

# Лабораторная работа №4

## Цель работы

Закрепить теоретический материал и реализацию алгоритмов растеризации отрезков и кривых:

- пошаговый алгоритм
- алгоритм ЦДА
- алгоритм Брезенхема
- алгоритм Брезенхема для окружностей

## Задачи работы

- Реализовать класс оконного приложения для растеризации кривых и отрезков
- Реализовать метод для пошагового алгоритма
- Реализовать метод для алгоритма ЦДА
- Реализовать метод для алгоритма Брезенхема
- Реализовать метод для алгоритма Брезенхема для окружностей

## Использованные средства разработки

- Kotlin
- JavaxSwing

## Временные характеристики

Был взят отрезок:  $x_0=-100$ ,  $y_0=-50$ ,  $x_1=100$ ,  $y_1=50$

И окружность:  $x_0=0$ ,  $y_0=0$ ,  $r=100$

	Время работы
Пошаговый алгоритм	1552 мкс
Алгоритм ЦДА	2751 мкс
Алгоритм Брезенхема	2222 мкс
Алгоритм Брезенхема для окружностей	17829 мкс

## Вывод

На основании приведенной в таблице информации можно сделать следующие выводы:

1. Пошаговый алгоритм оказался самым быстрым среди представленных алгоритмов растеризации с временем работы 1552 мкс.
2. Алгоритм ЦДА (цифровой дифференциальный анализатор) занял последнее место по скорости выполнения, затратив 2751 мкс.
3. Алгоритм Брезенхема оказался вторым по скорости, потратив 2222 мкс на растеризацию.

Таким образом, пошаговый алгоритм оказался наиболее эффективным для растеризации отрезков с заданными характеристиками. Это может быть связано с относительной простотой алгоритма и оптимизацией вычислений.