**Лабораторная работа 4**

**Документация**

Реализация лабораторной работы №4 с помощью фреймворка Qt и языка Python 3.11\_3.11.1776.0\_x64. .

Использованные библиотеки:

* decimal: Эта библиотека предоставляет поддержку для работы с десятичными числами с фиксированной точностью. В коде используются классы Decimal и функции ROUND\_HALF\_UP, ROUND\_FLOOR и ROUND\_CEILING для округления чисел.
* numpy (np): Библиотека для выполнения операций над многомерными массивами и матрицами. Она предоставляет высокопроизводительные структуры данных и функции для работы с числовыми данными. В коде используется для работы с числовыми массивами.
* math: Встроенная библиотека Python, предоставляющая функции для математических операций. В коде используется для выполнения общих математических вычислений.
* matplotlib: Библиотека для визуализации данных в виде графиков и диаграмм. В коде используется для создания и отображения графиков.
* matplotlib.pyplot (plt): Подмодуль библиотеки Matplotlib, предоставляющий функции для создания графиков и диаграмм. В коде используется для создания и настройки графиков.
* matplotlib.backends.backend\_qt5agg: Модуль Matplotlib, предоставляющий поддержку для интеграции библиотеки с фреймворком Qt. В коде используется для создания холста графика, который может быть встроен в приложение Qt.
* PySide6.QtCore: Библиотека для разработки приложений на языке Python с использованием фреймворка Qt. В коде используется для работы с ядром фреймворка Qt.
* PySide6.QtWidgets: Модуль PySide6, предоставляющий набор виджетов для создания графического интерфейса пользователя (GUI) в приложениях Qt. В коде используется для создания основного окна и виджетов.
* raster: Это мой файл с алгоритмами( пользовательская библиотека), которая содержит функции и классы для 4 алгоритмов растеризации.
* time: Встроенная библиотека Python для работы со временем. В коде используется для измерения времени выполнения определенных участков кода.

Основные компоненты приложения:

* Графический интерфейс пользователя (GUI): приложение создает графическое окно, в котором пользователь может взаимодействовать с программой.
* Combobox (список) методов растеризации отрезка
* Кнопка Draw для примения одного из методов растеризации для отрезка
* Масштабируемые оси координат, при нажатии любой кнопкой мыши и удержании, далее нужно провести мышкой в любом напрвлении, и мы получим отрезок.
* Кнопка Clear для удаления линий и других ненужных объектов с осей координат.
* Время выполнения: приложение отображает информацию о времени, затраченном на выполнение определенного метода.

Функционал:

* Масштабирование при помощи колёсика мыши.
* Выбор метода растеризации(пошаговый алгоритм; − алгоритм ЦДА; − алгоритм Брезенхема; − алгоритм Брезенхема (окружность)).
* Далее при помощи нажатия,удержания и затем отпускания кнопки мыши задаём отрезок, к которому хотим применить один из методов растеризации.
* При помощи кнопки Draw получаем на форме растр нашей линии определенным методом
* При помощи кнопки Clear мф очищаем нашу форму от всего того, что там было.

Общий функционал приложения заключается в применении методов растеризации, получения разложения в растр, замер времени(результат в консоли).Также для удобства в консоли есть точные координаты точек. Также все отрезки на форме будут разных цветов, для того, чтобы при добавлении на форму не только одного отрезка, а нескольких, их было лучше видно. В среднем время растеризации отрезка длиной 20, занимает не более 0.1 секунды.

**Пример:**

Для примера с отрезком координат (0,0) и (10,15), предположим, что цвет линии задан как "красный" (red).

Вычисления:

Заданные координаты начальной точки и конечной точки: start = [0, 0], end = [10, 15].

Поскольку start[0] (0) не больше end[0] (10), условие start[0] > end[0] не выполняется, и координаты остаются без изменений.

Поскольку start[1] (0) не больше end[1] (15), условие start[1] > end[1] также не выполняется, и координаты остаются без изменений.

Затем происходит проверка на горизонтальную линию. Поскольку start[1] (0) не равно end[1] (15), условие start[1] == end[1] не выполняется.

В цикле for происходит итерация от start[0] (0) до end[0] (10) с шагом 0.33. Значение x принимает следующие значения: 0.0, 0.33, 0.66, 0.99, 1.32, 1.65, 1.98, 2.31, 2.64, 2.97, 3.3, 3.63, 3.96, 4.29, 4.62, 4.95, 5.28, 5.61, 5.94, 6.27, 6.6, 6.93, 7.26, 7.59, 7.92, 8.25, 8.58, 8.91, 9.24, 9.57, 9.9, 10.23.

Для каждого значения x вычисляется соответствующее значение y на основе уравнения прямой: y = start\_y + (dy / dx) \* (x - start\_x). В данном случае dy = end[1] - start[1] = 15 - 0 = 15, dx = end[0] - start[0] = 10 - 0 = 10. Таким образом, уравнение прямой принимает вид y = 0 + (15 / 10) \* (x - 0) = 1.5x.  
Вычисленные значения y соответствующие каждому значению x: 0.0, 0.495, 0.99, 1.485, 1.98, 2.475, 2.97, 3.465, 3.96, 4.455, 4.95, 5.445, 5.94, 6.435, 6.93, 7.425, 7.92, 8.415, 8.91, 9.405, 9.9, 10.395, 10.89, 11.385, 11.88, 12.375, 12.87, 13.365, 13.86, 14.355, 14.85, 15.345.

Для каждого значения x и соответствующего значения y добавляется пиксель в список filled\_pixels в формате (x, y, color). В данном случае, список filled\_pixels будет содержать следующие значения:

(0, 0, "red")

(1, 1, "red")

(1, 2, "red")

(2, 3, "red")

(3, 4, "red")

(3, 5, "red")

(3, 5, "red")

(4, 6, "red")

(5, 7, "red")

(5, 8, "red")

(7,10, "red”)

(7,11, "red")

(8,12, "red")

(9,13, "red")

(9,14, "red")

(10,15, "red”)