



Microsoft Partner
Silver Learning

C++ Advanced

Функторы



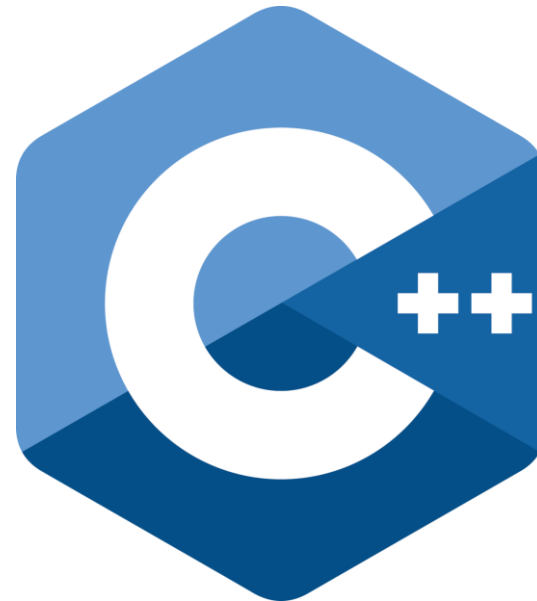
ITVVDN
IT VIDEO DEVELOPERS NETWORK

C++ Advanced

Автор курса



Кирилл Чернега



C++ Advanced

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

Функторы

C++ Advanced

Содержание урока

1. Что такое функторы?
2. Указатели на функции и их недостатки по сравнению с функторами.
3. `std::function`
4. Лямбда выражения (C++11, C++14).

C++ Advanced

Что такое функторы?

```
class Functor  
{  
  public:  
  void operator() ()  
  {  
    //some operation;  
  }  
};
```

Функтор, он же функциональный объект – экземпляр класса, где перегружен оператор круглых скобок.

C++ Advanced

Указатели на функции и их недостатки по сравнению с функторами

```
void (*func)(int a, int b);
```

```
int (MyClass::*pt2Member)(float, char, char);
```

Указатели на функции имеют следующий синтаксис – в целом, как у обычной функции, но перед именем функции ставится * и все вместе указывается в скобках, чтобы указатель не был присвоен к типу возвращаемого значения. Далее в скобках указывается сигнатура функции.

Указатели на функции и их недостатки по сравнению с функторами

```
char * ( *func (int, int) )( int, double );
```

Функция двух целочисленных параметров, возвращающая указатель на функцию, которая возвращает указатель на `char` и имеет собственный список формальных параметров вида: `(int, double)`

Функторы – так как они являются объектами, то можно расширять их описание класса, манипулировать данными и дополнительным поведением. Указатель на функцию – всего лишь указатель, мы не можем накладывать на него дополнительную логику. К тому же, этот указатель при обработке нужно проверять, как и любой указатель.

Недостаток указателей на функции – порой очень сложный синтаксис написания, а после – понимания того, как же должна выглядеть функция. Пример представлен слева.

C++ Advanced

Указатели на функции и их недостатки по сравнению с функторами

Замена с помощью typedef:

Было:

```
char * ( *func (int, int) )( int, double );
```

Стало:

```
typedef char* (*SomeFunction)(int, double);
```

```
SomeFunction func(int, int);
```

C++ Advanced

std::function

Шаблон класса `std::function` это полиморфная обёртка функции общего назначения. Экземпляры `std::function` могут хранить, копировать, и ссылаться на любой вызываемый объект - функцию, лямбда-выражение, привязку выражения (`std::bind`) или другой объект-функцию.

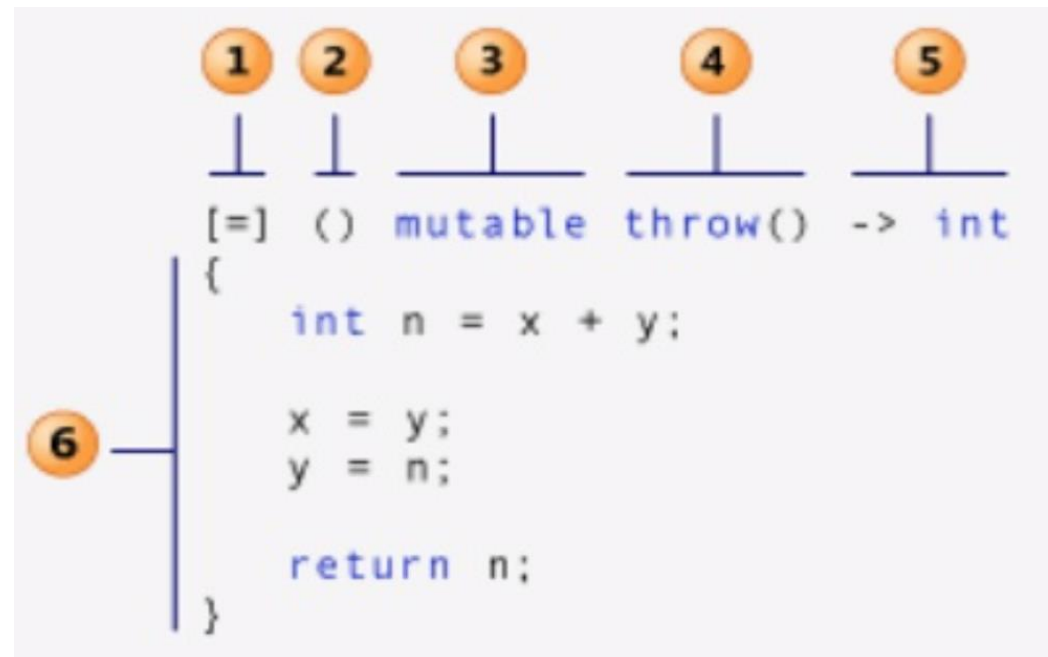
C++ Advanced

Лямбда-выражения

Лямбда – анонимная функция, то есть та, которая не имеет имени.

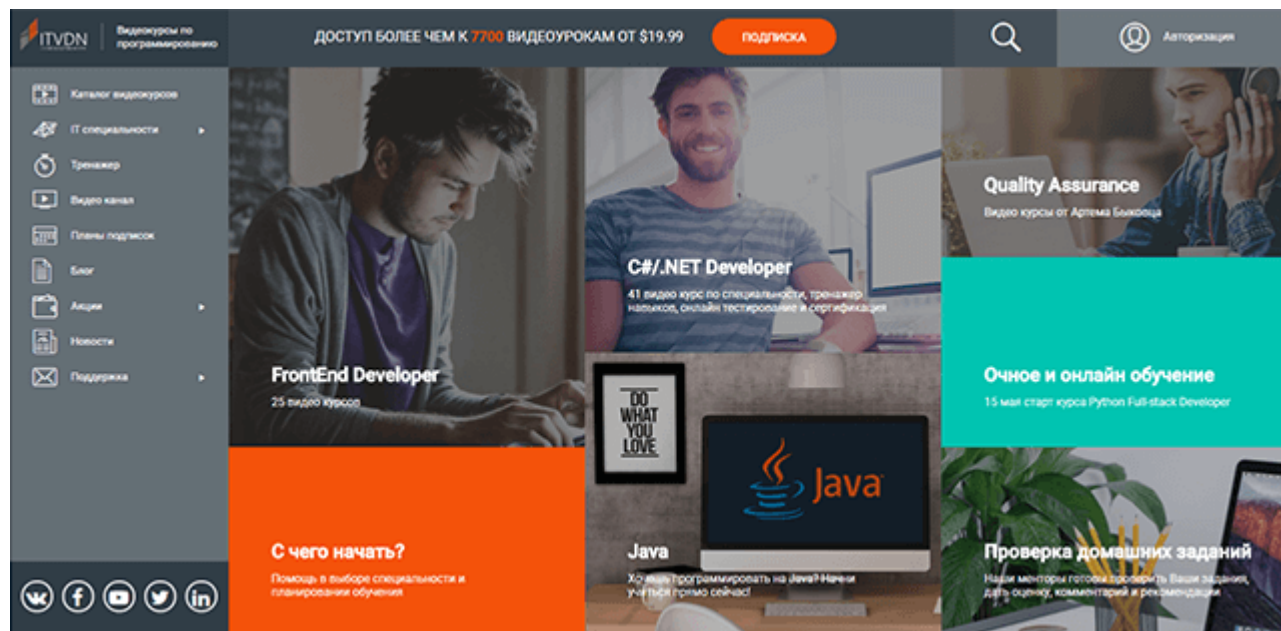
Главная отличительная черта – область захвата в лямбда-выражение.

Минимальная запись лямбды: `[]{}`



Смотрите наши уроки в видео формате

ITVDN.com



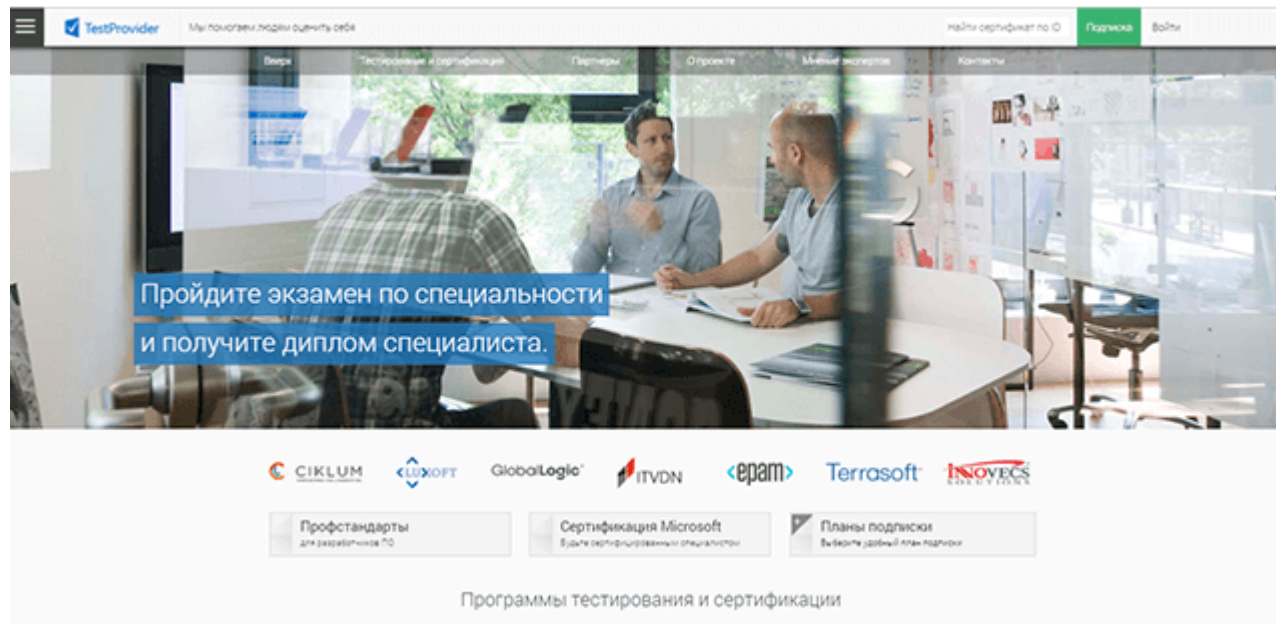
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

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



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](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Q&A

Информационный видеосервис для разработчиков программного обеспечения

