TP3: Manipulation des classes en JAVA, les bases

1 Concepts de base

1.1 Classe

Une classe est une structure qui définit les attributs (données) et les méthodes (comportements) d'un objet. Elle agit comme un plan à partir duquel des objets peuvent être créés.

En Java, une classe se déclare de cette manière :

```java
public class Personne {
    // Attributs de la classe
    String nom;
    int age;

    // Methodes de la classe
    public void afficherNom() {
        System.out.println(nom);
    }
}
```

- Le nom de la classe commencent par une majuscule.
- Les attributs d'une classe peuvent être des String, boolean, double, int...

1.2 Objet

Un objet est une instance d'une classe. Il représente une entité individuelle créée à partir de la classe et possède ses propres valeurs pour les attributs définis dans la classe.

Voici comment créer un objet en Java :

```java
Personne personne1 = new Personne();
```

1.3 Constructeur

Un constructeur est une méthode spéciale utilisée pour initialiser un objet lors de sa création. Il permet de définir les valeurs initiales des attributs de l'objet.

- Le constructeur porte le même nom que sa classe.
- Il existe deux types de constructeurs :

- Constructeur par défaut (pas d'arguments)
- Constructeu avec arguments

Exemple d'un constructeur :

```java
public class Personne {
    String nom;
    int age;

    // Constructeur avec les arguments: nom et age
    public Personne(String nom, int age) {
        this.nom = nom;
        this.age = age;
    }
}
```

1.4 Encapsulation

L'encapsulation est un principe fondamental de la POO qui consiste à protéger les attributs d'une classe en les déclarant privés (`private`). Ces attributs ne peuvent alors être modifiés ou lus qu'à travers des méthodes publiques appelées `getters` et `setters`.

1.5 Getters et Setters

Les `getters` et `setters` sont des méthodes qui permettent de lire et de modifier les attributs d'une classe depuis l'extérieur tout en préservant l'encapsulation. Ils offrent un contrôle sur l'accès et la modification des données de la classe.

Voici un exemple :

```java
public class Personne {
    private String nom;
    private int age;

    // Getter pour obtenir la valeur de 'nom'
    public String getNom() {
        return this.nom;
    }

    // Setter pour modifier la valeur de 'nom'
    public void setNom(String nom) {
        this.nom = nom;
    }
}
```

En resumé:

Pour implémenter l'encapsulation, il nous faut trois étapes :

1. C'est réduire la visibilité des variables à private. De ce fait, les attributs de la classe sont visibles uniquement au sein de la classe elle-même et les autres classes ne peuvent même pas les voir.
2. Au sein de la même classe et pour chaque attribut, implémenter des méthodes pour lire l'attribut (Getter) et une autre pour le modifier (Setter).
3. Dans le Setter, on fait des tests de contrôle avant de faire la modification : if(age > 0) { // modifier age }. (facultative)