

## Autre Solution : `AbstractAction`

- Classe `Action` : représente une action de Swing
- La classe `AbstractAction` est une implémentation abstraite qui nous évite de devoir implémenter toutes les méthodes d'`Action`
- `AbstractAction` est un `ActionListener` permettant de gérer, en plus de l'action générée par le composant, le texte, l'icône et l'état du composant (activé/désactivé)
- Implémentation : pour ajouter une action à un programme, création d'une classe héritant de `AbstractAction`
  - dans un nouveau fichier
  - en tant que classe interne
  - en tant que classe anonyme
- Exemple de création d'un bouton
  - `JButton bouton = new JButton(new IciAction("Cliquez ici !"));`

# Exemples de Classes Héritant de AbstractAction

```
public class IciAction extends AbstractAction {  
    public IciAction(String texte){  
        super(texte);  
    }  
  
    public void actionPerformed(ActionEvent e) {  
        System.out.println("Vous avez cliqué ici");  
    }  
}
```

```
public class LaAction extends AbstractAction {  
    public LaAction(String texte){  
        super(texte);  
    }  
  
    public void actionPerformed(ActionEvent e) {  
        System.out.println("Vous avez cliqué là");  
    }  
}
```

# Composants Texte

- Composants de saisie : permettent à l'utilisateur de saisir du texte
  - Le **JTextField** : le plus simple. Il permet d'entrer un texte sur une seule ligne.
  - Le JTextArea : Il permet d'entrer un texte complet sur plusieurs lignes.
  - Le JEditorPane : Très complet, possibilité de modifier police, taille et couleur du texte. Il permet même d'afficher des pages html et des images.
  - Le JTextPane : C'est le composant texte le plus évolué de Swing. Il permet la même chose que le JEditorPane et également d'insérer des composants personnalisés (e.g. composants Swing).
- Composants d'affichage : permettent d'afficher du texte
  - JLabel

# Listes de Choix

- Deux sortes de listes déroulantes
  - JList, liste déroulante qui permet d'afficher plusieurs éléments à la fois
  - JComboBox
    - addItem(Object item), ajoute un nouvel objet à la liste
    - removeItem(Object item), supprime l'objet de la liste
    - removeAllItems(), vide la liste déroulante
    - getSelectedItem(), retourne l'objet sélectionné

```
Object[] elements = new Object[]{"Element 1", "Element 2", "Element 3", "Element 4", "Element 5"};  
  
JComboBox listel = new JComboBox(elements);
```

# Exercices

- Développer 1 interface avec un `JTextField`, un label et un bouton de manière à ce que le texte écrit dans le `JTextField` soit copié dans le label lors du clic sur le bouton.
- Développer 1 interface avec 2 `JComboBoxes` et trois boutons permettant de : i) copier un élément du premier `JComboBox` dans le second, ii) supprimer un élément du second `JComboBox`, iii) vider tout le second `JComboBox`.