

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



Звіт до лабораторної роботи №12
З дисципліни : «Об’єктно орієнтоване програмування»
На тему: « Робота з файлами та винятками »
Варіант №3

Виконав:
Ст.гр. АП-21
Братейко Вадим

Прийняв:
Алтунін С.І.

Мета роботи: створити, відлагодити та протестувати програму, у якій реалізувати двонаправлений файловий ввід/вивід для обробки даних з використанням обробників винятків.

Хід роботи:

Варіант завдань:

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

Код програми.

```
#include <iostream>
#include <fstream>
#include <string>
#include <stdexcept>
#include <cstdlib> // Для роботи з файлом

using namespace std;

// Батьківський клас "Місто"
class City {
protected:
    string name; // Назва міста
    int population; // Населення

public:
    // Конструктор
    City(string n, int p) : name(n), population(p) {}

    // Віртуальна функція для виведення даних
    virtual void display() const {
        cout << "Місто: " << name << ", Населення: " << population << endl;
    }

    // Деструктор
    virtual ~City() {}

    // Перевантаження операторів для порівняння міст
    bool operator==(const City& other) const {
        return name == other.name && population == other.population;
    }

    bool operator!=(const City& other) const {
        return !(*this == other);
    }

    // Перевантаження оператора для виведення інформації про місто
    friend ostream& operator<<(ostream& os, const City& city);
};

ostream& operator<<(ostream& os, const City& city) {
    os << city.name << " (" << city.population << " осіб)";
    return os;
}

// Дочірній клас "Промисловий центр"
class IndustrialCenter : public City {
private:
    int factories; // Кількість фабрик

public:
    // Конструктор
```

```

IndustrialCenter(string n, int p, int f) : City(n, p), factories(f) {}

// Переозначення функції display для промислового центру
void display() const override {
    City::display();
    cout << "Кількість фабрик: " << factories << endl;
}

// Дочірній клас "Культурний центр"
class CulturalCenter : public City {
private:
    int theaters; // Кількість театрів
    int museums; // Кількість музеїв

public:
    // Конструктор
    CulturalCenter(string n, int p, int t, int m) : City(n, p), theaters(t), museums(m) {}

    // Переозначення функції display для культурного центру
    void display() const override {
        City::display();
        cout << "Кількість театрів: " << theaters << ", Кількість музеїв: " <<
museums << endl;
    }
};

// Функція для запису в файл
void writeToFile(const string& filename) {
    ofstream outFile(filename);
    if (!outFile) {
        throw ios_base::failure("Помилка при відкритті файлу для запису.");
    }

    // Записуємо дані про реальні міста
    outFile << "Дані про міста:\n";
    outFile << "Місто: Київ, Населення: 2967360, Кількість фабрик: 150,
Кількість театрів: 15, Кількість музеїв: 20\n";
    outFile << "Місто: Харків, Населення: 1431320, Кількість фабрик: 120,
Кількість театрів: 12, Кількість музеїв: 18\n";

    outFile.close();
}

// Функція для читання з файлу
void readFromFile(const string& filename) {
    ifstream inFile(filename);
    if (!inFile) {
        throw ios_base::failure("Помилка при відкритті файлу для читання.");
    }

    string line;
    while (getline(inFile, line)) {
        cout << line << endl;
    }

    inFile.close();
}

// Функція для перевірки коректності введеного населення
void checkValidPopulation(int population) {
    if (population <= 0) {
        throw invalid_argument("Населення має бути позитивним числом.");
    }
}

// Головна функція програми
int main() {
    try {
        // Введення даних для міста

```

```

string cityName;
int population;

cout << "Введіть назву міста: ";
getline(cin, cityName);
cout << "Введіть населення: ";
cin >> population;
cin.ignore(); // Очищення буферу вводу

// Перевірка валідності населення
checkValidPopulation(population);

// Створення об'єкта для міста
City city(cityName, population);
city.display();

// Створення об'єктів для промислових і культурних центрів
IndustrialCenter industrial("Харків", 1431320, 120); // Приклад для Харкова
CulturalCenter cultural("Київ", 2967360, 15, 20); // Приклад для Києва

// Виведення даних для обох міст
industrial.display();
cultural.display();

// Порівняння міст
if (industrial == cultural) {
    cout << "Ці міста однакові!" << endl;
} else {
    cout << "Ці міста різні!" << endl;
}

// Запис в файл
writeToFile("cities.txt");

// Зчитування з файлу
readFromFile("cities.txt");
} catch (const exception& ex) {
    cout << "Помилка: " << ex.what() << endl;
}

return 0;
}

```

Результати тестування:

```

Введіть назву міста: Львів
Введіть населення: 5426565
Місто: Львів, Населення: 5426565
Місто: Харків, Населення: 1431320
Кількість фабрик: 120
Місто: Київ, Населення: 2967360
Кількість театрів: 15, Кількість музеїв: 20
Ці міста різні!
Дані про міста:
Місто: Київ, Населення: 2967360, Кількість фабрик: 150, Кількість театрів: 15, Кількість музеїв: 20
Місто: Харків, Населення: 1431320, Кількість фабрик: 120, Кількість театрів: 12, Кількість музеїв: 18

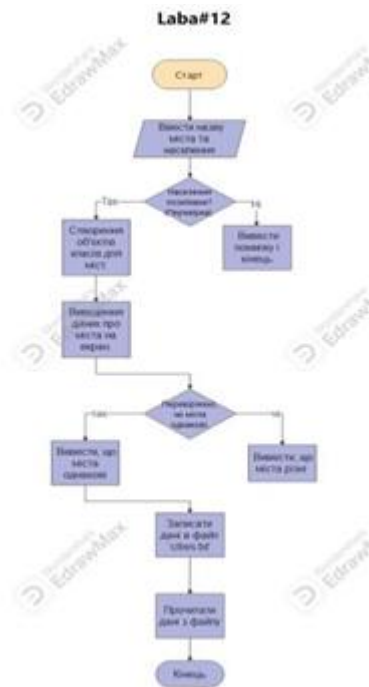
```

```

Дані про міста:
Місто: Київ, Населення: 2967360, Кількість фабрик: 150, Кількість театрів: 15, Кількість музеїв: 20
Місто: Харків, Населення: 1431320, Кількість фабрик: 120, Кількість театрів: 12, Кількість музеїв: 1

```

Блок-схема алгоритму програми:



Висновок: У ході виконання лабораторної роботи було реалізовано програму, яка використовує двонаправлений файловий ввід/вивід для обробки даних про міста. Програма зчитує інформацію з файлу, обробляє її, а також записує результати назад у файл, забезпечуючи коректну обробку помилок через використання обробників винятків.

