Bill Bohling
Course: Java Programming 1
Date: 9 November 2010
Assignment: HW7

Objective: Implement a Windchill Calculator

Notes: The windchill formula supplied with the assignment wasn't working with subzero temperatures, since subtracting a negative number is equivalent to adding. I ended up using a different formula I found online.

Code:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class WindChillCalculator extends Applet {  // implement the
GUI here

  // Panels
  Panel windPanel;
  Panel tempPanel;
  Panel chillPanel;

  // Controls
  // windspeed panel
  Scrollbar windSlider;
  Label windSpeed;
  CheckboxGroup windScale;
  Checkbox mph;
  Checkbox kph;
  // temperature panel
  Scrollbar tempSlider;
  Label temp;
  CheckboxGroup tempScale;
  Checkbox fahr;
  Checkbox cels;

  // Results
  Label windChill;

  public void init(){

    setLayout(new GridLayout(3,1));
    windPanel = new Panel();
    windPanel.setLayout(new GridLayout(4,1));
    windPanel.setBackground(new Color(125,200,125));
    tempPanel = new Panel();
```

```
    tempPanel.setLayout(new GridLayout(4,1));
    tempPanel.setBackground(new Color(150,150,200));
    chillPanel = new Panel();
    chillPanel.setLayout(new GridLayout(1,1));
    chillPanel.setBackground(new Color(225,200,150));
    // windspeed panel
    windScale = new CheckboxGroup();
    windSlider = new Scrollbar(Scrollbar.HORIZONTAL, 5, 2, 5, 52);
    windSpeed = new Label("Windspeed:
"+Integer.toString(windSlider.getValue())+"MPH");
    mph = new Checkbox("MPH", windScale, true);
    kph = new Checkbox("KPH", windScale, false);
    add(windPanel);
    windPanel.add(windSlider);
    windPanel.add(windSpeed);
    windPanel.add(mph);
    windPanel.add(kph);

    // temperature panel
    tempScale = new CheckboxGroup();
    fahr = new Checkbox("F", tempScale, true);
    cels = new Checkbox("C", tempScale, false);
    tempSlider = new Scrollbar(Scrollbar.HORIZONTAL, 72, 2, -50, 92);
    temp = new Label("Temperature:
"+Integer.toString(tempSlider.getValue())+"F");
    add(tempPanel);
    tempPanel.add(tempSlider);
    tempPanel.add(temp);
    tempPanel.add(fahr);
    tempPanel.add(cels);

    add(chillPanel);
    windChill = new Label();
    chillPanel.add(windChill);
    // let the logic class handle everything else
    WindChillLogician wcl = new WindChillLogician(this);
  }

}
```

```
class WindChillLogician
  extends Applet
  implements ItemListener,
             AdjustmentListener
{

  private WindChillCalculator gui;
  private String windScale;
  private String tempScale;
  private int temperature;
  private int windspeed;

  WindChillLogician(WindChillCalculator wcc){
    gui = wcc;
    gui.kph.addItemListener(this);
    gui.mph.addItemListener(this);
    gui.fahr.addItemListener(this);
    gui.cels.addItemListener(this);
    gui.tempSlider.addAdjustmentListener(this);
    gui.windSlider.addAdjustmentListener(this);
    this.windScale = gui.windScale.getSelectedCheckbox().getLabel();
    this.tempScale = gui.tempScale.getSelectedCheckbox().getLabel();
    this.temperature = gui.tempSlider.getValue();
    this.windspeed = gui.windSlider.getValue();
    gui.windChill.setText(currentWindchill());
  }

  public void adjustmentValueChanged(AdjustmentEvent ae){
    // get which slider changed
    Object slider = ae.getSource();
    if (slider == gui.windSlider){
      setWindspeed(gui.windSlider.getValue());
      gui.windSpeed.setText(currentWindspeed());

    }
    else {
      setTemp(gui.tempSlider.getValue());
      gui.temp.setText(currentTemp());
    }
    gui.windChill.setText(currentWindchill());
    repaint();
  }

  public void itemStateChanged(ItemEvent ie){
    // get which button was checked
    Object checked = ie.getItemSelectable();
    if ((checked == gui.mph) || checked == gui.kph){
      setWindScale(gui.windScale.getSelectedCheckbox().getLabel());
      gui.windSpeed.setText(currentWindspeed());
    }
    else {
```

```
      setTempScale(gui.tempScale.getSelectedCheckbox().getLabel());
      gui.temp.setText(currentTemp());
      gui.windChill.setText(currentWindchill());
    }

    repaint();
  }

  // private methods

  private String currentTemp(){
    return "Temperature: "+Integer.toString(getTemp())+this.tempScale;
  }

  private String currentWindspeed(){
    return "Windspeed: "+Integer.toString(getWindspeed())
+this.windScale;
  }

  private String currentWindchill(){
    return "Windchill Index: "+Integer.toString(getWindChill())
+this.tempScale;
  }

  private void setTemp(int temp){
    this.temperature = temp;
  }

  private void setTempScale(String scale){
    this.tempScale = scale;
  }

  private void setWindspeed(int speed){
    this.windspeed = speed;
  }

  private void setWindScale(String scale){
    this.windScale = scale;
  }

  private int getTemp(){
    if (this.tempScale == "F"){
      return this.temperature;
    }
    else {
      return (this.temperature - 32) * 5 / 9;
    }
  }

  private int getWindspeed(){
    if (this.windScale == "MPH"){
```

```
      return this.windspeed;
    }
    else {
      return (int)(this.windspeed * 1.6);
    }
  }

  private int getWindChill(){
    int wcIndex = (int)(35.74 + 0.6215 * this.temperature -
35.75*Math.pow(this.windspeed, 0.16) +
(0.4275*this.temperature*Math.pow(this.windspeed,0.16)));
    if (this.tempScale == "F"){
      return wcIndex;
    }
    else {
      return (wcIndex-32) * 5 / 9;
    }
  }
}// windChillLogician
```

Applet screenshots (running in Linux):