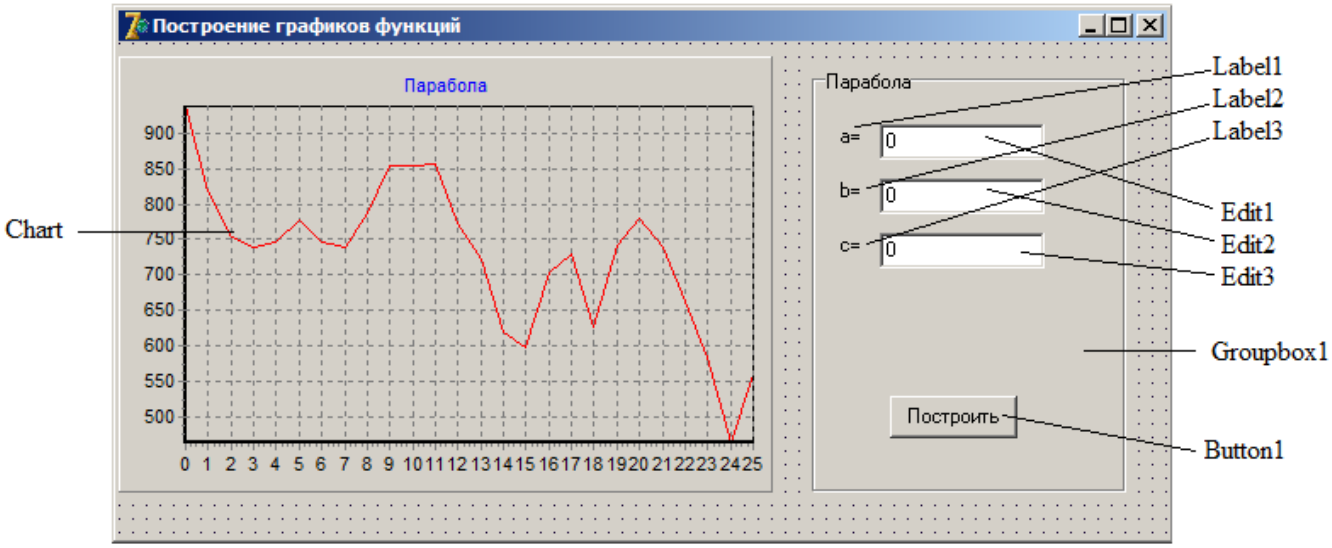
Практическая работа 6
Разработка приложений с отображением графической информации

Задание 1. Создайте приложение, которое выводит график функции $y=ax^2+bx+c$.

Решение:

1. Создайте папку «График функции».
2. Создайте новый проект в Delphi и сохраните его в папке «График функции».
3. Разместите на форме объекты по предложенному рисунку.



4. На форме поместите компонент **Chart**. Компонент Chart находится на странице Additional.
5. Двойным щелчком мыши на компоненте Chart запустите редактор диаграмм. На вкладке Chart выберите **Series** и щелкните на кнопке **Add**. В открывшемся окне выберите тип **Line**. Щелкните на кнопке ОК и закройте окно выбора типа графика или диаграммы.
6. На вкладке Chart перейдите к вкладке **3D** и снимите флажок с пункта **3 Dimension** для того, чтобы убрать трёхмерное отображение графика. Перейдите к вкладке **Legend** и снимите флажок с пункта **Visible**. Перейдите к вкладке **Titles** и в текстовом поле наберите следующее: «Парабола».
7. Для события Onclick компонента Button1 создайте обработчик:

```
procedure TForm1.Button1Click(Sender: TObject);
var
 x: integer;
 a,b,c:real;
begin
  begin
  a:=strtofloat(edit1.Text);
  b:=strtofloat(edit2.Text);
  c:=strtofloat(edit3.Text);
  Series1.Clear;
  for x:=(-50) to 50 do //диапазон изменения аргумента x
  Series1.AddXY(x, a*sqr(x)+b*x+c, inttostr(x), clblue);
  end;
end;
```

8. Кнопкой Close закройте редактор диаграмм.
9. Откомпилируйте проект. Запустите и протестируете созданное приложение.

Задание 2. Моделирование бегущих волн

Бегущими волнами называются волны, которые переносят в пространстве энергию. Уравнение бегущей волны следующее

$$\xi(x,t)=A\cos(\omega\cdot t-\kappa\cdot x+\phi_0)$$,

где:

A- амплитуда волны (в метрах), ω – циклическая частота волны (в радианах в секунду), x – координата, k – волновое число, φ0 – начальная фаза волны для простоты возьмем φ0=0.

2.1. Используя компонент Chart1 смоделируем бегущие волны.

Создайте новый проект. Поместите компонент Chart1 на форму. Также добавьте на форму компоненты, указанные в таблице, со следующими значениями их свойств, указанными в этой же таблице:

Компонент	Значение свойства Caption	Описание в контексте программы
Label1	W1	Частота первой волны
Label2	dT	Приращение времени
Label3	Ym1	Амплитуда первой волны
Label4	W2	Частота второй волны
Label5	Ym2	Амплитуда второй волны
Edit1	0,01	Значение частоты первой волны
Edit2	1	Значение приращения времени
Edit3	1	Значение амплитуды первой волны
Edit4	-0,05	Значение частоты второй волны
Edit5	3	Значение амплитуды второй волны

Компонент	Свойство	Значение свойства
checkbox1	Checked	False
	Caption	Запуск волны
checkbox2	Checked	False
	Caption	3D
Timer1	Interval	10
	Enabled	False
Button1	Caption	Установить

Разместите Label1 напротив Edit1, Label2 напротив Edit2, Label3 напротив Edit3, Label4 напротив Edit4, а также Label5 напротив Edit5. Разместите все эти компоненты так, как показано на рисунке. В компоненте Chart1 создайте графики Series1, Series2 и Series3.



Задача создаваемого приложения состоит в следующем: сначала устанавливаем значения всех параметров (амплитуды, частоты), затем щелкаем на кнопке «Установить», затем ставим галочку на «Запуск волны» и моделируемые волны должны отображаться в компоненте Chart1.

Создайте глобальные переменные Ym1,W1,Ym2,W2:double; t, Dt:word.

Проинициализируете их начальными значениями в обработчике события onCreate формы, для этого напишите там следующее:

```
t:=0; {начальное значение времени}
Ym1:=StrToFloat(Edit3.Text);
W1:=StrToFloat(Edit1.Text);
Ym2:=StrToFloat(Edit5.Text);
W2:=StrToFloat(Edit4.Text);
Dt:=StrToInt(Edit2.Text)
```

То же самое напишите в обработчике нажатия кнопки Button1, либо, (Важно!!!), чтобы не дублировать дважды один и тот же текст для Button1 на вкладке Events, для события onClick, щелкните справа на раскрывающемся списке и настройте это событие на обработчик FormCreate (путем простого выбора из списка). Аналогичные операции можно осуществлять с любыми событиями, настраивая события на любые другие обработчики, уже прописанные ранее, для других объектов.

В обработчике события onClick щелчка на CheckBox1 напишите следующее:

```
if CheckBox1.Checked then timer1.Enabled:=true else
timer1.Enabled:=not true;
```

Таким образом, если галочка установлена, то активируем таймер, в котором уже прописано моделирование волны, иначе останавливаем таймер.

Теперь непосредственно выведем бегущую волну в объект Chart1, для этого запрограммируем обработчик таймера следующим образом:

```
var x:integer; Y1:double;
begin
Series1.Clear;
t:=t+dt;
for x:=-360 to 360 do begin
  Y1:=Ym1*cos(x*pi/180+w1*t);
  Series1.AddXY(X,Y1,'');
end;
end;
```

Запустите программу. Установите галочку напротив «Запуск волны». Вы получите смоделированную бегущую волну. Меняйте значения параметров, ставьте как положительные, так и отрицательные значения частоты, а затем гажимайте кнопку «Установить». Изучайте, как меняется при этом скорость и направление волны. Чтобы зафиксировать масштаб, щелкните по компоненту Chart1 правой кнопкой мыши. Чтобы остановить движение волны, снова снимите галочку напротив «Запуск волны».

Закройте программу и вернитесь в режим программирования.

2.2. Добавим вторую волну, в обработчик этого же таймера.

Указание. Для расчетов будем использовать уже готовый цикл for, имеющийся в обработчике, то есть для второй волны не нужно еще раз прописывать второй for.

Немного измените предыдущий обработчик таймера следующим образом:

```
procedure TForm1.Timer1Timer(Sender: TObject);
var x:integer; Y1, Y2:double;
begin
Series1.Clear; Series2.Clear;
t:=t+dt;
```

```
for x:=-360 to 360 do begin
  Y1:=Ym1*cos(x*pi/180+w1*t);  {первая волна}
  Series1.AddXY(X,Y1,'');
  Y2:=Ym2*cos(x*pi/180+w2*t);  {вторая волна}
  Series2.AddXY(X,Y2,'');
end;
end;
```

Запустите программу и установите галочку на «Запуск волны». Вы получите две движущиеся бегущие волны.

2.3. Учитывая, что эти 2 волны складываются в пространстве (интерферируют), построим третью результирующую волну в этом же таймере.

У нас график Series3 будет соответствовать результирующей волне, то есть будет соответствовать интерференции двух волн, отображаемых графиками Series1 и Series2.

Смысл результирующей волны в следующем: к волнам применяется принцип суперпозиции волн – при распространении волн каждая из них распространяется так, как будто другие волны отсутствуют, а результирующее смещение частиц среды в любой момент времени равно геометрической сумме смещений, которые получают частицы, участвуя в каждом из слагающих волновых процессов.

Снова измените обработчик таймера следующим образом:

```
procedure TForm1.Timer1Timer(Sender: TObject);
var x:integer; Y1, Y2, Y3:double;
begin
Series1.Clear; Series2.Clear; Series3.Clear;
t:=t+dt;
for x:=-360 to 360 do begin
  Y1:=Ym1*cos(x*pi/180+w1*t);  {первая волна}
  Series1.AddXY(X,Y1,'');
  Y2:=Ym2*cos(x*pi/180+w2*t);  {вторая волна}
  Series2.AddXY(X,Y2,'');
  Y3:=Y1+Y2;
  Series3.AddXY(X,Y3,'');    {результирующая волна}
end;
end;
```

Запустите программу. Убедитесь, что теперь на графике отображается результирующая волна №3, равная сумме двух других. Мы проанализируем поведение этой третьей, результирующей волны чуть ниже. Закройте программу и вернитесь в режим программирования.

2.4. Запрограммируем компонент Checkbox2 следующим образом. Если на нем выбрана галочка, то график должен быть объемным, а иначе обыкновенным.

В обработчике события onClick щелчка на CheckBox2 напишите следующее:

```
if CheckBox2.Checked=true then Chart1.View3D:=true
else Chart1.View3D:=false.
```

Здесь мы используете булевское свойство View3D компонента chart1, отвечающее за объемность графика. Запустите программу. Установите обе амплитуды Ym1 и Ym2 двух складывающихся волн в одинаковое значение, например в 3, а их частоты W1 и W2 установите одинаковыми по модулю, но разными по знаку, например 0,05 и -0,05 соответственно. Установите галочку напротив «3D», чтобы график отображался в объемном виде и поставьте галочку напротив «Запуск волны».

2.5. Изменим параметры отображения объемного графика, с помощью компонента TrackBar.

Добавьте компонент TrackBar, расположенный на вкладке Win32 палитры компонентов, служит для установки конкретного целого значения, путем перемещения его указателя, при этом данное целое значение содержится в его свойстве Position.

Компонент	Свойство	Значение свойства
TrackBar1	Position	1
	min	1
	max	500

Запрограммируем так, чтобы при перемещении указателя TrackBar1 размер графика менялся от 1 до 500%. Для этого в обработчике изменения позиции указателя TrackBar1

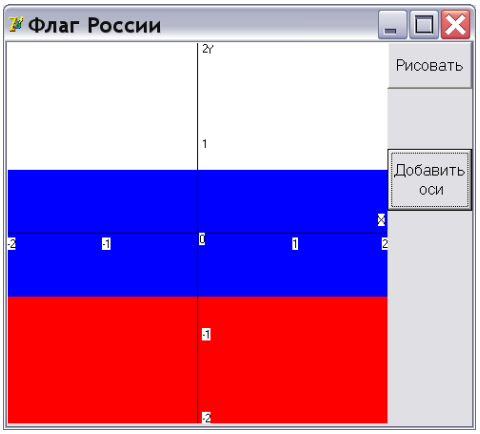
```
    procedure TForm1.TrackBar1Change(Sender: TObject) напишите
следующую простую инструкцию:
    chart1.View3DOptions.Zoom:=trackbar1.Position;
```

Снова запустите программу, установив объемный вид графика. Убедитесь в правильности ее работы. Закройте программу и вернитесь в режим программирования.

Задание 3. Нарисуем российский флаг в компоненте Image1

Покажем на данном примере, как работать с отдельным пикселом компонента Image1.

Создайте новый проект. Сбросьте на форму компонент Image1 и две кнопки Button.



У компонента Image1 установите значения его обоих свойств Height и Width равными 400. То есть компонент Image1 будет иметь число пикселей, равное 400x400. Заголовок Caption кнопки Button1 установите в значение «Рисовать», а заголовок Button2 – в «Добавить оси». При щелчке на Button1 будет нарисован флаг России, а при щелчке на Button2 к нему добавятся координатные оси в центре. Это сделано для того, чтобы показать, что компонент Image1, в принципе, годится и для построения обыкновенных графиков или диаграмм.

Создайте обработчик щелчка на Button1, который должен выглядеть следующим образом:

```
procedure TForm1.Button1Click(Sender: TObject);
var i,j,k,x,y:integer;
begin
{Рисуем флаг России 400x400}
for y:=1 to 400 do begin {цикл по всем y}
case y of
{цикл по всем x, первые 133 строчки закрашиваем белым цветом}
1..133: begin for x:=1 to 400 do
      Image1.Canvas.Pixels[x,y]:=clWhite;  end;
{цикл по всем x, вторые 133 строчки закрашиваем синим цветом}
```

```
    134..266: begin for x:=1 to 400 do
          Image1.Canvas.Pixels[x,y]:=clBlue;  end;
    {цикл по всем x, третьи 133 строчки закрашиваем красным
цветом}
    267..400: begin for x:=1 to 400 do
          Image1.Canvas.Pixels[x,y]:=clRed;  end;
    end;
    end;
```

Нужно сказать, что очень многие графические операции выполняются в контексте специального свойства Canvas, которое содержится во многих компонентах. Например, у объекта Form1 также имеется свойство Canvas.

Canvas является полноценным объектом, имеющим свои методы (то есть функции и процедуры), ориентированные на работу с графикой, рисованием на форме и всем, что связано с графическим интерфейсом Windows. В данном случае мы используем свойство Pixels[X, Y: Integer] объекта Canvas для доступа к цвету отдельного пикселя по его координатам относительно компонента Image1.

А теперь создайте обработчик щелчка на Button2, который должен выглядеть так:

```
    procedure TForm1.Button2Click(Sender: TObject);
    begin
    Image1.Canvas.MoveTo(1,200); Image1.Canvas.LineTo(400,200);
{Рисуем Ось Y}
    Image1.Canvas.TextOut(200+10,1,'Y');  {Подписываем Y}
    Image1.Canvas.MoveTo(200,1); Image1.Canvas.LineTo(200,400);
{Рисуем Ось X}
    Image1.Canvas.TextOut(400-10,200-20,'X'); {Подписываем X}
    with Image1 do  begin
    Canvas.TextOut(width div 2+2,Height div 2,IntToStr(0 ));
    Canvas.TextOut(width div 2-100,Height div 2+5,IntToStr(-1));
    Canvas.TextOut(width div 2-200,Height div 2+5,IntToStr(-2));
    Canvas.TextOut(width div 2+100,Height div 2+5,IntToStr(1));
    Canvas.TextOut(width div 2+194,Height div 2+5,IntToStr(2));
    Canvas.TextOut(width div 2+5,Height div 2-100,IntToStr(1));
    Canvas.TextOut(width div 2+5,Height div 2-200,IntToStr(2));
    Canvas.TextOut(width div 2+5,Height div 2+100,IntToStr(-1));
    Canvas.TextOut(width div 2+5,Height div 2+188,IntToStr(-2));
    end;
    end;
```

Здесь методом TextOut(X, Y: Integer; const Text: string) мы выводим нужную текстовую информацию, содержащуюся в переменной Text, в координаты X и Y, компонента Image1. Метод MoveTo объекта Canvas описан чуть ниже. Запустите программу. Щелкните сначала на кнопке Button1, в результате чего в компоненте Image1 нарисуется Российский флаг, а затем щелкните на кнопке Button2 и добавятся координатные оси в центре компонента.

Данную заготовку в принципе можно использовать не только для рисунков, но и для построения некоторых диаграмм или графиков.

Задания для самостоятельного решения

На форме создайте дополнительные объекты, которые позволят отобразить графики функций:

1. Линейная: $y=ax+b$;
2. Гипербола: $y=a/x;$
3. Кубическая парабола: $y=ax^2$;
4. Кубический трехчлен: $y=ax^3+bx^2+cx+d$;
5. Тригонометрические:

a) $y=\cos x$
b) $y=\sin x$

Контрольные вопросы

1. Каково назначение визуального компонента Chart?
2. Описать порядок создания приложения для построения и вывода графика функции.