L'UE ACO (édition parcours IL)

Noël PLOUZEAU

IRISA/ISTIC

Les intervenants

- Noël Plouzeau
  - Concepteur et responsable du cours
  - CM, TD, TP, évaluateur en chef
  - responsable du parcours IL du M1 info

Pourquoi ce cours

- Très grosse différence entre
  - écrire des lignes de code dans un langage qu'on maîtrise (plus ou moins)
  - faire la conception
- ACO traite essentiellement de conception
  - pour des architectures à objets (Java, C#, C++, Swift, etc)
  - même si en pratique reste valable pour du C

Une petite métaphore

- Tirée du domaine du bâtiment
- Différence entre
  - maçonnerie
  - architecture de bâtiment
- De la cabane de jardin à l'hôpital

Compétences et tâches d'un maçon

- Être capable de **lire** des plans, la capacité d'en rédiger n'est pas nécessaire
- Être capable de réaliser les opérations **élémentaires** de construction (monter un mur, couler du béton, etc)
- Garantir la **conformité** du bâtiment par rapport aux plans
- Garantir la **qualité** (verticalité des murs, prise du béton)
- La qualité est **contrôlée** par d'autres personnes

(suite)

- Un maçon doit aussi être capable de
  - déduire les détails pratiques concrets grâce aux **règles et bonnes pratiques** métier
  - déceler et signaler les **erreurs de conception**

Compétences et tâches de l'architecte

- Être capable de comprendre les **services** que le bâtiment doit rendre
- Être capable de déduire des informations **implicites**, grâce à sa connaissance du **domaine métier** du client (p. ex. normes hospitalières)
- Être capable de produire une architecture cohérente, **précise** mais faisant **abstraction des détails** de maçonnerie, et selon des vues de mises en œuvre spécialisées (électricité, plomberie, ventilation, etc)
- Assurer la conformité des services **rendus** par rapport aux services **attendus**
- L'évaluation finale **de conformité** (réception) est faite par des tiers

Et pour la réalisation de logiciel ?

- Les grandes phases de la réalisation d'un système logiciel sont très proches de ce qui se fait pour le bâtiment
- Néanmoins la métaphore a ses limites
  - Le logiciel est mou (« software ») pour permettre la réutilisabilité, les évolutions, etc
  - Et c'est là que les problèmes commencent (voir suite du cours)…

Construction logicielle

- La conception abstrait les détails
  - emploi d'une notation graphique pour la structure et le comportement
  - les détails du code n'apparaissent pas
- Les développeur doivent être capables de lire les « plans » et de compléter les détails (le code ;-)
- Contrôle continu de la qualité (tests, « visites de chantier »)

Objectifs généraux du cours ACO

- Vous devrez être capables d'employer de façon justifiée
  - les principes et techniques de conception
  - les principes et techniques de contrôle de la qualité
- En deux mots : de l'ingénierie logicielle

Prérequis

- Avoir un niveau intermédiaire in Java (v 8)
  - Le cours comporte des rappels sur des concepts Java importants
- Être capable de lire et d'écrire des diagrammes UML avec un niveau intermédiaire
  - Cela vous sera utile en TD et en TP

Mode d'organisation

- Des CM à distance, synchrones (permet les questions)
- Chaque séance sera enregistrée pour que vous puissiez la revoir
- Des TD en présence (si possible) : n'oubliez pas vos notes de cours
- Des TP en présence (si possible) : pour compléter les détails de la conception faite en TD **et tester**

Concernant les TP

- Nous emploierons Java 8 au minimum
  - Les tests emploieront JUnit 5, Coverage, sonarlink
  - Environnements recommandés
    - Eclipse (gratuit mais pataud voire bogué)
    - ou IntelliJIDEA (l'ISTIC a des licences, et vous pouvez en avoir à titre personnel)
  - Forge : l'instance gitlab de l'ISTIC
    - https://gitlab.istic.univ-rennes1.fr

Organisation des TP

- Conditions
  - Travail en binôme
  - Emploi de la forge ISTIC pour le partage du code source de vos TP avec votre binôme
  - Chaque personne du binôme se connecte avec son login UR1 sur la forge ISTIC
  - Vous m'ajouterez avec le rôle **reporter** (sinon je n'ai pas accès, et donc pas de rendu de TP…)

Évaluation

- Séance de TD (20 heures)
- Séance de TP (14 hours)
  - La dernière séance de TP comporte une courte démonstration
- Examen écrit (contrôle continu) en décembre : 2 heures, les documents de cours ACO sont autorisés
  - QCM, QROC, code pourri à corriger, etc
- Un contrôle continu en milieu de semestre est toujours possible ;-)…

Supports de cours

- Une version PDF des pages du CM sera placée dans /share/m1info/ACO)
- Mes exemples sont sur le gitlab ISTIC (et certains sur github)

Questions?