

Laboration 16

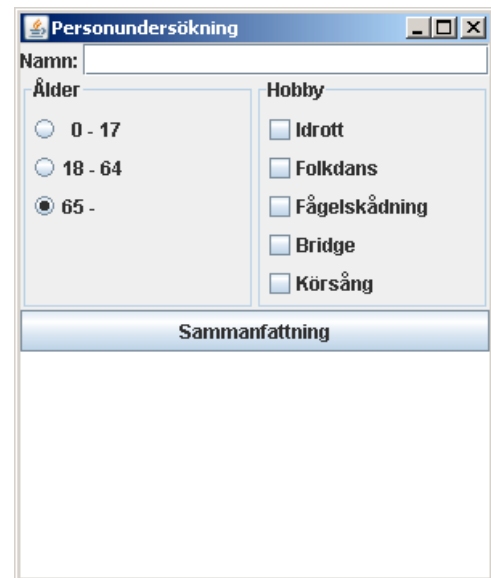
Avsikten med laborationen är att du ska använda ett par nya grafiska komponenter. Dessutom ska du göra designen med hjälp av LayoutManagers. Börja med att skapa paketet *laboration16*.

Grundläggande uppgifter

Uppgift 16a

Uppgiften går ut på att använda ett antal komponenter i ett program. De som används är:

- 3 st *JLabel* (varav två osynliga)
- 1 st *TextField*
- 75 st *JPanel*
- 3 st *JRadioButton*
- 5 st *JCheckBox*
- 1 st *JButton*
- 1 st *JTextArea*.

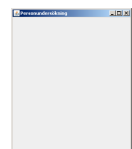


1. Skapa klasserna *SurveyPanel* och *SurveyApp*

Klassen *SurveyPanel* ska ärva *JPanel* och ges storleken 300x360. Sätt layout-managern till *BorderLayout*.

```
setPreferredSize( new Dimension(300,600) );  
setLayout( new BorderLayout() );
```

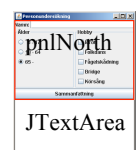
Klassen *SurveyApp* ska ha en main-metod. När main-metoden exekveras ska ett stängbart fönster skapas. I fönstret ska det vara en panel av typen *SurveyPanel*. Titelraden i fönstret ska vara "Personundersökning". Fönstret ska synas vid exekvering.



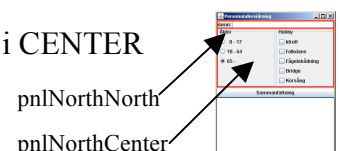
2. Användning av paneler

Det krävs totalt 5 paneler för att bygga *SurveyPanel*. Så här är det tänkt:

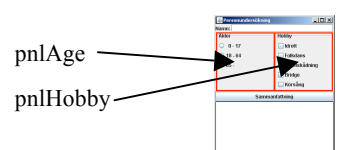
SurveyPanel ska ha en panel i NORTH (*pnlNorth*) och en *JTextArea* i CENTER.



pnlNorth ska innehålla två paneler, en i NORTH (*pnlNorthNorth*) och en i CENTER (*pnlNorthCenter*). I SOUTH ska det vara en *JButton*.



pnlNorthCenter ska delas i två paneler, *pnlAge* och *pnlHobby*.



Lägg till de 5 panelerna som instansvariabler i klassen. Panelerna ska ha följande LayoutManager:

pnlNorth – BorderLayout
pnlNorthNorth – BorderLayout
pnlNorthCenter – GridLayout(1,2)
pnlAge – GridLayout(5,1), 3 JRadioButton + två tomma platser
pnlHobby – GridLayout(5,1), 5 JCheckBox

Deklarera panelerna så här:

```
private JPanel pnlNorth = new JPanel(new BorderLayout());  
private JPanel pnlNorthNorth = new JPanel(new BorderLayout());  
private JPanel pnlNorthCenter = new JPanel(new GridLayout(1,2));  
private JPanel pnlAge = new JPanel(new GridLayout(5,1));  
private JPanel pnlHobby = new JPanel(new GridLayout(5,1));
```

Tänk på att paketet *java.awt* ska importeras.

3. Huvudpanelen - SurveyPanel

SurveyPanel ska ha en *JTextArea*-komponent i CENTER och *pnlNorth* i NORTH. Lägg till en instansvariabel av typen *JTextArea*.

```
private JTextArea taSummary = new JTextArea();
```

Placera sedan *pnlNorth* (i NORTH) och *taSummary* (i CENTER) i fönstret.

```
add( pnlNorth, BorderLayout.NORTH );  
add( taSummary, BorderLayout.CENTER );
```

Testkör programmet. Nu kommer ett helt vitt fönster visa sig. Det är *taSummary* som fyller hela fönstret. *pnlNorth* innehåller inte några komponenter och får därför ingen storlek.

4. pnlNorth

pnlNorth ska innehålla tre komponenter:

1. *pnlNorthNorth* (NORTH)
2. *pnlNorthCenter* (CENTER)
3. JButton, *btnSummary* (SOUTH)

Lägg till en instansvariabel av typen JButton:

```
private JButton btnSummary = new JButton( "Sammanfattning" );
```

Lägg sedan till de tre komponenterna i panelen *pnlNorth*:

```
pnlNorth.add(pnlNorthNorth, BorderLayout.NORTH);  
pnlNorth.add(pnlNorthCenter, BorderLayout.CENTER);  
pnlNorth.add(btnSammanfattning, BorderLayout.SOUTH);
```

```
add( pnlNorth, BorderLayout.NORTH );
```

Testkör programmet. Nu kommer *btnSummary* att synas högst upp i fönstret. Resten av fönstret fylls av *taSummary*.



5. Namn - pnlNorthNorth

Namn-delen består av en *JLabel* (texten Namn:) och en *JTextField*. Lägg till en *JLabel* och en *JTextField* som instansvariabler.

Namn:

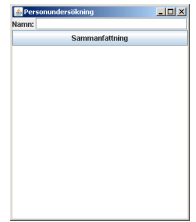
```
private JLabel lblName = new JLabel( "Namn: " );  
private JTextField tfName = new JTextField();
```

Placera sedan komponenterna i panelen *pnlNorthNorth*.

```
pnlNorthNorth.add(lblName, BorderLayout.WEST);  
pnlNorthNorth.add(tfName, BorderLayout.CENTER);
```

```
pnlNorth.add(pnlNorthNorth, BorderLayout.NORTH);
```

Testkör programmet. Nu kommer *pnlNorthNorth* att synas högst upp, därunder *btnSummary* och resten av fönstret fylls av *taSummary*.



6. Ålder och Hobby - pnlNorthCenter

pnlNorthCenter ska innehålla panelerna *pnlAge* och *pnlHobby*. Placera panelerna i *pnlNorthCenter*.

```
pnlNorthNorth.add(tfName, BorderLayout.CENTER);  
pnlNorthCenter.add(pnlAge);  
pnlNorthCenter.add(pnlHobby);
```

```
pnlNorth.add(pnlNorthNorth, BorderLayout.NORTH);
```

7. Ålder - pnlAge

pnlAge ska innehålla tre **JRadioButton**. Dessutom används en **ButtonGroup** (osynlig) för att radioknapparna ska fungera. Lägg till JRadioButton-komponenterna och ButtonGroup-komponenten som instansvariabler.

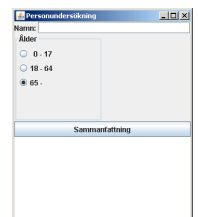
```
private JRadioButton rb0_17 = new JRadioButton( " 0 - 17" );  
private JRadioButton rb18_64 = new JRadioButton( "18 - 64" );  
private JRadioButton rb65 = new JRadioButton( "65 -" );  
private ButtonGroup buttonGroup = new ButtonGroup();
```

- Panelen ska ha en *TitledBorder* (i konstruktorn):

```
panelAge.setBorder( BorderFactory.createTitledBorder( "Ålder" ) );
```
- Placera *JRadioButton*-komponenterna i *buttonGroup* och i panelen *pnlAge*. Dessutom ska en radiobutton vara vald från början. Den sista sätter valet 65- som förvalt.

```
panelAge.setBorder( BorderFactory.createTitledBorder( "Ålder" ) );  
buttonGroup.add( rb0_17 );  
buttonGroup.add( rb18_64 );  
buttonGroup.add( rb65 );  
pnlAlder.add( rb0_17 );  
pnlAlder.add( rb18_64 );  
pnlAlder.add( rb65 );  
rb65.setSelected(true);
```

Testkör programmet.



8. Hobby - pnlHobby

pnlHobby ska innehålla fem **JCheckBox**. Lägg till komponenterna som instansvariabler.

```
private JCheckBox cbSport = new JCheckBox( "Idrott" );  
private JCheckBox cbDance = new JCheckBox( "Folkdans" );  
private JCheckBox cbBirds = new JCheckBox( "Fågelskådning" );  
private JCheckBox cbBridge = new JCheckBox( "Bridge" );  
private JCheckBox cbSong = new JCheckBox( "Körsång" );
```

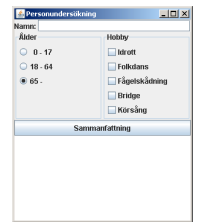
- Panelen ska ha en *TitledBorder* (se punkt 7).

```
rb65.setSelected(true);
```

```
pnlHobby.setBorder( BorderFactory.createTitledBorder( "Hobby" ) );
```

- Placera *JCheckBox*-komponenterna i panelen.

```
pnlHobby.setBorder( BorderFactory.createTitledBorder( "Hobby" ) );  
pnlHobby.add( cbSport );  
pnlHobby.add( cbDance );  
pnlHobby.add( cbBirds );  
pnlHobby.add( cbBridge );  
pnlHobby.add( cbSong );
```



Testkör programmet.

9. Händelse-hantering

När användaren klickar på Sammanfatta-knappen så ska lite sammanfattande information skrivas ut i *JTextArea*-komponenten (se figur).

- Importera paketet *java.awt.event*
- Skriv den inre klassen ***SummaryListener*** i klassen *SurveyPanel*. Klassen ska implementera interfacet *ActionListener*. Det innebär att du måste lägga till metoden *actionPerformed* i klassen.

```
private class SummaryListener implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        summary();  
    }  
}
```

- Klassen ska registreras som lyssnare till sammanfattnings-knappen:

```
add( taSummary, BorderLayout.CENTER );  
btnSummary.addActionListener( new SummaryListener() );
```

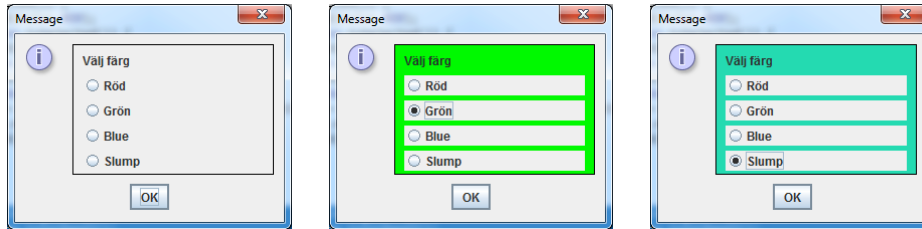
- Metoden *summary* måste skrivas. Nedan ser du en version av metoden:

```
public void summary() {  
    String res = tfName.getText() + "\n";  
    // Lägg till korrekt ålder  
    if( rb0_17.isSelected() )  
        res += "Ålder: " + "0 - 17 år\n";  
    else if( rb18_64.isSelected() )  
        res += "Ålder: " + "18 - 64 år\n";  
    else  
        res += "Ålder: " + "65 - år\n";  
    // Lägg till hobbies  
    res += "Hobby: ";  
    if( cbSport.isSelected() )  
        res += " Idrott";  
    if( cbDance.isSelected() )  
        res += " Folkdans";  
    if( cbBirds.isSelected() )  
        res += " Fågelskådning";  
    if( cbBridge.isSelected() )  
        res += " Bridge";  
    if( cbSong.isSelected() )  
        res += " Körsång";  
    res += "\n";  
    // Visa texten i JTextArea-komponenten  
    taSummary.setText( res );  
}
```

Nu är det åter dags att testköra programmet.

Uppgift 16b

Skriv klassen `ColorPanel` vilken ska innehålla fyra `JRadioButton`-komponenter. Om användaren väljer Röd så ska panelen färgas röd osv. Slump ska ge en slumpmässig färg.



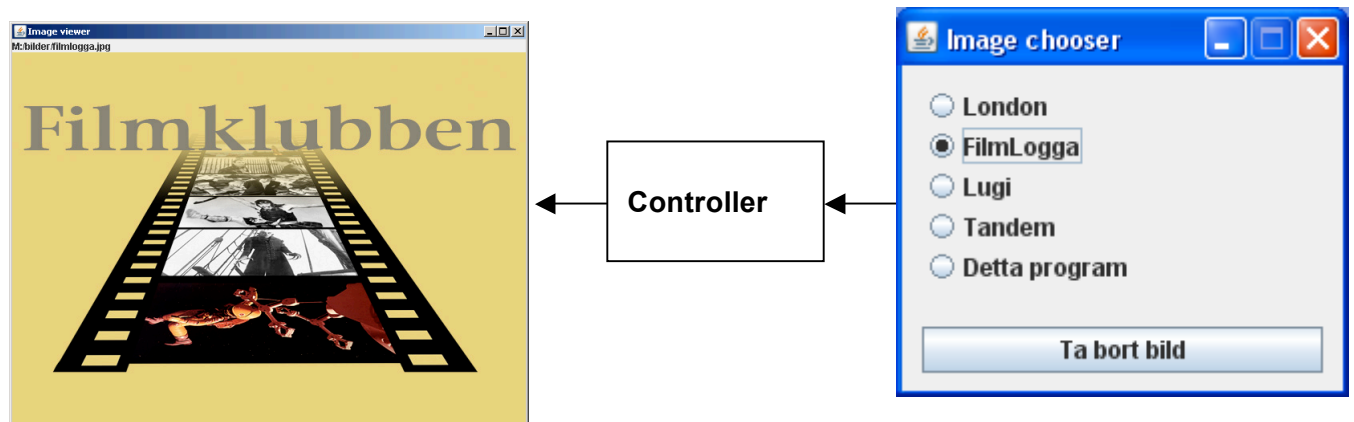
Om du visar panelen med nedanstående main-metod så kan den se ut som figurerna ovan.

```
public static void main(String[] args) {
    ColorPanel panel = new ColorPanel();
    panel.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    JOptionPane.showMessageDialog(null, panel);
}
```

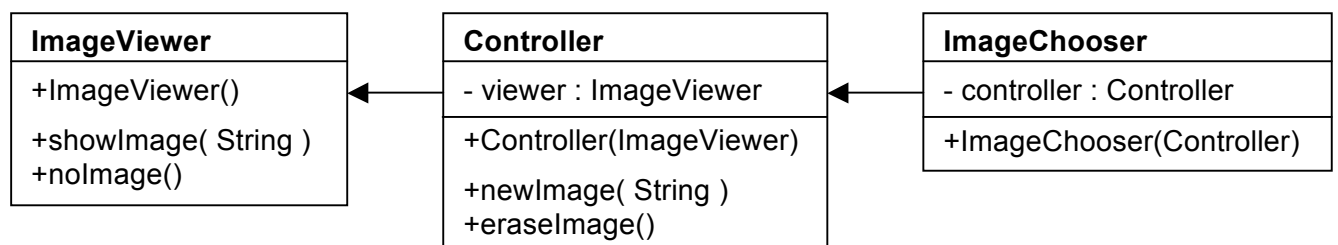
Uppgift 16c

Uppgiften handlar om att visa bilder. I uppgiften används fullständiga filnamn. Börja med att placera bilderna i `DA339AL16HT15.zip` i `M:\bilder` (eller där du önskar).

I ett fönster (det högra) väljer användaren den bild som ska visas med `JRadioButton`-komponenter. I det vänstra fönstret visas den valda bilden.



Ovanstående bild kan även beskrivas med hjälp av klassdiagram:



ImageViewer

Börja med att skapa en klass som ärver `JPanel`. Sätt layout-managern till `BorderLayout`. Ge klassen namnet ***ImageViewer***. Placera en ***JLabel***-komponent i `NORTH` och en ***JLabel***-komponent i `CENTER`. Den första ska visa bildfilens namn och den andra en bild. Glöm inte att ge ***JLabel***-komponenterna bra namn, t.ex. *lblTitle* och *lblImage*.

ImageViewer-klassen ska inte ha fler instansvariabler. Klassen används endast för att visa bilder (genom anrop till någon av metoderna *showImage(String)* eller *noImage()*).

- *public void noImage()* ska ta bort bilden som visas och i stället visa texten "INGEN BILD VALD" i *JLabel*-komponenten. *lblTitle* ska tömas på sitt innehåll.

```
lblImage.setIcon( null ); // ta bort bild  
lblImage.setText( "INGEN BILD VALD" );  
lblTitle.setText( "" );
```



- *public void showImage(String filename)* tar emot sökvägen till en bildfil vilken ska visas i *JLabel*-komponenten. *lblTitle* ska visa filnamnet.

```
lblImage.setText( "" ); // ta bort text som visas  
lblImage.setIcon( new ImageIcon( filename ) );  
lblTitle.setText( filename );
```



Du kan testa *ImageViewer*-klassen med main-metoden:

```
public static void main(String args[]) {  
    ImageViewer viewer = new ImageViewer();  
    viewer.showImage("M:/bilder/filmlogga.jpg"); // alt. viewer.noImage();  
    JOptionPane.showMessageDialog(null, viewer);  
}
```

Controller

Börja med att skapa en vanlig java-klass som du döper till **Controller**. Du ska lägga till en instansvariabel i klassen, en konstruktor och två metoder. Dessa beskrivs nedan och i klassdiagrammet till höger..

Instansvariabler

Controller-klassen ska ha en referens, till ett *ImageViewer*-objekt (instansvariabeln *viewer*).

```
private ImageViewer viewer;
```

Konstruktor

Controller-objektet får en referens till instansvariabeln när det skapas:

```
ImageViewer viewer = new ImageViewer();  
Controller cont = new Controller( viewer );
```

Denna referens ska lagras i instansvariabeln *viewer*.

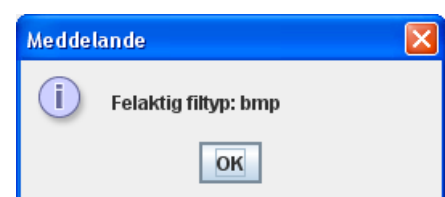
```
public Controller( ImageViewer inViewer ) {  
    this.viewer = inViewer;  
}
```

Metoder

Controller-klassen har två metoder:

- *newImage(String)* vilken tar emot sökvägen till en bildfil (t.ex. "C:/temp/sommar.jpg"). Metoden ska kontrollera att bildfilen är av korrekt typ (jpg, gif eller png). Om bildfilen är av korrekt typ så ska metoden anropa metoden *showImage(String)* i *ImageViewer*-objektet. Om bildfilen är av otillåten typ så ska ett felmeddelande visas (figur till

Controller
- viewer : ImageViewer
+Controller(ImageViewer)
+newImage(String)
+eraseImage()



höger).

```
public void newImage( String filename ) {
    int lastDot = filename.lastIndexOf( '.' );
    String suffix = filename.substring( lastDot+1 );
    suffix = suffix.toLowerCase(); // alla tecken små bokstäver
    if( suffix.equals("jpg") || suffix.equals("gif") || suffix.equals("png") ) {
        viewer.showImage( filename );
    } else {
        JOptionPane.showMessageDialog( null, "Felaktig filtyp: " + suffix );
    }
}
```

- *eraseImage()* vilken ska anropa metoden *noImage()* i *ImageViewer*-objektet.

```
public void eraseImage() {
    viewer.noImage();
}
```

Du kan testa *Controller*-klassen med följande main-metod. Endast en av de tre sista raderna ska vara aktiv vid exekvering (test av olika fall).

```
public static void main(String[] args) {
    ImageViewer viewer = new ImageViewer();
    Controller cont = new Controller( viewer );
    cont.newImage( "M:/bilder/filmlogga.jpg" );
    // cont.newImage( "M:/bilder/karta.bmp" ); // ej tillåten filtyp
    // cont.eraseImage();
}
```

ImageChooser

Börja med att skapa en klass ärver *JPanel*. Sätt layout-manager till *BorderLayout*. Ge klassen namnet *ImageChooser*.

Instansvariabler

1 *JPanel* (t.ex. *pnlRadioButtons*)
5 *JRadioButtons*
1 *JButton*

Glöm inte att ge komponenterna vettiga namn.

ImageChooser-klassen ska ha ytterligare en instansvariabel , nämligen referens till ett *Controller*-objekt:

```
private Controller controller;
```

Konstruktör

ImageChooser-objektet får en referens till ett *Controller*-objekt när det skapas.

```
Controller cont = new Controller( viewer );
ImageChooser chooser = new ImageChooser( cont );
```

Skriv en konstruktör, **public ImageChooser(Controller controller)**, vilken tilldelar instansvariabeln *controller* en referens till ett *Controller*-objekt. Dessutom ska fönstret och dess komponenter byggas i konstruktorn.

Sätt layout-manager i *pnlRadioButtons* till *GridLayout* med 5 rader och 1 kolumn. Placera *JRadioButton*-komponenterna i panelen.

Placera sedan *pnlRadioButtons* i *CENTER* och *JButton*-komponenten i *SOUTH* i huvudpanelen.



Händelsehantering

Nu är det dags att införa händelsehantering i *ImageChooser*-klassen.

Vid klick på knappen ska metoden *eraseImage()* anropas i *Controller*-klassen. Du ska skriva den inre klassen *EraseListener*:

```
private class EraseListener implements ActionListener {  
    public void actionPerformed( ActionEvent e ) {  
        controller.eraseImage();  
    }  
}
```

Vid val av *JRadioButton*-komponent ska metoden *newImage* anropas i *Controller*-klassen.

För att lyssna på *JRadioButton*-komponenterna ska du använda en *ActionListener* och lägga till metoden *actionPerformed(ActionEvent e)*.

Skriv den inre klassen *ImageListener*. Klassen ska implementera *ActionListener* och innehålla metoden *actionPerformed*:

```
public void actionPerformed(ItemEvent e) {  
    if( rbImage1.isSelected() )  
        controller.newImage("M:/bilder/london06.jpg");  
    else if( rbImage2.isSelected() )  
        controller.newImage("M:/bilder/filmlogga.jpg");  
    else if( rbImage3.isSelected() )  
        controller.newImage("M:/bilder/lugi.gif");  
    else if( rbImage4.isSelected() )  
        controller.newImage("M:/bilder/tandem1.jpg");  
    else if( rbImage5.isSelected() )  
        controller.newImage("M:/bilder/program.bmp");  
}
```

Slurligen måste du registrera att ett objekt av typen *EraseListener* ska lyssna på ”Ta bort bild”-knappen och att ett objekt av typen *ImageListener* ska lyssna på *JRadiobutton*-komponenterna.

Lägg till metoden *addListeners* och anropa metoden från konstruktorn.

```
public void addListeners() {  
    ImageListener listener = new ImageListener();  
    rbImage1.addActionListener( listener );  
    rbImage2.addActionListener( listener );  
    rbImage3.addActionListener( listener );  
    rbImage4.addActionListener( listener );  
    rbImage5.addActionListener( listener );  
    btnErase.addActionListener( new EraseListener() );  
}
```

Nu är det dags att testköra programmet. Skriv klassen *ImageApp* (se nästa sida):


```
package laboration11;
import javax.swing.*.*;

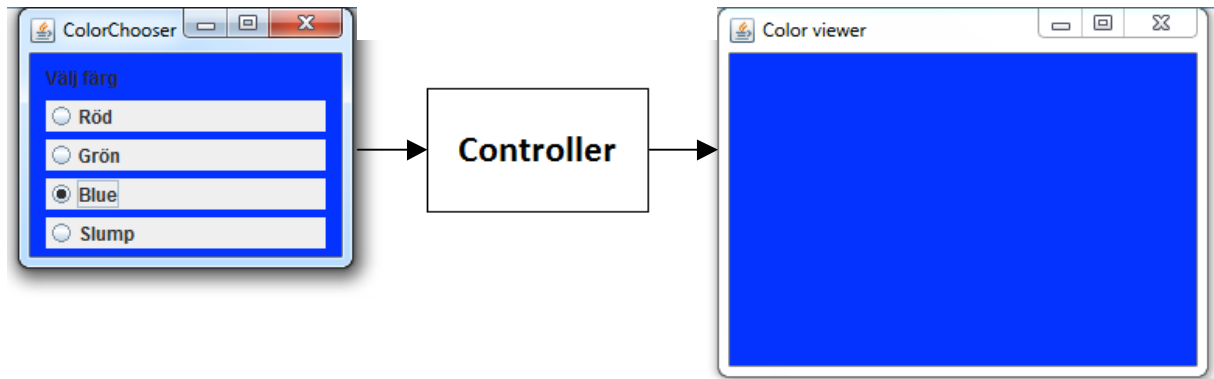
public class ImageApp {
    public void newFrame(JPanel panel, String title) {
        JFrame frame = new JFrame( title );
        frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        frame.add( panel );
        frame.pack();
        frame.setVisible( true );
    }
    public static void main(String[] args) {
        ImageApp app = new ImageApp();
        ImageViewer viewer = new ImageViewer();
        Controller cont = new Controller( viewer );
        ImageChooser chooser = new ImageChooser( cont );
        app.newFrame( viewer, "Image viewer" );
        app.newFrame( chooser, "Image chooser" );
    }
}
```

När programmet fungerar ska du förvissa dig om att

- du förstår hur anropen går till:
 1. Från val av radio-button till visning av bild / felmeddelande.
 2. Från klick på knappen till visningen av texten ”INGEN BILD VALD”.
- Hur nödvändiga referenser ges värde (*controller* i ImageChooser och *viewer* i ImageViewer)
- Hur fönsterna skapas i ImageApp.

Uppgift 16d

Använd klassen `ColorPanel` för att sätta färg på en `JLabel`-komponent (i fönstret med titeln `Color viewer` nedan). Figuren visar hur systemet är tänkt att hänga ihop. Som du ser måste `ColorPanel` känna till `Controller`-objekt (ha en instansvariabel av typen `Controller`) och en konstruktor som tar emot en instans av typen `Controller`.



Nedanstående main-metod ska starta programmet.

```
import javax.swing.JFrame;
import javax.swing.JPanel;

public class Laboration16d {

    private void showFrame(String title, int x, int y, JPanel panel) {
        JFrame frame = new JFrame(title);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.add(panel);
        frame.pack();
        frame.setLocation(x, y);
        frame.setVisible(true);
    }

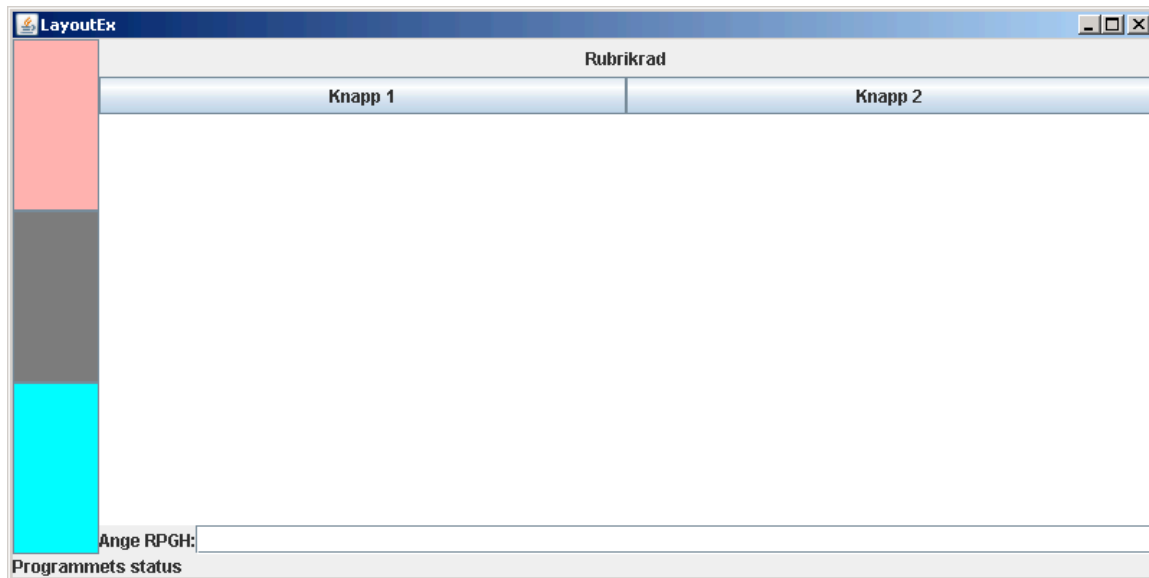
    public void action() {
        LabelPanel labelPanel = new LabelPanel();
        Controller controller = new Controller(labelPanel);
        ColorPanel colorPanel = new ColorPanel(controller);
        showFrame("ColorChooser", 50, 50, colorPanel);
        showFrame("Color viewer", 500, 50, labelPanel);
    }

    public static void main(String[] args) {
        Laboration16d lab16d = new Laboration16d();
        lab16d.action();
    }
}
```

Fördjupande uppgifter

Uppgift 16e

Uppgiften går ut på att placera komponenter med hjälp av LayoutManagerna BorderLayout och GridLayout. Fönstret är samma som på föreläsningen.



1. Skapa en klass som ärver JPanel

Börja uppgiften med att skapa en klass som ärver JPanel. Panelen ska vara 800 x 400 punkter stort.

Skriv en main-metod som visar panelen i ett fönster. Kontrollera att fönstret visar sig när du kör programmet.

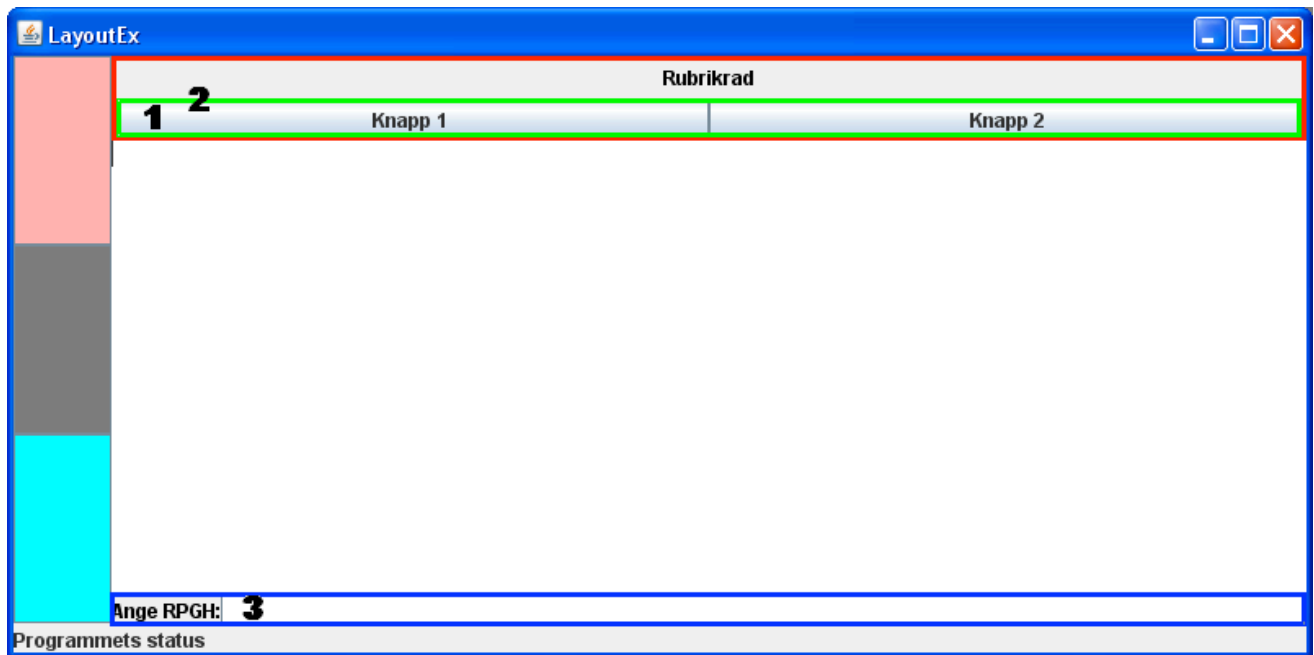
2. Deklarera komponenter som används

Nu ska du deklarerar alla komponenter som du hittar i fönstret ovan. Det är:

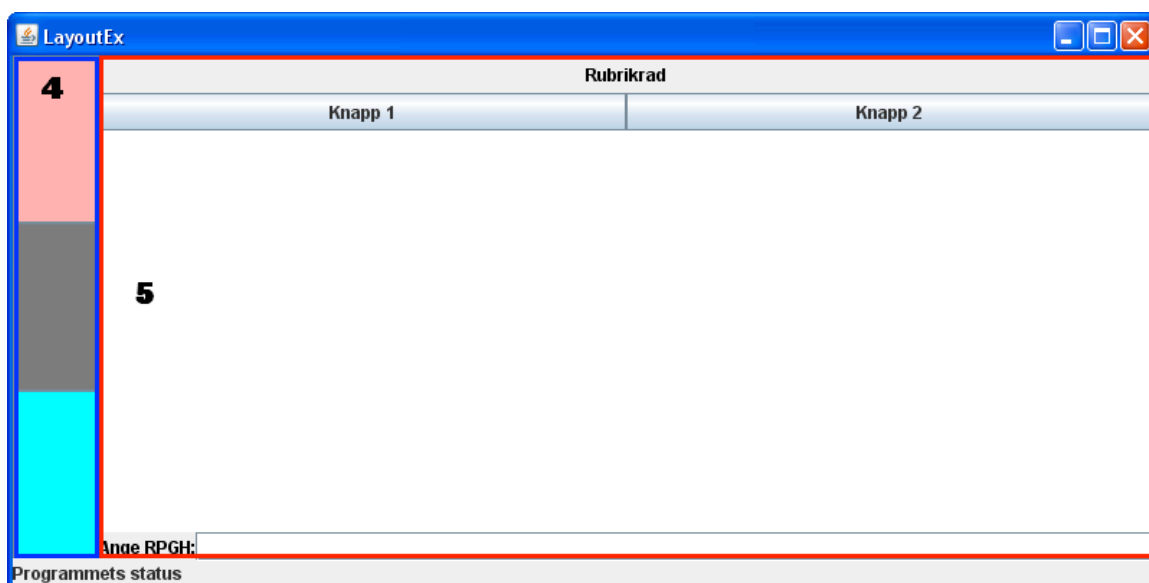
- 3 st *JLabel*-komponenter ("Rubrikrad", "Ange RPGH:" och "Programmets status")
- 5 st *JButton*-komponenter (rosa knapp, grå knapp, cyan knapp, "Knapp 1" och "Knapp 2")
- 1 st *JTextField* (inmatningsfönstret till höger om "Ange RPGH:")
- 1 st *JTextArea* (den stora vita ytan i mitten)
- 5 st *JPanel* vilka kommer att användas i layouten. LayoutManager för vardera panelen ska du försöka räkna ut med hjälp av fönstret.

3. Upptäck olika typer av layout-managers

Nu ska du försöka se olika typer av Layout-managers i fönstret:



1. Den gröna rektangeln visar en panel med två knappar. Båda får lika stort utrymme. I lösningen heter panelen **northButtonPanel**. Ange vid konstruktion av panelen den *LayoutManager* som ska användas. Placera knapparna på panelen.
2. Den röda rektangeln visar en panel med två rader. Texten "Rubrikrad" på den översta raden och **northButtonPanel** på den undre raden. I lösningen heter panelen **northPanel**. Ange vid konstruktion av panelen den *LayoutManager* som ska användas. Placera sedan komponenterna på panelen.
3. Den blå rektangeln (ganska långt ner) innehåller en *JLabel* till vänster och resten av rektangeln fylls av en *TextField*. I lösningen heter panelen **textFieldPanel**. Ange vid konstruktion av panelen den *LayoutManager* som ska användas. Placera sedan komponenterna på panelen.



4. Den blå rektangeln längst till vänster innehåller tre *JButton*-komponenter, samtliga lika stora. I lösningen heter panelen **westPanel**. Innan knapparna placeras kan du ange deras

färg och den storlek de önskar (`setPreferredSize`). Panelens bredd kommer att ges av önskad bredd på knapparna. Ange vid konstruktion av panelen den `LayoutManager` som ska användas. Designa sedan komponenterna och placera slutligen komponenterna på panelen.

5. Den röda rektangeln innehåller *northPanel* högst upp, `JTextArea`-komponenter fyller utrymmet i mitten och *textFieldPanel* är längst ner. I lösningen heter panelen ***centerPanel***. Ange vid konstruktion av panelen den `LayoutManager` som ska användas. Placera sedan komponenterna på panelen.
6. Nu är det endast huvudpanelen kvar. Det består av *westPanel* till vänster, en `JLabel` ("Programmets status") längst ner och resten fylls av panelen *centerPanel*. Detta är en `BorderLayout` och panelen ska därför ha `BorderLayout`. Placera komponenterna i panelen.

Lösningsförslag

Uppgift 16a

```
package laboration16;
import javax.swing.JFrame;

public class SurveyApp {
    public void start() {
        JFrame frame = new JFrame( "Personundersökning" );
        frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        frame.add( new SurveyPanel() );
        frame.pack();
        frame.setVisible( true );
    }
    public static void main(String[] args) {
        SurveyApp app = new SurveyApp();
        app.start();
    }
}

-----
package laboration16;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class SurveyPanel extends JPanel {
    // paneler
    private JPanel pnlNorth = new JPanel(new BorderLayout());
    private JPanel pnlNorthNorth = new JPanel(new BorderLayout());
    private JPanel pnlNorthCenter = new JPanel(new GridLayout(1, 2));
    private JPanel pnlAge = new JPanel(new GridLayout(5, 1));
    private JPanel pnlHobby = new JPanel(new GridLayout(5, 1));

    private JTextArea taSummary = new JTextArea();
    private JButton btnSummary = new JButton( "Sammanfattning" );
    // namn
    private JLabel lblName = new JLabel( "Namn: " );
    private JTextField tfName = new JTextField();
    // ålder
    private JRadioButton rb0_17 = new JRadioButton( " 0 - 17" );
    private JRadioButton rb18_64 = new JRadioButton( "18 - 64" );
    private JRadioButton rb65 = new JRadioButton( "65 -" );
    private ButtonGroup buttonGroup = new ButtonGroup();
    // hobby
    private JCheckBox cbSport = new JCheckBox( "Idrott" );
    private JCheckBox cbDance = new JCheckBox( "Folkdans" );
    private JCheckBox cbBirds = new JCheckBox( "Fågelskådning" );
    private JCheckBox cbBridge = new JCheckBox( "Bridge" );
    private JCheckBox cbSong = new JCheckBox( "Körsång" );

    public SurveyPanel() {
        setPreferredSize( new Dimension( 300, 360 ) );
        setLayout( new BorderLayout() );

        // pnlAlder
        pnlAge.setBorder( BorderFactory.createTitledBorder( "Ålder" ) );
        buttonGroup.add( rb0_17 );
        buttonGroup.add( rb18_64 );
        buttonGroup.add( rb65 );
        pnlAge.add( rb0_17 );
        pnlAge.add( rb18_64 );
        pnlAge.add( rb65 );
        rb65.setSelected(true);
        // pnlHobby
        pnlHobby.setBorder( BorderFactory.createTitledBorder( "Hobby" ) );
        pnlHobby.add( cbSport );
        pnlHobby.add( cbDance );
    }
}
```

```
        pnlHobby.add( cbBirds );
        pnlHobby.add( cbBridge );
        pnlHobby.add( cbSong );
        //pnlNorthNorth + pnlNorthCenter
        pnlNorthNorth.add(lblName, BorderLayout.WEST);
        pnlNorthNorth.add(tfName, BorderLayout.CENTER);
        pnlNorthCenter.add(pnlAge);
        pnlNorthCenter.add(pnlHobby);
        //pnlNorth
        pnlNorth.add(pnlNorthNorth, BorderLayout.NORTH);
        pnlNorth.add(pnlNorthCenter, BorderLayout.CENTER);
        pnlNorth.add(btnSummary, BorderLayout.SOUTH);

        add( pnlNorth, BorderLayout.NORTH );
        add( taSummary, BorderLayout.CENTER );
        btnSummary.addActionListener( new SummaryListener() );
    }

    private class SummaryListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            summary();
        }
    }

    public void summary() {
        String res = tfName.getText() + "\n";
        // Lägg till korrekt ålder
        if( rb0_17.isSelected() )
            res += "Ålder: " + "0 - 17 år\n";
        else if( rb18_64.isSelected() )
            res += "Ålder: " + "18 - 64 år\n";
        else
            res += "Ålder: " + "65 - år\n";
        // Lägg till hobbies
        res += "Hobby: ";
        if( cbSport.isSelected() )
            res += " Idrott";
        if( cbDance.isSelected() )
            res += " Folkdans";
        if( cbBirds.isSelected() )
            res += " Fågelskådning";
        if( cbBridge.isSelected() )
            res += " Bridge";
        if( cbSong.isSelected() )
            res += " Körsång";
        res += "\n";
        // Visa texten i JTextArea-komponenten
        taSummary.setText( res );
    }
}
```

Uppgift 16b

```
package laboration16;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class ColorPanel extends JPanel {
    private JLabel lblColor = new JLabel("Välj färg");
    private JRadioButton rbRed = new JRadioButton("Röd");
    private JRadioButton rbGreen = new JRadioButton("Grön");
    private JRadioButton rbBlue = new JRadioButton("Blue");
    private JRadioButton rbRandom = new JRadioButton("Slump");
    private Random rand = new Random();

    public ColorPanel() {
        setPreferredSize(new Dimension(200, 130));

        lblColor.setPreferredSize(new Dimension(180, 20));
        ButtonGroup group = new ButtonGroup();
        rbRed.setPreferredSize(new Dimension(180, 20));
        rbGreen.setPreferredSize(new Dimension(180, 20));
        rbBlue.setPreferredSize(new Dimension(180, 20));
        rbRandom.setPreferredSize(new Dimension(180, 20));
        group.add(rbRed);
        group.add(rbGreen);
        group.add(rbBlue);
        group.add(rbRandom);

        addListeners();

        add(lblColor);
        add(rbRed);
        add(rbGreen);
        add(rbBlue);
        add(rbRandom);
    }

    private void addListeners() {
        ColorListener listener = new ColorListener();
        rbRed.addActionListener(listener);
        rbGreen.addActionListener(listener);
        rbBlue.addActionListener(listener);
        rbRandom.addActionListener(listener);
    }

    private class ColorListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            Color color = null;
            if (rbRed.isSelected()) {
                color = Color.RED;
            } else if (rbGreen.isSelected()) {
                color = Color.GREEN;
            } else if (rbBlue.isSelected()) {
                color = Color.BLUE;
            } else if (rbRandom.isSelected()) {
                color = new Color(rand.nextInt(256), rand.nextInt(256),
rand.nextInt(256));
            }
            setBackground(color);
        }
    }

    public static void main(String[] args) {
        ColorPanel panel = new ColorPanel();
        panel.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        JOptionPane.showMessageDialog(null, panel);
    }
}
```


Uppgift 16c

```
package laboration16;
import javax.swing.*;
import java.awt.*;

public class ImageViewer extends JPanel {
    private JLabel lblTitle = new JLabel();
    private JLabel lblImage = new JLabel("INGEN BILD VALD", JLabel.CENTER);

    public ImageViewer() {
        setLayout( new BorderLayout() );
        lblImage.setPreferredSize( new Dimension( 640, 480 ) );
        add(lblTitle, BorderLayout.NORTH);
        add(lblImage, BorderLayout.CENTER);
    }

    public void showImage(String filename) {
        lblTitle.setText(filename);
        lblImage.setText("");
        lblImage.setIcon(new ImageIcon(filename));
    }

    public void noImage() {
        lblImage.setText("INGEN BILD VALD");
        lblImage.setIcon(null);
    }

    public static void main(String args[]) {
        ImageViewer viewer = new ImageViewer();
        viewer.showImage("M:/bilder/filmlogga.jpg"); // alt. viewer.noImage();
        JOptionPane.showMessageDialog(null, viewer);
    }
}

-----
package laboration16;
import javax.swing.*;

public class Controller {
    private ImageViewer viewer;

    public Controller( ImageViewer inViewer ) {
        this.viewer = inViewer;
    }

    public void newImage( String filename ) {
        int lastDot = filename.lastIndexOf( '.' );
        String suffix = filename.substring( lastDot+1 );
        suffix = suffix.toLowerCase(); // alla tecken små bokstäver
        if( suffix.equals( "jpg" ) || suffix.equals( "gif" ) || suffix.equals( "png"
) ) {
            viewer.showImage( filename );
        } else {
            JOptionPane.showMessageDialog( null, "Felaktig filtyp: " + suffix );
        }
    }

    public void eraseImage() {
        viewer.noImage();
    }

    public static void main(String[] args) {
        ImageViewer viewer = new ImageViewer();
        Controller cont = new Controller( viewer );
        cont.newImage( "M:/bilder/filmlogga.jpg" );
        // cont.newImage( "M:/bilder/karta.bmp" ); // ej tillåten filtyp
        // cont.eraseImage();
        JOptionPane.showMessageDialog( null, viewer );
    }
}

-----
```

```
package laboration16;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class ImageChooser extends JPanel {
    private Controller controller;
    private JRadioButton rbImage1 = new JRadioButton( "London" );
    private JRadioButton rbImage2 = new JRadioButton( "FilmLogga" );
    private JRadioButton rbImage3 = new JRadioButton( "Lugi" );
    private JRadioButton rbImage4 = new JRadioButton( "Tandem" );
    private JRadioButton rbImage5 = new JRadioButton( "Detta program" );
    private ButtonGroup buttonGroup = new ButtonGroup();
    private JButton btnErase = new JButton( "Ta bort bild" );

    /** Creates new form ImageChooser */
    public ImageChooser( Controller controller ) {
        JPanel pnlRadioButtons = new JPanel( new GridLayout(5,1) );
        this.controller = controller;
        setLayout( new BorderLayout() );
        setPreferredSize( new Dimension( 200, 190 ) );
        buttonGroup.add( rbImage1 );
        buttonGroup.add( rbImage2 );
        buttonGroup.add( rbImage3 );
        buttonGroup.add( rbImage4 );
        buttonGroup.add( rbImage5 );
        btnErase.setBounds(10, 130, 200, 23);

        addListeners();

        pnlRadioButtons.add( rbImage1 );
        pnlRadioButtons.add( rbImage2 );
        pnlRadioButtons.add( rbImage3 );
        pnlRadioButtons.add( rbImage4 );
        pnlRadioButtons.add( rbImage5 );
        add(pnlRadioButtons, BorderLayout.CENTER);
        add( btnErase, BorderLayout.SOUTH );
    }

    public void addListeners() {
        ImageListener listener = new ImageListener();
        rbImage1.addActionListener( listener );
        rbImage2.addActionListener( listener );
        rbImage3.addActionListener( listener );
        rbImage4.addActionListener( listener );
        rbImage5.addActionListener( listener );
        btnErase.addActionListener( new EraseListener() );
    }

    private class EraseListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            controller.eraseImage();
        }
    }

    private class ImageListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            if (rbImage1.isSelected())
                controller.newImage("M:/bilder/london06.jpg");
            else if (rbImage2.isSelected())
                controller.newImage("M:/bilder/filmlogga.jpg");
            else if (rbImage3.isSelected())
                controller.newImage("M:/bilder/lugi.gif");
            else if (rbImage4.isSelected())
                controller.newImage("M:/bilder/tandem1.jpg");
            else if (rbImage5.isSelected())
                controller.newImage("M:/bilder/program.bmp");
        }
    }
}
```

```
-----
package laboration16;
import javax.swing.*;

public class ImageApp {
    public void newFrame(JPanel panel, String title) {
        JFrame frame = new JFrame( title );
        frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        frame.add( panel );
        frame.pack();
        frame.setVisible( true );
    }
    public static void main(String[] args) {
        ImageApp app = new ImageApp();
        ImageViewer viewer = new ImageViewer();
        Controller cont = new Controller( viewer );
        ImageChooser chooser = new ImageChooser( cont );
        app.newFrame( viewer, "Image viewer" );
        app.newFrame( chooser, "Image chooser" );
    }
}
```

Uppgift 16d

```
// ColorPanel
package laboration16;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class ColorPanel extends JPanel {
    private Controller controller;
    private JLabel lblColor = new JLabel("Välj färg");
    private JRadioButton rbRed = new JRadioButton("Röd");
    private JRadioButton rbGreen = new JRadioButton("Grön");
    private JRadioButton rbBlue = new JRadioButton("Blue");
    private JRadioButton rbRandom = new JRadioButton("Slump");
    private Random rand = new Random();

    public ColorPanel( Controller controller) {
        this.controller = controller;
        setPreferredSize(new Dimension(200, 130));

        lblColor.setPreferredSize(new Dimension(180, 20));
        ButtonGroup group = new ButtonGroup();
        rbRed.setPreferredSize(new Dimension(180, 20));
        rbGreen.setPreferredSize(new Dimension(180, 20));
        rbBlue.setPreferredSize(new Dimension(180, 20));
        rbRandom.setPreferredSize(new Dimension(180, 20));
        group.add(rbRed);
        group.add(rbGreen);
        group.add(rbBlue);
        group.add(rbRandom);

        addListeners();

        add(lblColor);
        add(rbRed);
        add(rbGreen);
        add(rbBlue);
        add(rbRandom);
    }
}
```

```
private void addListeners() {
    ColorListener listener = new ColorListener();
    rbRed.addActionListener(listener);
    rbGreen.addActionListener(listener);
    rbBlue.addActionListener(listener);
    rbRandom.addActionListener(listener);
}

private class ColorListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        Color color = null;
        if (rbRed.isSelected()) {
            color = Color.RED;
        } else if (rbGreen.isSelected()) {
            color = Color.GREEN;
        } else if (rbBlue.isSelected()) {
            color = Color.BLUE;
        } else if (rbRandom.isSelected()) {
            color = new Color(rand.nextInt(256), rand.nextInt(256),
rand.nextInt(256));
        }
        setBackground(color);
        controller.newColor(color);
    }
}

}

package laboration16;
import java.awt.Color;

// Controller
public class Controller {
    private LabelPanel labelPanel;

    public Controller(LabelPanel panel) {
        this.labelPanel = panel;
    }

    public void newColor(Color color) {
        labelPanel.setColor(color);
    }
}

package laboration16;
import java.awt.*;
import javax.swing.*;

// LabelPanel
public class LabelPanel extends JPanel {
    private JLabel lblColor = new JLabel();

    public LabelPanel() {
        this.setPreferredSize(new Dimension(300,200));
        this.setLayout(new BorderLayout());
        lblColor.setOpaque(true);
        this.add(lblColor, BorderLayout.CENTER);
    }

    public void setColor(Color color) {
        lblColor.setBackground(color);
    }
}
```

Uppgift 16e

```
package laboration16;
import javax.swing.*;
import java.awt.*;

public class LayoutPanel extends JPanel {
    private JLabel lblTitle = new JLabel("Rubrikrad",JLabel.CENTER);
    private JButton btn1 = new JButton("Knapp 1"), // Andra raden uppifrån
        btn2 = new JButton("Knapp 2"),
        btnPink = new JButton(),
        btnGray = new JButton(),
        btnCyan = new JButton();
    private JTextArea taCenter = new JTextArea();
    private JLabel lblRpgh = new JLabel("Ange RPGH:");
    private JTextField tfRpgh = new JTextField();
    private JLabel lblStatus = new JLabel("Programmets status");
    private JPanel westPanel = new JPanel(new GridLayout(3,1));
    private JPanel northPanel = new JPanel(new GridLayout(2,1));
    private JPanel northButtonPanel = new JPanel(new GridLayout(1,2));
    private JPanel rpghPanel = new JPanel(new BorderLayout());
    private JPanel centerPanel = new JPanel(new BorderLayout());

    public LayoutPanel() {
        setPreferredSize( new Dimension( 800,400 ) );
        setLayout( new BorderLayout() );

        // knappraden till vänster
        btnPink.setBackground(Color.PINK);
        btnPink.setPreferredSize(new Dimension(60,60));
        btnGray.setBackground(Color.GRAY);
        btnGray.setPreferredSize(new Dimension(60,60));
        btnCyan.setBackground(Color.CYAN);
        btnCyan.setPreferredSize(new Dimension(60,60));
        westPanel.add(btnPink);
        westPanel.add(btnGray);
        westPanel.add(btnCyan);

        // panel med två knappar
        northButtonPanel.add(btn1);
        northButtonPanel.add(btn2);
        // panel i north
        northPanel.add(lblTitle);
        northPanel.add(northButtonPanel);

        // panel under textArea
        rpghPanel.add(lblRpgh,BorderLayout.WEST);
        rpghPanel.add(tfRpgh,BorderLayout.CENTER);

        //panel till höger om knapprad
        centerPanel.add(northPanel,BorderLayout.NORTH);
        centerPanel.add(taCenter,BorderLayout.CENTER);
        centerPanel.add(rpghPanel,BorderLayout.SOUTH);

        // layout i fönstret
        add(westPanel,BorderLayout.WEST);
        add(lblStatus,BorderLayout.SOUTH);
        add(centerPanel,BorderLayout.CENTER);
    }

    public static void main(String[] args) {
        JOptionPane.showMessageDialog( null, new LayoutPanel() );
    }
}
```