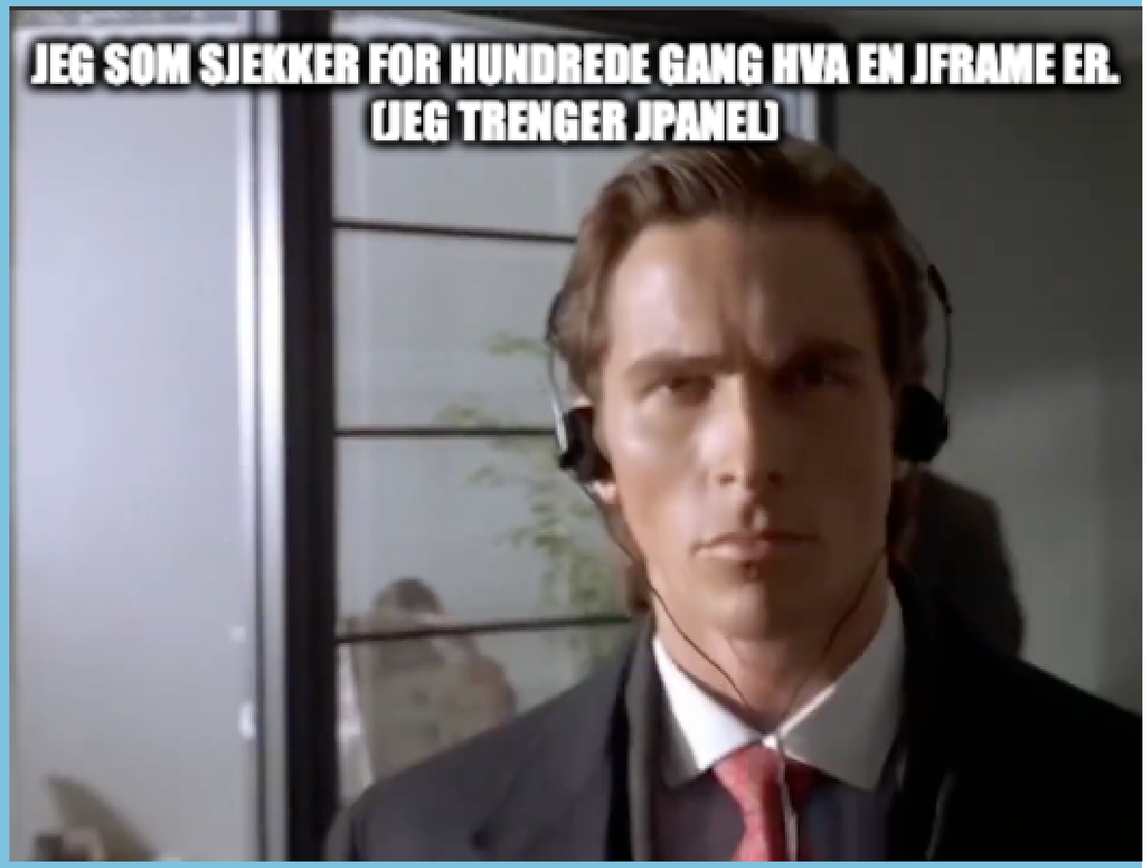
Uke 13 - GUI 1

Gruppe 5 - jonasbny@ifi.uio.no

Agenda



Hva er GUI

Hvorfor er det nyttig

GUI i IN1010

GUI og tråder

Swing og Awt

Oppsett

eksempler på objekter
(JFrame, JPanel, JLabel, Jbutton)

ukas shoutout

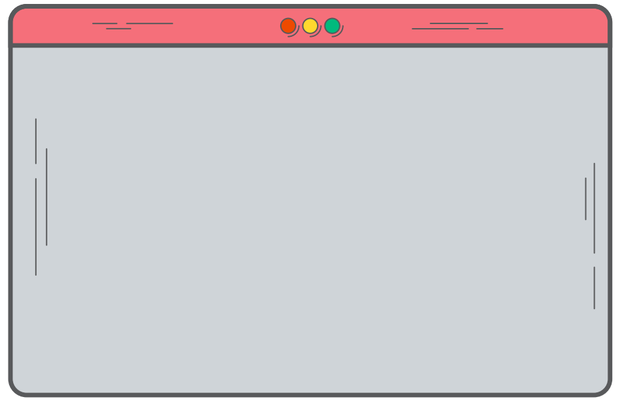
Hva er GUI

GUI står for Graphical User Interface, på norsk grafisk brukergrensesnitt

Programmer som er tenkt å bli brukt mye har sjeldent terminalen som eneste interaksjonspunkt

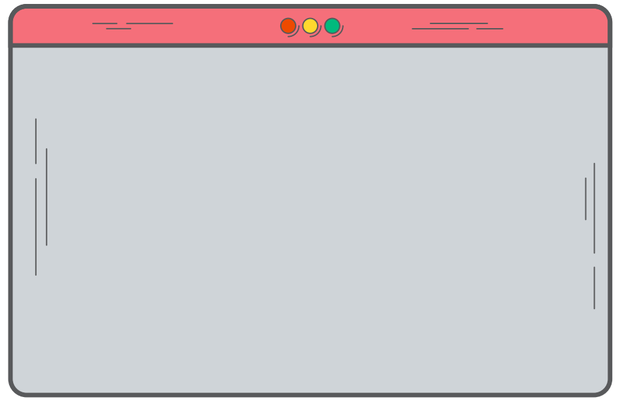
Man programmerer gjerne et eget interasjonslag som lar brukeren påvirke hva som skjer i programmet.

Dette kaller vi et brukergrensesnitt



Hvorfor er det nyttig?

Det forenkler interaksjonen mellom menneske og maskin



- Mer intuitivt å bruke
- Færre muligheter for brukerfeil
- Visuelt mer tiltalende

Noen ulemper:

- Mer komplisert å programmere
- Få biblioteker fungerer for både Linux, Mac og Windows

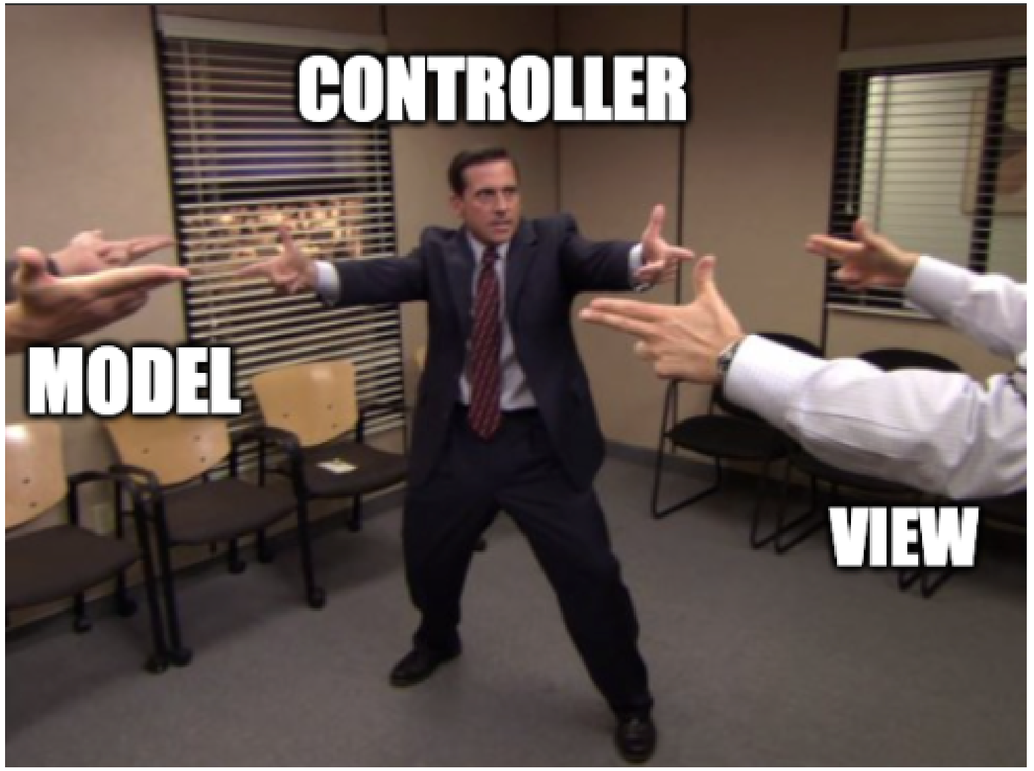
Hvorfor GUI i IN1010?

Allmennkunnskap for programmerere

Bra eksempel på bruk a OOP

Viser en ny programmeringstankegang:
Hendelsesorientert programmering

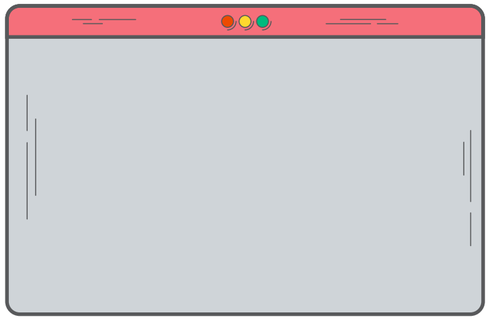
Viser programmeringsmønsteret MVC
model-view-controller



Diskusjonsoppgave

Hva er forskjellen på et bra GUI og et dårlig et?

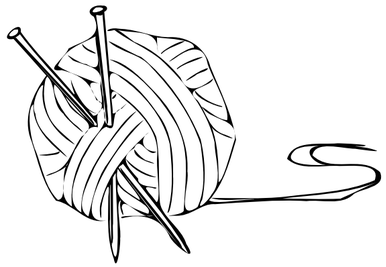
- Hva er de viktigste målene for et GUI?
- Hva er tegn på et dårlig GUI?
- Hva er noen bra eksempler på bra GUI?



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



Hvordan koder man det?

Dette er innebygd i bibliotekene Swing og Awt som vi skal bruke via ting som **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 bytte mellom flere forskjellige GUIer ved å bare endre hvilken klasse som blir brukt

- En klasse som gir et grensesnitt basert på terminalen (som innlevering 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 på den underliggende logikken



Swing og Awt

I IN1010 bruker vi to av Javas innebygde biblioteker for GUIer: Swing og AWT

Awt - Abstract Window ToolKit

- enkelt grensesnitt men gammelt (1995)



Swing (2007)

- mer avansert og generelt system for vinduer, trykkknapper etc bygget oppå Awt

I IN1010 bruker vi Awt+Swing selv om JavaFX er mer moderne

Generelt oppsett

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

Importerer bibliotekene vi trenger

```
try{ //setter UI lik uansett hva slags operativsystem man bruker
    UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
} catch(Exception e){
    System.exit(status:1);
}
```

Setter standard Swing-utseende

```
//setter standard verdier til vinduet
window = new JFrame(title:"Snake");
window.setSize(width:1200, height:600);
window.setLayout(new BorderLayout());
window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //
```

Oppretter vinduet

Gjør så programmet stopper når vinduet lukkes

```
window.setLocationRelativeTo(c:null);
window.setVisible(b:true);
```

Setter vinduet midt på skjermen

Gjør vinduet synlig

Eksempler på objekter

Et helt tomt vindu er ikke særlig interessant

Vi bruker innebygde klasser for å lage ulike ting

- JPanel - for å lage tegneflater
- JLabel - for å lage tekst
- JButton - for å lage knapper



Vi kan lage egne klasser som extender disse klassene hvis vi ønsker visse egenskaper

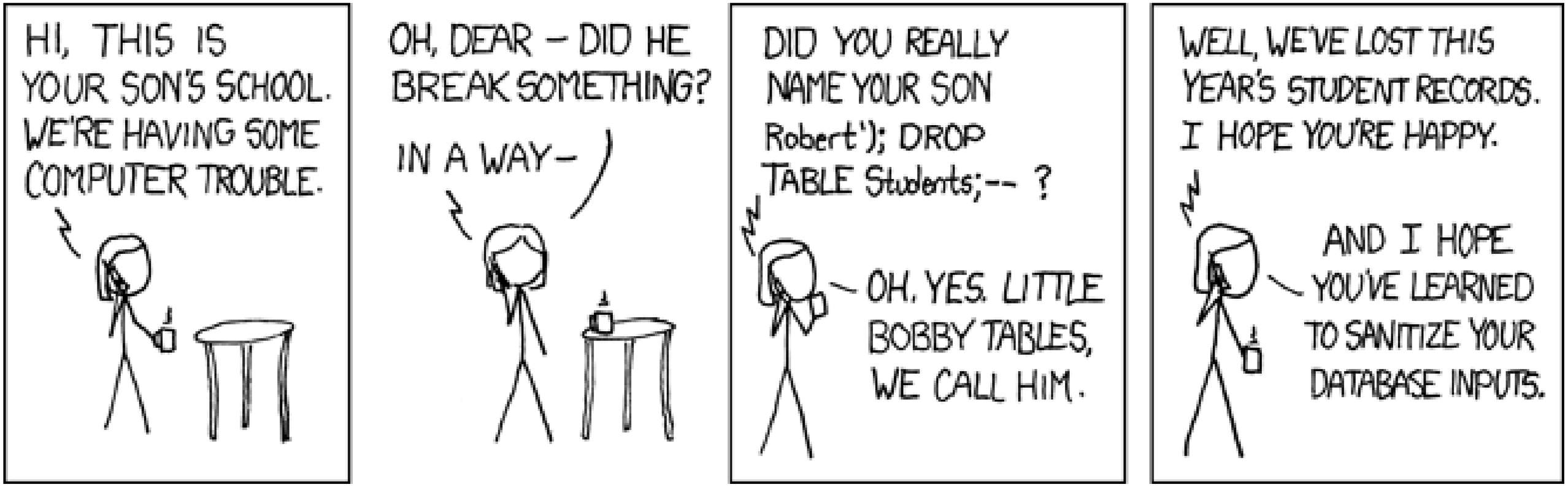
For eksempel extende JLabel for å kunne bruke mouselistener

Vanlig å ha et cheatsheet for å huske de forskjellige tingene

Diskusjonsoppgave

Hvorfor er det viktig å skille GUI kode fra logikken bak programmet?

- Hvilke utfordringer kan oppstå når man ikke skiller de?
- Har du sett eksempler på dårlig separasjon?
- Hvilken fordeler får vi ved å skille disse?


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.


ZU 0666', 0, 0); DROP DATABASE TABLICE;--

ukas shoutout

Soundcloud

For de som løper

Hør på en konsert setlist kontinuerlig

min go to:
Minna-no-kimochi | Boiler Room: Tokyo



Livekode

Oppgaver

