

4. TP 2 : Tableaux, Chaînes et Fonctions en Java

Objectifs

- Comprendre la déclaration et l'utilisation des tableaux en Java.
- Manipuler les chaînes de caractères avec la classe String.
- Créer et appeler des méthodes (fonctions) en Java.
- Utiliser les bibliothèques standard de Java.

Partie 1 : Questions guidées avec réponses

1.1 Tableaux (Arrays)

Question 1 :

Comment déclarer et initialiser un tableau d'entiers en Java ?

Réponse :

```
1 public class TableauEntiers {
2     public static void main(String[] args) {
3         // Déclaration et initialisation d'un tableau d'
entiers
4         int[] nombres = {1, 2, 3, 4, 5};
5
6         // Affichage des éléments du tableau
7         for (int i = 0; i < nombres.length; i++) {
8             System.out.println("nombres[" + i + "] = " +
nombres[i]);
9         }
10    }
```

```
11 }
```

Question 2 :

Comment créer un tableau de chaînes de caractères et accéder à ses éléments ?

Réponse :

```
1 public class TableauChaines {
2     public static void main(String[] args) {
3         // Déclaration et initialisation d'un tableau de chaî
4         nes
5         String[] fruits = {"Pomme", "Banane", "Cerise"};
6
7         // Affichage des éléments du tableau
8         for (int i = 0; i < fruits.length; i++) {
9             System.out.println("fruits[" + i + "] = " + fruits
10             [i]);
11         }
12     }
13 }
```

Question 3 :

Comment déclarer et utiliser un tableau multidimensionnel en Java ?

Réponse :

```
1 public class TableauMulti {
2     public static void main(String[] args) {
3         // Déclaration d'un tableau 2D (matrice)
4         int[][] matrice = {
5             {1, 2, 3},
6             {4, 5, 6},
7             {7, 8, 9}
8         };
9
10        // Affichage des éléments de la matrice
11        for (int i = 0; i < matrice.length; i++) {
12            for (int j = 0; j < matrice[i].length; j++) {
13                System.out.print(matrice[i][j] + " ");
14            }
15            System.out.println();
16        }
17    }
18 }
```

Question 4 :

Quelles sont les principales différences entre les tableaux en Java et en C ?

Réponse :

— En Java, les tableaux sont des objets et leur taille est fixe une fois créés.

- Il n'y a pas de pointeurs explicites en Java, ce qui réduit les erreurs liées à la mémoire.
- Java gère automatiquement la mémoire (garbage collector), donc pas besoin de libérer manuellement la mémoire.
- Les tableaux en Java connaissent leur propre longueur via l'attribut `length`.

1.2 Chaînes de caractères

Question 5 :

Comment créer une chaîne de caractères en Java et pourquoi est-elle immuable ?

Réponse :

```
1 public class ManipulationChaines {
2     public static void main(String[] args) {
3         // Création d'une chaîne de caractères
4         String salutations = "Bonjour le monde";
5
6         // Les chaînes en Java sont immuables, ce qui signifie
7         // que leur contenu ne peut pas être modifié après leur création.
8         // Toute opération qui semble modifier une chaîne en
9         // crée en fait une nouvelle.
10    }
```

Question 6 :

Comment utiliser les méthodes courantes de la classe `String` comme `length()`, `charAt()`, `substring()`, `equals()` ?

Réponse :

```
1 public class MethodesString {
2     public static void main(String[] args) {
3         String texte = "Programmation";
4
5         // Obtenir la longueur de la chaîne
6         int longueur = texte.length(); // 13
7
8         // Obtenir le caractère à une position spécifique
9         char caractere = texte.charAt(5); // 'a'
10
11        // Extraire une sous-chaîne
12        String sousChaine = texte.substring(0, 6); // "Program
13
14        // Comparer deux chaînes
15        String mot = "Programmation";
16        boolean estEgal = texte.equals(mot); // true
17    }
18 }
```

Question 7 :

Comment convertir une chaîne de caractères en majuscules ou en minuscules ?

Réponse :

```

1 public class ConversionCasse {
2     public static void main(String[] args) {
3         String texte = "Bonjour";
4
5         // Convertir en majuscules
6         String majuscules = texte.toUpperCase(); // "BONJOUR"
7
8         // Convertir en minuscules
9         String minuscules = texte.toLowerCase(); // "bonjour"
10    }
11 }

```

1.3 Fonctions (Méthodes)**Question 8 :**

Comment définir et appeler une méthode en Java qui additionne deux nombres entiers ?

Réponse :

```

1 public class Calculatrice {
2     // Définition de la méthode
3     public static int additionner(int a, int b) {
4         return a + b;
5     }
6
7     public static void main(String[] args) {
8         // Appel de la méthode
9         int resultat = additionner(5, 7);
10        System.out.println("Le résultat est : " + resultat);
11    }
12 }

```

Question 9 :

Quelle est la différence entre le passage de paramètres par valeur et par référence en Java ?

Réponse :

- En Java, les types primitifs (int, double, boolean, etc.) sont passés par valeur. La méthode reçoit une copie de la valeur.
- Les objets sont passés par référence de valeur, c'est-à-dire que la méthode reçoit une copie de la référence à l'objet. Les modifications sur l'objet affectent l'original.

Question 10 :

Comment utiliser les méthodes surchargées en Java ?

Réponse :

```
1 public class MathUtil {
2     // Méthode pour additionner deux entiers
3     public static int additionner(int a, int b) {
4         return a + b;
5     }
6
7     // Méthode surchargée pour additionner trois entiers
8     public static int additionner(int a, int b, int c) {
9         return a + b + c;
10    }
11
12    public static void main(String[] args) {
13        int somme2 = additionner(2, 3);    // Appelle la
première méthode
14        int somme3 = additionner(1, 4, 5); // Appelle la mé
thode surchargée
15    }
16 }
```

1.4 Utilisation des bibliothèques standard**Question 11 :**

Comment utiliser la classe Math pour calculer la racine carrée ou la puissance d'un nombre ?

Réponse :

```
1 public class UtilisationMath {
2     public static void main(String[] args) {
3         double nombre = 16.0;
4
5         // Calcul de la racine carrée
6         double racine = Math.sqrt(nombre); // 4.0
7
8         // Calcul de la puissance
9         double puissance = Math.pow(2, 3); // 8.0
10    }
11 }
```

Question 12 :

Comment utiliser la classe Arrays pour trier un tableau ?

Réponse :

```
1 import java.util.Arrays;
2
3 public class TriTableau {
4     public static void main(String[] args) {
5         int[] nombres = {5, 3, 8, 1, 2};
6     }
```

```

7      // Tri du tableau
8      Arrays.sort(nombres);
9
10     // Affichage du tableau trié
11     System.out.println(Arrays.toString(nombres)); // [1,
12     2, 3, 5, 8]
13 }

```

Question 13 :

Qu'est-ce qu'un package en Java et comment importer des classes à partir de packages ?

Réponse :

- Un package est un regroupement de classes et d'interfaces similaires.
- Pour utiliser une classe d'un package, on utilise l'instruction `import`.

```

1  import java.util.Scanner;
2
3  public class LectureClavier {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          System.out.print("Entrez votre nom : ");
7          String nom = scanner.nextLine();
8          System.out.println("Bonjour, " + nom + "!");
9      }
10 }

```

Partie 2 : Exercices sans réponses

2.1 Tableaux (Arrays)

Exercice 1 :

Déclarez un tableau de 10 entiers et initialisez-le avec les valeurs de 1 à 10. Calculez et affichez la somme de ses éléments.

Exercice 2 :

Créez un programme qui lit 5 noms de fruits entrés par l'utilisateur et les stocke dans un tableau. Affichez ensuite la liste des fruits saisis.

Exercice 3 :

Déclarez un tableau multidimensionnel représentant une grille de 3x3, initialisez-le avec des valeurs entières, puis affichez la somme de chaque ligne.

2.2 Chaînes de caractères

Exercice 4 :

Écrivez un programme qui demande à l'utilisateur de saisir une phrase, puis affiche le nombre de caractères de cette phrase.

Exercice 5 :

Demandez à l'utilisateur d'entrer une chaîne de caractères et un caractère. Comptez et affichez le nombre de fois que ce caractère apparaît dans la chaîne.

Exercice 6 :

Créez une méthode qui prend en paramètre une chaîne de caractères et retourne la chaîne inversée. Par exemple, "Bonjour" devient "ruojnoB".

2.3 Fonctions (Méthodes)

Exercice 7 :

Écrivez une méthode `factorielle` qui calcule la factorielle d'un nombre entier passé en paramètre. Utilisez cette méthode dans votre programme principal pour afficher la factorielle de 5.

Exercice 8 :

Créez une méthode `estPair` qui détermine si un nombre entier est pair ou impair. La méthode retourne un booléen (`true` si pair, `false` sinon).

Exercice 9 :

Implémentez une méthode `maximum` qui prend un tableau d'entiers en paramètre et retourne la valeur maximale du tableau.

2.4 Utilisation des bibliothèques standard

Exercice 10 :

Utilisez la classe `Math` pour générer un nombre aléatoire entre 1 et 100, puis demandez à l'utilisateur de deviner le nombre. Indiquez si la supposition est trop haute ou trop basse jusqu'à ce que l'utilisateur trouve le bon nombre.

Exercice 11 :

Utilisez la classe `Arrays` pour copier un tableau d'entiers dans un autre tableau.

Exercice 12 :

Écrivez un programme qui trie une liste de noms entrée par l'utilisateur en utilisant la méthode `Arrays.sort()`.

Exercice 13 :

Importez la classe `LocalDate` du package `java.time` et affichez la date actuelle.