Java Objet : Expériences en cours

Olivier Cailloux

LAMSADE, Université Paris-Dauphine

4 juin 2019

https://github.com/oliviercailloux/REPO

LAMSADE
UMR CNRS 7243
laboratoire d'analyse et modélisation de systèmes pour l'aide à la décision

DAUPHINE
UNIVERSITÉ PARIS

PSL

Outline

Outline

Cours

- Java Objet (et Java Projet), MIDO, L3 Apprentissage
  - 17×3h
  - 2016–, 2017–, 2018–
- Serveurs Java (Java EE), MIDO, M2 SITN Apprentissage
  - 8×3h
  - 2015–, 2016–, 2017–
- Serveurs Java (Java EE), MIDO, M2 IF Classique, option
  - 8×3h
  - 2015–, 2016–, 2017–

En pratique

- Appui sur livre ouvert de Eck pour cours (peu exploité) et exercices (très exploité)
- Tout sur internet (pas de papier)
- Exemples de code, diapos
- Page de suivi des séances avec liens vers les explications par sujet traité
- Manque (?) : syllabus cohérent

Outline

Objectifs

- Principes d'ingénierie logicielle : code maintenable plutôt que code fonctionnel
- Code en pratique : sur machine
- Usage des outils modernes d'aide (compilation, avertissements)
- Rentabilité du temps investi à maitriser un outil (t.q. git)
- Recherche de documentation : code copié d'une source hasardeuse VS source officielle / récente / crédible
- Volonté de les faire travailler entre les séances

Évaluation

Fluctuante selon les années !

- Évaluation projet et contrôle continu (pour suivi des explications en séance)
- Projet : contributions en binômes puis individuelles puis retour aux binômes
- Avec devoirs maison
- Avec évaluation orale en séance
- Avec exercices en séances
- Avec QCMs

Outline

Cadre

- Groupes 4 à 8
- Chaque groupe un projet différent
- Projets à objectif utile, publication open source visée
- Exemple : bibliothèque gestion de préférences ; gestion de bibliographie collaborative ; gestion des cours de Dauphine. . .
- Énoncés courts (2 pages)

En pratique

- Appui important sur GitHub et Pull Requests
- Exemple
- Année découpée en itérations
- Travail par binôme sur tâches lors d'une itération
- Binômes tournants
- Fusion dans master seulement sur mon accord
- Feuille de calcul pour garder trace des contributions par binôme par itération
- Étudiants fréquemment en difficultés avec git

Outline

Devoirs maison

Remise via GitHub

Devoirs du livre

- Ils ont les corrections à l'avance (!)
- Comptent en tout pour 10% de la note
- Notation binaire (fait / pas fait) ; zéro si en retard
- Pris sérieusement (?)

Devoirs hors livre

- $-3/20$ par heure de retard
- Correction automatique après coup
- Note transmise via MyCourse
- Correction : éléments en séance ou code complet

QCM

- Deux ou trois par cours
- Une minute par question
- Avec MyCourse (très insatisfaisant)
- Correction automatique par MyCourse, résultats immédiats
- Points négatifs
- Difficile à formuler et calibrer
- En pause en attendant meilleurs outils

NB : également utilisation de WooClap en séance, très appréciée, sans notation

Exercices notés en séance

- GitHub Class room
- Lien fourni en début de séance (exemple)
- Chaque étudiant démarre avec une copie du dépôt que j'ai créé au préalable
- Énoncé fourni sur le dépôt lui-même
- Doivent ajouter du code, des tests, ...
- Avec documents et internet
- 20 à 40 minutes, puis $-3/20$ par minute de retard
- Correction automatique après coup
- Outil *presque* formidable

Outline

Quelques bénéfices

- Commentaires personnalisés sur projets appréciés
- Temps important passé en dehors des séances
- Compréhension du point de vue de l'ingénierie logicielle (pas seulement code fonctionnel)
- Compréhension (trop tardive) de l'intérêt des explications en séances
- Capacité à coder
- Gestion de projet, sens des reponsabilités
- Projets réellement réutilisables au fil du temps

Quelques faiblesses

- Refus d'investir du temps pour maitriser ses outils
- Recours encore fréquent sans critique à des sources aléatoires
- Connaissance intuitive mais pas théorique : faible connaissance des mécanismes de Java, de la théorie des langages, des raisons des choix techniques et des alternatives
- Possiblité toujours existante de se cacher derrière les collègues
- Difficulté persistante d'évaluer les compétences des étudiants
- Difficulté de doser la vitesse d'avancement des séances : fournir assez de connaissances pratiques ; ne pas inonder les étudiants (manque de temps d'exercices selon étudiants)

Un échec temporaire

- Des bénéfices en termes pédagogiques
- Mais un échec tout de même

Analogie boiteuse

- « J'ai trouvé une méhode pour produire deux fois plus de carottes »
- . . . mais il faut deux fois plus de travail et de terre

- La ressource de l'enseignant : le temps (suivi et préparation)
- Ici : beaucoup trop important
- Espoir d'amélioration future !

Idées

- Vidéos pour (re-)accès au contenu nécessaire en fonction du besoin
- Démarrage des projets très tôt
- Entrainement sur exercices corrigés automatiquement en temps réel
- Jointure avec cours d'UML pour séparation ingénierie logicielle et implémentation

Thank you for your attention !