

SN/UCAD/ESP/DGI/DICI2/JAVA AVANCE

TME N°2

IMPORTANT

A la fin de la séance, et **avant 0h00**, chaque étudiant enverra les programmes réalisés via Google classroom.

Exercice: (Dessiner grâce à Java)

L'objet de l'exercice est de réaliser un simple programme de dessin. L'idée est de pouvoir dessiner sur une fenêtre en déplaçant la souris avec le bouton gauche enfoncé, cela est possible si l'on parvient à chaque instant à récupérer les coordonnées (x et y) du curseur de la souris et à dessiner une petite boule au point (x,y). Nous nous servons à cet effet de l'interface *MouseMotionListener*. Les méthodes abstraites de cette interface java sont: *public void mouseDragged(MouseEvent e)* et *public void mouseMoved(MouseEvent e)*.

Ecrire une classe Peintre qui étend la classe *javax.swing.JFrame*, implémente l'interface *MouseMotionListener* et qui comporte :

- Deux attributs entières valeurX et valeurY pour stocker les coordonnées de la souris sur l'écran
- D'un constructeur *Peintre()* qui fait appel au constructeur de la super-classe, ajoute un label d'indication au bas de la fenêtre en cours de création comme indiqué sur la capture d'écran, elle fait ensuite appel à *setSize(int, int)* et à *show()* de *JFrame*
- D'une méthode *public void paint(Graphics g)* qui trace une boule ovale pleine avec la méthode *fillOval(int,int,int,int)* de *g*.
- D'une méthode *public static void main(String[] args)* créant juste une instance de Peintre.
- Et naturellement les deux méthodes redéfinies de l'interface

Dans la redéfinition de la méthode *mouseDragged()*, on fait appel aux méthodes d'instance *getX()* et *getY()* de *MouseEvent* pour récupérer les coordonnées courantes de la souris; ensuite on fait appel à la méthode *paint()* de la classe Peintre grâce à un simple appel de fonction *repaint()* de *JFrame* héritée de la classe *Component*.



Capture de l'exercice