**Информатика, 2 курс ПД**

**Законспектировать продолжение предыдущей лекции**

*Операторы условного перехода*

Программирование, т.е. составление программы на алгоритмическом языке, состоит в написании последовательности операторов алгоритмического языка, реализующих алгоритм решения задачи.

Обычно операторы в программе выполняются последовательно: один за другим. В языке Бейсик, как и в любом другом языке программирования, имеются такие операторы, которые могут изменять порядок выполнения операторов в программе. В частности, те операторы, которые сообщают компьютеру о том, что управление должно быть передано некоторым другим частям программы, называются операторами передачи управления. Передача управления может быть условной или безусловной, т. е. управление будет передаваться в том случае, если некоторое условие истинно, или независимо от условия.

Условные операторы служат для изменения последовательности выполнения операторов программы в зависимости от некоторого условия. Они позволяют описывать разветвляющийся вычислительный алгоритм типа развилки, который был описан выше. По своей структуре условные операторы подразделяются на строчные и на блочные. Однако любой условный оператор начинается с ключевого слова IF, что в переводе с английского языка означает *если*. Затем записывается условие в виде логического выражения или выражения отношения и потом следует ключевое слове THEN, что в переводе c английского языка означает *тогда*. Далее следуют отличия в использовании блочного или строчного условных операторов, а также учитывается тип разветвляющегося алгоритма. В условных операторах могут использоваться еще два ключевых слова: ELSE, что в переводе c английского языка означает *иначе*, и ENDIF, что в переводе c английского языка означает *конец если*.

Синтаксис условного строчного оператора для полного варианта разветвления, т.е. алгоритм предусматривает одно действие (команду 1) при значении условия TRUE (истина, «Да») и одно действие (команду 2) при значении условия FALSE (ложь, «Нет»):

**IF условие THEN команда 1 ELSE команда 2**

Действие условного оператора состоит в том, что вычисляется значение условия. Если оно равно TRUE, то выполняется команда 1, следующая за ключевым словом THEN, и осуществляется переход к следующему оператору программы. Если оно равно FALSE, то выполняется команда 2, следующая за ключевым словом ELSE, и осуществляется переход к следующему оператору программы.

Синтаксис условного строчного оператора для неполного варианта разветвления («обход»), т.е. алгоритм предусматривает лишь одно действие (команду) при значении условия TRUE (истина, «Да»), а при значении условия FALSE (ложь, «Нет») действий нет:

**IF условие THEN команда**

Действие условного оператора состоит в том, что вычисляется значение условия. Если оно равно TRUE, то выполняется команда, следующая за ключевым словом THEN, и осуществляется переход к следующему оператору программы. Если оно равно FALSE, то при отсутствии ключевого слова ELSE сразу осуществляется переход к следующему оператору программы.

Синтаксис условного блочного оператора для полного варианта разветвления, т.е. алгоритм предусматривает одно или более действий (серию команд 1) при значении условия TRUE (истина, «Да») и одно или более действий (серию команд 2) при значении условия FALSE (ложь, «Нет»):

**IF условие THEN**

**серия команд 1**

**ELSE**

**серия команд 2**

**ENDIF**

Действие условного оператора состоит в том, что вычисляется значение условия. Если оно равно TRUE, то выполняется серия команд 1, которые записаны в следующих строках программы после заголовка условного блочного оператора, закачивающегося ключевым словом THEN. Сигналом для дальнейшего перехода является ключевое слово ELSE. Тогда программа анализирует появление ключевого слова ENDIF и осуществляется выполнение следующего оператора за этим ключевым словом. Если значение условия равно FALSE, то выполняется серия команд 2, которые записаны в следующих строках программы после ключевого слова ELSE до ключевого слова ENDIF, а осуществляется выполнение следующего оператора за этим ключевым словом.

Синтаксис условного блочного оператора для неполного варианта разветвления («обход»), т.е. алгоритм предусматривает лишь одно или более действий (серию команд) при значении условия TRUE (истина, «Да»), а при значении условия FALSE (ложь, «Нет») действий нет:

**IF условие THEN**

**серия команд**

**ENDIF**

Действие условного оператора состоит в том, что вычисляется значение условия. Если оно равно TRUE, то выполняется серия команд, которые записаны в следующих строках программы после заголовка условного блочного оператора, закачивающегося ключевым словом THEN. Не встретив ключевого слова ELSE, программа анализирует появление ключевого слова ENDIF и осуществляется выполнение следующего оператора за этим ключевым словом. Если значение условия равно FALSE, но, не встретив ключевого слова ELSE, а, получив для анализа ключевое слово ENDIF, осуществляется выполнение следующего оператора за этим ключевым словом.

*Оператор безусловного перехода*

Для осуществления переходов в программе используется оператор безусловного перехода. Ключевое слово этого оператора GOTO, что в переводе с английского языка означает: *иди к*.

Синтаксис оператора безусловного перехода:

**GOTO N**

где N – номер строки или метки оператора, на который происходит переход в программе.

Этот оператор служит для перехода из одной строки программы к другой, помеченной номером или меткой.

*Примеры* использования безусловного оператора перехода на языке Бейсик:

|  |  |  |  |
| --- | --- | --- | --- |
| Пример 1. | 10     X=3  20     GOTO 40  30     Y=2\*X  40     Z=5\*X | Пример 2. | X=3  GOTO W1  Y=2\*X  W1    Z=5\*X |

*Программа разветвляющегося алгоритма*

Рассмотрим реализацию разветвляющегося алгоритма на примере задачи, приведенной в разделе 1.3, а именно: вычислить значение функции для заданного x.

**REM вычисление значения функции y(x)**

**CLS**

**INPUT " Введите значения:  X "; X**

**IF X<=10 THEN Y=-5 ELSE Y=X^3**

**? “Y= “; Y**

**END**

В приведенной программе использовался условный строчный оператор, который можно всегда заменить на условный блочный, а именно:

**IF X<=10 THEN**

**Y=-5**

**ELSE**

**Y=X^3**

**ENDIF**

В программе использован оператор комментария Его ключевое слово REM (от английского слова remark – замечать, отмечать). Транслятор, встретив это ключевое слово, игнорирует эту строку программы для анализа синтаксиса и выполнения. Этот оператор служит для внесения пояснений в программу как для ее автора, так и для людей, которые будут ее читать.

Для самостоятельной работы на построение блок-схемы и записи программы разветвляющегося алгоритма можно предложить следующие задачи:

*Задача*1. Даны катеты одного треугольника (А1, В1) и катеты другого треугольника (А2, В2). Определить, будут эти треугольники равновеликими, т.е. имеют они равные площади?

*Задача* 2. Найти общее сопротивление двух проводников R1 и R2, если они соединены параллельно (R3) или последовательно (R4). Значения R1 и R2 вводятся с клавиатуры.