

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И
РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления
Кафедра интеллектуальных информационных технологий

ОТЧЁТ
по лабораторной работе №1
по дисциплине

Графический интерфейс интеллектуальных систем
“Алгоритмы построения отрезков”

Выполнил

Немкова Е.А. гр. 221701

Проверил

Жмырко А. В.

Минск 2025

Цель: разработать элементарный графический редактор, реализующий построение отрезков с помощью алгоритма ЦДА, целочисленного алгоритма Брезенхема и алгоритма Ву.

Дополнительно: вызов способа генерации отрезка задается из пункта меню и доступно через панель инструментов “Отрезки”. В редакторе кроме режима генерации отрезков в пользовательском окне должен быть предусмотрен отладочный режим, где отображается пошаговое решение на дискретной сетке.

Средства разработки:

1. Язык программирования - Python 3.x;
2. Реализация пользовательского интерфейса - Tkinter.

Теоретические сведения:

1. Алгоритм ЦДА.

С помощью алгоритма ЦДА решается дифференциальное уравнение отрезка, имеющее вид:

$$dy/dx = \Delta y / \Delta x, \text{ где } \Delta y = y_2 - y_1, \Delta x = x_2 - x_1,$$

При этом ЦДА формирует дискретную аппроксимацию непрерывного решения этого дифференциального уравнения. В симметричном ЦДА тем или иным образом определяется количество узлов N , используемых для аппроксимации отрезка. Затем за N циклов вычисляются координаты очередных узлов:

$$X_{i+1} = X_i + P_x / N,$$

$$Y_{i+1} = Y_i + P_y / N.$$

Получаемые значения x и y преобразуются в целочисленные значения координаты очередного подсвечиваемого пикселя. Генератор векторов, использующий этот алгоритм, имеет тот недостаток, что точки могут прописываться неоднократно, что увеличивает время построения, или отрезок будет выглядеть прерывистым.

2. Метод Брезенхема.

Брезенхем предложил алгоритм, построенный так, что требуется проверять лишь знак *ошибки* - изменение другой координаты (либо на ноль, либо на единицу) зависит от расстояния между действительным положением отрезка и ближайшими координатами

сетки. На рис. 1.1 а,б) это иллюстрируется для отрезка в первом октанте, т.е. для отрезка с угловым коэффициентом, лежащим в диапазоне от нуля до единицы. Из рисунка можно заметить, что если угловой коэффициент отрезка из точки (0,0) больше чем $1/2$, то его пересечение с прямой $x=1$ будет расположено ближе к прямой $y=1$, чем к прямой $y=0$. Следовательно, точка раstra (1,1) лучше аппроксимирует ход отрезка, чем точка (1,0). Если угловой коэффициент меньше $1/2$, то верно обратное. Для углового коэффициента, равного $1/2$ нет какого-либо предпочтительного выбора. В данном случае алгоритм выбирает точку (1,1). Не все отрезки проходят через точки раstra (рис. 1.3 в). Здесь отрезок с тангенсом угла наклона $3/8$ сначала проходит через точку раstra (0,0) и последовательно пересекает три пиксела. Так как желательно проверять только знак ошибки, то она первоначально устанавливается равной $-1/2$.

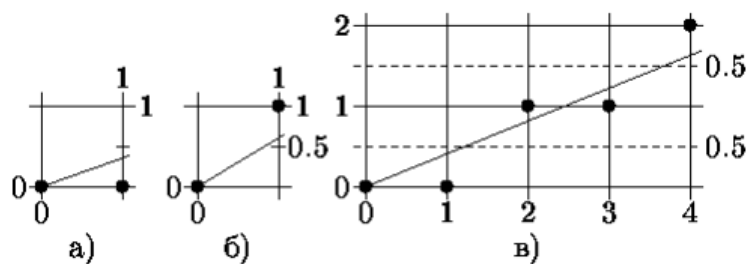


рис 1.1 - метод Брезенхема генерации отрезков

Возможны случаи

$E_1 > 0$	$E_1 \leq 0$
ближайшая точка есть:	
$X_1 = X_0 + 1; Y_1 = Y_0 + 1$	$X_1 = X_0 + 1; Y_1 = Y_0$
$E_2 = E_1 + P_y/P_x - 1$	$E_2 = E_1 + P_y/P_x$

ИЛИ

	$E_1 = 2 * P_y - P_x$
$E_1 > 0$	$E_2 = E_1 + 2 (P_y - P_x)$
$E_1 \leq 0$	$E_2 = E_1 + 2 * P_y$

3. Метод Ву.

Горизонтальные и вертикальные линии не требуют никакого сглаживания, поэтому их рисование выполняется отдельно. Для остальных линий алгоритм Ву проходит их вдоль основной оси, подбирая координаты по не основной оси аналогично алгоритму Брезенхема. Отличие состоит в том, что в алгоритме Ву на каждом шаге устанавливается не одна, а две точки. Например, если основной осью является Ох, то рассматриваются точки с координатами (х,у) и (х, у+1).

В зависимости от величины ошибки, которая показывает, как далеко ушли пиксели от идеальной линии по неосновной оси, распределяется интенсивность между этими двумя точками. Чем больше удалена точка от идеальной линии, тем меньше её интенсивность. Значения интенсивности двух пикселей всегда дают в сумме единицу, то есть это интенсивность одного пикселя, в точности попавшего на идеальную линию. Такое распределение придаст линии одинаковую интенсивность на всём её протяжении, создавая при этом иллюзию, что точки расположены вдоль линии не по две, а по одной.

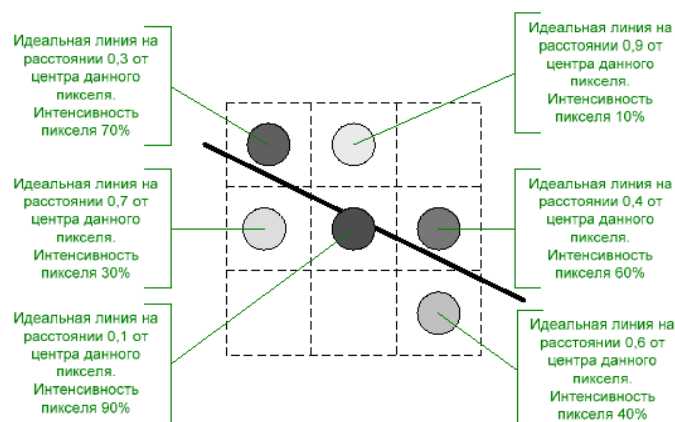


рис. 1.2 - распределение интенсивности пикселя в зависимости от расстояния

Демонстрация работы программы:

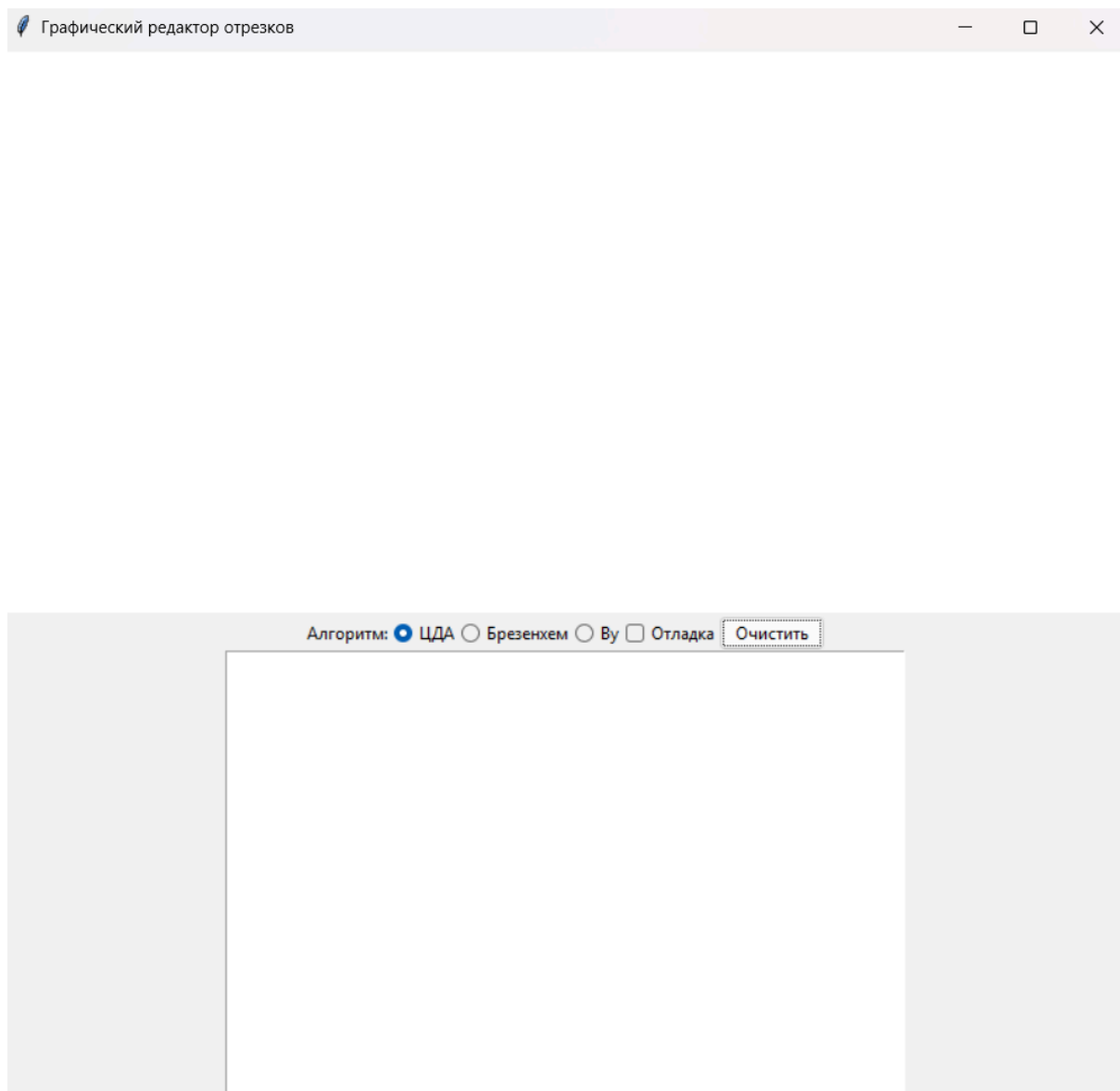


рис. 2.1 - главное окно программы

На рис. 2.1 представлено главное окно программы. Основное поле для рисования имеет белый фон панель управления и панель инструментов, которая позволяет пользователю рисовать отрезки, кликая и перетаскивая мышью. Внизу расположены кнопки, позволяющие выбрать метод отрисовки: ЦДА, Брезенхем, Бу. Также на панели инструментов есть кнопка “Отладка”, которая переводит пользователя в режим отладки. Рядом с отладкой расположена кнопка “Очистить” для очистки поля для рисования.

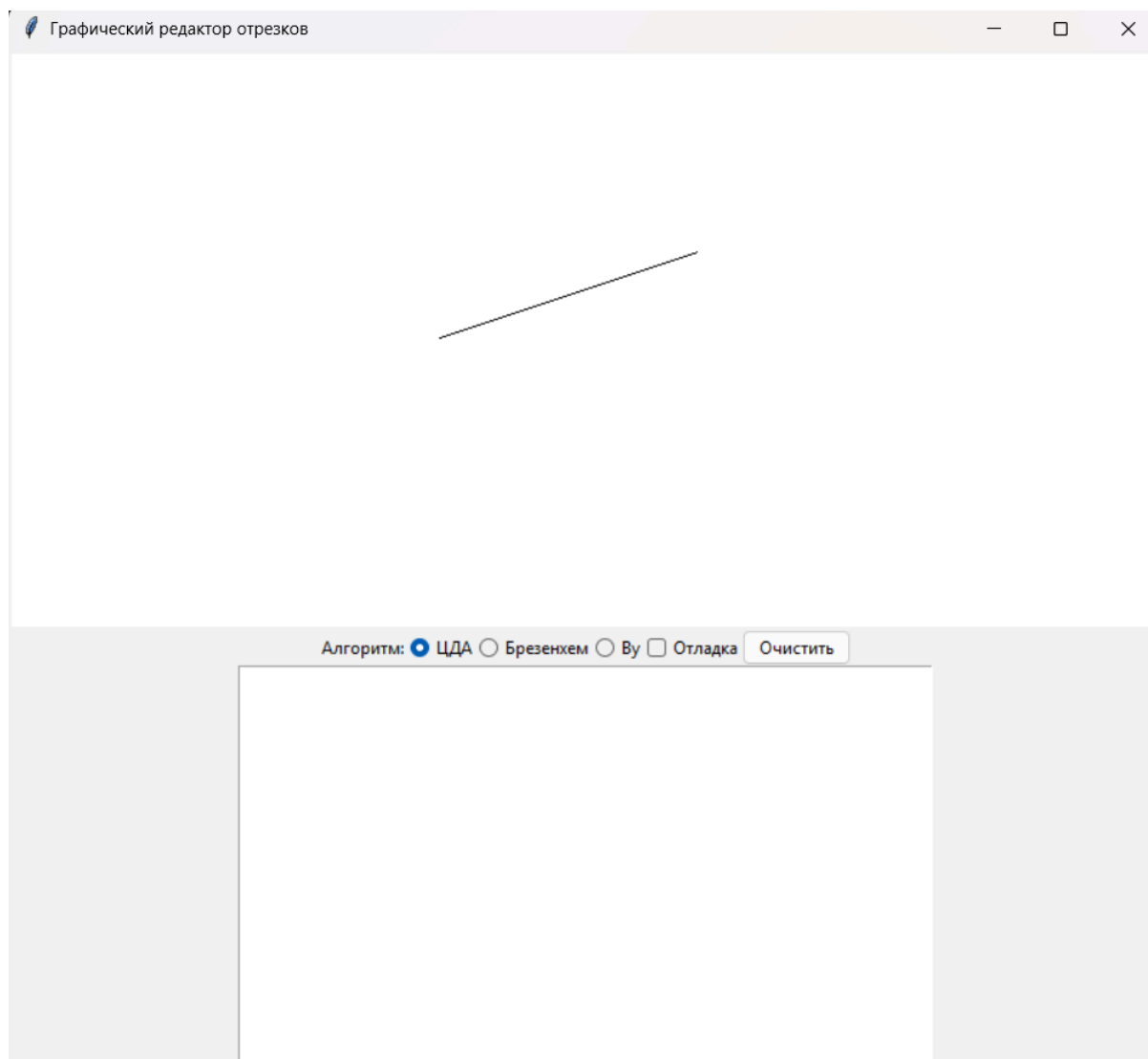


рис. 2.2 - построение отрезка с использованием алгоритма ЦДА в обычном режиме.

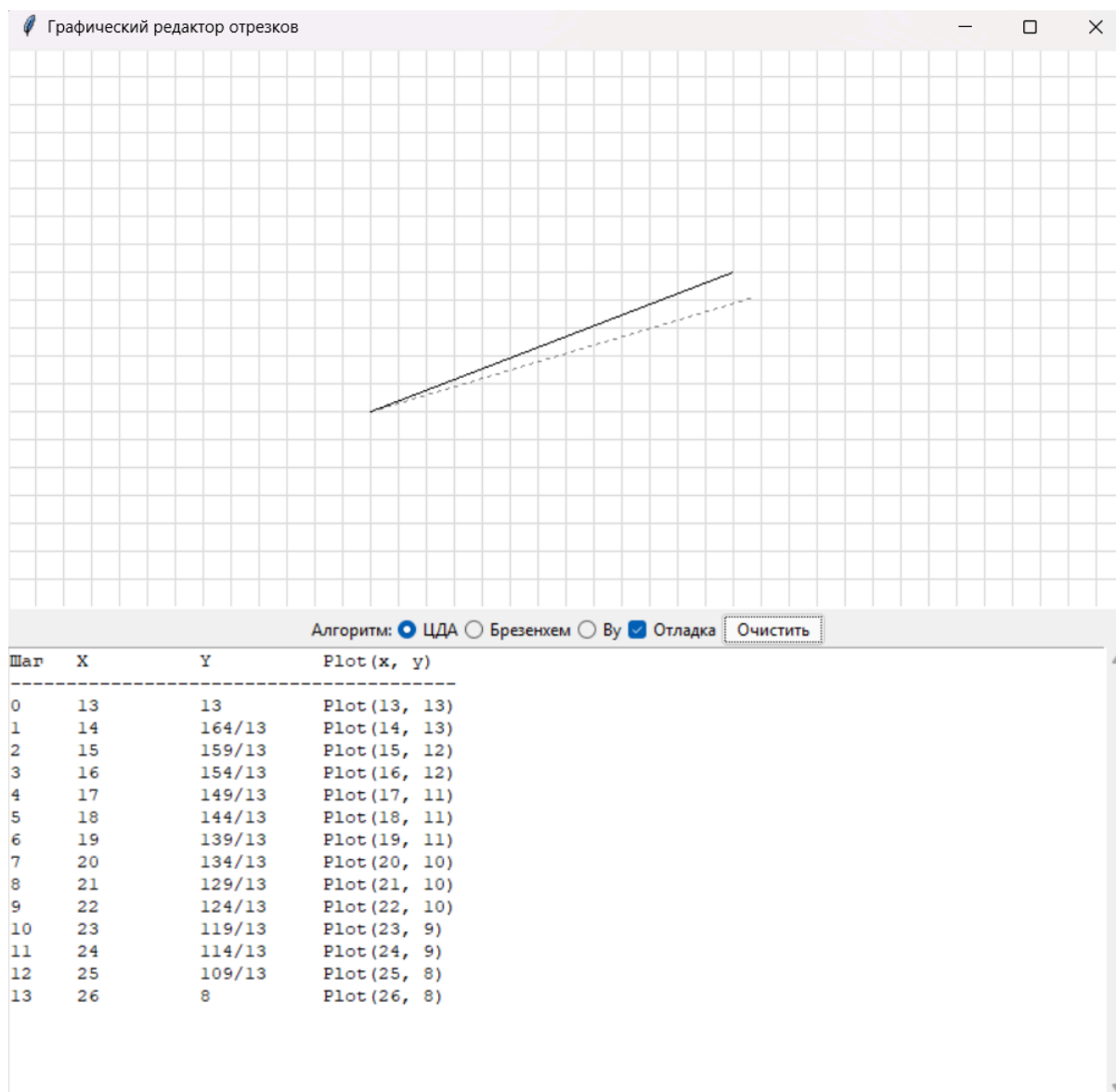


рис. 2.3 - построение отрезка с использованием алгоритма ЦДА в режиме отладки с таблицей построения отрезка.

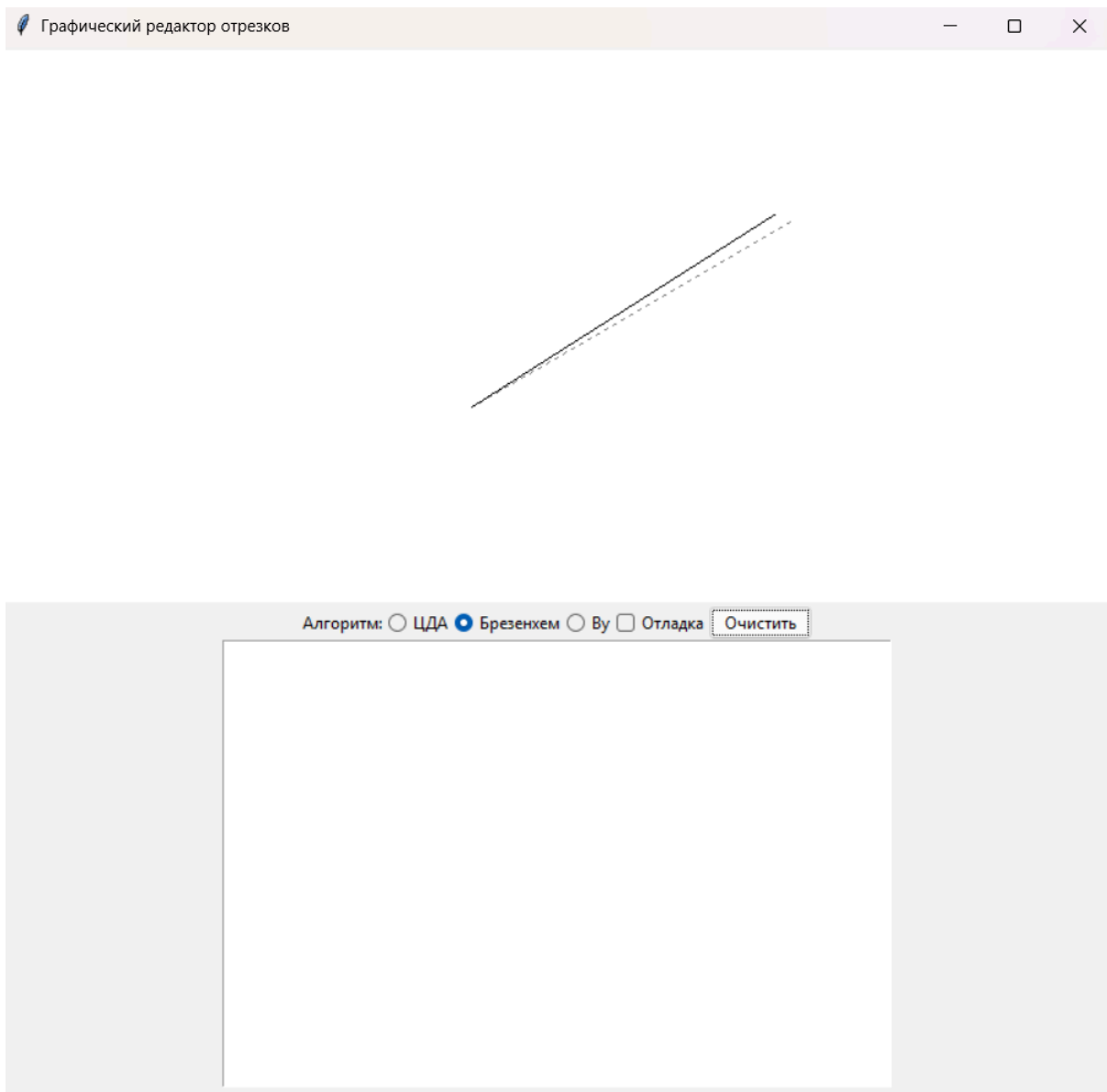


рис. 2.4 - построение отрезка с использованием алгоритма Брезенхема в обычном режиме.

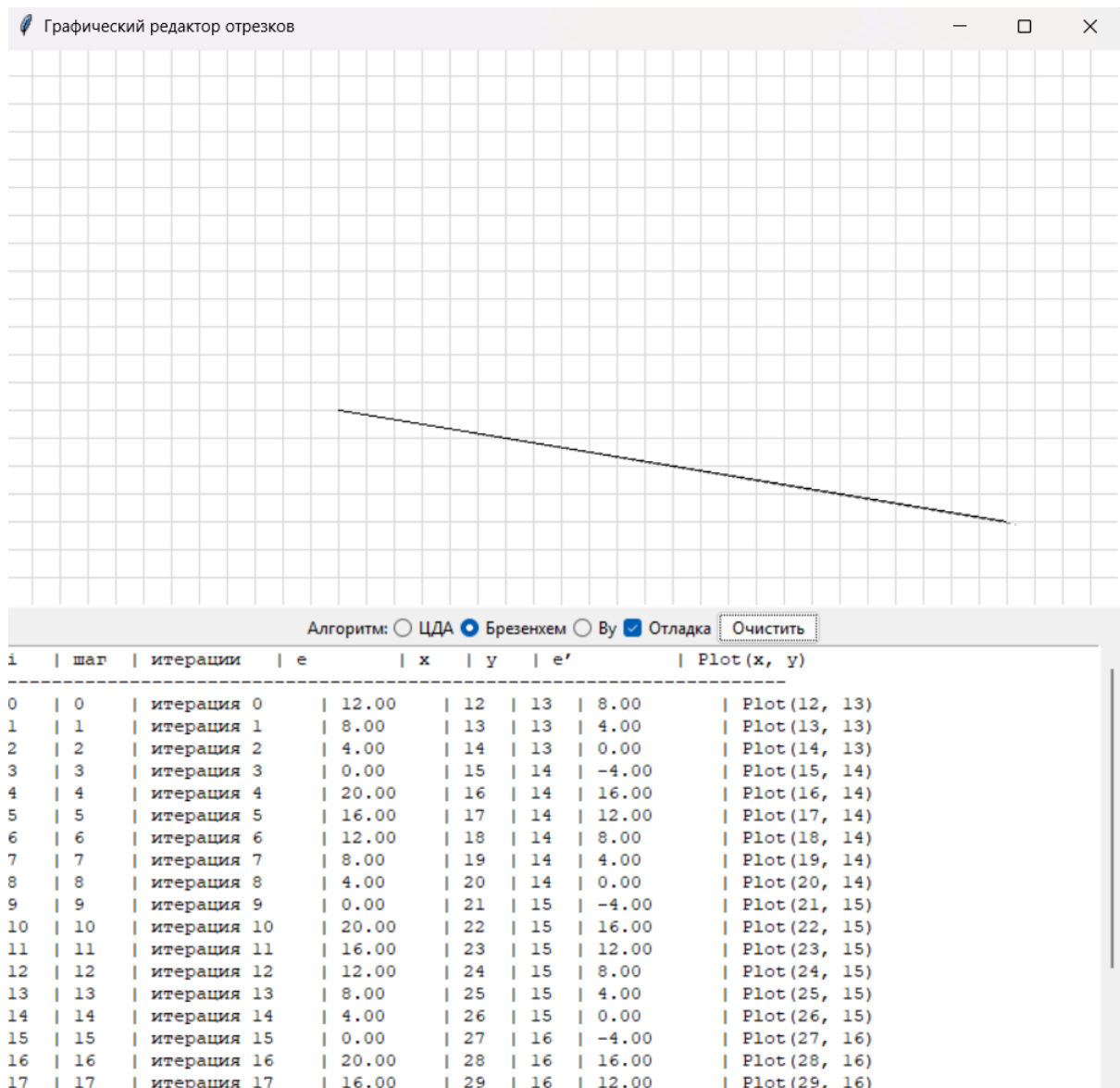


рис. 2.5 - построение отрезка с использованием алгоритма Брезенхема в режиме отладки с таблицей построения отрезков.

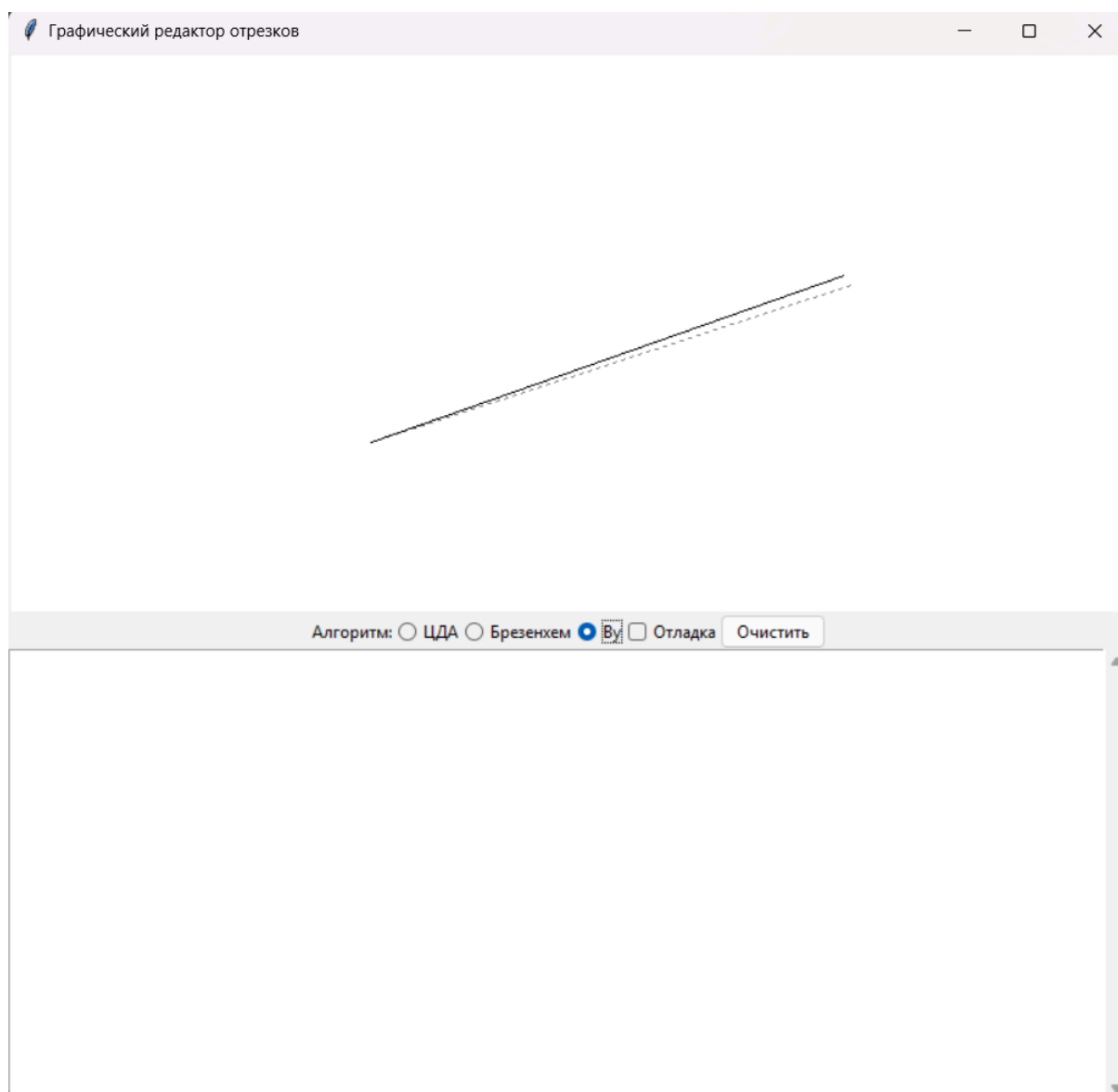


рис. 2.6. - таблица построение отрезка с использованием алгоритма Бу.

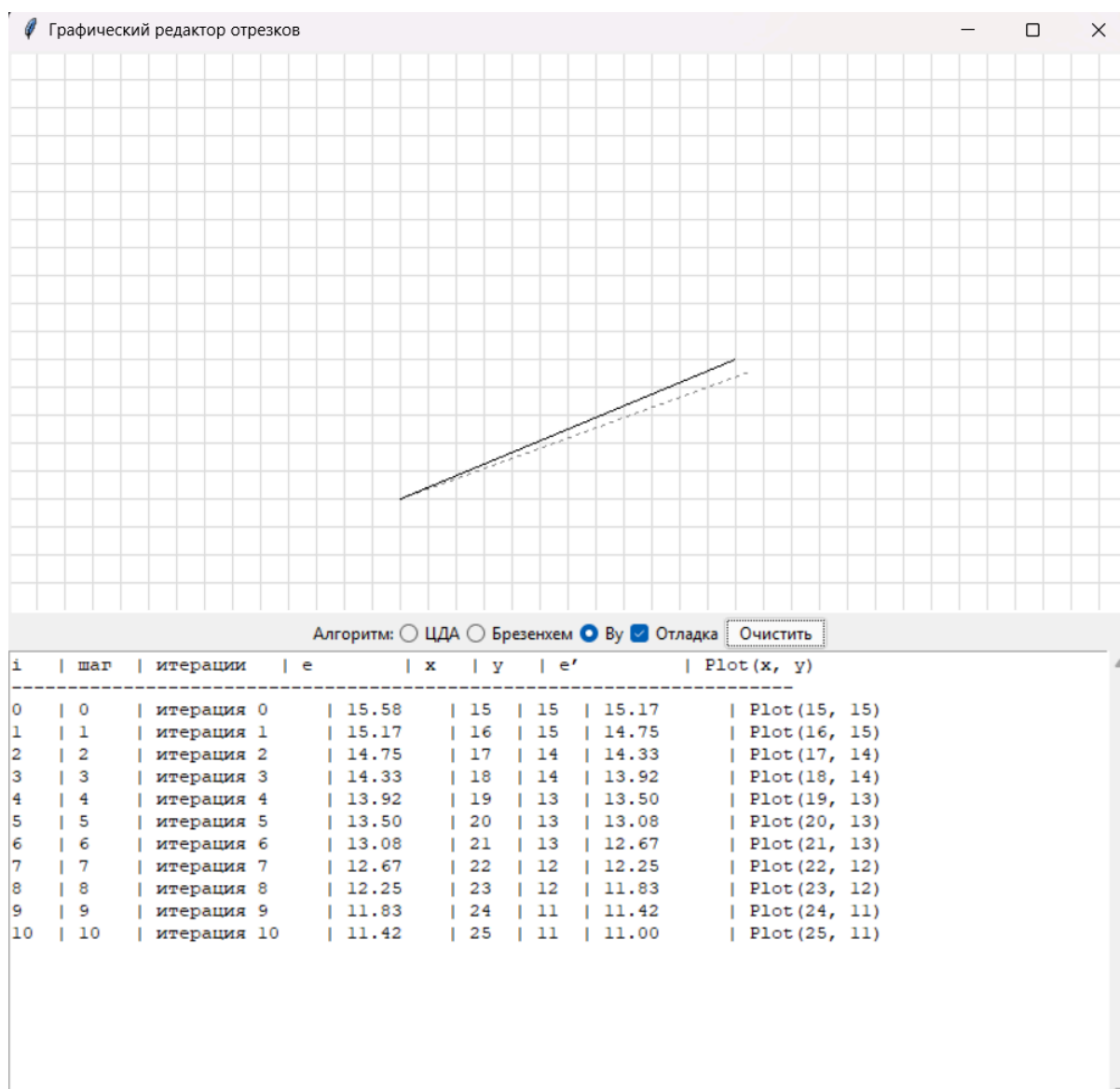


рис. 2.7 - построение отрезка с использованием алгоритма Vu в режиме отладки с таблицей шагов построения.

Вывод:

В ходе выполнения лабораторной работы были изучены три ключевых алгоритма построения отрезков на экране: Цифровой дифференциальный анализатор (ЦДА), алгоритм Брезенхема и метод Vu. Каждый из них обладает своими характерными чертами, достоинствами и ограничениями, что определяет область их применения в различных задачах компьютерной графики.

1. Цифровой дифференциальный анализатор (ЦДА):

Достоинства:

- прост в реализации;
- способен работать с дробными координатами, что обеспечивает более плавное построение линий;
- подходит для систем с высокой пропускной способностью.

Недостатки:

- высокая нагрузка на вычислительные ресурсы;
- погрешности из-за округления значений.

2. Алгоритм Брезенхема:

Достоинства:

- высокая скорость работы и вычислительная эффективность;
- стабильное качество линий при минимальных ошибках округления.

Недостатки:

- ограничен при работе с кривыми и сложными фигурами без доработки;
- требует доработок для корректного построения окружностей и кривых.

3. Метод Ву:

Достоинства:

- позволяет формировать сглаженные, визуально приятные линии;
- обеспечивает высокое качество изображения.

Недостатки:

- требует больше вычислительных ресурсов, чем метод Брезенхема;
- повышенная сложность реализации и обработки.

Заключение:

Выбор конкретного алгоритма построения линий зависит от

поставленных задач и требований к качеству графики, скорости выполнения и доступным ресурсам системы.

Метод Брезенхема зачастую выступает оптимальным выбором при разработке программ, где приоритетом является быстроедействие и эффективность.

Метод Ву оправдан в случаях, когда особенно важно добиться сглаженного и качественного визуального отображения линий.

ЦДА, в свою очередь, остается полезным инструментом для специфических задач, требующих работы с дробными значениями и высокой скорости при наличии достаточных вычислительных мощностей.