

Wydział: **Zarządzania i Modelowania Komputerowego**
Przedmiot: **Języki programowania - Delphi**
Rok **III** Semestr 5
Opracował: **dr inż. Krzysztof Strzałkowski**

Ćwiczenie 4 Proste obiekty

Przykład 1

Zdefiniuj typ obiektowy do obsługi obliczeń dla trójkąta. Składowe obiektu:

a, b, c długości boków,
obwód metoda-funkcja wyznaczająca obwód trójkąta na podstawie 3 boków.
Wykorzystaj obiekt w programie wyznaczania obwodu trójkąta.

Rozwiązanie:

```
program Trojkat;  
{ $APPTYPE CONSOLE }  
uses SysUtils;  
type TTrojkat = object  
    a, b, c: real;  
    function obwod:real;  
end;  
function TTrojkat.obwod:real;  
begin  
    obwod:=a+b+c;  
end;  
var t: TTrojkat;  
  
begin  
    writeln('Podaj 3 boki trojkata');  
    readln(t.a, t.b, t.c);  
    writeln('Obwod=', t.obwod:8:2);  
    readln;  
    writeln('Czekam na Enter...'); readln;  
end.
```

Wprowadź tekst powyższego programu, skompiluj i uruchom program dla przykładowych danych.

Zadanie 1

a) Rozbuduj definicję typu **TTrojkat** dodając definicję metody **Pole**. Wykorzystaj wzór Herona:

```
p = obwod/2;  
pole = sqrt((p-a)*(p-b)*(p-c)*p)
```

Przetestuj działanie metody.

b) Dokonaj dalszej rozbudowy obiektu o 3 metody wyznaczające 3 wysokości trójkąta. Wykorzystaj zależność:

$\text{Wysokość na bok } a = 2 * \text{pole} / a$

Przetestuj działanie metody.

Zadanie 2

Opracuj deklarację typu obiektowego **TMieszkanie** grupującego składowe:

metraz - powierzchnia całego mieszkania,

liczba_izb - ilość izb w mieszkaniu

pow_izby - **metoda** wyznaczająca średnią powierzchnie jednej izby w mieszkaniu

Napisz program wykorzystujący powyższy typ obiektowy do obliczenia średniej powierzchni jednej izby na podstawie wczytywanych metrażu mieszkania i liczby izb w mieszkaniu.

Zadanie 3

Rozbuduj program stanowiący rozwiązanie zadania 2 o definicję nowego typu obiektowego **TMiesz2** dziedziczącego składowe typu **TMieszkanie** oraz zawierającego dodatkowo składowe:

koszt - cena sprzedaży całego mieszkania

cena_metra - **metoda** obliczania średniej ceny jednego metra mieszkania

Program powinien wykonywać obliczenia jak w zadaniu 2 oraz dodatkowo wyświetlać cenę metra kwadratowego mieszkania na podstawie wczytywanego dodatkowo kosztu zakupu mieszkania.

Zadanie 4

Opracuj obiekt do obsługi wyznaczania oceny stylu skoku narciarskiego. Zastosuj deklarację:

```
type TNoty = object
    X: array[1..5] of real;
    function ZaStyl: real;
    function Max:real; {wyznaczanie oceny najwyzszej}
    function Min:real; {wyznaczanie oceny najnizszej}
    function Suma:real; {wyznaczanie sumy ocen}
end;
```

Przeznaczenie składowych typu **TNoty**:

X – tablica przeznaczona na oceny 5 sędziów,

Max – pomocnicza funkcja zwracająca wartość oceny najwyższej,

Min – pomocnicza funkcja zwracająca wartość oceny najniższej,

Suma – pomocnicza funkcja zwracająca wartość sumy 5 ocen,

ZaStyl – funkcja zwracająca obliczoną ocenę stylu skoku równą sumie ocen pięciu sędziów z odrzuceniem oceny najniższej i najwyższej ($ZaStyl = Suma - Max - Min$).

Napisz program testujący obiekt.

Zadanie 5

Rozbuduj program z zadania 2 do postaci dwumodułowej. Dodaj do projektu plik nowego modułu przy pomocy opcji:

File | New | Unit.

Zwróć uwagę, że dołączenie modułu spowodowało uzupełnienie specyfikacji **uses** w programie głównym - została dopisana pozycja nowego modułu (Unit1).

Definicję typu **TNoty** przenieś do nowego modułu. Sam opis typu powinien być umieszczony w sekcji **interface**, a opis metod w sekcji **implementation**.

Sprawdź, że po zmianach program działa dobrze.

Zmodyfikuj obiekt **TNoty** do postaci:

```
type TNoty = object
    procedure UstawDane(a,b,c,d,e: real);
    function ZaStyl: real;
private
    X: array[1..5] of real;
    function Max:real;
    function Min:real;
    function Suma:real;
end;
```

Dodaj zapis metody **UstawDane**:

```
procedure TNoty.UstawDane(a,b,c,d,e: real);
begin
X[1]:=a; X[2]:=b; X[3]:=c; X[4]:=d; X[5]:=e;
end;
```

W zmienionym obiekcie zaplanowano, że publiczne (dostępne w programie głównym) są jedynie metody **UstawDane** i **ZaStyl**. Pozostałe składowe mogą być wykorzystywane tylko w treści innych metod obiektu.

Sprawdź, że po zmianach program nie da się uruchomić.

Popraw program zmieniając sposób ustalania not za skok (do zapisu wartości not w obiekcie typu **TNoty** trzeba wykorzystać metodę **UstawDane**) i przetestuj działanie zmienionego obiektu.