

INSTRUKCJA POWTÓRZENIA FOR...TO...DO

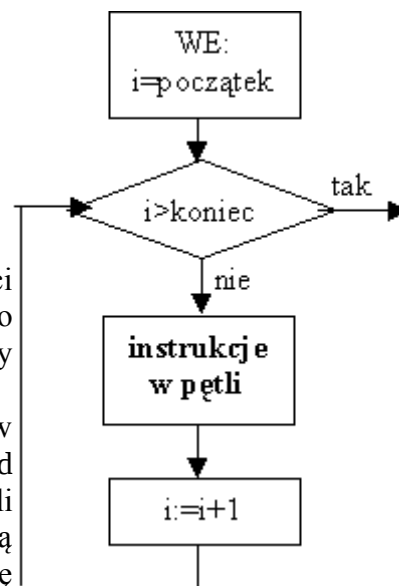
Bardzo często wykonujemy pewne działania cyklicznie - składające się z prostych, wielokrotnie powtarzanych czynności. Można by wyróżnić dwa rodzaje takich poleceń: zrób coś określoną ilość razy (dzisiejsza lekcja) albo powtarzaj pewne czynności, dopóki nie zostanie spełniony jakiś warunek (lekcja następna). Po albo przed każdym cyklem sprawdzamy, czy zadanie zostało zrealizowane. Jeśli tak - kończymy pracę, jeśli nie - kontynuujemy.

Schemat instrukcji FOR...TO...DO...

```
FOR i := 1 TO 10 DO line(100+i,100,100+i,200);  
{10 pionowych odcinków przylegających do siebie}
```

Zmienna "i", to tzw. zmienna sterująca pętlą. Przyjmuje ona kolejne wartości od 1 do 10 (wartość początkowa i końcowa zmiennej sterującej). Po przekroczeniu wartości końcowej przez zmienną sterującą, pętla kończy działanie.

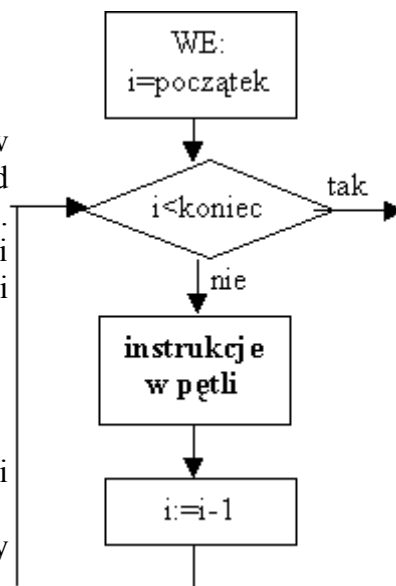
Za pomocą schematów blokowych pętla FOR może być przedstawiona w następujący sposób: zmienna sterująca przyjmuje wartość początkową. Przed wykonaniem instrukcji sprawdzany jest warunek zakończenia pętli. Jeśli zmienna sterująca nie przekroczyła wartości końcowej wykonywane są instrukcje w pętli po czym **zwiększana** jest zmienna sterująca o jeden i cykl się powtarza.



Jeśli chcemy by wartość zmiennej sterującej zmniejszała się za każdym razem o 1 stosujemy instrukcję ze słowem DOWNTO.

```
FOR i := 10 DOWNTO 1 DO line(100+i,100,100+i,200);
```

W schemacie blokowym instrukcja FOR...DOWNTO...DO wygląda w następujący sposób. Zmienna sterująca przyjmuje wartość początkową. Przed wykonaniem instrukcji w pętli sprawdzany jest warunek zakończenia pętli. Instrukcje są wykonywane, gdy zmienna sterująca nie jest mniejsza od wartości końcowej. Po wykonaniu instrukcji **zmniejszana** jest zmienna sterująca o 1 i cykl powtarza się na nowo.



Instrukcja złożona

Jeśli w pętli należy wykonać kilka instrukcji na przykład narysować kilka linii muszą one być "zgrupowane" w jedną instrukcję złożoną.

Poniższy zapis spowoduje że pierwsza linia narysuje się 10 razy i pętla skończy działanie a kolejne dwie instrukcje zostaną wykonane jeden raz.

```
FOR i := 1 TO 10 DO  
    line(100+i,100,100+i,200); {tylko to 10 razy, bo tu jest średnik kończący pętlę}  
    line(200+i,100,200+i,200); {jeden raz}  
    line(300+i,100,300+i,200); {jeden raz}
```

Aby te trzy instrukcje wykonały się 10 razy w tej pętli stosujemy instrukcję złożoną. Wszystko to, co znajduje się pomiędzy słowami begin i end wykona się w ramach jednej instrukcji.

```
FOR i := 1 TO 10 DO  
BEGIN  
    line(100+i,100,100+i,200); { 10 razy wszystkie trzy linie}  
    line(200+i,100,200+i,200);  
    line(300+i,100,300+i,200);  
END;
```

Liczby losowe

Każdy język programowania dysponuje możliwością generowania liczb losowych, Służą one m.in. do szyfrowania danych. W pascalu taką funkcją jest RANDOM(liczba), która losuje liczby z przedziału **0..liczba-1**. Dodatkowo na początku programu wywołuje się procedurę RANDOMIZE która ustawia generator, tak aby za każdym razem losowane było inne liczby.

```
x:=RANDOM(100);
```

W zmiennej X za każdym razem gdy instrukcja zostanie wywołana w programie znajdzie się losowo wybrana liczba całkowita z przedziału 0..99.

```
x:=RANDOM(5)+1; {losowo wybrane liczby z przedziały 1..6 . }
```

200 losowo narysowanych odcinków

```
FOR i := 1 TO 200 DO
```

```
BEGIN
```

```
    x1:=random(600);
```

```
    y1:=random(400);
```

```
    x2:=random(600);
```

```
    y3:=random(400);
```

```
    line(x1,y1,x2,y2);
```

```
END;
```

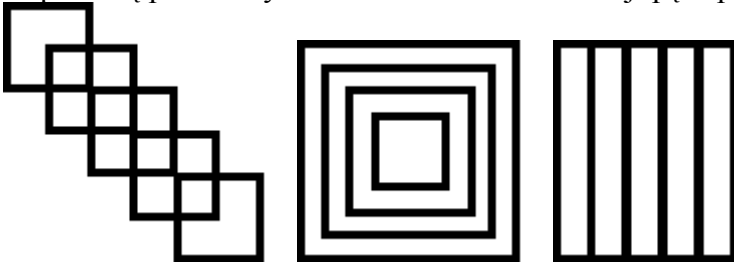
Przykładowe programy znajdują się na szkolnej stronie internetowej

Ćwiczenia

narysuj za pomocą LINE, LINETO, LINEREL następujące rysunki (długości i odległości dowolne):



za pomocą procedury PROSTOKATK i instrukcji pętli przygotuj następujące rysunki:



Za pomocą linii pionowych lub poziomych wypełnij kwadrat dowolnym kolorem

