

## Partie 1

### Q1 :

Créer un Classe Q1 et dans une méthode main :  
Créer deux optional de votre choix de deux façons différentes.  
Explorer l'API optional et utiliser au moins 5 méthodes différentes dans votre main.

### Q2 :

Créer un package q2  
Créer une functional interface nommée LengthCalculator avec une méthode calculate  
Créer une implémentation de cette interface nommée LastNameLengthCalculator  
Utiliser cette implémentation dans une classe main avec votre nom de famille.  
Créer une implémentation de LengthCalculator nommée FirstnameLengthCalculator dans votre Main  
Créer une lambda nommée CityLengthCalculator et utilisez là.  
Écrire cette même lambda avec 3 syntaxes différentes

Expliquer en commentaire avec **vos propres mots** et recherches ce qu'est une lambda et une functional interface

### Q3 :

Créer un Classe Q3:  
Créer une méthode qui récupère le deuxième élément d'une liste de string et la tester dans le main  
Créer une méthode qui récupère le deuxième élément d'une liste de Integer et la tester dans le main  
Créer une méthode générique qui récupère le deuxième élément d'une liste de type inconnu

Expliquer en commentaire avec **vos propres mots** et recherches ce qu'est que la généricité en java

### Q4 :

Créer un package Q4 :  
Créer une functional interface nommée StartWithGreg avec une méthode checkName qui renvoie un boolean en fonction de la string en paramètre  
Dans le main, créer et utiliser une lambda associée à cette interface  
Dans le main, créer et utiliser un predicate (java.util.function.Predicate) avec le même comportement.

Créer une functional interface nommée GenerateRandomNumber avec une méthode create qui renvoie un nombre entre 0 et 100 et qui n'a pas de paramètre

Dans le main, créer et utiliser une lambda associée à cette interface

Dans le main, créer et utiliser un supplier (java.util.function. Supplier) avec le même comportement.

Créer et utiliser une fonction initializeIntegerList avec deux paramètres, la taille du tableau ainsi que qu'un supplier de integer.

Créer et utiliser une fonction initializeList avec deux paramètres, la taille du tableau ainsi que qu'un supplier de type generic.

Utiliser cette dernière méthode pour générer une liste avec des nombres aléatoires, et une avec des lettres aléatoires

Créer et utiliser un consumer qui affichera un message de bienvenue en fonction du pseudo passé en paramètre.

Créer et utiliser une java.util.function.Function qui comme la Q2 calcule la taille d'une chaîne en paramètre

Créer et utiliser une java.util.function.BiFunction avec mettre le paramètre 1 à la puissance « paramètre 2 »

Expliquer en commentaire avec **vos propres mots** et recherches ce que sont une Predicate, Supplier, consumer et fonction. Quel lien y a-t-il avec la programmation fonctionnelle ?

Q5 :

Create package Q5

In this exercise you need to find the most suitable stream functionalities

Create a list of string with duplicate elements

count the number of distinct elements

check if one of the strings of the list contains a letter

Get a list of all the string starting with a letter of your choice (without duplicate)

Get a list of integers of the ascii code of the first letter of each non duplicated string (by order asc)

(Hard: so optional, only if you have a lot of time) Return a map with as key a letter and as value the occurrence in all the string of your list (with the duplicate)

Créer une class student avec deux attributs name et grade.

Créer une liste de plusieurs étudiants et trier les étudiants par notes utilisant une lambda pour comparer.

Q6

Create a Q6 package

Create a generic method named buildAnything with 3 parameters: a generic supplier : a generic consumer and a consumer of exception type

The method will get the supplier value then send it to the consumer.

If an error occurs in the catch the exception consumer can consume the raised exception.

You can try this method with a supplier supplying a number, a consumer who print, and an exception consumer who print the exception message.