

ФАЙЛЫ C++

ВСЁ О ФАЙЛАХ(ПОЧТИ ВСЁ)



ВИДЫ ФАЙЛОВ

- Текстовые
- Бинарные

ТЕКСТОВЫЕ ФАЙЛЫ

Текстовый файл содержит последовательность символов (в основном печатных знаков, принадлежащих тому или иному набору символов). Эти символы обычно сгруппированы в строки (англ. *lines, rows*)



Файлы с расширением .html

БИНАРНЫЕ ФАЙЛЫ

- Последовательность произвольных байтов. Название связано с тем, что байты состоят из бит, то есть двоичных цифр.
- Двоичные файлы противопоставляются текстовым файлам. При этом с точки зрения технической реализации на уровне аппаратуры, текстовые файлы являются видом двоичных файлов. Поэтому под определение «двоичный файл» подходит любой файл.

ВИДЫ ДОСТУПА

```
graph TD; A[ВИДЫ ДОСТУПА] --- B[ПРОИЗВОЛЬНЫЙ]; A --- C[ПОСЛЕДОВАТЕЛЬНЫЙ]; B --- D[ДЛЯ БИНАРНОГО ФАЙЛА]; C --- E[ДЛЯ ТЕКСТОВОГО И БИНАРНОГО ФАЙЛА];
```

ПРОИЗВОЛЬНЫЙ

ДЛЯ БИНАРНОГО ФАЙЛА

ПОСЛЕДОВАТЕЛЬНЫЙ

ДЛЯ ТЕКСТОВОГО И
БИНАРНОГО ФАЙЛА

ЧТЕНИЕ ИЗ ТЕКСТОВОГО ФАЙЛА

```
string ReadStringFromFile(string fileName) {  
    string result = "";  
    ifstream file(fileName);  
    //ifstream file;  
    //file.open();  
    if (!file.is_open()) {  
        return "";  
    }  
    getline(file, result);  
    file.close();  
    return result;  
}
```

ЗАПИСЬ В ТЕКСТОВЫЙ ФАЙЛ

```
void WriteStringInFile(string fileName, string data) {  
    ofstream file(fileName);  
    if (!file.is_open()) {  
        return;  
    }  
    file << data;  
    file.close();  
}
```

МОДИФИКАТОРЫ ДОСТУПА К ФАЙЛУ

- **ios_base::in** - Открытие для чтения.
- **ios_base::out** - Открытие для записи.
- **ios_base::app** - Автоматически переставляет указатель текущего символа потока в конец.
- **ios_base::binary** - Открытие файла в бинарном режиме.
- **ios_base::trunc** - Усекает поток после открытия.

ФУНКЦИИ ПОЗИЦИОНИРОВАНИЯ

- **eof()** – проверка на конец файла
- **seekg()** – перескочить на позицию в файле
 - **ios_base::end** – Отсчитать новую позицию с конца файла
 - **ios_base::beg** – Отсчитать новую позицию с начала файла (абсолютное позиционирование)
 - **ios_base::cur** – Перескочить на n байт начиная от текущей позиции в файле (по умолчанию)
- **tellg()** – получить текущую позицию