

# Velkommen!



Johanna  
johannph på mattermost  
[johannph@uio.no](mailto:johannph@uio.no) på mail!

# Kort: Praktisk informasjon

- Undervisningstilbud
  - <https://www.uio.no/studier/emner/matnat/ifi/IN1010/v21/undervisningstilbud/>
  - Jeg har konkrete spørsmål/problemer med min kode -> Labtime!
  - Jeg vil ha mer liveprogrammering -> Plenumstime!
  - Jeg vil jobbe med andre (og kanskje en kjapp recap av forelesning) -> Gruppetime!
  - Jeg vil ha en recap av de vanskeligste konseptene fra forelesning -> Repetisjonsgruppe!
- Skriv alt dere lurer på i chatten enten til everybody eller bare til meg 😊
  - Si i fra hvis dere faller av! Enten hvis noe er vanskelig, eller hvis dere glemte å følge med! Bare si i fra!
  - Vi har god tid, så det er masse tid til å gå gjennom ting flere ganger

**Repetisjon forrige uke**

# SOLID

- **Single responsibility principle**  
En klasse bør bare ha ett ansvarsområde, og dette bør være veldefinert
- **Open/closed principle**  
Komponenter bør være åpne for utvidelser, men lukket for modifikasjoner. Eksempel: klasser kontrollerer hva subklasser kan endre (private, protected, final, etc).
- **Liskov substitution principle**  
Objekter i et program bør kunne byttes ut med instanser av subtyper uten at dette endrer programmets korrekthet.
- **Interface segregation principle**  
Mange små grensesnitt er bedre enn store, monolittiske “super-grensesnitt”
- **Dependency inversion principle**  
Komponenter bør avhenge av abstraksjoner, og ikke konkrete implementasjoner

# Enhetstesting

Teste komponenter av koden hver for seg, f.eks. en metode eller en klasse.

Etter enhetstesting kan vi sette komponentene sammen og teste hvordan det fungerer sammen (integrasjonstest) og etterhvert hele systemet (systemtesting). Men i in1010 lærer vi bare litt enhetstesting.

Poenget er å finne feil så tidlig som mulig. Dere har sikkert selv erfart at det er vanskeligere å finne feil dess mer kode det er.

Repetisjon denne uka

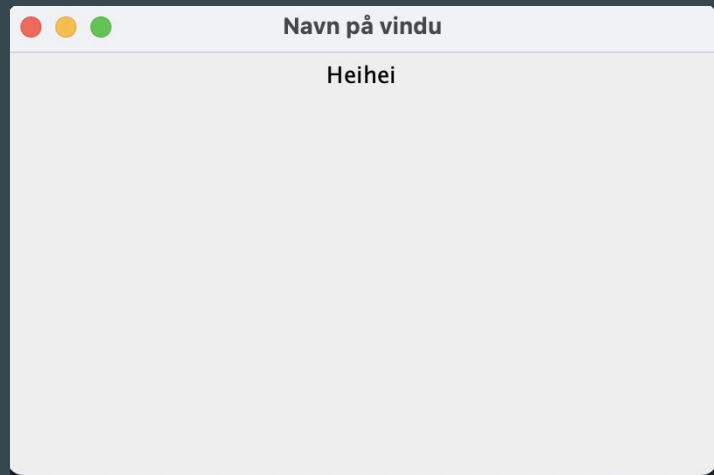
# Import + basics

```
1 import java.awt.*;↵
2 import java.awt.event.*;↵
3 import javax.swing.*;↵
4 ↵
5 class TestGraphic{↵
6     ··· public static void main(String[] args) {↵
7         ··· JFrame vindu = new JFrame("Navn på vindu");↵
8         ··· vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);↵
9         ↵
10        ··· JPanel panel = new JPanel();↵
11        ··· vindu.add(panel);↵
12        ↵
13        ··· vindu.pack();↵
14        ··· vindu.setVisible(true);↵
15    }↵
16 }
```



# Tekst

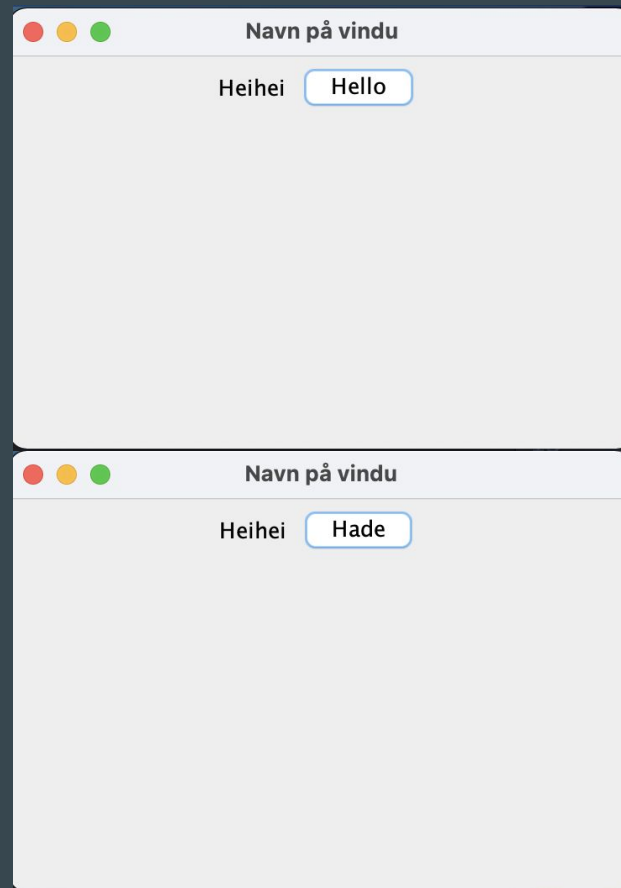
```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 class TestGraphic{
6     public static void main(String[] args) {
7         JFrame vindu = new JFrame("Navn på vindu");
8         vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9
10        JPanel panel = new JPanel();
11        vindu.add(panel);
12
13        JLabel tekst = new JLabel("Heihei");
14        panel.add(tekst);
15
16        vindu.pack();
17        vindu.setVisible(true);
18    }
19 }
```





# Knapper

```
5 class TestGraphic{-
6   public static void main(String[] args) {-
7     -
8     class Knapp extends JButton{-
9       public Knapp(String tekst){-
10        super(tekst);-
11      }-
12    }
13    class KnappAction implements ActionListener{-
14      @Override
15      public void actionPerformed(ActionEvent e){-
16        setText("Hade");-
17      }-
18    }
19    public void initGui(){-
20      addActionListener(new KnappAction());-
21    }-
22  }
23  JFrame vindu = new JFrame("Navn på vindu");
24  vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);-
25  -
26  JPanel panel = new JPanel();-
27  vindu.add(panel);-
28  -
29  JLabel tekst = new JLabel("Heihei");-
30  panel.add(tekst);-
31  -
32  Knapp knapp = new Knapp("Hello");-
33  knapp.initGui();-
34  panel.add(knapp);-
35  -
36  vindu.pack();-
37  vindu.setVisible(true);-
38  -
```

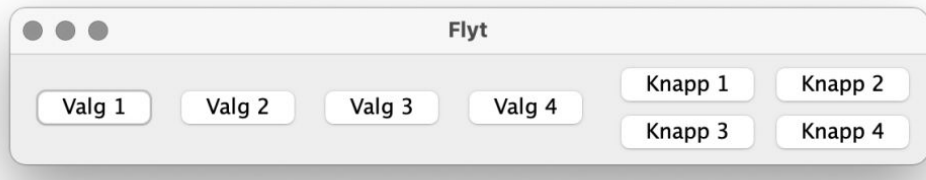


# FlowLayout vs GridLayout

FlowLayout er default, layout for et JPanel kan endres med `.setLayout()`.

```
JPanel flyt = new JPanel();
flyt.add(new JButton("Valg 1"));
flyt.add(new JButton("Valg 2"));
flyt.add(new JButton("Valg 3"));
flyt.add(new JButton("Valg 4"));
panel.add(flyt);

JPanel ruter = new JPanel();
ruter.setLayout(new GridLayout(2,2));
ruter.add(new JButton("Knapp 1"));
ruter.add(new JButton("Knapp 2"));
ruter.add(new JButton("Knapp 3"));
ruter.add(new JButton("Knapp 4"));
panel.add(ruter);
```



# Jobbe selv

Jobb med hva dere vil og rekk opp hånda hvis dere trenger hjelp med noe/har spørsmål så møtes vi i breakoutroom! 😊

