

Application d'une méthode à tous les éléments d'un conteneur

- On encapsule une méthode dans une interface :

```
interface Function {  
    abstract public int applyIt(int n);  
}
```

- On encapsule la fonction map dans une autre interface :

```
interface Map {  
    abstract public void map(Function f);  
}
```

- Une classe qui veut réaliser un map implémente cette interface :

```
import java.util.Arrays;  
  
public class Tableau implements Map {  
    int[] a;  
  
    Tableau(int n) {  
        a = new int[n];  
        for (int i = 0; i < n; i++)  
            a[i] = i + 1;  
    }  
  
    public void map(Function f) {  
        for (int i = 0; i < a.length; i++)  
            a[i] = f.applyIt(a[i]);  
    }  
  
    @Override  
    public String toString() {  
        return "Tableau [a=" + Arrays.toString(a) + "];"  
    }  
}
```

- Un exemple de fonction :

```
public class Carre implements Function {  
    public int applyIt(int n) {  
        return n*n;  
    }  
}
```

• **Un exemple d'utilisation:**

```
public class TestMap {  
    public static void main(String args[]) {  
        Map t = new Tableau(10);  
  
        Function square = new Carre();  
        System.out.println(t);  
  
        t.map(square);  
        System.out.println(t);  
    }  
}
```

Exécution

Tableau [a=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]]

Tableau [a=[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]]