

Programowanie obiektowe w C++: definicja klasy, konstruktora, destruktor, tworzenie obiektów

PRZYKŁAD: DEFINICJA KLASY. TWORZENIE OBIEKTÓW

```
#include <iostream>
#include <string>
class Color { //początek definicji
    klasy private:
        int r;
        int g; // składniki klasy
        int b;
        char name[20];
    public:
        Color () { //definicja konstruktora bezparametrowego
            r=255; g=255; b=255;
            strcpy (name, "biały");
        }
        //definicja konstruktora z parametrami
        Color (int c_r, int c_g, int c_b, char *n){
            r=c_r; g=c_g; b=c_b;          strcpy (name,
            n);
        }
        ~Color() { //definicja destruktor
        cout<<"Destruktor";
        }
        //metoda
        void setColor(int c_r, int c_g, int c_b){
            r=c_r; g=c_g; b=c_b;
        }
        //metoda
        char* getName(){ return name;}
}; void
main() {
    //utworzenie obiektu o nazwie biały przy pomocy konstruktora bezparametrowego
    Color biały;
    Color nowy; //utworzenie obiektu o nazwie nowy
    //utworzenie obiektu o nazwie czarny przy pomocy konstruktora z parametrami
    Color czarny(0, 0, 0, "czarny");          cout<<czarny.getName(); //wywołanie
    metody getName() na rzecz obiektu czarny          nowy.setColor(12, 14, 34);
    //wywołanie metody setColor na rzecz obiektu //nowy }
```

Powyższy program został skonstruowany w taki sposób, że jego całość może zostać umieszczona w jednym pliku. Należy przerobić program w taki sposób, aby definicję klasy wraz z nagłówkami metod, konstruktorów i destruktor umieścić w pliku nagłówkowym (.h), a implementacje zadeklarowanych składników funkcjonalnych umieścić w osobnym pliku (.cpp)

ZADANIA

1. Opracować klasę Osoba, która zawiera zmienne:

 Znakowe imię[20] i nazwisko[20];

 liczbową wiek;

2. Opracować funkcje publiczne, klasy Osoba:

- konstruktor bezparametrowy;

```

Osoba();
- konstruktor (przeciążony) o następującym prototypie:
Osoba(char *i, char *n, int w);
- destruktor ~Osoba();
- funkcję wpisującą informacje o imieniu, nazwisku i wieku do odpowiednich składników void
  zapisz(char *i, char *n, int w);
- funkcję wypisującą na konsoli dane składowe (informacje o osobie) void wypisz();
- funkcję zmieniającą pojedyncze pola struktury
void zmien_imie(char *i); void
zmien_nazwisko (char *n) void
zmien_wiek(int w);

```

Przed rozpoczęciem wykonywania poleceń w kolejnych punktach instrukcji, dokładnie przetestować działanie poprzednich funkcji. Na przykład:

```

void main()
{
    Osoba Janek;

```

Przetestować funkcję wypisującą informacje o osobie i funkcje zmieniające dane osoby:

```

    Janek.zapisz("Jan", "Nowak",7);
    Janek.wypisz();
    Janek.zmien_imie("Marek");
    Janek.wypisz();
    Janek.zmien_nazwisko("Nowe");
}

```

3. Opracować funkcję globalną w programie, która będzie wywoływana z jednym argumentem będącym obiektem klasy *Osoba*. Zadaniem tej funkcji jest wyświetlenie na konsoli komunikat: "Przedstawiam państwu Szanownego Gościa: *Imię Nazwisko i Wiek*". W miejscu Imię i Nazwisko powinno zostać wyświetlone imię, nazwisko i wiek osoby będącej argumentem funkcji. Wypisanie danych o osobie odbywa się przy użyciu funkcji *wypisz()* będącej składnikiem klasy.

4. Stworzyć funkcję *powiel(Osoba)*, która kopiuje instancję klasy *Osoba*.

```
void powiel(Osoba ktos);
```

5. Dla klasy *Osoba* z poprzedniego zadania opracować przeciążony konstruktor kopiujący.

6. Zmodyfikuj program, tak aby umieszczał obiekty w tablicy.

7. Stwórz dwie klasy dziedziczące po klasie *Osoba*, o nazwach *Man* oraz *Woman*, niech zawierają płeć. Stwórz ich instancję.