Author: ungcode
Email: ungcode076@gmail.com
Git: https://github.com/ungcode/java_space.git
Youtube Channel: https://www.youtube.com/channel/UCkIL5-6T1eefpG4aB-JVsTw?view_as=subscriber

Help us create more content. Hit the **like button** and **Subscribe** to the channel

Java 8 - Lambda Expressions

What is a Lambda Expression?

1. Lambda expression enables functional programming

Why should you use Lambda expressions ?

1. To provide the implementation of Functional interface
2. Less coding.

This is what you need to be aware of while defining lambda expressions

1. Syntax
   - `parameter` → *expression body*
2. Characteristics
   - **Type declaration** → parameter type are optional
   - **Parenthesis around parameter**
     a. No need to declare a single `parameter` in `parenthesis`.
     b. `Parentheses` are `required` for `multiple parameters`
   - `Curly braces` → ` curly braces ` are optional If the `body contain's a 'single statement`
   - Return keyword
     a. The compiler automatically returns the value if the body has a single expression
     b. `Curly braces` are required to indicate that expression returns a value.

Let's see Lambda expresions in practice:

```
package com.ung.java.lambda;
/**
*
*  Copyright 2020 by UNGCODE <ungcode076@gmail.com>
*
* This file is part of UNGCODE open source application.
```

```
*
* UNGCODE open source application is free software: you can redistribute
* it and/or modify it under the terms of the GNU General Public
* License as published by the Free Software Foundation, either
* version 3 of the License, or (at your option) any later version.
*
* UNGCODE open source application is distributed in the hope that it will
* be useful, but WITHOUT ANY WARRANTY; without even the implied warranty
* of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this project.  If not, see <http://www.gnu.org/licenses/>.
*
* @license GPL-3.0+ <http://spdx.org/licenses/GPL-3.0+>
*/
@FunctionalInterface
public interface Operation {

    int perform(int x,int y);
}

@FunctionalInterface
public interface Greeting {

    String perform(String message);
}

public class Lambda {


    public static int execute (int x,int y, Operation operation) {

        return operation.perform(x, y);
    }

   public static String execute (String message, Greeting greeting) {

        return greeting.perform(message);
    }

}

public class Execute {

public static void main(String[] args) {

        // With optional parameter type
        Operation addition = (x,y)->x+y;

        // With parameter type
```

```
        Operation subtraction = (int x,int y)->x-y;

        // With curly braces
        Operation multiplication = (x,y)->{return x+y;};

        // Without  curly braces and return statement
        Operation division = (int x,int y)->x/y;

        Greeting greeting = (message)-> {return message;};

        System.out.println("addition: "+ Lambda.execute(8, 2, addition));

        System.out.println("subtraction: "+ Lambda.execute(8, 2, subtraction));

        System.out.println("multiplication: "+ Lambda.execute(8, 2, multiplication));

        System.out.println("division: "+ Lambda.execute(8, 2, division));

        System.out.println("greeting: "+ Lambda.execute("Good morning",greeting));


    }
}
```

Author: ungcode
Email: ungcode076@gmail.com
Git: https://github.com/ungcode/java_space.git
Youtube Channel: https://www.youtube.com/channel/UCkIL5-6T1eefpG4aB-JVsTw?view_as=subscriber

Help us create more content. Hit the **like button** and **Subscribe** to the channel