EXEMPLE MODÈLE MVC

Cet exemple introduit le modèle MVC. Dans cet exemple, le modèle représente la valeur d'un volume quelconque stocké, tandis que les vues proposent l'interface utilisateur pour manipuler ce modèle et le contrôle gère les interactions entre les vues et le modèle. Quatre types de vues sont proposés, dont deux dites « actives », qui permettent à l'utilisateur de modifier la valeur du volume, et deux « passives », qui ne le permettent pas. Les deux premières utilisent les composants Swing *JSpinner* et *JSlider*, tandis que les deux dernières utilisent les composants *JList* et *JProgressBar*.

Par ailleurs, cet exemple est basé sur celui proposé par Baptiste Wicht ("*Implémentation du pattern MVC*"), sur le site http://baptiste-wicht.developpez.com/tutoriel/conception/mvc/ (dernière visite 29 Janvier 2009).

Table des matières

Le modèle : Class VolumeModel

```
/*
UNIVERSITE PARIS 1 (PANTHEON SORBONNE)
MIAGE - UFR 27
Cours ISI5 / INF2 - Developpement d'Interface
*/
package ihmexamples.volumeapplication;

import javax.swing.event.EventListenerList;

/**
 * Example d'usage du modele MVC
 * Cette classe definit le modele de donnees indicant un volume.
 *
 * Example base sur Baptiste Wicht, <b>"Implémentation du pattern MVC"</b>,
 * 24 Avril 2007 (derniere visite le 29 Janvier 2009).
 * @see http://baptiste-wicht.developpez.com/tutoriel/conception/mvc/
 * @author kirsch
 */
public class VolumeModel {

    public VolumeModel() {
```

```
        this(0);
    }

    public VolumeModel(int volume) {
        this.volume = volume;
        this.listeners = new EventListenerList();
    }

    /**
     * Cette methode permet de recuperer l'etat du modele, a savoir la valeur
     * du volume stocke.
     * @return int volume
     */
    public int getVolume() {
        return volume;
    }

    /**
     * Cette methode permet de modifier l'etat du modele, a savoir la valeur
     * du volume stocke.
     * @param volume nouvelle valeur de volume stocke
     */
    public void setVolume(int volume) {
        this.volume = volume;
        //on modifie l'etat du modele, on averti les listeners
        fireVolumeChanged();
    }

 /**
     * Methode utilisee pour abonner un nouveau listener a ce modele
     * @param listener nouveau listener
     */
    public void addVolumeListener(VolumeModelListener listener) {
        listeners.add(VolumeModelListener.class, listener);
    }

    /**
     * Methode utilisee pour desabonner un listener de ce modele
     * @param l listener qui veut se desabonner
     */
    public void removeVolumeListener(VolumeModelListener l) {
        listeners.remove(VolumeModelListener.class, l);
    }

    /**
     * methode declanchant la notification des tous les listeners suite a une
     * modification dans l'etat du modele.
     */
    protected void fireVolumeChanged() {
        VolumeModelListener[] listenerList =
                (VolumeModelListener[]) listeners.getListeners(VolumeModelListener.class);
        //foreach listener, on appelle la methode volumeChanged
        for (VolumeModelListener listener : listenerList) {
            listener.volumeChanged(new VolumeChangedEvent(this, this.getVolume()));
        }
    }

    //volume
    private int volume;
    //listeners
    private EventListenerList listeners;
}
```

Les événements : Classe VolumeChangedEvent

```
/*
UNIVERSITE PARIS 1 (PANTHEON SORBONNE)
MIAGE - UFR 27
Cours ISI5 / INF2 - Developpement d'Interface
*/

package ihmexamples.volumeapplication;

/**
 * Example d'usage du modele MVC
 * Cette classe definit un evennement de modification de l'etat du modele.
 * Un objet VolumeChangedEvent indique que le modele (indicant un volume)
 * a ete modifie.
 *
 * Example base sur Baptiste Wicht, <b>"Implémentation du pattern MVC"</b>,
 * 24 Avril 2007 (derniere visite le 29 Janvier 2009).
 * @see http://baptiste-wicht.developpez.com/tutoriel/conception/mvc/
 * @author kirsch
 */
class VolumeChangedEvent extends java.util.EventObject {

    public VolumeChangedEvent(Object source, int newVolume) {
        super(source);
        this.newVolume = newVolume;
    }

    public int getNewVolume() {
        return newVolume;
    }

    private int newVolume;
}
```

Les écouteurs : Interface VolumeModelListener

```
/*
 UNIVERSITE PARIS 1 (PANTHEON SORBONNE)
 MIAGE - UFR 27
 Cours ISI5 / INF2 - Developpement d'Interface
 */

package ihmexamples.volumeapplication;

/**
 * Example d'usage du modele MVC
 * Cette classe definit un listenener pour le modele de donnees.
 * A chaque modification sur le modele, un evennement du type VolumeChangedEvent
 * sera envoye a tous les listeners du modele.
 *
 * Example base sur Baptiste Wicht, <b>"Implémentation du pattern MVC"</b>,
 * 24 Avril 2007 (derniere visite le 29 Janvier 2009).
 * @see http://baptiste-wicht.developpez.com/tutoriel/conception/mvc/
 * @author kirsch
 */
public interface VolumeModelListener extends java.util.EventListener {
    public void volumeChanged(VolumeChangedEvent event);
}
```

Le contrôle : Class VolumeController

```
/*
 UNIVERSITE PARIS 1 (PANTHEON SORBONNE)
 MIAGE - UFR 27
 Cours ISI5 / INF2 - Developpement d'Interface
 */

package ihmexamples.volumeapplication;

import java.util.ArrayList;
import java.util.List;

/**
 * Example d'usage du modele MVC
 * Cette classe definit le controler pour modele de volume.
 *
 * Example base sur Baptiste Wicht, <b>"Implémentation du pattern MVC"</b>,
 * 24 Avril 2007 (derniere visite le 29 Janvier 2009).
 * @see http://baptiste-wicht.developpez.com/tutoriel/conception/mvc/
 * @author kirsch
 */
public class VolumeController {

    public VolumeController(VolumeModel model) {
        this.model = model;
        this.views = new ArrayList<VolumeView>();
    }

    public VolumeModel getModel() {
        return model;
    }

    public void setModel(VolumeModel model) {
        this.model = model;
    }

    /**
     * ce methode est utilise par les vue pour informer lors des
     * demandes (de l'utilisateur) de modification du volume.
     * @param newVolume nouveau valeur de volume
     */
    public void notifyVolumeChange(int newVolume) {
        this.model.setVolume(newVolume);
    }

    public void addView (VolumeView view) {
        this.views.add(view);
        this.model.addVolumeListener(view);
    }

    public void removeView (VolumeView view) {
        this.views.remove(view);
        this.model.removeVolumeListener(view);
     }

    public void displayViews() {
        for (VolumeView aView: this.views) {
            aView.display();
        }
     }

    public void closeViews() {
        for (VolumeView aView: this.views) {
            aView.close();
        }
    }

    private VolumeModel model;
    private List<VolumeView> views;
}
```

Les vues : Abstract Class VolumeView

```
/*
 UNIVERSITE PARIS 1 (PANTHEON SORBONNE)
 MIAGE - UFR 27
 Cours ISI5 / INF2 - Developpement d'Interface
*/

package ihmexamples.volumeapplication;

import java.awt.Dimension;

/**
 * Example d'usage du modele MVC
 * Cette classe definit le view pour modele de volume.
 *
 * Example base sur Baptiste Wicht, <b>"Implémentation du pattern MVC"</b>,
 * 24 Avril 2007 (derniere visite le 29 Janvier 2009).
 * @see http://baptiste-wicht.developpez.com/tutoriel/conception/mvc/
 * @author kirsch
 */
public abstract class VolumeView implements VolumeModelListener {

    public VolumeView(VolumeController controller) {
        this.controller = controller;
        this.controller.addView(this);
    }

    public final VolumeController getController(){
               return controller;
       }


    public abstract void display();
    public abstract void close();
    public abstract void setLocation(int x, int y);
    public abstract void setSize(Dimension d);
    public abstract Dimension getSize();

    private VolumeController controller = null;
}
```

Les vues : Class VolumeListView

```
/*
 UNIVERSITE PARIS 1 (PANTHEON SORBONNE)
 MIAGE - UFR 27
 Cours ISI5 / INF2 - Developpement d'Interface
*/

package ihmexamples.volumeapplication;

import java.awt.BorderLayout;
import java.awt.Dimension;
import javax.swing.DefaultListModel;
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JScrollPane;
import javax.swing.ListModel;

/**
 * Example d'usage du modele MVC
 * Cette classe definit une vue basé sur une liste de valeurs
 * pour un modele de volume
 *
```

```
 * Example base sur Baptiste Wicht, <b>"Implémentation du pattern MVC"</b>,
 * 24 Avril 2007 (derniere visite le 29 Janvier 2009).
 * @see http://baptiste-wicht.developpez.com/tutoriel/conception/mvc/
 * @author kirsch
 */
public class VolumeListView extends VolumeView {

    public VolumeListView(VolumeController controller, int volume) {
        super(controller);

        this.buildFrame(volume);
    }

    public VolumeListView(VolumeController controller) {
        this(controller, 0);
    }

    private void buildFrame(int volume) {
        this.jFrame = new JFrame();
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

         /* on definit le layoutmanager avec 2x plus d'espace vert. et horiz. */
        BorderLayout border = new BorderLayout();
        border.setHgap(border.getHgap() * 2);
        border.setVgap(border.getVgap() * 2);
        jFrame.setLayout(border);

        jListModel = new DefaultListModel();
              jListModel.addElement(volume);

        jVolumeList = new JList (jListModel);

        JScrollPane scrollPane = new JScrollPane (jVolumeList);
        jFrame.add(scrollPane, BorderLayout.CENTER);

        jFrame.pack();
    }

    @Override
    public void display() {
        this.jFrame.setVisible(true);
    }

    @Override
    public void close() {
        this.jFrame.setVisible(false);
    }

    public void volumeChanged(VolumeChangedEvent event) {
              jListModel.addElement(event.getNewVolume());
    }

    @Override
    public void setLocation(int x, int y) {
        this.jFrame.setLocation(x, y);
    }

    @Override
    public void setSize(Dimension d) {
        this.jFrame.setSize(d);
    }

    @Override
    public Dimension getSize() {
        return this.jFrame.getSize();
    }

    JFrame jFrame;
    JList jVolumeList;
    DefaultListModel jListModel;
}
```

Les vues : Class VolumeProgressView

```
/*
 UNIVERSITE PARIS 1 (PANTHEON SORBONNE)
 MIAGE - UFR 27
 Cours ISI5 / INF2 - Developpement d'Interface
*/

package ihmexamples.volumeapplication;

import java.awt.Dimension;
import java.awt.FlowLayout;
import javax.swing.JFrame;
import javax.swing.JProgressBar;
import javax.swing.SwingConstants;

/**
 * Example d'usage du modele MVC
 * Cette classe definit une vue basé sur une barre de progression qui
 * permet de visualiser la valeur du modele de volume
 *
 * @author kirsch
 */
public class VolumeProgressView extends VolumeView {
    private JFrame jFrame;
    private JProgressBar jProgress;


    public VolumeProgressView(VolumeController controller) {
        this(controller, 0);
    }

    public VolumeProgressView(VolumeController controller, int volume) {
        super(controller);
        this.buildFrame(volume);
    }


    private void buildFrame(int volume) {
        this.jFrame = new JFrame();
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        jFrame.setLayout(new FlowLayout());

        jProgress = new JProgressBar (0, 100);
        jProgress.setOrientation(SwingConstants.VERTICAL);
        jProgress.setValue(volume);

        jFrame.add(jProgress);

        jFrame.pack();
    }

    @Override
    public void display() {
        this.jFrame.setVisible(true);
    }

    @Override
    public void close() {
        this.jFrame.setVisible(false);
    }

    @Override
    public void setLocation(int x, int y) {
        this.jFrame.setLocation(x, y);
    }
```

```
    @Override
    public void setSize(Dimension d) {
        this.jFrame.setSize(d);
    }

    @Override
    public Dimension getSize() {
        return this.jFrame.getSize();
    }


    public void volumeChanged(VolumeChangedEvent event) {
        this.jProgress.setValue(event.getNewVolume());
    }

}
```

Les vues : Class VolumeSliderView

```
/*
UNIVERSITE PARIS 1 (PANTHEON SORBONNE)
MIAGE - UFR 27
Cours ISI5 / INF2 - Developpement d'Interface
*/
package ihmexamples.volumeapplication;

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JSlider;
import javax.swing.SwingConstants;

/**
 * Example d'usage du modele MVC
 * Cette classe definit une vue basé sur un "Slider", une "barre" deroulante qui
 * pour visualiser et modifier la valeur du modele de volume
 *
 * @author kirsch
 */
public class VolumeSliderView extends VolumeView {

    JFrame jFrame;
    JSlider jSlider;

    public VolumeSliderView(VolumeController controller) {
        this(controller, 0);
    }

    public VolumeSliderView(VolumeController controller, int volume) {
        super(controller);
        this.buildFrame(volume);
    }

    private void buildFrame(int volume) {
        this.jFrame = new JFrame();
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        /* on definit le layoutmanager avec 2x plus d'espace vert. et horiz. */
        BorderLayout border = new BorderLayout();
        border.setHgap(border.getHgap() * 2);
        border.setVgap(border.getVgap() * 2);
        jFrame.setLayout(border);

        //JSlider(int min, int max, int value)
        this.jSlider = new JSlider(0, 100, volume);
```

```
        this.jSlider.setOrientation(SwingConstants.HORIZONTAL);

        JButton jButton = new JButton("Change");
        jButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                notifyVolumeChange();
            }
        });

        jFrame.add(jSlider, BorderLayout.CENTER);
        jFrame.add(jButton, BorderLayout.SOUTH);

        jFrame.pack();
    }

    private void notifyVolumeChange() {
        this.getController().notifyVolumeChange(jSlider.getValue());
    }

    public void volumeChanged(VolumeChangedEvent event) {
        this.jSlider.setValue(event.getNewVolume());
    }

    @Override
    public void display() {
        this.jFrame.setVisible(true);
    }

    @Override
    public void close() {
        this.jFrame.setVisible(false);
    }

    @Override
    public void setLocation(int x, int y) {
        this.jFrame.setLocation(x, y);
    }

    @Override
    public void setSize(Dimension d) {
        this.jFrame.setSize(d);
    }

    @Override
    public Dimension getSize() {
        return this.jFrame.getSize();
    }
}
```

Les vues : Class VolumeSpinnerView

```
/*
UNIVERSITE PARIS 1 (PANTHEON SORBONNE)
MIAGE - UFR 27
Cours ISI5 / INF2 - Developpement d'Interface
*/
package ihmexamples.volumeapplication;

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JSpinner;
import javax.swing.SpinnerNumberModel;
```

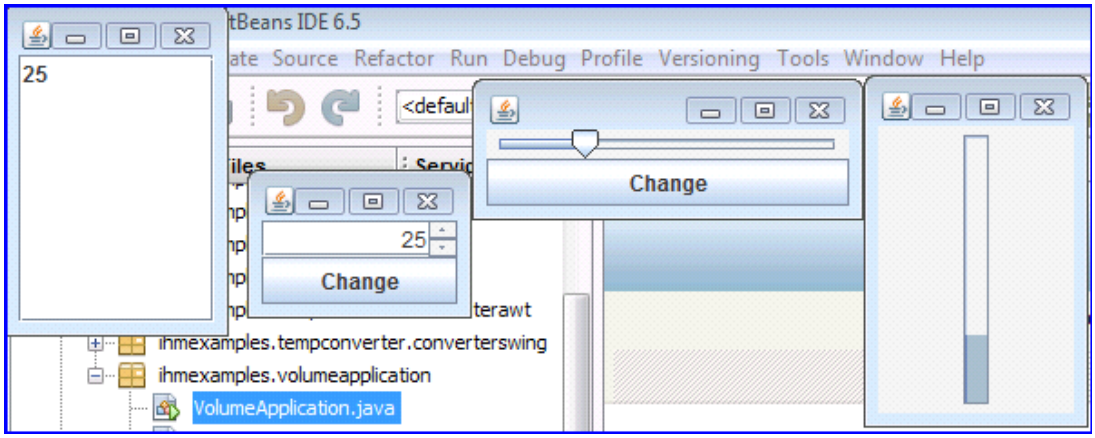
```
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;

/**
 * Example d'usage du modele MVC
 * Cette classe definit une vue basé sur un "Spinner" (textfield contenant
 * des boutons pour augmenter/reduire la valeur) pour visualiser et
 * modifier la valeur du modele de volume
 *
 * Example base sur Baptiste Wicht, <b>"Implémentation du pattern MVC"</b>,
 * 24 Avril 2007 (derniere visite le 29 Janvier 2009).
 * @see http://baptiste-wicht.developpez.com/tutoriel/conception/mvc/
 * @author kirsch
 */
public class VolumeSpinnerView extends VolumeView {

    private JFrame jFrame;
    private SpinnerNumberModel spinnerModel;
    private JSpinner jSpinner;

    public VolumeSpinnerView(VolumeController controller) {
        this(controller, 0);
    }

    public VolumeSpinnerView(VolumeController controller, int volume) {
        super(controller);
        this.buildFrame(volume);
    }

    private void buildFrame(int volume) {
        this.jFrame = new JFrame();
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        /* on definit le layoutmanager avec 2x plus d'espace vert. et horiz. */
        BorderLayout border = new BorderLayout();
        border.setHgap(border.getHgap() * 2);
        border.setVgap(border.getVgap() * 2);
        jFrame.setLayout(border);

        //SpinnerNumberModel (val, min, max, step);
        spinnerModel = new SpinnerNumberModel(volume, 0, 100, 1);
        jSpinner = new JSpinner(spinnerModel);

        JButton jButton = new JButton("Change");
        jButton.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {
                // notifyVolumeChange(e); //solution alternative
                notifyVolumeChange();
            }
        });

        jFrame.add(jSpinner, BorderLayout.CENTER);
        jFrame.add(jButton, BorderLayout.SOUTH);

        jFrame.pack();
    }

    // solution alternative
    /* private void notifyVolumeChange(ChangeEvent e) {
    JSpinner spinner = (JSpinner)e.getSource();
    int nv = spinner.getModel().getNumber().intValue();
    this.getController().notifyVolumeChange(nv);
    }
     */

    private void notifyVolumeChange() {
        int newVolume = this.spinnerModel.getNumber().intValue();
        this.getController().notifyVolumeChange(newVolume);
    }
```

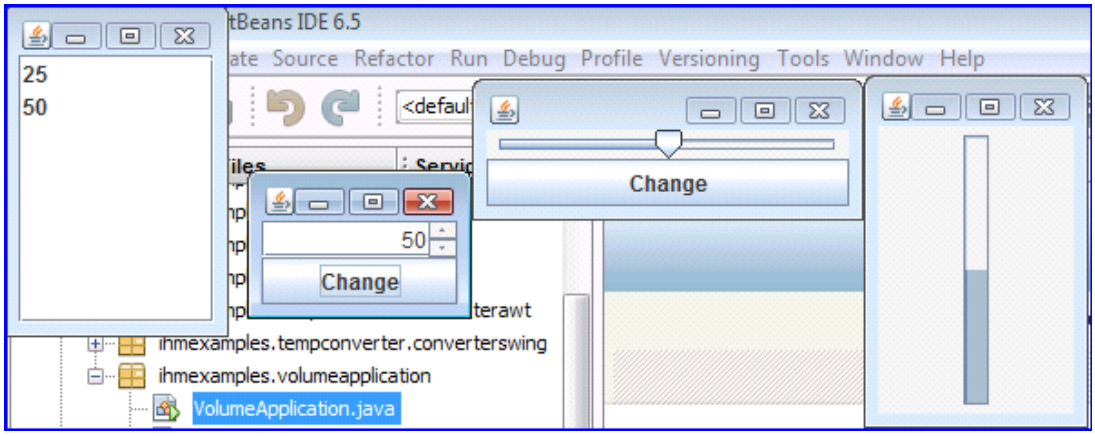
```
    @Override
    public void display() {
        this.jFrame.setVisible(true);
    }

    @Override
    public void close() {
        this.jFrame.setVisible(false);
    }

    public void volumeChanged(VolumeChangedEvent event) {
        spinnerModel.setValue(event.getNewVolume());
    }

    @Override
    public void setLocation(int x, int y) {
        this.jFrame.setLocation(x, y);
    }

    @Override
    public void setSize(Dimension d) {
        this.jFrame.setSize(d);
    }

    @Override
    public Dimension getSize() {
        return this.jFrame.getSize();
    }
}
```

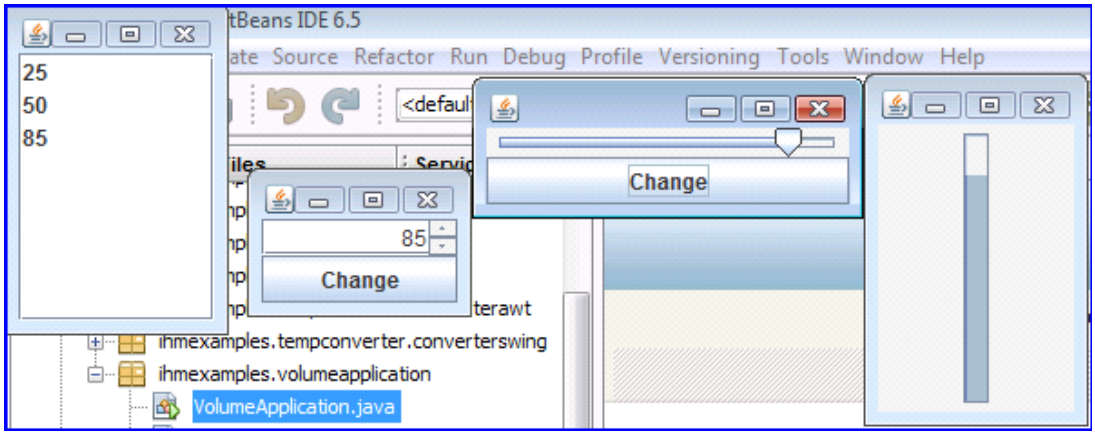
L'application : Class VolumeApplication

```
/*
UNIVERSITE PARIS 1 (PANTHEON SORBONNE)
MIAGE - UFR 27
Cours ISI5 / INF2 - Developpement d'Interface
*/
package ihmexamples.volumeapplication;


public class VolumeApplication {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        VolumeModel model = new VolumeModel(25);

        VolumeController controller1 = new VolumeController(model);

        VolumeListView listView = new VolumeListView(controller1, model.getVolume());
        VolumeSpinnerView spinnerView = new VolumeSpinnerView(controller1, model.getVolume());
        VolumeSliderView sliderView = new VolumeSliderView(controller1, model.getVolume());
        VolumeProgressView progressView = new VolumeProgressView(controller1, model.getVolume());

        spinnerView.setLocation((int) listView.getSize().getWidth() + 10,
                (int) (listView.getSize().getHeight() / 2));
        sliderView.setLocation((int) (listView.getSize().getWidth() +
                spinnerView.getSize().getWidth() + 10),
                (int) (spinnerView.getSize().getHeight() / 2));
        progressView.setLocation ((int) (listView.getSize().getWidth() +
                spinnerView.getSize().getWidth() +
                sliderView.getSize().getWidth() + 10),
                (int) (sliderView.getSize().getHeight() / 2));

        controller1.displayViews();
    }
}
```

Le résultat


25
tBeans IDE 6.5
ate Source Refactor Run Debug Profile Versioning Tools Window Help
Change
25
Change
terawt
ihmexamples.tempconverter.converterswing
ihmexamples.volumeapplication
VolumeApplication.java


25
50
tBeans IDE 6.5
ate Source Refactor Run Debug Profile Versioning Tools Window Help
Change
50
Change
terawt
ihmexamples.tempconverter.converterswing
ihmexamples.volumeapplication
VolumeApplication.java


25
50
85
tBeans IDE 6.5
ate Source Refactor Run Debug Profile Versioning Tools Window Help
Change
85
Change
terawt
ihmexamples.tempconverter.converterswing
ihmexamples.volumeapplication
VolumeApplication.java