# Работа с файлами в C++

Махнев А.А.

# Файл – это..

- именованный набор  байтов, который может быть сохранен на некотором накопителе и имеет уникальное имя

- file.txt
- file.dat

# Полное имя файлов

- Полный адрес к директории файла с указанием имени файла,

- D:\docs\file.txt

# Для работы с файлами

- Подключить заголовочный файл <fstream>

- Содержит несколько классов
- подключены заголовочные файлы
- <ifstream> — файловый ввод
- <ofstream>  — файловый вывод.

- Файловый ввод/вывод аналогичен стандартному вводу/выводу

# Пример 1

- Создать текстовый файл и записать в него строку

1. создать объект класса ofstream;
2. связать объект класса с файлом, в который будет производиться запись;
3. записать строку в файл;
4. закрыть файл.

# Пример 1 (реализация)

- // создаём объект для записи в файл
- ofstream **fout**;

- // связываем объект с файлом
- fout.open("cppstudio.txt");

- //запись строки в файл
- fout << "Работа с файлами в C++";

- // закрываем файл
- fout.close();

# Пояснения

- Указанный файл будет создан в текущей директории с программой. Если файл с таким именем существует, то существующий файл будет заменен новым

- одной строке создать объект и связать его с файлом
- ofstream  fout("cppstudio.txt");

# Как считать данные из файла?

- создать объект класса ifstream и связать его с файлом, из которого будет производиться считывание;
- прочитать файл;
- закрыть файл.

# Пример 2 (1 способ считывание данных)

- setlocale(LC_ALL, "rus"); // корректное отображение Кириллицы
- char buff[50]; // буфер промежуточного хранения считываемого из файла текста
- ifstream fin("cppstudio.txt"); // открыли файл для чтения
-
- fin >> buff; // считали первое слово из файла
- cout << buff << endl; // напечатали это слово
-
-
- fin.close(); // закрываем файл
-

# Пример 2  (2 способ считывание данных)

- setlocale(LC_ALL, "rus"); // корректное отображение Кириллицы
- char buff[50]; // буфер промежуточного хранения считываемого из файла текста
- 
- ifstream fin("cppstudio.txt"); // открыли файл для чтения
- 
- fin.getline(buff, 50); // считали строку из файла
- fin.close(); // закрываем файл
- cout << buff << endl; // напечатали эту строку

# Проверка доступа к файлу

```
char buff[50]; // буфер промежуточного хранения считываемого из файла текста
    ifstream fin("cppstudio.doc"); // (ВВЕЛИ НЕ КОРРЕКТНОЕ ИМЯ ФАЙЛА)

    if (!fin.is_open()) // если файл не открыт
        cout << "Файл не может быть открыт!\n"; // сообщить об этом
    else
    {
    fin >> buff; // считали первое слово из файла
    cout << buff << endl; // напечатали это слово

    fin.getline(buff, 50); // считали строку из файла
    fin.close(); // закрываем файл
    cout << buff << endl; // напечатали эту строку
    }
```

# Режимы открытия файлов

- установка режима в классе ios_base в виде константы

# Режимы открытия файлов

| Константа | Описание |
|---|---|
| **ios_base::in** | открыть файл для чтения |
| **ios_base::out** | открыть файл для записи |
| **ios_base::ate** | при открытии переместить указатель в конец файла |
| **ios_base::app** | открыть файл для записи в конец файла |
| **ios_base::trunc** | удалить содержимое файла, если он существует |
| **ios_base::binary** | открытие файла в двоичном режиме |

# Режим устанавливаем при открытии файла или создании объекта

- // открываем файл для добавления информации к концу файла
- ofstream fout("cppstudio.txt", ios_base::app);
- // открываем файл для добавления информации к концу файла
- fout.open("cppstudio.txt", ios_base::app);

# Режимы можно комбинировать

- ios_base::out | ios_base::trunc

- открытие файла для записи, предварительно очистив его.