MVC-mönstret

model-view-control i Swing

MVC

Mammaproffsen
Mödravårdscentral | Hälsocenter

Tanken bakom MVC (model view control) är att separera uppgifter i ett program från varandra.

Model - Den data som behandlas

View - Hur användargränssnittet ser ut

Control - Vad som ska hända när användaren agerar mot gränssnittet

Model

I vårt korta exempel så är Model en klass som representerar ett personregister.

Man kan söka efter en person och man kan ta bort en person.

En person representeras här för enkelhetens skull bara av en String med namnet.

Publikt gränssnitt mot Model

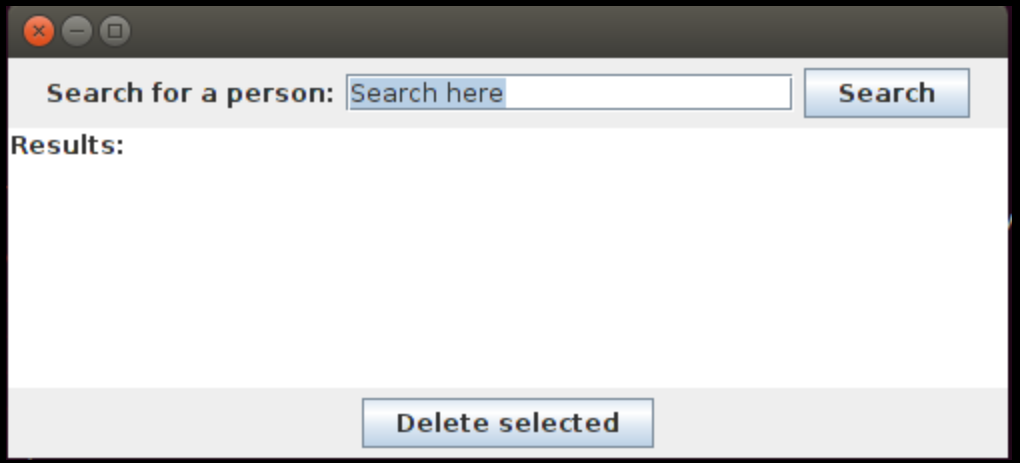
```
public Model(){...} // Skapa modellen

public void delete(String name){...}
// Ta bort en person

public ArrayList<String>
 search(String target){...} // Sök personer
```

View

Vår View är ett fönster med några få komponenter:



Komponenter i View

```
private JFrame frame;
// Komponenter i NORTH
private JPanel north;
private JLabel label;
private JTextField searchField;
private JButton searchButton;
// Komponenter i CENTER
private JList<String> resultList;
private DefaultListModel<String> result;
// Komponenter i SOUTH
private JPanel south;
private JButton deleteButton;
```

Publikt gränssnitt mot View

```
public View(){...} // Skapa en View
public JTextField getSearchField(){...} //Sökfältet
public JButton getSearchButton(){...}   //Sökknappen
public JButton getDeleteButton(){...}   //Deleteknappen
public DefaultListModel<String> getResultModel(){...}
                        //Listmodellen för resultat

public JList<String> getResultList(){...}
                        //Listan med resultat
```

Control

Control registrerar Listeners på komponenter och hanterar vad som ska hända när användaren triggar event på dem

- Ska View uppdateras? Hur?
- Ska Model förändras?
- Control måste alltså ha tillgång till View såväl som Model

“User actions” Control ska hantera

- Användaren vill söka efter person
  - Genom att trycka enter i sökfältet
  - Genom att trycka på sökknappen
- Avändaren vill radera person
  - Först markerar användaren personer som ska bort
  - Sedan trycker användaren på “Delete selected”

Användaren vill söka

Både sökfältet och sökknappen ska ha samma Listener. När eventet kommer, ska Control:

- Läsa sökfältet
- Fråga Model efter resultat genom att anropa seach() med sökfrasen
- Uppdatera resultatlistan med ev. resultat
- Återställa sökfältet (markerad text: sök här)

SearchAction

```
private class SearchAction implements ActionListener{
  public void actionPerformed(ActionEvent ae){
    searchForPerson();
    view.getSearchField().setText("Search here");
    view.getSearchField().requestFocus();
    view.getSearchField().selectAll();
  }
}
private void searchForPerson(){
  view.getResultModel().removeAllElements();
  for(String match :
      model.search(view.getSearchField().getText()) ){
          view.getResultModel().addElement(match);
  }
}
```

Användaren vill radera en person

Delete-knappen måste ha en DeleteAction som action listener.

Denna listener måste fråga listan vilken eller vilka element som är markerade, och radera dessa i Model.

För att ge feedback till användaren, så visas i listan vilka personer som raderats.

DeleteAction

```
private class DeleteAction implements ActionListener{
  public void actionPerformed(ActionEvent ae){
    List<String> values =
      view.getResultList().getSelectedValuesList();
    view.getResultModel().removeAllElements();
    for(String value : values){
      model.delete(value);
      view.getResultModel().addElement("Deleted: "+value);
    }
    view.getSearchField().setText("Search here");
    view.getSearchField().requestFocus();
    view.getSearchField().selectAll();
  }
}
```

Publikt gränssnitt mot Control

```
public Control(View view, Model model){...}//Skapa Control
public void controlIt(){...} // Take control!


//controlIt():
public void controlIt(){
  controlFieldAndButton(); // Register listeners
  controlDeleteButton();   // Register listener
}
```

Publika gränssnitt i hela MVC-appen

```
//Model:
public Model(){...} // Skapa modellen
public void delete(String name){...} // Ta bort en person
public ArrayList<String> search(String target){...} // Sök personer
// View:
public View(){...} // Skapa en View
public JTextField getSearchField(){...}  //Sökfältet
public JButton getSearchButton(){...}    //Sökknappen
public JButton getDeleteButton(){...}    //Deleteknappen
public DefaultListModel<String> getResultModel(){...}//Listmodellen
public JList<String> getResultList(){...} //Listan med resultat
// Control:
public Control(View view, Model model){...}//Skapa Control
public void controlIt(){...} // Take control!
```

Main-klassen

```
public static void main(String[] args){
  SwingUtilities.invokeLater(new Runnable(){
    public void run(){
      Model model = new Model();
      View view = new View();
      Control control = new Control(view, model);
      control.controlIt();
    }
  });
}
// Fyra rader kod (resten är överkurs ;-) )
```

Model.java

```
package mvc.model;
import java.util.ArrayList;
import java.util.Collections;

public class Model{

    private ArrayList<String> persons;
    public Model(){
        initList();
    }

    private void initList(){
        persons = new ArrayList<String>();
        persons.add("Arne Anka");
        persons.add("Sockerconny");
        persons.add("Ragnar Frunk");
        persons.add("Peter Parker");
        persons.add("Agent Carter");
        persons.add("Mandel Karlsson");
        persons.add("Archibald Haddock");
        persons.add("Stan Lee");
        persons.add("Lucky Luke");
        Collections.sort(persons);
    }
```

```
// continued

public void delete(String name){
   persons.remove(name);
 }
 public ArrayList<String> search(String target){
     ArrayList<String> matches = new ArrayList<String>();
     for(String person : persons){

         if(person.toLowerCase().contains(target.toLowerCase())){
             matches.add(person);
         }
     }
     return matches;
 }
}
```

View.java

```
package mvc.view;
import javax.swing.*;
import java.awt.*;

public class View{

    /* Components */
    private JFrame frame;
    private JTextField searchField;
    private JButton searchButton;
    private JButton deleteButton;
    private JList<String> resultList;
    private DefaultListModel<String> result;
    private JPanel north;
    private JPanel south;
    private JLabel label;

    /* Constructor */
    public View(){
        initAndLayout();
    }

    /* Initialize and lay out the components */
    private void initAndLayout(){
        initComponents();
        layoutComponents();
        frame.setSize(500,200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
```

```
private void initComponents(){
     frame = new JFrame();
     frame.setLayout(new BorderLayout());
     north = new JPanel();
     south = new JPanel();
     label = new JLabel("Search for a person:");
     searchField = new JTextField(20);
     searchField.setText("Search here");
     searchField.requestFocus();
     searchField.selectAll();
     searchButton = new JButton("Search");
     deleteButton = new JButton("Delete selected");
     result = new DefaultListModel<String>();
     result.addElement("Results:");
     resultList = new JList<String>(result);
     resultList.setVisibleRowCount(20);
 }

private void layoutComponents(){
     north.add(label);
     north.add(searchField);
     north.add(searchButton);
    south.add(deleteButton);
    frame.add(north, BorderLayout.NORTH);
    frame.add(resultList, BorderLayout.CENTER);
    frame.add(south, BorderLayout.SOUTH);
 }
```

View.java (continued...)

```
    /* Accessor methods for the control */
    public JTextField getSearchField(){
        return searchField;
    }
    public JButton getSearchButton(){
        return searchButton;
    }
    public JButton getDeleteButton(){
        return deleteButton;
    }
    public DefaultListModel<String> getResultModel(){
        return result;
    }
    public JList<String> getResultList(){
        return resultList;
    }
} // end class
```

Control.java

```
package mvc.control;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
import java.util.List;

import mvc.view.View;
import mvc.model.Model;
public class Control{
    private View view;
    private Model model;

    public Control(View view, Model model){
        this.view = view;
        this.model  = model;
    }
```

```
    public void controlIt(){
        controlFieldAndButton();
        controlDeleteButton();
    }


    private class SearchAction implements ActionListener{
        public void actionPerformed(ActionEvent ae){
            searchForPerson();
            view.getSearchField().setText("Search here");
            view.getSearchField().requestFocus();
            view.getSearchField().selectAll();
        }
    }
```

Control.java

```
private class DeleteAction implements ActionListener{
    public void actionPerformed(ActionEvent ae){
        List<String> values = view.getResultList().getSelectedValuesList();

        view.getResultModel().removeAllElements();
        for(String value : values){
            model.delete(value);
            view.getResultModel().addElement("Deleted: "+value);
        }
        view.getSearchField().setText("Search here");
        view.getSearchField().requestFocus();
        view.getSearchField().selectAll();
    }
}
```

Control.java (continuted)

```
    private void controlFieldAndButton(){
        SearchAction sa = new SearchAction();
        view.getSearchField().addActionListener(sa);
        view.getSearchButton().addActionListener(sa);
     }
     private void controlDeleteButton(){
        DeleteAction da = new DeleteAction();
        view.getDeleteButton().addActionListener(da);
     }
     private void searchForPerson(){
        view.getResultModel().removeAllElements();
        for(String match : model.search(view.getSearchField().getText()) ){
            view.getResultModel().addElement(match);
        }
     }
}
```