Практическая работа № 16, ПОСТРОЕНИЕ ГРАФИКА ФУНКЦИИ НА ОТРЕЗКЕ

Постановка задачи

Построить график функции на заданном отрезке. Для построения графика используется вся доступная область формы. При изменении размеров окна график выводится заново с учетом новых размеров.

В программе необходимо предусмотреть:

1. контроль вводимой информации;
2. масштабирование графика функции произвести по оси X по заданному отрезку, а по оси Y – по максимальному и минимальному значению функции на концах отрезка;
3. подписи осей на графике функции.

Ограничение:

1. формула функции задается непосредственно в программе.

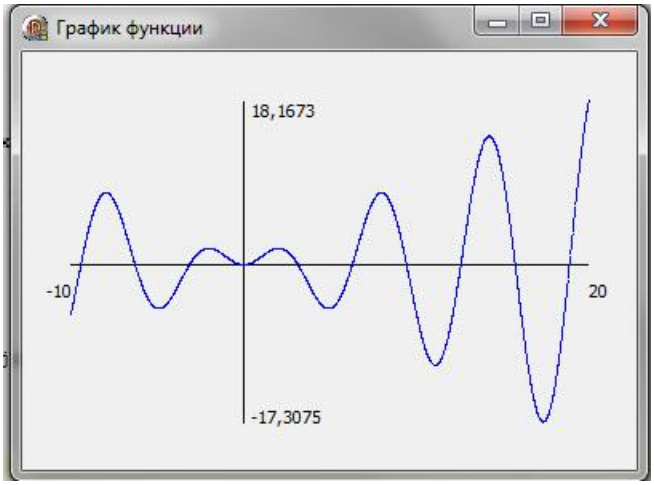

Рис.1

Новым в этой работе являются:

- построение графика функции с учетом масштабирования по X и по Y.

Немного теории

См. Практическая работа №9, Генератор случайных чисел

Карандаш и кисть

Методы, обеспечивающие вычерчивание на поверхности холста графических примитивов, используют:

- *карандаш* (**Pen**) – для вычерчивания линий и контуров,
- *кисть* (**Brush**) – для закрашивания областей, ограниченных контурами.

Карандаш и *кисть* представляют собой объекты типов **TPen** и **TBrush**.

Тип объекта	
TPen	**TBrush**
Color – цвет линии (контура)	**Color** – цвет закрашивания замкнутой области
Style – вид линии	**Style** – стиль заполнения области
Width – толщина линии в пикселях	

Вывод текста

Для вывода текста на поверхность графического объекта используется метод **TextOut**.

Инструкция вызова этого метода в общем виде:

```
Объект.Canvas.TextOut(x,y,Текст);
```

Объект – имя объекта, на поверхность которого выводится текст,

x, y – координаты точки вывода текста;

Текст – переменная или константа символьного типа (выводимый текст).

Методы вычерчивания графических примитивов

`Объект.Canvas.LineTo(x,y);`	Прямая линия от текущей позиции карандаша в точку с указанными координатами
`Объект.Canvas.MoveTo(x,y);`	Перемещение карандаша в указанную точку
`Объект.Canvas.Rectangle(x1,y1,x2,y2);`	Вычерчивание прямоугольника
`Объект.Canvas.Ellipse(x1,y1,x2,y2);`	Вычерчивание эллипса

Свойства объекта Canvas

1. Поверхности, на которую программа может осуществлять вывод графики, соответствует свойство **Canvas**. Используя свойство **Pixels** объекта **Canvas**, можно задать требуемый цвет для любой точки графической поверхности.

```
Form1.Canvas.Pixels[x,y]:= <цвет>;
```

2. Размер графической поверхности формы определяется значениями свойств

`ClientWidth` – ширина

`ClientHeight` – высота

3. Левая верхняя точка рабочей области – `Pixels[0,0]`
4. Правая нижняя точка рабочей области –

```
Pixels[ClientWidth-1,ClientHeight-1]
```

План разработки программы

Рассмотрим на примере $f(x) = x \times \sin x$ на отрезке [–10; 20].

Этапы создания программы

1. Разработка процедуры **Graph_Line**, которая

- вычисляет максимальное значение (Y2) и минимальное (Y1) значения функции на отрезке [X1, X2],
- вычисляет масштаб по осям X (MX) и Y (MY).

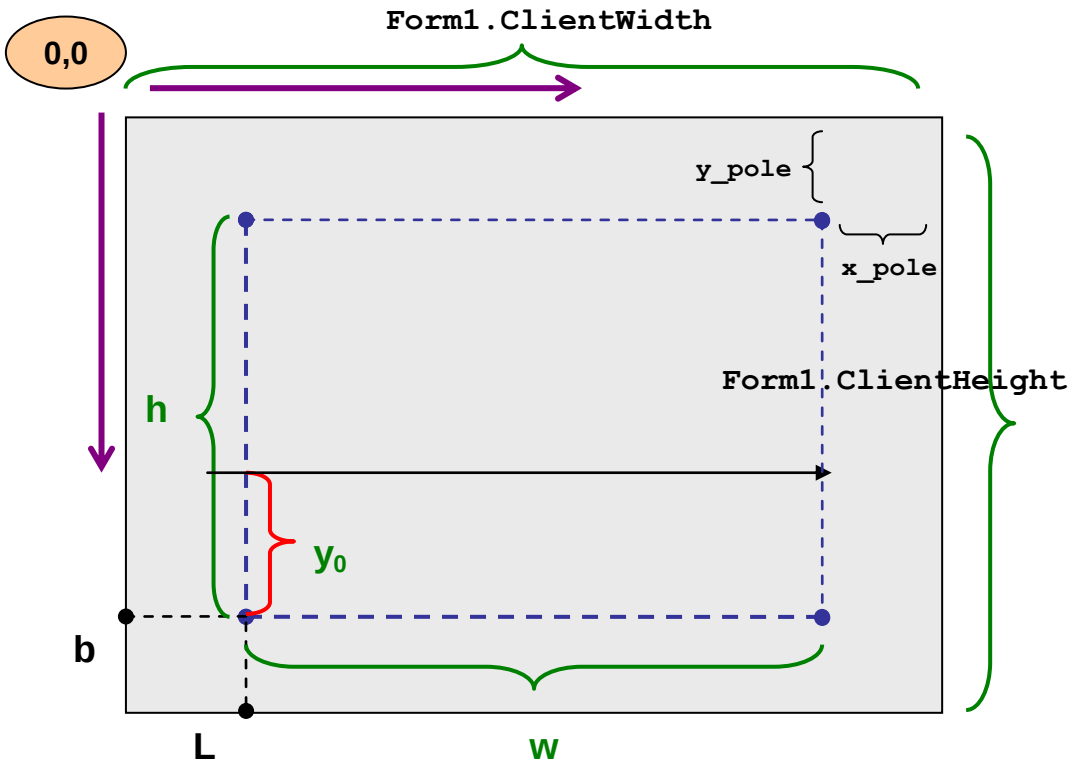

Рис.2

Пояснения к Рисунку 2

Все поле формы, на котором будем строить график функции, закрашено серым цветом и определяется размерами формы **Form1.ClientWidth** и **Form1.ClientHeight.**

Левый верхний угол соответствует точке с координатами (0, 0), при этом по оси X координата увеличивается слева направо, и по оси Y – сверху вниз.

Рабочее поле, относительно которого будет произведено масштабирование и на котором будет размещен график функции. На Рис.2 данная область выделена синей пунктирной линией.

y_pole и **x_pole** определяют пустые поля на форме.

h, **w** – ширина и высота поля, на котором будет размещен график функции.

Y0 – определяет расстояние от нижнего края рабочей области до оси Y.

2. Процедура построения графика функции помещается в программный модуль в раздел **implementation** (раздел описания процедур и функций модуля) после директивы {$R *.dfm}

```
procedure Graph_Line;    {Построение графика функции}

const x_pole = 30;
      y_pole = 30;
var
    x1,x2:real;         {границы изменения аргумента функции}
    y_min,y_max:real;   {границы изменения значения функции}
    x,xg : real;       {аргумент функции}
    y    : real;       {значение функции в точке x}
    dx   : real;       {приращение аргумента }
    l,b  : integer;     {левый нижний угол (x,y) области вывода графика}
    w,h  : integer;     {ширина и высота области вывода графика}
    mx,my: real;       {масштаб по оси x и y}
    x0,y0: integer;     {точка начала координат}

function f (x:real):real;
{Функция графика}
begin
        f:=sin(x)*x;
end;

begin
{Область вывода графика}
x1:= -10;           {нижняя граница диапазона аргумента}
x2:= 20;            {верхняя граница диапазона аргумента}
dx:= 0.01;          {шаг аргумента}
l:= x_pole;         {координата левого нижнего угла по x}
b:= Form1.ClientHeight-y_pole;{координата левого нижнего угла по y}
h:= Form1.ClientHeight-2*y_pole;    {высота  области вывода}
w:= Form1.ClientWidth -2*x_pole;    {ширина области вывода}

{Поиск минимума и максимума функции на отрезке [x1, x2]}
y_min := f(x1);
y_max := f(x1);
x := x1;

repeat
        y := f(x);
        if y < y_min then y_min := y;
        if y > y_max then y_max := y;
        x := x + dx;
until (x>=x2);

my := h/abs(y_max-y_min);   {масштаб по y}
mx := w/abs(x2-x1);         {масштаб по x}

x0 := l;
y0 := b-abs(Round(y_min*my));
```

```
 with Form1.Canvas do
   begin
     If x1*x2 < 0 Then          { вывод оси y}
      begin MoveTo(l+Round(abs(x1)*mx),b);
            LineTo(l+Round(abs(x1)*mx),b-h);
        end;
     If y_min*y_max < 0 Then   { вывод оси x}
      begin MoveTo(x0,y0);
            LineTo(x0+w,y0); end;

    TextOut(x0-15,y0+10,FloatToStrF(x1,ffGeneral,6,3));{вывод x1}
    TextOut(x0+w,y0+10,FloatToStrF(x2,ffgeneral,6,3)); {вывод x2}
   end;

{ Построение графика}
     x := x1;
     xg:=0;
     repeat
      y := f(x);
   Form1.Canvas.Pixels[x0+Round(xg*mx),y0-Round(y*my)]:=clBlue;
      x := x + dx;
      xg := xg + dx;
     until (x>=x2);
end;
```

7. Создайте следующие процедуры обработки событий:

Выделенный объект	Имя события	Действие
Form1	**OnPaint**	`Graph_Line;` **Комментарий** Эта процедура обеспечивает вычерчивание графика после появления формы на экране в результате запуска программы.

	OnResize	`with Form1 do`
		`Canvas.FillRect(Rect(0,0,ClientWidth,ClientHeight));`
		`Graph_Line;`
		Комментарий
		Для вычерчивания графика после изменения размера формы используется процедура **TForm1.FormResize**. Она обрабатывает событие **OnResize**, которое происходит при изменении размера формы.
		Данная процедура сначала очищает поверхность формы, а затем вызывает процедуру **Graph_Line** для построения графика функции.

Задание для самостоятельного выполнения

1. Дополните программу возможностью задавать размер формы и заголовок при создании формы.

2. Внесите изменения в программу, чтобы на экран выдавались максимальное и минимальное значения функции на заданном отрезке рядом с осью Y.

3. Внесите изменения в программу, чтобы была возможность ввести значения X1, X2 и dx с клавиатуры, предусмотрев контроль вводимой информации.

4. Внесите изменения в программу, чтобы была возможность выбора для построения из списка одной функции.