

# M2GL - V & V

## TP5 - Test d'IHM

Romain LE HÔ & Yves DEMIRDJIAN

5 novembre 2012

## 1 Notes

Toutes modifications apportées dans le code sont précédées d'un commentaire informant que ces dites modifications proviennent de nous.

## 2 Tests proposés dans l'énoncé du TP

### 2.1 All text fields should be disabled at application startup.

Au lancement manuel de l'application, les champs textes semblent Disabled. Un test nous montre cependant qu'à part deux composants, tous sont des LabeledText, qui ont bien leur état isEnabled à False. Mais State et ZIP, ne sont pas des LabeledText, mais des JTextComponent.

Ceux-ci sont Disabled, cependant l'inspection ne trouve pas de label pouvant les lier au champ texte correspondant ("ZIP" ou "State"). Cela a pour conséquence de faire échouer les tests JUnit quand on les lance une fois qu'ils ont été générés.

D'après le code (AddressPanel) nous montre que les labels des différentes rubriques sont toujours déclarées avant la boîte de texte liée, sauf pour ZIP et State. On déplace ceux-ci, et relançons les inspections : cette fois ZIP et State sont considérés comme des LabeledText et leur état isEnabled à False est testable grâce à JUnit.

### 2.2 All text fields should be enabled after clicking the new button.

Ce test fonctionne parfaitement dès la première exécution. Tous les champs de texte sont Enabled après qu'on ait cliqué sur New.

### 2.3 The button for delete should be enabled when there's a selected item in the list.

Ce test, fonctionne, une fois qu'on prend garde de tester juste, dans JListLocator(), le "Nom, prénom" en paramètre et sans le numéro d'index à la suite (comme initialisé

automatiquement). (voir code)

## 2.4 The Save button should not be enabled right after clicking the New button.

Le test manuel montre que le bouton Save est bien Enabled après un New, ce qui n'est pas conforme à ce qu'on attend. Il faut donc corriger l'application pour que seule une modification entraîne l'activation du bouton Save et rendre celui-ci désactivé par défaut.

Ceci est fait en ajoutant des KeyListeners dans l'AddressPanel et une fonction `addListenerToTextFields()`, de manière à ce qu'à chaque pression d'une touche dans un `TextField`, on informe `AddressActionPanel`. Pour simplifier, seuls les champs `FirstName`, `LastName` sont obligatoires pour activer le bouton.

## 2.5 The button for Delete should not be enabled where there isn't a selected item in the list.

Le test regarde si le bouton Delete est Disabled au démarrage. Le test échoue en première exécution, car Delete est toujours activé.

De nouveau, on ajoute un `addListenerToAddressList` à `AdressListPanel` (fonction `addListenerToList()` dans `AddressListPanel`), afin d'être notifié lorsqu'un élément est sélectionné ou non.

# 3 Tests supplémentaires

## 3.1 The button for edit should not be enabled where there isnt a selected item in the list.

De la même manière que pour le test précédent avec Delete, Edit ne doit être possible que si un item de la List est sélectionné. Nous avons donc rajouté les quelques lignes manquantes dans le listener que nous avons précédemment créé afin de gérer l'état Enable du bouton.

## 3.2 The button for edit should be enabled where there is a selected item in the list.

Nous ajoutons une personne, la sauvons et nous vérifions qu'après sélection de celle-ci le bouton Edit est Enabled. Le test a été concluant immédiatement grâce au rajout du code du test précédant.

### **3.3 The button for cancel should not be enabled when we are not editing an address**

Le test a tout d'abord échoué, car le bouton Cancel était toujours activé. Nous l'avons donc désactivé par défaut puis dans la fonction `setEditable` de `AddressPanel` il nous a suffi de changer l'état `Enabled` du bouton en fonction du paramètre passé à la fonction.

### **3.4 The button for cancel should not be enabled when we are not editing an address**

Le test consiste à vérifier que le bouton est désactivé avant l'appui sur le bouton New ou après l'appui sur le bouton Cancel. Le test a tout d'abord échoué, car le bouton Cancel était toujours activé. Nous l'avons donc désactivé par défaut puis dans la fonction `setEditable` de `AddressPanel` il nous a suffi de changer l'état `Enabled` du bouton en fonction du paramètre passé à la fonction.

### **3.5 The button for cancel should be enabled when we are editing an address**

Le test a immédiatement fonctionné suite aux modifications précédemment apportées. Le test consiste à simuler l'appui sur le bouton New et vérifier que le bouton Cancel est bien activé.