# Algo et Structure de donnees

Nouakchott, LE 14/12/2024

**Exo 1 (Code en JAVA) :**

```
package Helloworld;

public class HelloWorld {


   public static long fib(int n) {
      if (n <= 1) {
         return 1;
      }
      return fib(n - 1) + fib(n - 2);
   }

   public static void main(String[] args) {

      int n = 5;
      long result = fib(n);
      System.out.println("La " + n + " ème valeur de Fibonacci est : " + result);
   }
}
```

---

## 1. Structure du code

- Le programme est contenu dans un package nommé `Helloworld`.
- Il contient une classe publique nommée `HelloWorld` qui inclut :
  - Une méthode récursive `fib` pour calculer les valeurs de la suite de Fibonacci.
  - Une méthode `main`, qui est le point d'entrée du programme, pour exécuter la logique.

---

## 2. Méthode `fib(int n)`

La méthode `fib` calcule la **n^e valeur de Fibonacci** de manière récursive :

- **Entrée** : Un entier `n`.
- **Sortie** : Un entier long (`long`) représentant la valeur de Fibonacci correspondant à `n`.

*Fonctionnement logique :*

- Si `n <= 1`, la fonction retourne `1` directement (valeur de base).
- Sinon, elle fait appel à elle-même deux fois pour calculer :
  - `fib(n - 1)` : La (n-1)^e valeur de Fibonacci.
  - `fib(n - 2)` : La (n-2)^e valeur de Fibonacci.
- La somme de ces deux valeurs est retournée comme résultat.

## Explication logic:

Examinons comment le programme fonctionne pour n=5n = 5n=5. Voici une analyse étape par étape de l'exécution avec des appels récursifs détaillés.

```
fib(5)
├── fib(4)
│   ├── fib(3)
│   │   ├── fib(2)
│   │   │   ├── fib(1) → 1
│   │   │   └── fib(0) → 1
│   │   └── fib(1) → 1
│   └── fib(2)
│       ├── fib(1) → 1
│       └── fib(0) → 1
├── fib(3)
    ├── fib(2)
    │   ├── fib(1) → 1
    │   └── fib(0) → 1
    └── fib(1) → 1
```

---

Resultat fib(5) : La 5 ème valeur de Fibonacci est : 8

## **Exo 2 (Code en JAVA):**

```
package Helloworld;

public class HelloWorld {


   public static long fib(int n) {
      if (n <= 1) {
         return 1;
      }

      long f0 = 1;
      long f1 = 1;
      long fn = 1;


     for (int i = 2; i <= n; i++) {
         fn = f0 + f1;
         f0 = f1;
         f1 = fn;
      }

      return fn;
   }

   public static void main(String[] args) {
      int n = 6;
      long result = fib(n);
      System.out.println("La " + n + " ème valeur de Fibonacci est : " + result);
   }
```

}**Problèmes identifiés**

1. **Déclaration des variables** :
   - Les variables dans la méthode `fib` ne sont pas correctement initialisées ou déclarées (par exemple, `long int i` n'est pas valide en Java).
2. **Boucle `for`** :
   - La condition `i=n` est incorrecte. Une boucle `for` doit utiliser une comparaison logique (par exemple, `i <= n`).
3. **Retour dans la boucle** :
   - Vous utilisez `return fn` à l'intérieur de la boucle. Cela arrête l'exécution dès la première itération.
4. **Structure et logique** :
   - Vous mélangez des concepts de récursivité et d'itération. Le calcul de Fibonacci doit être soit complètement itératif, soit récursif.

## Explications

1. **Structure corrigée** :
   - La méthode `fib` est entièrement itérative.
   - Les valeurs f0f0f0, f1f1f1, et fnfnfn sont utilisées pour calculer la suite Fibonacci.
2. **Boucle `for`** :
   - La boucle commence à i=2i = 2i=2, car f0f0f0 et f1f1f1 sont déjà connus.
   - Elle s'arrête à i=ni = ni=n.
3. **Retour du résultat** :
   - Le résultat final fnfnfn est retourné après l'exécution de la boucle.

## Étape 1 : Appel de la méthode

La méthode `fib(6)` est appelée dans le `main`.

---

## Étape 2 : Initialisation

Les variables sont initialisées avant la boucle :

- f0=1f0 = 1f0=1 (la première valeur de Fibonacci)
- f1=1f1 = 1f1=1 (la deuxième valeur de Fibonacci)
- fn=1fn = 1fn=1 (la valeur courante de Fibonacci, initialisée à 1)

---

## Étape 3 : Boucle `for`

La boucle commence à i=2i = 2i=2 et continue jusqu'à i=6i = 6i=6. Voici le déroulement pour chaque itération :

*Itération 1 (i=2):*

- fn=f0+f1=1+1=2fn = f0 + f1 = 1 + 1 = 2fn=f0+f1=1+1=2
- Mise à jour :
  - f0=f1=1f0 = f1 = 1f0=f1=1
  - f1=fn=2f1 = fn = 2f1=fn=2

*Itération 2 (i=3):*

- fn=f0+f1=1+2=3fn = f0 + f1 = 1 + 2 = 3fn=f0+f1=1+2=3
- Mise à jour :
  - f0=f1=2f0 = f1 = 2f0=f1=2
  - f1=fn=3f1 = fn = 3f1=fn=3

*Itération 3 (i=4):*

- fn=f0+f1=2+3=5fn = f0 + f1 = 2 + 3 = 5fn=f0+f1=2+3=5

- Mise à jour :
  - f0=f1=3f0 = f1 = 3f0=f1=3
  - f1=fn=5f1 = fn = 5f1=fn=5

*Itération 4 (i=5):*

- fn=f0+f1=3+5=8fn = f0 + f1 = 3 + 5 = 8fn=f0+f1=3+5=8
- Mise à jour :
  - f0=f1=5f0 = f1 = 5f0=f1=5
  - f1=fn=8f1 = fn = 8f1=fn=8

*Itération 5 (i=6):*

- fn=f0+f1=5+8=13fn = f0 + f1 = 5 + 8 = 13fn=f0+f1=5+8=13
- Mise à jour :
  - f0=f1=8f0 = f1 = 8f0=f1=8
  - f1=fn=13f1 = fn = 13f1=fn=13

---

## Étape 4 : Résultat

À la fin de la boucle, fn=13fn = 13fn=13, qui correspond à la 6e valeur de Fibonacci.

Resultat : La 6 ème valeur de Fibonacci est : 13

---

**Cordialement.**

**Merci, j'espère que cela vous a été utile et que vous en avez pleinement profité.**