

<p>Cours 420-203-RE  <i>Développement de programmes dans un environnement graphique</i>  Automne 2020</p> <p>Cégep Limoilou, département d'informatique</p> <p>Professeur : Martin Simoneau</p>	<p>TP3</p> <p>MVC JavaFX  Fichiers FXML  Liaisons de propriétés</p> <p><b>12 %</b></p>
---	--

## OBJECTIFS

- Créer une application MVC complète
- Utiliser les styles CSS
- Utiliser une approche descriptive pour construire l'interface (FXML)
- Utiliser les liaisons et les propriétés

## Contexte :

- **Doit être fait en équipe de 2 personnes.**

On vous demande de concevoir une application de type calculatrice bonifiée. Le projet que l'on vous a fourni est vide, mais il est préconfiguré pour utiliser la librairie MXParser

## ÉTAPE 1 : COMPLÉTER LA VUE

Vous devez :

- Concevoir votre interface UI avec *SceneBuilder*. L'interface devra utiliser 2 ou 3 fenêtres. Elle comprendra:
  - Une zone d'affichage suffisamment large et avec une police claire et assez grande
  - Les touches numériques
    - Numéro de 0 à 9
    - La virgule
  - Les opérations
    - Les 2 parenthèses : ( )
    - Les opérations arithmétiques : + - \* /
  - Les opérations en mémoire
    - Un bouton *M get* pour mettre la valeur affichée en mémoire
    - Un bouton *M set* pour remettre la valeur en mémoire dans la zone d'affichage
  - Touches de fonctions

- 4 touches de fonctions configurables
- Une zone de conversion d'unités comprenant
  - Un Label pour recevoir la valeur convertie
  - 1 comboBox pour choisir le type de conversion à faire.
    - Mètre – pied
    - Aucune conversion
    - Kilogramme – livre
    - 2 autres conversions à votre goût
  - Un bouton pour ramener le contenu du label de la valeur convertie dans la zone d'affichage.
- Une zone de saisie de fonctions (ex :  $f(x)=x^2+2x-5$  ) comprenant :
  - Un contrôle pour saisir l'expression mathématique sous forme de chaîne de caractère
  - Un contrôle pour donner un nom à l'expression mathématique
  - Une liste pour contenir toutes les expressions saisies

## ÉTAPE 2 : CONCEVOIR VOS SQUELETTES DE CLASSES MVC

- Les vues sont :
  - les fichiers *FXML* (2 ou 3)
- Les modèles sont :
  - Un module d'enregistrement qui permet de conserver les expressions dans un fichier sur le disque.
  - Un module qui fait les calculs à partir de la librairie *MXParser* (<http://mathparser.org>). Ce module doit également gérer la valeur conservée en mémoire avec les touches *M get* et *M set*. Ce module doit finalement établir les liaisons nécessaires pour faire les conversions. Toutes les conversions sont faites par liaisons entre la zone d'affichage et le *label* de conversion.
  - Un module qui permet de saisir, éditer et effacer de nouvelles fonctions. Le module doit valider si la fonction est saisie correctement en essayant de l'évaluer. Une fonction mal saisie retournera toujours NaN. Les fonctions peuvent être conservées dans un *ObservableList<>*.
- Les contrôleurs sont :
  - Un contrôleur par vue
- Cette étape doit être faite par les 2 coéquipiers. Par la suite, il sera plus aisé de séparer le travail à faire.

## ÉTAPE 3 : PROGRAMMATION DE L'APPLICATION

- Fonctionnalités à réaliser
  - Effectuer les calculs arithmétiques de base avec des parenthèses.
  - Effectuer les conversions d'unité entre la zone d'affichage et l'étiquette de conversion selon le choix fait dans le ComboBox. La conversion doit être effectuée absolument par une liaison de propriété entre la zone d'affichage et le *label* de conversion.

- Garder une valeur en mémoire et pouvoir la récupérer lorsque l'utilisateur appuie sur les boutons de mémoire respectifs.
- Ramener la valeur convertie dans la zone de saisie lorsqu'on appuie sur le bouton
- Enregistrer et charger les fonctions saisies par l'utilisateur
- Saisir, modifier et effacer des fonctions
- Associer les fonctions spéciales à l'une des 4 touches de fonctions
- Lancer le calcul d'une fonction avec la valeur de la zone d'affichage.

## CRITÈRES D'ÉVALUATION

- Qualité du code
  - Javadoc
  - Formatage
  - Structure
  - Commentaires
  - Séparation MVC du code.
- Qualité des interfaces
  - Usage approprié des CSS
  - Interface **intuitive** et **claire (documentez-la au besoin avec des labels ou des tooltips)**
  - Interface agréable et invitante
  - Originalité
- Fonctionnement
  - Absence de bogues
  - Toutes les fonctionnalités présentes
  - Respect des contraintes (liaisons javafx, css...)

## À REMETTRE :

- Restructurer le nom du projet pour y ajouter vos initiales.
- Mettre vos noms dans l'en-tête de chaque classe
- Mettre vos noms dans les titres des fenêtres javaFX
- Remettre votre projet complet archivé en format **zip** sur Léa avant la date indiquée.
- Vous pouvez faire plusieurs remises, seule la dernière sera considérée.