

```

package consProp;

import java.beans.*;

public class Client {
    Boiler itsBoiler = new Boiler();

    public Client() {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        Client client = new Client();
        double temperature;
        try {
            temperature = 55.0D;
            System.out.println("Attempting to set temperature to: "+
                temperature);
            client.itsBoiler.setTemperature(temperature);
            System.out.println("itsBoiler temperature changed to: "+
                client.itsBoiler.getTemperature());
        }
        catch(PropertyVetoException e) {
        }

        System.out.println("itsBoiler temperature is: "+
            client.itsBoiler.getTemperature());
        System.out.println("----- ");

        try {
            temperature = 150.0D;
            System.out.println("Attempting to set temperature to: "+
                temperature);
            client.itsBoiler.setTemperature(temperature);
            System.out.println("itsBoiler temperature changed to: "+
                client.itsBoiler.getTemperature());
        }
        catch(PropertyVetoException e) {
        }

        System.out.println("itsBoiler temperature is: "+
            client.itsBoiler.getTemperature());
        System.out.println("----- ");

        try {
            temperature = 99.0D;

```

```

        System.out.println("Attempting to set temperature to: "+
            temperature);
        client.itsBoiler.setTemperature(temperature);
        System.out.println("itsBoiler temperature changed to: "+
            client.itsBoiler.getTemperature());
    }
    catch(PropertyVetoException e) {
    }

    System.out.println("itsBoiler temperature is: "+
        client.itsBoiler.getTemperature());
    System.out.println("----- ");
}

private void jbInit() throws Exception {
    itsBoiler.addVetoableChangeListener(new java.beans.VetoableChangeListener() {

        public void vetoableChange(PropertyChangeEvent e) throws PropertyVetoException {
            itsBoiler_vetoableChange(e);
        }
    });
}

void itsBoiler_vetoableChange(PropertyChangeEvent e) throws PropertyVetoException {
    Double newTemperature = (Double)e.getNewValue();
    if(newTemperature.doubleValue() > 100.0 ) {
        throw new PropertyVetoException("Veto Exception", e);
    }
}

}

/* --- output ---
Attempting to set temperature to: 55.0
itsBoiler temperature changed to: 55.0
itsBoiler temperature is: 55.0
-----
Attempting to set temperature to: 150.0
itsBoiler temperature is: 55.0
-----
Attempting to set temperature to: 99.0
itsBoiler temperature changed to: 99.0
itsBoiler temperature is: 99.0
-----
--- end output --- */

```

```
//Title:      Constrained Property Test
//Version:
//Copyright:  Copyright (c) 2000
//Author:     Furrukh Khan
//Company:    OSU
//Description: Your description
```

```
package consProp;
```

```
import java.io.*;
import java.beans.*;
```

```
public class Boiler implements Serializable {
```

```
    public Boiler() {
    }
```

```
    void readObject(ObjectInputStream ois) throws ClassNotFoundException, IOException {
        ois.defaultReadObject();
    }
```

```
    void writeObject(ObjectOutputStream oos) throws IOException {
        oos.defaultWriteObject();
    }
```

```
    public void setTemperature(double newTemperature) throws java.beans.PropertyVetoException {
        double oldTemperature = temperature;
        vetoableChangeListeners.fireVetoableChange("temperature", new Double(oldTemperature), new Double(newTemperature));
        temperature = newTemperature;
        propertyChangeListeners.firePropertyChange("temperature", new Double(oldTemperature), new Double(newTemperature));
    }
```

```
    public double getTemperature() {
        return temperature;
    }
```

```
    public synchronized void removePropertyChangeListener(PropertyChangeListener l) {
        propertyChangeListeners.removePropertyChangeListener(l);
    }
```

```
    public synchronized void addPropertyChangeListener(PropertyChangeListener l) {
        propertyChangeListeners.addPropertyChangeListener(l);
    }
```

```
    public synchronized void removeVetoableChangeListener(VetoableChangeListener l) {
        vetoableChangeListeners.removeVetoableChangeListener(l);
    }
```

```
    public synchronized void addVetoableChangeListener(VetoableChangeListener l) {
        vetoableChangeListeners.addVetoableChangeListener(l);
    }
```

```
    private double temperature;
```

```
    private transient PropertyChangeSupport propertyChangeListeners = new PropertyChangeSupport(this);
```

```
    private transient VetoableChangeSupport vetoableChangeListeners = new VetoableChangeSupport(this);
```

```
}
```