其中 `price` 为原始价格，`discount` 为折扣率。方法返回折扣后的价格，计算公式为：折扣后价格 = 原始价格 * (1 - 折扣率)。
6. 编写静态方法 `convert_currency(price, exchange_rate)` 用于将价格转换为其他货币的金额，其中 `price` 为原始价格，`exchange_rate` 为汇率。方法返回转换后的金额，计算公式为：转换后金额 = 原始价格 * 汇率。
7. 实例化对象，运行程序测试结果。

三、小马过河

1. 获得小马的数量以及各个小马过河需要的时间。
2. 将各个小马过河时间从小到大排序，计算最少需要的时间。
3. 先赶最快的两匹小马过河，再返回赶最慢的两匹小马过河，再处理剩下的小马。
4. 计算输出结果。

完整代码:

(必须有详细的注释)

一、Country 类 Country.py

```
class Country:

    continent = 'Asia'

    def __init__(self, name):

        self.name = name

#用于返回国家所在的大洲
```

```
def get_continent(self):  
    return Country.continent  
  
#用于返回国家的名称  
def get_name(self):  
    return self.name  
  
country1 = Country("China")  
print(country1.get_continent())  
print(country1.get_name())
```

二、购物车

```
class Cart:  
    def __init__(self):  
        self.items = []  
  
    #添加实例方法，用于向购物车中添加商品  
    def add_item(self, name, price):  
        if price < 0:  
            print("请输入合法的价格信息！")  
        else:  
            self.items.append((name, price))  
  
    #添加实例方法，用于从购物车移除指定名称的商品  
    def remove_item(self, name):  
        for item in self.items:  
            if item[0] == name:  
                self.items.remove(item)  
                return  
        print("未找到该商品信息！")
```

```
#添加实例方法，用于计算购物车中商品的总价格

def calculate_total(self):

    total = 0

    for item in self.items:

        total += item[1]

    return total


#类方法，用于返回折扣后的价格

@classmethod

def get_discounted_price(cls, price, discount):

    return price * (1 - discount) #折扣后价格 = 原始价格 * (1 - 折扣率)


#静态方法，用于将价格转换为其他货币的金额

@staticmethod

def convert_currency(price, exchange_rate):

    return price * exchange_rate #转换后金额 = 原始价格 * 汇率


cart = Cart()
cart.add_item("Shirt", 20)
cart.add_item("Pants", 30)
cart.add_item("Shoes", 50)


print(cart.calculate_total()) #期望输出 100


cart.remove_item("Pants")
print(cart.calculate_total()) #期望输出 70
```

```
discounted_price = Cart.get_discounted_price(100, 0.2)
print(discounted_price) #期望输出 80.0
```

```
converted_price = Cart.convert_currency(100, 0.8)
print(converted_price) #期望输出 80.0
```

```
cart.add_item("Hat", -80) #期望输出错误信息提示
cart.remove_item("Doors") #期望输出错误信息提示
```

三、小马过河 horse.py

读取输入

```
N = int(input("请输入需要过河的小马数量:"))
times = list(map(int, input("请输入每匹小马过河需要花费的时间(分钟), 正整数之
间以一个空格隔开: ").split()))
```

将过河时间从小到大排序

```
times.sort()
```

计算最少需要的时间

```
total_time = 0
```

```
while N > 3:
```

赶最快的两匹小马过河

```
time1 = times[0]
```

```
time2 = times[1]
```

```
total_time += time2
```

返回

```
total_time += time1
```

赶最慢的两匹小马过河

```
time3 = times[N-1]
```

```
time4 = times[N-2]
```



```

    total_time += time3

    # 返回

    total_time += time2

    N -= 2

# 处理剩下的 1-3 匹小马
if N == 2:
    total_time += times[1]
elif N == 3:
    total_time += sum(times)

# 输出结果
print("赶 N 匹小马全部过河至少需要花费的时间为:", total_time)

```

五、 实验结果

（运行结果，截图）

一、Country 类



```

>>>
===== RESTART: D:\MineP\Program
所在的大洲是: Asia
该国家为: China
>>> |

```

二、购物车

```

cart = Cart()
cart.add_item("Shirt", 20)
cart.add_item("Pants", 30)
cart.add_item("Shoes", 50)

print(cart.calculate_total()) #期望输出 100

cart.remove_item("Pants")
print(cart.calculate_total()) #期望输出 70

discounted_price = Cart.get_discounted_price(100, 0.2)
print(discounted_price) #期望输出 80.0

converted_price = Cart.convert_currency(100, 0.8)
print(converted_price) #期望输出 80.0

cart.add_item("Hat", -80) #期望输出错误信息提示
cart.remove_item("Doors") #期望输出错误信息提示

```

```

IDLE Shell 3.9.6
File Edit Shell Debug Options
Python 3.9.6 (tags/v3.9.6:db3ff77, Oct 8 2021, 14:10:04) [AMD64]
Type "help", "copyright", "credits()" or "quit()" for more
>>>
===== RESTART: D:\MineP\Python\
100
70
80.0
80.0
请输入合法的价格信息!
未找到该商品信息!
>>>

```

三、小马过河

```

===== RESTART: D:\MineP\Program\Python.py\Modern_Program\Task11\h
请输入需要过河的小马数量:4
请输入每匹小马过河需要花费的时间(分钟),正整数之间以一个空格隔开: 1 2 3 4
赶N匹小马全部过河至少需要花费的时间为: 11
>>>

```

六、实验心得

（本次实验遇到的问题，解决过程，有什么收获等）

1. 熟练掌握了类的各类方法的构造。
2. 弄懂了类属性、类方法、实例属性、实例方法、静态方法等属性的概念和编写规则。
3. 对面向对象编程有了更深一步的了解。

成绩评定:

实验过程（60 分）	实验结果（30 分）	心得体会（10 分）	总分（100 分）

指导教师签字： 年 月 日

备注:

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。