

Instrukcje podsumowanie

Proste:

- przypisania
- wejścia-wyjścia (read, readln, write, writeln)
- pusta - po prostu ; (średnik)

Strukturalne:

- **grupująca**

```
begin <zestaw instrukcji> end
```

- **warunkowa**

```
if <wyrażenie> then <instrukcja1>;
```

lub

```
if <wyrażenie> then <instrukcja1>  
    else <instrukcja2>;
```

- **pętli**

```
repeat
```

```
    <instrukcje>
```

```
until <warunek_końca>;
```

```
while <warunek kontynuacji> do
```

```
    <instrukcja>;
```

```
suma := 0;  
writeln('Liczby zakończ 0');  
readln(a)  
while a <> 0 do  
    begin  
        suma := suma + a;  
        readln(a);  
    end;  
writeln('Suma=', suma:8:2);
```

```
suma := 0;  
writeln('Liczby zakończ 0');  
repeat  
    readln(a);  
    suma := suma + a;  
until a = 0;  
writeln('Suma=', suma:8:2);
```

```
for <zmienna> := <wyr_pocz> to <wyr_kon> do <instrukcja>;
```

```
for <zmienna> := <wyr_pocz> downto <wyr_kon> do <instrukcja>;
```

Przykłady

```
for i := 1 to n do  
  writeln('i=', i, ' i*i=', i*i);
```

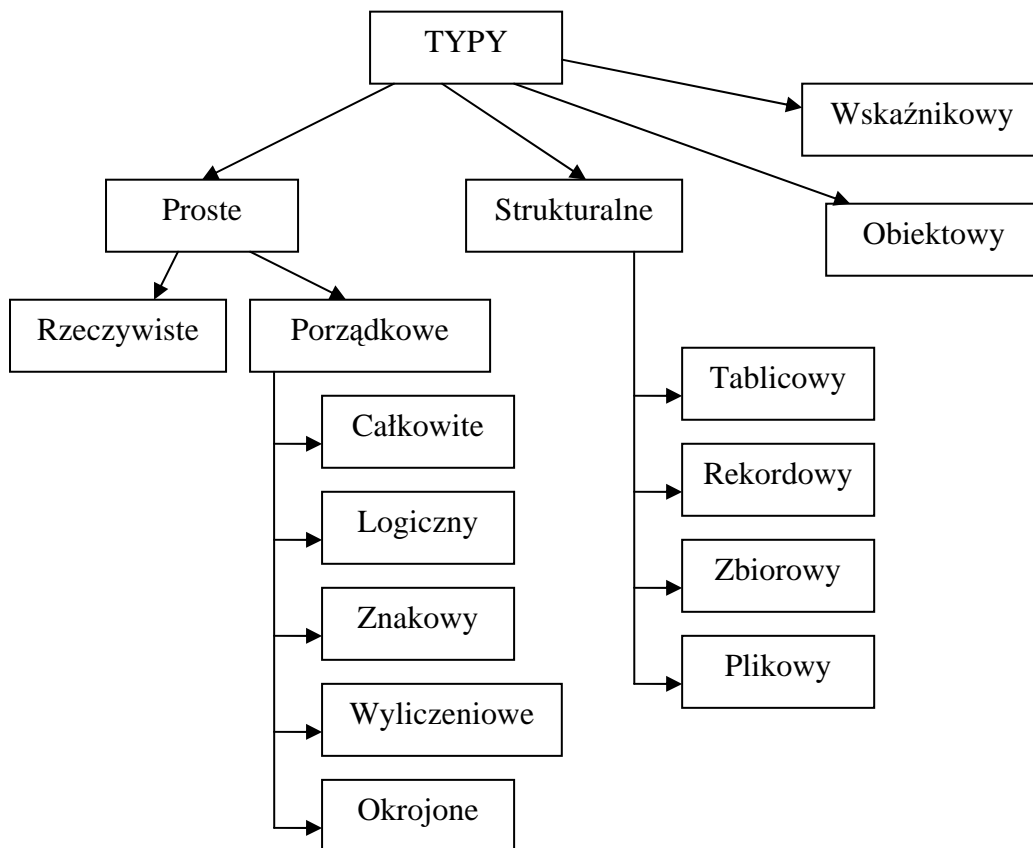
Ile dodatnich z 2, 3, 4 liczb

```
readln(a, b);  
ile := 0;  
if a > 0 then ile := ile+1;  
if b > 0 then ile := ile+1;  
writeln('Ilosc liczb dodatnich: ', ile);
```

Ile dodatnich z N liczb:

```
readln(n);  
ile := 0;  
for i := 1 to n do  
  begin  
    readln(a);  
    if a > 0 then ile := ile+1;  
  end;  
writeln('Ilosc liczb dodatnich: ', ile);
```

Klasyfikacja typów danych w języku Pascal:



Definiowane typy proste w pascalu.

Typ okrojony:

[*<wartosc_dolna>* .. *<wartosc_gorna>*]

Typ taki obejmuje wartości określone zakresem. Oba ograniczenia muszą być jednakowego typu prostego nie rzeczywistego.

Typ wyliczeniowy

(*<stała1>* , *<stała2>* , ...)

Stałe muszą być identyfikatorami. Typ zawiera tyle wartości ile jest wymienione w deklaracji. Deklaracja ta służy jednocześnie do zdefiniowania stałych.

Problem:

Dane jest N liczb. Obliczyć ile z nich jest większe od średniej wszystkich liczb.

```
readln(n);
suma := 0;
for i := 1 to n do
  begin
    readln(a);
    suma := suma+a;
  end;
srednia:=suma/n;
ile := 0;
for i := 1 to n do
  begin
    readln(a);
    if a > srednia then ile := ile+1;
  end;
writeln('Ilosc liczb wiekszych od sredniej: ', ile);
```

Zamieszczony wyżej przykład wymaga dwukrotnego podawania liczb przez użytkownika co jest niedopuszczalne. Komputer powinien raz wczytać liczby, zapamiętać je, by następnie można było wielokrotnie z nich korzystać.

Tablice

Tablica to pojemnik na ciąg wartości (liczb). Pojemnik ma jedną nazwę, a składa się z wielu jednakowych komórek Każda komórka przeznaczona jest do przechowywania jednego elementu tablicy.

Komórki są ponumerowane (indeksowane). Indeksami komórek są zwykle kolejne liczby całkowite.

A

1	3
2	-4.5
3	12
	⋮
	⋮
	⋮
N	-2

Element tablicy identyfikujemy podając jej nazwę i indeks elementu. Obowiązuje składnia:

<nazwa tablicy> [<indeks elementu>]

Na przykład

A[1] element 1-szy
A[2] element 2-gi

A[i+1] element i+1 –szy

Deklaracja tablic

Obowiązująca składnia opisu typu tablicowego:

```
array [ <typ_porządkowy>, ... ] of <typ_elementu>;
```

Tablica może mieć wiele wymiarów.

Deklaracja tablicy jednowymiarowej:

```
array [ <typ_porządkowy> ] of <typ_elementu>
```

<typ_porządkowy> określa zakres indeksów tablicy (a więc także liczbę elementów tablicy)

<typ_porządkowy to najczęściej okrojenie.

Przykłady deklaracji tablic:

Type

```
Tab1 = array [ 1 .. 10 ] of char;
```

```
Tab2 = array[ 1.. 10, 1..10 ] of integer;
```

Var

```
Nap: array [ 1..20 ] of Tab1;
```

```
X, Y : array [ 1 .. 10 ] of real;
```

```
Y1, Y2 : Tab2;
```

Można deklarować tablice o bardzo złożonej strukturze. Najczęściej używane są jednowymiarowe o indeksach wyrażanych liczbami całkowitymi od 1.

W obsłudze tablic przydatna jest instrukcja **for**.

Rozwiązanie problemu (liczba większych od średniej) z wykorzystaniem tablic:

```
var i, n, ile: integer;  
      suma, srednia: real;  
      a: array [1..100] of real;  
begin  
writeln('Podaj liczbe elementow');  
readln(n);  
writeln('Podaj elementy');  
for i := 1 to n do                    { wczytywanie tablicy }  
      readln(a[i]);
```

```

suma := 0;
for i := 1 to n do           { sumowanie elementow tablicy }
    suma := suma+a[i];
srednia:=suma/n;
ile := 0;
for i := 1 to n do         { zliczanie liczby el. > sredniej }
    if a[i] > srednia then ile := ile+1;
writeln('Ilosc liczb wiekszych od sredniej: ', ile);
end.

```

Przykład – wyznaczenie maksimum z użyciem tablic:

```

var i, n, imax: integer;
    max: real;
    a: array [1..100] of real;
begin
writeln('Podaj liczbe elementow');
readln(n);
writeln('Podaj elementy');
for i := 1 to n do
    readln(a[i]);
max := a[1]; imax:=1;
for i := 1 to n do
    if a[i] > max then
        begin max := a[i]; imax := 1; end;
writeln('Najwiekszy element na pozycji: ', imax,
        ' ma wartosc = ', max:8:2);
end.

```

Procedury i funkcje

Przykład – pole trójkąta na podstawie 3 boków wykorzystaniem funkcji

```

var a, b, c, p: real;

function pole (a, b, c: real) : real;
var p: real;
begin
p:=(a + b +c)/2;
pole:=sqrt((p-a)*(p-b)*(p-c)*p);
end;

begin
writeln('Podaj boki');
readln(a, b, c);
p:=pole(a, b, c);
writeln('Pole = ', p:8:2);
readln;
end.

```

Składnia funkcji

```
function <nazwa>(<lista_parametrów>) : <typ_wyniku>;  
  <deklaracje_lokalne>  
begin  
  <instrukcje>  
  <nazwa> := <wyrażenie>;           { lub return <wyrażenie>; }  
end;
```

Przykład – rysowanie gwiazdek

```
var k: integer;  
procedure gwiazdki (n: integer);  
var i: integer;  
begin  
  for i := 1 to n do write('*');  
  writeln;  
end;  
begin  
  writeln('Podaj liczbe');  
  readln(k);  
  gwiazdki(k);  
  readln;  
end.
```

```
var i, n: integer;  
    a: array [1..100] of integer;  
procedure gwiazdki (n: integer);  
var i: integer;  
begin  
  for i := 1 to n do write('*');  
  writeln;  
end;  
begin  
  writeln('Podaj liczbe paskow');  
  readln(n);  
  writeln('Podaj ilosci gwiazdek');  
  for i := 1 to n do  
    readln(a[i]);  
  for i := 1 to n do gwiazdki(a[i]);  
  readln;  
end.
```

Składnia procedury

```
procedure <nazwa>(<lista_parametrów>);  
  <deklaracje_lokalne>  
begin  
  <instrukcje>  
end;
```

Przykład – wyznaczenie maksimum z użyciem procedur:

```
type t1=array [1..100] of real;  
var i, n: integer;  
    max: real;  
    a: t1;  
procedure emax(n: integer; x: t1; var max: real);  
var i: integer;  
begin  
max := a[1];  
for i := 1 to n do  
    if a[i] > max then max := a[i];  
end;  
begin  
write('Podaj liczbe elementow: ');  
readln(n);  
writeln('Podaj elementy');  
for i := 1 to n do  
    readln(a[i]);  
writeln('Poczatek obliczen');  
Emax(n, a, max);  
writeln('Najwiekszy element =', max:8:2);  
readln;  
end.
```