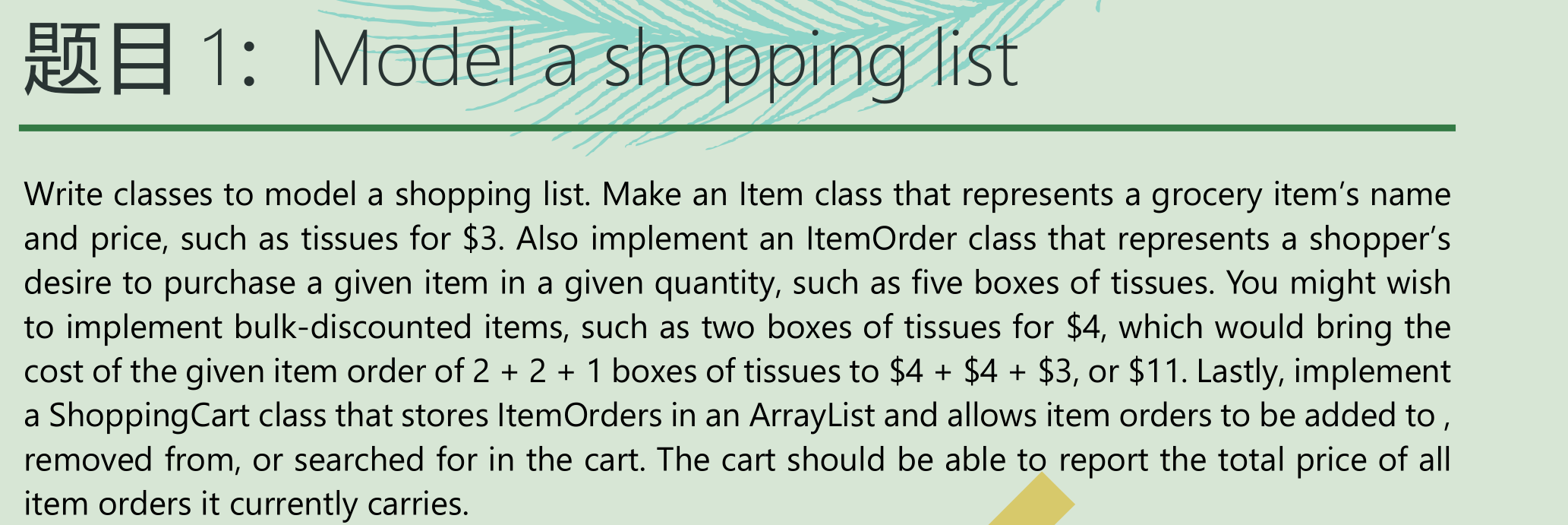
|  |  |
| --- | --- |
| **面向对象程序设计 实验报告** | |
| 第 4 次 | |
| 图片包含 标牌  已生成极高可信度的说明 | |
|  | |
|  | |
|  | |
| **姓名** | 郑新宇 |
| **班级** | 软件91班 |
| **学号** | 2191312205 |
| **电话** | 15600782016 |
| **Email** | 1370909470@qq.com |
| **日期** | 2019-08-15 |

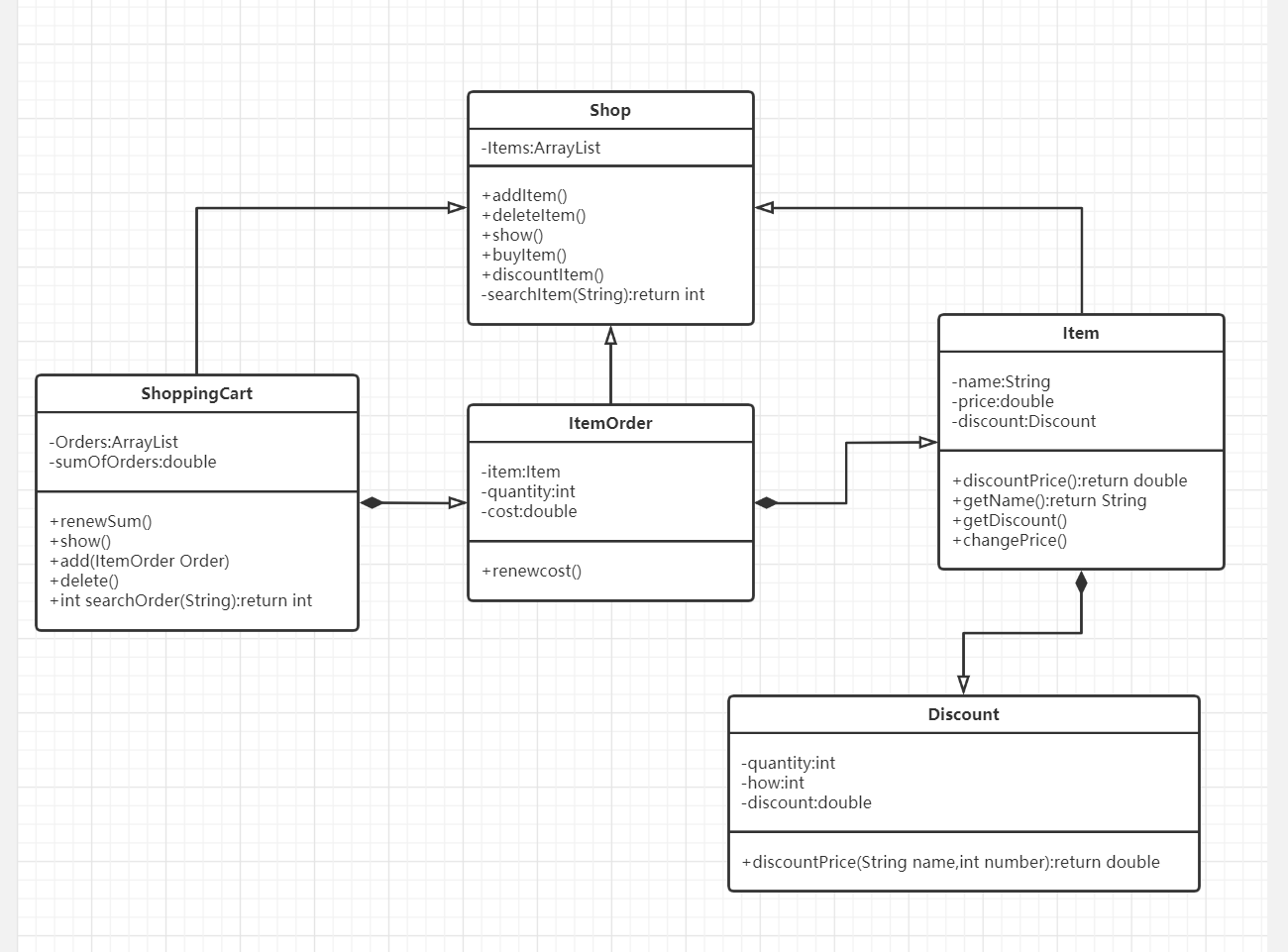
目录

实验1

1. 题目

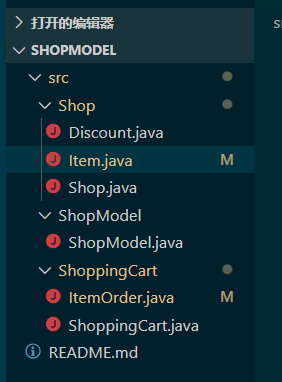


1. 设计过程（包含类的设计以及类和类之间的关系描述）



1. 主干代码说明

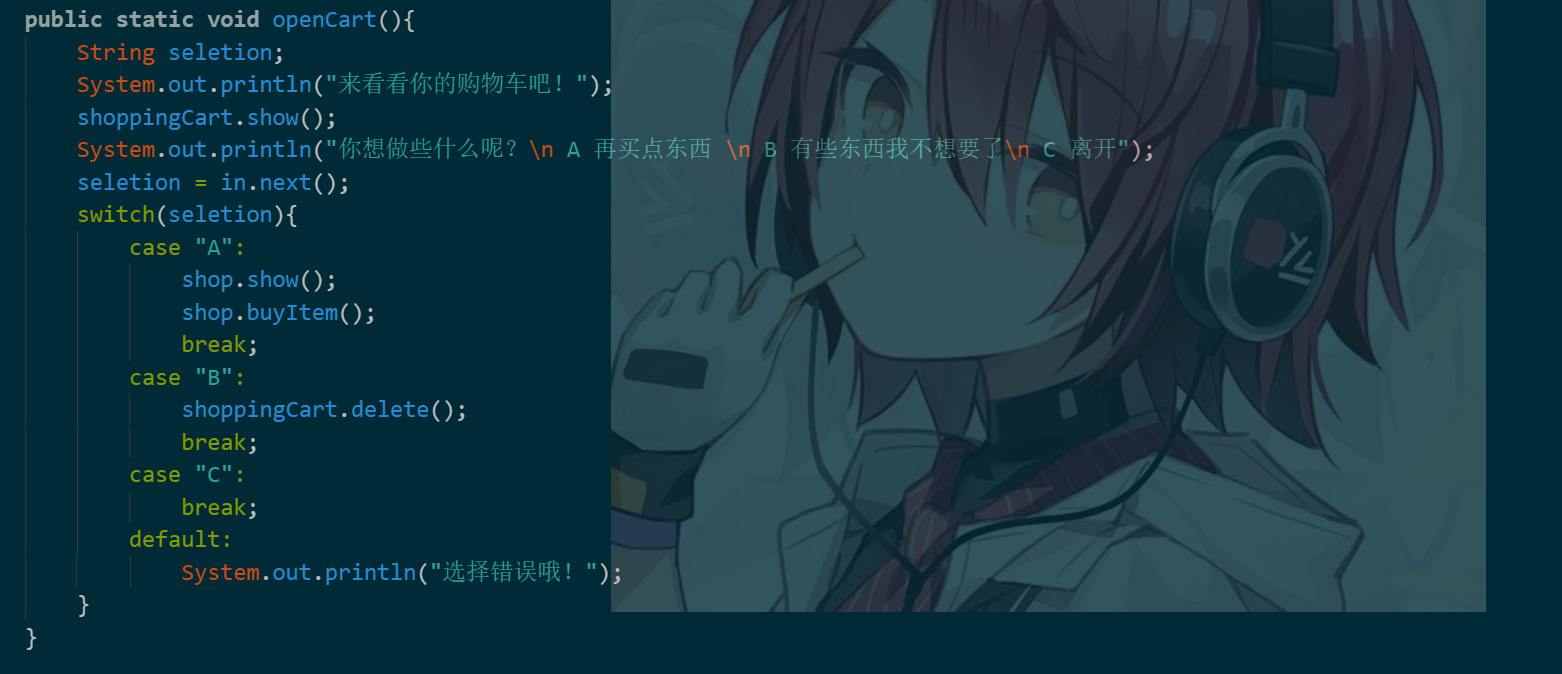
整体目录



ShopModel(主函数)







Discount.java



Item.java



Shop.java





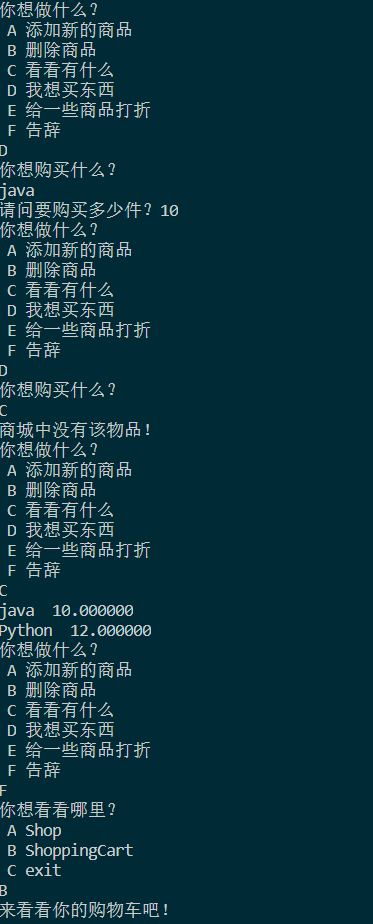
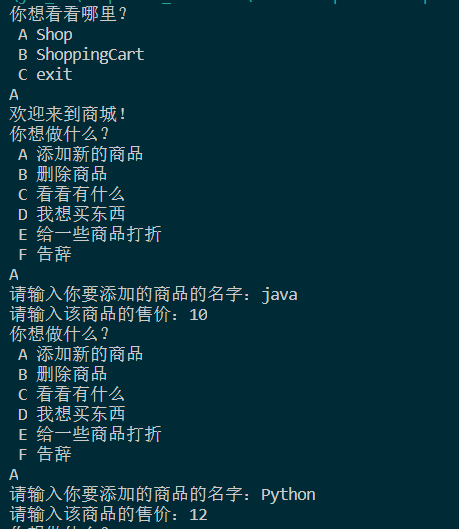
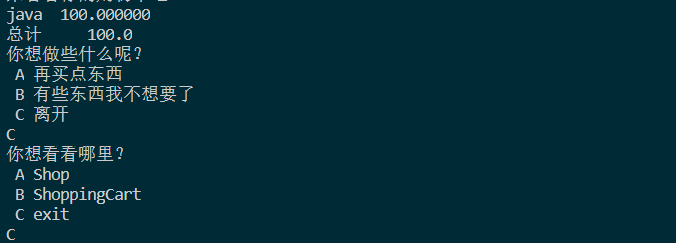
ItemOrder.java



\

ShoppingCart.java

1. 运行结果展示

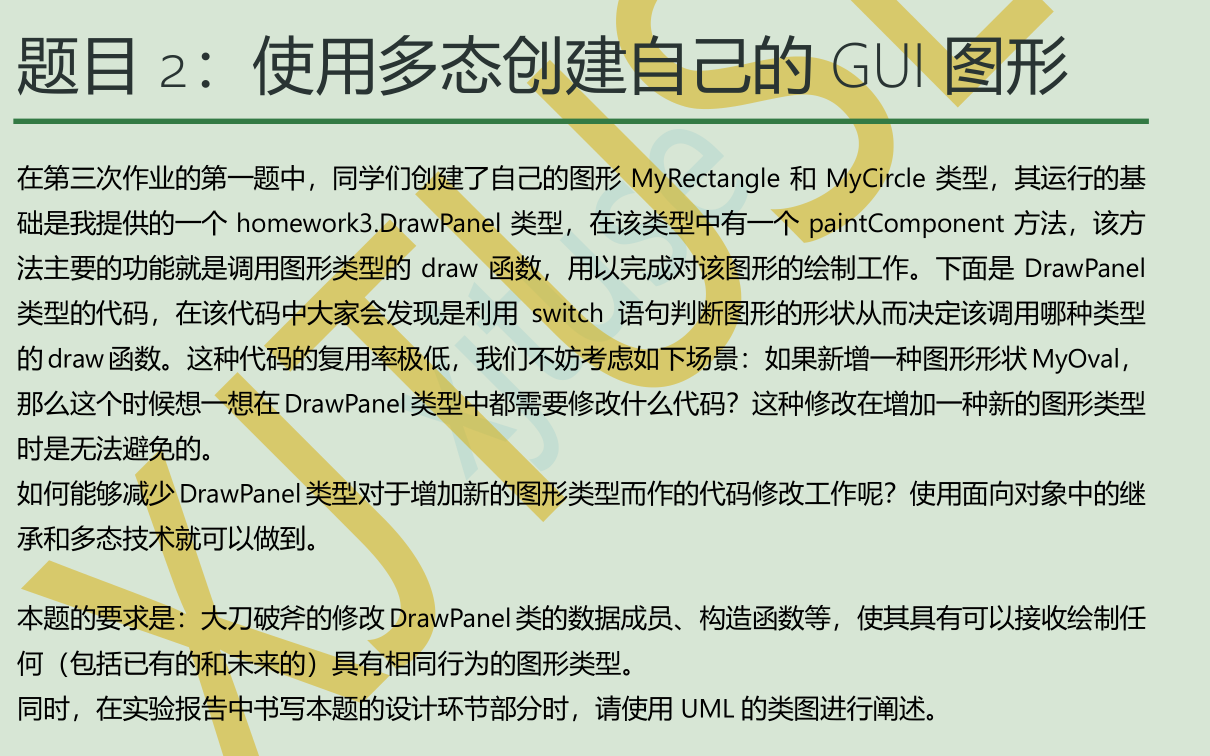
1. 总结和收获

学会了面向对象编程时的思考方式，从哪里入手去思考

学会了UML图的绘制，并了解其作用

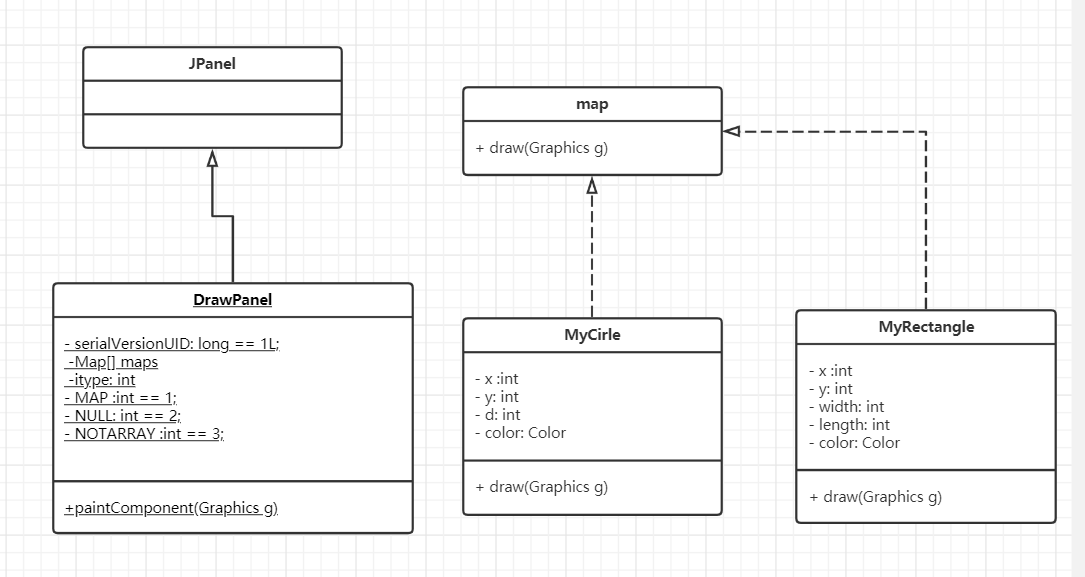
实验2

1. 题目



1. 设计过程（包含类的设计以及类和类之间的关系描述）

同一个函数，执行相同函数操作不同的类，使用接口较为方便



1. 主干代码说明

DrawPanel.java

package homework3;

import javax**.**swing**.**JPanel;

import java**.**awt**.**Color;

import java**.**awt**.**Graphics;

import Map**.**Map;

**public** **class** DrawPanel **extends** JPanel{

*/\*\**

*\**

*\*/*

**private** **static** **final** **long** serialVersionUID = 1L;

**private** Map[] maps;

**private** **int** type;

*// private static final int NONE = 0;*

**private** **static** **final** **int** MAP = 1;

**private** **static** **final** **int** NULL = 2;

**private** **static** **final** **int** NOTARRAY = 3;

**public** DrawPanel(Map[] maps){

        setBackground(Color.WHITE);

        if( maps == null ){

            type = DrawPanel.NULL;

            return ;

        }else if(!maps.getClass().isArray()){

            type = DrawPanel.NOTARRAY;

            return;

        }

        this.maps = maps;

        type = DrawPanel.MAP;

    }

**public** DrawPanel() {

        setBackground(Color.BLACK);

        this.type = 0;

      }

**public** **void** paintComponent(Graphics g){

        super.paintComponent(g);

        switch(this.type){

            case DrawPanel.MAP:

                for(Map map:maps)

                    map.draw(g);

                break;

            case DrawPanel.NULL:

                g.drawString("在DrawPanel的构造函数中，你传递的引用参数是NULL!",50,50);

                break;

            case DrawPanel.NOTARRAY:

                g.drawString("在DrawPranel的构造函数中，你传递的参数必须是某个形状的数组类型", 50, 50);

                break;

        }

    }

}

Map.java

package Map;

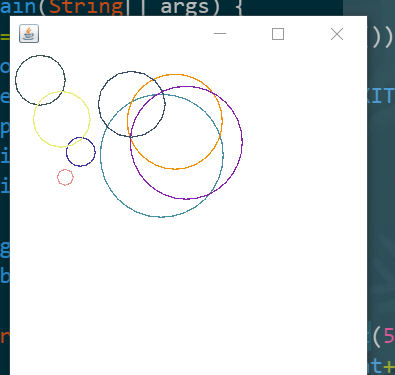
import java**.**awt**.**Graphics;

**public** **interface** Map {

**public** **void** draw(Graphics g);

}

1. 运行结果展示

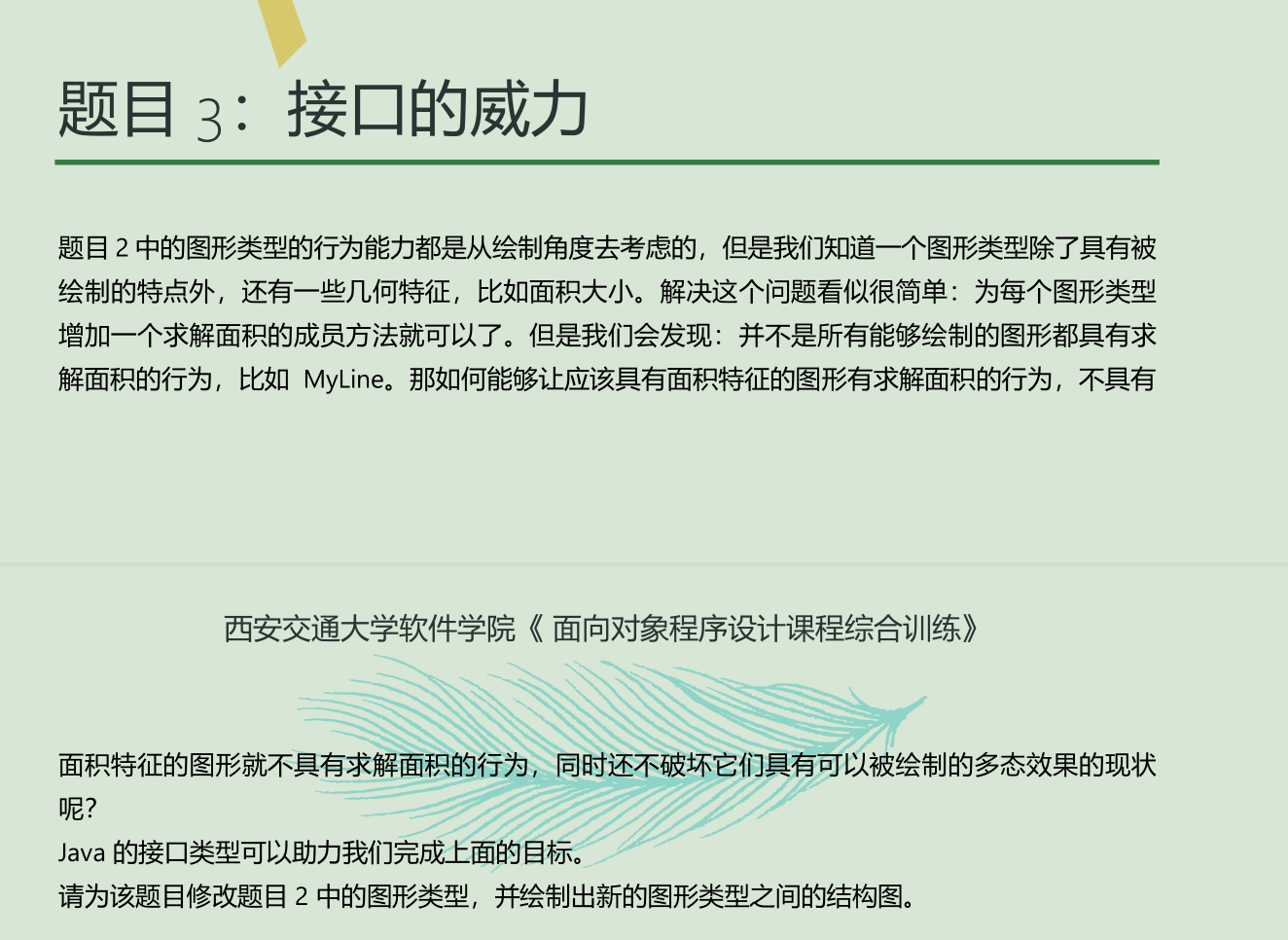


1. 总结和收获

理解了接口的使用方法以及规范，明白了接口出现的意义

实验3

题目



设计过程（包含类的设计以及类和类之间的关系描述）

同一个函数，执行相同函数操作不同的类，使用接口较为方便

主干代码说明

运行结果展示

总结和收获