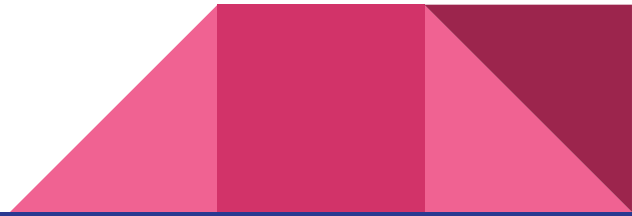
Java 기초 - 람다식

백성애

Lambda Expression - 람다식

- 자바는 객체 지향 프로그래밍 언어. 기능이 필요하다면 클래스를 먼저 만들고 클래스 안에 기능을 구현한 메서드를 만든 뒤 그 메서드를 호출해야 함
- Java 8부터는 함수형 프로그래밍을 지원

- 함수형 프로그래밍이란?
- 함수의 구현과 호출만으로도 프로그램을 만드는 프로그래밍 방식
- 자바에서 제공하는 함수형 프로그래밍 방식을 Lambda Expression이라고 한다



람다식 구현 방식

- 함수 이름이 없는 **익명 함수**를 만드는 것
-

```
(매개변수 ) -> { 실행문; }
```

```
public int add(int x, int y){
    return x+y
}
```

```
(int x, int y) ->{ return  x+y}
```

람다식 문법

[1] 매개변수 자료형과 괄호 생략하기

- 매개변수의 자료형을 생략할 수 있다.
- 매개변수가 하나인 경우 괄호도 생략 가능하다

str->{ System.out.println(str);}

- 매개변수가 2 개 이상인 경우는 괄호를 생략해서는 안된다.

x, y ->{ System.out.println(x+y);} [x]

(x,y)->{ System.out.println(x+y);} [o]

람다식 문법

[2] 중괄호 생략하기

- 중괄호 안의 구현 부분이 한 문장인 경우 중괄호를 생략할 수 있다.

  str -> System.out.println(str);

- 하지만 중괄호 안의 구현 부분이 한 문장이더라도 return 문은 중괄호를 생략할 수 없다.

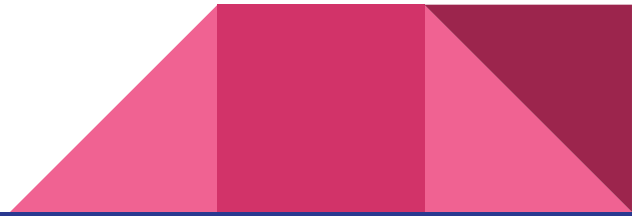
  str -> return str.length(); [x]

  str ->{ return str.length(); } [o]

람다식 문법

[3] return 생략하기

- 중괄호 안의 문장이 return문 하나라면 중괄호와 return을 모두 생략하고 식만 쓸 수 있다.
- (x,y) -> x+y
- str -> str.length()



[4] 함수형 인터페이스에는 하나의 메서드만 선언

- 람다식은 메서드 이름이 없고 메서드를 실행하는데 필요한 매개변수와 이를 활용한 실행 코드를 구현하는 것이다.
- 자바는 참조 변수 없이 메서드를 호출할 수 없으므로 함수형 인터페이스를 만들고 인터페이스에 람다식으로 구현할 메서드를 선언하는 것.
- 따라서 함수형 인터페이스에는 2 개 이상의 메서드를 가져서는 안된다.
-
- 혹시 2개 이상의 메서드를 선언하는 실수가 생길 수 있으므로

  @FunctionalInterface라는 어노테이션을 사용한다.

```
@FunctionalInterface
interface MyNum{
        int getMax(int a, int b);// 추상 메서드 선언
        //함수형 인터페이스에는 추상메서드는 1개만 있어야 한다
}
```

람다식은 단순히 메서드를 선언하는 것이 아니라, 이 메서드를 갖고있는 객체를 생성해낸다.

인터페이스 변수 = 람다식

```
MyNum my=(x, y)-> (x>=y)? x: y;
System.out.println(my.getMax(3, 10));
```

람다식에서의 지역변수 사용

- 람다식 실행 블럭에서 클래스의 멤버(멤버변수와 멤버메서드)를 제약없이 사용할 수 있다.
- 하지만 지역변수를 사용할 경우 이 지역변수는 final 특성을 가지고 있으므로 지역변수의 값을 수정할 수는 없다

- 즉 지역변수를 람다식으로 읽는 것은 허용되나,

  람다식 내부 또는 외부에서 변경할 수는 없다.

-