

Conception d'applications web en Java

Présentation du cours

Olivier Cailloux

LAMSADE, Université Paris-Dauphine

Version du 13 décembre 2016

L'enseignant

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

Objectifs pédagogiques

- Programmer des applications d'envergure
- De qualité
- Portables

Complexité :

- Appui sur API ouvertes
- Programmer selon les specs
- Beaucoup de paramètres \Rightarrow comprendre !

Objectifs pédagogiques

Mise en œuvre des patterns


- Mise en œuvre des patterns dans les spécifications Java EE
 - Quand les mettre en œuvre ?
 - Applications dans programmes propres
-
- Prise en main d'outils de dév avancés :
 - Eclipse ;
 - Maven ;
 - Git...

Un environnement imparfait

Log Viewer – Mozilla Firefox

SP servlet best pract... x Maven – Introduc... x http://loc...ityManager x PersistenceCont... x PersistenceCont... x hibernate - When... x

localhost:4848/common/logViewer/logViewer.jsf?instanceName=server&loglevel=INFO&viewResults=true

 **An error has occurred**
Check server log for more information.

Log Viewer

View, search, and filter a server log file using basic and advanced options. Refer to the Log Levels page for information about log levels you can filter here.

[Advanced Search](#)

Search Criteria

Text search:

Only log entries containing the specified text will be displayed. Search is case sensitive.

Timestamp: ☒ **Most Recent** ☐ **Specific Range:**

Log Level: ☐ **Do not include more severe messages**

Log entries are limited to those stored in the log file. Set appropriate log level in the Log Level page to ensure data is logged.

[Modify Search](#)

Instance:

Log File:

Log Viewer Results (0)

Un environnement complexe

De <http://graphserver.github.io/graphserver/> :

First, get Java 1.6. Or Java6. I think they're the same thing. Get the JDK, but not the JVM, the JRE, SDK, SDN, or the JCE. Note Java SE comes with Jave EE, which is apparently at version 5, and may or may not have anything to do with J2EE. I don't know what those have to do with anything. Google it or something. I don't know. They don't make it particularly easy for you. It's like, they've got more money than god and nothing pleases them better than pouring all that expertise into baffling the hell out of you. Honestly, I can't stand Java, but you need it to run Osmosis.

Objectifs pédagogiques, suite

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 !

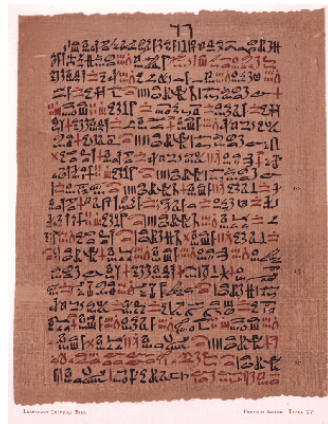
- Livraisons fréquentes
- Travail en binôme
- Réusinage intense

Intérêt pratique

- Qu'on soit programmeur, qu'on discute avec des programmeurs
- Prendre de la hauteur, éviter les tâches répétitives et se concentrer sur le conceptuel
- Respect et compréhension des standards (aperçu de la façon dont ils sont construits) : compétence essentielle
- ... dans de multiples domaines
- Importance des patterns dans de multiples domaines
- Technologie en vogue
 - Java EE VS Spring VS ... ?

Les bienfaits de la complexité

- Activité plus difficile : souvent plus attrayante
- Évite les activités répétitives : complexité amène diversité
- Récompenses plus grandes
- Recherche d'un accomplissement personnel
- Cercle vertueux : accès à activités plus complexes
- Pas un bien positionnel : accessible à tous



Écriture hiératique, égypte ancienne

Prérequis

- Programmation en Java, manipulation d'un environnement de développement, compréhension des notions algorithmiques élémentaires.
- Capacité à comprendre des textes en anglais liés à l'informatique.
- HTTP, HTML, XML, XSD, SQL.
- Bases théoriques de Java EE.

Mise en œuvre

- Pédagogie par projet et *partiellement* inversée
- Central : travail sur projet
- Objectif : un projet utile
- Un projet \neq par groupe
- Durant cours x : constituez un binôme \neq du précédent
- Programmation en binôme, équipes tournantes :
diversification, pas spécialisation
- Travaillez sur une fonctionnalité de votre projet
- À remettre avant le jour du cours $x + 1$
- 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 cours : 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
- $\{[(25 \text{ à } 30\text{h} / \text{ECTS}) \times 6 \text{ ECTS}] - 48 \text{ h}\} / 7$ inter-séances
- 18 heures de travail entre chaque séance
- Autres étudiants : 9 heures entre chaque séance
- Peuvent donc sauter une séance sur deux
- Ou auront meilleure note
- Ou binôme se sépare pour une partie du travail

Plan

- Git, Maven
- CDI
- JSF
- JAX-RS
- Et plus selon demandes

Plus : Éléments d'ingénierie (programmation par contrat...)

Astuces importantes

- Éviter l'essai / erreur : *comprendre*
- Poser des questions !
- Commencer simple
- Si ça ne fonctionne pas : faire plus simple
- Exclure bugs possibles pas à pas
- Éviter débuggage pas à pas
- Se méfier du code auto-généré
- Choisir *une* approche cohérente. Mélange deux tutos : problèmes presque assurés (\neq versions ; \neq annotations...)

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

À faire

Avant chaque cours :

- Constituez un binôme \neq du précédent
- Par binôme : Création d'un problème git (*issue*) sur projet Git
- Par binôme : Commit branche perso, nom branche = n° issue
- Par chacun : Indication sur MyCourse de l'URL issue travaillée + nom binôme
- Conseil : *vérifier* que je verrai votre code en effectuant un nouveau clone vous-même
- Voir détails sur MyCourse

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.