

**Московский авиационный институт
(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

Лабораторная работа № 1

Тема: Построение изображений 2D-кривых

Студент: Мукин Юрий

Группа: 80-304

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

1. Постановка задачи

Задание: Написать и отладить программу, строящую изображение заданной замечательной кривой.

Изначально функция дана в полярных координатах:

$$r = a \cdot \cos(3 \cdot \phi)$$

для удобства преобразуем в декартовы координаты и дальше будем работать именно с этой формой записи:

$$x = r \cdot \cos(\phi) = a \cdot \cos(3 \cdot \phi) \cdot \cos(\phi)$$

$$y = r \cdot \sin(\phi) = a \cdot \cos(3 \cdot \phi) \cdot \sin(\phi)$$

2. Описание программы

Задание выполнено на язык C# с помощью стандартного элемента управления **chart**. Этот элемент позволяет отображать различные типы графиков, в данном случае был использован метод построения **Line**. Этот метод строит не сглаженную прямую по набору точек.

Программа позволяет задавать следующие значения:

1. **const** - в постановке задачи это значение этого поля соответствует **a**.
2. **from** - начальное значение угла **phi**.
3. **to** - конечное значение угла **phi**.
4. **step** - шаг с которым происходит итерирование. (меньше шаг - выше точность).

По нажатию кнопки **draw** считываются введенные значения и если они удовлетворяют требованиям рассчитывается новый набор точек и перерисовывается график.

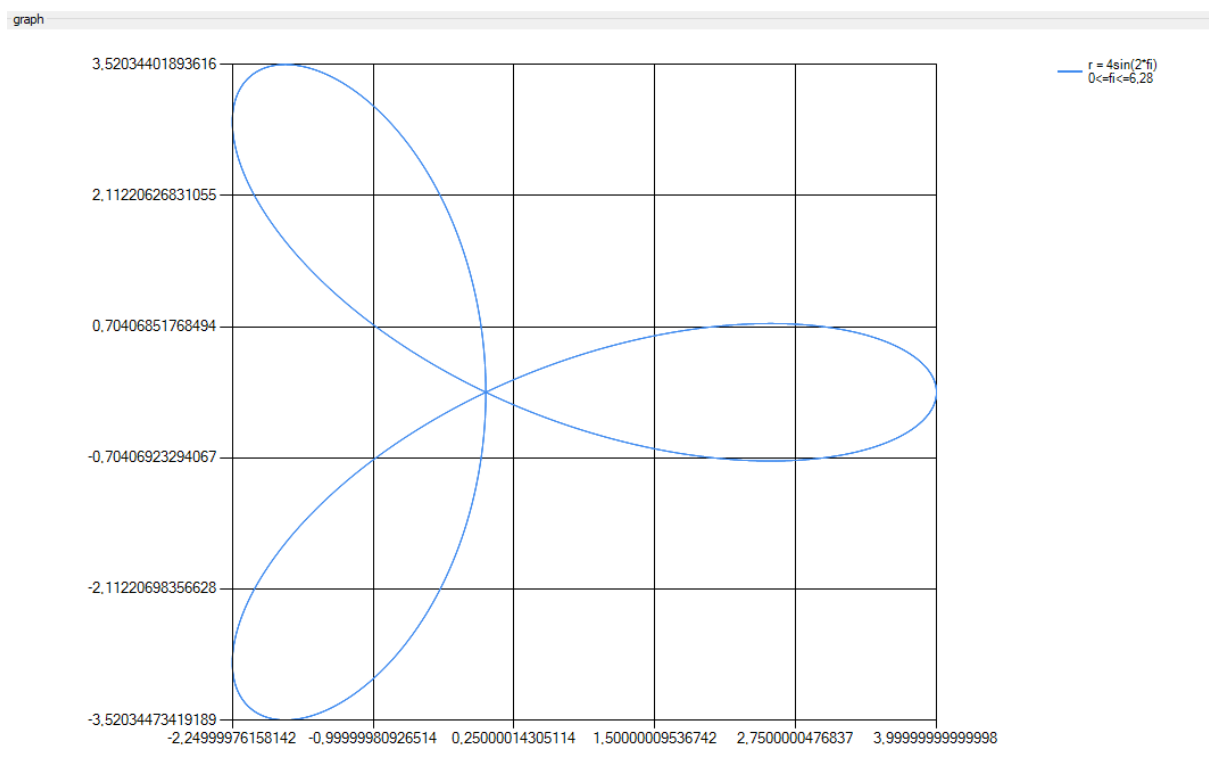
В постановке задачи нет требования задавать промежуток в котором находятся значения угла, поэтому были установлены значения по умолчанию: **from = 0; to = 6.28; step = 0.001**.

3. Набор тестов

input		input	
const:	<input type="text" value="4"/>	const:	<input type="text" value="1"/>
from:	<input type="text" value="0"/>	from:	<input type="text" value="0"/>
to:	<input type="text" value="6,28"/>	to:	<input type="text" value="6,3"/>
step:	<input type="text" value="0,001"/>	step:	<input type="text" value="0,001"/>
<input type="button" value="draw"/>		<input type="button" value="draw"/>	

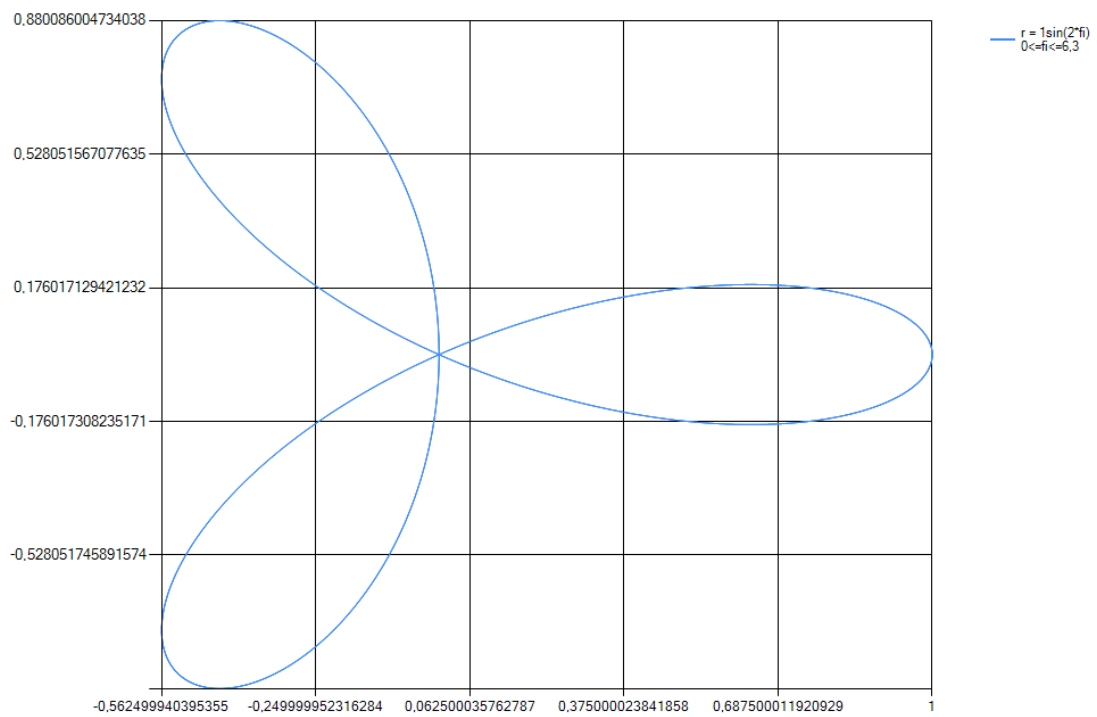
input		input	
const:	<input type="text" value="5"/>	const:	<input type="text" value="5"/>
from:	<input type="text" value="0"/>	from:	<input type="text" value="0"/>
to:	<input type="text" value="3,14"/>	to:	<input type="text" value="1,7"/>
step:	<input type="text" value="0,001"/>	step:	<input type="text" value="0,1"/>
<input type="button" value="draw"/>		<input type="button" value="draw"/>	

4. Результаты выполнения тестов

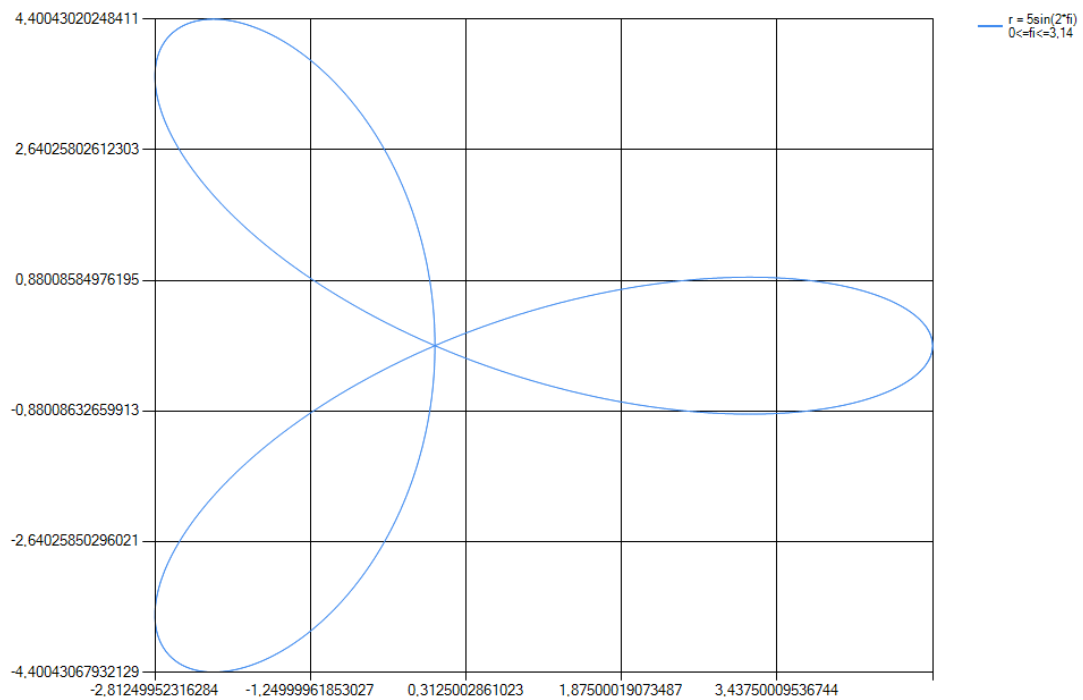


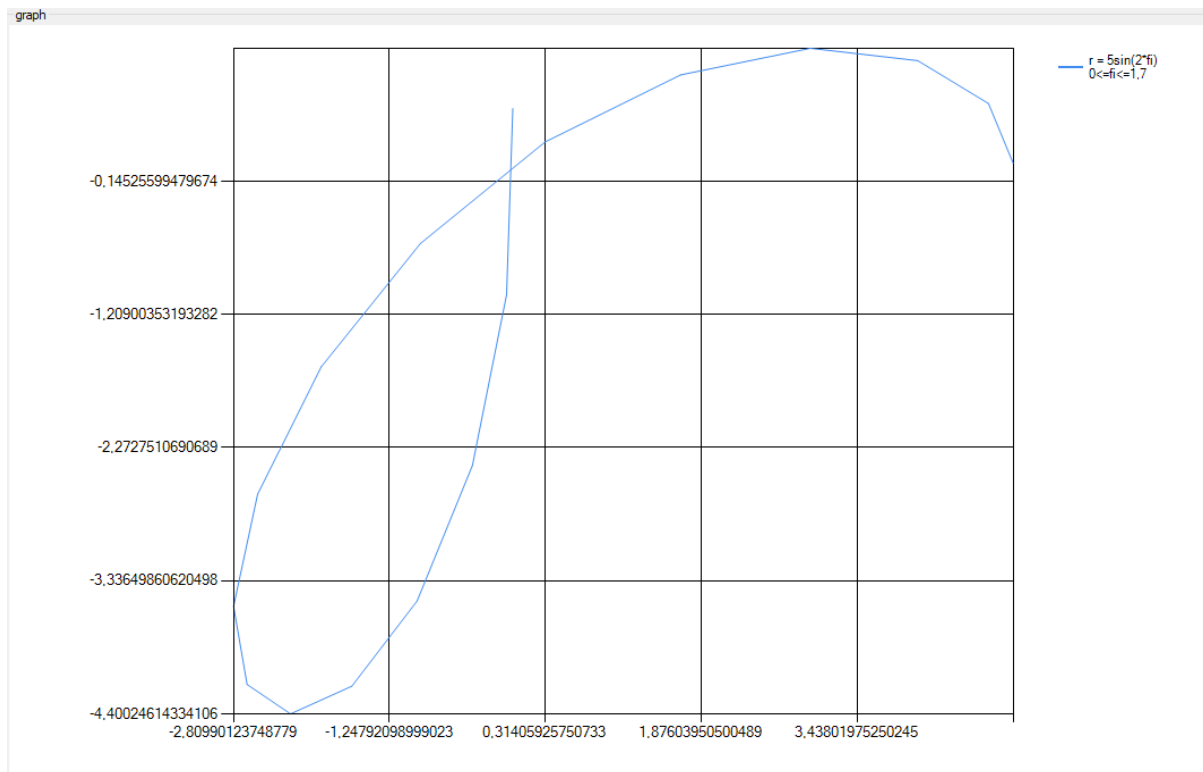
Form1

graph



graph





5. Листинг программы

Сама функция:

```
class function
```

```
{
    private double a;
    public float Ymax, Ymin, Xmax, Xmin;
    public Stack<PointF> points;
    public function(double st, double ed, double sp, double a)
    {
        this.a = a;
        this.VoVal(st, ed, sp);
    }
}
```

//пройдемся по промежутку заданному пользователем с заданным шагом и найдем значения минимумов и максимумов

```
public void VoVal(double st, double ed, double sp)
```

```
{
    PointF tmp = value(st);
    Ymax = tmp.Y;
    Xmax = tmp.X;
    Ymin = Ymax;
    Xmin = Xmax;
    this.points = new Stack<PointF>();
    for (double i = st; i <= ed; i += sp)
    {
        tmp = value(i);
        this.points.Push(tmp);
        if (tmp.Y > Ymax)
            Ymax = tmp.Y;
        if (tmp.Y < Ymin)
            Ymin = tmp.Y;
        if (tmp.X > Xmax)
```

```

        Xmax = tmp.X;
        if (tmp.X < Xmin)
            Xmin = tmp.X;
    }
}
// преобразуем полярные координаты в декартовы
public PointF value(double phi)
{
    return new PointF((float)(this.r(phi) * Math.Cos(phi)), (float)(this.r(phi) *
Math.Sin(phi)));
}
//вычисляем значение радиус вектора
private double r(double phi)
{
    return this.a * Math.Cos(3*phi);
}
}

```

Метод для вывода графика:

```

private void drawing()
{
    //пересчитываем значение функции с заданными значениями
    f = new function(Convert.ToDouble(from.Text), Convert.ToDouble(to.Text),
Convert.ToDouble(step.Text), Convert.ToDouble(constanta.Text));
    this.chart1.ChartAreas[0].AxisX.Minimum = f.Xmin;//устанавливаем актуальные
границы
    this.chart1.ChartAreas[0].AxisX.Maximum = f.Xmax;
    this.chart1.ChartAreas[0].AxisY.Minimum = f.Ymin;
    this.chart1.ChartAreas[0].AxisY.Maximum = f.Ymax;
    chart1.Series[0].Points.Clear();//очищаем панель
    chart1.Series[0].Name = "r = " + constanta.Text + "sin(2*fi)" + '\n' +
from.Text + "<=fi<=" + to.Text;//подписываем график
    foreach (PointF p in f.points)//добавляем все вычисленные значения
        chart1.Series[0].Points.AddXY(p.X, p.Y);
}

```

Список литературы

- 1) <https://docs.microsoft.com/ru-ru/>