

## Лабораторная работа №4

### Цель работы:

Закрепить теоретический материал и практически освоить основные возможности по использованию базовых алгоритмов растеризации отрезков и кривых:

- пошаговый алгоритмов
- алгоритм ЦДА
- алгоритм Брезенхема
- алгоритм Брезенхема(окружность)
- алгоритм Ву

### Задачи работы:

- Создать функцию для отображения растеризованного отрезка на экране
- Создать удобный и понятный пользовательский интерфейс
- Реализовать пошаговый алгоритм
- Реализовать алгоритм ЦДА
- Реализовать алгоритм Брезенхема
- Реализовать алгоритм Брезенхема для окружности
- Реализовать алгоритм Ву для сглаженных линий

### Использованные средства разработки:

Python и Tkinter

### Ход работы:

1. Создание функций `draw_shape` и `draw_grid` для отображения растеризованного отрезка на экране с поддержкой координатной сетки и изменения масштаба.

2. Проектировка и создание удобного пользовательского интерфейса с возможностью выбора алгоритма, введением координат исходного отрезка, очисткой экрана от всех фигур.

3. Реализация пошагового алгоритма в виде функций `step_by_step_line`, `step_by_step_line_x_based` и `step_by_step_line_y_based`

4. Реализация алгоритма ЦДА в виде метода `dda_line`
5. Реализация алгоритма Брезенхема в виде метода `bresenham_line`
6. Реализация алгоритма Брезенхема для окружности в виде метода `bresenham_circle`
7. Реализация алгоритма Ву для сглаженных линий в виде метода `wu_line`
8. Добавление поддержки измерения прошедшего времени для каждого метода

### **Временные характеристики:**

Были введены большие входные данные для отрезка (хоть на экране они не видны, алгоритмы всё равно высчитываются, а скорость их печати не учитывается):

$x_0 = -1000$     $y_0 = -1000$

$x_1 = 1500$     $y_1 = 1500$

Для окружности:

$x_0 = -1000$     $y_0 = -1000$     $R = \text{abs}(x_1 - x_0) = 2500$

Пошаговый алгоритм	210 мс
Алгоритм ЦДА	1.786 мс
Алгоритм Брезенхема	1.094 мс
Алгоритм Ву	0.005 мс
Алгоритм Брезенхема для окружности	4.7 мс

Пошаговый алгоритм работает медленнее всего, так как при его работе на каждой итерации идут операции умножения с вещественными числами, в то время как в других алгоритмах, не используются операции умножения. В ЦДА алгоритме идёт прибавление вещественных чисел и их округление. В алгоритме Брезенхема идут операции сложения и вычитания с целыми числами, что и обуславливает его быструю работу по сравнению с другими алгоритмами. В алгоритме Ву кроме операций  $+1$  и  $-1$  вообще не используются другие в цикле, он работает быстрее всех.

### **Вывод:**

В ходе выполнения данной работы я:

- создал приложение, позволяющее проводить растеризацию отрезков и кривых базовыми алгоритмами
- закрепил полученные лекционные знания по различным алгоритмам растеризации
- получил дополнительный опыт по проектировке приложений
- углубил знания библиотеки TKinter, а также языка python
- получил дополнительный опыт работы с системой контроля версий Git