Laboratoire 4 – Programmation évènementielle

ITI 1521. Introduction à l'informatique II – Hiver 2021

SOLUTION

/10

Question 1: (1 POINT)

/********Classe GUI**********/**

```
import javax.swing.*;
public class GUI extends JFrame{
  public static final int DRAW_SIZE = 400;
  public GUI(){
     super("GUI 1");
     setSize(DRAW_SIZE, DRAW_SIZE);
     setDefaultCloseOperation(EXIT_ON_CLOSE);
  }
  public static void main(String[] args) {
    GUI gui;
    gui = new GUI();
    gui.setVisible(true);
  }
}
```

Question 2: (1 POINT)

/**********Classe GUI**********/

```
import javax.swing.*;
import java.awt.*;
public class GUI extends JFrame{
      public static final int DRAW_SIZE = 400;
      public GUI(){
      super("GUI 2");
      setSize(DRAW_SIZE, DRAW_SIZE);
      setDefaultCloseOperation(EXIT_ON_CLOSE);
      JPanel panneau1 = new JPanel();
      panneau1.setBackground(Color.BLUE);
       add(panneau1, BorderLayout.NORTH);

       JButton bouton1 = new JButton("Shape");
       panneau1.add(bouton1);

       JButton bouton2 = new JButton("Quit");
       panneau1.add(bouton2);

       JPanel panneau2 = new JPanel();
       add(panneau2, BorderLayout.CENTER);
       panneau2.setBackground(Color.WHITE);

  }
   public static void main(String[] args) {
     GUI gui;
     gui= new GUI();
     gui.setVisible(true);
   }
}
```

Question 3: (1 POINT)

/**********Classe GUI**********/

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;


public class GUI extends JFrame implements ActionListener {

    public static final int DRAW_SIZE = 400;

     public GUI(){
     super("GUI 3");
     setSize(DRAW_SIZE, DRAW_SIZE);
     setDefaultCloseOperation(EXIT_ON_CLOSE);
     JPanel panneau1 = new JPanel();
      panneau1.setBackground(Color.BLUE);
      add(panneau1, BorderLayout.NORTH);


      JButton bouton1, bouton2;
      bouton1 = new JButton("Shape");
      bouton1.addActionListener(this);
      panneau1.add(bouton1);


      bouton2 = new JButton("Quit");
      bouton2.addActionListener(this);
      panneau1.add(bouton2);


      JPanel panneau2 = new JPanel();
      add(panneau2, BorderLayout.CENTER);
      panneau2.setBackground(Color.WHITE);


  }

   public void actionPerformed(ActionEvent e) {

         System.out.println("actionPerformed was called");

   }

  public static void main(String[] args) {
    GUI gui;
    gui= new GUI();
    gui.setVisible(true);
  }
}
```

Question 4: (1 POINT)

/**********Classe GUI**********/

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;


public class GUI extends JFrame implements ActionListener {

    public static final int DRAW_SIZE = 400;

      public GUI(){

      super("GUI 4");
      setSize(DRAW_SIZE, DRAW_SIZE);
      setDefaultCloseOperation(EXIT_ON_CLOSE);
      JPanel panneau1 = new JPanel();
       panneau1.setBackground(Color.BLUE);
       add(panneau1, BorderLayout.NORTH);


      JButton bouton1, bouton2;
       bouton1 = new JButton("Shape");
       bouton1.addActionListener(this);
       panneau1.add(bouton1);


       bouton2 = new JButton("Quit");
       bouton2.addActionListener(this);
       panneau1.add(bouton2);


       JPanel panneau2 = new JPanel();
       add(panneau2, BorderLayout.CENTER);
       panneau2.setBackground(Color.WHITE);
  }
   public void actionPerformed ( ActionEvent e ) {

   System.out.println ( " actionPerformed was called : " + e.getActionCommand ( ) ) ;

}


   public static void main(String[] args) {

     GUI gui;
     gui= new GUI();
     gui.setVisible(true);
  }
}
```

Question 5: (2.5 POINTS)

/**********Classe GUI**********/

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.Point;
public class GUI extends JFrame implements ActionListener {

   public static final int MAX_POINTS = 10;
   private Point[] tabPoints = new Point[MAX_POINTS];
   private int number = 0;
   private JPanel draw;
   public static final int DRAW_SIZE = 400;

  public GUI(){
     super("GUI 5");
     setSize(DRAW_SIZE, DRAW_SIZE);

     JPanel panneau1 = new JPanel();
      panneau1.setBackground(Color.BLUE);
      add(panneau1, BorderLayout.NORTH);

      JButton bouton1, bouton2;
      bouton1 = new JButton("Shape");
      bouton1.addActionListener(this);
      panneau1.add(bouton1);

      bouton2 = new JButton("Quit");
      bouton2.addActionListener(this);
      panneau1.add(bouton2);

      draw = new DrawPanel();
      draw.setBackground(Color.WHITE);
      add(draw, BorderLayout.CENTER);

       setDefaultCloseOperation(EXIT_ON_CLOSE);

  }

  public void actionPerformed(ActionEvent e) {
      if (e.getActionCommand().equals("Shape")){
        addPoint(new Point((int)(Math.random() * DRAW_SIZE), (int)(Math.random() *
DRAW_SIZE)));
         draw.repaint();

 }

      else
          System.out.println("Pression du bouton Quit");
          }
```

```
private void addPoint(Point p) {
      if (number < MAX_POINTS){
       tabPoints[number++] = p;
        //draw.repaint();
        }
   }


   private class DrawPanel extends JPanel {
      public void paint(Graphics g) {
         super.paint(g);
         for (int i = 1; i < number; i++)
            g.drawLine((int)(tabPoints[i - 1].getX()), (int)(tabPoints[i - 1].getY()),
                  (int)(tabPoints[i].getX()), (int)(tabPoints[i].getY()));
      }
   }


  public static void main(String[] args) {
    GUI gui;
    gui = new GUI();
    gui.setVisible(true);
  }
}
```

Question 6: (2.5 POINTS)

/**********Classe GUI**********/

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.Point;

public class GUI extends JFrame implements ActionListener {
   public static final int MAX_POINTS = 10;
   private Point[] tabPoints = new Point[MAX_POINTS];
   private int number = 0;
   private JPanel draw;
   public static final int DRAW_SIZE = 400;
   private Color changeColorDraw = Color.BLACK;

   public GUI(){
     super("GUI 6");
     setSize(DRAW_SIZE, DRAW_SIZE);

      JPanel panneau1 = new JPanel();
      panneau1.setBackground(Color.BLUE);
      add(panneau1, BorderLayout.NORTH);

      JButton bouton1, bouton2;
      bouton1 = new JButton("Shape");
      bouton1.addActionListener(this);
      panneau1.add(bouton1);

      bouton2 = new JButton("Quit");
      bouton2.addActionListener(this);
      panneau1.add(bouton2);

      draw = new DrawPanel();
      draw.setBackground(Color.WHITE);
      add(draw, BorderLayout.CENTER);

      setJMenuBar(createMenu());

       setDefaultCloseOperation(EXIT_ON_CLOSE);

 }

    JMenuBar createMenu() {
      JMenuBar bar = new JMenuBar();

      JMenu menu = new JMenu("Color Menu");
      bar.add(menu);

      JMenuItem item = new JMenuItem("Black");
```

```
        item.addActionListener(this);
        menu.add(item);

        item = new JMenuItem("Red");
        item.addActionListener(this);
        menu.add(item);

        item = new JMenuItem("Green");
        item.addActionListener(this);
        menu.add(item);

        item = new JMenuItem("Blue");
        item.addActionListener(this);
        menu.add(item);

        return bar;
    }

    public void actionPerformed(ActionEvent e) {

          if (e.getActionCommand().equals("Black")) {
            changeColorDraw = Color.BLACK;
            draw.repaint();
        }
        else if (e.getActionCommand().equals("Red")) {
            changeColorDraw = Color.RED;
            draw.repaint();
        }
        else if (e.getActionCommand().equals("Blue")) {
            changeColorDraw = Color.BLUE;
            draw.repaint();
        }
        else if (e.getActionCommand().equals("Green")) {
            changeColorDraw = Color.GREEN;
            draw.repaint();
        }
       else if (e.getActionCommand().equals("Shape")){
            addPoint(new Point((int)(Math.random() * DRAW_SIZE), (int)(Math.random() *
DRAW_SIZE)));
            draw.repaint();
         }

        else
            System.out.println("Pression du bouton Quit");
    }

    private void addPoint(Point p) {
        if (number < MAX_POINTS)
            tabPoints[number++] = p;
          // draw.repaint();
    }
```

```
   private class DrawPanel extends JPanel {
      public void paint(Graphics g) {
         super.paint(g);
         Color colorDraw = g.getColor();
         g.setColor(changeColorDraw);

         for (int i = 1; i < number; i++)
            g.drawLine((int)(tabPoints[i - 1].getX()), (int)(tabPoints[i - 1].getY()),
                  (int)(tabPoints[i].getX()), (int)(tabPoints[i].getY()));
         g.setColor(colorDraw);
      }
   }


  public static void main(String[] args) {
    GUI gui;
    gui = new GUI();
    gui.setVisible(true);
  }
}
```

Question 7: (1 POINT)

/**********Classe GUI**********/

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.Point;

public class GUI extends JFrame implements ActionListener {
   public static final int MAX_POINTS = 10;
   private Point[] tabPoints = new Point[MAX_POINTS];
   private int number = 0;
   private JPanel draw;
   public static final int DRAW_SIZE = 400;
   private Color changeColorDraw = Color.BLACK;

   public GUI(){
     super("GUI 6");
     setSize(DRAW_SIZE, DRAW_SIZE);

      JPanel panneau1 = new JPanel();
      panneau1.setBackground(Color.BLUE);
      add(panneau1, BorderLayout.NORTH);

      JButton bouton1, bouton2;
      bouton1 = new JButton("Shape");
      bouton1.addActionListener(this);
      panneau1.add(bouton1);
```

```
        bouton2 = new JButton("Quit");
        bouton2.addActionListener(this);
        panneau1.add(bouton2);

        draw = new DrawPanel();
        draw.setBackground(Color.WHITE);
        add(draw, BorderLayout.CENTER);

        setJMenuBar(createMenu());

         setDefaultCloseOperation(EXIT_ON_CLOSE);

    }

       JMenuBar createMenu() {
        JMenuBar bar = new JMenuBar();

        JMenu menu = new JMenu("Color Menu");
        bar.add(menu);

        JMenuItem item = new JMenuItem("Black");
        item.addActionListener(this);
        menu.add(item);

        item = new JMenuItem("Red");
        item.addActionListener(this);
        menu.add(item);

        item = new JMenuItem("Green");
        item.addActionListener(this);
        menu.add(item);

        item = new JMenuItem("Blue");
        item.addActionListener(this);
        menu.add(item);

        return bar;
     }

    public void actionPerformed(ActionEvent e) {

          if (e.getActionCommand().equals("Black")) {
            changeColorDraw = Color.BLACK;
            draw.repaint();
        }
        else if (e.getActionCommand().equals("Red")) {
            changeColorDraw = Color.RED;
            draw.repaint();
        }
        else if (e.getActionCommand().equals("Blue")) {
```

```
            changeColorDraw = Color.BLUE;
            draw.repaint();
        }
        else if (e.getActionCommand().equals("Green")) {
            changeColorDraw = Color.GREEN;
            draw.repaint();
        }
       else if (e.getActionCommand().equals("Shape")){
            addPoint(new Point((int)(Math.random() * DRAW_SIZE), (int)(Math.random() *
DRAW_SIZE)));
            draw.repaint();
         }

        else if (e.getActionCommand().equals("Quit"))
            System.exit(0);
    }

   private void addPoint(Point p) {
        if (number < MAX_POINTS)
            tabPoints[number++] = p;
          // draw.repaint();
    }

    private class DrawPanel extends JPanel {
        public void paint(Graphics g) {
            super.paint(g);
            Color colorDraw = g.getColor();
            g.setColor(changeColorDraw);

            for (int i = 1; i < number; i++)
                g.drawLine((int)(tabPoints[i - 1].getX()), (int)(tabPoints[i - 1].getY()),
                        (int)(tabPoints[i].getX()), (int)(tabPoints[i].getY()));
            g.setColor(colorDraw);
        }
    }


   public static void main(String[] args) {
     GUI gui;
     gui = new GUI();
     gui.setVisible(true);
   }
}
```