

Programmation Orientée Objet

Programmation JAVA

FISE 3A ICy / FISA 3A Informatique

CM1 : Paradigme de POO et qualité d'un logiciel

Mohamed Amine BOUDIA

UPHF, CNRS, UMR 8201 - LAMIH, F-59313 Valenciennes, France

Email : mohamedamine.boudia@uphf.fr



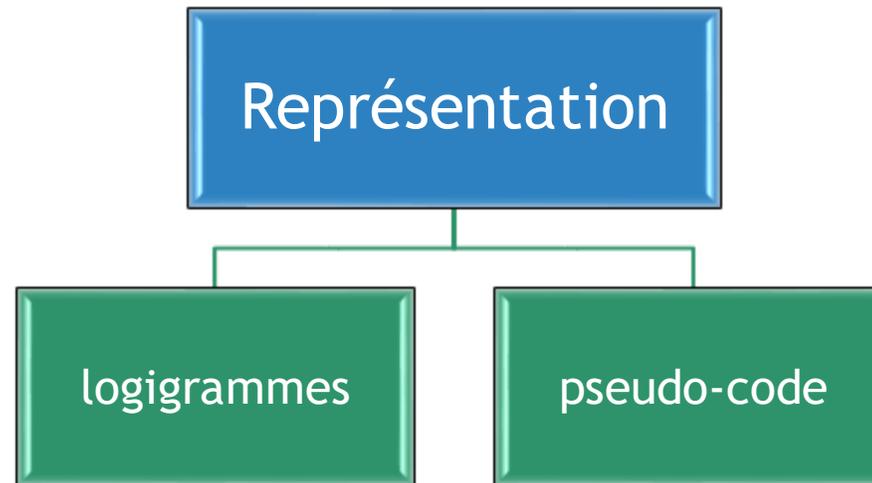
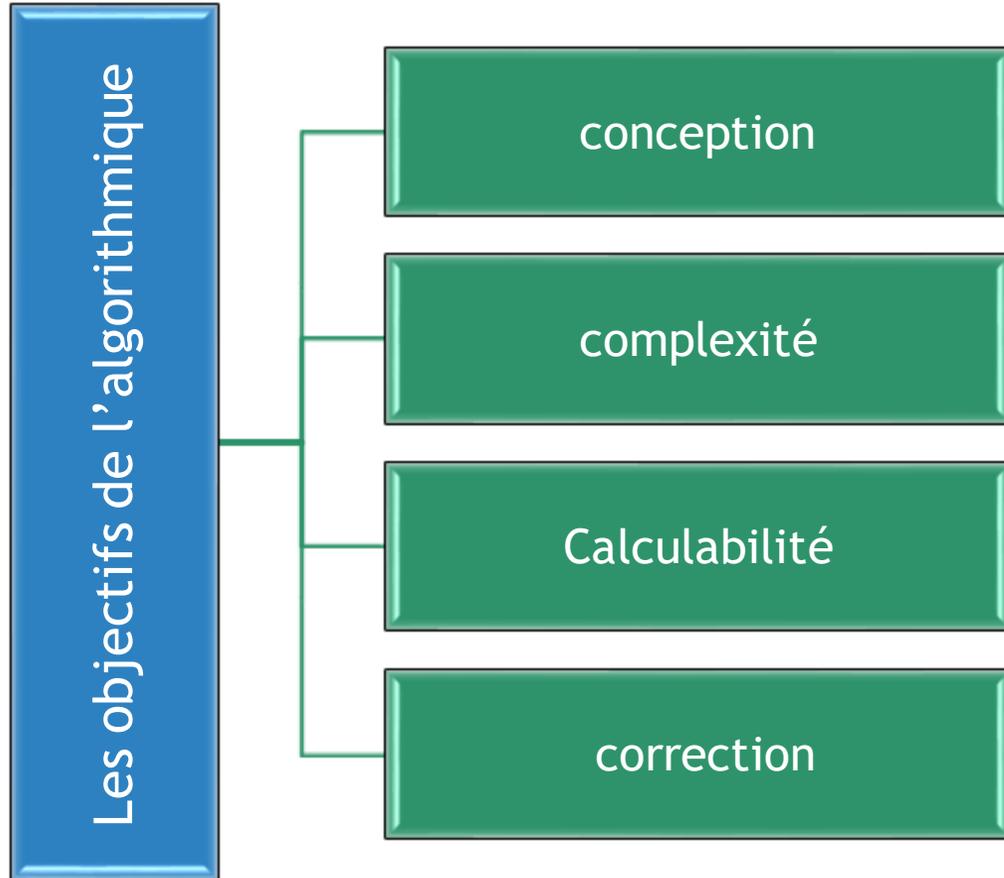
Objectif Zéro Papier



Les objectifs de l'algorithmique

- ▶ 1. La conception : Dans ce cas, la problématique est : quelle est la suite d'instructions permettant la résolution du problème posé ? En d'autres termes, l'objectif est de trouver un algorithme permettant de réaliser le travail souhaité.
- ▶ 2. La complexité : Il n'y a pas qu'un seul algorithme permettant de résoudre un problème donné. Tout comme pour reprendre l'exemple de la notice de montage, il n'y a pas forcément qu'une seule façon d'assembler les pièces. Par contre, certaines manières de faire sont peut-être plus aisées ou plus rapides... La complexité a pour objectif de noter l'efficacité d'un algorithme en comptant le nombre d'opérations réalisées. La problématique est donc : quel est l'algorithme le plus économe ? La complexité cherche généralement à minimiser le temps nécessaire pour exécuter l'algorithme, mais il est également possible de chercher à optimiser l'utilisation de la mémoire.
- ▶ 3. La calculabilité : Pour certains problèmes, il n'existe pas d'algorithme permettant de les résoudre. Par exemple, il n'existe pas d'algorithme prenant en paramètre un autre algorithme et qui indique si ce dernier terminera ou ne terminera jamais. La problématique de la calculabilité est donc : existe-t-il un algorithme permettant de résoudre le problème posé ?
- ▶ 4. La correction : La dernière finalité de l'algorithmique est de s'assurer de la fiabilité de l'algorithme. La question

L'algorithmique





app.wooclap.com/GKNKZF



- 1 Allez sur wooclap.com
- 2 Entrez le code d'événement dans le bandeau supérieur

Code d'événement
GKNKZF



- 1 Envoyez **@GKNKZF** au **06 44 60 96 62**
- 2 Vous pouvez participer

 Désactiver les réponses par SMS

Deux questions clés :

- ▶ **C'est quoi** l'orientée objet **théoriquement**?
- ▶ **Pourquoi** l'orientée objet?

C'est quoi l'orientée objet théoriquement?

- ▶ C'est quoi l'orientée objet théoriquement?
 - Paradigme de programmation :. Un paradigme est une façon d'approcher la programmation informatique, c'est-à-dire de concevoir son code et de structurer ses différents éléments.
 - La programmation impérative : recette de cuisine C, Python PHP (assignation, boucle, branchement et séquence d'instructions), **sépare état et comportement**
 - La programmation déclarative : SQL, Prolog. **État et pas de comportement**
 - Programmation fonctionnelle : dérivé de déclarative comme JavaScript
 - Programmation événementielle : HTML+JavaScript (boucle asynchrone) **comportement mais pas état**
 - Programmation Orientée Objet : Java , **état et comportement**

- ▶ Programmation impérative : programme = séquence d'ordres, d'instructions
- ▶ Programmation procédurale : programme = ensemble de procédures impératives
- ▶ Programmation fonctionnelle : programme = ensemble de fonctions (pas de variables, pas d'itération)
- ▶ Programmation logique : programme = ensemble de formules logiques
- ▶ Programmation orientée objet : programme = ensemble d'objets qui interagissent entre eux. Extension de la programmation procédurale.

Paradigme n'est pas un langage

- ▶ Un **paradigme** est un ensemble de **concepts de programmation**
- ▶ Un **langage de programmation**, c'est un **moyen** de programmer à partir d'une **syntaxe**
- ▶ Un **langage de programmation** peut **exploiter un ou plusieurs paradigmes** de programmation
(**langage multi-paradigmes**)
- ▶ POO (**P**rogrammation **O**rientée **O**bjet) : un ensemble de concepts s'appliquant à de nombreux langages de programmation

Pourquoi l'orientée objet?

Qualité d'un logiciel

▶ Qualité Interne

- ❑ Modulaire
- ❑ Facile à lire

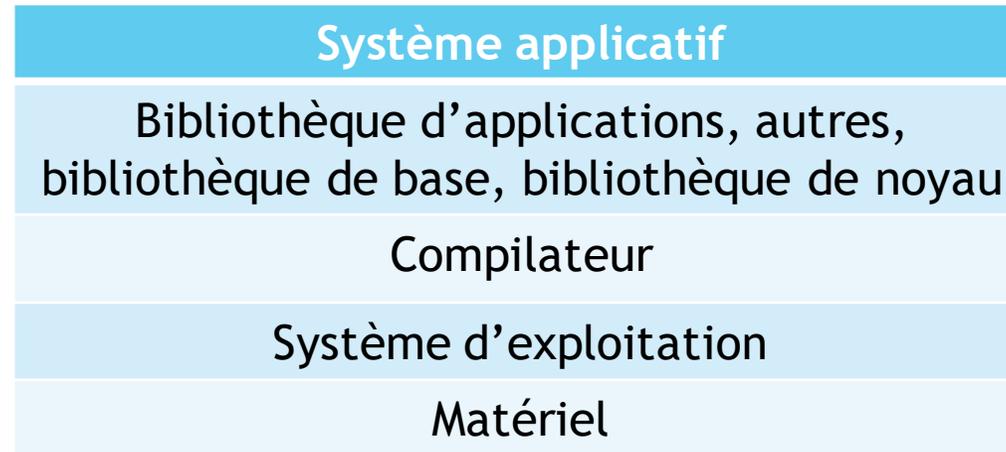
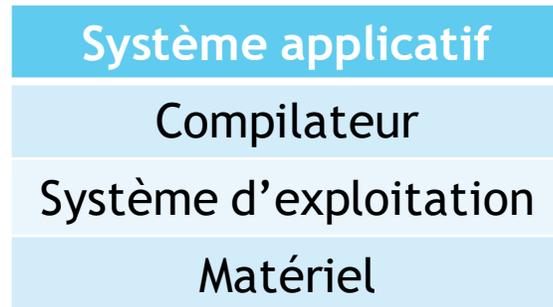
▶ Qualité externe

- ❑ Correction
- ❑ Robustesse
- ❑ Extensibilité
- ❑ Réutilisabilité
- ❑ Comptabilité
- ❑ Efficacité
- ❑ Portabilité
- ❑ Facilité d'utilisation
- ❑ Fonctionnalité
- ❑ Ponctualité

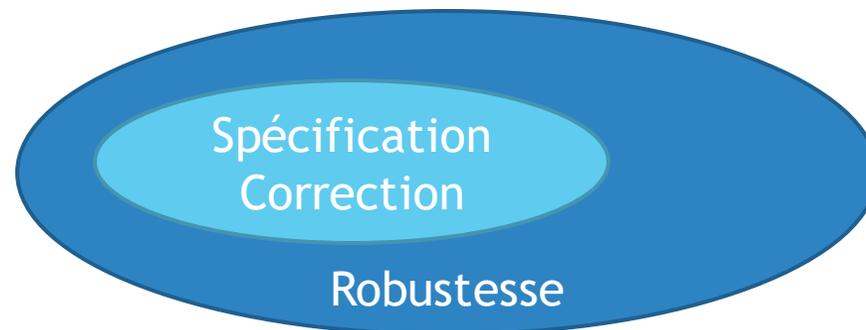
▶ Autres qualité

- ❑ Vérifiabilité
- ❑ Intégrité
- ❑ Réparabilité
- ❑ Economie
- ❑ Documentation
- ❑ Compromis
- ❑ Préoccupations essentielles

- ▶ **Correction** : la correction est la capacité que possède un produit logiciel de mener à bien sa tâche telle qu'elle a été définie par sa spécification.



- ▶ **Robustesse** : La robustesse est la capacité qu'offrent des systèmes logiciels à réagir de manière appropriée à la présence de condition anormales



- ▶ **Extensibilité** : l'extensibilité est la facilité d'adaptation des produits logiciels aux changements de spécification .
- ▶ **Réutilisabilité** : la réutilisabilité est la capacité des éléments logiciels à servir à la construction de plusieurs applications différentes.
- ▶ **Compatibilité** : la comptabilité est la facilité avec laquelle des éléments logiciels peuvent être combinés à d'autres
- ▶ **Efficacité** : l'efficacité est la capacité d'un système logiciel à utiliser le minimum de ressources matérielles, que ce soit le temps machine, l'espace occupé en mémoire externe et interne, ou la bande passante des moyens de communication.

- ▶ **Portabilité** : la portabilité est la facilité avec laquelle des produits logiciels peuvent être transférés d'un environnement logiciel ou matériel à un autre.

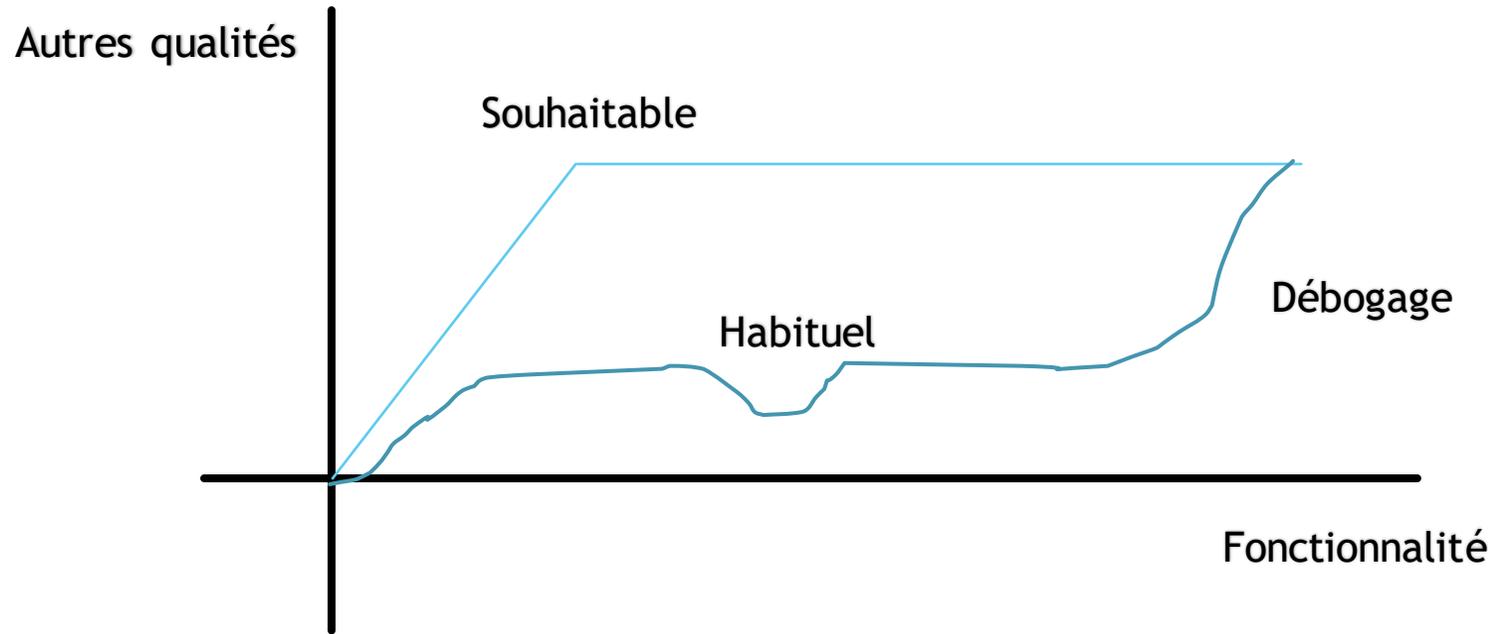
- ▶ **Facilités d'utilisation** : La facilité d'utilisation est la facilité avec laquelle des personnes présentant des formations et des compétences différentes peuvent apprendre à utiliser les produits logiciels et à l'en servir pour résoudre des problèmes. Elle recouvre également la facilité d'installation, d'opération et de contrôle.

Principe de conception d'interface utilisateur :

Ne prétendez pas connaître l'utilisateur,

Vous ne le connaissez pas

- ▶ **Fonctionnalité** : La fonctionnalité est l'étendue des possibilités offertes par le système.



- ▶ **Ponctualité** : la ponctualité est la capacité d'un système logiciel à être livré au moment Désiré par ses utilisateur, ou avant.

De la maintenance logicielle

