

WIESLAW ZIELONKA
ZIELONKA@IRIF.FR
WWW.IRIF.FR/~ZIELONKA

INTERFACES GRAPHIQUES

JAVA FX PROPRIÉTÉS

C'est quoi une propriété?

- contient une valeur de la variable d'instance et est observable
- les listeners enregistrés sur une propriété sont notifiés quand la valeur de la propriété change
- utilisation de binding : la valeur de la propriété est calculée à partir de sources et change automatiquement quand les valeurs fournies par les sources changent

JAVA FX PROPERTIES

- read-write propriétés
- read-only propriétés
- propriétés immuables
- java fx propriété est un "wrapper" autour de java bean.

la convention sur les noms les méthodes d'accès aux propriétés

Pour chaque propriété de type read-write trois méthodes.

Par exemple la classe Rectangle possède la propriété

`DoubleProperty width;`

et elle fournit trois méthodes:

`DoubleProperty getWidthProperty()`

`double getWidth()`

`void setWidth(double value)`

La même règle de nommage s'applique à toutes les propriétés :

`get....Property()` `set...()` `get...()`

pour obtenir la propriétés, changer la valeur, obtenir la valeur.

Propriétés

- Les classes abstraites:

IntegerProperty	FloatProperty	DoubleProperty
BooleanProperty	LongProperty	StringProperty
ObjectProperty<T>	ListProperty<E>	SetProperty<E>
MapProperty<K, V>	ReadOnlyXXX	

- Wrapper classes

SimpleIntegerProperty	SimpleFloatProperty
SimpleBooleanProperty	SimpleLongProperty
SimpleObjectProperty<T>	SimpleListProperty<E>
SimpleMapProperty<K, V>	SimpleDoubleProperty
SimpleSetProperty<E>	

JavaFX listeners

C'est quoi un Listener?

- Listener observe les changements de la valeur d'une propriété
- il est possible d'enregistrer autant de Listeners qu'on a besoin
- possibilité de supprimer les Listeners qui étaient enregistrés auparavant

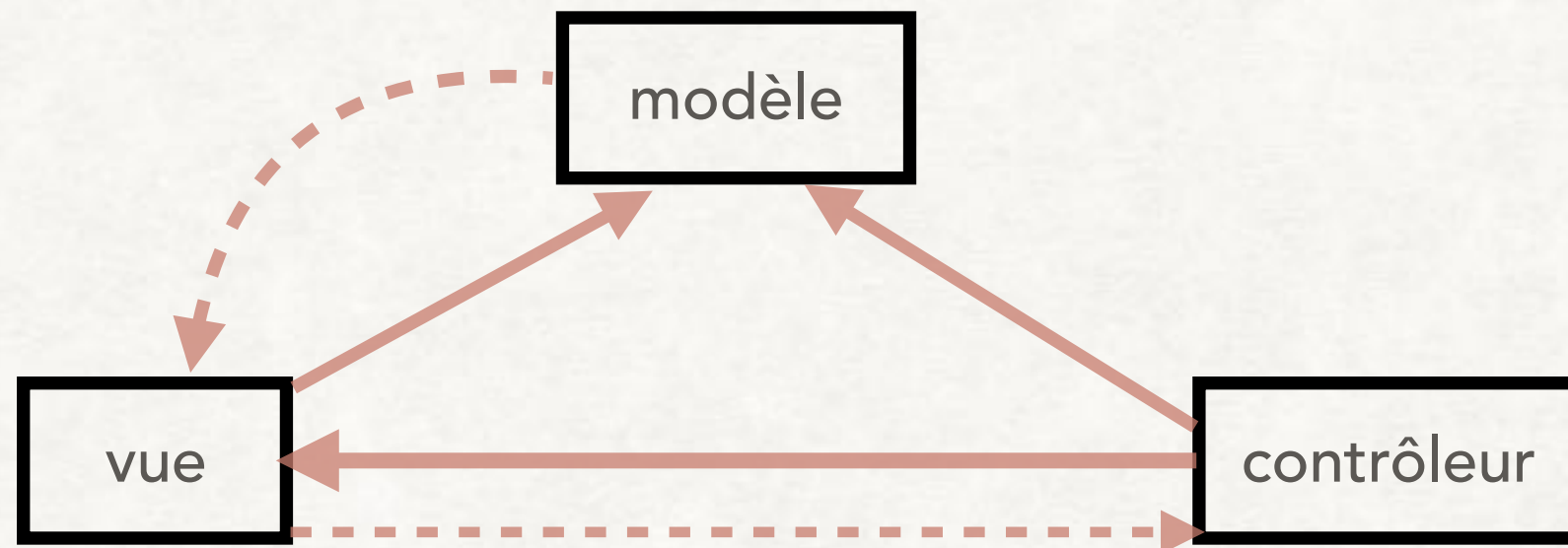
`InvalidationListener` est appelé quand les données ne sont plus valables.

`ChangeListener` est appelé quand les données ne sont plus correctes.

Comment enregistrer un Listener

```
ChangeListener<Boolean> chListener = new  
ChangeListener<Boolean>() {  
  
    @Override  
    public void changed(  
        ObservableValue<? extends Boolean> ov,  
        Boolean old_val, Boolean new_val) {  
        // code de ChangeListener  
    }  
  
};  
  
checkBox.selectedProperty()  
    .addListener(chListener);
```


Modèle - vue - contrôleur



- Modèle — les données de l'application et le traitement de données
- vue — l'interface graphique lui-même
- contrôleur — est responsable de la synchronisation de modèle et de vue