



# IN1010 uke 12

Gruppe 4

# Agenda

- Litt repetisjon av GUI (AWT+Swing)
- Layouts
  - BorderLayout
  - GridLayout
- Bestemme utseende på komponenter
  - Fonts, farger, rammer
  - Størrelse
- MVC (Model - View - Controller)
- Jobbe med oppgaver/trix/oblig!

# Litt repetisjon av forrige uke



# Vanlige komponenter

- Tekst → JLabel
- Tekst inputboks → JTextField
- Knapp → JButton
- Tegneflate → JPanel
- Vindu for programmet → JFrame

# Setup for hvert GUI-program

Start med:

```
try {  
    UIManager.setLookAndFeel(  
        UIManager.getCrossPlatformLookAndFeelClassName());  
} catch (Exception e) {  
    System.exit(1);  
}  
  
JFrame vindu = new JFrame("Mitt Program");  
vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
JPanel panel = new JPanel();  
vindu.add(panel);
```

# Setup for hvert GUI-program

Avslutt med:

```
vindu.pack();  
vindu.setVisible(true);
```

# Layouts



# Layouts: Hvordan ting plasseres på en JPanel

- `FlowLayout` (Standard)
- `BorderLayout`
- `GridLayout`



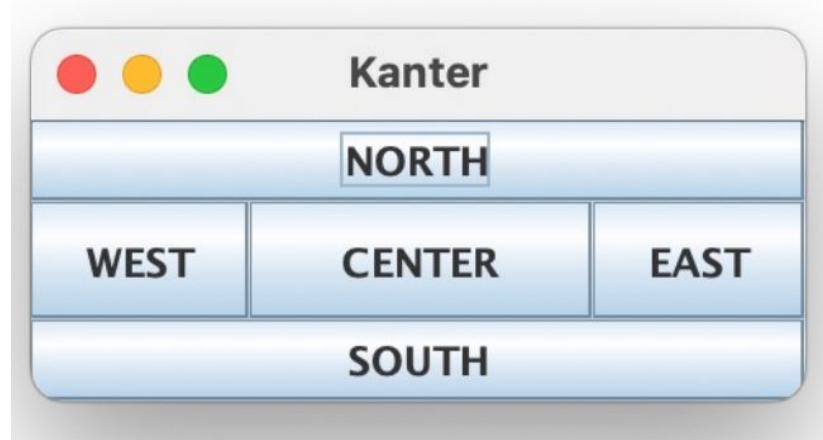
# FlowLayout (Standard)

- Komponentene legges etter hverandre for å få plass inni vinduet
- Kommer i rekkefølgen de blir addet til JPanel



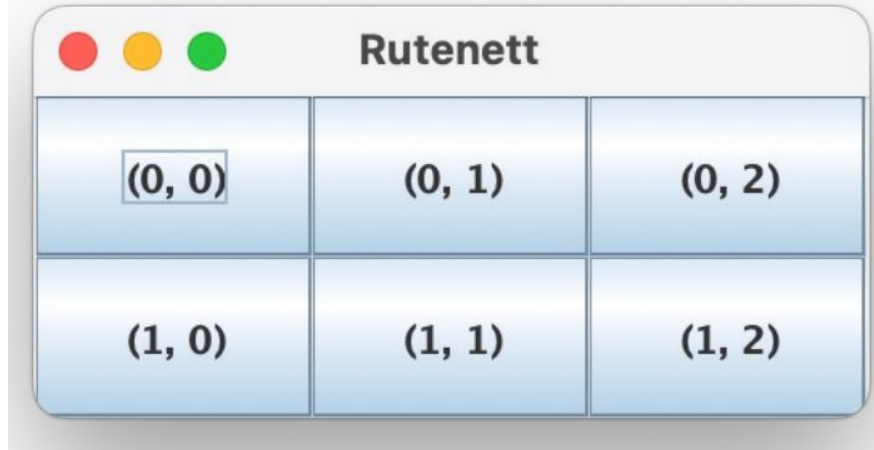
# BorderLayout

- Velg å plassere noe i enten *SOUTH*, *NORTH*, *WEST*, *EAST* eller *CENTER*



# GridLayout

- Lager rutenett med valgt antall rader og kolonner
- Ting legges til i neste ledige rute i rekkefølgen de blir add til JPanel



# Kodeeksempel

*Ulike layouts for 5  
knapper*



# Kodeeksempel

*Binærtall converter med  
penere layout*



# Bestemme utseende på komponenter



# JComponent

- JLabel, JButton osv. arver alle fra samme klasse, *JComponent*

→ Mye av metodene er de samme

Fin å printe ut til eksamen! *(Bilde fra forelesningen uke 12)*

**ActionEvent**

**interface ActionListener**

actionPerformed()

**BorderFactory**

createLineBorder(Color c)

**BorderLayout** ext **LayoutManager**

String CENTER, EAST, NORTH, SOUTH, WEST

**Color**

Color BLACK, BLUE, ...

**Dimension**

**FlowLayout** ext **LayoutManager**

**Font**

int BOLD, PLAIN

String MONOSPACED, SANS\_SERIF

**GridLayout** ext **LayoutManager**

**JButton** ext **JComponent**

addActionListener (ActionListener a)

**JComponent**

setBackground(Color c)

setFont (Font f)

setForeground(Color c)

setOpaque(boolean b)

setText (String t)

**JFrame**

add (JComponent c)

int EXIT\_ON\_CLOSE

pack()

setDefaultCloseOperation (int i)

setVisible (boolean b)

**JLabel** ext **JComponent**

**JPanel** ext **JComponent**

add (JComponent c)

setLayout(LayoutManager m)

**interface LayoutManager**



# Endre font/tekststørrelse, endre farge

- Endre font: `.setFont(new Font(...))`

```
new Font(Font.MONOSPACED, Font.PLAIN, 20)
new Font(Font.MONOSPACED, Font.PLAIN, 30)
new Font(Font.MONOSPACED, Font.BOLD, 30)
new Font(Font.SANS_SERIF, Font.PLAIN, 30)
new Font(Font.SANS_SERIF, Font.BOLD, 30)
```

- Endre farge:
  - `.setForeground(Color.BLUE)`
  - eller med RGB-fargekode:  
`.setForeground(new Color(218, 41, 28))`

# Endre border



- Svart ramme:  
`.setBorder(BorderFactory.createLineBorder(Color.BLACK))`
- Kan også lage “usynlig ramme” (padding), f.eks:  
`.setBorder(BorderFactory.createEmptyBorder(10, 20, 10, 20))`

# Endre størrelse

- Eksempel:

```
knapp.setPreferredSize(new Dimension(100,50))
```

(Dimension bestemmes av lengde, høyde)

# **MVC**

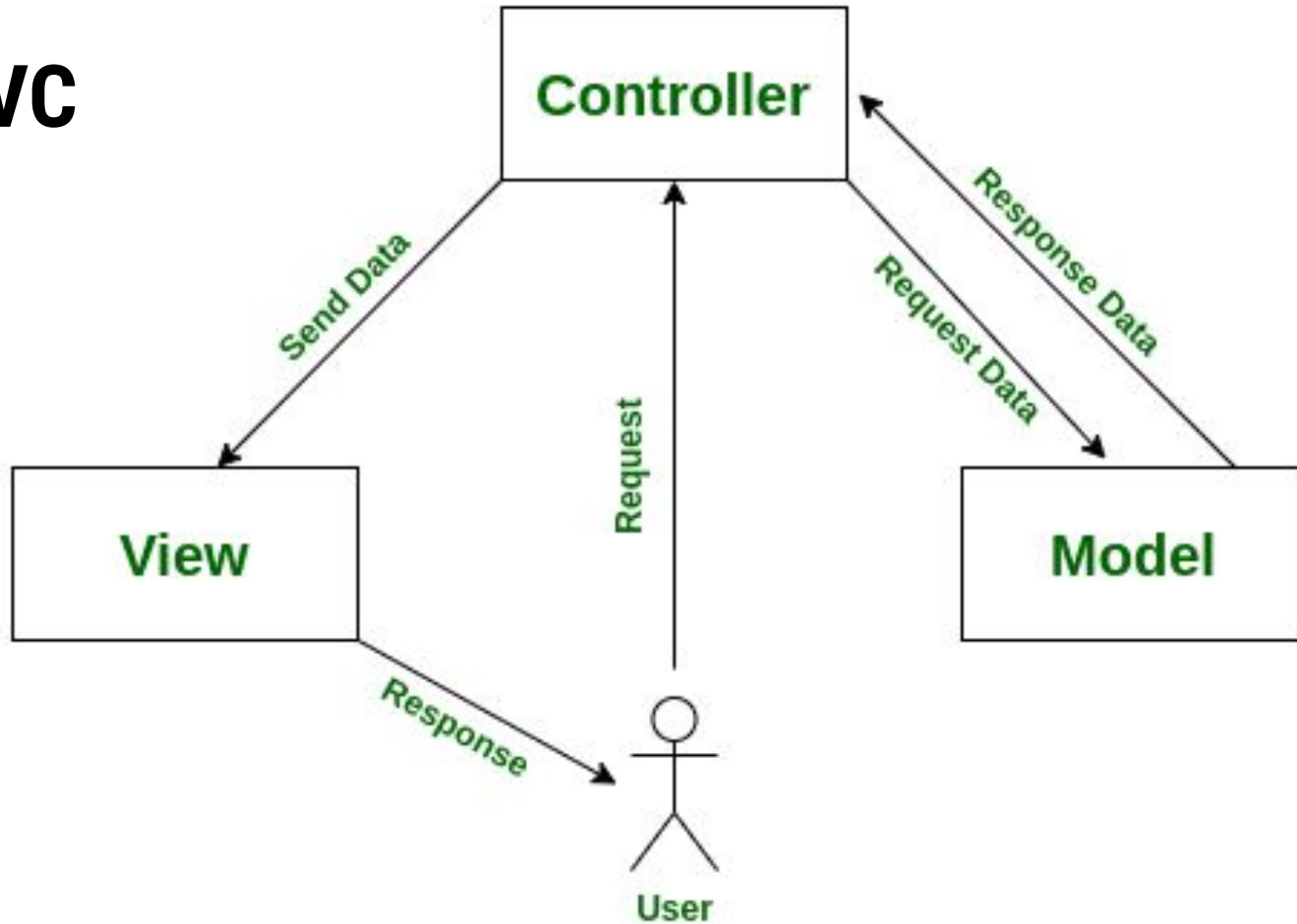
## **(Model - View - Controller)**



# MVC

- Hvordan koble sammen det vi ser (GUI) og logikken bak programmet på strukturert måte?  
→ MVC
- Model: selve logikken bak programmet
- View: Det vi ser (GUI)
- Controller: Mellomledd mellom Model og View, styrer alt

# MVC



# Jobb med oppgaver!

IN1010 Emnesiden → Grupper →  
Gruppe 4 → Uke12

