**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 10-8 класса

Иванов А.И.

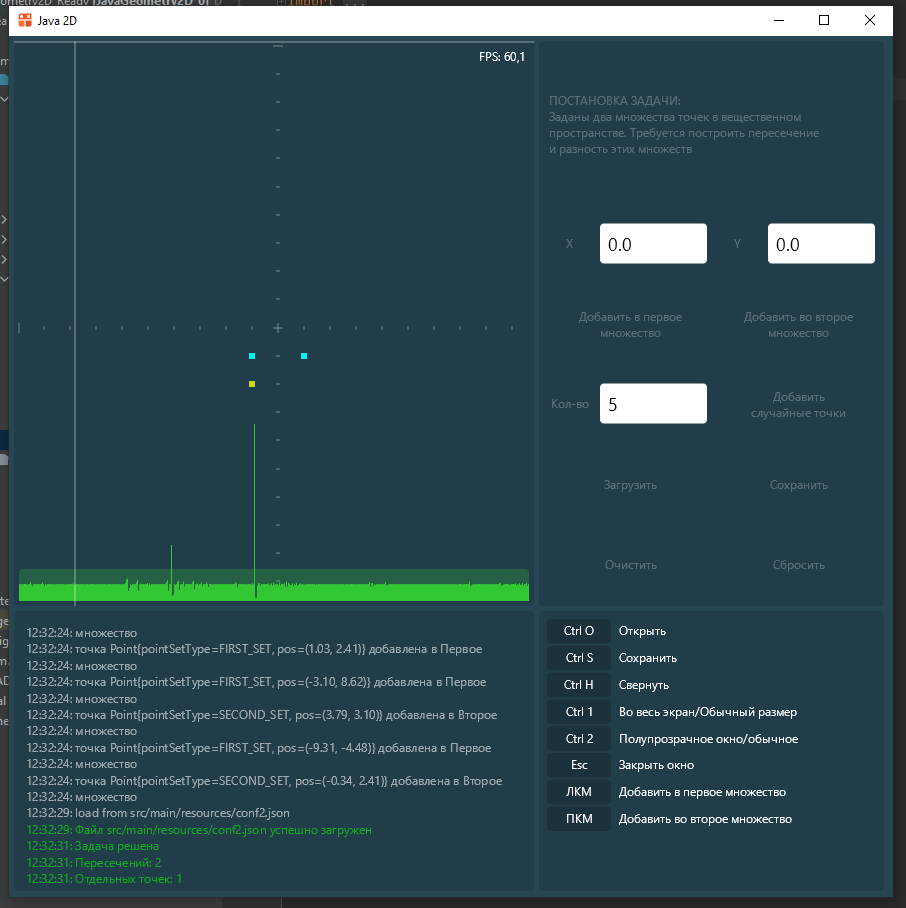
Преподаватель:

Клюнин А.О.

Санкт-Петербург – 2022 год

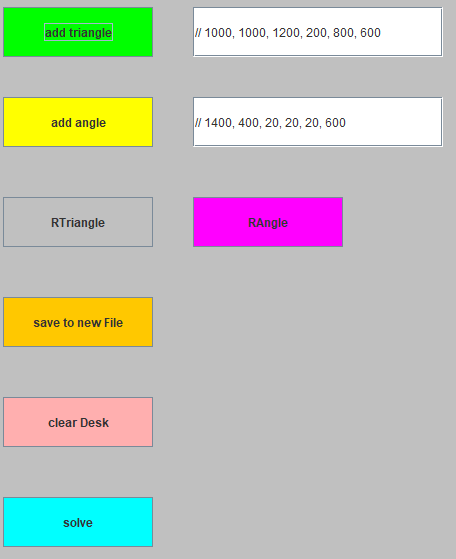
# 1. Постановка задачи

Заданы два множества точек в вещественном пространстве. Требуется построить пересечение и разность этих множеств.



# 2. Элементы управления

В рамках данной задачи необходимо было реализовать следующие элементы управления:

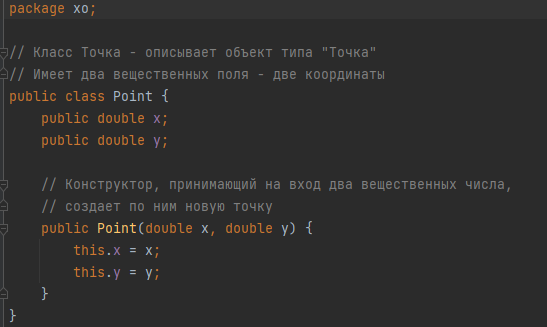


Для добавления точки по координатам было создано два поля ввода: «X» и «Y». Чтобы различить, в какое множество точка должна быть добавлена, используются две кнопки «Добавить в первое множество», «Добавить во второе множество».

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.

3. Структуры данных

Для того чтобы хранить точки, был разработан класс **Point.java.**



# 4. Рисование

Чтобы нарисовать треугольник и угол были разработаны два класса

**AcuteAngle.java** и **Triangle.java**, также класс **Line.java**, рисующий линию по

двум точкам.

# 5. Решение задачи

// Метод, нахождения точек пересечения

// Обозначаем за AB, BC, AC стороны треуг-ка  
 Line AB = new Line(triangle.a, triangle.b);  
 Line BC = new Line(triangle.b, triangle.c);  
 Line AC = new Line(triangle.a, triangle.c);  
 // Обозначаем за firstSide, secondSide стороны угла  
 Line firstSide = new Line(angle.top, new Point(angle.v1.x, angle.v1.y));  
 Line secondSide = new Line(angle.top, new Point(angle.v2.x, angle.v2.y));  
  
 // лежит ли вершина угла внутри треугольника  
 if (*isInsideTheTriangle*(angle.top, triangle)) {  
 *result*.add(angle.top);  
 }  
 // лежат ли какие-то вершины треугольника внутри угла  
 if (*isInsideTheTriangle*(triangle.a, new Triangle(angle))) {  
 *result*.add(triangle.a);  
 }  
  
 if (*isInsideTheTriangle*(triangle.b, new Triangle(angle))) {  
 *result*.add(triangle.b);  
 }  
  
 if (*isInsideTheTriangle*(triangle.c, new Triangle(angle))) {  
 *result*.add(triangle.c);  
 }  
 // И дальше просто добавляем всевозможные точки пересечения  
 addPoint(AB, firstSide);  
 addPoint(BC, firstSide);  
 addPoint(AC, firstSide);  
  
 addPoint(AB, secondSide);  
 addPoint(BC, secondSide);  
 addPoint(AC, secondSide);  
}

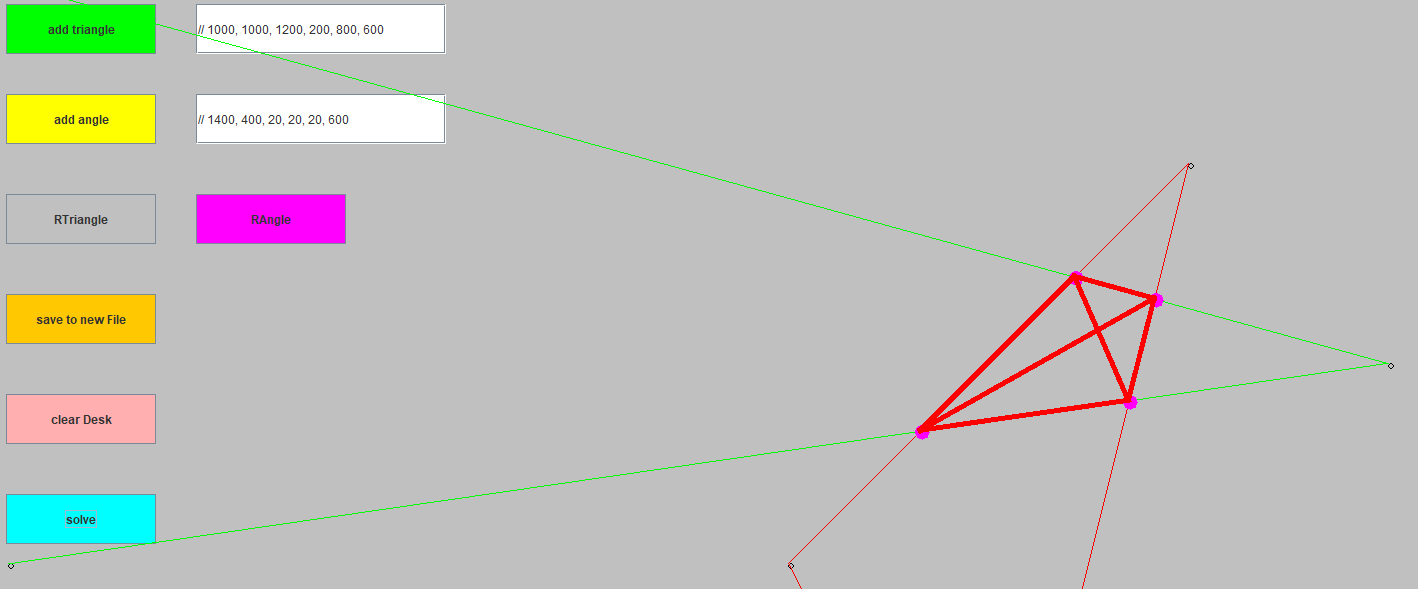
// Метод, рисующий область пересечения угла и треугольника  
private void drawShape(Graphics g, List<Point> points) {  
 g.setColor(Color.*MAGENTA*);  
 int len = points.size();  
 for (int i = 0; i < len; i++) {  
  
 Point p1 = points.get(i); // p1 = points[i]  
 Point p2 = points.get((i + 1) % len); // p2 = points[ (i + 1) % len]  
 g.drawOval((int) (p1.x - 3), (int) (p1.y - 3), 15, 15); // рисуем точку p1

g.fillOval((int) (p1.x - 3), (int) (p1.y - 3), 15, 15); // заливаем точку p1   
 g.drawOval((int) (p2.x - 3), (int) (p2.y - 3), 15, 15); // рисуем точку p2   
 g.fillOval((int) (p2.x - 3), (int) (p2.y - 3), 15, 15); // заливаем точку p2   
 }  
 for (int i = 0; i < len; i++) {  
 Point p1 = points.get(i); // p1 = points[i]  
 for (int j = i; j < len; j++) {  
 Point p2 = points.get(j); // p2 = points[j]  
 g.drawLine((int) p1.x, (int) p1.y, (int) p2.x, (int) p2.y); // рисуем линию (p1, p2)  
 }  
 }

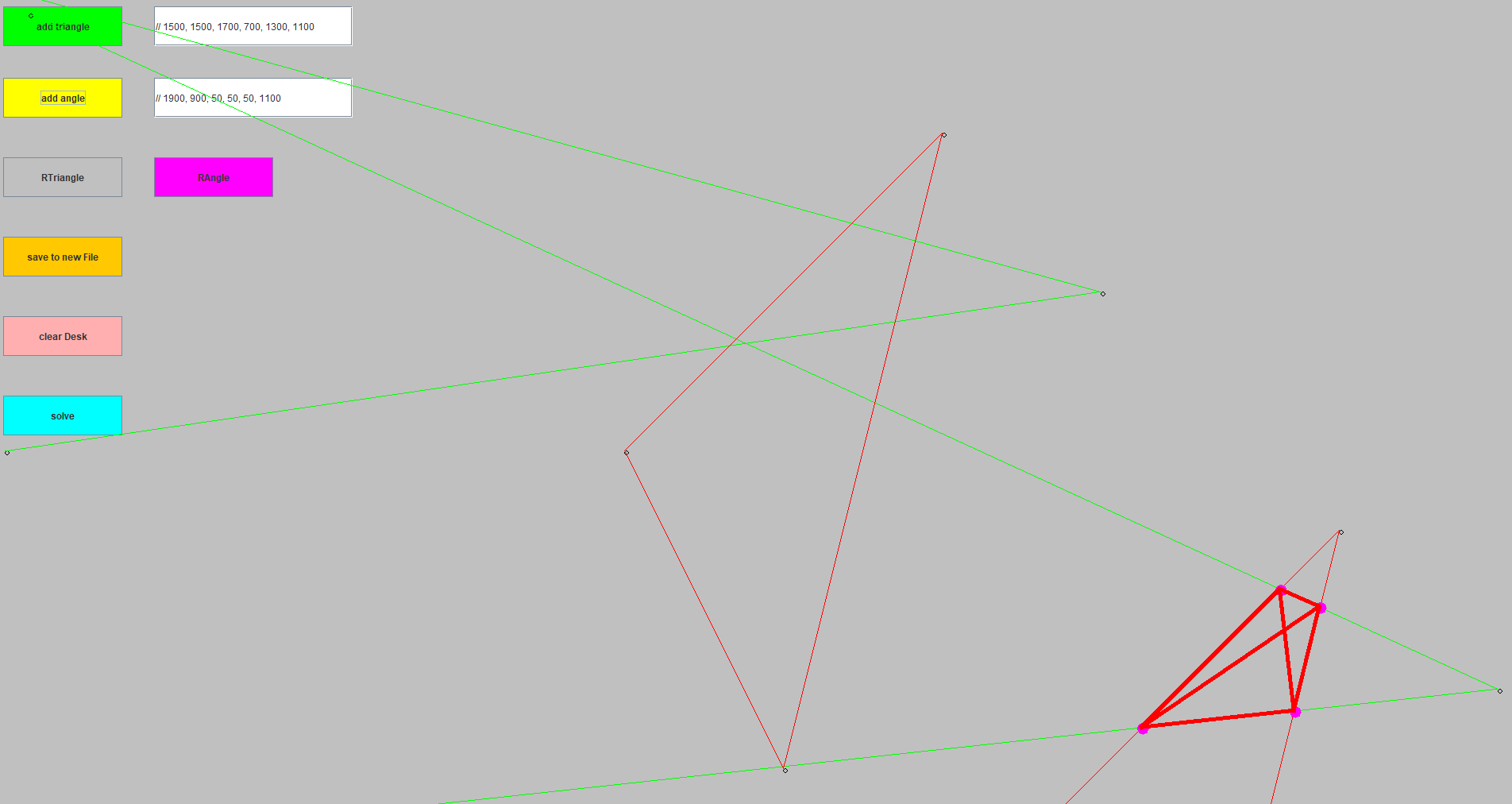
6. Проверка

Для проверки правильности решённой задачи были разработаны unit-тесты.

Тест 1



Тест 2



# 7. Заключение

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

# Приложение А.

# **AcuteAngle.java**

package xo;  
  
// Класс "Острый угол"  
// Как и треугольник, имеет 3 точки - вершина и по одной на каждой стороне  
public class AcuteAngle {  
 public Point top;  
 public Point v1;  
 public Point v2;  
  
 // Создаём острый угол по 3-м точкам, указанным в параметрах  
 public AcuteAngle(Point p1, Point p2, Point p3) {  
 this.top = p1;  
 this.v1 = p2;  
 this.v2 = p3;  
 }  
}

**Triangle.java**

package xo;  
  
// Класс треугольник, описывает треугольник  
// Имеет три вершины - три точки (a, b, c)  
public class Triangle {  
 public Point a;  
 public Point b;  
 public Point c;  
  
 // Создаем треугольник по 3-м точкам, которые были переданы в параметры  
 public Triangle(Point p1, Point p2, Point p3) {  
 this.a = p1;  
 this.b = p2;  
 this.c = p3;  
 }  
  
 // Конструктор, принимает на вход угол,  
 // Создает треугольник по вершине угла и двум его (угла) точкам  
 public Triangle(AcuteAngle angle) {  
 this.a = angle.top;  
 this.b = angle.v1;  
 this.c = angle.v2;  
 }  
}

# **Line.java**

package xo;  
  
// Класс Line - описывает объект типа "Отрезок"  
// Линию задаем по двум точкам  
public class Line {  
 public Point p1;  
 public Point p2;  
  
 // Создаем новую линию по двум точкам,  
 // которые были переданы в параметрах  
 public Line(Point p1, Point p2) {  
 this.p1 = p1;  
 this.p2 = p2;  
 }  
  
}