

## 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
- алгоритм Кастла–Питвея

### Задачи работы:

- Создать класс для отображения растеризованного отрезка на экране
- Создать класс для отображения пояснительной информации по ходу алгоритма 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:

- Фреймворк Qt и язык C++

### Ход работы:

1. Создание класса `PlotArea` для отображения растеризованного отрезка на экране с поддержкой координатной сетки и изменения масштаба. Были реализованы основные методы `DrawGrid`, `DrawAxis`, `DrawTicks`, `DrawPixels`, `AddPixel`
2. Проектировка и создание удобного пользовательского интерфейса с возможностью выбора алгоритма и введением координат исходного отрезка
3. Реализация пошагового алгоритма в виде метода `paintSteps`

4. Реализация алгоритма ЦДА в виде метода `paintDDA`
5. Реализация алгоритма Брезенхема в виде метода `paintBrezzenham`
6. Реализация алгоритма Брезенхема для окружности в виде метода `paintCircleBrezzenham`
7. Реализация алгоритма Ву для сглаженных линий в виде метода `paintWu`
8. Реализация алгоритма Кастла–Питвея в виде метода `paintCastlePitvey`

#### 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 фреймворка Qt, а также языка C++
- получил дополнительный опыт работы с системой контроля версий Git