

HardCore in Programming

[진리는어디에/C++20](#)

[C++20] 코루틴(Coroutine) - co_return

kukuta 2021. 5. 4. 00:43

코루틴을 종료하기 위해선 co_return을 호출한다.

```
1 Task foo() {  
2     co_return;  
3 }
```

co_return은 값을 리턴 할 수도 있고, 아무런 값도 리턴하지 않고 단순히 코루틴을 종료하기만 할수도 있다. 값을 리턴하느냐 그렇지 않느냐에 따라 코루틴 함수의 리턴 타입이 가져야 하는 요구사항이 달라진다. 코루틴의 리턴 타입에 어떠한 것들이 필요한지 아래 예제를 통해 알아 보도록 한다.

컴파일러는 co_return 키워드를 만나면 아래와 비슷한 코드를 생성한다.

```
1 Task foo()  
2 {  
3     promise.return_void();  
4     co_await promise.final_suspend();  
5 }
```

컴파일러는 co_return 대신에 promise와 co_await를 생성한다. foo() 함수는 아무것도 리턴하지 않으므로 코루틴을 종료하기 전에 return_void()를 호출 한다. 만일 아래와 같이 값을 리턴하는 코루틴이라면

```
1 Task<int> foo() {  
2     co_return 7;  
3 }
```

promise의 return_value() 함수를 대신 호출 한다.

```
1 Task<int> foo()  
2 {  
3     promise.return_value(7);  
4     co_await promise.final_suspend();  
5 }
```

참고로 한 promise 타입에 이 두 함수는 동시에 구현 될수 없다. 값을 리턴하던지, 그렇지 않던지 둘중에 하나만 택해야 한다.

결론은 `co_return`을 사용하기 위해서는 리턴 값을 유무에 따라 `promise_type`에 `void return_void()` 또는 `void return_value(Type value)` 를 추가해 주어야 한다.

코루틴이 `co_return` 에 도달하면 다음과 같이 처리 된다

- `co_return`을 만나면 스택에 생성 되었던 모든 변수들이 생성 반대 순서로 해제 된다.
- `promise.return_value(exprt)` 이나 `promise.return_void()` 후에 `co_await promise.final_suspend()` 가 호출 된다
NOTE - `final_suspend()`가 호출 된다는 것은 코루틴의 완전 종료를 의미하고 `final_suspend()` 호출 이후 코루틴 객체의 `done()`함수를 호출하면 `true`가 리턴 된다.



공감

구독하기

태그 #c++20, #Coroutine, #co_return, #코루틴

'진리는어디에/C++20' Related Articles

한국어

속성

sogseong

NO IMAGE

C++20

2020

Core Language

• `constexpr` comparison operator
• `constexpr` literals as template arguments
• `constexpr` virtual functions
• `constexpr` definition of volatile
• `constexpr` signed initializers
• `constexpr` lambda improvements
• `constexpr` standard attributes
• `constexpr` `constexpr` and `constexpr` word
• `constexpr` `constexpr` location

Library

• `constexpr` Calendar and time-zone
• `constexpr` `std::span` as a view on a contiguous array
• `constexpr` containers such as `std::string` and `std::vector`
• `std::format`

• `std::`
• `std::`
• `std::`
• `std::`
• `std::`
• `std::`
• `std::`
• `std::`

NO IMAGE

[C++20] 어트리뷰트 (Attribute)

[C++20] Feature Test Macro

[C++20] Concepts

[C++20] Ranges



유익한 글이었다면 공감(♥) 버튼 꼭!! 추가 문의 사항은 댓글로!!

이름

암호

☐ Secret

여러분의 사랑과 관심이 한 사람을 살립니다. 댓글 한번만 남겨 주세요. 관심 받고 싶습니다!! 하얏하얏!!

댓글달기