Mercado Salinas Fhernando

- ***Máster Universitario en Sistemas Embebidos***
  - Mondragon Unibertsitatea, Basque Country, Spain

- ***Ingeniería en Sistemas Computacionales***
  - Tecnológico de Estudios Superiores de Jocotitlán

Programación Lógica y Funcional

v 0.0, Advanced Level

Functional Programming with Java

FP_Structured_01.java

```java
import java.util.List;

/* Structured Approach */

public class FP_Structured_01{

    public static void main(String[] args){

        List<Integer> numbers = List.of(12, 9, 13, 4, 6, 2, 4, 12, 15);

        FP_Structured_01.printAllNumbersInListStructured(numbers);
        System.out.println("");
        FP_Structured_01.printEvenNumbersInListStructured(numbers);
        System.out.println("");

        printAllNumbersInListStructured(List.of(12, 9, 13, 4, 6, 2, 4, 12, 15));
        System.out.println("");
        printEvenNumbersInListStructured(List.of(12, 9, 13, 4, 6, 2, 4, 12, 15));
        System.out.println("");
    }

    private static void printAllNumbersInListStructured(List<Integer> numbers) {
        // How to loop the numbers?
        for(int number : numbers){
            System.out.print(number + ", " );
        }
        System.out.println("");
    }

    private static void printEvenNumbersInListStructured(List<Integer> numbers) {
        // How to loop the numbers?
        for(int number : numbers){
            if(number % 2 == 0){
                System.out.print(number + ", " );
            }
        }
        System.out.println("");
    }
}

```

Structured Approach

Functional Programming with Java

FP_Structured_01.java

```java
import java.util.List;

/* Structured Approach */

public class FP_Structured_01{

    public static void main(String[] args){

        List<Integer> numbers = List.of(12

        FP_Structured_01.printAllNumbersIn
        System.out.println("");
        FP_Structured_01.printEvenNumbersI
        System.out.println("");

        printAllNumbersInListStructured(Li
        System.out.println("");
        printEvenNumbersInListStructured(L
        System.out.println("");
    }

    private static void printAllNumbersInL
        // How to loop the numbers?
        for(int number : numbers){
            System.out.print(number + ", "
        }
        System.out.println("");
    }

    private static void printEvenNumbersIn
        // How to loop the numbers?
        for(int number : numbers){
            if(number % 2 == 0){
                System.out.print(number + ", " );
            }
        }
        System.out.println("");
    }
}
```

_Ejemplo_006_ — fm5@Air-Fhm5-5 — .._Ejemplo_006_ — -zsh — 80×24

```
┌─[fm5 Air-Fhm5-5] - [~/Documents/_reposGit_gitHub/_Programacion_Logica_y_Funcio
nal_/_Ejemplo_006_] - [Wed Mar 25, 14:19]
└─[$] <> javac FP_Structured_01.java
┌─[fm5 Air-Fhm5-5] - [~/Documents/_reposGit_gitHub/_Programacion_Logica_y_Funcio
nal_/_Ejemplo_006_] - [Wed Mar 25, 14:19]
└─[$] <> java FP_Structured_01
12, 9, 13, 4, 6, 2, 4, 12, 15,

12, 4, 6, 2, 4, 12,

12, 9, 13, 4, 6, 2, 4, 12, 15,

12, 4, 6, 2, 4, 12,

┌─[fm5 Air-Fhm5-5] - [~/Documents/_reposGit_gitHub/_Programacion_Logica_y_Funcio
nal_/_Ejemplo_006_] - [Wed Mar 25, 14:20]
└─[$] <>
```

Structured Approach

Functional Programming with Java

FP_Functional_01.java

FP_Structured_01.java | FP_Functional_01.java

```java
import java.util.List;

/* Structured Functional */

public class FP_Functional_01{

    public static void main(String[] args) {
        List<Integer> numbers = List.of(12, 9, 13, 4, 6, 2, 4, 12, 15);

        System.out.println("Using to --> [System.out::print] by default");
        printAllNumbersInListFunctional_Two(numbers);
        System.out.println("\nprintAllNumbersInListFunctional: ");
        printAllNumbersInListFunctional(numbers);
        System.out.println("\nprintEvenNumbersInListFunctional: ");
        printEvenNumbersInListFunctional(numbers);
        System.out.println("\nprintSquaresOfEvenNumbersInListFunctional: ");
        printSquaresOfEvenNumbersInListFunctional(numbers);
        System.out.println("");
    }

    private static void print(int number){
        System.out.print(number + ", " );
    }

    private static boolean isEven(int number){
        return (number % 2 == 0);
    }

    private static void printAllNumbersInListFunctional_Two(List<Integer> numbers){
        // What to do?
        numbers.stream()                // ---> Convert to Stream
        .forEach(System.out::print);    // ---> Method Reference
        System.out.println("");
    }

    private static void printAllNumbersInListFunctional(List<Integer> numbers){
        // What to do?
        numbers.stream()                    // ---> Convert to Stream
        .forEach(FP_Functional_01::print);  // ---> Method Reference
        System.out.println("");
    }
```

Functional Approach

Functional Programming with Java

FP_Functional_01.java

FP_Structured_01.java × | FP_Functional_01.java ×

```java
20
21      private static void print(int number){
22          System.out.print(number + ", " );
23      }
24
25      private static boolean isEven(int number){
26          return (number % 2 == 0);
27      }
28
29      private static void printAllNumbersInListFunctional_Two(List<Integer> numbers){
30          // What to do?
31          numbers.stream()                // ---> Convert to Stream
32          .forEach(System.out::print);    // ---> Method Reference
33          System.out.println("");
34      }
35
36      private static void printAllNumbersInListFunctional(List<Integer> numbers){
37          // What to do?
38          numbers.stream()                    // ---> Convert to Stream
39          .forEach(FP_Functional_01::print);  // ---> Method Reference
40          System.out.println("");
41      }
42
43      // number -> number % 2 == 0
44      private static void printEvenNumbersInListFunctional(List<Integer> numbers){
45          // What to do?
46          numbers.stream()                                // ---> Convert to Stream
47                  .filter(FP_Functional_01::isEven)       // --->  Method Reference --> Filter = Only Allow Even Numbers
48                  .forEach(FP_Functional_01::print);      // --->  Method Reference
49          System.out.println("");
50      }
51
52      private static void printSquaresOfEvenNumbersInListFunctional(List<Integer> numbers){
53          numbers.stream()                                // ---> Convert to Stream
54                  .filter(number -> number % 2 == 0)      // ---> Lambda Expression
55                  .map(number -> number * number)         // ---> Lambda --> mapping = x -> x * x
56                  .forEach(FP_Functional_01::print);      // ---> Method Reference
57          System.out.println("");
58      }
59  }
60
```

Functional Approach

Functional Programming with Java

FP_Functional_01.java

FP_Structured_01.java × FP_Functional_01.java ×

```
20
21      private static void print(int number){
22          System.out.print(number + ", " );
23      }
24
25      private static boolean isEven(int
26          return (number % 2 == 0);
27      }
28
29      private static void printAllNumbe
30          // What to do?
31          numbers.stream()
32          .forEach(System.out::print);
33          System.out.println("");
34      }
35
36      private static void printAllNumbe
37          // What to do?
38          numbers.stream()
39          .forEach(FP_Functional_01::pr
40          System.out.println("");
41      }
42
43      // number -> number % 2 == 0
44      private static void printEvenNumb
45          // What to do?
46          numbers.stream()
47                  .filter(FP_Functional
48                  .forEach(FP_Functiona
49          System.out.println("");
50      }
51
52      private static void printSquaresOfEvenNumbersInListFunctional(List<Integer> numbers){
53          numbers.stream()                                    // ---> Convert to Stream
54                  .filter(number -> number % 2 == 0)          // ---> Lambda Expression
55                  .map(number -> number * number)             // ---> Lambda --> mapping = x -> x * x
56                  .forEach(FP_Functional_01::print);          // ---> Method Reference
57          System.out.println("");
58      }
59  }
60
```

_Ejemplo_006_ — fm5@Air-Fhm5-5 — .._Ejemplo_006_ — -zsh — 86×24

```
┌─[fm5 Air-Fhm5-5] - [~/Documents/_reposGit_gitHub/_Programacion_Logica_y_Funcional_/_
Ejemplo_006_] - [Wed Mar 25, 18:22]
└─[$] <> javac FP_Functional_01.java
┌─[fm5 Air-Fhm5-5] - [~/Documents/_reposGit_gitHub/_Programacion_Logica_y_Funcional_/_
Ejemplo_006_] - [Wed Mar 25, 18:22]
└─[$] <> java FP_Functional_01
Using to --> [System.out::print] by default
1291346241215

printAllNumbersInListFunctional:
12, 9, 13, 4, 6, 2, 4, 12, 15,

printEvenNumbersInListFunctional:
12, 4, 6, 2, 4, 12,

printSquaresOfEvenNumbersInListFunctional:
144, 16, 36, 4, 16, 144,

┌─[fm5 Air-Fhm5-5] - [~/Documents/_reposGit_gitHub/_Programacion_Logica_y_Funcional_/_
Ejemplo_006_] - [Wed Mar 25, 18:22]
└─[$] <>
```

Functional Approach