

# ФАЙЛОВЕ В C++

1. Четене информация от даден текстов файл и да я извежда като текст в конзолата
2. Създаване текстови файлове и да вписвате в тях дадена информация въведена от клавиатурата или извлечена от друг файл.
3. Режими за работа с файлове, кои са основните и за какво служат.
4. Указатели „get“ и „put“. Накратко: put указателя се поставя там от където искате да започнете да пишете във файла, а get се поставя там от където искате да започнете да четете от файла. И двата указателя се местят автоматично и последователно към края на файла, но могат и да им се задават определени позиции. Ето и кои са основните режими за работа:

## Режими на работа с файлове

| Режим на работа             | Описание  |
|-----------------------------|---|
| <code>ios::in</code>        | Отваря файла за четене.   |
| <code>ios::out</code>       | Отваря файла за запис, като изтрива съдържанието му, ако той съществува.  |
| <code>ios::app</code>       | Отваря файла за запис (добавяне) в края на файла.   |
| <code>ios::ate</code>       | Отваря файла за запис и установява put указателя в края на файла, като предишното съдържание се запазва. Указателят може да се премества на произволни места. |
| <code>ios::trunc</code>     | Отваря файл, ако той съществува изтрива съдържанието му.  |
| <code>ios::nocreate</code>  | Ако файлът съществува той се отваря. Не се създава файл с посоченото име.   |
| <code>ios::noreplace</code> | Създава и отваря файл с посоченото име, само ако такъв файл не съществува.  |
| <code>ios::binary</code>    | Превключва режима за достъп до файл от текстов в двоичен.   |

Сега ще преминем към писането на самите примерни програми, за да разберете как работи всичко. Много е важно в началото на „сpp“ файла да включите библиотеката „fstream“, за да можете да извършвате операции с файлове. Има два начина за деклариране на променливи от тип файл.

„ofstream“ се използва, когато ще записваме информация във файл.  
„ifstream“ се използва, когато ще извличаме данни от файл.

Самото деклариране става по следния начин.

### 1. ifstream име\_на\_променлива

или съответно

### 1. ofstream име\_на\_променлива

Сега трябва да отворим дадения файл като определим режимът на работа с него. Ето и как става това.

### 1. open("път до файла и неговото разширение", "режим на работа");

Ето и как затваряме по-горе отвореният файл.

### 1. close();

**ВАЖНО:** Винаги когато приключвате работата с даден файл го затваряйте.

Пример за създаване и записване във файл.

1. `ofstream file;`
2. `file.open("filepath.txt", ios::out);`
3. `file<<"Текстът, който ще записваме във файла"<<endl;`

#### 4. `file.close()`;

Пример за четене на информация от файл.

1. `string promenliva;`
2. `ifstream file;`
3. `file.open("file.txt", ios::in);`
4. `file>>promenliva;`
5. `file.close();`
6. `cout<<promenliva<<endl;`

Програмата от следващия пример ще чете от един текстов файл и ще записва информацията в друг.

1. `#include <iostream>`
2. `#include <conio.h>`
3. `#include <string>`
4. `#include <fstream>`
5. `using namespace std;`
- 6.
7. `int main() {`
8. `string promenliva;`
9. `ifstream file1;`
10. `file1.open("file1.txt", ios::in);`
11. `ofstream file2;`
12. `file2.open("file2.txt", ios::out);`
13. `while(!file1.eof()) {`
14. `//Цикълът ще се върти, докато не се достигне края на файла`
15. `file1>>promenliva;`
16. `file2<<promenliva<<" ";`
17. `}`
18. `file1.close();`
19. `file2.close();`
20. `getch();`
21. `return 0;`
22. `}`

Тази команда означава, че ще се извършва някакво действие в цикъла докато не се достигне края на файла. Или по-просто казано, докато не се прочете целия файл.

#### 1. `!file.eof()`

Програма, която чете целия текст от даден файл и го визуализира.

1. `#include <iostream>`
2. `#include <conio.h>`
3. `#include <string>`
4. `#include <fstream>`
5. `using namespace std;`
- 6.
7. `int main() {`
8. `string prom;`
9. `ifstream file;`
10. `file.open("fail.txt", ios::in);`
11. `while(!file.eof()) {`
12. `file >> prom;`
13. `cout << prom<<" ";`
14. `}`

```
15. file.close();
16. getch();
17. return 0;
18. }
```

Програма, която чете всичко от даден файл, разделя отделните редове и ги въвежда като елементи на даден масив. Примерният код извежда само първия елемент на масива!

```
1. #include <iostream>
2. #include <conio.h>
3. #include <string>
4. #include <fstream>
5. using namespace std;
6.
7. int main() {
8.     string prom, massive[255];
9.     ifstream file;
10.    file.open("fail.txt", ios::in);
11.    int i=0;
12.    while(!file.eof()) {
13.        getline(file, prom);
14.        massive[i]=prom;
15.        i++;
16.    }
17.    cout<<massive[0]<<endl;
18.    getch();
19.    return 0;
20. }
```

При всяко въвеждане и извеждане **put** и **get** указателите се позиционират на мястото където е завършило въвеждането или извеждането. За вземане (връщане) и промяна на техните позиции се използват следните стандартни функции включени в библиотеката **<fstream>**:

- 1) **име\_на променлива.tellg();** - връща цяло число, указващо позицията на **get** указателя във файла
- 2) **име\_на променлива.tellp();** - връща цяло число, указващо позицията на **put** указателя във файла
- 3) **име\_на променлива.seekg(цяло\_число);** - премества **get** указателя на позиция с номер **цяло\_число**, от където ще се извърши следващото **извеждане**.
- 4) **име\_на променлива.seekp(цяло\_число);** - премества **put** указателя на позиция с номер **цяло\_число**, от където ще се извърши следващото **въвеждане**.