

<p align="center">Cours 420-EAD-LI POOIV Hiver 2015 Cégep Limoilou Département d'informatique</p> <p>Professeur : Martin Simoneau</p>	<p align="center">Laboratoire 1</p> <p align="center">Modèle MVC, Observateur (7%) Fabriques et dépendances</p>
--	---

Nom1 : _____

Nom2 : _____

Objectifs

- Comprendre l'importance des dépendances
- Comprendre le patron de design Observer
- Comprendre le patron de design MVC

À remettre :

- Le travail sera remis sur Léa. Compressez chacun des 3 projets et ajoutez un fichier Word comprenant la réponse aux questions de la dernière section. Assurez-vous de mettre votre nom dans chacun de vos fichiers (dans la javadoc de vos classes)
- Mettre vos noms à la fin de chaque artifactID dans les fichiers POM.xml.
- Vos 3 projets Maven complets et compressés dans une archive **portant votre nom complet**.
- Vous avez 2 semaines pour faire le travail.
- 5% sont attribués pour remettre les Exercices 1, et 2
- Votre projet doit contenir les fichiers .jar générés par MAVEN

Contexte :

- Remettre vos projets sur Léa
- Le laboratoire peut être fait en équipe de 2 personnes (maximum).
- Le modèle et le contrôleur seront évalués dans le cadre de POOIV
- La vue javaFX sera faite en parallèle dans le cadre du cours Multimedia.

À faire

On vous demande de fabriquer une application simple en respectant les principes MVC vus dans le cours.

Votre application doit effectuer les tâches suivantes:

- Lire un fichier contenant des mots.
- À l'aide d'une classe fabrique singleton,
 - créez une classe *Mot* qui contient
 - un mot (1 seul mot)
 - une définition (max 20 mots configurables par propriétés)
 - le nom d'un fichier contenant une image (en format jpeg) qui est associée à la définition (peut être null)

- la date de saisie du mot (allez voir la classe `LocaleDate`)
 - la date de la dernière modification (`LocaleDate` aussi)
- La fabrique est la seule instance qui a le droit de consulter le fichier de propriétés.
- La fabrique doit avoir une méthode pour créer les ~300 000 mots contenus dans le fichier qui vous a été fourni avec le travail. La date est saisie sur le moment. La définition et l'image sont vides par défaut.
- Votre application doit offrir les fonctionnalités suivantes:
 - Créer tous les mots du fichier
 - Permettre d'ajouter et d'effacer un mot à la fois.
 - Permettre de modifier les instances de *Mot* (la date ne peut pas être changée)
 - La définition
 - Le mot
 - L'image associée
 - Moteur de Recherche qui recherche:
 - le texte du mot.
 - Les mots saisis avant ou après la date de saisi ou la date de modification.
 - Les mots qui ont une image associée ou non.
 - Rechercher tous les mots qui contiennent le groupe de lettres demandées (minimum de 3 lettres).
 - Pour vous démarquer (optionnel pour l'instant):
 - Utilisez des wildcards dans le critère de recherche (* remplace un groupe de lettres ? remplace une seule lettre).
 - Optimisez votre moteur de recherche en utilisant des structures de données efficaces et appropriées.
 - Faire une recherche respectant plusieurs critères simultanément (exemple le nom est « automobile et la date est après).

Contraintes

- Vous devez faire au moins un Observateur. Vous pouvez le faire vous même ou avec les classes et interfaces du JDK. L'observateur doit être utilisé de façon appropriée.

Critères d'évaluation

- Respect intégral du modèle MVC
- Utilisation adéquate de MAVEN
- Découpage approprié du code (classes et méthodes)
- Fonctionnement du programme.
- Efficacité du moteur de rechercher
- Respect des fonctionnalités demandées.
- Code bien documenté et formaté

- javadoc conforme.