Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра “Електронних обчислювальних машин”



**Звіт з лабораторної роботи №2**

**Програмування, частина 2**

**(Об’єктно - орієнтоване програмування)**

на тему:

“Потоковий ввід-вивід ”

Варіант - 25

**Виконала:**

ст. гр. КІ-15

Романішина А. Ю.

**Перевірив:**

асист. каф. ЕОМ

Козак Н. Б.

Львів - 2020

**Мета:** познайомитися із потоковим вводом-виводом.

**Теоретичні відомості:**

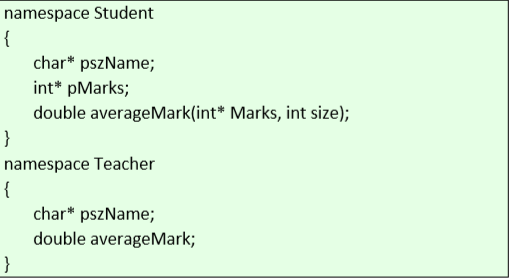
*Простори імен:*

У мові програмування С++ так само як і у багатьох об‘єктноорієнтованих мовах програмування існує поняття просторів імен, що розділяються, на відміну від мови С в якій існує один глобальний простір імен. Простори імен служать для об‘єднання класів, що написані різними програмістами або мають схоже призначення в групи. Завдяки цьому з‘являється можливість усувати неоднозначності, що пов‘язані з використанням класів, що мають однакові імена, але різне призначення.

Також простори імен дозволяють безпечно підключати бібліотеки класів не боячись співпадінь у назвах існуючих і підключених класів.

Все, що оголошене в межах простору імен є видимим лише в його межах. Для оголошення простору імен слід вжити ключове слово namespace після якого слід вказати назву простору імен. Після цього в фігурних дужках визначається його тіло. Тіло простору імен може містити як класи і структури, так і функції зі змінними та константами, а також вкладені простори імен.

Приклад просторів імен:



Простори імен є відкритими. Це означає, що можна в будь-якому місці програми оголосити простір імен з існуючою назвою, додати в нього нові елементи і при компіляції ці простори імен об‘єднаються в один, що міститиме існуючі і нововведені елементи. Простір імен може бути безіменним, якщо його ім‘я при оголошенні не вказується. Безіменні простори імен можна використовувати для оголошення глобальних статичних змінних з внутрішнім зв‘язуванням, оскільки згідно стандарту ключове слово static є застарілим. Для того, щоб звернутися до елементу з простору імен слід вказати назву простору імен, оператор розширення області видимості (::) та здійснити відповідне звернення до елементу простору імен:

**Student :: averageMark(arr, size);**

**Teacher :: averageMark = 5.2;**

Щоб постійно не вказувати простори імен при зверненні до їх елементів, їх можна підключити до програми за допомогою ключового слова using частково або повністю. Для підключення окремих елементів простору імен використовується using оголошення. Для підключення всього простору імен з використовується using директива. Синтаксис підключення окремих елементів простору імен з використанням using оголошення:

**using назваПросторуІмен :: назваЕлементу**

*Потоки вводу-виводу в С++:*

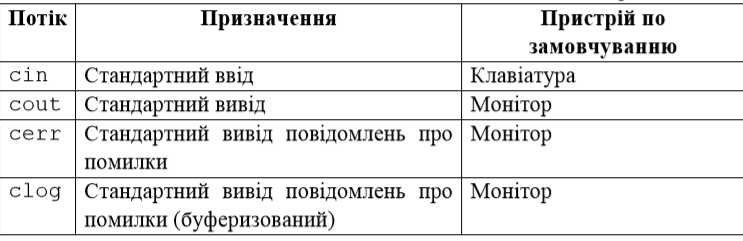
Система вводу-виводу в стандартній бібліотеці С++ реалізована у вигляді потоків. Потік вводу-виводу – це логічний пристрій, який приймає та видає інформацію користувача. Кожен потік зв‘язаний з фізичним пристроєм (клавіатура, монітор) або з файлом. Бібліотека потоків iostream реалізована

як ієрархія класів та забезпечує широкі можливості для виконання операцій вводу-виводу. Далі наведено призначення деяких класів потокового вводувиводу:

* 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

*Стандартні потоки:*

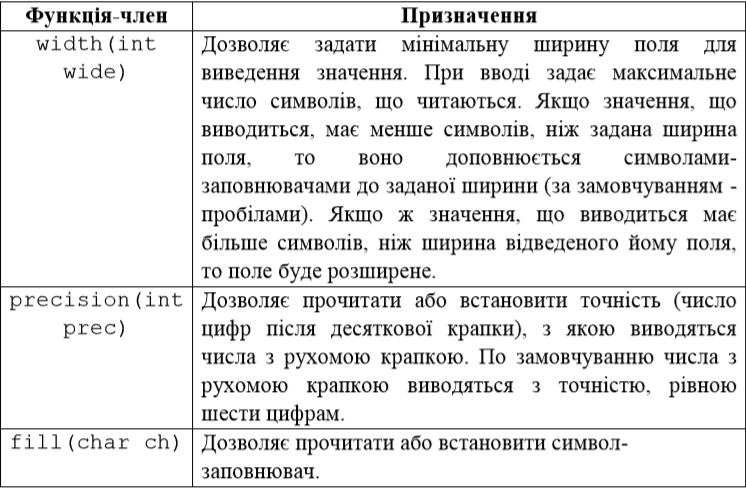
Коли запускається програма на С++, автоматично створюються чотири об‘єкти, що реалізують стандартні потоки.



Щоб мати можливість використовувати стандартні потоки необхідно підключити заголовочний файл iostream або iostream.h

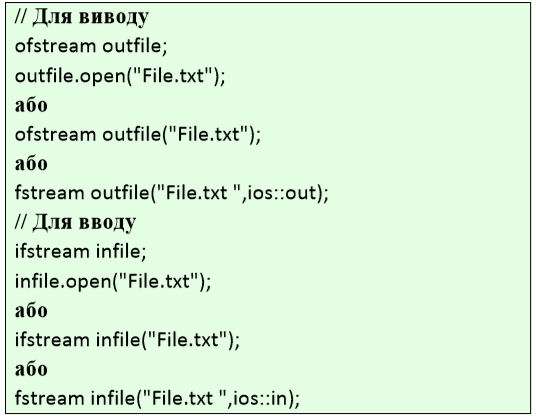
*Засоби форматування потоку*

Система вводу-виводу дозволяє виконувати форматування даних та змінювати визначені параметри вводу інформації. Дані операції реалізовані за допомогою функцій форматування, прапорців та маніпуляторів



*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. Відкрити файл для вводу чи виводу можна наступним чином:



**Індивідуальне завдання:**

25.З клавіатури вводиться назва шахти, кількість людей, що у ній працює на зміну, тривалість зміни у годинах, розмір видобутку за зміну, а у файл виводиться назва шахти, кількість людей, що у ній працює на зміну, тривалість зміни у годинах, розмір видобутку за зміну та розмір видобутку за місяць.

**Код програми:**

#include <iostream>

#include <fstream>

#include <iomanip>

#include <math.h>

#include <vector>

#include <string>

**using** **namespace** std;

#define div ";"

**extern** **void** menu();

vector <string> split(string str, string divider)

{

vector <string> res;

**int** start = 0, end, delta = divider.length(), i = 0;

**while** ((end = str.find(divider, start)) != string::npos)

{

res.push\_back(str.substr(start, end - start));

start = end + delta;

}

res.push\_back(str.substr(start));

**return** res;

}

**struct** type {

string name;

**int** people, duration, production;

};

**void** read()

{

**int** i = 0;

vector<type> Data;

string temp;

vector <string> buff;

ifstream file("file.txt");

**if** (file.fail())

{

cout << "Неможливо вiдкрити файл" << endl;

}

**else**

{

**while** (file.eof() != 1)

{

file >> temp;

buff = split(temp, div);

Data.resize(Data.size() + 1);

Data[i].name = buff[0];

Data[i].people = stoi(buff[1]);

Data[i].duration = stoi(buff[2]);

Data[i].production = stoi(buff[3]);

i++;;

}

Data.pop\_back();

file.close();

}

cout << setw(10) << "Ім'я" << setw(20) << "Кількість людей" << setw(20) << "Тривалість зміни" << setw(20) << "Продуктивність" << setw(30) << "Продуктивність з міс" << endl;

**if** (Data.empty() == 0)

{

**for** (**unsigned** **int** i = 0; i < Data.size(); i++)

{

cout << setw(10) << Data[i].name << setw(20) <<

Data[i].people << setw(20) <<

Data[i].duration << setw(20) <<

Data[i].production << setw(30) <<

Data[i].production \* 30 << endl;

}

}

system("pause");

menu();

}

**void** write()

{

string name;

**int** people, duration, production;

cout << "Введіть ім'я" << endl;

cin >> name;

cout << "Введіть кількість людей" << endl;

cin >> people;

cout << "Введіть тривалість зміни" << endl;

cin >> duration;

cout << "Введіть продуктивність" << endl;

cin >> production;

ofstream file("file.txt", ios\_base::app);

file << name << div << people << div << duration << div << production << endl;

file.close();

system("pause");

menu();

}

**void** menu()

{

system("cls");

cout << "Виберіть дію:" << endl;

cout << "1. Записати у файл" << endl;

cout << "2. Прочитати з файлу" << endl;

cout << "3. Вихід" << endl;

**int** ch;

cin >> ch;

**switch** (ch)

{

**case** 1:

{

write();

**break**;

}

**case** 2:

{

read();

**break**;

}

**default**:

exit(0);

**break**;

}

}

**int** main()

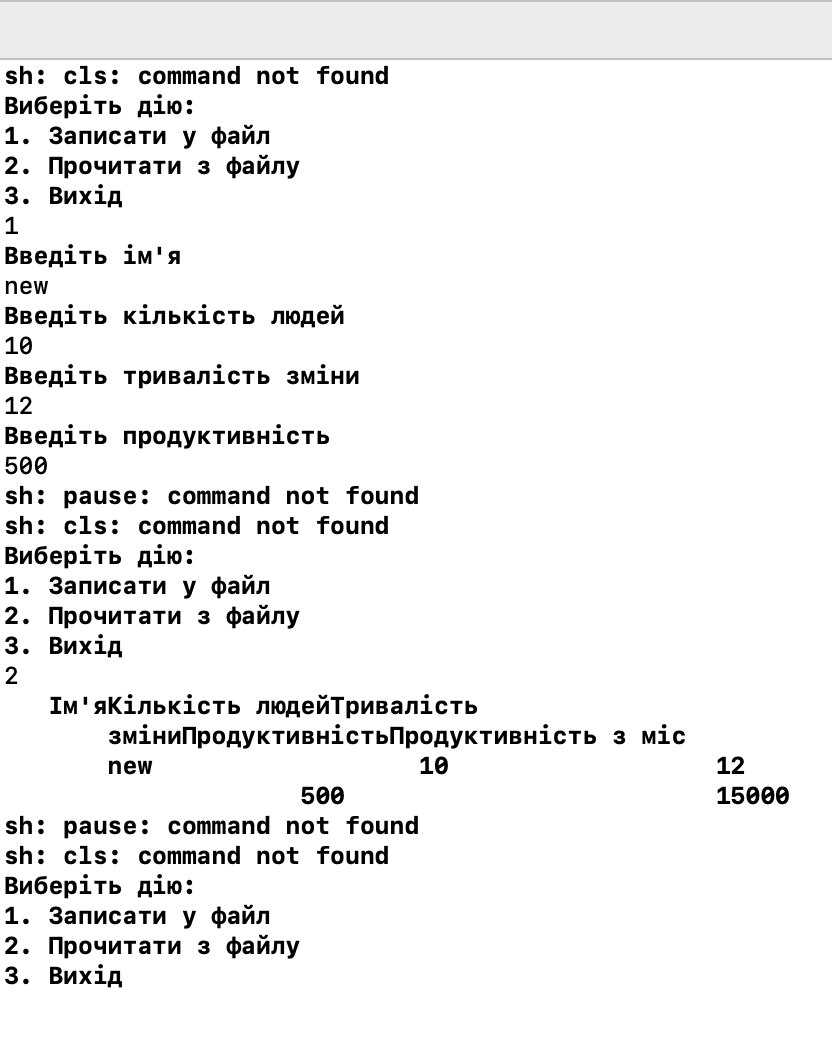
{

setlocale(LC\_ALL, "Russian");

menu();

}

**Результат виконання функції:**

****

**Висновок**: на даній лабораторній роботі я познайомилась з потоковим вводом-виводом, просторами імен та файловим вводом-виводом.