



PROGRAMMATION ORIENTÉE OBJET JAVA

LICENCE 2 INGÉNIERIE - INFORMATIQUE

2020– 2021

Marie NDIAYE



PRÉSENTATION DU COURS

INFORMATIONS GÉNÉRALES

Enseignante	Marie NDIAYE DIOP
Adresse mail	marie.ndiaye@univ-zig.sn
Objectifs généraux	<ul style="list-style-type: none">• Connaître les principes de la programmation orientée objet• Comprendre les notions de classe et d'objet en Java• Connaître les bases du langage Java• Savoir utiliser les possibilités offertes par Java pour créer des applications
Séances	CM : 24 heures TD : 12 heures TP : 12 heures
Evaluation	Devoir/Projet (30%) Examen (70%)

PROGRAMME DU COURS

- Introduction à la programmation Orientée Objet et Java
- Les bases du langage Java
- L'agrégation
- L'héritage et le polymorphisme
- Les exceptions
- La programmation graphique

BIBLIOGRAPHIE

■ Livres

- Programmer en Java – Claude DELANNOY – Editions EYROLLES
- Le livre de Java premier langage avec 90 exercices corrigés – Anne Tasso – Editions EYROLLES
- JAVA 7 – Les fondamentaux du langage Java – Thierry GROUSSARD – Editions ENI
- JSP et servlets efficaces – Production de sites dynamiques en Java – Jean-Luc DELEAGE – Editions DUNOD
- Java Enterprise Edition – Le développement d'applications web avec JEE 6 [2ième édition] – Thierry GROUSSARD – Editions ENI

■ World Wide Web

- Développez.com → <https://java.developpez.com/cours/>
- Openclassrooms → <https://openclassrooms.com/courses/apprenez-a-programmer-en-java>
- Misfu → <http://www.misfu.com/cours/tutoriel/apprendre-java-363.html>

RÉFÉRENCES

■ Supports de cours et tutoriels

- APPRENTISSAGE DU LANGAGE JAVA – Serge Tahé – ISTIA - Université d'Angers
- Introduction à Java – Borland® Jbuilder

■ World Wide Web

- <http://www.dicodunet.com>
- <http://www.webmaster-hub.com/publication/java-un-langage-de-programmation.html>
- <http://www.techno-science.net>
- <http://membres.multimania.fr/deneau/jsp.html>



INTRODUCTION À LA POO ET À JAVA

LA PROGRAMMATION ORIENTEE OBJET (POO)

- Style de programmation proposé par **Alan Kay** dans les années 1970
- Consiste en la définition et l'interaction de **briques logicielles** appelées **objets**.
 - un objet représente un concept, une idée ou toute entité du monde physique
 - Exemple d'objets : une voiture, une personne, un dossier médical

LES CONCEPTS OBJET

- La classe
- L'encapsulation
- L'héritage
- Le polymorphisme

L'OBJET

- Un objet est une structure de données qui regroupe les données et les moyens de traitement de ces données (méthodes).
- Il est caractérisé par
 - Un état : combinaison de ses propriétés
 - Un comportement : code qui sera inséré dans les méthodes de l'objet afin de le faire agir
 - Un identifiant : permet de différencier les objets

LE CONCEPT DE CLASSE

- Une classe est la description d'un ensemble d'objets ayant une structure de données commune et disposant des mêmes méthodes.
- Le concept de classe correspond à la généralisation de la notion de type que l'on rencontre dans les langages classiques.

Classe = Type	Objet = Variable
Classe = Moule	Objet = Gâteau
Integer	i



EXEMPLE

■ Classe personne

Attributs

Personne

prénom : String
nom : String
âge : int

Méthodes

afficherPrénom () : void
incrémenterAge() : void
ModifierNom(String)

...

■ Objet de type Personne

= **instance** de la classe Personne

P : Personne

prénom = Alpha
nom = DIATTA
âge = 25

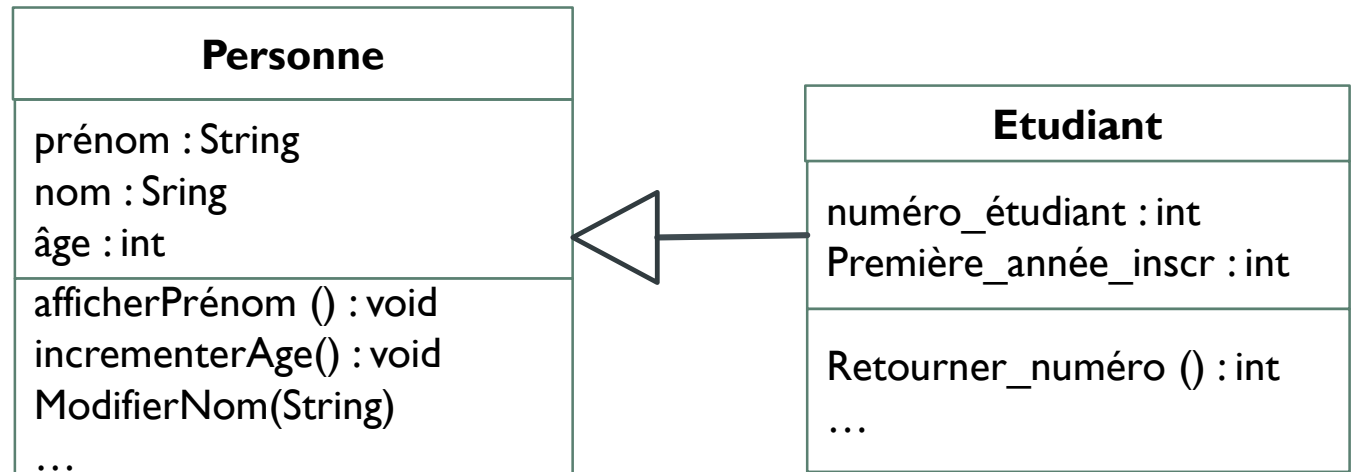
LE CONCEPT D'ENCAPSULATION

- L'encapsulation des données consiste à cacher l'implémentation de l'objet, c'est-à-dire en empêchant l'accès aux données par un autre moyen que les services (méthodes) proposés.
- **Avantages** : intégrité, réutilisabilité, maintenance
- **Exemple** : On ne peut accéder au prénom, au nom et à l'âge d'une personne qu'en utilisant les méthodes *afficherPrénom*, *incrémenterAge* et *ModifierNom* définies dans la classe *Personne*.

LE CONCEPT D'HÉRITAGE

- L'héritage est une technique qui permet de définir une classe (classe fille) à partir d'une classe existante (classe de base) à laquelle on ajoute de nouveaux attributs et de nouvelles méthodes.
- **Avantages** : réutilisabilité, éviter la réécriture d'un même code

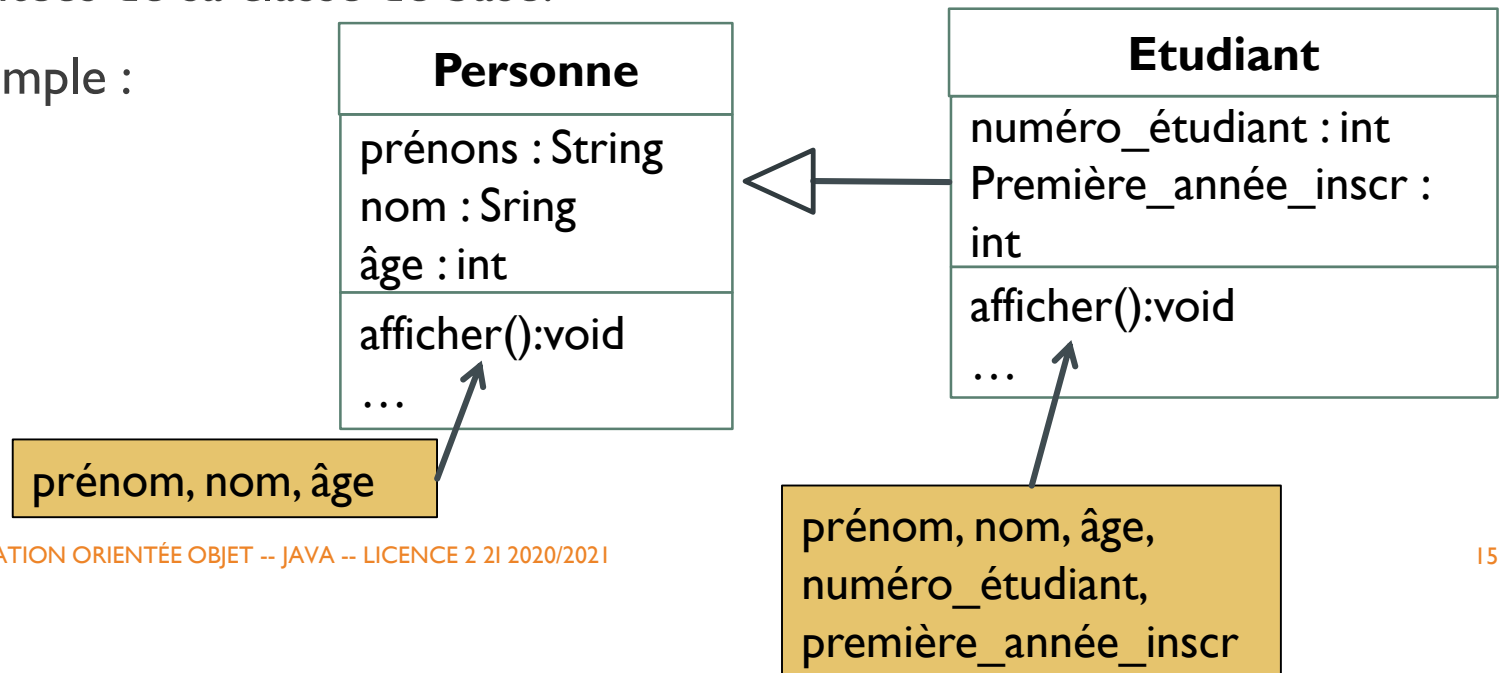
■ Exemple :



LE CONCEPT DE POLYMORPHISME

- Le polymorphisme représente la faculté d'une méthode à pouvoir s'appliquer à des objets de classes différentes.
- Mise en œuvre : une classe peut redéfinir certaines des méthodes héritées de sa classe de base.

■ Exemple :



AGRÉGATION

- L'agrégation est une relation entre deux classes, spécifiant que les objets d'une classe sont des composants de l'autre classe.
- L'agrégation permet d'assembler des objets de base, afin de construire des objets plus complexes.
- Exemple :

Chaise
...
...

Table
...
...

Bureau
une_table : Table
une_chaise : Chaise
...
...



INTRODUCTION À JAVA

NAISSANCE ET ÉVOLUTION DE JAVA

■ Origines

- Créateurs: James GOSLING, Patrick NAUGHTON, avec le soutien de Bill JOY

- Naissance: 1991 (logiciels pour appareils électroniques)

- Présentation officielle: 23 mai 1995 (navigateur web HotJava)



■ Evolution

- 1996 → versions 1.01 & 1.02

- 1998 → version 1.2 (rebaptisé Java 2)

- De nos jours → Version 8 Update 231 (15 octobre 2019)

LE LANGAGE JAVA 1/2

- Est **orienté objet**

- Reprend en grande partie la syntaxe du langage **C++**.

- Est **compilé**

- En bytecode, i.e., code intermédiaire indépendant de la machine

- Est **interprété**

- Reprend le concept de machine virtuelle du langage **Pascal**

- Code source → byte code (pseudo langage universel) → code exécutable

- Le bytecode est interprété par une machine virtuelle Java

LE LANGUAGE JAVA 2/2

- Est **fortement typé**
 - Toute variable doit être déclarée avec un type
 - Le compilateur vérifie que les utilisations des variables sont compatibles avec leur type (notamment via un soustypage correct)
- Tout est objet à l'exception des types primitifs
 - Les types sont d'une part fournis par le langage, mais également par la définition des classes
- Remplace l'**héritage multiple** par l'implémentation des interfaces.

JAVA : UN "PUR" LANGAGE DE POO

■ **Orienté objet**

- Tout est objet (instance de classe)
- Encapsulation des données
- Les procédures sont des méthodes → pas de procédure pouvant s'exécuter indépendamment de tout objet
- Programme → classe ou réunion de plusieurs classes

■ **Pas orienté objet**

- Types primitifs
- Méthodes de classe : utilisables de façon indépendant d'un objet
- Encapsulation non absolue

PROGRAMMATION ÉVÈNEMENTIELLE EN JAVA

- **Principe** : le programme sera principalement défini par ses réactions aux différents événements qui peuvent se produire. L'utilisateur pilote le programme qui répond à ses demandes.
- **Programmes à interface graphique** (*G.U.I : graphical user interface*) : la communication se fait par l'intermédiaire de composants tels que les menus déroulants, les barres d'outils, les boîtes de dialogue, etc.
- **Mise en œuvre** → l'application a une boucle principale divisée en deux sections :
 - Section qui détecte les événements
 - Section qui gère les évènement

EXEMPLE : HELLO WORD

- Dans un fichier de nom **HelloWorld.java**
- Règles
 - Toute classe publique doit être dans un fichier qui a le même nom que la classe.
 - Tout code doit être à l'intérieur d'une classe.

```
public class HelloWorld {  
    /* Un style de commentaire  
    sur plusieurs lignes. */  
    public static void main(String[ ] args) {  
        // Un commentaire sur une seule ligne  
        System.out.println("Bonjour a vous les étudiants!");  
    }  
}
```

LES OUTILS QUE NOUS UTILISERONS

- Java SE Development Kit 8u171 (Compilateur, machine virtuel, etc.)



- Sublim Text (Editeur)



- Eclipse (Environnement de développement intégré)

