

java

23강_람다식

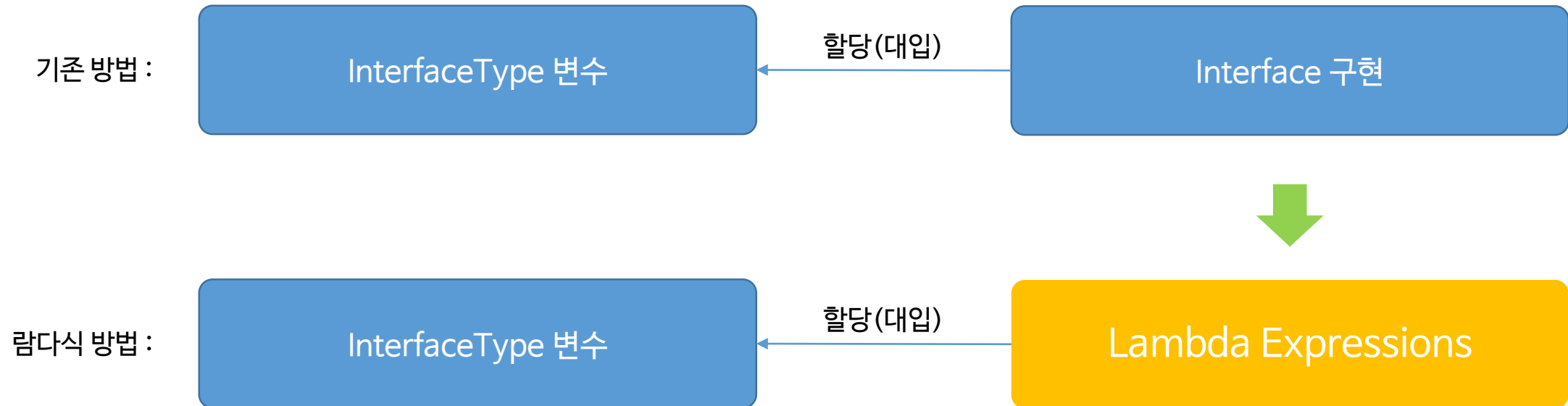
기존의 객체 지향이 아닌 함수 지향으로 프로그래밍 방법에 대해서 학습합니다.

23-1 람다식이란?

23-2 람다식 구현

23-1 : 랴다식이란?

익명 함수 (anonymous function)를 이용해서 익명 객체를 생성하기 위한 식이다.



23-2 : 랴다식 구현

람다식은 기본적으로 함수를 만들어 사용한다고 생각하면 된다.

```
// 매개변수와 실행문만으로 작성한다.(접근자, 반환형, return 키워드 생략)
LambdaInterface1 li1 = (String s1, String s2, String s3) ->
{ System.out.println(s1 + " " + s2 + " " + s3); };
li1.method("Hello", "java", "World");
```

```
System.out.println();
```

```
// 매개변수가 1개이거나 타입이 같을 때, 타입을 생략할 수 있다.
LambdaInterface2 li2 = (s1) -> { System.out.println(s1); };
li2.method("Hello");
```

```
// 실행문이 1개일 때, '{}'를 생략할 수 있다.
LambdaInterface2 li3 = (s1) -> System.out.println(s1);
li3.method("Hello");
```

```
// 매개변수와 실행문이 1개일 때, '()'와 '{}'를 생략할 수 있다.
LambdaInterface2 li4 = s1 -> System.out.println(s1);
li4.method("Hello");
```

```
// 매개변수가 없을때, '()'만 작성한다.
LambdaInterface3 li5 = () -> System.out.println("no parameter");
li5.method();
```

```
// 반환값이 있는 경우
LambdaInterface4 li6 = (x, y) -> {
int result = x + y;
return result;
};
System.out.printf("li6.method(10, 20) : %d\n", li6.method(10, 20));
```

```
li6 = (x, y) -> {
int result = x * y;
return result;
};
System.out.printf("li6.method(10, 20) : %d\n", li6.method(10, 20));
```

```
li6 = (x, y) -> {
int result = x - y;
return result;
};
System.out.printf("li6.method(10, 20) : %d\n", li6.method(10, 20));
```