

Laboratorium Informatyki II

Ćwiczenie 7

1 Konstruktory klasy

Elementami składowymi każdej klasy są jej cechy i metody. Cechy umożliwiają przechowywanie danych opisujących tworzone na bazie klasy obiekty, a metody dostarczają narzędzi do przetwarzania tych informacji. W wielu sytuacjach projektując klasę wymagane może być aby w momencie tworzenia obiektu cechy klasy były inicjowane określonymi wartościami. Standard języka C++ nie dopuszcza inicjacji cech.

Inicjację cech można zrealizować za pomocą specjalnej metody zwanej **konstruktorem**. Jest ona automatycznie wywoływana wyłącznie w momencie tworzenia obiektu. Jej podstawowym zadaniem jest przypisanie wartości do cech klasy. Konstruktor charakteryzuje się tym że jest metodą bez określonego typu oraz ma nazwę taką samą jak nazwa klasy. Definiuje się go jako metodę publiczną. Konstruktor może być metodą bez lub z atrybutami wywołania. W klasie można zdefiniować dowolną liczbę konstruktorów, różniących się liczbą i typami atrybutów.

```
#include <iostream>

class dane{
private:
    int x;
    float y;
    std::string nazwa;
public:
    dane()
    {
        x=10;
        y=3.15;
        nazwa = "Domyślny";
    }
    dane(int x, float y, std::string nazwa)
    {
        this->x = x;
        this->y = y;
        this->nazwa = nazwa;
    }
    void pokaz()
    {
        std::cout << nazwa << "(" << x << "," << y << ")" << std::endl;
    }
};
```

```
int main()
{
    dane A,B(10,45.123,"Nazwa");
    A.pokaz();
    B.pokaz();
}
```

Dla konstruktora z atrybutami wejściowymi można przypisać do nich wartości domyślne. W kodzie klasy konstruktor taki zastępuje konstruktor bez atrybutowy. W kodzie można zdefiniować tylko jeden konstruktor z parametrami domyślnymi.

```
#include <iostream>
class dane{
private:
    int x;
    float y;
    std::string nazwa;
public:
    dane(int x=10, float y=3.14, std::string nazwa="Domyślny")
    {
        this->x = x;
        this->y = y;
        this->nazwa = nazwa;
    }
    void pokaz()
    {
        std::cout << nazwa << "(" << x << "," << y << ")" << std::endl;
    }
};
int main()
{
    dane A,B(10,45.123,"Nazwa");
    A.pokaz();
    B.pokaz();
}
```

2 Destruktor

Często w momencie kasowania obiektu zachodzi konieczność "posprzątania" po nim. Zazwyczaj taka sytuacja zachodzi gdy chociaż jedna z cech jest tworzona dynamicznie. W kodzie klasy w języku C++ można zdefiniować specjalną metodę wywoływaną w momencie kasowania obiektu. Metodę taką określa się mianem **destruktora**. W kodzie klasy może być tylko jedna taka metoda, musi być ona bezparametrowa. Destruktor ma nazwę taką samą jak klasa i konstruktor, ale jest poprzedzony znakiem "~".

```
#include <iostream>
class dane{
private:
    int *x,n;
    std::string nazwa;
```

```

public:
    dane(int n=5, std::string nazwa="Domyślny")
    {
        this->n = n;
        this->nazwa = nazwa;
        x = new int[n];
    }
    ~dane()
    {
        delete [] x;
        std::cout << "Obiekt <" << this << "> skasowany" << std::endl;
    }
    void pokaz()
    {
        std::cout << nazwa << " : ";
        for(int i=0;i<n;i++)
            std::cout << x[i] << " ";
        std::cout << std::endl;
    }
};
int main()
{
    dane A,B(10,"Nazwa");
    A.pokaz();
    B.pokaz();
}

```

Zadania

Napisać program obiektowy generujący dowolną liczbę zestawów o zadanej długości liczb losowych w określonym przedziale.

- klasa posiada kilka konstruktorów
- metody interfejsu:
 - wyświetlenie wszystkich zestawów obiektu lub jednego z wskazanych
 - posortowanie w określonej kolejności liczb w zestawach obiektu lub wskazanym zestawie
 - wygenerowanie liczb dla obiektu lub zestawu w obiekcie
- klasa posiada destruktory sprzątające po obiekcie