



POA

Programmation Objet Avancée

C.M. L2 Info
Pierre-Alain MASSON



- Moodle (enregistré en L2S3) : **L2INFO-POA-2024**
- Partie Objets : 3 intervenants
 - François Bonneville : 1TD + 2TP
 - Ismael Jecker : 1TD + 1TP
 - Pierre-Alain Masson : resp. UE, CM, 1TP
- Codage en Java en TP **et en TD** (pas de PDL)
 - Celles et ceux qui le peuvent amèneront leur machine en TD
- Complément de formation (changement maquette)
 - **Docker** – CM+TP – Louis-Claude Canon
 - **GIT** – CM+TP – Frédéric Dadeau



- TP partie Objets
 - Plusieurs sujets *non notés* mais à *rendu obligatoire*
 - Un projet de fin de semestre : par binôme, avec *démo*, *noté*
- Evaluations : 2 DS sur table + projet
- Calcul de la note (2^{nde} chance)
 - Chance 1 : DS1 /3 + DS2 /3 + projet /3
 - Chance 2 : DS1 30% + DS2 30% + projet 40%
- Partie complément de formation (Docker, GIT)
 - Modalités à préciser par les enseignants responsables

Rappel des notions principales vues en L1 (1)

- **Classe** : une entité qui regroupe les données (**attributs**) et les traitements (**méthodes**)
- Une classe définit un **type** (pour typer des variables)
- L'instanciation d'une classe produit un **objet**
 - **Constructeurs** : méthodes de la classe pour allouer l'espace mémoire à l'objet, et initialiser ses attributs
 - L'opérateur d'instanciation **new** invoque un constructeur
- Dans un programme, une variable de type objet contient une **référence** à l'objet instancié

- Les méthodes autorisées pour manipuler un objet constituent son API
- **Encapsulation** : les attributs sont cachés (**privés**) pour forcer le recours à l'API (méthodes **publiques**)
- Les attributs et les méthodes peuvent être attachés :
 - Aux instances (**non-statique**, par défaut) : autant d'exemplaires que d'instances
 - A la classe elle-même (**statique**) : un seul exemplaire, partagé par toutes les instances
 - Pas besoin d'instanciation pour invoquer les attributs ou méthodes statiques : on les invoque sur un nom de classe plutôt que sur un nom d'objet



1. Introduction à la POO
2. Une **classe** : définition de nouveaux objets
3. **Instanciation** et utilisation d'objets
4. Création des objets : les **constructeurs**
5. **Références**, visibilité des variables
6. **Encapsulation** et masquage des données
7. **Statique**, ou d'instance ?
8. Héritage
9. Polymorphisme
10. Classes abstraites et interfaces
11. Introduction aux types génériques
12. Exceptions en java
13. *Compléments syntaxiques*

Contenu du cours (partie 2)



1. Collections (utilisation de l'API Java)
2. Complément de formation : Docker + GIT