

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

Turbo Pascal. Leksykon kieszonkowy

Autorzy: Jan Zahorski, Andrzej Orłowski

ISBN: 83-7361-741-8

Format: B6, stron: 136



Turbo Pascal, mimo tego, że może być uważany za weterana na rynku języków programowania, nadal cieszy się sporą popularnością, szczególnie wśród osób poznających podstawy samodzielnego tworzenia aplikacji. Prosta i czytelna składnia, spore możliwości i sprawdzone środowiska programistyczne to przyczyny powodzenia Turbo Pascala. Za poznawaniem tego języka przemawia również fakt wykorzystywania jego nieco bardziej rozbudowanej wersji w środowisku Delphi – jednym z najpowszechniej wykorzystywanych narzędzi do tworzenia aplikacji.

Książka „Turbo Pascal. Leksykon kieszonkowy” to zbiór najbardziej niezbędnych informacji o Turbo Pascalu. Przedstawia najistotniejsze elementy języka, koncentrując się na tworzeniu aplikacji pracujących pod kontrolą systemu Windows. Zawarte w niej informacje będą szczególnie przydatne dla tych Czytelników, którzy już znają podstawowe zasady programowania w języku Turbo Pascal, a w książce będą szukali pomocy w rozwiązaniu konkretnych problemów związanych z samodzielnym tworzeniem aplikacji użytkowych.

- Struktura programu
- Deklaracje
- Typy i operatory
- Instrukcje
- Procedury i funkcje
- Struktury
- Elementy zewnętrzne



Spis treści

1. Struktura programu	9
2. Komentarze	12
3. Deklaracje.....	14
Deklaracje etykiet	14
Deklaracje podprogramów	14
Deklaracje typów	16
Zakres i zasięg deklaracji	16
Deklaracje zmiennych	17
4. Typy	18
Typy Windows API	18
Typy literałów	19
Typy proste	19
Typy mnogościowe	22
Typy okrojone	23
Typy plikowe	23
Typy rekordowe	23
Typy tablicowe	24
Typy wskazujące	24
Typy wyczeniowe	25
Zgodność typów	26

5. Operatory	27
Operatory arytmetyczne	27
Operatory logiczne	27
Operatory łańcuchowe	28
Operatory mnogościowe	29
Operatory relacji	30
Priorytety operatorów	30
6. Instrukcje	31
Instrukcja grupująca	31
Instrukcja kodowa	31
Instrukcje powtarzające	32
Instrukcja procedury (procedure)	34
Instrukcja skoku (goto)	34
Instrukcja przypisania (:=)	35
Instrukcja pusta	35
Instrukcja warunkowa (if)	35
Instrukcja wiążąca (with)	36
Instrukcja wyboru (case)	37
7. Podprogramy.....	39
Podprogramy kodowe	39
Podprogramy obsługi przerwania (interrupt)	40
Podprogramy skojarzenia parametrów z argumentami	40
Podprogramy rekurencyjne	42
8. Dyrektywy kompilatora.....	43
Symbole predefiniowane	46
9. Struktury.....	49
TMsg	49
TOpenFilename	49
TPoint	51

TPaintStructure	51
TRect	51
TWndClass	52
10. Procedury i funkcje	53
11. Elementy zewnętrzne programu	115
Moduły	115
Biblioteki DLL	118
Pliki włączane do programu dyrektywą {\$I Nazwa_programu}	120
A API.....	121
Przykład programu z wykorzystaniem funkcji API	121
Skorowidz	135

Rozdział 6. Instrukcje

Instrukcja grupująca

Instrukcja grupująca to jedna z najważniejszych instrukcji w programie, ponieważ każdy program w Turbo Pascalu musi mieć przynajmniej jedną taką instrukcję tworzącą główny blok programu. Instrukcja grupująca rozpoczyna się słowem kluczowym **begin**, a kończy słowem kluczowym **end**. Instrukcja grupująca umożliwia wykonanie kilku instrukcji, w przypadku gdy ze względów składniowych użyć możemy tylko jednej instrukcji (pętle) lub przy budowie procedur i funkcji, np.

```
program test;
uses WinCrt;
var
  A, C : Integer;
begin
  C := 0;
  while C < 10 do
    begin
      A := A + C;
      Inc(C);
    end;
end.
```

Instrukcja kodowa

Instrukcja kodowa **inline** (argument) przeznaczona jest do umożliwienia użycia w programie instrukcji asemblera, czyli skorzystania z języka niskiego poziomu, ale za to dającego możliwość pełnej kontroli nad maszyną.

```
inline($58); {odpowiada rozkazowi pop AX}
```

Instrukcje powtarzające

Dla (for)

Instrukcja powtarzająca **for** przeznaczona jest głównie do wykonywania pojedynczej instrukcji lub grupy instrukcji założoną a priori ilość razy. Ilość powtórzeń pętli deklarujemy w momencie jej pisania, zakładając, że nic nie zakłuci jej przebiegu. Pętla *dla* składa się ze słowa kluczowego **for**, nazwy i deklaracji wartości początkowej zmiennej sterującej, słowa kluczowego **to** (**down to**), deklaracji wartości końcowej zmiennej sterującej, słowa kluczowego **do**, po którym następuje *instrukcja (instrukcje)*, które mają być wykonywane w pętli. Najprostszą realizację pętli ilustruje listing:

```
for a := 0 to 9 do a := a + 5; {Pętla będzie się wykonywać 10 razy}
```

Nieco bardziej skomplikowany przykład przedstawia listing:

```
for i := 0 to 127 do
  for j := 0 to 127 do
    begin
      Bufor[i, j] := i + j;
    end;
```

W przykładzie tym mamy do czynienia z zagnieżdżeniem pętli, co jest zjawiskiem dosyć powszechnym i często stosowanym w praktyce.

Dopóki (while)

Instrukcja powtarzająca *dopóki* składa się ze słowa kluczowego **while**, warunku i słowa kluczowego **do**. Istotną cechą tej instrukcji jest to, że warunek początkowy jest sprawdzany każdorazowo przed kolejnym wykonaniem pętli i jeżeli przybierze wartość **False**, wykonywanie pętli zostaje zakończone. Znajomość

momentu sprawdzania warunku ma niezwykle istotne znaczenie dla określenia ilości powtórzeń pętli. Jeżeli o tym zapomnimy to możemy po zakończeniu pętli otrzymać wynik niezgodny z założonym czyli fałszywy. Przykłady wykorzystania instrukcji **while** przedstawiają listingi:

```
X := 0;
while X < 10 do {Fałsz gdy X >= 10}
begin
  Inc(X); {Uwaga. Jeżeli zapomnisz nie wyjdiesz z pętli!}
end;
Writeln('X < 10)= ', X); {Wynik X = 10}
```

```
X := 0;
while X <= 10 do {Fałsz gdy X = 11}
begin
  Inc(X); {Uwaga. Jeżeli zapomnisz nie wyjdiesz z pętli!}
end;
Writeln('X <= 10 = ', X); {Wynik X = 11}
```

Dopóty (repeat)

Instrukcja powtarzająca *dopóty* składa się ze słowa kluczowego **repeat**, instrukcji do wykonania, słowa kluczowego **until** i warunku zakończenia pętli. Przykłady przedstawiają listingi:

```
X := 0;
repeat
  Inc(x); {Uwaga. Jeżeli zapomnisz nie wyjdiesz z pętli!}
until X > 10;
Writeln('X > 10 = ',X); {Wynik X = 11}
```

```
X := 0;
repeat
  Inc(x); {Uwaga. Jeżeli zapomnisz nie wyjdiesz z pętli!}
until X >= 10;
Writeln('X >= 10 = ',X); {Wynik X = 11}
```

Instrukcja procedury (procedure)

Instrukcja procedury składa się z : słowa kluczowego **procedure**, nazwy procedury, opcjonalnie w nawiasach argumentów oraz instrukcji grupującej **begin..end**.

Procedura z parametrami

```
procedure Dodaj(X, Y : Integer);  
begin  
  X := Y + 10;  
end;
```

i procedura bez parametrów

```
procedure EkranTekstowy;  
begin  
  X := ScreenSize.X;  
  Y := ScreeSize.Y;  
  Writeln('Długość ekranu tekstowego = ', X);  
  Writeln('Wysokość ekranu tekstowego = ', Y);  
end;
```