

Работа с файлами

C++ забезпечує класи для операції з файлами для записи і читання

- **ofstream**: клас потоку записи на файл
- **ifstream**: клас потоку для читання з файла
- **fstream**: клас потоку для читання і записи в файл.

Основные операции с файлами – запись данных в файл example.txt

```
#include <iostream>
#include <fstream>
using namespace std;

int main () {
    ofstream myfile;
    myfile.open ("example.txt");
    myfile << "Writing this to a file.\n";
    myfile.close();
    return 0;
}
```

Открытие файла

- Операция, выполняемая для закрепления объекта файловых потоков к физическим файлам.

`open(название_файла, режим)`

Когда есть выбор режима открываемого файла можно выбрать из следующих режимов:

<code>ios::in</code>	Open for input operations.
<code>ios::out</code>	Open for output operations.
<code>ios::binary</code>	Open in binary mode.
<code>ios::ate</code>	Set the initial position at the end of the file. If this flag is not set to any value, the initial position is the beginning of the file.
<code>ios::app</code>	All output operations are performed at the end of the file, appending the content to the current content of the file. This flag can only be used in s operations.
<code>ios::trunc</code>	If the file opened for output operations already existed before, its previous content is deleted and replaced by the new one.

- Все эти операторы могут быть скомбинированы посредством битовых операторов:

```
ofstream myfile;  
myfile.open ("example.bin", ios::out | ios::app | ios::binary);
```

- Функция `open()` классов `ofstream`, `ifstream` и `fstream` имеет режим по умолчанию и используется при открытии файла без второго параметра:

class	default mode parameter
<code>ofstream</code>	<code>ios::out</code>
<code>ifstream</code>	<code>ios::in</code>
<code>fstream</code>	<code>ios::in ios::out</code>

- Также файл может быть открыт с помощью конструктора

```
ofstream myfile ("example.bin", ios::out | ios::app | ios::binary);
```


- Чтобы проверить успешно ли открылся файл, можно использовать метод `is_open()`

```
if (myfile.is_open()) { /* ok, proceed with output */ }
```

Флаги проверки статуса

- `bad()`
- `fail()`
- `eof()`
- `good()`

`tellp()` & `tellg()` – функции получения
позиции курсора в файле

`seekg()` & `seekp()` – операторы перевода курсора в файле

- `seekg(position)`
- `seekp(position)`

- `seekg(смещение, направление)`
- `seekp(смещение, направление)`

- Направления: `ios::beg`, `ios::end`, `ios::cur`

Заккрытие файла

- После завершения всех операции с файлом, нужно его закрыть для освобождения его ресурсов

```
myfile.close();
```

Текстовые файлы

```
// writing on a text file
#include <iostream>
#include <fstream>
using namespace std;

int main () {
    ofstream myfile ("example.txt");
    if (myfile.is_open())
    {
        myfile << "This is a line.\n";
        myfile << "This is another line.\n";
        myfile.close();
    }
    else cout << "Unable to open file";
    return 0;
}
```

Пример чтения из файла и вывода на консоль

```
int main(){
    int a[10];
    ifstream myf("example.txt");
    while(!myf.eof()){
        myf>>*a;
        cout<<*a<<endl;
    }
    myf.close();

    return 0;
}
```

Ввод данных из файла

```
// reading a text file
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main () {
    string line;
    ifstream myfile ("example.txt");
    if (myfile.is_open())
    {
        while ( myfile.good() )
        {
            getline (myfile,line);
            cout << line << endl;
        }
        myfile.close();
    }

    else cout << "Unable to open file";

    return 0;
}
```


Задания

- Прочитать содержимое файла в массив
- Записать массив строк в файл