**集美大学计算机工程学院实验报告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：Java程序设计 | **班级**：软件1912 | **实验成绩**： |
| **指导教师**：李　旺 | **姓名**：游嘉升 |  |
| **实验项目名称**：图形程序设计 | **学号**：201921122050 | **上机实践日期**：11月6日 |
| **实验项目编号**： | **组号**： | **上机实践时间**： 2 学时 |

**一、目的**

掌握图形程序设计；掌握Swing组件；掌握事件处理。

**二、实验内容与设计思想**

1. 绘图组件

(1)编写一个用于绘图的DrawLineComponent extends JComponent。覆盖其paintComponent方法，进行图形的绘制(**要求尽量使用Graphics2D.draw(Shape)方法绘制**)。该组件可以根据给定的点(Point2D)的集合，绘制出对应的曲线（相邻的两点构成线段，曲线实际上是折线）。

(2)使用上述的DrawLineComponet，分别绘制正弦曲线、余弦曲线或者画出自己编写表达式(如y=kx、y=sinx+cosx、y=e-x等)的曲线。**注意函数值的(x,y)跟组件中的坐标(x,y)的对应关系。**

**说明：图形可以看作是线的组合，线则是点的组合，只要构造出点的集合，就可以构造出对应的线，进而画出整个图形。**

进阶1：如果还想通过上面的程序画其他曲线？如何进行？试编写一个通用的绘图程序。

进阶2：如果想在绘制曲线的同时，绘制坐标轴，或者同时绘制多条曲线而不互相干扰，上述的DrawLineComponent能不能实现？如果能，请写出一个示例；如果不能，请改造它，并写出一个示例。

2. 事件处理

(1)实现对第1小题绘制出来的正弦曲线使用鼠标右键拖动(即平移，整个曲线随鼠标移动而移动)。

(2)使用鼠标左键拖拽曲线上的某个点以改变曲线的图形。

3. 画笔程序

实现简单的鼠标画笔功能（仅实现鼠标拖动画线功能即可），参考Windows的“画图”程序。

思路：

(1)需要有一个集合专门存储鼠标按下时产生的点的坐标，但要注意：只能存储一条连续的曲线，鼠标松开时这个List就不再存储接下来的点。

(2)不同的连续曲线存储在不同的集合，所以可能有多个集合。

(3)每次移动鼠标时，要重绘所有集合中所有的点。

(4)关闭程序时保存内存中所有List中的所有点(可使用文件)。

进阶(选做)：学有余力的同学，可以尝试扩充功能，如线条颜色、常见几何图形(矩形、椭圆等)，并使这些图形绘制后仍然可以选择并移动。

4. 简易计算器

使用Swing组件编制简易计算器，实现简单的四则运算功能。参考附件calculator.png。

思路：

(1)使用合适的布局管理器，实现界面的排版；

(2)添加事件处理代码，分别处理操作数与操作码。

进阶(选做)：学有余力的同学，可以尝试扩充功能，如三角函数、幂次、对数、括号表达式等，兼容键盘操作等。

**三、实验使用环境**

Java version "13.0.2"

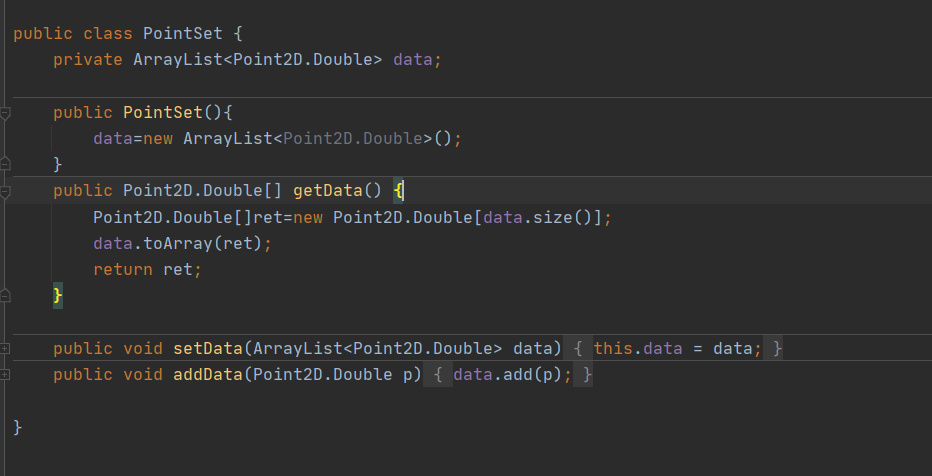
IntelliJ IDEA 2020.2.2 x64

**四、实验步骤和调试过程**

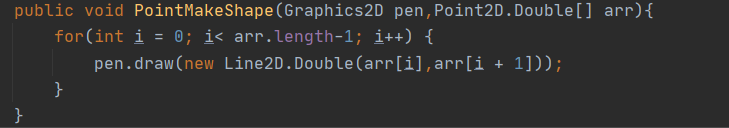
1. 绘图组件

1. 设计思想：创建一个MainFrame类继承JFrame类并在新的线程里创建一个匿名MainFrame。在MainFrame类里添加一个继承于Component类的DrawLineComponent类用于画出点集。
2. 关键代码：

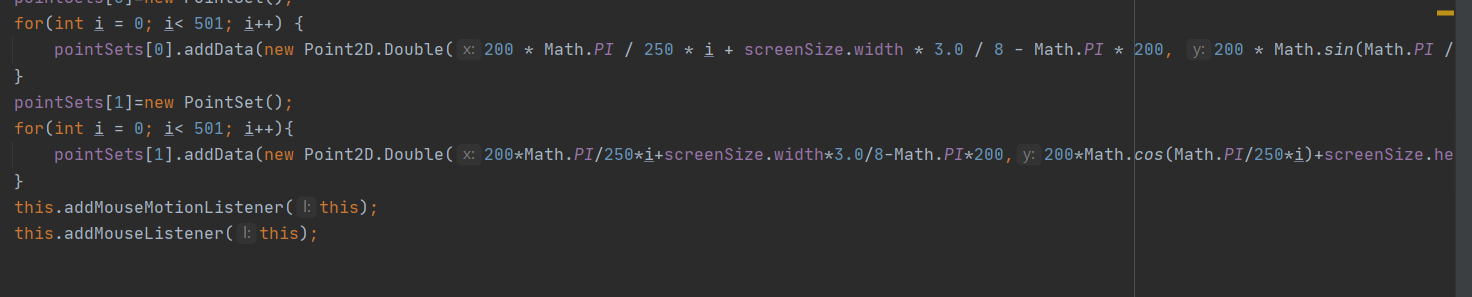
点集的定义



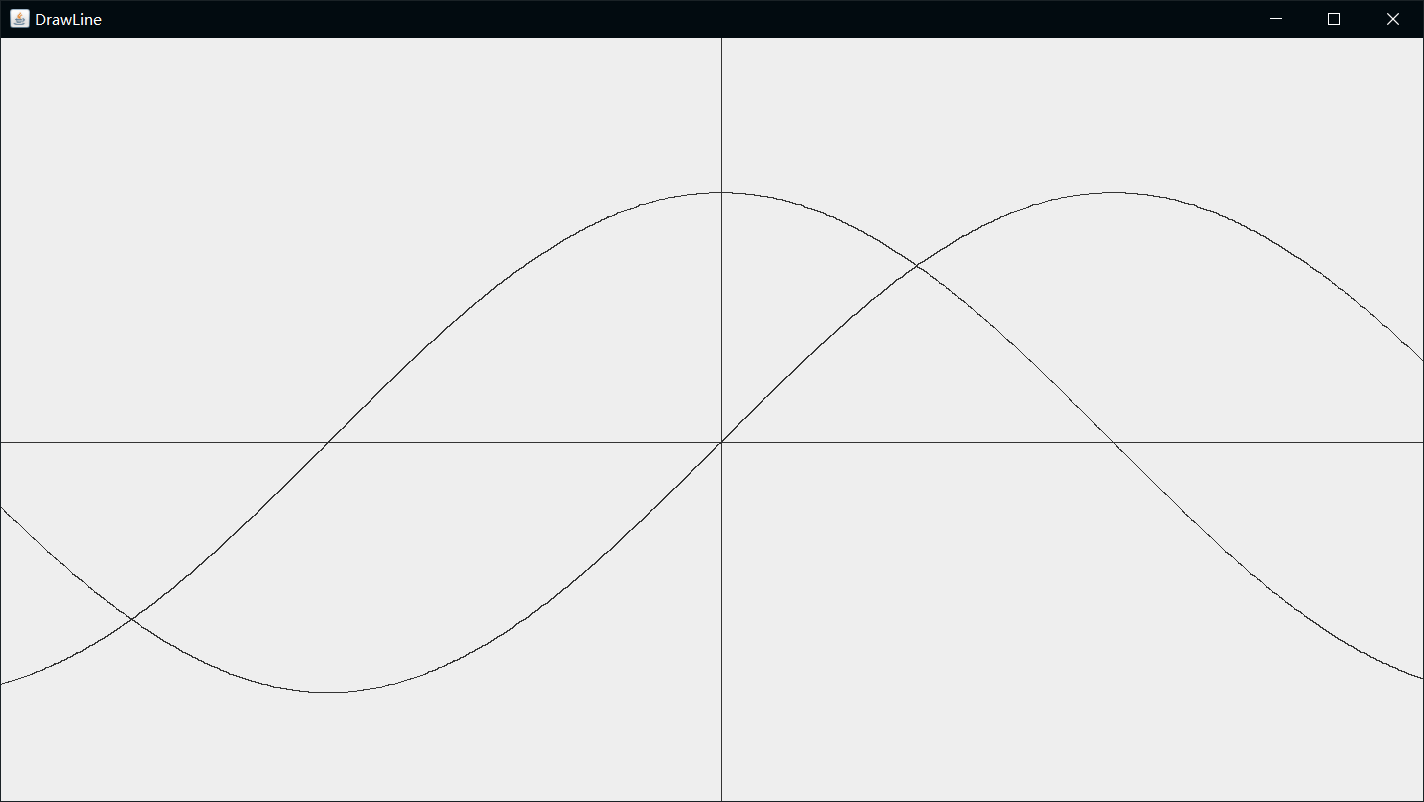
用于根据点集画出曲线的方法



1. 测试数据：



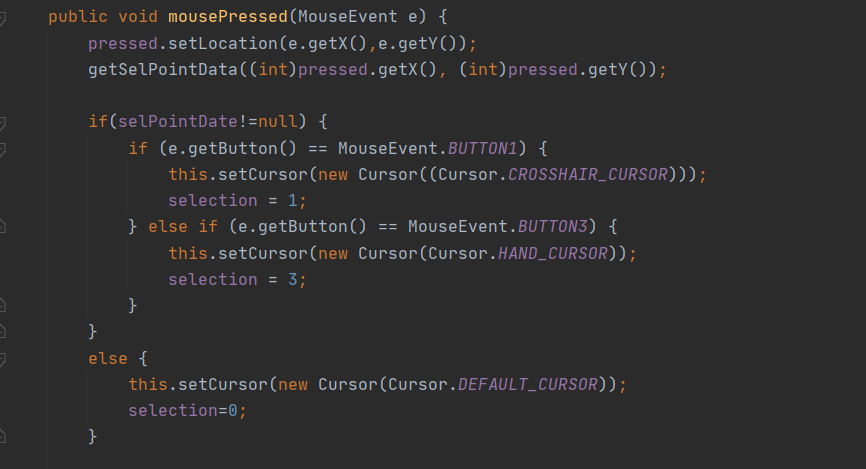
1. 测试结果：



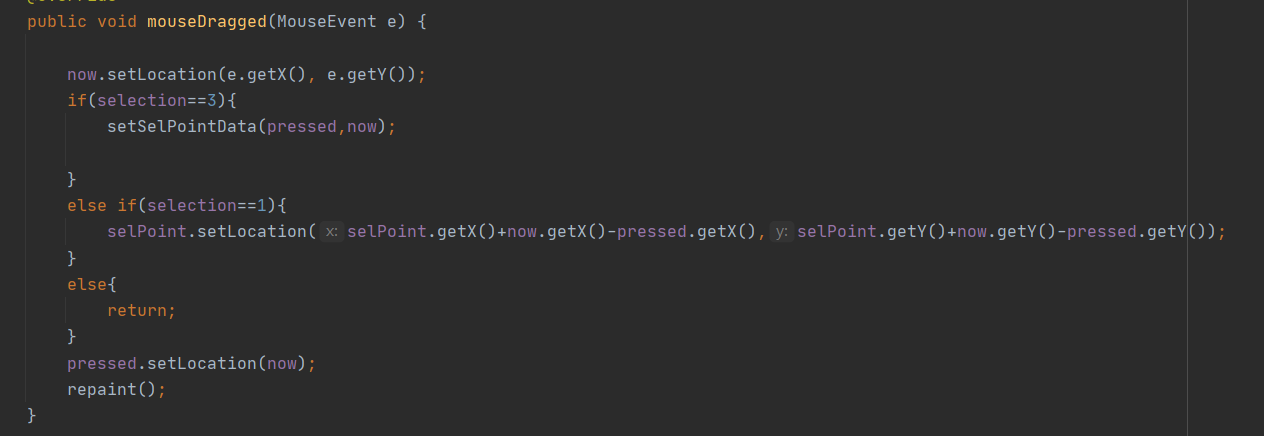
2. 事件处理

1. 设计思想：让上题的DrawLineComponent拥有MouseListener和MouseMotionListenner接口并定义鼠标事件方法，对拖拽事件左键采取记录拖拽点位置然后重绘图形，右键采取记录拖拽点轨迹并重新计算点集位置并重绘。
2. 关键代码：

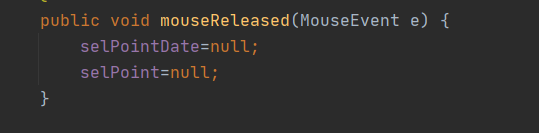
鼠标按下事件



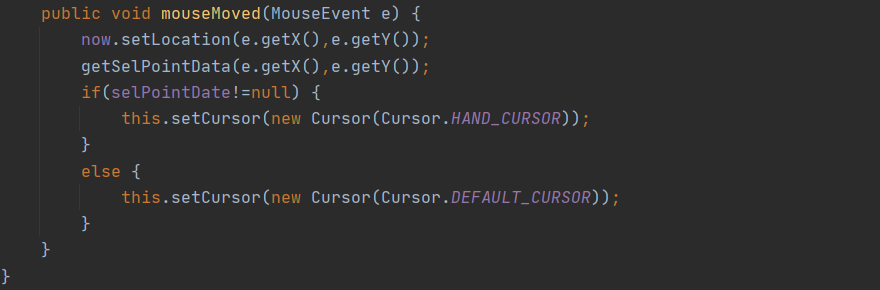
鼠标拖拽事件



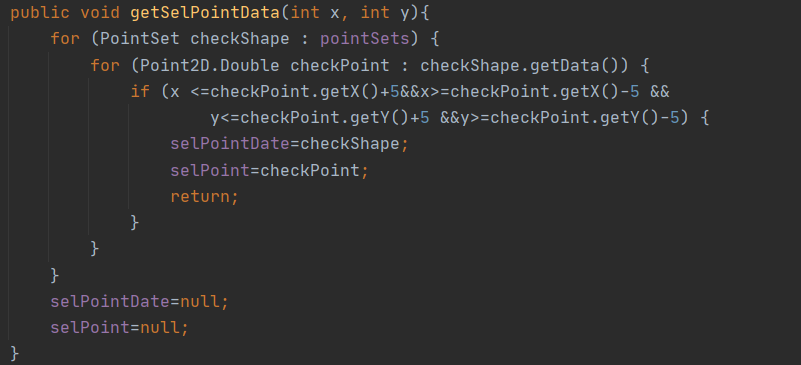
鼠标松开事件



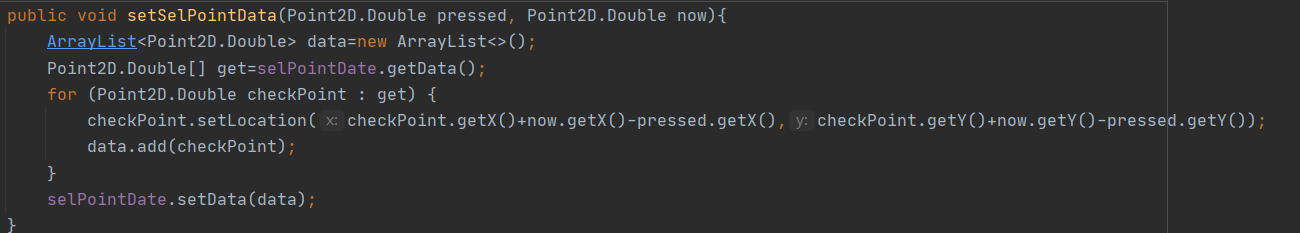
鼠标移动事件



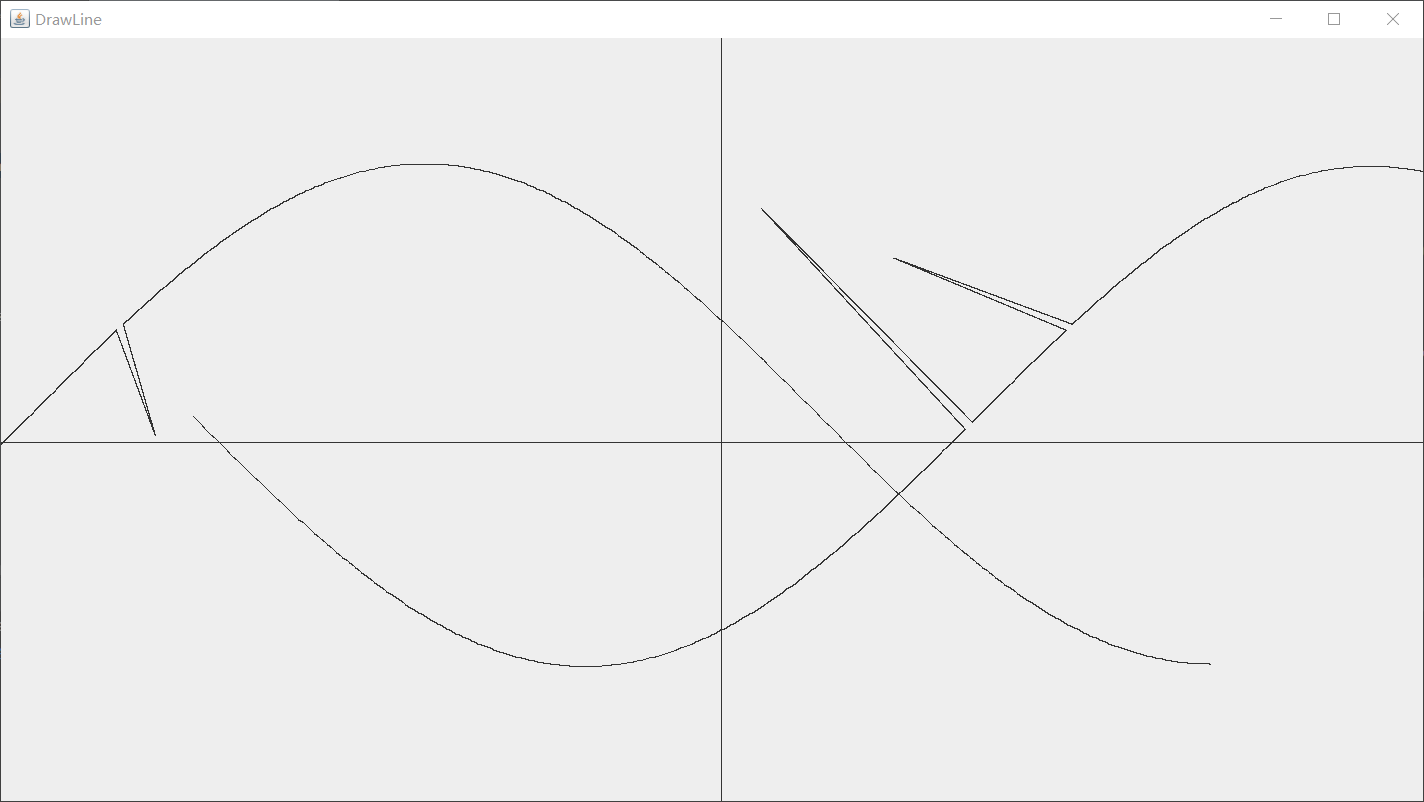
得到鼠标点击的点



计算点集位移并对应用于点集



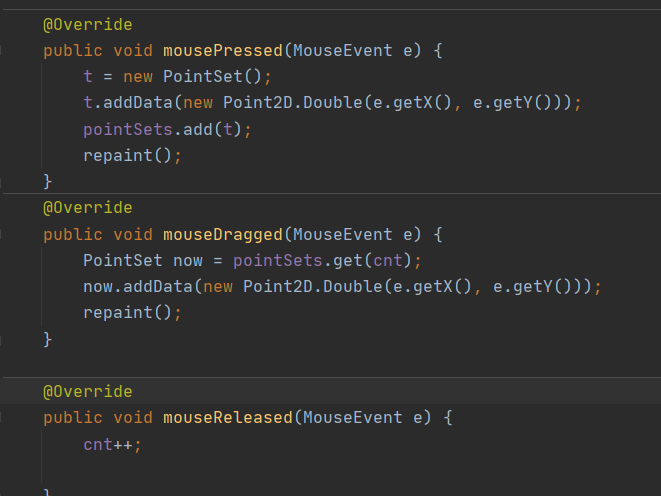
1. 测试数据：同上题
2. 测试结果：



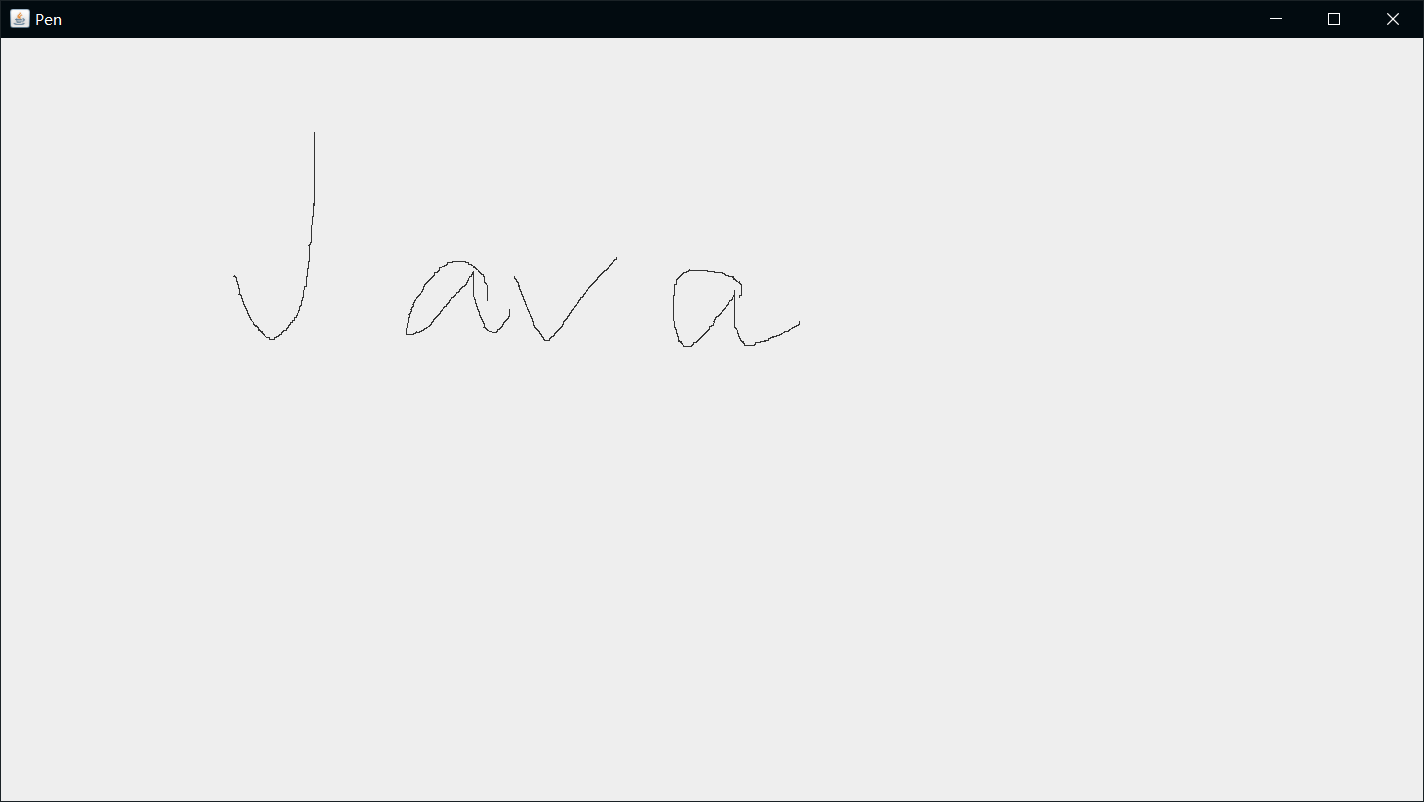
3. 画笔程序

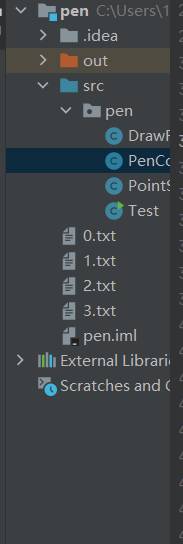
1. 设计思想：在鼠标按下后，移动鼠标，记录鼠标移动的位置，生成鼠标移动的点集，再将这个点集两点之间相连并绘制出来。在窗口关闭时，将生成的多个点集写入文件中。
2. 关键代码：

鼠标按压，拖拽，松开时的事件处理



1. 测试数据：无
2. 测试结果：





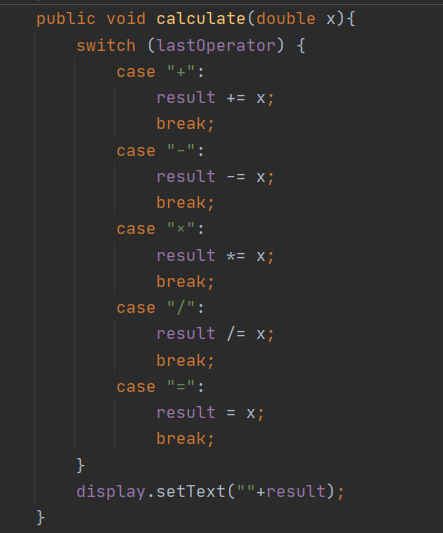
4. 简易计算器

1. 设计思想：添加多个按钮，按钮分为两个不同的属性，一个是操作数按钮，一个是操作符按钮，对于不同按钮按下产生不同的事件处理，达到计算的效果。
2. 关键代码：

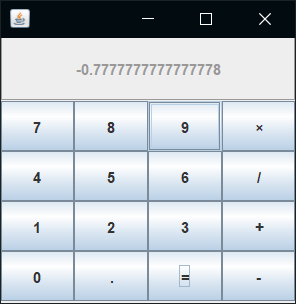
操作数按钮点击时产生的作用，操作符按钮点击时产生的作用



计算模块：



1. 测试数据：1+3-5\*7/9
2. 测试结果：



**五、实验小结**

1．实验中遇到的问题及解决过程

①在第一题时遇到点击点精度过高而点不到点的问题

解决过程：给一个点的判定为一个正方形的10px\*10px的位置，这样就可以在一定范围里判定到点击的某个点。

缺点：精度过低有些点实际上无法被点到。

②第四题用边框布局使实际数据输出栏按钮过小影响视觉。

解决过程：强制用setPreferredSize方法定义按钮的大小为一定大小

2．实验中产生的错误及原因分析

有好多错误。。但是主要错误原因还是对Swing中的各种方法不熟悉不会用导致错误出现。

3．实验体会和收获。

初步了解MVC模型，对于Swing编程有一定的了解与使用。

**六、附录**

[1][美] 凯.S.霍斯特曼（Cay S. Horstmann）.Java核心技术·卷 I（原书第10版）[M].机械工业出版社