

Université Paris Est Créteil
IUT de Sénart-Fontainebleau
Département informatique
IS3
Algorithmique Programmation
2010–2011

Partiel
Durée : 2 heures

Seuls sont autorisés, à titre de documents, les listings comprenant explicitement le nom de l'étudiant (et celui-ci seulement) et les notes manuscrites.

Exercice 1.- (Balle se déplaçant)

Écrire une application Java qui affiche un cadre (fermant) de titre « Cadre avec balle qui se déplace », à fond bleu dans lequel apparaît une balle jaune de rayon dix pixels que l'on peut déplacer avec la souris.

Exercice 2.- (Distinguer les boutons de souris)

Écrire une applet Java qui affiche un segment dans sa fenêtre active. Son origine sera définie par un clic sur le bouton gauche de la souris (elle se modifie à chaque nouveau clic sur ce bouton). Son extrémité sera définie par un clic sur le bouton droit de la souris.

Ceci constitue le début d'un logiciel de dessin.

Exercice 3.- (Affichage d'une liste chaînée)

- 1^o) Écrire une classe Java liste chaînée d'entiers, comprenant une méthode d'affichage de cette liste chaînée dans l'ordre dans lequel on l'a entrée.

[On pourra utiliser une méthode récursive.]

- 2^o) Écrire un programme de test demandant d'entrer un certain nombre d'entiers (-1 pour terminer) puis affichant ces entiers dans l'ordre dans lequel ils ont été entrés.

Documentation

1 Les couleurs

- Les couleurs sont les objets de la classe `Color` du paquetage `java.awt`.
- Il y a un certain nombre de couleurs prédéfinies sous la forme de données statiques :

```
black blue cyan darkGray gray green lightGray  
magenta orange pink red white yellow
```

- On peut modifier la couleur de fond d'un composant grâce à la méthode :
`setBackground(Color)`

2 Dessin d'une surface ovale

- La méthode :

```
setColor(Color)
```

de la classe `Graphics` permet de changer la couleur de que l'on est en train de dessiner (jusqu'au prochain changement de couleur).

- La méthode :

```
fillOval(int x, int y, int width, int height)
```

de la classe `Graphics` dessine une surface ovale dans la couleur en cours contenue dans le rectangle carré défini par les coordonnées (x, y) de son coin supérieur gauche, de largeur `width` et de hauteur `height`, exprimés en pixels.

3 Déplacement de la souris

- L'interface `MouseMotionListener` permet de prendre en compte le déplacement de la souris.

- Les deux méthodes, que l'on doit surcharger absolument, sont :

```
public void mouseMoved(MouseEvent)  
public void mouseDragged(MouseEvent)
```

Il est fait appel à la première lorsque le curseur de la souris passe dans la fenêtre sans appuyer sur un bouton alors que pour la seconde on a appuyé sur un bouton et on déplace le curseur (un événement est engendré sans cesse jusqu'à ce qu'on relâche le bouton).

- Les méthodes :

```
getX()  
getY()
```

de `MouseEvent` permettent de savoir où se situe le curseur.

4 Événements souris (rappels)

- L'interface `MouseListener` permet de prendre en compte les événements souris.
- Les cinq méthodes, que l'on doit surcharger absolument, sont :

```
public void mousePressed(MouseEvent)
public void mouseReleased(MouseEvent)
public void mouseClicked(MouseEvent)
public void mouseEntered(MouseEvent)
public void mouseExited(MouseEvent)
```

5 Événements d'entrée

- La classe `InputEvent` permet de préciser les événements (clavier et souris).
- La méthode de `Event` (et ses classes dérivées) :

```
public int getModifiers()
```

permet d'obtenir un entier (dit *modificateurs*) dont chaque bit est un drapeau.

- Entre autres, ses constantes statiques :

```
BUTTON1_MASK
BUTTON3_MASK
```

sont des masques permettant de déterminer si c'est sur le bouton gauche (respectivement le bouton droit) de la souris que l'on a cliqué, grâce au modificateur obtenu à partir de l'événement.

- Rappelons que la conjonction bit à bit est `&` en Java.