**Лабораторна робота №19.**. **Розробка програм із використанням бінарних файлів для введення/виведення даних.**

**Мета:** Набуття навичок в розробці програм, де використовується перевантаження операторів потокового введення/виведення та обробка виключень..

**На період** карантину в дистанційній формі навчання результати надсилати на електронну адресу викладача [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com)у вигляді cpp- та h- файлів з іменем у форматі

**<Номер групи><Номер лабораторної>[-<Номер завдання>] <Прізвище англійською>**

та викладати на платформу коледжу у власний репозиторій, під’єднаний до репозиторію викладача.

Приклад іменування файлів, IPZ-31LAB11-1KRASNOPOROV.cpp, IPZ-31LAB11-1KRASNOPOROV.h.

При відсутності можливості доступу до Інтернету текст програми набрати в Блокноті або WordPad Windows та надіслати на електронну адресу викладача.

**Строк виконання цієї роботи 29.05.2020**

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача.

**Завдання**

1. Опрацювати теоретичні відомості та переглянути лекції "Тема16.Класи потоків введення-виведення", "Тема18. Стандартні консольні потоки введення/виведення". Використання файлів для введення/виведення даних. Класи ofstream та ifstream. Бінарні та текстові файли".
2. Вся ця робота виконується з програмою, яка розроблювалась на ЛР №№ 5 – 15, 17. Перелік завдань, які мають бути реалізовані в програмі надані в ЛР№№ 5 – 15, 17-18, а також вимоги до оформлення програми надані в завданні до ЛР№11.
3. Модифікувати функціональність позиції меню "71 Test write – read binary", додавши запит "***Rewrite?*** ". При позитивній відповіді (y або yes) відкрити бінарний файл для дозапису, при негативній (n або no) - для запису.
4. Створити нову позицію меню "75 Test Record search in binary ", що викликає функцію, в який виконуються такі дії:

* Запитує фразу;
* Записує її до бінарного файлу;
* Для перевірки запису файл відкривається для читання, зчитується інформація та виводиться на консоль;
* Запитує номер позиції для читання та кількість літер;
* Через команди прямого доступу зчитується запитаний фрагмент та виводиться на консоль.
* В функції повинен проводитися аналіз результатів звернення до файлу з обробленням помилок.

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

**Функція прямого доступу до файлу**

**Прямий доступ до файлу** – це доступ до вмісту файлу засобами операційної системи.

**Функції введення-виведення низького рівня (прямого доступу)** –це функції, які здійснюють обмін з файлами або периферійними пристроями шляхом прямого звернення до відповідних функцій операційної системи.

Функція read при коректному виконанні здійснює послідовне читання даних з файлу. При роботі с файлами на нижньому рівні можна змінювати порядок доступу до даних з послідовного на довільний, прямий доступ. Для цього використовується функція fseek.

Синтаксис:

**int fseek(FILE \*f, long off, int org);**

де f – покажчик на файл;

off – позиція зміщення;

org – початок відліку.

**Зміщення** – це відносна координата переміщення покажчика при зверненні до даних файлу на нижньому рівні. Зміщення off задається виразом або змінною та може бути від’ємним, тобто можливе переміщення і в прямому, і в зворотному напрямках. Початок відліку org задається однією з визначених в файлі <stdio.h> констант:

|  |  |  |
| --- | --- | --- |
| **Константа** | **Значення** | **Опис** |
| SEEK\_SET | 0 | начало файлу |
| SEEK\_CUR | 1 | поточна позиція |
| SEEK\_END | 2 | кінець файлу |

Функція повертає 0, якщо переміщення в потоці виконано успішно, інакше повертається ненульове значення.

Функцію fseek потрібно використовувати тільки при роботі с бінарними файлами. В текстових файлах початок відліку org повинен встановлюватися як SEEK\_SET, а зміщення вказується відносно початку файлу (можливе нульове зміщення, яке інтерпретується як позиціонування в початок файлу).

Наприклад, для текстових файлів:

//переміщення до початку потоку з поточної позиції

**fseek(f,0L,SEEK\_SET);**

// переміщення до кінця потоку з поточної позиції

**fseek(f,0L,SEEK\_END);**

Наприклад, для бінарних файлів:

// переміщення вперед від початку на довжину змінної ***а***

**fseek(f,(long)sizeof(a),SEEK\_SET);**

// переміщення назад від поточної позиції на довжину змінної ***а***

**fseek(f,-(long)sizeof(a),SEEK\_CUR);**

Для прямого доступу до файлу використовуються також функції:

//отримує значення покажчика поточної позиції в потоці

**long ftell(FILE \*f);**

//встановлює значення покажчика на початок потоку

void rewind(FILE \*f);

***Приклад 1*.** Використання функцій доступу до даних на нижньому рівні.

#include "stdafx.h"

#include <iostream>

using namespace std;

#include <fcntl.h>

#include <sys\stat.h>

#include <io.h>

int \_tmain(int argc, \_TCHAR\* argv[]){

int fh1, fh2;

//відкриття файлу на нижньому рівні

fh1 = open("data1.dat", O\_RDONLY);

//перевірка коректності відкриття файлу

if (fh1 == -1)

perror("Open failed on input file");

fh2 = open("data2.dat",O\_WRONLY|O\_TRUNC|O\_CREAT,S\_IREAD|S\_IWRITE);

if (fh2 == -1)

perror("Open failed on output file");

system("pause");

return 0;

}

***Приклад 2*.** Використання позиціонування файлового покажчика за допомогою функції fseek в бінарному файлі.

#include "stdafx.h"

#include <iostream>

using namespace std;

#include <io.h>

int \_tmain(int argc, \_TCHAR\* argv[]){

FILE \*f;//покажчик на бінарний файл

int i,n=10;

char s[]="String";

float r;

f=fopen("file\_bin","wb");

//створення бінарного файлу для запису

for(i=1;i<=n;i++){

r=pow(i,1.0/3);

fwrite(s,sizeof(s),1,f);

//запис рядка String до файлу

fwrite(&i,sizeof(int),1,f);

//запис цілого числа (номера рядка) до файлу

fwrite(&r,sizeof(float),1,f);

//запис дійсного числа (кореня кубічного) до файлу

printf("\n%s %d %f",s,i,r);//контрольне виведення на екран

}

fclose(f);//закриття файлу

printf("\n");

f=fopen("file\_bin","rb");

//відкриття бінарного файлу для читання

for(i=n; i>0; i--) {

//переміщення файлового покажчика

fseek(f,(i-1)\*(sizeof(s)+sizeof(int)+sizeof(float)),SEEK\_SET);

fread(&s,sizeof(s),1,f);//читання рядка

fread(&n,sizeof(int),1,f);//читання цілого числа

fread(&r,sizeof(float),1,f);

//читання дійсного числа

printf("\n%s %d %f",s,n,r);

//виведення на екран вмісту файлу

}

system("pause");

return 0;

}