

## **Aujourd'hui:**

Introduction à la programmation informatique et aux différents langages de programmation.

- L'univers de la programmation.
- Les langages de programmation.
- Le concept Orienté objet ( OO).
- Les Propriétés de base de l'OO.
- Des TDs applicatifs.

# L'univers de la programmation (1/5)

## Qu'est ce qu'un programme:

- C'est la description d'une méthode de résolution d'un problème donné,
- Cette description est donnée par une suite d'instructions d'un langage de programmation.
- Ces instructions ont pour but de traiter et transformer les données du problème à résoudre, jusqu'à aboutir à une solution.

**En résumé: Un programme informatique est une séquence d'instructions qui spécifie étape par étape les opérations à effectuer pour obtenir un résultat. Il est exprimé sous une forme qui permet de l'utiliser avec une machine comme un ordinateur pour exécuter les instructions. ...**

## L'univers de la programmation (2/5)

### Étapes clés pour l'élaboration d'un programme:

1. **Analyse:** détermination de le problème à résoudre, les données de départ et le résultat à obtenir.
2. **Conception:** méthode de résolution. Cette méthode est ensuite énoncée sous forme de suite de pas à accomplir pour aboutir aux solutions : c'est l'algorithme de resolution du problème.
3. **Développement:** traduction de l'algorithme en langage de programmation, appelé habituellement par le codage.

## L'univers de la programmation (3/5)

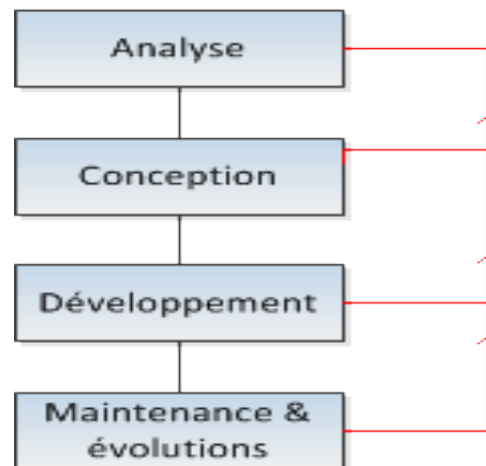
### 4. Mise au point:

- **Compilation** : Traduction du code source du programme vers le langage natif de la machine (ou parfois vers du code intermédiaire). Le résultat: un nouveau fichier écrit en langage machine (appelé code objet).
- **Tests** : exécution du code objet avec divers cas typiques des entrées, ou jeu de tests. C'est le moment où la plupart des erreurs apparaissent.

## L'univers de la programmation (4/5)

- **Correction d'erreurs**: on modifie le code de manière à corriger les erreurs au fil des tests, et l'on recommence la compilation, exécution et tests, etc
- **Maintenance** : la correction des erreurs apparues après la mise en service, la modification du programme pour l'adapter à de nouvelles spécifications du problème.

En résumé:



# L'univers de la programmation (5/5)

## Traducteurs de programmes

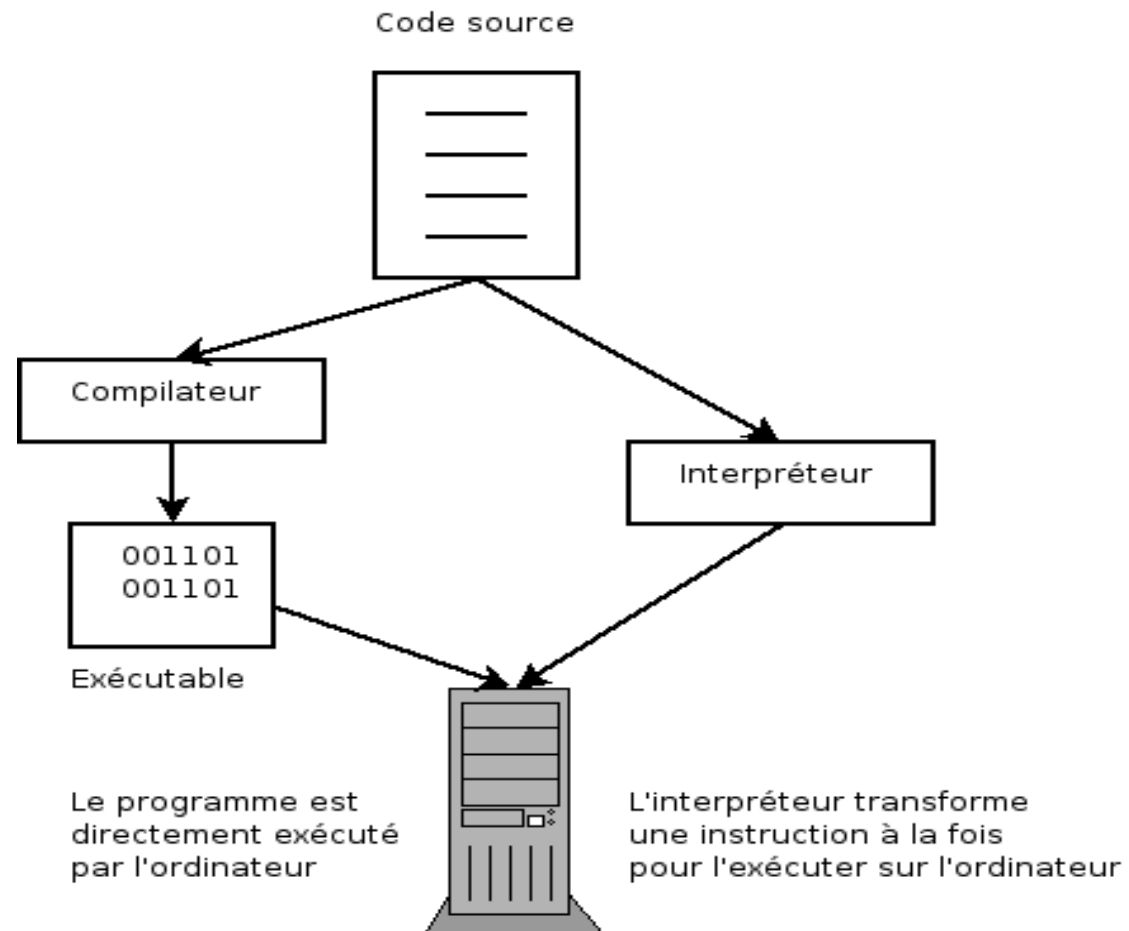
1- Comment une machine comprend un programme:

- Grâce à des traducteurs de programmes.
- Logiciel de traduction des programmes écrits dans ce langage.
- Principalement, il traduit chaque instruction de haut niveau en plusieurs instructions machine équivalentes.

2- Deux sortes de traducteurs: les interprètes et les compilateurs.

- Compilateur réalise trois traitements: Analyse syntaxique, Typage et traduction ( code source ---> code code cible).
- Interprète: traduction en code puis exécution.

## L'univers de la programmation (6/6)



# Introduction à la programmation informatique et aux différents langages de programmation

L'univers de la programmation: un programme, ses étapes clés, types de traducteurs

Les langages de programmation.

Le concept Orienté objet ( OO).

Les Propriétés de base de l'OO.

Des exemples applicatives.



## Les langages de programmation (1/4)

- **Langages de bas niveau:** Ce sont les différents ensembles d'instructions propres à chaque machine (SPARC/Sun, Intel/PC, etc). Appelés également langages cibles ou natifs. Ils sont codés en binaire et directement exécutables par chaque machine.
- **Langages de haut niveau:** Ils fournissent des nombreuses constructions sophistiquées qui facilitent l'écriture des programmes (Ex : C, Ada, Pascal, Cobol, Java, OCaml, Python). Ils sont compréhensibles par les humains, mais pas directement exécutables par les machines. Un programme écrit en langage de haut niveau devra être traduit en langage machine avant son exécution.

# Programmation Procédurale VS Programmation Orientée-Objet (POO) (2/4)

## Programmation procédurale:

- Centrée sur les procédures (opérations): décomposition des fonctionnalités d'un programme en procédures qui vont s'exécuter séquentiellement.
- Couplage procédures/données.
- Ressemble peu à notre schéma de penser.
- Tend à générer du code "Spaghetti"
  - La maintenance et l'ajout de nouvelles fonctionnalités demandent de modifier ou d'insérer des séquences dans ce qui existe déjà
  - Peu devenir complexe très rapidement
  - Modularité et abstraction absente (ou presque)
  - Réutilisation ardue => "Couper-coller" = DANGER!
  - Travail d'équipe difficile (peu modulaire), donc la qualité du code en souffre

# Programmation Procédurale VS

## Programmation Orientée-Objet (POO) (3/4)

### **La programmation Orientée objet:**

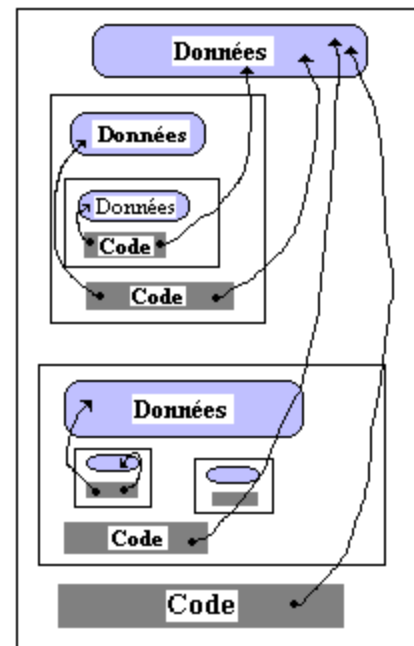
- Centrée sur les données
- Tout tourne autour des "objets" qui sont des petits ensembles de données représentant leurs propriétés

Ex.: Un chat a 4 pattes, une queue et fait partie de la famille des félins

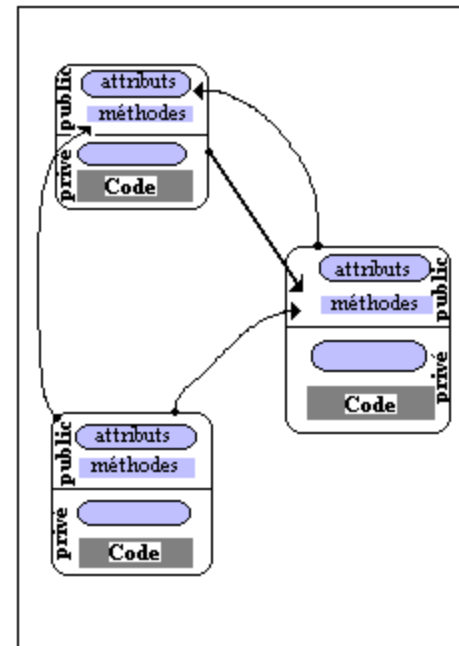
Ex.: Une automobile 4 portes, familiale, moteur 6 cylindres, automatique, etc.

# Programmation Procédurale VS

## Programmation Orientée-Objet (POO) (4/4)



Programmation procédurale



Programmation objet

## Aujourd'hui:

Introduction à la programmation informatique et aux différents langages de programmation.

- L'univers de la programmation.
- Les langages de programmation: langage bas/haut niveau, programmation procédurale vs POO
- Le concept Orienté objet ( OO).
- Les Propriétés de base de l'OO.
- Des exemples applicatives.

## Le concept Orienté objet (OO) (1/4)

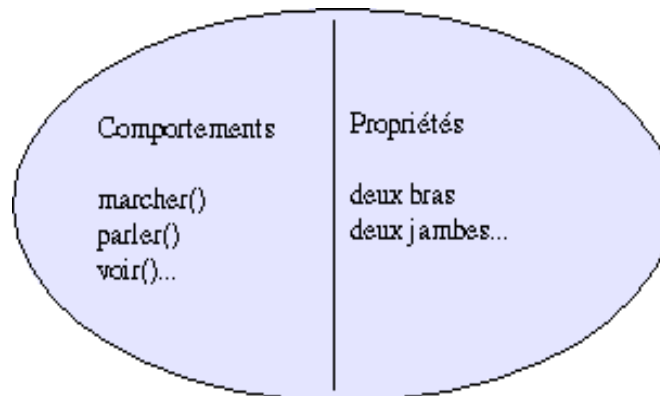
- Le concept orienté objet veut une modélisation du monde réel via des objets: l'élément de décomposition unique est « l'OBJET ».
- L'objet se veut alors une représentation (abstraite) d'une « chose » (concrète) du monde réel.
- Un objet regroupe à la fois les données qui le caractérise et les actions qu'on peut effectuer sur lui.
- Un objet regroupe une partie statique (un ensemble de données) et une partie dynamique (un ensemble de procédures manipulant ces données).

Ex.: Un chat a 4 pattes, une queue et fait partie de la famille des félins

Ex.: Une automobile 4 portes, familiale, moteur 6 cylindres, automatique, etc.

## Le concept Orienté objet (OO) (2/4)

**Notion de classe:** Une classe encapsule, c'est-à-dire regroupe des propriétés et des comportements. Par exemple, la classe Humain *définit* des propriétés ( deux bras, deux jambes...) et des comportements ( marcher, parler, voir...).

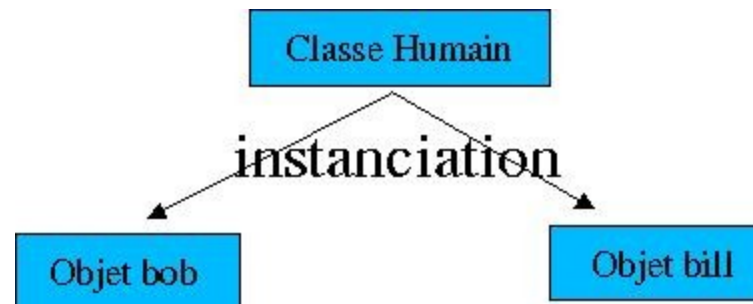


En OO, les comportements sont appelés *méthodes* et les propriétés *variables d'instance*.

## Le concept Orienté objet (OO) (3/4)

**Notion d'objet:** Un objet est une instance de classe, c'est-à-dire un exemplaire utilisable crée à partir de cette classe et en valorisant certaines propriétés.

Exemple, *bob* ou *bill* sont des instances de la classe Humain, c'est-à-dire des humains ayant des propriétés spécifiques ( *bob* est un humain aux yeux noirs et *bill* un humain aux yeux marrons ).





## Le concept Orienté objet (OO) (4/4)

Le concept de **classe** est **abstrait** alors que le concept d'**objet** est fondamentalement **concret**: il est concevable de croiser des personnes physiques, c'est-à-dire des instances d'*Humain*.

**Ex-1**: Application du concept Classe et Objet à des formes géométriques 'Rectangle' et 'un point'.

**Ex-2**: Application du concept Classe et Objet à des véhicules.