# Міністерство освіти і науки України

## Національний університет "Львівська політехніка"

Інститут телекомунікацій, радіоелектроніки та електронної техніки



Звіт до лабораторної роботи №8

3 дисципліни : «Об'єктно орієнтоване програмування»

На тему: « Конструювання базових та похідних класів»

Варіант: 3

Виконав:

Ст.гр АП-21

Братейко Вадим

Прийняв:

Алтунін С.І

**Мета роботи:** створити, відлагодити та протестувати програму, у якій створити базовий та похідні класи для обробки даних.

### Хід роботи:

#### Варіант завдань:

	Батьківський клас	Дочірні класи
3	Місто	Промисловий центр, культурний центр

#### Код програми.

```
Main.cpp
#include <iostream>
#include <vector>
#include "IndustrialCity.h"
#include "CulturalCity.h"
int main() {
  vector<City*> cities;
   cities.push_back(new IndustrialCity("Kharkiv", 1500000)); cities.push_back(new CulturalCity("Lviv", 800000));
   for (City* city: cities) {
  city->displayData();
   for (City* city : cities) {
   delete city;
   return 0;
Sity.h
#ifndef CITY_H
#define CITY_H
#include <string>
using namespace std;
class City {
public:
   City(string name, int population);
   virtual ~City();
```

```
virtual void inputData();
  virtual void displayData() const;
protected:
  string name;
  int population;
};
#endif// CITY_H
Sity.cpp
#include "City.h"
#include <iostream>
City::City(string name, int population): name(name), population(population) {}
City::~City() { }
void City::inputData() {
  cout << "Enter city name: ";</pre>
  cin >> name;
  cout << "Enter population: ";</pre>
  cin >> population;
  if (population < 0) {
     cout << "Population cannot be negative. Setting to 0." << endl;
    population = 0;
  }
}
```

```
void City::displayData() const {
  cout << "City: " << name << ", Population: " << population << endl;
}
CulturalCity.h
#ifndef CULTURALCITY_H
#define CULTURALCITY_H
#include "City.h"
class CulturalCity : public City {
public:
  CulturalCity(string name, int population);
  void displayData() const override;
};
#endif// CULTURALCITY_H
CulturalCity.cpp
#include "CulturalCity.h"
#include < iostream>
Cultural City:: Cultural City(string name, int population): City(name, population) \left\{\right.\right\}
void CulturalCity::displayData() const {
  cout << "Cultural City - ";</pre>
  City::displayData();
}
IndustrialCity.h
#ifndef INDUSTRIALCITY_H
#define INDUSTRIALCITY_H
#include "City.h"
```

```
class IndustrialCity: public City {
public:
    IndustrialCity(string name, int population);
    void displayData() const override;
};

#endif// INDUSTRIALCITY_H
IndustrialCity.cpp
#include "IndustrialCity.h"
#include <iostream>

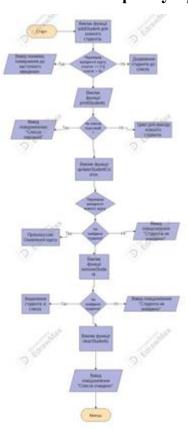
IndustrialCity::IndustrialCity(string name, int population): City(name, population) { }

void IndustrialCity::displayData() const {
    cout << "Industrial City - ";
    City::displayData();
}</pre>
```

Результати тестування для різних вхідних умов.

```
Industrial City - City: Kharkiv, Population: 1500000
Cultural City - City: Lviv, Population: 800000
```

#### Блок-схема алгоритму програми:



**Висновок:** У ході виконання лабораторної роботи було створено клас для обробки даних, який містить динамічний масив структур. Було реалізовано механізми додавання, видалення та обробки даних у масиві, а також проведено тестування функціональності класу. Робота дозволила закріпити знання з роботи з динамічною пам'яттю, створення структурованих типів даних та об'єктно-орієнтованого програмування.