JAVA Lambda Expression

Furkan Ekici
`Turkcell Camp`
furkanekici1997@gmail.com

February 4, 2022

Contents

Abstract

Lambda Expession that comes with Java 8 can be defined as methods that do not have a name. Just like methods, they have a list of parameters, a body of code, and can return a variable. This report covers what Lambda Expressions are, their uses and examples.

1 What is Lambda Expression

Lambda Expression are structures that work in a similar way to methods. It has a parameter list and body. This build came with Java 8. It makes it possible to write more compact and less code. It saves a lot of code.
It implements functional interfaces. Functional interface is an interface that has only one method inside it. This interface reports the structure of the Lambda expression. [1] [2]

2 Usage of Lambda Expression

Lambda Expressions consist three parts.

- Parameter List: Lambda Expression takes parameters like a method. But parameters don't have variable type. Variable type of parameters are defined in functional interface.
- Arrow Operator: It separates parameter list and body section of Lambda Expression.
- Body Section: This is where the logic is done. There are expressions as in the body of a function. Parentheses are not needed if there is only one expression. Parentheses are used if there is more than one expression.

2.1 Lambda Expression Syntax

In left side, there is parameter list, in right side, there is body section and there is arrow operator between parameter list and body section.

`(parameter list) -> {body section}`

In the below, there is Lambda Expression implementation example:

```
@FunctionalInterface
interface Greet{
    void sayHi(String name);
}
public class LambdaExpressionExample{
    public static void main(String[] args){
        Greet greet = (name) -> {
            System.out.println("Hi " + name);
        };
        greet.sayHi("Furkan");
    }
}
```

[1]
OUTPUT :
Hi Furkan

In the first line of above code '@FunctionalInterface' part is optional. This program execute even if optional part is not written. But it can make code more readable.

As you can see, interface method is implemented by using Lambda Expression. After implementation, sayHi method can be used like a classic method.

2.2 Lambda Expression with Arraylist

By using Lambda Expression and ArrayList forech method, elements of ArrayList can be accessed iteratively. In this way, ArrayList can be managed easily and progremmers write less code.
For example :

```
public class LambdaExpressionWithArrayList{
    public static void main(String[] args){
        ArrayList<String> list = new ArrayList<>();
        list.add("Furkan");
        list.add("Mehmet");
        list.add("Eren");

        list.forEach((element) -> {
           System.out.println(element);
        });
    }
}
```

[2]
OUTPUT :
Furkan
Mehmet
Eren

References

[1] JavatPoint. *Java Lambda Expressions*. URL: https://www.javatpoint.com/java-lambda-expressions. (accessed: 04.02.2022).

[2] W3Schools. *Java Lambda Expressions*. URL: https://www.w3schools.com/java/java_lambda.asp. (accessed: 04.02.2022).