

Génie Logiciel et Projet – Contrôle écrit – 21 mars 2023

Durée : 120 minutes - Une seule feuille A4 recto-verso (manuscrite ou imprimée) autorisée

Exercice 1. Questions basées sur les 6 programmes vus en cours (12 points total)

1.1 Considérons le programme « Chronomètre graphique », expliquez **avec un schéma** les avantages à avoir fait la classe « Dashboard » (la sous-fenêtre) au lieu de gérer tous les aspects graphiques dans la classe de la fenêtre principale (ChronometerGUI). (2 points)

1.2 Considérons le programme « Tree » version 1, illustrez avec un **schéma** comment les classes TreeBuilder, TreeNodeFactory et VariableRepository travaillent **techniquement ensemble**. (2 points)

1.3 Considérons le programme « Aircraft », **sans** écrire de programme, **listez** les modifications concrètes à effectuer sur le programme existant (tous les endroits où il faut mais de façon concise) pour un nouvel aspect à implémenter : A chaque tour, chaque ennemi a 40% de chance de se multiplier. Si c'est le cas, il crée une copie de lui-même qui se trouvera sur la même ligne dans une case différente que celle dans laquelle se trouve l'ennemi original. Cette copie avancera avec la même allure. En revanche, on n'autorise jamais 4 ou plus d'ennemis sur la même ligne. Les copies d'ennemis sont toujours en couleur « vert ». (2 points)

1.4 Considérons le programme « Tree » version 2 (**celui avec « visitor »**), sans écrire de programme, **listez** les modifications concrètes à effectuer sur le programme existant (tous les endroits où il faut mais de façon concise), si on doit gérer une nouvelle opération « Modulo » (qui aura la même priorité que l'opération « Multiplication »). (2 points)

1.5 Considérons le programme « Tree » version 3, illustrez avec un **schéma** annoté le fonctionnement du programme avec Log4j pour gérer les journaux (traces), en mentionnant les types de classes et fichiers concernés. (2 points)

1.6 Inspirez-vous du programme « Train », illustrez de **façon générale** avec un **schéma** comment un programme gère l'accès concurrent à une source partagée, à l'aide des méthodes « wait » et « notify ». (2 points)

Exercice 2. Conception de logiciel (8 points total)

On souhaite réaliser un logiciel qui simule le contrôle du trafic aérien (*ATC : Air Traffic Control*). L'ATC est un service rendu par des contrôleurs aériens aux aéronefs afin d'aider à l'exécution sûre, rapide et efficaces des vols. Le service est assuré dans les buts suivants : 1) Prévenir les collisions entre aéronefs ou entre un aéronef et un obstacle 2) Accélérer et ordonner la circulation aérienne. L'objectif est ainsi de réaliser une simulation logicielle du trafic aérien en temps réel. Le logiciel fournit une IHM graphique en 2D (vue aérienne verticale) simplifiée permettant de visualiser l'état des avions en un espace aérien fixe. On suppose que chaque aéronef a plusieurs pistes (pour décollage ou atterrissage).

Les avions circulent entre les aéronefs en suivant les lignes de vol prédéfinies. Une ligne de vol est composée de différents allers-retours entre deux aéronefs et il y a plusieurs vols par jour sur la ligne. Chaque avion en l'air a une altitude qui est une information importante pour éviter la collision quand deux avions se croisent dans une zone en l'air. Les obstacles sont principalement les montagnes qui ont différentes altitudes. Les problèmes (incidents) des avions surviennent aléatoirement qui obligent une gestion de l'atterrissage dans un aéronef plus proche que celui comme destination originale.

L'utilisation souhaite pouvoir démarrer la simulation avec quelques paramètres simples. Il peut visualiser surtout l'espace aérien contenant les avions en l'air, avec toutes les informations importantes et pertinentes. Il souhaite également pouvoir gérer l'atterrissage urgent le cas échéant sur une piste d'un aéronef.

2.1 Classes de données : modélisez les **données nécessaires** pour la réalisation de ce logiciel avec un diagramme de classe UML (à dessiner en orientation « paysage » si nécessaire). **Dans ce digramme, vous mentionnerez uniquement les noms et attributs des classes, ainsi que les relations entre les classes.** Vous n'y indiquerez **aucune méthode, ni constructeur**. (4 points)

2.2 Proposez une conception de l'IHM graphique de ce logiciel. **Vous illustrez avec un schéma bien annoté** (à dessiner en orientation « paysage » si nécessaire), les détails de différentes parties de votre conception. (4 points)