

Java Lambda Expressions

Java 8 ile gelen ve Java 8'in en büyük yeniliklerinden biri olan Lambda Expressions amacı, fonksiyonel programlamayı kolaylaştırıp daha az kod ile hızlı yazılım geliştirmeyi sağlamaktır. Lambda Expressions , Stream API pipeline ile çok çekirdekli ortamların paralel işlem özelliklerinden yararlanır. Lambda Expressions, parametre alabilen ve bir değer döndürebilen kod bloklarıdır. Bir diğer deyişle temel olarak Functional Interface örnekleridir. Functional Interface ise bir tane abstract metodu olan interfacedir. Lambda Expressions tek başına yürütülmez, Functional Interface'ten tanımlanan bir metodu implement edebilmek için kullanılır. Aslında Lambda Expressions, metotlara benzerler fakat bir isimleri olmasına gerek yoktur (anonim) ve metotların gövdesinde implement edilir.

Tanımlama

Parametresiz tanımlanırsa;

- **() -> expressions**

Parametre ile tanımlanırsa tanımlanırsa;

- **(birinciParametre, ikinciParametre) -> expressions**

Parametresiz Örnek

- **void merhaba() {**
 System.out.println("Merhaba");
 }

Yukarıdaki metot Lambda Expression ile aşağıdaki gibi yazılır.

- **() -> System.out.println("Merhaba");**

() : Parametreler burada yazılır. Eğer parametre yoksa boş bırakılır.

-> : Arrow-token olarak adlandırılır. Parametreler ile expression arasında kullanılır.

System.out.println("Merhaba"): Body yani gövdedir. Expression içerir.

Lambda Expression’da dikkat edilmesi bir konu metodun isimsiz yani anonim bir şekilde oluşturulmasıdır. Lambda Expression kullanılmadığı zaman merhaba ismini alan metod Lambda Expression kullanıldığı zaman anonimdir.Bu örnekte Lambda Expression parametre almamıştır bu yüzden parantezlerin içi boştur.

Parametrelili Örnek

Öncelikle bir interface oluşturulur.Interface içinde bir abstract metod oluşturulur.Oluşturulan abstract metod **mathOperation**.Bu metod bir tane parametre alır.Sonra main içinde Lambda Expression kullanılır.Bu örnekte tek ifadeli oluşturulmuştur.

```
1 package lambdaExpressions;
2
3
4 interface Operation{
5     boolean mathOperation(int number);
6 }
7
8 public class LambdaExp {
9     public static void main(String[] args) {
10         Operation isOddNumber = (number) -> (number % 2) !=0;
11         System.out.println(isOddNumber.mathOperation(-3));
12
13     }
14
15 }
16
```

Şekil 1:Parametrelili Örnek Kod

true

Şekil 2:Output

Blok Lambda Expression Örnek

Bu örnekte birden fazla ifade (expression) bulunmaktadır.Birden fazla ifadenin yer aldığı kodlarda bu ifadeler { } parantezler içinde yazılmaktadır.İfadeler yazıldıktan sonra kapatılan } parantezden sonra noktalı virgül (;) kullanılmalıdır.

```

interface Operation{
    void mathOperation(int number);
}

public class LambdaExp {
    public static void main(String[] args) {

        Operation isOddNumber = (number) ->{
            int result;
            result=(number % 2);
            if(result != 0) {
                System.out.println("Tek sayıdır.");
            }
            else {
                System.out.println("Tek sayı değildir.");
            }
        };

        isOddNumber.mathOperation(9);
    }
}

```

Şekil 3:Blok Lambda

Tek sayıdır.

Şekil 4:Output