

Java Objet

Présentation du cours

Olivier Cailloux

LAMSADE, Université Paris-Dauphine

Version du 26 février 2025

L'enseignant

- Olivier Cailloux
- olivier.cailloux@dauphine.psl.eu
- P405ter
- Anciennement : Développeur sur projets de recherche
- Enseignant chercheur au LAMSADE

Objectifs pédagogiques

- Programmer de « vraies » applications
- De qualité
- Fournir et utiliser des composants réutilisables
- Conception objet
- Prise en main d'outils de dév. avancés :
 - Visual Studio Code ;
 - Maven ;
 - git (livraisons exclusivement via GitHub)

Contenu

- Syntaxe Java
- Programmation objets : responsabilités ; techniques
- Maven
- Éléments d'ingénierie : programmation par contrat ; patrons de conception. . .
- Collections
- Tests unitaires
- Utilisation de bibliothèques tierces
- Exceptions
- Logging
- Et plus selon demandes

Intérêt pratique

- Technologies omniprésentes et très demandées (15 In-Demand Tech-Focused (And Tech-Adjacent) Skills And Specialties, 2022, [Forbes Technology Council](#) ; Emploi informatique 2023, [Developpez.com](#))
- Qu'on soit programmeur, qu'on discute avec des programmeurs
- Décomposition en responsabilités, en sous-problèmes
- Respect des spécifications
- Utile au-delà de la programmation

Demander simplement au chat ?

- Le chat doit savoir ce qui est à faire
- Il faut dialoguer avec le chat, pas l'écouter aveuglément
- Comment savoir si le chat se trompe ? Lui demander d'expliquer ? Et si le chat **baratine** ?
- Le chat **aide** (pour le moment) surtout les débutants

Donc

- Savoir calculer, savoir coder
- Connaître la plomberie sans faire de la plomberie
- Avec aide
- Comprendre et critiquer

Prérequis

- Notions algorithmiques élémentaires (boucles, structures de listes, d'arbres. . .)
- Familiarité avec un langage tel que C++ ou Python
- Manipulation de votre système d'exploitation : installation de logiciels, navigation dans le système de fichiers, démarrage de programmes
- Capacité à comprendre des textes en anglais liés à l'informatique
- Un ordinateur fonctionnel

Évaluation

100% CC

- \approx 7 tests réguliers en séance (annoncés)
- Devoirs maison et Remises de projet
- Aggrégation des notes reçues au long de l'année
- Pondérations (« owa », sous réserve de modifications) :

tests 1 à 6 au mieux par individu parmi $\{1/10, 1/10, 1, 1, 1, 1\}$ (plafond possible pour tests simples),

dernier test 2

projet VS tests au pire par individu parmi $\{1/5, 4/5\}$

Évaluation des tests

- Code doit compiler
- Livraison via git
- Respect précis des instructions
- Évaluation généralement automatique
- Points pour chaque aspect fonctionnel
- Rattrapage possible avec pénalité

Dans ce cours

- Exercices en séance : avec ressources pré-existantes, sans LLM
- Projet : avec LLM

Travail attendu

- $\{[(25 \text{ h} / \text{ECTS}) \times 5 \text{ ECTS}] - 51 \text{ h}\} / 16$ inter-séances
- 5 heures de travail entre chaque séance en moyenne
- [Prenez des notes](#)
- Poursuivre les exercices chez vous, cf. page GitHub du cours

Licence

Cette présentation, et le code LaTeX associé, sont sous [licence MIT](#). Vous êtes libres de réutiliser des éléments de cette présentation, sous réserve de citer l'auteur.

Le travail réutilisé est à attribuer à [Olivier Cailloux](#), Université Paris-Dauphine.