C++ Advanced

Вывод типов при использовании шаблонов, auto и decltype.

Автор курса

Кирилл Чернега

После урока обязательно

Повторите этот урок в видео формате на ITVDN.com

Проверьте как Вы усвоили данный материал на TestProvider.com

Вывод типов при использовании шаблонов, auto и 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.

Вывод типов C++

Type inference or type deduction.

Automatically (auto).

```
int a = 5;
double b = 4.5;

auto c = a; // int
auto d  = 5ul; // unsigned long
```

Вывод типов шаблонов

```
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&
```

Вывод типов шаблонов

```
template<typename T>
void f ( T&& param ) ; // ParamType – T&&

f (expression);
```

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

```
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); // x – lvalue => T == param - int&
f(cx); // cx – lvalue => T == param – const int&
f (rx) ; // rx – lvalue => T == param – const int&

f(27); // 27 – rvalue => T – int, param – int&&
```

Вывод типов шаблонов

```
template<typename T>
void f ( T param ) ; // передача по значению

f (expression);
```

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; }
```

Вывод типов auto

Вывод типов шаблона == вывод типов auto

Исключение – std::initializer_list:

auto xl = 27; // Тип int, значение - 27

auto x2(27); // Тип int, значение – 27

Universal reference:

auto&& uref1 = x; //x - int и lvalue, => urefl - int& cx;

auto&& uref2 = cx; // cx - const int и lvalue, => uref2 - const int&

auto&& uref3 = 27; // 27 - int и rvalue, => uref3 - int&&

auto x3 = { 27 } ; // std::initializer_list<int>, значение (27}

auto x4{ 27 }; // std::initializer_list<int>, значение (27}

decltype, decltype(auto)

Declared type

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

decltype(auto):
auto указывает, что тип должен быть выведен,
а decltype говорит о том, что в процессе вывода следует использовать правила decltype.

Будучи именем, x представляет собой lvalue, и C++ также определяет выражение (x) как lvalue.
Следовательно, decltype ((x)) представляет собой int&.
Добавление скобок вокруг имени может изменить тип, возвращаемый для него decltype !

Средства определения типов переменных

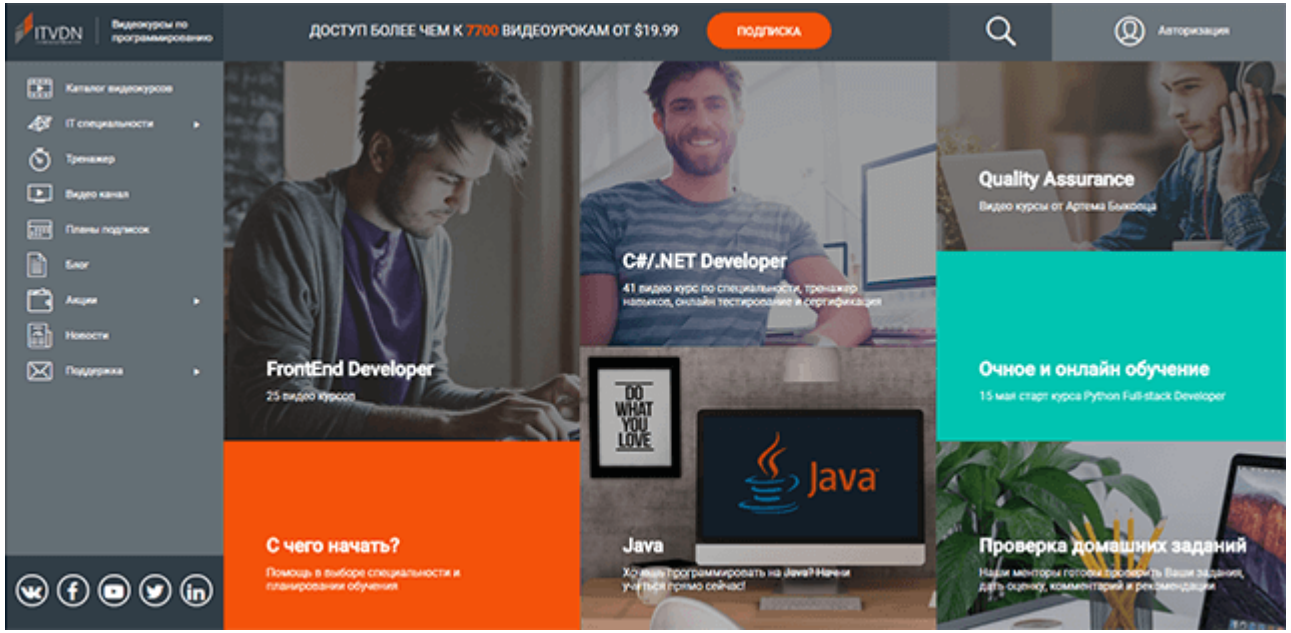
Выводимые типы часто можно просмотреть с помощью редакторов IDE, сообщений об ошибках компиляции и с использованием библиотеки Boost.TypeIndex.

Результаты, которые выдают некоторые инструменты, могут оказаться как неточными, так и бесполезными, так что понимание правил вывода типов в C++ является совершенно необходимым.

Материал по книге «Эффективный и современный C++» (Скотт Мейерс).

Смотрите наши уроки в видео формате

ITVDN.com



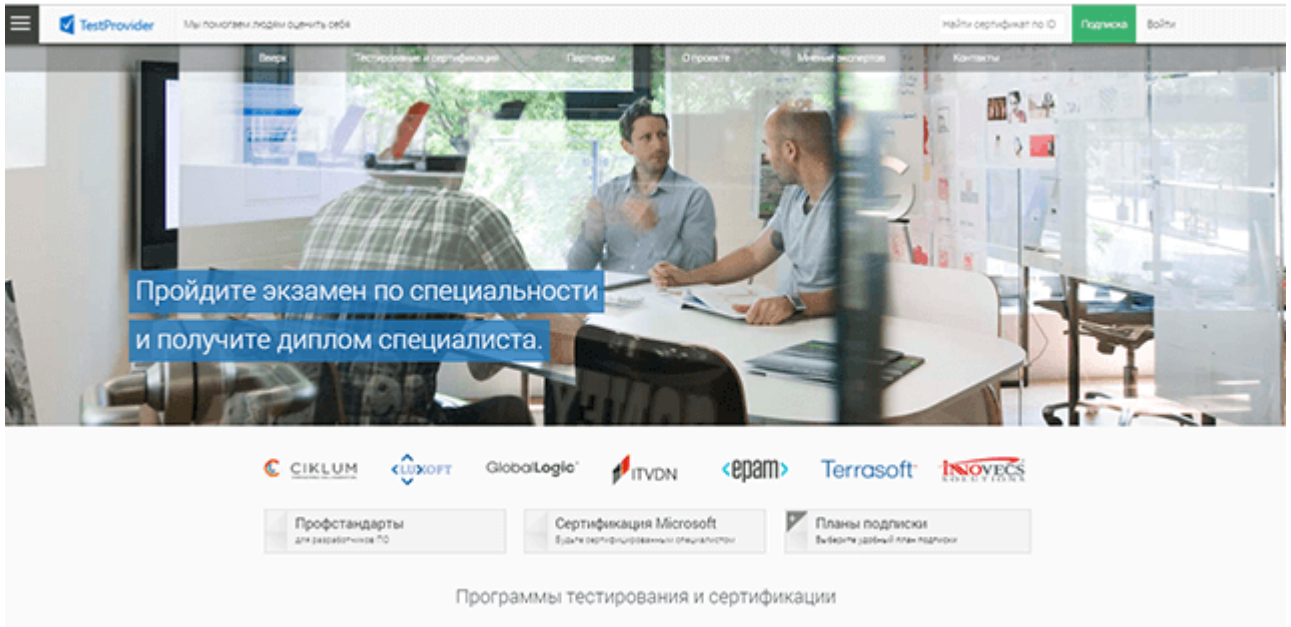
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.

ITVDN
IT VIDEO DEVELOPERS NETWORK

Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на TestProvider.com

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

TestProvider

Q&A

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association