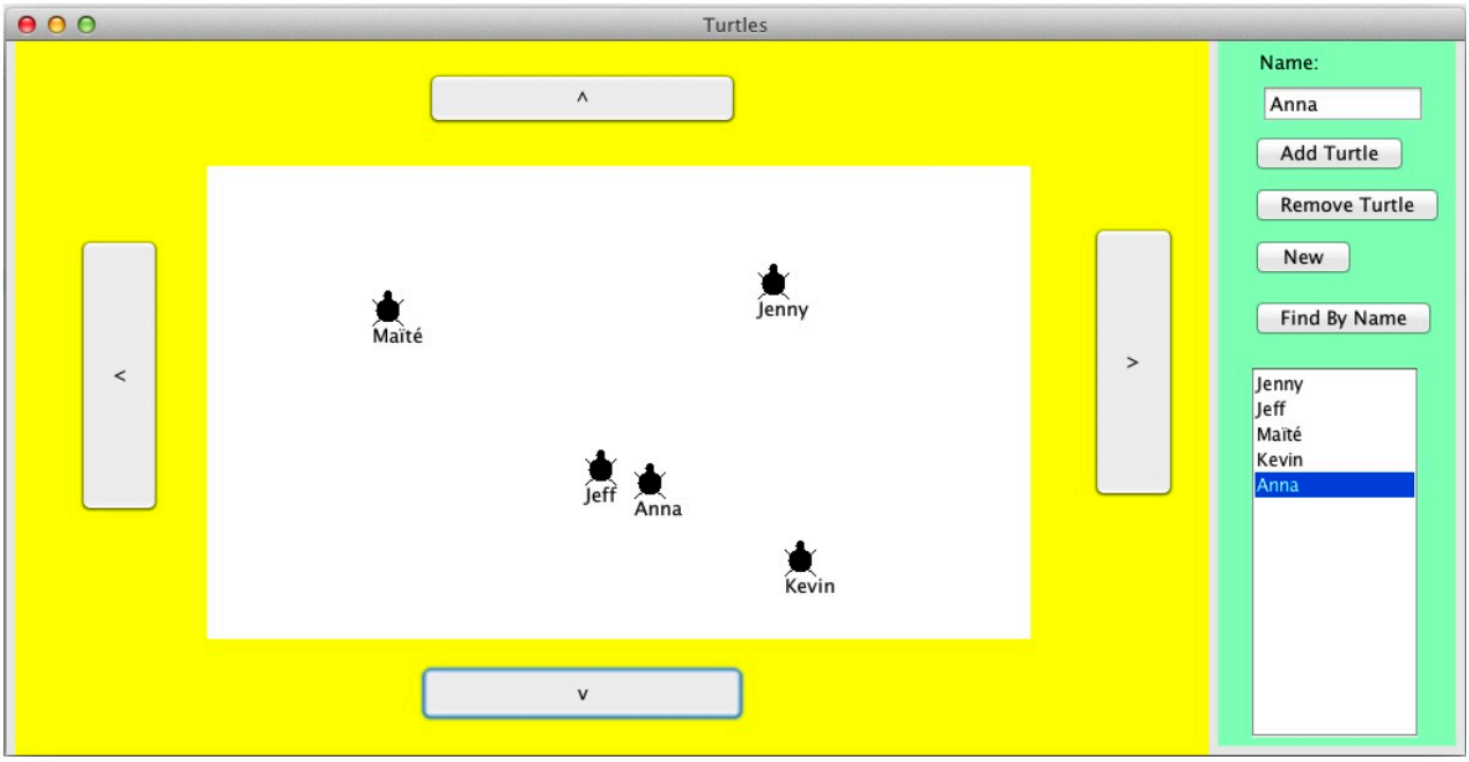
Exercice 2: Bouger plusieurs tortues

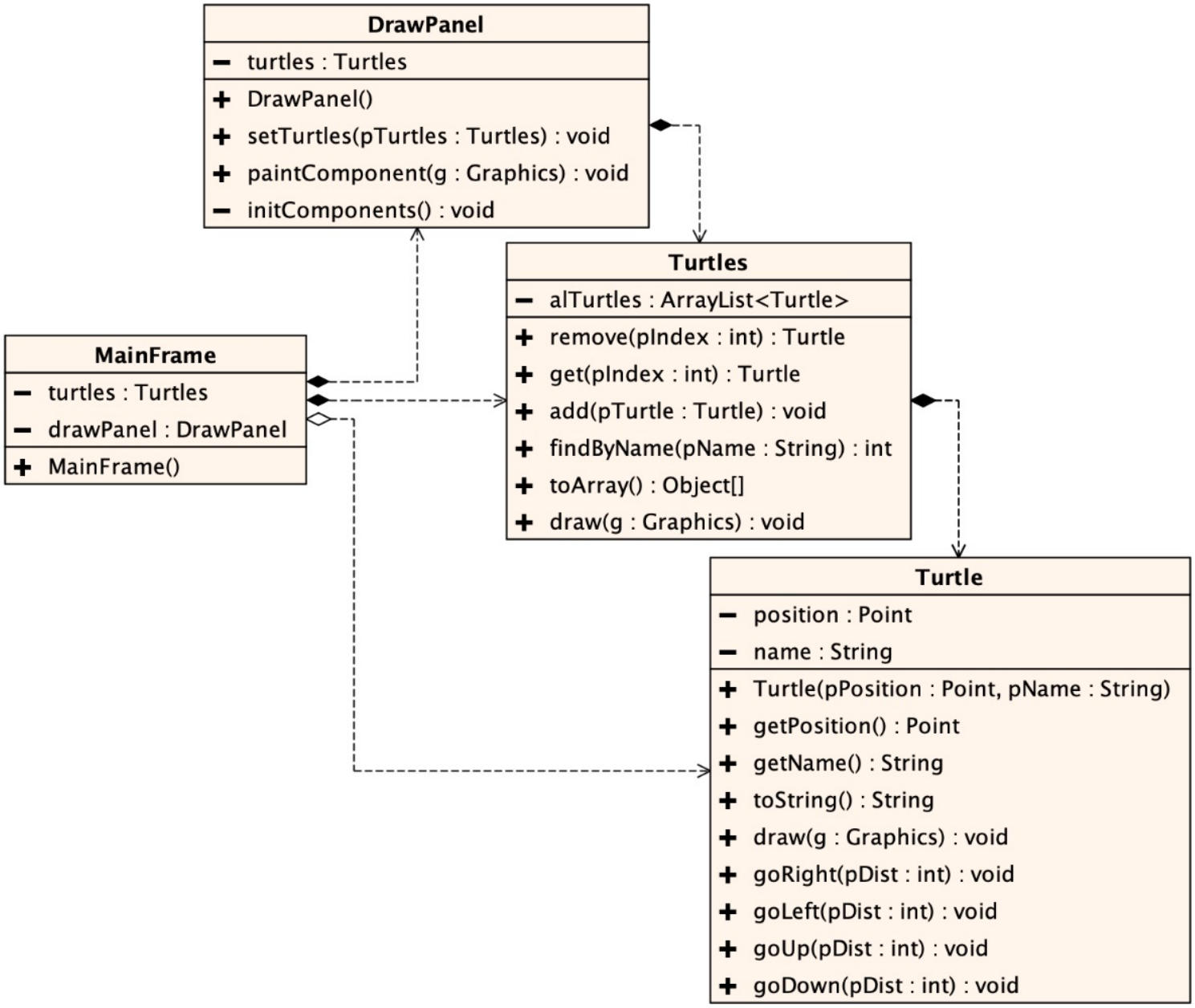
Ce programme consiste à étendre le programme précédent (**version 1A**) de la manière suivante :

- Plusieurs tortues peuvent être affichées et déplacées sur le panneau
- Une tortue a un nom **`name`** qui est affiché sur le panneau
- Les noms des tortues sont également affichés dans un composant de type **`JList`**
- La tortue sélectionnée dans la liste peut être déplacée sur le panneau
- Le bouton **Add Turtle** permet d'ajouter une nouvelle tortue qui est ensuite placée à une position aléatoire sur le panneau. Cette tortue porte le nom saisi dans la boîte d'édition **Name**
- Le bouton **Remove Turtle** supprime la tortue sélectionnée dans la liste
- Le bouton **New** crée une nouvelle liste de tortues
- Le bouton **Find By Name** permet de chercher dans la liste la tortue qui porte le nom saisi dans la boîte d'édition **Name**. Si une tortue est trouvée alors elle est marquée dans la liste. Si plusieurs tortues portent le nom recherché alors la première est sélectionnée.
- la classe **Turtles** possède une méthode **`draw(Graphics g)`** qui dessine toutes les tortues de la liste sur un canevas **`g`**.



Pour vous aider dans la conception du programme, vous trouvez dans la suite le schéma UML des **CLASSES** :

- la classe **`Turtle`** représentant une tortue,
- la classe **`Turtles`** qui gère une liste de tortues,
- la classe **`DrawPanel`** qui fait dessiner les tortues sur son canevas,
- la classe **`MainFrame`** qui est représentée de façon simplifiée (sans les composants et les méthodes de réaction). Mais il est bien visible qu'elle contrôle le déroulement et les interactions entre les autres classes.


DrawPanel
- turtles : Turtles
+ DrawPanel()
+ setTurtles(pTurtles : Turtles) : void
+ paintComponent(g : Graphics) : void
- initComponents() : void
Turtles
- alTurtles : ArrayList<Turtle>
+ remove(pIndex : int) : Turtle
+ get(pIndex : int) : Turtle
+ add(pTurtle : Turtle) : void
+ findByName(pName : String) : int
+ toArray() : Object[]
+ draw(g : Graphics) : void
MainFrame
- turtles : Turtles
- drawPanel : DrawPanel
+ MainFrame()
Turtle
- position : Point
- name : String
+ Turtle(pPosition : Point, pName : String)
+ getPosition() : Point
+ getName() : String
+ toString() : String
+ draw(g : Graphics) : void
+ goRight(pDist : int) : void
+ goLeft(pDist : int) : void
+ goUp(pDist : int) : void
+ goDown(pDist : int) : void

```
1
2   import java.awt.Color;
3   import java.awt.Graphics;
4
5   /*
5    * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.
6   txt to change this license
6    * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to
7   edit this template
8    */
9
10  /**
11   *
12   * @author luxformel
13   */
14  public class DrawPanel extends javax.swing.JPanel {
15      private Turtles turtles;
16
17      public void setTurtles(Turtles t) {
18          this.turtles = t;
19      }
20
21
22      /**
23       * Creates new form DrawPanel
24       */
25      public DrawPanel() {
26          initComponents();
27      }
28
29      /**
29       * This method is called from within the constructor to initialize the
30  form.
30       * WARNING: Do NOT modify this code. The content of this method is
31  always
32       * regenerated by the Form Editor.
33       */
36      @SuppressWarnings("unchecked")
48
49
50      @Override
51      protected void paintComponent(Graphics g) {
51          super.paintComponent(g); // Generated from nbfs:
52  //nbhost/SystemFileSystem/Templates/Classes/Code/OverriddenMethodBody
53          //Dessin d'un fond blanc sur le drawPanel
54          g.setColor(Color.white);
55          g.fillRect(0, 0, getWidth(), getHeight());
56
57          //Ce contrôle doit toujours être effectué, afin d'éviter des erreurs
58          //genre "Null pointer exception" directement dans le MainFrame
59          if (turtles != null)
60              turtles.draw(g);
```

```
1
2   import java.awt.Point;
3   import javax.swing.JColorChooser;
4
5   /*
5    * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.
6   txt to change this license
6    * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to
7   edit this template
8    */
9
10  /**
11   *
12   * @author luxformel
13   */
14  public class MainFrame extends javax.swing.JFrame {
15
16      //Instanciation / Création d'un nouvel objet de type Turtles
17      private Turtles turtles = new Turtles();
18
19      /**
20       * Creates new form MainFrame
21       */
22      public MainFrame() {
23          initComponents();
24
25          //L'objet instancié de type turtles doit être passé au drawPanel
26          drawPanel.setTurtles(turtles);
27
28          //Raffraîchir l'interface
29          updateView();
30          //positionLabel.setText("X:" + position.x + "; Y:" + position.y);
31      }
32
33      public void updateView(){
34          //Redéssine la surface de dessin
35          drawPanel.repaint();
36
36          //Affichage de la ArrayList contenant les tortues dans la JList
37  (Vitrine)
38          turtlesList.setListData(turtles.toArray());
39      }
40
41      /**
41       * This method is called from within the constructor to initialize the
42  form.
42       * WARNING: Do NOT modify this code. The content of this method is
43  always
44       * regenerated by the Form Editor.
45       */
48      @SuppressWarnings("unchecked")
60
```

```
62
73
80
87
94
101
103
105
112
119
125
193
196
197
197     private void rightButtonActionPerformed(java.awt.event.ActionEvent evt)
198 {
199         //Récupérer l'indice de la tortue sélectionnée
200         int index = turtlesList.getSelectedIndex();
201         //Si une tortue est sélectionnée
201         //Alors la tortue sélectionée de la ArrayList est mise dans la
202 variable t
203         if (index != -1){
204             //Récupère la tortue à l'indice index
205             Turtle t = turtles.get(index);
206 
207             //Déplace la tortue
208             t.goRight(10, drawPanel.getWidth());
209 
210 
211             updateView();
212 
213             //Reseléctionne la tortue dans la JList
214             turtlesList.setSelectedIndex(index);
215         }
216     }
217 
217     private void leftButtonActionPerformed(java.awt.event.ActionEvent evt)
218 {
219         //Récupérer l'indice de la tortue sélectionnée
220         int index = turtlesList.getSelectedIndex();
221         //Si une tortue est sélectionnée
221         //Alors la tortue sélectionée de la ArrayList est mise dans la
222 variable t
223         if (index != -1){
224             Turtle t = turtles.get(index);
225             t.goLeft(10);
226             updateView();
227             turtlesList.setSelectedIndex(index);
228         }
229     }
230 
230     private void downButtonActionPerformed(java.awt.event.ActionEvent evt)
```

```
231 {
232         //Récupérer l'indice de la tortue sélectionnée
233         int index = turtlesList.getSelectedIndex();
234         //Si une tortue est sélectionnée
234         //Alors la tortue sélectionée de la ArrayList est mise dans la
235 variable t
236         if (index != -1){
237             Turtle t = turtles.get(index);
238             t.goDown(10, drawPanel.getHeight());
239             updateView();
240             turtlesList.setSelectedIndex(index);
241         }
242     }
243 
244     private void upButtonActionPerformed(java.awt.event.ActionEvent evt) {
245         //Récupérer l'indice de la tortue sélectionnée
246         int index = turtlesList.getSelectedIndex();
247         //Si une tortue est sélectionnée
247         //Alors la tortue sélectionée de la ArrayList est mise dans la
248 variable t
249         if (index != -1){
250             Turtle t = turtles.get(index);
251             t.goUp(10);
252 
253             updateView();
254 
255             turtlesList.setSelectedIndex(index);
256         }
257 
258     }
259 
259     private void addTurtleButtonActionPerformed(java.awt.event.ActionEvent
260 evt) {
261         int minX = 10;
262         int maxX = drawPanel.getWidth() - 10;
263         int minY = 13;
264         int maxY = drawPanel.getHeight() - 10;
265 
266 
267         int x = (int)(Math.random() * (maxX - minX + 1)) + minX;
268         int y = (int)(Math.random() * (maxY - minY + 1)) + minY;
269 
270         Point position = new Point (x, y);
271         String name = nameTextField.getText();
272 
273         //if (! name.isEmpty()){
274         if (! nameTextField.getText().equals("")){
275             Turtle t = new Turtle(position, name);
276             turtles.add(t);
277             updateView();
278         }
279     }
```

```
280
280     private void nameTextFieldActionPerformed(java.awt.event.ActionEvent
281 evt) {
282         // TODO add your handling code here:
283     }
284
285     /**
286      * @param args the command line arguments
287      */
288     public static void main(String args[]) {
310         /* Set the Nimbus look and feel */
311
312         /* Create and display the form */
313         java.awt.EventQueue.invokeLater(new Runnable() {
314             public void run() {
315                 new MainFrame().setVisible(true);
316             }
317         });
318     }
319
320     // Variables declaration - do not modify//GEN-BEGIN:variables
321     private javax.swing.JButton addTurtleButton;
322     private javax.swing.JButton downButton;
323     private DrawPanel drawPanel;
324     private javax.swing.JLabel jLabel1;
325     private javax.swing.JScrollPane jScrollPane1;
326     private javax.swing.JButton leftButton;
327     private javax.swing.JTextField nameTextField;
328     private javax.swing.JLabel positionLabel;
329     private javax.swing.JButton rightButton;
330     private javax.swing.JList turtlesList;
331     private javax.swing.JButton upButton;
332     // End of variables declaration//GEN-END:variables
333 }
334
```

```java

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Point;


/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author luxformel
 */
public class Turtle {
    //Déclaration des attributs
    private Point position;
    private String name;

    //Déclaration et définition du constructeur
    public Turtle(Point position, String pName) {
        this.position = position;
        name = pName;
    }

    //Getters & setters
    public Point getPosition() {
        return position;
    }

    //Cette méthode permet de déplacer la tortue vers la droite
    //Le paramètre width correspond à la largeur de la surface de dessin (DrawPanel)
    //Avant de faire un pas à droite, un contrôle est effectué. Si la tortue
    //en faisant le pas ne sors pas de la surface de dessin, alors
    //la tortue peut faire le pas.
    //Sinon, la tortue est placée à la limite droite de la surface de dessin
    //Les valeurs +10 et -10 correspondent aux pixels qui séparent la position de la tortue
    //et ses extrémités
    public void goRight(int pDist, int width){
        if (position.x + 10 + pDist <= width)
            position.x = position.x + pDist;
        else
            position.x = width - 10;
    }

```

```
47      //Cette méthode permet de déplacer la tortue vers la gauche
47      //Avant de faire un pas à gauche, un contrôle est effectué. Si la
48  tortue
49      //en faisant le pas ne sors pas de la surface de dessin, alors
50      //la tortue peut faire le pas.
50      //Sinon, la tortue est placée à la limite gauche de la surface de
51  dessin
51      //Les valeurs +10 et -10 correspondent aux pixels qui séparent la
52  position de la tortue
53      //et ses extrémités
54      public void goLeft(int pDist){
55          if (position.x - 10 - pDist >= 0)
56              position.x = position.x - pDist;
57          else
58              position.x = 10;
59      }
60
61      //Cette méthode permet de déplacer la tortue vers le haut
61      //Avant de faire un pas vers le haut, un contrôle est effectué. Si la
62  tortue
63      //en faisant le pas ne sors pas de la surface de dessin, alors
64      //la tortue peut faire le pas.
64      //Sinon, la tortue est placée à la limite supérieure de la surface de
65  dessin
65      //Les valeurs +13 et -13 correspondent aux pixels qui séparent la
66  position de la tortue
67      //et ses extrémités
68      public void goUp(int pDist){
69          if (position.y - 13 - pDist >= 0)
70              position.y = position.y - pDist;
71          else
72              position.y = 13;
73      }
74
75      //Cette méthode permet de déplacer la tortue vers le bas
75      //Le paramètre height correspond à la hauteur de la surface de dessin
76  (DrawPanel)
76      //Avant de faire un pas ver le bas, un contrôle est effectué. Si la
77  tortue
78      //en faisant le pas ne sors pas de la surface de dessin, alors
79      //la tortue peut faire le pas.
79      //Sinon, la tortue est placée à la limite inférieure de la surface de
80  dessin
80      //Les valeurs +10 et -10 correspondent aux pixels qui séparent la
81  position de la tortue
82      //et ses extrémités
83      public void goDown(int pDist, int height){
84          if (position.y + 10 + pDist <= height)
85              position.y = position.y + pDist;
86          else
87              position.y = height - 10;
88      }
```

```
89
90      //Cette méthode dessine la tortue
91      public void draw(Graphics g){
92          g.setColor(Color.black);
93          g.fillOval(position.x - 7, position.y - 7, 15, 15);
94          g.fillOval(position.x - 2 , position.y - 13, 5, 6);
94          g.drawLine(position.x - 10, position.y - 7, position.x + 10,
95  position.y + 10);
95          g.drawLine(position.x - 10, position.y + 10, position.x + 10,
96  position.y - 7);
97      }
98
99      public String toString(){
100         return name + " (X:" + position.x + ";" + " Y:" + position.y +")";
101     }
102
103 }
104
```

```java

import java.awt.Graphics;
import java.util.ArrayList;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.
txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
this template
 */

/**
 *
 * @author luxformel
 */
public class Turtles {
    //Liste contenant toutes les tortues
    private ArrayList<Turtle> alTurtles = new ArrayList<>();

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

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

    public boolean add(Turtle e) {
        return alTurtles.add(e);
    }

    //Cette méthode dessine toutes les tortues contenues dans la liste
    //La liste est parcourue un élément après l'autre
    //Pour chaque élément, la méthode draw est appelée
    public void draw(Graphics g){
        for(int i = 0; i < alTurtles.size(); i++){
           alTurtles.get(i).draw(g);
        }
    }
}

```