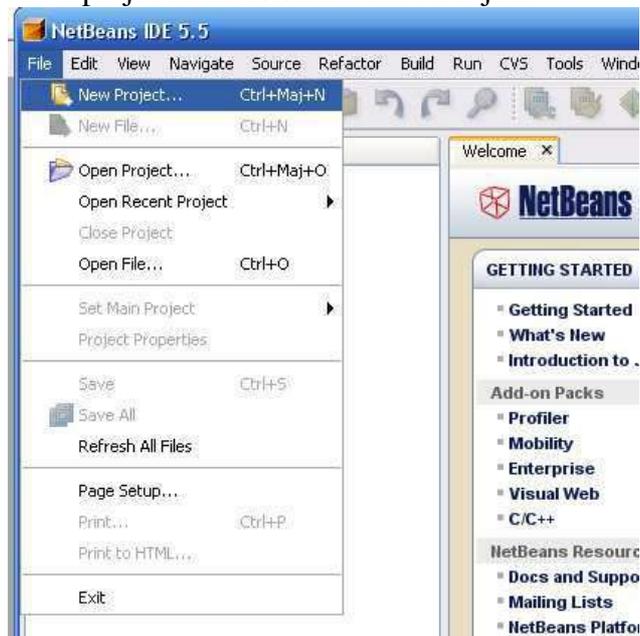


# **Réalisation d'interfaces graphiques Avec NetBeans**

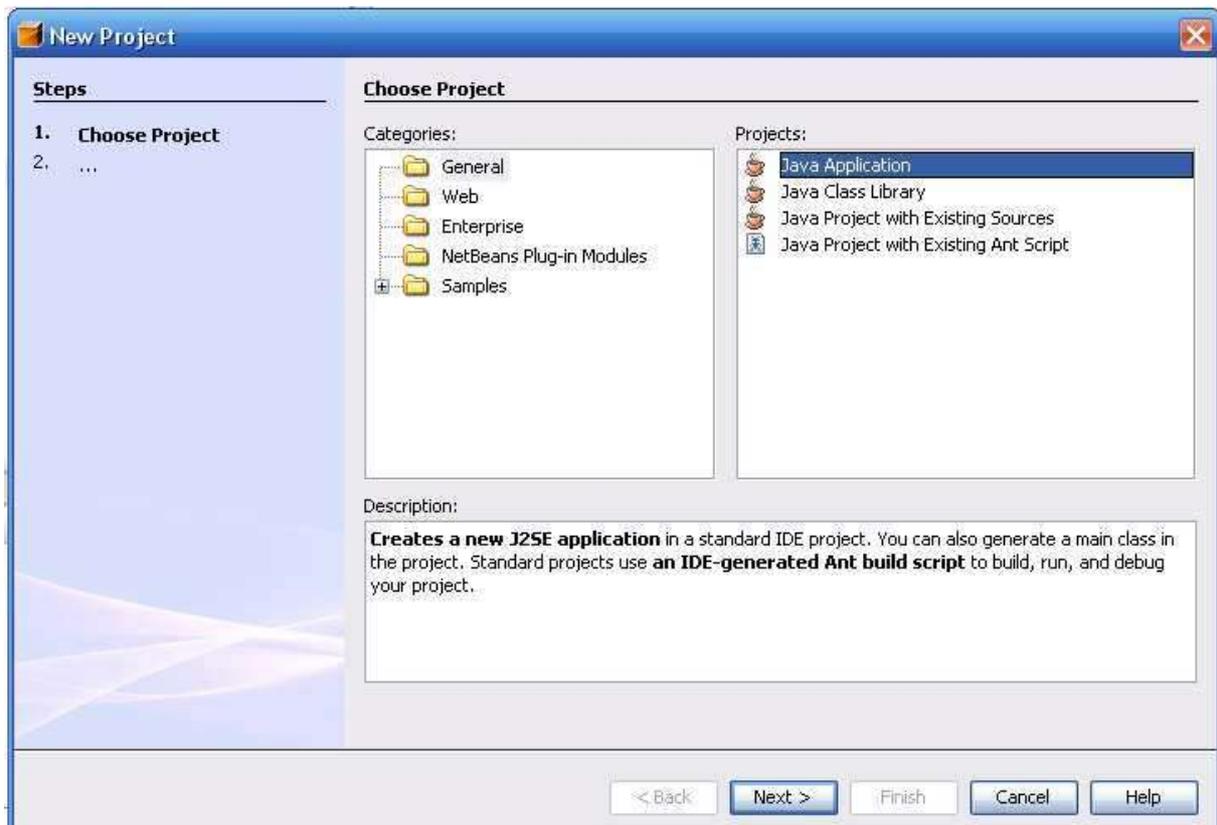
**Version 1.0, du mardi 30 octobre 2007**

## 1. Création d'un projet

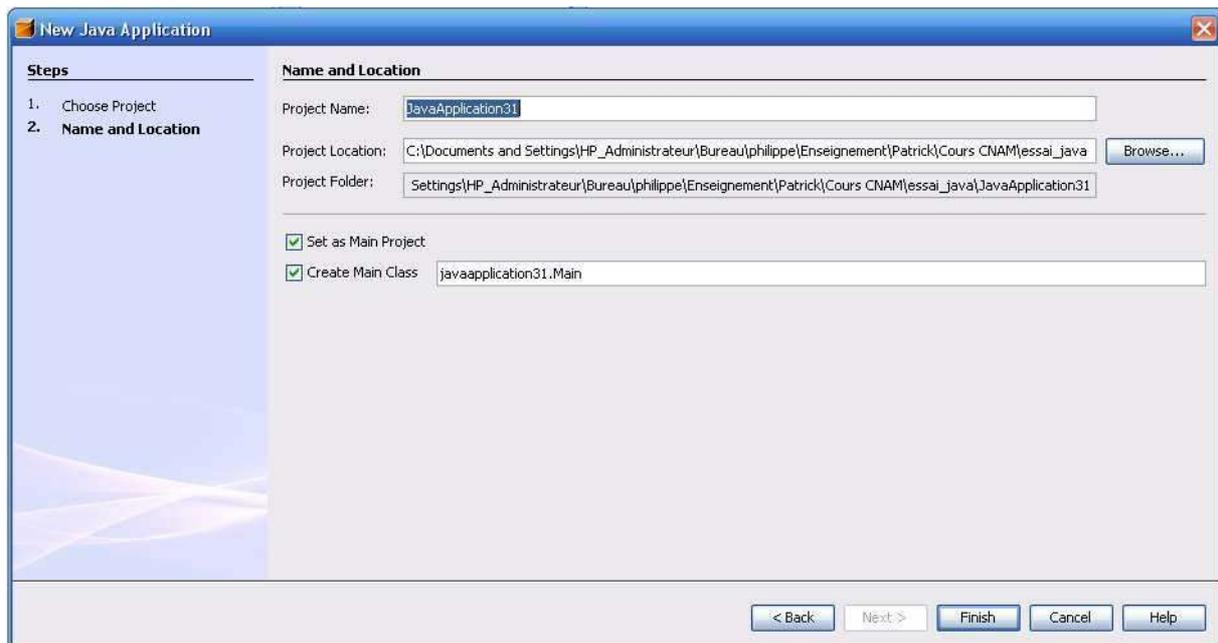
Dans le menu File, créez un projet en choisissant : New Project.



Dans le menu New Project, choisissez Java Application.

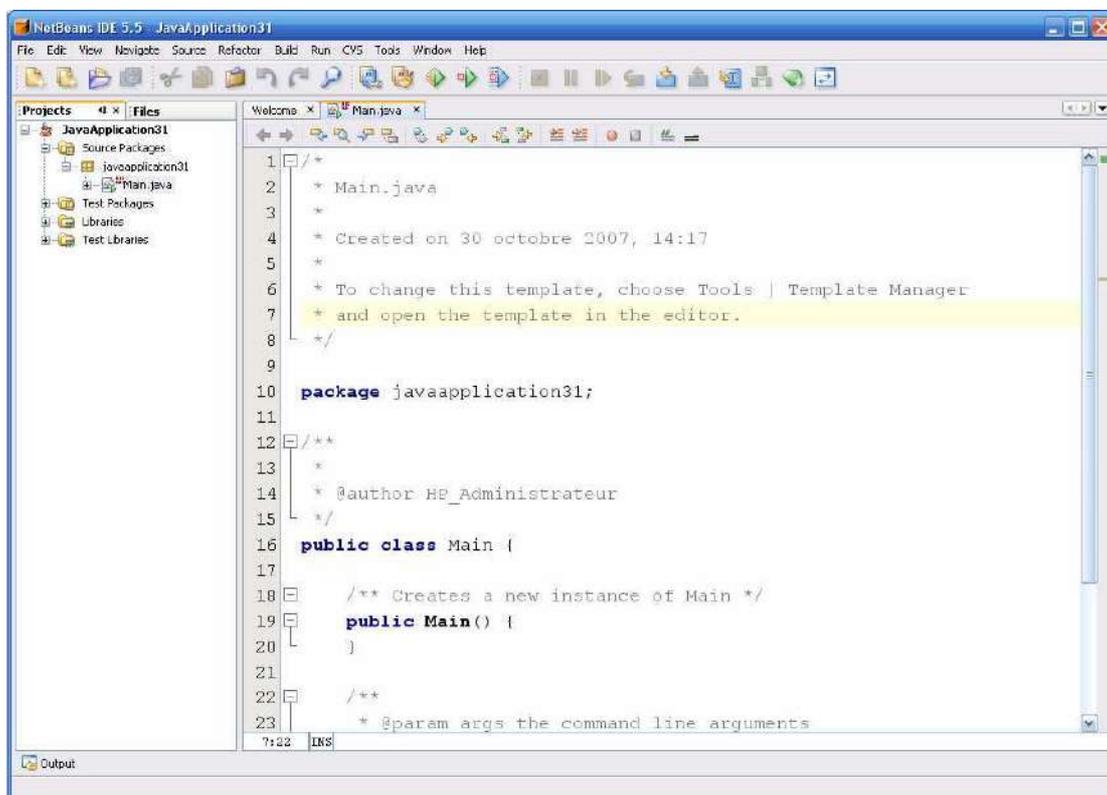


Vérifiez ensuite le chemin de sauvegarde du projet (Champs Project Location).



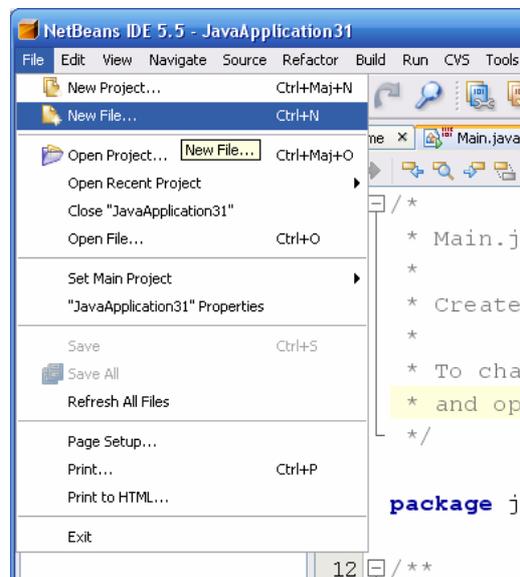
## 2. Ajout d'une interface dans le projet

Le projet initial se compose uniquement d'un fichier Main.java. Il s'agit par défaut d'une application console. Nous allons rapidement faire mieux !

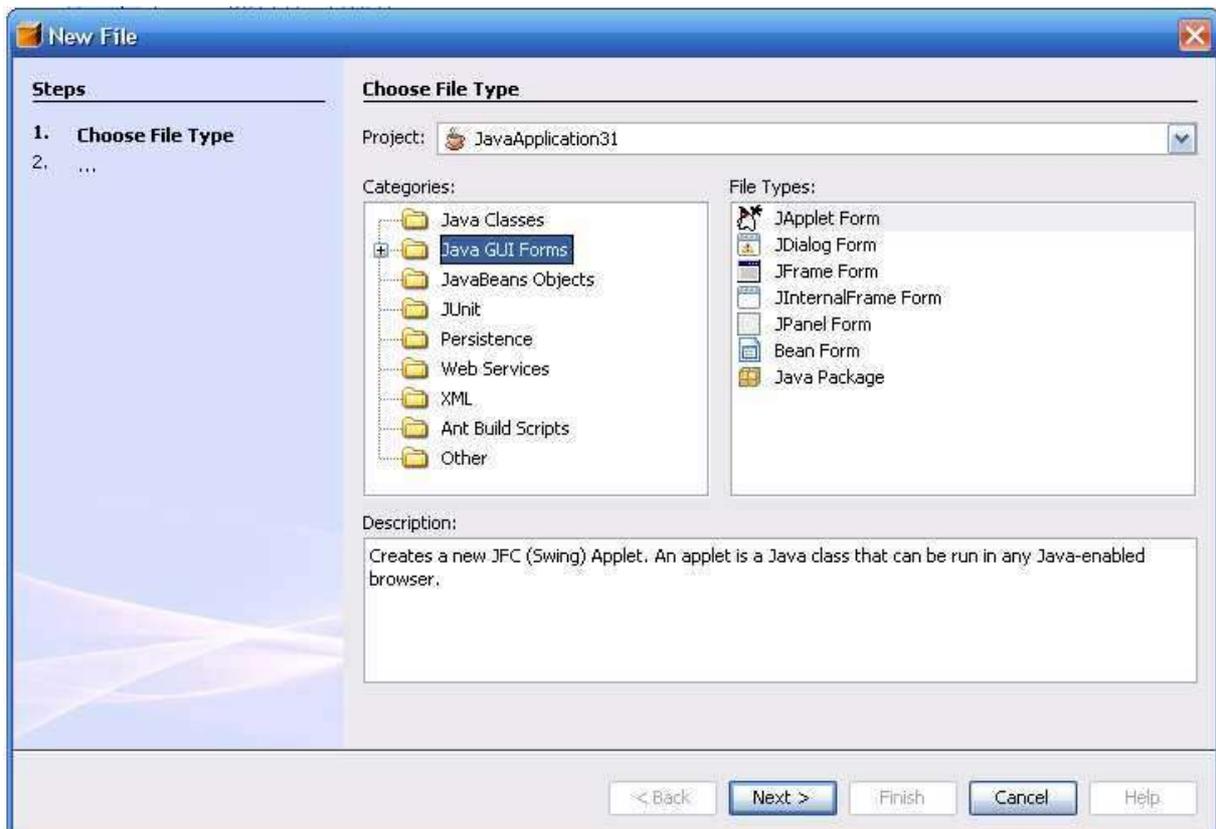


## 2.1. Ajouter un composant visuel à votre projet.

Pour cela utilisez le sous menu New File dans le menu File.

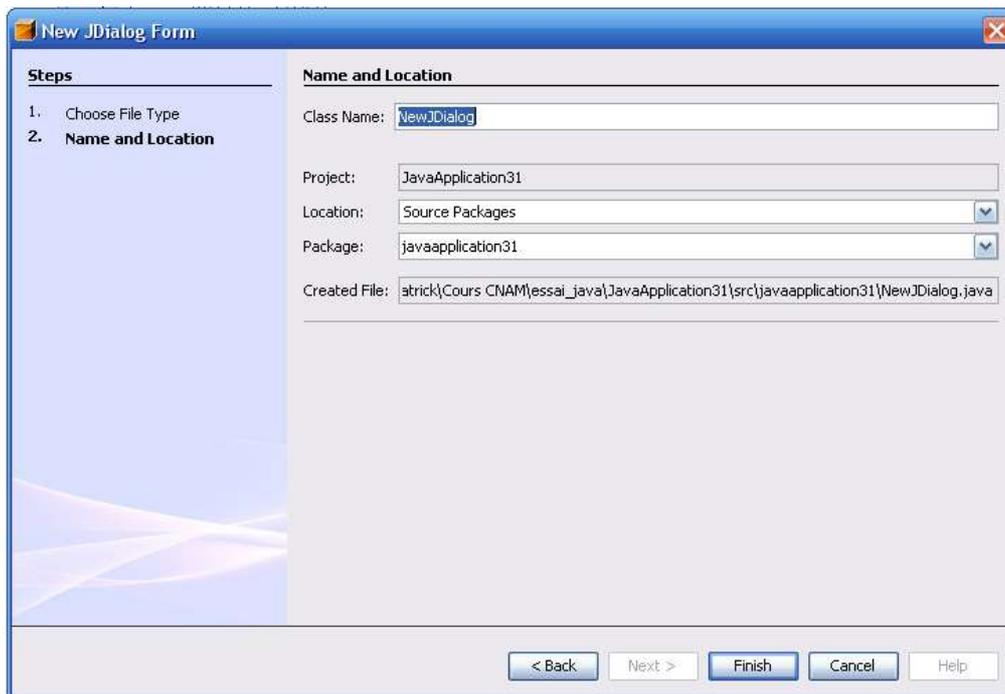


Les fichiers permettant d'obtenir des éléments graphiques sont dans Java Gui Forms.

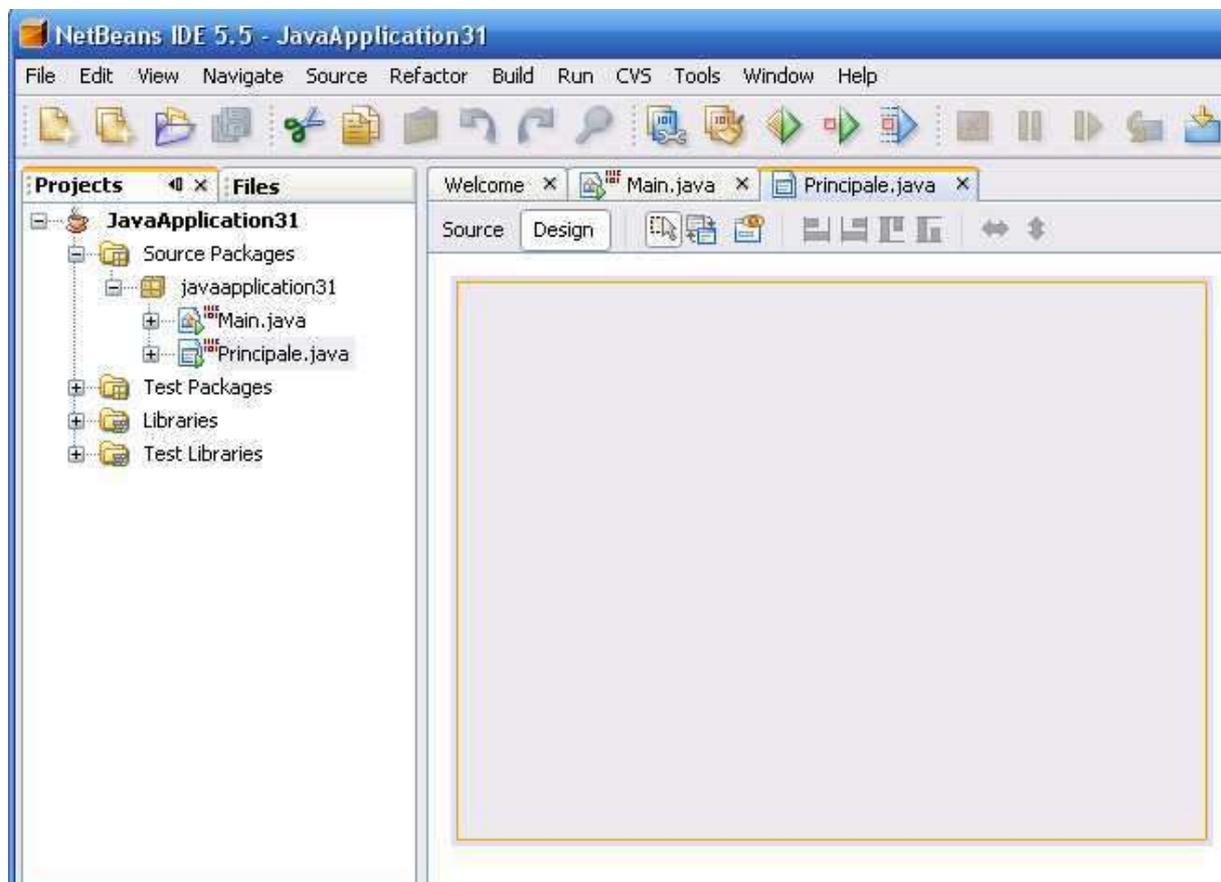


Il y a de nombreux types différents. Nous utiliserons un composant simple : **JDialog Form**.

Choisissez ensuite un nom adapté par exemple **Principale**.

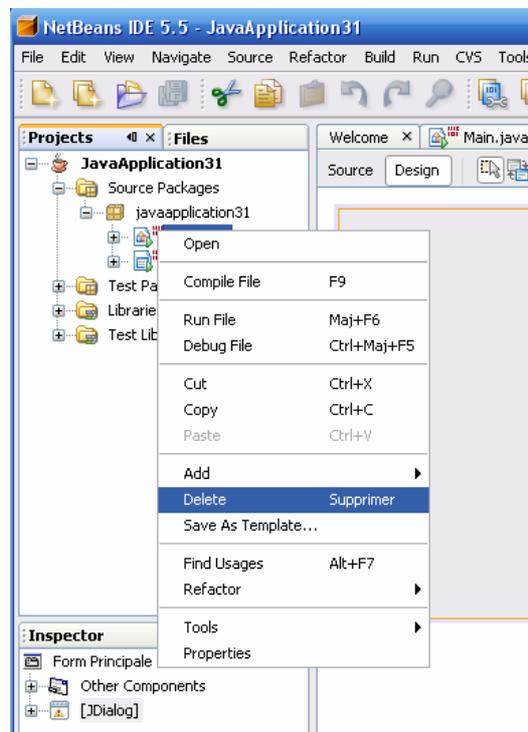


Le projet en cours est mis à jour et possède maintenant un fichier Main.java et un fichier Principale.java.

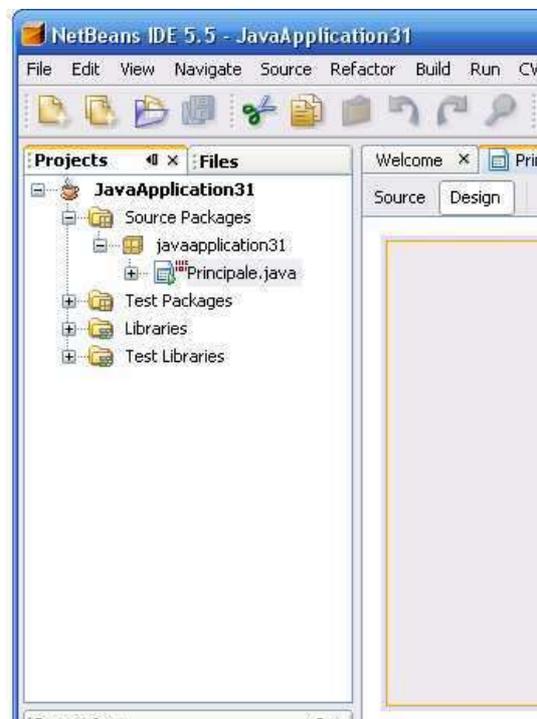


## 2.2. Suppression du fichier Main.java

Faire un click droit sur le fichier Main.java et choisir Delete dans le menu contextuel.

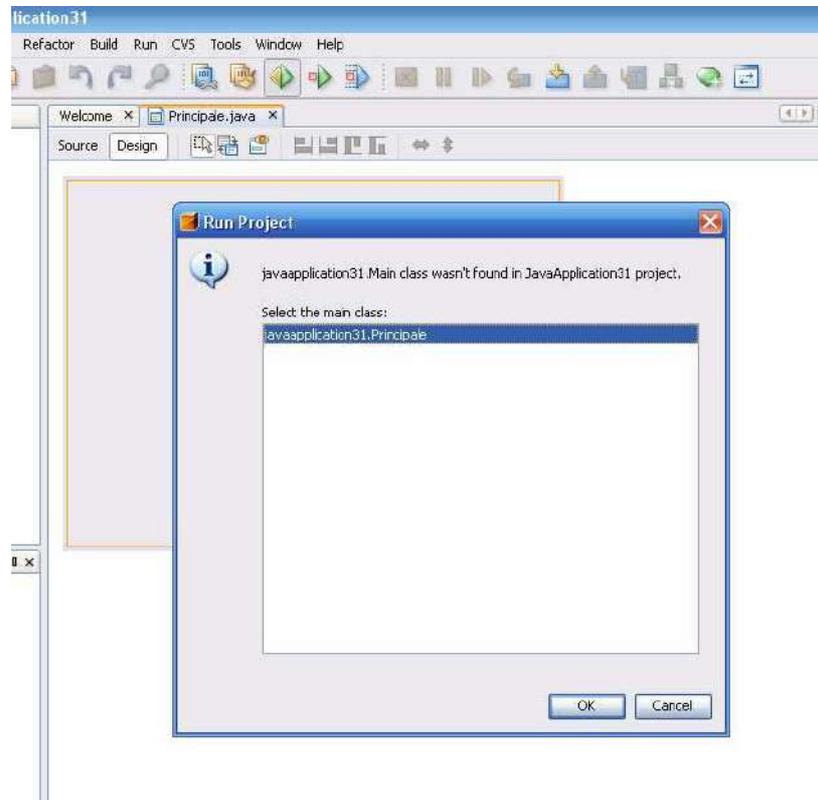


Le projet se présente alors comme suit :

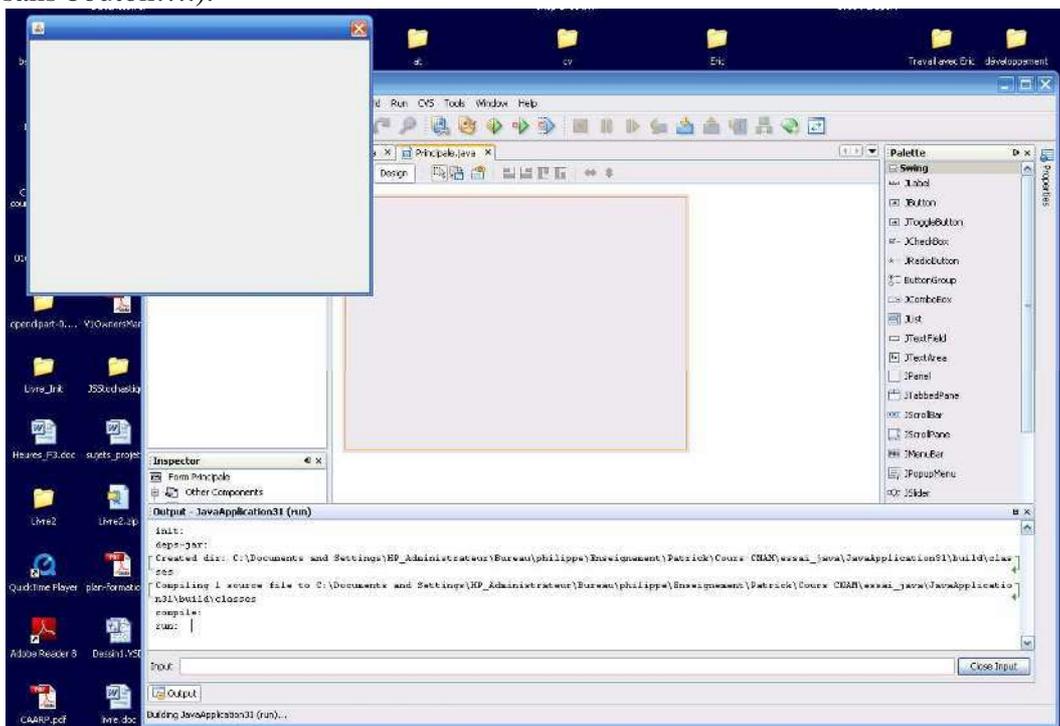


## 2.3. Compiler le projet

En cliquant sur , NetBeans vous demandera de sélectionner la classe principale.



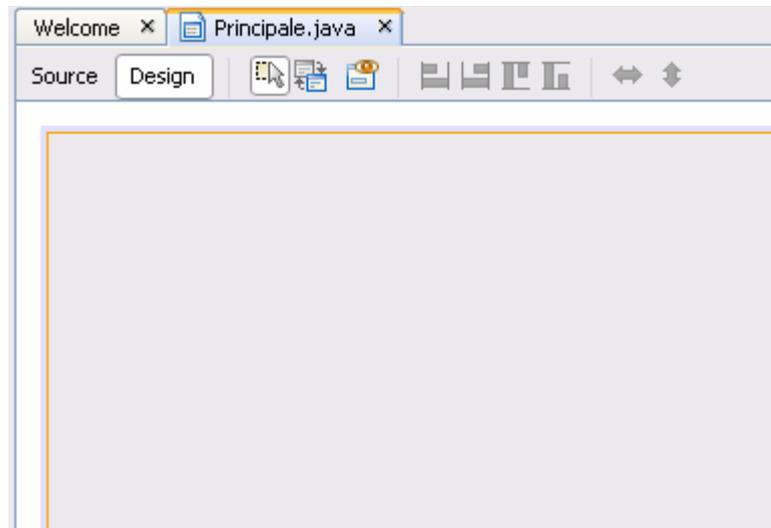
En validant Principale comme étant la classe on obtient comme résultat une fenêtre vide (sans nom, sans bouton...).



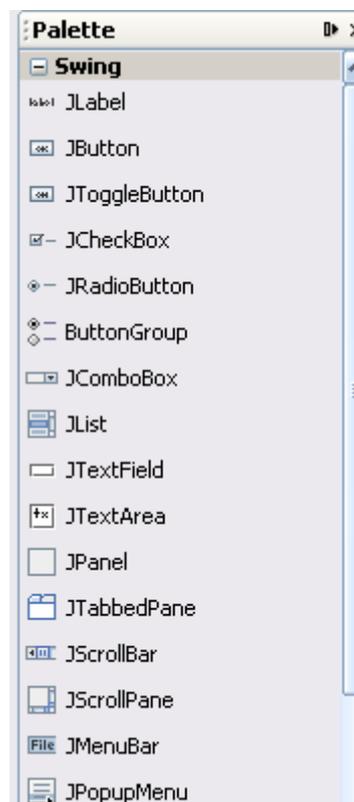
### 3. Les éléments de base d'une fenêtre

#### 3.1. Utilisation des objets de l'interface

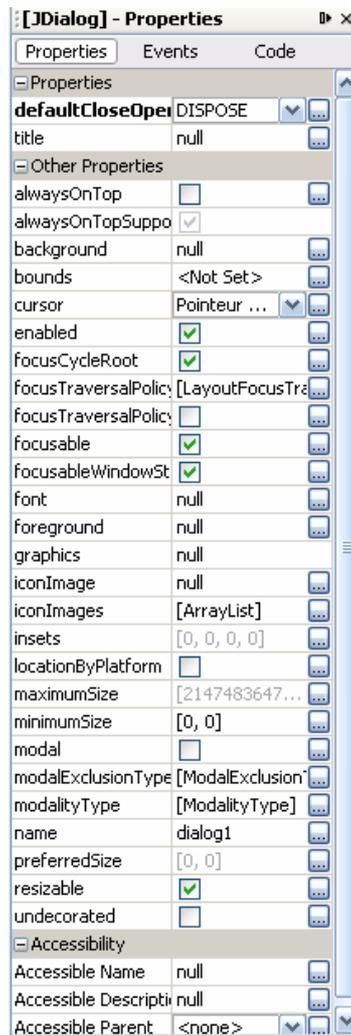
Une fenêtre graphique peut se manipuler sous forme graphique (utilisez l'onglet Design) ou sous forme textuelle (onglet Source). La construction d'une interface se fait en mode Design.



Tous les objets graphiques nécessaire à une interface sont regroupés dans l'onglet Palette.



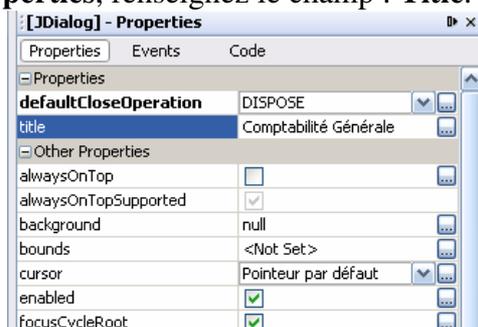
Lorsqu'un objet graphique est sélectionné à la souris, l'onglet Properties permet de manipuler ses propriétés (couleur, position, forme,...).



### 3.2. Renseigner les éléments de base de la fenêtre

Afin d'illustrer l'utilisation de l'onglet **Properties**, nous allons donner un nom à cette fenêtre en l'appelant : Comptabilité Générale.

Pour cela, dans l'onglet **Properties**, renseignez le champ : **Title**.



Ceci devrait vous donner après compilation :



### 3.3. Ajouter un bouton Quitter

Dans la liste des composants sélectionnez **JToggleButton** et dessinez ensuite un bouton sur la fenêtre.

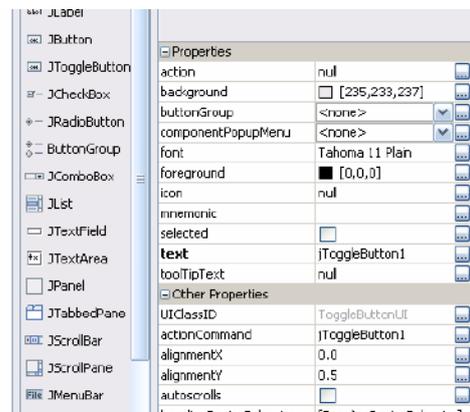


Vous devriez obtenir une fenêtre similaire à celle-ci :



Nous allons changer l'apparence de ce bouton en faisant apparaître Quitter (ne faisons pas comme Windows, ne l'appelons pas Demarrer ☺).

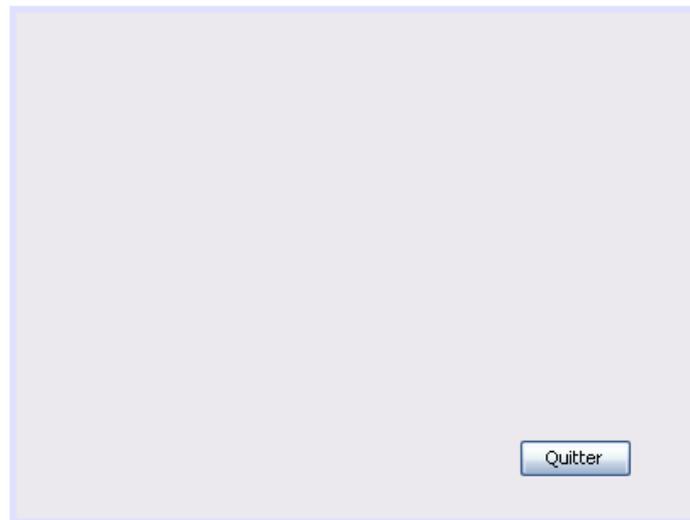
Sélectionnez le bouton à la souris, et examinez le panneau **Propriétés**. Modifiez ensuite le champ **Text**.



Après avoir saisi Quitter et une fois validé par la touche Entrée, vous obtiendrez :

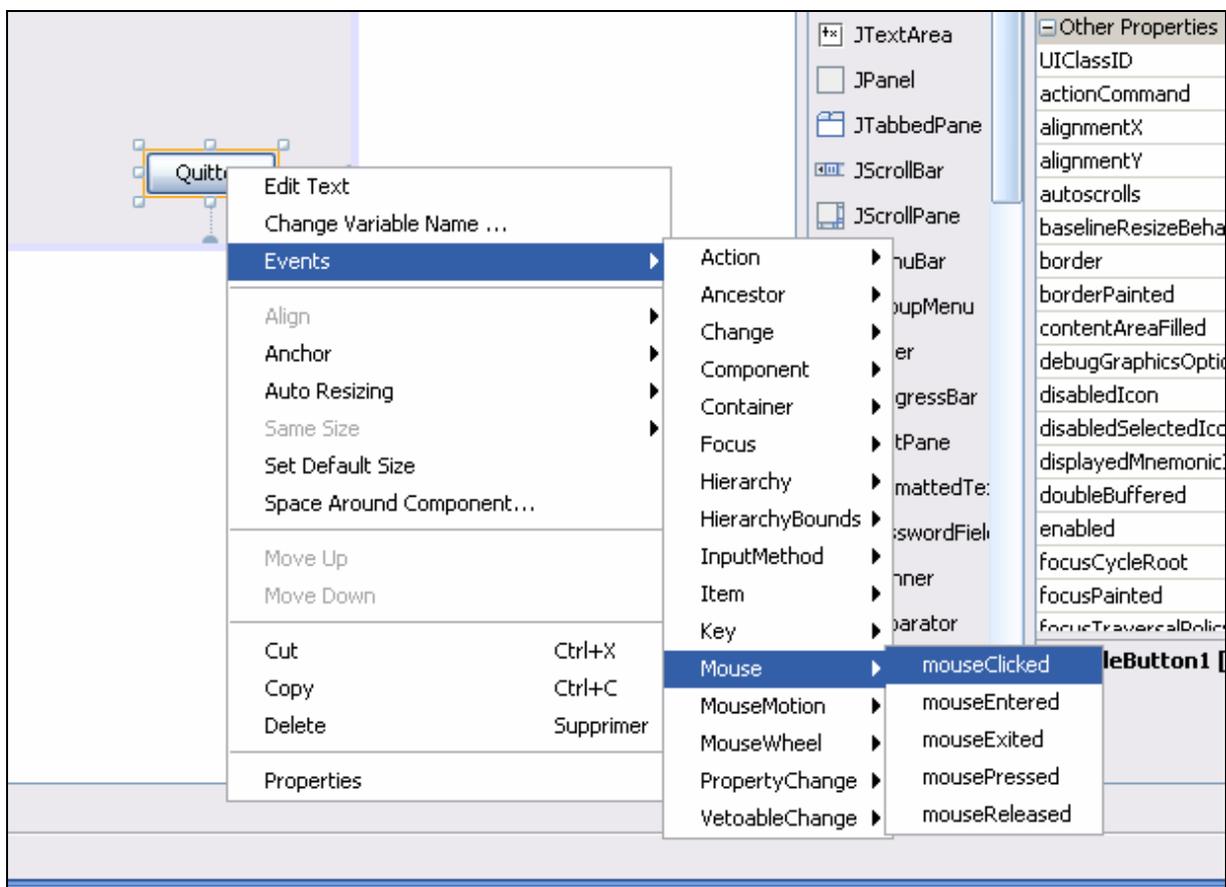


Et la fenêtre en cours de construction se présente comme suit (cela commence à prendre forme)...



### 3.4. Attachez une action au bouton Quitter

Faire un click droit sur le bouton pour faire apparaître le menu contextuel. Et choisir ensuite : **Events/Mouse/mouseClicked**. Cela signifie que nous désirons attacher un événement lorsque l'évènement click se produira sur le bouton.



Automatiquement, NetBeans vous propose de remplir le code qui sera associé à l'événement Click sur le bouton. Le nom de la procédure est clair de ce point de vue.



```
Generated Code

private void jToggleButton1MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
}
```

Je vous propose de mettre le code suivant :



```
private void jToggleButton1MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:

    System.exit(0);
}
```

Ce code permettra de terminer l'application sur l'événement Click.

Je vous laisse vérifier que cela fonctionner (prenez le temps de compiler et de tester l'application).

## 4. Communiquer avec l'utilisateur

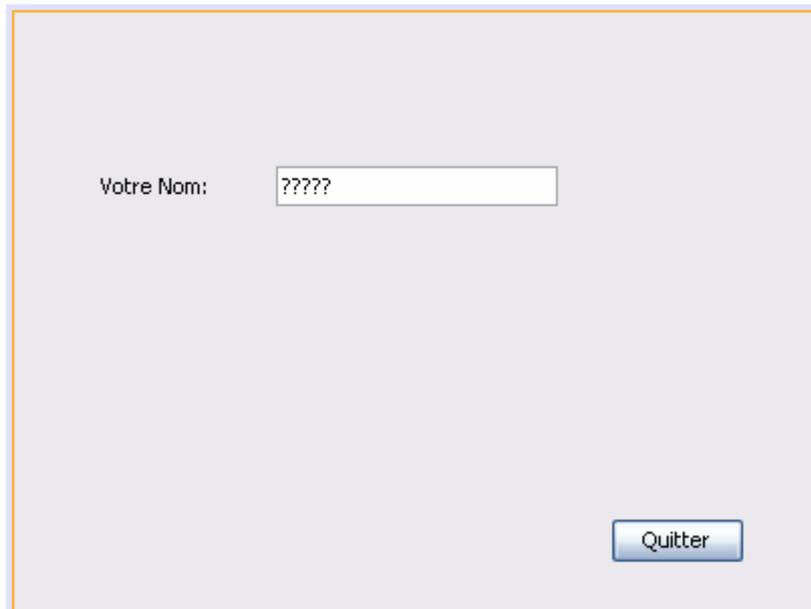
### 4.1. Lire les données saisies et afficher

Utilisez les JLabel et les JTextField pour construire une fenêtre similaire à la fenêtre ci-dessous.



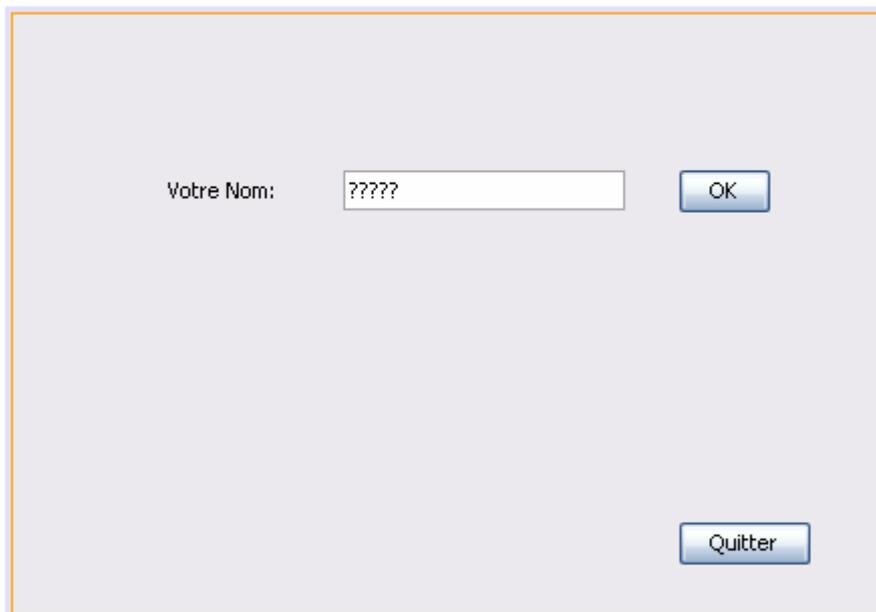
Je vous laisse deviner comment j'ai modifier le texte initial du JLabel et le contenu du JTextField (☺).

Vous devriez obtenir ceci :



A screenshot of a Java Swing dialog box with a light gray background. On the left, the text "Votre Nom:" is followed by a white text input field containing "?????". In the bottom right corner, there is a blue button with the text "Quitter".

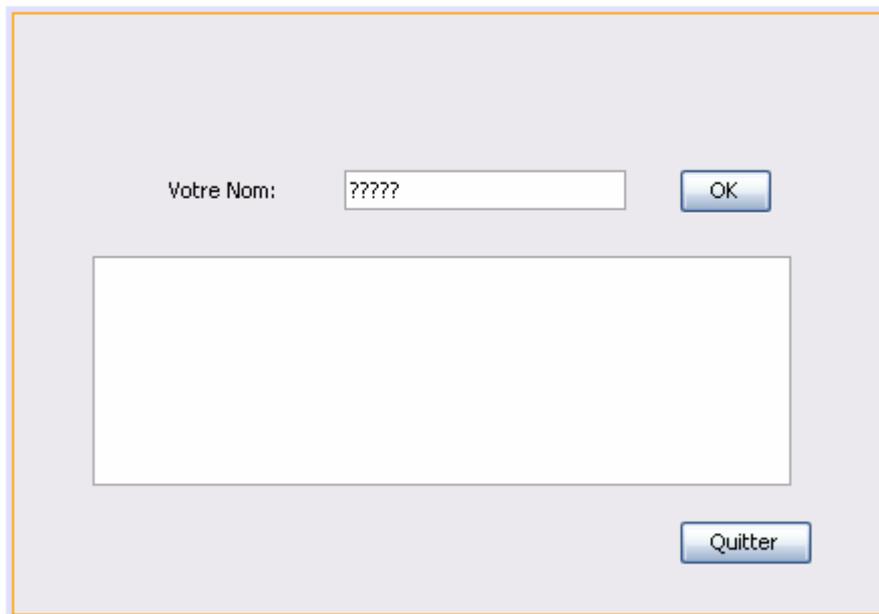
Ajoutez ensuite un bouton :



A screenshot of a Java Swing dialog box with a light gray background. On the left, the text "Votre Nom:" is followed by a white text input field containing "?????". To the right of the text field is a blue button with the text "OK". In the bottom right corner, there is a blue button with the text "Quitter".

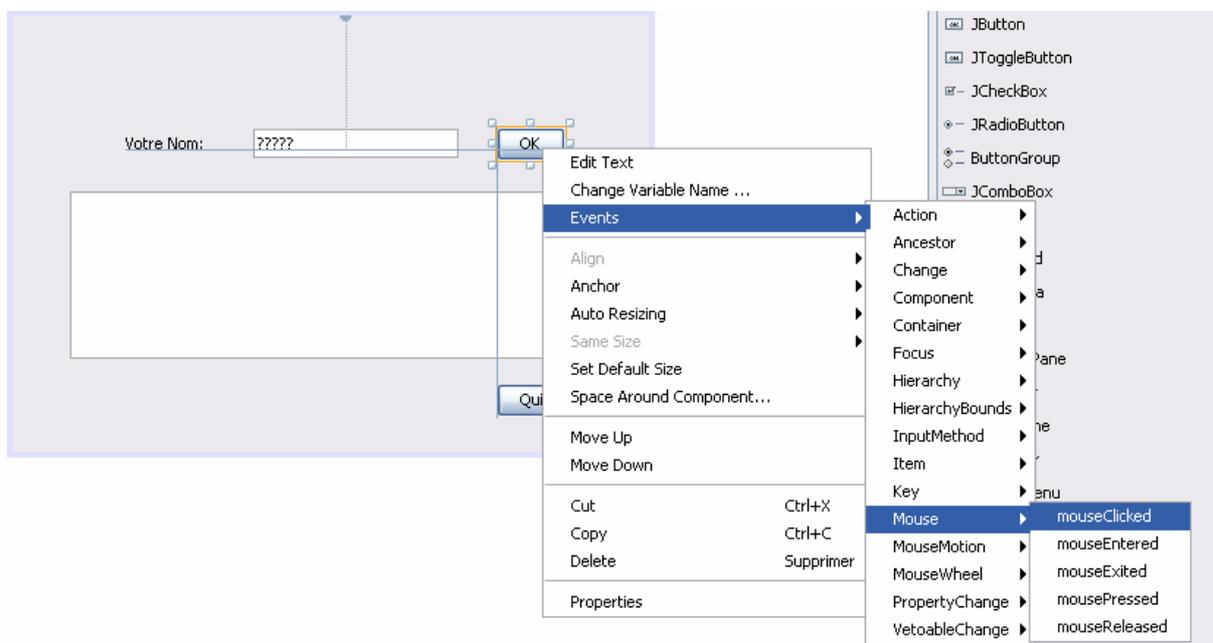
Et pour conclure ajoutez un JTextArea

Au final voici à quoi pourrait ressembler votre interface :



Nous allons réaliser l'opération (simple ☺ ) suivante : lorsque l'utilisateur « clique » sur le bouton **ok**, nous allons afficher dans le JTextArea (ce composant se comporte comme une console ☺) le message suivant : « Bonjour Monsieur XXXX ».

Pour cela faites comme précédemment en appelant le menu contextuel de votre bouton OK.



NetBeans devrait alors vous proposer ceci :

```
private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
}
```

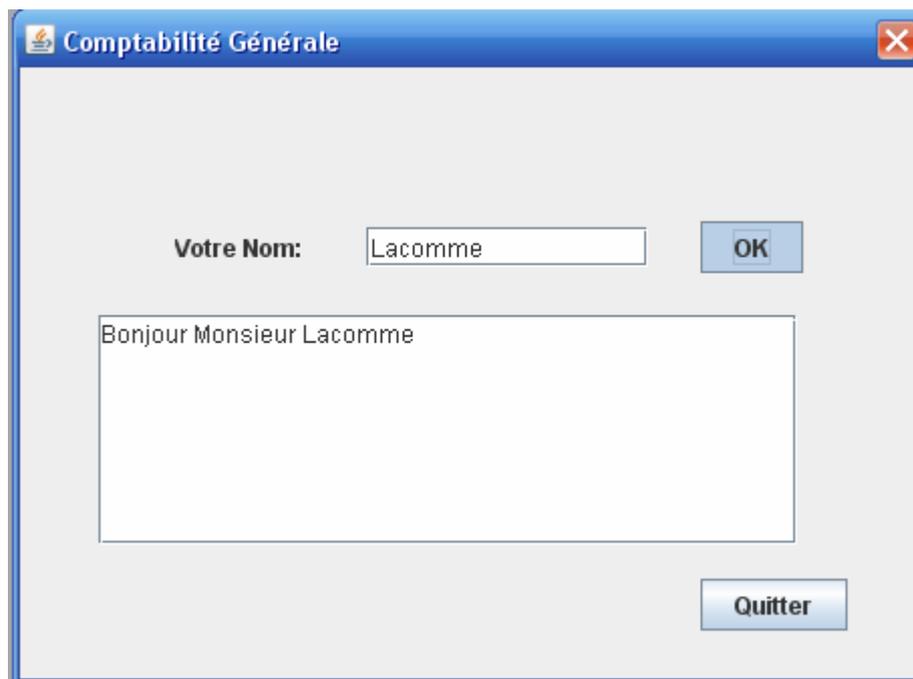
**Remarque :**

Le texte sur le bouton est bien OK, mais d'un point de vue informatique, le bouton s'appelle **JToggleButton2**.

Maintenant ajoutez le code Java suivant :

```
private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    String chaine;  
  
    chaine = jTextField1.getText();  
    jTextArea1.setText("Bonjour Monsieur "+chaine);  
}
```

Vous obtiendrez à l'exécution ceci :



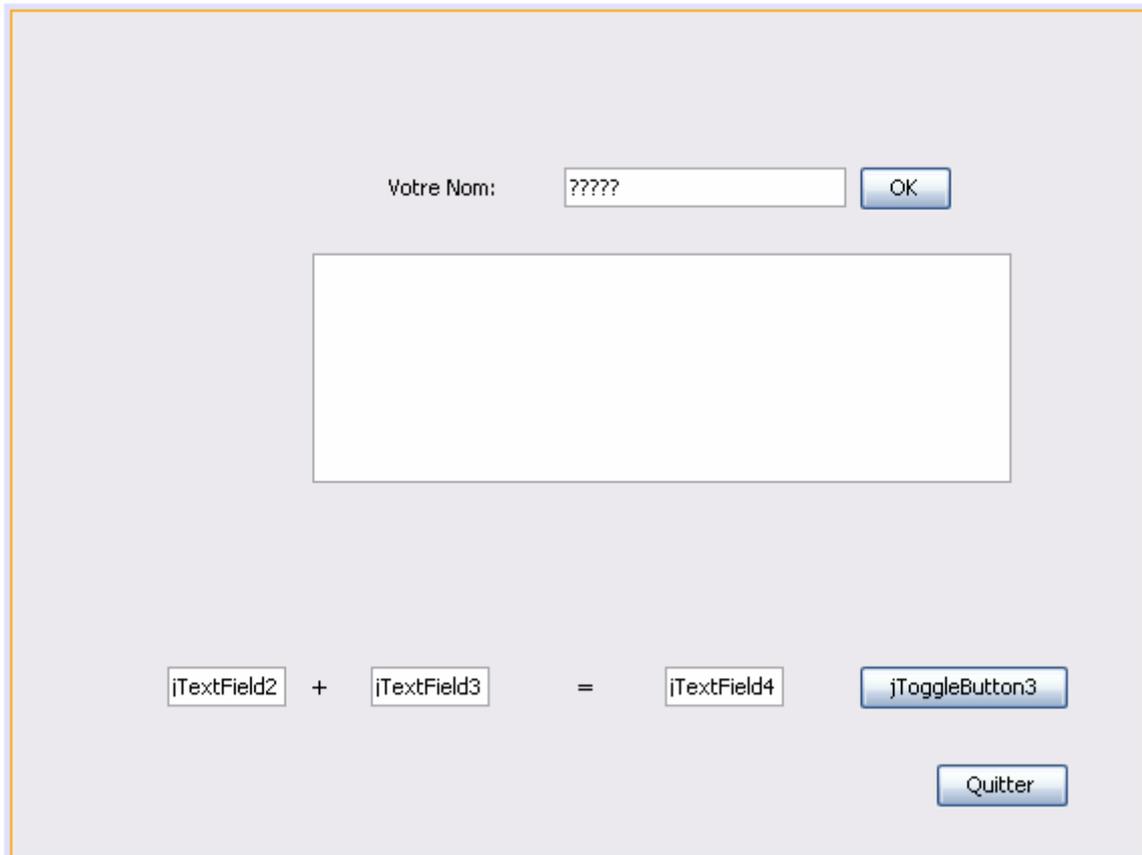
Pas mal pour un début...

## 4.2. Attention aux nombres entiers et aux nombres réels.

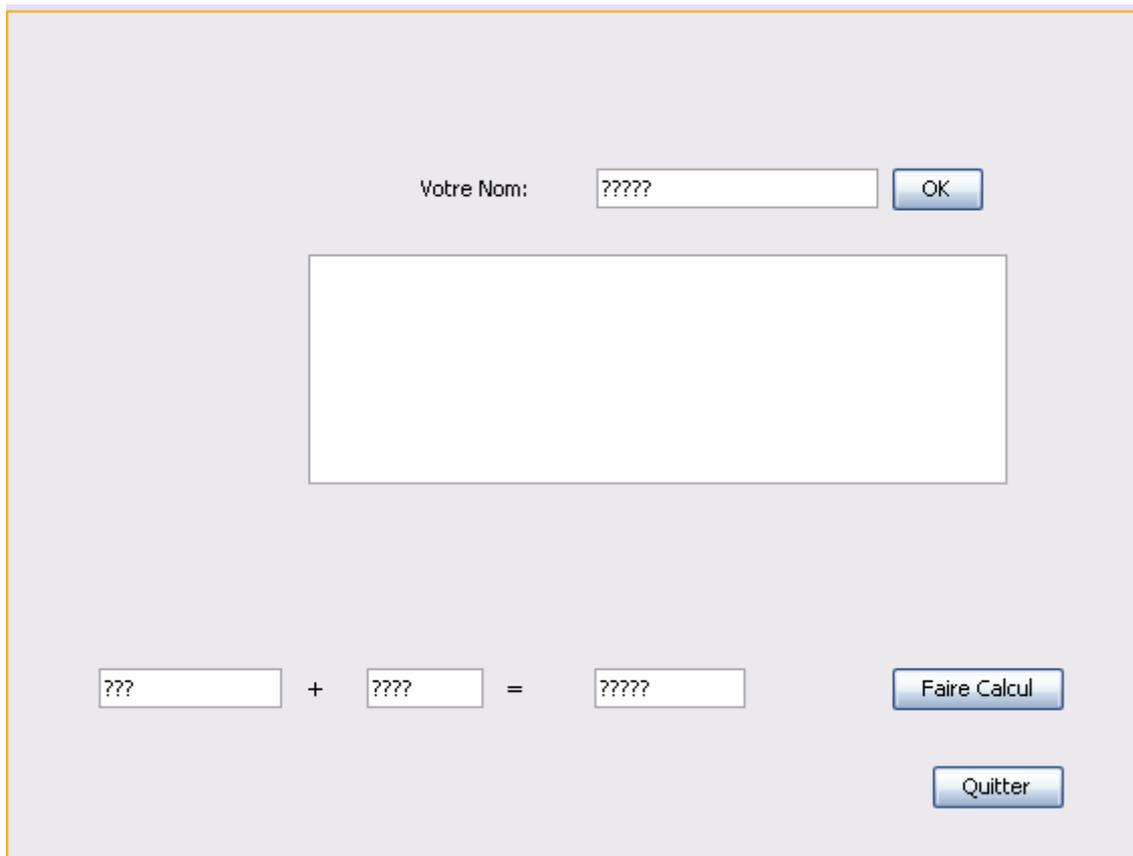
Ils sont un cauchemar à eux seuls....

Modifier la fenêtre précédente (vous n'avez plus besoin de moi pour le faire) :

Comme ceci dans un premier temps :

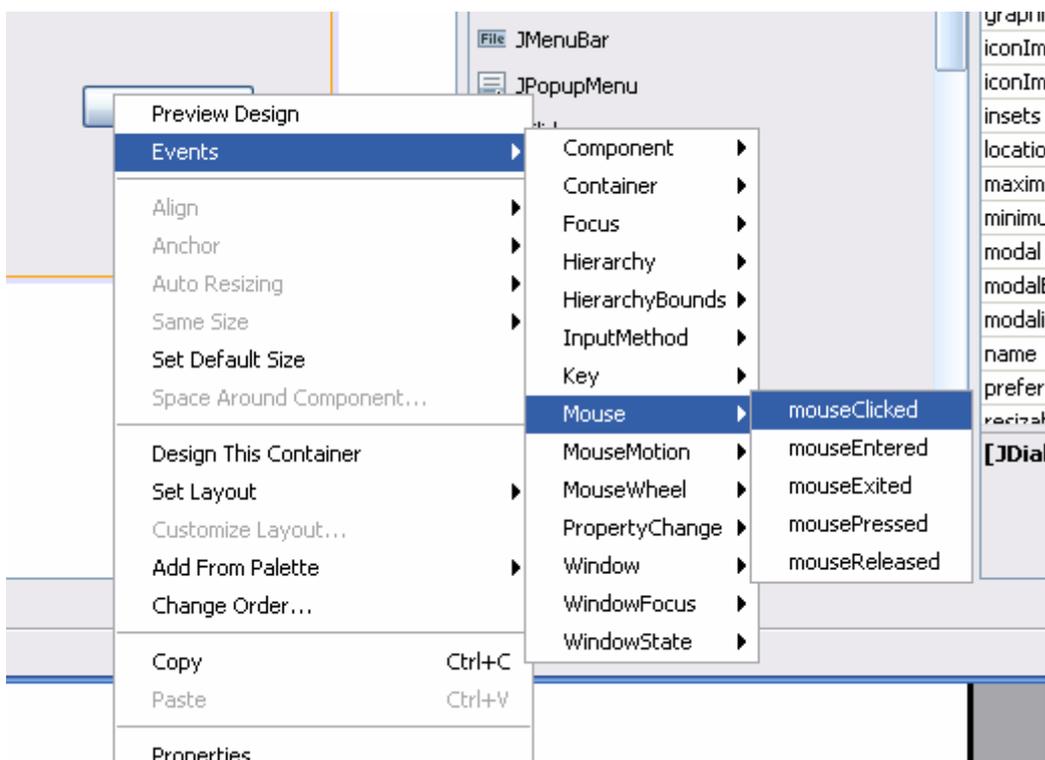


Puis finalement :



Le plus pénible est à venir.....

Déclarez un événement sur le bouton que vous venez de définir :



Et attaquons nous au code ... pas difficile mais particulièrement pénible...

Le code Java à inclure dans la procédure `jToggleButton2MouseClicked` doit ressembler à ceci :

```
int i,j,resultat;
String Chain1, Chain2, Chain3;

// récupération des deux données au format String...
Chain1 = jTextField2.getText();
Chain2 = jTextField3.getText();

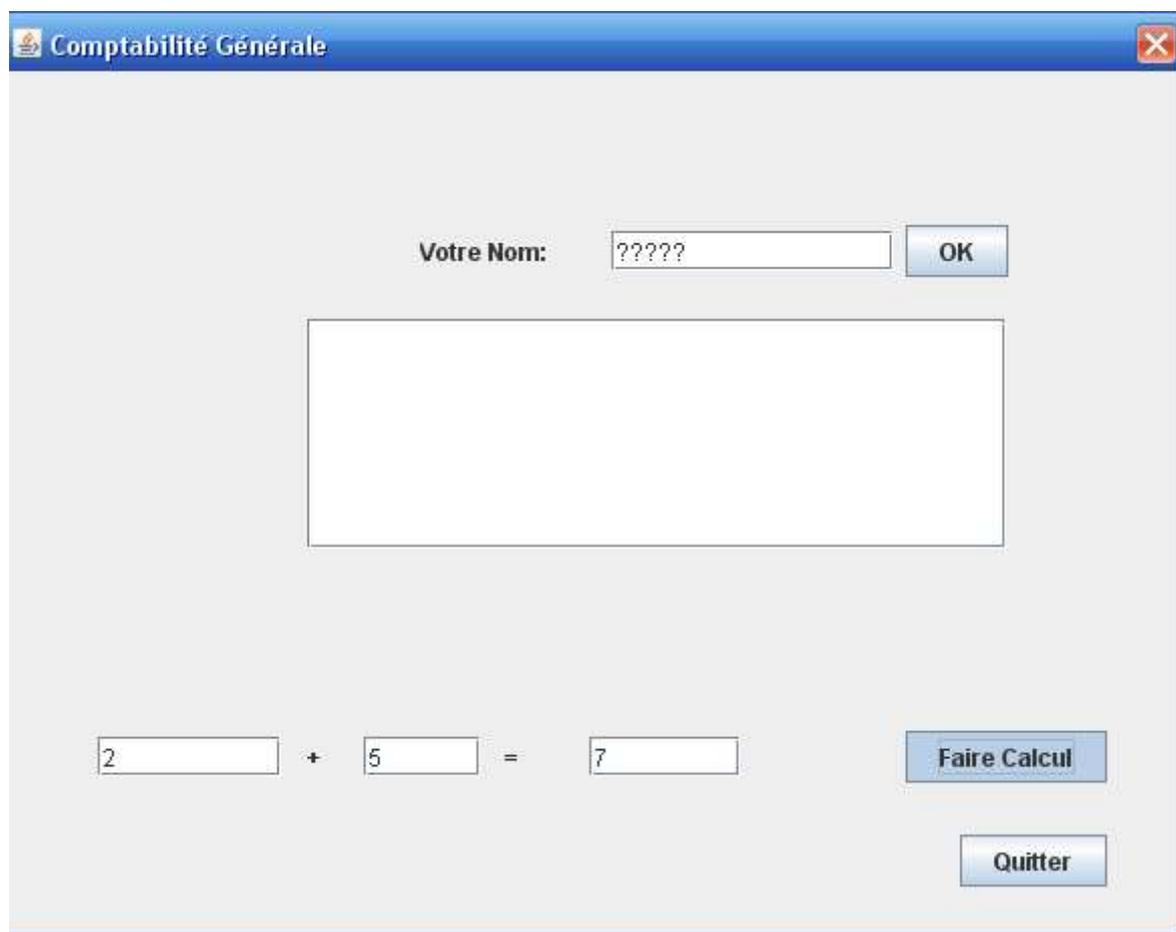
// conversion en int
i = Integer.parseInt(Chain1);
j = Integer.parseInt(Chain2);

// addition
resultat = i + j;

// conversion en chaine
Integer rr = new Integer(resultat);
Chain3 = rr.toString();

// on affiche le résultat
jTextField4.setText(Chain3);
}
```

Vous pourrez ensuite vérifier que les calculs sont justes :



## 5. Conclusion

Ce document vous aidera (je l'espère à faire vos premières interfaces).

Toutes les critiques sont les bienvenues y compris pour la correction des fautes de frappe et des fautes d'orthographe.