

## Лабораторная работа 7

### Тема Введение в графику

**Рекомендации по выполнению работы.** Для вывода линий, геометрических фигур, текста и изображений необходимо создать экземпляр класса Graphics, описанного в пространстве имен System Drawing.

Существует три способа создания объекта данного класса.

1. Ссылку на объект Graphics получают из параметра PaintEventArgs, передаваемого в обработчик события Paint, возникающего при необходимости прорисовки формы или элемента управления:

```
Private void Form1_Paint (object sender, PaintEventArgs e)
{ Graphics g=e.Graphics; // использование объекта
}
```

2. Использование метода CreateGraphics, описанного в классах формы и элемента управления:

```
Graphics g;
g= this.CreateGraphics;
```

3. Создание объекта с помощью объекта-потомка Image. Этот способ используется для изменения существующего изображения:

```
Bitmap bm = new Bitmap ("d: \picture.bmp");
Graphics g = Graphics.FromImage ( bm);
```

После создания объекта типа Graphics его можно использовать для вывода линий, геометрических фигур, текста и изображений. Основными объектами, которые при этом применяются, служат:

Pen – рисование линий и контуров геометрических фигур

Brush – заполнение областей,

Font – вывод текста,

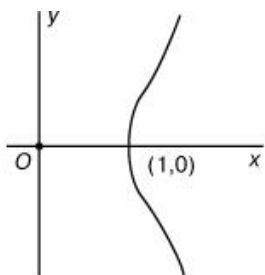
Color – выбор цвета.

В примере приводится текст программы для вывода на форме линии, эллипса и текста.

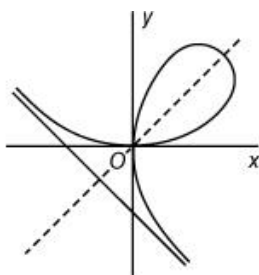
```
public partial class Form1 : Form
{
    public Form1( ) { InitializeComponent();}
    private void Form1_Paint ( object sender, PaintEventArgs e )
    using ( Graphics g = e.Graphics)
    {
        using ( Pen pen = new Pen (Color.Red) )
        {
            g.DrawLine (pen, 0, 0, 200,100);
            g.DrawEllipse (pen, new Rectangle(50, 50, 100, 150) );
        }
        string s = "Sample Text";
        Font font = new Font( "Arial", 18);
        float x =100.0;
        float y =20.0;
        g.DrawString (s, font, brush, x, y);
        font.Dispose( );
        brush.Dispose( );
    }
}
```

Графические объекты потребляют системные ресурсы, поэтому для них рекомендуется использовать метод `Dispose` – освобождение ресурсов.

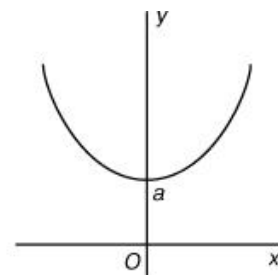
В вариантах заданий 1-18 наравне с графиками простейших функций (линии, параболы, гиперболы, окружности) обязательно продемонстрировать построение трансцендентных кривых в параметрической форме (в полярных координатах), например:



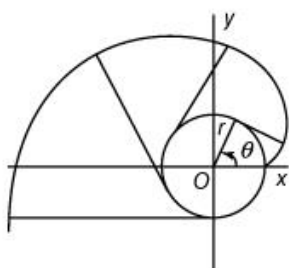
**1. Кривая с изолированной точкой.** Уравнение:  $y^2 = x^3(x - 1)$ .



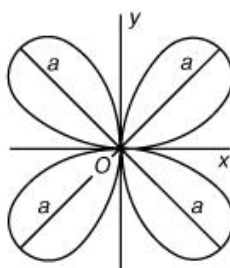
**2. Декартов лист.** Уравнение:  $x^3 + y^3 = 3axy$ .



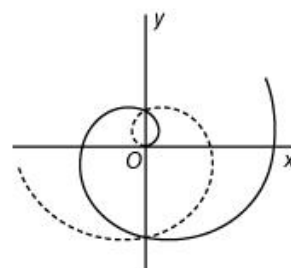
**3. Цепная линия.** Уравнение:  $y = \frac{a}{2}(e^{x/a} + e^{-x/a})$ .



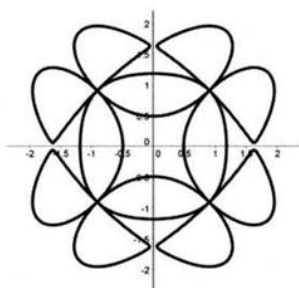
**4. Инволюта окружности.** Уравнения:  $x = r \cos \theta + r \theta \sin \theta$ ,  $y = r \sin \theta - r \theta \cos \theta$ .



**5. Четырехлепестковая роза.** Уравнение:  $r = a \sin 2 \theta$ .

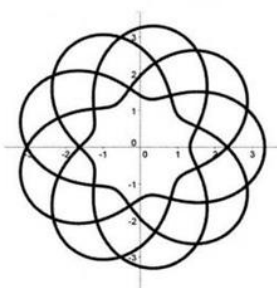


**6. Архимедова спираль.** Уравнение:  $r = a \theta$ .



Симметричная роза с четырьмя лепестками. Параметрические уравнения:

$$\begin{cases} x = \pm \sqrt{3 \cos 2t \cdot \cos t + 0,82} \\ y = \pm \sqrt{3 \cos 2t \cdot \sin t + 0,82} \end{cases}$$

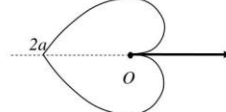


Узорчатая кривая. Уравнение в полярных координатах:

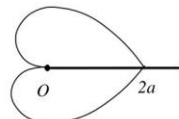
$$r = \cos \frac{9\theta}{4} + \frac{7}{3}$$

В полярной системе координат

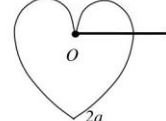
Кардиоиды:  $\rho = a(1 - \cos \varphi)$



$$\rho = a(1 + \cos \varphi)$$



$$\rho = a(1 - \sin \varphi)$$



$$\rho = a(1 + \sin \varphi)$$

В вариантах заданий 19-28 использование трансцендентных кривых не обязательно, но графический объект должен обязательно состоять из нескольких графических примитивов (кругов, прямоугольников, линий и прочего)

## **Варианты**

### **Варианты 1 – 6**

Создать Windows-приложение на языке программирования C#, которое выполняет построение нескольких различных графиков одновременно в одной системе координат (на одном поле) различными цветами.

Для реализации интерфейса пользователя создать меню с командами: Выбор линий, Построение графиков, Очистка поля, Информация о графике, Выход из программы.

Команда «Выбор линий» открывает диалоговое окно, содержащее:

- количество линий для построения (от 1 до 5);
- список для выбора цвета графика типа (компонент ListBox);
- список для выбора типа графика (компонент ListBox), содержащий виды кривых в аналитической форме;
- выбор диапазона построения графика (от и до по осям x и y).

Остальные команды реализуются произвольно.

**Варианты 7 – 12.** Написать Windows-приложение на языке программирования C#, которое выполняет построение четырех различных графиков трансцендентных кривых в параметрической форме на одной форме одновременно, причем каждый из графиков строится в своей системе координат (в собственной области) собственным цветом.

Для реализации интерфейса пользователя создать меню с командами: Параметры кривых, Построение графиков, Очистка полей, Информация о графике, Выход.

Команда «Параметры кривых» открывает диалоговое окно, в котором:

- список для выбора цвета графика типа ListBox;
- выбор толщины линии;
- выбор параметров функции (случайным образом или из диапазона значений);
- выбор наличия координатной сетки (да/нет).

Остальные команды реализуются произвольно.

**Варианты 13 – 18.** Написать Windows-приложение на языке программирования C#, которое выполняет построение на форме одного графика из предлагаемого набора функций. Обязательными в наборе являются: прямая, парабола, гипербола, эллипс, а также не менее двух трансцендентных кривых в параметрической форме.

Для реализации пользовательского интерфейса создать меню с командами: Выбор функции, Выбор вида графика, Построение графика, Очистка поля, Информация о графике, Выход.

Команда «Выбор функции» открывает диалоговое окно, содержащее:

- список кривых для построения графиков (в аналитической записи) типа ListBox;
- выбор значений параметров уравнений (случайным образом и из диапазона).

Команда «Выбор вида графика»

- выбор диапазона координат построения графика;
- выбор наличия координатной сетки;
- цвет координатных линий;
- цвет графика;
- толщина линии.

Остальные команды реализуются произвольно.

**Варианты 19 – 23.** Написать Windows-приложение на языке программирования C#, которое выполняет анимацию некоторого графического объекта.

Для реализации пользовательского интерфейса создать меню с командами: Выбор параметров, Показать изображение, Анимация, Стоп, Выход.

Команда «Выбор параметров» открывает диалоговое окно, содержащее:

- список для выбора цвета изображения;
- группу Direction из двух переключателей типа RadioButton для выбора направления движения (сверху вниз, слева направо);
- размер графического объекта (маленький, большой).

При выборе команды «Показать изображение» в центре формы рисуется объект, состоящий из нескольких графических закрашенных примитивов (кругов, прямоугольников, линий и прочего), например, человечек, самолет или цветок.

По команде «Анимация» объект начинает движение в выбранном направлении до края формы и обратно.

По команде Стоп движение останавливается.

Команда Выход завершает работу приложения.

### **Варианты 24 - 28.**

Написать Windows-приложение на языке программирования C#, которое выполняет анимацию графического объекта.

Создать меню с командами: Выбор параметров, Показать изображение, Анимация, Стоп, Выход.

Команда «Выбор параметров» открывает диалоговое окно, содержащее:

- поле типа TextBox с меткой Speed для ввода скорости движения объекта;
- группу Direction из двух переключателей (по часовой стрелке, против часовой стрелки) типа RadioButton для выбора направления движения.

При выборе команды «Показать изображение» в центре экрана появляется изображение графического объекта, состоящий из нескольких графических не закрашенных примитивов (кругов, прямоугольников, линий и прочего), например, человечек, самолет или цветок.

По команде «Анимация» объект начинает движение в выбранном направлении;

По команде «Стоп» движение останавливается.

Команда «Выход завершает работу приложения».