**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 10-1 класса

Шонин Г.А.

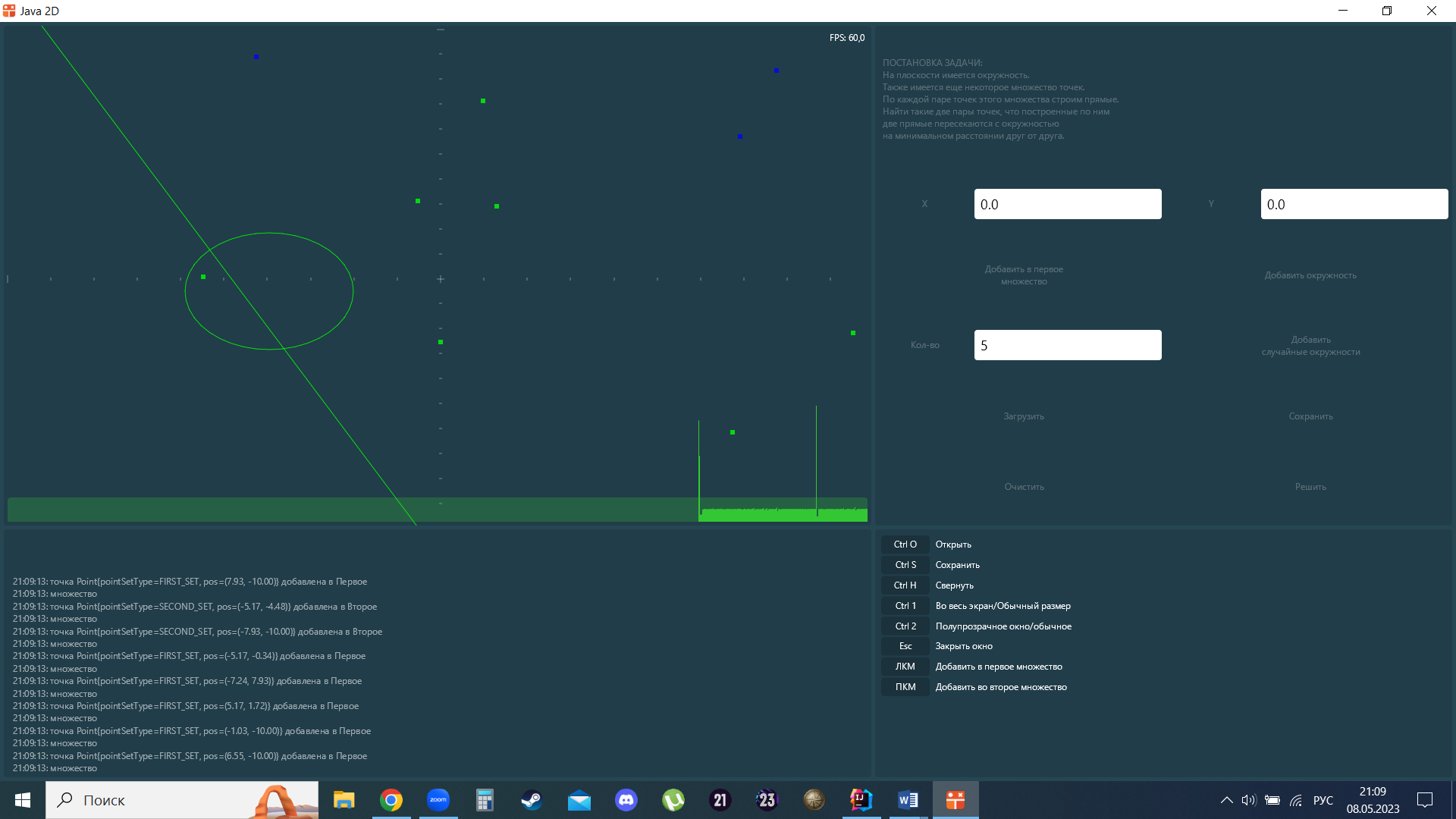
Преподаватель:

Клюнин А.О.

Санкт-Петербург - 2023 год

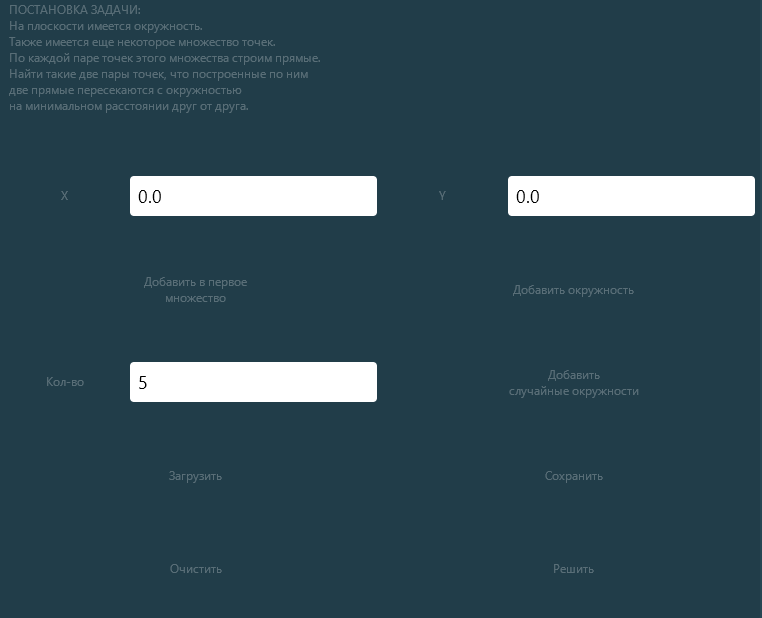
1.Постановка задачи

На плоскости имеется окружность. Также имеется еще некоторое множество точек. По каждой паре точек этого множества строим прямые. Найти такие две пары точек, что построенные по ним две прямые пересекаются с окружностью на минимальном расстоянии друг от друга. Выделить найденные точки, нарисовать прямые, построенные по ним. Выделить точки пересечения этих прямых с окружностью. Выделить дугу между этими точками пересечения. Дополнительная опция: искомые две пары точек не должны иметь общих точек.



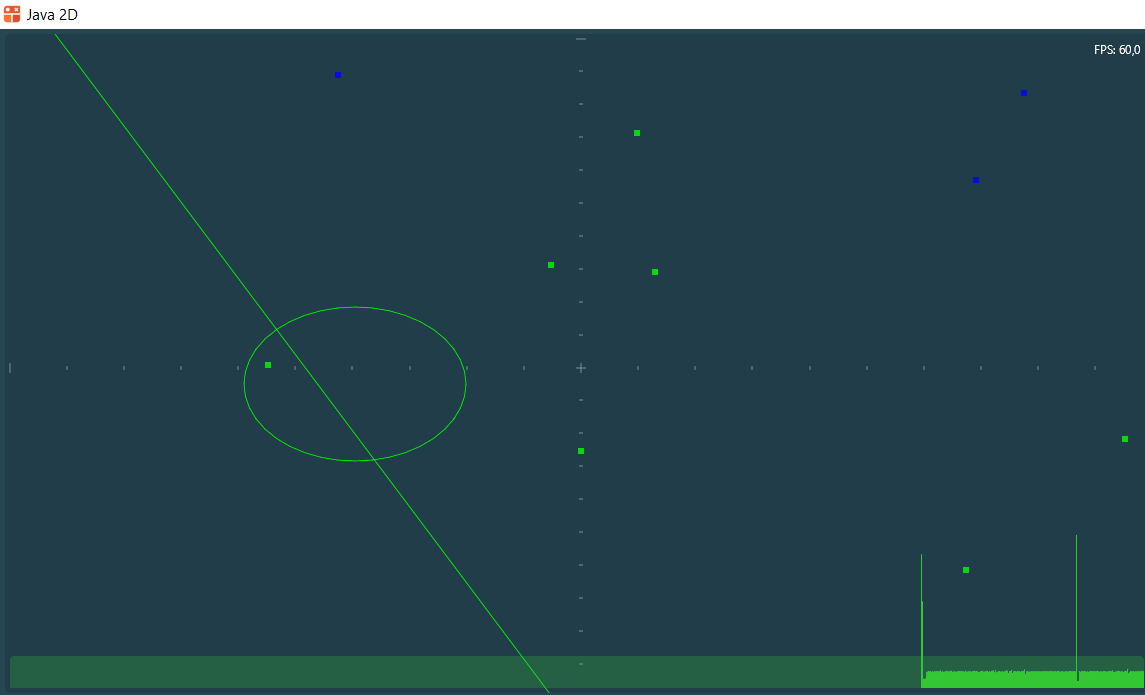
2.Элементы управления

В рамках данной задачи нужно было реализовать множество элементов управления. Они показаны ниже:



Для добавления точки по координатам было создано два поля ввода: «X» и «Y». Для добавления окружности использовалась функция от четырех координат(координат центра и координаты точки на окружности, по ним считался радиус). Также были созданы примитивы прямых и окружностей, для построения прямых по двум точкам был написан специальный метод.

Кроме того, написанная программа способна добавлять случайные точки, принадлежащие разным множествам по нажатию кнопок мыши. Аналогичные варианты действий написаны для добавления случайной окружности. Все это можно наблюдать в области рисования.



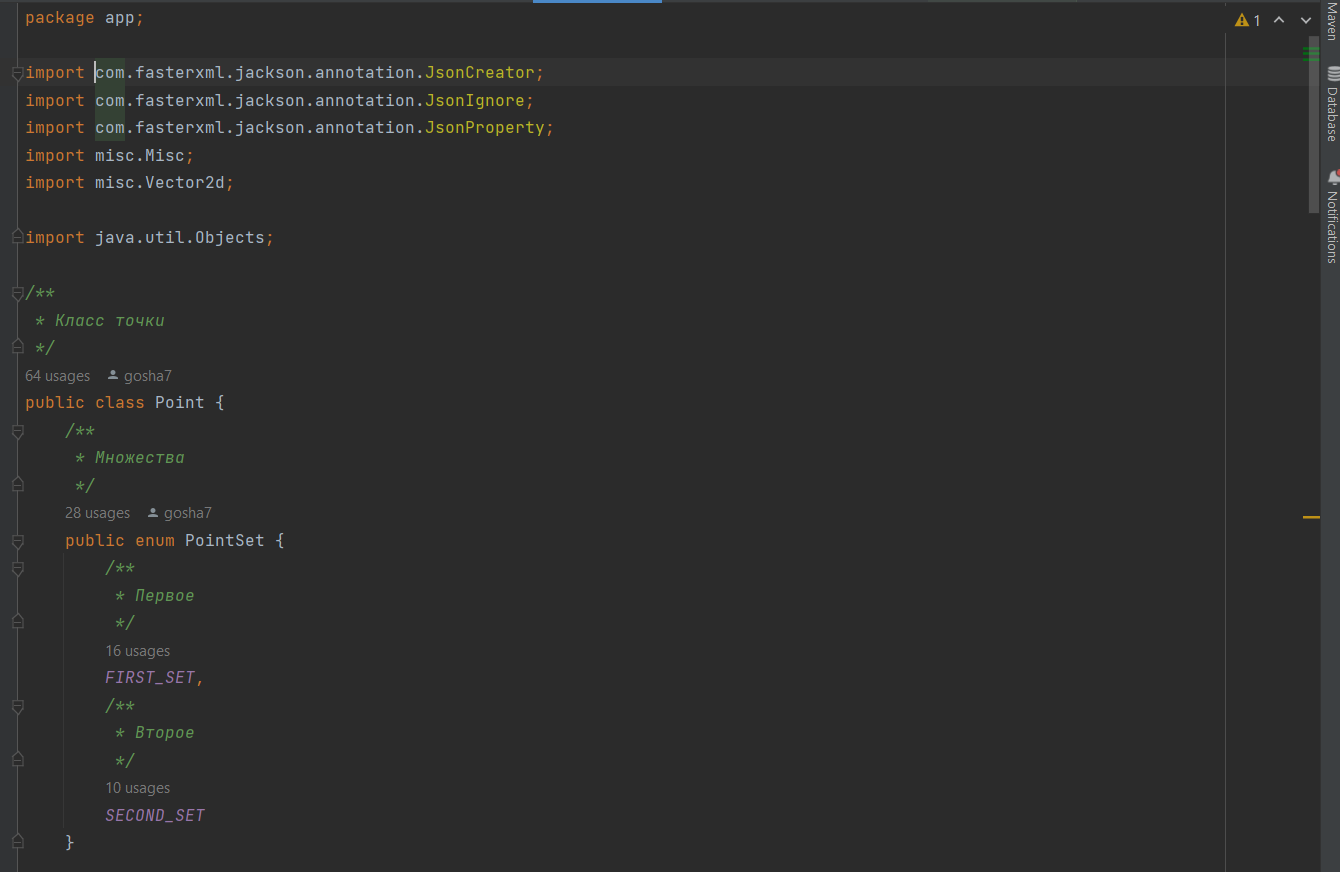
По нажатию левой или правой кнопки мыши будут добавлять случайные точки в области рисования. Так, при нажатии на левую кнопку мыши точка добавляется в 1 множество(зеленый цвет), а при нажатии на правую—во второе множество(синий цвет). Аналогично при нажатии кнопки Shift также будут добавлять окружности разных цветов.

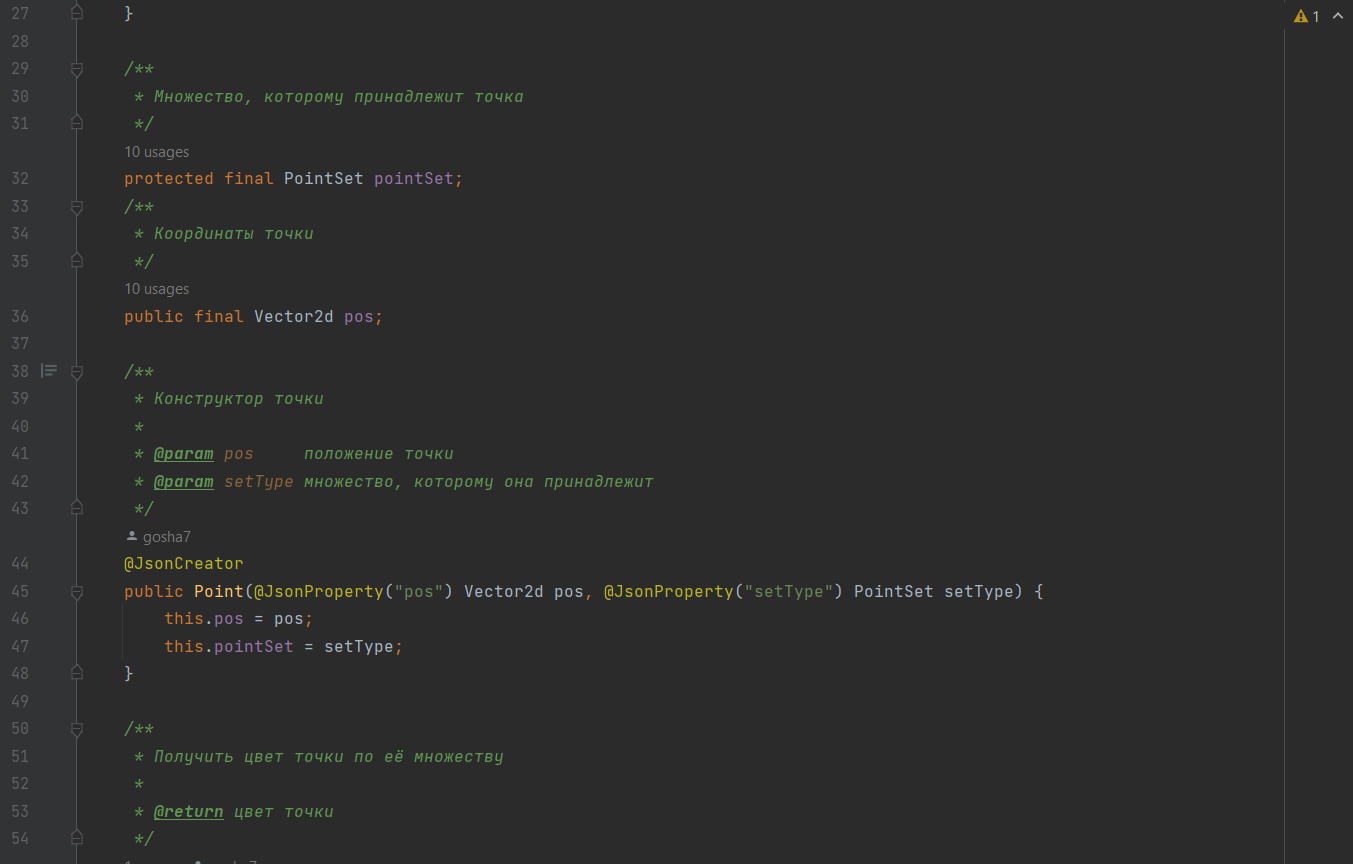
3.Структуры данных

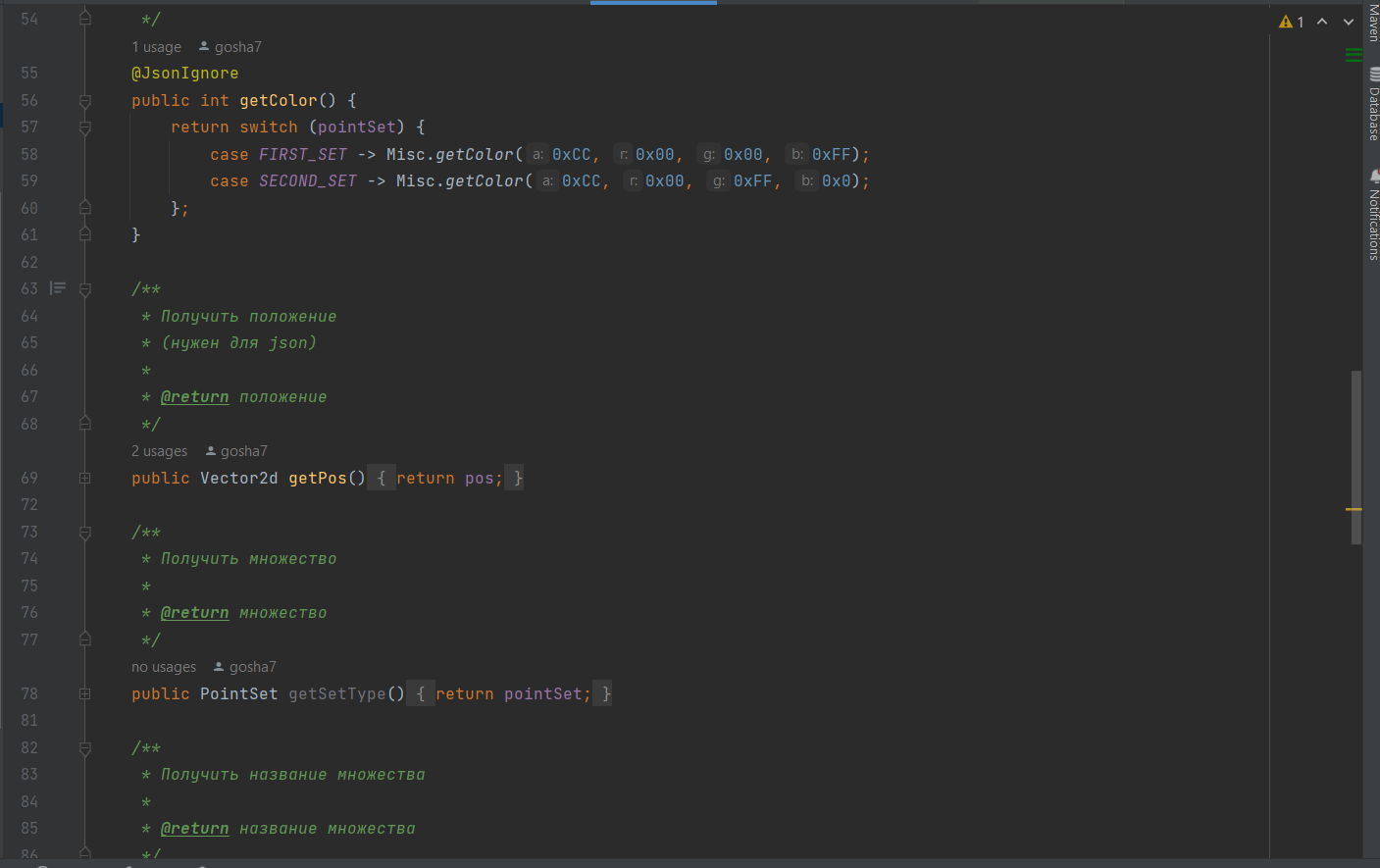
Для того чтобы хранить точки, был разработан класс **Point.java**. Аналогичные классы были разработаны для хранения окружностей и прямых.

В него были добавлены поля **pos**, соответствующее положению точки в пространстве задачи и тип множества **pointset**. Хранение типа множества обеспечено за счёт введения нового перечисления **PointSet**.

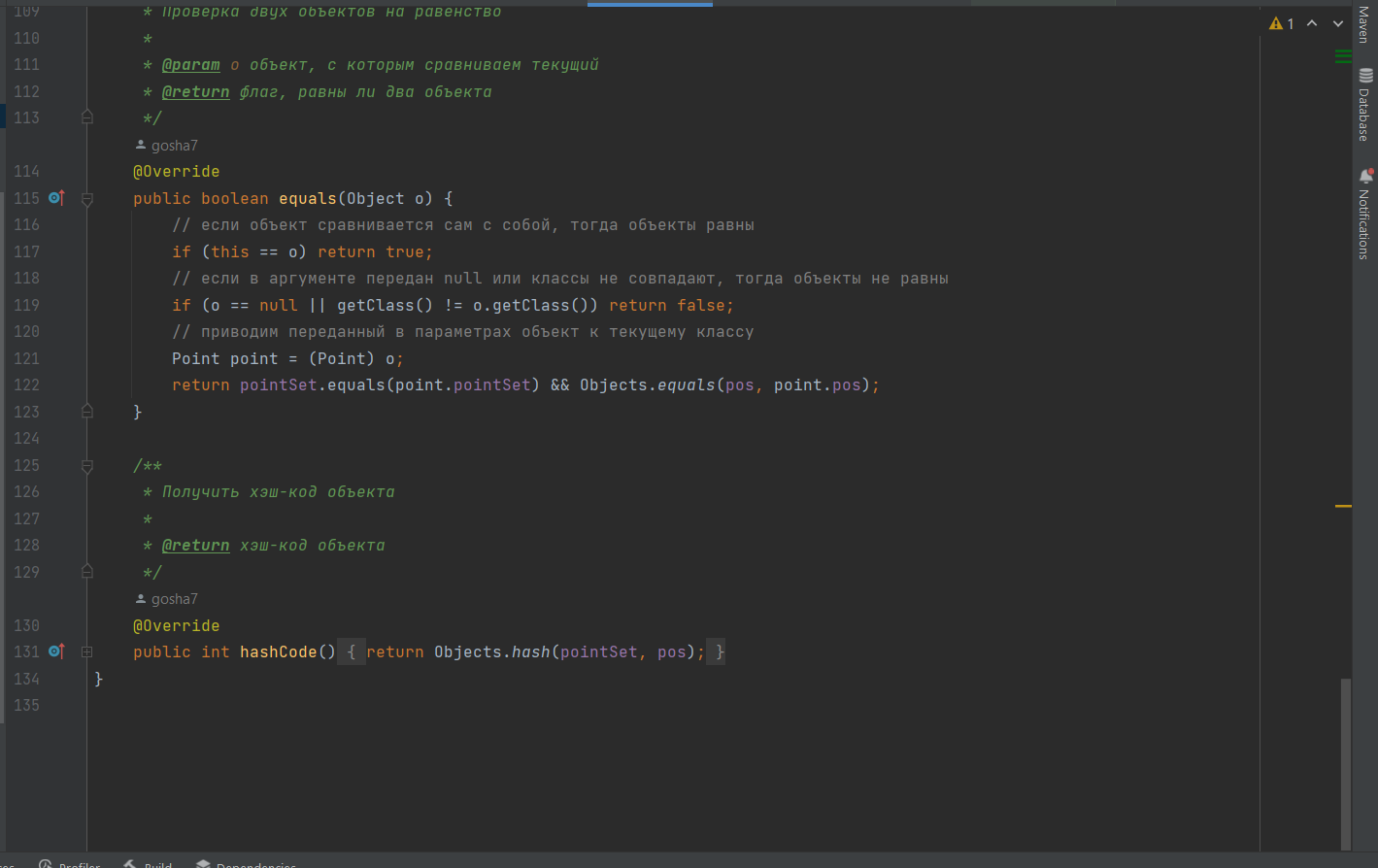
В случае с окружностями и прямыми ситуация была несколько более сложной:нужно было хранить и задавать объекты по нескольким координатам, а также создавать большее количество типов множеств. Пример класса и методов для него на примере **Point.java** представлен ниже.





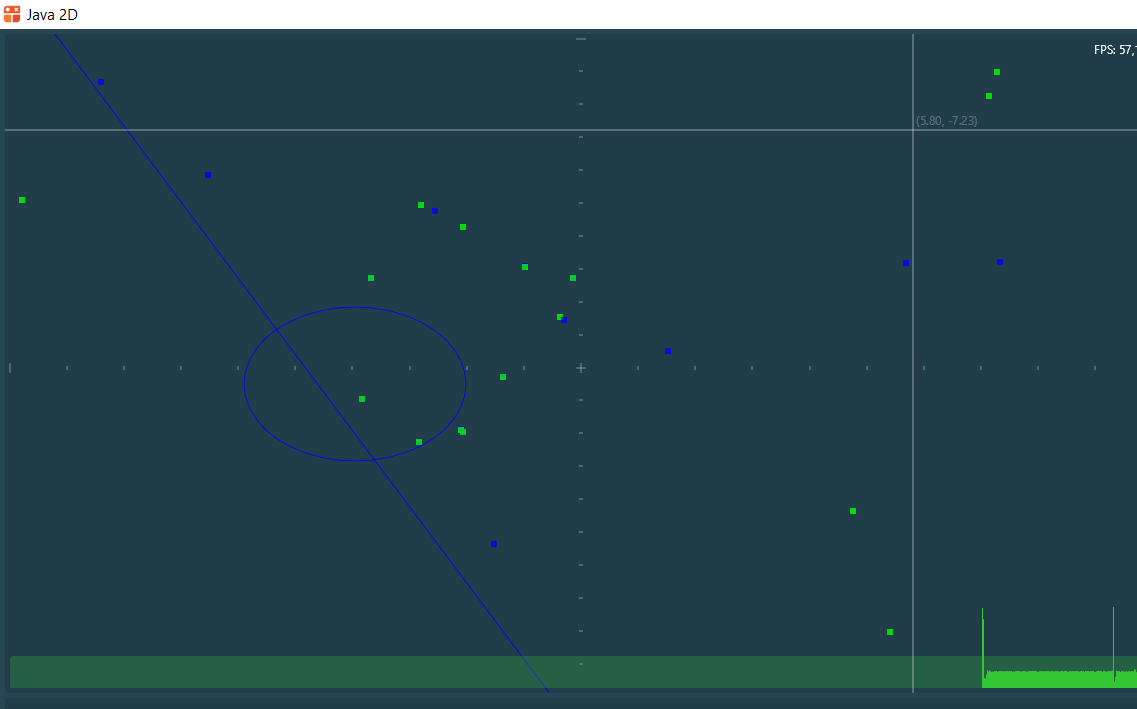






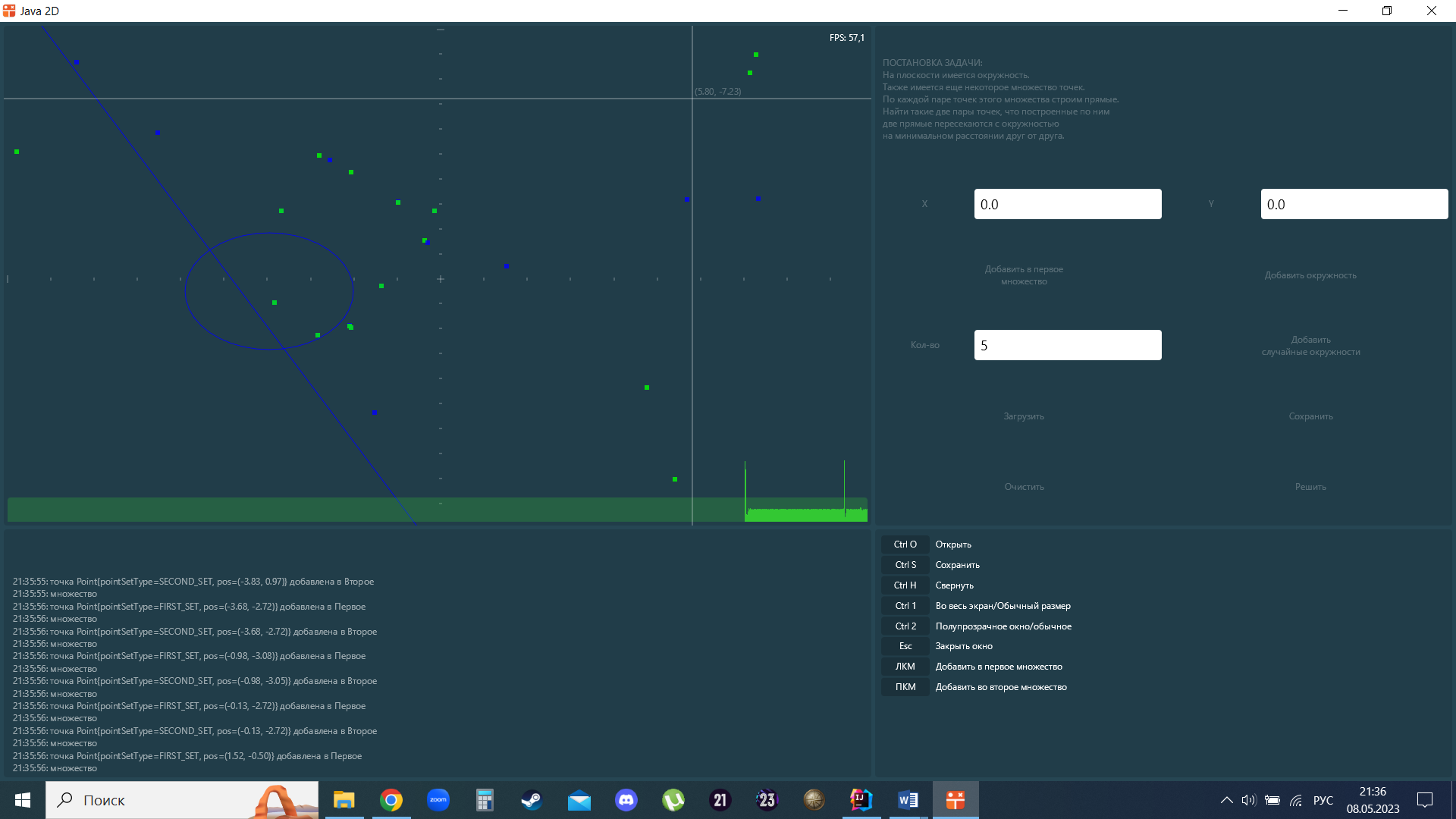
4.Рисование

Чтобы нарисовать точку, использовалась команда рисования прямоугольников **canvas.drawRect().** Аналогичные методы с более сложным методом были применены для рисования окружности и прямой. Ниже будет показана область рисования:



5.Решение задачи

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



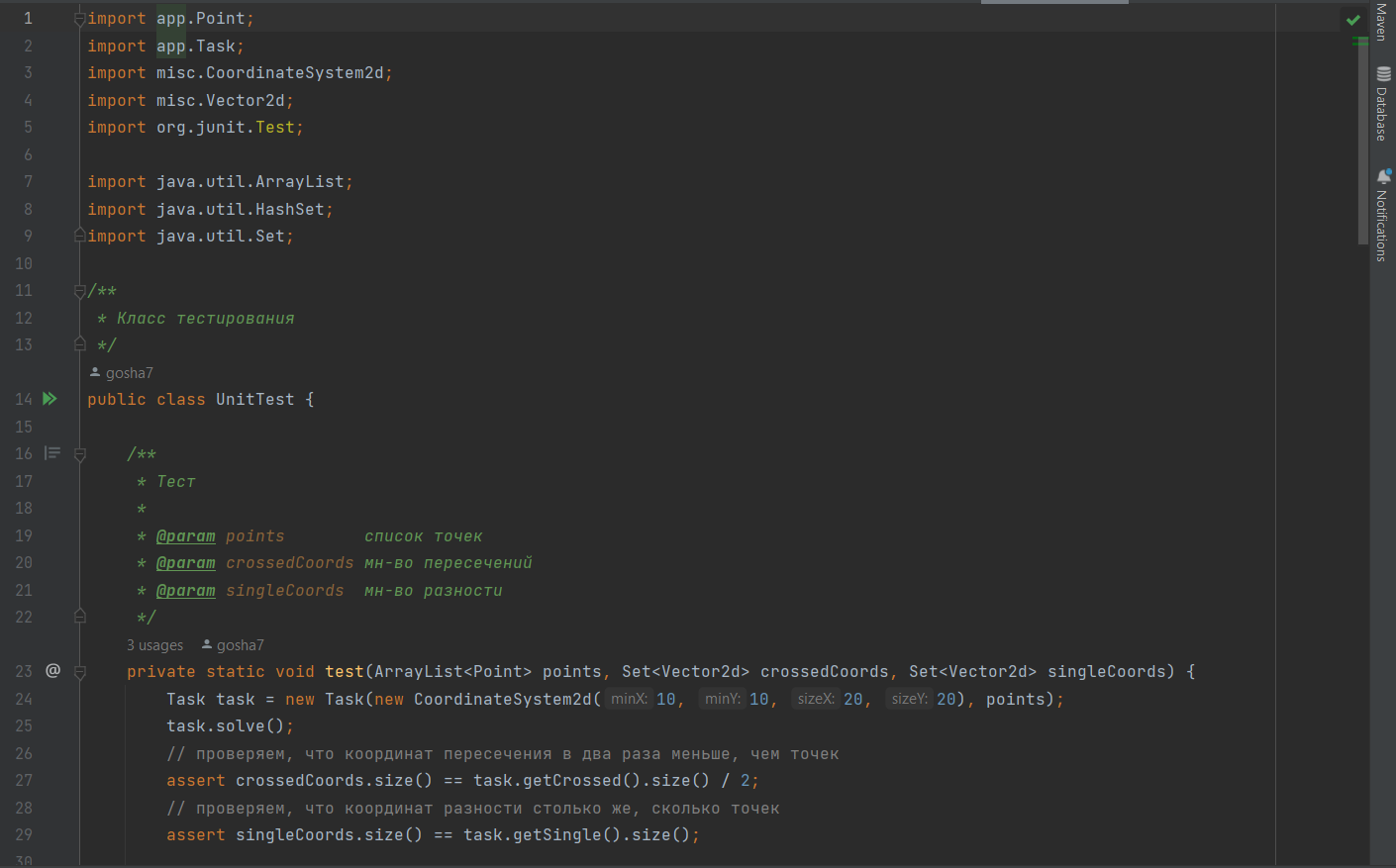
Программа проводит через все точки прямые и с помощью метода поиска координаты пересечения окружности и прямой находит для каждой пары точек нужные координаты. Далее по полученным координатам находит каждое расстояние между этими точками и ищет минимальное, после этого отмечает эти прямые и точки.

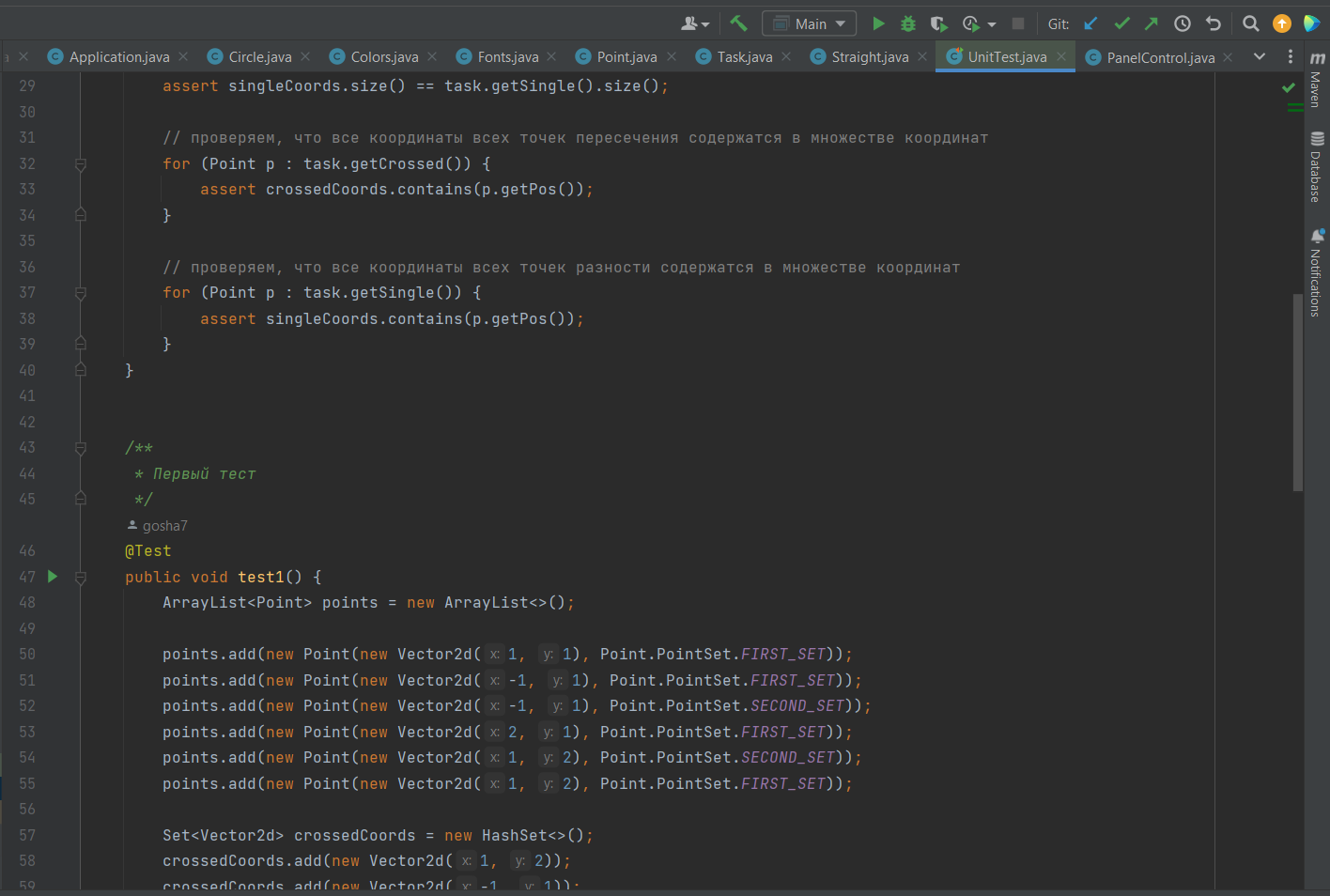
Программа сама решает задачу при нажатии на кнопку решить. Кроме того, можно сохранять файлы с решением задачи, а также загружать нужные файлы с помощью библиотеки json.

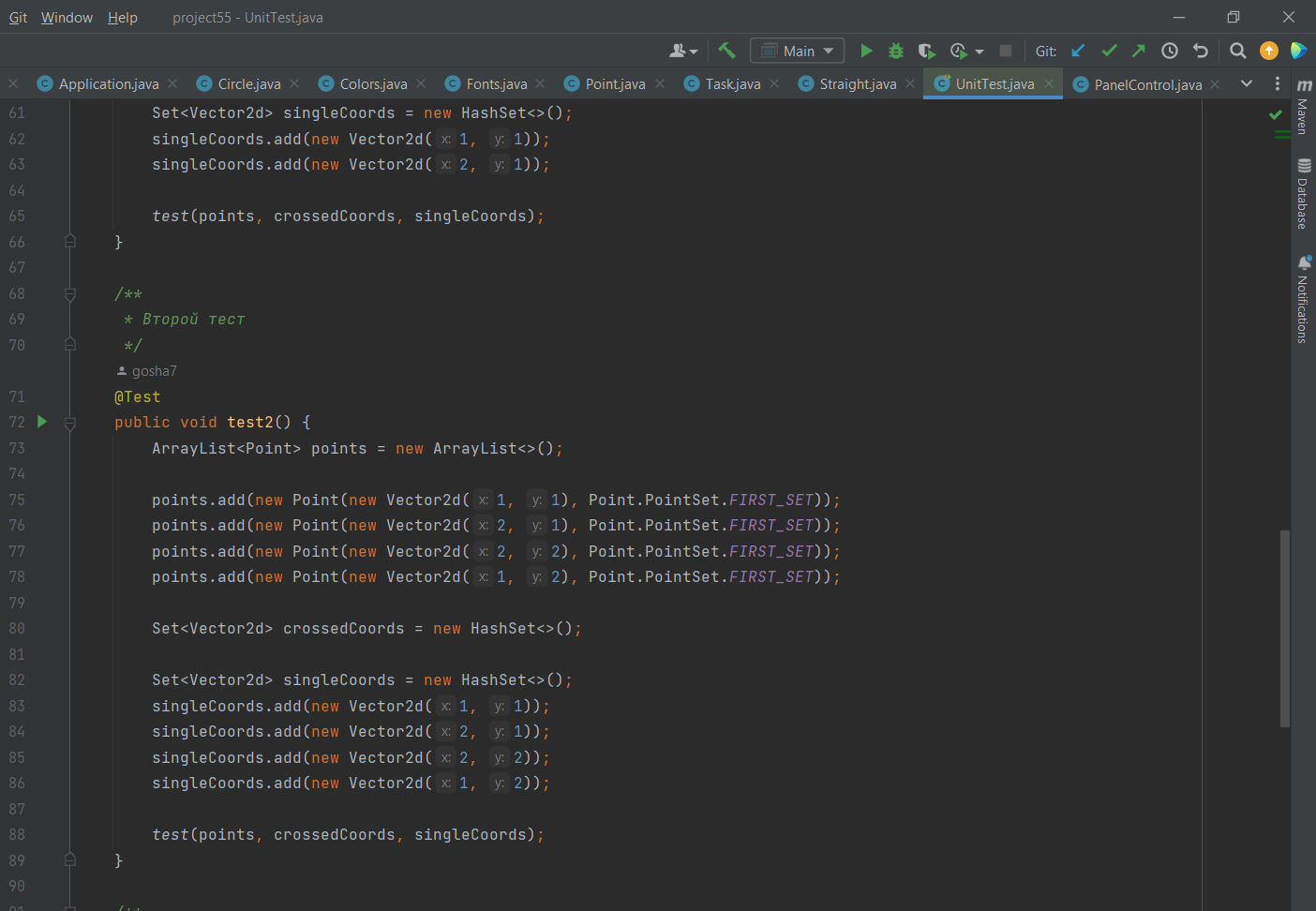
6.Проверка

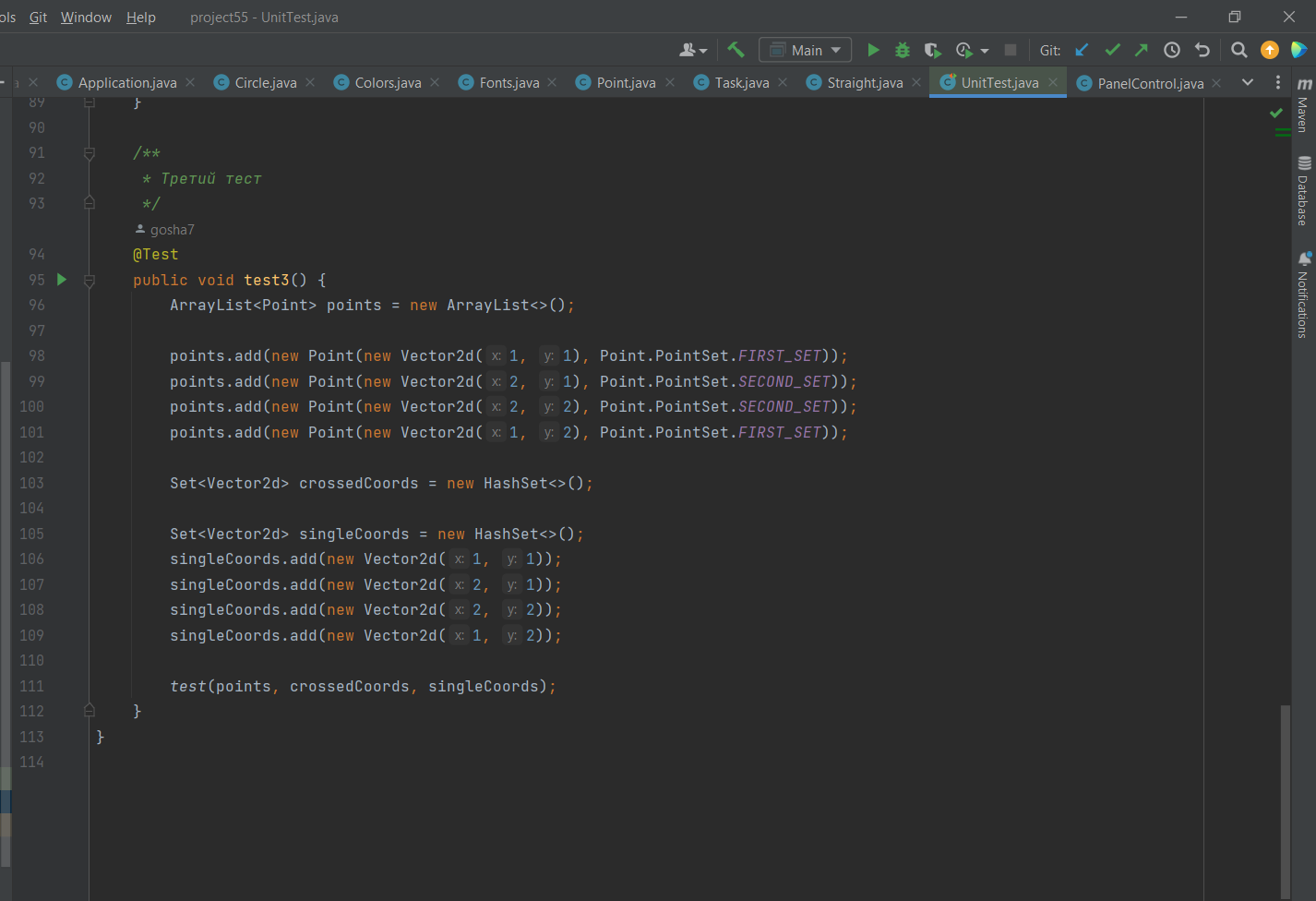
Для проверки правильности решенной задачи были разработаны unit-тесты.

Приведем код этих тестов. Программа верно решила поставленную задачу.









7.Заключение

В рамках выполнения поставленной задачи было создано графическое приложение с требуемым функционалом. Правильность решения задачи проверена с помощью юнит-тестов. Были изучены множество функций и методов, примененные в ходе создания программы.