**Лабораторная работа №1**

**ПОСТРОЕНИЕ ГРАФИКОВ ФУНКЦИЙ**

**Упражнение**. Создать приложение, отображающее на форме график функции y= ‑x2+3.

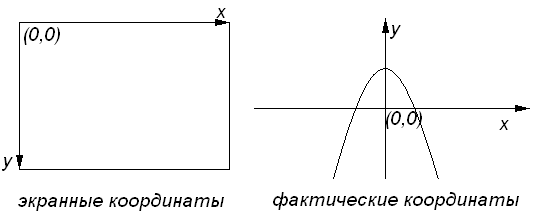
**Решение**

Создайте новый проект, назовите его Ex1, установите его стартовым.

Измените значения свойств формы следующим образом:

|  |  |
| --- | --- |
| Свойство | Значение |
| *Name* | *MainForm* |
| *Text* | График |
| *Windowstate* | *Maximized* |
| *BackColor* | *Window* |

При построении на экране любого графика используются как фактические координаты графика функции в декартовой системе координат для расчета значений функции, так и экранные координаты для отображения графика на форме (рис. 1). Отметим, что начало координат экранной системы координат находится в левом верхнем углу, ось абсцисс направлена вправо, а ось ординат – вниз.



*Рис.  1.* Экранные и фактические координаты

Для преобразования фактических координат в экранные воспользуемся следующими формулами:

хэ=х0+хф×k,

уэ=у0–уф×k,

где координаты (х0,у0) задают положение начала координат на форме, (хф, уф) – это фактические координаты в декартовой системе координат, (хэ, уэ) – экранные координаты, k – коэффициент масштабирования.

При отображении графика на форме будем использовать метод DrawLine() класса Graphics. В этом случае на форме будет отображаться отрезок прямой, соединяющий две соседние точки графика. Опишите и проинициализируйте объект Graph класса Graphics, и объект MyPen класса Pen, а также освободите занимаемые ими ресурсы в обработчике события FormClosing. При этом:

описаны в классе формы:

public partial class Form1 : Form

{

Graphics Graph;

Pen MyPen;

int x,y;

записано в конструкторе формы;

public partial class Form1 : Form

{

Graph = CreateGraphics();

MyPen = new Pen(Color.Magenta);

InitializeComponent();

указано в обработчике FormClosing:

MyPen.Dispose();

Graph.Dispose();

Итак, для отображения графика функции нам потребуются следующие переменные (опишите их в обработчике события Paint):

целочисленные переменные х0 и у0, определяющие положение начала координат на форме. Для нашего примера зададим начало координат в середине формы (свойство ClientSize определяет размер клиентской области элемента управления – в нашем случае для формы размер без учета заголовка и границ):

int x0 = this.ClientSize.Width / 2;

int y0 = this.ClientSize.Height / 2;

целочисленные переменные х1, у1 и х2, у2 – экранные координаты концов отрезка графика:

int x1, y1, x2, y2;

фактические координаты текущей точки графика функции:

double x, y;

Будем отображать график функции в заданном диапазоне изменения аргумента [xMin, xMax] с заданным приращением аргумента step. Введем соответствующие константные значения:

const double xMin = -5;

const double xMax = 5;

const double step = 0.01;

Также опишем константу для определения коэффициента масштабирования:

const double k = 5.5;

Для отображения графика функции вставьте в обработчик события Paint следующие операторы:

//фактические координаты в начальной точке заданного диапазона

x = xMin;

y = - x \* x + 3;

//соответствующие им экранные координаты

x1 = (int)(x0 + x \* k);

y1 = (int)(y0 - y \* k);

while (x < xMax)

{

//определение фактических координат графика в следующей точке

x = x + step;

y = - x \* x + 3;

//соответствующие им экранные координаты

x2 = (int)(x0 + x \* k);

y2 = (int)(y0 - y \* k);

//вывод отрезка графика на экран

Graph.DrawLine(MyPen, x1, y1, x2, y2);

//запоминаем текущие координаты

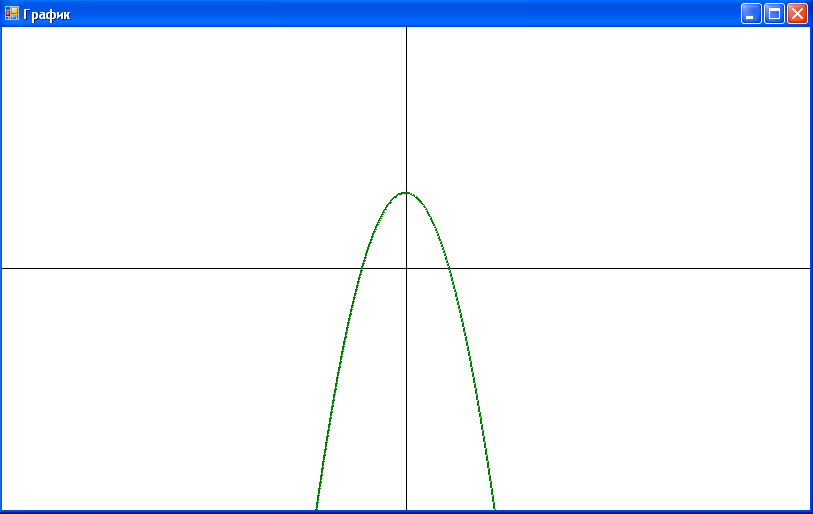
x1 = x2;

y1 = y2;

}

Отметим, что каждый раз отрезок графика рисуется от предыдущей сохраненной точки графика (*x1*, *y1*) до текущей (*x2*, *y2*).

Сохраните изменения, внесенные в проект. Запустите приложение, убедитесь в правильности его работы. Для большей наглядности, самостоятельно добавьте отображение осей координат (рис. 2).



*Рис. 2.* Приложение «График»

***Задание 1. Создать визуальный проект и построить график функции, используя точечный метод. Выбрать произвольно цвет фона, цвет изображения. Построить разметки осей координат***

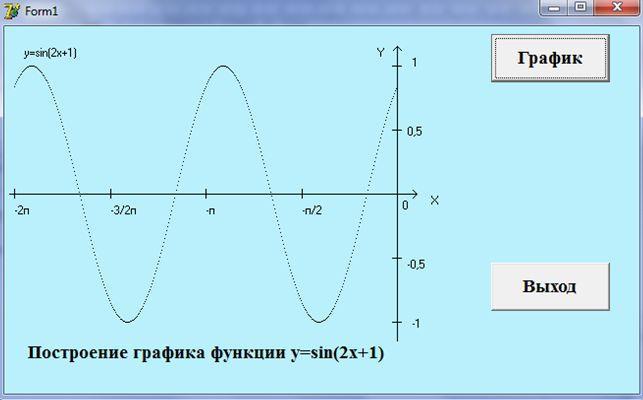
Решение этой задачи удобно проводить в следующем порядке:

1. Определим границы значений аргумента в декартовых координатах, в пределах которых будет строиться график X [Xmin, Xmax].
2. Для данной области значений аргумента определим предельные значения функции: Y[Ymin, Ymax]. Эти значения не обязательно должны быть точными. Они могут быть оценочными снизу и сверху соответственно.
3. Зададим границы графического окна в графических координатах, в пределах которого будет рисоваться график: [Xgmin, Xgmax] – по горизонтали, [Ygmin, Ygmax] – по вертикали.
4. Учесть, что Ygmin>Ygmax, поскольку в графических координатах вертикальная ось направлена вниз.

**Варианты заданий**

1. *y=1/3cos(2x)* на отрезке[-2π, 2π].
2. *y=2sin(2x)* на отрезке[0, 3π].
3. *y=sin(3x+1)* на отрезке[-π,π].
4. *y=3sin(x)* на отрезке[-2π, 0].
5. *y=sin(1/2x)* на отрезке[-2π, 2π].
6. *y=cos(2x+1)* на отрезке[-2π, 0].
7. *y=1/2cos(x)* на отрезке[-π,π].
8. *y=1/2cos(2x+1)* на отрезке[-2π, 2π].
9. *y=3cos(x-1)* на отрезке[-2π, 2π].
10. *y=1+cos(x+1/2)* на отрезке[-2π, 2π].
11. *y=1-cos(x)* на отрезке[-π, 0].
12. *y=1+2cos(2x)* на отрезке[0, 3π].
13. *y=1-3sin(x)* на отрезке[-2π, 2π].
14. *y=2+sin(x)* на отрезке[-π,π].
15. *y=1-sin(1/2x)* на отрезке[0, 3π].
16. *y=1+-sin(x-1)* на отрезке[0, 3π].
17. *y=1-sin(x+1)* на отрезке[-2π, 2π].
18. *y=1-sin(2x)* на отрезке[0, 4π].
19. *y=1-2sin(1/2x)* на отрезке[-π, 4π].
20. *y=1-3sin(1/2x)* на отрезке[-π, 5π].

***Результат работы программы***

****