Labo 5 – Révisions avant l'AI

Objectifs (2h)

Au terme de ce laboratoire, tu auras revu les principales tâches qui te seront demandées durant l'AI :

→ Concevoir une synthèse sur le choix des collections ;
→ Trouver les pré- et postconditions d'un algorithme ;
→ Rédiger un plan de test exhaustif ;
→ Justifier le choix d'une interface de Collection du Java Collections Framework (List, Queue, Set et Map) sur base de leurs principales fonctionnalités ;
→ Justifier le choix de l'implémentation concrète d'une Collection du Java Collections Framework (List, Queue, Set et Map) sur base des complexités des principales méthodes utilisées par un programme ;
→ Calculer la complexité d'un programme.

Préambule

Comme tu as pu le voir jusqu'ici, les laboratoires d'algorithmiques te laissent libre de choisir les exercices dans lequel tu ressens le plus de difficulté ou qui te t'intéressent le plus. Ce labo ne fait pas exception. Voici comment il est conseillé de se préparer à l'AI :

- La première partie t'invite à concevoir une synthèse **personnelle** sur le choix des collections. Elle t'aidera fortement durant toutes les itérations de l'activité intégrative ;
- La seconde partie te propose des exercices supplémentaires sur des notions déjà vues. Comme d'habitude, focalise-toi sur les exercices dans lesquels tu rencontres des difficultés : il n'y a pas assez de 2 heures pour tout faire réaliser. Dans la suite de tes études (et de ta vie professionnelle), savoir diagnostiquer toi-même comment t'améliorer (le terme technique, c'est « **s'autoréguler** ») est une qualité essentielle de l'expert. Si tu as des questions ou des doutes , pose-les à ton encadrant de labo, il est là pour t'aider.

Concevoir une synthèse (1h)

L'idée de cette partie est de créer un document de la taille d'un écran ou d'une page A4 qui résume ce que **tu** dois faire quand **tu** es confronté au choix de collections. On parle bien de toi, qui lit cet énoncé, pas de Christiane, Arnaud², Cyril, Nicolas, François, Jean ou Simon ! Ce doit donc être un travail **personnel** : il te sera d'autant plus utile.

Quelle forme doit-il prendre ? Ce peut être une carte mentale (« *mindmap* ») ou un arbre de décision (pourquoi pas ?) ou encore une « *check-list* » si le monde de l'aviation t'inspire. N'hésite pas à imaginer ta propre solution.

Au minimum, ta réalisation devrait te permettre de répondre aux questions suivantes :

- *Sur quel·s critère·s dois-je choisir une interface de collection ?* N'hésite pas à y inclure des exemples qui te parlent ou des pièges dans lesquels tu serais déjà tombé·e afin de ne pas te faire avoir une seconde fois.
- *Mon interface choisie, sur quel·s critère·s dois-je choisir l'implémentation ?* Tu pourrais par exemple faire figurer les complexités de certaines méthodes souvent utilisées avec ces collections (lesquelles ? Base-toi sur ton expérience, tes habitudes ou tes besoins !).

N'hésite pas à demander l'avis de ton responsable de laboratoire. Veille surtout à ne pas inclure de « *fakenews* » dans ta production. Ici encore, le responsable de ton labo peut t'aider. Je te conseille de te méfier des synthèses toutes faites que tu pourrais trouver ailleurs, en particulier venant de sites non spécialisés comme les réseaux sociaux. En outre, tu passerais à côté de l'intérêt de l'exercice.

Test de ta synthèse

Ta synthèse est censé t'aider à choisir correctement des collections. Valide son utilité en justifiant quelques choix de collections, au choix parmi (fais-en quelques-unes) :

- → Reviens sur les labos de POO ou d'algorithmique qui utilisaient des tableaux et détermine par quelles collections on peut remplacer ces tableaux ;
- → Tu peux aussi valider les choix de collections du labo 3 d'algorithmique (ta synthèse devrait te permettre d'arriver aux mêmes conclusions que ce que recommande les énoncés) ;
- → Tu peux aussi valider les choix de collections pour la cave à vin (POO/Labo 5), les catalogue des produits, caddie et collection de règles (POO Labo 6 / Interro), etc.

Exercices de révision

Voici des exercices de révision. Nous te conseillons de pratiquer les sujets qui te posent des problèmes.

Trouver les pré- et postconditions d'un algorithme

Fournis les préconditions et postconditions des méthodes publiques des classes que tu as rédigées lors de l'interrogation formative de POO. En particulier, peut-on toujours appeler la méthode `BaseRule.applyTo(Cart)` ? Que se passe-t-il si on le fait ?

Rédiger un plan de test exhaustif

Établis un plan de test le plus complet possible pour la méthode `algo.labo04.Mot.contientLettre`. Après avoir trouvé le bug[†], propose une correction de cette méthode.

Calculer la complexité d'un programme

Dans la dernière leçon d'algorithmique (sur la programmation dynamique), nous avons revu le problème de la mine d'or. Calcule les complexités temporelles pour les fonctions `solve(int[][] mine)` de toutes les implémentations de `GoldmineSolver` (il y en a trois).

Pour t'aider, voici quatre possibilités (il y en a donc une en trop). Dans la suite, P est la profondeur de la mine (le nombre de lignes du tableau) et Lsa largeur (le nombre de colonnes).

1. $O(P + L)$
2. $O(P \times L)$
3. $O(P3^L)$
4. $O(P^2L)$

[†] Le programme du pendu ne provoque pas le bug mais la classe `Mot` peut être mis dans un état incohérent si elle était utilisée différemment. *J'aurais pu* moi-même le voir si je l'avais correctement testée. À quelque chose, malheur est bon : ça me donne un bon énoncé à te soumettre.