|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № \_\_4\_\_**

|  |  |
| --- | --- |
| **Тема РЕАЛИЗАЦИЯ И ИССЛЕДОВАНИЕ АЛГОРИТМОВ ГЕНЕРАЦИИ ОКРУЖНОСТИ И ЭЛЛИПСА**  **Студент \_Брянская Е.В.\_\_\_\_\_\_\_\_**  **Группа \_ИУ7-42Б\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель \_Куров А.В.** |  |

Москва.

2020 г.

**Цель работы:** реализация алгоритмов построения окружности и эллипса, исследование и сравнение визуальных и временных характеристик алгоритмов.

Задание:

1. Реализовать алгоритмы построения окружности/эллипса на основе

- Канонического уравнения

- Параметрического уравнения

- Алгоритма Брезенхема

- Алгоритма средней точки

- построение окружности с помощью библиотечной функции

1. Сравнить визуальные характеристики (также учесть сравнение спектров)
2. Сравнить временные характеристики разных алгоритмов на окружности и на эллипсе (построить графики зависимости времени работы алгоритма от радиуса (для окружности) или от изменения полуоси (для эллипса))

При построении окружности пользователь выбирает из списка определенный алгоритм, задает координаты центра, радиус, цвет рисования.

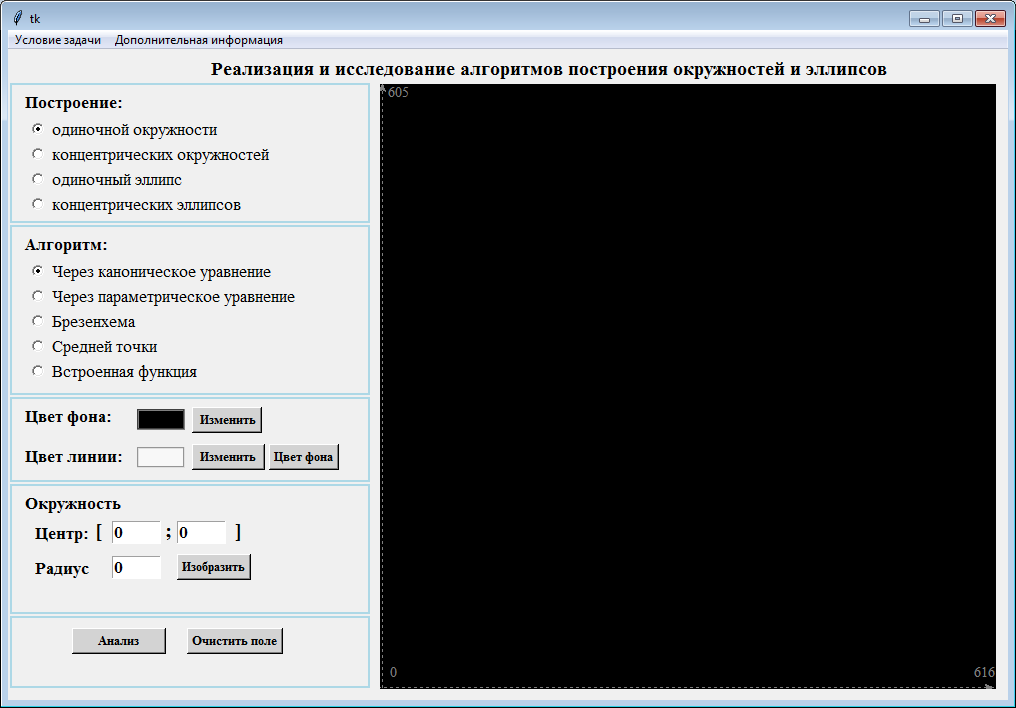
При построении спектра концентрических окружностей пользователь выбирает из списка определенный алгоритм, задает координаты центра, цвет рисования, три из следующих четырех параметров: начальный радиус, конечный радиус, шаг изменения радиуса, количество окружностей.

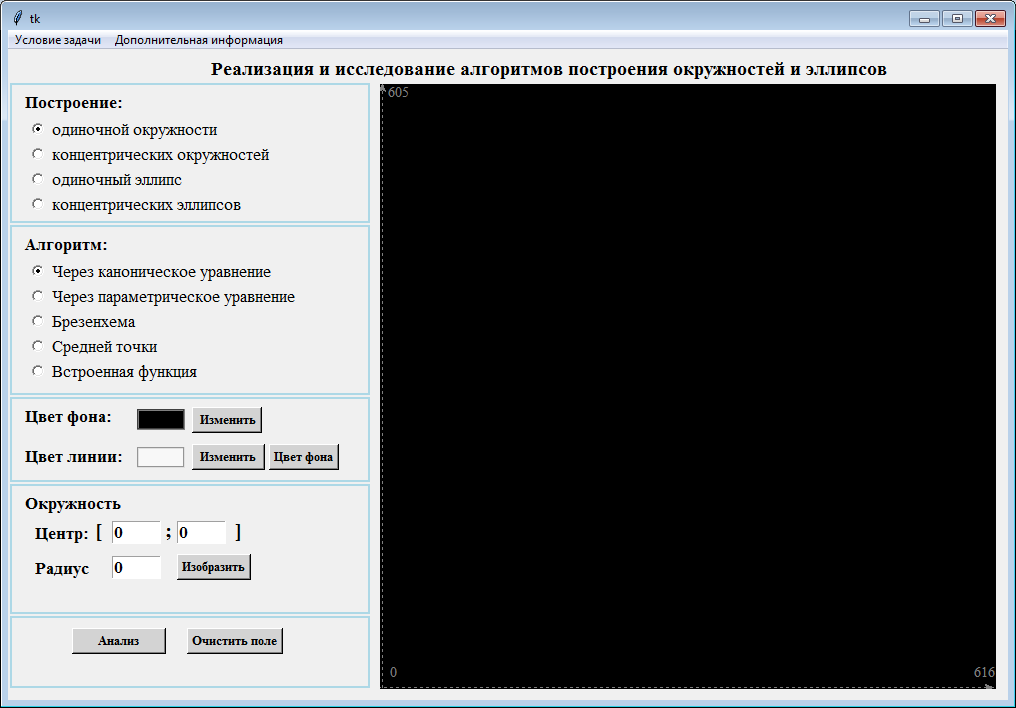
При построении эллипса пользователь выбирает из списка определенный алгоритм, задает координаты центра, полуоси, цвет рисования.

При построении спектра концентрических эллипсов пользователь выбирает из списка определенный алгоритм, задает координаты центра, цвет рисования, начальные значения полуосей, шаг изменения одной из полуосей, количество эллипсов.

**Интерфейс**

Пользователю предоставляется следующий интерфейс для работы:

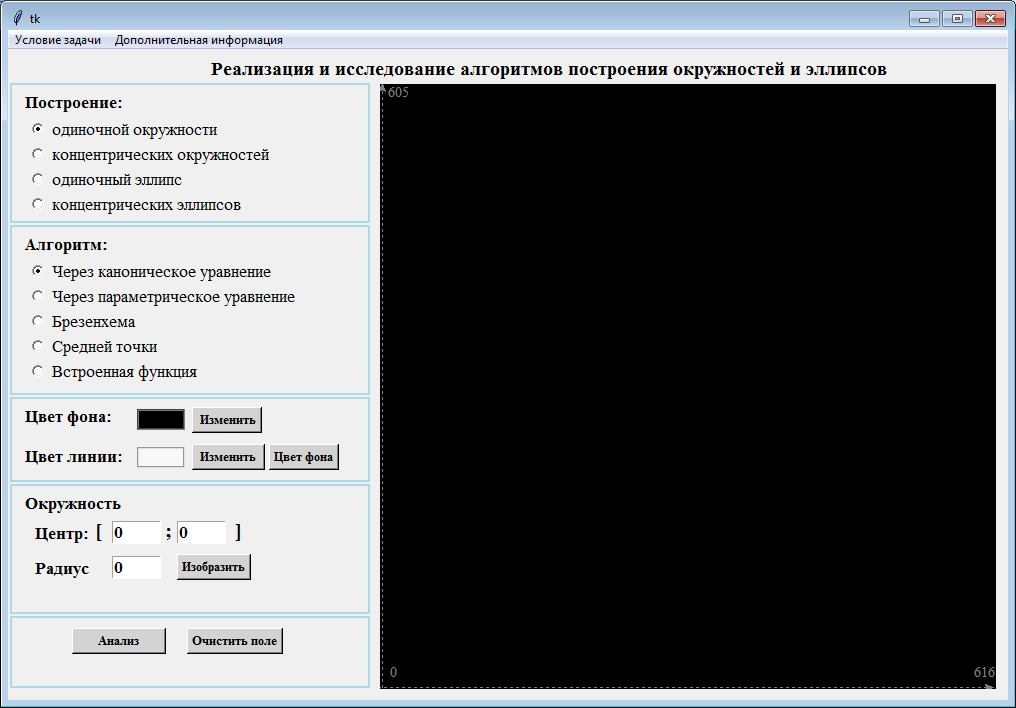




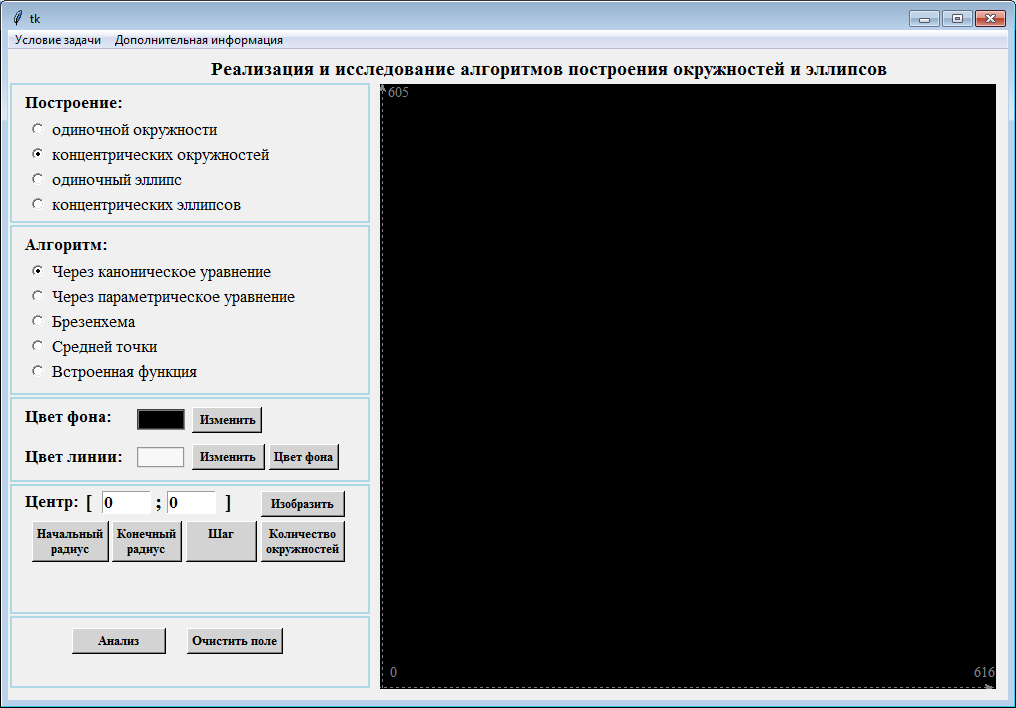
Интерфейс предоставляет возможность пользователю выбрать необходимый режим работы:

Причём в соответствии с режимом появляется необходимое меню.

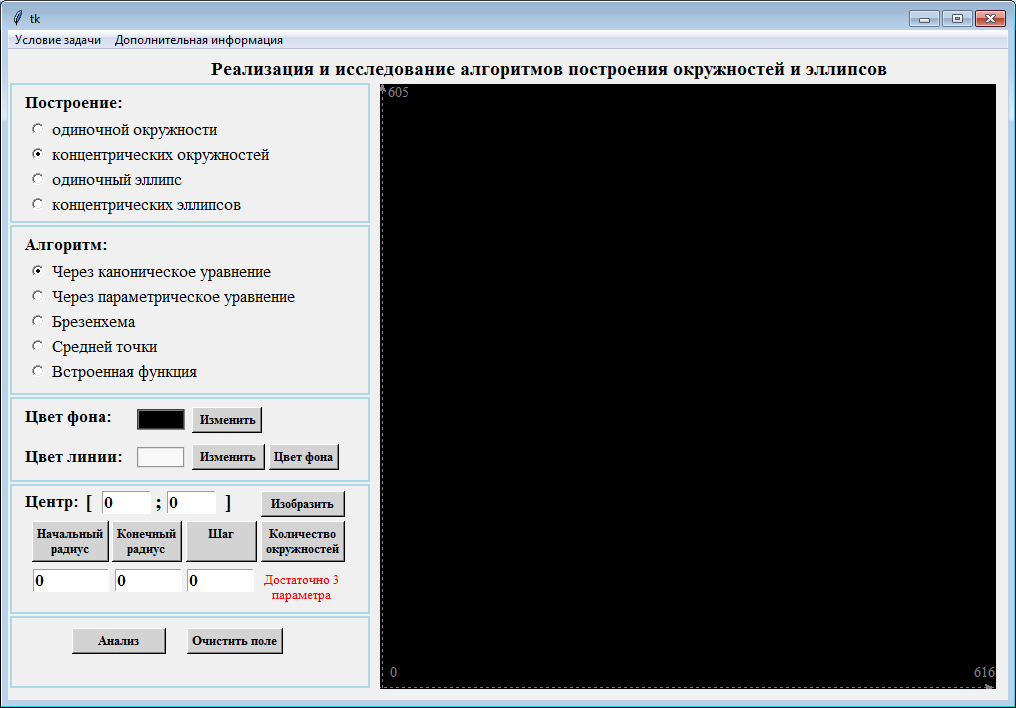
Так, для «одиночной окружности» меню выглядит следующим образом:



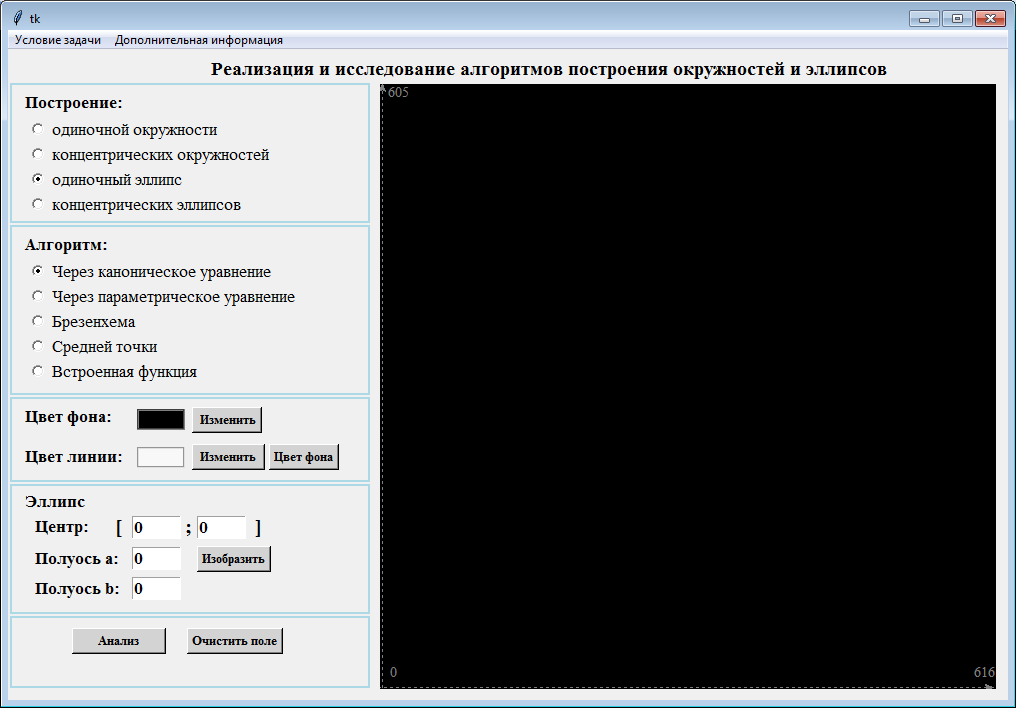
Для «концентрических окружностей»:



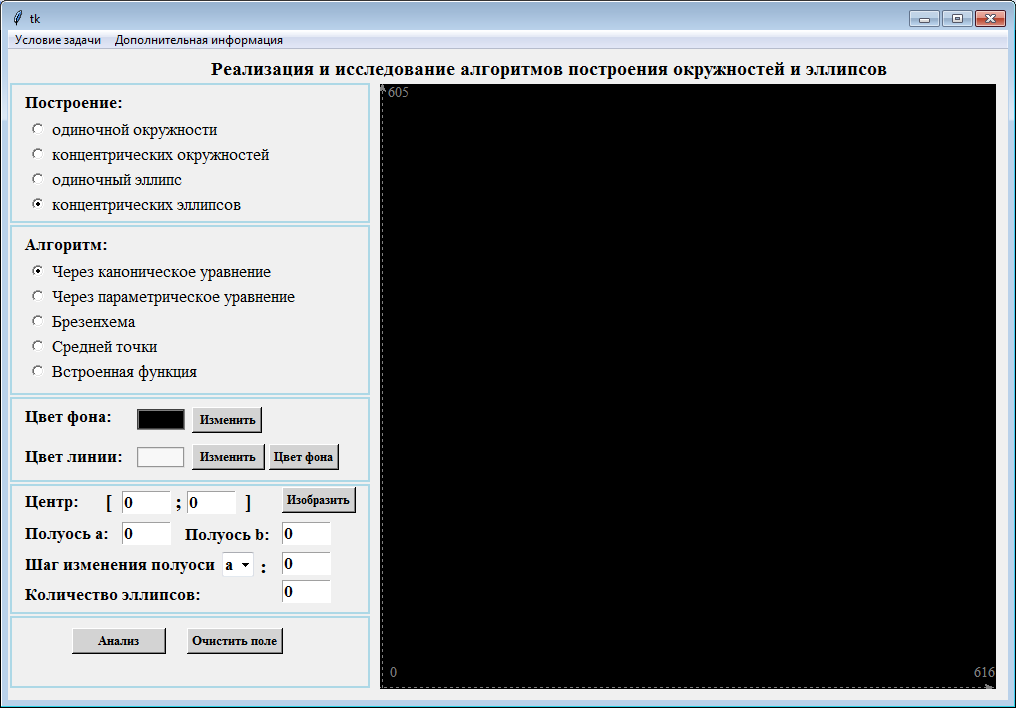
Так как пользователь должен выбрать только три параметра из четырёх, то осуществляется контроль над вводимыми значениями. Например, нельзя выбрать все доступные поля для работы – высвечивается предупреждение:



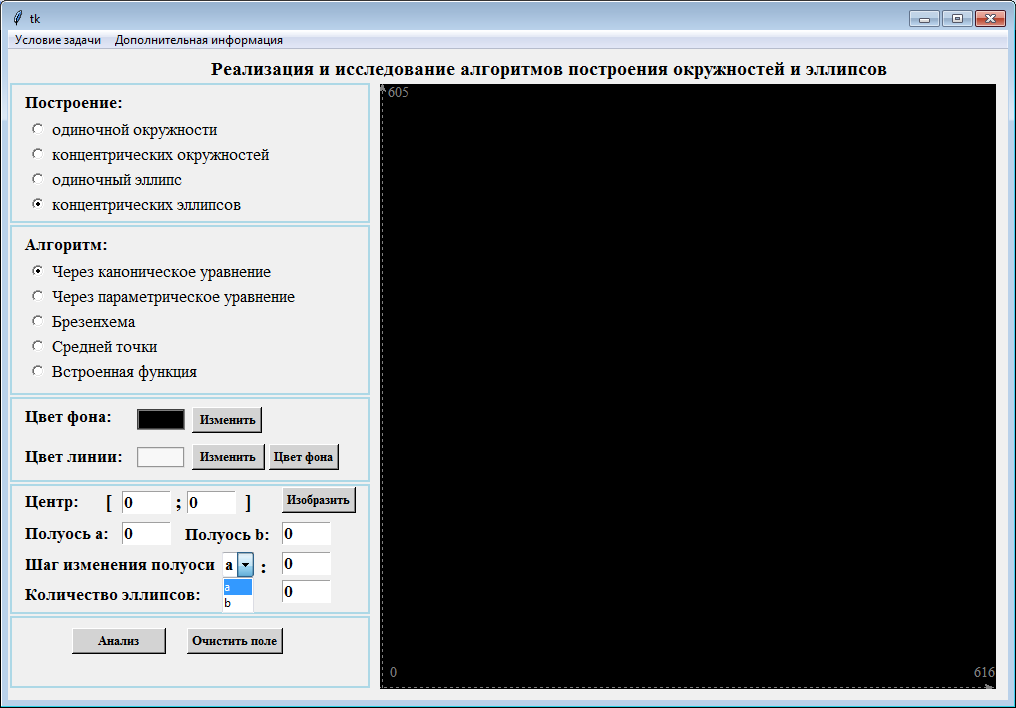
Меню для «одиночных эллипсов» выглядит следующим образом:



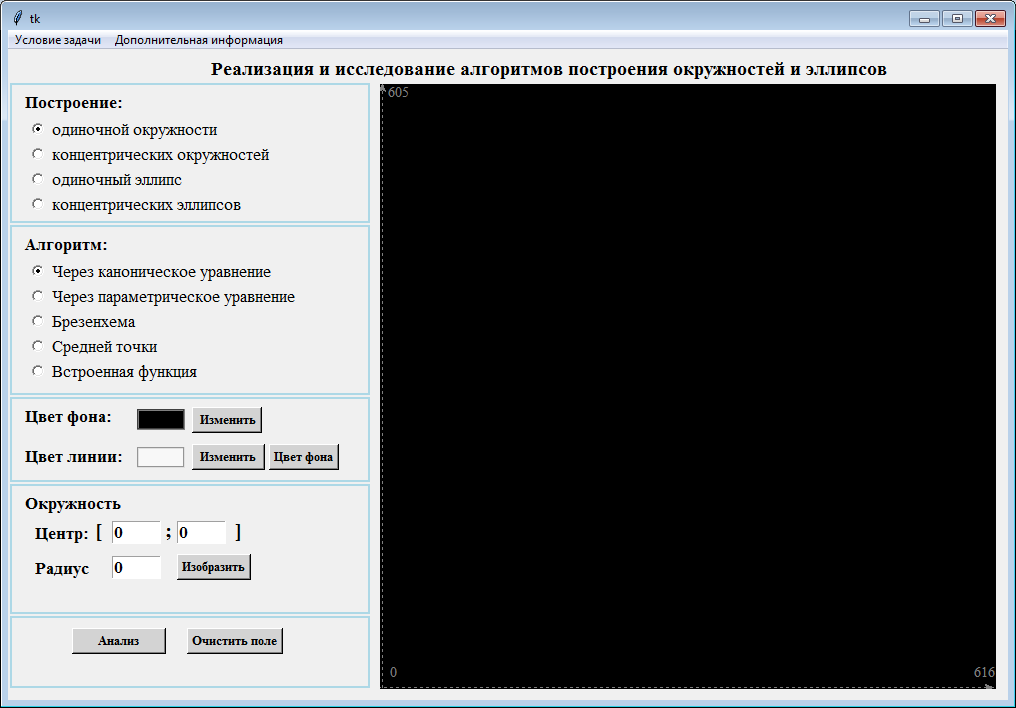
Для «концентрических эллипсов»:

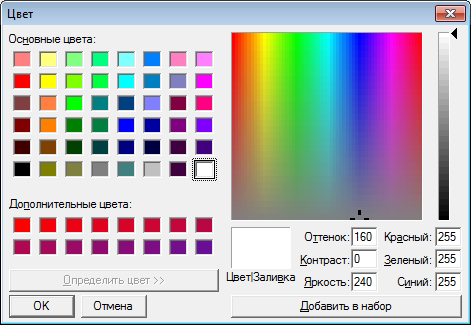
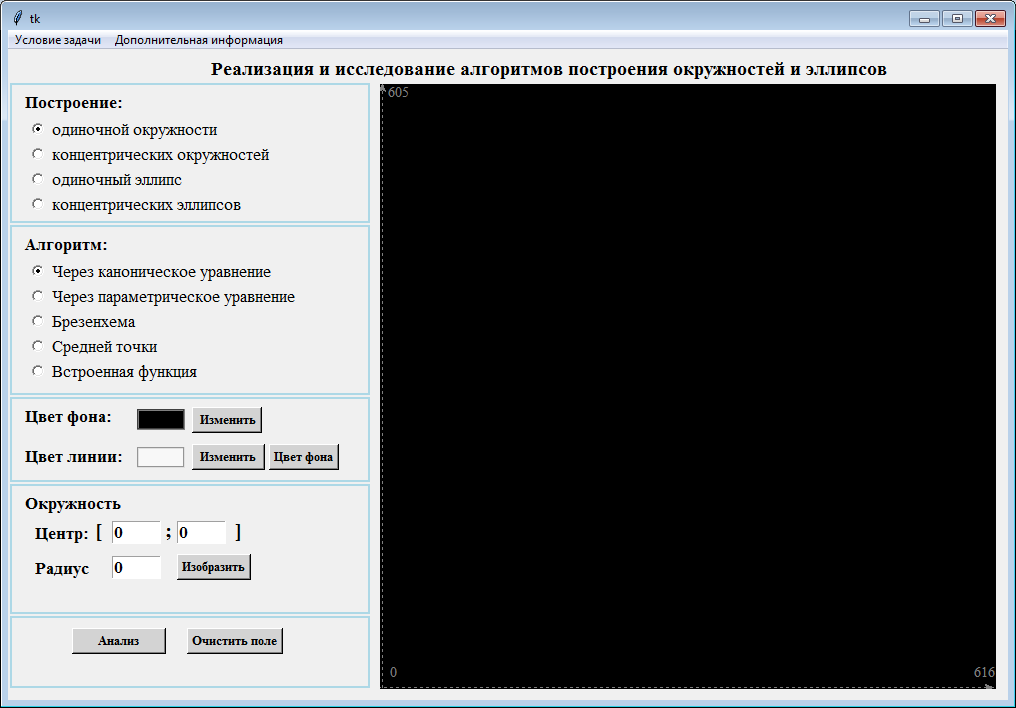


Причём, задавать шаг изменения можно как для полуоси а, так и для полуоси b (пользователь может выбрать):

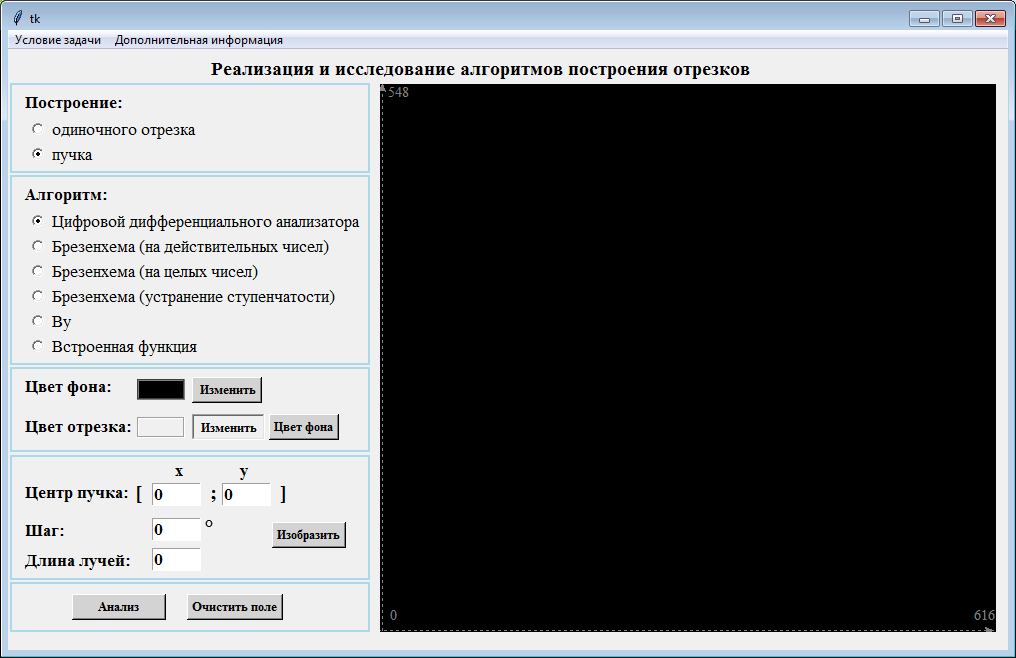


Можно выбрать алгоритм:



А также можно выбрать цвет линии и фона при нажатии на соответствующую кнопку (для отрезка предусмотрена дополнительная кнопка рисования цветом фона). По умолчанию цвет фона – чёрный, линии – белый.

Ещё предусмотрены такие кнопки, как «Очистить поле» и «Анализ».



Кроме того, за каждым полем ввода осуществляется проверка на корректные данные, и, в случае ошибки, выдаётся соответствующее сообщение.

**Рассматриваемые алгоритмы**

Общие замечания

Для изображения окружности/эллипса достаточно сгенерировать только часть фигуры, остальную можно получить путём последовательных отображений. Так, для окружности достаточно получить часть, а для эллипса .

Используемые формулы для вычисления точек окружности/эллипса написаны для случая, когда центр находится в начале координат, а само смещение осуществляется на этапе отрисовки путём переноса.

**Алгоритм построения на основе канонического уравнения**

*Об алгоритме*

В этом алгоритме используется каноническое уравнение для построения окружности и эллипса:

Для окружности уравнение имеет вид:

Для эллипса:

Из этих уравнений получаются зависимости x(y), y(x), которые используются для конкретных координатных четвертей.

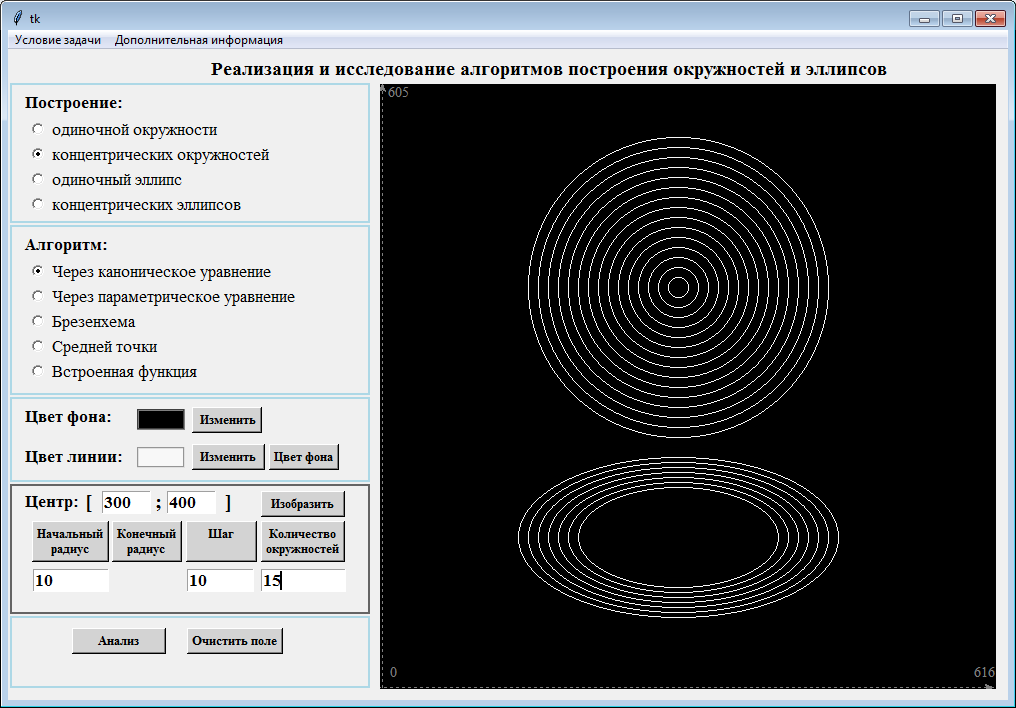
Для построения окружности используется только один цикл по х от 0 до , так как это значение описывает 1/8 окружности. Учитывая симметрию, отрисованные этим циклом пиксели можно отразить до полной окружности.

Для построения эллипса используется уже два цикла (по х и по у). Цикл по х проходит по значениям от 0 до , и цикл по у от 0 до . Таким образом, отрисовывается 1/4 эллипса, и далее отображается относительно координатных осей до полной фигуры.

*Программная реализация:*

|  |  |
| --- | --- |
| окружность | эллипс |
|  |  |

*Пример работы программы:*



**Алгоритм построения на основе параметрического уравнения**

*Об алгоритме*

В этом алгоритме используется параметрическое уравнение для построения окружности и эллипса:

Для окружности уравнение имеет вид:

Для эллипса:

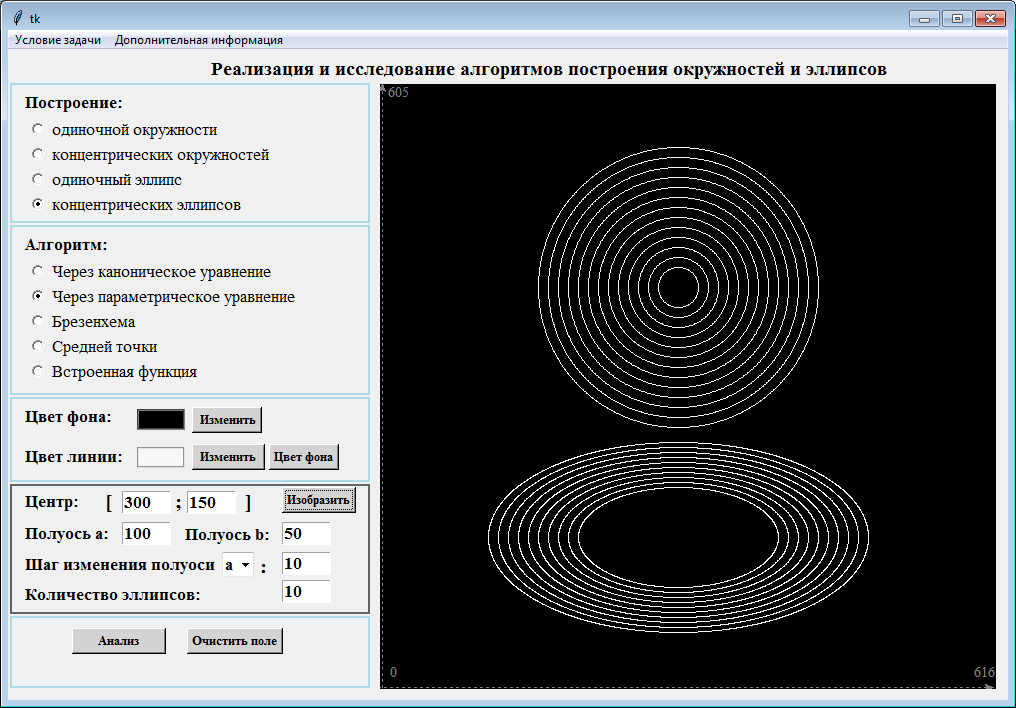
t – угол, который проходит интервал от 0 до , а в эллипсе от 0 до . После чего уже отрисованный участок отображается до полной фигуры.

Шаг выбирается согласно формуле , где R – радиус кривизны. Т.е. для окружности шаг равен , где r – радиус окружности. Для эллипса меняется от до , поэтому для удобства и полной отрисовки выбирается наименьшая из этих двух величина.

*Программная реализация:*

|  |  |
| --- | --- |
| окружность | эллипс |
|  |  |

*Пример работы программы:*

****

**Алгоритм Брезенхема**

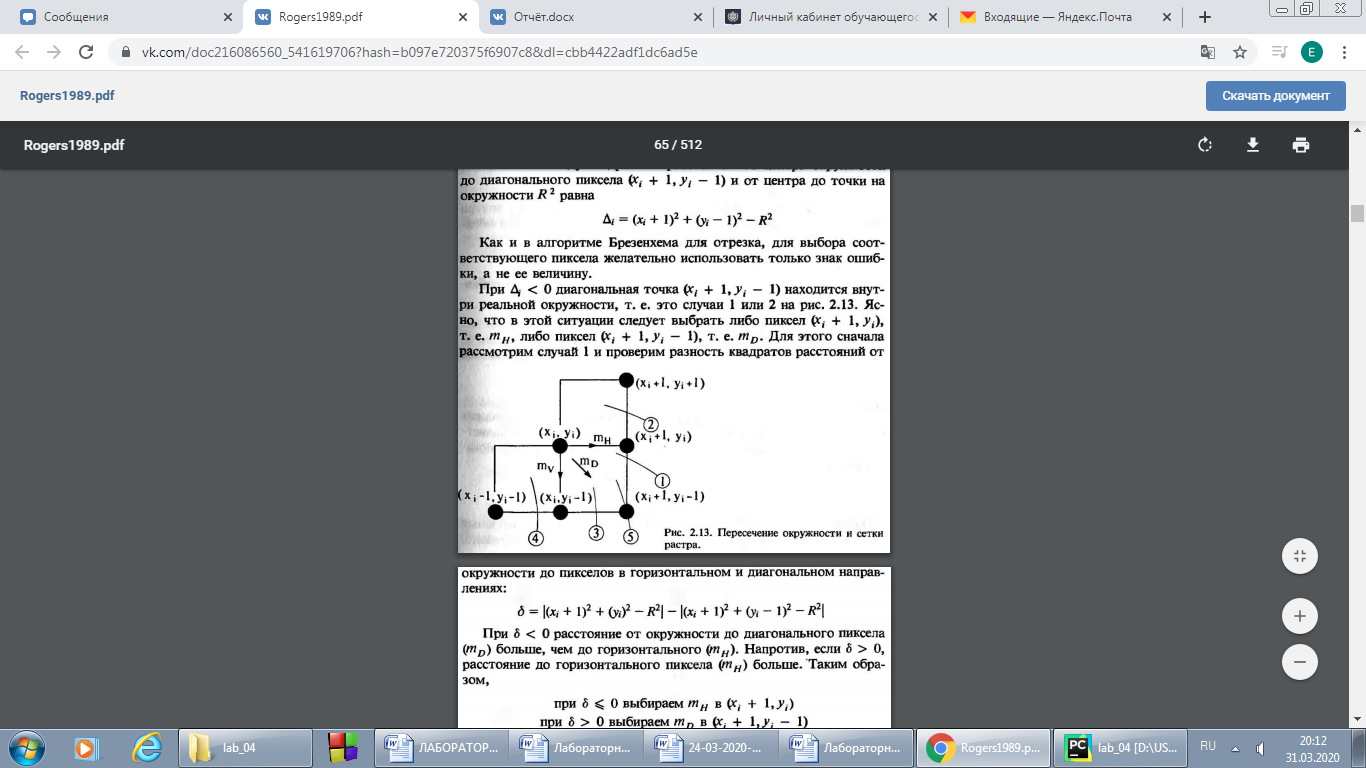
*Об алгоритме*

Этот алгоритм один из наиболее эффективных. Стартовой точкой является точка с координатами (0, r) для окружности и (0, b) для эллипса.

Так же, как и в предыдущих алгоритмах, построение производится только части фигуры.

Построение производится только в трёх направлениях: горизонтально вправо, вертикально вниз, по диагонали (вниз и вправо), и высвечиваются необходимые пиксели. Цикл, в котором вычисляются нужные координаты, продолжается до тех пор, пока у больше или равен (для окружности) или 0 (для эллипса). Тем самым отрисовывается часть, по которой дальше будет генерироваться целая фигура.

В процессе вычислений выбирается точка, которая наиболее приближена к идеальной фигуре (окружности или эллипсу). Для оценки расстояний используется такое понятие как «ошибка». Сравнивая её с нулём, определяется положение диагонального пикселя (меньше нуля – внутри, равно нулю – на самой фигуре, больше нуля – вне фигуры).

Возможны пять случаев прохождения окружности/эллипса.

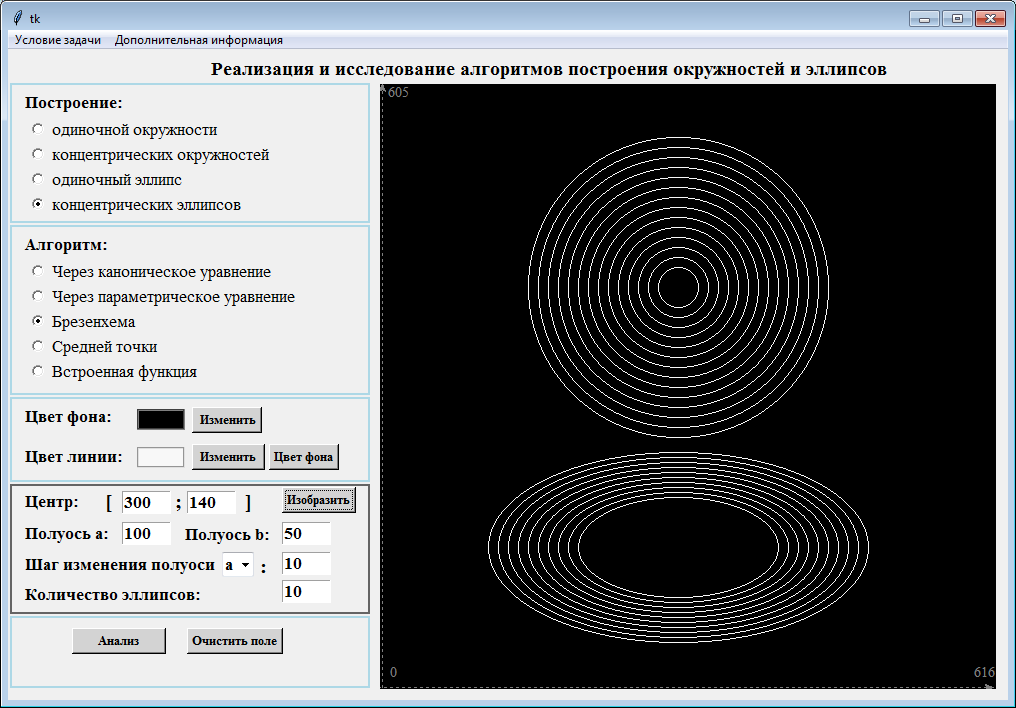
В каждом случае рассматривается положение диагонального пикселя и делается соответствующий вывод о подходящем шаге (горизонтальный, диагональный или вертикальный)

Но так как у окружности рассматривается только 1/8 часть, то выбор в двух случаях очевиден, зато при построении эллипса нужно их учесть.

*Программная реализация:*

|  |  |
| --- | --- |
| окружность | эллипс |
|  |  |

*Пример работы программы:*

****

**Алгоритм средней точки**

*Об алгоритме*

Это пошаговый алгоритм, название которого связано с тем, что на каждом шаге работы выбирается ближайший к фигуре пиксель из двух возможных, анализируя, находится ли средняя точка между этими пикселями внутри или вне фигуры.

В этом алгоритме используется такое понятие, как «пробная функция». Если в пробную функцию подставить точку, лежащую на эллипсе, то получится нулевое значение. Если точку, лежащую внутри эллипса, то значение будет отрицательно. А если точку вне эллипса - положительный результат.

Анализируя знак пробной функции для средней точки, расположенной между двумя альтернативными пикселями, нужно выбрать ближайший к фигуре.

Выделяются два интервала, которые разграничиваются точкой, где .

Выбор очередного пикселя зависит от значения производной. При выбор должен осуществляться между двумя вертикальными пикселями, а при – между двумя горизонтальными.

Интервал 1

* Функция средней точки
* Горизонтальный шаг:
* Диагональный шаг:

Интервал 2

* Функция средней точки
* Вертикальный шаг:
* Диагональный шаг:

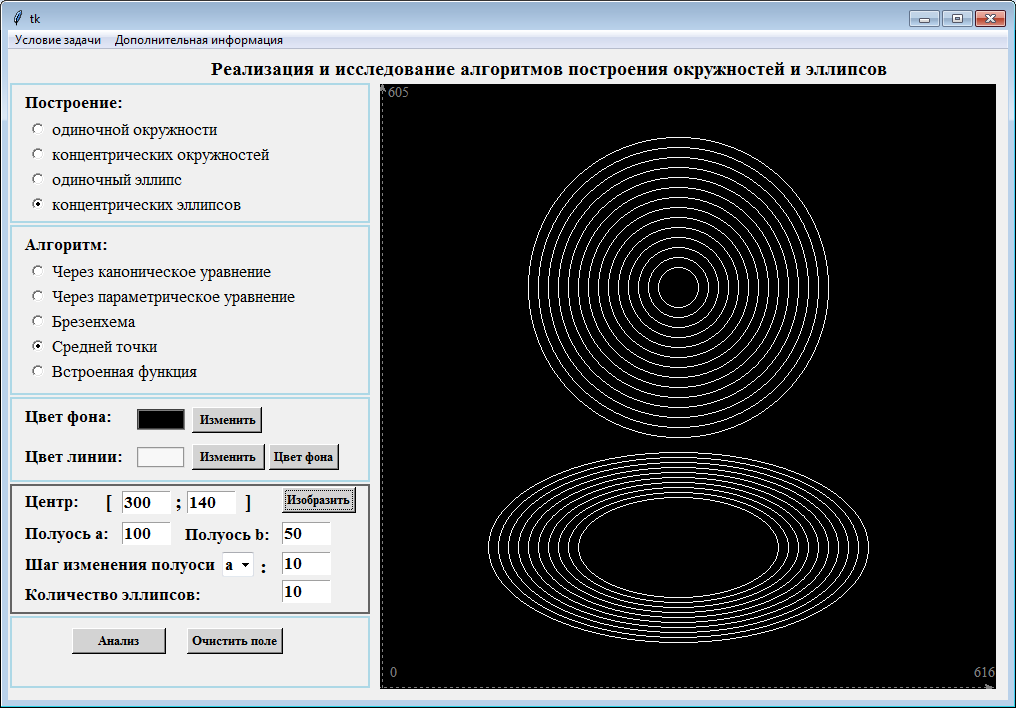
(приведенные формулы справедливы для эллипса, для окружности нужно принять a = b = r)

Рекомендуется оптимизировать алгоритм путём отказа от операции умножения и ограничиться сложением, также запоминать некоторые значения в отдельные переменные и использовать их, чтобы не вычислять повторно.

*Программная реализация:*

|  |  |
| --- | --- |
| окружность | эллипс |
|  |  |

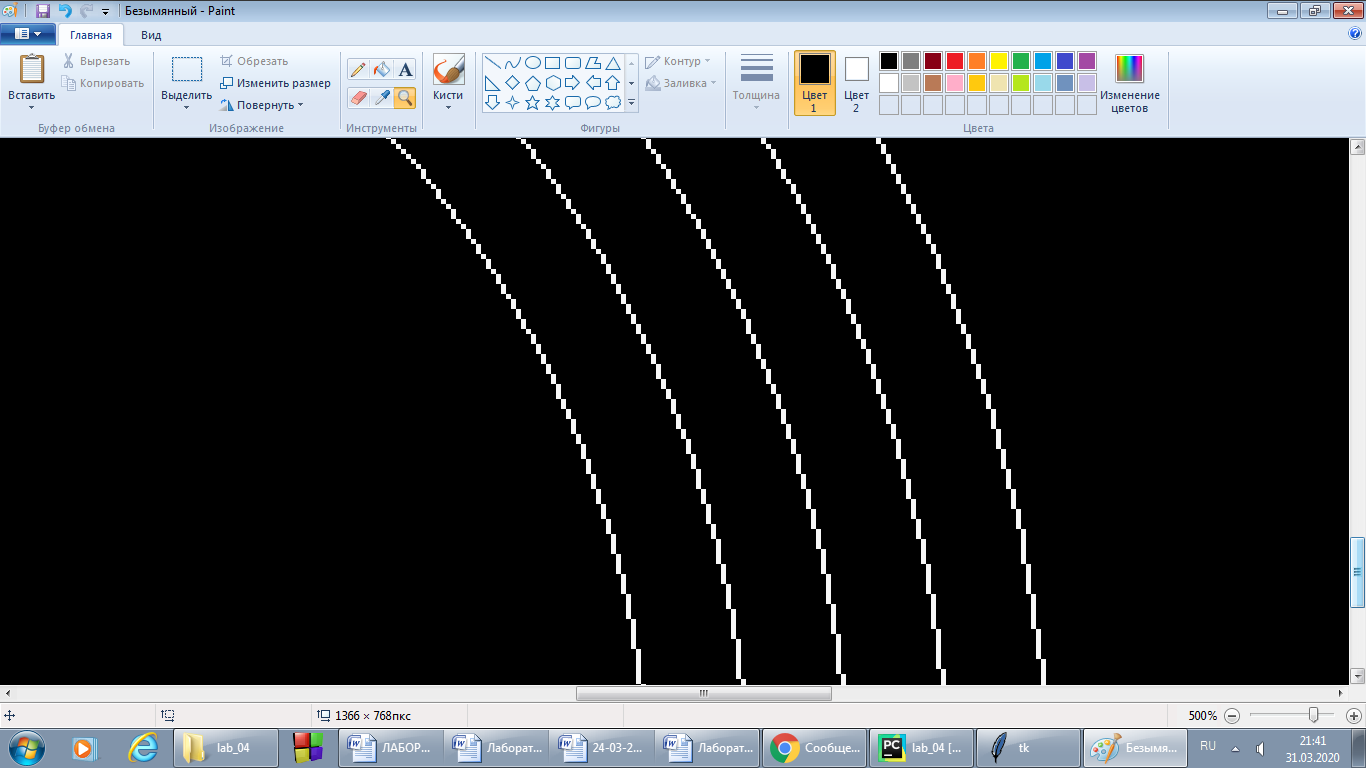
*Пример работы программы:*



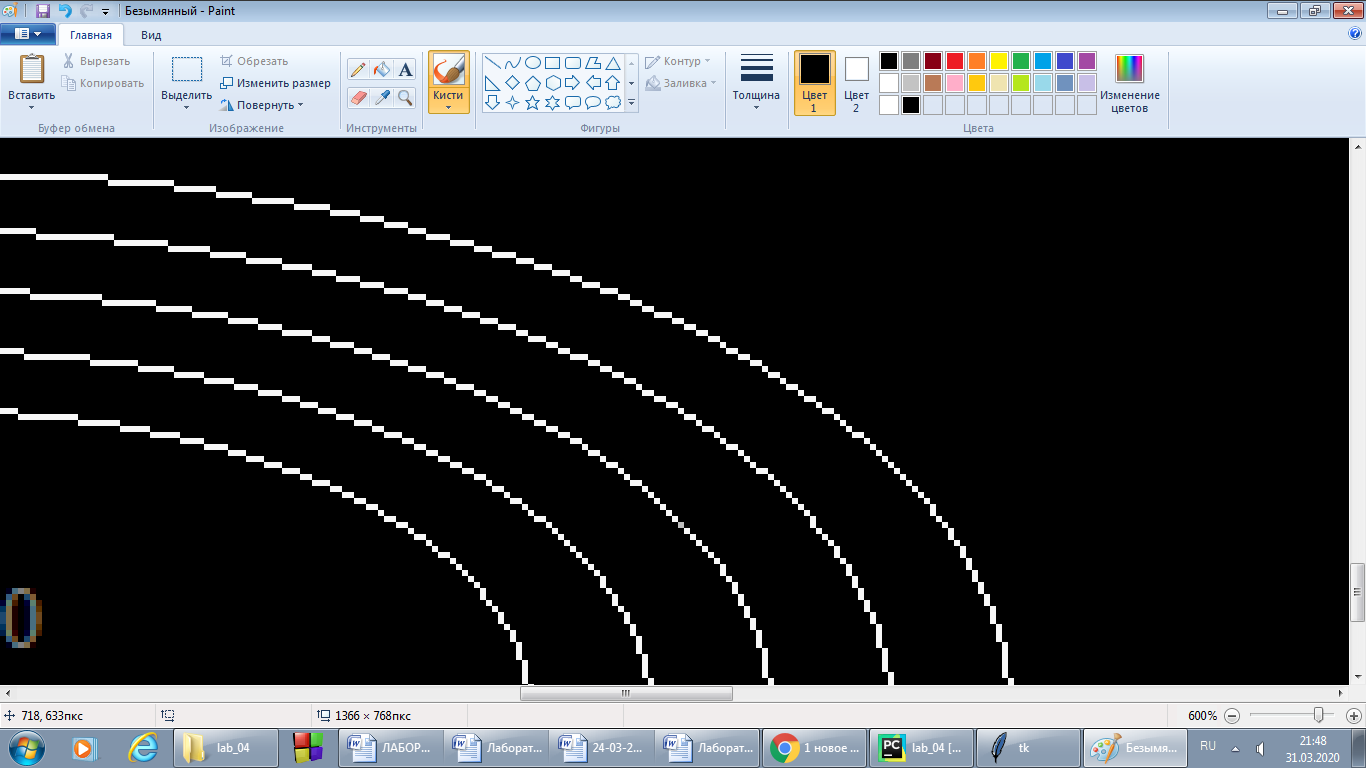
**Исследование временных и визуальных характеристик**

Произведём сравнение визуальных характеристик путём изображения одного и того же объекта разными алгоритмами.

Каноническое уравнение -> Параметрическое уравнение -> Брезенхем -> Средняя точка -> Встроенная функция



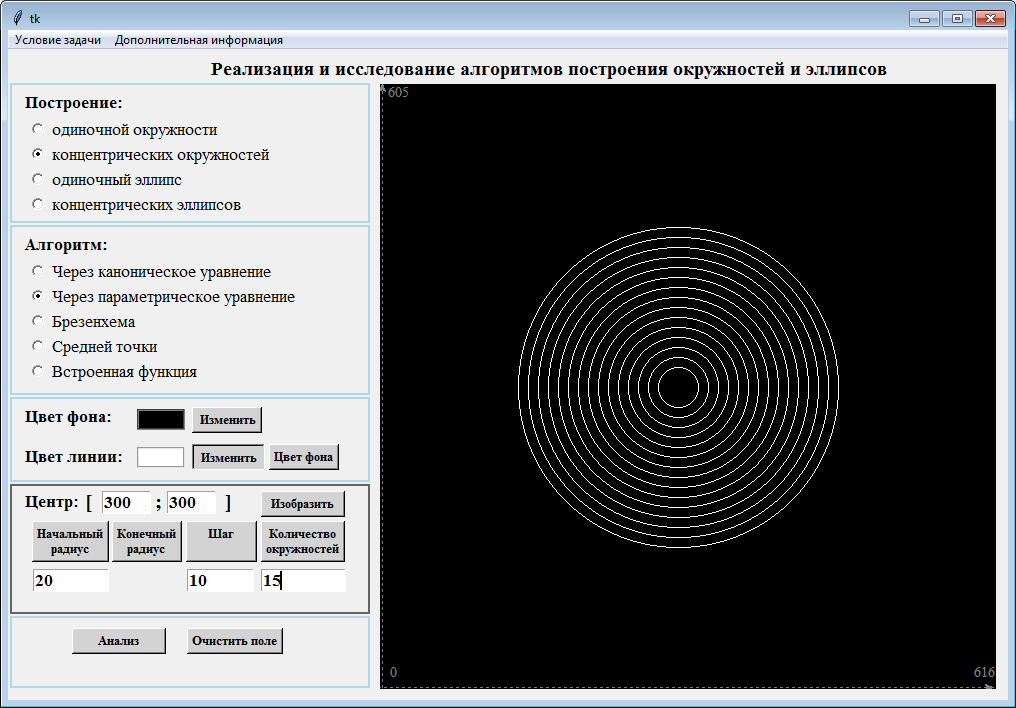
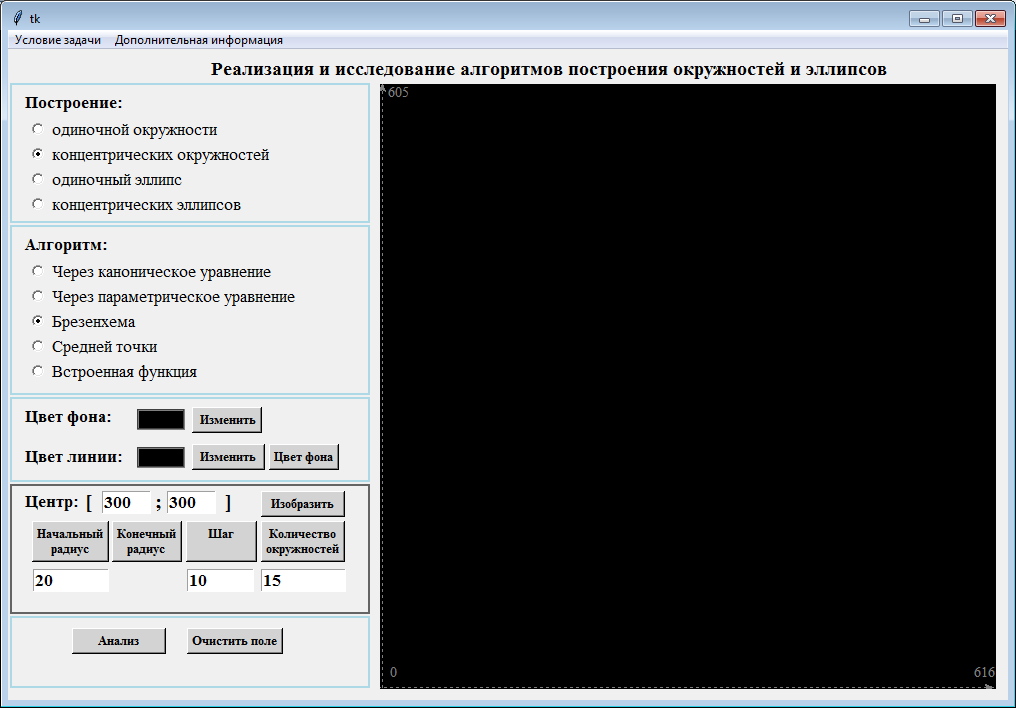
(построение окружностей с одним центром разными методами)



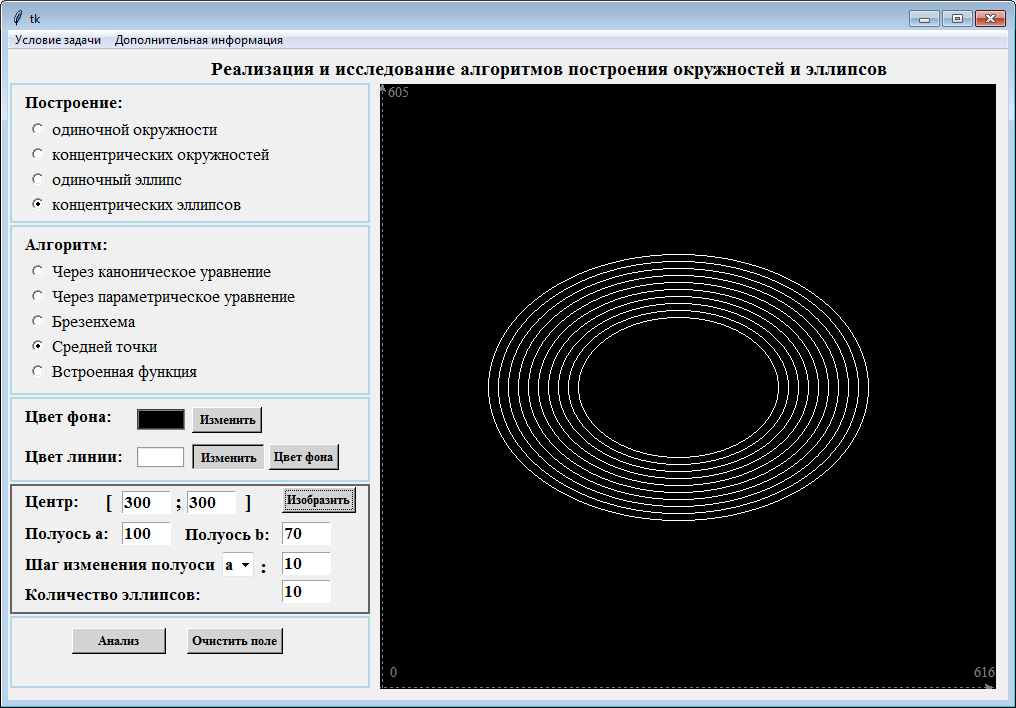
(построение эллипсов с одним центром разными методами)

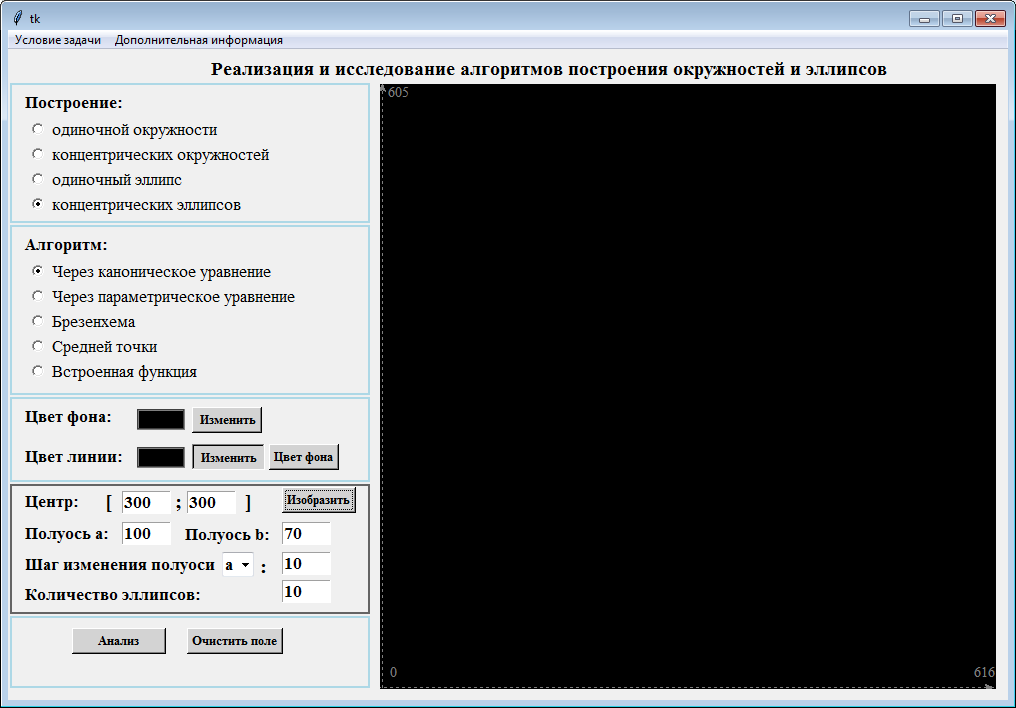
При наложении было выявлено, что алгоритм средней точки, канонического, параметрического уравнения, алгоритм Брезенхема и встроенный взаимно закрашивают друг друга.

Например, сравним окружности, построенные с помощью параметрического уравнения и алгоритма Брезенхема:



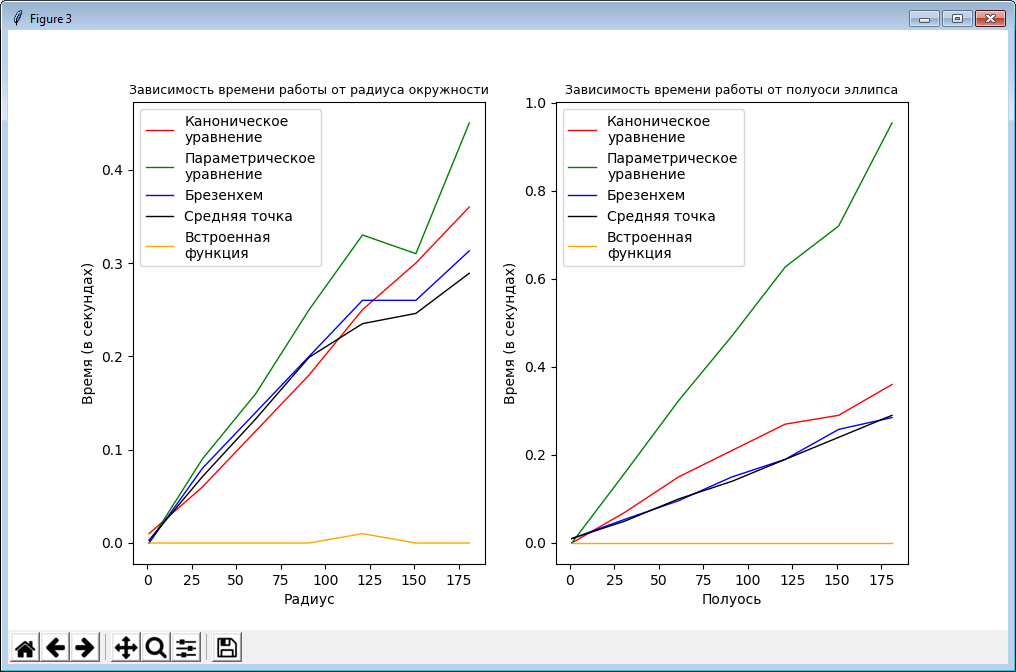
(Закрасилось)

И ещё один пример на эллипсах (алгоритм средней точки и канонического уравнения):



(Закрасилось)

Ниже представлены диаграммы анализа (зависимость времени работы разных алгоритмов от радиуса (для окружностей) и от полуоси (для эллипсов)):



Таким образом, наиболее затратными по времени получились методы, связанные с уравнениями (параметрическое и каноническое уравнения). Менее затратными оказались алгоритм Брезенхема и средней точки. На эллипсах работают практически одинаково.

Глядя на графики, можно сказать, что зависимость практически линейная. Например, линейная зависимость для ситуации с окружностью исходит из формулы длины окружности C = 2πR.

**Заключение**

Таким образом, в ходе лабораторной работы были изучены и реализованы несколько алгоритмов построения окружностей и эллипсов, проанализирована их работа, и проведён визуальный и временной сравнительный анализ, по итогам которого были сделаны соответствующие выводы.