Examen de Génie logiciel avancé

Note : Vous avez 3 heures.

Rédigez chacune des deux parties sur des feuilles indépendantes.

Première partie

Ce sujet porte sur la conception d'un environnement de développement intégré de nouvelle génération, nommé Absence. Le logiciel à concevoir est à la base un éditeur de texte, capable en plus de gérer les activités typiques de la programmation déjà supportées par les concurrents (p.ex. Eclipse). Une liste non exhaustive de ces activités, que vous avez déjà rencontrées dans votre expérience comme développeurs, comprend :

- la coloration syntaxique
- la complétion automatique de membres de structures, classes, etc.
- la compilation, en appelant un outil externe
- l'exécution de batteries de test, en appelant aussi un outil externe
- le débogage, à travers activités comme l'exécution pas à pas et l'observation des valeurs de variables
- l'aide à la refactorisation, et plus particulièrement au renommage sûr de noms de variables

Le comportement correct de ces activités dépend du langage de programmation utilisé. Absence vise à supporter nativement les langages de programmation les plus populaires, comme Java, C, C++, Python, et OCaml, mais veut aussi permettre à la communauté d'utilisateurs d'étendre les langages supportés avec un système de greffons.

Ce qui rend Absence révolutionnaire est le support pour la pratique agile de programmation en binôme ("pair programming") à distance sur Internet. Dans l'interface d'Absence, l'utilisateur devra pouvoir d'abord exécuter un serveur Absence et/ou demander de se connecter aux serveurs Absence de ses collègues. Pour chaque requête entrante, l'utilisateur du serveur devra pouvoir vérifier l'identité d'un collègue et ensuite accepter la connexion. Une fois la connexion établie, toutes modifications effectuées par un des collègues dans la session seront immédiatement visibles par les autres. La communication en chat textuelle devra aussi être possible.

Exercice 1 (Spécification informelle des charges) *Votre tâche est de concevoir le logiciel Absence en respectant les meilleures pratiques de réutilisation du code et d'extensibilité. Le système sera développé dans un langage de programmation à objets.*

1. *Quelle est la frontière du système ?*
2. *Quelles sont les charges fonctionnelles ? Quelles sont les charges non fonctionnelles ?*
3. *Quelles sont les charges liées au domaine d'activité ?*
4. *D'après vous, quels sont les acteurs principaux de ce système ?*
5. *Donner le diagramme (en syntaxe UML) des cas d'utilisation ("use case") principaux pour Absence et expliquer ses différentes composantes.*

6. Donner le diagramme (en syntaxe UML) de séquence correspondant à la création d'une connexion entre deux instances d'Absence

Justifiez vos réponses.

Exercice 2 (Conception à objet)

1. Qu'est-ce qu'une relation de généralisation et comment est elle représentée dans un diagramme UML ? Quelles annotations sont possibles pour une relation de généralisation ?
2. Donner le diagramme de classe du système, en syntaxe UML, en incluant les attributs principaux et les relations entre classes.
3. On désire étendre le système dans plusieurs directions énumérées ci-après. Pour chacune de ces extensions, indiquer si elle est facilitée par la modélisation proposée. Dans le cas contraire, proposez une modélisation alternative qui s'appuie sur un patron de conception et qui aurait simplifié cette extension. Dans les deux cas, montrer les classes fondamentales de la modélisation nécessaires à supporter l'extension.

Justifiez vos réponses.

- (a) **Génération du code.** Nous voulons étendre le système de greffons pour permettre de décrire de transformations de code à la fois simple (p.ex. supprimer les commentaires) et complexes (p.ex. obfuscation de code ou même compilation personnalisée).
 - (b) **Commentaires externes.** Nous voulons stocker dans des fichiers différents des fichiers sources des commentaires écrits par les collègues qui travaillent sur un même fichier. Cela permettre de mettre en ouvre des revues de code et améliorer la qualité du code.
4. Présenter le patrons de conception Visitor. Discuter si et comment, dans le cadre du logiciel Absence et ses extensions, ils peut être appliqués.

Deuxième partie

Exercice 3 (Types abstraits) Une **double-queue** est une queue que l'on peut augmenter ou diminuer à la fois par la gauche et par la droite. Elle contient des éléments de type **element**. Modéliser un type abstrait de données **double-queue** (fonctions de construction, fonctions d'accès, fonctions de test, etc).

Exercice 4 (Spécification en B) Soit **PAYS** l'ensemble de tous les pays, **LANGUES** l'ensemble de toutes les langues. Donner des expressions du langage de spécification B pour modéliser les faits suivants :

- **parle** est une relation associant un pays à ses langues officielles. .
- **UE** est l'ensemble de tous les pays de l'Union Européenne.
- **anglophones** est l'ensemble de tous les pays anglophones.
- **francophones-hors-ue** est l'ensemble de tous les pays francophones hors de l'union européenne.
- **plus-d'une** est l'ensemble de tous les pays possédant plus d'une langue officielle.
- **meme-langage** est un ensemble de pays possédant une même langue.