

# Interface graphique Swing

Package javax.swing.\*

**Projects**

- AppDebug
- AppJeu
  - Source Packages
    - <default package>
    - appljeu
      - fenetre
        - AjoutHeros.java
        - JAjoutJoueur.java
        - JMenu.java
      - lesException
      - option
    - Test Packages
    - Libraries
    - Test Libraries
    - Bd
    - BD-AppelJava
    - CVDATPTEST
      - Source Packages
        - Exception
        - cvdatptest
          - Bibliotheque.java
          - CVDATPTEST.java
          - ListePersonne.java
          - Livre.java
          - Personne.java
      - Test Packages
        - <default package>
          - BibliothequeTest.java
          - ListePersonneTest.java
      - Libraries
      - Test Libraries
    - Devoir
    - Geometrie
    - GestionException

Source Design History

To change layout manager of a container use Set Layout submenu from its context menu.

Création d'un héros

Nom

Points de vie

Défense

Poids

Agilité

Ajouter

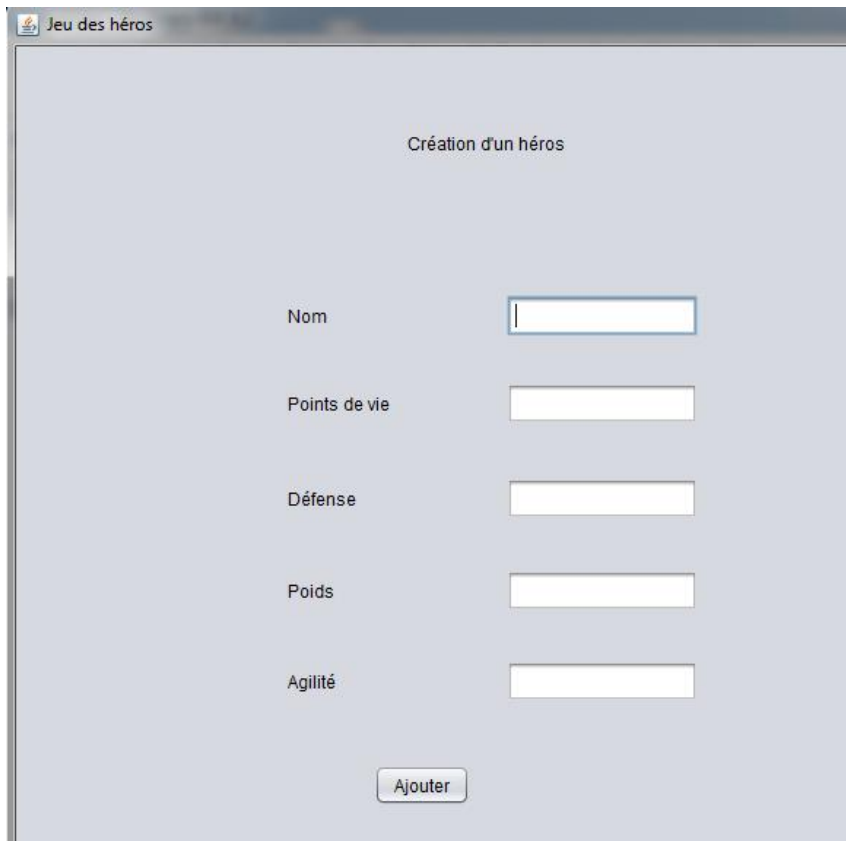
**Palette**

- Swing Containers
  - Panel
  - Tabbed Pane
  - Split Pane
  - Scroll Pane
  - Tool Bar
  - Desktop Pane
  - Internal Frame
  - Layered Pane
- Swing Controls
  - Label
  - Button
  - Toggle Button
  - Check Box
  - Radio Button
  - Button Group
  - Combo Box
  - List
  - Text Field
  - Text Area
  - Scroll Bar
  - Slider
  - Progress Bar
  - Formatted Field
  - Password Field
  - Spinner
  - Separator
  - Text Pane
  - Editor Pane
  - Tree
  - Table
- Swing Menus
- Swing Windows
- Swing Fillers
- AWT
- Beans
- Java Persistence

**AjoutHeros.java - Properties**

Properties	
Name	AjoutHeros
Extension	java
File Size	9388
Modification Time	2 mai 2018 17:39:29
All Files	D:\Dropbox\NetBeansProject\AppliJ...
Classpaths	
Compile Classpath	
Runtime Classpath	D:\Dropbox\NetBeansProject\AppliJ...
Boot Classpath	C:\Program Files\Java\jdk1.8.0_131\...

# Création de JFrame



The screenshot shows a Java Swing window titled "Jeu des héros". Inside the window, there is a form titled "Création d'un héros". The form contains five text input fields, each with a label to its left: "Nom", "Points de vie", "Défense", "Poids", and "Agilité". Below these fields is a button labeled "Ajouter".

- Chaque JFrame possède une méthode main (Run ou classe de démarrage)
- Propriétés



Properties	
defaultCloseOperation	DISPOSE
title	Jeu des héros

# Classes - interface graphique

<b>nomClasse</b>	<b>Permet de gérer</b>
JFrame	Une fenêtre
JButton	Un bouton
JTextField	Un champ de saisie
JRadioButton	Un bouton de sélection
JComboBox	Une liste déroulante
JCheckBox	Une case à cocher
JMenu	Un menu affiché dans la fenêtre
JDialog	Une boîte de dialogue

# Comportement des objets

## Propriétés

Properties	Binding	Events	Code
[-] Properties			
editable		<input checked="" type="checkbox"/>	...
background		<input type="checkbox"/> [255,255,255]	...
columns		0	...
document		<default>	...
font		Tahoma 11 Plain	...
foreground		■ [0,0,0]	...
horizontalAlignment		LEADING	...
text			...
toolTipText			...
[-] Other Properties			

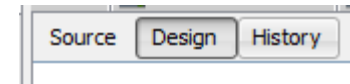
## Evènements

Properties	Binding	Events	Code
[-] Events			
actionPerformed		<none>	...
ancestorAdded		<none>	...
ancestorMoved		<none>	...
ancestorMoved		<none>	...
ancestorRemoved		<none>	...
ancestorResized		<none>	...
caretPositionChanged		<none>	...
caretUpdate		<none>	...
componentAdded		<none>	...
componentHidden		<none>	...

# Gestion des propriétés

- Gestion par l'interface graphique de l'environnement de développement
- Gestion dans le code avec les accesseurs sur les propriétés

# Génération de code



```
public class AjoutHeros extends javax.swing.JFrame {

    /**
     * Creates new form AjoutHeros
     */
    public AjoutHeros() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jTextNom = new javax.swing.JTextField();
        jTextPoints = new javax.swing.JTextField();
        jTextDefense = new javax.swing.JTextField();
        jTextPoids = new javax.swing.JTextField();
        jTextAgilite = new javax.swing.JTextField();
        jButtonAjout = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
    }
}
```

← Non modifiable

# Gestion des objets

## Classe Infos

- Créer la classe Infos
- Créer 2 listes public static à initialiser :
  - lesHeros
  - lesJoueurs

## Utilisation

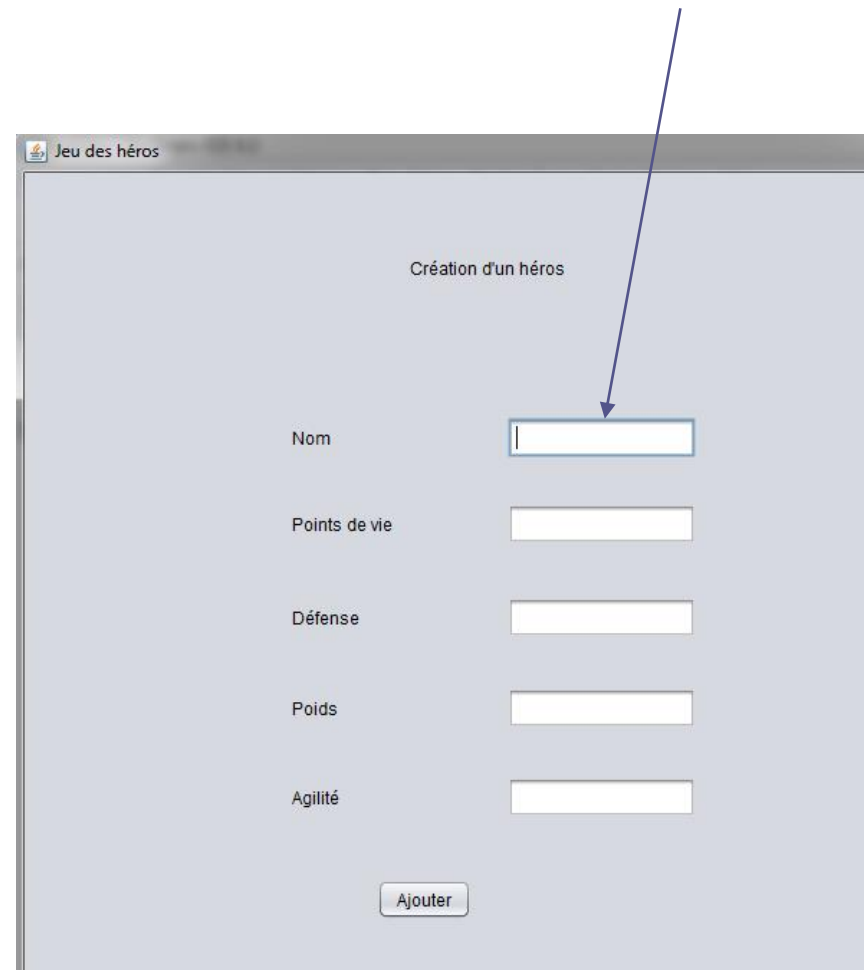
- Dans les fenêtres
- Importer la classe
- Utilisation des listes
  - `Infos.lesHeros.methode`
  - `Infos.lesJoueurs.methode`



# Exemple

Pour accéder à la valeur d'un champ JTextField :  
`nomChamp.getText();`

- Créer un package fenetres
- Créer la fenêtre JAjoutHeros qui permet d'ajouter un héros de terre au jeu
- Un clic sur le bouton Ajouter doit créer un objet de la classe HerosTerre et l'ajouter à la liste lesHeros

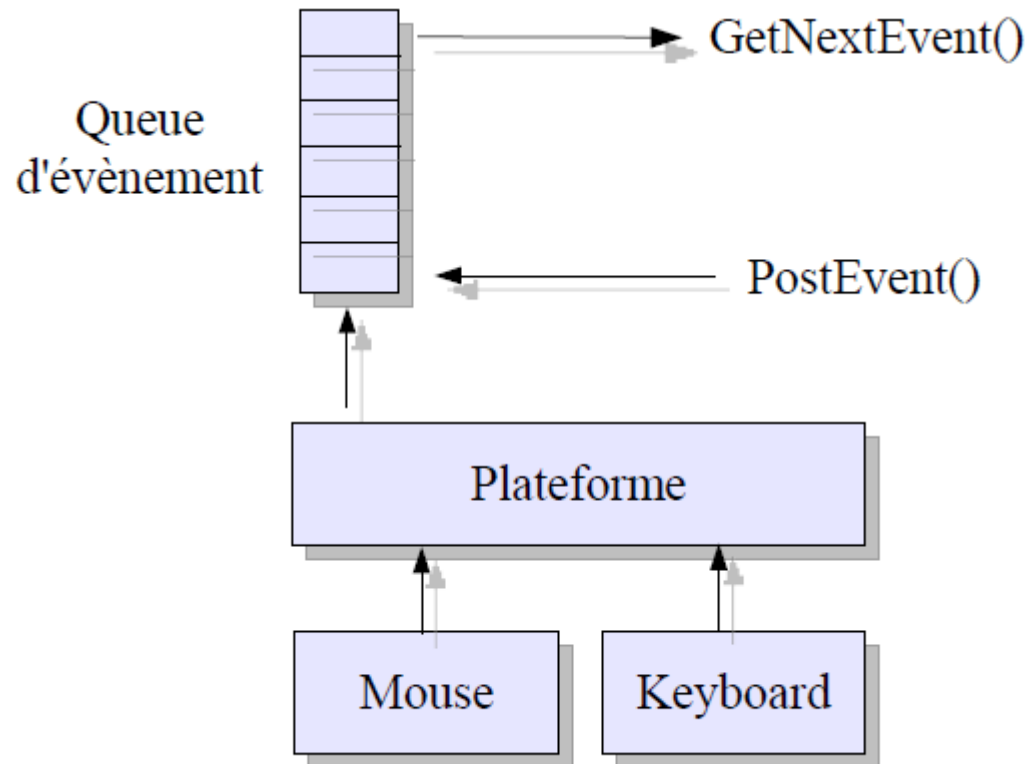


The screenshot shows a Java Swing window titled "Jeu des héros". Inside the window, there is a form titled "Création d'un héros". The form contains five text input fields, each with a label to its left: "Nom", "Points de vie", "Défense", "Poids", and "Agilité". At the bottom of the form, there is a button labeled "Ajouter". A blue arrow points from the top right of the slide to the "Nom" input field.

# Exemple d'évènements

Evènement	Action associée
actionPerformed	Clic sur l'objet
focusGained/focusLost	Focus sur l'objet ou focus perdu par l'objet
keyPressed/keyReleased	Touche appuyée ou relachée
mouseClicked/mouseEntered/mouseExited...	Actions de la souris sur l'objet

# Traitement des évènements



# Principe

- Création d'un écouteur (listener) en attente d'un évènement sur une ressource
- Une action de l'utilisateur est captée par le listener correspondant qui déclenche le programme associé

# Génération automatique avec Swing

```
jButtonAjout.setText("Ajouter");  
jButtonAjout.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButtonAjoutActionPerformed(evt);  
    }  
});
```

Ajout d'un listener

Définition de l'action  
écoutée

Méthode exécutée lorsque  
le listener détecte l'action

# Capter des évènements

- Plusieurs évènements peuvent être captés
- Action réalisée : clic
- Focus sur ou hors de l'objet
- Actions de la souris :
  - Bouton pressé
  - Déplacement
  - Pointeur entré dans une zone
  - Pointeur sortir d'une zone

# Exemple

- Faire afficher l'aide lors d'un survol des boutons (évènement FocusGained)
- Propriété ToolTipText à modifier

# Exercice

- Ajouter une aide dans la fenêtre JAjoutHeros pour indiquer à l'utilisateur les informations attendues dans chacun des champs
- Au choix
  - Utilisation des propriétés ToolTipText des champs
  - Modification d'un label d'aide intégré dans la fenêtre

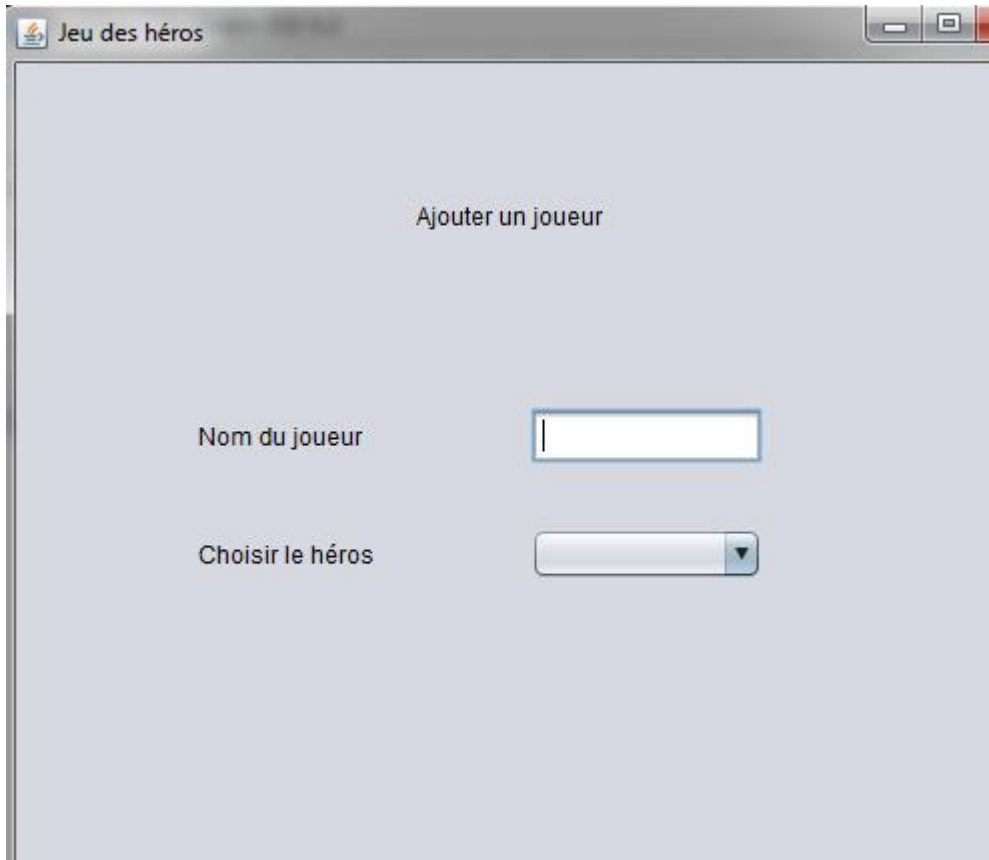


# Ouverture de fenêtres

- Créer la fenêtre JMenu suivante
- Créer les fenêtres JAjoutJoueur et JDemarrer vides
- Configurer la méthode actionPerformed afin d'afficher les fenêtres correspondant à chaque bouton lors d'un clic



# Fenêtre JAjoutJoueur



The image shows a Java Swing window titled "Jeu des héros". The window contains a form with the following elements:

- A title "Ajouter un joueur" centered at the top of the form area.
- A label "Nom du joueur" followed by a text input field.
- A label "Choisir le héros" followed by a dropdown menu.