

Wydział: **Zarządzania i Modelowania Komputerowego**
Kierunek: **Inżynieria Danych**
Przedmiot: **Programowanie w środowisku RAD-C++**
Rok 2 Semestr 3

Ćwiczenie 5 - Proste obiekty

Przykład 1

Zdefiniuj typ (strukturę lub klasę) 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:

```
#include <conio.h>
#include <iostream.h>
struct TTrojkat
{
    float a, b, c;
    float obwod();
};
float TTrojkat::obwod()
{ return a+b+c; };
void main()
{
    TTrojkat t;
    cin >> t.a >> t.b >> t.c;
    cout << "Obwod=" << t.obwod();
    getch();
    return;
}
```

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

Zadanie 1

a) Rozbuduj definicję struktury **TTrojkat** dodając definicję metody **pole**. Wykorzystaj wzór Herona:

$$p = \text{obwod()}/2;$$
$$\text{pole} = \text{sqrt}((p-a)*(p-b)*\{p-c\}*p)$$

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 struktury obiektowej **TMieszkanie** grupującej 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 nowej klasy (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. Wypróbuj działanie programu przy zmianie specyfikatora dziedziczenia z public na private.

Zadanie 4

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

```
class TNoty
{
    public :
    float X[5];
    float ZaStyl();
    float Min();
    float Max();
    float Suma();
}
```

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

Napisz wersję programu z zadania 4. Zastosuj obiekt:

```
class TNoty
{
    public :
    void UstawDane(float a, float b, float c, float d, float e);
    float ZaStyl();
    private :
    float X[5];
    float Max();
    float Min();
    float Suma();
};
```

Dodaj zapis metody **UstawDane**:

```
void TNoty::UstawDane(float a, float b, float c, float d, float e)
{
    X[1]=a; X[2]=b; X[3]=c; X[4]=d; X[5]=e;
};
```

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.

Zadanie 6

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

punkty, **maxP** – ilość punktów które student dostał ze sprawdzianu oraz maksymalną ilość punktów z tego sprawdzianu

ocena – metoda wyznaczająca stopień za sprawdzian (powyżej 90% - ocena 5, powyżej 70% - ocena 4, 50% ocena 3, poniżej 50% - ocena 2)

Opracuj także typ **TPrzedmiot** dziedziczący składowe typu **TSprawdzian** oraz dodatkowo zawierający składowe:

maxPrzedmiot – maksymalna ilość punktów za wszystkie sprawdziany z przedmiotu.

pozostaje – metoda wyznaczająca ile punktów z pozostałych sprawdzianów musi uzyskać student by dostać ocenę trzy z przedmiotu (tj 50% punktów możliwych do uzyskania z wszystkich sprawdzianów)

Napisz program wykorzystujący powyższe typy obiektowe do obliczenia oceny ze sprawdzianu oraz ilości punktów które student musi jeszcze zdobyć żeby zaliczyć przedmiot na 3, na podstawie wczytywanych: punktów z pierwszego sprawdzianu, maksymalnej punktacji tego sprawdzianu oraz ilości punktów możliwych do uzyskania ze wszystkich sprawdzianów. Wyznacz także (bez definiowania dodatkowej funkcji składowej) jaki maksymalny stopień może jeszcze uzyskać student.