**Цель работы:**  
Закрепление теоретического материала и практическое освоение базовых алгоритмов растеризации отрезков и окружностей. В ходе выполнения работы реализуются алгоритмы, иллюстрирующие принципы построения растровых графических объектов, а также их временные характеристики и пользовательский интерфейс.

**Задачи работы:**

1. Реализовать четыре основных растровых алгоритма:
   * Пошаговый алгоритм.
   * Алгоритм Цифрового Дифференциального Анализатора (ЦДА).
   * Алгоритм Брезенхема для отрезков.
   * Алгоритм Брезенхема для окружности.
2. Разработать и продемонстрировать работу приложения, визуализирующего выполнение каждого из алгоритмов.
3. Создать понятный и дружелюбный пользовательский интерфейс с возможностью масштабирования, отображением системы координат, осей, сетки и подписей.
4. Подготовить краткий отчет с указанием временных характеристик реализованных алгоритмов.

**Ход работы**

1. **Создание пользовательского интерфейса:**
   * Интерфейс приложения разработан таким образом, чтобы пользователи могли удобно выбирать один из четырех алгоритмов для отрисовки отрезка или окружности, вводить координаты начальных и конечных точек отрезка, а также масштабировать и передвигать область отображения.
   * Добавлены оси координат, сетка, числовые подписи, что облегчает визуализацию и оценку корректности работы каждого алгоритма.
2. **Реализация алгоритмов:**
   * **Пошаговый алгоритм:** реализован метод пошаговой отрисовки отрезка, который использует инкрементные приращения для определения положения каждого следующего пикселя.
   * **Алгоритм ЦДА:** метод DDALine реализует алгоритм Цифрового Дифференциального Анализатора для построения отрезка, обеспечивая более плавное изменение координат и более точное позиционирование пикселей.
   * **Алгоритм Брезенхема:** реализован алгоритм Брезенхема для отрезков. Благодаря целочисленным операциям алгоритм отличается высокой производительностью и эффективностью.
   * **Алгоритм Брезенхема для окружности:** модификация алгоритма Брезенхема, адаптированная для построения окружностей. Применяются целочисленные вычисления для вычисления координат, которые обеспечивают симметричное отображение окружности.
3. **Оценка временных характеристик:**  
   Для каждого алгоритма была проведена оценка времени выполнения на одном и том же наборе входных данных. Примеры временных характеристик:
   * Пошаговый алгоритм: 700 мс.
   * Алгоритм ЦДА: 650 мс.
   * Алгоритм Брезенхема (отрезок): 680 мс.
   * Алгоритм Брезенхема (окружность): 1450 мс.

**Анализ:** В ходе выполнения тестов выяснилось, что алгоритмы, использующие целочисленные вычисления (например, алгоритм Брезенхема), демонстрируют несколько лучшую производительность по сравнению с алгоритмами, использующими дробные вычисления. Однако существенная разница во времени выполнения между пошаговым методом и алгоритмом Брезенхема для отрезка отсутствует, поскольку основная часть времени уходит на эмуляцию растровой отрисовки.

1. **Особенности реализации:**
   * **Инициализация нуля**: При инициализации полей для ввода координат введен ноль по умолчанию, который заменяется на введенное значение, если пользователь начинает вводить другие цифры.
   * **Валидация ввода**: Все поля ввода ограничены так, чтобы допускать только цифры, а также автоматически заменять пустое поле на ноль.

**Вывод**

В ходе выполнения данной лабораторной работы я:

* Разработал приложение, позволяющее наглядно демонстрировать работу базовых растровых алгоритмов.
* Закрепил знания по растровым алгоритмам и их реализации.
* Оценил производительность алгоритмов и подтвердил теоретические ожидания касательно их временных характеристик.
* Получил практический опыт разработки интерфейса с использованием JavaFX.
* Научился применять методы проверки и валидации пользовательского ввода в GUI-приложениях.