TD3

Exercise 1

1)

```
import java.util.Collections;
import java.util.List;
import java.util.function.BiFunction;

public class TD3Exo1 {

    public static void main(String[] args)
    {
        List<String> list = List.of("hello", "world", "hello", "lambda",
"hello");

        System.out.println(count(list, "hello"));
    }

    private static long count(List<String> list, String word) {
        BiFunction<List, String, Long> biFunction = (li, s) ->
Long.parseLong(String.valueOf(Collections.frequency(li, s)));
        return biFunction.apply(list, word);
     }
```

2)

```
import java.util.Collections;
import java.util.List;
import java.util.function.BiFunction;

public class TD3Exo1 {

    public static void main(String[] args)
    {
        List<String> list = List.of("hello", "world", "hello", "lambda",
"hello");

        System.out.println(count(list, "hello"));
        System.out.println(count2(list, "hello"));
    }

    private static long count(List<String> list, String word) {
        BiFunction<List, String, Long> biFunction = (li, s) ->
Long.parseLong(String.valueOf(Collections.frequency(li, s)));
        return biFunction.apply(list, word);
     }

    private static long count2(List<String> list, String word) {
        BiFunction<List, String, Long> biFunction = (li, s) ->
li.stream().filter(x -> x.equals(s)).count();
        return biFunction.apply(list, word);
    }
}
```

Exercise 2

1)

```
import java.util.ArrayList;
import java.util.List;

public class TD3Exo2 {

    public static void main(String[] args)
    {
        List<String> list = List.of("hello", "world", "hello", "lambda",
"hello");

        System.out.println(upperCase(list));
    }

    private static List<String> upperCase(List<String> list){
        List<String> uppercases = new ArrayList<String>(list);

        uppercases.replaceAll(s -> s.toUpperCase());
        return uppercases;
    }
}
```

2)

```
import java.util.ArrayList;
import java.util.List;

public class TD3Exo2 {

    public static void main(String[] args)
    {
        List<String> list = List.of("hello", "world", "hello", "lambda",
"hello");

        System.out.println(upperCase(list));
        System.out.println(upperCase2(list));
    }

    private static List<String> upperCase(List<String> list){
        List<String> uppercases = new ArrayList<String>(list);

        uppercases.replaceAll(s -> s.toUpperCase());
        return uppercases;
    }

    private static List<String> upperCase2(List<String> list){
        List<String> uppercases = new ArrayList<String>();

        list.stream().map(s -> s.toUpperCase()).forEach(s ->
uppercases.add(s));
        return uppercases;
    }
}
```

3)

```java
import java.util.ArrayList;
import java.util.List;

public class TD3Exo2 {

    public static void main(String[] args)
    {
        List<String> list = List.of("hello", "world", "hello", "lambda",
"hello");

        System.out.println(upperCase(list));
        System.out.println(upperCase2(list));
        System.out.println(upperCase3(list));
    }

    private static List<String> upperCase(List<String> list){
        List<String> uppercases = new ArrayList<String>(list);

        uppercases.replaceAll(s -> s.toUpperCase());
        return uppercases;
    }

    private static List<String> upperCase2(List<String> list){
        List<String> uppercases = new ArrayList<String>();

        list.stream().map(s -> s.toUpperCase()).forEach(s ->
uppercases.add(s));
        return uppercases;
    }

    private static List<String> upperCase3(List<String> list){
        List<String> uppercases = new ArrayList<String>(list);

        uppercases.replaceAll(String::toUpperCase);
        return uppercases;
    }
}
```

4)

```java
import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

public class TD3Exo2 {

    public static void main(String[] args)
    {
        List<String> list = List.of("hello", "world", "hello", "lambda",
"hello");

        System.out.println(upperCase(list));
        System.out.println(upperCase2(list));
        System.out.println(upperCase3(list));
        System.out.println(upperCase4(list));
    }

    private static List<String> upperCase(List<String> list){
        List<String> uppercases = new ArrayList<String>(list);

        uppercases.replaceAll(s -> s.toUpperCase());
        return uppercases;
    }

    private static List<String> upperCase2(List<String> list){
        List<String> uppercases = new ArrayList<String>();

        list.stream().map(s -> s.toUpperCase()).forEach(s ->
uppercases.add(s));
        return uppercases;
    }

    private static List<String> upperCase3(List<String> list){
        List<String> uppercases = new ArrayList<String>(list);

        uppercases.replaceAll(String::toUpperCase);
        return uppercases;
    }

    private static List<String> upperCase4(List<String> list){
        List<String> uppercases = new ArrayList<String>();

        uppercases = list.stream().map(s ->
s.toUpperCase()).collect(Collectors.toList());
        return uppercases;
    }
}
```

Exercise 3

1) Car celle-ci va nous permettre de récupérer directement les bits de la list

2)

```
import java.util.List;
import java.util.function.BiFunction;

public class TD3Exo3 {

    public static void main(String[] args)
    {
        List<String> list = List.of("hello", "world", "hello", "lambda",
"hello");

        System.out.println(count3(list, "hello"));
    }

    private static long count3(List<String> list, String word){

        BiFunction<List, String, Long> biFunction = (li, w) ->
list.stream().filter(x -> x.equals(word)).mapToLong(x -> 1).sum();
        return biFunction.apply(list, word);
    }

}
```

Exercice 4

1) La variable locale list2 contient 1 million d'entier compris entre 1 et 100 réparties de façon aléatoire.

2)

```
import java.util.Collections;
import java.util.List;
import java.util.Random;
import java.util.function.BiFunction;
import java.util.function.Function;
import java.util.function.LongFunction;
import java.util.function.Supplier;
import java.util.stream.Collectors;

public class TD3Exo4 {

    public static void main(String[] args)
    {
        List<String> list2 =
                new Random(0)
                        .ints(1_000_000, 0, 100)
                        .mapToObj(Integer::toString)
                        .collect(Collectors.toList());

        //System.out.println(count(list2, "33"));
        printAndTime(() -> count(list2, "33"));
        printAndTime(() -> count2(list2, "33"));
        printAndTime(() -> count3(list2, "33"));
    }

    private static void printAndTime(Supplier<Long> result){
        long start = System.nanoTime();
        long end = System.nanoTime();
        System.out.println("result " + result.get());
        System.out.println(" elapsed time " + (end - start));
    }

    private static long count(List<String> list, String word) {
        BiFunction<List, String, Long> biFunction = (li, s) ->
Long.parseLong(String.valueOf(Collections.frequency(li, s)));
        return biFunction.apply(list, word);
    }

    private static long count2(List<String> list, String word) {
        BiFunction<List, String, Long> biFunction = (li, s) ->
li.stream().filter(x -> x.equals(s)).count();
        return biFunction.apply(list, word);
    }

    private static long count3(List<String> list, String word){

        BiFunction<List, String, Long> biFunction = (li, w) ->
list.stream().filter(x -> x.equals(word)).mapToLong(x -> 1).sum();
        return biFunction.apply(list, word);
    }
}
```