

# UEF4.3.Programmation Orientée Objet

Mme Sadeg et Mme Bousbia

[s\\_sadeg@esi.dz](mailto:s_sadeg@esi.dz), [n\\_bousbia@esi.dz](mailto:n_bousbia@esi.dz)

Ecole nationale Supérieure d'Informatique  
(ESI)

# A propos du cours (1)

- C'est un cours *introductif* à la programmation orientée objet.
- Pré-requis: le cours d'algorithmique
- **Java** sera utilisé comme exemple de langage orienté objet pour illustrer les notions vues en cours
- **BlueJ** et **eclipse** seront utilisés comme environnement de développement (d'autres environnements peuvent être choisis selon vos préférences)

# Pourquoi java ? (1 / 2)

- Java est orienté objet (utilise des classes et des types de base).
- Java est facile à apprendre (Sa syntaxe se rapproche de celle du C/C++)
- Java est portable (indépendant de toute plate-forme )
- Java est fiable(Typage de données fort et pas de pointeurs en apparence)
- Java est sûr (Mécanisme d'exception, vérification du code à l'exécution et des accès réseau et/ou fichiers)
- Java assure la gestion de la mémoire (Garbage Collector)

## Pourquoi java ? (2/2)

Java est un langage très populaire et très utilisé dans le monde de l'entreprise, Il est très souvent classé numéro 1 des langages les plus populaires. (<https://www.tiobe.com/tiobe-index/>)

Jan 2019	Jan 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.904%	+2.69%
2	2		C	13.337%	+2.30%
3	4	▲	Python	8.294%	+3.62%
4	3	▼	C++	8.158%	+2.55%
5	7	▲	Visual Basic .NET	6.459%	+3.20%
6	6		JavaScript	3.302%	-0.16%
7	5	▼	C#	3.284%	-0.47%
8	9	▲	PHP	2.680%	+0.15%
9	-	▲▲	SQL	2.277%	+2.28%
10	16	▲▲	Objective-C	1.781%	-0.08%

## A propos du cours (2)

- Objectifs:
  - Introduire l'approche orientée objet,
- A l'issue du cours,
  - Vous aurez pris conscience de l'importance d'appliquer les principes de l'orienté objet (Abstraction, encapsulation, héritage, polymorphisme..etc)
  - Vous serez capables de concevoir une petite application en utilisant l'approche orientée objet,
  - Vous serez capable de l'implémenter en java,

# A propos du cours (3)

- Volume horaire
  - 23 heures de cours (1 séance de 2h/semaine)
  - 18h de TD
  - 15 heures de TP
- Crédits: 4
- Coefficient: 4
- Evaluation:
  - Contrôle continu
  - Contrôle intermédiaire
  - Examen semestriel
  - Note de TP

# Contenu du cours

- Introduction à la P.O.O
- Objets et classes
- Héritage et polymorphisme
- Exceptions
- Tableaux et collections
- Classes internes et classes anonymes
- Programmation graphique et gestion des évènements
- Flux et fichiers

# Bibliographie

- Fondements de la programmation orientée objet, Mochale Kerboeuf, Ellipses.
- Programmer en Java, Claude Delannoy, Eyrolles
- Head First Java, Second Edition, By Kathy Sierra, Bert Bates, O'Reilly Media.
- Programmer en JAVA, Deitel et Deitel, Les éditions reynald Goulet
- Le Programmeur JAVA 2, Lemay L, Campus Press.
- Au coeur de Java 2 Volume I - Notions fondamentales, Horstmann et Cornell, The Sun Microsystems Press Java Series



Chapitre I

# Introduction à la Programmation Orientée Objet

# Introduction (1)

Pour résoudre un problème, il faut:

- Comprendre et analyser le problème
- Concevoir une solution
- Implémenter la solution (programmer)
- Tester la solution

Ceci doit être fait en considérant

- Maintenance
- Évolution
- Réutilisation

du logiciel

# Introduction (2)

## Quelques bugs informatiques (1)

- **Perte de Mars Climate Orbiter**, le 23 septembre 1999, après 9 mois de voyage. Coût : 120 M\$

Cause : confusion entre pieds et mètres.

- **Mission Vénus** : passage à 5 000 000 de Km de la planète, au lieu de 5 000 Km prévus.

Cause : remplacement d'une virgule par un point (au format US des nombres).

- **Mariner 1** : la première sonde spatiale du programme Mariner, envoyée par la NASA le 27 juillet 1962. La sonde fut détruite peu de temps après son envol. Coût : 80 millions de dollars.

Cause : un trait d'union oublié dans un programme Fortran (« plus coûteux trait d'union de l'histoire », Arthur C. Clarke).

# Introduction (3)

## Quelques bugs informatiques (2)

### **Echec du premier lancement d'Ariane V.**

Au premier lancement de la fusée Ariane V, celle ci a explosé en vol.

La cause : logiciel de plate forme inertielle repris tel quel d'Ariane IV sans nouvelle validation. Ariane V ayant des moteurs plus puissants s'incline plus rapidement que Ariane IV, pour récupérer l'accélération due à la rotation de la Terre. Les capteurs ont bien détecté cette inclinaison d'Ariane V, mais le logiciel l'a jugée non conforme au plan de tir (d'Ariane IV), et a provoqué l'ordre d'auto destruction. En fait tout se passait bien... Coût du programme d'étude d'Ariane V : 38 milliards de Francs, pour 39 secondes de vol après 10 années de travail....

# Paradigmes de programmation

- Un paradigme est un "Modèle théorique de pensée qui oriente la recherche et la réflexion scientifiques" (Larousse).
- Un paradigme de programmation est un outil de pensée qui oriente la façon d'analyser, de concevoir et de coder un programme
- Il existe 4 principaux paradigmes de programmation:
  - Procédural: Fortran ,C, Pascal, ...
  - Fonctionnel: LISP, Scheme,Haskell...
  - Logique:Prolog,GHC...
  - Orienté Objet: Simula, Smalltalk,C++, Java, C#

# Programmation Orientée Objet (POO)

- Définition:

La POO est une méthode d'implémentation dans laquelle les programmes sont organisés sous forme de collections coopératives d'objets, dont chacun représente une instance d'une classe quelconque et dont toutes les classes sont membres d'une hiérarchie de classes unies à travers des relations d'héritage.

# Procédural Vs Orienté objet

## Dans la programmation procédurale

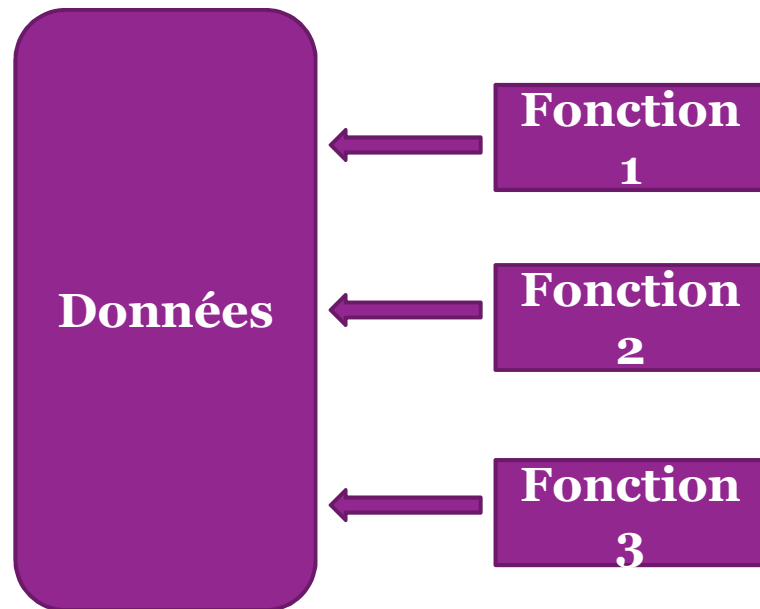
- on apprend à subdiviser un problème en une série d'étapes simples (des fonctions) permettant de résoudre un problème.
- D'abord, on décide de la manière dont on va manipuler les données, ensuite le type de structure de données les plus appropriés pour faciliter cette manipulation.
- Donne des logiciels difficiles à maintenir et à réutiliser

## Dans La programmation orientée objet

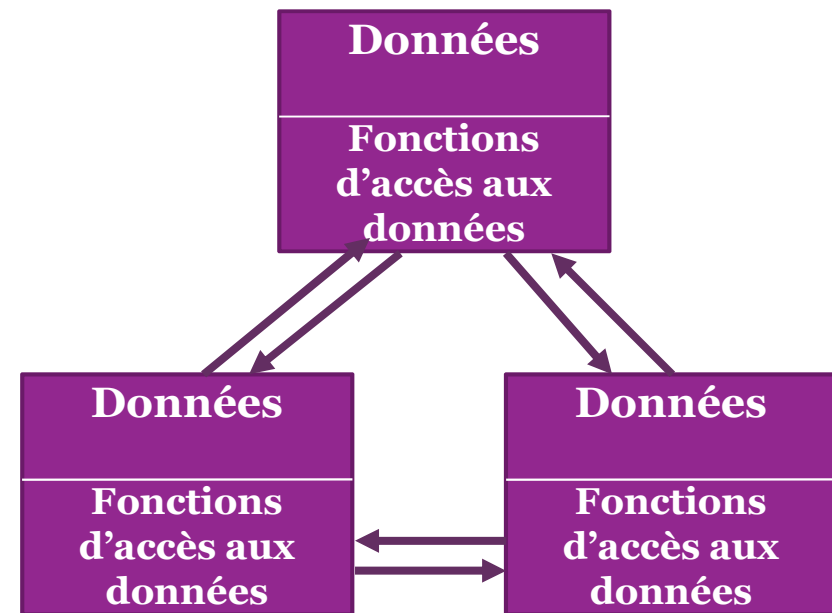
- un programme informatique est considéré comme étant un ensemble d'objets fonctionnant ensemble pour effectuer une tâche.
- On s'intéresse d'abord aux données, avant de déterminer l'algorithme qui sera utilisé pour opérer sur ces données.
- Les objets coopèrent par envois de messages (Appel de méthodes)

# Procédural Vs Orienté Objet

## Programmation procédurale



## Programmation Orientée Objet





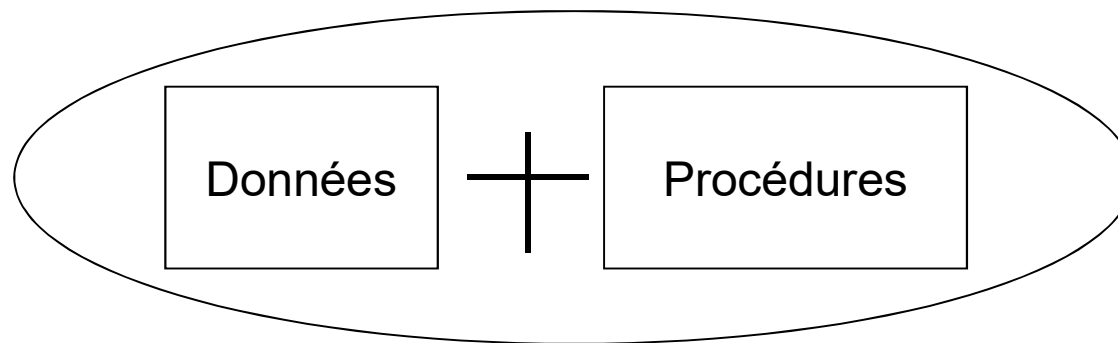
# Qu'est ce qu'un objet ?

- La Programmation orientée objet est basée sur la notion d'objet
- Dans le monde réel, un objet peut être:
  - Une voiture,
  - Une bibliothèque,
  - Un enseignant
  - Un livre
  - ...

# Qu'est ce qu'un objet ?

- Dans le monde réel, les objets ont des attributs et des comportements:

<b>Objet</b>	<b>attributs</b>	<b>comportements</b>
▫ Une voiture,	Accélérateur	Accélère
▫ Une bibliothèque,	abonnés	prête des livres
▫ Un enseignant	Cours	Enseigne
▫ Un ordinateur	Processeur	Calcule



Objet

# Objets et classes

- Ce sont des notions essentielles en P.O.O
- **Objet:** Un *objet* est un élément dans un programme rassemblant des données et du code manipulant ces données. Il est défini par trois caractéristiques : Une identité, un état et un comportement.

**Objet = Identité + état + comportement**

# Objet (1)

## Identité

L'objet possède une identité unique qui le distingue des autres objets indépendamment de son état et de son comportement.

Deux objets peuvent avoir le même état et le même comportement mais ont toujours des identités différentes, et réciproquement, le comportement et l'état d'un objet peuvent changer durant l'exécution sans que cela affecte son identité.

## Objet (2)

### **Etat**

ce sont les données qui caractérisent l'objet, i.e., des informations sur l'objet stockées dans des variables appelées aussi *attributs* ou *champs*.

### **Comportement** (méthodes ou fonctions)

C'est l'ensemble de traitements que peut accomplir un objet.

# Classe

- Une *classe* peut être considérée comme un moule ou un modèle à partir duquel on peut créer des objets.
- Des objets créés à partir de la même classe auront des aspects semblables (mêmes attributs et mêmes méthodes).
- Un objet d'une classe est aussi appelé *instance* de cette classe.

# Objets et classes

Ces objets ont les mêmes valeurs d'attributs, on dit qu'ils sont égaux car ils ont le même *état*. Ce sont néanmoins deux objets distincts car ils ont des *identités* différentes. Les objets d'une même classe ont toujours une identité distincte et généralement un état distinct.

- Exemple:

Classe véhicule

Demarrer ()  
Avancer ()  
Reculer()

Objets  
(instances)

Clio  
Renault  
Noire  
2005

Demarrer ()  
Avancer ()  
Reculer()

Clio  
Renault  
Noire  
2005

Demarrer ()  
Avancer ()  
Reculer()

Golf  
Wolswagen  
Rouge  
1998

Demarrer ()  
Avancer ()  
Reculer()

## Exercice

Proposer une modélisation objet ( attributs et méthodes) pour chacun des objets suivants :

- Une voiture
- Un téléviseur
- Un compte postal
- Une bibliothèque



# Concepts de base de la programmation orientée objet

- La P.O.O. est basée sur les principaux concepts suivants:
  - L'abstraction
  - L'encapsulation
  - L'héritage
  - Le polymorphisme

# Abstraction

**Définition:** C'est le fait d'identifier (selon le point de vue de l'observateur) et de regrouper les caractéristiques et comportements communs et essentiels à des entités afin d'en faciliter la manipulation.

**Exemple:** les caractéristiques d'un étudiant qui intéressent la direction des études de son établissement sont: le niveau d'étude, le matricule, le nom, le prénom, ses notes d'examens...etc, alors que pour le médecin se sont l'âge, le sexe, son dossier médical...etc.

# Encapsulation (1)

**Définition:** Elle consiste à regrouper des données (attributs) et un comportement (méthodes) dans une même classe et à réglementer l'accès aux données de l'extérieur (par d'autres objets).

- Les détails de l'implémentation des méthodes sont masqués aux autres objets
- Cela signifie qu'il n'est pas possible d'agir directement sur les données d'un objet ; il est nécessaire de passer par ses méthodes.
- L'appel de la méthode d'un objet est aussi appelé *envoi de message* à l'objet.

## Encapsulation (2)

- Le principe de l'encapsulation est que, vu de l'extérieur, un objet se caractérise uniquement par ses méthodes visibles appelées *interface*. Les données, elles, restent invisibles et inaccessibles.
- Ce principe permet de concevoir des programmes (logiciels) plus sûrs, plus lisibles et plus faciles à maintenir, car une modification de la structure des données d'un objet ne se répercute pas sur les objets qui communiquent avec lui (invoquent ses méthodes).

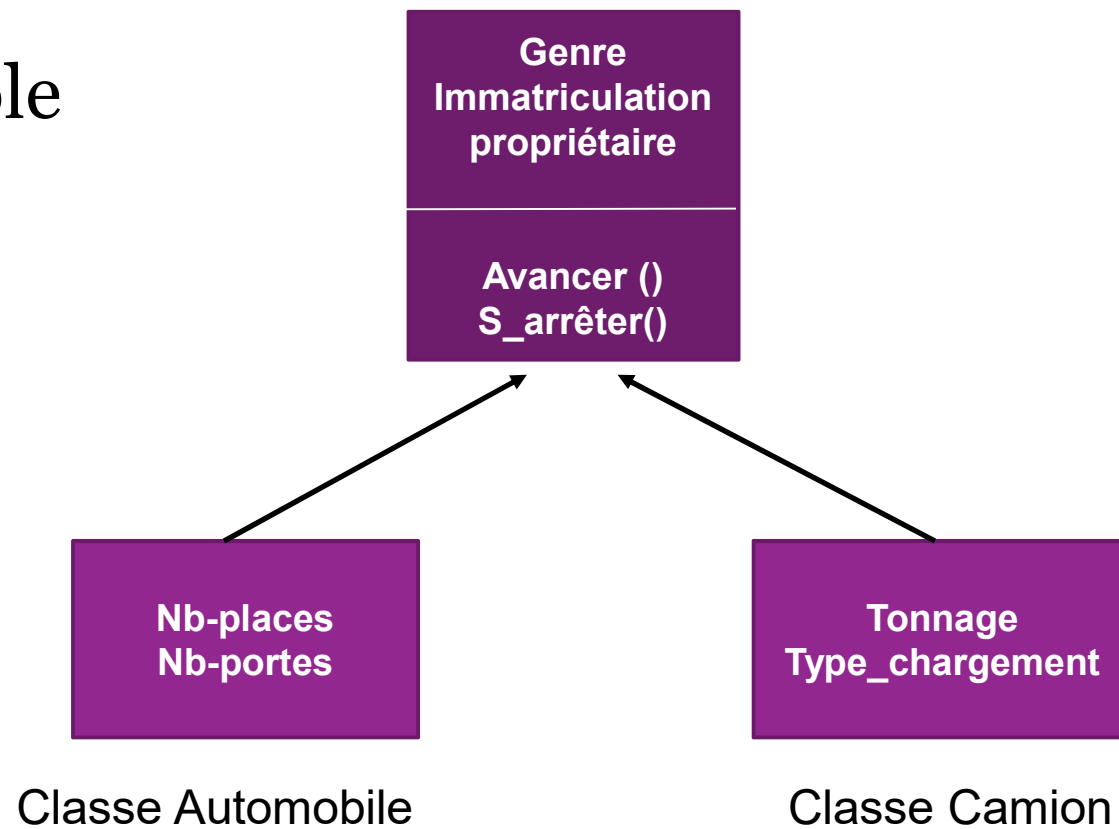
# Héritage

- L'héritage est un autre concept important en P.O.O.
- Il permet de définir une (ou plusieurs) nouvelle classe (appelée *classe dérivée*, *classer fille* ou *sous-classe*) à partir d'une classe existante (appelée *classe de base*, *classe mère* ou *super-classe*), à laquelle on ajoute de nouvelles données et de nouvelles méthodes.
- L'héritage facilite largement la réutilisation de produits existants, ce qui est un avantage important de la P.O.O.

# Héritage

Classe Véhicule

- Exemple



# Polymorphisme

- C'est un concept puissant de la P.O.O. qui vient compléter celui de l'héritage.
- Le terme polymorphisme décrit la caractéristique d'un élément qui peut se présenter sous différentes formes.
- En P.O.O., cela peut s'appliquer aussi bien aux objets qu'aux méthodes.

# Polymorphisme

## **Polymorphisme d'objets**

Il permet une certaine flexibilité dans la manipulation des objets en exploitant la relation d'héritage entre les classes. Pour cela, il applique la règle suivante : Si la classe A est dérivée de la classe B alors un objet de la classe A peut aussi être considéré comme étant un objet de la classe B.



# Polymorphisme

## Polymorphisme de méthodes

Il exprime le fait que

- Des méthodes d'une même classe puissent avoir le même nom et des signatures différentes (ceci est appelé ***surcharge*** ou ***surdéfinition*** de méthodes)

```
public int min(int i1, int i2){}  
public float min(float f1, float f2){}
```

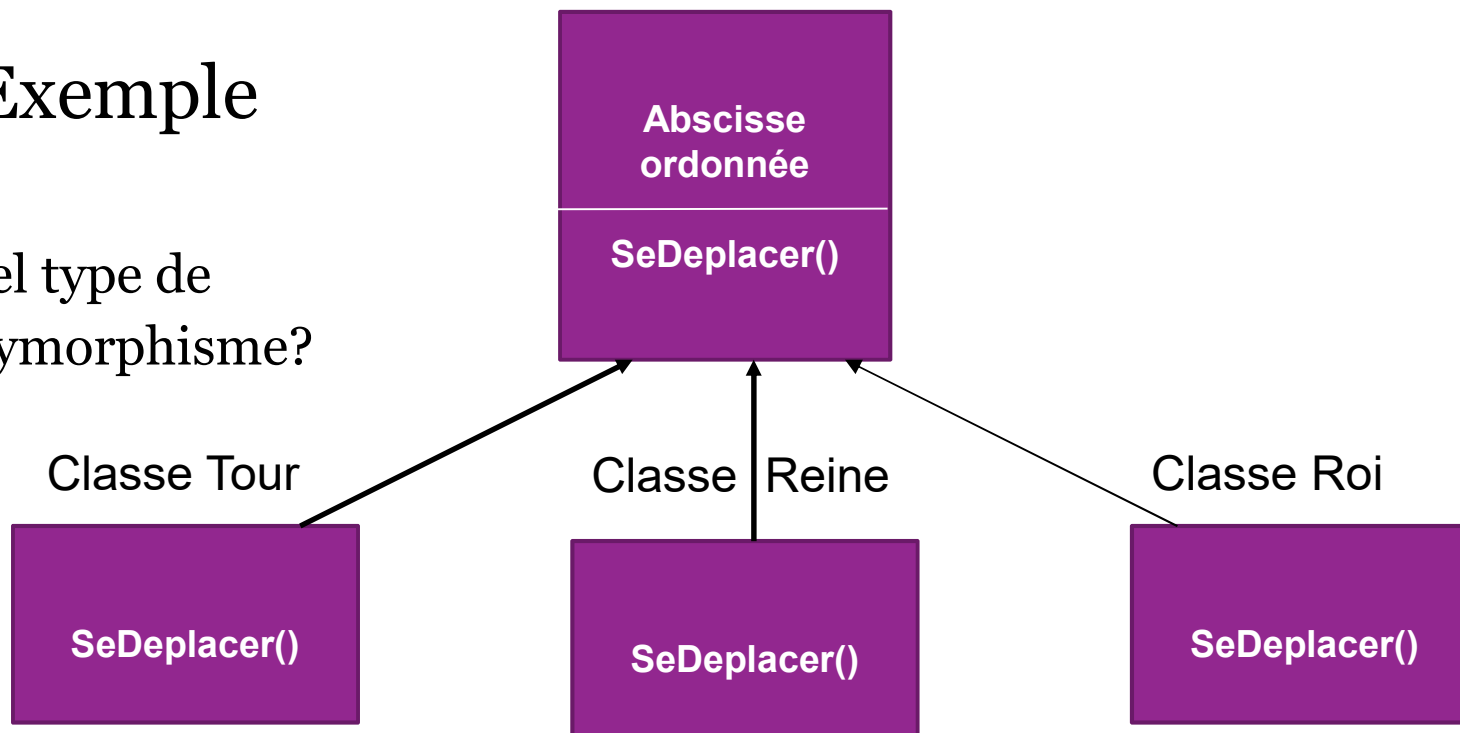
- La même méthode puisse se comporter différemment sur différentes classes de la hiérarchie. Autrement dit, des méthodes peuvent avoir des noms similaires dans une hiérarchie de classes et être différentes. Ceci est appelé ***redéfinition*** ou ***spécialisation*** de méthodes.

# Polymorphisme

Classe PieceJeuEchec

- Exemple

Quel type de polymorphisme?



# Conclusion

- La programmation orientée objet permet de concevoir, maintenir et réutiliser plus facilement les programmes grâce à ses principes d'encapsulation, d'abstraction, d'héritage et de polymorphisme.
- Chacun de ces concepts sera abordé plus en détails lors des prochains cours.

# Outils à installer

- JDK <https://jdk.java.net/11/>
- BlueJ <https://www.bluej.org/>
- Eclipse  
<https://www.eclipse.org/downloads/packages/release/2018-12/r/eclipse-ide-java-developers>