

# Программирование на языке C++

## Лекция 7

Указатели на методы и поля класса

Александр Смаль

## Указатели на методы: параметризация алгоритмов

Для вызова метода по указателю нужен объект.

```
struct Unit
{
    virtual unsigned id()    const;
    virtual unsigned hp()    const;
};

typedef unsigned (Unit::*UnitMethod)() const;

void sort(Unit* p, Unit* q, UnitMethod mtd)
{
    for (Unit * m = p; m != q; ++m)
        for (Unit * r = m; r + 1 != q; ++r)
            if ( (r->*mtd)() > ((r+1)->*mtd)() )
                swap(*r, *(r+1));
}

sort(p, q, &Unit::hp);
```

## Указатели на поля: параметризация алгоритмов

Для обращения к полю по указателю нужен объект.

```
struct Unit
{
    unsigned id;
    unsigned hp;
};

typedef unsigned Unit::*UnitField;

void sort(Unit* p, Unit* q, UnitField f)
{
    for (Unit * m = p; m != q; ++m)
        for (Unit * r = m; r + 1 != q; ++r)
            if ( (r->*f) > ((r+1)->*f) )
                swap(*r, *(r+1));
}

sort(p, q, &Unit::id);
```

# Резюме по синтаксису

Указатели на методы и поля класса.

```
struct Unit
{
    unsigned id() const;
    unsigned hp;
};
```

```
unsigned (Unit::*mtd)() const = &Unit::id;
unsigned  Unit::*fld          = &Unit::hp;
```

```
Unit u;
Unit * p = &u;
```

```
(u.*mtd)() == (p->*mtd)();
(u.*fld)   == (p->*fld);
```

Как такие указатели устроены?

Как такие указатели устроены?

Что хранится в указателе на функцию?

Как такие указатели устроены?

Что хранится в указателе на функцию?

Хранится адрес функции.

Как такие указатели устроены?

Что хранится в указателе на функцию?

Хранится адрес функции.

Что хранится в указателе на поле класса?



## Как такие указатели устроены?

Что хранится в указателе на функцию?

Хранится адрес функции.

Что хранится в указателе на поле класса?

Хранится смещение поля от начала объекта.

## Как такие указатели устроены?

Что хранится в указателе на функцию?

Хранится адрес функции.

Что хранится в указателе на поле класса?

Хранится смещение поля от начала объекта.

Что хранится в указателе на метод?

## Как такие указатели устроены?

Что хранится в указателе на функцию?

Хранится адрес функции.

Что хранится в указателе на поле класса?

Хранится смещение поля от начала объекта.

Что хранится в указателе на метод?

Там хранятся:

## Как такие указатели устроены?

Что хранится в указателе на функцию?

Хранится адрес функции.

Что хранится в указателе на поле класса?

Хранится смещение поля от начала объекта.

Что хранится в указателе на метод?

Там хранятся:

1. адрес метода,

## Как такие указатели устроены?

Что хранится в указателе на функцию?

Хранится адрес функции.

Что хранится в указателе на поле класса?

Хранится смещение поля от начала объекта.

Что хранится в указателе на метод?

Там хранятся:

1. адрес метода,
2. номер в таблице виртуальных методов,

## Как такие указатели устроены?

Что хранится в указателе на функцию?

Хранится адрес функции.

Что хранится в указателе на поле класса?

Хранится смещение поля от начала объекта.

Что хранится в указателе на метод?

Там хранятся:

1. адрес метода,
2. номер в таблице виртуальных методов,
3. смещение.

## Зачем нужно смещение?

```
struct Elf {  
    string secretName;  
};  
  
struct Archer {  
    unsigned arrows() const { return arrows_; }  
    unsigned arrows_;  
};  
  
struct ElfArcher : Elf, Archer {};  
  
void foo() {  
    ElfArcher ea;  
    unsigned (ElfArcher::*m)() = &Archer::arrows;  
    (ea.*m)();  
}
```

## Важные моменты



## Важные моменты

- Использование неинициализированных указателей на функции и методы влечёт неопределённое поведение.

## Важные моменты

- Использование неинициализированных указателей на функции и методы влечёт неопределённое поведение.
- Для использования указателей на методы и поля классов нужны экземпляры этих классов.

## Важные моменты

- Использование неинициализированных указателей на функции и методы влечёт неопределённое поведение.
- Для использования указателей на методы и поля классов нужны экземпляры этих классов.
- Указатели на методы и поля класса ни к чему не приводятся.

## Важные моменты

- Использование неинициализированных указателей на функции и методы влечёт неопределённое поведение.
- Для использования указателей на методы и поля классов нужны экземпляры этих классов.
- Указатели на методы и поля класса ни к чему не приводятся.
- Указатель на статический метод — это указатель на функцию, а указатель на статическое поле — это обычный указатель.

## Важные моменты

- Использование неинициализированных указателей на функции и методы влечёт неопределённое поведение.
- Для использования указателей на методы и поля классов нужны экземпляры этих классов.
- Указатели на методы и поля класса ни к чему не приводятся.
- Указатель на статический метод — это указатель на функцию, а указатель на статическое поле — это обычный указатель.
- В шаблонном коде указатель на функцию ведёт себя так же, как и объект класса с оператором `()`. Это позволяет использовать указатели на функции в качестве функторов.

## Важные моменты

- Использование неинициализированных указателей на функции и методы влечёт неопределённое поведение.
- Для использования указателей на методы и поля классов нужны экземпляры этих классов.
- Указатели на методы и поля класса ни к чему не приводятся.
- Указатель на статический метод — это указатель на функцию, а указатель на статическое поле — это обычный указатель.
- В шаблонном коде указатель на функцию ведёт себя так же, как и объект класса с оператором `()`. Это позволяет использовать указатели на функции в качестве функторов.
- Используйте `typedef!` `=`).