Комитет по образованию г. Санкт-Петербург

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ПРЕЗИДЕНТСКИЙ ФИЗИКО-МАТЕМАТИЧЕСКИЙ
ЛИЦЕЙ №239

Отчет о практике
«Создание графических приложений на языке Java»

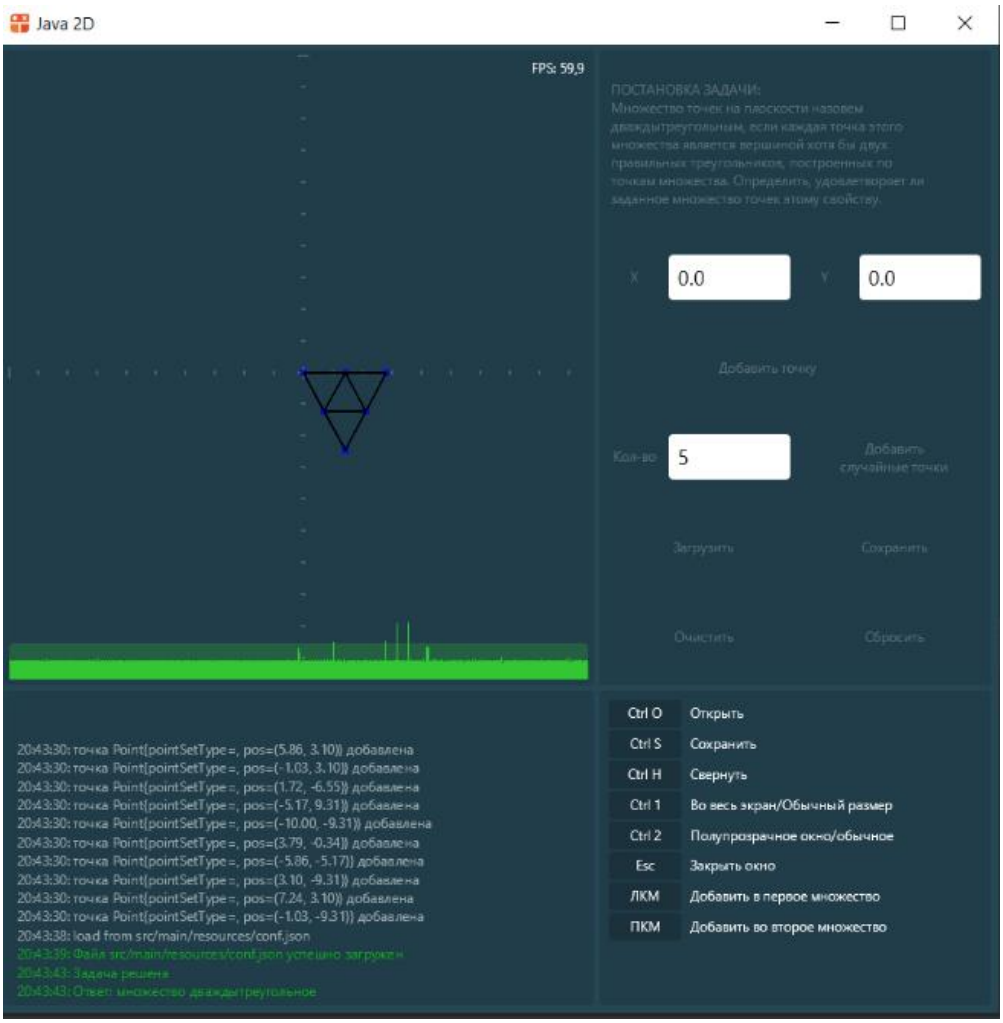
Учащийся 10-3 класса
Иванов А.В.

Преподаватель:
Клюнин А.О.

Санкт-Петербург – 2023 год

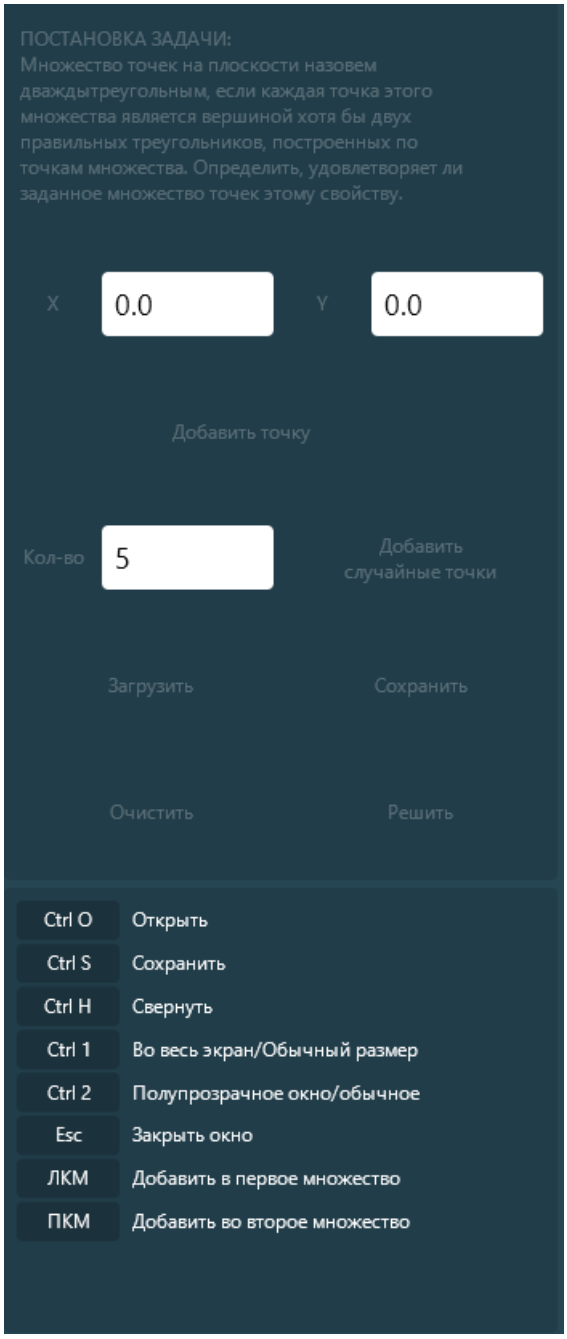
1. Постановка задачи

Множество точек на плоскости назовём дважды треугольным, если каждая точка этого множества является вершиной хотя бы двух правильных треугольников, построенных по точкам множества. Определите, удовлетворяет ли заданное множество точек этому свойству.



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», а также кнопка «Добавить точку», по нажатию которой в область рисования добавляется точка с введёнными координатами.

Т.к. задача предполагает ввод только одного вида 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.

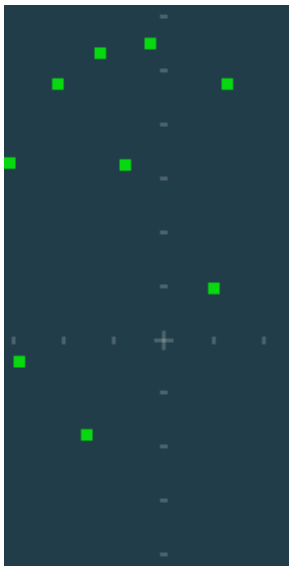
3. Структуры данных

Для того чтобы хранить точки, был разработан класс **Point.java.** Его листинг приведён в приложении А. В него было добавлено поле **pos**, соответствующее положению точки в пространстве задачи.

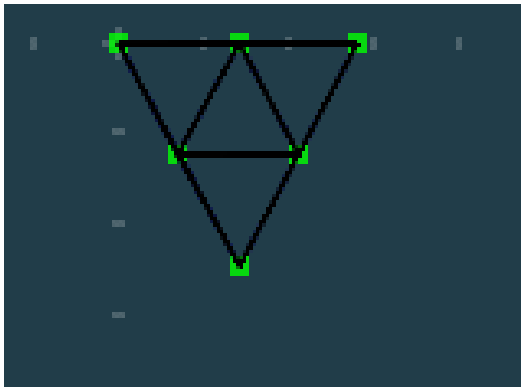
Для хранения треугольников был разработан класс **Triangle.java.** Его листинг приведён в приложении Б. В него были добавлены метод **Triangle** для хранения вершин треугольника и метод **render** для рисования треугольника.

4. Рисование

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

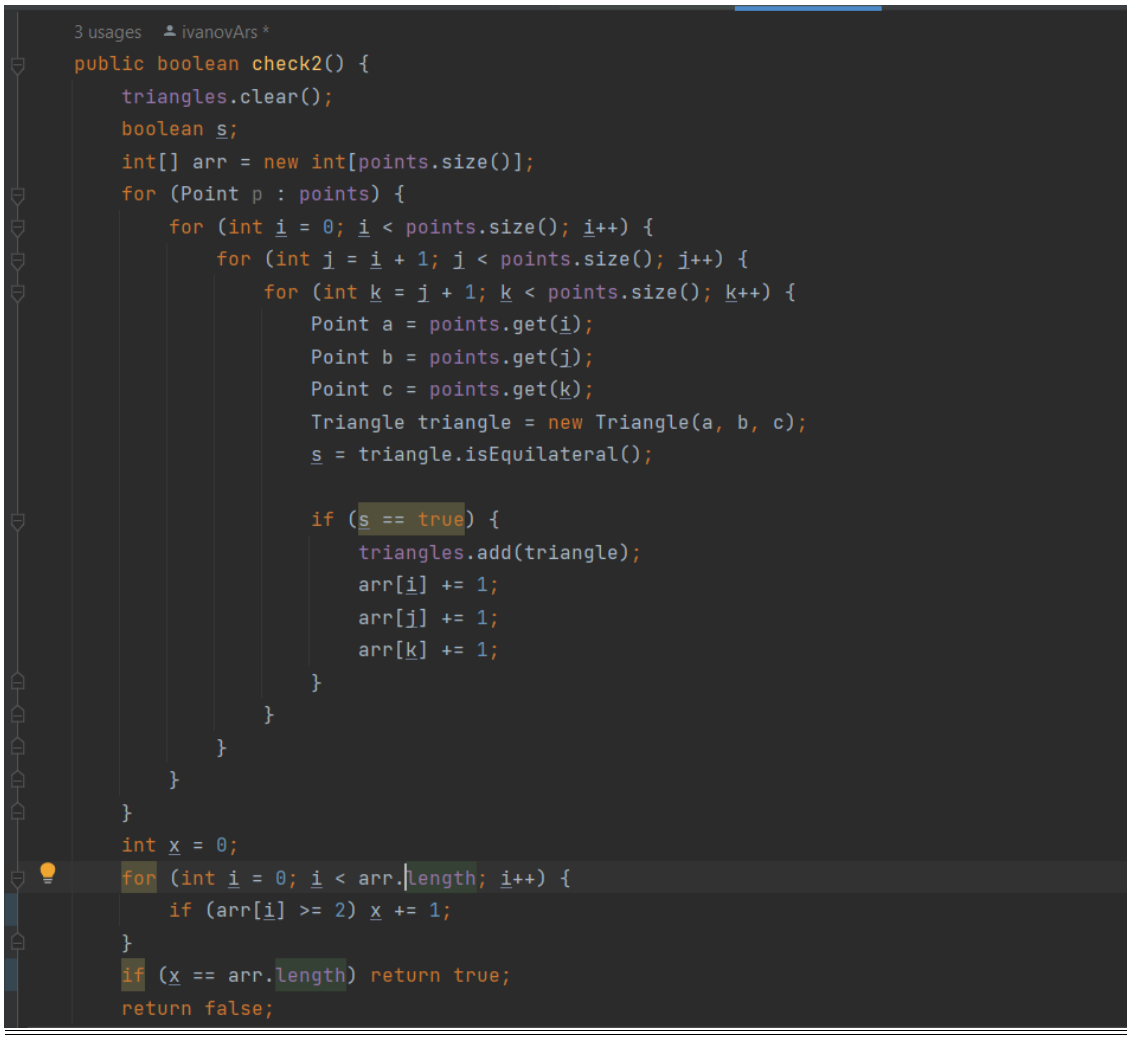


Для рисования треугольников использовалась команда рисования прямых между двумя точками с заданными координатами **canvas.drawLine().**



5. Решение задачи

Для решения поставленной задачи в классе **Task** был разработан метод **check2().**



```
3 usages  ivanovArs *
public boolean check2() {
    triangles.clear();
    boolean s;
    int[] arr = new int[points.size()];
    for (Point p : points) {
        for (int i = 0; i < points.size(); i++) {
            for (int j = i + 1; j < points.size(); j++) {
                for (int k = j + 1; k < points.size(); k++) {
                    Point a = points.get(i);
                    Point b = points.get(j);
                    Point c = points.get(k);
                    Triangle triangle = new Triangle(a, b, c);
                    s = triangle.isEquilateral();

                    if (s == true) {
                        triangles.add(triangle);
                        arr[i] += 1;
                        arr[j] += 1;
                        arr[k] += 1;
                    }
                }
            }
        }
    }
    int x = 0;
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] >= 2) x += 1;
    }
    if (x == arr.length) return true;
    return false;
```

В нём создаётся целочисленный массив, отвечающий за количество равносторонних треугольников, вершиной которых является та или иная точка множества (по умолчанию все элементы начального массива равны 0). Затем в методе перебираются все тройки точек. Если треугольник, образуемый тремя какими-то точками, равносторонний (расстояния между всеми парами точек равны между собой), для всех этих точек счётчик правильных треугольников увеличивается на 1 (элементы массива с соответствующими индексами увеличиваются на 1).

Если после перебора всех возможных троек количество правильных треугольников для каждой точки больше 1, это означает, что каждая точка является вершиной минимум двух равносторонних треугольников, а значит данное множество дваждытреугольное. В противном случае – нет.

6. Проверка

Для проверки правильности решённой задачи были разработаны unit-тесты. Их листинг приведён в приложении В.

Тест 1

Точки : $\{(0, 0); (2.829, 0); (1.415, 2.449); (0.707, 1.225); (1.415, 0); (2.121, 1.225)\}$

Ответ : множество дваждытреугольное

Тест 2

Точки : $\{(0, 0); (6, 0); (1.3, 2.7); (0.1, 1.2); (1, 0); (2.5, 2.3)\}$

Ответ : множество недваждытреугольное

7. Заключение

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

Приложение А. Point.java

```java
package app;

import ...

/**
 * Класс точки
 */
60 usages  ivanovav.24 *
public class Point {
    /**
     * Координаты точки
     */
    public final Vector2d pos;

    /**
     * Конструктор точки
     *
     * @param pos     положение точки
     */
    ivanovav.24
    @JsonCreator
    public Point(@JsonProperty("pos") Vector2d pos) { this.pos = pos; }


    /**
     * Получить цвет точки по её множеству
     *
     * @return цвет точки
     */
    1 usage  ivanovav.24 *
    @JsonIgnore
```

```java
    @JsonIgnore
    public int getColor() { return Misc.getColor( a: 0xCC,  r: 0x00,  g: 0xFF,  b: 0x00); }

    /**
     * Получить положение
     * (нужен для json)
     *
     * @return положение
     */
    ivanovav.24
    public Vector2d getPos() { return pos; }


    /**
     * Строковое представление объекта
     *
     * @return строковое представление объекта
     */
    ivanovav.24
    @Override
    public String toString() {
        return "Point{" +
                "pointSetType=" +
                ", pos=" + pos +
                '}';
    }

    /**
     * Проверка двух объектов на равенство
     *
     * @param o объект, с которым сравниваем текущий
     * @return флаг, равны ли два объекта
```

```java
    /**
     * Проверка двух объектов на равенство
     *
     * @param o объект, с которым сравниваем текущий
     * @return флаг, равны ли два объекта
     */
    ivanovav.24
    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Point point = (Point) o;
        return equals(point) && Objects.equals(pos, point.pos);
    }

    /**
     * Получить хэш-код объекта
     *
     * @return хэш-код объекта
     */
    ivanovav.24
    @Override
    public int hashCode() { return Objects.hash(pos); }
    5 usages
    static final int POINT_SIZE = 3;
}
```

Приложение Б. Triangle.java

```java
package app;

import ...

16 usages  ivanovav.24 +1 *
public class Triangle {
    @Getter
    Point a;
    @Getter
    Point b;
    @Getter
    Point c;


    ivanovav.24
    @JsonCreator
    public Triangle(@JsonProperty("a") Point a, @JsonProperty("b") Point b, @JsonProperty("c") Point c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }


    /**
     * Получить цвет точки по её множеству
     *
     * @return цвет точки
     */
    1 usage  ivanovav.24 *
    @JsonIgnore
    public int getColor() { return Misc.getColor( a: 0xCC,  r: 0x00,  g: 0xFF,  b: 0x00); }
```

```java
    /**
     * Строковое представление объекта
     *
     * @return строковое представление объекта
     */
    ivanovav.24
    @Override
    public String toString() {
        return "Point{" +
                "pointSetType=" +
                ", a=" + a.getPos() + ", b=" + b.getPos() + " ,c=" + c.getPos() +
                '}';
    }


    ivanovav.24
    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Triangle triangle = (Triangle) o;
        return Objects.equals(a, triangle.a) && Objects.equals(b, triangle.b) && Objects.equals(c, triangle.c);
    }

    ivanovav.24
    @Override
    public int hashCode() { return Objects.hash(a, b, c); }

    5 usages  ivanovav.24
    public boolean isEquilateral() {
        double dx1 = a.pos.x - b.pos.x;
        double dy1 = a.pos.y - b.pos.y;
        double dx2 = b.pos.x - c.pos.x;
        double dy2 = b.pos.y - c.pos.y;
        double dx3 = c.pos.x - a.pos.x;
        double dy3 = c.pos.y - a.pos.y;

        System.out.println(dx1 * dx1 + dy1 * dy1);
        System.out.println(dx2 * dx2 + dy2 * dy2);
        System.out.println(dx3 * dx3 + dy3 * dy3);

        return Math.abs((dx1 * dx1 + dy1 * dy1) - (dx2 * dx2 + dy2 * dy2)) > 0.01 ||
                Math.abs((dx2 * dx2 +
                Math.abs((dx3 * dx3 +
    }

    1 usage  ivanovArs
    public void render(Canvas canvas, CoordinateSystem2i windowCS, CoordinateSystem2d ownCS) {
        try (Paint paint = new Paint()) {
            // вершины треугольника
            Vector2i pointA = windowCS.getCoords(a.pos, ownCS);
            Vector2i pointB = windowCS.getCoords(b.pos, ownCS);
            Vector2i pointC = windowCS.getCoords(c.pos, ownCS);
            // рисуем его стороны
            canvas.drawLine(pointA.x, pointA.y, pointB.x, pointB.y, paint);
            canvas.drawLine(pointB.x, pointB.y, pointC.x, pointC.y, paint);
            canvas.drawLine(pointC.x, pointC.y, pointA.x, pointA.y, paint);
        }
    }
}
```

```
double dy1 = a.pos.y - b.pos.y
untitled1
```

Приложение В. UnitTest.java

```java
import ...

ivanovArs +1
public class UnitTest {
    // тест на равносторонность треугольников (и равносторонние, и не равносторонние) - хотя бы два
    // тест на дважды треугольность множеств точек (и дважды треугольные, и недважды треугольные)
    // Причём несколько тестов: 1 - нет треугольников, 2 - однотреугольное, 3 - частично двухтреугольные
    // 4 - полность двухтрегуольное множество
    ivanovav.24
    @Test
    public void test1() {
        Triangle triangle = new Triangle(
                new Point(new Vector2d( x: 0,  y: 0)),
                new Point(new Vector2d( x: 2.829,  y: 0)),
                new Point(new Vector2d( x: 1.415,  y: 2.449))
        );
        assert triangle.isEquilateral();
    }
    ivanovArs
    @Test
    public void test2() { // дважды треугольное множество
        Triangle triangle = new Triangle(
                new Point(new Vector2d( x: 0,  y: 0)),
                new Point(new Vector2d( x: 10,  y: 0)),
                new Point(new Vector2d( x: 1.415,  y: 2.449))
        );
        assert !triangle.isEquilateral();
    }
```

```java
public void test3() { // однотреугольное множество
    CoordinateSystem2d cs = new CoordinateSystem2d( minX: -10,  minY: -10,  sizeX: 20,  sizeY: 20);
    ArrayList<Point> points = new ArrayList<>();
    points.add(new Point(new Vector2d( x: 0,  y: 2.449)));
    points.add(new Point(new Vector2d( x: 2.829,  y: 2.449)));
    points.add(new Point(new Vector2d( x: 1.415,  y: 4.898)));
    points.add(new Point(new Vector2d( x: 1.415,  y: 0)));
    Task t = new Task(cs, points);
    assert t.check1();
}
ivanovArs
@Test
public void test4() { // однотреугольное множество
    CoordinateSystem2d cs = new CoordinateSystem2d( minX: -10,  minY: -10,  sizeX: 20,  sizeY: 20);
    ArrayList<Point> points = new ArrayList<>();
    points.add(new Point(new Vector2d( x: 0,  y: 0)));
    points.add(new Point(new Vector2d( x: 1,  y: 1)));
    point                                                    );
    point
    Task
    assert !t.check1();
}
ivanovArs
@Test
public void test5() { // дваждытреугольное множество
    CoordinateSystem2d cs = new CoordinateSystem2d( minX: -10,  minY: -10,  sizeX: 20,  sizeY: 20);
    ArrayList<Point> points = new ArrayList<>();
    points.add(new Point(new Vector2d( x: 0,  y: 0)));
    points.add(new Point(new Vector2d( x: 2.829,  y: 0)));
    points.add(new Point(new Vector2d( x: 1.415,  y: 2.449)));
    points.add(new Point(new Vector2d( x: 0.707,  y: 1.225)));
```

ArrayList<Point> points = new ArrayList<Point>()
untitled1

```java
public void test5() { // дваждытреугольное множество
    CoordinateSystem2d cs = new CoordinateSystem2d( minX: -10,  minY: -10,  sizeX: 20,  sizeY: 20);
    ArrayList<Point> points = new ArrayList<>();
    points.add(new Point(new Vector2d( x: 0,  y: 0)));
    points.add(new Point(new Vector2d( x: 2.829,  y: 0)));
    points.add(new Point(new Vector2d( x: 1.415,  y: 2.449)));
    points.add(new Point(new Vector2d( x: 0.707,  y: 1.225)));
    points.add(new Point(new Vector2d( x: 1.415,  y: 0)));
    points.add(new Point(new Vector2d( x: 2.121,  y: 1.225)));
    Task t = new Task(cs, points);
    assert t.check2();
}
ivanovArs
@Test
public void test6() { // дваждытреугольное множество
    CoordinateSystem2d cs = new CoordinateSystem2d( minX: -10,  minY: -10,  sizeX: 20,  sizeY: 20);
    ArrayList<Point> points = new ArrayList<>();
    points.add(new Point(new Vector2d( x: 0,  y: 0)));
    points.add(new Point(new Vector2d( x: 6,  y: 0)));
    points.add(new Point(new Vector2d( x: 1.3,  y: 2.7)));
    points.add(new Point(new Vector2d( x: 0.1,  y: 1.2)));
    points.add(new Point(new Vector2d( x: 1,  y: 0)));
    points.add(new Point(new Vector2d( x: 2.5,  y: 2.3)));
    Task t = new Task(cs, points);
    assert !t.check2();
}
}
```