

Глава 1. Основы программирования

Программа

Программа, работающая на компьютере, нередко отождествляется с самим компьютером, т. к. человек, использующий программу, "вводит в компьютер" исходные данные, как правило, при помощи клавиатуры, а компьютер "выдает результат" на экран, на принтер или в файл. На самом деле, преобразование исходных данных в результат выполняет процессор компьютера. Процессор преобразует исходные данные в результат по определенному алгоритму, который, будучи записан на специальном языке, называется программой. Таким образом, чтобы компьютер выполнил некоторую работу, необходимо разработать последовательность команд, обеспечивающую выполнение этой работы, или, как говорят, написать программу.

Этапы разработки программы

Выражение "написать программу" отражает только один из этапов создания компьютерной программы, когда разработчик программы (программист) действительно пишет команды (инструкции) на бумаге или при помощи текстового редактора.

Программирование — это процесс создания (разработки) программы, который может быть представлен последовательностью следующих шагов:

1. Спецификация (определение, формулирование требований к программе).
2. Разработка алгоритма.
3. Кодирование (запись алгоритма на языке программирования).
4. Отладка.
5. Тестирование.
6. Создание справочной системы.
7. Создание установочного диска (CD-ROM).

Спецификация

Спецификация, определение требований к программе — один из важнейших этапов, на котором подробно описывается исходная информация, формулируются требования к результату, поведение программы в особых случаях (например, при вводе неверных данных),

разрабатываются диалоговые окна, обеспечивающие взаимодействие пользователя и программы.

Разработка алгоритма

На этапе разработки алгоритма необходимо определить последовательность действий, которые надо выполнить для получения результата. Если задача может быть решена несколькими способами и, следовательно, возможны различные варианты алгоритма решения, то программист, используя некоторый критерий, например, скорость решения алгоритма, выбирает наиболее подходящее решение. Результатом этапа разработки алгоритма является подробное словесное описание алгоритма или его блок-схема

Кодирование

После того как определены требования к программе и составлен алгоритм решения, алгоритм записывается на выбранном языке программирования. В результате получается исходная программа.

Отладка

Отладка — это процесс поиска и устранения ошибок. Ошибки в программе разделяют на две группы: синтаксические (ошибки в тексте) и алгоритмические. Синтаксические ошибки — наиболее легко устранимые. Алгоритмические ошибки обнаружить труднее. Этап отладки можно считать законченным, если программа правильно работает на одном-двух наборах входных данных.

Тестирование

Этап тестирования особенно важен, если вы предполагаете, что вашей программой будут пользоваться другие. На этом этапе следует проверить, как ведет себя программа на как можно большем количестве входных наборов данных, в том числе и на заведомо неверных.

Создание справочной системы

Если разработчик предполагает, что программой будут пользоваться другие, то он обязательно должен создать справочную систему и обеспечить пользователю удобный доступ к справочной информации во время работы с программой. В современных программах справочная информация представляется в форме CHM- или HLP-файлов. Помимо справочной информации, доступ к которой осуществляется из программы во время ее работы, в состав справочной системы включают инструкцию по установке (инсталляции) программы, которую оформляют в виде Readme-файла в одном из форматов: TXT, DOC или HTML.

Процесс создания справочной системы и механизмы доступа к справочной информации описаны в гл. 14.

Создание установочного диска

Установочный диск или CD-ROM создаются для того, чтобы пользователь мог самостоятельно, без помощи разработчика, установить программу на свой компьютер. Обычно помимо самой программы на установочном диске находятся файлы справочной информации и инструкция по установке программы (Readme-файл). Следует понимать, что современные программы, в том числе разработанные в Delphi, в большинстве случаев (за исключением самых простых программ) не могут быть установлены на компьютер пользователя путем простого копирования, так как для своей работы требуют специальных библиотек и компонентов, которых может и не быть у конкретного пользователя. Поэтому установку программы на компьютер пользователя должна выполнять специальная программа, которая помещается на установочный диск. Как правило, установочная программа создает отдельную папку для устанавливаемой программы, копирует в нее необходимые файлы и, если надо, выполняет настройку операционной системы путем внесения дополнений и изменений в реестр.

Процесс создания установочного диска (CD-ROM) при помощи входящей в состав Delphi утилиты InstallShield Express описан в гл. 18.

Алгоритм и программа

На первом этапе создания программы программист должен определить последовательность действий, которые необходимо выполнить, чтобы решить поставленную задачу, т. е. разработать алгоритм. Алгоритм — это точное предписание, определяющее процесс перехода от исходных данных к результату.

Алгоритм решения задачи может быть представлен в виде словесного описания или графически — в виде блок-схемы. При изображении алгоритма в виде блок-схемы используются специальные символы (рис. 1.1).

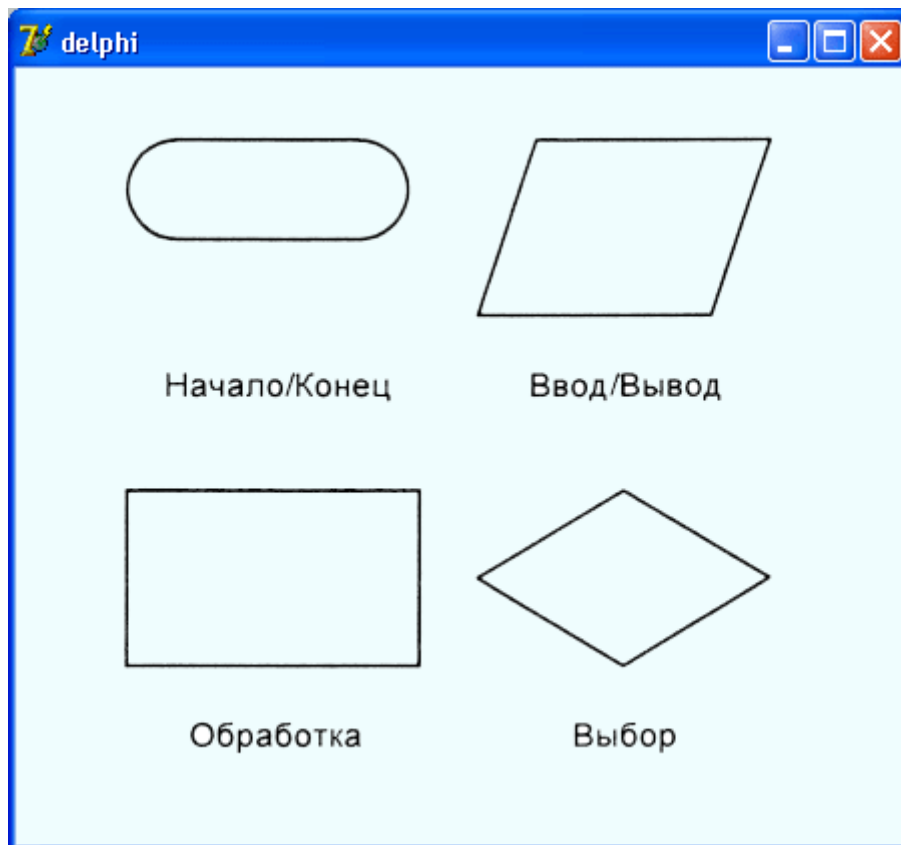


Рис. 1.1. Основные символы, используемые для представления алгоритма в виде блок-схемы

Представление алгоритма в виде блок-схемы позволяет программисту уяснить последовательность действий, которые должны быть выполнены для решения задачи, убедиться в правильности понимания поставленной задачи.

При программировании в Delphi алгоритм решения задачи представляет собой совокупность алгоритмов процедур обработки событий.

В качестве примера на рис. 1.2 приведена совокупность алгоритмов программы **Стоимость покупки**, а на рис. 1.3 — ее диалоговое окно. После разработки диалогового окна и алгоритмов обработки событий можно приступить к написанию программы. Ее текст приведен в листинге 1.1.

Листинг 1.1. Программа **Стоимость покупки**

```
unit покупка_1;
```

```
interface uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls;
```

```
type
```

```
TForm1 = class(TForm)
Edit1: TEdit;
Edit2: TEdit;
Label1: TLabel;
Label2: TLabel;
Button1: TButton;
Label3: TLabel;
procedure Button1Click(Sender: TObject);
procedure Edit2KeyPress(Sender: TObject;
var Key: Char);
procedure Edit1KeyPress(Sender: TObject;
var Key: Char); private
{ Private declarations } public
{ Public declarations }
end;
var
Form1: TForm1;
implementation
{$R *.dfm}
// подпрограмма
procedure Summa;
var
цена: real; // цена
кол: integer; // количество
s: real; // сумма
```

```

mes: string[255]; // сообщение

begin

cena := StrToFloat(Form1.Edit1.Text);

kol := StrToInt(Form1.Edit2.Text);

s := cena * kol;

if s > 500 then

begin

s := s * 0.9;

mes := 'Предоставляется скидка 10%' + #13;

end;

mes := mes+ 'Стоимость покупки: '
+ FloatToStrF(s,ffFixed,4,2) +' руб.';

Form1.Label3.Caption := mes;

end;

// щелчок на кнопке Стоимость

procedure TForm1.Button1Click(Sender: TObject);

begin

Summa; // вычислить сумму покупки

end;

// нажатие клавиши в поле Количество

procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);

begin

case Key of

'0' .. '9',#8: ; // цифры и клавиша <Backspace>

#13: Summa; // вычислить стоимость покупки

```

```
else Key := Chr(0); // символ не отображать
end;
end;
// нажатие клавиши в поле Цена
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
case Key of
'0' .. '9', #8 : ; // цифры и клавиша <Backspace>
#13: Form1.Edit2.SetFocus; // клавиша <Enter>
',' , ':' :
begin
if Key = ':'
then Key := ',' if Pos(',', Edit1.Text) <> 0
then Key := Chr(0);
end;
else // все остальные символы запрещены
Key := Chr(0);
end;
end;
end.
```

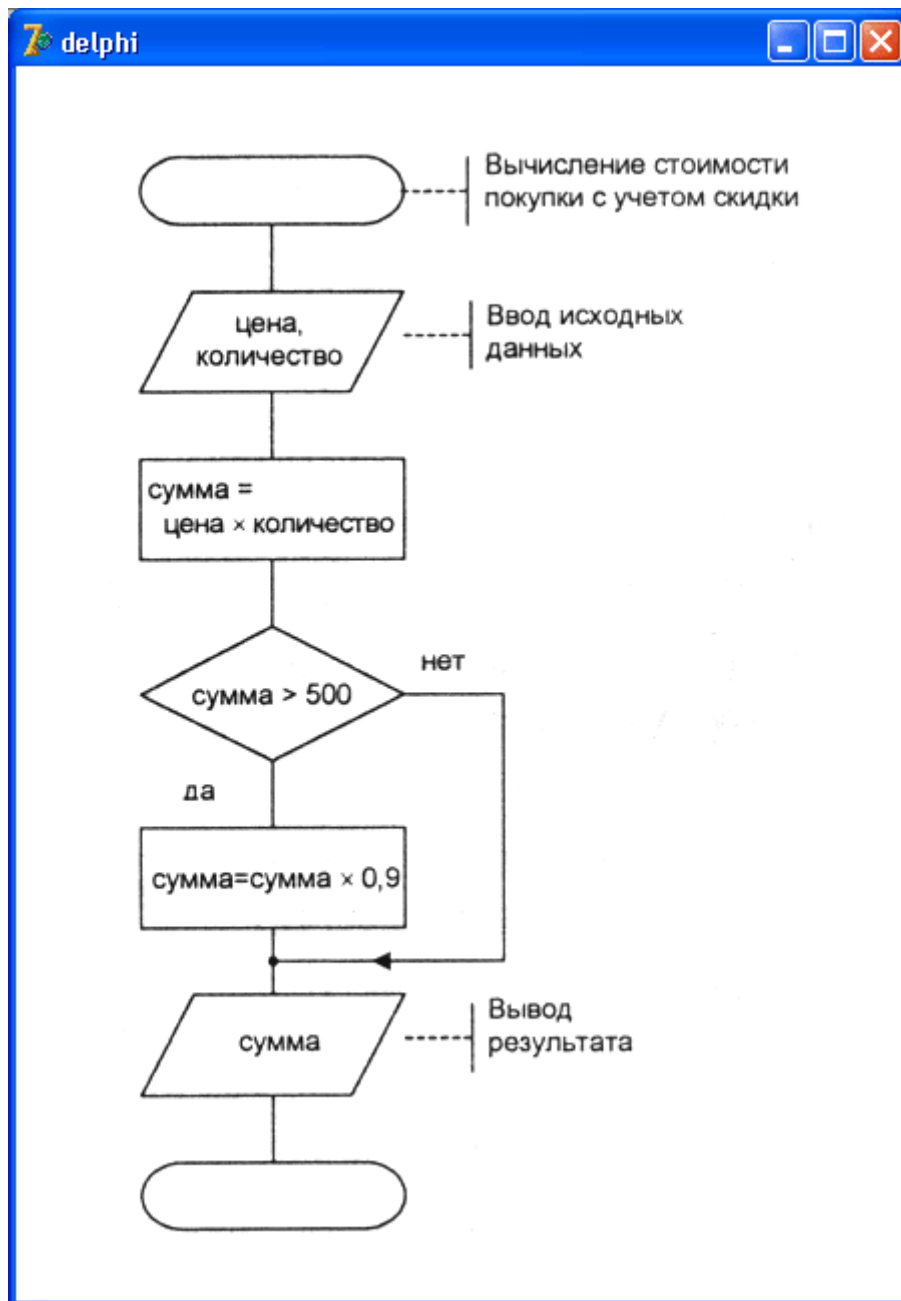


Рис. 1.2. Алгоритм программы вычисления стоимости покупки — совокупность алгоритмов обработки событий на компонентах формы

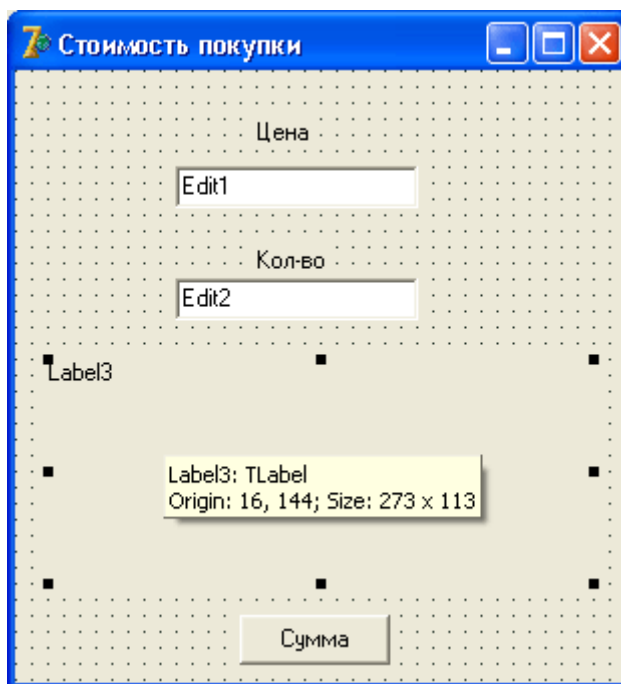


Рис. 1.3. Окно (форма) программы **Стоимость покупки**

Компиляция

Программа, представленная в виде инструкций языка программирования, называется исходной программой. Она состоит из инструкций, понятных человеку, но не понятных процессору компьютера. Чтобы процессор смог выполнить работу в соответствии с инструкциями исходной программы, исходная программа должна быть переведена на машинный язык — язык команд процессора. Задачу преобразования исходной программы в машинный код выполняет специальная программа — компилятор.

Компилятор, схема работы которого приведена на рис. 1.4, выполняет последовательно две задачи:

1. Проверяет текст исходной программы на отсутствие синтаксических ошибок.
2. Создает (генерирует) исполняемую программу — машинный код.

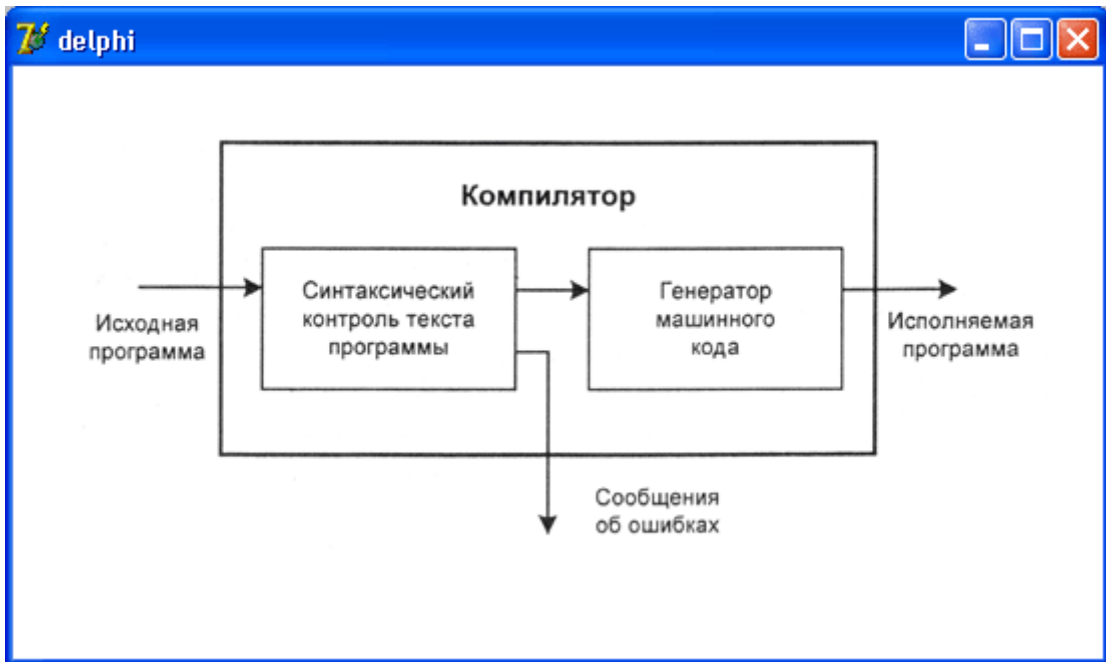


Рис. 1.4. Схема работы компилятора

Следует отметить, что генерация исполняемой программы происходит только в том случае, если в тексте исходной программы нет синтаксических ошибок.

Генерация машинного кода компилятором свидетельствует лишь о том, что в тексте программы нет синтаксических ошибок. Убедиться, что программа работает правильно можно только в процессе ее тестирования — пробных запусках программы и анализе полученных результатов. Например, если в программе вычисления корней квадратного уравнения допущена ошибка в выражении (формуле) вычисления дискриминанта, то, даже если это выражение будет синтаксически верно, программа выдаст неверные значения корней.