Mer om GUI

Repetisjon

Oppbygning av brukergrensesnitt

- **JFrame**
  - Vinduet brukeren ser
  - Holder på alle komponenter
- **JPanel**
  - Holder på komponenter
- **Komponenter**
  - Visuelle elementer i vinduet
  - JLabel, JButton, JTextField…

```
class FrameEksempel {
    public static void main(String[] args) {
        JFrame vindu = new JFrame("Dette er et vindu!”);
        vindu.setPreferredSize(new Dimension(400, 400));

        vindu.pack();
        vindu.setLocationRelativeTo(null);
        vindu.setVisible(true); // Starter en ny Swing-tråd
    }
}
```

Oppbygning av brukergrensesnitt

- **JFrame**
  - Vinduet brukeren ser
  - Holder på alle komponenter
- **JPanel**
  - Holder på komponenter
- **Komponenter**
  - Visuelle elementer i vinduet
  - JLabel, JButton, JTextField…

```
class FrameEksempel {
    public static void main(String[] args) {
        JFrame vindu = new JFrame("Dette er et vindu!");
        vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        vindu.setPreferredSize(new Dimension(400, 400));

        try {
            UIManager.setLookAndFeel(
                UIManager.getCrossPlatformLookAndFeelClassName()
            );
        } catch (Exception e) {
            e.printStackTrace();
        }

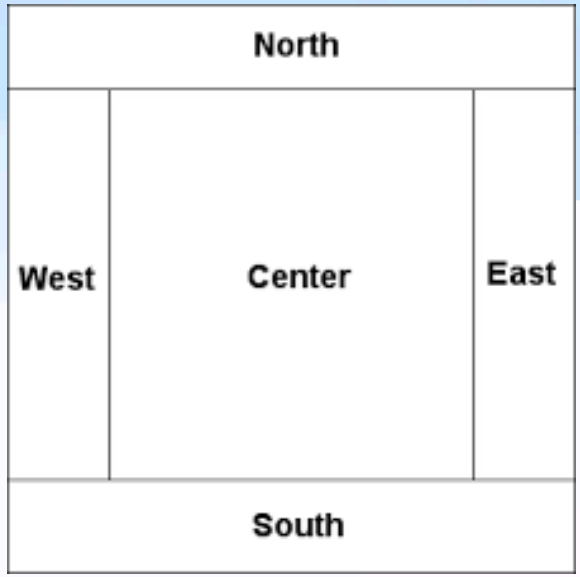
        JPanel panel = new JPanel();
        panel.setBorder(new EmptyBorder(40, 40, 40, 40));
        panel.setLayout(new GridLayout(2, 1));

        JLabel tekst = new JLabel("Dette er en JLabel", SwingConstants.CENTER);
        JButton knapp = new JButton("Trykk meg!");

        panel.add(tekst); panel.add(knapp);
        vindu.add(panel);

        vindu.pack();
        vindu.setLocationRelativeTo(null);
        vindu.setVisible(true); // Starter en ny Swing-tråd
    }
}
```

Layout



```java
public class LayoutEksempel {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Inner Class Layout Demo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setLayout(new BorderLayout(8, 8)); // Tallene gir spacing mellom paneler

        frame.add(new ColorPanel("Header", new Color(255, 102, 102)), BorderLayout.NORTH);
        frame.add(new ColorPanel("Venstre", new Color(102, 255, 102)), BorderLayout.WEST);
        frame.add(new ColorPanel("Hoved panel", new Color(102, 102, 255)),
BorderLayout.CENTER);
        frame.add(new ColorPanel("Høyre", new Color(255, 255, 102)), BorderLayout.EAST);
        frame.add(new ColorPanel("Footer", new Color(102, 255, 255)), BorderLayout.SOUTH);

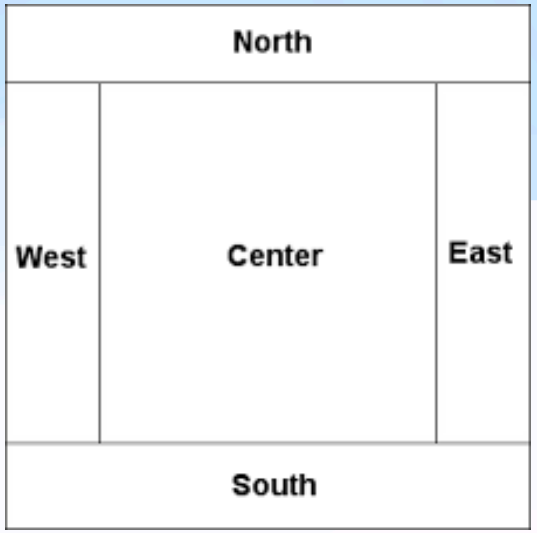
        frame.pack();
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }

    public static class ColorPanel extends JPanel {

        public ColorPanel(String title, Color backColor) {
            this.setBackground(backColor);
            this.setPreferredSize(new Dimension(150, 80));
            this.setLayout(new GridBagLayout());

            JLabel label = new JLabel(title);
            label.setForeground(Color.BLACK);
            this.add(label);

            this.setBorder(BorderFactory.createLineBorder(Color.DARK_GRAY, 1));
        }
```



Hendelseshåndtering

- Du utføre handlinger ved knappetrykk med bruk av ***.addActionListener()***
  - Metoden tar inn et ActionListener-grensesnitt
  - Som har metoden ***public void actionPerformed(ActionEvent e);***

Hendelseshåndtering

- Du utføre handlinger ved knappetrykk med bruk av ***.addActionListener()***
  - Metoden tar inn et ActionListener-grensesnitt
  - Som har metoden ***public void actionPerformed(ActionEvent e);***

```
public static void main(String[] args) {
        JFrame vindu = new JFrame();
        vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        vindu.setPreferredSize(new Dimension(200, 200));

        JPanel panel = new JPanel();
        panel.setLayout(new BorderLayout());
        vindu.add(panel);

        JButton knapp = new JButton("Trykk meg!");
        class HåndterKnappTrykk implements ActionListener {

            @Override
            public void actionPerformed (ActionEvent e) {
                System.out.println("Knappen trykkes!");
            }
        }
        knapp.addActionListener(new HåndterKnappTrykk());
        vindu.add(knapp);
        vindu.pack();
        vindu.setLocationRelativeTo(null);
        vindu.setVisible(true);
    }
```

Mer om hendelseshåndtering

Model, View, Controller (MVC)

- Separasjon av logikk til egne klasser
- View
  - Hva brukeren ser og interagerer med
- Model
  - Inneholder data og logikk
- Controller
  - Laget mellom *model* og *View*
  - Sørger for at View holdes oppdatert

Oppgaver

(Anbefaler å gjøre dem 🤓)

https://www.uio.no/studier/emner/matnat/ifi/IN1010/v26/grupper/gruppe-12-inf-ars/uke-13/oppgaver---uke-13.pdf

