

## Thread Java TD/TP N°1

### *Simulation du mouvement d'un ou plusieurs mobiles*

#### **Exercice I**

La classe JPanel permet de créer des widgets qui sont des conteneurs mais qui sont aussi des fenêtres sans bord sur lesquelles on peut écrire et dessiner. Ils sont ajoutés par la méthode add() à une fenêtre dérivant de JFrame, comme les autres widgets. On donne le code qui suit:

```
class UnMobile extends JPanel implements Runnable
{
    int saLargeur, saHauteur, sonDebutDessin;
    final int sonPas= 10, sonTemps= 500. sonCote= 40;
    UnMobile( int telleLargeur, int telleHauteur)
    {
        super();
        saLargeur= telleLargeur;saHauteur= telleHauteur;
        setSize( telleLargeur, telleHauteur );
    }//UnMobile

    public void run()
    {
        for(sonDebutDessin= 0;sonDebutDessin < saLargeur – sonPas; sonDebutDessin+= sonPas)
        {
            repaint();
            try{ (Thread.sleep( son Temps ))//try
            catch(InterruptedException telleExcp)
            {telleExcp.printStackTrace();};//catch
        }//for
    }//run()

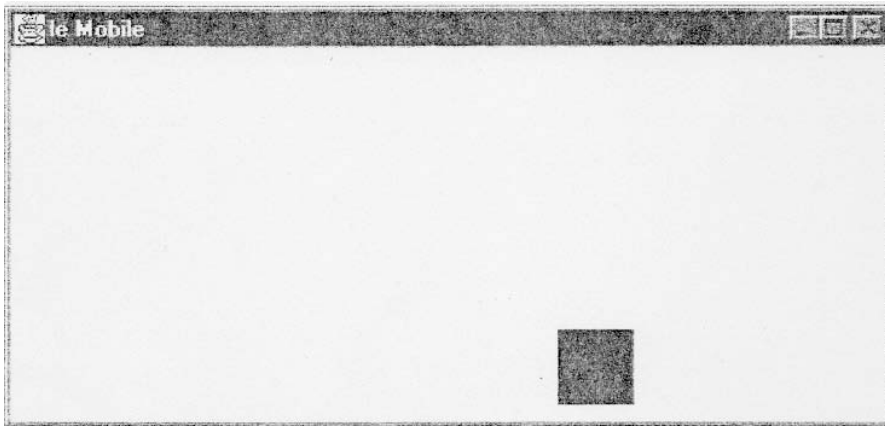
    public void paintComponent( Graphics telContexteGraphique)
    {
        super.paintComponent( telContexteGraphique );
        telContexteGraphique.fillRect( sonDebutDessni, saHauteur/2, sonCote, SonCote);
    }//paintComponent()

} //classe UnMobile
```

Cette classe permet de créer des threads ayant leur propre espaces d'affichage puisqu'à chaque instantiation de la classe UnMobile on crée une fenêtre sans bord dont on fixe la taille par setSize() et sur laquelle on dessine par paintComponent() (toujours appelée par repaint()). Il faut noter que cette fenêtre est donnée par telContexteGraphique dans paintComponent().

1- Ecrire une classe UneFenetre dérivant de JFrame, cette classe aura un champ UnMobile sonMobile, son constructeur devra ajouter sonMobile à la fenêtre, créer une thread avec sonMobile, afficher la fenêtre et lancer la thread. Ecrire une classe Main testant la classe UneFenetre.

2- Faire en sorte que le mobile reparte en sens inverse lorsqu'il atteint une extrémité de la fenêtre



## **Exercice II**

Ajouter un bouton à la dernière fenêtre de l'exercice I. Ce bouton devra arrêter ou relancer le mobile, on pourra pour cela utiliser les méthodes `suspend()` et `resume()` de la classe `Thread`. Vous aurez un warning vous indiquant qu'elles sont deprecated pour le jdk1.3, ne pas en tenir compte.

## **Exercice III**

L'organisateur de la classe `JFrame` est un objet de la classe `BorderLayout`, il utilise les points cardinaux pour disposer les widgets. Il existe d'autres organisateurs, par exemple ceux issus de la classe `GridLayout` qui disposent les widgets selon une grille dont on a donné le nombre de lignes et de colonnes au constructeur. Pour illustrer cela on donne le code qui suit:

```
Container leConteneur = getContentPane();  
leConteneur.setLayout (new GridLayout(NBRLIG, NBRCOL);  
sonMobile1=new unMobile(LARGE_MOBILE, HAUT_MOBILE);  
sonBouton1= new JButton ("Start/Stop");  
sonBouton1.addActionListener(this);  
leConteneur.add(sonBouton1);  
leConteneur.add(sonMobile1);  
sonMobile2=new unMobile(LARGE_MOBILE, HAUT_MOBILE);  
sonBouton2= new JButton ("Start/Stop");  
sonBouton2.addActionListener(this);  
leConteneur.add(sonBouton2);  
leConteneur.add(sonMobile2);
```

Modifier le code de l'exercice II de façon à afficher 2 mobiles, chaque bouton et chaque zone d'affichage sont disposés sur une grille de 2 lignes et 2 colonnes.