

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 <vector>
#include <memory>

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);

    // Вивести всі дані

    std::cout << "\nДані про міста:\n";
    for (const auto& city : cities) {
        city->displayData();
    }
}
```

```

// Очистити пам'ять

for (auto& city : cities) {

    delete city;

}

return 0;

}

City.h

#ifndef CITY_H
#define CITY_H
#include <string>
#include <iostream>
class City {
protected:
    std::string name;
    int population;

public:

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

    virtual void displayData() const; // Віртуальний метод для виведення даних
    virtual void inputData(); // Віртуальний метод для введення даних

    // Гетери

    std::string getName() const;
    int getPopulation() const;

    // Сетери

    void setName(const std::string& name);
    void setPopulation(int population);
};

```

```
#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;  
}
```

```
// Гетери
```

```
std::string City::getName() const { return name; }  
int City::getPopulation() const { return population; }
```

```
// Сетери
```

```
void City::setName(const std::string& name) { this->name = name; }  
void City::setPopulation(int population) { this->population = population; }
```

```
IndustrialCity.h
```

```
#ifndef INDUSTRIALCITY_H
```

```

#define INDUSTRIALCITY_H
#include "City.h"
class IndustrialCity : public City {

private:
    int factories; // Кількість заводів

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

    void displayData() const override; // Перевизначений метод
    void inputData() override; // Перевизначений метод

    // Додаткові методи
    int getFactories() const;
    void setFactories(int factories);
};

#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;
}

// Методи
int IndustrialCity::getFactories() const { return factories; }
void IndustrialCity::setFactories(int factories) { this->factories = factories; }

```

CulturalCity.h

```

#ifndef CULTURALECITY_H
#define CULTURALECITY_H
#include "City.h"

class CulturalCity : public City {
private:
    int museums; // Кількість музеїв
public:

    CulturalCity(const std::string& name = "", int population = 0, int museums =
0);
    ~CulturalCity();
    void displayData() const override; // Перевизначений метод
    void inputData() override; // Перевизначений метод
    // Додаткові методи
    int getMuseums() const;
    void setMuseums(int museums);
};

#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;
}
```

```
// Методи
```

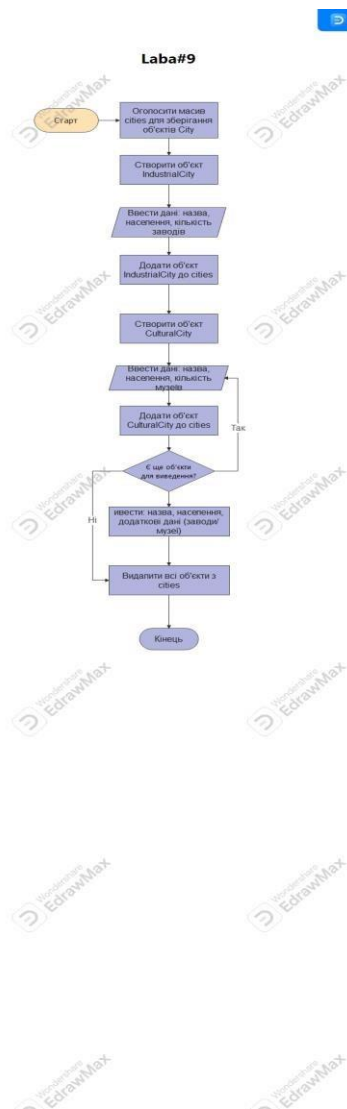
```
int CulturalCity::getMuseums() const { return museums; }
void CulturalCity::setMuseums(int museums) { this->museums = museums; }
```

Результати тестування

```
Введіть назву міста: Lviv
Введіть населенн?: 50000
Введіть кількіст? заводів: 34
Введіть назву міста: Kyiv
Введіть населенн?: 800000
Введіть кількіст? музеїв: 24

Дані про міста:
Місто: Lviv, Населення: 50000
Кількість заводів: 34
Місто: Kyiv, Населення: 800000
Кількість музеїв: 24
```

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.