**Общая схема организации ввода-вывода :**

1. Создать объекта для организации ввода-вывода.
2. Убедиться в том, что объект создан успешно (с помощью метода is\_open()).
3. Выполнить необходимые операции по вводу данных (с помощью оператора >>) или по выводу данных (с помощью оператора <<).
4. Закончить работу с объектом (с помощью метода close()).

#include <iostream>

#include <fstream> // Библиотека функций для работы с файлами

#include<math.h>

#include<cstring>

using namespace std;

int main()

{

int i;

ofstream fout;

fout.open("fout1.txt"); //Открытие потока для записи в файл "fout1.txt"

if (fout1.is\_open()) // Проверка , что файл создан успешно

{

fout<<" Good morning! "<<endl; // Запись строки в файл

fout<<" Be happy! "<<endl; // Запись строки в файл

for( i=0; i<10; i++)

{

fout << i << endl ;

}

for( i=0; i<10; i++)

{

fout << i << '\t' << pow(i,2) << endl ; // Запись в файл аргумента и функции

}

fout.close(); //Закрытие потока

}

else

{

cerr<<" error!"; // **cerr** - стандартный поток ошибок без буферизации.

// который используется для вывода ошибок.

// Это также экземпляр класса ostream.

//Поскольку cerr небуферизован, он используется, // когда нам нужно немедленно отобразить

//сообщение об ошибке.

}

char str[50];

int a[10];

int b[10][2];

int y;

ifstream fin("fout1.txt"); //Открытие потока для чтения из файла "fout1.txt"

if ( fin.is\_open() )

{

//fin>>str; // Чтение слова из файла

// cout<<str<<endl; // Вывод слова на экран

fin.getline(str, 50); //Чтение строки из файла

cout<<str<<endl; // Вывод строки на экран

fin.getline(str, 50);

cout<<str<<endl;

for( int i=0; i<10; i++)

{

fin>>a[i]; // Чтение данных из файла в массив **а**

y=pow(a[i],3);

cout<<a[i]<<'\t'<< y<<endl; // Вывод элемента массива на экран

}

for(int i=0;i<10;i++)

{

for(int j=0;j<2;j++)

{

fin>>b[i][j]; // Чтение данных из файла в массив **b**

cout<<'\t'<<b[i][j]; // Вывод прочитанных данных на экран

}

cout<<endl;

}

}

fin.close(); //Закрытие потока

return 0;

}



