

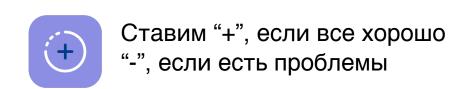


C++ developer. Professional Сравнение C++11/C++14 с C++ 03

otus.ru

• REC Проверить, идет ли запись

Меня хорошо видно && слышно?



Тема вебинара

Сравнение С++11/С++14 с С++ 03



Карина Дорожкина

Research Development Team Lead

10 лет опыта разработки на С/С++.

Долгое время занималась развитием ПО в области безопасности транспортного

Имею опыт руководства несколькими командами и проектами с разнообразным стеком технологий.

Спикер конференций С++ Russia, escar Europe.

dorozhkinak@gmail.com

Правила вебинара



Активно участвуем



Off-topic обсуждаем в telegram Otus C+ +-2023-08



Задаем вопрос в чат



Вопросы вижу в чате, могу ответить не сразу

Агенда

- **1.** move семантика (C++11)
- 2. list initialization (C++11/C++14)
- default, delete, override, final (C++11)
- 4 constexpr (C++11/C++14)
- 5 static_assert (C++11)
- 6 enumeration (C++11)
- 7 nullptr (C++11)
- 8 using (C++11)
- 9 User-defined literals (C++11)
- 10 Binary literal (C++14)
- 11 Ranged-based for loop (C++11)
- 12 Constructor delegation (C++11)



Список нововведений по стандартам

https://github.com/AnthonyCalandra/modern-cpp-features

Как будут сконструированы строки?

```
std::string s(48, 'a');
std::string s{48, 'a'};
```

Скомпилируется ли код?

```
struct Base
{
    virtual void Function(int a, double b) {};
};

struct A : Base
{
    int Function(int a, double b) override {}
};
```

Скомпилируется ли код?

```
enum class Color
    red,
    green,
    blue
};
void ProcessColor(Color color)
void func()
{
    ProcessColor(red);
```

В чем разница?

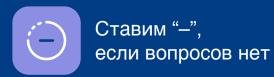
```
int foo()
{
    return 6;
}

constexpr int l = foo();
const int k = foo();
```

Спасибо за внимание!

Вопросы?





Не забудьте принять участие в опросе

Следующий вебинар



15 сентября 2023

Реализация умных указателей

