

## Теория для Занятия №20-21.

### Оператор цикла с постусловием REPEAT...UNTIL(цикл ДО)

Применяется для организации циклов с неизвестным заранее числом повторений, логика которых предполагает выполнение цикла до тех пор пока не станет истинным некоторое условие.

**Синтаксис оператора REPEAT...UNTIL:**

**REPEAT**

Оператор1;

Оператор2;

...

ОператорN;

**UNTIL** Условие;

Операторы (Оператор1, Оператор2, ... ОператорN), образующие тело цикла, будут выполняться циклически, до тех пор пока значение “Условия” не станет равно TRUE (истина). Иными словами, в цикле REPEAT...UNTIL условием продолжения итераций является невыполнение “Условия” (его значение FALSE – ложь).

Условие цикла может быть логической константой, переменной или логическим выражением.

Оператор REPEAT...UNTIL имеет две части: начальную (слово REPEAT) и завершающую (слово UNTIL), которые охватывают группу операторов, составляющих тело цикла. Поэтому необходимости в использовании составного оператора для построения тела цикла из нескольких простых операторов не возникает.

Принципиальное отличие оператора REPEAT...UNTIL от оператора WHILE в том, что проверка условия производится не перед началом выполнения тела цикла, а после его выполнения, когда решается вопрос, повторить ли еще раз действия. **Поэтому тело цикла всегда выполняется, по крайней мере, один раз.**

#### Пример 1. Проверка ввода

**VAR**

N : Integer;

**BEGIN**

...

**REPEAT**

Write('Введите целое число от 0 до 10: ');

ReadLn(N);

**UNTIL** (N>=0) and (N<=10);

...

**END.**

В этом примере программа будет запрашивать ввод числа до тех пор, пока пользователь не введет “правильное” число.

#### Пример 2. “Вечный цикл”

**REPEAT UNTIL** False;

Этот цикл пустой и никогда не прекращающийся. Он хорош только в том случае, когда нужно заблокировать программу, и, возможно весь компьютер. Но если

отбросить шутки, то можно и его пристроить в дело. Обычно так организуют программы с повторяющимися действиями: вначале программы ставят REPEAT, а в конце – UNTIL False. А прервать цикл можно специальными операторами EXIT и HALT. Это имеет смысл, если условий завершения программы много или они очень сложны.

### Практическая часть

1. Дано натуральное число  $N$ . Определить является ли оно простым (имеет всего два делителя, 1 и само число).
2. Составить программу нахождения НОД чисел  $a$  и  $b$ , используя алгоритм Евклида.
3. Числа вида  $2^p - 1$ , где  $p$ -простое число, называются числами Мерсенна. Напишите программу, которая определяет, являются ли числа Мерсенна при значениях  $p=2,3,5,7,11,\dots,29,31$  простыми.
4. Совершенным называется число, равное сумме всех своих делителей, меньших, чем оно само. Например:  $6=1+2+3$ ,  $28=1+2+4+7+14$ . Напишите программу, которая проверяет является ли введенное число совершенным. (Евклидом доказано, что каждое число вида  $2^{p-1} \cdot (2^p - 1)$  является совершенным, если  $2^p - 1$  – простое. Л.Эйлер доказал, что все четные совершенные числа находятся по формуле Евклида, а относительно нечетных совершенных чисел ничего неизвестно до сих пор)