Kotlin Scope Function

20191629 이주허

Kotlin의 표준 라이브러리는 "scope function"이라고 불리는 함수 몇 가지를 제공한다. let, run, with, apply, also가 이 함수에 해당하는데, 세부적인 동작에 차이는 있지만 이 다섯 함수 모두 인자로 주어진 함수를 다른 객체의 스코프에서 실행한다는 공통점이 있다. 이 중, also, with, run 함수의 동작 원리에 대해 알아본다.

각 함수에 대하여 Kotlin Standard Library의 원본 소스코드를 제시하고 분석하되, 각 함수의 동작을 이해하는 데 불필요한 KDoc 주석, 어노테이션과 Kotlin Contracts 구현체는 삭제하였다.

also 함수

```
public inline fun <T> T.also(block: (T) -> Unit): T {
   block(this)
   return this
}
```

also 함수는 타입 T와 T를 인수로 받아 아무것도 반환하지 않는 함수를 인자로 받는 제네릭확장 함수(generic extension function)이다. 즉, 어떤 타입이 also 함수를 정의하지 않더라도 kotlin 패키지에 의해 이 "메서드"가 정의된다. 여기서 타입 인수 T는 컴파일러에 의해 추론되어 also 함수를 사용할 때 명시할 필요는 없다.

이 함수는 자신이 호출된 객체를 인자로 받은 함수의 첫 번째 인자로 넘겨서 호출한 뒤, 자신이 호출된 객체를 반환한다. 따라서, 이 함수는 객체의 상태를 변경하고 싶을 때 주로 사용하며, this를 다시 반환한다는 점에서 함수 여러 개를 엮어서 사용할 수 있다는 장점이 있다. also 함수에서는 인자를 통해 다른 객체의 스코프에 접근한다. 즉, block 인자에 람다 함수를 준 경우, 람다 함수 안의 기본 인자 변수명인 it를 사용해서 스코프를 바꿀 수 있다.

with 함수

```
public inline fun <T, R> with(receiver: T, block: T.() -> R): R {
    return receiver.block()
```

with 함수 역시 제네릭 확장 함수이나, T 타입 객체와 그 객체의 "람다 메서드"를 인자로 받는다. 타입 인수 T, R의 값 모두 위와 마찬가지로 컴파일러가 자동으로 추론한다.

이 함수는 also 함수와 달리 스코프를 제공하는 객체가 아닌, 인자로 주어진 블록의 반환값을 그대로 반환한다. 따라서 이 함수는 receiver 객체보다 그 객체를 사용하는 작업의 결과가 더중요할 때 주로 사용한다.

여기서 주목할 점은 block 인자가 일반 함수 타입이 아닌 "메서드 타입"이라는 점이다. block은 일반적인 함수가 아니라, 타입 T를 this로 갖는 "확장 메서드" 타입이므로, 다른 객체의 스코프에 접근할 때 함수의 인자가 아닌 this 문맥을 사용해서 스코프에 접근한다.

run 함수

```
public inline fun <R> run(block: () -> R): R {
    return block()
}

public inline fun <T, R> T.run(block: T.() -> R): R {
    return block()
}
```

run 함수는 Kotlin에서 제공하는 스코프 함수 중 유일하게 오버로딩이 되어 있는 함수이다. 이함수는 어떤 타입의 값을 반환하는 함수를 인자로 받는데, 어떤 객체의 멤버로서 호출될 수도 있고, 어떤 객체에도 종속되지 않고 혼자서 호출될 수도 있다. 하지만 여기에서는 스코프를 바꾸는 기능을 가진 아래쪽 정의만 살펴본다.

아래쪽 정의를 보면, with 함수와 같은 일을 한다는 점을 알 수 있다. 다만 run 함수와 with 함수의 차이점은 호출 방식이 다르다는 것이다. with 함수는 스코프에 접근할 객체를 인자로 받는 반면, run 함수는 스코프를 변경할 함수를 this 문맥으로 받는다.

어떤 객체에 대한 작업인지에 따라 사용할 함수가 달라지겠지만, run 함수는 사용할 스코프를 this 문맥으로 받는다는 점에서 다른 스코프 함수와 함께 사용하기 좋다. 반대로 with 함수는 어떤 객체에 대한 작업을 수행하나, 다른 스코프 함수와 묶을 필요가 없을 때 주로 사용한다.