

Московский авиационный институт
(Национальный исследовательский университет)

Факультет: «Информационные технологии и
прикладная математика»

Кафедра: «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

Лабораторная работа №1 по курсу «Компьютерная графика»

Тема: Построение изображений 2D-кривых.

Студент: Тимофеев А. В.

Преподаватель: Морозов А. В.

Группа: М80-307Б

Дата:

Оценка:

Подпись:

Москва, 2022

Постановка задачи

Написать и отладить программу, строящую изображение заданной замечательной кривой.
Вариант 3: $x = a \cdot \cos 3\phi$, $y = a \cdot \sin 3\phi$

ρ, ϕ - полярные координаты, x, y – декартовы координаты t – независимый параметр.

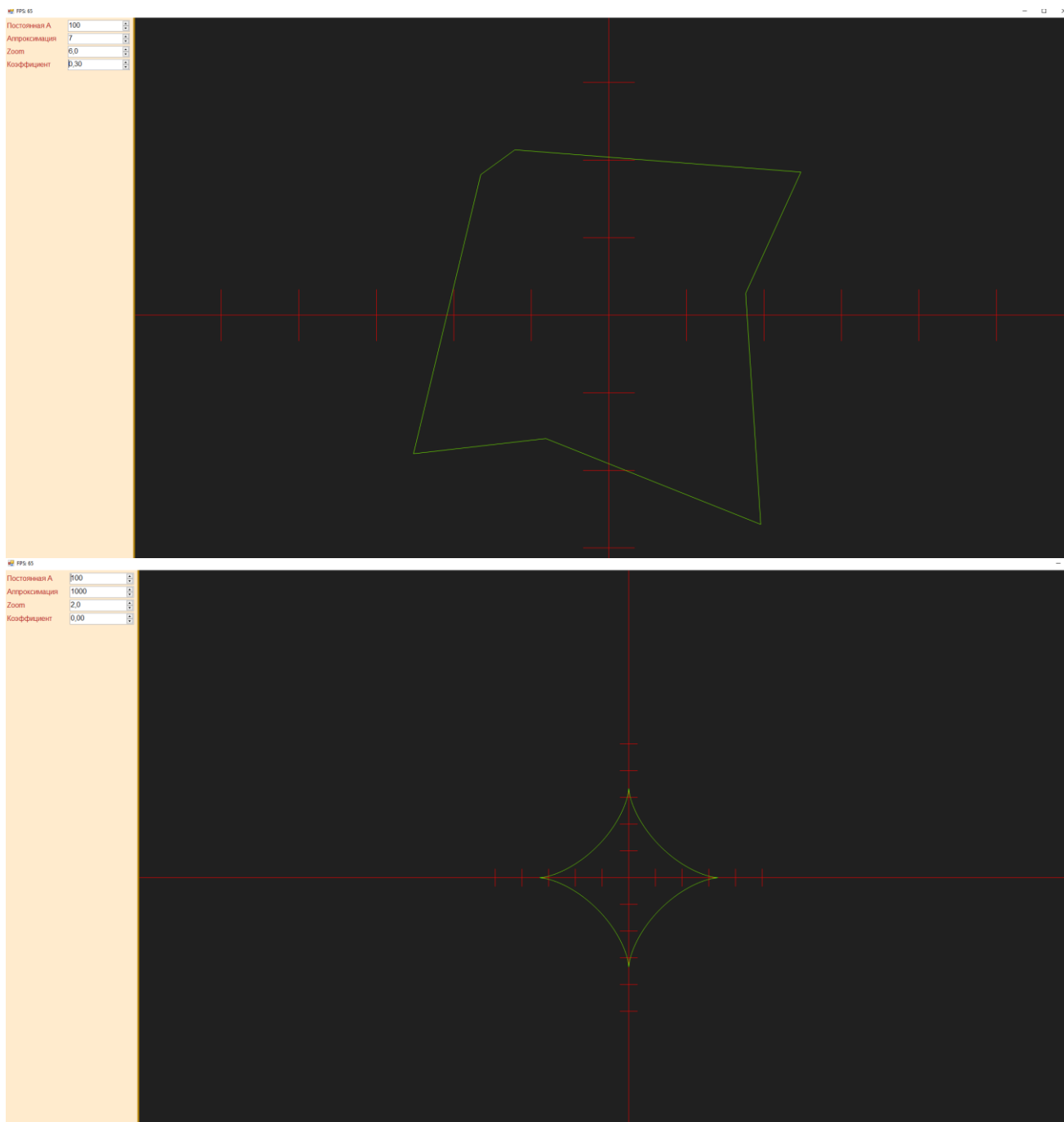
a, b, k, A, B , - константы, значения которых выбираются пользователем (вводятся в окне программы). $a, b > 0$

Обеспечить автоматическое масштабирование и центрирование кривой при изменении размеров окна.

Решение задачи

По условию переведем координаты в полярные координаты: $x = a \cdot \cos 3\phi$, $y = a \cdot \sin 3\phi$.
В итоге параметризация происходит по ϕ : 2π делится на заданное пользователем число и рисовка происходит с соответствующим шагом. Также здесь реализованы сдвиг, поворот графика, автомасштаб/масштаб и рисовка осей. Для корректного отображения к соответствующим координатам прибавлены половины от высоты/ширины окна.

Пример работы



Листинг программы

(основная часть)

```
protected override void OnDeviceUpdate(object s, DeviceArgs e){
    var step = 2 * Math.PI / VertexCount;
    double angle = 0;
    double X, Y;
    //создаем OX, OY и разметку
    var mOX = new DVector2(-800, 0);
    var OX = new DVector2(800, 0);
    var mOY = new DVector2(0, -800);
    var OY = new DVector2(0, 800);
    e.Surface.DrawLine(Color.Red.ToArgb(), FromViewToPhysicalSpace(mOX * zoom + Shift),
        FromViewToPhysicalSpace(OX * zoom + Shift));
    e.Surface.DrawLine(Color.Red.ToArgb(), FromViewToPhysicalSpace(mOY * zoom + Shift),
        FromViewToPhysicalSpace(OY * zoom + Shift));
    DrawScaleOY(e);
    DrawScaleOX(e);
    //создали OX, OY и разметку
    var points = new List<DVector2>();
    while (angle < 2 * Math.PI) { // просчет точек графика
        X = A * Math.Pow(Math.Cos(angle), 3) * zoom;
        Y = A * Math.Pow(Math.Sin(angle), 3) * zoom;
        points.Add(new DVector2(X, Y) + Shift);
        angle += step;
    }
    X = A * Math.Pow(Math.Cos(angle), 3) * zoom;
    Y = A * Math.Pow(Math.Sin(angle), 3) * zoom;
    points.Add(new DVector2(X, Y) + Shift);
    for (var i = 1; i < points.Count; ++i) // рисуем график
        e.Surface.DrawLine(Color.LawnGreen.ToArgb(),
Rotation(FromViewToPhysicalSpace(points[i])),
        Rotation(FromViewToPhysicalSpace(points[i - 1])));
    }
}
```

Вывод

В результате выполнения данной лабораторной работы я познакомился с возможностью отрисовки 2D изображений кривых посредством вызова методов рисования отрезков в форме приложения. Также я научился применять такие простые аффинные преобразования как сдвиг, поворот и масштабирование.

