# AP4B - Examen final du 20 Juin 2023

## Durée 2h – aucun document autorisé

**Exercice 1 (3 pts).** **Pattern Composite.** Considérons le schéma de conception Composite. Il permet la composition récursive d’objets, de représenter une hiérarchie d'objets, d’ignorer la différence entre un composant simple et un composant en contenant d'autres (interface uniforme).

1. Donner le modèle de classe UML de ce pattern conception.
2. Donner un programme JAVA du pattern appliqué à des formes géométriques Cercle, Carré, Groupe de formes, chaque forme affichant un message approprié. Donner un exemple de main(). On doit pouvoir créer des formes, en ajouter à un composite, qui peut lui-même contenir d’autres composites.

**Exercice 2 (4 pts).** **Inscription aux UV.** Dans une université de technologie, une tâche complexe réalisée en début de semestre par le service scolarité est de produire l’emploi du temps des enseignants et étudiants. Cela est réalisé automatiquement à partir des informations enregistrées dans le système d’information. Dans un premier temps les enseignants responsables des UV (1 seul par UV) saisissent les prévisions d’interventions de chaque enseignant de l’UV : nombre de séances d’enseignant dans chaque modalité CM/TD/TP. Ensuite, une phase d’inscription pédagogique permet aux étudiants de s’inscrire dans 6 UV choisies. Sur la base de ces informations, le service pédagogique génère les emplois du temps hebdomadaires effectifs et met à jour le SI. Il se peut que des CM/TD/TP soient dédoublés en sous-groupes (C1, C2, TD1, TD2, TD3 …) selon les effectifs. Le système doit pouvoir retrouver les interventions exactes pour chaque enseignant dans chaque UV, et déterminer la salle et l’horaire associés. De même, les affectations des étudiants aux groupes de CM/TD/TP sont déterminées. Pour chaque étudiant et chaque UV, une note globale est déterminée. Pour chaque enseignant et chaque cours, le nombre séances est déterminé. Une entité/classe « Cours » servira à modéliser une séance effective de CM/TD/TP, un jour de semaine, à un horaire précis, dans une salle donnée et fera le lien avec les autres entités.

1. Donner un digramme de cas d’utilisation du SI.
2. Donner un diagramme de classe UML permettant de répondre aux besoins.

**Exercice 3 (4 pts).** **Téléphone portable.** On considère un téléphone portable avec ses fonctions standards de communication. La touche « verte » du téléphone permet d’accepter la réception d’un appel ou de lancer un appel vers un destinataire, sachant que le clavier alphanumérique comporte les dix chiffres de « 0 » à « 9 » pour la numérotation préalable. En plus de cela, un ensemble de menus sont fournis. On se déplace dans un menu à l’aide des boutons « en bas », « en haut » et « ok ». Une fois entré dans le menu « répertoire », l’utilisateur se sert des mêmes touches pour se déplacer, choisir un nom, un numéro de téléphone, puis valider pour lancer l’appel. Le but est de produire un diagramme Etat/Transition spécifiant les aspects dynamiques des différents modes d’utilisation du téléphone portable en fonction des séquences d’appui-touche. On ne détaillera que les modes précisés ci-dessus (veille, numérotation, appel, réception, saisie dans le répertoire). On attachera une importance particulière à proposer un diagramme bien lisible en procédant par étapes successives (approche descendante) d’imbrication des états.

1. Donner un diagramme Etat/Transition hiérarchisé présentant le système dans sa globalité sans entrer ici dans le détail des sous-états principaux.
2. Préciser les sous-automates principaux.

**Exercice 5 (5 pts). Réseau de transport global.** Donner un modèle de classes UML d'un réseau de transport global couvrant un territoire géographique. Sur le territoire est répartie une population dont les individus utilisent les moyens de transport à leur disposition. Le territoire est maillé par un ensemble de routes interconnectées. Il est également couvert par un réseau d’antennes radio-cellulaires permettant la communication par téléphones mobiles et le suivi des déplacements de personnes. Des lignes ferroviaires passent par des gares d'échanges ou des lieux importants à desservir. Les usagers du réseau de transport sont des piétons, des cyclistes, des automobiles, des camions, etc. Une personne peut évidemment aller à pied, en vélo, en auto. Les véhicules circulent sur les routes, tandis que les camions peuvent également être transbordés par chemin de fer (ferroutage). Les compagnies de transport envoient des ordres à leurs flottes de véhicules pour desservir des demandes particulières de transport de marchandises et autres. Les trains, gérés par une grande compagnie de transport, effectuent des liaisons et s'arrêtent dans les gares à des horaires prédéfinis. Le réseau routier comporte des autoroutes, surveillées par des unités de sécurité et de secours, il comporte des routes nationales et départementales surveillées par la gendarmerie, etc...

**Exercice 5 (4 pts). Philosophes dînant.** Dans le problème des "philosophes dînant", il y a cinq philosophes autour d’une table et 5 fourchettes posées sur la table à gauche et à droite de chaque philosophe. Il s’agit d’un problème typique d’allocation de ressource. Chaque philosophe a accès à deux fourchettes situées à sa gauche et sa droite. Une contrainte est qu’un philosophe ne peut manger que s’il a saisi ses deux fourchettes adjacentes, gauche et droite (une dans chaque main). Lorsqu’il a mangé, il repose les 2 fourchettes puis entame une discussion. Lorsqu’il a fini de parler, il décide de manger à nouveau. Et le cycle se répète. On veut réaliser un programme de simulation Java qui simule les philosophes dînant par des threads synchronisés. Chaque philosophe doit tenter de s’emparer des 2 fourchettes, s’il réussit, il mange, repose les deux fourchettes, puis discute. Et le cycle manger-discuter reprend, sans fin. Notez que tous les philosophes doivent manger équitablement au bout d’un temps fini.

Une solution consiste à se synchroniser sur un seul et unique objet partagé (prenons la Table) qui gère l’accès aux fourchettes et mémorise l’état de chaque fourchette (posée, prise). La saisie des 2 fourchettes est simultanée si elles sont libres en même temps. Le philosophe mange puis repose les deux fourchettes en même temps.

1. Donner un modèle de classe UML du programme Java et représenter les threads.
2. Donner le programme JAVA de simulation complet (classe Table, classe Philosophe, fonction main(), mettre en évidence la gestion de la synchronisation.