

Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Інститут телекомунікацій, радіоелектроніки та електронної техніки



Звіт до лабораторної роботи №9
З дисципліни : «Об’єктно орієнтоване програмування»
На тему: « Конструювання класів з використанням дружніх функцій »
Варіант №3

Виконав:
Ст.гр АП-21
Братейко Вадим

Прийняв:
Алтунін С.І

Львів - 2024

Мета роботи: створити, відлагодити та протестувати програму, у якій створити базовий та похідні класи з використанням дружніх функцій для обробки Даних.

Хід роботи:

Варіант завдань:

3	Місто	Промисловий центр, культурний центр
---	-------	-------------------------------------

Код програми.

Main.cpp

```
#include "IndustrialCity.h"
```

```
#include "CulturalCity.h"
```

```
#include "CityManager.h"
```

```
#include <vector>
```

```
int main() {
```

```
    std::vector<City*> cities;
```

```
    // Додати промислове місто
```

```
    IndustrialCity* industrialCity = new IndustrialCity();
```

```
    industrialCity->inputData();
```

```
    cities.push_back(industrialCity);
```

```
    // Додати культурне місто
```

```
    CulturalCity* culturalCity = new CulturalCity();
```

```
    culturalCity->inputData();
```

```
    cities.push_back(culturalCity);
```

```
    // Вивід даних через дружній клас
```

```
    CityManager::displayAllCities(cities);
```

```

// Очистити пам'ять
for (auto& city : cities) {
    delete city;
}

return 0;
}
City.h

#ifndef CITY_H
#define CITY_H

#include <string>
#include <iostream>

class CityManager; // Оголошення дружнього класу

class City {
protected:
    std::string name;
    int population;

public:
    City(const std::string& name = "", int population = 0);
    virtual ~City();

    virtual void displayData() const;
    virtual void inputData();

    // Дружній клас
    friend class CityManager;
};

```

```
#endif
```

```
City.cpp
```

```
#include "City.h"
```

```
// Конструктор
```

```
City::City(const std::string& name, int population) : name(name),  
population(population) {}
```

```
// Деструктор
```

```
City::~City() {}
```

```
// Виведення даних
```

```
void City::displayData() const {  
    std::cout << "Місто: " << name << ", Населення: " << population << std::endl;  
}
```

```
// Введення даних
```

```
void City::inputData() {  
    std::cout << "Введіть назву міста: ";  
    std::cin >> name;  
    std::cout << "Введіть населення: ";  
    std::cin >> population;  
}
```

```
IndustrialCity.h
```

```
#ifndef INDUSTRIALCITY_H
```

```
#define INDUSTRIALCITY_H
```

```
#include "City.h"
```

```
class IndustrialCity : public City {
```

```
private:
```

```
    int factories;
```

```
public:
```

```
    IndustrialCity(const std::string& name = "", int population = 0, int factories = 0);
```

```
    ~IndustrialCity();
```

```
void displayData() const override;  
void inputData() override;
```

```
friend class CityManager; // Дружній клас  
};
```

```
#endif
```

```
IndustrialCity.cpp
```

```
#include "IndustrialCity.h"
```

```
// Конструктор
```

```
IndustrialCity::IndustrialCity(const std::string& name, int population, int factories)  
    : City(name, population), factories(factories) { }
```

```
// Деструктор
```

```
IndustrialCity::~IndustrialCity() { }
```

```
// Виведення даних
```

```
void IndustrialCity::displayData() const {  
    City::displayData();  
    std::cout << "Кількість заводів: " << factories << std::endl;  
}
```

```
// Введення даних
```

```
void IndustrialCity::inputData() {  
    City::inputData();  
    std::cout << "Введіть кількість заводів: ";  
    std::cin >> factories;  
}
```

```
CulturalCity.h
```

```

#ifndef CULTURALCITY_H
#define CULTURALCITY_H

#include "City.h"

class CulturalCity : public City {
private:
    int museums;

public:
    CulturalCity(const std::string& name = "", int population = 0, int museums = 0);
    ~CulturalCity();

    void displayData() const override;
    void inputData() override;

    friend class CityManager; // Дружній клас
};

#endif

CulturalCity.cpp

#include "CulturalCity.h"

// Конструктор
CulturalCity::CulturalCity(const std::string& name, int population, int museums)
    : City(name, population), museums(museums) { }

// Деструктор
CulturalCity::~~CulturalCity() { }

// Виведення даних
void CulturalCity::displayData() const {
    City::displayData();
    std::cout << "Кількість музеїв: " << museums << std::endl;
}

```

```

// Введення даних
void CulturalCity::inputData() {
    City::inputData();
    std::cout << "Введіть кількість музеїв: ";
    std::cin >> museums;
}
CityManager.h
#ifndef CITYMANAGER_H
#define CITYMANAGER_H

#include "City.h"
#include "IndustrialCity.h"
#include "CulturalCity.h"
#include <vector>

class CityManager {
public:
    static void displayAllCities(const std::vector<City*>& cities);
};

#endif
CityManager.cpp
#include "CityManager.h"

// Вивід усіх міст
void CityManager::displayAllCities(const std::vector<City*>& cities) {
    std::cout << "\nДані про міста:\n";
    for (const auto& city : cities) {
        city->displayData();
    }
}

```

Результати тестування

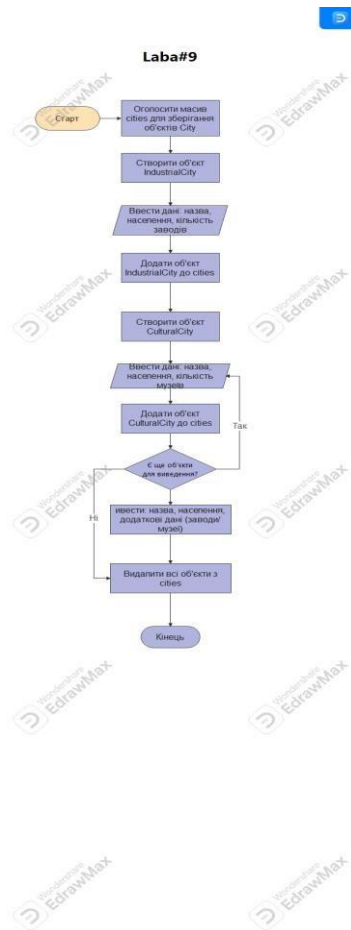
```

Введіть назву міста: Львів
Введіть населення: 610000
Введіть кількість заводів: 200
Введіть назву міста: Київ
Введіть населення: 1000000
Введіть кількість музеїв: 25

Дані про міста:
Місто: Львів, Населення: 610000
Кількість заводів: 200
Місто: Київ, Населення: 1000000
Кількість музеїв: 25

```

Блок-схема алгоритму програми:



Висновок: У лабораторній роботі було створено базовий та похідний класи з використанням дружніх функцій для доступу до приватних даних класів. Це дозволило спростити взаємодію між класами та підвищити ефективність обробки даних. Програма успішно скомпільована і протестована.