

Alain Plantec

Vous utiliserez Eclipse pour développer. Il est aussi demandé d'utiliser *Junit* pour tester vos méthodes. Voir par exemple <http://gfx.developpez.com/tutoriel/java/junit/>

Préambule

Voici un exemple de code pour dessiner un polygone dans une fenêtre.

```
import java.awt.Graphics;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.JComponent;
import javax.swing.JFrame;

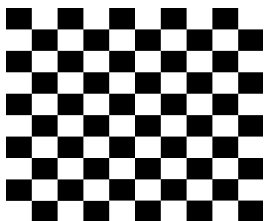
class MyCanvas extends JComponent {
    private static final long serialVersionUID = 1L;
    public void paint(Graphics g) { // Fonction appelee pour dessiner
        int[] x = {10,120,120,10} ; // Tableau pour les x
        int[] y = {40,40,80,80} ; // Tableau pour les y
        int[] x2 = {30,150,150,30} ; // Tableau pour les x
        int[] y2 = {90,90,170,170} ; // Tableau pour les y
        Color previousColor = g.getColor();
        g.setColor(Color.BLACK);
        g.drawPolygon (x, y, x.length); // Dessin du polygone
        g.fillPolygon (x2, y2, x.length); // Dessin du polygone
        g.setColor(previousColor);
    }
}

class GestFenetre extends WindowAdapter {
    public GestFenetre(JFrame window) {
        window.addWindowListener(this);
    }
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
}

public class DrawPolygon {
    public static void main(String[] args) { // Notre fonction main
        JFrame window = new JFrame(); // On cree une fenetre
        window.setBounds(30, 30, 300, 300); // On indique la taille de la fenetre
        window.getContentPane().add(new MyCanvas()); // On ajoute un composant pour dessiner
        window.setVisible(true); // Et hop, on demande à la fenetre de s'ouvrir
        new GestFenetre(window);
    }
}
```

Exercice 1

Inspirez vous de ce code pour programmer une classe *Damier* et en dessiner les instances de la façon suivante :



Exercice 2

Loli est un petit ver qui me donne le tournis : je le vois toujours se déplacer et tourner dans tous les sens sur son territoire. Dès qu'il parvient à une de ses limites, il tourne dans un sens ou un autre et continue à se déplacer.

Programmez *Loli* dans son territoire. Vous représentez le territoire en dessinant un Polygone et *Loli* comme une série d'anneaux. Chaque anneau peut être représenté par un petit carré noir.

Voici comment générer un nombre aléatoire. Vous pouvez vous en servir pour simuler les changements de direction quand *Loli* parvient à un bord.

```
Random ran = new Random();
System.out.println( ran.nextInt(3) ); // 0, 1, 2 ou 3
```