```
/*
 * IsingPanel.java
 * ---------------------------
 * Separated from Ising.java by David Wong
 */

import java.awt.*;
import java.awt.event.*;
import java.text.NumberFormat;
import java.util.Observer;
import java.util.Observable;

class IsingPanel extends Panel implements Observer {
    private Ising is;
    private IsingCanvas can;
    private GridBagLayout GBLay;
    private GridBagConstraints c;
    private Double Temp;
    private TextField TempText;
    private Checkbox check;
    private IsingLattice lattice;
    private Temperature tempValueHolder;  /* object keep track of temperature */
    private NumberFormat format;
    private final int MAX_NO_OF_DISPLAY_DIGIT = 5;

    public IsingPanel(Ising aIs, IsingCanvas can, IsingLattice lattice,
                      Temperature aTempValueHolder) {
        this.lattice = lattice;
        this.is=aIs;
        this.can=can;
        this.tempValueHolder = aTempValueHolder;

        GBLay=new GridBagLayout();
        c=new GridBagConstraints();
        setLayout(GBLay);

        c.gridx=0; c.gridy=0;
        c.gridheight=1;
        c.fill=GridBagConstraints.BOTH;
        Label Blubber = new Label ("Temperature");
        Blubber.setFont (new Font ("TimesRoman", Font.PLAIN, 14));
        Blubber.setForeground (Color.red);
        GBLay.setConstraints(Blubber,c);
        add(Blubber);

        c.gridy=1;
        Temp=new Double(tempValueHolder.getTemperature ());
        if (Temp.doubleValue() == lattice.TempMax)
            Temp=new Double(Double.POSITIVE_INFINITY);
        TempText=new TextField(Temp.toString(),10);
        GBLay.setConstraints(TempText,c);
        add(TempText);
        TempText.addActionListener (new ActionListener () {
            public void actionPerformed (ActionEvent e) {
                tempValueHolder.setTemperature
                    (Double.valueOf
                     (TempText.getText ().trim ()).doubleValue ());
            }
        });
        makeButtons (c);
        format = NumberFormat.getInstance ();
        format.setMaximumFractionDigits (MAX_NO_OF_DISPLAY_DIGIT);
    }
```

```
    protected Button makebutton(String name,
                                GridBagLayout gb,
                                GridBagConstraints gbc) {
        Button button=new Button(name);
        gb.setConstraints(button,gbc);
        add(button);
        return button;
    }


    private void makeButtons (GridBagConstraints c) {
        c.gridx=1; c.gridy=0;
        makebutton("Init cold",GBLay,c).addActionListener
            (new ActionListener () {
                public void actionPerformed (ActionEvent evt) {
                    is.initColdRequested ();
                }
            });

        c.gridx=2;
        //  makebutton("Init hot",GBLay,c);
        c.gridx=3;
        makebutton("Init warm",GBLay,c).addActionListener
            (new ActionListener () {
                public void actionPerformed (ActionEvent evt) {
                    is.initWarmRequested ();
                }
            });

        c.gridx=4; c.gridheight=2;
        check=new Checkbox("Grid");
        check.setState(false);
        GBLay.setConstraints(check,c);
        add(check);
        check.addItemListener (new ItemListener () {
            public void itemStateChanged (ItemEvent evt) {
                if (can.gridIsForbidden ())
                    check.setVisible(false);
                else
                    can.setGridFlag(check.getState());
            }
        });

        c.gridx=1; c.gridy=1; c.gridheight=1;
        makebutton("Start",GBLay,c).addActionListener
            (new ActionListener () {
                public void actionPerformed (ActionEvent evt) {
                    is.startRequested ();
                }
            });
        c.gridx=2;
        makebutton("Stop",GBLay,c).addActionListener
            (new ActionListener () {
                public void actionPerformed (ActionEvent evt) {
                    is.stopRequested ();
                }
            });
        c.gridx=3;
        makebutton("Step",GBLay,c).addActionListener
            (new ActionListener () {
                public void actionPerformed (ActionEvent evt) {
                    is.stepRequested ();
                }
```

```
            });
    }

    public void update (Observable o, Object arg) {
        TempText.setText
            (format.format (tempValueHolder.getTemperature ()));
    }
}
```