

Modernes C++

...für Programmierer

Unit 06: Attribute

by

Dr. Günter Kolousek

Überblick

- ▶ `[[deprecated]]`
- ▶ `[[maybe_unused]]`
- ▶ `[[nodiscard]]`
- ▶ `[[fallthrough]]`
- ▶ `[[noreturn]]`

`g++ -Wall -Wextra -std=c++2a ...`

[[deprecated]]

```
#include <iostream> // deprecated.cpp  
using namespace std;
```

```
[[deprecated]] int f() { return 1; }
```

```
[[deprecated("g() is legacy; use h() instead")]]  
int g() { return 2; }
```

```
int h() { return 3; }
```

```
int main() {  
    cout << g() << endl; // warning!  
}
```

2

[[nodiscard]] und [[maybe_unused]]

```
#include <iostream> // nodiscard_unused.cpp
using namespace std;
```

```
[[nodiscard]]
int f(int x) { return x; }
```

```
// warning: unused parameter...
```

```
void g(int y) { cout << 42 << endl; }
```

```
void h([[maybe_unused]] int y) { cout<<42<<endl; }
```

```
int main() {
    f(42); // warning: ignoring return value...
    g(1);
    h(1);
}
```

[[fallthrough]]

```
#include <iostream> // fallthrough.cpp
using namespace std;
int main() {
    enum class X {A, B, C, D};
    X x{X::A};
    switch (x) {
        case X::A:
            cout << "A" << endl;
            break;
        case X::B:
            cout << "B" << endl;
            [[fallthrough]];
        case X::C:
            // warning: this statement may fall through
            cout << "C" << endl;
        case X::D:
            cout << "D" << endl;
    }
}
```

[[noreturn]]

```
#include <cstdlib> // fallthrough.cpp
#include <string>
using namespace std;

// UB if f() returns!
[[noreturn]] void f(bool x){
    if (x)
        throw "error";
}

[[noreturn]] void usage(std::string msg) {
    std::exit(1);
}

int main() {
}
```