Université de Carthage
Institut National des Sciences
Appliquées et de Technologie

Année Universitaire : 2022-2023
Niveau : RT2
Enseignante : Hazar MLIKI

Atelier 2 : Initiation à la programmation graphique

Objectifs

- Introduire Swing
- Manipuler les objets JFrame & JPanel
- Dessiner avec Graphics

Partie 1 : L'objet JFrame

1. Créez un nouveau projet : File > New > JAVA PROJECT
2. Créez une fenêtre de type Jframe :

```
import javax.swing.JFrame;
 public class Test {
  public static void main(String[] args){
    JFrame fenetre = new JFrame();
  }
}
```

Lorsqu' on exécute ce code, on n'obtient rien, car par défaut, *JFrame* est invisible.

Pour le rendre visible, on doit ajouter :

```
fenetre.setVisible(true);
```

3. Pour obtenir une fenêtre plus développée, il faut :
   - qu'elle soit plus grande ;
   - qu'elle comporte un titre ;
   - qu'elle figure au centre de l'écran ;
   - que le programme s'arrête lorsqu'on clique sur la croix rouge, car sinon le processus tourne toujours même après la fermeture de la fenêtre.

Pour répondre à ces besoins, on peut modifier le code par :

```
import javax.swing.JFrame;
 public class Test {
  public static void main(String[] args){
    JFrame fenetre = new JFrame();
    //Définit un titre pour la fenêtre
    fenetre.setTitle("Ma première fenêtre Java");
```

```
    //Définit sa taille : 400 px de large et 100 px de haut
     fenetre.setSize(400, 100);
   // positionner la fenêtre au centre
     fenetre.setLocationRelativeTo(null);
  //Termine le processus lorsqu'on clique sur la croix rouge
     fenetre.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  // rendre la fenêtre visible
     fenetre.setVisible(true);
   } }
```

4. Pour ne pas redéfinir les attributs à chaque fois, créez une classe *Fenetre* qui hérite de *JFrame* :

```
import javax.swing.JFrame;
public class Fenetre extends JFrame {
 public Fenetre(){
  this.setTitle("Ma première fenêtre Java");
  this.setSize(400, 500);
  this.setLocationRelativeTo(null);
  this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  this.setVisible(true);
  }   }
```

Pour tester cette classe, il suffit de l'instancier dans le *main* d'une autre classe *Test* :

```
Fenetre fen = new Fenetre();
```

5. Pour personnaliser la fenêtre, on peut :
   - Positionner la fenêtre à un emplacement spécifique à l'écran (Les coordonnées sont exprimées en pixels). L'origine des coordonnées coïncide avec le coin supérieur gauche de l'écran. L'axe des abscisses est orienté vers la droite, celui des ordonnées vers le bas.

     ```
     setLocation(int x, int y)
     ```

   - Empêcher le redimensionnement de la fenêtre (**True** : autorise le redimensionnement ; False : l'empêche)

     ```
     setResizable(boolean b)
     ```

   - Garder la fenêtre au premier plan (**True** : laisse la fenêtre au premier plan ; **False** : l'annule)

     ```
     setAlwaysOnTop(boolean b)
     ```

   - Retirer les contours et les boutons de contrôle (**True** : retire les contours et les boutons de contrôle ; **False** : l'annule)

     ```
     setUndecorated(boolean b)
     ```

<u>À Vous !</u>
Modifier la taille et le titre de la fenêtre graphique à partir des informations entrées dans la fenêtre console.

Partie 2 : L'objet JPanel

1. JPanel est un composant de type conteneur qui permet d'accueillir d'autres objets de même type ou des objets de type composant (boutons, cases à cocher…).
   La démarche à suivre :
   a. Importer la classe **javax.swing.JPanel** dans le classe héritée de JFrame.
   b. Instancier un **JPanel** puis lui spécifier une couleur de fond pour mieux le distinguer.
   c. Avertir le JFrame que ça sera le JPanel qui constituera son **ContentPane**.

```
import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JPanel;
public class Fenetre extends JFrame {
  public Fenetre(){
    this.setTitle("Ma première fenêtre Java");
    this.setSize(400, 100);
    this.setLocationRelativeTo(null);
    //Instanciation d'un objet JPanel
    JPanel pan = new JPanel();
    //Définition de sa couleur de fond
    pan.setBackground(Color.ORANGE);
    //On prévient le JFrame que le JPanel sera son ContentPane
    this.setContentPane(pan);
    this.setVisible(true);
  } }
```

2. Créer une classe Panneau héritée de Jpanel

```
import java.awt.Graphics;
import javax.swing.JPanel;
public class Panneau extends JPanel {
  public void paintComponent(Graphics g){
//afficher ce message à chaque fois que la méthode est appelée
    System.out.println("Je suis exécutée !");
    g.fillOval(20, 20, 75, 75);
  }  }
```

Partie 3 : Dessiner dans un JPanel avec Graphics

Un dessin en Java doit passer par un objet Graphics qui offre des méthodes pour dessiner des motifs, des images et du texte.

La méthode *paintComponent* reçoit un paramètre, de type *Graphics*. Un objet Graphics comporte une suite de paramètres (police, couleur, etc.) pour dessiner d'images et de texte.

Pour créer un panneau, sur lequel vous pourrez dessiner :

```
class MyComponent extends JComponent
{
   public void paintComponent(Graphics g)
  {
    code de dessin
  }
}
```

Mettre Panneau comme content pane de Fenetre

```
import javax.swing.JFrame;
public class Fenetre extends JFrame {
  public Fenetre(){
    this.setTitle("Ma première fenêtre Java");
    this.setSize(100, 150);
    this.setLocationRelativeTo(null);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setContentPane(new Panneau());
    this.setVisible(true);
  }
  }
```

Les méthodes suivantes permettent de dessiner différentes formes géométriques.

- ***drawOval()*** permet de dessiner un rond vide (comme fillOval())

```
import java.awt.Graphics;
import javax.swing.JPanel;
public class Panneau extends JPanel {
  public void paintComponent(Graphics g){
    int x1 = this.getWidth()/4;
    int y1 = this.getHeight()/4;
    g.drawOval(x1, y1,this.getWidth()/2, this.getHeight()/2);
  }     }
```

- ***drawOval()*** : permet de dessiner un rond vide (comme fillOval())

```
import java.awt.Graphics;
```

```
import javax.swing.JPanel;
public class Panneau extends JPanel {
  public void paintComponent(Graphics g){
    int x1 = this.getWidth()/4;
    int y1 = this.getHeight()/4;
    g.drawOval(x1, y1,this.getWidth()/2, this.getHeight()/2);
  }     }
```

- **drawRect()** : permet de dessiner un rectangle vide (comme fillRect())

```
import java.awt.Graphics;
import javax.swing.JPanel;
 public class Panneau extends JPanel {
  public void paintComponent(Graphics g){
    g.drawRect(10, 10, 50, 60); //x1, y1, width, height
    g.fillRect(65, 65, 30, 40); }  }
```

- ***drawRoundRect():*** dessiner un rectangle vide arrondi (fillRoundRect())

```
import java.awt.Graphics;
import javax.swing.JPanel;
 public class Panneau extends JPanel {
  public void paintComponent(Graphics g){
    //x1, y1, width, height, arcWidth, arcHeight
    g.drawRoundRect(10, 10, 30, 50, 10, 10);
    g.fillRoundRect(55, 65, 55, 30, 5, 5);
  }  }
```

- ***drawLine()*** **:** dessiner une ligne droite (x1, y1, x2, y2)

```
import java.awt.Graphics;
import javax.swing.JPanel;
 public class Panneau extends JPanel {
  public void paintComponent(Graphics g){
    g.drawLine(0, 0, this.getWidth(), this.getHeight());
    g.drawLine(0, this.getHeight(), this.getWidth(), 0); } }
```

- ***drawPolygon() :*** dessiner un polygone vide (comme fillPolygon()) en définissant les coordonnées de tous les points qui le forme :

```
drawPolygon(int[] x, int[] y, int nbrePoints);
```

```
import java.awt.Graphics;
import javax.swing.JPanel;
 public class Panneau extends JPanel {
  public void paintComponent(Graphics g){
    int x[] = {20, 30, 50, 60, 60, 50, 30, 20};
    int y[] = {30, 20, 20, 30, 50, 60, 60, 50};
```

```
    g.drawPolygon(x, y, 8);
    int x2[] = {50, 60, 80, 90, 90, 80, 60, 50};
    int y2[] = {60, 50, 50, 60, 80, 90, 90, 80};
    g.fillPolygon(x2, y2, 8);
  }
}
```

- ***drawString() :*** écrire un texte, modifier la couleur et la police d'écriture

```
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import javax.swing.JPanel;
public class Panneau extends JPanel {
 public void paintComponent(Graphics g){
  Font font = new Font("Courier", Font.BOLD, 20);
  g.setFont(font);
  g.setColor(Color.red);
  g.drawString("Il faut laisser du temps au temps!", 10, 20);
  }
}
```

- ***drawImage()*** **:** afficher une image

drawImage(Image img, int x, int y, int width, int height, Observer obs)

```
import java.awt.Graphics;
import java.awt.Image;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;
import javax.swing.JPanel;
public class Panneau extends JPanel {
 public void paintComponent(Graphics g){
  try { Image img = ImageIO.read(new File("images.jpg"));
        g.drawImage(img, 0, 0, this);
    //Pour une image de fond
   //g.drawImage(img, 0, 0, this.getWidth(), this.getHeight(), this);
      } catch (IOException e) {
         e.printStackTrace();  }
}
}
```

<u>À Vous :</u> Dessinez les formes géométriques suivantes