

## ЛАБОРАТОРНА РОБОТА 3

### РОБОТА З ПОТОКАМИ ЧИТАННЯ/ЗАПИСУ ФАЙЛІВ. МАСИВИ ОБ'ЄКТІВ

**Мета роботи:** ознайомитися з способами запису у файли та зчитування з файлів та отримати практичні навички у написанні програм з використанням файлових потоків, створенням та обробкою динамічних масивів об'єктів.

#### Основні теоретичні відомості

C++ надає наступні класи вхідних і вихідних файлових потоків, які призначені для читання символів з файлу та запису символів у файл:

ofstream: клас потоку для запису у файли

ifstream: клас потоку для читання файлів

fstream: клас потоку для читання і запису в / з файлів.

Розглянемо простий приклад запису у файл:

```
#include <iostream>
#include <fstream>
using namespace std;

int main() {
    ofstream myfile;
    myfile.open("example.txt");
    myfile << "This text is written in the file";
    myfile.close();
}
```

Як результат ми отримаємо файл "example.txt", у якому буде записано "This text is written in the file";

Для того, щоб відкрити файл використовується функція open():

open (<назва файлу>, <режим відкриття>);

Режими відкриття представлені в таблиці 3.1:

ios::in	Відкриття для зчитування
ios::out	Відкриття для запису
ios::binary	Відкриття у бінрному режимі
ios::ate	Встановити вихідне положення в кінці файлу.
ios::app	Усі вихідні операції виконуються в кінець файлу, додаючи вміст поточного вміст файлу.

<code>ios::trunc</code>	Якщо файл відкрито для виведення операцій, які існували до нього то попередній вміст видаляється і замінюється новим.
-------------------------	---

Всі ці режими можуть комбінуватися за допомогою оператора OR (|).

Наприклад:

```
ofstream myfile;
myfile.open("example.bin", ios::out | ios::app | ios::binary);
```

Кожна функція `open()` у класах `ofstream`, `ifstream`, `fstream` має своє значення за замовчанням, як показано в таблиці 3.2:

<code>ofstream</code>	<code>ios::out</code>
<code>ifstream</code>	<code>ios::in</code>
<code>fstream</code>	<code>ios::in   ios::out</code>

Функція використовує ці режими за замовчанням, коли поле режиму у функції `open()` відсутнє.

Файл можна відкрити і без функції `open()`, створюючи потоковий об'єкт з параметрами:

```
#include <fstream>
#include <string>
using namespace std;

int main() {
    string str;
    ifstream fin("example.txt");
    fin >> str;
    cout << str;
    fin.close();
}
```

Ця програма зчитує з файлу перший рядок і виводить його на екран.

### Перевіряючі функції:

`is_open()` – повертає значення `true`, якщо відкриття файлу відбулось успішно.

`bad()` – повертає значення `true`, якщо при виконання зчитування або запису відбулись помилки.

`fail()` – дещо схожа функція з `bad()`, оскільки повертає значення `true` в ситуаціях зазначених щодо `bad()`, але в цій також додається перевірка на правильність алфавіту.

`eof()` – повертає значення `true`, якщо було досягнуто кінця файлу.

Наприклад, використаємо функцію `eof()`, для зчитування рядково:

```
#include <fstream>
#include <string>
using namespace std;

int main() {
    string str;
    ifstream fin("data.txt");
    if (fin.is_open())
    {
        while (!fin.eof()) // поки не кінець файлу
        {
            getline(fin, str, '\n');// зчитуємо рядок
            cout << str << endl;
        }
        fin.close();
    }
    else
        cout << "Unable to open file";
}
```

### Функції навігації у файлі

`tellg()` і `tellp()` – ці функції не мають параметрів та повертають значення про поточну позицію каретки, `tellg()` – при зчитуванні, `tellp()` – при записі.

`seekg()` і `seekp()` - дозволяють змінювати позицію каретки при зчитуванні – `seekg()`, при записі – `seekp()`.

### Сортування масиву об'єктів

Приклад програми.

Є текстовий файл `data.txt` з будь-якою кількістю рядків:

Файл	Правка	Формат	Вид
Справка			
Зайнаб	90		
Джамиля	67		
Гюзель	60		
Саида	75		
Хафиза	85		
Зухра	95		
Лейла	70		
Зульфия	97		
Гюльчатай	100		

Прізвище і кількість балів розділені символом “Tab”

Нижче наведений текст програми, яка зчитує дані з файлу data.txt, створює динамічний масив об’єктів і сортує цей масив за алфавитом:

```
#include <fstream>
#include <string>
using namespace std;

class Student // початок оголошення класу
{
    string surname; // прізвище студента
    int scores;     // бали
public:

    void Set(string surname, int scores)
    {
        this->surname = surname;
        this->scores = scores;
    }
    string GetName()
    {
        return surname;
    }
    int GetScores()
    {
        return scores;
    }
    void Print()
    {
        cout <<"Студент(ка) " << surname << " має " << scores << "
балів" << endl;
    }
}; // кінець оголошення класу
```

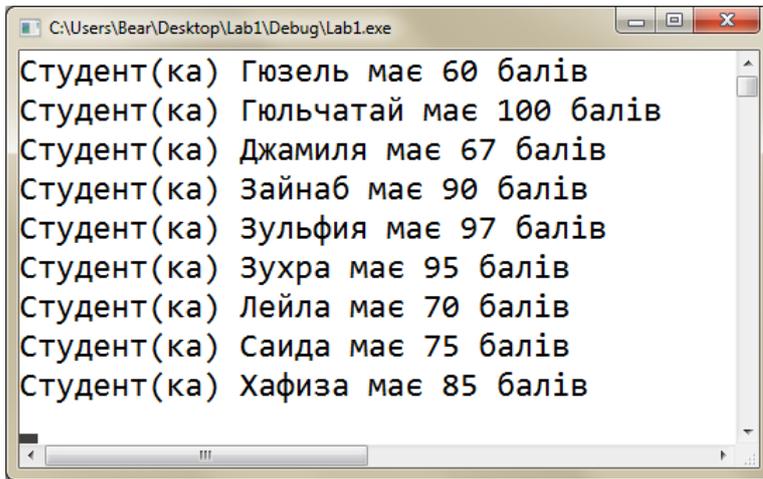
```

int main() {
    int kilkist = 0;
    string str;
    ifstream fin("data.txt"); // створення потоку читання файлу
    if (fin)
    {
        while (!fin.eof()) // поки не кінець файлу
        {
            getline(fin, str, '\n');// зчитуємо рядок
            kilkist++; // підраховуємо кількість рядків
        }
        // створюється масив об'єктів (екземплярів класу Student)
        Student *P = new Student[kilkist];
        fin.clear(); // скидання прапорця "кінець файлу"
        fin.seekg(0); // позиція каретки на початок файлу
        for (int i = 0; i < kilkist; i++)
        {
            string name;
            int b;
            fin >> name; // читання прізвища і-го студента з файлу
            fin >> b; // читання бала і-го студента з файлу
            P[i].Set(name, b); // виклик методу Set для і-го студента
        }
        // сортування за алфавітом
        for (int i = 0; i < kilkist - 1; i++)
            for (int j = i + 1; j < kilkist; j++)
                if (P[i].GetName() > P[j].GetName())
                {
                    swap(P[i], P[j]); // міняємо місцями P[i]та P[j]
                }
        // вивід списку за алфавітом
        for (int i = 0; i < kilkist; i++)
            P[i].Print();

        fin.close(); // закриття потоку
        delete []P; // звільнення динамічної пам'яті
    }
    else
        cout << "Unable to open file";
}

```

Результат роботи програми:



```
C:\Users\Bear\Desktop\Lab1\Debug\Lab1.exe
Студент(ка) Гюзель має 60 балів
Студент(ка) Гюльчатай має 100 балів
Студент(ка) Джамиля має 67 балів
Студент(ка) Зайнаб має 90 балів
Студент(ка) Зульфия має 97 балів
Студент(ка) Зухра має 95 балів
Студент(ка) Лейла має 70 балів
Студент(ка) Саида має 75 балів
Студент(ка) Хафиза має 85 балів
```

### Завдання до лабораторної роботи

1. Створить текстовий файл на ім'я data.txt, с довільною кількістю рядків, але не менше 10 зі студентами та їх оцінками з трьох предметів: математики, фізики, інформатики. Формат файлу: кожний рядок містить прізвище, ім'я та три числа. Дані в рядку розділені одним пробілом або символом «Tab». Оцінки приймають значення від 1 до 5.
2. Створить масив екземплярів класу та зчитати дані з файлу data.txt для визначення відповідних полів. Виконати завдання відповідно варіанту.
3. Створити звіт.

### Варіанти завдань

Варіант	Виконати
1	Визначте середній бал кожного студента зі всіх предметів. Виведіть три дійсні числа: середній бал усіх студентів за зростанням.
2	Виведіть за алфавітом прізвища та імена студентів, які не мають трійок (а також двійок).
3	Визначте студентів з найкращою успішністю, тобто з максимальним середнім балом з трьох предметів. Виведіть за алфавітом список студентів, які мають середній бал більш за 4.
4	Виведіть прізвища та імена студентів в порядку убутання їх середнього балу.

5	Виведіть за алфавітом список студентів, що мають найкращу успішність по заданому предмету. Предмет має вводитися користувачем з клавіатури.
6	Виведіть за алфавітом список студентів, що мають найгіршу успішність по заданому предмету. Предмет має вводитися користувачем з клавіатури.
7	Вивести успішність для певного студента, ім'я якого буде задано з клавіатури.
8	Підрахувати середній бал по заданому предмету. Предмет має вводитися користувачем з клавіатури.
9	Підрахувати кількість студентів з заданим балом по фізиці. Виведіть за алфавітом список цих студентів. Бал має вводитися користувачем з клавіатури.
10	Вивести за алфавітом імена студентів, що не мають жодної п'ятірки.
11	Відсортувати студентів за середнім балом та вивести на екран тих, хто має бал більш за 4.
12	Вивести за алфавітом імена студентів, що мають лише п'ятірки.

### **Контрольні запитання**

1. Які класи призначені для введення і виведення символів в / з файлів?
2. Яким чином виконується запис у файл?
3. Яка функція використовується для відкриття файлів та які параметри вона в собі включає?
4. Які режими відкриття файлу ви знаєте? Опишіть їх та дайте стислу характеристику.
5. Яким чином здійснюється зчитування з текстового файлу?
6. Які функції навігації ви знаєте? Дайте стислий опис.
7. Які перевіряючі функції ви знаєте? Дайте стислий опис.
8. Яким чином відбувається закриття файлу?