Міністерство освіти і науки

Національний університет “Львівська політехніка”

**Кафедра ЕОМ**



**Звіт**

з лабораторної роботи № 2

# з дисципліни: “ Програмування, частина 2 (Об'єктно-орієнтоване програмування”

на тему: “ ПОТОКОВИЙ ВВІД-ВИВІД ”

Виконав: ст. гр. КІ-15 Блищак Р.С.

Прийняв: Козак Н. Б.

Львів – 2020

Мета: познайомитися із потоковим вводом-виводом.

ТЕОРЕТИЧНІ ВІДОМОСТІ

**Потоки вводу-виводу в С++**

Система вводу-виводу в стандартній бібліотеці С++ реалізована у вигляді потоків. Потік вводу-виводу – це логічний пристрій, який приймає та видає інформацію користувача. Кожен потік зв‘язаний з фізичним пристроєм (клавіатура, монітор) або з файлом. Бібліотека потоків iostream реалізована 20 як ієрархія класів та забезпечує широкі можливості для виконання операцій вводу-виводу. Далі наведено призначення деяких класів потокового вводу-виводу:

• istream – підтримує операції по вводу;

• ostream – підтримує операції по виводу;

• iostream – підтримує операції по вводу-виводу;

• іfstream – підтримує операції по вводу з файлу;

• ofstream – підтримує операції по виводу у файл;

• fstream – підтримує операції з файлами по вводу-виводу.

**Файловий ввід-вивід**

Робота з файлами в мові С++ як і у мові С передбачає 3 етапи: відкривання файлу (файлового потоку), обмін даними з файловим потоком, закривання файлового потоку. Для виконання операцій з файлами в мові С++ передбачено три класи: іfstream, ofstream і fstream. Ці класи є похідними від класів іstream, ostream і іostream. Всі функціональні можливості (перевантажені операції << та >> для вбудованих типів, функції і прапорці форматування, маніпулятори й ін.), що застосовуються до стандартного вводу та виводу, можуть застосовуватися і до файлів. Існує деяка відмінність між використанням стандартних та файлових потоків. Стандартні потоки можуть використовуватися відразу після запуску програми, тоді як файловий потік спочатку слід зв'язати з файлом. Для реалізації файлового вводу-виводу потрібно підключити заголовочний файл fstream, що знаходиться в просторі імен std. Режими відкриття файлу являють собою бітові маски, тому можна задавати два або більш режими, поєднуючи їх побітовою операцією АБО. Слід звернути увагу, що по замовчуванню режим відкриття файлу відповідає типові файлового потоку. У потоці вводу або виводу прапорець режиму завжди встановлений неявно. Між режимами відкриття файлу іos::ate та іos::app існує певна відмінність. Якщо файл відкривається в режимі додавання (іos::app), весь вивід у файл буде здійснюватися в позицію, що починається з поточного кінця файлу, безвідносно до операцій позиціонування у файлі. У режимі відкриття іos::ate (від англійського "at end") можна змінити позицію виводу у файл і здійснювати запис, починаючи з неї. Файли, які відкриваються для виводу, створюються, якщо вони ще не існують. Якщо при відкритті файлу не зазначений режим іos::bіnary, файл відкривається в текстовому режимі. Якщо відкриття файлу завершилося невдачею, об'єкт, що відповідає потокові, буде повертати нуль. Перевірити успішність відкриття файлу можна також за допомогою функції-члена іs\_open(). Дана функція повертає 1, якщо потік вдалося зв'язати з відкритим файлом. Для перевірки, чи досягнутий кінець файлу, можна використовувати функцію eof(). Завершивши операції вводу-виводу, необхідно закрити файл, викликавши функцію-член close(). Далі наведений приклад, що демонструє файловий ввід-вивід з використанням потоків.

**Завдання:**

3. З клавіатури вводиться ім‘я, адреса, вартість і рік

заснування компанії.

**Виконання завдання:**

Код програми:

#include<iostream>

#include<fstream>

#include<string>

#include<iomanip>

using namespace std;

int main()

{

while (true)

{

setlocale(LC\_ALL, "ukr");

cout << "Показати данi - натиснiть 1" << endl;

cout << "Записати данi - натиснiть 2" << endl;

cout << "Вихiд - натиснiть 3" << endl;

int choice;

cin >> choice;

if (choice == 1)

{

string name;

string address;

int cost;

int year;

ifstream infile;

infile.open("File.txt");

if (!infile)

{

cout << "Не можна вiдкрити файл" << endl;

return -1;

}

cout << setw (9)<< "Iмя" << setw(10) << "Адреса" << setw(13) << "Вартiсть" << setw(8) << "Рiк" << endl;

while (!infile.eof())

{

infile >> name;

infile >> address;

infile >> cost;

infile >> year;

if (!infile.eof())

{

cout.width(9);

cout << name;

cout.width(10);

cout << address;

cout.width(12);

cout << cost;

cout.width(9);

cout << year;

cout << endl;

}

}

infile.close();

}

if (choice == 2)

{

string name;

string address;

int cost;

int year;

cout << "Iмя: ";

cin >> name;

cout << "Адреса: ";

cin >> address;

cout << "Вартiсть: ";

cin >> cost;

cout << "Рiк: ";

cin >> year;

fstream outfile("File.txt", ios::app);

if (!outfile)

{

cout << "Не можна вiдкрити файл" << endl;

return -1;

}

outfile.setf(ios::left);

outfile.width(9);

outfile << name << ' ';

outfile.width(9);

outfile << address << ' ';

outfile.width(9);

outfile << cost << ' ';

outfile.width(9);

outfile << year << endl;

outfile.close();

}

if (choice == 3)

{

break;

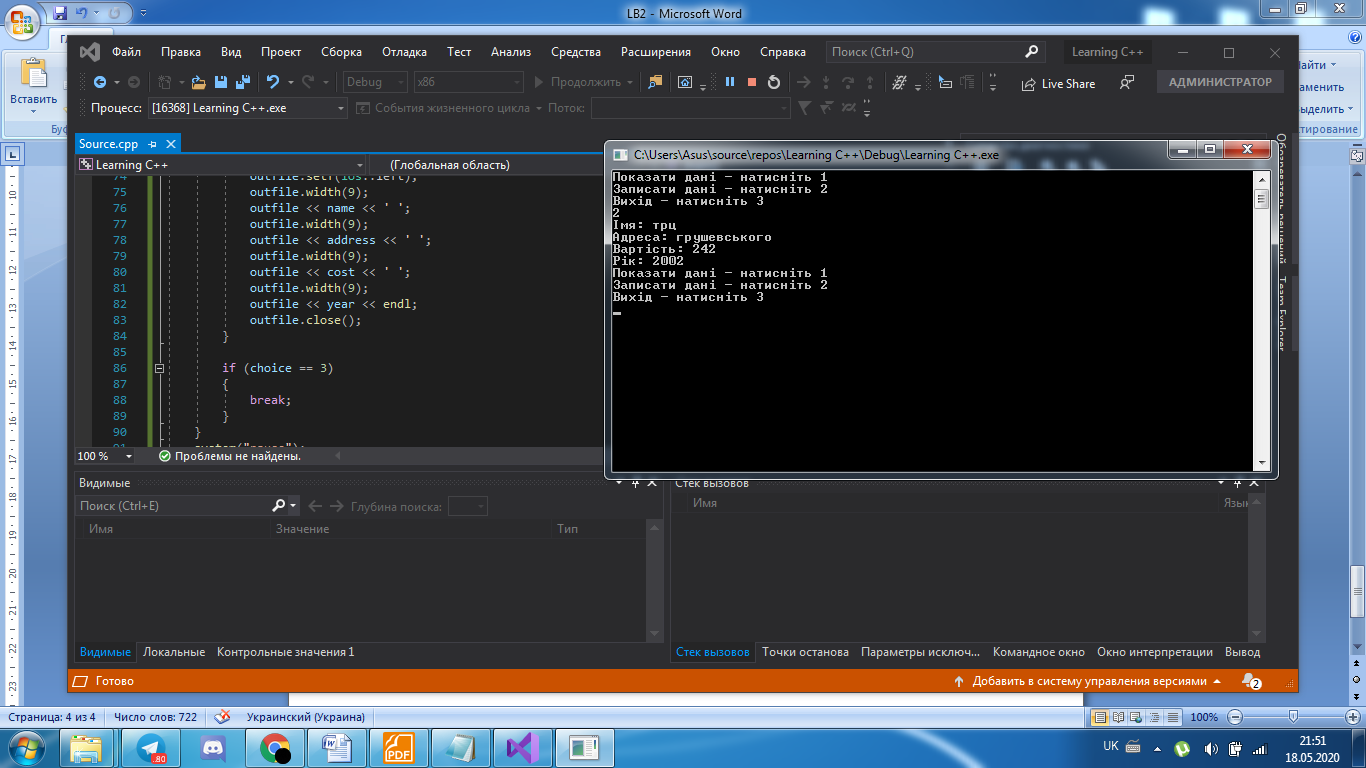
}

}

system("pause");

}

Результат роботи програми:



Висновок: на цій лабораторній роботі я познайомився із потоковим вводом-виводом.