

Метки и оператор перехода.

Современная технология структурного программирования основана на принципе «*программировать без GOTO*»: считается, что злоупотребление операторами перехода *затрудняет* понимание программы, делает её запутанной и сложной в отладке. Тем не менее, в некоторых случаях использование операторов перехода может *значительно упростить* программу.

Оператор	Модуль	Назначение	Пример
GOTO	SYSTEM	Предназначен для <i>принудительного</i> изменения хода выполнения программы. Форма записи: GOTO <метка> метка – метка;	LABEL 1, LOOP; ... 1: ... GOTO 1; ...

Метка – это произвольный идентификатор, позволяющий именовать некоторый оператор программы и таким образом ссылаться на него. Допускается использовать в качестве меток *целые числа* без знака.

Метка располагается непосредственно перед помеченным оператором и отделяется от него *двоеточием*. Оператор можно помечать *несколькими* метками, которые в этом случае отделяются друг от друга двоеточием. Перед тем как появиться в программе, метка должна быть *описана*. Описание меток состоит из зарезервированного слова **LABEL** (*метка*), за которым следует список меток.

Действие оператора **GOTO** состоит в передаче управления соответствующему *меченому* оператору. При использовании меток необходимо руководствоваться следующими правилами:

- метка, на которую ссылается оператор **GOTO**, должна быть описана в разделе описаний и она должна встретиться где-нибудь в теле программы;
- метки, описанные в процедуре (функции), локализируются в ней, поэтому передача управления извне процедуры (функции) на метку внутри её невозможна.

Пример программы.

```

Program Label_Demo;
  Var
    St : String;
    I : Integer;
  Label
    First;
BEGIN
  Write('Введите слово: ');
  Readln(St);
  Writeln('Повторяю слово 10 раз. ');
  I:=0;
First:
  Writeln(St);
  I:=I+1;
  If I<10 Then Goto First;
END.

```

