



# Videregående programmering i Java



Dag 7 - Swing og MVC

Swing: MVC, Datamodeller, JTable, evt. JTree

Evt.: Swing: Præsentations- og kontrol del

Læsning: VP 6



# Swing-demo

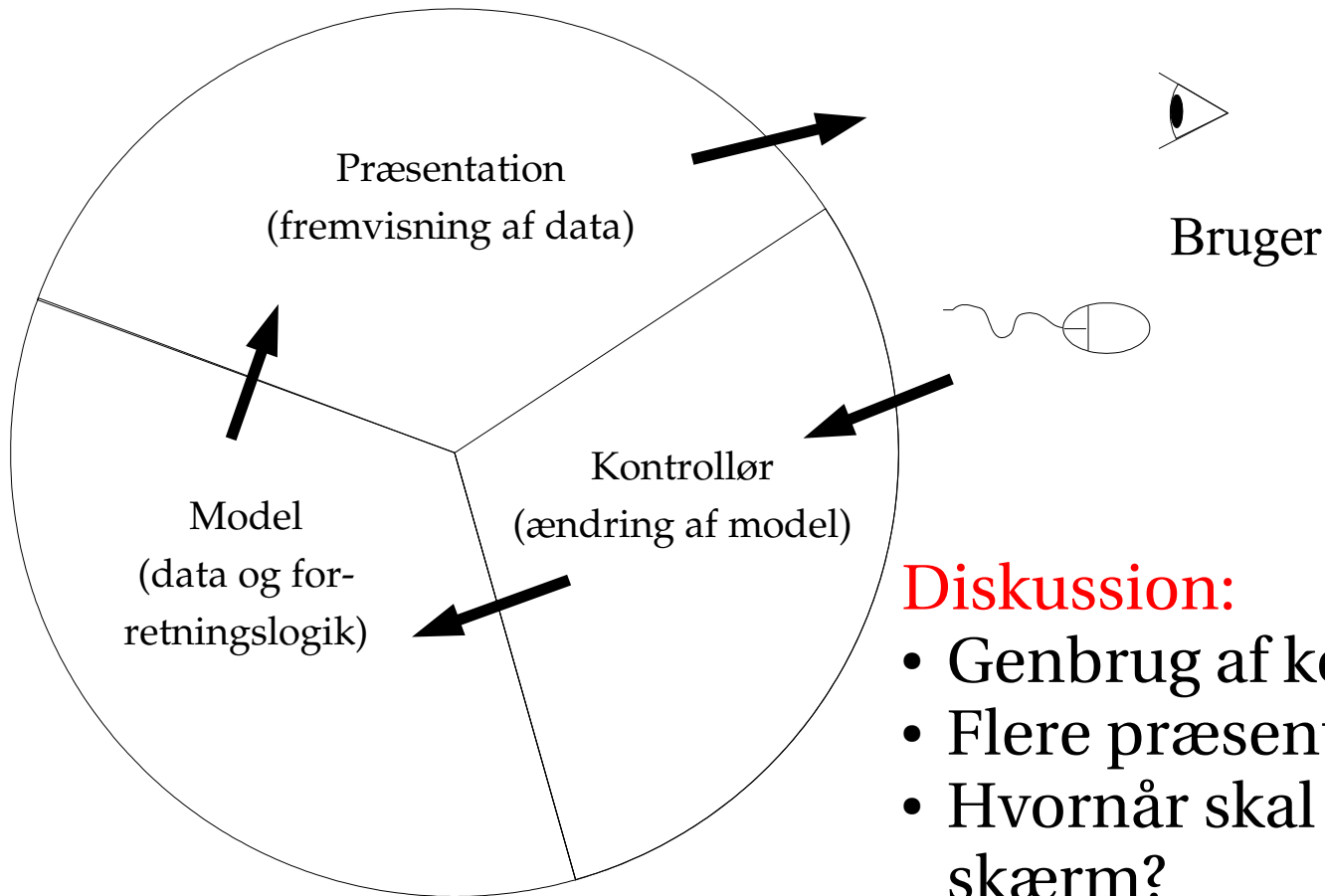
The screenshot shows the SwingSet application window with the following components:

- Menu Bar:** File, Look & Feel, Themes, Tool Tips
- Toolbar:** Contains icons for file operations, text editing, and window management.
- Table Demo Tab:** Active tab showing configuration options for a table.
- Configuration Options:**
  - Reordering allowed
  - Horiz. Lines
  - Vert. Lines
  - Column selection
  - Row selection
  - Inter-cell spacing:
  - Row height:
  - Selection mode: **Multiple ranges** (dropdown)
  - Autosize mode: **Subsequent columns** (dropdown)
- Table:** A table with 6 columns and 6 rows. The first row is highlighted in blue, and the second row is highlighted in red.

First Name	Last Name	Favorite Color	Favorite Movie	Favorite Number	Favorite Food
Michael	Martak	Green	City of Lost Children	3	
David	Mendenhall	Forest Green	Schindler's List	7	
Phil	Milne	Suspect Pink	Withnail & I	3	
Lynn	Monsanto	Cyber Green	Das Boot	52	
Hans	Muller	Rust Red	Eraserhead	0	
Joshua	Outwater	Blue	Labyrinth	3	

**Prøv til øvelserne:** Demo i `jdk1.4/demo/jfc/SwingSet2/SwingSet2.jar`

# Model-View-Controller



## Diskussion:

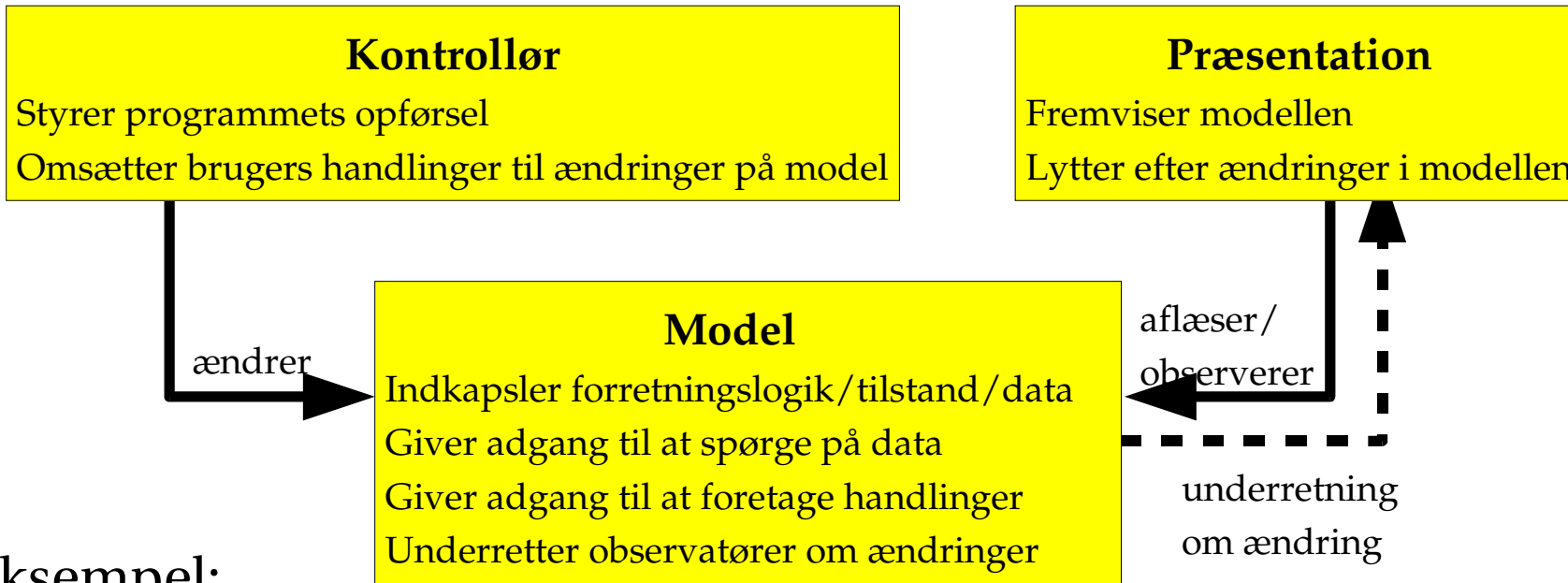
- Genbrug af kode?
- Flere præsentationer af samme data?
- Hvornår skal præsentation opdatere skærm?

Muligheder for opdatering af præsentation:

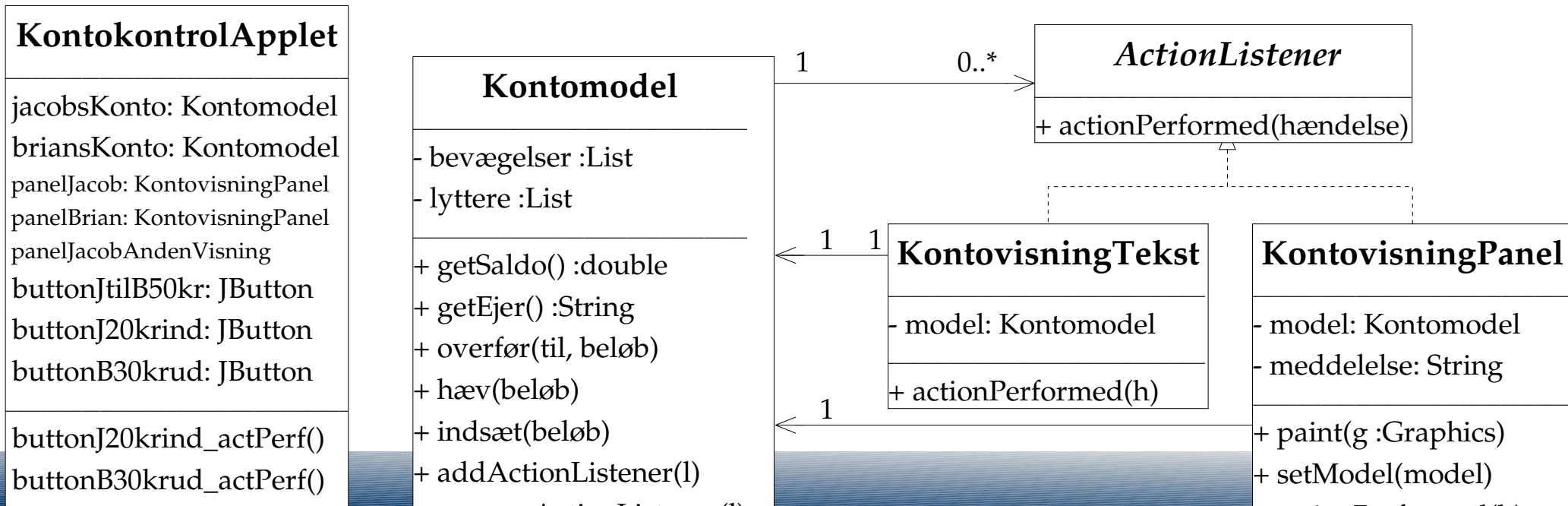
- A - Præsentationer undersøger modellen
- B - Kontrol del underretter præsentationer
- C - Modellen underretter præsentationer



# C - Model underretter præsentation



Eksempel:





# Swing-komponenter og MVC



- Datamodellen er skilt ud fra komponenten.
- Programmøren kan selv bestemme hvilket objekt der skal være datamodel



# Swing og MVC



Alle Swing-komponenter har deres 'datamodel' skilt ud i et separat objekt  
Hvilken type model (klasse) der anvendes afhænger af komponenten

I JTextField, JTextArea og de andre tekstredigeringsfelter er datamodellen et objekt af typen Document, og læses hhv. sættes med metoderne getDocument() og setDocument().  
Har man f.eks. to tekstfelter...

```
JTextField jTextField1 = new JTextField();  
JTextField jTextField2 = new JTextField();
```

kan man få dem til at redigere *i den samme tekst* ved at sætte den ene til at bruge den andens datamodel:

```
Document datamodel = jTextField1.getDocument();  
jTextField2.setDocument( datamodel );
```

**Øvelse:** Lav program med to tekstfelter, der deler samme Document-objekt.



# Datamodellen i jTable



```
import javax.swing.table.*;

class DenLilleTabel extends AbstractTableModel
{
    public int getColumnCount()    { return 10; }
    public int getRowCount()      { return 10;}
    public Object getValueAt(int r, int k) {
        return new Integer((r+1)*(k+1));
    }
}

public class Frame1 extends JFrame {
   .JTable jTable1 = new.JTable();
    ...

    DenLilleTabel dlt=new DenLilleTabel();

    public Frame1() {
        jbInit();
        jTable1.setModel( dlt );
    }
    ...
}
```

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

# Præsentation i JTable

Navn	Yndlingsfarve	Yndlingstal	Yndlingsret	Enlig?	Sexet?
Hans Petersen	farve	44.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>
Jacob Sørens...	farve	42.0		true	<input checked="" type="checkbox"/>
	farve			true	

```
public class Farvepraesentation implements TableCellRenderer
{
    JLabel jLabel = new JLabel();           // genbrug den samme komponent

    public Component getTableCellRendererComponent(JTable tabel, Object værdi,
        boolean erValgt, boolean harFokus, int række, int kolonne)
    {
        if (værdi instanceof Color)        // er værdien af type Double?
        {
            Color f = (Color) værdi;
            jLabel.setBackground(f);         // hele baggrunden viser farven
            jLabel.setForeground(f.darker()); // tekstfarven er lidt mørkere
            jLabel.setOpaque(true);         // uigennemsigtig så baggrunden kan ses
            jLabel.setText("farve");        // vis bare en eller anden tekst
        } else {
            jLabel.setBackground(Color.white); // er objektet ikke en farve, så vis
            jLabel.setForeground(Color.black); // det som toString() giver
            jLabel.setText(værdi.toString());
        }
        return jLabel;                       // returnér komponenten der skal vises
    }
}
```



# Datamodellen i JTree

```
public class MinTreeModel implements TreeModel
{
    public Object getRoot() { return "rod"; }

    public int getChildCount(Object parent) { return 4; }

    public Object getChild(Object parent, int index) {
        return "hej"+index;
    }

    public boolean isLeaf(Object node) { return false; }

    public void valueForPathChanged(TreePath sti, Object nyV) {}
    public int getIndexOfChild(Object far, Object barn){return 0;}
    public void addTreeModelListener(TreeModelListener l) {}
    public void removeTreeModelListener(TreeModelListener l) {}
}
```

