

# Partie I : Programmation graphique en Swing

Programmation objet 2 (2012-2013)

Chapitre 7 Beans

- Composants logiciels
- Bean
- JavaBeans
- Composants graphiques de Swing

# Composants logiciels

- Composants logiciels aux différents niveaux d'abstraction :
  - Une suite de code source
  - Fonction
  - Classe
  - Bean

# Bean

- « bean » = haricot en français
- En informatique,
  - Bean = Object + Pattern
  - Un bean est un « objet » qui encapsule plusieurs objets de manière à répondre à un besoin métier

# Bean

- Beans possèdent
  - des propriétés
  - des évènements reconnus
  - des méthodes de traitement des évènements
- Les composants graphiques de Swing sont des beans
  - JavaBeans

# Convention JavaBeans

- Un composant JavaBean est une simple classe Java qui respecte certaines conventions sur le nommage, la construction et le comportement des méthodes.
- Les conventions à respecter sont les suivantes :
  - La classe doit être "Serializable" pour pouvoir sauvegarder et restaurer l'état d'instances de cette classe ;
  - La classe doit posséder un constructeur sans paramètre (constructeur par défaut);
  - Les attributs privés de la classe (variables d'instances) doivent être accessibles publiquement via des méthodes accesseurs construit avec *get* ou *set* suivi du nom de l'attribut avec la première lettre transformée en majuscule. Le couple d'accesseurs est appelé Propriété.
  - La classe ne doit pas être déclarée « final ».

# Convention JavaBeans

Classe :

implements Serializable

***thread-safe***

Les propriétés simples (property) sont

private ou protected <Propertytype> <PropertyName>;

accesseurs possibles : R, W ou RW

public <Propertytype> get<PropertyName>()

public void set<PropertyName>(<Propertytype> valeur)

public boolean is<PropertyName>()

Les propriétés indicées (indexed property) = les tableaux

public <Propertytype>[ ] get<PropertyName>()

public void set<PropertyName>(<Propertytype>[ ] valeur)

public <Propertytype> get<PropertyName>(int indice)

public void set<PropertyName>(int indice, <Propertytype> valeur)

Les événements des beans :

xxxEvent extends java.util.EventObject interface XxxListener public void addXxxListener  
(XxxListener) public void removeXxxListener(XxxListener)

# Exemple

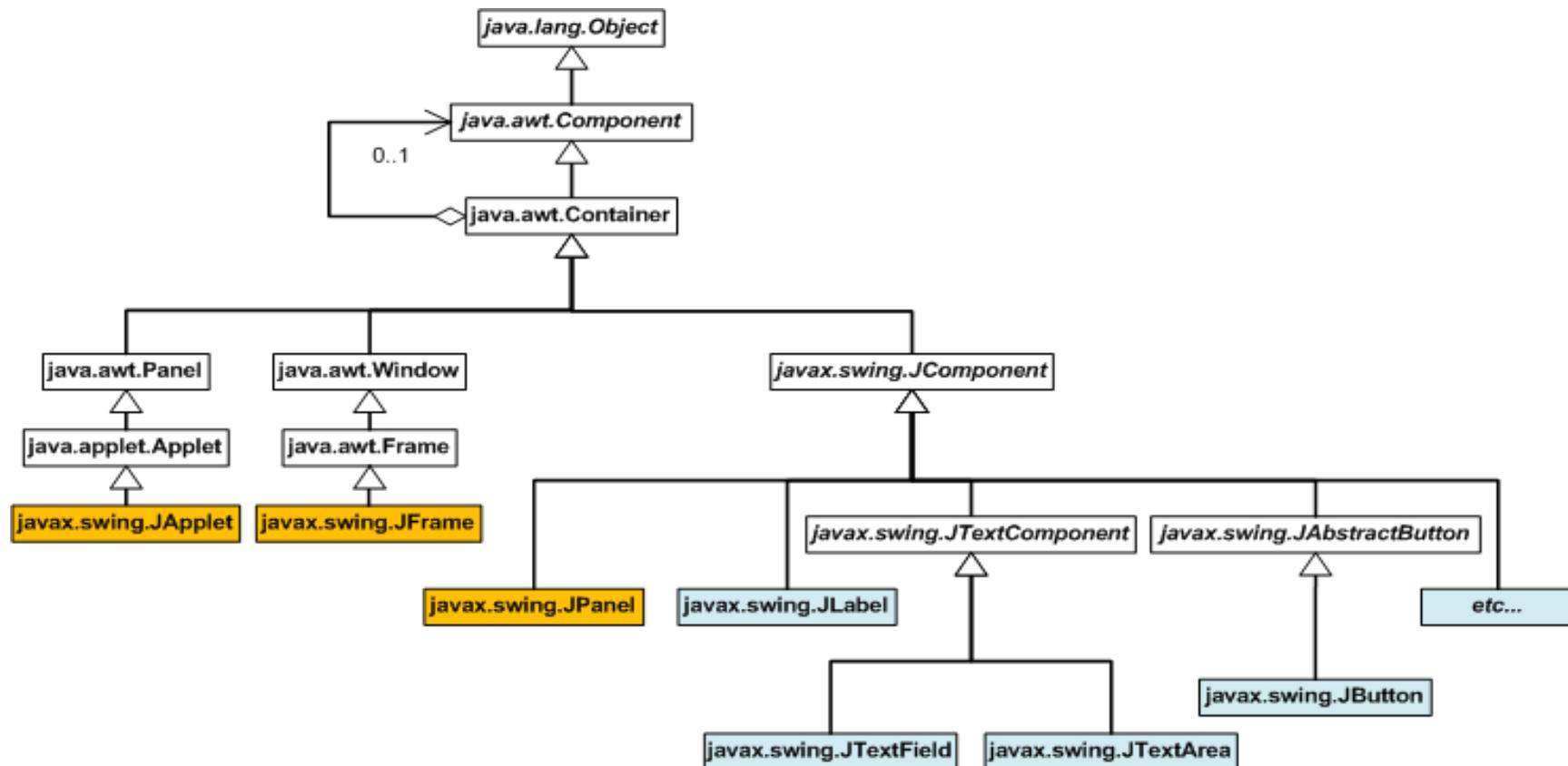
```
public class PersonBean implements Serializable {
    private String name;
    private boolean deceased;
    public PersonBean() {
    }
    public String getName() {
        return (this.name);
    }
    public void setName(String name) {
        this.name = name;
    }
    public boolean isDeceased() {
        return (this.deceased);
    }
    public void setDeceased(boolean deceased) {
        this.deceased = deceased;
    }
}
```

```
public static void main(String[] args) {
    PersonBean person = new PersonBean
    ();
    person.setName("Bob");
    person.setDeceased(true);

    // Affichage : "Bob [décédé]"
    System.out.print(person.getName());
    System.out.println(person.isDeceased
    () ? " [décédé]" : "[vivant]");
    }
}
```

# Composants graphiques de Swing

- Swing utilise le pattern Composite pour définir les composants graphiques



# Composite

- Problème
  - On souhaite disposer de la même interface pour manipuler un objet seul ou un ensemble d'objets
- Solution
  - Component
    - L'abstraction de tous les composants, y compris ceux qui sont composés
  - Composite
    - Des composants composés
  - Feuille
    - Des composants non-composés

# Composite

