```java
public class Rectangle
{
    // private - comme il faut, mais il faut modifier les autres classe pour utiliser les getters/setters
    private int x;
    private int y;
    private int width;
    private int height;

    public Rectangle(int x, int y, int width, int height)
    {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
    }

    public int getX()
    {
        return x;
    }

    public int getY()
    {
        return y;
    }

    public int getWidth()
    {
        return width;
    }

    public int getHeight()
    {
        return height;
    }

    public void setLocation(int x, int y)
    {
        this.x = x;
        this.y = y;
    }

    public boolean contains(Rectangle o)
    {
        // les 4 lignes virtuelles de "o" doivent être l'intérieur du rectangle
        return (o.x >= x) && (o.x + o.width  <= x + width) &&
               (o.y >= y) && (o.y + o.height <= y + height);
    }

    public boolean intersects(Rectangle o)
    {
        // les 4 lignes virtuelles doivent de "o" être toutes OK
        return (o.x <= x + width)  && (o.x + o.width  >= x) &&
               (o.y <= y + height) && (o.y + o.height >= y);
    }
}
```

```java
import java.awt.Color;
import java.awt.Graphics;
public class Crosshair extends Rectangle
{
    public Crosshair(int x, int y)
    {
        super(x, y, 1, 1);
    }

    public void draw(Graphics g)
    {
        int x = getX();
        int y = getY();

        g.setColor(Color.BLACK);
        g.drawLine(x, y - 40, x, y + 40);
        g.drawLine(x - 40, y, x + 40, y);
        g.drawOval(x - 10, y - 10, 21, 21);
        g.drawOval(x - 25, y - 25, 51, 51);
    }
}
```

```java
import java.awt.Color;
import java.awt.Graphics;
public class Turtle extends Rectangle
{
    private Color color;

    public Turtle(int x, int y, Color color)
    {
        super(x, y, 21, 25);
        this.color = color;
    }

    public void draw(Graphics g)
    {
        int x = getX();
        int y = getY();

        g.setColor(color);
        g.fillOval(x + 2, y + 6, 16, 16);
        g.fillOval(x + 7, y + 0, 6, 8);
        g.drawLine(x + 0, y + 7, x + 20, y + 24);
        g.drawLine(x + 20, y + 7, x + 0, y + 24);
    }

    public void goRight(int dist)
    {
        setLocation(getX() + dist, getY());
    }

    public void goLeft(int dist)
    {
        setLocation(getX() - dist, getY());
    }

    public void goUp(int dist)
    {
        setLocation(getX(), getY() - dist);
    }

    public void goDown(int dist)
    {
        setLocation(getX(), getY() + dist);
    }
}
```

```java
import java.awt.Color;
import java.awt.Graphics;
public class TurtleHunting
{
    private Turtle turtle = null;
    private Crosshair crosshair = null;
    private int shots;
    private int hits;
    private int width;
    private int height;

    public TurtleHunting(int width, int height)
    {
        this.width = width;
        this.height = height;
        crosshair = new Crosshair(width / 2, height / 2);
        turtle = new Turtle(0, 0, Color.BLACK);

        positionTurtleRandomly();
        shots = 0;
        hits = 0;
    }

    public void fire()
    {
        shots++;
        if (turtle.intersects(crosshair))
        {
            hits++;
            positionTurtleRandomly();
        }
    }

    private void positionTurtleRandomly()
    {
        turtle.setLocation((int) (Math.random() * (width - turtle.getWidth())),
                (int) (Math.random() * (height - turtle.getHeight())));
    }

    public void moveCrosshairUp()
    {
        turtle.goDown(10);
    }

    public void moveCrosshairDown()
    {
        turtle.goUp(10);
    }

    public void moveCrosshairRight()
    {
        turtle.goLeft(10);
    }

    public void moveCrosshairLeft()
    {
        turtle.goRight(10);
    }

    public void draw(Graphics g)
    {
        g.setColor(Color.BLACK);
        g.fillRect(0, 0, width, height);
        g.setColor(Color.WHITE);
        int side = Math.min(width, height);
        g.fillOval((width - side) / 2, (height - side) / 2, side, side);

        g.setColor(Color.YELLOW);
        g.drawString("Shots: " + shots, 5, 20);

        Turtle turtleIcon = new Turtle(0, 0, Color.YELLOW);
        turtleIcon.setLocation(width - 50, 2);
        turtleIcon.draw(g);

        g.drawString(String.valueOf(hits), width - 20, 20);

        turtle.draw(g);
        crosshair.setLocation(width / 2, height / 2);
        crosshair.draw(g);
    }
}
```

```java
import java.awt.Color;
import java.awt.Graphics;
public class DrawPanel extends javax.swing.JPanel
{
    private TurtleHunting turtleHunting = null;

    public DrawPanel()
    {
        initComponents();
    }

    public void setTurtleHunting(TurtleHunting turtleHunting)
    {
        this.turtleHunting = turtleHunting;
    }

    public void paintComponent(Graphics g)
    {
        g.setColor(Color.WHITE);
        g.fillRect(0, 0, getWidth(), getHeight());

        if (turtleHunting != null)
            turtleHunting.draw(g);
    }
// Skipped: ... initComponents { ... }
    // Variables declaration - do not modify//GEN-BEGIN:variables
    // End of variables declaration//GEN-END:variables
}
```

```java
public class MainFrame extends javax.swing.JFrame
{
    private TurtleHunting turtleHunting = null;

    public MainFrame()
    {
        initComponents();

        turtleHunting = new TurtleHunting(drawPanel.getWidth(), drawPanel.getHeight());
        drawPanel.setTurtleHunting(turtleHunting);
        repaint();
    }
// Skipped: ... initComponents { ... }
    private void upButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_upButtonActionPerformed
    {//GEN-HEADEREND:event_upButtonActionPerformed
        turtleHunting.moveCrosshairUp();
        repaint();
    }//GEN-LAST:event_upButtonActionPerformed

    private void downButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_downButtonActionPerformed
    {//GEN-HEADEREND:event_downButtonActionPerformed
        turtleHunting.moveCrosshairDown();
        repaint();
    }//GEN-LAST:event_downButtonActionPerformed

    private void leftButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_leftButtonActionPerformed
    {//GEN-HEADEREND:event_leftButtonActionPerformed
        turtleHunting.moveCrosshairLeft();
        repaint();
    }//GEN-LAST:event_leftButtonActionPerformed

    private void rightButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_rightButtonActionPerformed
    {//GEN-HEADEREND:event_rightButtonActionPerformed
        turtleHunting.moveCrosshairRight();
        repaint();
    }//GEN-LAST:event_rightButtonActionPerformed

    private void fireButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_fireButtonActionPerformed
    {//GEN-HEADEREND:event_fireButtonActionPerformed
        turtleHunting.fire();
        repaint();
    }//GEN-LAST:event_fireButtonActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[])
    {
        java.awt.EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                new MainFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton downButton;
    private DrawPanel drawPanel;
    private javax.swing.JButton fireButton;
    private javax.swing.JButton leftButton;
    private javax.swing.JButton rightButton;
    private javax.swing.JButton upButton;
    // End of variables declaration//GEN-END:variables
}
```