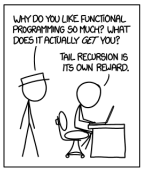
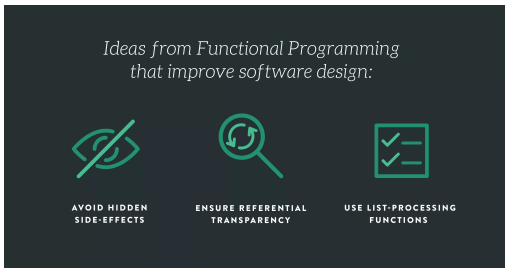
La programmation procédurale en Java



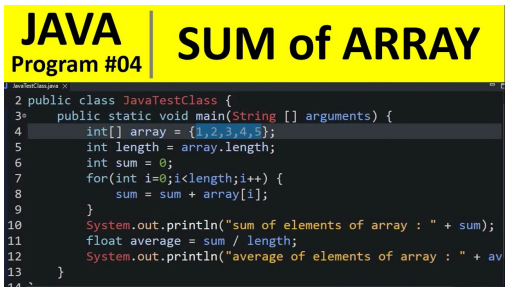
La programmation procédurale est un paradigme de programmation dérivé de la programmation structurée, basé sur le concept d'appel de procédures, également connu sous le nom de routines ou fonctions. En Java, bien que ce soit un langage orienté objet, il est possible d'écrire du code en suivant un style procédural. Voici comment cela peut être réalisé :

Concepts de base de la programmation procédurale



1. **Procédures/Fonctions** : Blocs de code réutilisables qui effectuent une tâche spécifique.
2. **Variables globales et locales** : Variables accessibles dans tout le programme (globales) ou dans une procédure spécifique (locales).
3. **Contrôle de flux** : Utilisation de structures conditionnelles et de boucles pour contrôler l'exécution du programme.

Exemple de programmation procédurale en Java



Pour illustrer la programmation procédurale en Java, considérons un programme simple qui calcule la somme des éléments d'un tableau.

Définir les procédures (fonctions)

En Java, nous définissons les fonctions au sein d'une classe (car tout doit être encapsulé dans une classe), mais nous évitons d'utiliser des objets et des méthodes d'instance pour suivre le style procédural.

```
public class ProceduralExample {

    // Fonction pour initialiser un tableau avec des valeurs
    public static int[] initializeArray(int size) {
        int[] array = new int[size];
        for (int i = 0; i < size; i++) {
            array[i] = i + 1; // Remplit le tableau avec les valeurs 1, 2, ..., size
        }
        return array;
    }

    // Fonction pour calculer la somme des éléments d'un tableau
    public static int sumArray(int[] array) {
        int sum = 0;
        for (int value : array) {
            sum += value;
        }
        return sum;
    }

    // Fonction principale qui sert de point d'entrée au programme
```

```
    public static void main(String[] args) {
        int size = 10;
        int[] myArray = initializeArray(size);
        int sum = sumArray(myArray);
        System.out.println("The sum of the array elements is: " + sum);
    }
}
```

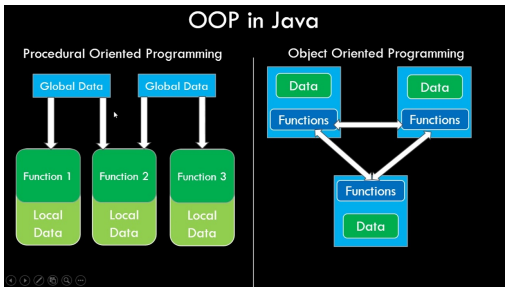
Explication du programme

1. **Classe principale** : `ProceduralExample`
   - En Java, nous devons définir au moins une classe pour contenir notre méthode `main`, même si nous suivons un style procédural.
2. **Fonction `initializeArray`** :
   - Initialise un tableau d'entiers avec une taille donnée.
   - Remplit le tableau avec les valeurs de 1 à `size`.
   - Retourne le tableau initialisé.
3. **Fonction `sumArray`** :
   - Calcule la somme des éléments d'un tableau donné.
   - Utilise une boucle `for-each` pour itérer sur les éléments du tableau et les ajouter à la somme.
4. **Méthode `main`** :
   - Point d'entrée du programme.
   - Appelle les fonctions `initializeArray` et `sumArray` pour créer un tableau et calculer sa somme.
   - Affiche le résultat.

Aspects procéduraux

- **Pas d'objets** : Le programme n'utilise pas d'instances d'objets pour accomplir ses tâches, mais appelle directement des fonctions statiques.
- **Contrôle de flux** : Utilisation de boucles et de structures de contrôle pour gérer l'exécution du programme.
- **Encapsulation minimale** : Les données (tableau) sont passées de fonction en fonction plutôt que d'être encapsulées dans des objets.

Comparaison avec la programmation orientée objet



En programmation orientée objet, nous encapsulerions les données et les comportements dans des classes et des objets. Par exemple :

```
public class ArrayProcessor {
    private int[] array;

    public ArrayProcessor(int size) {
        array = new int[size];
        for (int i = 0; i < size; i++) {
            array[i] = i + 1;
        }
    }

    public int sumArray() {
        int sum = 0;
        for (int value : array) {
            sum += value;
        }
        return sum;
    }

    public static void main(String[] args) {
        ArrayProcessor processor = new ArrayProcessor(10);
        int sum = processor.sumArray();
        System.out.println("The sum of the array elements is: " + sum);
    }
}
```

Ici, nous voyons l'utilisation d'une classe `ArrayProcessor` pour encapsuler les données et les méthodes, conformément aux principes de la programma-

tion orientée objet.

Conclusion

Bien que Java soit principalement conçu pour la programmation orientée objet, il est possible d'écrire du code de manière procédurale en utilisant des méthodes statiques et en évitant l'utilisation d'instances d'objets. Cela peut être utile pour les développeurs venant de langages procéduraux ou pour des programmes simples où l'utilisation complète de la POO n'est pas nécessaire.