C++ Club UK Meeting 117

Gleb Dolgich

2020-11-26

WG21 Index

https://wg21.link/index.html

C#-like events in C++

Reddit

(Unrelated: Variable name prefixes)

- Code on Pastebin
- Signal-Slot library benchmarks
- Boost.Signals2

Uses of immediately invoked function expressions (IIFE) in C++

- Erik Rigtorp
- Jonathan Müller

Overloading by Return Type in C++

Philip Trettner

- Reddit

```
1  struct to_string_t {
2    std::string_view s;
3
4    // int  from_string(std::string_view s);
5    operator int() const;
6    // bool from_string(std::string_view s);
7    operator bool() const;
8  };
9
10 int i = to_string_t{"7"};
11 bool b = to_string_t{"true"};
```

The Defold game engine code style

Article

Code style

- C-like C++
- No classes (*huh? – GD*)
- No exceptions
- No STL
  - Custom containers
- Data ownership tracking
- C++98

(*It's the end of 2020, by the way. – GD*)

A Buffers Library for C++20

Colby Pike

vcpkg: Accelerate your team development environment with binary caching and manifests

Microsoft

- Reddit

Related

Why is it such an abysmal pain to use libraries in C++ compared to pretty much anything else?

Raymond Chen on structured bindings

- Structured binding may be the new hotness, but we'll always have `std::tie`
  - Reddit
- How to add C++ structured binding support to your own types
  - Reddit

dont_deduce

- artificial::mind
  - Reddit

C++11

```
template <class T> struct foo_t { using type = T; };
template <class T> using foo = typename foo_t<T>::type;
```

C++20

```
template <typename T>
auto operator+(
  vec3<T> const& a,
  std::convertible_to<T> auto const& b
) -> vec3<T>;
```

Calendar and Time-Zones in C++20: Time of Day

Rainer Grimm

FTL - A functional template library for containers processing in C++

- GitHub
- Reddit

FTL is a C++17-based library that provides a more convenient API for collections processing. The API is heavily inspired by the Rust programming language.

```
const auto totalVolume = cityTrees.iter()
  .filter([](c onst auto &tree) { return tree.kind ==
      Tree::Oak && tree.diameter > 25; })
  .map([](const auto &tree) { return tree.volume(); })
  .sum();
```