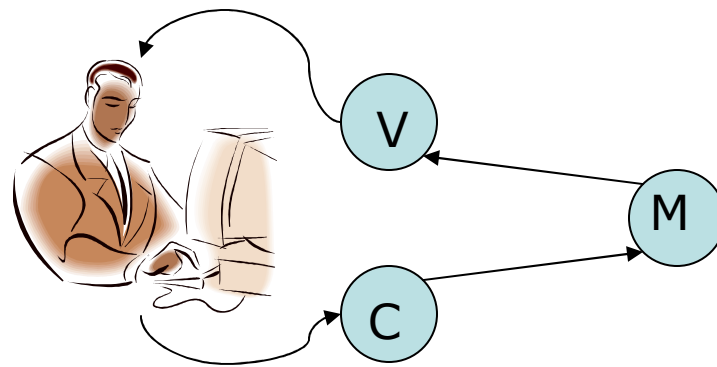


Organisation

- Introduction à Swing
- Composants et Conteneurs
- Gestion de la Mise en Page
- Gestion des Événements
- **Modèles**

Modèle-Vue-Contrôleur

- Architecture mise en oeuvre pour la conception d'IHM :
 - Modèle gère les données
 - Vue affiche les données du modèle pour l'utilisateur
 - Contrôleur gère les entrées utilisateur et communique avec le modèle



Modèles et Vues

- Plusieurs composants Swing utilisent des modèles.
- La vue est donc complètement séparée des données qui sont stockées dans le modèle.
- Ex. : une `JList` est une Vue-Contrôleur qui communique avec un `ListModel` correspondant.

Comment Créer un Modèle ?

Exemple des Listes

- Le constructeur de `JList` permet de passer en paramètre un tableau d'éléments, qui est alors utilisé pour créer un `ListModel` par défaut.
- Quand on utilise les méthodes de `JList`, on va opérer sur ce modèle.
- Implémentation de l'interface `ListModel` pour créer un modèle de données sur mesure.

Création de Modèles de JComboBox

- Manipulation de JComboBoxes comprenant des chaînes de caractère
 - Implémenter `ComboBoxModel<String>`
 - Hériter de `AbstractListModel<String>`

- `ModeleCBString`

`ModeleCBString ()`

`ModeleCBString (String[] tabStrings)`

`List<String> accesStrings ()`

`int accesIndice (Object element)`

`void ajoutString (String string)`

`void effaceString (String string)`

`void effaceTout ()`

`@Override :`

`String getSelectedItem ()`

`void setSelectedItem (Object o)`

`String getElementAt (int indice)`

`int getSize ()`

cf. méthodes

`fireIntervalAdded` et

`fireIntervalRemoved`