

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №1
з дисципліни «Об'єктно-орієнтоване програмування»

Тема:
«Основні поняття ООП. Класи та об'єкти. Функції доступу. Вказівник this»

Виконав: студент групи КБ-24мб

Воробйов Д.С.

Перевірив:

Козірова Н.Л.

Кропивницький 2024

Мета: ознайомитись з основними поняттями ООП. Вивчити поняття клас, об'єкт, сеттер, геттер та навчитись їх програмно реалізовувати мовою C++.

Завдання:

Завдання 1:

1. Створіть клас "Country" (країна), який має наступні властивості та функціональність:

Приватні поля класу:

- name (назва країни);
- capital (столиця країни);
- population (населення країни).

Публічні методи класу:

- метод setName(), який дозволяє задати назву країни;
- метод getName(), який повертає назву країни;
- метод setCapital(), який дозволяє задати столицю країни;
- метод getCapital(), який повертає столицю країни;
- метод setPopulation(), який дозволяє задати населення країни;
- метод getPopulation(), який повертає населення країни.

2. Створіть об'єкт класу "Country";

3. Використайте метод setName() для задання назви країни;

4. Використайте метод getName() для отримання назви країни та виведіть її на екран;

5. Використайте метод setCapital() для задання столиці країни;

6. Використайте метод getCapital() для отримання столиці країни та виведіть її на екран;

7. Використайте метод setPopulation() для задання населення країни;

8. Використайте метод getPopulation() для отримання населення країни та виведіть його на екран;

9. Реалізувати програму за допомогою роздільної компіляції.

Завдання 2

Реалізувати вище наведену задачу за допомогою структурного програмування.

У висновку описати різницю цих методів.

Результати (обробка даних, графіки, діаграми, таблиці)

Завдання 1:

Лістинг Country.h

```
#ifndef COUNTRY_H
#define COUNTRY_H
#include <string>

class Country {
public:
    void setName(std::string name);
    std::string getName();

    void setCapital(std::string capital);
    std::string getCapital();

    void setPopulation(int population);
    int getPopulation();
private:
    std::string name;
    std::string capital;
    int population;
};

#endif
```

Лістинг Country.cpp

```
#include "Country.h"

void Country::setName(std::string name) {
    this->name = name;
}

std::string Country::getName() {
    return name;
}

void Country::setCapital(std::string capital) {
    this->capital = capital;
}
```

```

}
std::string Country::getCapital() {
    return capital;
}
void Country::setPopulation(int population) {
    this->population = population;
}
int Country::getPopulation() {
    return population;
}

```

Лістинг main.cpp

```

#include <iostream>
#include "Country.h"

using namespace std;

int main() {
    Country UKR;
    UKR.setName("Ukraine");
    UKR.setCapital("Kyiv");
    UKR.setPopulation(41258478);
    cout << "Country: " << UKR.getName() << endl;
    cout << "Capital: " << UKR.getCapital() << endl;
    cout << "Population: " << UKR.getPopulation() << endl;
    return 0;
}

```

```

Country: Ukraine
Capital: Kyiv
Population: 41258478

```

Рисунок 1 – Результат роботи програми

Завдання 2

Лістинг main.cpp

```

#include <iostream>
#include <string>

```

```
using namespace std;

struct Country {
    string name;
    string capital;
    int population;
};

void setName(Country& country, string countryName) {
    country.name = countryName;
}

string getName(Country& country) {
    return country.name;
}

void setCapital(Country& country, string countryCapital) {
    country.capital = countryCapital;
}

string getCapital(Country& country) {
    return country.capital;
}

void setPopulation(Country& country, int countryPopulation) {
    country.population = countryPopulation;
}

int getPopulation(Country& country) {
    return country.population;
}

int main() {
    Country UKR;

    setName(UKR, "Ukraine");
    setCapital(UKR, "Kyiv");
}
```

```
setPopulation(UKR, 41258478);

cout << "Country: " << getName(UKR) << endl;
cout << "Capital: " << getCapital(UKR) << endl;
cout << "Population: " << getPopulation(UKR) << endl;

return 0;
}

Country: Ukraine
Capital: Kyiv
Population: 41258478
```

Рисунок 2 – Результат роботи програми

Висновки: різниця між об'єктно-орієнтованим програмуванням та структурним програмуванням полягає в тому, що об'єктно-орієнтоване програмування забезпечує інкапсуляцію для захисту даних від прямого доступу, а також використовує спадковість та поліморфізм для простішого доповнення або зміни коду. У структурному програмуванні, навпаки, дані доступні на пряму, даний метод є менш гнучким що не підходить для складних проєктів.