

Applications graphiques Swing sous Java

Exercice 0 : Écrire un programme Java qui affiche "bonjour" dans la console via `System.out.println(...)`. Sauvegarder le programme dans un fichier de même nom que la classe ⊕ extension `.java`. L'exécution comporte 2 étapes : on doit taper 2 commandes dans la console : (click sur K en haut à gauche → Applications pédagogiques

→ Terminal → Konsole)

```
compilation # javac NOMCLASSE.java
exécution   # java NOMCLASSE
```

Soit le programme Java ci-après. On va utiliser ce code comme squelette pour les programmes suivants.

```
1 import javax.swing.*;
2 public class Cadre{
3     public static void main(String [] args){
4         JPanel monPanel = new JPanel();
5         JButton monBouton = new JButton("OK");
6         JLabel monLabel = new JLabel("Un_texte");
7         monPanel.add(monBouton);
8         monPanel.add(monLabel);
9
10        JFrame monCadre = new JFrame();
11        monCadre.setSize(600, 200);
12        monCadre.setVisible(true);
13        monCadre.setContentPane(monPanel);
14        monCadre.setLocation(0,0);
15        monCadre.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
16    }
17 }
```



Familiarisez vous avec les principes Swing et Java pour comprendre le code ci-dessus et les similarités entre Java et Processing.

1 Mouvements de Fenêtres

Exercice 1.1 Modifier le programme plus haut pour faire une fenêtre similaire à celle ci :



Exercice 1.2 Ajouter encore 50 boutons à la fenêtre. Vous avez besoin d'une boucle genre : `for (int i=1;i<=50;i++)`. Le bloc de code à l'intérieur du `for` doit contenir une instruction pour ajouter le bouton.

Exercice 1.3 Modifier la position de la fenêtre pour la placer en bas à droite. Les deux arguments de la méthode `setLocation` indiquent les coordonnées x et y par rapport au coin en haut à gauche.

Indication : modifiez juste la ligne 14 du programme ci-dessus.

Exercice 1.4 Continuer le programme (en Java, vous continuez la méthode `main`) afin de déplacer progressivement la fenêtre (un pas de 10 pixel) sur la trajectoire suivante : coin haut gauche → coin bas droite. Après chaque déplacement, le programme fait une pause de 0.2 secondes. Pour cela, vous pouvez utiliser la méthode `sleep`. Cette méthode n'est pas implémentée sous Java, mais on peut l'écrire dans la classe `Cadre` :

```
public static void sleep (double seconds){
    try{Thread.currentThread().sleep((int)(1000*seconds));}
    catch(java.lang.Exception ie){};
}
```

2 Dessins et animations

Pour tracer des figures, nous avons besoin d'une nouvelle classe qui gère les dessins. Pour cela, il faut construire *un nouveau fichier .java* (attention à lui donner le même nom que le nom de la classe \oplus l'extension) :

```
1 import javax.swing.*;
2 public class Toile extends java.awt.Canvas{
3     public void paint(java.awt.Graphics g){
4         g.fillRect(50, 50, 300, 310);
5     }
6 }
```

Par la suite, il faut ajouter cette "toile" au panel dans l'autre fichier. On peut ajouter dans `Cadre.java` :

```
Toile maToile = new Toile();
maToile.setPreferredSize(new java.awt.Dimension(500,300));
monPanel.add(maToile);
```

Cette "toile" représente l'objet qui réalise le dessin, qui sera utilisé pour tracer des objets géométriques. Pour cela, la fonction *paint* (appelée automatiquement par le système) utilise un objet *g* de classe *Graphics*.

Compiler les deux fichiers Java et lancer java Cadre

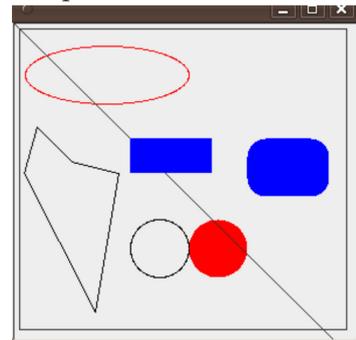
Exercice 2.1 Continuer la la méthode `paint` de la classe `Toile` afin de faire un dessin.

1. D'abord, pour écrire un texte, vous pouvez utiliser les instructions suivantes.

```
g.setFont(new java.awt.Font("Arial",java.awt.Font.ITALIC,20));
g.setXORMode(java.awt.Color.RED);
g.drawString("Salut",20,20);
```

2. Quel est l'effet de l'appel `{g.setFont(new Font("Arial",Font.ITALIC,20))}`, trouvez vous facilement la documentation de la méthode `setFont` sur Internet? La même question pour `g.setXORMode(java.awt.Color.RED)`. Tester le fonctionnement si vous écrivez ces instructions *avant ou après monDessin.drawString(...)*; que observez vous? Pourquoi le dessin n'a pas la couleur rouge?
3. Chercher sur Internet la documentation de la classe `Graphics` (e.g. taper "java class Graphics") et étudier ses méthodes. Ensuite, trouver les bonnes méthodes à utiliser pour dessiner :

- une ligne
- un rectangle
- un rectangle rempli
- un rectangle avec les coins arrondis
- un cercle (comme sous `processing`, on doit faire une ellipse avec hauteur=largeur)
- un cercle rempli
- une ellipse
- un polygone avec 5 sommets



Exercice 2.2 Pour réaliser une animation, nous avons besoin de faire en boucle les opérations suivantes : (i) afficher une figure géométrique à une position genre $(10 * i, 10 * i)$, (ii) faire une pause à l'aide d'un appel à la méthode statique `Cadre.sleep(...)`, (iii) effacer le dessin actuel et répéter le point (i) avec une valeur incrémentée de *i*. Pour effacer un dessin, il suffit de le re-faire le même dessin à la même position : comme on trace toutes les figures en mode XOR, "re-faire" = "effacer".