

Pętla logiczne Do...Loop

Pętla logiczne służą do wielokrotnego wykonywania polecenia lub poleceń w bloku. Pełnią dokładnie taką samą rolę jak omawiana pętla For...Next lecz w odróżnieniu od niej pętle logiczne wykonywane są **nieokreśloną z góry ilość razy**. W typowym zastosowaniu pętla „For i=1 To 10” wykona się dokładnie 10 razy (chyba że stosujemy wewnątrz instrukcje GoTo, co nie jest zalecane). Pętli logicznych używamy, gdy nie wiemy dokładnie ile razy ma być wykonane polecenie lub blok poleceń w pętli. Ilość powtórzeń zależy od warunku logicznego i kod wewnątrz pętli będzie wykonywany **aż warunek zostanie spełniony** (ma wartość False, a uzyska wartość True) lub **dopóki jest spełniany** (ma Wartość True, a uzyska wartość False). Dodatkowo warunki te mogą być sprawdzane na początku lub na końcu, tzn. przed lub po wykonaniu poleceń w pętli, co daje cztery różne kombinacje. Oczywiście ich zastosowanie wynika z rodzaju zadania, które trzeba wykonać i z przyzwyczajień programisty, gdyż każdą z tych pętli (nawet pętlą For...Next) można wykonać dowolną lub zaplanowaną ilość powtórzeń. Dodać do tego należy, że również w pętlach warunkowych stosować można instrukcję **Exit Do**, która powoduje natychmiastowe zakończenie pętli albo instrukcję **Exit Sub**, która kończy działanie procedury. Wytrawny programista potrafi jednak tak skonstruować swoje pętle, by nie zawierały takich zakończeń.

Do While <i>warunek instrukcje</i> Loop	Dopóki warunek jest prawdziwy wykonywane są instrukcje. Warunek sprawdzany na początku. Instrukcje mogą nie wykonać się ani jeden raz, gdy warunek jest fałszywy na początku.
Do <i>instrukcje</i> Loop While <i>warunek</i>	Wykonywane są instrukcje dopóki warunek jest prawdziwy . Warunek sprawdzany na końcu. Najpierw wykonujemy a potem sprawdzamy, więc instrukcje wykonają się przynajmniej jeden raz.
Do Until <i>warunek instrukcje</i> Loop	Dopóki warunek jest fałszywy wykonywane są instrukcje. Warunek sprawdzany na początku. Instrukcje mogą nie wykonać się ani jeden raz, gdy warunek jest fałszywy na początku. (wykonywane są instrukcje aż warunek stanie się prawdziwy)
Do <i>instrukcje</i> Loop Until <i>warunek</i>	Wykonywane są instrukcje dopóki warunek jest fałszywy (aż warunek stanie się prawdziwy). Warunek sprawdzany na końcu. Najpierw wykonujemy a potem sprawdzamy, więc instrukcje wykonają się przynajmniej jeden raz.

Największym problemem programisty, który stosuje pętle logiczne jest takie konstruowanie warunków logicznych, aby pętla zawsze się skończyła. Istnieje niebezpieczeństwo, że nieodpowiednio skonstruowany warunek (który nie uwzględni wszystkich przypadków) spowoduje, że pętla nigdy nie zostanie zakończona (program się zawiesi).
nierówności nieostre

Wszystkie poniższe makra wykonują identyczne operacje: wpisanie do kolejnych 10 komórek liczb od 1 do 10.

<pre>i = 1 Do While i <= 10 Cells(i, 1) = i i = i + 1 Loop</pre>	<pre>i = 1 Do Cells(i, 1) = i i = i + 1 Loop While i <= 10</pre>	<pre>i = 1 Do Until i > 10 Cells(i, 1) = i i = i + 1 Loop</pre>	<pre>i = 1 Do Cells(i, 1) = i i = i + 1 Loop Until i > 10</pre>
<pre>i = 10 Do While i >= 1 Cells(i, 1) = i i = i - 1 Loop</pre>	<pre>i = 10 Do Cells(i, 1) = i i = i - 1 Loop While i >= 1</pre>	<pre>i = 10 Do Until i < 1 Cells(i, 1) = i i = i - 1 Loop</pre>	<pre>i = 10 Do Cells(i, 1) = i i = i - 1 Loop Until i < 1</pre>

```
Do
  l = Int(Rnd(10) * 10) + 1
  If Cells(l, 1) = "" Then Cells(l, 1) = l
  i = 0
  Do
    i = i + 1
  Loop Until i > 10 Or Cells(i, 1) = ""
  If i > 10 Then Exit Sub
Loop
```

Choć można też wpisać kolejnych 10 liczb za pomocą takiego algorytmu. Pętla zewnętrzna Do...Loop jest nieskończona (nie ma określonego warunku zakończenia. zakończy się dopiero wtedy, gdy zmienna „i” będzie większa od 10. Na początku pętli losujemy do zmiennej „l” losową liczbę z przedziału 1..10. Wpisujemy ją do komórki, gdy ta jest pusta. W pętli wewnętrznej sprawdzamy czy są jeszcze puste komórki lub czy już sprawdziliśmy pierwsze dziesięć. Gdy są puste – wracamy do losowania, gdy sprawdziło pierwszych i nie znalazło pustych – koniec wpisywania.

Ćwiczenia

1. Do komórki D1 arkusza „OBSŁUGA” wpisać liczbę studentów. Student istnieje jeśli pole NUMER nie jest puste. Sprawdzamy do pierwszego wystąpienia pustego pola.
2. Wyszukujemy najstarszego i najmłodszego studenta korzystając tylko pętli logicznych. Wynik do okienka MsgBox. Przycisk „URODZONY min-max”
3. W okienku InputBox wpisujemy imię studenta, a w okienku MsgBox wypisujemy wszystkie nazwiska (jedno pod drugim). Przycisk „IMIĘ”. Sprawdzaj w taki sposób, aby wyszukiwanie „było odporne” na duże i małe litery.
4. W okienku InputBox wpisujemy imię studenta, a w okienku MsgBox wypisujemy ilu jest takich studentów. Przycisk „ILU”. Można wpisać imię w dowolnej postaci – duże, małe litery, itp
5. W okienku InputBox wpisujemy przedział punktów, w postaci „min max” (np. 5 8) z jedną spacją w środku. W okienku MsgBox podajemy ilu jest studentów w tym przedziale punktowym. Przycisk „PUNKTY”.
6. Napisz makro, które posortuje studentów według nazwisk i imion. Wynik sortowania do arkusza OBSŁUGA. Przycisk „SORTUJ”. Algorytm sortowania:
 - przepisz wszystkich do tablicy z arkusza STUDENCI
 - weź kolejne nazwisko z tablicy
 - znajdź w następnych komórkach wartość minimalną
 - zamień miejscami minimalne z kolejnym
 - powtarzaj trzy poprzednie polecenia aż dojdiesz do końca tablicy
 - przepisz nazwiska z tablicy do arkusza OBSŁUGA
7. W okienku InputBox podajemy liczbę, a w okienku MsgBox wypisujemy ile jest ulic oraz nazwy ulic, których długość jest równa tej liczbie. Ponieważ nazw ulic może być dużo, dlatego w okienku MsgBox wypisujemy po trzy ulice w jednym wierszu. Przycisk „ULICE”.
8. Użyj paska przewijania do zmiany wysokości wierszy na arkuszu „STUDENCI”. Zarejestruj najpierw operacje związane ze zmianą wiersza.
9. Chcemy w polu kombi otrzymać listę miast, z których pochodzą studenci, ale tylko po jednym mieście. Napisz makro, które z całej listy makr wybierze tylko po jednym i umieści w kolumnie A arkusza „OBSŁUGA”. Przycisk „MIASTA po jednym”. Utwórz pole kombi. We właściwościach pola wstaw tę listę, a łącze komórki ustaw na D2 w arkuszu „OBSŁUGA”.
Program jest skomplikowany, dlatego został podzielony na kilka części: KOPIOWANIE, WYSZUKIWANIE, PORZĄDKOWANIE – wszystkie zostały wykonane za pomocą pętli Do...Loop.
Algorytm KOPIOWANIA po jednym:
 - jeżeli komórka w arkuszu STUDENCI jest niepusta to przepisz ją do arkusza OBSŁUGA
 - powtarzaj poprzedni krok aż kolejna komórka będzie pustaAlgorytm WYSZUKIWANIA po jednym:
 - skopiuj wszystkie miasta do wolnej kolumny
 - wyszukaj komórkę z miastem
 - wyczyść do końca te komórki, które zawierają to miasto
 - powtarzaj poprzednie dwa punkty aż do końca tabeliAlgorytm PORZĄDKOWANIA pozostałości po usuwaniu miast z tabeli
 - szukaj niepustej komórki
 - przepisz na kolejną pozycję od góry
 - powtarzaj poprzednia kroki aż do końca tabeli
10. Lista miast, jak w poprzednim ćwiczeniu, ale na zmiennych tablicowych. Gotowy wynik do arkusza OBSŁUGA.
11. Do pola kombi przypisz makro, które pokoloruje wiersze z wybranym miastem. Przygotuj makro i przycisk „KOLORUJ”, który odkoloruje cały arkusz.