```
//KLASA GŁÓWNA

public class HeaterMain {

	public static void main(String[] args)
	{
		Heater heater=new Heater();
		HeaterView heaterView=new HeaterView();
		Supervisor supervisor=new Supervisor();
		heater.addPropertyChangeListener(heaterView);
		heater.addVetoableChangeListener(supervisor);
		HeaterGUI heaterGUI=new HeaterGUI(heater);
	}
}

//*************************************************************************

//KLASA ZIARNA Z WŁAŚCIWOŚCIĄ OGRANICZONĄ

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

public class Heater implements Serializable
{
	private int temperature;

	private PropertyChangeSupport propertyChange=new PropertyChangeSupport(this);

	private VetoableChangeSupport vetoableChange=new VetoableChangeSupport(this);

	public Heater()
	{
		temperature=25;
	}

	public synchronized void setTemperature(int temp) throws PropertyVetoException
	{
		int oldTemp=temperature;

		vetoableChange.fireVetoableChange("temperature", new Integer(oldTemp), new
		   Integer(temp));

		temperature=temp;

		propertyChange.firePropertyChange("temperature", new Integer(oldTemp), new
		  Integer(temperature));
	}

	public synchronized int getTemperature()
	{
		return temperature;
	}

	public synchronized void increment()
	{
		try
		{
			setTemperature(getTemperature()+1);
		}
		catch (PropertyVetoException e)
		{}
	}

	public synchronized void decrement()
```

```
	{
		try
		{
			setTemperature(getTemperature()-1);
		}
		catch (PropertyVetoException e)
		{}
	}

	public synchronized void addPropertyChangeListener(PropertyChangeListener listener)
	{
		propertyChange.addPropertyChangeListener(listener);
	}

	public synchronized void addVetoableChangeListener(VetoableChangeListener listener)
	{
		vetoableChange.addVetoableChangeListener(listener);
	}
}

//*************************************************************************

//KLASA ZIARNA WETUJĄCEGO ZMIANY

import java.beans.PropertyChangeEvent;
import java.beans.PropertyVetoException;
import java.beans.VetoableChangeListener;
import java.io.Serializable;

public class Supervisor implements Serializable, VetoableChangeListener
{
	public Supervisor()
	{
	}

	public void vetoableChange(PropertyChangeEvent e) throws PropertyVetoException
	{
		int t=((Integer)e.getNewValue()).intValue();

		if(t>30)
		{
			throw new PropertyVetoException("Za wysoka temperatura", e);
		}
	}
}
```