CHAPITRE I:
Introduction à la Programmation Orientée Objet

Centre Universitaire de Mila
2ième Année Licence Informatique
Matière: Programmation Orientée Objet
Responsable de la matière: DR. SADEK BENHAMMADA

Plan du cours

1. **Paradigmes de programmation**
2. **De la programmation procédurale à la programmation orientée objet**
3. **Concepts de base de l'orienté objet**
4. **Langage Java**

Plan du cours

1. **Paradigmes de programmation**
2. De la programmation procédurale à la programmation orientée objet
3. Concepts de base de l'orienté objet
4. Langage Java

1. Paradigmes de programmation

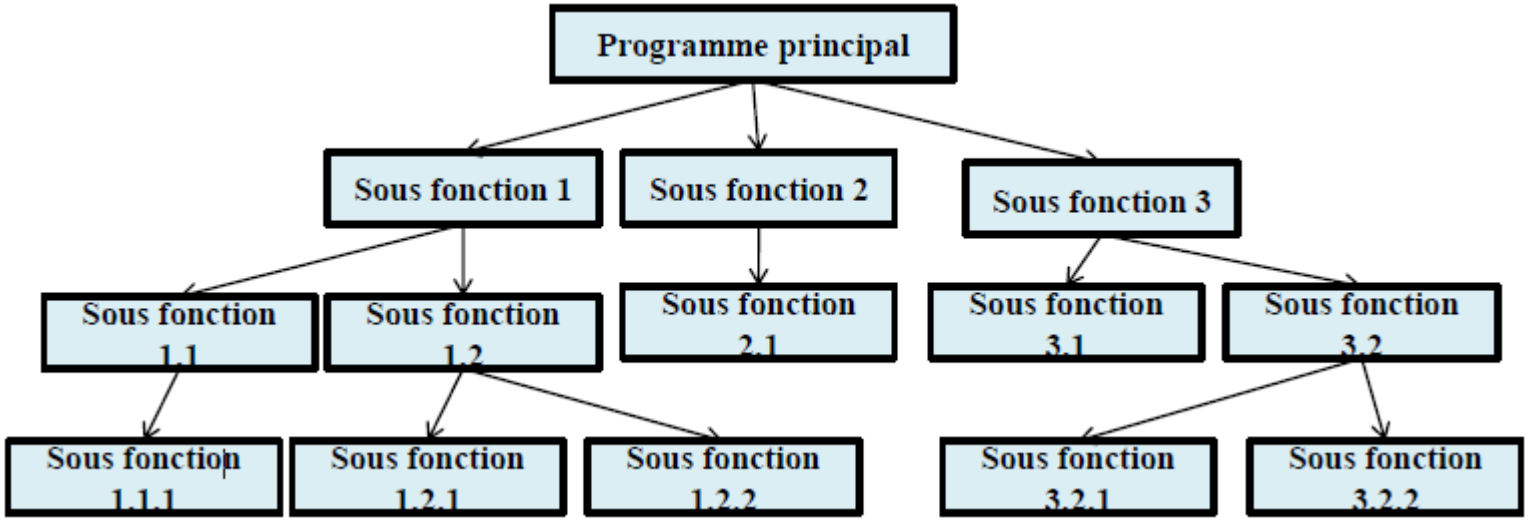
- Un paradigme de programmation est une manière de programmer, basée sur un ensemble de principes ou une théorie.
- Les langages de programmation peuvent être classés en plusieurs paradigmes:

1. **Le paradigme procédural (ou impératif):** C, Pascal, Fortran, etc.
2. **Le paradigme orienté objet :** Java, C++, Python, C#.
3. **Le paradigme déclaratif :**
   - **Programmation descriptive:** HTML, XML, LaTeX.
   - **Programmation fonctionnelle:** Scheme, Lisp, etc.
   - **Programmation logique:** Prolog, etc.

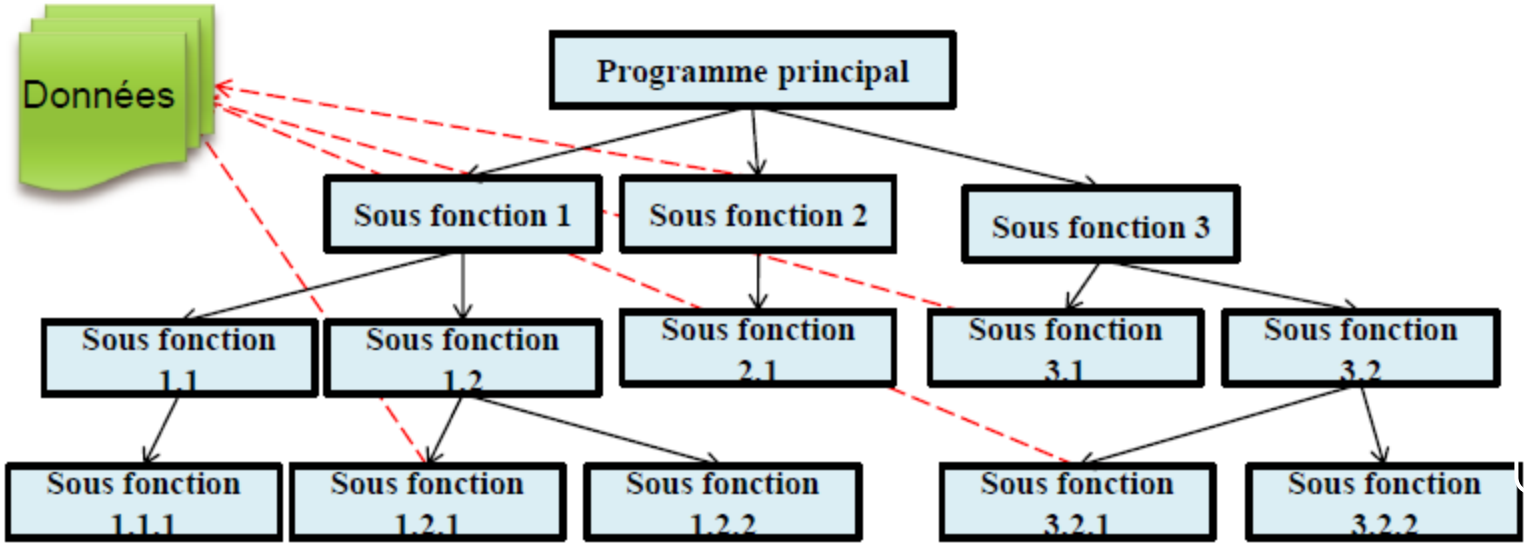
Plan du cours

1. Paradigmes de programmation
2. **De la programmation procédurale à la programmation orientée objet**
3. Concepts de base de l'orienté objet
4. Langage Java

- Le programme principal est décomposé en **modules** (**fonctions** ou **procédures**),
- Chaque module est en suite décomposé en sous-modules,
- La décomposition se poursuit jusqu'à arriver à des composants maîtrisables (la longueur ne dépasse pas si possible une page).
- Le programme principal est donc une succession d'appels à des fonction et procédures.

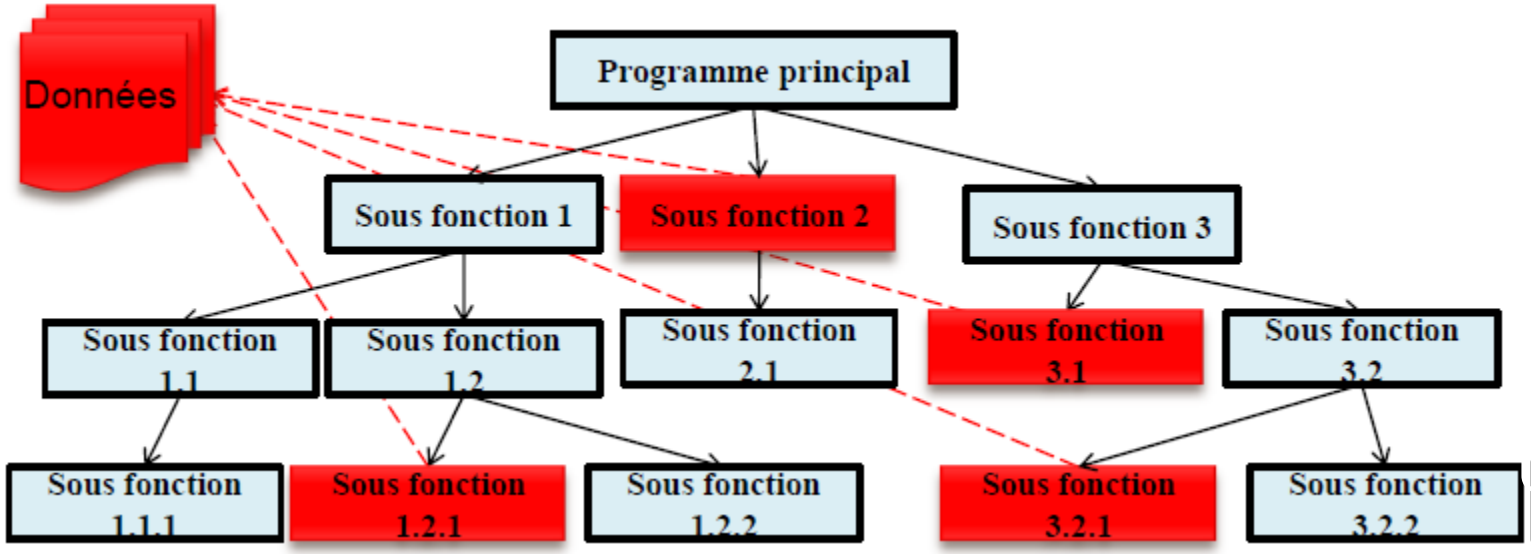


- La programmation procédurale sépare entre les données et les traitements (programmes) qui les manipulent.
- L'accès aux données peut-être direct (base de données), ou par le passage de paramètres depuis le programme principal.



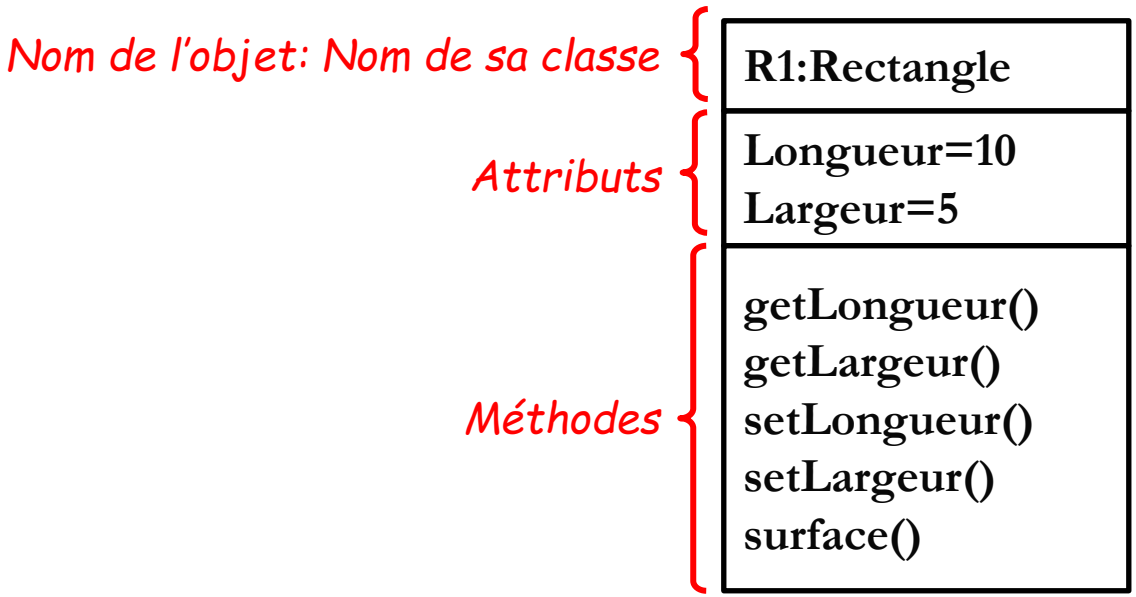
Limites

- **Limites: Maintenance et réutilisabilité difficile:** Une modification d'une structure de données est répercutée sur tous les programmes qui les manipulent.

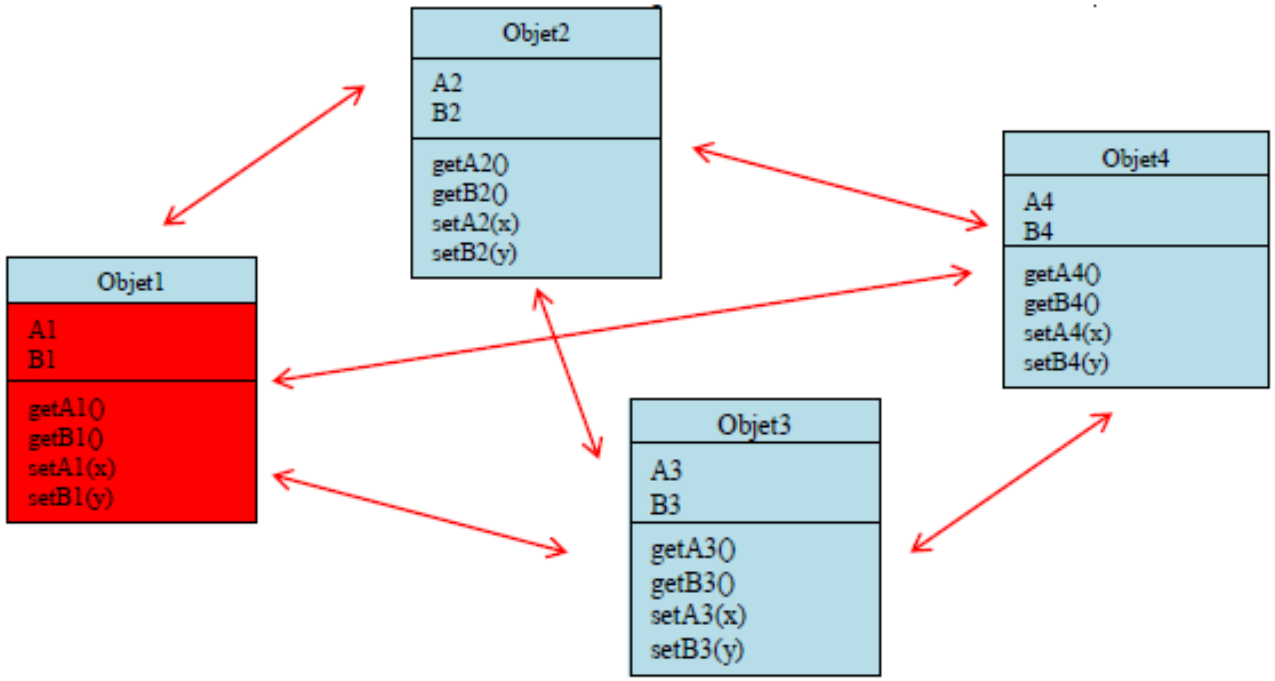


B. La programmation orientée objet

- Association des structures de **données** et les **traitements** qui les manipulent dans des **entités cohérentes**
- Ces entités sont les **objets.**
- Les structures de données associées à un **objet** sont ses **attributs.**
- Les traitement associées à un **objet** sont ses **méthodes.**
- Les **attributs** d'un objet **ne sont accessibles que par ses méthodes**.
- **Exemple**



- L'approche orientée objet considère le logiciel comme une collection d'objets.
- Les fonctionnalités du logiciel émergent alors de l'interaction entre les différents objets qui le constituent.



Plan du cours

1. Paradigmes de programmation
2. De la programmation procédurale à la programmation orientée objet
3. **Concepts de base de l'orienté objet**
4. Langage Java

3. Concepts de base de l'orienté objet

A. L'objet

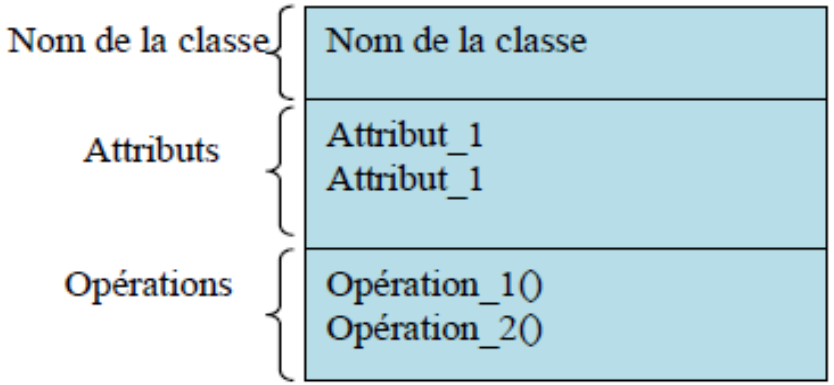
- Un objet est une entité cohérente rassemblant des données et du code travaillant sur ses données.
- Chaque objet possède une *identité*, des *attributs* et des *méthode*:
  - Les attributs: Ce sont des variables stockant des informations sur l'état de l'objet.
  - Les méthodes: décrivent le comportement de l'objet, c'est-à-dire l'ensemble des actions que l'objet est capable de réaliser.
  - L'identité: L'objet possède une identité, qui permet de le distinguer des autres objets, indépendamment de son état: (Deux objets demeurent distincts même si leurs attributs contiennent les mêmes valeurs)

Objet = identité + état (attributs) + comportement (méthodes)

3. Concepts de base de l'orienté objet

B. La classe

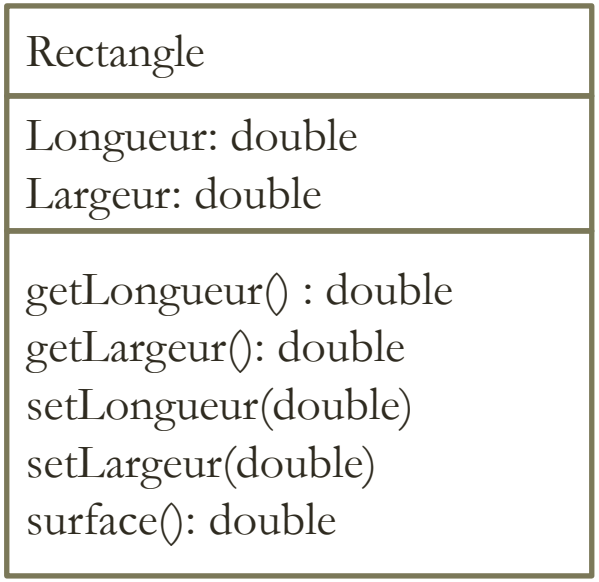
- Un ensemble d'objets partageant les mêmes caractéristiques (attributs et méthodes), sont regroupés dans une **classe**.
- **Une classe** est la description d'un ensemble d'objets ayant une sémantique et des caractéristiques communes.
- Une classe définit :
  - Des **attributs** avec leurs **noms** et leurs **types** (mais pas leurs valeurs qui sont particulières à chaque objet).
  - Des **méthodes** (**opération**) avec leurs **signatures** et le **code** qui décrit le comportement associé.
- Un **objet** est une **instance** (**occurrence**) d'une **classe**.



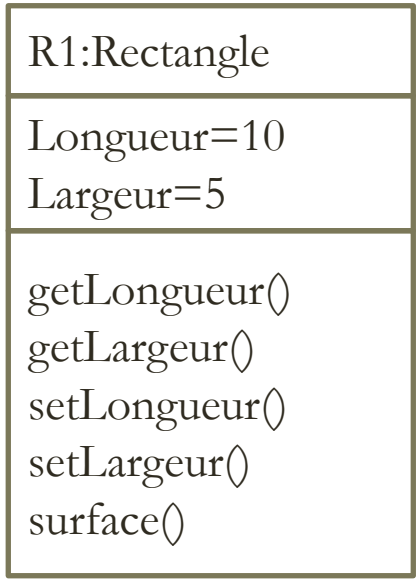
3. Concepts de base de l'orienté objet

B. La classe

Exemple



Classe Rectangle

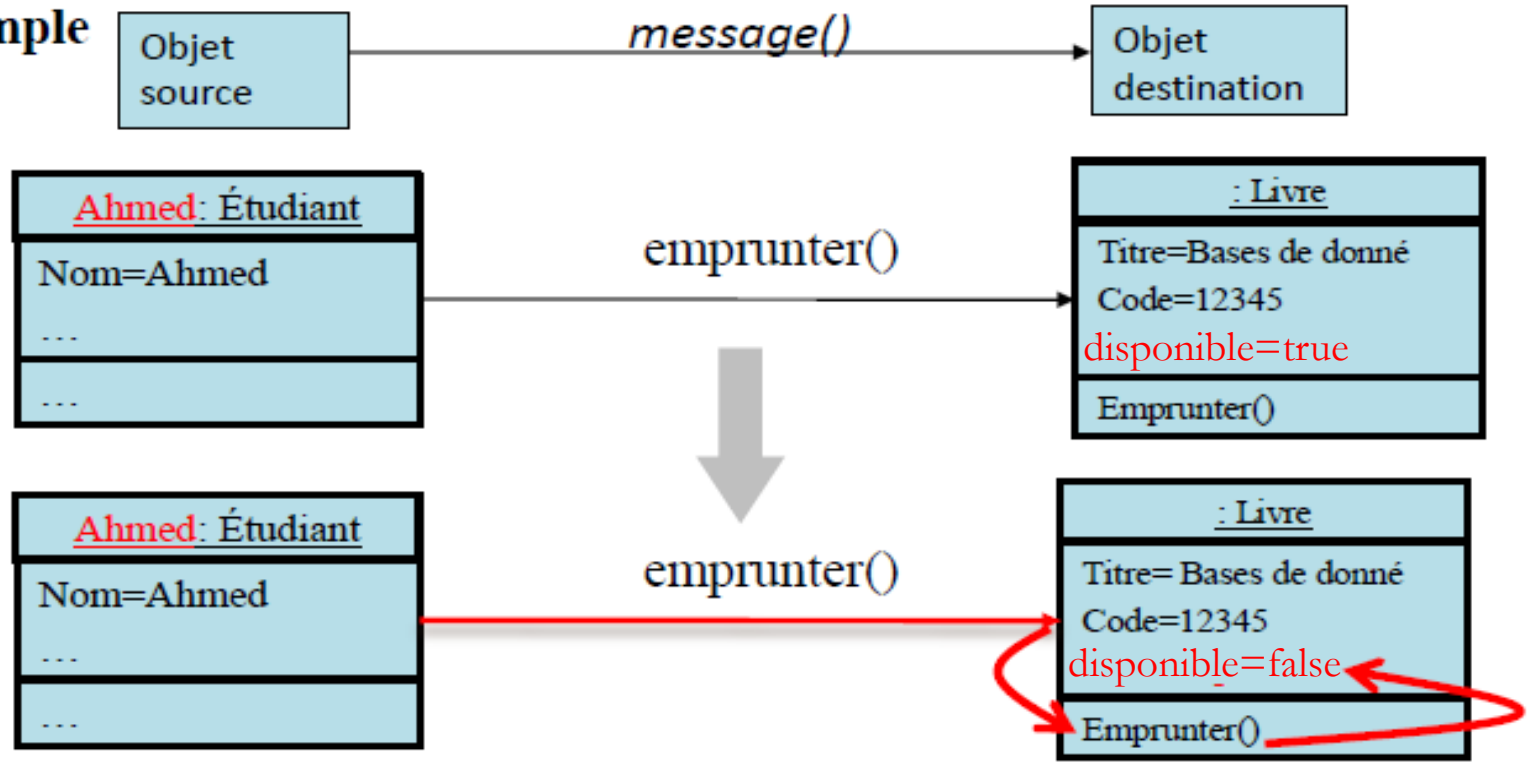


Objet de la classe Rectangle

C. Message

- Les objets communiquent entre eux par envoie de **messages**.
- Un **message** représente **l'appel d'une méthode** de l'objet **destination** par un objet **source**.

Exemple



3. Concepts de base de l'orienté objet

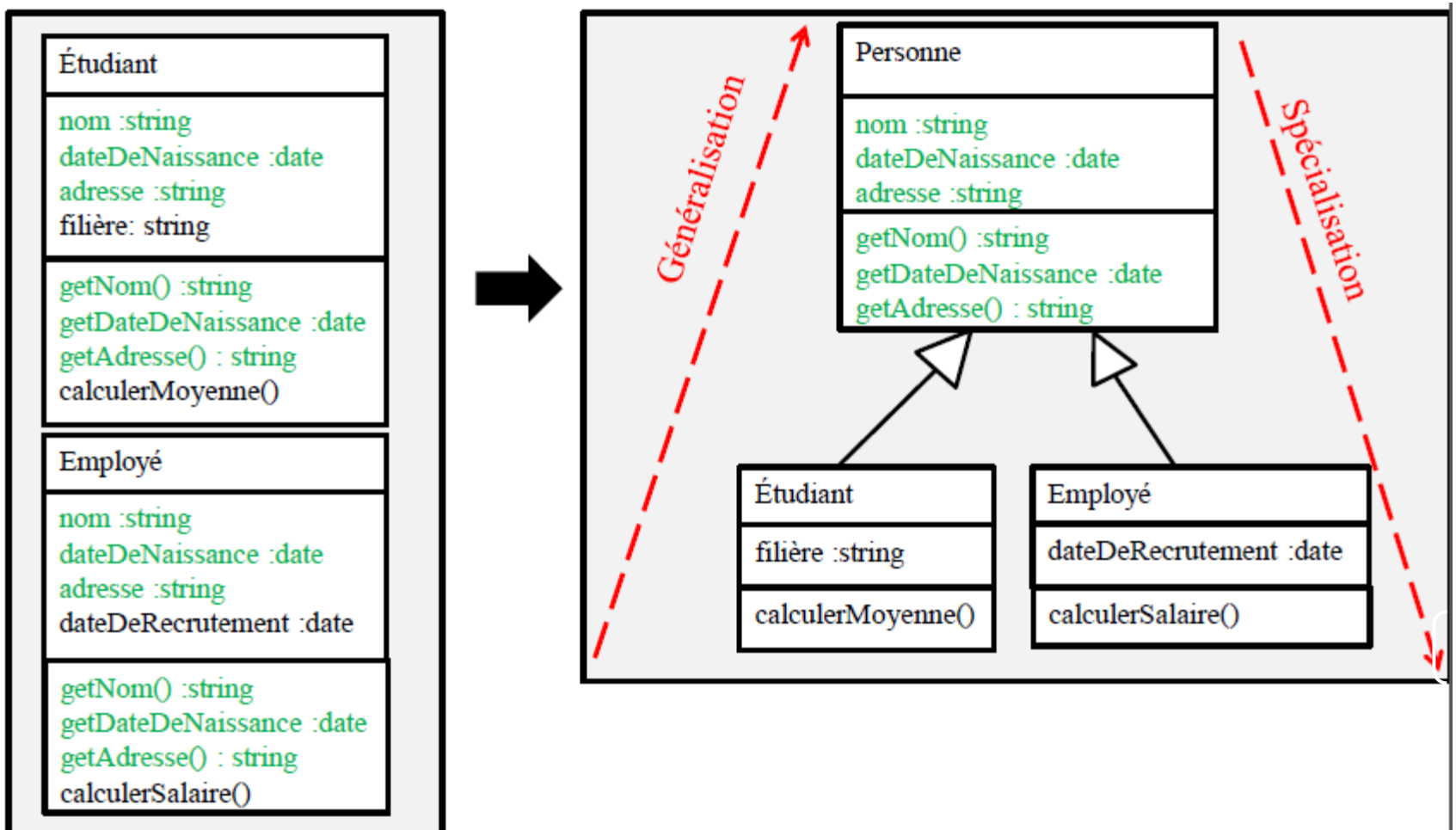
D. L'héritage:

- L'héritage décrit une relation entre une classe générale (classe de base ou classe parent) et une classe spécialisée (classe fille ou sous-classe).
- La classe **spécialisées:**
  - **Hérite** tous les attributs et toutes les méthodes de la **classe générale**
  - Peut ajouter de nouveau attributs et de nouvelles méthodes

3. Concepts de base de l'orienté objet

D. L'héritage

Exemple



3. Concepts de base de l'orienté objet

D. L'héritage:

- L'héritage est mis en œuvre grâce à deux propriétés qui sont:
  - **1.La spécialisation:** Une classe peut être spécialisée en d'autres classes, afin d'y ajouter de nouveaux attributs et de nouvelles méthodes.
  - **2.La généralisation:** Plusieurs classes peuvent être généralisées en une classe, afin de regrouper les caractéristiques communes d'un ensemble de classes.

- **Avantage de l' héritage**
  - L'héritage permet la réutilisation du code et évite la duplication: le code des attributs et des méthodes de la classe générale sera réutilisé dans les classes spécialisées et ne sera pas réimplémenté.

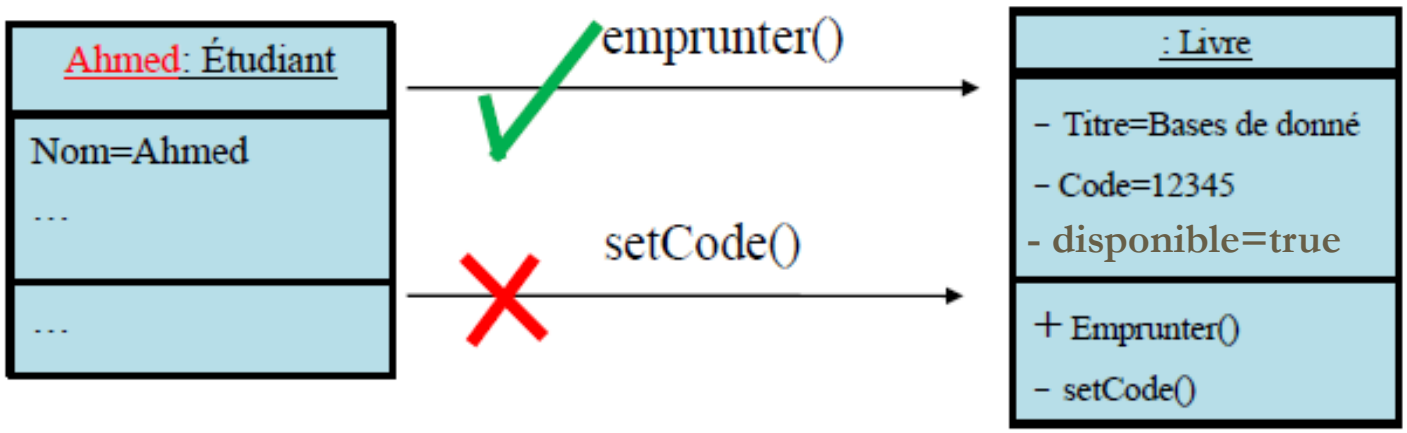
E. L'encapsulation

- L'**encapsulation** consiste à **masquer** une parties des membres d'un objet (attributs et méthodes). c'est-à dire en **empêchant l'accès direct à ces membres par les autres objets.**
- Les attributs masqués sont accessibles par les autres objets à travers les **services (méthodes visibles)**.
- Les **services** d'un objet sont invocables à travers les **messages**.
- La liste des messages auxquels est capable de répondre un objet constitue son **interface** (**sa vue externe**).

- **Avantages:**
  - L'encapsulation **facilite l'évolution** d'une application car elle stabilise l'utilisation des objets: on peut modifier l'implémentation des attributs d'un objet sans modifier son interface.
  - L'encapsulation **garantit l'intégrité** des données, car elle permet d'interdire l'accès direct aux attributs des objets (utilisation d'accesseurs).

E. L'encapsulation

- L'encapsulation permet de définir des niveaux de visibilité des attributs et des méthodes. Il existe trois niveaux de visibilité:
- ***Publique(+):*** Tous les objets peuvent accéder aux attributs ou méthodes d'un objet définis avec le niveau de visibilité publique. Il s'agit du plus bas niveau de protection.
- ***Protégée(#):*** l'accès est réservé aux objets dérivés.
- ***Privée(-):*** l'accès est limité aux méthodes de la même classe. Il s'agit du niveau de protection des données le plus élevé



F. Le polymorphisme

- Le polymorphisme est la possibilité pour un même message de déclencher des traitements différents, suivant les objets auxquels il est adressé.
- Le message déclenche des méthodes qui portent le même nom mais qui sont implémentée de différentes manières.
- **Exemple**

a: Rectangle
Longueur=10 Largeur=5
getLongueur() getLargeur() setLongueur() setLargeur() **calculerSurface()**

b: Cercle
rayon=10
getRayon() setRayon() **calculerSurface()**

Lorsque le message « **calculerSurface()** » est exécuté, il n'a pas le même effet sur l'objet «a» et sur l'objet «b». La méthode «**calculerSurface()**» est polymorphe.

Plan du cours

1. Paradigmes de programmation
2. De la programmation procédurale à la programmation orientée objet
3. Concepts de base de l'orienté objet
4. **Langage Java**

Présentation de Java

- Java est un langage de programmation orienté objet.

- Il a été mis au point en 1991 par la firme *Sun Microsystems* (acheté par *Oracle Corporation* en 2009).

- Java est largement utilisé pour le développement d'**applications d'entreprises** et les **applications mobiles**.

- Classement des langages de programmation (https://www.tiobe.com)

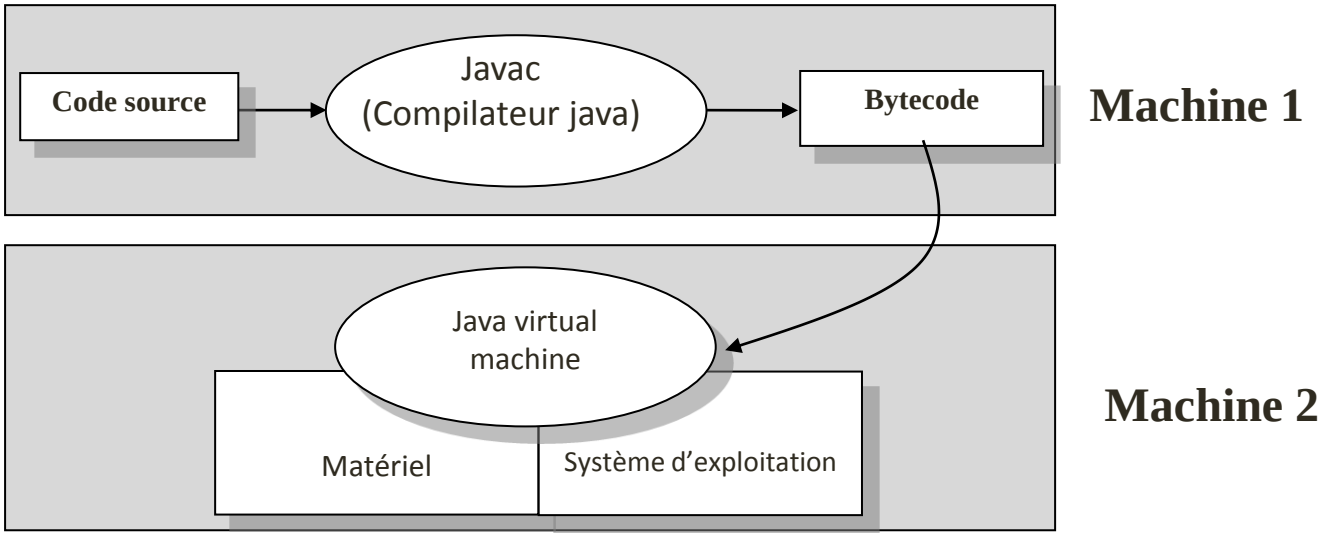
Sep 2018	Sep 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.436%	+4.75%
2	2		C	15.447%	+8.06%
3	5		Python	7.653%	+4.67%
4	3		C++	7.394%	+1.83%
5	8		Visual Basic .NET	5.308%	+3.33%
6	4		C#	3.295%	-1.48%
7	6		PHP	2.775%	+0.57%
8	7		JavaScript	2.131%	+0.11%
9	-		SQL	2.062%	+2.06%
10	18		Objective-C	1.509%	+0.00%
11	12		Delphi/Object Pascal	1.292%	-0.49%
12	10		Ruby	1.291%	-0.64%
13	16		MATLAB	1.276%	-0.35%
14	15		Assembly language	1.232%	-0.41%
15	13		Swift	1.223%	-0.54%
16	17		Go	1.081%	-0.49%
17	9		Perl	1.073%	-0.88%
18	11		R	1.016%	-0.80%
19	19		PL/SQL	0.850%	-0.63%
20	14		Visual Basic	0.682%	-1.07%

- **Quelques chiffres à propos de Java (2011):**
  - 97% des machines d'entreprises ont une JVM installée
  - Java est téléchargé plus d'un milliards de fois chaque année
  - Il y a plus de 9 millions de développeurs Java dans le monde
  - Java est un des langages les plus utilisés dans le monde
  - Tous les lecteurs de Blue-Ray utilisent Java
  - Plus de 3 milliards d'appareils mobiles peuvent mettre en œuvre Java
  - Plus de 1,4 milliards de cartes à puce utilisant Java sont produites chaque année

Caractéristiques de Java

Java possède un certain nombre de caractéristiques qui ont largement contribué à son énorme succès:

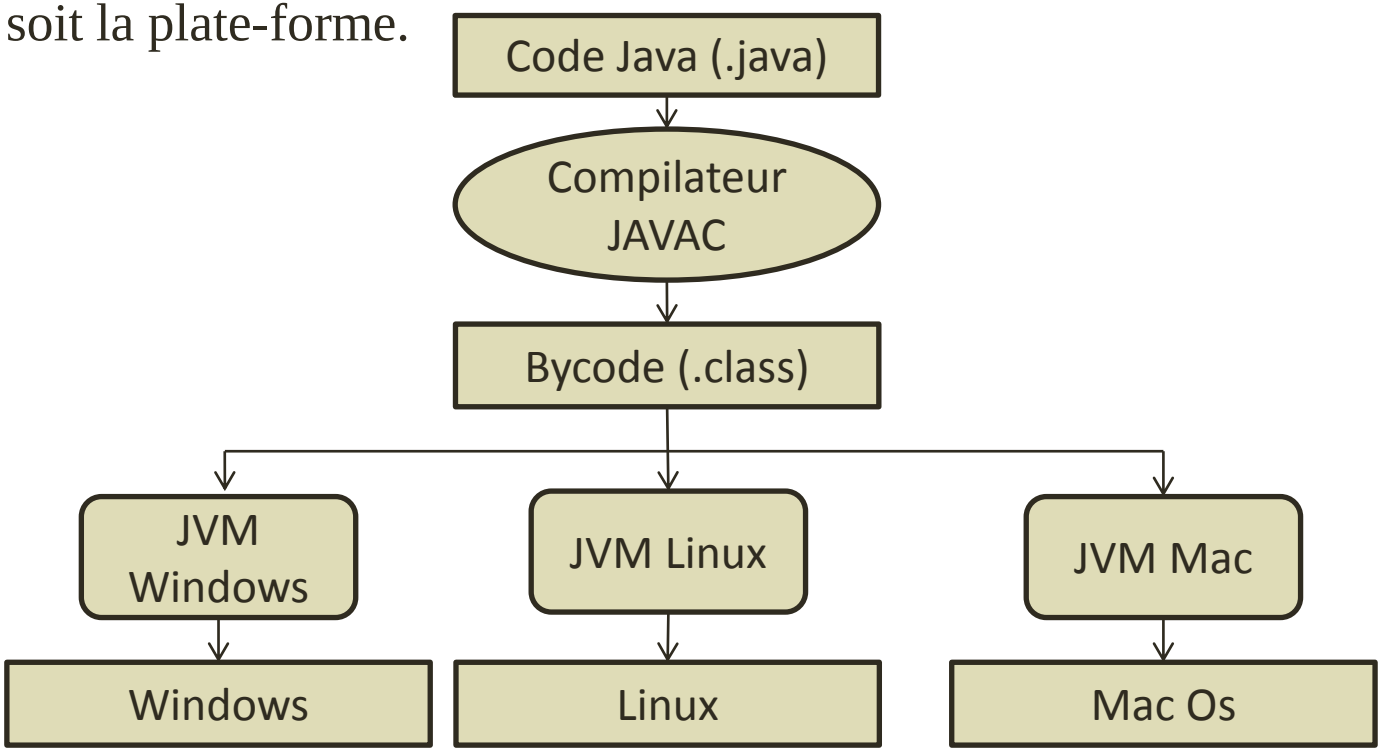
1. **Java est interprété:**
   - Le code source est compilé en *Bytecode*
   - *Le Bytecode* est un code intermédiaire entre le code source et le code machine
   - *Le Bytecode* n'est exécutable sur **aucun processeur**, il est exécuté par un interpréteur Java : la Java Virtual Machine (JVM).



Caractéristiques de Java

2. Java portable

- Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine (JVM).
- La JVM est spécifique à chaque plate-forme (couple SE) et permet aux applications Java compilées en *bytecode* de produire les mêmes résultats quelle que soit la plate-forme.



Caractéristiques de Java

3. **Java est orienté objet:** Comme la plupart des langages récents, Java est orienté objet.
4. **Java est simple:** Java est basé sur le langage C/C++ mais laisse de coté les éléments qui posent des problèmes(pointeurs, structures, gestion de la mémoire, héritage multiple, etc.).
5. **Java est fortement typé:** Toutes les variables sont typées et il n'existe pas de conversion automatique qui risque une perte de données.

   - Exemple: le code suivant est correct en C, mais il est incorrect en java:

```
double a=5.5;
int y=a;
```

6. **Java est multitâche**: Une applications java peut être décomposée en unités d'exécution (**threads**) fonctionnant simultanément.
7. **Java est économe:** Le pseudo code a une taille relativement petite car les bibliothèques de classes requises ne sont liées qu'à l'exécution.
8. **etc.**

Bibliographie

- Le site officiel d'Oracle: https://www.oracle.com/technetwork/java/index.html
- Sigaud, Olivier. **Introduction à la programmation orientée objets en Java**. Ecole Nationale Supérieure de Techniques Avancées (ENSTA).
- DOUDOUX, Jean Michel. **Developpons en JAVA**.
- Esnard Aurélien1. **Cours de Java** .ENSERB informatique.
- Eckel, Bruce. *Thinking in JAVA*. Prentice Hall Professional

Merci pour votre attention