

Тема 3

• Байт - число от 0 до 255

ofstream of(out.dat, ios::out | ios::binary)

of.write(указ. към данните, колко байта да запише)
const char *

int a=16;

of.write((const char *) &a, sizeof(a))

* Четене от двубитен файл

```
int x;
ifstream if(....)
if.read()
```

• Двубитни файлове - нямат точна деф.
~ Ако срещне символ < 32 е двубитен файл

* Записване / четене на обекти / инстанции в дв. файл

```
struct A {
    bool b;
    int a;
}
```

0	1	1	1	a	a	a	a
---	---	---	---	---	---	---	---

A obj { false, 7 }

ofs.write((const char *) &obj, sizeof(A))

~ little endian не разменя последов. на елементите на структурата, а се прилага за \forall ст. по отделно

Hex view: 00cccccccc04cccccc

⚠ Записване на масив от инстанции в двоичен файл

A obj[4] = { ... }

`f.write((char *)obj, sizeof(A) * 4);`
 ↳ $\text{усп. е пойнтер на масив}$

$A^* \text{ агг} = \text{new } A[\text{elCount}]$

`f.write((char *)agg, elCount * sizeof(A))`

* Записване/четене - използване на външен ресурс

```
struct Person {
    char * name;
    int age;
}
```

```
{
    Person p;
    p.age = 3;
    p.name = "ABC";
    p.name = new char[5];
    strcpy(p.name, "Ivan");
}
```

→ като излезе от scope - а се уникува


```
Person * ptr = new Person;
ptr->name = new char [5]
strcpy(ptr->name, "Ivan")
delete[] ptr->name;
delete ptr;
```

* Писане и четене на масив от обекти от файл

В началото пишем големината на масива

• Работа с fstream - в интерф. му има функции до-ни на ifstream и ofstream



ofstream - изход
ifstream - вход
fstream - вход и изход

Δ При fstream има 1 указател \rightarrow tellg seekp() = seekg()

Δ Когато прилагаме входна операция след изходна трябва да обновим буфера