```java
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Point;
public class Turtle
{
    private Point position;
    private String name;

    public Turtle(Point position, String name)
    {
        this.position = position;
        this.name = name;
    }

    public Point getPosition()
    {
        return position;
    }

    public String getName()
    {
        return name;
    }

    public String toString()
    {
        return name;
    }

    public void goRight(int dist)
    {
        position.x += dist;             // pareil à: position.x = position.x + dist;
    }

    public void goLeft(int dist)
    {
        position.x -= dist;
    }

    public void goUp(int dist)
    {
        position.translate(0, -dist);   // pareil à: position.y -= dist;
    }

    public void goDown(int dist)
    {
        position.translate(0, dist);
    }

    public void draw(Graphics g)
    {
        int x = position.x;
        int y = position.y;

        g.setColor(Color.BLACK);
        g.fillOval(x - 2, y - 10, 5, 7);
        g.fillOval(x - 7, y - 5, 15, 15);
        g.drawLine(x - 10, y - 5, x + 10, y + 12);
        g.drawLine(x - 10, y + 12, x + 10, y - 5);

        // Dessiner le nom en dessous en bleu
        g.setColor(Color.BLUE);
        g.drawString(name, x - 10, y + 24);
    }
}
```

```java
import java.awt.Graphics;
import java.util.ArrayList;
public class Turtles
{
    private ArrayList<Turtle> alTurtles = new ArrayList<>();

    public Turtle remove(int index)
    {
        return alTurtles.remove(index);
    }

    public Turtle get(int index)
    {
        return alTurtles.get(index);
    }

    public void add(Turtle turtle)
    {
        alTurtles.add(turtle);
    }

    public Object[] toArray()
    {
        return alTurtles.toArray();
    }

    public void draw(Graphics g)
    {
        for (int i = 0; i < alTurtles.size(); i++)
            alTurtles.get(i).draw(g);
    }

    public int findByName(String name)
    {
        boolean found = false;
        int i = 0;
        while ((!found) && (i < alTurtles.size()))
        {
            if (alTurtles.get(i).getName().equals(name))
                found = true;
            else
                i++;
        }
        if (found)
            return i;
        else
            return -1;
    }
}
```

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

    public void setTurtles(Turtles turtles)
    {
        this.turtles = turtles;
    }

    public DrawPanel()
    {
        initComponents();
    }

    public void paintComponent(Graphics g)
    {
        int w = getWidth();
        int h = getHeight();

        g.setColor(Color.WHITE);
        g.fillRect(0, 0, w, h);

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

```java
import java.awt.Point;
public class MainFrame extends javax.swing.JFrame
{
    private Turtles turtles = null;

    public MainFrame()
    {
        initComponents();

        turtles = new Turtles();
        drawPanel.setTurtles(turtles);

        updateView();
    }

    public void updateView()
    {
        turtleList.setListData(turtles.toArray());
        repaint();
    }
// Skipped: ... initComponents { ... }
    private void rightButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_rightButtonActionPerformed
    {//GEN-HEADEREND:event_rightButtonActionPerformed
        int i = turtleList.getSelectedIndex();
        if (i >= 0)
            turtles.get(i).goRight(10);

        repaint(); // ne PAS utiliser la méthode updateView()  car sinon la selection sera effaçée
    }//GEN-LAST:event_rightButtonActionPerformed

    private void leftButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_leftButtonActionPerformed
    {//GEN-HEADEREND:event_leftButtonActionPerformed
        int i = turtleList.getSelectedIndex();
        if (i >= 0)
            turtles.get(i).goLeft(10);

        repaint(); // ne PAS utiliser la méthode updateView()  car sinon la selection sera effaçée
    }//GEN-LAST:event_leftButtonActionPerformed

    private void upButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_upButtonActionPerformed
    {//GEN-HEADEREND:event_upButtonActionPerformed
        int i = turtleList.getSelectedIndex();
        if (i >= 0)
            turtles.get(i).goUp(10);

        repaint(); // ne PAS utiliser la méthode updateView()  car sinon la selection sera effaçée
    }//GEN-LAST:event_upButtonActionPerformed

    private void downButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_downButtonActionPerformed
    {//GEN-HEADEREND:event_downButtonActionPerformed
        int i = turtleList.getSelectedIndex();
        if (i >= 0)
            turtles.get(i).goDown(10);

        repaint(); // ne PAS utiliser la méthode updateView()  car sinon la selection sera effaçée
    }//GEN-LAST:event_downButtonActionPerformed

        // continue sur la page suivante
```

```java
    private void addButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_addButtonActionPerformed
    {//GEN-HEADEREND:event_addButtonActionPerformed
        int x = (int) (Math.random() * drawPanel.getWidth());
        int y = (int) (Math.random() * drawPanel.getHeight());

        Point position = new Point(x, y);
        Turtle turtle = new Turtle(position, nameTextField.getText());
        turtles.add(turtle);

        updateView();
    }//GEN-LAST:event_addButtonActionPerformed

    private void removeButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_removeButtonActionPerformed
    {//GEN-HEADEREND:event_removeButtonActionPerformed
        int i = turtleList.getSelectedIndex();
        if (i >= 0)
            turtles.remove(i);

        updateView();
    }//GEN-LAST:event_removeButtonActionPerformed

    private void newButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_newButtonActionPerformed
    {//GEN-HEADEREND:event_newButtonActionPerformed
        turtles = new Turtles();
        drawPanel.setTurtles(turtles);

        updateView();
    }//GEN-LAST:event_newButtonActionPerformed

    private void findButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_findButtonActionPerformed
    {//GEN-HEADEREND:event_findButtonActionPerformed
        int i = turtles.findByName(nameTextField.getText());
        if (i >= 0)
            turtleList.setSelectedIndex(i);
    }//GEN-LAST:event_findButtonActionPerformed
// Skipped: ... Look & Feel
     // Variables declaration - do not modify//GEN-BEGIN:variables
     private javax.swing.JButton addButton;
     private javax.swing.JButton downButton;
     private DrawPanel drawPanel;
     private javax.swing.JButton findButton;
     private javax.swing.JLabel jLabel1;
     private javax.swing.JPanel jPanel1;
     private javax.swing.JPanel jPanel2;
     private javax.swing.JScrollPane jScrollPane1;
     private javax.swing.JButton leftButton;
     private javax.swing.JTextField nameTextField;
     private javax.swing.JButton newButton;
     private javax.swing.JButton removeButton;
     private javax.swing.JButton rightButton;
     private javax.swing.JList turtleList;
     private javax.swing.JButton upButton;
     // End of variables declaration//GEN-END:variables
}
```