import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

class move extends JPanel{

  Timer timer;

  int flagg=0;

  int x=0,y=50,width=40,height=40;

  int radius=(width/2);

  int frXPos=500;

  int speedX=1;

  JButton bt;

  JButton bt1;

  JButton bt2;

  JButton bt3;

 /\* You might be asking why I use the radius of the ball in the code above. The x - position of the   \*ball is its middle point, which means that the left or the right border of the ball is its x - position   \*plus or less the radius. To make us believe that the ball would be bouncing when it hits the wall,   \*we have to add or substract the radius to / from the the value of x\_pos.

  \*/

  move() {

 bt = new JButton("->");

 add(bt);

 bt1 = new JButton("<-");

 add(bt1);

 bt2 = new JButton("up");

 add(bt2);

 bt3 = new JButton("down");

 add(bt3);

     final ActionListener taskPerformer = new ActionListener () {

         public void actionPerformed(ActionEvent ae) {

if( x<= (500 - width))

x++;

             repaint();

          }

     };

final ActionListener taskPerformer1 = new ActionListener () {

         public void actionPerformed(ActionEvent ae) {

if(x>0)

x--;

             repaint();

          }

     };

final ActionListener taskPerformer2 = new ActionListener () {

         public void actionPerformed(ActionEvent ae) {

            if(y>0)

y--;

             repaint();

          }

     };

final ActionListener taskPerformer3 = new ActionListener () {

         public void actionPerformed(ActionEvent ae) {

            if(y<500-70)

y++;

             repaint();

          }

     };

     bt.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

if(flagg==1)

timer.stop();

timer= new Timer(0,taskPerformer);

   timer.start();

flagg=1;

            repaint();

}

});

     bt1.addActionListener(new ActionListener() {

  public void actionPerformed(ActionEvent arg0) {

if(flagg==1)

timer.stop();

timer= new Timer(0,taskPerformer1);

   timer.start();

flagg=1;

              repaint();

  }

  });

     bt2.addActionListener(new ActionListener() {

  public void actionPerformed(ActionEvent arg0) {

              if(flagg==1)

timer.stop();

timer= new Timer(0,taskPerformer2);

   timer.start();

flagg=1;

              repaint();

  }

  });

     bt3.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent arg0) {

    if(flagg==1)

timer.stop();

timer= new Timer(0,taskPerformer3);

   timer.start();

flagg=1;

              repaint();

    }

    });

  }

  public void paintComponent(Graphics g) {

     super.paintComponent(g);

     g.setColor(Color.red);

     g.fillOval(x,y,width,height);

 }

}

public class MovingBall {

  MovingBall() {

      JFrame fr=new JFrame("Moving Ball");

      move o=new move();

      fr.add(o);

     fr.setVisible(true);

     fr.setSize(500,500);          // width of the frame is 500

     fr.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

  }

 public static void main(String args[]) {

   new MovingBall();

 }

}