

TP JAVA

Partie 1: Introduction aux Classes et Objets en Java

Objectifs:

- Comprendre ce que sont les classes et les objets.
- Savoir comment définir et utiliser des classes et des objets en Java.

Contenu:

- Définition d'une classe en Java: structure et syntaxe.
- Création d'objets à partir de classes.
- Introduction aux constructeurs: définition, utilité et types (par défaut, paramétré).

Exercices Pratiques:

- Création d'une classe `Person` avec des attributs comme `name` et `age`.
- Écriture d'un constructeur pour initialiser les attributs de `Person`.
- Instanciation d'objets de type `Person` et utilisation des instances.

Partie 2: Encapsulation et Accesseurs

Objectifs:

- Comprendre le concept d'encapsulation.
- Apprendre à implémenter l'encapsulation en Java à l'aide de modificateurs d'accès.

Contenu:

- Principe d'encapsulation: pourquoi et comment.
- Modificateurs d'accès: `private`, `public`.
- Getters et Setters: définition et utilisation.

Exercices Pratiques:

- Modifier la classe `Person` pour encapsuler les attributs.
- Ajouter des getters et setters pour manipuler les attributs de manière contrôlée.
- Tester l'encapsulation en modifiant les attributs via les méthodes d'accès.

Partie 3: Héritage en Java

Objectifs:

- Comprendre le principe d'héritage.
- Apprendre à utiliser l'héritage pour réutiliser et organiser le code.

Contenu:

- Concept d'héritage: définition et utilité.
- La syntaxe de l'héritage en Java (`extends`).
- Surcharge et redéfinition de méthodes (Overloading vs Overriding).

Exercices Pratiques:

- Création d'une classe `Student` qui hérite de `Person`.
- Ajout de caractéristiques spécifiques à `Student`.
- Surcharge et redéfinition des méthodes pour montrer la différence entre les deux.

Partie 4: Polymorphisme et Interfaces

Objectifs:

- Comprendre le concept de polymorphisme.
- Apprendre à utiliser des interfaces en Java.

Contenu:

- Polymorphisme: définition et types (statique et dynamique).
- Interfaces en Java: définition, implémentation et utilité.
- Différences entre classes abstraites et interfaces.

Exercices Pratiques:

- Implémentation d'une interface `Walkable` dans la classe `Person`.
- Utilisation du polymorphisme pour appeler des méthodes de différentes manières.
- Création et utilisation de classes abstraites et interfaces pour comprendre leurs différences.