Оголошення файлової змінної та відкриття текстового файлу для читання має наступний синтаксис:

ifstream f\_in(“filename.txt”, ознака1|ознака2|…|ознакаN);

У даному записі f\_in – назва файлової змінної, filename.txt –назва файлу на диску. Ознаки слугують для визначення прав доступу до файлу. Перелік ознак відкриття файлу та їх значень наведено в табл. 13.3

Оголошення файлової змінної та відкриття файлу для запису має синтаксис:

ofstream f\_out(“filename.txt”, ознака1|ознака2|…|ознакаN);

У даному записі f\_out – назва файлової змінної, filename.txt –  назва файлу на диску.

Якщо не використовувати ознаки відкриття файлу для читання чи запису, на диску буде створений новий файл із зазначеним ім'ям, або перезаписаний файл із таким же ім'ям, якщо він вже існує.

Наприклад, для того, щоб записати потрібну інформацію в кінець файлу text1.dat, необхідно скористатися фрагментом програмного коду:

ofstream flags(“text1.dat”, ios::app);

Таблиця 13.3 – Ознаки відкриття файлу

| **Ознака** | **Призначення** |
| --- | --- |
| ios::in | Відкриває файл для читання. Вміст файлу зберігається |
| ios::out | Відкриває файл для запису. Якщо файл не існує, то буде створений |
| ios::app | Відкриває файл для дозапису. Дані будуть записані у кінець файлу |
| ios::trunc | Якщо  файл, який відкривають для запису, вже існує, то його вміст буде видалено. |
| ios::nocreate | Забороняє створювати файл, який відкривають |
| ios::inoreplace | Забороняє перезаписувати існуючий  файл |
| ios::binary | открытие файла в двоичном режиме |

Наприклад, для того, щоб створити текстовий файл і записати в нього рядок «Робота в С++» необхідно виконати наступні кроки:

ofstream fout;

fout.open("cppstudio.txt"); // це можна було зробити, записавши

 // ofstream fout(“cppstudio.txt ”, app);

fout << “ Робота в С++ ”;

fout.close();

1. створити об’єкт класу ofstream, наприклад, з назвою fout:
2. пов’язати об’єкт класу з файлом, в який буде проводитися запис:
3. записати рядок в файл;
4. закрити файл:

**Приклад 6. Записати у файл три рядки тексту.**

#include "stdafx.h"

#include<iostrem.h>

#include<fstream.h>

int main()

{

ofstream fout(“output.txt”);

fout<<”Prog na C++ “<<”I like”<<endl;

fout<<”My name is”<<endl;

fout<<”I am a student”<<endl;

fout.close();

return 0;

}

Для того щоб прочитати створений файл, знадобитися виконати аналогічні кроки, що й при запису у файл з невеликими змінами:

ifstream fin("cppstudio.txt");

fin >> buff;

fin.close().

1. створити об'єкт класу ifstream і пов'язати його з файлом, який відкриємо в режимі для читання:
2. прочитати файл;
3. закрити файл.

# Ввід по кінцю файлу

**Приклад 7. Прочитати файл, записаний в результаті виконання прикладу 6.**

#include "stdafx.h"

int main()

{

ifstream fin(“output.txt”);

char line[64];

while (!fin.eof())

{

fin.getline(line,sizeof(line));

cout<<line<<endl;

}

fin.close();

getch();

return 0;

}

У даному випадку зчитування інформації з файлу відбувається доти, доки не буде досягнуто його кінця

# Розпізнавання помилок при відкритті файлу

відкривши файл, бажано прописати перевірку: відкрився він? Так як є ряд причин, за якими файл може не відкритися, а ми цього не побачимо. Наприклад, файлу з вказаним ім'ям немає в прописаної папці або шлях вказано невірно. Можна піти двома шляхами: перевірити змінну файлу в логічному виразі (застосувавши оператор “!”,  наприклад) або використовувати метод **is\_open()** :

Варіант перевірки з використанням методу **is\_open()** :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | #include <iostream>  #include <fstream>  using namespace std;    int main()  {  setlocale(LC\_ALL, "rus");    ifstream file ("d:\\1\\файл.txt");    if (!file.is\_open()) // вызов метода is\_open()  {  cout << "Файл не открыт!\n\n" << endl;  return -1;  }  if(file.eof())  {  cout << “файл пустой”;  return -2;  }  } |

Метод **is\_open()** поверне 1, якщо файл знайдений і успішно відкритий. інакше поверне 0 і спрацює код прописаний в блоці **else**.

Якщо файл не відкритий – бажано обробити помилку. Як правило, якщо вся робота програми пов'язана з файлом пишуть якесь повідомлення в консоль, і ставлять вихід з програми. При серйозних помилках прийнято повертати якийсь код виконання (число),  який буде характеризувати ту чи іншу помилку. Коди для кожного виду помилок автор програми може придумувати свої.