

# Java Objet

## Présentation du cours

Olivier Cailloux

LAMSADE, Université Paris-Dauphine

Version du 25 décembre 2016

# L'enseignant

- Olivier Cailloux
- [olivier.cailloux@dauphine.fr](mailto:olivier.cailloux@dauphine.fr)
- Coordonnées : cf. [annuaire](#) de Dauphine

# Objectifs pédagogiques

- Programmer des vraies applications
- De qualité
- Fournir et utiliser des composants réutilisables
- Conception objet
- Prise en main d'outils de dév avancés :
  - Eclipse ;
  - Maven ;
  - Git...

# Objectifs pédagogiques : modélisation

## Modélisation

- Réponse à des besoins exprimés vaguement
- Appui sur les standards du web actuels
- Réusinage fréquent
- Dosage du réalisme et de l'intérêt des fonctionnalités

Approche agile ? À moitié !

- Livraisons fréquentes
- Travail en binôme
- Réusinage intense

# Intérêt pratique

- Technologies utiles
- Qu'on soit programmeur, qu'on discute avec des programmeurs
- Résolution de problèmes
- Décomposition abstraite en responsabilités
- ... au-delà de la programmation

# Prérequis

- Programmation élémentaire en Java, manipulation d'un environnement de développement, compréhension des notions algorithmiques élémentaires.
- Capacité à comprendre des textes en anglais liés à l'informatique.

# Approche

- Pédagogie par projet et *partiellement* inversée
- Central : travail sur projet
- Objectif : un projet utile
- Un projet  $\neq$  par groupe

# Mise en œuvre

- Durant séance  $x$  : constituez un binôme  $\neq$  du précédent
- Programmation en binôme, équipes tournantes :  
diversification, pas spécialisation
- Travaillez sur une fonctionnalité de votre projet *ou* des exercices
- À remettre avant le jour de la séance  $x + 1$
- Livraisons exclusivement via Git



# Agilité

- À gérer de façon agile car fonctionnalités ajoutées au fil du cours : cycles courts, réusinage...
- Je joue le rôle du client : détail des fonctionnalités à implémenter
- Fonctionnalités de l'application décrites de manière vague : à vous de m'interroger

# Évaluation

L'évaluation se fait par contrôle continu exclusivement.

- Durant chaque séance : je tire des personnes au hasard
- J'évalue, avec le binôme, le travail fourni depuis la dernière évaluation
- Note finale : agrégation des notes reçues au long de l'année (nombre de notes : aléatoire)
- *Vous êtes toujours censés comprendre votre code*
- Si  $\neq$  niveaux : l'avancé aide le débutant
- Fin d'année : présentation collective de vos projets
- Vote pour la meilleure application

## Aspects pris en compte

- Qualité du code
- Respect des demandes de l'utilisateur
- Mise en œuvre adéquate des technologies dans l'application
- Livraisons régulières
- Ampleur des fonctionnalités
- Qualité générale de l'application
- Qualité de la présentation finale...

## Travail attendu

- Étudiants en commun avec cours Dan Dimcea : 6 ECTS
- $\{[(25 \text{ à } 30\text{h} / \text{ECTS}) \times 5 \text{ ECTS}] - 48 \text{ h}\} / 7 \text{ inter-séances}$
- 18 heures de travail entre chaque séance
- Autres étudiants : 9 heures entre chaque séance

# Plan

- Git, Maven
- Programmation objets : responsabilités ; techniques
- Éléments d'ingénierie : programmation par contrat ; patrons de conception...
- Exceptions
- Logging
- Utilisation de bibliothèques tierces
- Interfaces graphiques (SWT)
- Et plus selon demandes

# Attendu

- Rediriger vos e-mails @ Dauphine si nécessaire pour vous assurer de recevoir les annonces
- Vous êtes supposé lire les annonces publiées sur MyCourse
- Suivre *scrupuleusement* les instructions qui s'y trouvent SVP

# 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.