

《Java 程序设计》

教师评价：

☐ 算法/实验过程正确； ☐ 源程序/实验内容提交 ☐ 程序结构/实验步骤合理；

☐ 实验结果正确； ☐ 语法、语义正确； ☐ 报告规范；

其他：

评价教师签名：

一、实验目的

运用面向对象程序设计思想，基于 java 的 GUI、套接字和多线程机制，实现基绘图板程序。

二、实验项目内容

1. 借鉴 作业 2 的类的架构构建绘图类集合，shape、shape 类的派生类和 graphics 类；

2. 将作业 2 的 OpenGLApp 改为 JFrame 类的派生类，包含 Graphics 类实例作为数据成员，利用特定的布局管理器（layout）构建主窗口，在窗口中创建相应的控件允许用户选择当前绘制图形的形状、线条颜色、填充颜色等；

3. 添加相应的事件监听和相应方法（处理用户的输入），例如绘制图形的形状、线条颜色、填充颜色的选择变化；

4. 重载主窗口的 paintComponent（Graphics g）方法，用于在主窗口中绘制 Graphics 类实例的图形数据，绘制过程中调用这些图形数据自己的 draw() 方法，可以把主窗口的 g 传给每个图形元素类的 draw 方法；

5. 在主窗口中注册的 MouseListener 接口中相关处理方法，根据当前的用户选择实现类似 window 附件中 painter 工具的功能，例如：绘制指定图形和移动图形。

6. 利用 sock、多线程机制，修改程序结构实现一个基于 java sock 的网络白板程序（多个用户协同白板绘图程序）的设计与实现。

报告创建时间：

三、实验过程或算法（参照 MVC 模式进行类的设计，程序实现流程设计和算法设计）

基于 MVC 的 GUI 实现绘图程序的设计和实现，借助于前面已经实现的 MVC 绘图框架设计实现功能向 GUI 图形化界面的封装。

MVC 设计：经典 MVC 模式中，M 是指业务模型，V 是指用户界面，C 则是控制器，使用 MVC 的目的是将 M 和 V 的实现代码分离，从而使同一个程序可以使用不同的表现形式。其中，View 的定义比较清晰，就是用户界面。

设计了两个包 App 和 Graph，App 包只有主类 OpenGLApp 类，Graph 包含有 Graphic、Cube、custom_draw、Point、Rectangle、Shape、Triangle 类分别实现不同的功能，其中 Graph 包里的 Shape 类是 Rectangle、Cube、Triangle 三个类的抽象父类，有他们需要实现的 draw 方法，Point 类为点集合类，用户为图形化对象提供点对象。

设计了 OpenGLApp 类为程序主入口，主要用于存储全局变量和定义图形化窗口，布局窗口和添加相应的辅助操作入口，如添加一个帮助按钮，为按钮添加监听，用户可以通过点击帮助按钮查看帮助文档，实现对用户的指导似乎用方法。在主类中要添加各个功能的入口，且要具有选择性，让用户自主选择需要的功能。

本设计功能主要包括两大块：1、用户可能需要存储一些图形数据而暂时不需要查看图形，那么这时候就是要给用户提供存储功能。2、用户有时候需要查看相应图形的样式以加深对图形的理解和加深记忆。为解决这两大问题，所以就在主类提供了存储信息入口和自定义绘图入口，以满足用户需求。

为体现可选择性充分对用户友好，提供了选择命令输入框，用户可以输入对应的命令获取相应的服务，并添加了处理错误命令的能力以增强程序的强壮性和准确性。

加入了图形信息查询功能，因为要存储的信息不知道长度，所以选用 ArrayList 来存储存储信息，因为这相当于数据结构中介绍的线性链表，长度可变，可认为长度无限长，可以满足对信息的存储需求。如果用户需要查看存储的图形信息，那么就在面板设置文本域，用文本的形式将所有图形信息显式的展示到界面上以供用户查看，同时为面板布局了滚动条，避免了文本过多而无法完全显示的问题。

布局确定按钮，用于触发进一步的图形展示功能，离开首页，进入主要功能界面，离开主类。

进入图形类 Graphic 类，Graphic 类的功能主要就是处理从主类过来的命令，以选择特定的功能进行活动，以及实现将得到的对象实例加入到对象集合中存储下来，便于用户查看图形信息，这个类的功能就是分辨命令并提供相应的服务以及具有一定的纠错能力，可以对用户输入的指令作出相应的判断和处理。值得注意的是，Graphic 类与主类 OpenGLApp 类不在一个包里面，因

为 Graphic 等类要实现对主类对象的处理，所以要通过导入 App 包实现对主类 OpenGLApp 类的对象访问。

图形实例化类 Cube、Rectangle、Triangle 三个类的模型是相同的，功能也大致相同，只是实例化的对象不一样，图形不一样，所以放到一起进行讲解分析。以 Cube 类为例，通过 Graphic 类选择相应的命令并处理之后进入模块，如进入 Cube 类，在界面右方添加了信息输入框可以输入相应的图形信息，如棱长、一个点的坐标(x,y,z)，添加确定按钮并添加监听，当点击按钮的时候会把图形信息加入到 ArrayList 中保存起来方便后期对信息的访问查看。同时在界面左侧设置了样图显示框，将样例图显示出来，给用户一个直观的图形化演示加深用户对图形的直观感受。添加一个返回首页按钮，并添加事件监听，当用户点击的时候可以快速返回首页。

进入 Rectangle、Triangle 类的功能与 Cube 类相似，在此不再赘述，值得一提的是他们都有样例展示，信息输入和确定添加以及快速返回首页入口。还有为了避免用户不输入而直接添加信息，此时信息为空，所以设置信息默认值为 1，以填补用户不输入而直接添加的错误操作。

针对所有需要输入信息或命令的文本框，因为军设置了默认值，为便于用户输入信息，为文本框添加了点击监听，获取焦点当点击时就自动全选文本框内容以便于用户输入，添加的时间监听为：

```
public class myFocusListener extends FocusAdapter {  
    public void focusGained(FocusEvent e) {  
        JTextField c=(JTextField)e.getSource();  
        c.setSelectionStart(0);  
        c.setSelectionEnd(c.getText().length());  
    }  
}
```

继承了 FocusAdapter 类来实现相关功能，值得一提的是使用 FocusAdapter 和 FocusListener 类的不同之处，后者需要实现所有相关方法包括不需要的方法，前者只需要添加实现需要的方法从而减轻代码编写者的负担而可以集中精力于算法分析。

最后就是自定义画板类 custom_draw 类，此类用于给用户自定义画图服务，同样的里面还具有三角形、矩形、立方体的绘图功能，同时急啊如了自定义绘图模块，就是用户可以根据自己的需求画出任意图形（自定义画板），添加“清空画板”按钮并添加监听，当点击时清空画板内容满足用户的重回需求。同样的添加了返回首页的快速按钮，可直接返回命令首页。

当选择三角形、矩形、立方体按钮的时候，可以分别绘制相应的图形，其中为了避免混乱，面板只允许绘制相同的图形，选择其他命令的时候自动清空面板。选择这三个命令的时候为鼠标添加了点击监听，当选择这三个命令的其中一个之后，将鼠标在绘图区点击一点时，自动获取该点坐标，并以改点为顶点绘制一个定的该图形，目的在于为用户提供直观显示服务。

自定义画板就是满足用户自己需要画出的图形，而不局限于特定图形，是个自由的画图区，也是绘图功能的巅峰功能。

为了给用户选择图形颜色的服务，添加了选择颜色按钮以及添加了监听，当用户点击该按钮的时候会更改线条颜色，并在按钮显示颜色名以给用户直观的信息，告知用户当前颜色种类。

导航栏，界面上方和甲方填充了导航栏，每次进入不同的界面，导航栏都会显示相应的提示文字以告知用户简洁的信息，是一个友好的设置，其中上方主要显示当前位置，下方主要显示一些注意的点或者是操作的简要描述。

自此，图形化绘图程序的介绍基本完毕，值得注意的是，部分功能以及网络功能暂未实现，后期会填充进去，敬请期待。

四、实验结果展示及分析和（或）源程序调试过程

问题：Graphic 类继承主类 OpenGLApp 时对主类成员的修改无效

分析及解决办法：因为子类规定是没有覆盖父类成员的权限的，子类继承父类，那么就具有和父类相同的规模和成员并可以添加新成员或方法，简单来说，父类到子类通过继承实现的就是传值继承，是彼此同名但分离的，不具备传引用的特性。

解决办法：将父类成员赋值给子类成员，通过这样实现覆盖和重写，即是简介改值的方法实现传引用的特性实现，如 Graphic 类中：

```
public Graphic(JPanel panel,JLabel label2,JLabel label3,
               ArrayList<Triangle> list1,ArrayList<Rectangle>
list2,ArrayList<Cube> list3) {
    this.panel=panel;
    this.label2=label2;//子类不能直接修改父类的成员域,要搭建链接
才能直接修改，否则修改的对象是一个值，即子类自己的成员变量
    this.label3=label3;
    this.list1=list1;
    this.list2=list2;
    this.list3=list3;
}
```

实现了对父类对象的链接，可以在子类更改父类成员。

问题：不能很好的使用 Graphics 画笔

分析及解决办法：原来是要继承需要绘画的对象类，并在其中重写 paint（）方法，才能正确使用，如 Cube 类中：

```

}
public class drawpanel extends JPanel{
    private static final long serialVersionUID = 1L;

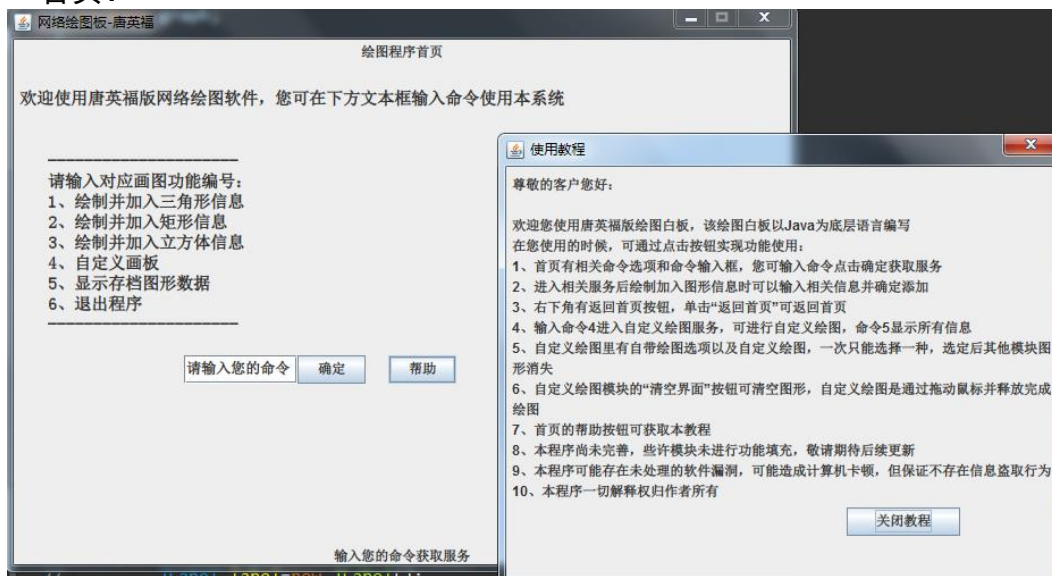
    public void paint(Graphics pen) {
        super.paint(pen);pen.setColor(Color.red);
        pen.drawRect(80, 80, 200, 200);
        pen.drawRect(40, 140, 200, 200);
        pen.drawLine(80, 80, 40, 140);
        pen.drawLine(280, 80, 240, 140);
        pen.drawLine(40, 340, 80, 280);
        pen.drawLine(240, 340, 280, 280);
    }
}
}

```

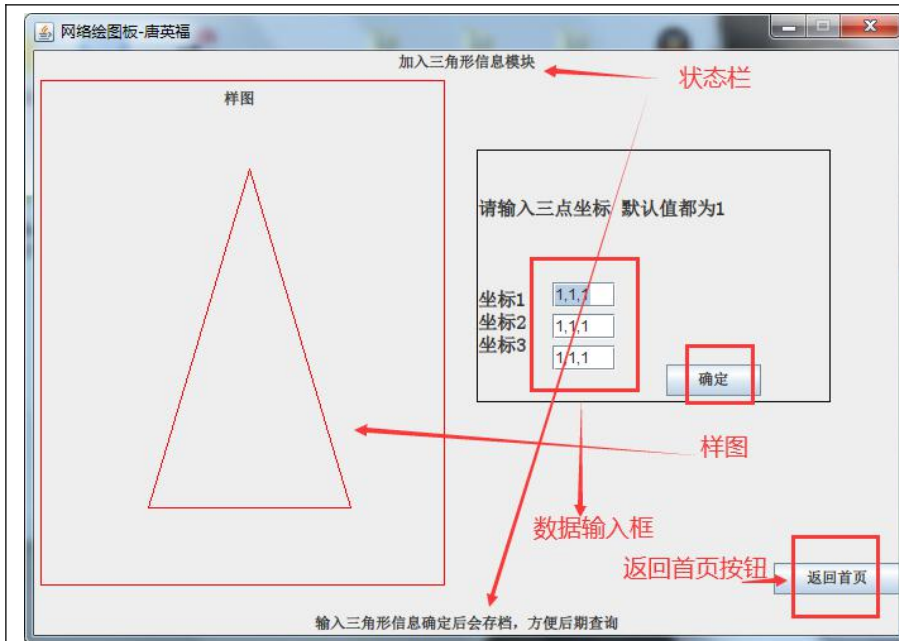
继承 JPanel 类实现在 JPanel 上画立方体。

结果展示：

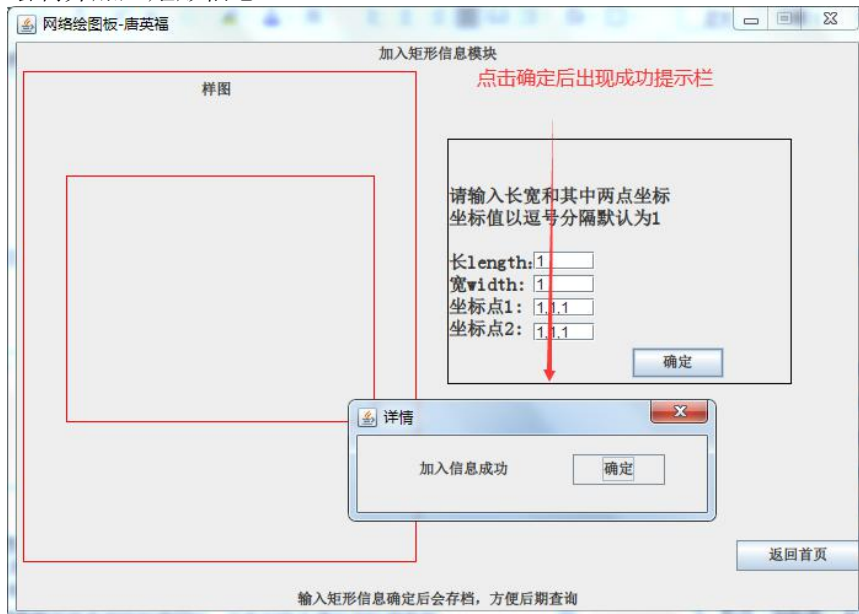
首页：



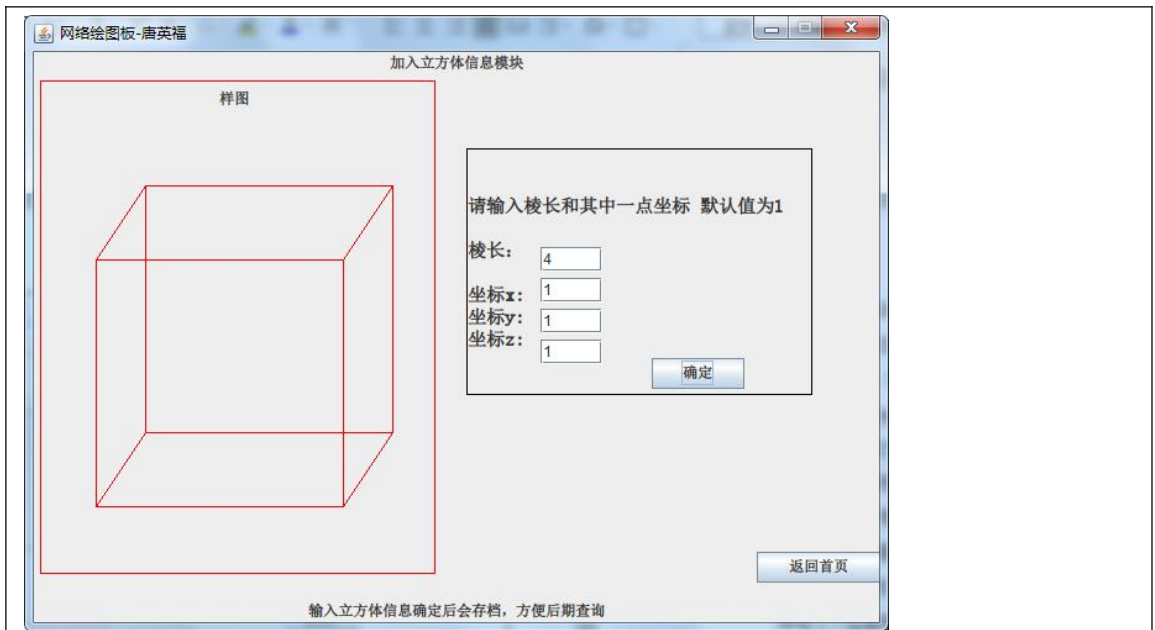
绘制并加入三角形模块：



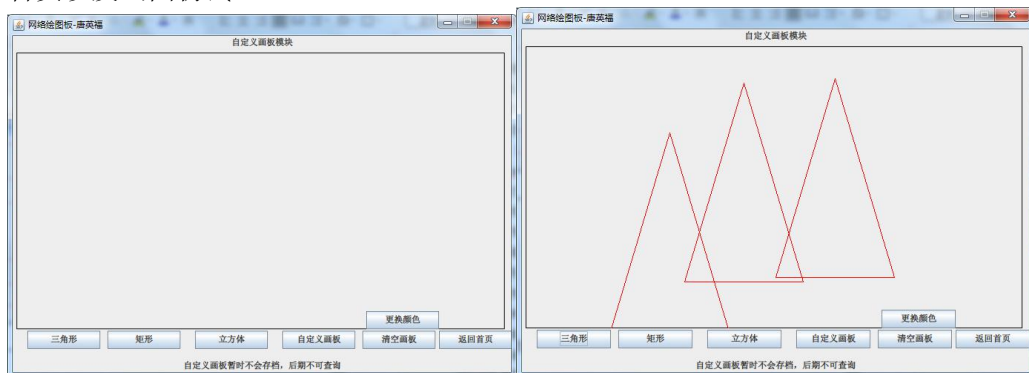
绘制并加入矩形信息:



绘制并加入立方体信息:



自定义绘图板:
首页以及画图初试:



自定义画图以及更改颜色:

