



section1-class01-람다 표현식



구현을 위한 사전 지식 (1)



구현을 위한 사전 지식 (1)

✓ 함수형 인터페이스(Functional Interface)

1 함수형 인터페이스란?

2 함수형 인터페이스의 형태

3 함수형 인터페이스 예제 코드 설명

✓ 람다 표현식(Lambda Expression)

1 람다 표현식이란?

2 람다 표현식의 기본 구조

3 람다 표현식과 함수형 인터페이스의 관계

✓ 함수 디스크립터(Function Descriptor)

✓ 함수형 인터페이스(Functional Interface)

1 함수형 인터페이스란?

- 구현해야 할 추상 메서드가 딱 하나만 있는 인터페이스
- Java 8 이전에 있었던 인터페이스 + Java 8 이후부터 새로 추가된 인터페이스

2 함수형 인터페이스의 형태

익명 구현 클래스 -> 함수형 인터페이스 -> 람다 표현식 -> 메서드 레퍼런스

3 함수형 인터페이스 예제 코드 설명

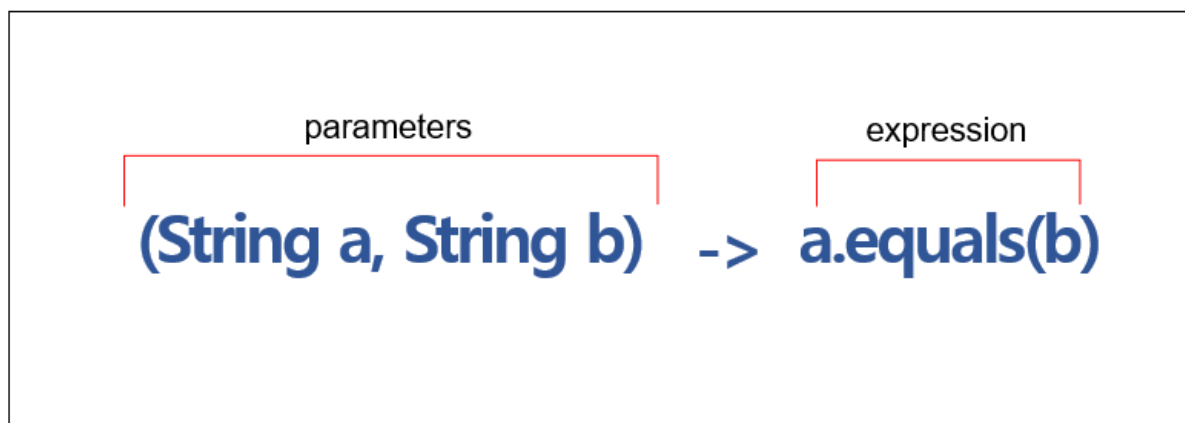
- 함수형 인터페이스 예제 코드 링크
 - <https://github.com/ITVillage-Kevin/mini-project-p1-s1-stream-examples/tree/main/src/main/java/com/itvillage/section01/class01>

✓ 람다 표현식(Lambda Expression)

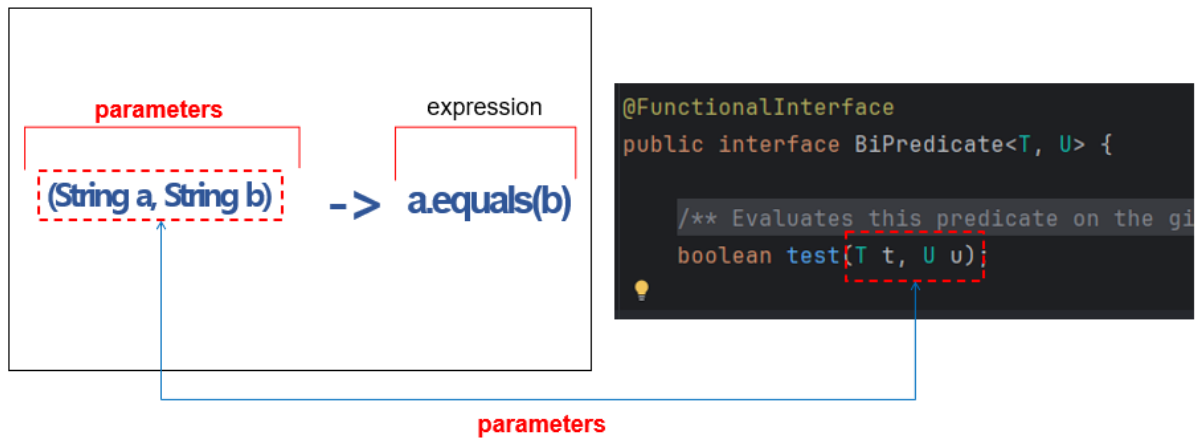
1 람다 표현식이란?

- 함수형 인터페이스를 구현한 **익명 클래스 구현 기법을 단순화 한 표현식**이다.

2 람다 표현식의 기본 구조



3 람다 표현식과 함수형 인터페이스의 관계



✓ 함수 디스크립터(Function Descriptor)

- 함수형 인터페이스의 추상 메서드를 설명해 놓은 시그니처
- Java 8부터 java.util.function 패키지에서 다양한 함수형 인터페이스를 제공하며, 제공되는 함수형 인터페이스가 없다면 직접 만들어서 사용할 수 있다.

함수형 인터페이스	함수 디스크립터(Function Descriptor)
Predicate<T>	T -> boolean
Consumer<T>	T -> void
Function<T, R>	T -> R
Supplier<T>	() -> T
BiPredicate<L, R>	(L, R) -> boolean
BiConsumer<T, U>	(T, U) -> void
BiFunction<T, U, R>	(T, U) -> R