

# Бинарни(двоични) файлове

Обектно ориентирано програмиране - семинар 2022/2023

Какво са бинарните файлове?



# Какво е бинарен файл:

Бинарните (двоичните) файлове са файлове, които не са текстови. Тяхното съдържание представлява последователност от групи от битове всяка с по 8 бита. За да може да се използва тяхното съдържание трябва процесорът и компилатора (интерпретатора) на програмата, която ги използва да знае предварително по какъв начин се съдържанието е сериализирано, и как може да го десериализира.



# Видове формати бинарни файлове

- Графични - jpg, png, gif, bmp, tiff, psd
- Видео - mp4, mkv, avi, mov, mpg, vob
- Аудио - mp3, aac, wav, flac, ogg, mka, wma
- Документи: pdf, doc, xls, ppt, docx, odt
- Архиви: zip, rar, 7zip, tar, iso
- Бази от данни: mdb, accde, frm ,sqlite



# Защо се използват?

- Записването и четенето в/от бинарни файлове е много по-бързо
- Много по-малък е като размер от текстов файл. Това е много важно за аудио, видео и графичните файлове, тъй като биха били огромни като размер
- Не всеки тип информация е удобен за текстов файл



# Работа с бинарни файлове в C++



# Отваряне и затваряне на двоичен файл

```
std::fstream file(s: FILENAME, mode: std::ios::out |  
                  std::ios::in |  
                  std::ios::binary);  
  
file.close();
```

```
std::fstream file2;  
file2.open(s: FILENAME, mode: std::ios::out |  
           std::ios::in |  
           std::ios::binary);  
  
file2.close();
```

# Записване на текст в двоичен файл

Използваме метода `write`, който приема 2 аргумента - указател към буфер и размер на буфера.

Пример:

```
char greeting[10] = "Hello!";  
file.write(s: greeting, n: sizeof(greeting));
```





# Записване на обекти в двоичен файл

```
struct Student {  
    int fn;  
    char name[15];  
    double grade;  
};
```

```
Student student[3] = {  
    [0]: { .fn: 1, .name: "Ivan Ivanov", .grade: 5.20},  
    [1]: { .fn: 2, .name: "Georgi Nikolov", .grade: 4.30},  
    [2]: { .fn: 3, .name: "Maria Petrova", .grade: 2.00},  
};  
file.open(s: FILENAME, mode: std::ios::out |  
          std::ios::binary);  
for(int i = 0; i < 3; i++) {  
    file.write(s: (char*)&student[i], n: sizeof(Student));  
}  
file.close();
```

## Четене от двоичен файл (пример с обект)

```
Student student2[3];
```

```
file.open(s: FILENAME, mode: std::ios::in |  
        std::ios::binary);
```

```
for(int i = 0; i < 3; i++) {
```

```
    file.read(s: (char*)&student[i], n: sizeof(Student));  
}
```

```
file.close();
```


```
assert(strcmp(student2[2].name, "Maria Petrova"));
```

# Методи за файловия указател

## ifstream

- seekg - мести файловия указател на определена позиция
- tellg - дава ни текущата позиция на файловия указател

## ofstream

- seekp - мести файловия указател на определена позиция
  - tellp - дава ни текущата позиция на файловия указател
- 

# Задача (минипроект)

Да се напише конзолно приложение, което да представлява регистър на работници в една софтуерна компания. Работниците са 3 вида (developer, quality assurance и devops). За всеки от тях трябва да пазим следната информация: име, години, работна позиция, имейл и телефон. Трябва да можем да изпълним следните операции (CRUD): добавяне на нов служител, зареждане на служители от двоичен файл, промяна на служител и изтриване на служител. Да се напише подходящ конзолен интерфейс.

