

INF3522

Développement d'Applications N-tier



Dr El Hadji Bassirou TOURE

Département Informatique – DI

UFR Sciences et Techniques – UFR ST

2022 - 2023

Dr. E_H_B. TOURE / Inf3522 / UASZ

Programmation structurelle

■ Exemple

```
2
3 import java.util.List;
4
5 public class FP01Structured {
6
7     public static void main(String[] args) {
8
9         printAllNumbersInListStructured(List.of(12,9,13,4,6,2,4,12,15));
10
11     }
```

```
12
13     private static void printAllNumbersInListStructured(List<Integer> numbers) {
14         //How to loop the numbers?
15         for(int number:numbers) {
16             System.out.println(number);
17         }
18
19     }
20
21 }
```

Programmation structurelle

■ Exemple

```
2
3 import java.util.List;
4
5 public class FP01Structured {
6
7     public static void main(String[] args) {
8
9         printAllNumbersInListStructured(List.of(12,9,13,4,6,2,4,12,15));
10
11     }
12
13
14
15
16
17     private static void printAllNumbersInListFunctional(List<Integer> numbers) {
18         //What to do?
19         numbers.stream()
20             .forEach(System.out::println); //Method Reference
21     }
22
23 }
```

Programmation structurelle

■ Exemple

```
21
22 private static void printEvenNumbersInListStructured(List<Integer> numbers) {
23     // How to loop the numbers?
24     for (int number : numbers) {
25         if (number % 2 == 0) {
26             System.out.println(number);
27         }
28     }
29 }
```

Writable

Smart Insert

10 : 81

Udemy

Programmation structurelle

■ Exemple

```
16
17 private static boolean isEven(int number) {
18     return number%2 == 0;
19 }
20
```

```
20
27 private static void printEvenNumbersInListFunctional(List<Integer> numbers) {
28     //What to do?
29     numbers.stream()
30         //Filter - Only Allow Even Numbers
31         .filter(FP01Functional::isEven)
32         .forEach(System.out::println); //Method Reference
33 }
34
35 }
36
```

Writable

Smart Insert

31 : 42

Udemy

Programmation structurelle

■ Exemple

```
16
17 private static boolean isEven(int number) {
18     return number%2 == 0;
19 }
20
```

```
29 // number -> number%2 == 0
30 private static void printEvenNumbersInListFunctional(List<Integer> numbers) {
31     //What to do?
32
33     numbers.stream()
34         .filter(number -> number%2 == 0) //Lambda Expression
35         .forEach(System.out::println); //Method Reference
36
37     //filter(FP01Functional::isEven)//Filter - Only Allow Even Numbers
38
39 }
```

Programmation structurelle

■ Exemple

Exercise 1

Print Only Odd Numbers from the List

Exercise 2

Print All Courses individually

```
List<String> courses = List.of("Spring", "Spring Boot", "API", "Microservices", "AWS", "PCF", "Azure", "Docker", "Kubernetes")
```

Exercise 3

Print Courses Containing the word "Spring"

Exercise 4

Print Courses Whose Name has atleast 4 letters

Programmation structurelle

■ Exemple

```
private static void printSquaresOfEvenNumbersInListFunctional(List<Integer> numbers) {  
    numbers.stream() // Convert to Stream  
        .filter(number -> number % 2 == 0) // Lambda Expression  
        //mapping - x -> x * x  
        .map(number -> number * number)  
        .forEach(System.out::println); // Method Reference  
  
    // .filter(FP01Functional::isEven) // Filter - Only Allow Even Numbers  
}
```


Programmation structurelle

■ Exemple

```
1 import java.util.List;
2 import java.util.Optional;
3 import java.util.function.Predicate;
4
5 public class PlayingWithOptional {
6     public static void main(String[] args) {
7         List<String> fruits = List.of("apple", "banana", "mango");
8
9         Predicate<? super String> predicate = fruit -> fruit.startsWith("b");
10        Optional<String> optional = fruits.stream().filter(predicate).findFirst();
11        System.out.println(optional);
12        System.out.println(optional.isEmpty());
13        System.out.println(optional.isPresent());
14        System.out.println(optional.get());
15    }
16 }
17
18
```

Programmation fonctionnelle



En résumé

