

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





ОНЛАЙН-ОБРАЗОВАНИЕ



C++14

decltype, move, param packs

Дмитрий Шебордаев

Ведущий специалист



Сегодня

- ▶ вспомним `lambda`, `tuple`, `auto`
- ▶ `http://cpp.sh/`
- ▶ `decltype`
- ▶ `move`
- ▶ `param packs`



вспомним lambda, tuple, auto

lambda

► `[x](int i){return x * i;}`



вспомним lambda, tuple, auto

lambda

- ▶ `[x](int i){return x * i;}`
- ▶ `[x]` - конструктор и захват



вспомним lambda, tuple, auto

lambda

- ▶ `[x](int i){return x * i;}`
- ▶ `[x]` - конструктор и захват
- ▶ `(int i)` декларация оператора `()`



вспомним lambda, tuple, auto

lambda

- ▶ `[x](int i){return x * i;}`
- ▶ `[x]` - конструктор и захват
- ▶ `(int i)` декларация оператора `()`
- ▶ `{return x * i;}` тело



вспомним lambda, tuple, auto

tuple

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



вспомним lambda, tuple, auto

tuple

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



вспомним lambda, tuple, auto

tuple

- ▶ `std::tuple<double, int>(1.0, 2);`
- ▶ `std::make_tuple(1.0, 2);`
- ▶ `std::tie` - кортеж ссылок



вспомним lambda, tuple, auto

auto

```
► auto i = 42;
```



вспомним lambda, tuple, auto

auto

- ▶ `auto i = 42;`
- ▶ `auto foo() { return 42; };`



вспомним lambda, tuple, auto

auto

- ▶ `auto i = 42;`
- ▶ `auto foo() { return 42; };`
- ▶ а как же лямбда?



decltype

- ▶ `decltype(expr)` просто тип



decltype

- ▶ `decltype(expr)` просто тип
- ▶ `decltype(auto)` без ослаблений `auto`



decltype

- ▶ `decltype(expr)` просто тип
- ▶ `decltype(auto)` без ослаблений `auto`
- ▶ `decltype(foo()), decltype(arr[0])`



decltype

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



move

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



variadic template

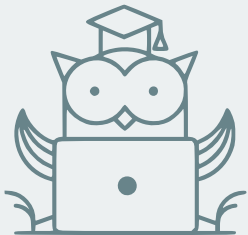
- ▶ parameter pack
- ▶ variadic template
- ▶ разворачивание
- ▶ шаблон разворачивания
- ▶ рекурсия или нет?



ПОЧИТАТЬ

- ▶ http://thbecker.net/articles/auto_and_decltype/section_01.html
- ▶ http://thbecker.net/articles/rvalue_references/section_01.html
- ▶ http://en.cppreference.com/w/cpp/language/parameter_pack





Спасибо за
внимание!

