

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
Чашин А. Е.

Преподаватель:
Клюнин А.О.

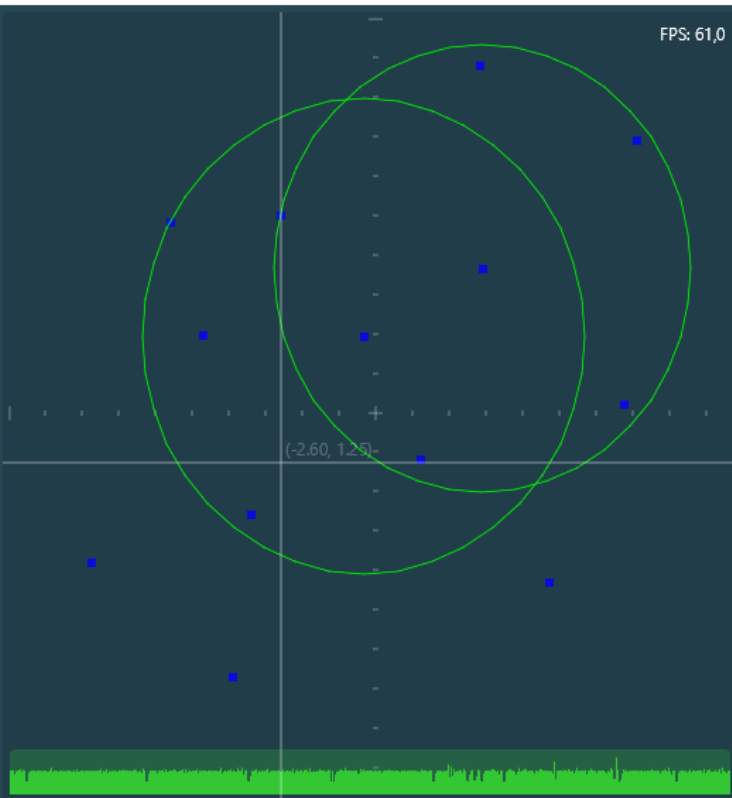
Санкт-Петербург – 2023 год

1. Постановка задачи

На плоскости задано множество точек. Найти такие две окружности, что их центры находятся в точках заданного множества, внутри каждой из этих окружностей находятся хотя бы половина из всех точек заданного множества, и меньший из двух радиусов минимален.

Java 2D

FPS: 61,0



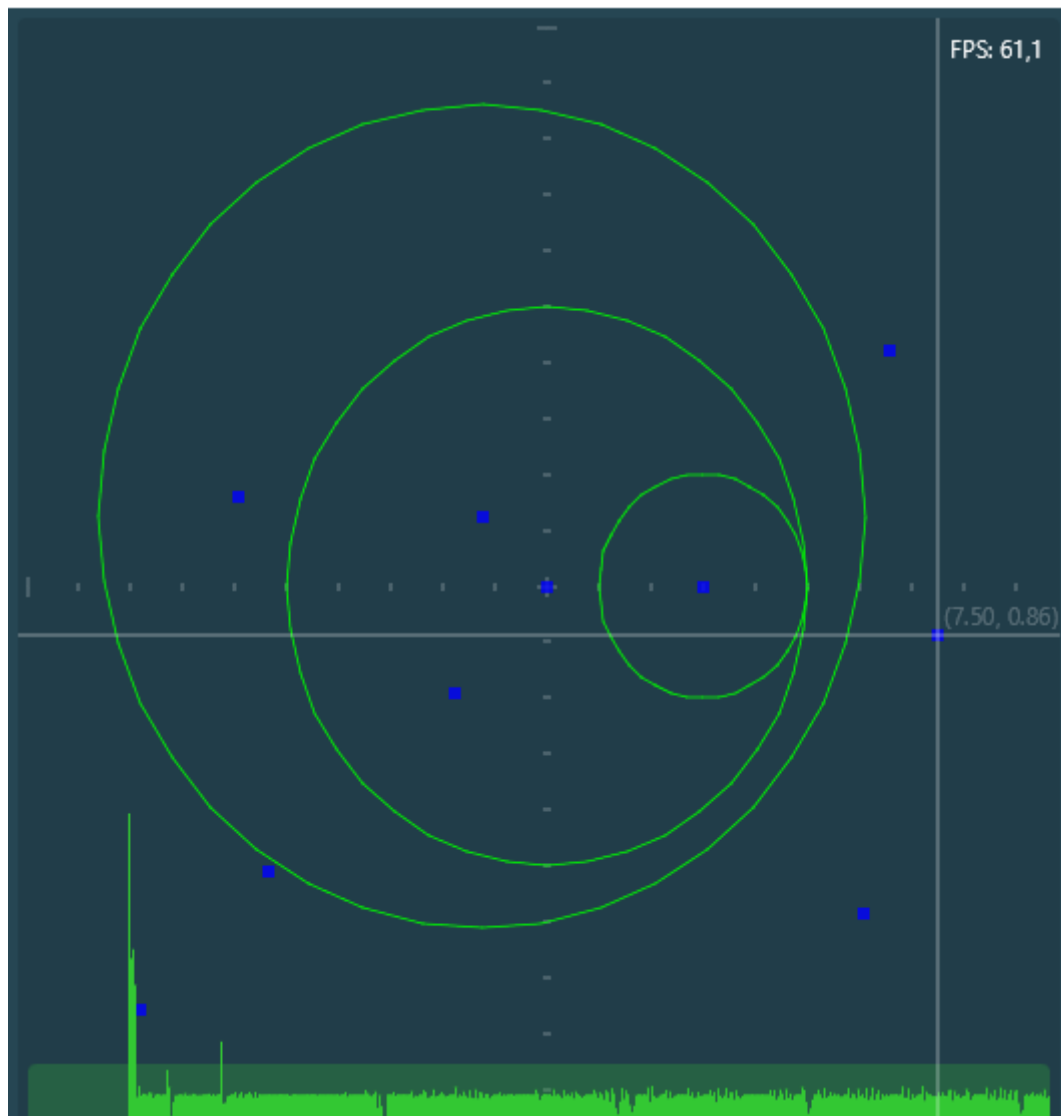
15:02:50: точка Point{pos=(-5.61, -4.84)} добавлена
15:02:51: точка Point{pos=(2.91, -3.66)} добавлена
15:02:52: точка Point{pos=(1.21, 1.18)} добавлена
15:02:52: точка Point{pos=(-3.41, 2.58)} добавлена
15:02:52: точка Point{pos=(4.72, 4.30)} добавлена
15:02:53: точка Point{pos=(6.76, -0.22)} добавлена
15:02:53: точка Point{pos=(-0.33, -1.94)} добавлена
15:02:54: точка Point{pos=(-3.91, 6.70)} добавлена
15:02:55: точка Point{pos=(-7.76, 3.80)} добавлена
15:02:55: точка Point{pos=(-4.72, -1.97)} добавлена
15:02:55: точка Point{pos=(2.83, -8.78)} добавлена
15:02:56: точка Point{pos=(7.11, -6.92)} добавлена
15:02:57: точка Point{pos=(-2.60, -5.02)} добавлена
15:02:58: Задача решена
15:02:58: окружность №1:Point{center=(2.91, -3.66) radius=5.676420188578931} окружность
15:02:58: №2:Point{center=(-0.33, -1.94) radius=6.024998712095019}

ПОСТАНОВКА ЗАДАЧИ:
На плоскости задано множество точек. Найти такие две окружности, что их центры находятся в точках заданного множества, внутри каждой из этих окружностей находятся хотя бы половина из всех точек заданного множества, и меньший из двух радиусов минимален.

R	Отменить решение
S	Решить
Ctrl H	Свернуть
Ctrl 1	Во весь экран/Обычный размер
Ctrl 2	Полупрозрачное окно/обычное
Esc	Заккрыть окно
ЛКМ/ПКМ	Добавить точку

2. Элементы управления

Программа позволяет добавлять точки с помощью мыши. При клике на область рисования, появляется новая точка.



3. Структуры данных

Для того чтобы хранить точки, был разработан класс **Point.java.**, окружности - **Circle.java** Их листинг приведён в приложении А.

У окружности были добавлены поля **pos**, соответствующее положению центра окружности в пространстве задачи и её радиус - **r**.

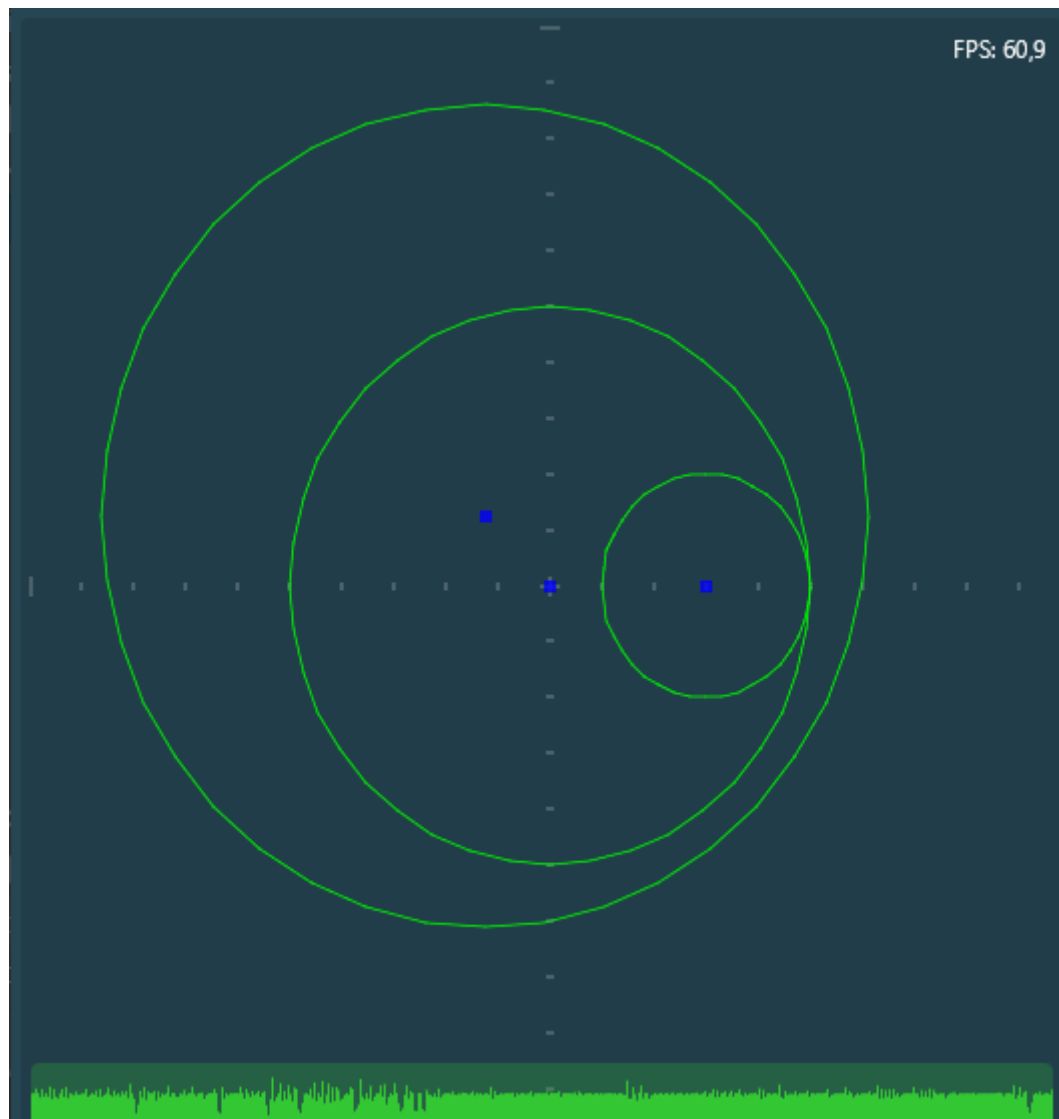
У точки было добавлено поле **pos**, соответствующее положению точки в пространстве задачи.

4. Рисование

Для рисования точки использовалась команда **canvas.drawRect(...)**

Для рисования окружности команда **paint()** просчитывала точки, которые, соединив их линиями, образовывали заданную окружность.

Для рисования линий использовалась команда **canvas.drawLines(...)**



6. Решение

Для каждой точки создать список расстояний до всех остальных. Отсортировать эти списки. Числа в центре этих списков и будут минимальные радиусы окружностей, с центрами в соответствующей точке, которые захватывают хотя бы половину всех точек. После найти 2 минимальных радиуса и по соответствующим точкам построить окружности.

7. Заключение

В рамках выполнения поставленной задачи было создано графическое приложение с требуемым функционалом.

Приложение А. Point.java; Circle.java

```
package app;

import misc.Misc;
import misc.Vector2d;

import java.util.Objects;

/**
 * Класс точки
 */
public class Point {
    /**
     * Координаты точки
     */
    public final Vector2d pos;

    /**
     * Конструктор точки
     *
     * @param pos положение точки
     */
    public Point(Vector2d pos) {
        this.pos = pos;
    }

    /**
     * Получить цвет точки
     *
     * @return цвет точки
     */
    public int getColor() {
        return Misc.getColor(0xCC, 0x00, 0x00, 0xFF);
    }

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

    /**
     * Получить хэш-код объекта
     *
     * @return хэш-код объекта
     */
    @Override
    public int hashCode() {
        return Objects.hash(pos);
    }
}
```

```

package app;

import misc.Misc;
import misc.Vector2d;

import java.util.Objects;

/**
 * Класс окружности
 */
public class Circle {
    /**
     * Координаты центра
     */
    public final Vector2d pos;

    /**
     * радиус
     */
    public final double r;

    /**
     * Конструктор окружности
     *
     * @param pos    положение окружности
     * @param r      радиус окружности
     */
    public Circle(Vector2d pos, double r) {
        this.pos = pos;
        this.r = r;
    }

    /**
     * Получить цвет окружности
     *
     * @return цвет окружности
     */
    public int getColor() {
        return Misc.getColor(0xCC, 0x00, 0xFF, 0x0);
    }

    /**
     * получить точки для рисования окружности
     * @return точки
     */
    public float[] paint() {
        int loopCnt = 40;
        // создаём массив координат опорных точек
        float[] points = new float[loopCnt * 4];
        for (int i = 0; i < loopCnt; ++i) {
            // x координата первой точки
            points[i * 4] = (float) (pos.x + r * Math.cos(Math.PI / 20 * i));
            // y координата первой точки
            points[i * 4 + 1] = (float) (pos.y + r * Math.sin(Math.PI / 20 *
i));

            // x координата второй точки
            points[i * 4 + 2] = (float) (pos.x + r * Math.cos(Math.PI / 20 *
(i + 1)));
            // y координата точки
            points[i * 4 + 3] = (float) (pos.y + r * Math.sin(Math.PI / 20 *
(i + 1)));
        }
    }
}

```

```
        return points;
    }

    /**
     * Строковое представление объекта
     *
     * @return строковое представление объекта
     */
    @Override
    public String toString() {
        return "Point{" +
            "center=" + pos +
            " radius=" + r +
            '}';
    }

    /**
     * Получить хэш-код объекта
     *
     * @return хэш-код объекта
     */
    @Override
    public int hashCode() {
        return Objects.hash(pos, r);
    }
}
```