西北大学

面向对象程序设计（Java）

上机实验报告

学号：\_\_\_\_\_\_\_\_\_\_

班级：\_\_\_ \_\_\_\_\_\_\_\_\_

姓名：\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

日期：\_\_\_\_\_\_\_\_\_\_

一、需求分析

问题为用JavaFX设计一个图形用户界面面板程序。程序需要接收用户的输入从广义上来讲有用户选择的功能、选择图形的属性（包括大小、颜色等）、图形创建在画板上的位置等，以上操作均由鼠标和键盘进行输入，除此之外，程序的输入可能还来自文件。程序的输出为图形用户界面的显示以及程序数据保存在文件中。

二、详细设计

（请使用方程组、流程图等表示问题模型和流程，并使用文字说明）

首先是设计类，考虑到面板大致分为五个部分，上部为含有保存、导出等功能的菜单区、中间为画板主体、左部为画笔、形状等的功能区、右部为选择图形后的显示属性的属性区、下部则为鼠标的状态区，主要显示鼠标的位置以及所画图形的编号。

如图1，画板界面的各个类关系如下，MainPane主要负责画板五个部分的布局设计，其余类则完成其相关功能，这降低了程序类之间的耦合性。

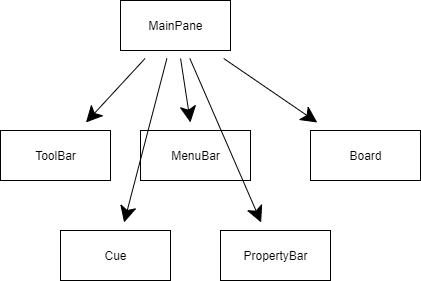


图 1 Board主体设计

大体设计完之后细化小点，分别设计每个类，关联每个类之间的关系，如下图为GUI所需的各个类的UML图，其中所有类都围绕着MainPane类所衍生。

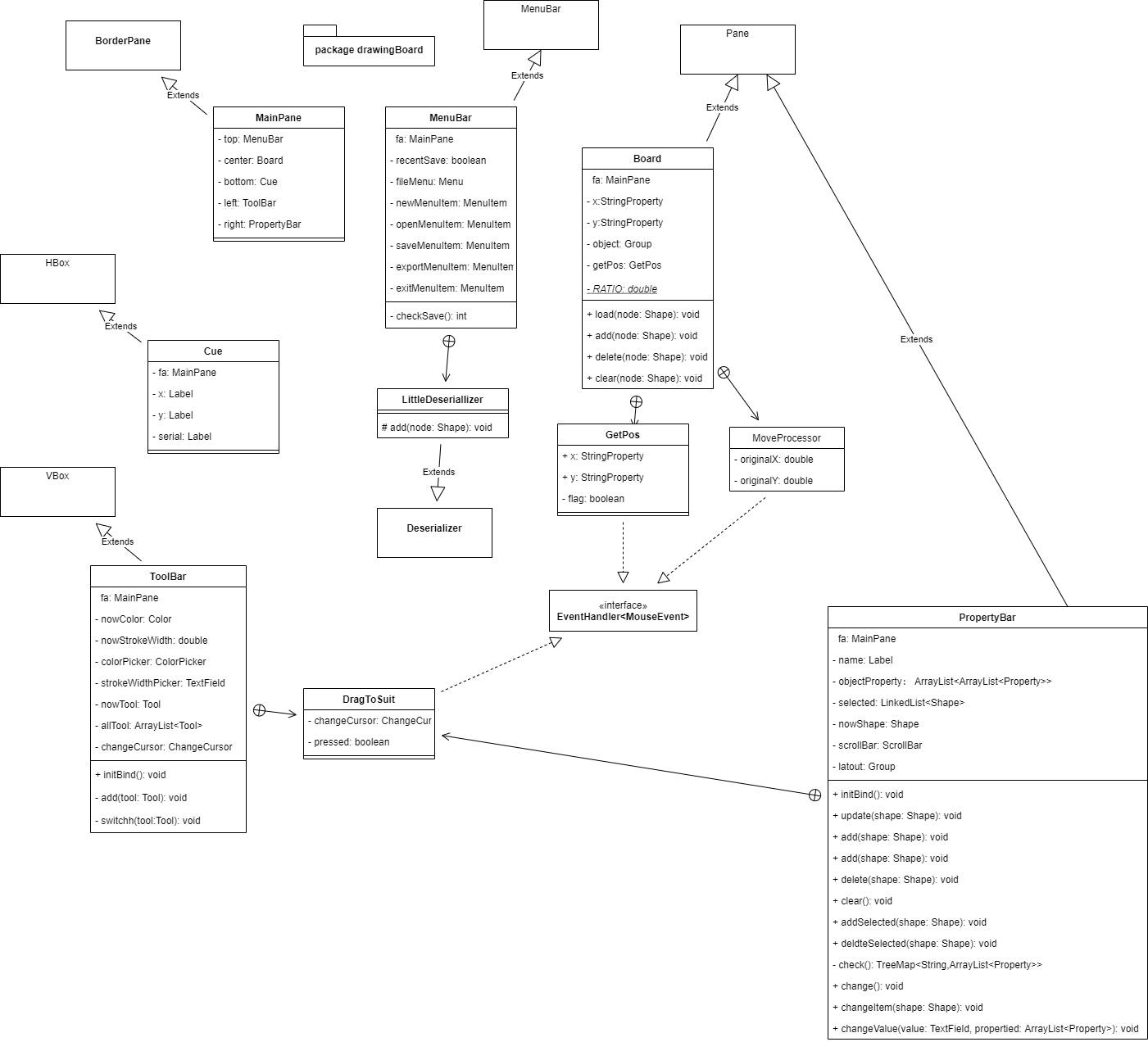


图 2 GUI类总览

在设计以上类时需要一些工具类，以及把其中共有的属性或者方法抽象为类或者接口，这里我设计了Geometry类来对绘画的Shape类提供静态方法。如图3为Geometry类的具体设计。

图 3 Geometry类

在形状的绘制中会有鼠标的动作，这里我把鼠标的press、release、drag动作抽象为Tool接口，具体设计如图4

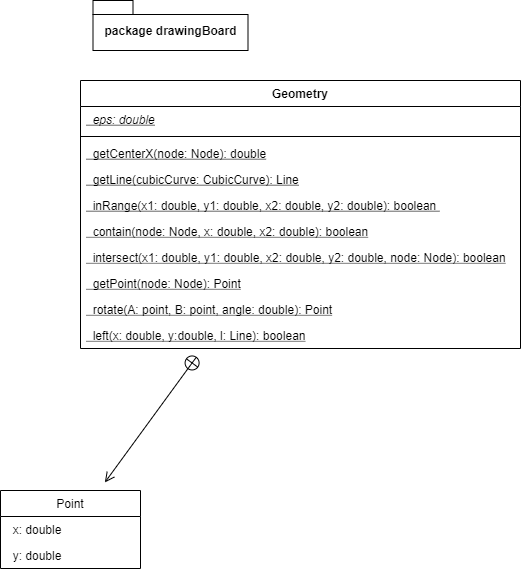


图 4 Geometry类

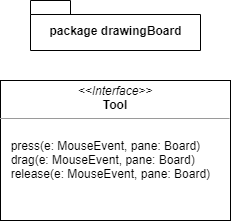


图 5 Tool接口

最后在画板保存时会牵扯文件保存的功能，这里自行设计文件结构，实现对象数据的持久化，方便随时重复读写，这里就涉及到对象的序列化与反序列化。这里主要有Deserializer与SyntaxError类。

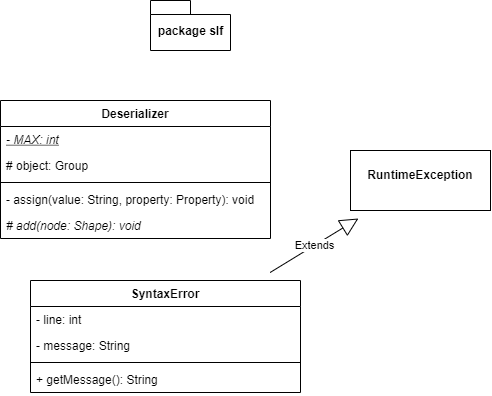


图 6 序列化对象类

文件内容大致如下：

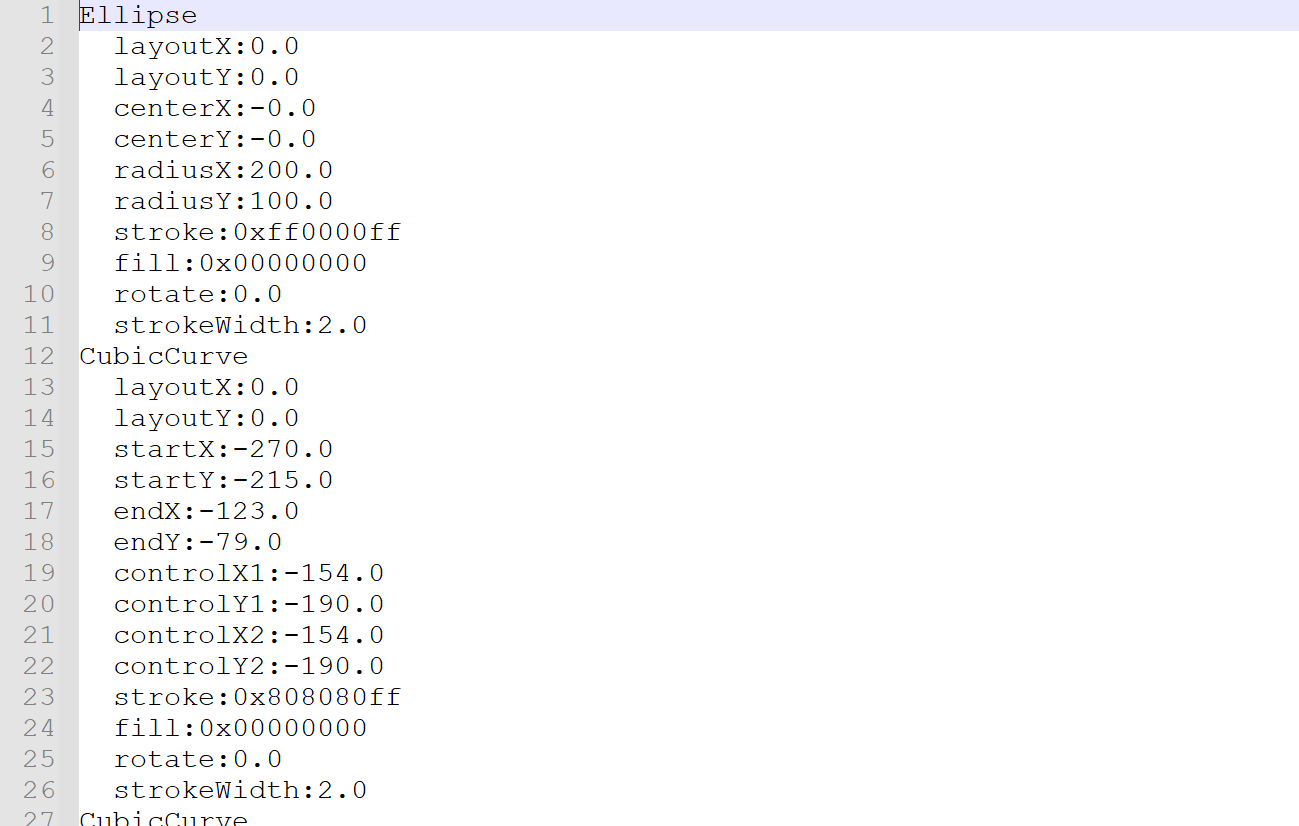


图 7 slf文件结构

对每个要绘制的形状也进行了定制化，分别实现了Tool接口。

三、交互界面设计

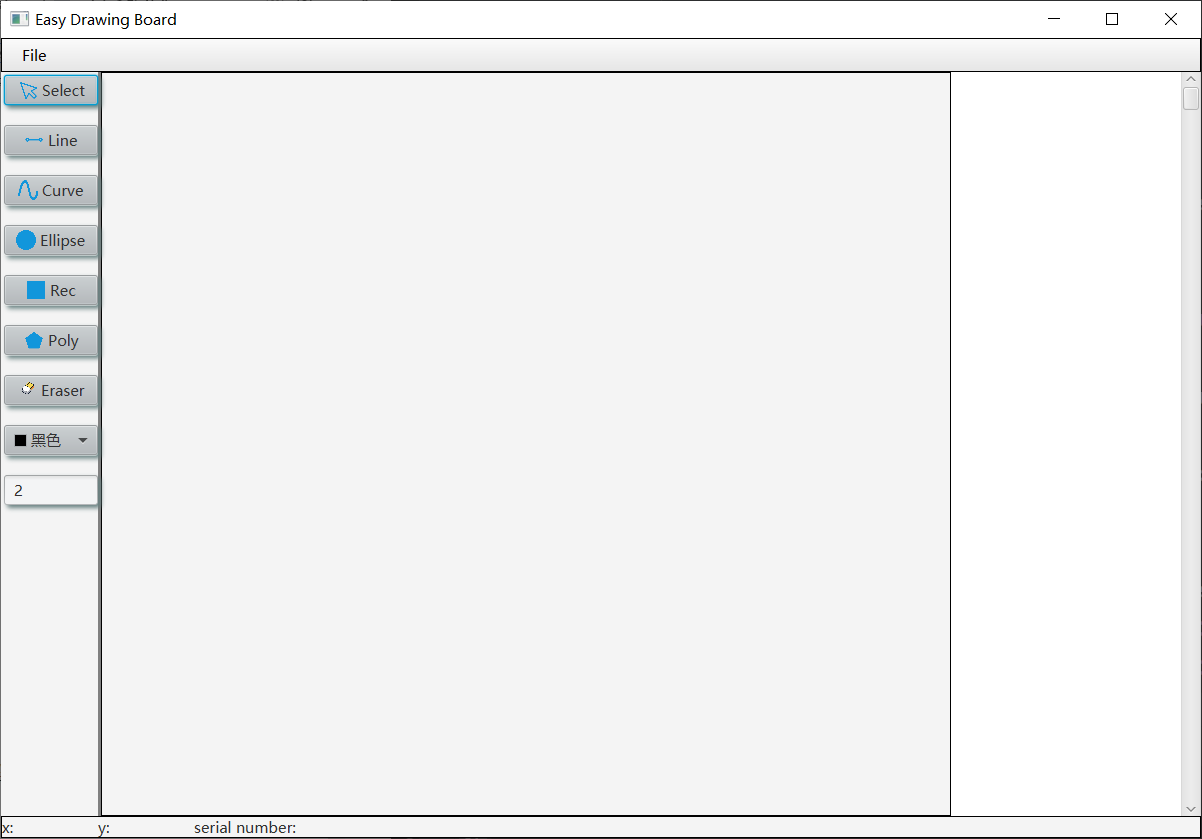


图 8 画板主页面

在绘制一个图形后右侧会限制这个形状的参数，以便修改形状的各个属性，底部显示鼠标在形状中的坐标信息与形状的编号，左侧则为形状的选择。

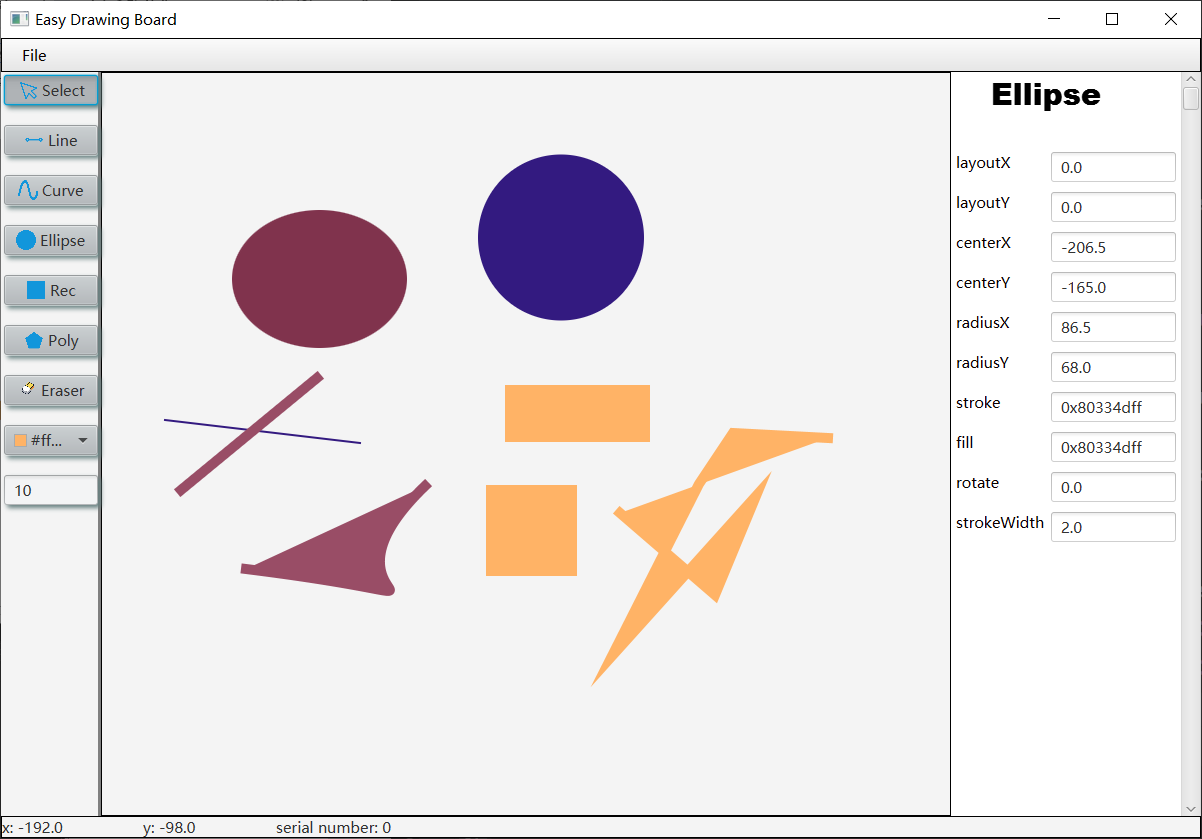


图 9 绘制椭圆后的画板

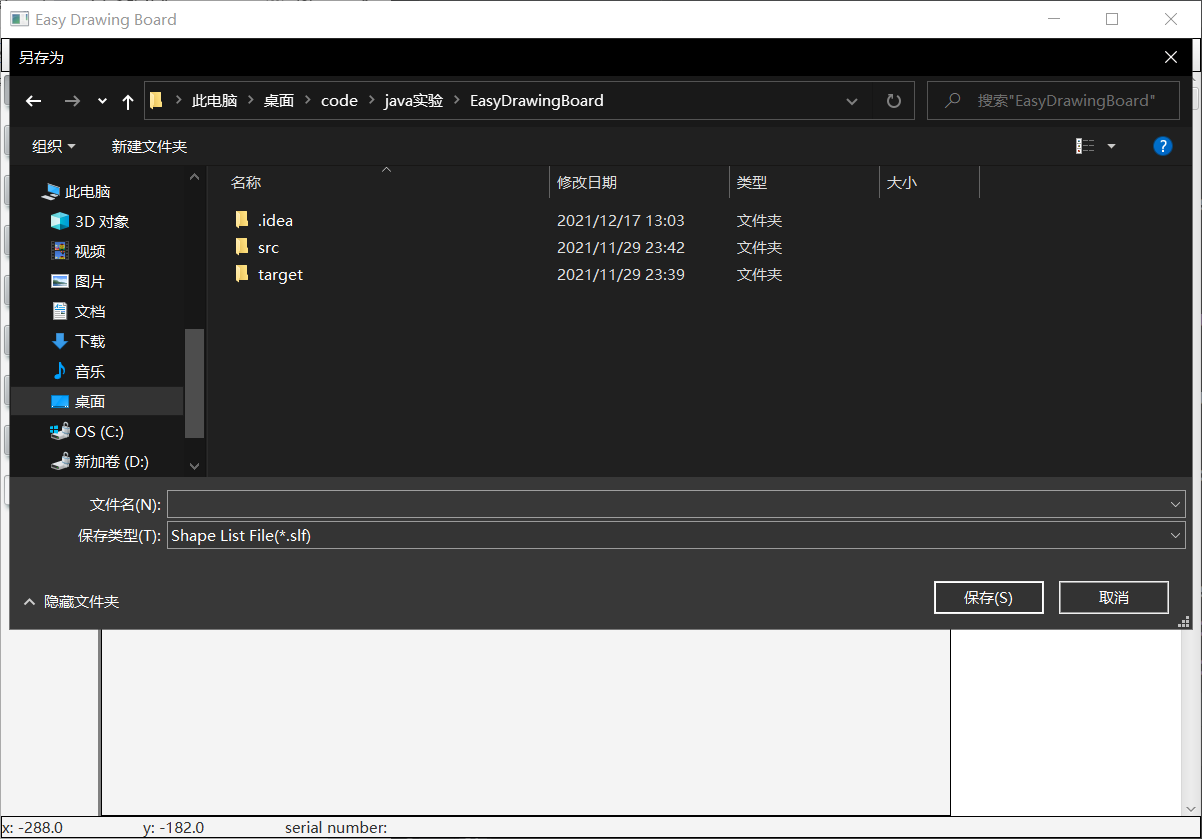
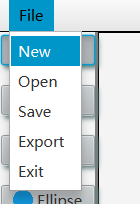


图 10 MenuBar 图 11 选择保存后

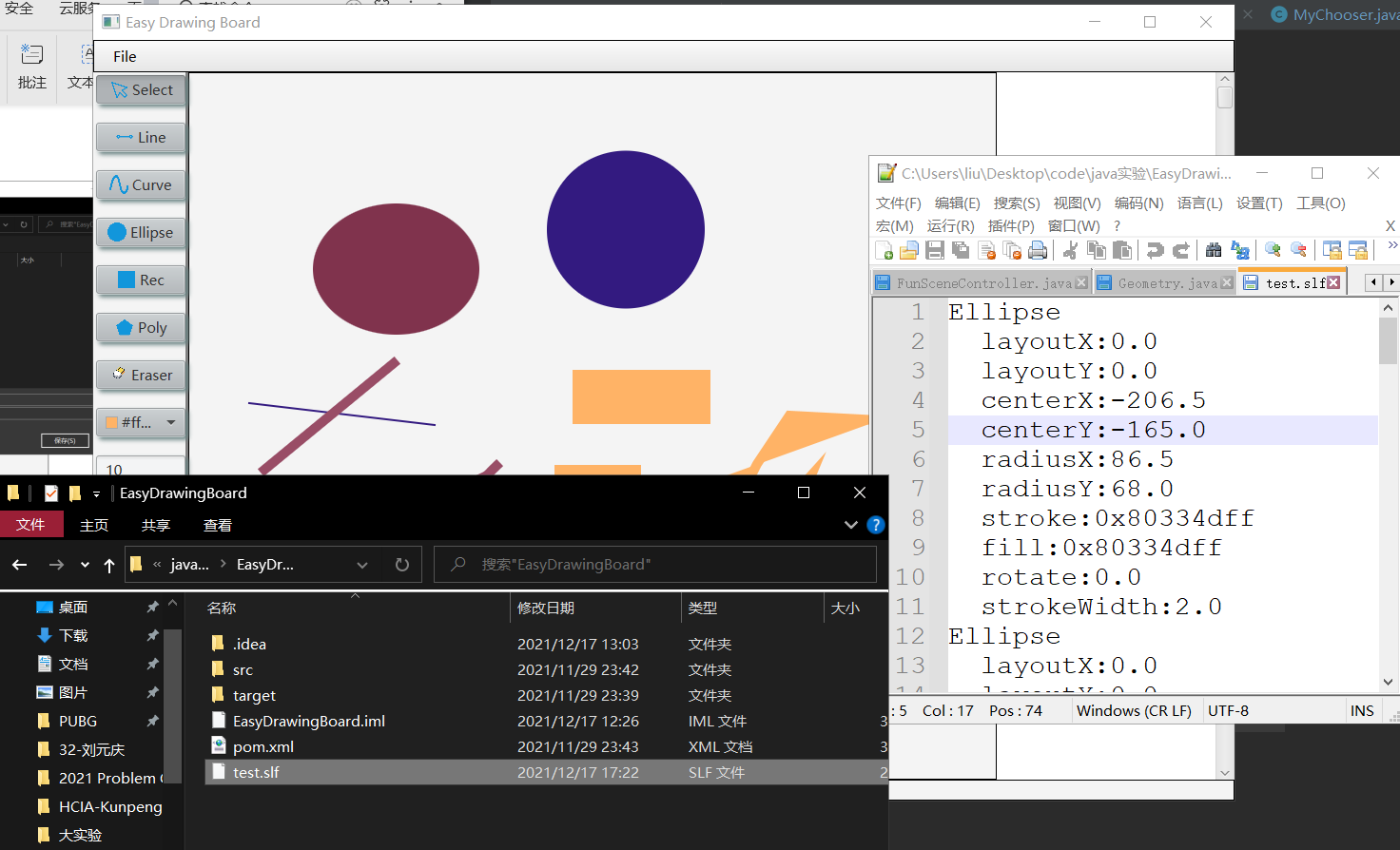


图 12 保存后的文件

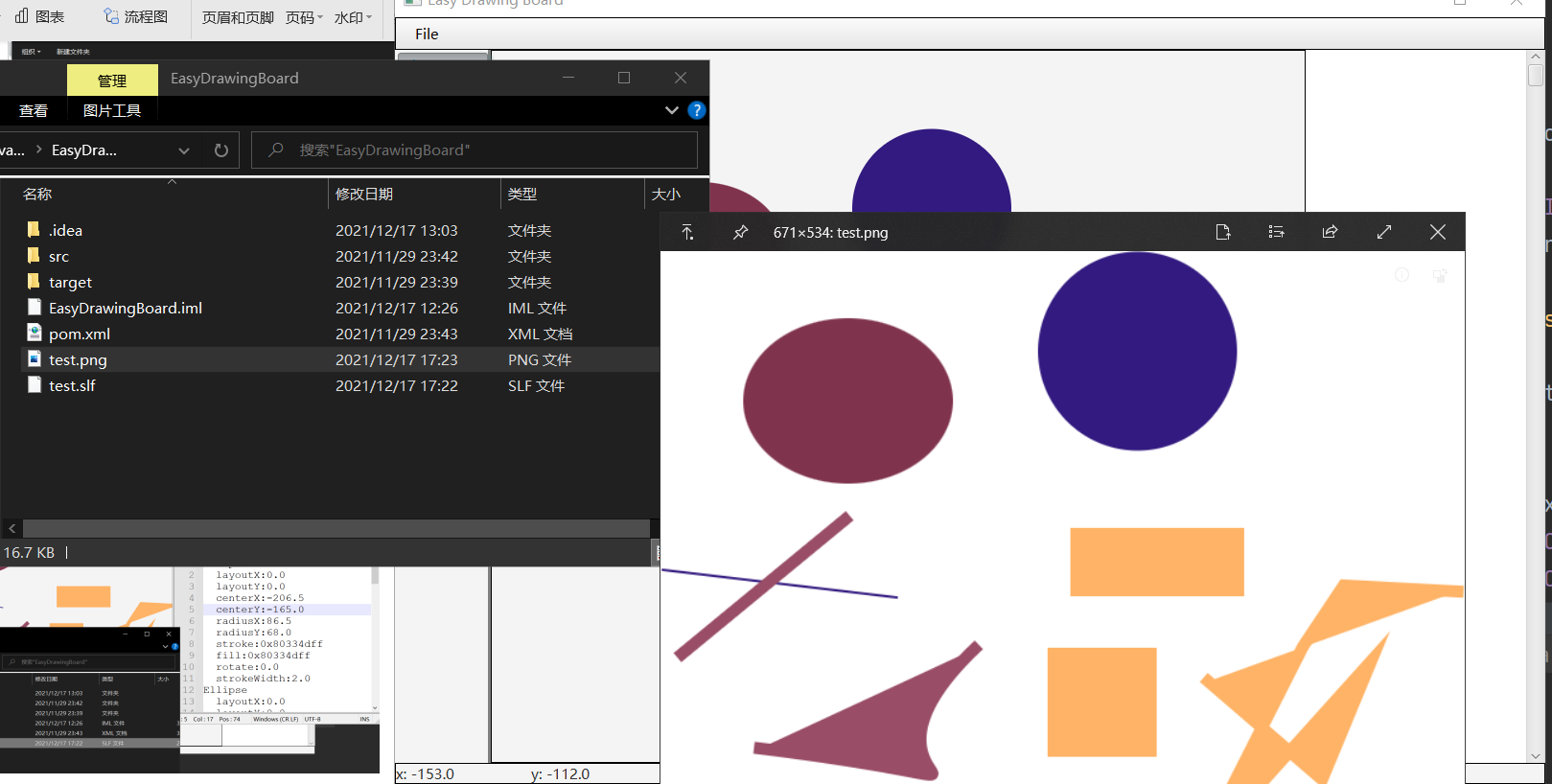


图 13 导出为PNG格式

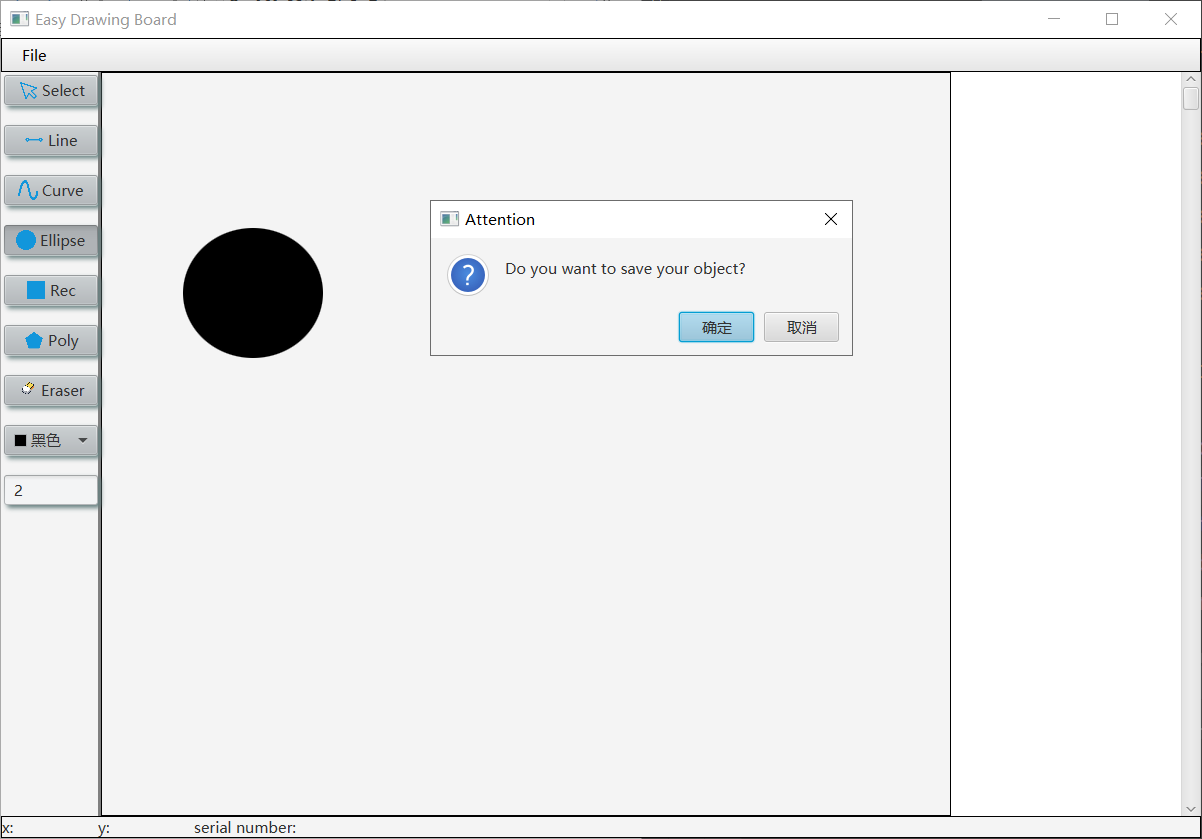


图 14 alert提示1

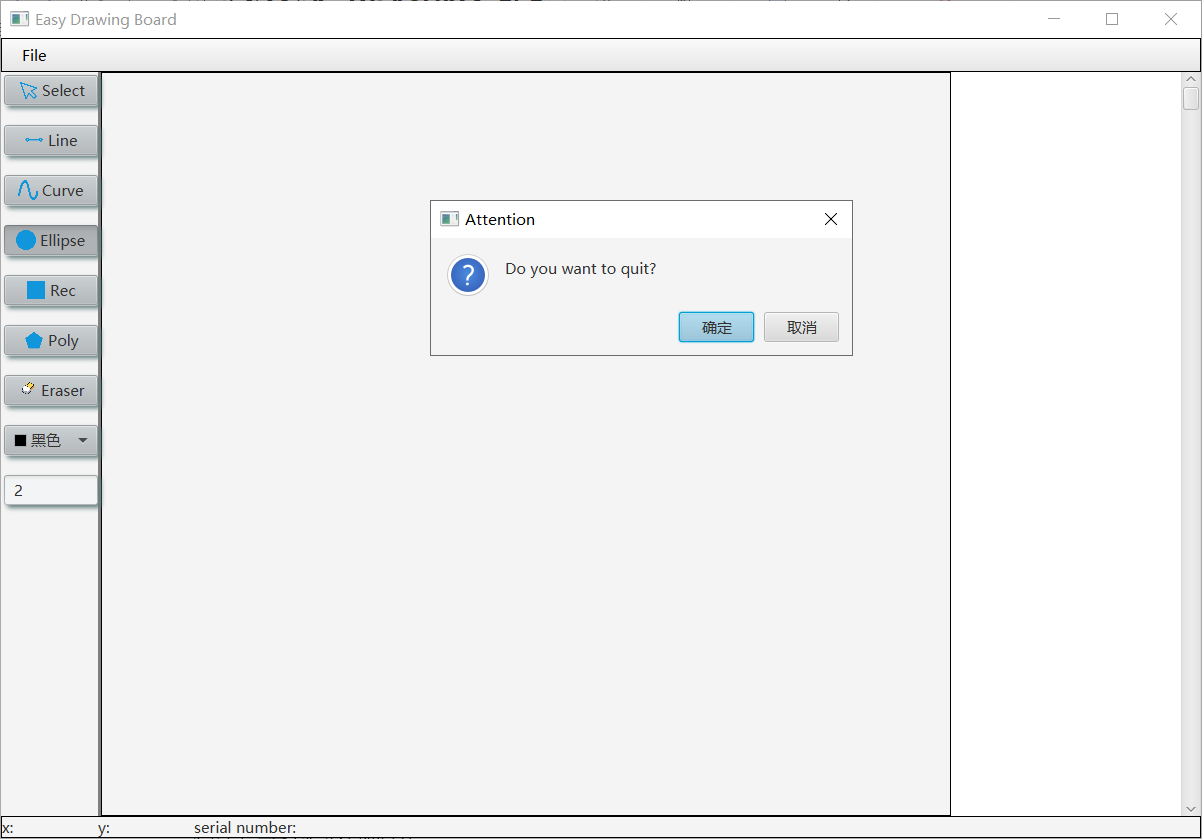


图 15 alert提示2

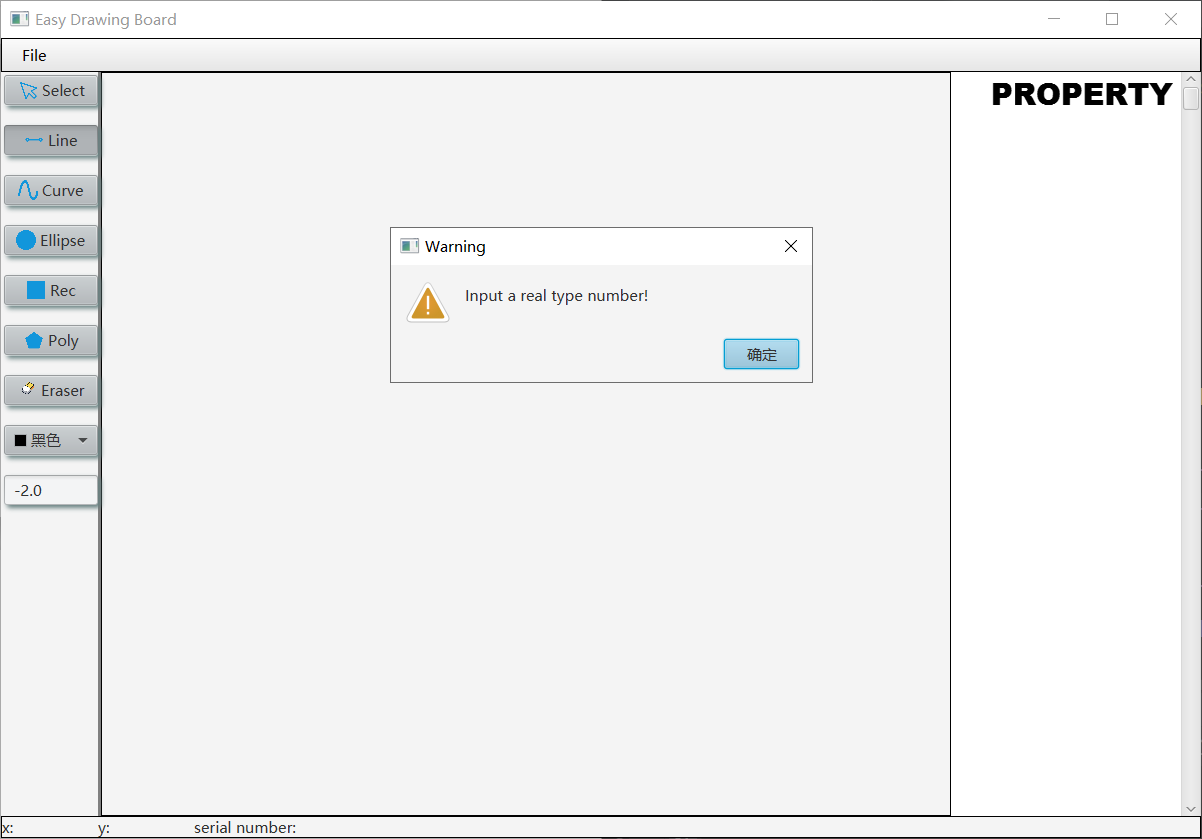


图 16 对线条宽度输入格式的检测

1. 程序操作方法和效果

这次实验我主要用了结构化程序设计、面向对象程序设计、事件驱动的程序设计方法。

结构化程序设计方法：

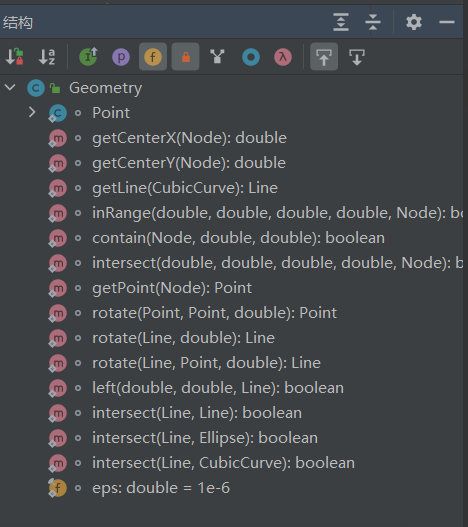


图 17 Geometry类的结构

因为Geometry是一个工具类，在设计时主要采取的时结构化的程序设计方法，将Shape类所需的方法封装起来并设置为静态，以便于其他类进行直接的调用。

面向对象程序设计方法：

从画板这个具象的事物出发，逐渐抽象出画板的各个部分，包括构建画板的五个部分，MenuBar、ToolBar、PropertyBar、Cue、Board，以及组合这五个部分的MainPane。对这些具体的对象进行建模设计，自顶向下、逐步求精。

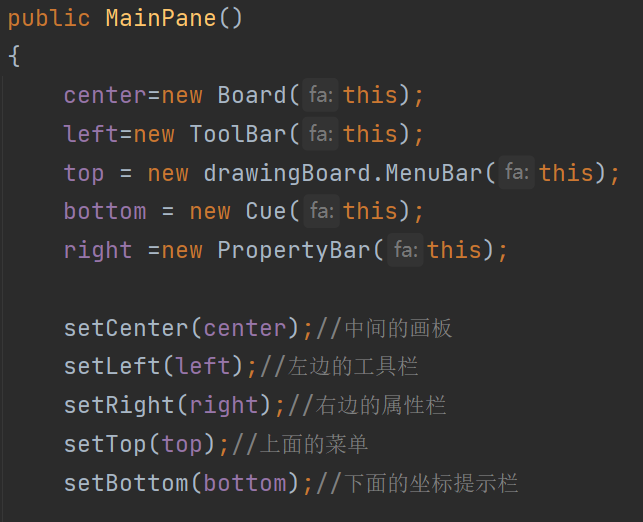


图 18 MainPane构造方法

事件驱动的程序设计方法：

我针对各个形状都需要的鼠标操作，包括press、release、drag动作，把他们抽象为一个接口Tool，面向所有的形状，MyLine、MyEllipse等都需要实现Tool接口，这样统一了操作，在鼠标或者键盘操作时对事件进行响应。

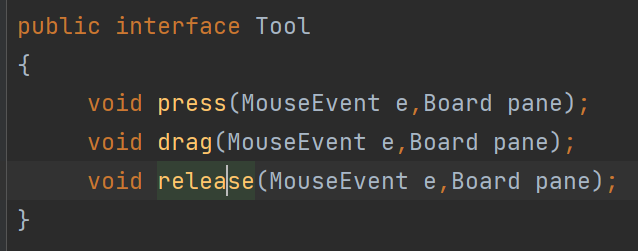


图 19 Tool接口

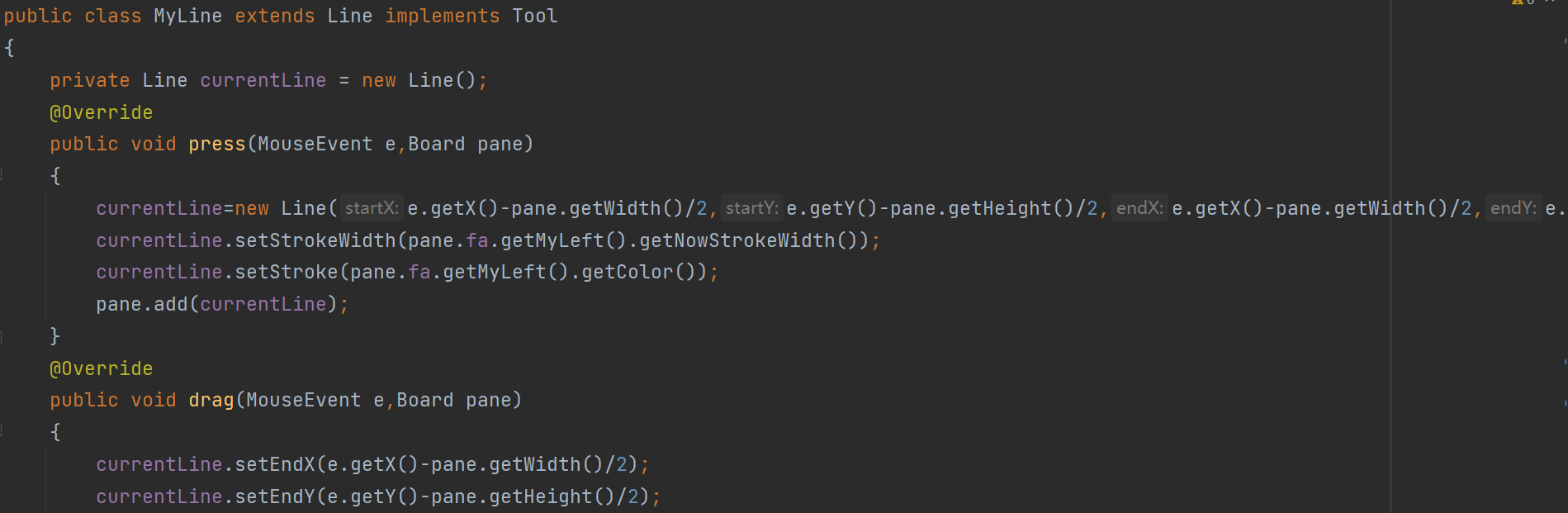


图 20 MyLine类

这样可以直接对对象进行操作，针对不同的鼠标事件进行不同的响应。