

# 6 ОТЛАДКА ПРОГРАММ



Написать действующую сложную программу с первого раза очень непросто. Конечно, среда Delphi поможет в исправлении синтаксических ошибок, однако допустить неточность или ошибку можно и в формально правильно записанном операторе. В подобных случаях нам поможет режим отладки и пошагового выполнения, в котором программу можно выполнить оператор за оператором, анализируя значения промежуточных переменных. В этом разделе мы займемся отладкой программ, познакомимся с методикой исправления ошибок.

## 6.1

**Использование отладчика программ в визуальной среде программирования. Пошаговое выполнение программ, просмотр значений переменных во время выполнения программы. Разновидности ошибок, методы их поиска и исправления**

Квалификация программиста определяется не его умением писать безошибочные программы (написать сложную программу без ошибок практически невозможно). Мастерство определяется умением разработчика быстро, эффективно и надежно отлаживать и тестировать свое приложение. На этом занятии мы рассмотрим технические возможности отладки приложений в среде Delphi.

Компиляция с последующим выполнением приложения осуществляется посредством команды Run (в основном меню) → Run либо горячей

клавиши F9 (или выбор ). Выполнение будет осуществляться только в случае, если при компиляции не обнаружены ошибки и загрузочный модуль создан. Компиляция без последующего выполнения приложения осуществляется с помощью команды Project → Compile Project или горячих клавиш Ctrl + F9. В результате компиляции могут быть сделаны замечания — Hint, предупреждения — Warning и сообщения об ошибках — Error. Например, в листинге программы, представленной на рис. 6.1, имеется строка с предупреждением о необходимости использовать в качестве параметра цикла FOR локальную переменную и строка с сообщением об ошибке несоответствия типов Integer и Extended.

Если наше приложение откомпилировалось и стало выполняться, это, увы, еще не означает, что оно правильно работает. В нем может быть еще множество ошибок, допущенных во время выполнения приложения. Это могут быть ошибки, постоянно возникающие при выполнении некоторых частей нашей программы, а также проявляющиеся только при каких-то сочетаниях данных: ошибки деления на нуль,

```

Unit1.pas
Unit1
  Variables/Const
  Uses
  TForm1
  Form1: TForm1;
  i: integer;
  implementation
  {$R *.dfm}
  procedure TForm1.Button1Click(Sender: TObject);
  var
    x, y: integer;
  begin
    x:=strtofloat(edit1.Text);
    for i:=1 to 10 do
      y:=y+x;
    end;
  end.
  
```

31: 29 Modified Insert \Code\Diagram/

[Error] Unit1.pas(31): Incompatible types: 'Integer' and 'Extended'  
 [Warning] Unit1.pas(33): For loop control variable must be simple local variable  
 [Fatal Error] Project1.dpr(5): Could not compile used unit 'Unit1.pas'

Рис. 6.1. Листинг программы, содержащей предупреждение об ошибке

переполнения, открытия несуществующего файла и т. п. Наконец, могут быть случайные перемежающиеся ошибки, когда одна и та же задача иногда выполняется нормально, а иногда — нет. Такие ошибки, которые наиболее трудно обнаружить, обычно связаны с отсутствием инициализации переменных в различных режимах работы. В этом случае выполнение приложения зависит от случайного состояния памяти компьютера.

Очень часто для того, чтобы выяснить, в чем причина ошибки, необходимо, выполняя фрагмент программы, отслеживать изменения переменных при выполнении каждой команды. Для пошагового прохода фрагмента программы можно использовать команды, приведенные ниже в табл. 6.1.

Таблица 6.1

Команды пошагового выполнения

Команда	«Горячие» клавиши	Пояснения
Step Over (По шагам без захода в ...)	F8	Пошаговое выполнение строк программы (вызов функции или процедуры считается за одну строку), вход в функции и процедуры не производится
Trace Into (Трассировка с заходом в ...)	F7	Пошаговое выполнение программы с заходом в вызываемые функции и процедуры
Trace to Next Source Line (Трассировка до следующей строки)	Shift + F7	Переход к следующей исполняемой строке