

Programowanie obiektowe C++

Proste obiekty, dziedziczenie

Proste obiekty, dziedziczenie

1. Przykład

Zdefiniuj typ obiektowy do obsługi obliczeń dla trójkąta. Klasa powinna zawierać:

- pola: **a**, **b** i **c** do zapamiętywania długości boków,
- metodę **obwod()** do wyznaczania obwodu trójkąta na podstawie długości boków.

```
#include <iostream>

using namespace std;

class TTrojkat
{
public:
    float a, b, c;
    float obwod();
};

float TTrojkat::obwod()
{
    return a + b + c;
};

int main()
{
    TTrojkat t;

    cin >> t.a >> t.b >> t.c;
    cout << "Obwod: " << t.obwod() << endl;

    return 0;
}
```

Wprowadź program i przetestuj jego działanie dla przykładowych danych.

2. Zadanie

- Rozbuduj definicję klasy **TTrojkat** o dodatkową metodę **pole()**. Do obliczeń wykorzystaj zależności:
 - $s = \frac{1}{2} \cdot o$, gdzie: o to obwód trójkąta,
 - $p = \sqrt{s \cdot (s - a) \cdot (s - b) \cdot (s - c)}$.
- Dokonaj dalszej rozbudowy klasy o dodatkowe funkcje obliczające wysokości trójkąta według zależności: $h = 2 \cdot \frac{p}{a}$, gdzie: h to wysokość (w tym wypadku dla boku a), a - długość boku.

3. Zadanie

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

- **metraz** - powierzchnia całego mieszkania,
- **liczbaIzb** - liczba izb w mieszkaniu,
- **powIzby()** - metoda wyznaczająca średnią powierzchnię izby w mieszkaniu.

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

Zadania dodatkowe

1. Rozbuduj program z zadania 3 definicję nowej klasy **TMieszkNowy** dziedziczącą z typu **TMieszkanie** oraz zawierającą dodatkowe składowe:

- **koszt** - cena sprzedaży całego mieszkania,
- **cenaMetra()** - metoda obliczająca średnią cenę jednego metra mieszkania.

```
class TMieszkNowy : public TMieszkanie
{
public:
    float koszt;
    float cenaMetra();
};
```

Program powinien wykonywać obliczenia jak w zadaniu 3 oraz dodatkowo wyświetlać cenę metra kwadratowego mieszkania na podstawie wczytywanego kosztu zakupu mieszkania.

2. Obiekt do obsługi wyznaczania oceny stylu skoku narciarskiego ma deklarację:

```
class TNoty
{
public:
    float oceny[5];
    float min();
    float max();
    float suma();
    float zaStyl();
}
```

Przeznaczenie poszczególnych obiektów:

- **oceny** - tablica ocen sędziów,
- **max()** - metoda zwracająca wartość oceny najwyższej,
- **min()** - metoda zwracająca wartość oceny najniższej,
- **suma()** - metoda zwracająca wartość sumy ocen,
- **zaStyl()** - metoda zwracająca ocenę stylu skoku równą sumie ocen sędziów pomniejszoną o wartość najwyższej i najniższej oceny.

3. Opracuj wersję programu z poprzedniego zadania z wykorzystaniem koncepcji interfejsu. Zastosuj obiekt o deklaracji:

```
class TNotyInt
{
```

```
public:
    void ustawOceny(float a, float b, float c, float d, float e);
    float zaStyl();

private:
    float oceny[5];
    float min();
    float max();
    float suma();
}
```

4. Opracuj deklarację typu obiektowego **TSprawdzian** o składowych:

- **punkty** - liczba punktów, które student uzyskał na sprawdzianie,
- **maxP** - maksymalna możliwa liczba punktów,
- **ocena()** - metoda wyznaczająca stopień za sprawdzian (0%÷49% - ocena 2, 50%÷59% - ocena 3, 60%÷69% - ocena 3.5, 70%÷79% - ocena 4, 80%÷89% - ocena 3.5, 90%÷100% - ocena 5).

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

- **maxPrzedmiot** - maksymalna liczba punktów za wszystkie sprawdziany z przedmiotu,
- **pozostaje()** - metoda wyznaczająca liczbę punktów z pozostałych sprawdzianów, które musi uzyskać student, żeby uzyskać zaliczenie.

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

5. Opracuj wersję zadania z poprzedniego punktu z wykorzystaniem **enkapsulacji** oraz metod typu **modyfikatory** i **akcesory**.