

Université de Jijel
Faculté des sciences exactes et de l'informatique
Département d'informatique

Application avec interface graphique (API Swing)

Objectif de ce TP

Construire un programme à interface graphique qui somme deux nombres réels.



The image shows a screenshot of a graphical user interface (GUI) window titled "Addition". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is light gray and contains three input fields and three buttons. The first input field is labeled "NB1", the second is labeled "NB2", and the third is labeled "Résultat". Below the input fields are two buttons labeled "Somme" and "Annuler". At the bottom right of the window is a button labeled "Quitter".

Etape 1

- Création du projet

- File > **New Project**
- Categories / **Java** & Projects/ **Java Application** > Next
- Project NameNommer le programme (eg. **Add**)
- **Décocher** (Create Main Class)
- Finish

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

Use Dedicated Folder for Storing Libraries

Libraries Folder:

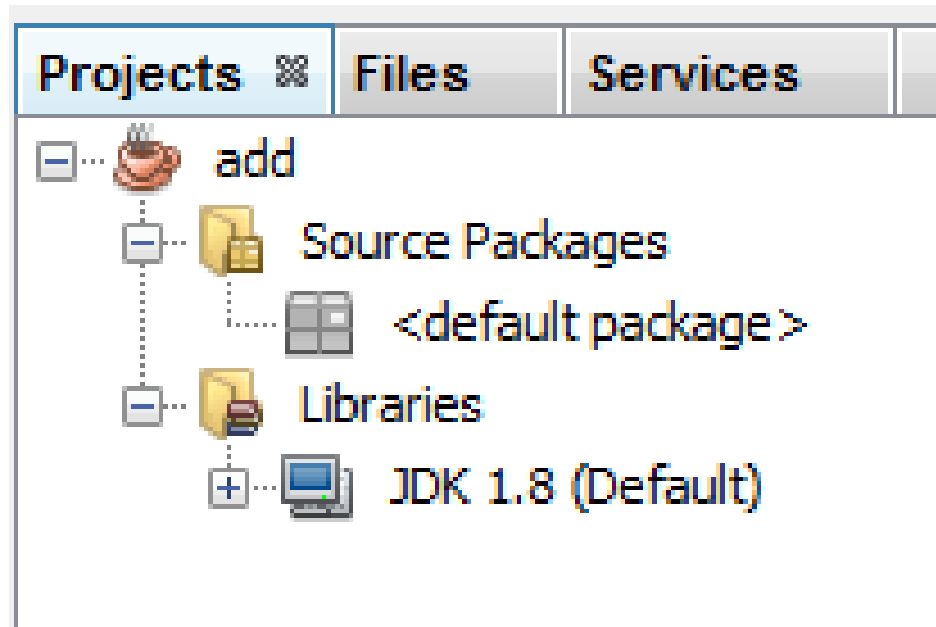
Different users and projects can share the same compilation libraries (see Help for details).

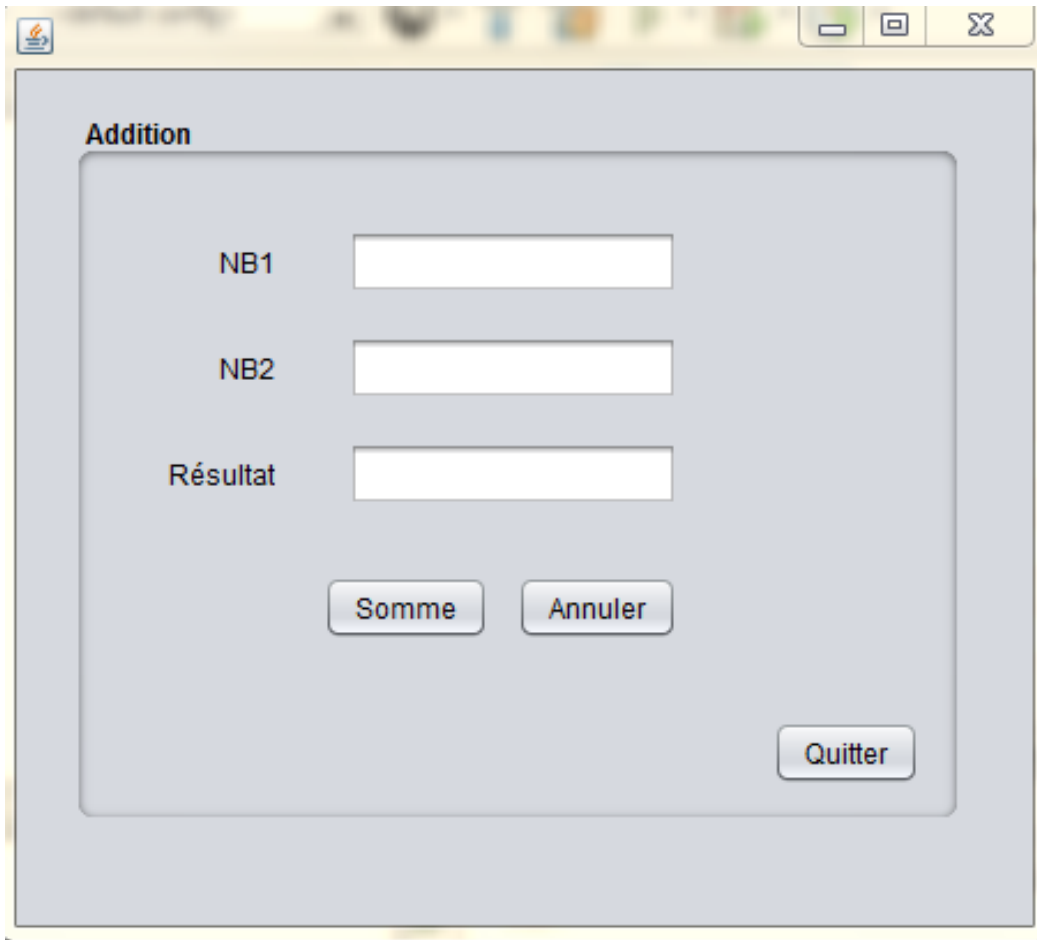
Create Main Class

< Back Next > **Finish** Cancel Help

4

Ce qu'on obtient





Fenêtre : JFrame

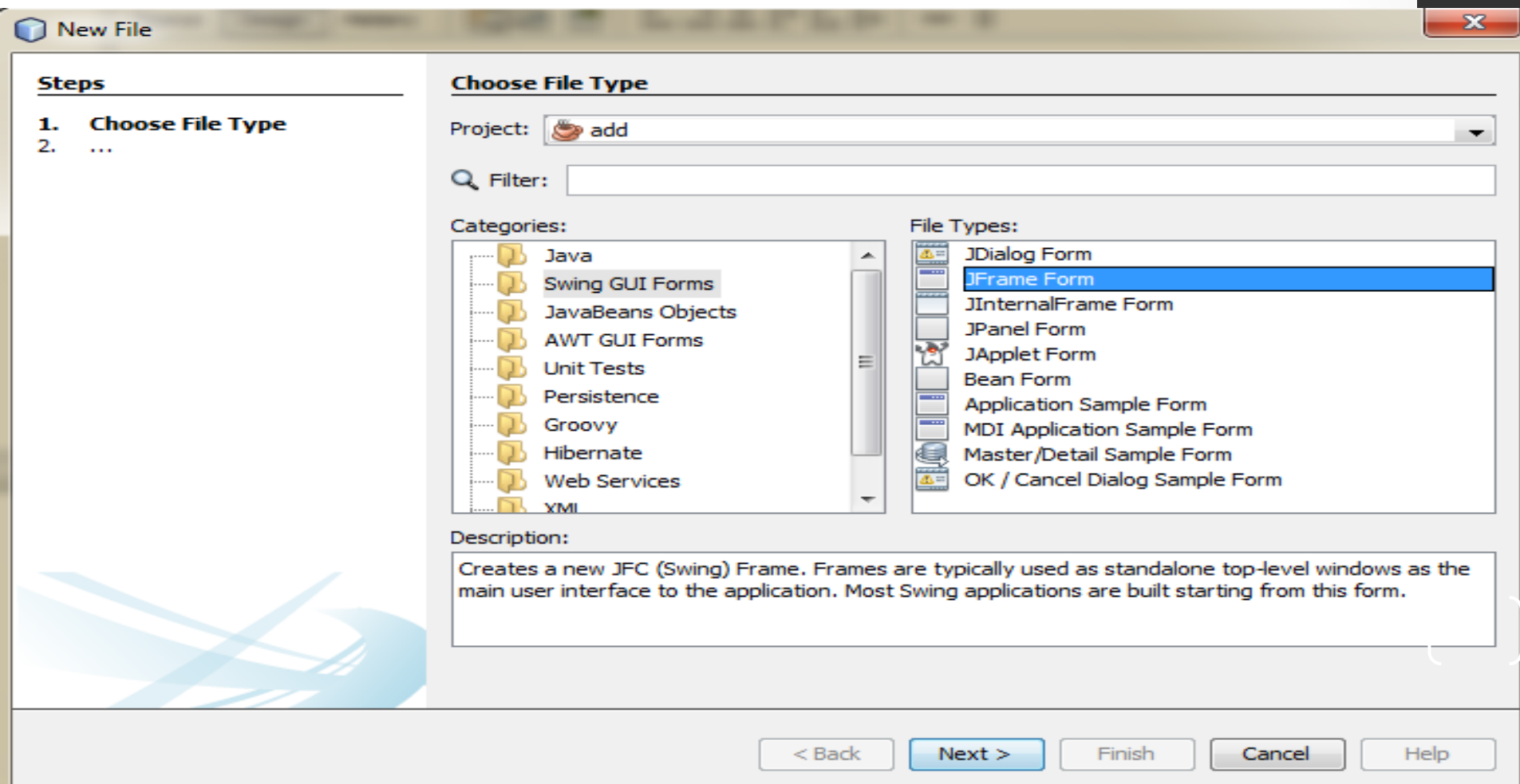


Rôle: conteneur de composants

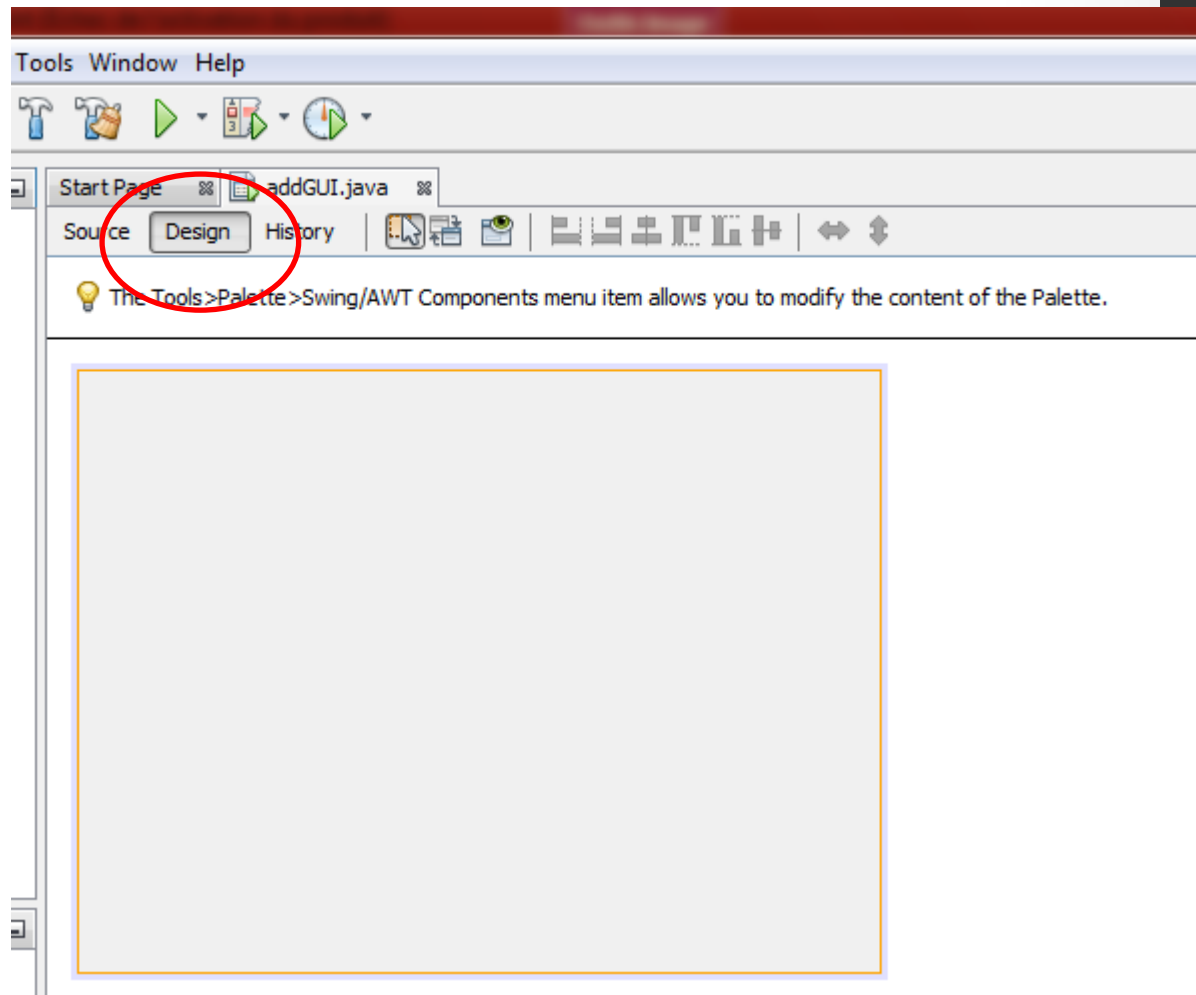
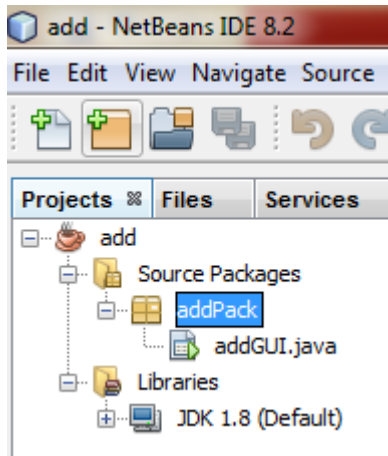
Etape 2

- Création de l'interface graphique

- File > New File
- Categories / Swing Gui Forms & File Types/ JFrame Form > Next
- Class Name Nommer la classe (eg, AddGUI)
- Package Nommer le package (eg, AddPack)
- Finish

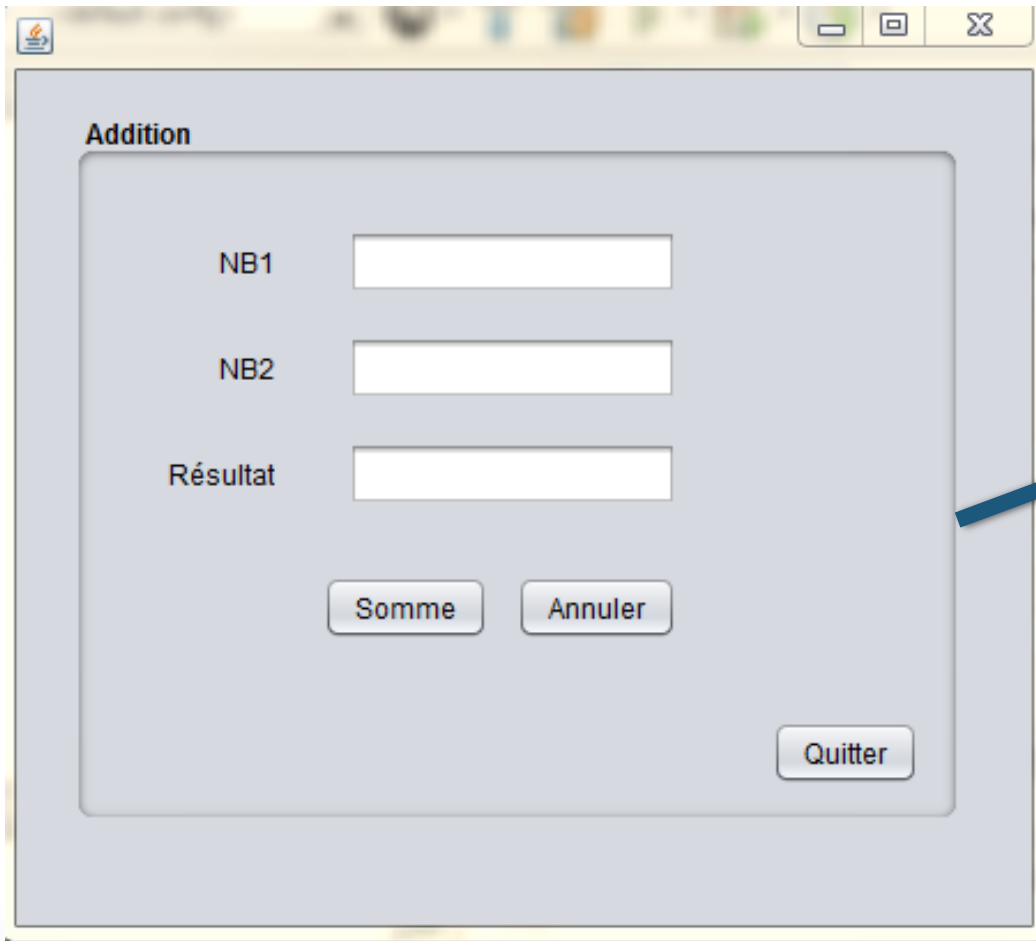


Ce qu'on obtient

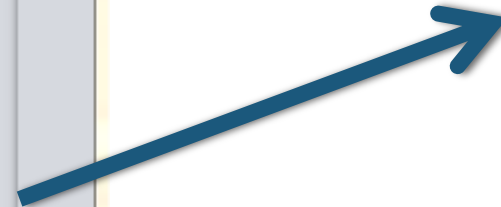


Etape 3

- Insertion des différents composants



Panneau (avec un titre) : Panel



Rôle: Sous fenêtre

jPanel1 [JPanel] - border

Set jPanel1's border property using:

Available Borders

- (No Border)
- Bevel Border
- Compound Border
- Empty Border
- Etched Border
- Line Border
- Matte Border
- Soft Bevel Border
- Titled Border

Properties

Border	[XpImageBorder]	...
Title	Addition	...
Color	■ [255,51,51]	...
Font	Tahoma 11 Plain	...
Justification	Default Justification	▼
Position	Default Position	▼

Titled Border
javax.swing.border.TitledBorder
A border with a String title in a specified position and justification.

OK Reset to Default Cancel Help

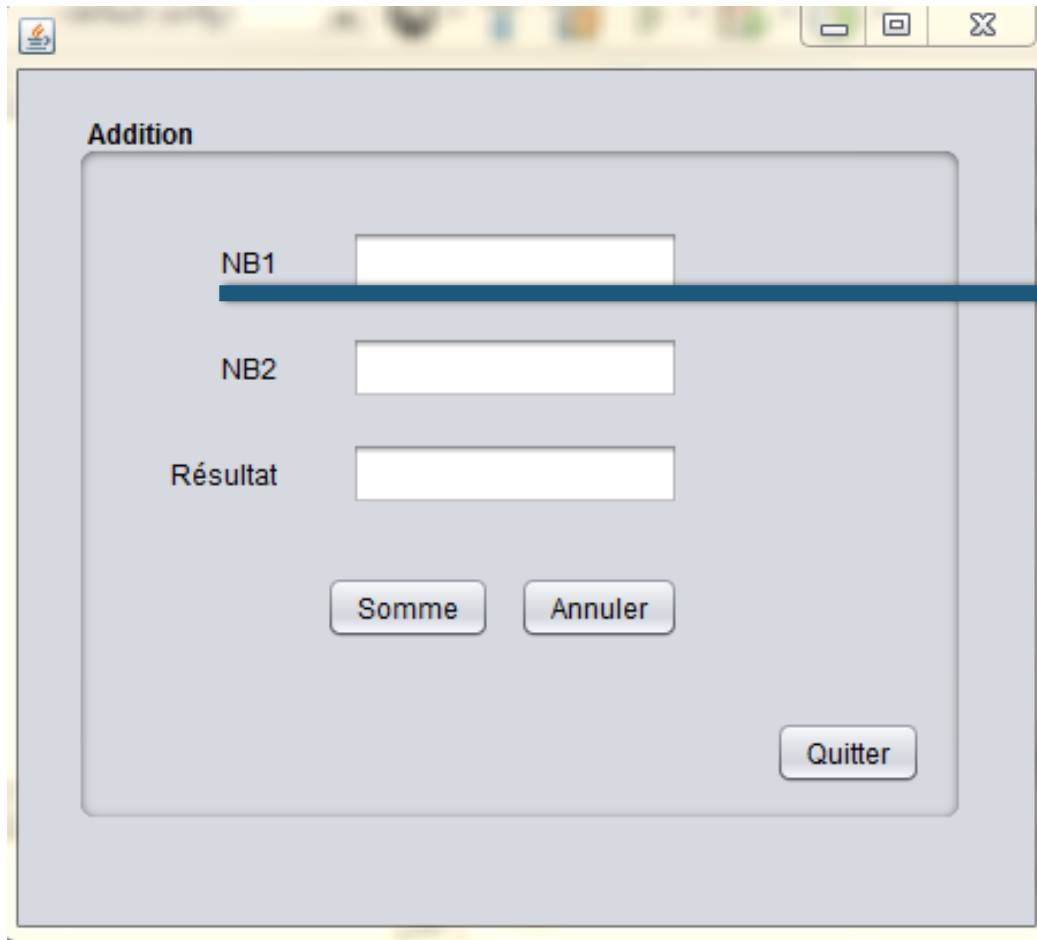
Palette

- Swing Containers
 - Panel**
 - Split Pane
 - Tool Bar
 - Internal Frame
- Swing Controls
 - Label
 - Toggle Button
 - Radio Button
 - Combo Box
 - Text Field
 - Tabbed Pane
 - Scroll Pane
 - Desktop Pane
 - Layered Pane
 - OK Button
 - Check Box
 - Button Group
 - List
 - Text Area

jPanel1 [JPanel] - Properties

Properties	Binding	Events	Code
background		<input type="checkbox"/> [240,240,240]	...
border		[TitledBorder]	...
foreground		<input checked="" type="checkbox"/> [0,0,0]	...
toolTipText			...
Other Properties			
UIClassID		PanelUI	...
alignmentX		0.5	...
alignmentY		0.5	...
autoscrolls		<input type="checkbox"/>	...

border
(javax.swing.border.Border) The component's border.



3 étiquettes : JLabel

Non modifiable par l'utilisateur

Palette ☒

Swing Containers

- Panel
- Split Pane
- Tool Bar
- Internal Frame
- Tabbed Pane
- Scroll Pane
- Desktop Pane
- Layered Pane

Swing Controls

- Label
- Toggle Button
- Radio Button
- Combo Box
- Text Field
- Button
- Check Box
- Button Group
- List
- Text Area

jLabel1 [JLabel] - Properties ☒

Properties Binding Events Code

background	<input type="checkbox"/>	[240,240,240]	...
displayedMnemonic			...
font		Tahoma 11 Plain	...
foreground	<input checked="" type="checkbox"/>	[0,0,0]	...
horizontalAlignment		LEADING	▼ ...
icon		<none>	▼ ...
labelFor		<none>	▼ ...
text		NB1	...
toolTipText			...

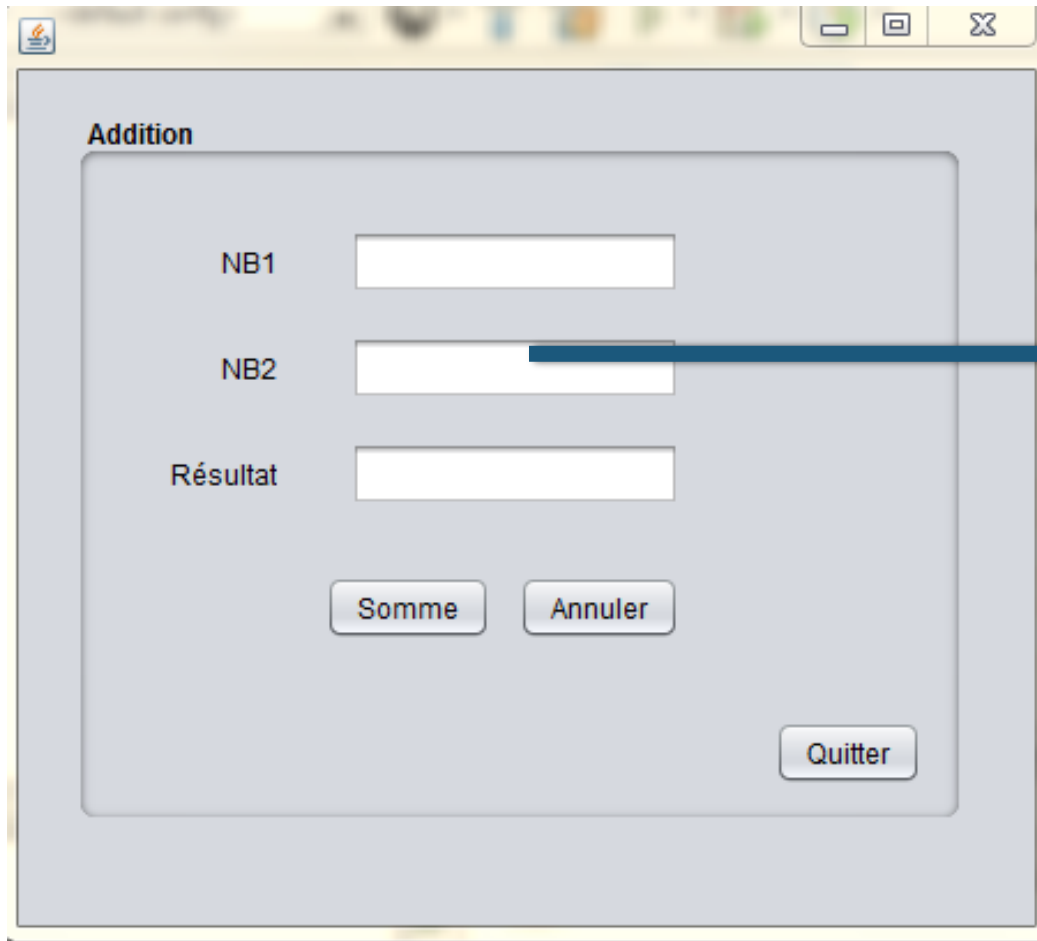
jLabel1 [JLabel] ?

De même pour

NB2

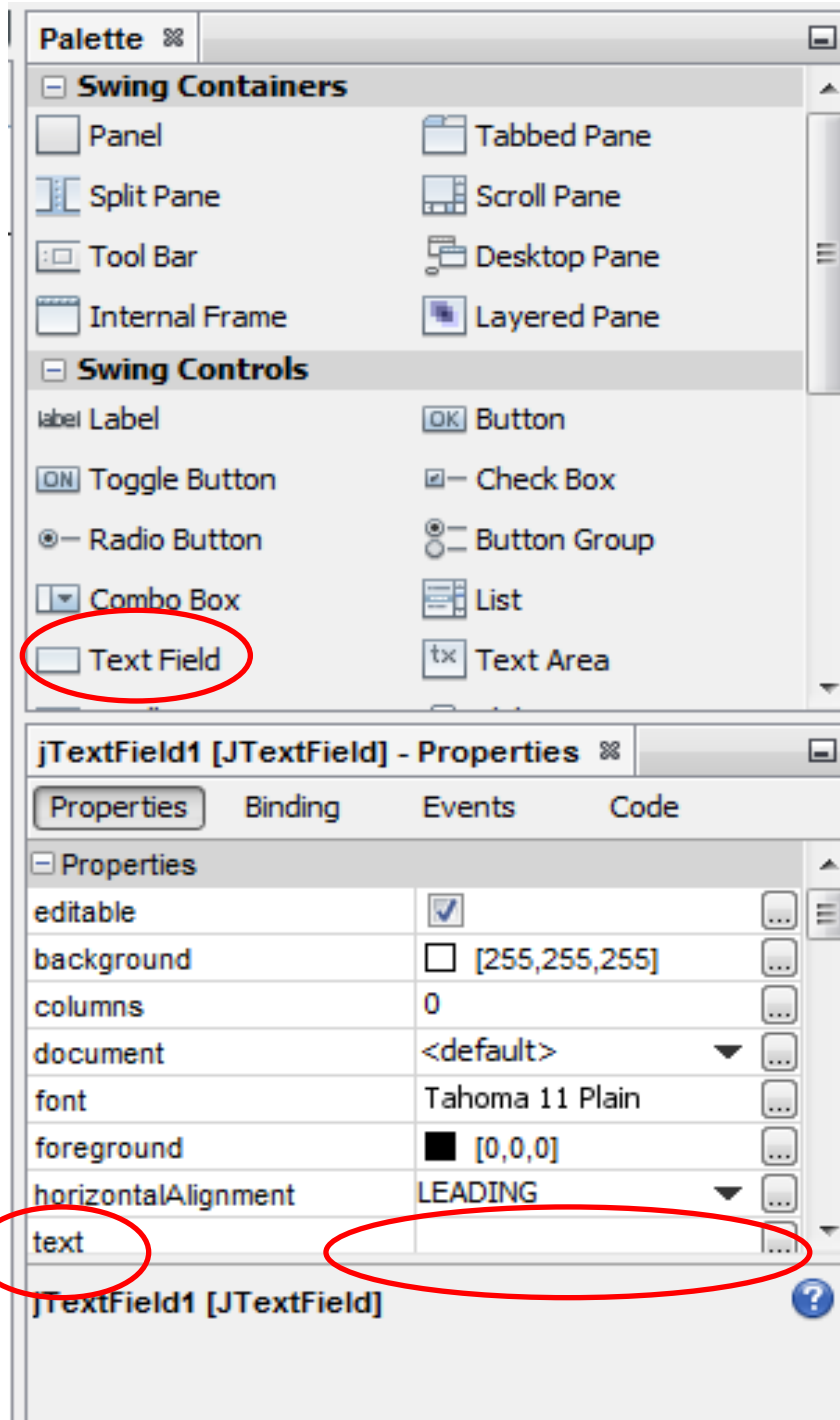
Résultat

À quoi ressemble notre interface ?



The image shows a Java Swing window titled "Addition". Inside the window, there are three text input fields labeled "NB1", "NB2", and "Résultat". Below the input fields, there are three buttons: "Somme", "Annuler", and "Quitter". A blue arrow points from the text "3 champs de texte: JTextField" to the three input fields.

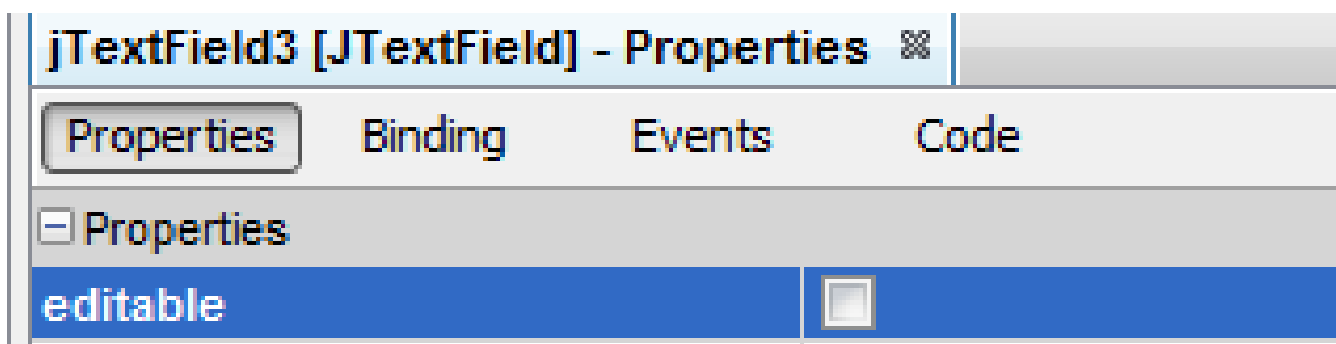
3 champs de texte:
JTextField



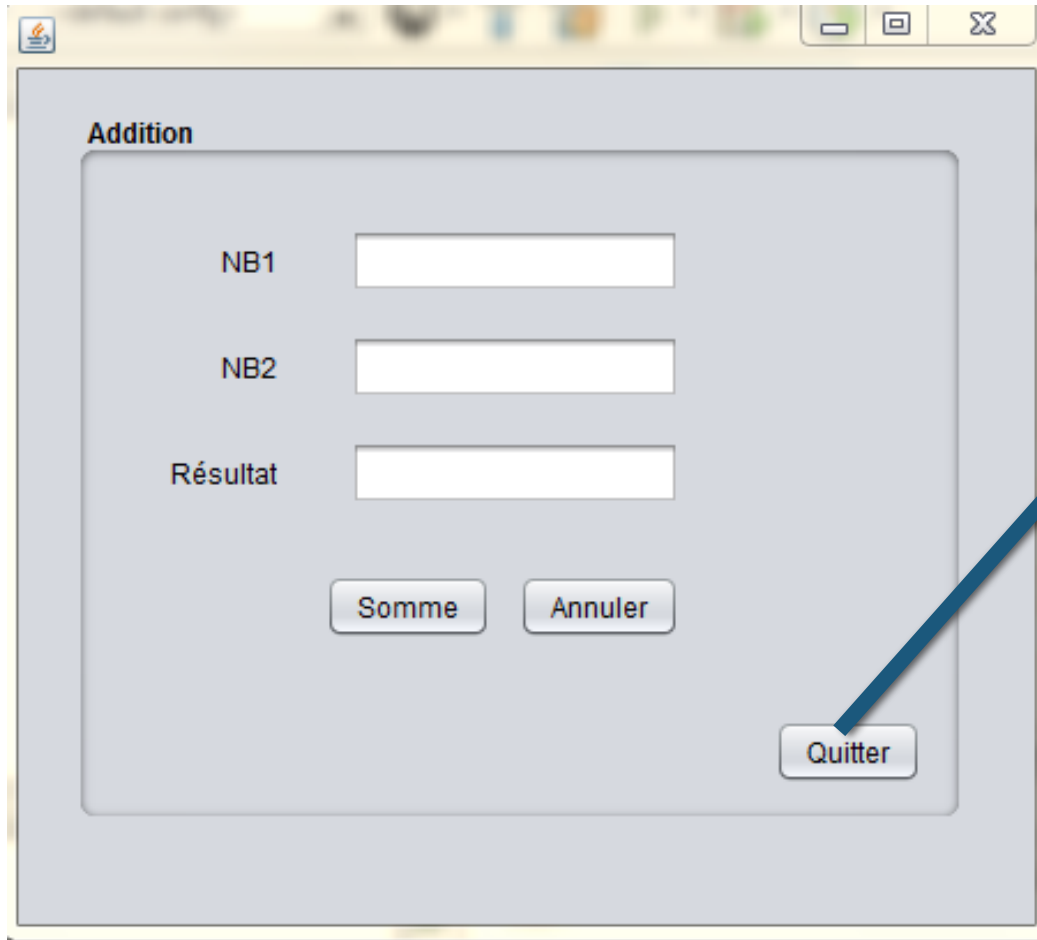
De même pour

Les deux autres champs

Pour un champ résultat , on peut désactiver la saisie en allant dans Propriétés/ (décocher editable) :

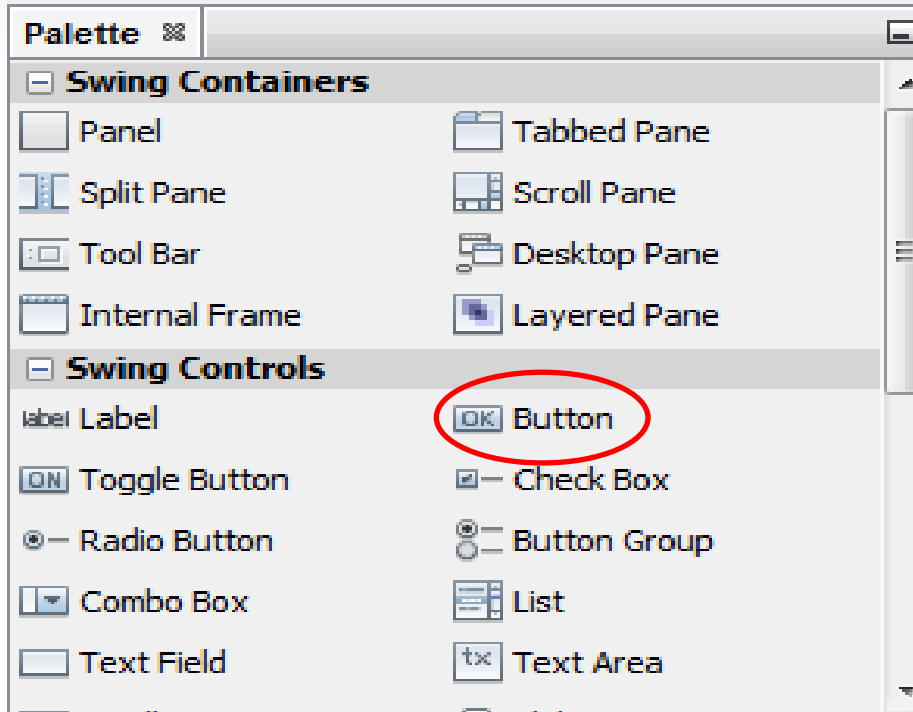


À quoi ressemble notre interface ?



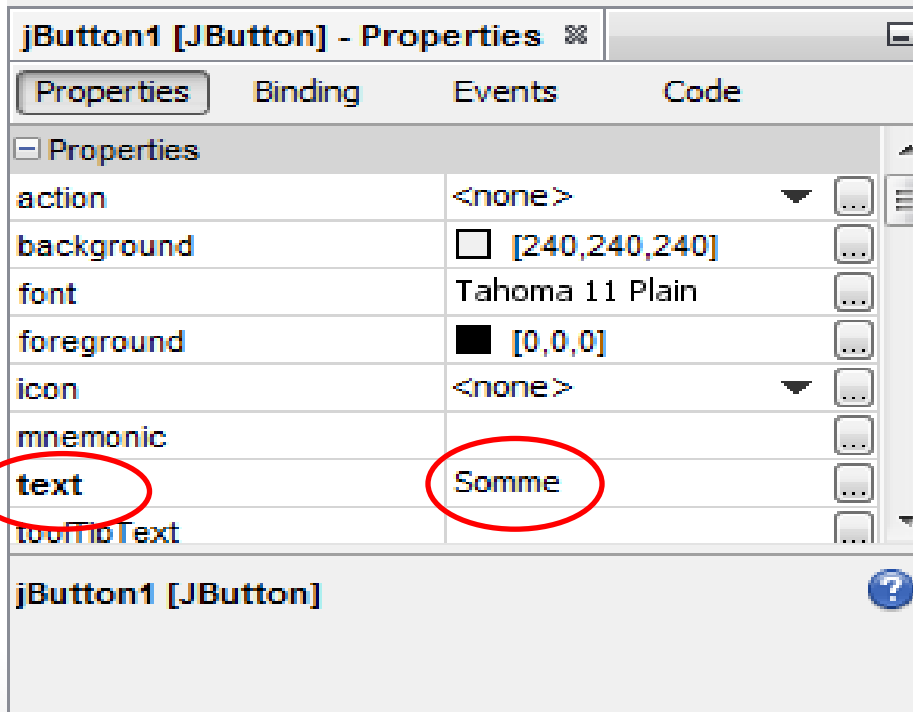
The image shows a Java Swing window titled "Addition". Inside the window, there are three text input fields labeled "NB1", "NB2", and "Résultat". Below the input fields, there are three buttons: "Somme", "Annuler", and "Quitter". The window has a standard title bar with minimize, maximize, and close buttons.

3 boutons: Button



De même pour

Les deux autres boutons



Etape 4

- Écriture du Code

Bouton Quitter

- Double clic sur le bouton

```
private void jButton3ActionPerformed(java.awt.event.MouseEvent evt) {  
  
    System.exit(0);  
  
}
```

Source

Design

History



```
1  [+ ...5 lines
6  package addPack;
7
8  [+ /**...4 lines */
12 public class addGUI extends javax.swing.JFrame {
13
14  [+ /** Creates new form addGUI ...3 lines */
17  [+ public addGUI () {...3 lines }
20
21  [+ /** This method is called from within the constructor to initialize
26  @SuppressWarnings ("unchecked")
27  [+ Generated Code
132
133 [- private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {
134     // TODO add your handling code here:
135     System.exit(0);
136 }
```

Bouton Annuler

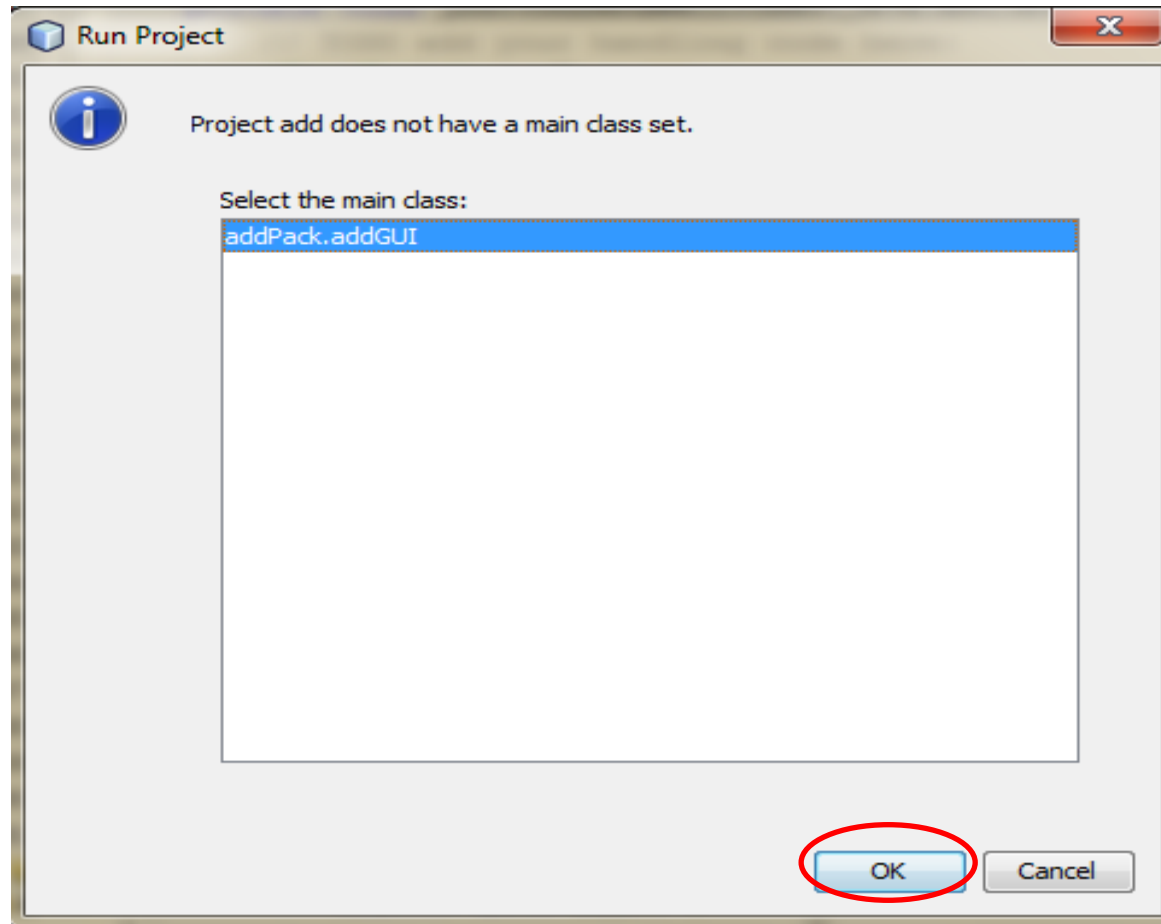
```
private void jButton2ActionPerformed(java.awt.event.MouseEvent evt) {  
    jTextField1.setText("");  
    jTextField2.setText("");  
    jTextField3.setText("");  
}
```

Bouton Somme

```
float n1,n2,res;  
n1=Float.parseFloat(jTextField1.getText());  
n2=Float.parseFloat(jTextField2.getText());  
res=n1+n2;  
jTextField3.setText(String.valueOf(res));
```

Exécution

- Run / Run Project (Add)



Exemple d'exécution

The image shows a Java Swing window titled "Addition". The window has a standard title bar with minimize, maximize, and close buttons. Inside the window, there is a light gray panel containing three text input fields and three buttons. The first field is labeled "NB1" and contains the value "5.5". The second field is labeled "NB2" and contains the value "6.2". The third field is labeled "Résultat" and contains the value "11.7". Below the input fields are two buttons: "Somme" and "Annuler". At the bottom right of the window is a "Quitter" button.

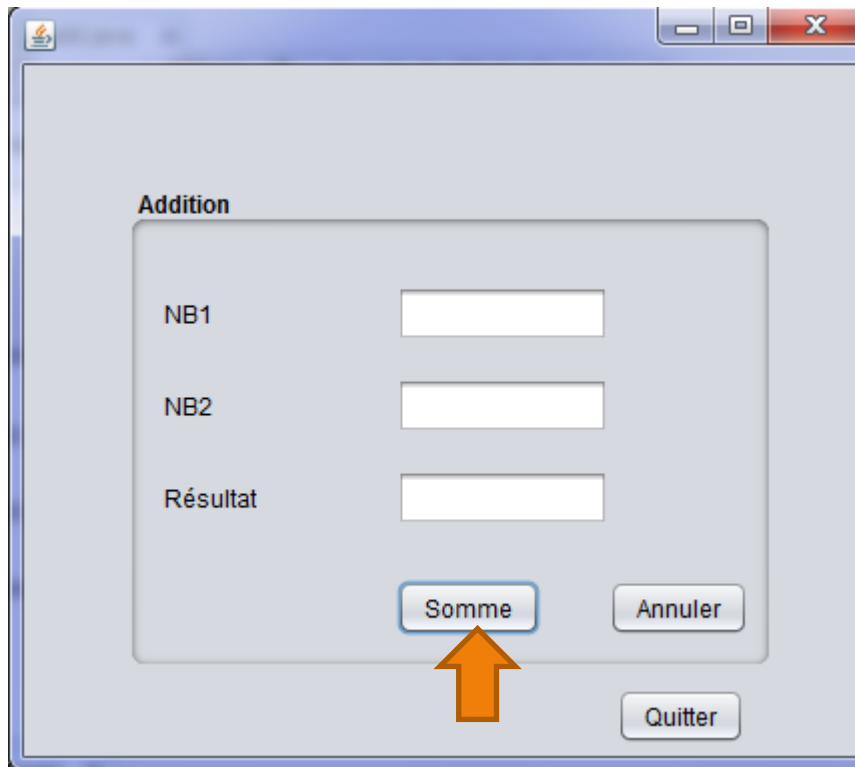
Label	Value
NB1	5.5
NB2	6.2
Résultat	11.7

Buttons: Somme, Annuler, Quitter

Traitement des exceptions

Exceptions ?

Le terme *exception* désigne tout événement arrivant **durant l'exécution** d'un programme interrompant son fonctionnement normal. Autrement, les exceptions sont des erreurs survenant lors de l'exécution d'un programme.



Exemple 1:

Lacer le calcul sans données d'entrée

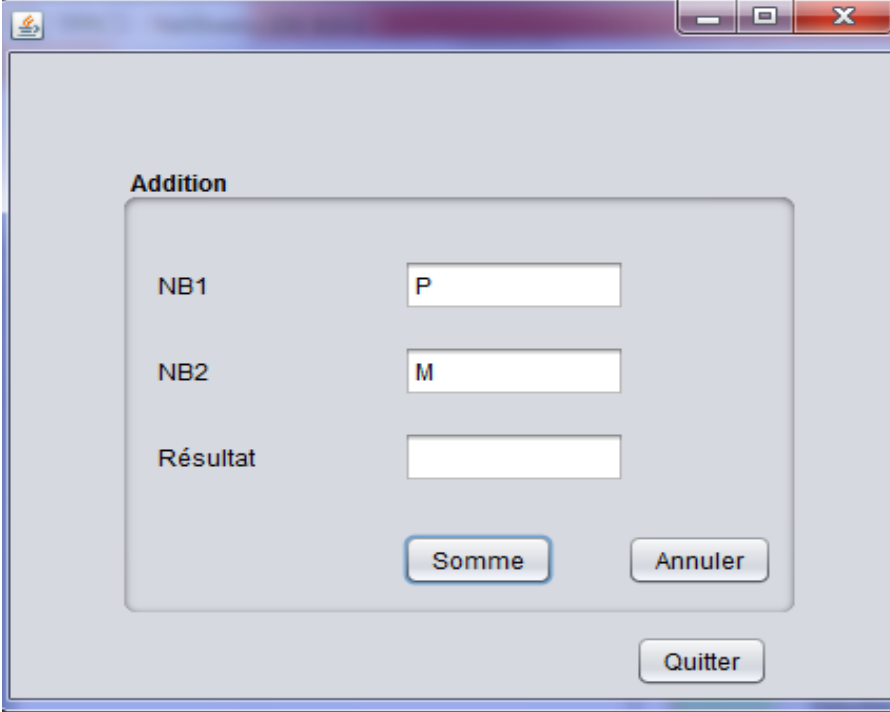


The image shows a Java Swing window titled "Addition". Inside the window, there is a panel with three text labels: "NB1", "NB2", and "Résultat", each followed by an empty text input field. Below the input fields, there are three buttons: "Somme", "Annuler", and "Quitter". The window has a standard title bar with minimize, maximize, and close buttons.

Exception in thread "AWT-EventQueue-0" java.lang.NumberFormatException:
empty String at java.lang.Float.parseFloat

Exemple 2:

Saisir des données non numériques



The screenshot shows a Java Swing window titled "Addition". Inside the window, there is a panel with the following elements:

- A label "Addition" at the top left of the panel.
- Two input fields: "NB1" with the value "P" and "NB2" with the value "M".
- A label "Résultat" next to an empty input field.
- Three buttons: "Somme", "Annuler", and "Quitter".

Exception in thread "AWT-EventQueue-0" java.lang.NumberFormatException:

For input string: "P" at java.lang.Float.parseFloat

Traitement de l'exception

Try {

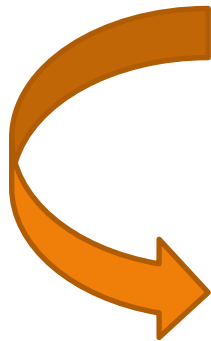
Instructions susceptibles de lever des exceptions

}

Catch (**Exception** e) {

Traitement de l'exception

}

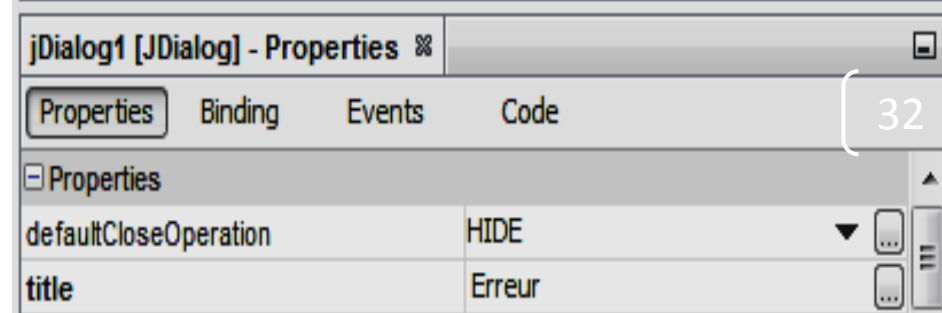
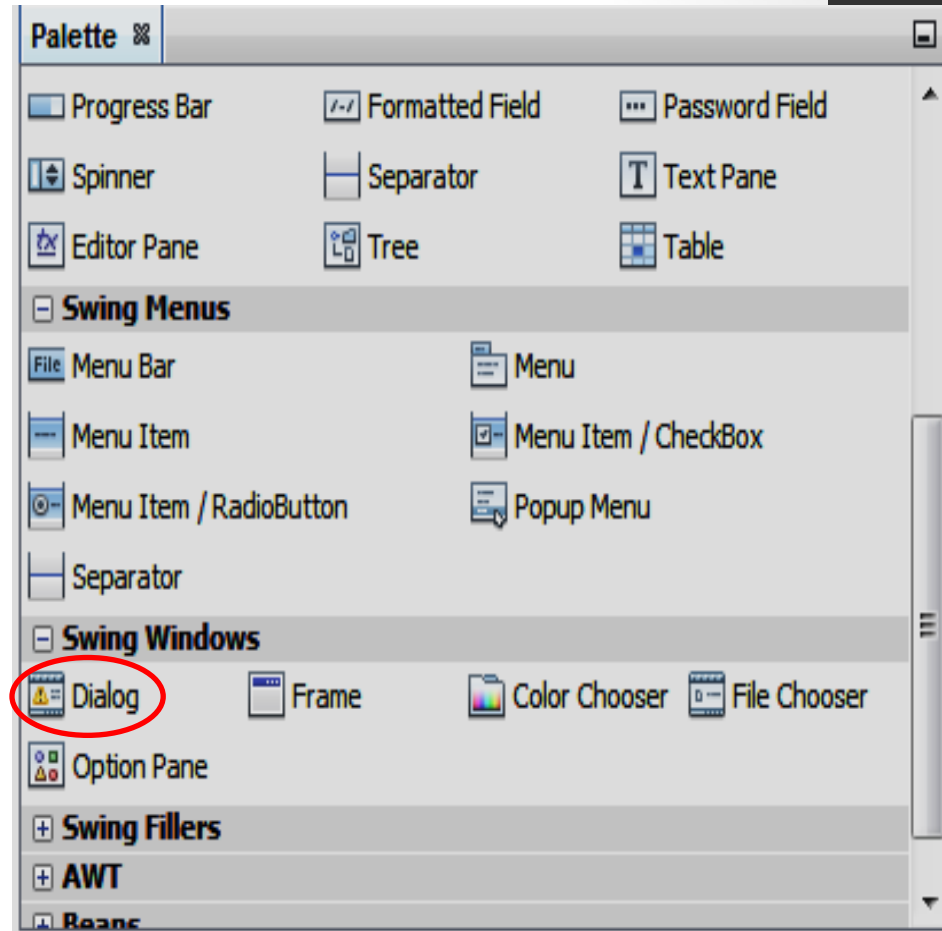
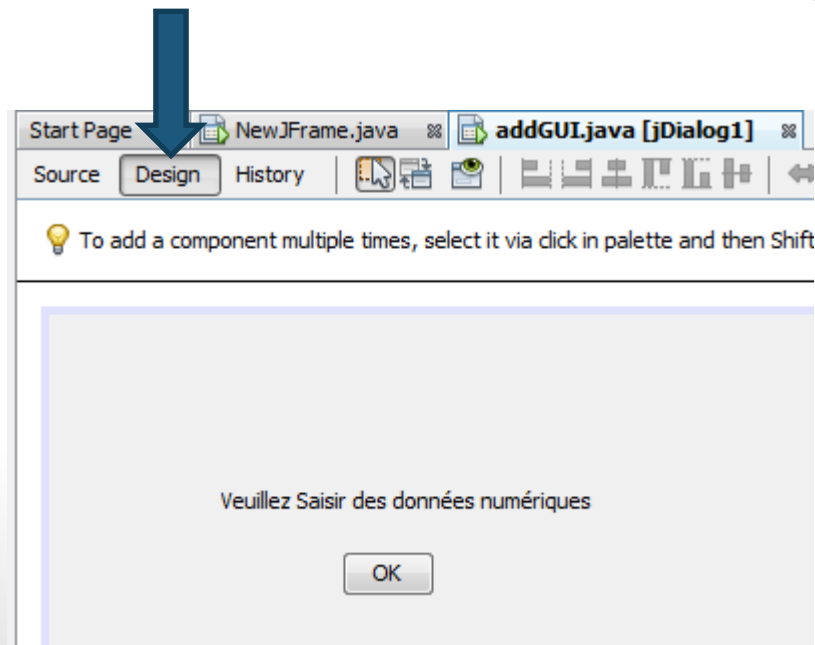
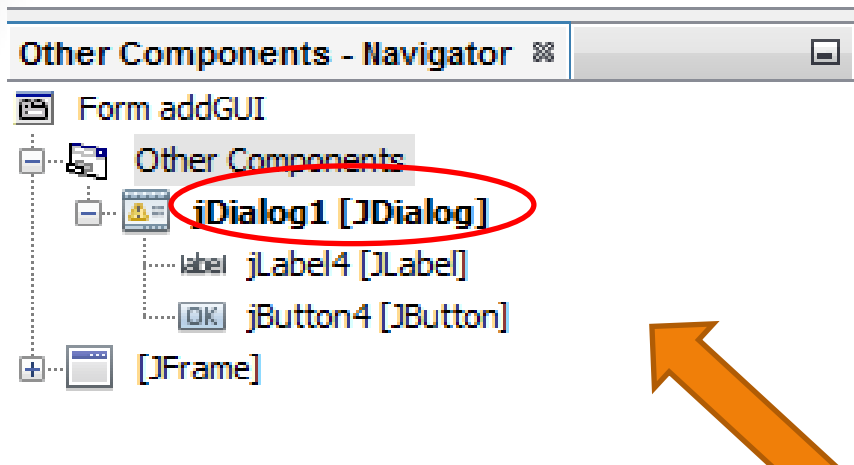


On veut afficher **un message**

(une boîte de dialogue)

pour avertir l'utilisateur

Création de la boîte de dialogue



Code pour le bouton Ok

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    jDialog1.setVisible(false);  
}
```

**Masquer ou fermer
la boîte de dialogue**

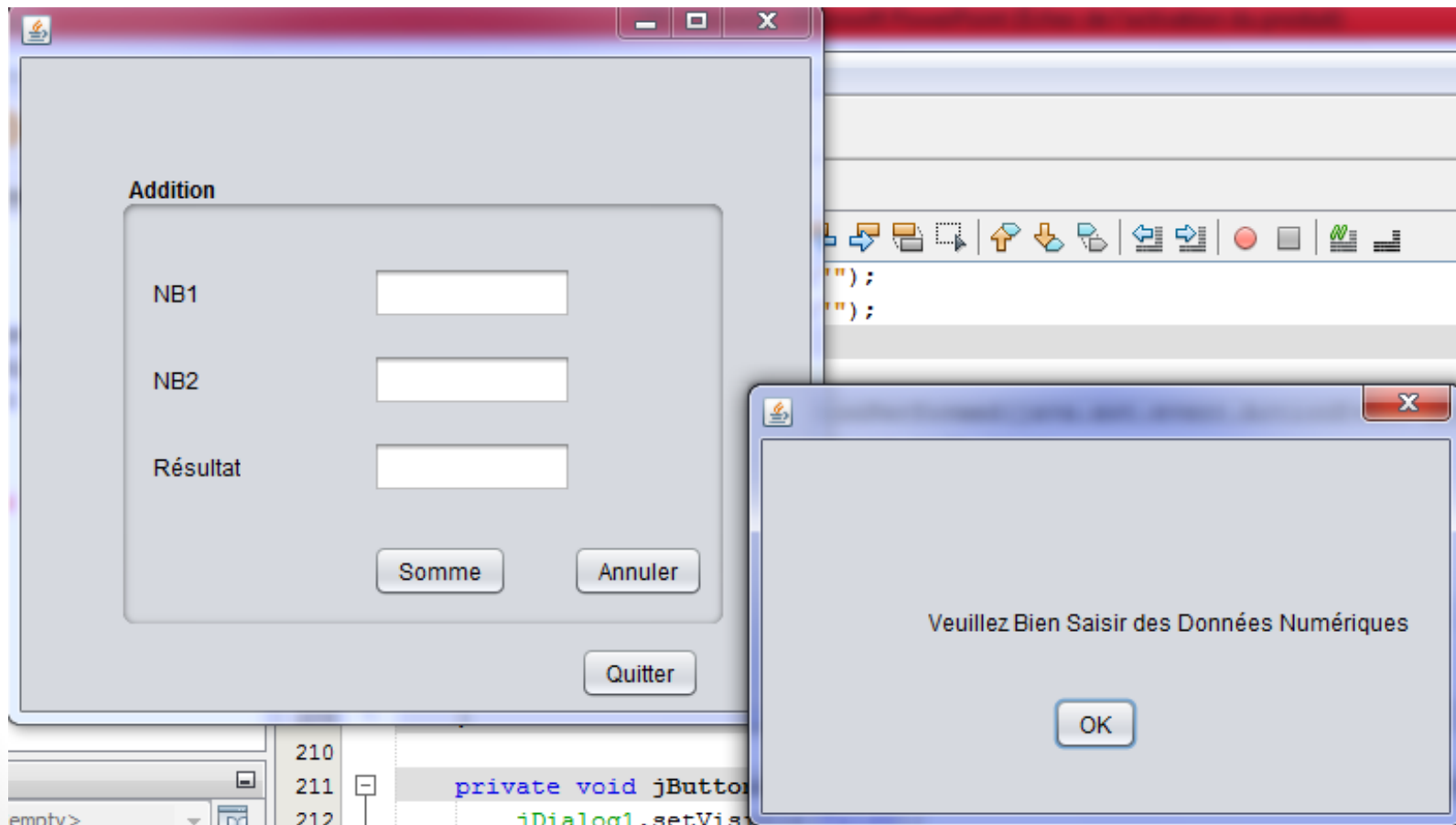
Bouton Somme : Insertion du try - catch

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try{  
        float n1,n2,res;  
        n1=Float.parseFloat(jTextField1.getText());  
        n2=Float.parseFloat(jTextField2.getText());  
        res=n1/n2;  
        jTextField3.setText(String.valueOf(res));  
    } catch (NumberFormatException e){  
        jDialog1.setVisible(true);  
    }  
}
```

Type de l'exception

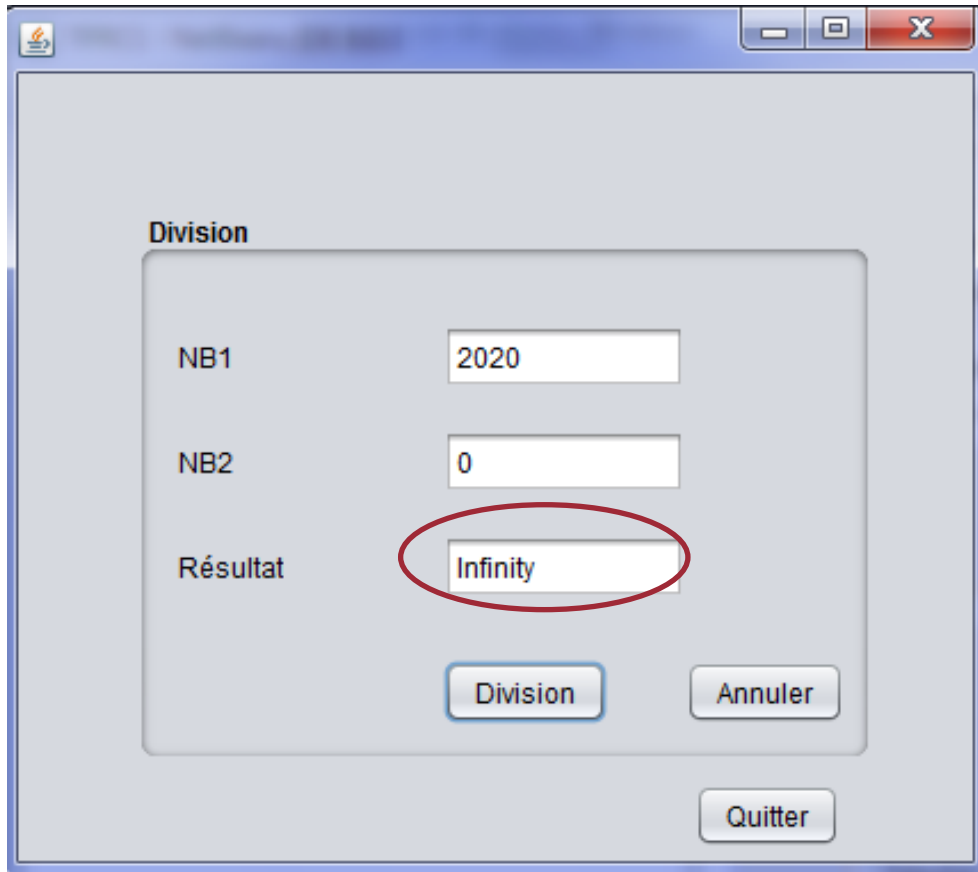
Visualiser la boite de dialogue

Exécuter ...



Remplacer la somme par une division

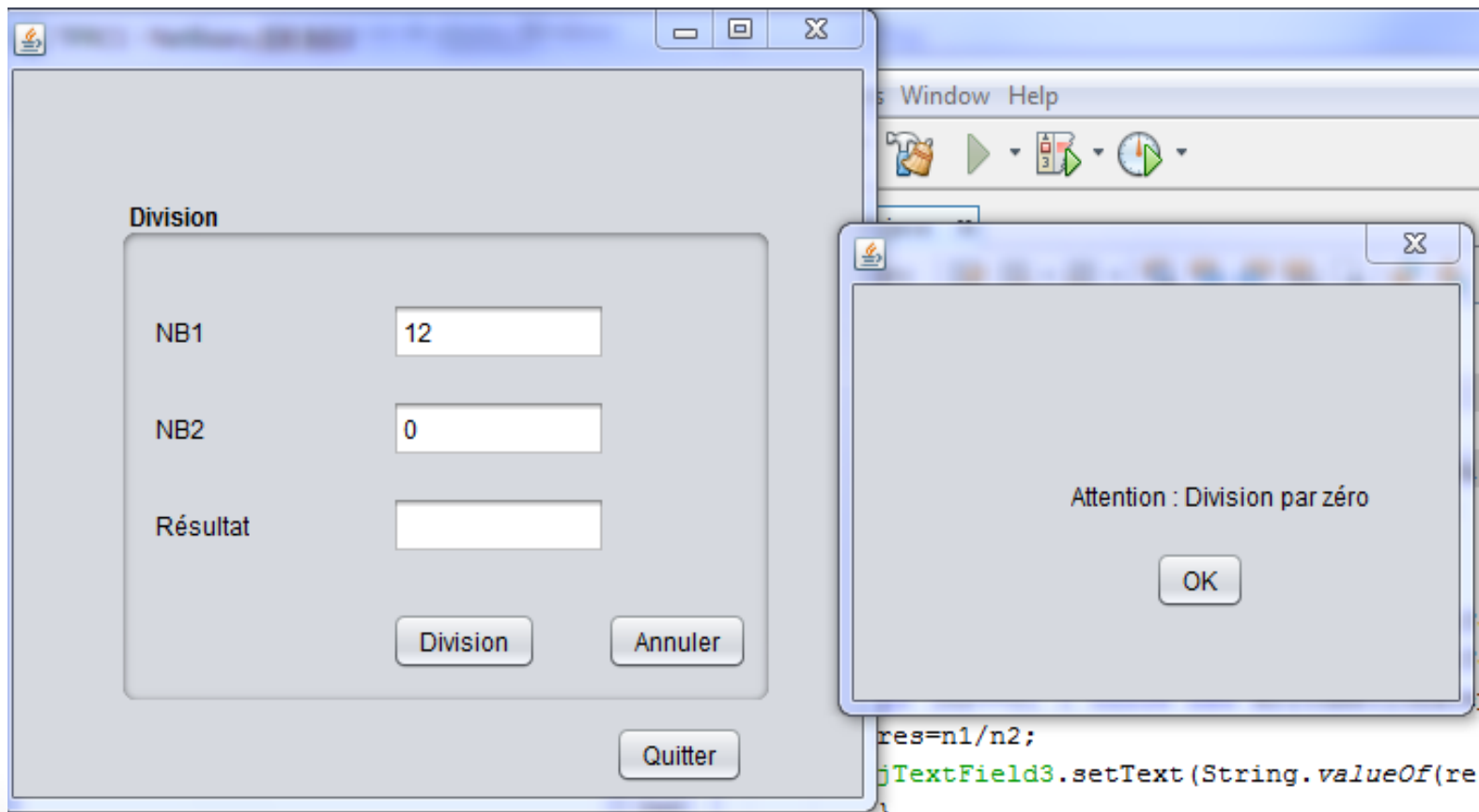
➤ Essayer la division par zéro



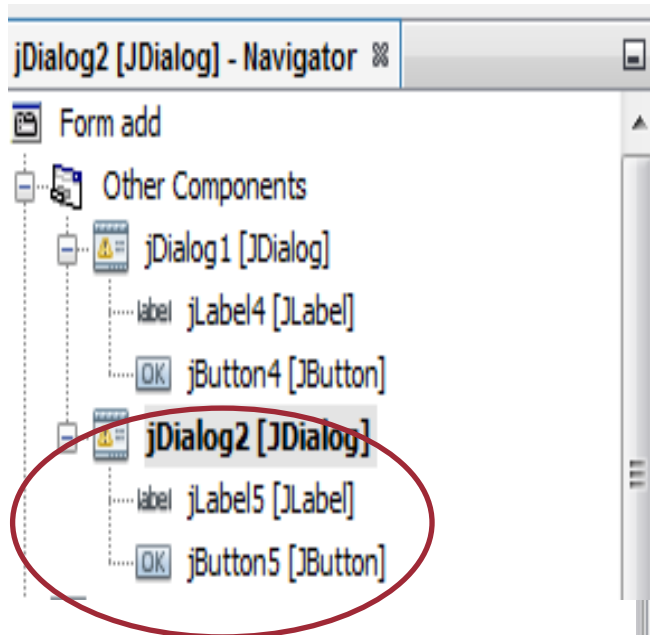
The image shows a Windows-style dialog box titled "Division". It contains three input fields: "NB1" with the value "2020", "NB2" with the value "0", and "Résultat" with the value "Infinity". The "Résultat" field is circled in red. Below the input fields are three buttons: "Division", "Annuler", and "Quitter".

**On veut signaler
cette exception**

On veut afficher une boîte de dialogue pour avertir l'utilisateur



Design & code



```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
    jDialog2.setVisible(false);  
}
```

throw = lancer ou signaler

```
try{
float n1,n2,res;
n1= Float.parseFloat(jTextField1.getText());
n2= Float.parseFloat(jTextField2.getText());
if (n2==0) { throw new ArithmeticException ();}
res=n1/n2;
jTextField3.setText(String.valueOf(res));
}
catch (ArithmeticException e1) {jDialog2.setVisible(true);}
catch (NumberFormatException e2) {jDialog1.setVisible(true);}
```