

Powtórzenie Pascal:

1. Typy zmiennych:
 - liczbowe: stałoprzecinkowa i zmiennoprzecinkowa
 - deklaracja zmiennych
2. Operatory:
 - przypisania
 - porównania
 - negacji
3. Instrukcje warunkowe:
 - IF, IF ELSE, CASE
4. Pętle:
 - FOR (TO i DOWNTO)
 - WHILE
 - REPEAT
5. Tablice:
 - deklaracja tablicy jedno, dwu i trójwymiarowej
 - Czytanie i wyświetlanie tablic
6. Obiekty:
 - deklaracja obiektu, pól i metod
 - implementacja (oprogramowanie) metod
 - odwołanie do obiektu w programie
7. Budowa części głównej programu
8. Funkcje i procedury
 - deklaracja i zwracane wartości
 - pojęcie parametrów formalnych
 - przekazywanie parametrów przez zmienną i przez wartość

-
1. Integer -32768..32767
Byte 0..255
Word 0..65535
LongInt ~ -2*10⁹ - 2*10¹⁰
Real 2,9*10⁻³⁹ - 1,7*10³⁸ zmiennoprzecinkowa
 2. Przypisania :=
Porównania a>b, a<b , a>=b, a<=b, a<>b
negacji not a
 3. Instrukcje warunkowe
 - a. IF
 - i. if (a>b) then writeln('a > b');
 - b. IF ELSE
 - i. if (a>b) then writeln('a > b')
 else writeln('a <= b')
 - c. CASE
 - case b of
 - 1: instrukcja1;
 - 2: instrukcja2;
 - 3: begin
 - intsrukcja3;
 - instrukcja4;
 - end;
 - else
 - instrukcja5;
 - endcase;
 4. Pętle
 - a. FOR ... TO ...
 - i. For i:=1 to 10 do
 - write(i)
 - ii. For i:=10 downto 1 do
 - write(i)
 - iii. i:=1;
 - While (i<=10) do
 - Begin

```

        Writeln(i);
        Inc(i);
    End;
iv. i:=1;
    Repeat
        witeLn(i);
        inc(i);
    Until (i>10)

```

5. Tablice

a. 1-wymiarowe

- i. Nielelegantkie:


```
var a: array [1..10] of integer;
```
- ii. 2 sposób


```
Type _tab = array [1..10] of integer;
var a:_tab;
```
- iii. Odwołanie do element tablicy


```
a[5], a[i]
```

b. 2- wymiarowe

- i.

```
Type _tab = array [1..10,1..4] of integer;
var a:_tab;
```
- ii. Odwołanie do element tablicy


```
a[3,4], a[i,j]
```

c. 3-wymiarowe

- i.

```
Type _tab = array [1..10,1..4,1..5] of integer;
var a:_tab;
```
- ii. Odwołanie do element tablicy


```
a[3,4,5], a[i,j,z]
```

6. Obiekty

```

type _Strzal = object
    x,y:integer;
    function punkty:integer;
    procedure czytaj;
end;

```

```

function _Strzal.punkty:integer;
var odl:real;
    pkt,i:integer;
    ok:boolean;
begin
    odl:=sqrt(sqr(self.x)+sqr(self.y));
    pkt:=0;
    ok:=true;
    i:=1;
    while ((i<=10) and ok) do
        begin
            if (odl<=i) then
                begin
                    pkt:=10-i+1;
                    ok:=false;
                end;

            inc(i);    {i:=i+1}
        end;

    punkty:=pkt;
end;

```

[...]

```

begin
    s.czytaj;
    writeln('WYNIK = ',s.punkty);

    repeat until keypressed;
end.

```

7. Budowa części głównej programu

```
program .....          nazwa programu
uses ...                biblioteki
type ...               deklaracje typów
var ...                deklaracje zmiennych
funkcje i procedury ... implementacja funkcji i procedur
Begin
  Cześć główna programu...
End.
```

8. Funkcje i procedury

```
program strzelanie;
uses crt;

var x,y:integer;
    i,punkty:integer;
    ok:boolean;
    odl:real;

procedure Czytaj(var rx,ry:integer); //przekazywanie przez zmienną
// rx,ry parametry formalne
begin
  clrscr;
  writeln('Podaj punkt P(x,y):');
  write(' x = ');
  readln(rx);
  write(' y = ');
  readln(ry);
end;

function Odleglosc(rx,ry:integer):real; //przekazywanie przez wartość
// rx,ry parametry formalne
begin
  return sqrt(sqr(x)+sqr(y));
end;

begin
  Czytaj(x,y);
  odl:=Odleglosc(x,y);
  punkty:=0;
  ok:=true;
  i:=1;
  while ((i<=10) and ok) do
    begin
      if (odl<=i) then
        begin
          punkty:=10-i+1;
          ok:=false;
        end;
      inc(i); //i:=i+1
    end;
  writeln('WYNIK = ',punkty);
  repeat until keypressed;
end.
```