



mcc

@mcclure111

Follow



In C++ we don't say "Missing asterisk" we say "error C2664: 'void std::vector<block,std::allocator<_Ty>>::push_back(const block &)': cannot convert argument 1 from 'std::_Vector_iterator<std::_Vector_val<std::_Simple_types<block>>>' to 'block &&'" and i think that's beautiful

4:30 PM - 1 Jun 2018

292 Retweets 926 Likes



ООП/C++: Лекция 9

Шаблоны

ака «Что под капотом у STL»

О чём лекция сегодня

1. Что такое шаблоны

2. Нюансы шаблонов

<https://github.com/avasyukov/oop-2nd-term/tree/master/2019/lection09>



http://judge2.vdi.mipt.ru/cgi-bin/new-client?contest_id=911146



Что такое шаблоны

Откуда взялись шаблоны

- Очень не хочется постоянно писать много одинакового кода
 - Логика ровно одна и та же
 - Отличаются только типы данных (или параметры)

Давайте как-нибудь от этого избавимся?

Что такое шаблоны

Если отличаются только типы данных, а алгоритм ровно один, пусть компилятор сам сгенерирует нужное количество однотипных функций/классов по шаблону

- Следствие 1: если алгоритм разный для разных данных – шаблоны применять не стоит
- Следствие 2: это именно генерация кода на стадии компиляции, опираться может только на то, что есть на этапе компиляции

Шаблоны: базовые примеры

Смотрим примеры:

- 01_simple_function.cpp
- 02_templated_function.cpp

Специализация шаблонов

- Если нужно – можно явно указать, какую сгенерированную реализацию вызвать
- Если нужно – можно написать разную логику для разных параметров шаблона

Внимание: это не штатные сценарии! Каждый раз, когда обнаруживаете такое в коде, нужно хорошо подумать, уместны ли тут шаблоны вообще.

Шаблоны: примеры странного

Смотрим примеры:

- 03_explicit_call.cpp
- 04_template_specialization.cpp

Нюансы шаблонов

Нюансы использования шаблонов

Автоматическая генерация кода имеет свои проблемы и ограничения

- Тяжело читаемые сообщения об ошибках из недр генерируемого кода
- Использование friend-ов требует аккуратности
- Использование одновременно с наследованием может оказаться небанальным
- При сложной структуре проекта компилятор может «не догадаться», что именно и для каких типов он должен был сгенерировать

Подробнее: <https://isocpp.org/wiki/faq/templates>

Шаблоны: примеры

Смотрим жизненные примеры:

- `05_simple_stack.cpp`
- `06_templated_stack.cpp`

Что стоит запомнить из лекции

- В C++ есть шаблоны. Технически это кодогенерация на этапе компиляции.
- Очень много вещей в C++ держится именно на шаблонах.
- У шаблонов своя специфичная область применения и ограничения. С ними нужно обходиться осторожно.

<http://tinyurl.com/y6f828jn>



**НЕЛЬЗЯ ПРОСТО ТАК
ВЗЯТЬ И ВЫУЧИТЬ ШАБЛОНЫ C++**

**ПО КРАЙНЕЙ МЕРЕ С ТЕХ ПОР, КАК БЫЛО
ОБНАРУЖЕНО, ЧТО ЭТО ОТДЕЛЬНЫЙ ФУНКЦИОНАЛЬНЫЙ
ЯЗЫК ПРОГРАММИРОВАНИЯ ВНУТРИ КОМПИЛЯТОРА C++**