



## Wykład V

### PASCAL

- operacje na znakach i łańcuchach;
- dźwięk,



## Łańcuchy i znaki

# Operacje na znakach w łańcuchu

```
1 Program lit;  
2 uses Crt;  
3 var  
4     s:string;  
5     i:integer;  
6 begin  
7     ClrScr;  
8     Write('Podaj ciąg znakow: ');  
9     ReadLn(s);  
10    for i:=1 to Length(s) do  
11        Write(s[i], ' ');  
12        Readkey;  
13    end.
```

Program wypisuje zawartość łańcucha dzieląc litery spacjami

# Operacje na znakach w łańcuchu

```
1 Program lit;  
2 uses Crt;  
3 var  
4     s:string;  
5     i:integer;  
6 begin  
7     ClrScr;  
8     Write('Podaj ciąg znakow: ');  
9     ReadLn(s);  
10    for i:=1 to Length(s) do  
11        if (s[i] IN ['a'..'z','A'..'Z']) OR (s[i]=' ') then  
12            write(s[i]);  
13            Readkey;  
14    end.
```

Program wypisuje tylko litery i spacje - ignoruje inne znaki



## Znak <-> liczba

Składnia:

```
znak := Chr (liczba) ;
```

```
znak := #liczba ;
```

Każdy litera, cyfra czy inny znak posiadają swój własny numer odczytywany z tablicy kodów ASCII, np. znakowi 'A' odpowiada liczba 65, znakowi 'a' liczba 97, a znakowi '2' liczba 50. Aby można było zamieniać liczbę na znak używa się właśnie funkcji Chr, która zwraca znak typu CHAR

```
1 Begin
2   WriteLn (chr (70)) ;
3   WriteLn (chr (43)) ;
4   WriteLn (#43) ;
5   WriteLn (chr (39)) ;
6 End.
```

## Znak <-> liczba

Składnia:

```
liczba := Ord (znak) ;
```

Ta funkcja jest podobna do funkcji Chr, tylko że w niej zamieniane jest na odwrót, nie liczby na znaki, tylko znaki na liczby :) np. ord('A')=65 ord('a')=97 ord('2')=50 itd.

```
1  USES CRT;
2  Var
3      znak:char;
4  Begin
5      Repeat
6          znak:=ReadKey;
7          WriteLn (znak, '      ', ord (znak) );
8      Until znak=Chr (27) ;
9  End.
```

27 – numer znaki Esc



```
1 Program ascii;
2 Uses Crt;
3 Var i:integer;
4 begin
5   ClrScr;
6   for i:=32 to 255 do
7     begin
8       Write (i:3,'-', Chr(i), ' ');
9       if (i-31) mod 10 = 0 then WriteLn('');
10    end;
11  Readkey;
12 end.
```

Pierwsze 31 znaków kodu ASCII to znaki sterujące

Wypisujemy znak o numerze w kodzie ASCII podanym przez zmienną "i"

jeżeli "i" jest podzielne przez 10 (reszta z dzielenia równa zero) przechodzimy do nowej linii

Program wypisuje na ekranie znaki kodu ASCII, wraz z ich numerami.

## Obsługa Klawiszy Rozszerzonych

Klawisze rozszerzone podobnie jak zwykle odczytuje się funkcją `ReadKey`. Jest jednak jedna różnica, jeżeli to jest klawisz rozszerzony to funkcja `ReadKey` zwróci nam wartość `chr(0)`, aby rozpoznać który to klawisz z rozszerzonych, to należy **POWTÓRNIIE ODCZYTAĆ KLAWISZ** funkcją `ReadKey`.

```
1  USES CRT;
2  Var
3      normalny, rozszerzony:char;
4  Begin
5      WriteLn( 'Wcisnij jakis klawisz...');
6      normalny:=ReadKey;
7      if normalny=chr(0) then
8      Begin
9          rozszerzony:=ReadKey;
10         WriteLn( 'Wcisniety zostal rozszerzony klawisz: ', rozszerzony );
11     End Else WriteLn( 'Wcisniety zostal klawisz: ', normalny );
12 End.
```



# Pobieranie znaku z klawiatury

```
1 Program lit;  
2 uses Crt;  
3  
4 var  
5   c: char;  
6   nr: integer;  
7 begin  
8   repeat  
9     c := ReadKey;  
10    if c = #0 then nr :=Ord(ReadKey) else nr:=Ord(c);  
11    WriteLn(nr);  
12  until c=#27;  
13 end. _
```

Funkcja Readkey  
Pobiera pojedynczy znak i  
zwraca go do zmiennej typu  
Char

Klawisze sterujące (strzałki)  
wysyłają 2 znaki #0 i  
#numer

Kod znaku ESC



## Przykład – „kod PIN”

---

### **Zadanie:**

Program ma wczytać 4-ro cyfrowy kod PIN i sprawdzić jego poprawność.

### **Wymogi:**

- Kod wpisywany niejawnie (na ekranie pojawiają się gwiazdki),
- Program przyjmuje tylko cyfry, inne znaki są ignorowane,
- Kodu nie trzeba zatwierdzać,
- Użytkownik ma tylko 3 próby.



```
1 Program paselko;
2 Uses Crt;
3 var i, ile: Integer;
4     pass, pin: String;
5     {pass - wczytywane haslo
6     |pin - prawidlowe haslo zapamietane w programie}
7     c: char;
8     ok:Boolean;
9 begin
10    pin:='1234';
11    ok:=false;
12
13    repeat
14        Write('Podaj PIN');
15        pass:='';
16        i:=0;
17
18        repeat
19            c:=ReadKey;
20            if c IN ['0'..'9'] then
21                begin
22                    Write('*');
23                    pass:=pass+c;
24                    i:=i+1;
25                end;
26            until i=4;
27
28            ile:=ile+1;
29            if pass=pin then
30                begin
31                    ok:=true;
32                    WriteLn('OK');
33                end
34            Else
35                WriteLN('zle');
36            until (ile=3) OR (ok);
37
38    Repeat until Keypressed;
39 end.
```

Zmienna "ok" informuje, czy podano poprawne haslo - na początku ustawiona jest na "falsz"

Jeżeli wczytany znak "c" jest cyfrą:  
- na ekran wypisywany jest znak \*,  
- wczytany znak dołączany jest na koniec łańcucha "pass",  
zwiększana jest licznik wczytanych znaków (zmienna "i")

Pętla wczytująca hasło znak po znaku.  
Zakończona zostaje gdy wczytane zostają 4 znaki (zmienna "i" równa 4)

Jeżeli wczytane hasło jest prawidłowe (pass=pin)  
ustawiamy zmienną "ok" na wartość prawda.

Pętla zakończona zostanie gdy zmienna ile wyniesie 3 (trzy łądne próby wpisania hasła) lub też gdy zmienna "ok" będzie miała wartość "prawda (wpisano prawidłowe hasło)



## Dźwięk w PASCALU



## Dźwięk w PASCALU

---

Składnia:

**Sound(wysokość);**     *[częstotliwość podana w Hercach]*

Polecenia SOUND( ) powoduje rozpoczęcie odtwarzania dźwięku o podanej częstotliwości.

Składnia:

**NoSound;**     *[bez parametrów]*

Polecenia NOSOUND powoduje zakończenie odtwarzania dźwięku.

Składnia:

**Deleay(czas);;**     *[czas podany w milisekundach]*

Polecenia DELAY( ); powoduje, że program odczeka określony w parametrze czas.

## Dźwięk w PASCALU

---



Wygenerowanie sygnału dźwiękowego to zwykle trzy polecenia:

```
Sound (wysokość)  
Deleay (opóźnienie) ;  
NoSound;
```

# Dźwięk w PASCALU

Program „Pianinko” – pod wybrane klawisze podpinamy dźwięki o różnych częstotliwościach.

```
1 Program Pianinko;
2 USES CRT;
3 Var
4   klawisz:char;
5 Begin
6   WriteLn( 'ESC to koniec programu');
7   Repeat
8     klawisz:=readkey;
9     case klawisz of
10      q: sound(2000);
11      w: sound(3000);
12      e: sound(4000);
13      r: sound(5000);
14      t: sound(6000);
15      y: sound(7000);
16      u: sound(8000);
17    end;
18    deley(50);
19    nosound;
20 until klawisz = #27;
21 End.
```

Program kończy naciśnięcie klawisza ESC  
(kod znaku 27)

## Literatura:

---

- R. Jarża, Turbo Pascal. Szkoła programowania, Wydawnictwo Robomatic 2000. *(dostępne w bibliotece uczelni)*
- J. Bishop, Turbo Pascal, Wydawnictwo RM, Warszawa 1999.
- K. Jakubczyk, Turbo Pascal i Borland C++ Przykłady, Helion, Gliwice 2002. *(dostępne w bibliotece uczelni)*

