

# Структуры, указатели, модули, файлы

Юрий Литвинов  
yurii.litvinov@gmail.com

05.10.2018

# Структуры

- ▶ Способ группировки родственных по смыслу значений
- ▶ Структура — это тип
  - ▶ В памяти представляется как поля, лежащие друг за другом, возможно, с “дырками” (padding)

# Указатели и ссылки

- ▶ Указатель — адрес ячейки в памяти
- ▶ Ссылка — “синоним”, просто другое название для ячейки в памяти
  - ▶ Можно считать, что ссылка — это указатель, который не надо разыменовывать (и нельзя менять)
- ▶ Структуры и указатели настолько часто используются вместе, что есть оператор `->` (разыменовывать указатель на структуру и обратиться к её полю)
  - ▶ `Point *p = new Point { 10, 20 };  
printf("(%d, %d)", p->x, p->y);`  
Или  
`auto p = new Point { 10, 20 };  
printf("(%d, %d)", p->x, p->y);`
  - ▶ То же самое, что `(*p).x` и `(*p).y`

# Файлы

- ▶ Последовательность байтов на диске
  - ▶ Бывают “сырые” и “текстовые”
    - ▶ Самому файлу всё равно, это лишь способы интерпретации его содержимого
  - ▶ Режимы доступа: r, w, a, r+, w+, a+
  - ▶ Курсор
  - ▶ EOF
- ▶ Сишные функции
  - ▶ fopen, fclose, fprintf, fscanf, fseek, ftell, fgetc
- ▶ Файлы надо не забывать закрывать

# Модули

- ▶ Способ группировки кода в логически обособленные группы
- ▶ В C++ это реализуется с помощью заголовочных файлов и файлов с реализацией
  - ▶ .h и .cpp
- ▶ В отдельный модуль выносятся объявления типов данных и функции, которые делают одно дело
  - ▶ Например, разные функции сортировки
  - ▶ Или всё для работы с матрицами
- ▶ В интерфейсную часть модуля выносится только то, что может использовать другой код
  - ▶ Меньше знаешь — крепче спишь
- ▶ Функции, используемые только для реализации, пишутся только в .cpp-файле
  - ▶ Например, функция разделения массива для быстрой сортировки или `swar`

# Модули

Заголовочный файл:

```
#pragma once
```

```
// Комментарий к функции 1
```

```
int function1(int x, int y);
```

```
// Комментарий к функции 2
```

```
void function2();
```

.cpp-файл:

```
#include <имя заголовочного файла.h>
```

```
#include <все остальные библиотеки>
```

```
int function1(int x, int y)
```

```
{  
    ...  
}
```

```
void function2()
```

```
{  
    ...  
}
```