

## Instructions importantes

**Ceci est un devoir effectué en groupe.** Vous travaillerez avec les membres de votre groupe de laboratoire sur ce devoir. Soumettez tous les fichiers du devoir sur Brightspace.

## Environnement de développement intégré (IDE)

Eclipse est présenté dans le deuxième tutoriel. Vous pouvez utiliser Eclipse n'importe quel environnement de développement intégré pour résoudre cette tâche.

## Exercice 1 - Analyse de diverses variantes d'un design

Veuillez lire la section 2.9 du manuel, à partir de la page 57. **J'ai posté un scan des pages pertinentes si vous n'avez pas le livre.**

### Procédure

1. Créez un projet sur votre IDE. Importez les classes PointCP et PointCPTest de la répertoire design1 (original) à votre projet (voir le répertoire compressé associé avec ce devoir) .
2. Modifiez la classe PointCP pour implémenter les conceptions décrites dans le tableau 1. Toutes les modifications doivent être effectuées dans le répertoire /pointcp. En modifiant la version Design 1 (original), implémentez les designs suivants.
  - design 2 (éditez le code dans le répertoire /design2). Ajoutez votre code dans PointCP2.java.
  - design 3 (éditez le code dans le répertoire /design3). Ajoutez votre code dans PointCP3.java.
  - design 4 (modifiez le code dans le répertoire /design4). Ajoutez votre code dans PointCP4.java.
  - design 5 (éditez le code dans le répertoire /design5). Ajoutez votre code dans PointCP5.java et modifiez vos PointCP2.java et PointCP3.java existants (pour les adapter aux exigences de ce design).
3. Modifiez la classe PointCPTest pour vous permettre de tester les designs que vous avez développés. Faites une série complète de tests pour vous assurer que vos classes fonctionnent correctement. Vous pouvez implémenter d'autres classes pour faciliter la tâche ou ajouter des méthodes supplémentaires à la classe PointCPTest (si vous le jugez nécessaire).
4. **Dans un document texte** (par exemple, un document MS Word), répondez aux exercices suivants de la section 2.9 (page 57) du manuel: **E26, E28, E29, E30. J'ai publié cette section du manuel avec ce devoir.** **Dans un document texte** (par exemple un document MS Word), répondez aux exercices suivants du livre: **E26, E28, E29, E30.** Pour répondre à **E28, E29 et E30**, pour les designs 1 et 5, créez un grand nombre d'instances PointCP aléatoires (c'est-à-dire les coordonnées et le type de système de coordonnées choisis au hasard). Dans une boucle, appelez toutes les méthodes publiques sur chaque instance. Calculez le temps écoulé en millisecondes pour terminer l'exécution de la boucle. Assurez-vous que votre programme s'exécute pendant environ 10 secondes, afin d'obtenir une bonne mesure des performances. Exécutez le test pour chaque design plusieurs fois pour vous assurer que

vos résultats sont cohérents et indiquez les mesures de temps médian, minimum et maximum obtenues. De plus, dans le même document texte, fournissez les informations suivantes:

- a. Comment est-ce que vous avez fait les tests ;
- b. Échantillon des sorties lorsque vous avez exécuté les tests ; et
- c. Discussion des résultats.

Tableau 1 - Conceptions PointCP

Design	Comment les coordonnées cartésiennes sont calculées	Comment les coordonnées polaires sont calculées
Design 2: stocker uniquement les coordonnées polaires	Calculé à la demande, mais non stocké	Simplement retourné
Design 3: stocker uniquement les coordonnées cartésiennes	Simplement retourné	Calculé à la demande, mais non stocké
Design 4: stocker les deux types de coordonnées à l'aide de quatre variables d'instance	Simplement retourné	Simplement retourné
Design 5: Classe abstraite avec les designs 2 et 3 comme sous-classes	Dépend de la classe de concrète utilisée	Dépend de la classe de concrète utilisée

## Exercice 2 - Tableaux

Pour cet exercice, vous allez comparer les performances de LinkedList, ArrayList et tableau (array) java. Vous ajouterez un grand nombre d'éléments aux listes (LinkedList et ArrayList ) et au tableau pour évaluer leurs performances.

### Procédure

1. Vous découvrirez le nombre d'éléments à ajouter aux listes et au tableau via une méthode d'erreur d'essai :
  - a. Ajoutez un nombre X de caractères avec une valeur comprise entre «a» et «z» à une ArrayList . X est un grand nombre avec lequel vous avez choisi de commencer (par exemple 10 000).
  - b. Mesurez le temps nécessaire pour ajouter les éléments à la ArrayList .
  - c. S'il faut moins de 10 secondes pour remplir la ArrayList , augmentez la valeur de X et revenez à l'étape a, sinon, passez à l'étape 2.
2. Remplissez ArrayList, LinkedList et tableau avec un nombre X de caractères avec une valeur comprise entre «a» et «z». Le ArrayList et LinkedList peuvent être créés en ajoutant successivement des éléments et leur permettant de grandir, alors que le tableau doit être créé à sa pleine taille, puis rempli avec son contenu.

3. Comparez les performances des trois structures de données comme suit:
  - a. Mesurer le temps (en millisecondes) nécessaire pour additionner les éléments dans ArrayList
  - b. Mesurer le temps (en millisecondes) nécessaire pour additionner les éléments de la LinkedList
  - c. Mesurer le temps (en millisecondes) nécessaire pour additionner les éléments du tableau
4. **Dans un document texte** (par exemple **un document** MS Word), présentez vos résultats dans des tableaux appropriés. Tirez des conclusions d'une analyse de vos résultats et développez des recommandations aux concepteurs.