

```
//Copyright:   Copyright (c) 2000
//Author:     Furrukh Khan
//Company:    OSU
```

```
package beantest4;
```

```
import java.util.*;
```

```
public class PressureEvent extends EventObject {
```

```
    public PressureEvent(Object source) {
        super(source);
    }
```

```
    private double pressure;
```

```
    public double getPressure() {
        return pressure;
    }
```

```
    public void setPressure(double newPressure) {
        pressure = newPressure;
    }
```

```
}
```

```
//Copyright:    Copyright (c) 2000
//Author:      Furrukh Khan
//Company:     OSU
```

```
package beantest4;

import java.util.*;

public interface PressureListener extends EventListener {

    public void pressureChanged(PressureEvent e);
}
```

```
//Copyright:    Copyright (c) 2000
//Author:      Furrukh Khan
//Company:     OSU
```

```
package beantest4;

import java.io.*;
import java.util.*;

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 synchronized void removePressureListener(PressureListener l) {
        if(pressureListeners != null && pressureListeners.contains(l)) {
            Vector v = (Vector) pressureListeners.clone();
            v.removeElement(l);
            pressureListeners = v;
        }
    }

    public synchronized void addPressureListener(PressureListener l) {
        Vector v = pressureListeners == null ? new Vector(2) : (Vector) pressureListeners.clone();
        if(!v.contains(l)) {
            v.addElement(l);
            pressureListeners = v;
        }
    }

    private transient Vector pressureListeners;
    private double threshold;

    protected void firePressureChanged(PressureEvent e) {
        if(pressureListeners != null) {
            Vector listeners = pressureListeners;
            int count = listeners.size();
            for (int i = 0; i < count; i++) {
                ((PressureListener) listeners.elementAt(i)).pressureChanged(e);
            }
        }
    }

    public void run() {
        double pressure;
        for(int i = 0; i < 100; ++i) {
            pressure = (double)i;
            if( pressure > getThreshold() ) {
                PressureEvent ev = new PressureEvent(this);
                ev.setPressure(pressure);
                firePressureChanged(ev);
            }
        }
    }

    public void setThreshold(double newThreshold) {
        threshold = newThreshold;
    }

    public double getThreshold() {
        return threshold;
    }
}
```

```
//Copyright:    Copyright (c) 2000
//Author:      Furrukh Khan
//Company:     OSU
```

```
package beantest4;

public class Client {
    boiler mainBoiler = new boiler();

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

    public static void main(String[] args) {
        Client client = new Client();
        client.mainBoiler.run();
    }

    private void jbInit() throws Exception {
        mainBoiler.setThreshold(80.0);
        mainBoiler.addPressureListener(new beantest4.PressureListener() {

            public void pressureChanged(PressureEvent e) {
                mainBoiler_pressureChanged(e);
            }
        });
    }

    void mainBoiler_pressureChanged(PressureEvent e) {
        System.out.println("Pressure: "+e.getPressure());
    }
}
```