

От редакции. Уважаемые читатели, с этого номера мы начинаем публикацию цикла, состоящего из девяти статей, посвященных начальным навыкам программирования в среде Delphi (программированию различных типов алгоритмов). В основу данного цикла были положены лабораторные работы по курсу программирования на Delphi. Разумеется, в рамках

отдельных статей, тем более, посвященных конкретным темам, невозможно дать ответы на все возможные вопросы, поэтому в затруднительных ситуациях не забывайте обращаться к встроенной в Delphi справочной системе Help, кроме того, в конце каждой статьи приводится список дополнительной литературы.

А.ШАКИРИН,
г.Минск, БАТУ, каф. ВТ.

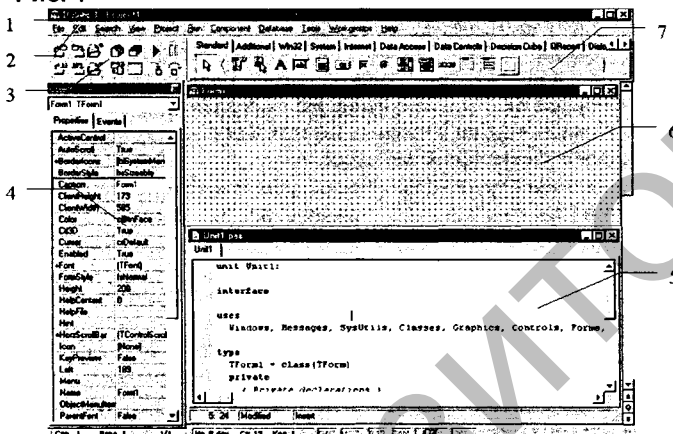
Программирование линейных алгоритмов

Цель этой статьи — изучить основные элементы визуальной среды, освоить использование простейших компонентов Delphi для ввода/вывода данных и создать приложение, которое использует линейный алгоритм.

1. Визуальная среда Delphi

При запуске Delphi на экране появляется панель интерфейса, показанная на рис. 1. Среда Delphi визуально реализуется в виде нескольких одновременно раскрытых на экране монитора окон. Количество, расположение, размер и вид окон могут изменяться программистом.

Рис. 1



Главное окно (1) всегда присутствует на экране и предназначено для управления процессом создания приложения.

Главное меню (2) содержит все необходимые средства для управления проектом.

Пиктограммы главного меню (3) облегчают доступ к наиболее часто применяемым командам.

Окно Инспектора Объектов (4) предназначено для изменения свойств выбранных компонентов и состоит из двух страниц. Страница *Properties* (Свойства) предназначена для изменения необходимых свойств компонента. Страница *Events* (События) — для определения реакции компонента на то или иное событие (например, на щелчок кнопки мыши).

Окно Редактора Кода (5) предназначено для просмотра, создания и редактирования текстов модулей проекта. При первоначальной загрузке в окне Редактора Кода находится текст модуля, содержащий минимальный набор операторов для нормального функционирования пустой Формы в качестве Windows-приложения. При размещении некоторого компонента в окне Формы, текст модуля автоматически дополняется необходимыми операторами.

Окно Формы (6) представляет собой интерфейс проектируемого Windows-приложения. В это окно на этапе проектиро-

вания приложения помещаются необходимые компоненты, которые разработчик берет из Палитры Компонентов. Каждой Форме проекта соответствует модуль (*Unit*), текст которого на языке Object Pascal размещается в окне Редактора Кода.

Палитра Компонентов (7) обеспечивает доступ к набору библиотечных программ среды Delphi, которые описывают некоторый элемент (компонент), помещенный программистом в окно Формы. Каждый компонент имеет определенный набор свойств, которые программист может выбирать и изменять по своему усмотрению, например, заголовок окна, надпись на кнопке, размер, цвет и тип шрифта и др. Свойства компонентов приведены в Help.


Обо всех происходящих в системе событиях, таких как создание Формы, нажатие кнопки мыши или клавиатуры и т.д., ядро Windows информирует окна путем отправки соответствующих сообщений. Среда Delphi принимает и обрабатывает сообщения с помощью обработчиков событий (например, щелчок кнопки мыши — событие *OnClick*, создание Формы — *OnCreate*). Наиболее часто применяемые события представлены в табл. 1.

Табл. 1

Событие	Описание события
<i>OnActivate</i>	Возникает при активизации Формы
<i>OnCreate</i>	Возникает при создании Формы. В обработчике данного события следует задавать действия, которые должны происходить в момент создания Формы (например, установка начальных значений)
<i>OnClick</i>	Возникает при нажатии кнопки мыши в области компонента
<i>OnDblClick</i>	Возникает при двойном нажатии кнопки мыши в области компонента
<i>OnKeyPress</i>	Возникает при нажатии клавиши на клавиатуре. Параметр <i>Key</i> имеет тип <i>Char</i> и содержит ASCII-код нажатой клавиши (клавиша <i>Enter</i> клавиатуры имеет код #13, клавиша <i>Esc</i> — #27 и т.д.). Обычно это событие используется в том случае, когда необходима реакция на нажатие одной из клавиш
<i>OnKeyDown</i>	Возникает при нажатии клавиши на клавиатуре. Обработчик этого события получает информацию о нажатой клавише и состоянии клавиш <i>Shift</i> , <i>Alt</i> и <i>Ctrl</i> , а также о нажатой кнопке мыши

Для создания обработчика события программисту необходимо раскрыть список компонентов в верхней части окна Инспектора Объектов и выбрать необходимый компонент. Затем на странице *Events* Инспектора Объектов нажатием левой клавиши мыши выбрать название обработчика и дважды щелкнуть по его правой (белой) части. В ответ Delphi активизирует окно Редактора Кода модуля и покажет заготовку процедуры обработки выбранного события. Для каждого обрабатываемого события в тексте модуля организуется процедура (*procedure*), между ключевыми словами *begin* и *end* которой программист на языке Object Pascal записывает требуемый алгоритм обработки события.



Переключение между окном Формы и окном Редактора Кода осуществляется кнопкой главного меню с пиктограммой  или клавишей F12.

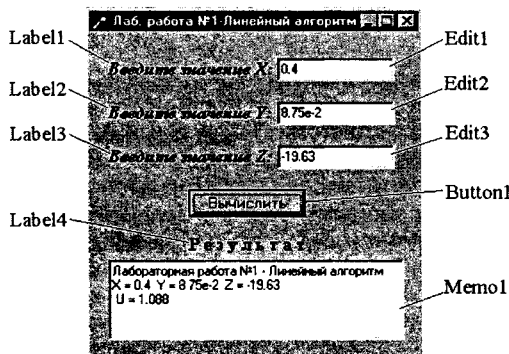
2. Пример создания приложения

Необходимо создать Windows-приложение для вычисления выражения

$$u = tg^5(\sqrt{x} - y^3) + e^{y/z} \cdot \sin z^2.$$

Численные значения данных x , y и z вводятся с клавиатуры в соответствующие поля панели интерфейса. Один из возможных вариантов панели интерфейса создаваемого приложения показан на рис.2.

Рис. 2




2.1. Сохранение проекта

В процессе проектирования приложения Delphi создает несколько файлов — *проект*. Каждый проект целесообразно хранить в отдельной, заранее созданной папке. С помощью подходящего приложения Windows создадим папку и назовем ее, например, LAB1.

Для сохранения проекта откройте в главном меню пункт *File* и щелкните мышью на опции *Save Project As...* (*Сохранить проект как...*). Сначала Delphi откроет панель диалога *Save Unit1 As (Сохранить модуль как)* для сохранения модуля проекта. В этой панели найдем созданную папку LAB1 и сохраним в ней модуль под именем, например, UnLinAlg. Обратите внимание на то, что Delphi по умолчанию присвоит этому файлу тип Delphi unit с расширением *.pas. Затем откроется панель диалога *Save Project1 As*. Назовем наш проект, например, PrLinAlg и сохраним его в этой же папке. Здесь Delphi даст файлу тип Delphi project и расширение *.dpr. Убедитесь в том, что главное окно Delphi теперь называется PrLinAlg, окно главного файла проекта — PrLinAlg.dpr, а окно модуля проекта — UnLinAlg.pas.





Старайтесь давать файлам осмысленные имена вместо однообразных Unit1 и Project1, предлагаемых Delphi.

Чтобы избежать потери файлов проекта в аварийных ситуациях, связанных, например, с выключением питания, зависанием системы и т.д., рекомендуется периодически сохранять проект, используя пиктограмму  главного меню или опцию *Save All* в меню *File*.

2.2. Настройка окон

Чтобы работать с окном, необходимо сделать его активным, щелкнув мышью в любом месте окна. У активного окна заголовок становится выделенным, например, на рис.1 активным является окно Редактора Кода.

Окна Формы и Редактора Кода модуля в правом верхнем углу имеют кнопки управления, которые предназначены:

-  — для свертывания окна в пиктограмму;
-  — для разворачивания окна на весь экран и возвращения к исходному размеру;
-  и  — для закрытия окна.

С помощью мыши, захватывая одну из кромок окна или выделенную строку заголовка, отрегулируйте нужные разме-

ры окон Формы, Редактора Кода, Инспектора Объектов и их положение на экране.


2.3. Изменение заголовка Формы

Новая Форма имеет одинаковые имя (Name) и заголовок (Caption) — FORM1. Начинаям программистам имя Формы менять не рекомендуется, т.к. оно используется в тексте модуля.


Для изменения заголовка активизируйте окно Инспектора Объектов и на странице Properties в свойстве Caption замените заголовок Form1 на Лаб. работа №1 – Линейный алгоритм. Убедитесь, что одновременно изменился заголовок окна Формы.


2.4. Размещение компонентов на Форме

Будем размещать компоненты на Форме так, чтобы они соответствовали панели, показанной на рис.2.

Для нанесения надписей на Форму используется компонент Label. Выберите в Палитре Компонентов на странице Standard пиктограмму  компонента Label и щелкните на ней мышью. Затем в нужном месте Формы щелкните мышью — появится надпись Label1. В свойстве Caption Инспектора Объектов замените надпись Label1 на *Введите значение X:*. В свойстве Font подберите шрифт. Аналогично нанесите на Форму остальные надписи. Щелкнув мышью на любом из размещенных компонентов, отрегулируйте его местоположение на Форме и размер.

Для ввода/вывода данных в простейших случаях используются компоненты Edit и Memo. Компонент Edit применяется в тех случаях, когда данные представляются одной строкой. Если данные представляют собой несколько строк, то используется компонент Memo.


Для создания полей ввода численных значений переменных x , y и z используем компонент Edit. Выберите в Палитре Компонентов на странице Standard пиктограмму  и разместите компонент Edit в нужных местах Формы — так же как вы это делали с компонентом Label.

Для вывода результатов используем компонент Memo. Выберите в Палитре Компонентов на странице Standard пиктограмму , поместите компонент Memo на Форму и откорректируйте его местоположение и размеры.

2.5. Написание процедуры обработки события создания Формы (FormCreate)

Если программист желает, чтобы при появлении панели интерфейса на экране в соответствующих полях находились начальные значения данных, он должен учесть, что при запуске приложения возникает событие — создание Формы (OnCreate). Создадим процедуру обработки этого события, которая занесет начальные значения переменных x , y , z в поля Edit1, Edit2 и Edit3 соответственно, а в поле Memo1 поместит строку *Лабораторная работа №1 – Линейный алгоритм*. Для этого дважды щелкните мышью на любом свободном месте Формы. На экране появится текст модуля UnLinAlg, в котором Delphi автоматически создает заготовку процедуры-обработчика события создания Формы: *Procedure TForm1.FormCreate(Sender:TObject)*. Между операторами *begin* и *end* этой процедуры вставьте операторы, которые выполняют необходимые действия (текст модуля приведен в разделе 2.7).

2.6. Написание процедуры обработки события нажатия кнопки Button1 (Button1Click)

Поместим на Форму кнопку, нажатие которой приведет к вычислению выражения. Выберите в Палитре Компонентов на странице Standard пиктограмму  компонента Button. В свойстве Caption Инспектора Объектов замените надпись на кнопке Button1 на *Вычислить*. В свойстве Font подберите шрифт. Отрегулируйте положение и размер кнопки. Затем дважды щелкните мышью на кнопке, после чего курсор установится в тексте процедуры-обработчика события нажатия кнопки Button1: *Procedure TForm1.Button1Click(Sender:TObject)*.



Внимательно наберите операторы этой процедуры, используя текст модуля UnLinAlg.

2.7. Текст модуля UnLinAlg

Unit UnLinAlg;

interface

uses
Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls;

type

```
TForm1 = class(TForm)
  Label1: TLabel;
  Edit1: TEdit;
  Label2: TLabel;
  Edit2: TEdit;
  Label3: TLabel;
  Edit3: TEdit;
  Label4: TLabel;
  Memo1: TMemo;
  Button1: TButton;
  procedure FormCreate(Sender: TObject);
  procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

```
Form1: TForm1;
```

implementation

```
($R *.DFM)
// Процедура обработки события создания Формы:
procedure TForm1.FormCreate(Sender: TObject);
begin
  Edit1.Text:='0.4'; // начальное значение X
  Edit2.Text:='8.75e-2'; // начальное значение Y
  Edit3.Text:='-19.63'; // начальное значение Z
  Memo1.Clear; // очистка Memo1
// Вывод строки в Memo1:
Memo1.Lines.Add('Лабораторная работа #1 - Лнейный алгоритм');
end;
// Процедура обработки события нажатия кнопки Button1:
procedure TForm1.Button1Click(Sender: TObject);
var
  x,y,z,a,b,c,u : extended; // объявление локальных переменных
begin
  x:=StrToFloat(Edit1.Text); // X присваивается содержимое Edit1
  y:=StrToFloat(Edit2.Text); // Y присваивается содержимое Edit2
  z:=StrToFloat(Edit3.Text); // Z присваивается содержимое Edit3
// Вычисляется выражение
  a:=sqrt(x)-y*y*y;
  b:=sin(a)/cos(a);
  c:=Exp(5*Ln(b));
  u:=c+exp(y/z)*sin(z*z);
Memo1.Lines.Add('X = '+Edit1.Text+' Y = '+Edit2.Text+'
  Z = '+Edit3.Text); // контрольный вывод X, Y, Z в Memo1
// Вывод результата в Memo1:
Memo1.Lines.Add('U = '+FloatToStrF(u,ffFixed,8,3));
end;
end.
```

Данные, с которыми работают компоненты Edit и Memo, имеют тип String. Поэтому в процедуре TForm1.Button1Click при присваивании содержимого полей Edit1, Edit2, Edit3 переменным X, Y, Z с помощью функции StrToFloat осуществляется преобразование данных типа String в действительные значения с плавающей точкой типа Extended. Если необходимо работать с данными целого типа, используется функция StrToInt.

При выводе данных в Memo1 используется метод Add свойства Lines, причем для преобразования данных из действи-

тельного значения в строковое и управления формой представления выводимого результата используется функция FloatToStrF.

2.8. Работа с приложением

Для запуска созданного приложения нажмите пиктограмму главного меню или клавишу F9. При этом происходит компиляция модулей и, если нет ошибок, компоновка проекта и создание выполняемого файла PrLinAlg.exe. На экране появляется панель интерфейса приложения (рис.2).

Щелкните мышью на кнопке "Вычислить", и в поле Memo1 появится результат. Измените исходные значения x, y, z в полях Edit и снова нажмите кнопку "Вычислить". Убедитесь, что в поле Memo1 отображаются новые результаты. Завершить работу приложения можно нажатием кнопки в правом верхнем углу панели интерфейса.

В случае нештатного функционирования приложения, восстановить первоначальный режим работы с проектом можно путем выбора в меню Run опции ProgramReset или нажав клавиши Ctrl+F2.

3. Индивидуальные задания

Необходимо создать Windows-приложение для вычисления соответствующего выражения. В соответствии с условием задания, количеством и типами исходных данных следует оформить дизайн панели интерфейса проектируемого приложения, установить необходимое количество полей Edit, текстов заголовков на Форме, размеры шрифтов, а также типы переменных и функции преобразования при вводе и выводе результатов.

$$1. \quad t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{0,5 + \sin^2 y} \left(1 + \frac{z^2}{3 - \frac{z^2}{5}} \right).$$

При x=14,26, y=-1,22, z=3,5×10⁻², t=0,564849.

$$2. \quad u = \frac{\sqrt[3]{8 + |x - y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x - y|} (tg^2 z + 1)^x.$$

При x=-4,5, y=0,75×10⁻⁴, z=0,845×10⁻², u=-55,6848.

$$3. \quad v = \frac{1 + \sin^2(x + y)}{\left| x - \frac{2y}{1 + x^2 y^2} \right|} x^{|y|} + \cos^2\left(\arctg \frac{1}{z}\right).$$

При x=3,74×10⁻², y=-0,825, z=0,16×10⁻², v=1,0553.

$$4. \quad \omega = |\cos x - \cos y|^{(1 + 2 \sin^2 y)} \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4} \right).$$

При x=0,4×10⁻², y=-0,875, z=-0,475×10⁻³, ω=1,9873.

$$5. \quad \alpha = \ln\left(y^{-\sqrt{|x|}}\right) \left(x - \frac{y}{2}\right) + \sin^2 \arctg(z).$$

При x=-15,246, y=4,642×10⁻², z=20,001×10⁻², α=-182,036.

Литература

1. В.В.Феофанов. Delphi 3. Учебный курс. — М.: Нолидж, 1981.
2. Э.Возневич. Delphi. Освой самостоятельно. — М.: Восточная книжная компания, 1996.
3. Дж.Матчо, Д.Р.Фолкнер. Delphi. — М.: БИНОМ, 1995.
4. М.Канту. Delphi 2 для Windows 95/NT. — М.: ООО "Малип", 1997.