

**Ceri - Université d'Avignon  
Master Informatique 2012-2013**

**M1: Génie Logiciel Avancé  
Prototypage d'interface utilisateur**

**TP1 : Prototypage simple (énoncé)**

**Objectif du TP**

L'objectif de ce TP est de réaliser le prototypage d'une application de gestion de carnets d'adresses en utilisant l'environnement de développement NetBeans. La maquette du logiciel se limitera à une version « statique », sans interactivité. Le prototype sera constitué de plusieurs formulaires présentant les choix d'ergonomie, sans interconnexion.

Lors de ce TP on vous demandera de vous familiariser avec l'outil de développement NetBeans et plus particulièrement son éditeur d'interfaces utilisateurs.

## 1. Création du projet

→ Créer un nouveau projet vierge Java de type « Java Application ». Définir un paquetage « m1.pi » qui contiendra le code source.

## 2. Construction de la fenêtre applicative

→ Créer un formulaire Swing de type **JFrame** pour le design de la fenêtre principale. Nommer ce formulaire « MainFrame ».

→ Ajouter une barre de menu avec les items suivants:

- File → New , Open..., Save, Save as..., et Exit.
- Help → About

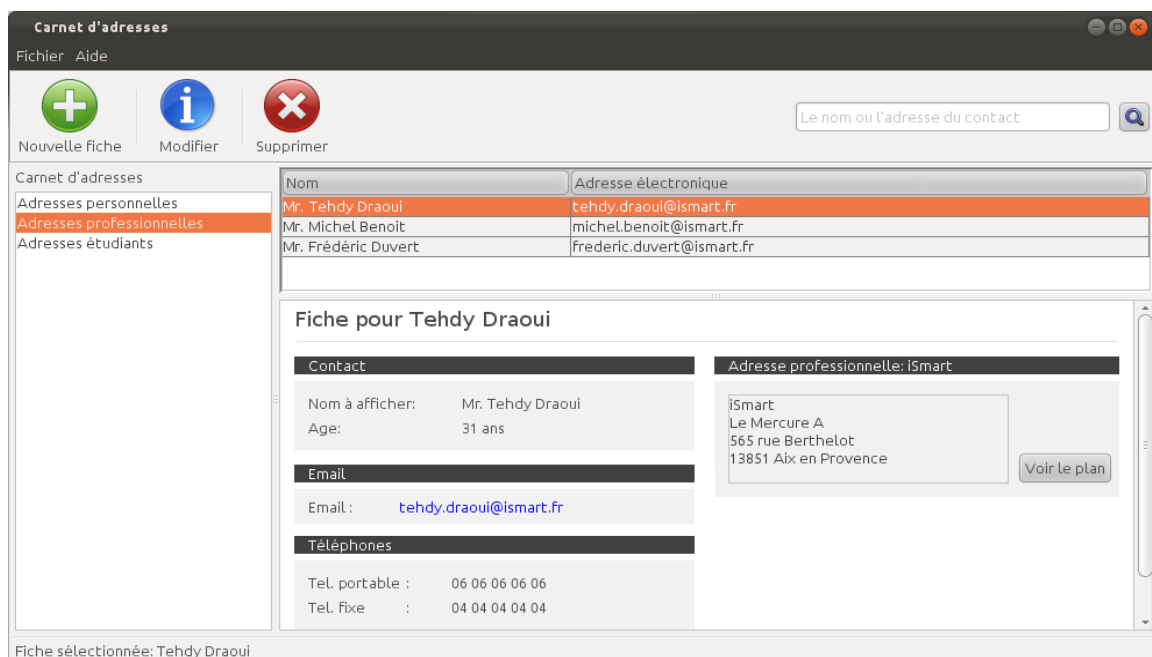
→ Ajouter une barre d'outils avec des boutons pour ajouter, modifier et supprimer une fiche, ainsi qu'un champs de recherche de fiches. Vous pouvez également ajouter ces actions dans la barre de menu.

→ Ajouter une barre d'état en bas de la fenêtre qui affichera par exemple le nom du contact sélectionné.

## 3. Élaboration de l'écran principal

→ Créer un nouveau formulaire Swing de type **JPanel** pour le design de la consultation du carnet d'adresses (zone entre la barre d'outil et la barre d'état). Nommer ce formulaire « MainPanel ».

Vous devez mettre au point l'interface de consultation des carnets d'adresses. L'interface doit être ergonomique et permettre à l'utilisateur de naviguer à travers la liste des carnets d'adresses, de parcourir le tableau des contacts du carnet d'adresses sélectionné, et d'afficher la fiche du contact sélectionné :

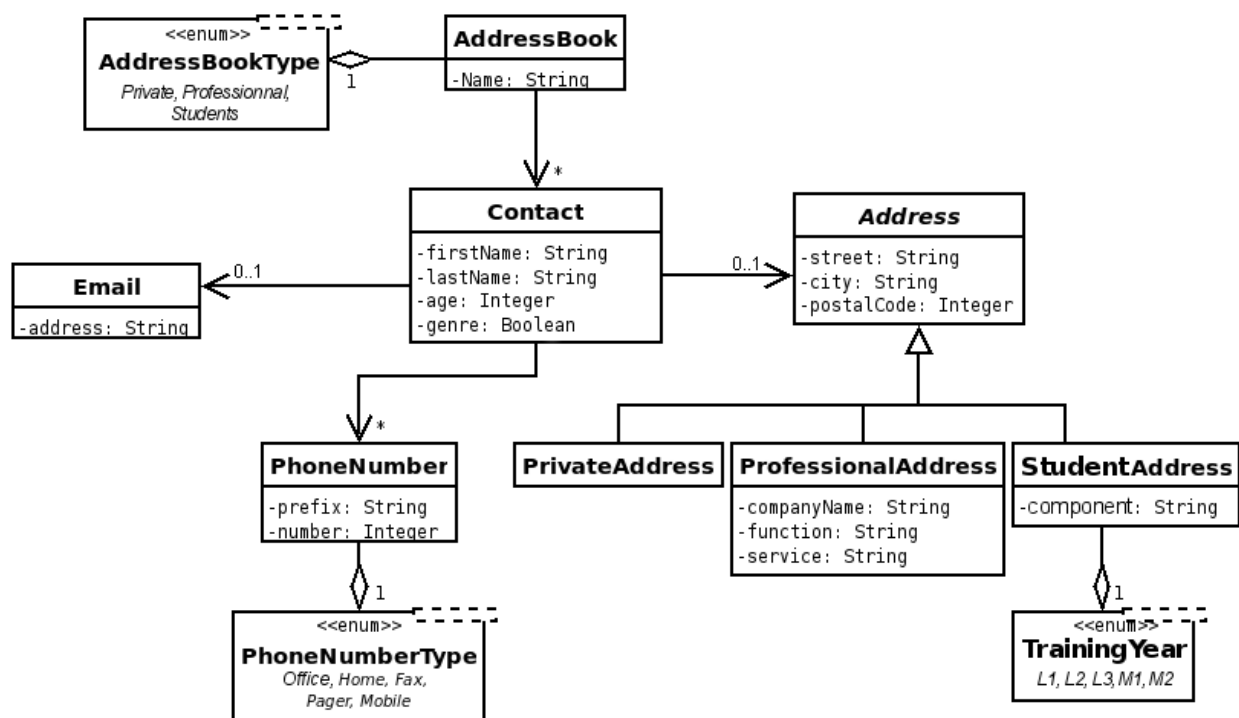


→ Pour chaque élément de votre interface utilisateur, essayer de penser à la réutilisabilité de vos composants en décomposant le plus possible votre prototypage.

#### 4. Conception de la maquette de saisie d'un contact

→ Créer un nouveau formulaire Swing de type **JPanel** pour le design de la saisie d'un contact. Nommer ce formulaire « **ContactPanel** ».

→ Concevoir la maquette de l'écran de saisie d'un contact (prototypage simple uniquement). L'interface doit être ergonomique et permettre à l'utilisateur de renseigner les informations propres à un contact, ainsi que de saisir son email, son adresse et plusieurs numéros de téléphone. Le diagramme de classe suivant vous permettra de distinguer les informations nécessaires pour le formulaire de création d'un contact:



#### 5. Construction d'une boîte de dialogue OK/Cancel réutilisable

→ Créer un formulaire Swing de type **JDialog** pour l'affichage d'une boîte de dialogue, avec les boutons « OK » et « Cancel ». Nommer ce formulaire « **OkCancelDialog** ».

Ce formulaire sera réutilisé pour toutes les boîtes de dialogues nécessitant une action de type « OK/Cancel ». Pour qu'un formulaire puisse être réutilisé, il faut respecter les règles suivantes :

- doit être une sous classe de « `java.awt.Container` »
- doit être un « bean » (avec un constructeur sans paramètre)
- doit être vide ou avoir un conteneur délégué vide

→ Rendre le formulaire « OkCancelDialog » réutilisable. Pour pouvoir changer le contenu de cette boîte de dialogue, vous devez définir un conteneur délégué. Ce dernier se déclare dans le bean info associé au formulaire:

- ajouter un bean info au formulaire : menu contextuel → « BeanInfo Editor ... »
- dans la classe du bean info, déclarer le nom de la méthode qui retourne le conteneur délégué vide dans la méthode « `getBdescriptor()` » : « `beanDescriptor.setValue("containerDelegate", "getjPanell");` »

## 6. Assemblage de la boîte de dialogue de saisie d'un contact

→ Créer un formulaire Swing de type **Bean** pour l'affichage d'une boîte de dialogue contenant le formulaire de saisie d'un contact. Nommer ce formulaire « ContactDialog ». Il doit étendre le formulaire « OkCancelDialog ».

→ Ajouter le composant de saisie d'un contact « ContactPanel » dans « ContactDialog » pour former une boîte de dialogue de saisie d'un contact.

**Note** : la prévisualisation du formulaire ainsi assemblé n'affiche pas la boîte de dialogue complète avec les boutons. Pour tester et évaluer le résultat final, il est nécessaire d'ajouter un « `main()` » dans « ContactDialog » et d'y coder la création et l'affichage de la boîte de dialogue.

## 7. Analyse du code généré

→ Vous pouvez analyser le code que l'outil a généré pour la création de l'interface utilisateur. Il est important de ne jamais modifier directement à la main cette partie du code car elle est maintenue de façon automatique par NetBeans, non pas dans le code Java, mais par l'intermédiaire d'un fichier XML « .form ».