

## Day 07 作业题

### 简答题

#### 1. 简述继承的语法?

```
class 类B(类A):  
    pass  
# 类 B继承类 A
```

#### 2. 为什么要重写,什么是重写,重写后如何调用父类的方法?

```
# 为什么要重写  
父类中的方法,不能满足子类对象的需求  
# 什么是重写  
子类实现了和父类同名的方法  
# 重写后如何调用父类的方法  
super().方法名()
```

#### 3. 描述打开文件的函数以及参数表示的意思?

打开文件: 将文件从磁盘(硬盘)中读取到内存中

语法:

```
open(file, mode='r', encoding=None)
```

> 参数 file: 是要打开的文件, 类型是字符串, 文件的路径可以是相对路径, 也可以是绝对路径(从根目录开始书写的路径), 建议使用相对路径(相对于当前代码文件所在的路径, ./ ../ )

> 参数 mode: 默认参数(缺省参数), 表示的是打开文件的方式

> r: read 只读打开

> w: write 只写打开

> a: append 追加打开, 在文件的末尾写入内容

> 参数 encoding: 编码方式, (文字和二进制如何进行转换的)

> gbk: 将一个汉字转换为 2 个字节的二进制

> utf-8: 常用, 将一个汉字转换为 3 个字节的二进制

> 返回值: 返回的是 文件对象, 后续对文件的操作, 都需要这个对象

### 代码题

#### Game 案例完成

定义一个游戏类 `Game`, 包含 属性 玩家名字

1. 要求记录游戏的最高分( `top_score` 类属性)和获得最高分的玩家名字( `top_score_player` ),
2. 定义方法: `show_help` 显示游戏的帮助信息 输出 这是游戏的帮助信息

3. 定义方法: `show_top_score`, 打印输出游戏的最高分

4. 定义方法: `start_game`, 开始游戏, 规则如下

1. 使用随机数获取本次游戏得分 范围 `(10 - 100)` 之间
2. 判断本次得分和最高分之间的关系
  - 如果本次得分比最高分高,
    - 修改最高分和获得最高分的玩家名字
  - 如果分数小于等于最高分,则不操作
3. 输出本次游戏得分

5. 主程序步骤

- 1, 查看游戏的帮助信息
- 2, 查看当前的最高分
- 3, 创建一个玩家小王,玩一局游戏, 输出当前最高分和玩家名字
- 4, 创建一个玩家小李,玩一局游戏, 输出当前最高分和玩家名字
- 5, 创建一个玩家小张,玩一局游戏, 输出当前最高分和玩家名字

```
import random

class Game:
    """游戏类"""
    # 直接在类内部,方法外部定义的变量,就是类属性
    top_score = 0 # 最高分
    top_score_player = '' # None 最高分玩家名字,

    # 想要定义实例属性,需要定义 init 方法,
    def __init__(self, name):
        self.player_name = name # 玩家的名字

    # 定义静态方法
    @staticmethod
    def show_help():
        print('这是游戏的帮助信息')

    # 定义类方法
    @classmethod
    def show_top_score(cls):
        if cls.top_score: # 数字中的 0 , 容器中的空容器 都是 False
            print(f"历史最高分玩家为: {cls.top_score_player}, 分数为: {cls.top_score}")
        else:
            print('还没有人开始游戏, 暂无最高分, 快去游戏吧')

    # 实例方法 直接书写
    def start_game(self):
        # 本次游戏得分
        score = random.randint(10, 100)
        print(f"{self.player_name} 开始了一局游戏,本次游戏得分为 {score}")
        # 判断本次游戏得分和历史最高分的关系
```

```

        if score > Game.top_score:
            # 修改历史最高分
            Game.top_score = score
            Game.top_score_player = self.player_name

if __name__ == '__main__':
    Game.show_help()
    Game.show_top_score()
    Game('小王').start_game()
    Game.show_top_score()
    Game('小李').start_game()
    Game.show_top_score()
    Game('小张').start_game()
    Game.show_top_score()

```

## 题目2

定义Animal动物类，具有 吃肉 `eat` 的方法，喝 `drink` 的方法，睡觉 `sleep` 的方法 定义 狗Dog 继承了动物类，具有汪汪叫 `bark` 的方法

- 狗 去调用吃的方法
- 狗 去调用喝的方法

```

class Animal:
    """动物类"""
    def eat(self):
        """吃方法"""
        print("动物都爱吃")

    def drink(self):
        """喝的方法"""
        print("动物可以喝水")

    def sleep(self):
        """睡觉的方法"""
        print("动物都喜欢睡觉")

class Dog(Animal):    # 继承 子类名(父类名)
    """子类 狗类"""
    def bark(self):
        """狗叫的方法"""
        print("狗汪汪叫的方法")

# 创建子类对象
dog = Dog()
dog.bark()

dog.eat()    # 子类对象可以调用父类的方法

dog.drink()

```

## 题目 3

1.按如下要求完成代码的编写：

- 1). 定义一个Person类，包含姓名和年龄属性，要求姓名是公有属性，年龄是私有属性
- 2). 提供获取私有属性的公有方法 get\_age方法
- 3). 提供可以设置私有属性的set\_age方法，要求如果输入的年龄在 0--120 之间，设置年龄，否则提示输入不正确
- 4). 重写 `__str__` 要求打印对象时，把姓名和年龄都打印出来

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.__age = age # 私有属性

    def get_age(self):
        """获取年龄"""
        return self.__age

    def set_age(self, new_age):
        """设置年龄"""
        if new_age >= 0 and new_age <= 120:
            self.__age = new_age
        else:
            print("输入的年龄有误")

    def __str__(self):
        return f"姓名:{self.name}, 年龄: {self.__age}"

xw = Person('小王', 18)
print(xw)
```

## 题目4

1. 使用代码在test.txt文件中写入"wow,so beautiful!".
2. 使用 代码读取 `test.txt` 文件中的内容 并打印

```
# 写文件
# 1. 打开文件
f = open('test.txt', 'w', encoding='utf-8')
# 2. 写文件
f.write('wow, so beautiful!')
# 3. 关闭文件
f.close()

# 读文件
```

```
# 1. 打开文件
f = open('test.txt', encoding='utf-8')
# 2. 读文件
data = f.read()
print(data)
# 3. 关闭文件
f.close()
```

## 题目 5

1. 使用代码向文件 `a.txt` 中写入以下内容

```
张三,李四,王五,赵六
```

2. 使用代码读取 `a.txt` 文件中的内容,使用字符串的方法,将每个名字存入列表 `my_list`, 打印列表

```
# 向文件中写入内容
f = open('a.txt', 'w', encoding='utf-8')
f.write('张三,李四,王五,赵六')
f.close()

# 读取文件中的内容,并转换为列表
f = open('a.txt', 'r', encoding='utf-8')
buf = f.read() # str
my_list = buf.split(',') # []
f.close()
print(my_list)
```

## 题目 6

1. 将如下内容 存入b.txt 文件中

```
1,2,3,4,5,6,7
```

2. 读取文件中的内容,并对这些数字进行求和计算

```
# 1,2,3,4,5,6,7

def write_data():
    f = open('b.txt', 'w', encoding='utf-8')
    f.write('1,2,3,4,5,6,7')
    f.close()

def read_data():
    f = open('b.txt', 'r', encoding='utf-8')
    data = f.read() # '1,2,3,4,5,6,7'
    # print(data)
```

```
f.close()
data_list = data.split(',') # ['1' , '2' , '3' , '4' , '5' , '6' , '7']
print(data_list)
return data_list

data_list = read_data()
num = 0
for i in data_list:
    num += int(i)
print(num)
# write_data()
```