

"Si vous n'aimez pas les tests de votre produit, vos clients n'aimeront probablement pas non plus le tester"



Java pour les Testeurs

Zied HANNACHI

Test lead automation chez ALL4TEST, pure player QA

Sommaire

- ▶ Introduction
- ▶ Classe
- ▶ Objets
- ▶ Méthodes
- ▶ Modificateurs
- ▶ Constructeur
- ▶ Héritage
- ▶ Polymorphisme
- ▶ Abstraction
- ▶ Encapsulation
- ▶ Garbage Collection
- ▶ Variables



Introduction à Java

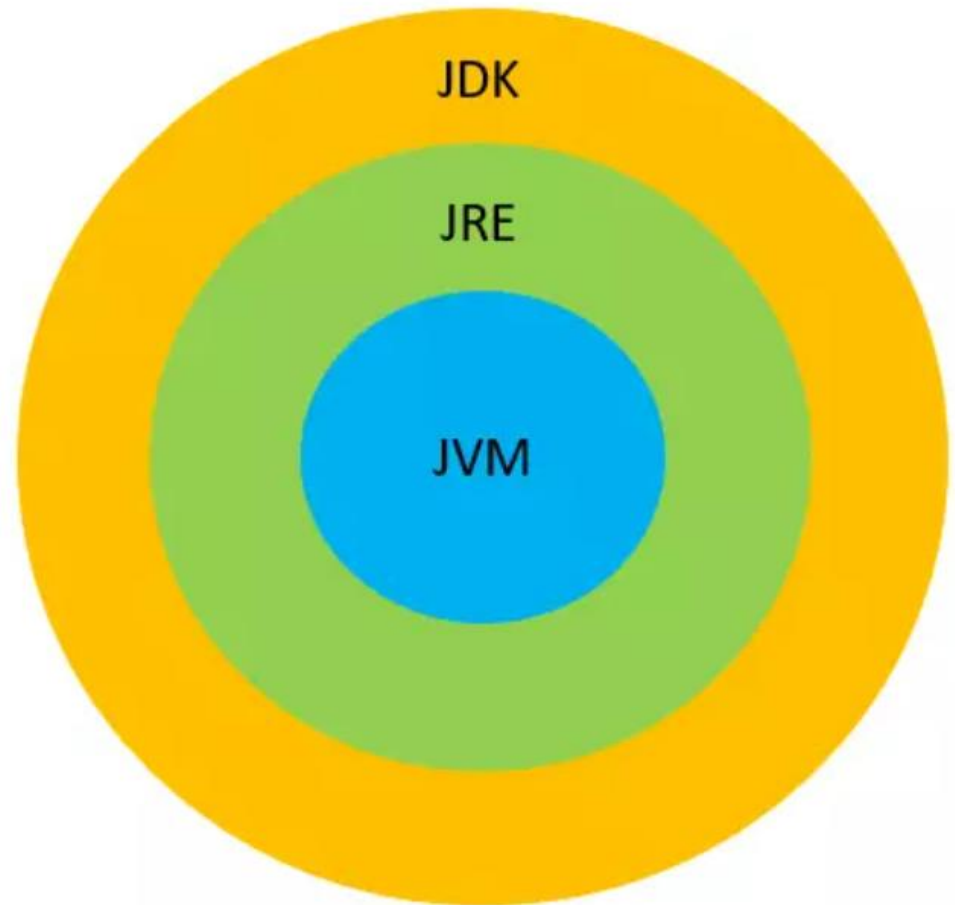
2

Java est un langage de programmation de haut niveau développé à l'origine par Sun Microsystems en 1995. Java est un langage indépendant de la plate-forme. Il fonctionne sur plusieurs plates-formes telles que UNIX, Windows, Mac OS. La programmation du langage Java est basée sur le concept de POO.



JDK, JRE et JVM

Nous devons trois terminologies en Java telles que JDK, JRE et JVM. Ici, je donne une idée de base de ces termes dans le prochain post, nous verrons une explication détaillée.



JDK, JRE et JVM

- ▶ JDK signifie Java Development Kit.

En utilisant JDK, nous pouvons développer, compiler et exécuter de nouvelles applications et également modifier des applications existantes. Nous devons installer JDK dans la machine des développeurs où nous voulons développer de nouvelles applications ou modifier des applications existantes. JDK comprend JRE et des outils de développement (environnement pour développer, déboguer et surveiller les programmes Java).

- JRE signifie Java Runtime Environment.

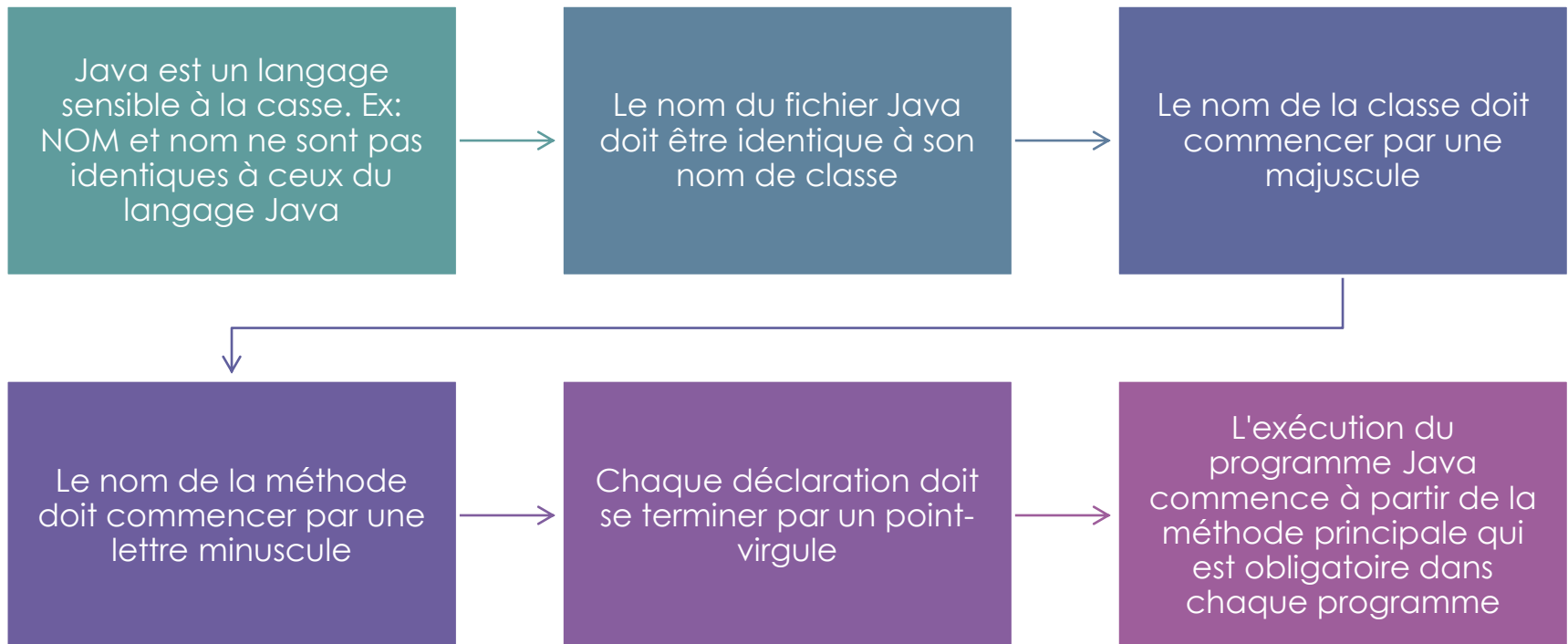
En utilisant JRE, nous ne pouvons exécuter que des applications déjà développées. Nous ne pouvons pas développer de nouvelles applications ou modifier des applications existantes. Comme son nom l'indique, JRE fournit uniquement l'environnement d'exécution.

- JVM signifie Java Virtual Machine.

JVM pilote le code java. En utilisant JVM, nous pouvons exécuter du code d'octet java en les convertissant en langage machine OS actuel.

Syntaxe Java

5



Qu'est-ce qu'une classe Java



Une classe Java peut être définie comme un modèle ou un plan qui décrit l'état / le comportement de son objet.



En d'autres termes, une classe est utilisée pour créer des objets.

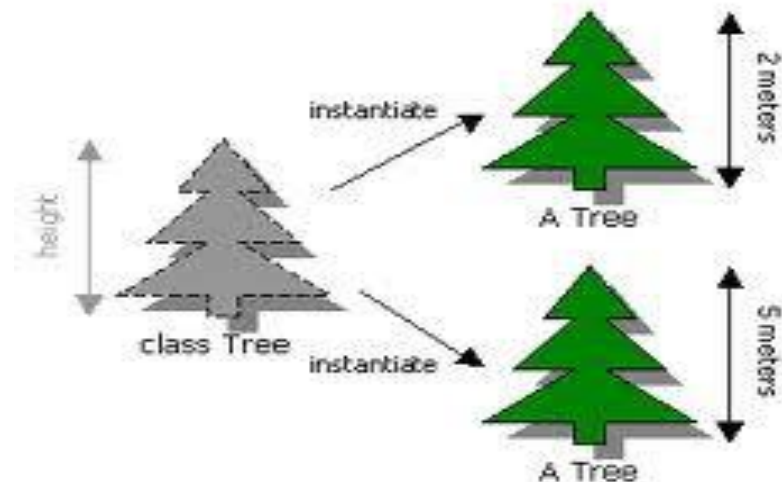


Fig. 1: Instantiating two Trees from the Tree class

Créer une classe

- ▶ Une classe est déclarée en utilisant le mot-clé «class».

- ▶ `public class Tree`

- ▶ `{`

- ▶ `//This is our Tree class`

- ▶ `//All codes goes here`

- ▶ `}`

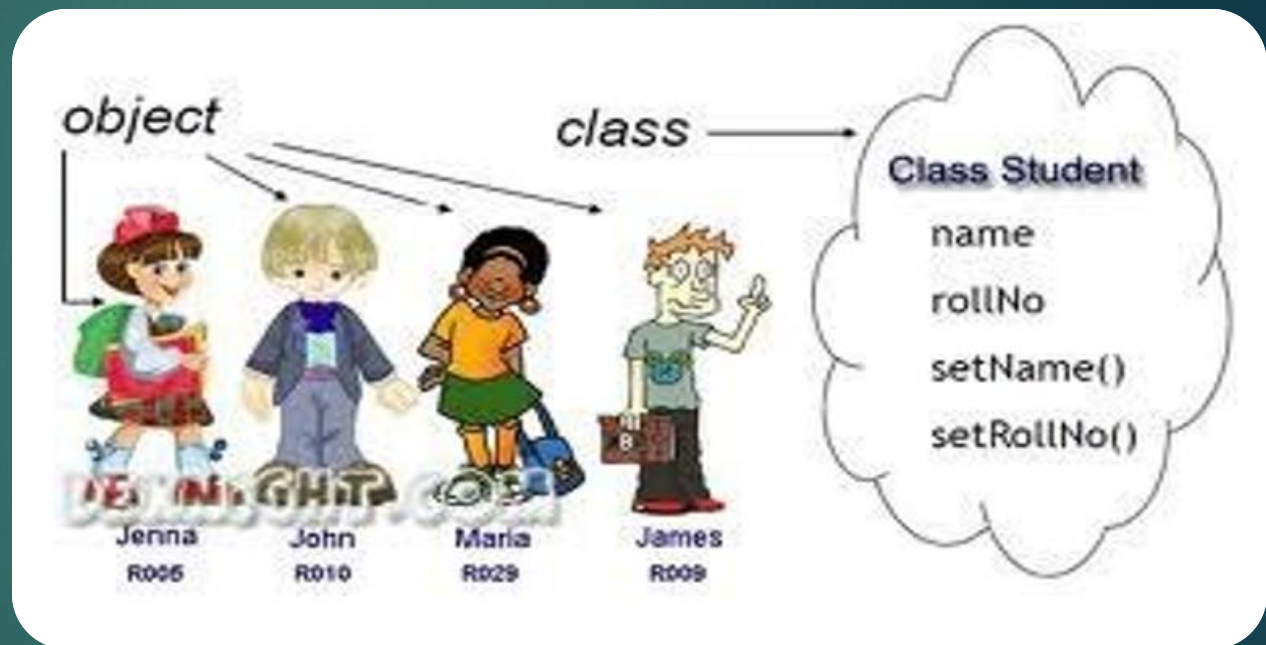
Quels sont les objets



LES OBJETS NE SONT
QUE L'INSTANCE DE LA
CLASSE.



UNE SEULE CLASSE PEUT
CRÉER UN NOMBRE
ILLIMITÉ D'OBJETS
UNIQUES.



Création d'objets

- En java, un objet est créé lorsque quelqu'un dit «new».
- A chaque «new», un nouvel objet d'une classe est créé.

Par exemple:



`NEW DOG();`



`NEW TREE();`



`NEW
STUDENT();`

Où vivent les objets

- ▶ Les objets vivent dans Java heap.

Qu'est-ce que Java heap?

Java heap n'est rien d'autre que l'espace mémoire pris par JVM du système d'exploitation.

Tous les objets sont créés dans ce heap (espace). Chaque fois que la JVM rencontre un «new» mot clé, elle crée un objet en heap.

- Lorsqu'il n'y a pas d'espace dans le heap pour créer des objets, la JVM renvoie l'erreur «Out of Memory».

Création du premier objet

```
public class Student  
{  
    public static void main(String[] args)  
    {  
        new Student();  
    }  
}
```



Qu'est-ce qu'une méthode

La méthode décrit le comportement d'un objet. Une méthode consiste en une collection d'instructions qui exécute une action. Les méthodes sont également appelées procédures ou fonctions

```
public int sum(int a, int b, int c){  
    // method body  
}
```

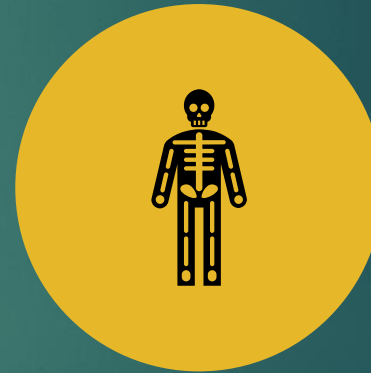
```
public void sum(int a, int b, int c){  
    // method body  
}
```

Les modificateurs

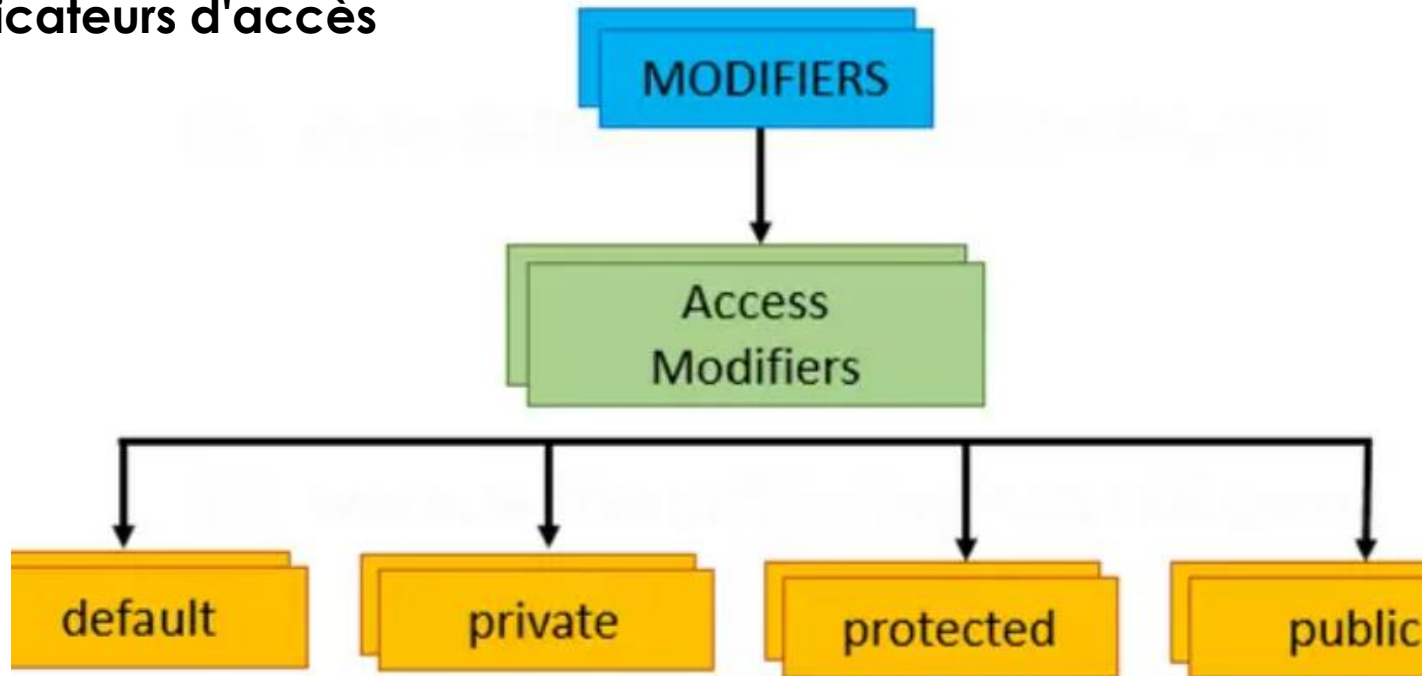
En Java, il existe deux types de modificateurs



MODIFICATEURS D'ACCÈS: LES
MODIFICATEURS D'ACCÈS SONT
SUBDIVISÉS EN QUATRE TYPES TELS QUE
DEFAULT, PUBLIC, PRIVATE, PROTECTED



MODIFICATEURS SANS ACCÈS: LES
MODIFICATEURS SANS ACCÈS SONT
SUBDIVISÉS EN QUATRE TYPES TELS QUE
STATIC, FINAL, ABSTRACT, SYNCHRONIZED

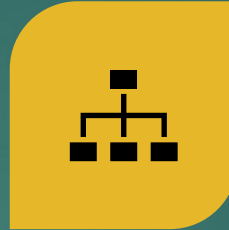


Les modificateurs

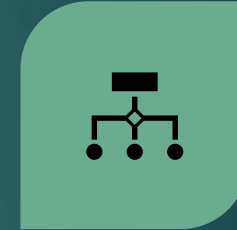
Les modificateurs



LA PORTÉE DU MODIFICATEUR D'ACCÈS **DEFAULT** EST LIMITÉE AU PACKAGE UNIQUEMENT. SI NOUS NE MENTIONNONS AUCUN MODIFICATEUR D'ACCÈS, IL AGIT COMME UN MODIFICATEUR D'ACCÈS PAR DÉFAUT.



LA PORTÉE DU MODIFICATEUR D'ACCÈS **PRIVATE** NE S'APPLIQUE QU'AUX CLASSES.



LA PORTÉE DU MODIFICATEUR D'ACCÈS **PROTECTED** SE TROUVE DANS UN PACKAGE ET ÉGALEMENT EN DEHORS DU PACKAGE VIA L'HÉRITAGE UNIQUEMENT.



LA PORTÉE DU MODIFICATEUR D'ACCÈS **PUBLIC** EST PARTOUT. IL N'A AUCUNE RESTRICTION. LES MEMBRES, MÉTHODES ET CLASSES DE DONNÉES DÉCLARÉS PUBLICS SONT ACCESSIBLES DE N'IMPORTE OÙ.

Consultez ce tableau simple pour comprendre facilement les modificateurs d'accès

16

MODIFIER	ACCESS LEVELS			
	Class	Package	Subclass	Everywhere
public	Y	Y	Y	Y
protected	Y	Y	Y	N
default	Y	Y	N	N
private	Y	N	N	N

Les modificateurs

Constructeur

Le constructeur en Java est utilisé dans la création d'un objet qui est une instance d'une classe. Le nom du constructeur doit être identique au nom de la classe. Cela ressemble à une méthode mais ce n'est pas une méthode. Il ne renverra aucune valeur. Nous avons vu que les méthodes peuvent renvoyer une valeur. S'il n'y a pas de constructeur dans une classe, le compilateur crée automatiquement un constructeur par défaut.



Héritage

L'HÉRITAGE EST UN PROCESSUS DANS LEQUEL UNE CLASSE HÉRITE DES PROPRIÉTÉS D'UNE AUTRE CLASSE.



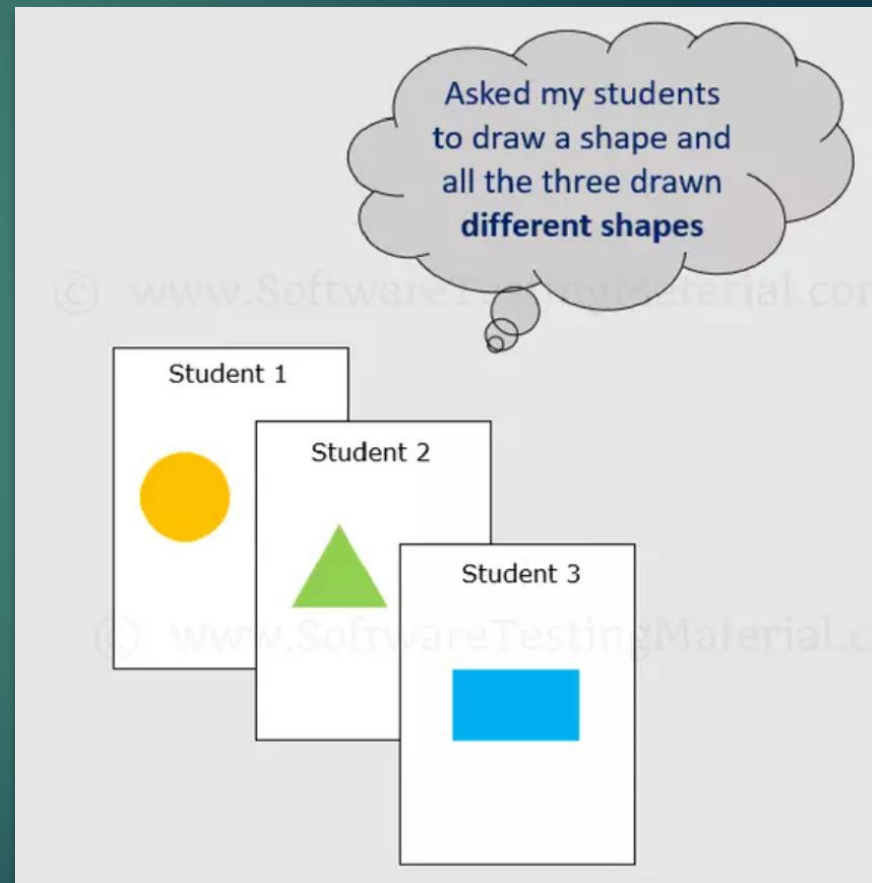
Héritage

► Disons que nous avons deux classes à savoir la classe parent et la classe enfant. La classe enfant est également connue sous le nom de classe dérivée. La classe enfant hérite des propriétés de la classe parent. Le principal objectif de l'héritage est la réutilisation du code.

► Supposons que nous ayons une classe nommée Laptop, Apple MacBook Pro, Lenovo Yoga. Les cours Apple MacBook Pro et Lenovo Yoga étendent la classe Laptop pour hériter des propriétés de la classe Laptop.

Polymorphisme

- ▶ Le polymorphisme nous permet d'effectuer une tâche de plusieurs façons. Brisons le mot polymorphisme et voyons-le, 'Poly' signifie 'Beaucoup' et 'Morphos' signifie 'Formes'.



Polymorphisme

21

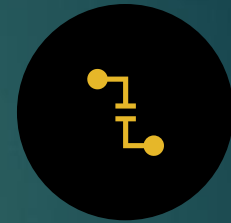


VOYONS UN EXEMPLE. SUPPOSONS QUE NOUS AYONS QUATRE ÉLÈVES ET NOUS LEUR AVONS DEMANDÉ DE DESSINER UNE FORME. TOUS LES QUATRE PEUVENT DESSINER DES FORMES DIFFÉRENTES COMME LE CERCLE, LE TRIANGLE ET LE RECTANGLE.

Il existe deux types de polymorphisme en Java



POLYMORPHISME AU
MOMENT DE LA
COMPILATION (LIAISON
STATIQUE) - SURCHARGE
DE MÉTHODE



POLYMORPHISME AU
MOMENT DE
L'EXÉCUTION (LIAISON
DYNAMIQUE) -
SUBSTITUTION DE
MÉTHODE

Abstraction

- L'abstraction est la méthodologie utilisée pour masquer la mise en œuvre des détails internes et montrer la fonctionnalité aux utilisateurs.



Abstraction

23



Exemple: téléphone portable.



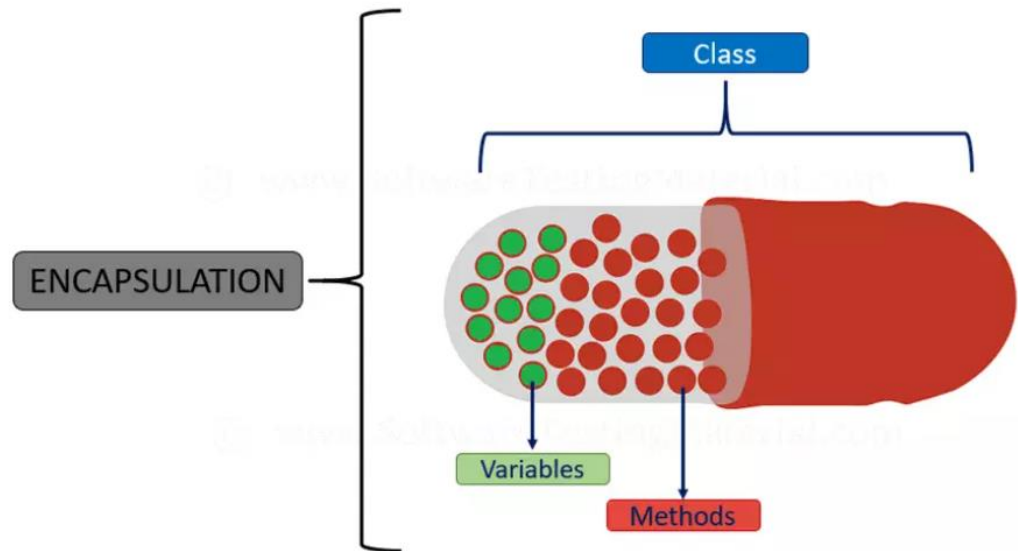
Un profane qui utilise un téléphone mobile ne sait pas comment cela fonctionne en interne, mais il peut passer des appels téléphoniques.



L'abstraction en Java est réalisée à l'aide de classes et d'interfaces abstraites. Voyons ce qu'est la classe abstraite et l'interface en détail.

Encapsulation

► L'encapsulation est un mécanisme de liaison du code et des données dans une seule unité. Prenons un exemple de capsule. Différents médicaments en poudre ou liquides sont encapsulés dans une capsule. De même en encapsulation, toutes les méthodes et variables sont regroupées dans une seule classe.



Garbage Collection



Garbage Collection est le mécanisme fourni par JVM pour nettoyer le heap, afin que de nouveaux objets puissent être créés.



Il détruit les objets «non utilisés» ou éligibles à la “garbage collection”.



Tout objet est considéré comme éligible pour la garbage collection (GC) s'il n'y a pas de «variable de référence» qui lui est attachée.



Alors, qu'est-ce qu'une «variable de référence» ?

Affectation d'objets à une variable de référence



AVANT DE CRÉER
UNE VARIABLE DE
RÉFÉRENCE, NOUS
DEVONS SPÉCIFIER
«OBJET DE QUELLE
CLASSE» ELLE VA SE
RÉFÉRER.



EN D'AUTRES TERMES,
NOUS DEVONS
SPÉCIFIER LE «TYPE DE
CLASSE» DE LA
VARIABLE DE
RÉFÉRENCE. PAR
EXEMPLE:

```
public class Test
{
    public static void main(String[] args)
    {
        Test t = new Test();
        //where t is the reference variable
        which stores the Address of Test object
        in heap
    }
}
```

Déclaration et initialisation des variables primitives

27

- Integer

- **int** i = 234242425;

- **long** l = 284798247287427427428947l;

- //suffix 'l' is must otherwise compiler will treat it as integer.

- Float

- **float** = 2342.34f

- //suffix 'f' is must otherwise compiler will treat it as a double

- **double** = 298472847242478927.2942949274

- Character

- Char** c = 'j'; // only single character is allowed

- Boolean

- Boolean** b = true // only true or false is allowed

Java String et String Concatenation

28

En java, "String" est une classe et non un type de données et il peut être instancié comme les autres classes

```
String s = new String ();
```

La concaténation de chaînes est essentiellement un moyen de combiner deux chaînes ou plus en une seule chaîne. Cela se fait en utilisant l'opérateur «+».

```
String s = "We" + "are" + "learning" + "java" + "!"
```

Les valeurs de chaîne peuvent être concaténées avec tout autre type de données.

```
boolean b = true;
```

```
String s = "this is" + " " + b
```

Auteur

► Zied Hannachi est ingénieur d'automatisation de test Full Stack, il évolue aujourd'hui en tant qu'Expert de Test. Il a ensuite enchaîné les missions et a pu ainsi se bâtir une solide expérience qu'il continue à développer. Il a passé 8 ans à jouer avec l'automatisation dans des projets gigantesques en France.

► Il est passionné par la conception des frameworks d'automatisation qui suivent les concepts OOP et les design pattern. Pour l'automatisation, ces armes sont Selenium (Java, C #, JavaScript, Python et Ruby), Appium, REST-Sharp, REST-Assured, Cucumber, Specflow, etc.

► Il se consacre aux formations de test manuel et automatique et accompagne les sociétés dans l'audit, implémentation de stratégie de test et implémentation des framework d'automatisation de test.



Merci