Projet Java 2ème session 2020

L'évaluation de seconde session pour l'UE Développement informatique avancé orienté application se déroulera sous la forme d'une épreuve unique pour les deux activités d'apprentissage, à savoir un projet à développer individuellement qui sera présenté et défendu le jour de l'examen.

ActA Théorie

Les étudiants ayant échoué à l'ActA de théorie seront interrogés sur la matière théorique lors de la défense. Les questions posées pourront être en lien avec le projet développé.

ActA Pratique

En plus de la défense du projet, il pourrait être demandé de développer pour démontrer la maitrise des pratiques de programmation enseignées dans le cours.

Enoncé du projet

Chaque étudiant devra développer une petite application Java (jeu ou application utilitaire) sur base d'un cahier des charges, application qui :

- exploite de manière adéquate les concepts orientés-objet,
- présente une interface graphique
- respecte l'architecture MVC
- comporte au moins une Collection ou une Map utilisée à bon escient
- comporte au moins un héritage utilisé à bon escient
- comporte au moins une interface utilisée à bon escient

Les exigences sont détaillées dans la grille critériée annexe.

Modalités pratiques

Il est demandé aux étudiants souhaitant présenter cette UE en seconde session d'envoyer un mail à l'équipe enseignante avant le 30/6 avec :

- Le descriptif de son sujet et un premier jet de cahier des charges
- La/Les ActA qu'il représente (Théorie, pratique ou les deux)
- L'adresse de son repository Github.

Le code de l'application sera hébergé sur un repository Github dont l'adresse sera renseignée à l'équipe enseignante dès que possible. Les commits devront être :

- réguliers et petits, porter sur de petites fonctionnalités (on ne veut pas de gros bouts de code qui arrivent d'un coup !)
- comporter des messages de commits explicites
- ne pas être effectués depuis l'interface Github, mais bien depuis un logiciel approprié (outil git dédié ou intégré à votre IDE)

Le Wiki de ce Github constituera le rapport.

Le repository ne pourra plus être modifié au-delà de l'échéance du projet, dont la date vous sera communiquée en temps voulu (a priori, 3 jours avant la date de l'examen).

Rapport / Wiki

Le Wiki regroupera tous les éléments de rapports selon la structure ci-dessous. La page d'accueil du Wiki présentera une table des matières permettant une navigation aisée entre les éléments demandés. Vous devez respecter la numérotation proposé ci-dessous.

- **A. Analyse et conception**
  - A.1 **Description du sujet** (thématique, objectif, client, utilisateurs, …)
  - A.2 **Cahier des charges** (idéalement, les besoins fonctionnels sont décrits sous forme de User Stories)
  - A.3 **Diagramme de classe UML** et commentaire explicatif
  - A.4 Sur base du diagramme UML, présentation et justification des **concepts Orienté-Objets** exploités dans le projet (dont héritage et interface)
  - A.5 **Architecture MVC** : Explication de l'implémentation du pattern MVC dans votre projet
  - A.6 **Collection** : Présentation de la ou des structures de données utilisées dans le projet et justification de leur ** utilisation
  - A.7 **Interface graphique** : Explication de la structure des écrans de l'application
- **B. Implémentation et robustesse**
  - B.1 **Gestion des erreurs** : Présentation de la manière dont les erreurs et les cas limites sont gérés dans le code (encapsulation, exception, programmation défensive, spécifications), avec éventuellement des exemples
  - B.2 **Tests unitaires** : Présentation des tests unitaires effectués et réflexion sur la couverture du code
  - B.3 **Git** : Explication de la manière dont l'outil Git a été exploité
  - B.4 **Propreté du code** : Présentation des éventuels outils de formatage de code, des conventions de codage respectées, et d'un exemple de refactoring de code prenant en compte les bonnes pratiques de programmation vues au cours (lisibilité, simplification, niveau d'imbrication raisonnable, conditions booléennes, choix des noms de variables,…)
- **C. Documentation**
  - C.1 **Javadoc**
  - C.2 **Manuel d'installation** (explication brève de comment exécuter le programme)
- **D. Bilan**
  - D.1 **Conclusion sur le résultat obtenu** (qualité du code, objectifs atteints, bugs résiduels)
  - D.2 **Conclusion sur vos apprentissages**
  - D.3 **Points forts et pistes d'amélioration**
  - D.4 **Bonus** description des qualités de votre projet qui ne sont pas reprises dans la grille (ex : client réel, utilisation d'outils intéressants, mise à disposition de l'application sous forme d'un package installable ou exécutable, utilisation d'un socket ou de threads, …)

Il vous est demandé d'illustrer autant que possible vos affirmations par des extraits de code (lien hypertexte vers les instructions concernées sur le Github)

Défense du projet

Lors de l'examen oral, l'étudiant présente son projet en **5 minutes** : Présentation du sujet et du cahier des charges, démonstration brève de l'application et bilan du projet.

Il indiquera également les éléments dont il est particulièrement fier et qui pourraient mener à des bonus.

Suivra ensuite une séance de **questions/réponses** pour vérifier que l'étudiant maitrise bien sa réalisation et les concepts du cours.

Evaluation

Le projet sera évalué sur base d'une grille critériée fournie ci-jointe.

Pour l'ActA théorie, la base de la note sera constituée par le projet, ajustée sur base des discussions sur la théorie lors de la défense.

Pour l'ActA TP, même chose, sur base cette fois d'un éventuel supplément de développement.