

ZADANIA DO SAMODZIELNEGO ĆWICZENIA

Ćwiczenie 0

Uzupełnij poniższą klasę o pole statyczne "count" liczące ilość stworzonych obiektów tej klasy. Wywołanie poprawnego programu powinno wyświetlić "Liczba obiektów: 4"

```
#include <iostream>

class Klasa
{
private:
    double x;
public:
    Klasa(double a=0.0) : x(a) {}
};

//przykładowy main
int main()
{
    Klasa(3.);
    Klasa(6.);
    Klasa(1.);
    Klasa();
    std::cout<<"Liczba obiektów: "
    << Klasa::count << std::endl;
    return 0;
}
```

Ćwiczenie 1

Użyj `std::map`, żeby powiązać klucze – liczby typu `unsigned`, z wartościami – nazwami dni tygodnia. Użyj typu `std::string` do przechowywania nazw dni tygodnia. Dodaj w funkcji `main` wszystkie dni z odpowiednimi numerami a następnie je wypisz za pomocą pętli `for` i iteratorów.

Output:

```
1 poniedziałek
2 wtorek
3 sroda
4 czwartek
5 piątek
6 sobota
7 niedziela
```

Podpowiedź: Mapa jest w pliku nagłówkowym `<map>` a typ `string` w `<string>`. Obiekt typu `string` można utworzyć z łańcucha znaków w następujący sposób: `"std::string s("Napis");`

Ćwiczenie 2

Uzupełnij podaną niżej funkcję `main` tak, żeby używając `std::set` zachować informację jedynie o unikalnych elementach z wektora. Najpierw stwórz `set` korzystając z iteratorów wektora, później wypisz jego

elementy.

```
#include <set>
#include <vector>
#include <iostream>
using namespace std;

int main()
{
    std::vector<std::string> kraje =
        {"Hiszpania", "Holandia", "Portugalia", "Wlochy", "Wlochy",
         "Czechy", "Szwecja", "Hiszpania", "Niemcy", "Hiszpania",
         "Portugalia", "Niemcy", "Norwegia"};

    // tu stwórz set

    //a tu go wypisz
    return 0;
}
```

Jako output powinieneś dostać wszystkie nazwy krajów, ale bez powtórzeń.

Ćwiczenie 3

Zaprojektuj klasę Pracownik, która będzie mieć pola typu *std :: string* na imię i nazwisko. Napisz dla klasy konstruktor pozwalający ustawić oba pola. Dodatkowo napisz dwie metody:

```
void przedstawSie();
void funkcja();
```

Pierwsza z metod ma wyświetlić na cout napis "Jestem Adam Adamowicz", gdzie imię i nazwisko są wzięte z pól obiektu. Druga metoda ma wyświetlić na cout napis "Jestem szeregowym pracownikiem.". Druga metoda ma być wirtualna a pierwsza nie.

Dodatkowo napisz klasę Szef, która dziedziczy publicznie z klasy Pracownik. Napisz dla niej konstruktor i zdefiniuj metody przedstawSie i funkcja o takich samych sygnaturach jak w klasie Pracownik. W pierwszej z nich wypisz na cout napis "Jestem szefem i nazywam się Adam Adamowicz.", gdzie wstawisz odpowiednie imię i nazwisko. Druga metoda ma wypisać napis "Jestem szefem!".

Działanie obu klas przetestuj na załączonym mainie.

```
#include <vector>
#include <iostream>
#include <string>

using namespace std;

int main()
{
    Pracownik p1 = Pracownik("Jan", "Kowalski");
    Szef p2 = Szef("Andrzej", "Nowak");

    vector<Pracownik*> v = {&p1, &p2};
    for(auto val : v)
    {
```

```
        val->przedstawSie();  
        val->funkcja();  
    }  
  
    return 0;  
}
```

Output:

Jestem Jan Kowalski.

Jestem szeregowym pracownikiem.

Jestem Andrzej Nowak.

Jestem szefem!

Czy rozumiesz dlaczego output wygląda tak, a nie inaczej?