Chaos opgave:

Opgave 1(tælle semaphore)
Jeg har ikke lavet denne opgave, jeg har ærlig talt ikke leget så meget med semaphore.

Opgave 2: bolde… den her virker okay – den blinker ikke, jeg har prøvet at implementere invokeLater for at optimere, men når jeg implementere det så sletter den ikke bolden hver gang den bliver tegnet, så jeg får nogle lange striber, så du kan se bort fra doThreadSafeWork() metoden, den er der bare for at vise jeg har prøvet.

```
package ball;
/**
 * @author ArneK
 */
import java.awt.*;

import javax.swing.SwingUtilities;

public class  Ball implements Runnable, Konstanter
{
  double x, y, fartx, farty;
  int d;
  private Color c;
  Graphics g;
  private static int caseNumber = 0;
  public Ball(Graphics g1, int x1, int y1,Color c, int d1)
  {
            g = g1; x = x1; y = y1; this.c = c; d = d1;
            this.fartx = 1 * Math.random();
            this.farty = 1 * Math.random();

    Thread t = new Thread(this);
    t.start();
  }
  protected void doThreadSafeWork()
            {
                        if(SwingUtilities.isEventDispatchThread())
                        {
                                    switch (caseNumber)
                                    {
                        case 1:
                                    synchronized(g)
                                    {
                                                g.setColor(Color.WHITE);
                                                g.fillOval((int) x, (int) y,
d, d);
                                    }

                        break;

                        case 2:
```

```
                                    synchronized(g)
                                    {
                                                            g.setColor(c);
                                                            g.fillOval((int) x, (int) y, d, d);

                            }


                                    break;

                                      }
                        }



                        else
                        {
                                    Runnable callDoThreadSafeWork = new Runnable()
                                    {
                                                public void run()
                                                {
                                                            doThreadSafeWork();
                                                }
                                    };
                                    SwingUtilities.invokeLater(callDoThreadSafeWork);
                        }
            }
@Override
public void run() {
            while (true)
    {

                        //jeg kalder slet ikke InvokeLater and wait metoden - jeg kan ikke få den til at virke ordentligt,
                        //den kan ikke finde ud af at slette boldene ordentligt, så jeg har nøjes med at sætte synchronised(g) rundt om
//                      caseNumber = 1;
//          doThreadSafeWork();

             synchronized(g)
            {
                        g.setColor(Color.WHITE);
                                    g.fillOval((int) x, (int) y, d, d);
         }


      x = x + fartx;
      y = y + farty;

//      caseNumber = 2;
//      doThreadSafeWork();
      synchronized(g)
                        {
                                    g.setColor(c);
                                    g.fillOval((int) x, (int) y, d, d);
                        }
```

```
      farty = farty + gravity;


      if (x    < 0)     fartx =  Math.abs(fartx);
      if (x+d > sizeJFrameX) fartx = -Math.abs(fartx);
      if (y    < 0)     farty =  Math.abs(farty);
      if (y+d >sizeJFrameY)  farty = -Math.abs(farty);




      try { Thread.sleep(20); } catch (Exception e) {};
   }

   }

}


package ball;
/**
 * @author ArneK
 */

public interface Konstanter {
            final static int sizeJFrameX = 800;
            final static int sizeJFrameY = 600;
            final static double gravity = 0.2;
}


package ball;
/**
 * @author ArneK
 */
import java.awt.*;

import javax.swing.*;

public class TestBolde  implements Konstanter
{
  public static void main(String[] arg) throws InterruptedException
  {
    JFrame f = new JFrame();
    f.setSize(sizeJFrameX,sizeJFrameY);
    f.getContentPane().setBackground(Color.WHITE);
    f.setVisible(true);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


    Graphics g = f.getGraphics();

    f.update(g);
    new Ball(g,300,40, Color.BLACK,60);
    new Ball(g,300,40, Color.BLUE,60);
```

```
        new Ball(g,600,40, Color.RED,60);



    }
}
```