**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.

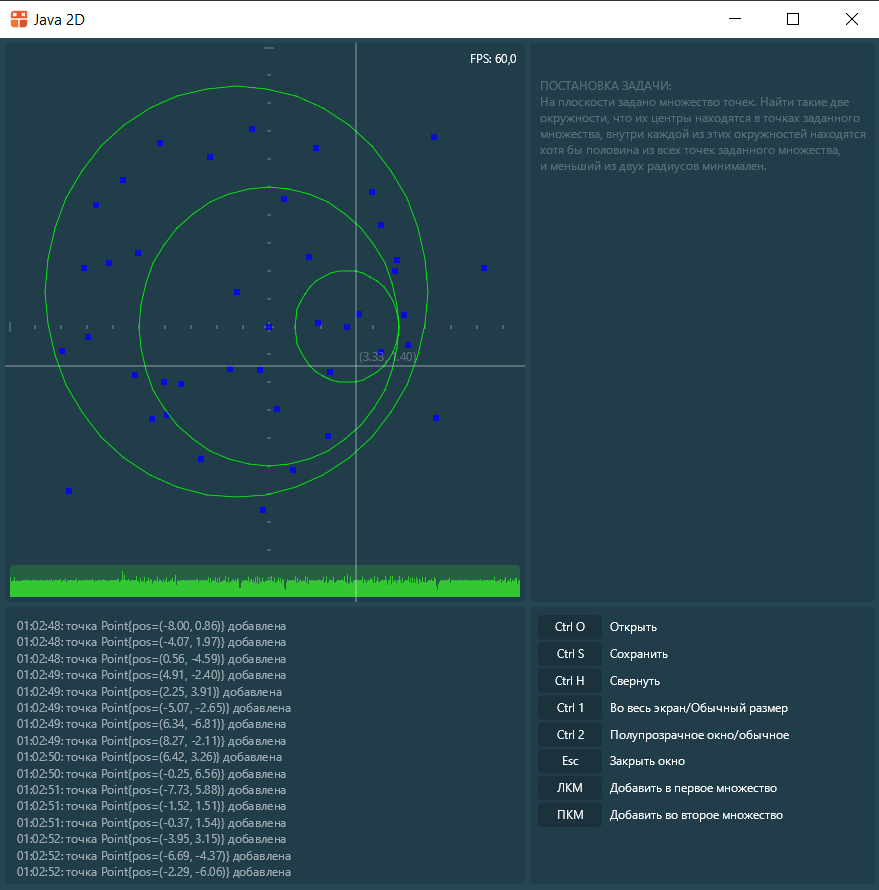
Преподаватель:

Клюнин А.О.

Санкт-Петербург – 2023 год

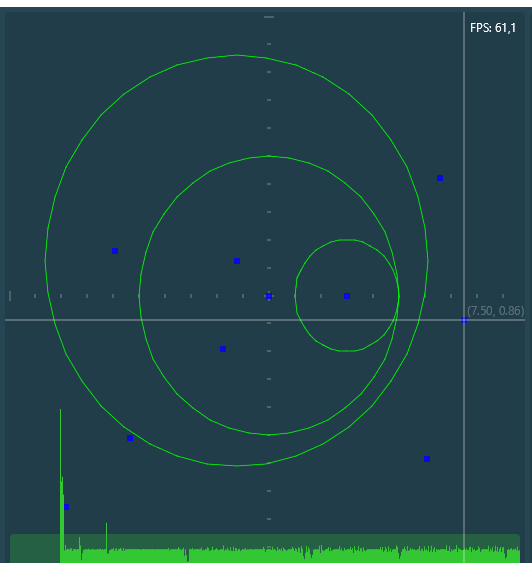
# 1. Постановка задачи

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.



# 2. Элементы управления

Программа позволяет добавлять точки с помощью мыши. При клике на область рисование, 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.**

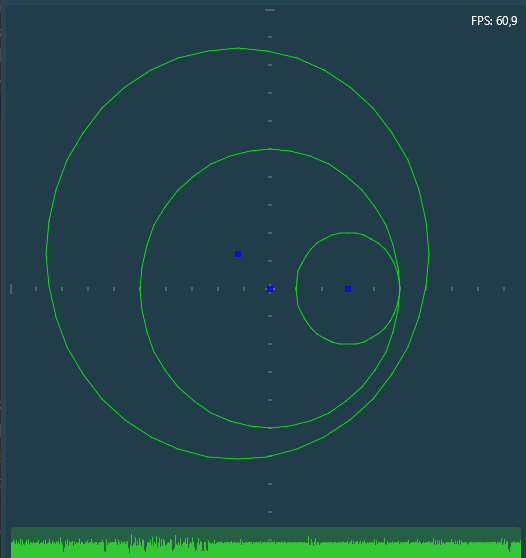
У точки было добавлено поле **pos**, соответствующее положению точки в пространстве задачи.

# 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(…)**



# 5. 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);  
 }  
}