**Les getters et setters en Java**

Cf. : https://melo-code.com/les-getters-et-setters-en-java/

Getters et setters en Java.

Lorsque vous faites de la programmation orienté objet, il est important de respecter quelques principes. Dans ce chapitre, on va aborder les getters et les setters tout en appuyant sur le principe d’encapsulation. **L’encapsulation** signifie que l’état d’un objet ne peut être modifié que par les [méthodes](https://melo-code.com/appeler-les-methodes-sur-un-objet/)de cet objet. Pour manipuler un objet il est important de passer par des méthodes, mais pas directement à travers ses attributs.

Dans cet exemple, on va violer le principe d’encapsulation pour avoir une idée plus claire de ce que nous allons faire. Une classe personne pour la création de notre objet personne et une classe main pour le lancement de notre test.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **package** accesseurs;    **public** **class** Personne {    **public** String nom;  **public** String prenom;  **public** **int** age;    } |

La classe Main dans laquelle nous allons créer notre objet personne en utilisant l’instance suivie d’un point pour modifier les valeurs de ses attributs.

package accesseurs;

public **class** **Main** {

public **static** **void** main(String[] args) {

Personne pers = **new** Personne();

pers.nom = "Joseph";

pers.prenom = "Joe";

pers.age = 30;

System.out.println("Je suis " + pers.nom + " " + pers.prenom + " et j'ai " + pers.age + " ans");

}

}

Code language: JavaScript (javascript)

Ce code s’exécutera sans problème. La visibilité des attributs sont **public**, c’est-a-dire qu’ils sont accessibles partout.

**Application du principe d’encapsulation**

Nous allons protéger nos données en utilisant le modificateur de visibilité : **private**. Lorsqu’un attribut (propriété) d’une classe est **private**, il est accessible uniquement depuis sa propre classe. Pour accéder aux propriétés **private**d’une classe dans une autre classe, on utilisera des **getters** et des **setters**;

* Les **getters** nous permettent d’accéder au contenu des propriétés privées.
* Les **setters** nous permettent de modifier les contenus des propriétés privées.

**Exemples de getters et setters**

package accesseurs;

public **class** **Personne** {

private String nom;

private String prenom;

private int age;

public String getNom() {

**return** nom;

}

public **void** setNom(String nom) {

**this**.nom = nom;

}

public String getPrenom() {

**return** prenom;

}

public **void** setPrenom(String prenom) {

**this**.prenom = prenom;

}

public int getAge() {

**return** age;

}

public **void** setAge(int age) {

**this**.age = age;

}

}

Code language: JavaScript (javascript)

Les **getters** et **setters** sont des méthodes publiques qui nous permettent d’avoir accès au contenu d’une variable (propriété) privée d’un objet.

package accesseurs;

public **class** **Main** {

public **static** **void** main(String[] args) {

Personne pers = **new** Personne();

pers.setNom("Joseph");

pers.setPrenom("Joe");

pers.setAge(30);

System.out.println("Je suis " + pers.getNom() + " " + pers.getPrenom() + " et j'ai " + pers.getAge() + " ans");

}

}

Code language: JavaScript (javascript)

**N.B. :**Une [classe](https://melo-code.com/notion-de-classes-en-java/) est constituée d’attributs et de méthodes.

Lorsque vous créez un objet d’une classe, tous les attributs et méthodes appartiendront aussi à l’objet créé.