

Prénom _____

NOM _____

20

Interrogation de Java

(Date)

Remarques :

- ↪ vous travaillerez dans un répertoire interro¹,
- ↪ vous pouvez utiliser **toutes les notes** que vous voulez ainsi que les programmes qui se trouvent dans votre répertoire,
- ↪ l'interro se fait sur linux1,
- ↪ vous disposez de 2 heures

L E PROBLÈME. Nous allons faire des "statistiques"² sur les *nombre de Fibonacci*. Nous allons chercher la fréquence de nombres pairs, et impairs dans une suite de tels nombres.

Rappel Les nombres de Fibonacci se construisent comme suit ;

$$F_0 = 1$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2} \text{ où } n \geq 2$$

1. Ecrire une fonction entête, affichant un cartouche tel que celui-ci

(2 points)

=====

Nombres de Fibonacci

Auteur : Prénom NOM

Groupe : Prof

=====

2. Ecrire, au choix, une fonction estPair ou estImpair retournant la valeur booléenne adéquate en fonction du nombre entier passé en paramètre.

(3 points)

```

/**
 * estPair
 *
 * @param nombre: la valeur a tester
 * @return vrai si nombre est pair, faux sinon
 */
public static boolean estPair (long nombre) {...}
```

3. Ecrire une fonction getSuiteFibonacci prenant en entrée le nombre de nombres de Fibonacci³ à générer et retournant un tableau contenant cette suite de nombres.

(4 points)

¹Remarquez l'absence de majuscule au nom de répertoire

²Ne soyez pas effrayés, *statistique* est un bien grand mot.

³On entend bien par là, l'indice jusqu'auquel il faut générer tous les nombres de Fibonacci précédents

```

/**
 * getSuiteFibonacci
 *
 * @param nombre: nombre de nombres a generer
 * @return un tableau de nombres de Fibonacci
 */
public static long [] getSuiteFibonacci (int nombre) {...}

```

4. Ecrire une fonction `affiche` qui affiche le tableau passé en paramètre. (2 points)

```

/**
 * affiche
 *
 * @param t: un tableau de long
 */
public static void affiche (long [] t) {...}

```

5. Ecrire une fonction `statistiques` effectuant les diverses statistiques sur la suite de nombres. Ces statistiques sont : (3 points)

- ↪ calculer et afficher le nombre de nombres pairs,
- ↪ calculer et afficher le nombre de nombres impairs,

```

/**
 * statistiques
 *
 * @param t, tableau de long
 */
public static void statistiques ( long [] t) {...}

```

6. Ecrire la fonction `main` afin qu'elle mette en oeuvre les différentes fonctions. Le programme : (3 points)

- ↪ affiche un *entête*,
- ↪ crée un tableau de `NOMBRENOMBRES` entiers remplis de la suite de nombre de Fibonacci,
- ↪ affiche ce tableau,
- ↪ affiche les statistiques sur la suite.

7. Modifier la fonction `main` afin qu'elle reçoive le nombre de nombre de Fibonacci à générer en paramètre au programme. Si le nombre de paramètre est différent de 1, le programme se termine et un message est affiché ; (3 points)

Usage : `java Fibonacci <nombre>`