**Операторы цикла. Прерывания**

По английски break означает "перерыв", "прервать", и это слово хорошо знакомо боксерам. При его использовании программа немедленно выходит из цикла, продолжая выполнять последующие за циклом операторы. Так, если вы выполняете в цикле некоторую работу (вычисления), но при определенных условиях все дальнейшие вычисления бессмысленны, то смело задействуйте оператор break – он мгновенно выведет вас из цикла.

Так и оператор break или exit, МГНОВЕННО завершающие работу в нужный момент. Но следует помнить, что оператор break прерывает только тот цикл, в котором он вызван, и это надо учитывать при использовании вложенных циклов. Приведем примеры.

**Код Pascal**

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13 | **var**  ch: **char**; *{ <-- переменная ch символьного типа }*  **begin**  *{ Перебираем все символы английского алфавита }*  **for** ch := 'a' **to** 'z' **do**  **begin**  write(ch:2); *//выводим символ*  *{ Когда встречаем символ 'm', выходим из цикла: }*  **if** ch = 'm' **then** **break**  **end**;  readln  **end**. | |

В строке 8 кода выводим букву ch (параметр цикла), а ниже проверяем равенство её букве "m". Если равенство выполняется, то выходим из цикла. Поскольку после цикла в коде программы есть только оператор readln – ожидание нажатия клавиши Enter, – то после её нажатия программа на этом завершается.

Сначала в строках 5, 6 задаем границы a и b интервала случайных чисел (вместо указанных 1000 и 1999 можете взять свои параметры). В 9 строке находим случайное число n с диапазона [1000, 1999], а в 11 строке после запроса вводим делитель d по своему желанию. Если этот делитель не равен 0 и делит нацело число n (строчка 12 кода), то выводим соответствующее сообщение; в противном случае выходим из цикла (строка 14) и сообщаем о неудаче (строка 17).

**Код Pascal**

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | **var**  a, b, n, d: **integer**;  **begin**  a := 1000;  b := 1999;  randomize;  **repeat**  n := a + random(b - a + 1); *{ число 1000..1999 }*  write('Введите целое число --> ');  readln(d);  **if** (d <> 0) **and** (n **mod** d = 0) **then**  writeln(' ', n, ' делится на ', d)  **else** **break**  **until** **false**;  writeln;  writeln(n, ' НЕ ДЕЛИТСЯ НА ', d, '!');  readln  **end**. | |

Вывод

Введите целое число --> 1

1590 делится на 1

Введите целое число --> 2

1616 делится на 2

Введите целое число --> 3

1080 делится на 3

Введите целое число --> 5

1584 НЕ ДЕЛИТСЯ НА 5!

**оператор continue**

На английском continue означает "продолжать". Исходя из определения, если в цикле прогаммы встречается оператор continue, то все последующие за ним операторы пропускаются, осуществляя переход к концу цикла. Далее все зависит от вида цикла – с параметром или с условием. Если цикл с параметром (цикл for) или цикл с предусловием (while), то сразу переходим в начало и проверяем условие входа в цикл; если цикл с послеусловием (repeat), то прорверяется условие выхода из цикла.

**Код Pascal**

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13 | **var**  i, n: **integer**;  **begin**  randomize;  **for** i := 1 **to** 10 **do** *{ <-- повторяем 10 раз }*  **begin**  n := -100 + random(200); *{ диапазон чисел - [-100, 99] }*  **if** n < 0 **then** **continue**;  write(' ', n) *{ <-- выводим число }*  **end**;  readln  **end**. | |

75 96 30 24 93 80 92

Каждый раз, как мы получим отрицательное число (строки кода 8, 9), оператор continue отправит нас в самый конец цикла, пропуская вывод этого числа в строке 10.

Литература

<https://progmatem.ru/pascal/break-continue-exit.html>