**Комитет по образованию г. Санкт-Петербург**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ПРЕЗИДЕНТСКИЙ ФИЗИКО-МАТЕМАТИЧЕСКИЙ**

**ЛИЦЕЙ №239**

**Отчет о практике**

**«Создание графических приложений на языке Java»**

Учащийся 10-2 класса

Андреев. А.П.

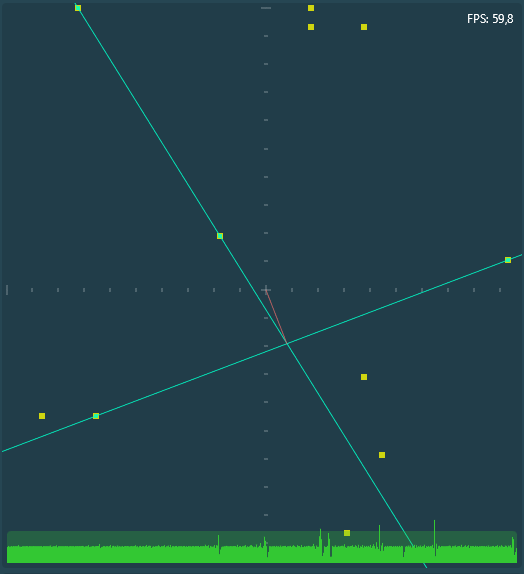
Преподаватель:

Клюнин А.О.

Санкт-Петербург – 2023 год

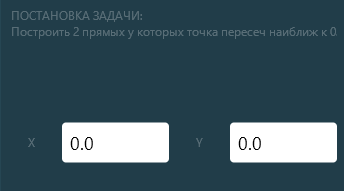
# 1. Постановка задачи

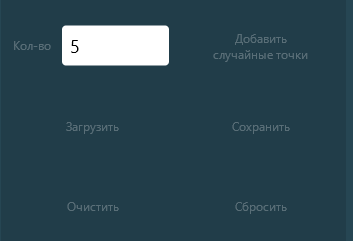
1. На плоскости задано множество точек. Найти среди них такие две пары, что точка пересечения прямых, проведенных через эти пары точек, находится ближе всего к началу координат. То есть, если рассмотреть все возможные прямые, которые могут быть построены по парам указанных точек, и все возможные точки пересечения этих прямых между собой, то нужно найти такую точку пересечения этих прямых, которая находится ближе всего к началу координат. Ответом должна быть: эта точка пересечения, те две прямые, которые в этой точке пересекаются, а также отрезок, проведенный из начала координат до точки пересечения.



# 2. Элементы управления

В рамках данной задачи необходимо было реализовать следующие элементы управления:





Для добавления точки по координатам было создано два поля ввода: «X» и «Y».

Т.к. задача предполагает только один вид геометрических объектов, то для добавления случайных элементов достаточно одного поля ввода. В него вводится количество случайных точек, которые будут добавлены.

Также программа позволяет добавлять точки с помощью клика мышью по области рисования

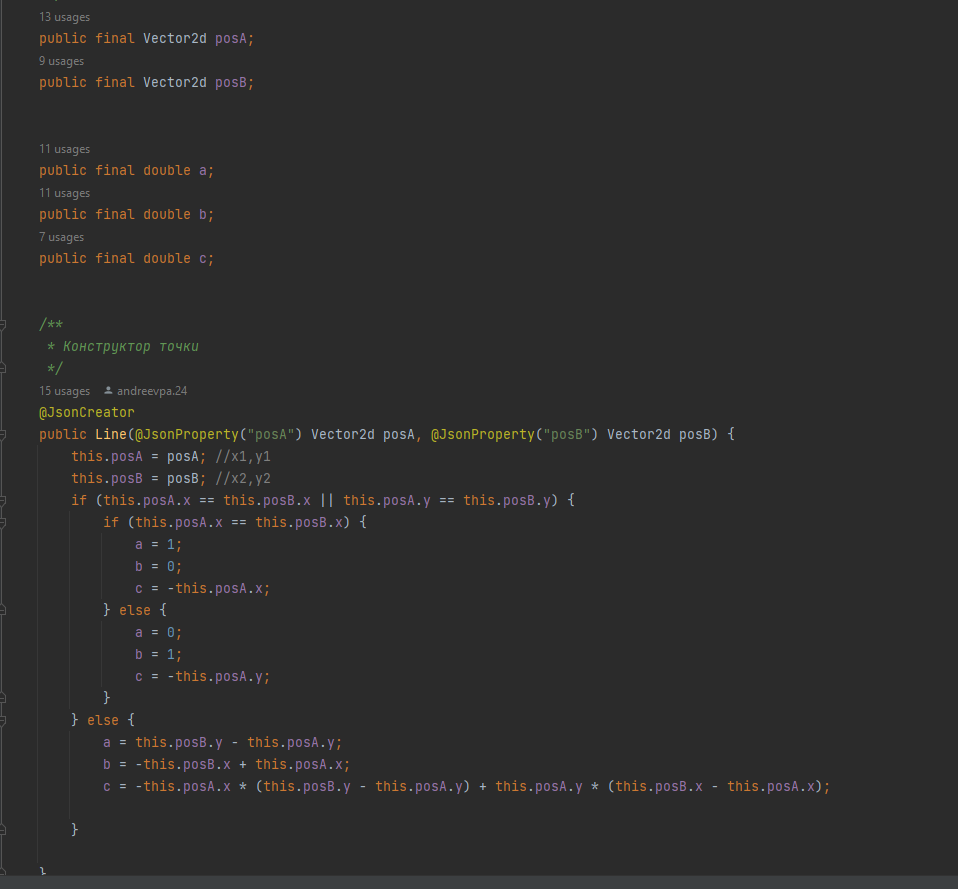


При клике левой кнопкой мыши по области рисования в месте клика создаётся точка.

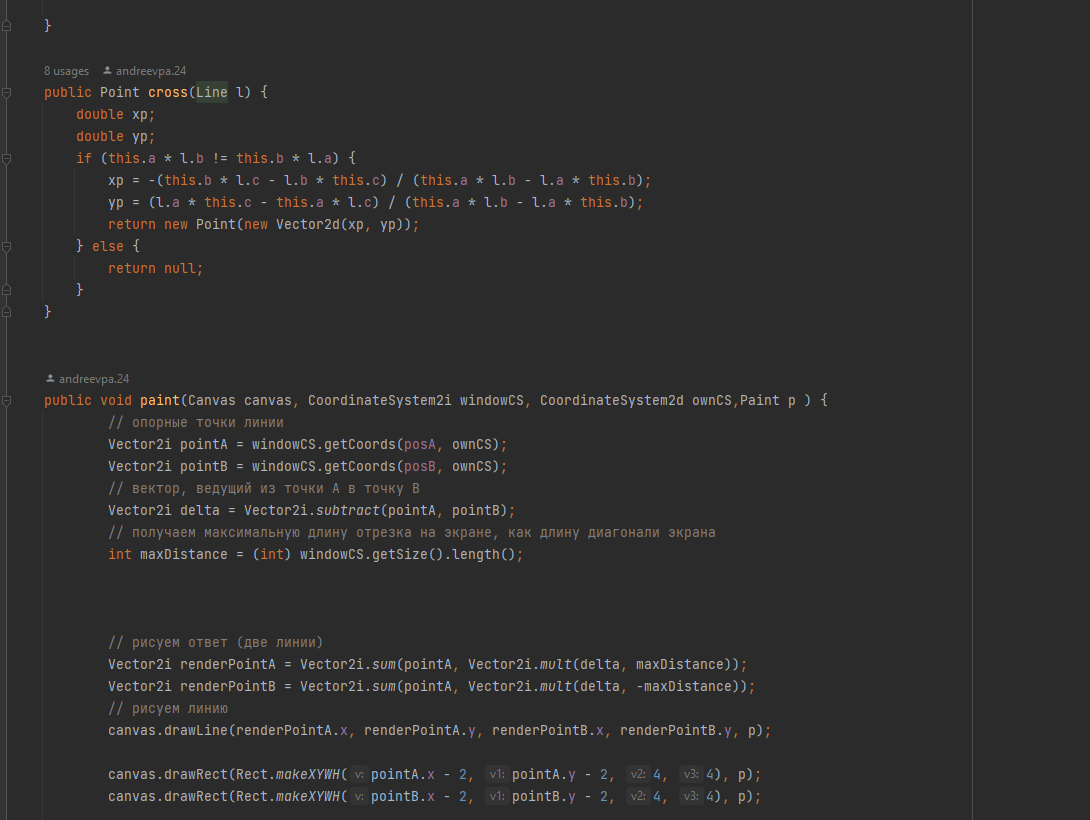
# 3. Структуры данных

Для того чтобы хранить линии, был разработан класс **Line.java**

**В нем содержится метод cross- ищущий пересечения, с помощью общего уравнения прямой.**



1



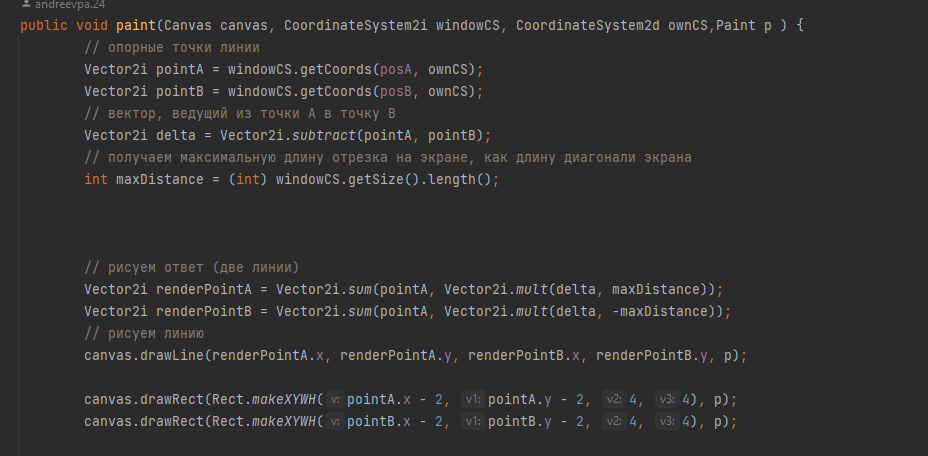
2

# 4. Рисование

Чтобы нарисовать точку, использовалась команда рисования прямоугольников **canvas.drawRect().**

**Для рисования линий была использована другая команда:**

*public* *static* Vector2i mult(Vector2i v, *int* s) {  
 *return* *new* Vector2i(v.x \* s, v.y \* s);

****

# 5. Решение задачи

Для решения поставленной задачи в классе **Task** был разработан метод **solve().**

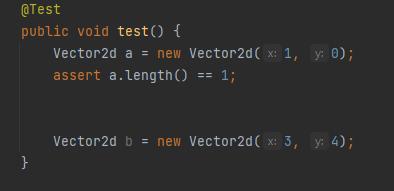


В нём перебираются пары точек и ищет точку пересечения, а после ищет ближайшую к 0 0 эти точки отображаются.

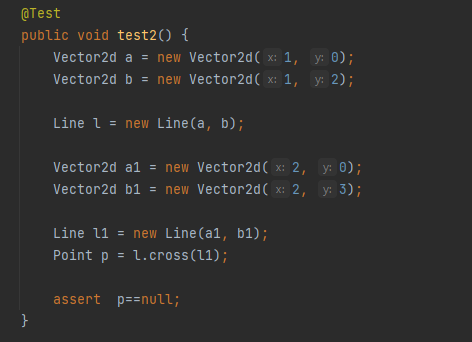
# 6. Проверка

Для проверки правильности решённой задачи были разработаны unit-тесты.

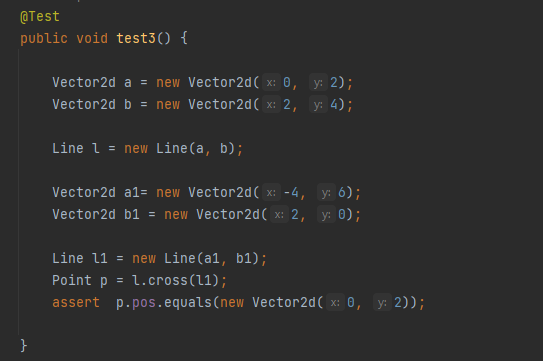
Тест 1



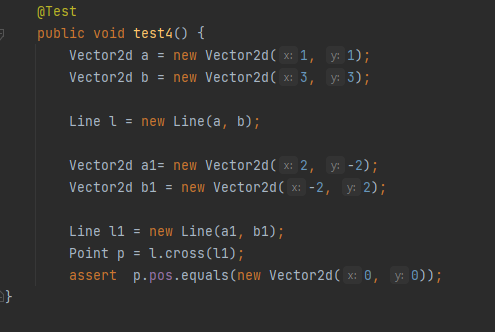
Тест 2



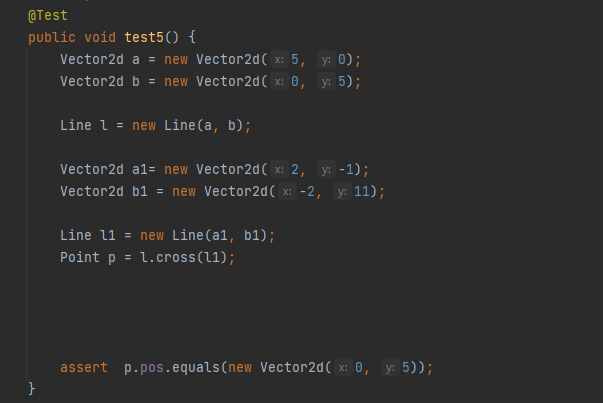
Тест 3



Тест 4



Тест 5



# 7. Заключение

В рамках выполнения поставленной задачи было создано графическое приложение с требуемым функционалом. Правильность решения задачи проверена с помощью юнит-тестов.