

Laboratorium Informatyki II

Ćwiczenie 6

1 Klasa

Klasa jest typem deklaratywnym o budowie zbliżonej do struktury, która poza danymi dostarcza także narzędzi (metod) do przetwarzania tych danych. Program obiektowy jest zbiorem obiektów wymieniających i przetwarzających informacje pomiędzy sobą.

W klasie definiuje się dane zwane atrybutami lub polami za pomocą których opisywany jest jego stan, oraz funkcji, procedur zwanych metodami które dostarczają narzędzi interfejsu do przetwarzania atrybutów obiektu.

Dostęp do poszczególnych atrybutów i metod obiektu jest precyzyjnie definiowany. W ramach projektowanej klasy definiuje się które z jej elementów są ogólnie dostępne, a do których dostęp jest ograniczony lub zabroniony. W klasie można określić trzy poziomy dostępu do jej elementów:

- **public** - publiczny, można odwołać się bezpośrednio do tych elementów obiektu z poziomu programu. W obszarze tym definiuje się interfejs klasy
- **private** - prywatny, dostęp do elementów z tej grupy możliwy tylko dla metod klasy. W obszarze tym deklaruje się atrybuty klasy (zazwyczaj) oraz metody wewnętrzne klasy.
- **protected** - chroniony, dostęp tylko dla metod klasy i klas dziedziczących na osobno określonych zasadach. W obszarze tym deklaruje się wszystkie elementy (atrybuty i metody) do których dostęp mają mieć klasy pochodne.

Poziom **private** jest traktowany jako domyślny i nie musi być formalnie deklarowany, ale ze względu na czytelność kodu zaleca się umieszczanie go w kodzie. Deklaracja trybu dostępu polega na umieszczeniu jego nazwy w kodzie klasy zakończonej dwukropkiem. Wszystko co zostanie umieszczone po zgłoszeniu trybu, aż do zgłoszenia innego trybu lub końca deklaracji klasy, będzie z nim związane. Zazwyczaj projektując klasę określa się elementy publiczne i prywatne, natomiast chronione tylko wtedy gdy dodatkowo projektujemy klasy dziedziczące po bazowej. Poniżej zapisano przykład programu obiektowego.

```
#include <iostream>
using namespace std;
class dane{
private:
    int a;
    float b;
public:
    void set(char *nazwa)
    {
```

```

        cout << "Podaj " << nazwa << ".a= "; cin >> a;
        cout << "Podaj " << nazwa << ".b= "; cin >> b;
    }
    int get_a(){
        return a;
    }
    float get_b(){
        return b;
    }
};
int main()
{
    dane X,Y;
    X.set("X");
    cout << "X.a= " << X.get_a() << " : X.b= " << X.get_b() << endl;
    cout << "Y.a= " << Y.get_a() << " : Y.b= " << Y.get_b() << endl;
    Y=X;
    cout << "Y.a= " << Y.get_a() << " : Y.b= " << Y.get_b() << endl;
}

```

2 Wskaźnik *this*

W pewnych sytuacjach zachodzi konieczność aby wewnątrz klasy odwołać się do atrybutów klasy które mają taką samą nazwę jak atrybuty metody. Wskaźnik *this* pozwala na precyzyjne odniesienie się do obiektu w kodzie obiektu. Poniżej pokazano kod programu obiektowego wykorzystującego wskaźnik *this*.

```

#include <iostream>
using namespace std;
class dane{
private:
    int a;
    float b;
public:
    void set(int a, float b)
    {
        this->a = a;
        this->b = b;
    }
    void draw()
    {
        cout << "a = " << a << " : b = " << b << endl;
    }
};
int main()
{
    dane X;
    X.set(5,13.23);
    X.draw();
}

```

Zadania

W oparciu o informacje wykładowe i treść niniejszej instrukcji zrealizować poniższe zadania:

- Zaprojektować klasę opisującą pojedynczy punkt wycieczki:
 - ID punktu
 - nazwę punktu
 - współrzędne punktu
 - wysokość nad poziomem morza
- Zaprogramować w projektowanej klasie metody:
 - pobieranie danych od użytkownika
 - wyświetlanie danych o punkcie
 - obliczanie odległości punktu od wskazanego punktu
 - obliczania różnicy wysokości pomiędzy punktem a wskazanym punktem
- Napisać program obiektowy z wykorzystaniem zaprojektowanej klasy zbierający dane o wycieczce góry, a następnie wyświetlający jej podsumowanie
- Program oprzeć na dynamicznej alokacji danych
- Zaimplementować możliwość określania pozycji wprowadzanego punktu wycieczki na liście punktów (linked list)