Université Grenoble-Alpes
Licence 2 MIASHS
Inff3 – Algorithmique et programmation par objets
Jérôme David - CC BY-SA 4.0
2024-2025

TP2 : Création d'objets et interfaces graphiques

L'objectif de ce TP est de manipuler au travers de la création d'interfaces graphique, l'instanciation d'objets et l'appels de méthodes. Ce TP ne traite pas de la gestion des événements (i.e. les actions associées aux clics sur des boutons par exemple).

Le langage Java dispose de librairies pour créer des interfaces graphiques. Toutes le classes dont on a besoin pour cela sont disponibles dans les packages `java.awt` et `javax.swing`. En règle générale, lors de la création d'interface graphique, on importera l'ensemble des classes de ces packages en ajoutant les 2 lignes suivantes au début des fichiers .java :

```
import java.awt.*;
import javax.swing.*;
```

1- Première fenêtre

Le code minimum pour créer une fenêtre vide est donné par la classe exécutable `InterfaceGraphique1`. Lisez ce code et exécutez-le.

Un objet de type `JFrame` permet de représenter une fenêtre. Une fenêtre est composée de plusieurs éléments dont le panneau de contenu accessible par appel à la méthode `getContentPane()` sur un objet de type `JFrame`.

Dans ce panneau de contenu, on peut ajouter des composants graphiques via appel à la méthode `add(…)` qui prend en paramètre le composant que l'on veut ajouter.

Etudiez et exécutez, la classe exécutable `InterfaceGraphique2` qui instancie un objet de type `JFrame` et ajoute dans son panneau de contenu un objet de type `JLabel`. Cet objet `JLabel` permet de représenter une étiquette textuelle.

Il existe une multitude de composants graphiques. Modifiez la classe `InterfaceGraphique2` pour tester successivement les composants suivants :

```
JTextField tf = new JTextField();
JButton bt = new JButton();
JCheckBox jtb = new JCheckBox();
JSlider js = new JSlider();
JComboBox jcombo = new JComboBox();
```

Pour l'instant, nous pouvons ajouter qu'un seul composant à fois. Tout nouvel appel à la méthode `add(…)` remplace le composant précédemment présent par le composant passé en paramètre. Essayez pour vous en persuader.

2- Gestionnaire de placement

Si on veut ajouter plusieurs composants dans une fenêtre alors, il faut spécifier leur placement. Pour cela, il existe des gestionnaires de placements. Le gestionnaire de placement par défaut est appelé

`BorderLayout`. Il permet de placer les composants sur 5 places différentes : début de page (ou nord), fin de page (ou sud), début de ligne (ou ouest), fin de ligne (ou est).
Les classes `InterfaceGraphique3` et `InterfaceGraphique4` sont des exemples de ce gestionnaire de placement.
Exécutez le programme `InterfaceGraphique4`, redimensionnez la fenêtre et observez la répartition de l'espace supplémentaire. Que pouvez-vous en conclure sur les règles de répartition de l'espace de ce gestionnaire de placement ?

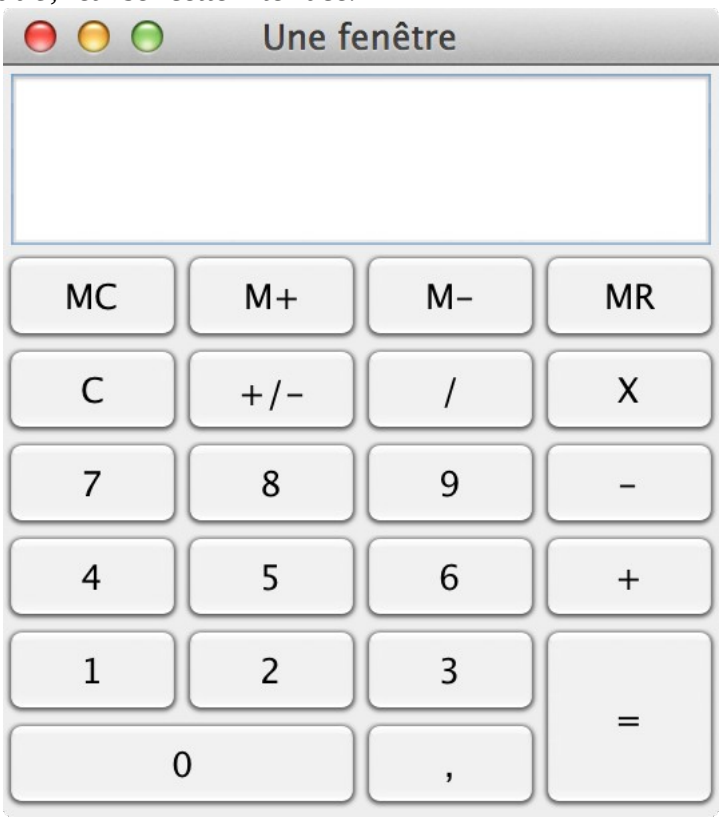
Il existe plusieurs gestionnaires de placement : `FlowLayout`, `GridLayout`, etc. La classe `InterfaceGraphique5` illustre l'utilisation du gestionnaire `GridLayout` qui positionne les composants dans une grille. Testez en changeant le gestionnaire de placement par `FlowLayout`. Qu'observez-vous ?

3- Imbrication de panneaux et gestionnaire de placements

Pour réaliser des placements complexes, il existe des gestionnaires très perfectionnés mais également difficile à appréhender (`SpringLayout`, `GridBagLayout`). Une façon de créer une interface « complexe » est de découper son interface en sous-groupes de composants que l'on appellera panneaux puis de placer ces panneaux dans le panneau de contenu de la fenêtrer. On peut ainsi imbriquer plusieurs niveaux de panneaux.

Pour faire cela, on peut créer des objets de type `JPanel` (des panneaux). Dans ces panneaux on peut ajouter des composants (ou des panneaux) via la méthode `add`. Chaque panneau possède son propre gestionnaire de placement. Ces panneaux peuvent être eux-mêmes placés dans le panneau de contenu de la fenêtre.

La classe illustre `InterfaceGraphique6` l'utilisation de `JPanel`.

4- Exercice

En découpant cette interface en plusieurs niveaux de panneaux et en utilisant le gestionnaire de placement `GridLayout`, réaliser cette interface.