Institut Supérieur d'Informatique et de Multimédia de Gabes

Programmation Orientée Objet
JAVA

Sofiane HACHICHA

sofiane.hachicha@isimg.tn

CPI2 **2025/2026**

Institut Supérieur d'Informatique et de Multimédia de Gabes

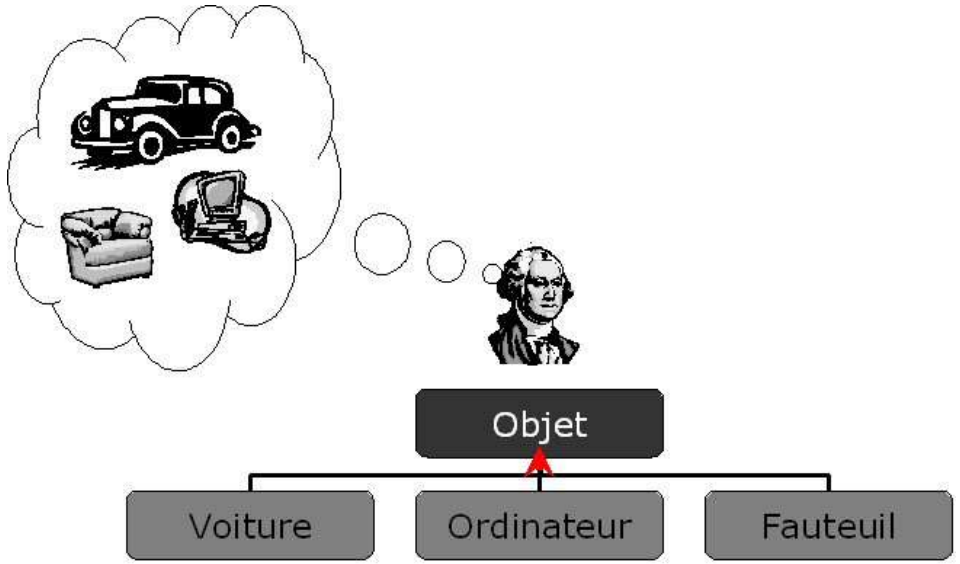
POO : Java

Chapitre 2

Concepts de base de la programmation orientée objet

Sofiane HACHICHA
ISIMG 2025 - 2026

- **L'abstraction dans l'approche objet permet la représentation des entités du monde réel sous forme d'entités informatique de la manière la plus naturelle.**
  - Etablir une association entre le modèle du problème à résoudre et le modèle de la machine :



Sofiane HACHICHA
ISIMG 2025 - 2026

2.2 Les bienfaits de l'abstraction (1/2)

- **L'abstraction est une représentation des éléments du monde réel « objets réels » dans l'espace problème (la machine) en tant qu'« objets informatiques».**
  - décrire le problème avec les termes du problème plutôt qu'avec les termes de la machine.
    - un programme traitant des images doit manipuler des structures de données représentant des images, et non leur traduction sous forme de suite de 0 et de 1.
    - un programme de gestion de personnel doit représenter des personnes avec toutes les informations pertinentes de type texte, date, nombres ou autre.
- **L'idée est d'adapter le programme à l'esprit du problème réel en ajoutant de nouveaux types « objets ».**
  - Quand on lit le code décrivant la solution, on lit aussi quelque chose qui décrit le problème.

2.2 Les bienfaits de l'abstraction (2/2)

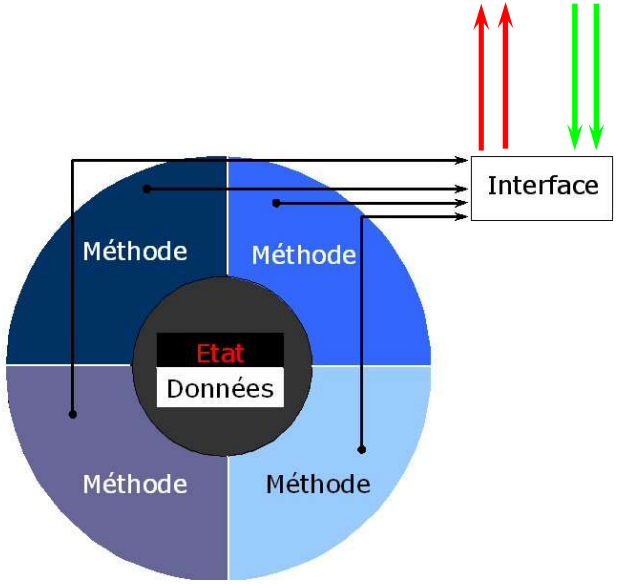
- **La complexité des problèmes et la capacité à les résoudre sont directement proportionnelles au type et à la qualité de nos capacités d'abstraction.**
  - Un Objet (un micro-ordinateur par exemple) :
    - **plusieurs niveaux d 'abstraction :**
      - de point de vue concepteur, un micro-ordinateur est un objet formé d'un ensemble d'éléments physiques appelés matériels (hardware).
      - de point de vue informaticien, un micro-ordinateur est un objet résultant d'un assemblage hardware et d'un ensemble de programme appelé logiciels (software).

      - de point de vue utilisateur, un micro-ordinateur est une boite noire qui offre un certain nombre de fonctions ou de services qui permettent d'interagir avec elle.

- **Un objet est semblable à une variable améliorée :**
  - elle stocke des données qui décrivent son « état »;
  - mais qui possède aussi un ensemble de fonctions ou méthodes « comportement », pour répondre aux requêtes des utilisateurs.
- **L'ensemble des services (méthodes) proposés par un objet est appelé l'interface de cet objet.**
- **Un objet est encapsulé par son interface :**
  - la seule manière d'interagir (demander un service) avec cet objet est d'invoquer une des méthodes de son interface.

Sofiane HACHICHA
ISIMG 2025 - 2026

- **Exemple 1 :**
  - Un poste de Télévision est une boite ayant : un écran, des haut-parleurs et une télécommande.
    - Pour changer de chaîne il suffit de demander à cette boite de le faire pour nous, en appuyant simplement sur le bouton correspondant. Peu importe ce qui se passe réellement dans le poste.

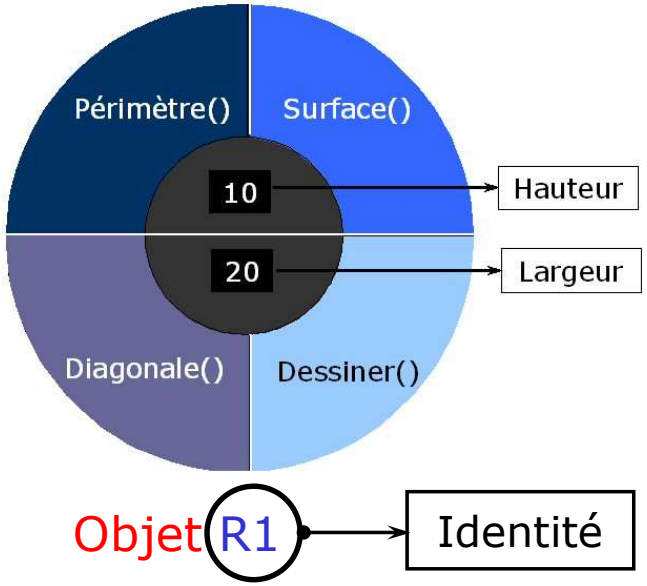


Principe d'Objet

Sofiane HACHICHA
ISIMG 2025 - 2026

Exemple 2 : Objet Rectangle R1

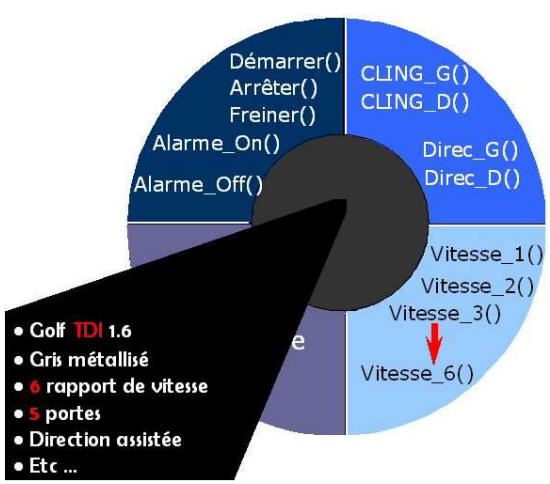
- R1 est un objet Rectangle de hauteur 10 de largeur 20, ayant une interface qui permet de :
  - calculer le périmètre du rectangle,
  - calculer sa surface,
  - calculer la diagonale
  - et un dernier service pour dessiner le rectangle.

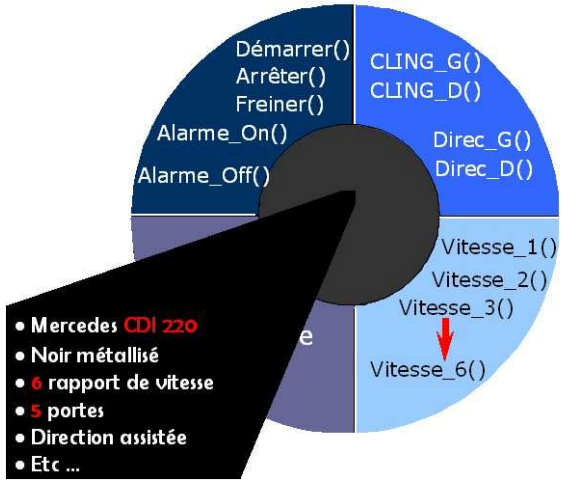


- Pour utiliser l'un de ces services, il suffit d'invoquer la méthode correspondante.

POO : Java

- **Un objet représente informatiquement une entité précise du monde à modéliser, exemple : ma_voiture.**





- **Il existe généralement de nombreux objets représentant le même concept « voiture », exemple : ma_voiture et présidence_voiture**
  - Il est intéressant de définir le concept (classe voiture), puis de créer autant de représentant que voulu basé sur ce concept.

Sofiane HACHICHA
ISIMG 2025 - 2026

- **Une classe est un moule pour fabriquer des objets de même structure et de même comportement.**
  - En programmation, une classe apparaît donc comme un nouveau type construit par l'utilisateur.
    - **Le nom d'une classe peut être utilisé en Java comme le type d'un attribut ou d'une variable ou aussi comme le type de retour d'une fonction (méthode).**
  - Le processus de création d'un objet à partir d'une classe est appelé en jargon objet instanciation d'un objet ou création d'instance d'une classe .
    - **Exemple :**

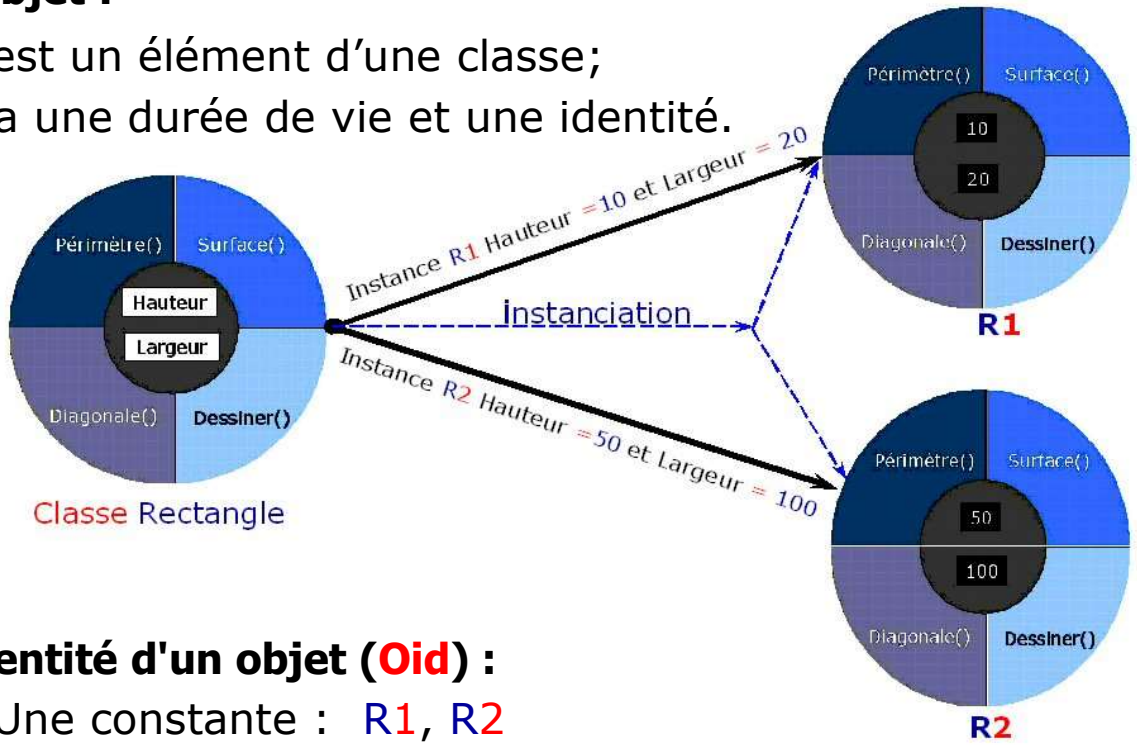
```
String s1 = "Hello" ;
```

      - String est le nom de la classe, la variable s1 est un objet instancié ou instance de la classe String.

Sofiane HACHICHA
ISIMG 2025 - 2026

POO : Java

- **Un objet :**
  - est un élément d'une classe;
  - a une durée de vie et une identité.



- **L'identité d'un objet (Oid) :**
  - Une constante : R1, R2
    - Unique et indépendante de la valeur des propriétés de l'objet;
    - invisible à l'utilisateur.

2.4 Attribut de classe, attribut d'instance

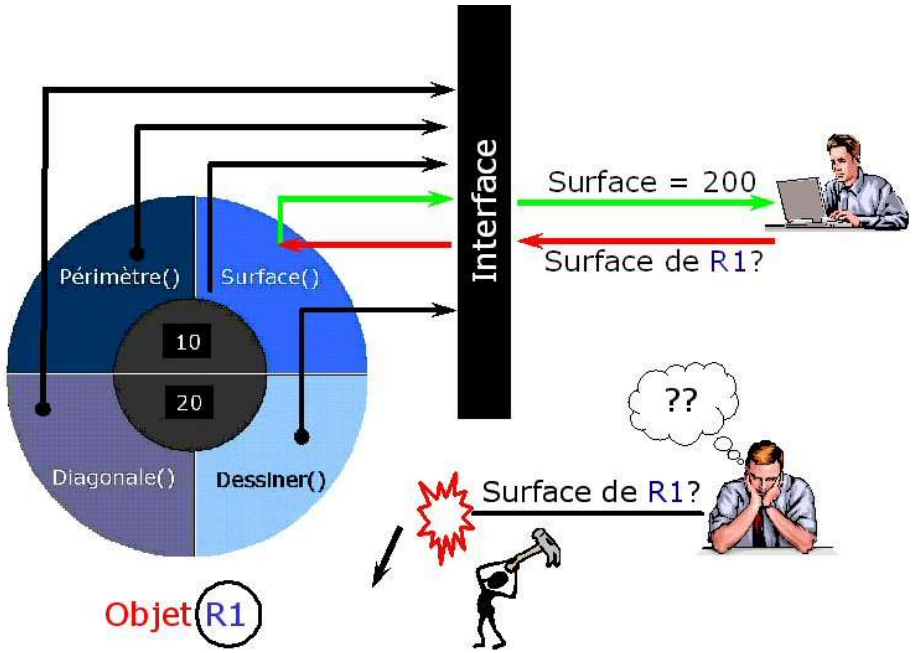
- **Les données membres d'une classe sont appelées dans le jargon informatique attributs.**
- **Java distingue deux types d'attributs :**
  - Attribut d'instance : chaque instance (ou objet) possède sa propre valeur pour cet attribut, qui peut varier d'une instance à une autre.
    - Exemple : un attribut « couleur » dans la classe « Voiture » peut avoir une valeur différente selon chaque voiture créée.
  - Attribut de classe (appelé aussi attribut statique) : sa valeur est partagée par toutes les instances de la classe. En **Java**, les attributs de classe sont déclarés avec le mot-clé **static**.
    - Exemple : un attribut « nbreDeRoues » dans la classe « Voiture » est généralement fixé à 4 pour toutes les voitures.

2.5 Encapsulation (1/2)

- **L'encapsulation d'un objet par son interface permet de masquer son contenu :**
  - montrer uniquement ce qui est nécessaire pour son utilisation :
    - Les données sont généralement considérées comme données privées. Elles ne sont pas accessibles directement.
    - Les méthodes constituent l'interface d'interaction avec un objet d'une classe. Elles sont donc accessibles (publiques).
  - ➡ Un objet n'est accessible qu'à travers l'interface de sa classe ;
- **Java propose plusieurs niveaux de visibilité (modificateur d'accès) utilisable sur les données d'une classe pour assurer le principe de l'encapsulation.**

POO : Java

- **Pour respecter ce principe d'encapsulation, on ne doit interagir avec un objet que par l'invocation d'une de ces méthodes de son interface.**



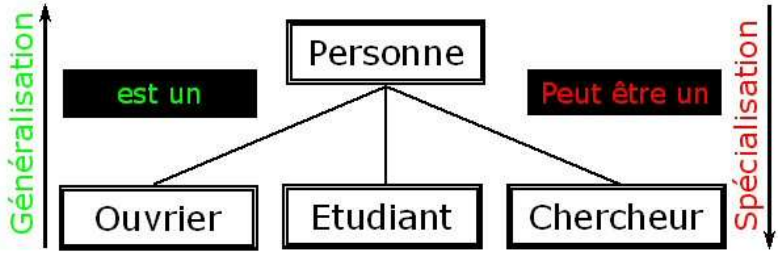
- **On dit à l'objet ce qu'il doit faire, sans se préoccuper de comment il doit le faire.**

2.6 Exemple d'un programme Java

- **Écrire un programme Java qui comprend deux classes :**
  - Classe **Rectangle** :
    - Cette classe représente un rectangle caractérisé par sa longueur et sa largeur.
    - Elle encapsule les données en utilisant des attributs privés pour la longueur et la largeur.
    - Elle fournit des méthodes publiques pour obtenir la longueur et la largeur, ainsi que pour calculer le périmètre et la surface du rectangle.
  - Classe **TestRectangle** :
    - Cette classe sert de programme principal.
    - Elle crée deux objets de type Rectangle, en spécifiant leurs dimensions respectives.
    - Elle affiche ensuite les caractéristiques de chaque rectangle, notamment la longueur, la largeur, le périmètre et la surface.

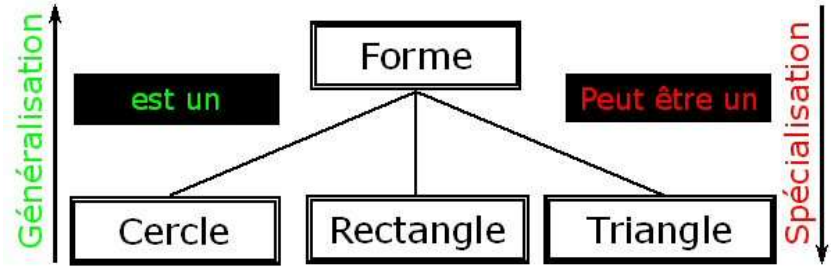
- **L'héritage est l'un des grands intérêts des langages orienté objet :**
  - pouvoir définir des dépendances entres classes en factorisant les propriétés communes à plusieurs classes :
    - ordonner hiérarchiquement les classes d'une application,
    - réaliser des programmes parfaitement modulaires,
    - disposer de modules réutilisables.
    - Exemple :
      - Une classe Personne étant définie.
      - On peut créer une classe Etudiant qui hérite de la classe Personne en précisant simplement qu'un étudiant est une personne possédant un numéro d'inscription.
      - Une instance de la classe Etudiant contient à la fois les informations héritées de la classe Personne et celles spécifiques à la classe Etudiant.

- Les deux classes **Personne** et **Forme** sont appelées classe de base



- Les classes **Ouvrier**, **Etudiant** et **Chercheur** héritent toutes les trois de la classe **Personne**.

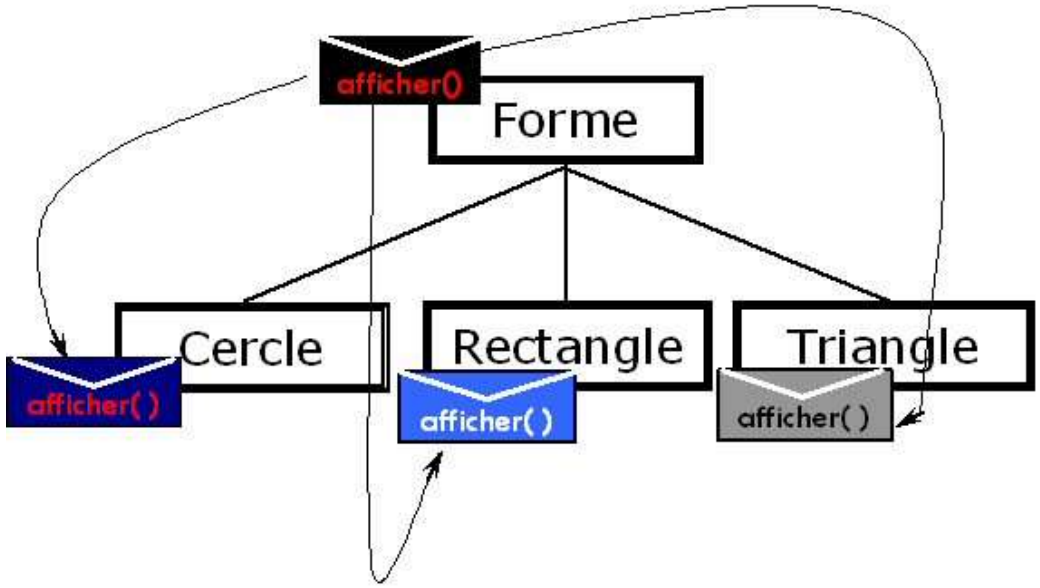
- Les classes **Cercle**, **Rectangle** et **Triangle** sont appelées des classes **dérivées** de la classe **Forme**.



Sofiane HACHICHA
ISIMG 2025 - 2026

POO : Java

- **Le polymorphisme est la possibilité pour deux classes séparées, mais dépendantes l'une de l'autre, de recevoir le même message mais d'agir dessus de différentes façons.**
  - En d'autres termes, c'est la faculté d'une méthode héritée à pouvoir s'appliquer à des classes dérivées.



2.9 Pourquoi utiliser l'approche Objet ? (1/2)

- **La motivation essentielle de cette approche est d'augmenter les possibilités de réutilisation :**
  - L'encapsulation des données et du code dans une même entité permet de garantir la cohérence des objets.
    - Cette cohérence est indispensable pour envisager de réutiliser un objet dans un autre contexte.
- **La notion d'encapsulation par une interface permet de normaliser le fonctionnement des objets :**
  - Il est possible de changer le fonctionnement interne d'un objet particulier, sans modifier la manière dont il est utilisé dans le reste du programme.

2.9 Pourquoi utiliser l'approche Objet ? (2/2)

- **La notion d'héritage permet la réutilisation efficace du code déjà défini lors de la conception d'objets pour en créer de nouveaux.**

- **La notion de polymorphisme, permet de manipuler de manière identique et naturelle des objets ayant des comportements différents :**

  - des comportements qui ne sont pas obligatoirement connus au moment où l'on définit ces manipulations :

    - Vouloir afficher un rectangle, un cercle ou un triangle, on est toujours dans le même espace de problème, qui est afficher une forme géométrique. C'est la manière d'afficher chaque forme qui est différente.

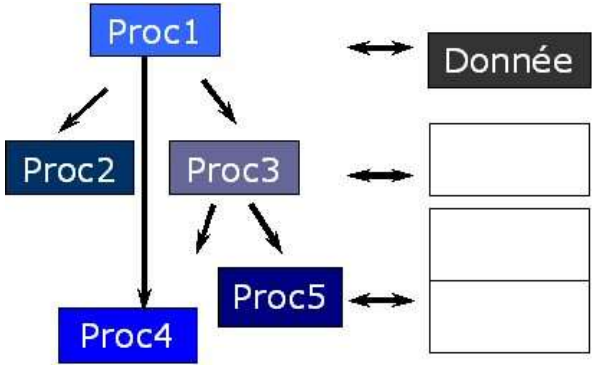
2.10 Apports de l'approche Objet

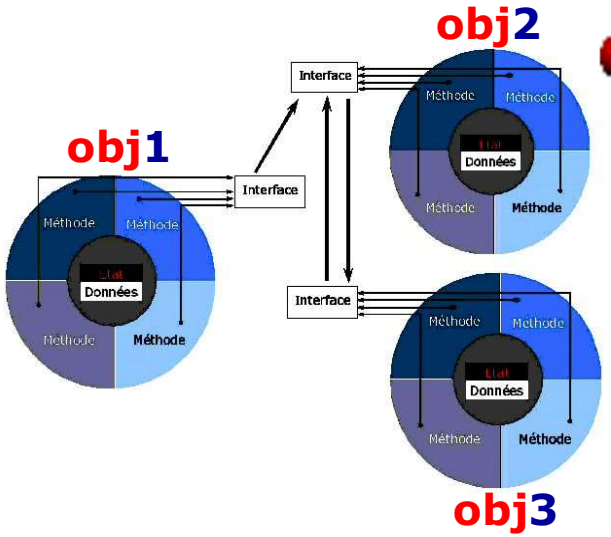
- **L'amélioration de la qualité et de la productivité des logiciels.**
- **L'abstraction permet de modéliser des concepts du monde réel de manière plus proche de la réalité.**
- **La modularité qui permet de :**
  - décomposer un problème en modules élémentaires faciles à écrire, à maintenir et à étendre,
  - créer des modules complexes par composition de modules simples.
- **L'extensibilité :**
  - les logiciels objets : ensemble de modules incrémentalement extensibles;
- **L'uniformité des différents niveaux de représentation des données.**
- **...**

2.11 Approche Procédurale vs Approche Objet

POO : Java

- **Approche procédurale :**
  - Opérations et données sont séparées.
  - Les actions sont décomposées de manière hiérarchique.





- **Approche Objet :**
  - Pas de séparation des données et des actions.
  - Chaque objet peut invoquer une méthode d'un autre objet qui coopère en répondant à cette demande.

Sofiane HACHICHA
ISIMG 2025 - 2026