

实验五 面向对象程序设计

实验目的

1. 掌握 Python 中类的定义方法；
2. 掌握 Python 中对象的概念和使用方法；
3. 掌握类的封装和数据隐藏技术；

实验内容

1. 为电子商务网站编写一个商品类 **Product**。该类包括数据成员：商品编号、商品类别、商品名称、价格；包括方法：构造函数、为数据成员赋值的方法、获取数据成员数值的方法、显示商品信息的方法等。

【提示】考虑数据隐藏

2. Design a class named **QuadraticEquation** for a quadratic equation $ax^2 + bx + c = 0$. The class contains:

- ✓ The private data fields a,b, and c that represent three coefficients.
- ✓ A constructor for the arguments for a, b, and c.
- ✓ Three get methods for a, b, and c.
- ✓ A method name `getDiscriminant()` that returns the discriminant, which is $b^2 - 4ac$
- ✓ The methods named `getRoot1()` and `getRoot2()` for returning the two roots of the equation using the following formula:

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

These methods are useful only if the discriminant is nonnegative. Let these methods return 0 if the discriminant is negative.

Write a test program the prompts the user to enter values or a, b, and c and displays the result based on the discriminant. If the discriminant is positive, display two roots.

If the discriminant is 0, display one root. Otherwise, display *The equation has no real roots*.

3. 定义一个 `Identifier` 类，具有一个属性 `id`，存放身份证号，以及如下方法：

(1) 构造函数

(2) `isValid()`方法用于判断身份证号码是否有效；

➤ 身份证号码为 18 位

➤ 一个合法的身份证号码由 17 位地区、日期编号和顺序编号加 1 位校验码组成。校验码的计算规则如下：

首先对前 17 位数字加权求和，权重分配为：{7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2}；然后将计算的和对 11 取模得到值 Z；最后按照以下关系对应 Z 值与校验码 M 的值：

Z: 0 1 2 3 4 5 6 7 8 9 10

M: 1 0 X 9 8 7 6 5 4 3 2

(3) `getBirth()`方法用于从身份证号码中提取出生年份；

(4) `dispInfo()`方法用于输出身份证号码。

【提示】考虑数据隐藏

4. 以下代码先定义了一个 `Student` 类，其数据域包含学生姓名和高数、英语、C 语言三科成绩，方法 `cal_mean` 计算并返回学生的平均分，方法 `get_c_score` 获取 C 语言成绩，方法 `display_info` 返回学生信息。然后，测试此类，输出所有学生信息；找出平均分最高的学生；按 C 语言成绩按高到低排序，并输出。请为程序填空。

```
class Student:
    def __init__(self, name, math_score, english_score, c_score):
        self.name = name
        self.scores = {
            'math': math_score,
            'english': english_score,
```

```

        'c_lan': c_score
    }

    def cal_mean(self):
        """计算学生的平均分"""
        return _____ (1)

    def get_c_score(self):
        """获取 C 语言成绩"""
        return self.scores['c_lan']

    def display_info(self):
        mean_score = self.cal_mean()
        return (f"姓名: {self.name} | "
                f"高数: {self.scores['math']} | "
                f"英语: {self.scores['english']} | "
                f"C 语言: {self.scores['c_lan']} | "
                f"平均分: {mean_score:.2f}")

if __name__ == "__main__":
    # 创建学生对象
    students = [
        Student("张三", 85, 92, 78),
        Student("李四", 88, 95, 82),
        Student("王五", 92, 89, 95),
        Student("赵六", 76, 82, 79),
        Student("钱七", 90, 87, 91)
    ]
    print("=== 所有学生信息 ===")
    for student in students:
        print(student.display_info())

```

```
print("\n=== 平均分最高的学生 ===")
highest_mean = max(____(2)____ , ____ (3) ____ )
print(highest_mean.display_info())

print("\n=== 按 C 语言成绩从高到低排序 ===")
# 按 C 语言成绩从高到低排序
sorted_students = ____ (4) ____ (____ (5) ____ , ____ (6) ____ , ____ (7) ____ )

for i in range(len(____ (8) ____)):
    student = ____ (9) ____
    print(f" {i+1} . {____ (10) ____}")
```