THERMOMETER 1.0

Project Design Document

Team Solo

Guy One <guy1@rit.edu>

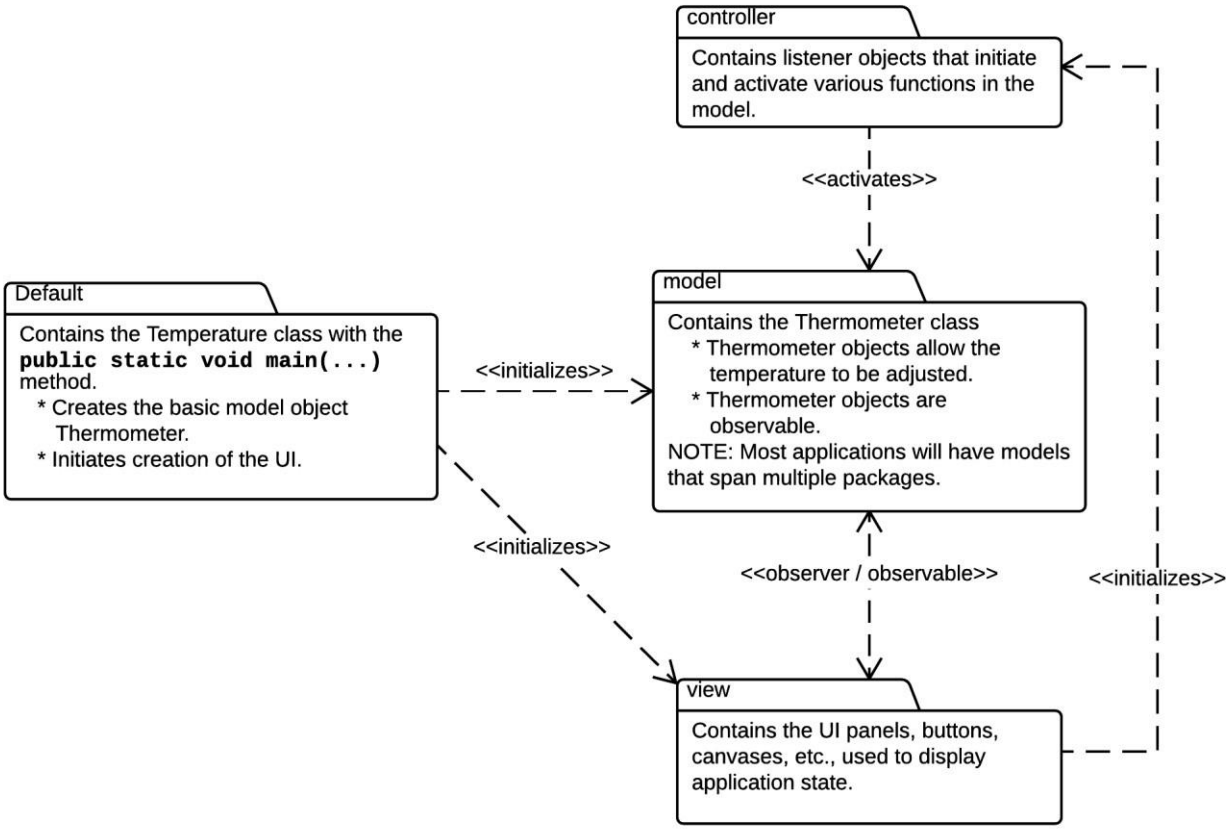
Project Summary

The project models a thermometer whose temperature is reported in either Fahrenheit or Celsius mode (to the nearest 1/10th of a degree). A bar graph showing the last 10 readings is also provided, where the graph range is restricted: from freezing to boiling. In addition, controls exist to increment and decrement the thermometer's temperature by 1 or 5 degrees (in the current mode).

Design Overview

From the start the design incorporated the Model-View-Controller (MVC) architectural pattern, with embedded use of the Observer pattern to connect the model elements (the thermometer) to the various views displaying thermometer information (the temperature as text and the bar chart showing the last few readings). However, the initial design used anonymous classes for listeners in the GUI, and this made the separation into View and Controller elements hard to see. What is more, all the classes were originally in the Java *default* package, so the separation into distinct subsystems was not clearly visible. The final design has only the main driver class (Temperature) in the *default* package, with the other classes placed in the *model*, *view*, or *controller* packages, representing the subsystems in the design. This is reflected in the subsystem diagram that follows, and resulted in a cleaner partition of concerns into distinct, interacting subsystems.

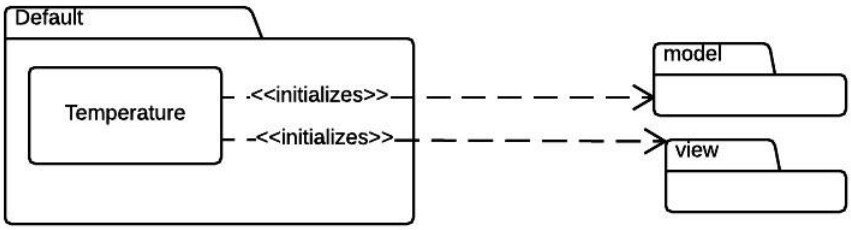
Subsystem Structure



Subsystems

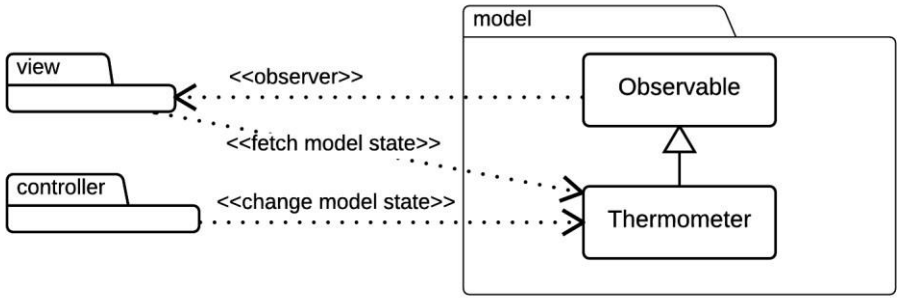
Default Subsystem

Class Temperature	
Responsibilities	Create the model object(s). Create the graphic user interface. Display the GUI so that the application can be used.
Collaborators (uses)	**model.Thermometer** – the primary model class. j**avax.swing.JFrame** - the outermost window for the application. **java.awt.BorderLayout** - layout manager for the main window. **view.ModeChangePanel** - panel supporting temperature mode change. **view.TemperatureChangePanel** - panel supporting changes to temperature. **view.TextViewPanel** - panel used to see the temperature as text. **view.BarGraphCanvas** - canvas showing the last few temperatures



Model Subsystem

Class Thermometer	
Responsibilities	Notify observers of any changes (with the type of change in the optional argument). Set and get the current mode (Celsius or Fahrenheit) Set the temperature to a value based on the current mode. Return the temperature in the current mode. Return the temperature as Celsius or Fahrenheit) Display the GUI so that the application can be used.
Collaborators (inherits)	j**ava.util.Observable** - so temperature changes can be observed by others.
Collaborators (uses)	**java.util.Observer -** for notifications of temperature changes to views.



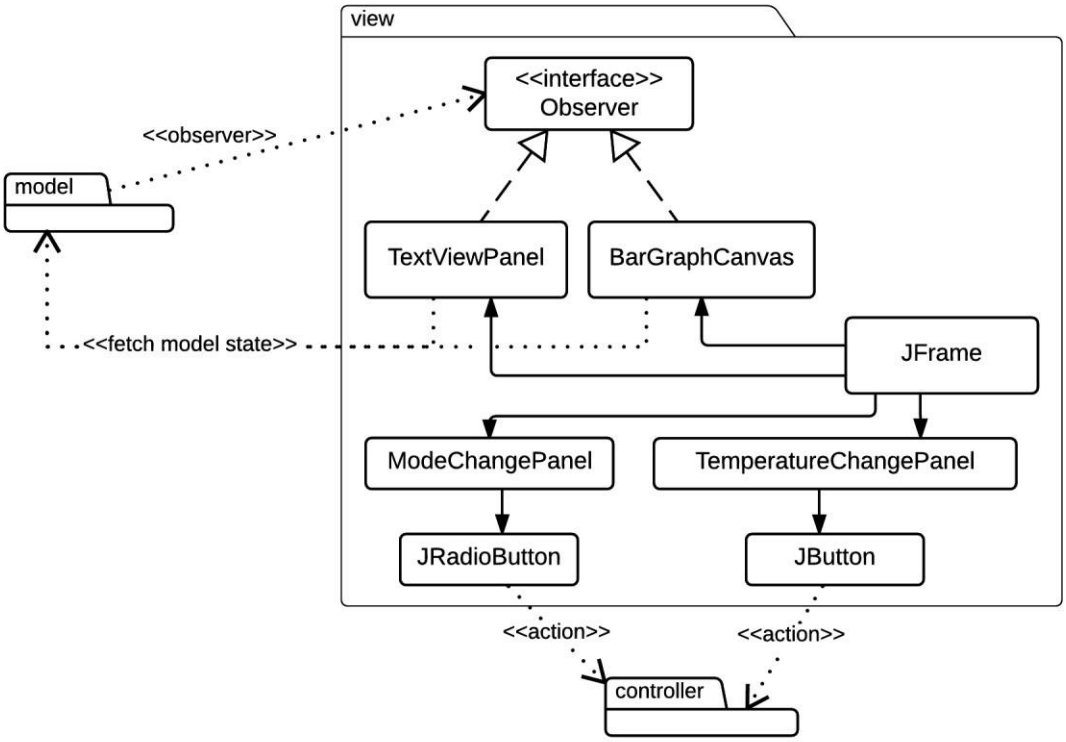
View Subsystem

Class ModeChangePanel	
Responsibilities	Set up mode change radio buttons in a button group. Associated each radio button with an appropriate mode change listener (a controller). Place the radio buttons in this panel.
Collaborators (inherits)	j**ava.swing.JPanel** - holds the mode change radio buttons.
Collaborators (uses)	**java.awt.GridLayout** - layout control for the buttons. **java.swing.JRadioButton** - one button per supported mode. **java.swing.ButtonGroup** - group buttons so exactly one is active. **model.Thermometer** - the thermometer whose mode is affected. **controller.ModeChangeListener** - activated to change thermometer modes.

Class TemperatureChangePanel	
Responsibilities	Setup the temperature change buttons. Associate each button with an appropriate temperature change listener (a controller). Place buttons in this panel.
Collaborators (inherits)	j**ava.swing.JPanel** - holds the temperature change buttons.
Collaborators (uses)	**java.awt.GridLayout** - layout control for the buttons. **java.swing.JButton** - one button per supported change amount. **model.Thermometer** - the thermometer whose temperature is affected. **controller.TemperatureChangeListener** - activated to change temperature.

Class TextViewPanel	
Responsibilities	Display the current temperature. Respond (as Observer) to updates from the thermometer, either to change the temperature or update the value to reflect a mode change.
Collaborators (implements)	j**ava.swing.JPanel** - holds the label displaying the temperature. **java.util.Observer** - to observe the thermometer.
Collaborators (uses)	**java.util.Observable -** to register for thermometer change notifications. **java.awt.GridLayout** - layout control for the labels. **java.swing.JLabel** - labeling and holding the temperature value. **java.text.DecimalFormat** - to format the temperature as a string. **model.Thermometer** - the thermometer whose temperature is affected.

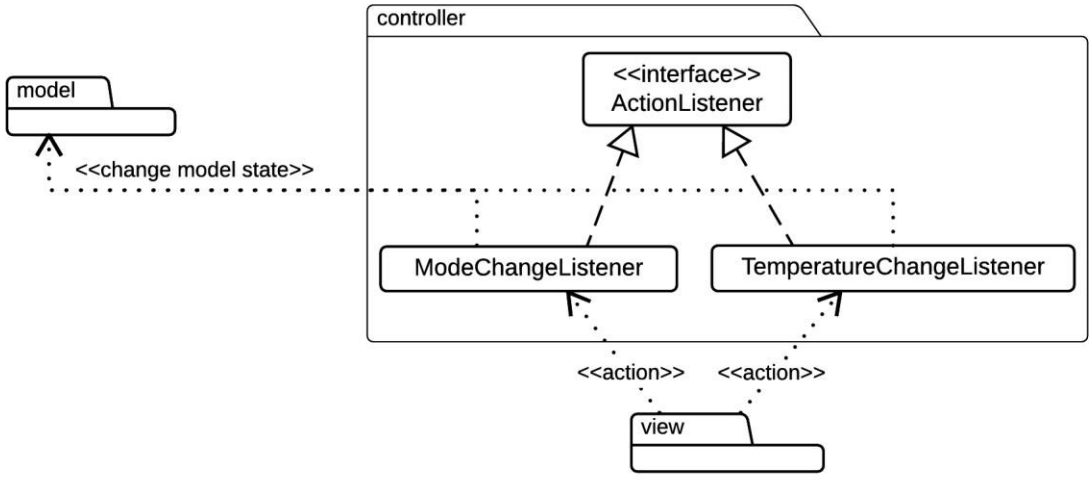
Class BarGraphCanvas	
Responsibilities	Display the past few temperature changes as a bar graph. Respond (as Observer) to updates from the thermometer to change the temperature.
Collaborators (implements)	j**ava.awt.Canvas** - display the bar graph of recent thermometer values. **java.util.Observer** - to observe the thermometer.
Collaborators (uses)	**java.util.Observable -** to register for thermometer change notifications. **java.awt.Color** - control the color of the bar graph. **java.awt.Graphics** - control draw the overall graph. **java.awt.Rectangle** - scale the graph to the actual canvas size. **model.Thermometer** - the thermometer whose temperature is affected.



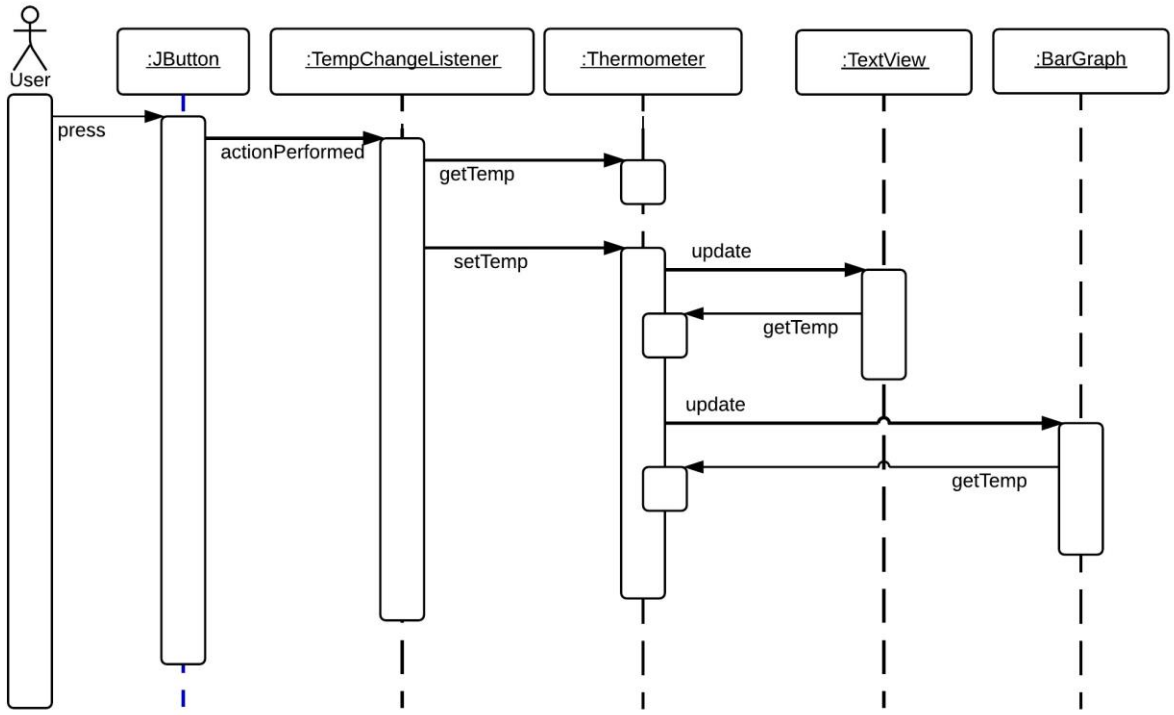
Controller Subsystem

Class ModeChangeListener	
Responsibilities	Record the mode of the change. On activation, change the thermometer mode to that being held.
Collaborators (inherits)	j**ava.swing.ActionListener** - interface used by GUI buttons to activate the listener.
Collaborators (uses)	**java.awt.ActionEvent** - event generated by pressing a button. **model.Thermometer** - the thermometer whose mode is affected.

Class TemperatureChangeListener	
Responsibilities	Record a specific temperature change amount. On activation, change the thermometer temperature by the given amount.
Collaborators (inherits)	j**ava.swing.ActionListener** - interface used by GUI buttons to activate the listener.
Collaborators (uses)	**java.awt.ActionEvent** - event generated by pressing a button. **model.Thermometer** - the thermometer whose mode is affected.



Sequence Diagrams



Sequence 1 - Change Temperature

Pattern Usage

Pattern #1 Observer

The various display entities in the GUI observe the thermometer and, when the thermometer changes, update the information displayed.

Observer Pattern	
Observers	BarGraphCanvas TextViewPanel
Observable	Thermometer

Pattern #2 MVC

The application is organized using the model view controller pattern to (a) recognize commands to the controllers, manipulate the thermometer as the model, and reflect changes via the views.

MVC Pattern	
Model	Thermometer
Views	BarGraphCanvas TextViewPanel
Controllers	ModeChangeListener TemperatureChangeListener