

Simple Count Documentation

Etienne DEBAS

23 Novembre 2014

Table des matières

1	Choix des composants	2
1.1	Ecran	2
1.2	Les buttons	2
2	Choix du design	2
3	Conception	2
3.1	IDE	2
3.2	Package	3
3.3	Heritage	3
3.4	Pattern MVC	3

1 Choix des composants

Le package Swing de java a été utilisé pour l'interface homme machine.

1.1 Ecran

L'ecran de la calculatrice est composé de 3 JTextField permettant :

- d'afficher la totalité l'expression courante.
- d'ajouter un élément à l'expression en cours.
- d'afficher un message en cas d'erreur.

1.2 Les buttons

Les boutons de la calculatrice ont été créés grace à l'objet JButton. Pour l'insertion des JTextField et des JButton au sein de l'interface j'ai utilisé l'objet JPanel avec les conteneur suivant :

- GridLayout
- Box
- GridBagLayout

L'ensemble des composants est mis en place dans une JFrame

2 Choix du design

Trouvant le thème de base proposé par Swing trop clair et trop sobre j'ai choisi de customiser la plupart des composants de la calculatrice comprenant les JTextField et les JButton. Tous les éléments ont été initialisés avec un background DARK_GREY, avec un foreground WHITE pour la police et avec une custom font.

Les bordures des JTextField ont été supprimés afin de n'avoir aucune séparation apparente entre eux. Les boutons **Undo** et **Clear** implémentent des icons permettant de mieux se représenter leurs fonctionnalités.

3 Conception

3.1 IDE

L'IDE IntelliJ IDEA a été utilisé pour le développement de ce projet car celui ci propose des fonctionnalités très intéressante pour l'élaboration d'un projet java :

- une autocomplétion de qualité
- une intégration du versionning avec Git
- permet de développer du J2EE et de l'android

3.2 Package

Les classes ont été placées dans des packages différents afin de gagner en visibilité.

Le package **ui** regroupe les classes en rapport avec l'interface graphique.

Le package **math** regroupe l'évaluateur d'expression et les unicodes mathématiques pour l'interface graphique.

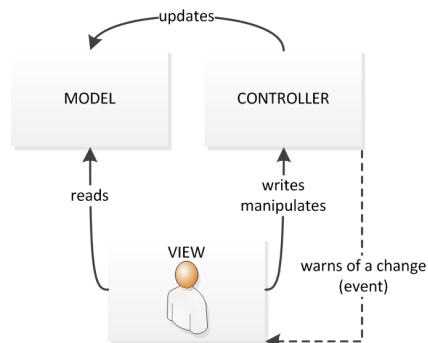
Le package **exception** contient la classe d'exception pour l'évaluateur d'expression.

Les packages **model** et **controller** regroupent les classes du pattern **MVC**.

3.3 Heritage

Afin d'éviter les répétitions de code les objets de l'IHM héritent de ceux de Swing afin d'initialiser leurs attributs dès leur construction (SimpleCountButton, SimpleCountField) Utilisé aussi pour le listener des boutons, pour l'exception mais aussi pour le pattern **MVC**.

3.4 Pattern MVC



Le pattern MVC (Modele - Vue - Controller) est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective (*Wikipedia*)

L'intégration de ce pattern au sein du projet est donc approprié.

Lorsque l'utilisateur appuie sur un bouton, le listener (Controller) est appelé puis le model update les données pour ensuite notifier la vue des changements graphique à effectuer.