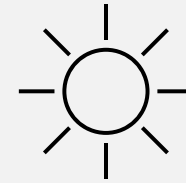


# GRUPPE 8 IN1010 – UKE 11

Sivert Fjeldstad Madsen

# I DAG

- GUI
- Generelt oppsett
- De mest brukte byggeklossene
- JButton og ActionListener
- Oppgaver



# GUI – GRAPHICAL USER INTERFACE

- Frem til nå har den eneste måten å interagere med programmene vi har skrevet vært gjennom terminalen
- Programmer som er tenkt å bli brukt mye har sjelden terminalen som eneste interaksjonspunkt
- Man programmerer gjerne et eget *interaksjonslag* som lar brukeren påvirke programmet
  - Det er dette vi kaller et brukergrensesnitt
- Her vil en bruker kunne få se og interagere med programmet på ulike måter

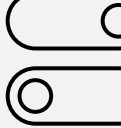


# GUI OG TRÅDER

- Grafiske brukergrensesnitt og tråder er på mange måter uadskillelige
- Hvis hele grensesnittet hadde kjørt på én tråd hadde det ikke vært mulig at flere ting kunne skje samtidig
  - Feks. ville du ikke kunnet trykke på en knapp samtidig som en animasjon ble spilt av
- Hvordan koder vi dette?
- Vi slipper det, heldigvis
- Dette er innebygd i Swing og AWT, og skjer bl.a. via **ActionEvent** og **ActionListener**



# GUI - KODESKIKK

- En lur ting å tenke på når man programmerer større programmer som skal ha et GUI, er å holde det adskilt fra resten av koden 
- Tenk at du skal kunne enkelt bytte mellom flere forskjellige GUIer ved bare å endre hvilken klasse som blir brukt. F.eks:
  - En klasse som gir et grensesnitt basert på **terminalen** (som i oblig 4)
  - En klasse som gir et grensesnitt basert på Swing og AWT
- På denne måten er det enkelt å gjøre endringer på grensesnittet uten å måtte endre masse underliggende logikk, og vice versa
- Dette er en annen form for innkapsling

# DEKLARATIV PROGRAMMERING

- De fleste moderne rammeverk for GUIer bruker det som kalles **deklarativ programmering**
- Fram til nå har vi i all hovedsak drevet med **imperativ** programmering, med noen få smakebiter av **funksjonell** programmering
- Deklarativ programmering går ut på at programmereren (deg) forteller *hva* de vil ha, men ikke i veldig stor grad *hvordan*
- Dette gjør at vi slipper å forholde oss til mange ting, sånn som hvordan man oppretter tråder for de ulike GUI-delene og hvordan man faktisk tenger et vindu på en skjerm
- Det gjør også at vi må godta å ikke ha like god kontroll over alt som vi har blitt vant til



# SWING OG AWT

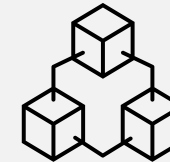
- I INI010 bruker vi to av Javas innebygde biblioteker for GUIer: **Swing** og **AWT**
- Begge bibliotekene tilbyr mange av de samme tingene, og derfor bruker vi *deler* av begge
- Disse bibliotekene er litt utdaterte, men...
  - De er relativt enkle å bruke
  - De fungerer veldig likt som mange nyere biblioteker
  - De gjør alt det vi trenger i INI010



# GENERELT OPPSETT

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

Importerer bibliotekene vi trenger



```
...  
try {  
    UIManager.setLookAndFeel(  
        UIManager.getCrossPlatformLookAndFeelClassName()  
    );  
} catch (Exception e) {  
    System.exit(1);  
}
```

Setter utseendet til å matche operativsystemet (hvis mulig)

```
JFrame vindu = new JFrame("Dette er navnet på vinduet!");  
vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
...  
vindu.pack();  
vindu.setLocationRelativeTo(null);  
vindu.setVisible(true);
```

Oppretter vinduet

Gjør så programmet stopper når vinduet lukkes

Legger alt innhold i vinduet

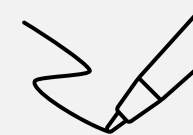
Setter vinduet midt på skjermen

Gjør vinduet synlig



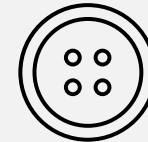
# LEGG TIL ELEMENTER I VINDUET

- Et helt tomt vindu er ikke spesielt interessant
- Vi bruker innebygde klasser for å representere ulike ting:
  - **JPanel** – for å lage tegneflater
  - **JLabel** – for tekst
  - **TextField** og **TextPane** – for tekstfelter
  - **Button** – for knapper
- Elementer legges til en tegneflate med **.add()**
- Tegneflater legges til vinduet (JFrame) med **.add()**
- Mer om **layout** neste uke...



# JBUTTON

- Knapper er en av de vanligste måtene for en bruker å interagere med et program
- Vi må definere hva som skal skje når en knapp blir trykket på, og dette gjør vi med indre klasser
  - Ganske likt som arbeiderklassene når vi programmerer med tråder
- «Hendelses»-klassen vår må implementere interfacet **ActionListener**
  - Som krever at vi skriver metoden **public void actionPerformed (ActionEvent e)**
  - Denne metoden definerer hva som skjer når knappen blir trykket på



# OPPSUMMERING

- GUI i Java gir flere muligheter for brukerinteraksjon!
- Lag et ark med de vanligste konstruksjonene i Swing og AWT, siden disse brukes hele tiden
  - Det kan være lurt å ha til f.eks. eksamen...
- Ukas tips:
  - Sjekk ut <https://userinyerface.com/> for tips til hvordan å *ikke* designe brukergrensesnitt

