Feuille de travaux pratiques n° 1

Projet de Développement Logiciel

UML, OCL, Z, preuve, Java, Tests, Intégration

1 Prérequis

Pour réaliser ce projet, l'étudiant doit maîtriser les langages et outils suivants :

1. UML, le langage de modélisation unifié.
2. OCL, le langage d'expression de contraintes de UML.
3. Z, le langage de spécification formelle et Z-EVES un environnement de vérification et preuve.
4. Un langage à objets comme Java, Smalltalk, Python, Scala, Groovy, Ruby ou Kotlin.
5. Un outil de test unitaire de type xUnit, associé à ce langage.
6. Apache Maven, l'outil pour la gestion et l'automatisation de production des projets logiciels Java, ou son similaire pour les autres langages : rake, setuptools, etc.
7. Un environnement d'automatisation de tests unitaires de type xUnit.

2 Le cas d'étude

Le cas d'étude est détaillé en annexe sur Madoc.
Le cas peut varier d'un groupe à l'autre.

3 Objectif du travail

Il s'agit de concevoir et mettre en œuvre un logiciel pour le cas d'étude, dont l'analyse du domaine et les spécifications des exigences logicielles sont fournies en annexe. La conception sera composée de trois parties, une spécification formelle de la partie critique, la conception préliminaire (spécification des composants) et la conception détaillée.

La conception sera utilisée pour spécifier la mise en œuvre de la solution. Elle doit prioriser l'évolutivité : l'ajout de nouvelles formules de conversion doit être simple, sans la modification du code source. La conception comprendra un diagramme de classes, des diagrammes d'objet et pour les opérations complexes, des diagrammes de séquence et d'activités.

La mise en œuvre doit respecter les exigences décrites dans le document SEL fourni en annexe.

4 Architecture technique

Les concepteurs sont libres de choisir l'architecture technique qui leur convient. Ils devront toutefois motiver leur choix.

5 Travail à rendre

Le projet se compose de trois livrables :

1. Le dossier de spécification formelle.
2. Le dossier de conception.
3. Le code source.

Le dossier de spécification formelle comprend à la fois la spécification et les preuve des propriétés (obligations de preuves, propriété de sûreté). Le dossier de conception se divise en deux parties : la spécification des composants et la conception détaillée. Le dossier de conception se divise en deux parties : la spécification des composants et la conception détaillée. Il doit être organisé en fonction de la solution proposée (réservation d'une salle, gestion du matériel, etc.) et non comme une présentation d'UML (Diagrammes de séquence, diagrammes d'activités, etc.). Les diagrammes UML doivent être précis et cohérents entre eux.

Le code source doit être simple à comprendre et à évaluer : il doit être possible de les compiler et d'exécuter les tests unitaires à partir de la ligne de commandes.

6 Critères d'évaluation

Les critères qui seront utilisés pour évaluer le travail rendu son les suivants :

1. Le respect des exigences logicielles.
2. La qualité du dossier de spécification rendu.
3. La qualité et la pertinence des preuves.
4. La qualité du dossier de conception rendu.
5. La précision, la correction et la clarté de la conception : diagrammes, descriptions textuelles, etc..
6. La qualité et la pertinence des tests.
7. La cohérence entre la conception et le code.
8. La qualité du code.

7 Organisation

1. La réalisation du projet sera faite par des groupes de 3 personnes.
2. Trois livraisons sont prévues.
3. Chaque groupe déposera ses livrables sur le serveur Madoc : `http://madoc.univ-nantes.fr/`, **exclusivement**.
4. Les rapports seront rendus en format PDF, **exclusivement**.
5. Le code Z/Eves sera fourni dans une archive lisible sous Windows.
6. Le code source et les tests seront rendus en un fichier archive `tar.gz`, qui contiendra aussi un fichier `pom.xml` (Maven) ou `build.xml` (Apache Ant) permettant la compilation du système et le lancement de la validation.
7. Les fichiers Java compilés (*.class) ainsi que les éventuelles bibliothèques utilisées ne doivent pas être rendus avec le code source.