

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 №2
по дисциплине

Графический интерфейс интеллектуальных систем
“Алгоритмы построения линий второго порядка”

Выполнил

Немкова Е.А. гр. 221701

Проверил

Жмырко А. В.

Минск 2025

Цель: разработать элементарный графический редактор, реализующий построение линий второго порядка: окружность, эллипс, гипербола, парабола.

Дополнительно: Выбор кривой задается из пункта меню и доступен через панель инструментов “Линии второго порядка”. В редакторе кроме режима генерации линий второго порядка 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 Брезенхема для генерации окружности.

Алгоритм генерации окружности заключается в нахождении такого множества пикселей, которое наилучшим образом аппроксимирует кривую. Главное требование, предъявляемое к алгоритму, - эффективность.

Уравнение окружности задается по формуле: $x^2 + y^2 = R^2$

При генерации окружности достаточно иметь алгоритм, генерирующий одну восьмую окружности, остальные фрагменты могут быть получены последовательными отражениями.

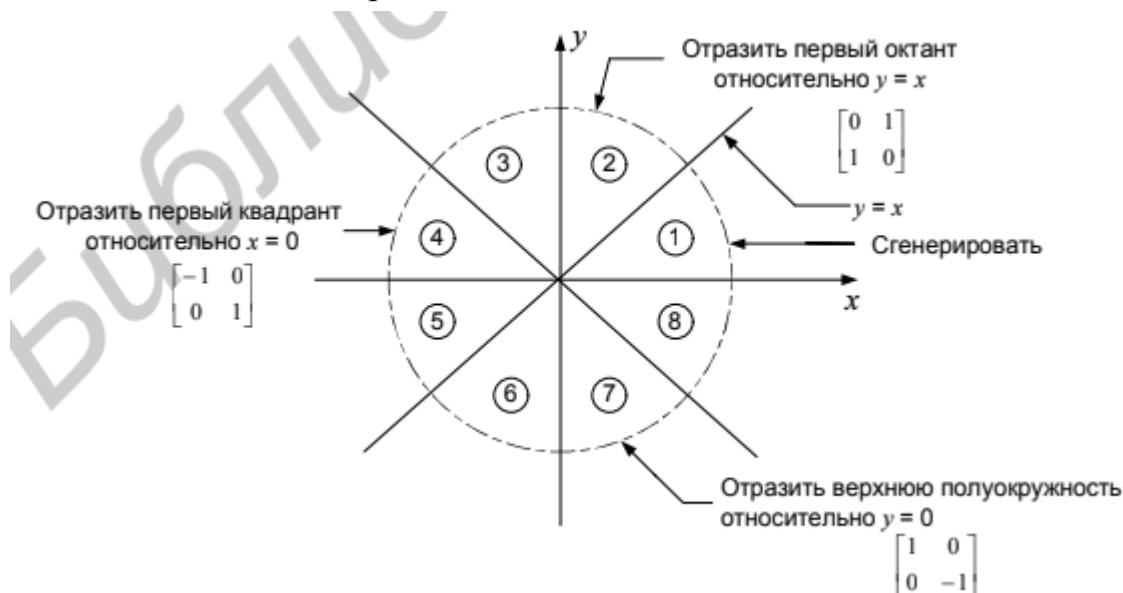


Рис. 1.1 - генерация полной окружности из дуги в первом октанте

Алгоритм Брезенхема начинает работать с точки $(0, R)$ и рисует часть окружности по часовой стрелке, лежащую в первом квадранте. Предполагается, что центр окружности и начальная точка находятся точно в точках растра. Как и при рисовании линии, центральным понятием

является ошибка – разность между центром пикселя и действительным положением окружности.

Сущность метода состоит в следующем:

1. вычислить ошибки для пикселей претендентов;
2. выбрать среди них пиксель с минимальным абсолютным значением разности квадратов расстояний от центра окружности до пикселя-претендента и до окружности.

Имеется только 3 пикселя-претендента - горизонтальный, вертикальный и диагональный.

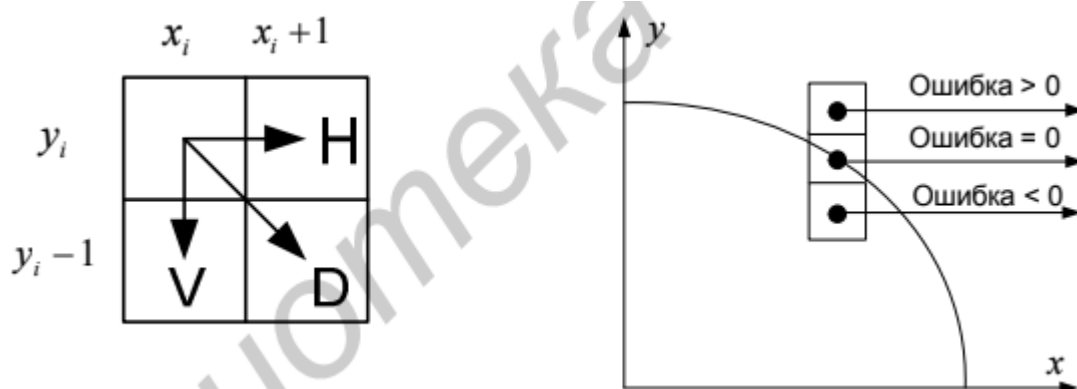


Рис 1.2 - пиксели-претенденты и значения ошибок

В общем случае положения точки относительно абсолютной величины можно задать в виде таблицы:

	δ		Пиксель	x_{i+1}	y_{i+1}	Δ_{i+1}
Случай А $\Delta_i < 0$	$\delta = 2(\Delta_D + y_i) - 1$	$\delta \leq 0$	H	$x_i + 1$	y_i	$\Delta_i + (2x_i + 1)$
		$\delta > 0$	D	$x_i + 1$	$y_i - 1$	$\Delta_i + 2(x_i - y_i + 1)$
Случай Б $\Delta_i > 0$	$\delta^* = 2(\Delta_D - x_i) - 1$	$\delta^* \leq 0$				
		$\delta^* > 0$	V	x_i	$y_i - 1$	$\Delta_i + (-2y_i + 1)$
Случай В $\Delta_i = 0$			D	$x_i + 1$	$y_i - 1$	$\Delta_i + 2(x_i - y_i + 1)$

Демонстрация работы программы:

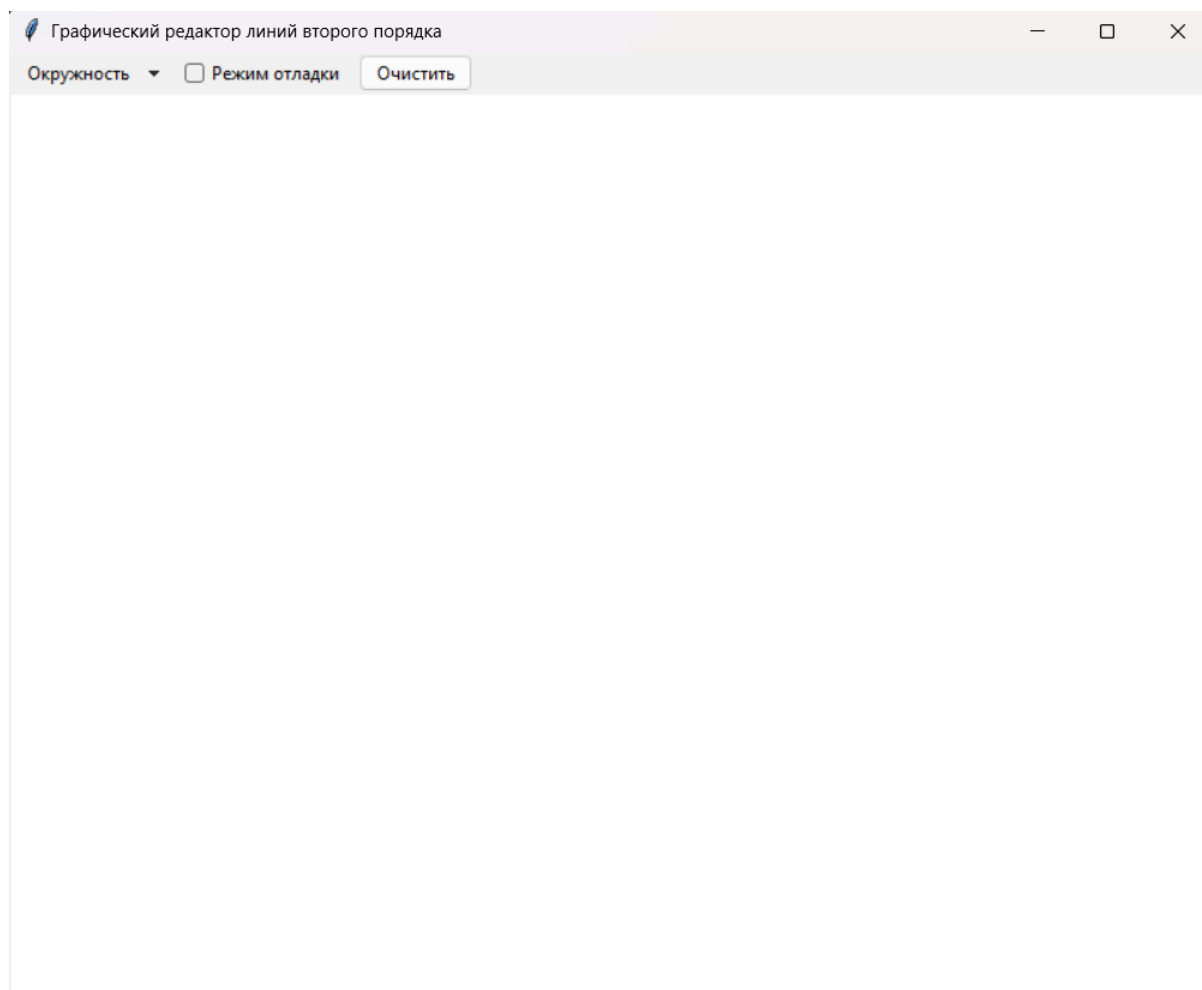


рис. 2.1 - главное окно программы.

На рис. 2.1 представлено обновленное главное окно программы. Основное поле для рисования имеет белый фон, на котором пользователь может строить кривые второго порядка (окружности, эллипсы, параболы и гиперболы). Панель инструментов дополнена новыми функциями, которые позволяют работать с графическими примитивами. Ниже приведено описание основных элементов интерфейса:

1. Панель инструментов:

- **Первая кнопка:** Открывает выпадающий список для выбора типа кривой (окружность, эллипс, гипербола, парабола). После выбора пользователь задаёт опорные точки на канве для построения выбранной кривой.

- **Кнопка “Отладка”:** Переключает пользователя в режим отладки, при котором после построения примитива открывается таблица с подробными вычислениями.

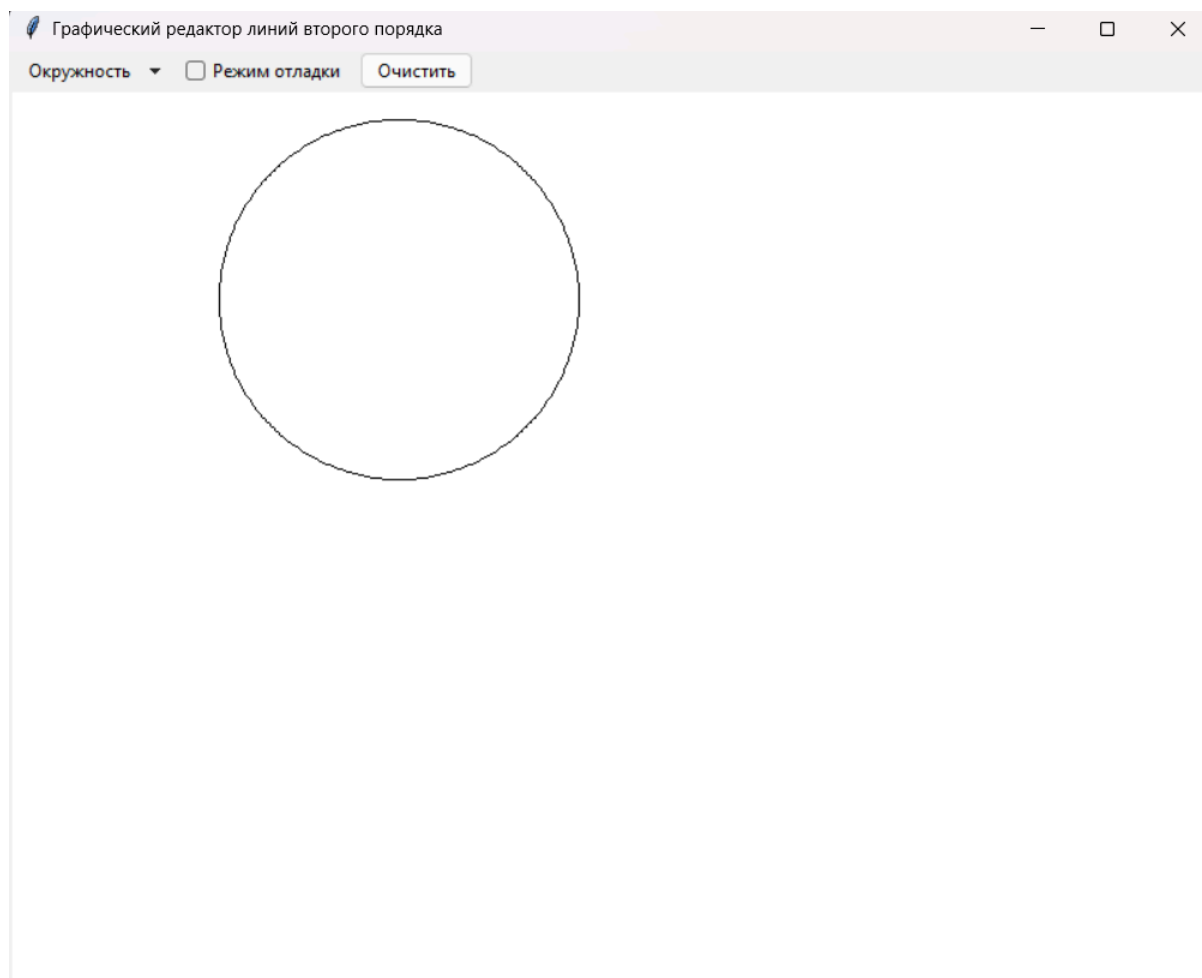


рис. 2.2 - построение окружности в обычном режиме.

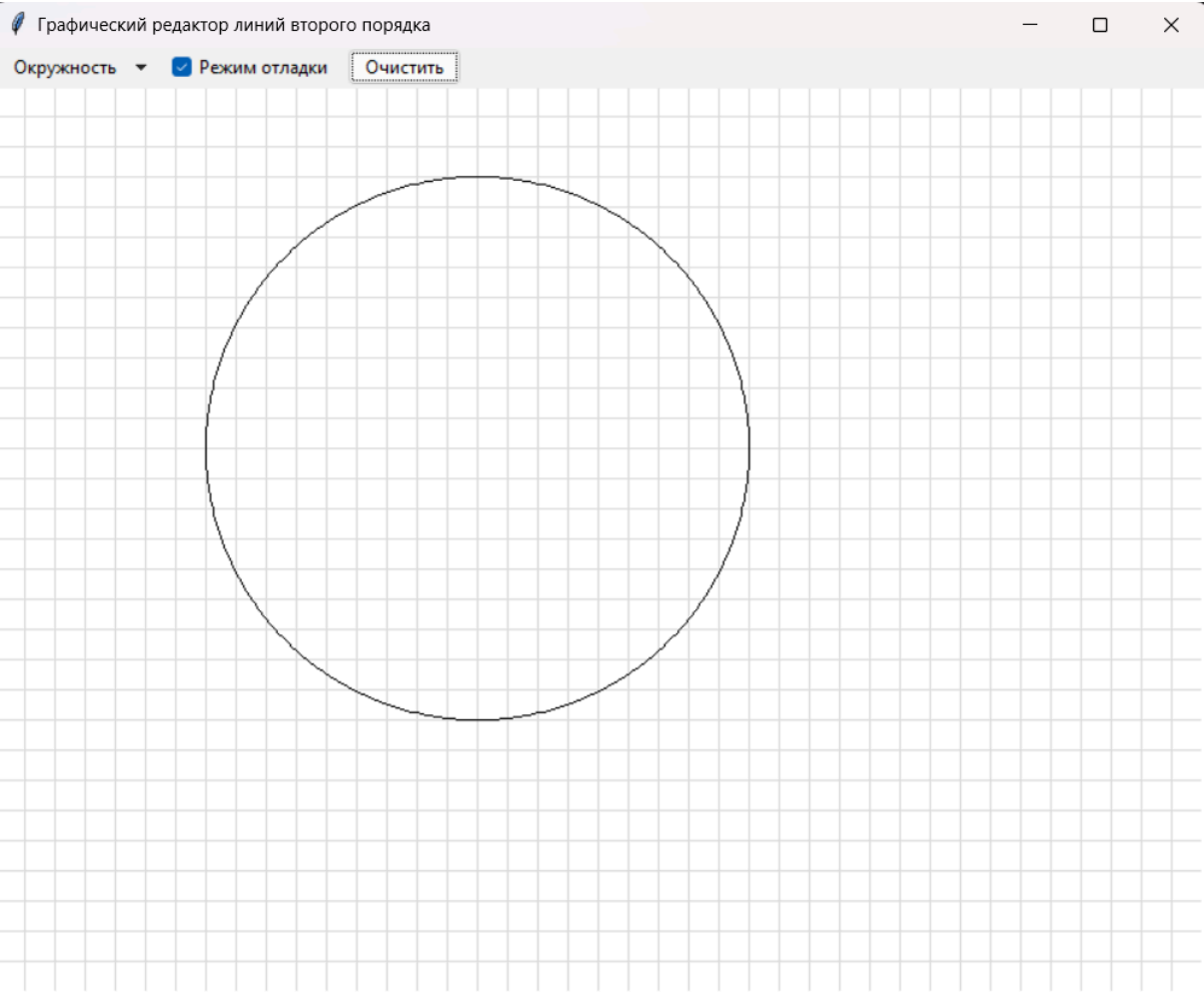


рис. 2.3. Построение окружности в режиме отладки (таблица шагов показана на рис. 2.4.)

Таблица отладки								
Окружность (алгоритм Брезенхема)								
Шаг	Δ_1	δ	x	y	Пиксель 1	Пиксель 2	Пиксель 3	Пиксель 4
1	-16	0	0	9	(16, 21)	(16, 21)	(16, 3)	(16, 3)
2	-13	-15	1	9	(17, 21)	(15, 21)	(17, 3)	(15, 3)
3	-8	-9	2	9	(18, 21)	(14, 21)	(18, 3)	(14, 3)
4	-20	-21	3	8	(19, 20)	(13, 20)	(19, 4)	(13, 4)
5	-11	-25	4	8	(20, 20)	(12, 20)	(20, 4)	(12, 4)
6	0	-7	5	8	(21, 20)	(11, 20)	(21, 4)	(11, 4)
7	-4	-11	6	7	(22, 19)	(10, 19)	(22, 5)	(10, 5)
8	-4	-21	7	6	(23, 18)	(9, 18)	(23, 6)	(9, 6)
9	0	-23	8	5	(24, 17)	(8, 17)	(24, 7)	(8, 7)
10	8	-17	9	4	(25, 16)	(7, 16)	(25, 8)	(7, 8)
11	20	-3	10	3	(26, 15)	(6, 15)	(26, 9)	(6, 9)
12	17	19	10	2	(26, 14)	(6, 14)	(26, 10)	(6, 10)
13	16	13	10	1	(26, 13)	(6, 13)	(26, 11)	(6, 11)
14	17	11	10	0	(26, 12)	(6, 12)	(26, 12)	(6, 12)

рис. 2.4. Таблица построения окружности.

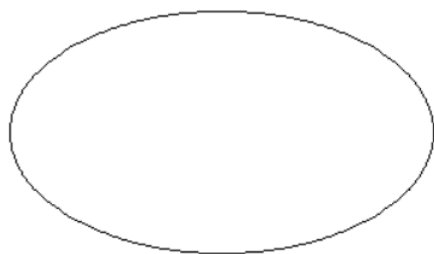


рис. 2.5 - построение эллипса в обычном режиме.

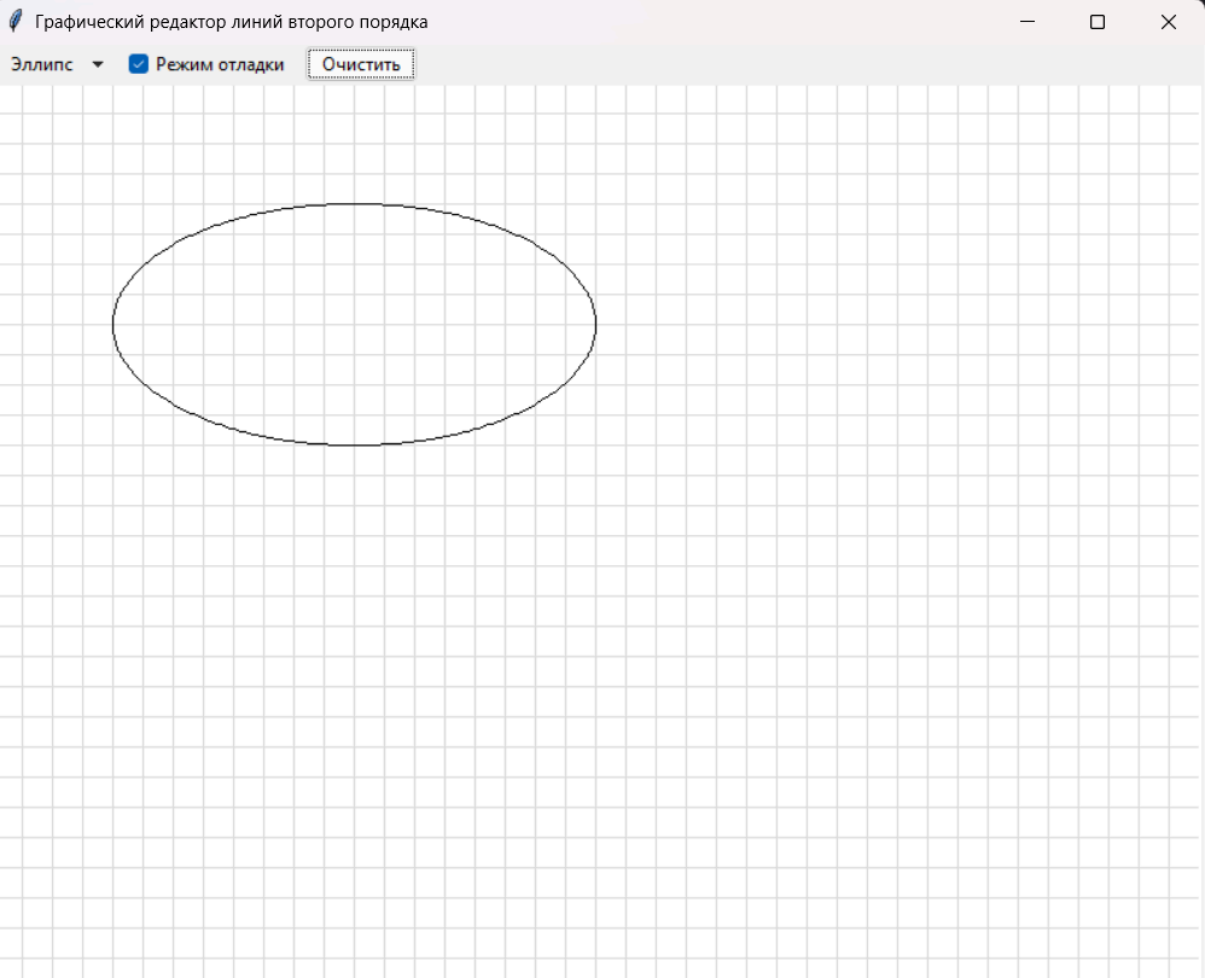


рис. 2.6. Построение эллипса в режиме отладки (таблица шагов показана на рис. 2.7.)

Таблица отладки							
Эллипс (алгоритм средней точки)							
Шаг	Регион	Δ	dx	dy	x	y	Пиксель
0	Region 1	-224.0	0	512	0	4	(12, 12)
1	Region 1	-176.0	32	512	1	4	(13, 12)
2	Region 1	-96.0	64	512	2	4	(14, 12)
3	Region 1	16.0	96	512	3	4	(15, 12)
4	Region 1	-224.0	128	384	4	3	(16, 11)
5	Region 1	-48.0	160	384	5	3	(17, 11)
6	Region 1	160.0	192	384	6	3	(18, 11)
7	Region 1	144.0	224	256	7	2	(19, 10)
8	Region 1	288.0	256	128	8	1	(20, 9)
8	Region 2	132.0	256	128	8	1	(20, 9)
9	Region 2	196.0	256	0	8	0	(20, 8)
10	Region 2	388.0	256	-128	8	-1	(20, 7)

рис. 2.7. Таблица построения эллипса.



рис. 2.8. - построение гиперболы в обычном режиме.

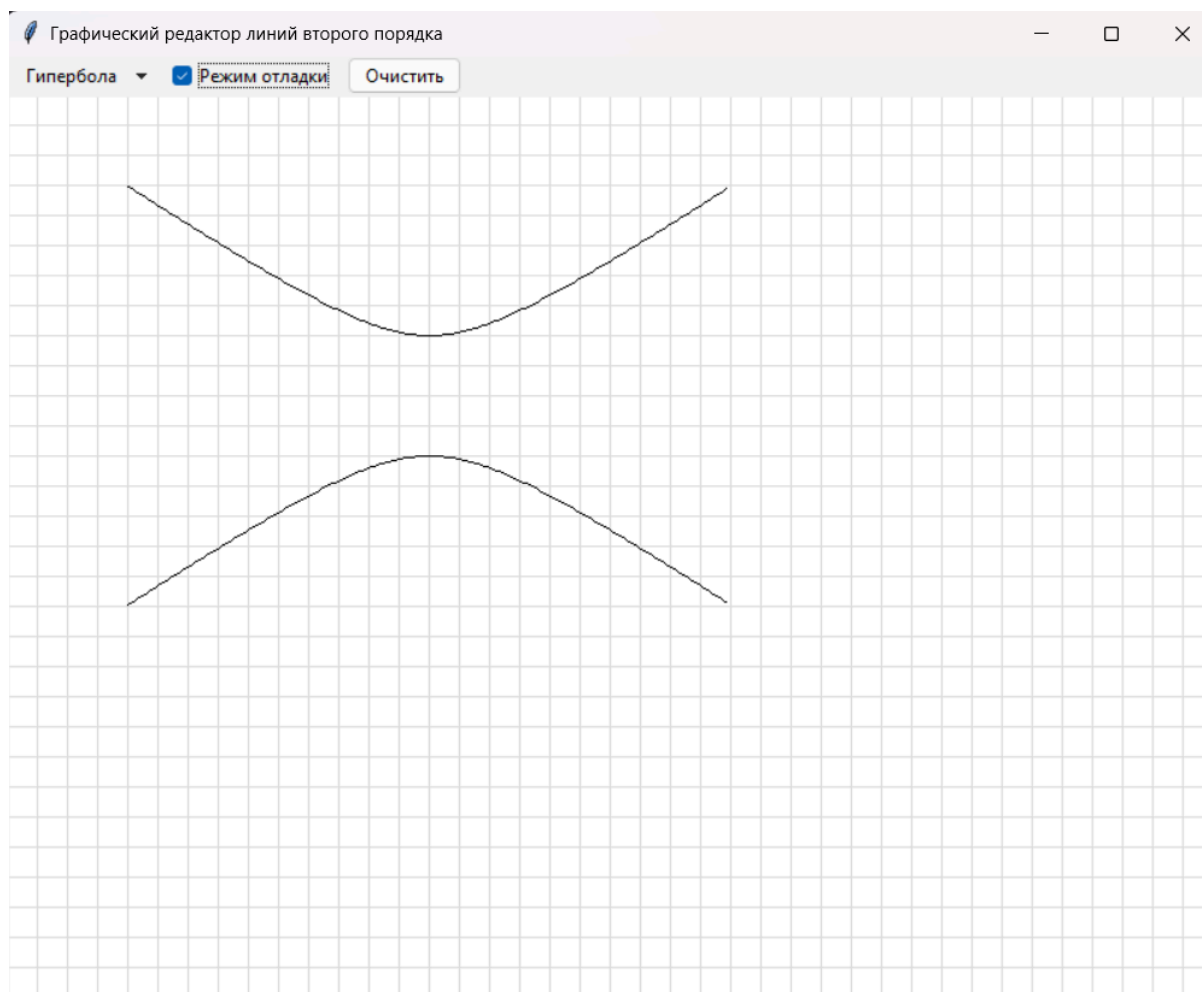


рис. 2.9. Построение гиперболы в режиме отладки (таблица шагов показана на рис. 2.10.)

Гипербола (алгоритм Брезенхема)

Шаг	x	y	Пиксель 1	Пиксель 2
0	3	0	(17, 10)	(17, 10)
1	4	1	(18, 11)	(18, 9)
2	5	2	(19, 12)	(19, 8)
3	6	3	(20, 13)	(20, 7)
4	7	4	(21, 14)	(21, 6)
5	8	5	(22, 15)	(22, 5)
6	9	5	(23, 15)	(23, 5)
7	10	6	(24, 16)	(24, 4)
8	11	7	(25, 17)	(25, 3)
9	12	8	(26, 18)	(26, 2)
10	13	8	(27, 18)	(27, 2)
11	14	9	(28, 19)	(28, 1)
12	15	10	(29, 20)	(29, 0)
13	16	10	(30, 20)	(30, 0)
14	17	11	(31, 21)	(31, -1)
15	18	12	(32, 22)	(32, -2)
16	19	12	(33, 22)	(33, -2)
17	20	13	(34, 23)	(34, -3)
18	21	14	(35, 24)	(35, -4)
19	22	14	(36, 24)	(36, -4)
20	23	15	(37, 25)	(37, -5)
21	24	16	(38, 26)	(38, -6)
22	25	16	(39, 26)	(39, -6)

рис. 2.10. Таблица построения гиперболы.

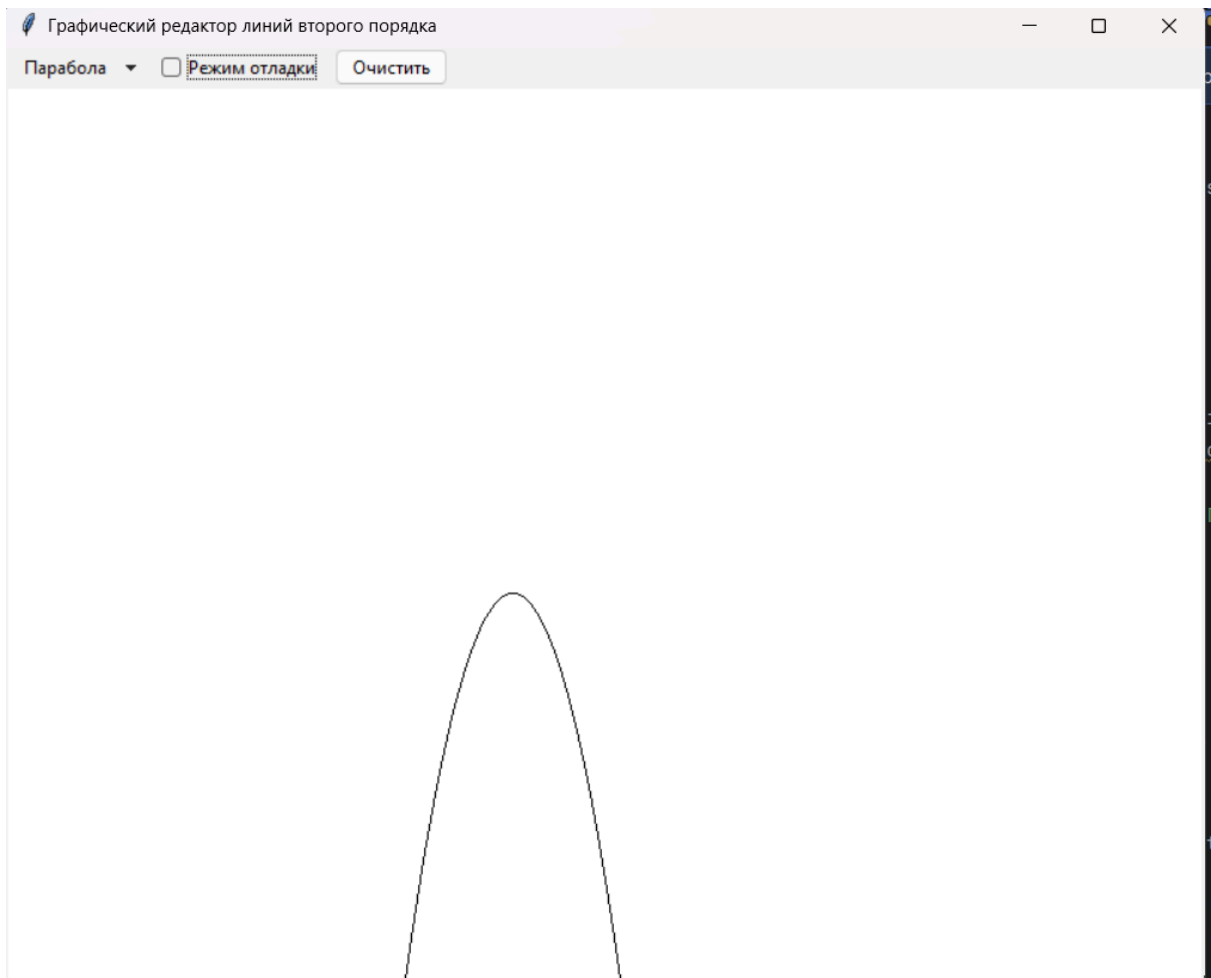


рис. 2.11. - построение параболы в обычном режиме.

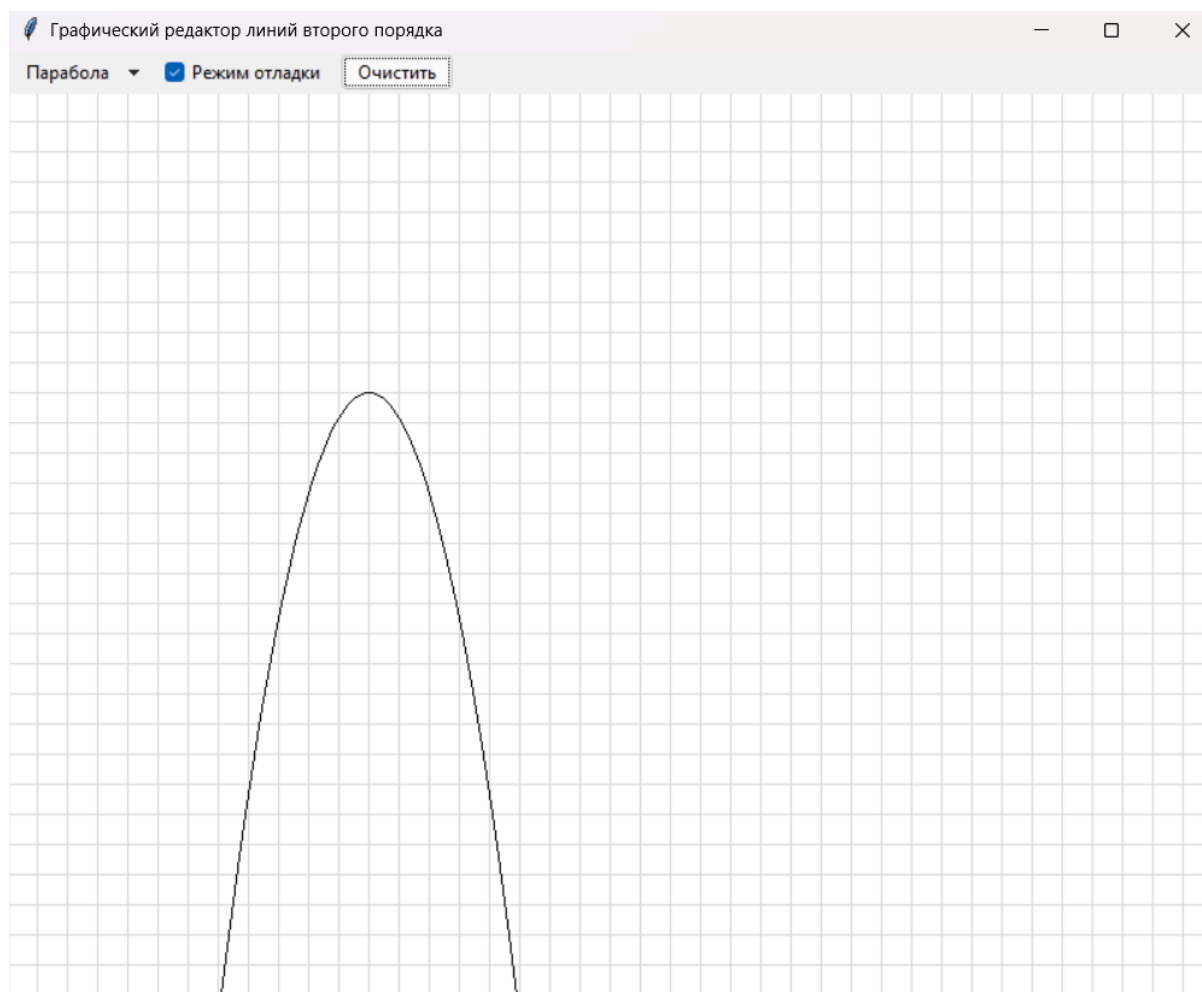


рис. 2.12. Построение параболы в режиме отладки (таблица шагов показана на рис. 2.13.)

Таблица отладки				
Парабола (алгоритм Брезенхема)				
Шаг	Δ	x	y	Пиксель
0	-11	0	0	(12, 10)
1	-8	1	0	(13, 10)
2	-3	2	0	(14, 10)
3	4	3	0	(15, 10)
4	1	4	1	(16, 11)
5	0	5	2	(17, 12)
6	1	6	3	(18, 13)
7	4	7	4	(19, 14)
8	9	8	5	(20, 15)
9	16	9	6	(21, 16)
10	25	10	7	(22, 17)
11	36	11	8	(23, 18)
12	49	12	9	(24, 19)
13	64	13	10	(25, 20)
14	81	14	11	(26, 21)
15	100	15	12	(27, 22)
16	121	16	13	(28, 23)
17	144	17	14	(29, 24)
18	169	18	15	(30, 25)
19	196	19	16	(31, 26)
20	225	20	17	(32, 27)
21	256	21	18	(33, 28)
22	289	22	19	(34, 29)

рис. 2.13. Таблица построения параболы

Вывод: В ходе лабораторной работы на тему "Алгоритм построения линий второго порядка" была реализована алгоритмическая схема построения окружностей по алгоритму Брезенхема. Этот метод обеспечивает эффективное и качественное рисование окружностей на растровых дисплеях, основываясь на целочисленной арифметике, что минимизирует вычислительные затраты и повышает скорость отрисовки.

Преимущества:

1. **Эффективность:** Алгоритм использует целочисленные вычисления, что позволяет значительно сократить время обработки по сравнению с плавающей точкой.
2. **Отсутствие арифметических ошибок:** Использование целых чисел минимизирует вероятность ошибок округления, что особенно важно при построении кривых.

3. **Простота реализации:** Алгоритм относительно прост для понимания и реализации, что делает его доступным для изучения и практического применения.
4. **Гладкость линий:** Благодаря выбору пикселей, алгоритм обеспечивает визуально приемлемое качество линий и кривых.

Недостатки:

1. **Ограниченная гибкость:** Алгоритм предназначен в первую очередь для построения окружностей и может потребовать адаптации для других кривых, что увеличивает время разработки.
2. **Невозможность работы с антиалипингом:** В базовой реализации алгоритм не учитывает сглаживание, что может привести к менее гладким линиям на экране.
3. **Сложность в обработке сложных фигур:** При построении сложных фигур, состоящих из нескольких окружностей или других кривых, алгоритм может стать сложным в реализации и менее эффективным.

В целом, алгоритм Брезенхема для построения окружностей демонстрирует эффективные характеристики, особенно в контексте растровых графиков, однако его использование требует учета специфики задач и возможных ограничений.