

Вывод типов при использовании шаблонов, auto и decltype.

№ урока: 1 **Курс:** C++ Advanced

Средства обучения: Qt Creator

Обзор, цель и назначение урока

Научить студентов понимать и применять на практике базовые механизмы вывода типов C++, рассмотреть историю вывода типов, начиная с шаблонов, заканчивая auto & decltype.

Изучив материал данного занятия, учащийся сможет

- Понимать, что такое вывод типов C++.
- Уметь объяснить разницу между правилами вывода для шаблонов и auto.
- Понимать, что такое decltype.
- Знать особенности правил вывода в стандартах C++11, C++14, C++17.
- Знать ключевые особые правила для auto & decltype.

Содержание урока

1. Вывод типов C++
2. Вывод типов шаблонов
3. Вывод типов auto
4. decltype
5. Совместное использование auto и decltype
6. Средства определения типов переменных

Резюме

```
template<typename T>
void f ( ParamType param ) ; // ParamType – T& или T* в данном примере
```

f (expression);

ParamType:

Указатель или ссылка – для T игнорируется ссылочная/указательная часть:

```
int x= 27; // x имеет тип int
const int cx = x; // cx имеет тип const int
const int& rx = x; // rx является ссылкой на x как на const int
```

```
f(x); //T - int, тип param - int&f(cx); // T - const int, тип param - const int&
```

```
f(rx); // T - const int, тип param - const int&
```

ParamType:

Универсальная ссылка:

- Если expression представляет собой lvalue, то T и ParamType выводятся как lvalue- ссылки.
- В случае rvalue – обычные правила, как в предыдущем пункте

ParamType:

Не указатель, не ссылка

•Отбрасывается как константность, так и ссылочность – создается новый объект

Ссылка на массив

```
template<typename T, std::size_t N>
```

```
constexpr std::size_t arraySize (T (&) [N] ) noexcept
```

```
{ return N; }
```

Исключение для вывода типов auto – std::initializer_list.

Declared type

Основное применение decltype в C++11 - объявление шаблонов функций, в которых возвращаемый тип функции зависит от типов ее параметров.

Будучи именем, x представляет собой lvalue, и C++ также определяет выражение (x) как lvalue.

Следовательно, decltype (x) представляет собой int&.

Добавление скобок во-круг имени может изменить тип, возвращаемый для него decltype !

Закрепление материала

- Чем отличается вывод типов шаблонов от вывода типов auto?
- Что такое универсальная ссылка?
- Для чего необходим decltype?
- Как можно вычислить длину Сишного массива на этапе компиляции с помощью ссылок на массив?
- В каких случаях лучше всего использовать auto?

Дополнительное задание

Задание

Изучите особенности std::underlying_type, std::declval, typeid & typeid (C++filt).

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Рассмотрите разницу взаимодействия с std::initializer_list в стандартах C++11, C++14, C++17.

Задание 3

Зайдите на сайт MSDN.

Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

<https://en.cppreference.com/w/cpp/language/typeid>

<https://docs.microsoft.com/en-us/cpp/cpp/auto-cpp?view=vs-2017>

<https://www.oreilly.com/library/view/effective-modern-c/9781491908419/ch01.html>

<https://github.com/CppCon/CppCon2014/blob/master/Presentations/Type%20Deduction%20and%20Why%20You%20Care/C%2B%2B%20Type%20Deduction%20and%20Why%20You%20Care%20-%20Scott%20Meyers%20-%20CppCon%202014%20-%20CppCon%202014.pdf>

<https://en.cppreference.com/w/cpp/language/decltype>