```
/*******************************************
 * Ising.java                              *
 * ---------------------------------       *
 * Created by Bernd Nottelmann in 1996     *
 * Last update (by Peter Young): March 25, 1997*
 * Updated by David Wong for Matse 390AS   *
 *    Update the AWT API to support AWT 1.1    *
 *******************************************/

import java.lang.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;


public class Ising extends java.applet.Applet
    implements Runnable {
    /* Grid  parameter and Field content */
    private IsingLattice lattice;  // actual grid, here is where
                                   // all the physics happens

    /* Public Parameter */
    /* Java-Objekte und -Parameter */
    private Thread is;

    private short plotValue=1;           // Spin value

    private IsingCanvas can;
    private IsingPanel pan;

    private final int[] Panxy={100,15,50,10};// room for the control panel
                                            //elements

    final int timeout=100;        // pause interval for the picture generation
                                 // in milli-sec time variables

    private Temperature tempValueHolder; // value holder of temperature
    private boolean stopRequested; // want to stop the is thread?

    public void init() {
        // Dimension of Model
        int dim=2; // square and practical (from German chocolate commercial)

        int n=Integer.valueOf(getParameter("latt")).intValue();
        // fetch the grid extent
        tempValueHolder=new Temperature
            (Double.valueOf(getParameter("temp")).doubleValue());
        // temperature from the applet tag

        int period=2*n;   // time variable

        /* Grid installization */
        lattice=new IsingLattice(dim,n, period,tempValueHolder);
        lattice.initHot();

        /* Java Initialization Part */
        setLayout(new BorderLayout());

        add("Center", can = new IsingCanvas
            (this, Panxy, plotValue, n, lattice, tempValueHolder));
        add("South", pan = new IsingPanel(this, can,
                                          lattice, tempValueHolder));
        setUpObservers ();
    }
```

```
    private void setUpObservers () {
        tempValueHolder.addObserver (pan);
        tempValueHolder.addObserver (lattice);
    }

    public void start() {
        if (is == null) {
            is=new Thread(this);
            is.start();
        }
        pan.setEnabled(true);
    }

    public void stop() {
        pan.setEnabled(false);
        is.stop ();
        is = null;
    }

    public void brk() {
        is.suspend();
    }

    public void cont() {
        is.resume ();
    }

    public void run() {
        while (true) {
            long startTime = System.currentTimeMillis ();
            lattice.ssdUpdate();
            if (stopRequested) {
                stopRequested = false;
                brk ();
            }
            can.repaint ();
            try{
                long endTime = System.currentTimeMillis ();
                if (timeout - (endTime - startTime)>0) {
                    Thread.currentThread().sleep
                        (timeout - (endTime - startTime));
                }
                Thread.currentThread ().sleep (timeout);
            } catch (InterruptedException e) {}
        }
     }

    void initColdRequested () {
        lattice.initCold(-1);
        can.reset();
        //      can.redraw(false,true,true,true);
        can.repaint ();
    }

    void initHotRequested () {
        lattice.initHot();
        can.reset();
        //      can.redraw(false,true,true,true);
        can.repaint ();
    }

    void initWarmRequested () {
        lattice.initWarm();
```

```
        can.reset();
        //      can.redraw(false,true,true,true);
        can.repaint ();
    }

    void startRequested () {
        is.resume ();
    }

    void stopRequested () {
        stopRequested = true;
        //      can.SpinsDrawn=false;
        //      can.redraw(false,true,true,true);
        can.repaint ();
    }

    void stepRequested () {
        lattice.ssdUpdate ();
        can.repaint();
    }

    public static void main(String args[]) {
        //System.out.println("Here is main!");
        Frame frame=new Frame("Ising-Modell");
        Ising ising=new Ising();

        ising.init();
        ising.start();

        frame.add("Center",ising);
        frame.setSize(frame.getPreferredSize());
        frame.setVisible(true);
    }
}
```