



ОНЛАЙН-ОБРАЗОВАНИЕ

# Разработчик C++

C++14 move, variadic templates, POD

Сергей Кольцов  
профессиональный программист



# Откуда?

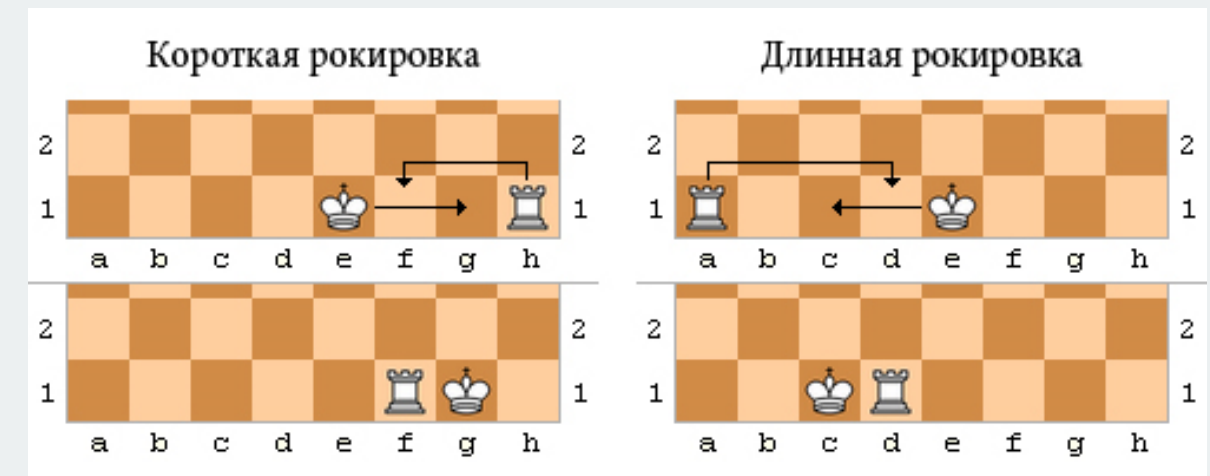


**ES.56: Write `std::move()` only when you need to explicitly move an object to another scope**

**<https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines>**



# Что если...



Несколько занятий (одно-два) проведёт другой преподаватель?

- П – я против
- Н – мне не важно
- К – было бы круто!
- ... – напишите свой вариант коротечко, в пару предложений.



# Что если...



Я попрошу вас объединиться в небольшие группы для выполнения задания

- П – я против
- Н – мне не важно
- К – было бы круто!
- ... – напишите свой вариант коротечко, в пару предложений.



# Каков наш план?

- ВСПОМНИМ
- decltype
- move
- param packs



# Вспомним lambda

```
[x] (int i) { return x * i; }
```

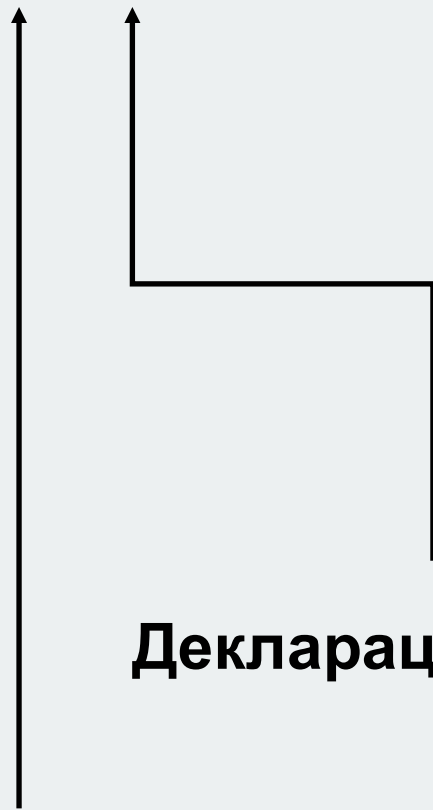


Список захвата (&, =)



# Вспомним lambda

```
[x] (int i) { return x * i; }
```



Декларация operator()

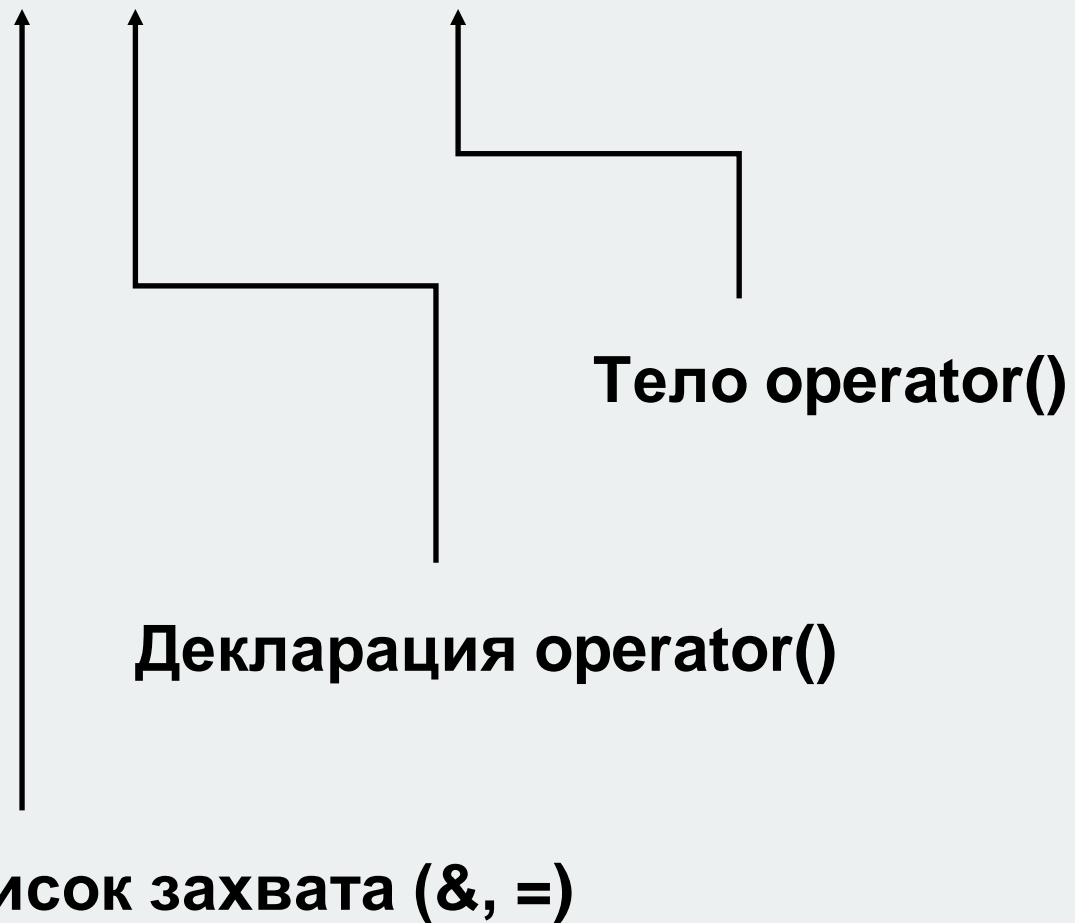
Список захвата (&, =)





# Вспомним lambda

```
[x] (int i) { return x * i; }
```



# Вспомним tuple

`std::tuple<double, int>(1.0, 2);`

`std::make_tuple(1.0, 2);`

`std::tie` - кортеж ссылок



# Вспомним auto

- Инициализация переменных:  
`auto i = 42;`
- Вывод возвращаемого типа:  
`auto foo() { return 42; };`
- Вывод типа параметра лямбды?  
`[] (auto arg) { return arg; }`



# decltype

- `decltype(expr)` - просто тип
- `decltype(auto)` - без ослаблений `auto`
- `decltype(foo())`
- `decltype(arr[0])`
- `auto foo(int x) -> decltype(x)`



# Online-компиляция



- <http://cpp.sh/>
- <https://wandbox.org/>
- <https://www.onlinegdb.com/>
- <https://godbolt.org/>



# Move-semantics

- rvalue vs lvalue
- дорогое копирование
- copy and ctor
- noexcept
- компилятор меня не понял, `static_cast`
- я не понял компилятор, `ctor`

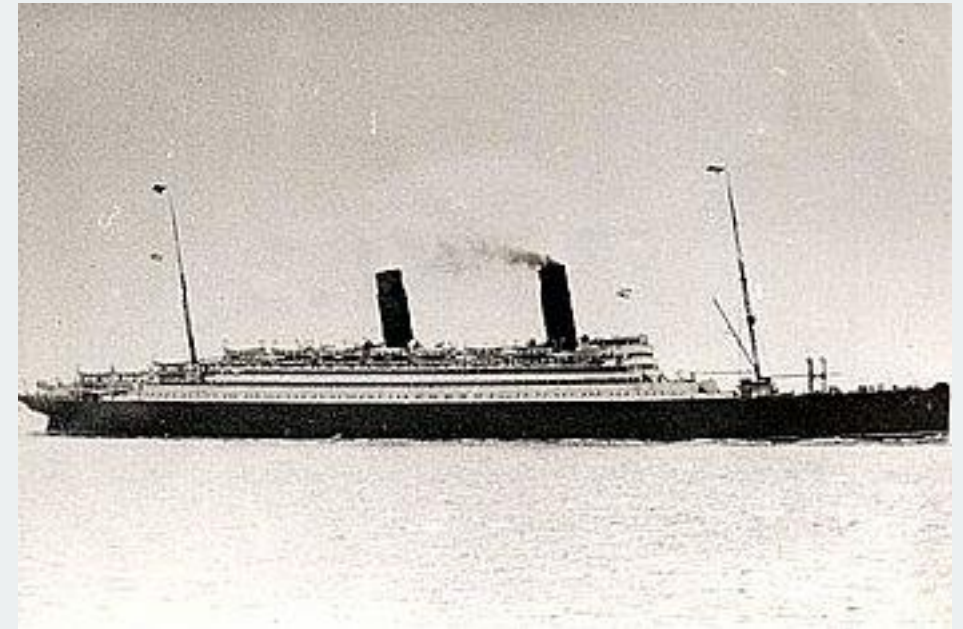




# Класс-помощник

```
struct Tracer {  
    std::string m_name;
```

```
    explicit Tracer(const char* name) : m_name(name) {  
        std::cout << "ctor " << m_name << std::endl;  
    }  
};
```



# Variadic templates

- parameter pack
- variadic template
- разворачивание
- шаблон разворачивания
- рекурсия или нет?







**Спасибо  
за внимание!**

# Ответы на вопросы

