Оператор цикла REPEAT.

Является условным оператором цикла с *постпроверкой* условия, т.е. вначале выполняется *оператор*, а потом проверяется *условие*.

Оператор	Модуль	Назначение	Пример

REPEAT	**SYSTEM**	Повторение *заданной* последовательности операторов в зависимости от *условия*.
		Форма записи:
		REPEAT
		<оператор>;
		UNTIL <условие>;
		REPEAT, UNTIL – зарезервированные слова (*повторять до тех пор, пока не будет выполнено условие*);
		<условие> – выражение логического типа;
		<оператор> – исполнительный оператор.
			VAR
			I:INTEGER;
			BEGIN
			...
			I:=0;
			REPEAT
			READLN(I);
			UNTIL I>100;
			...
			END.

<**Оператор**> выполнится хотя бы один раз, после чего проверяется <**условие**>: если его значение есть **FALSE** (условие не выполняется), <**оператор**> повторяется, в противном случае (условие выполняется) оператор **REPEAT – UNTIL** завершает свою работу.

Пара **REPEAT – UNTIL** подобна операторным скобкам **BEGIN – END;** (составной оператор), поэтому перед **UNTIL** ставить точку с запятой необязательно. **REPEAT** воспринимается как **BEGIN**, а **UNTIL** воспринимается как **END;**.

Для более гибкого управления циклическими операторами **FOR**, **WHILE** и **REPEAT** в состав **Turbo Pascal** включены две процедуры:

BREAK	– реализует немедленный выход из цикла; действие процедуры заключается в передаче управления оператору, стоящему сразу за концом циклического оператора;
CONTINUE	– обеспечивает досрочное завершение очередного прохода цикла; эквивалент передачи управления в самый конец циклического оператора.

Введение в язык этих процедур почти исключает необходимость использования оператора безусловного перехода **GOTO**.

Пример программы.

```
Program Repeat_Demo;
 Var
  I  : Integer;
  St : String;
BEGIN
 Write('Введите слово: ');
 Readln(St);
 Writeln('Повторяю 10 раз.');
 I:=0;
 Repeat
  Writeln(St);
  I:=I+1;
 Until I=10;
END.
```