App-programmering med FXML

God måte å få inn objektorientert tankegang

Oppsett

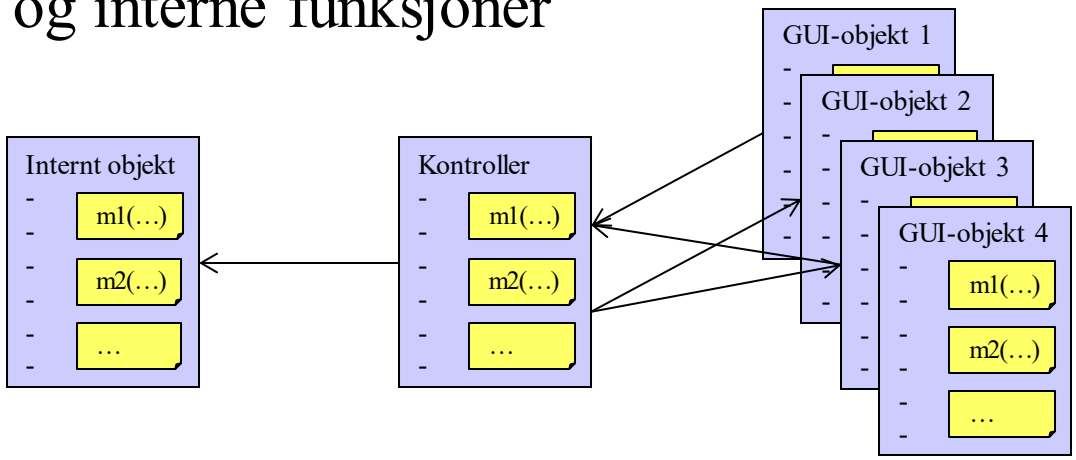
- https://www.ntnu.no/wiki/display/tdt4100/Oppsett+av+Scenebuilder+i+VSCode
- Dette er uansett ikke noe bråhast – det kommer med om det senere! (ØF)
- Vi bruker det også for å vise hva som kommer (prosjekt), og for å gi dere mer forståelse av hvordan objekter lages og virker sammen.

Første gang jeg laget noe på denne måten tenkte jeg…

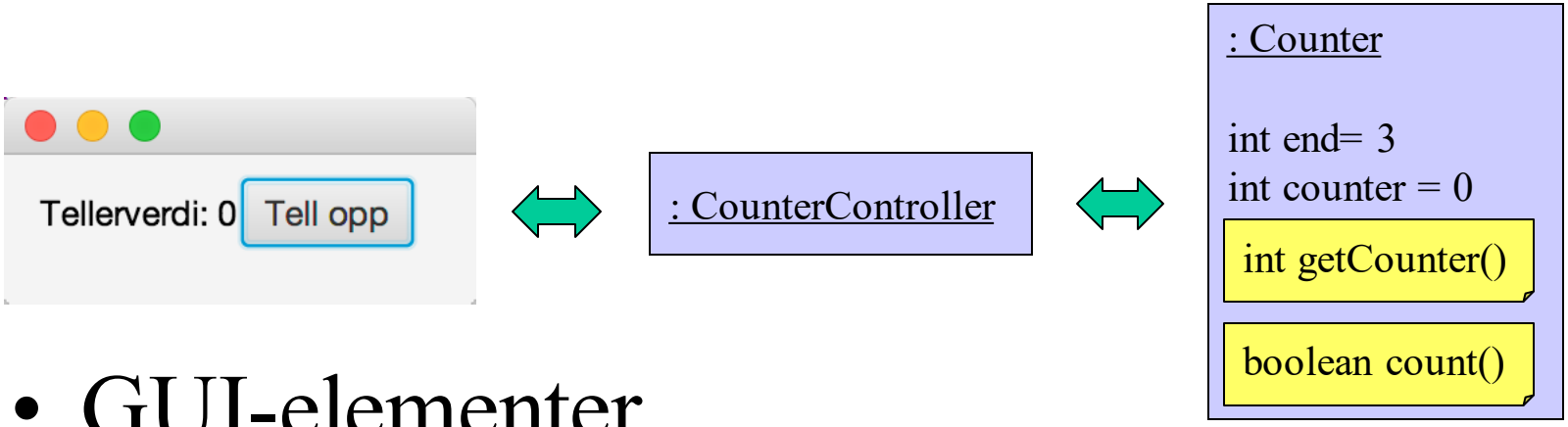
- WTF?!?!1!

App-er

- (G)UI (ratt, girspak, dashbord, ...)
  - GUI-objekter – interaktive og rent grafiske
  - kontroller-objekter - koordinerer (G)UI-objekter og interne objekter
- Intern tilstand (motor)
  - objekter med app-logikk og app-data, altså intern tilstand og interne funksjoner



App for Counter



- GUI-elementer
  - tekst (grafikk) viser nåværende tellerverdi
  - knapp brukes for å telle opp
- Kontroller-logikken
  - reagerer på knapp ved å telle opp
  - oppdaterer tekst
- Intern tilstand - Counter

GUI-elementer og FXML

- GUI-elementer
  - ren grafikk – tekst, streker, rektangler og andre figurer
  - interaktive elementer – tekstfelt, knapper, lister osv.
  - grupperingselement – layout, f.eks. horisontalt, rutenett
- FXML
  - opprettes kankje enklest ved å kopiere inn en eksisterende…
  - redigeres som tekst i VS Code
  - grafisk med SceneBuilder
  - kan kobles sammen, men ingen integrasjon…

Kontroller og FXML

- Kjøring av FXML
  - det opprettes JavaFX-objekter tilsvarende hvert element (tag)
  - hvert JavaFX-objekt sin oppførsel styres av attributtene
  - men hva med koblingen til den app-spesifikke interne tilstanden? det er kontrolleren sin oppgave!
- Kontrolleren må kunne
  - reagere på brukerinteraksjon, f.eks. knappetrykk
  - oppdatere GUI-objektene, og trenger derfor referanser til dem (altså de som skal kunne oppdateres)
  - FXML har spesielle koder for rigging av kontrolleren

Rigging av kontroller

- Kontroller-klasse
  - **fx:controller**-attributt angir kontroller-klasse
  - kontroller-objekt opprettes automagisk
- Trigging ved brukerinteraksjon:
  - FXML-attributt på element: **onXYZ="#metode"**
  - Java-metode i kontroller: **@FXML void metode() { ... }**
- Referanse til JavaFX-objekt
  - FXML-attributt på element: **fx:id="attributt"**
  - Java-attributt i kontroller: **@FXML Type attributt;**
  - settes automagisk etter opprettelse av kontroller
- Initialisering
  - Java-metode i kontroller: **@FXML void initialize() { ... }**
  - kalles automagisk etter at referansene er satt

Kafe.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.text.Text?>

<HBox xmlns="http://javafx.com/javafx/11.0.1"   xmlns:fx="http://javafx.com/fxml/1"
fx:controller="uke5.kaffe.KafeController">
    <Button onAction="#handleServer" text="Server"></Button>
    <Button onAction="#handleNew" text="Ny kunde"></Button>
    <Text fx:id="output" text="Info"></Text>
</HBox>
```

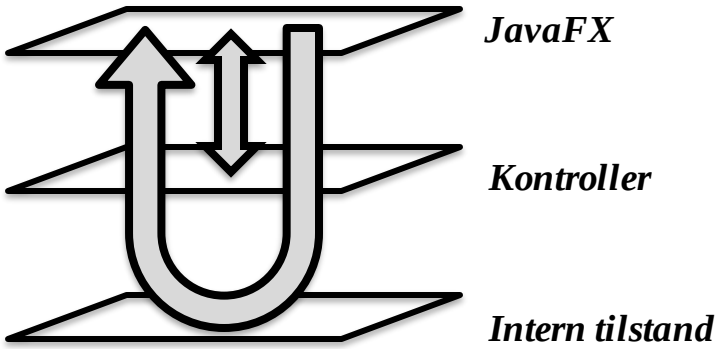
KafeApp

```
package uke5.kaffe;

import javafx.fxml.FXML;
import javafx.scene.control.TextField;
import javafx.scene.text.Text;

public class KafeController {

    MiniKafe mk = new MiniKafe(); // Men status vil ikke synes siden vi ikke oppdaterer
    Person p;

    @FXML TextField input;
    @FXML Text output;

    @FXML
    void handleNew() {
        p = new Person("Even");
        output.setText("Har "+p.getNavn()+" fått nok kaffe: "+p.nokKaffe());
    }

    void updateOutput() {
        String outputString = "Ferdig? ";
        if (p.nokKaffe()) {
            outputString += " Ja!";
        } else {
            outputString += " Nei";
        }
        output.setText(outputString);
    }

    @FXML
    void handleServer() {
        p.drikkKaffe();
        updateOutput();
    }
}
```

Initialisering av intern tilstand

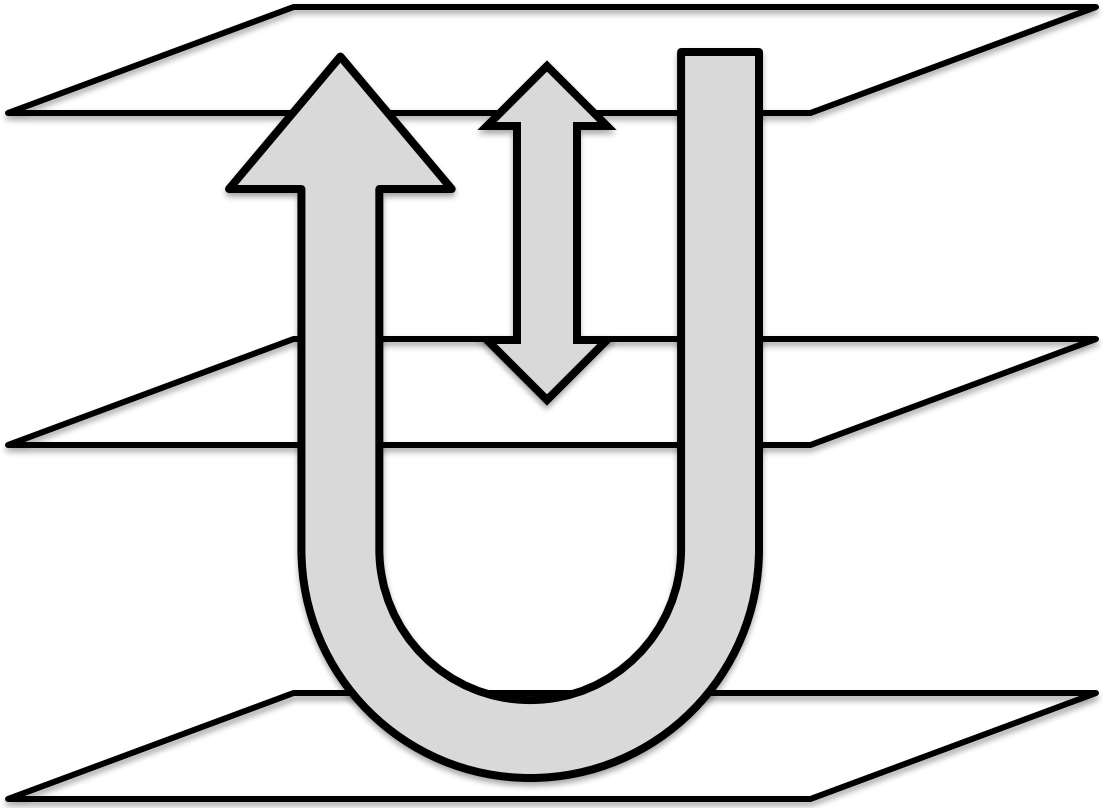
- Konstruktør
  - (tom) konstruktør brukes når kontroller-objekt opprettes
  - kalles for tidlig til at GUI-objektene kan nås og brukes
  - kan brukes til initialisering av intern tilstand
- **@FXML void initialize() { ... }**
  - kalles automagisk etter at GUI er satt opp, merk at dette er spesifikt for FXML-applikasjoner
  - kan brukes til initialisering av intern tilstand
  - brukes til å oppdatere GUI-objektene, slik at de stemmer med den initielle interne tilstand
- Vi kan med andre ord initialisere objektene, men ikke vise tilstanden i GUI uten initialize

Kontroller-logikk



- Reagere på brukerinteraksjon
  - typisk handleXYZ-metoder
  - henter ut innfylte data med get-metoder, konverterer evt. til ønskede typer
  - kaller metoder på interne objekter
- Oppdatere
  - leser ut ny tilstand med get-metoder
  - oppdaterer GUI-objekter vha. set-metoder

Tenk på det som en U og en I
= UI (User Interface)

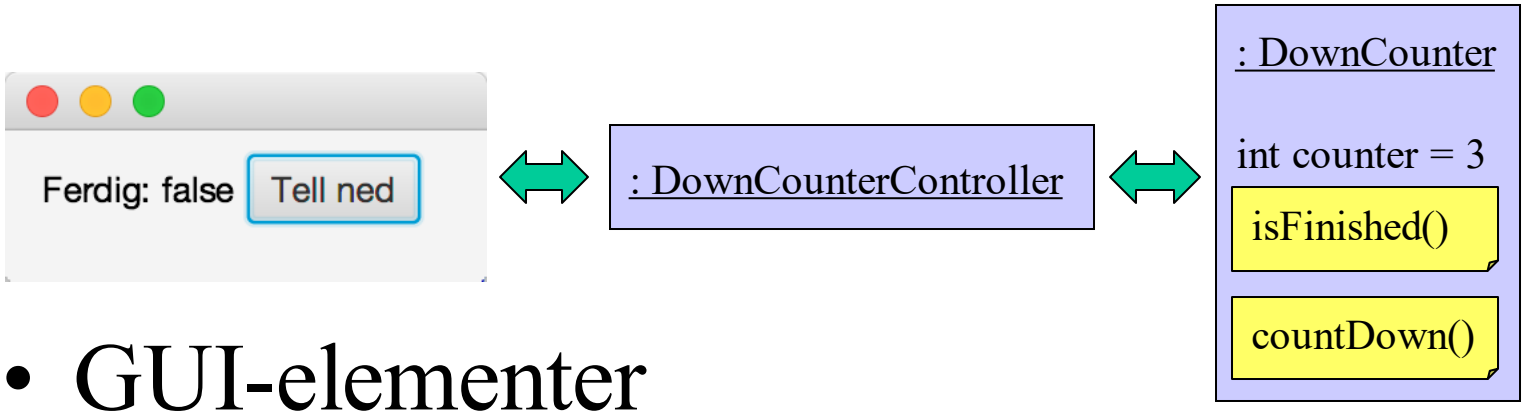


JavaFX

Kontroller

Intern tilstand

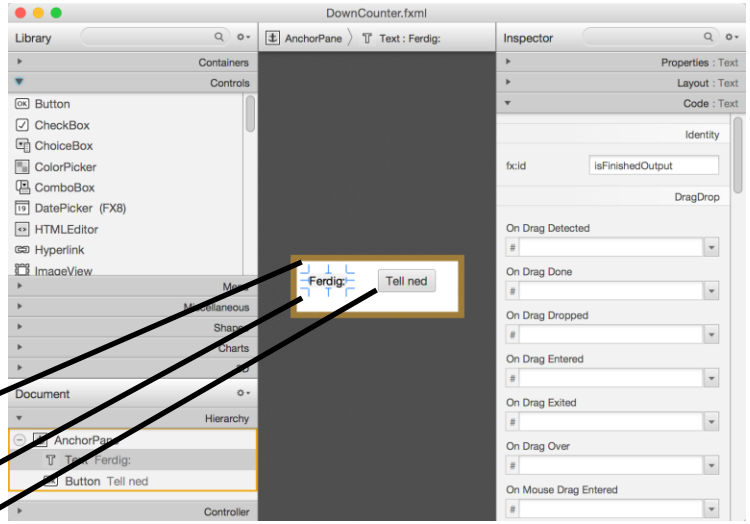
App for DownCounter



- GUI-elementer
  - tekst (grafikk) viser om nedtelling er ferdig
  - knapp brukes for å telle ned
- Kontroller-logikken
  - reagerer på knapp ved å telle ned
  - oppdaterer tekst
- Intern tilstand - DownCounter

GUI med JavaFX

(Kafe.fxml er slik)



SceneBuilder

```
<?import javafx.scene.control.Button?>
<?import javafx.scene.text.Text?>
<?import javafx.scene.layout.AnchorPane?>

<AnchorPane xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
    prefHeight="57.0" prefWidth="181.0"
    fx:controller="counter.DownCounterController">
    <Text fx:id="isFinishedOutput" text="Ferdig:" layoutX="14.0" layoutY="27.0"/>
    <Button onAction="#handleCountDownAction" text="Tell ned" layoutX="91.0" layout
</AnchorPane>
```

FXML

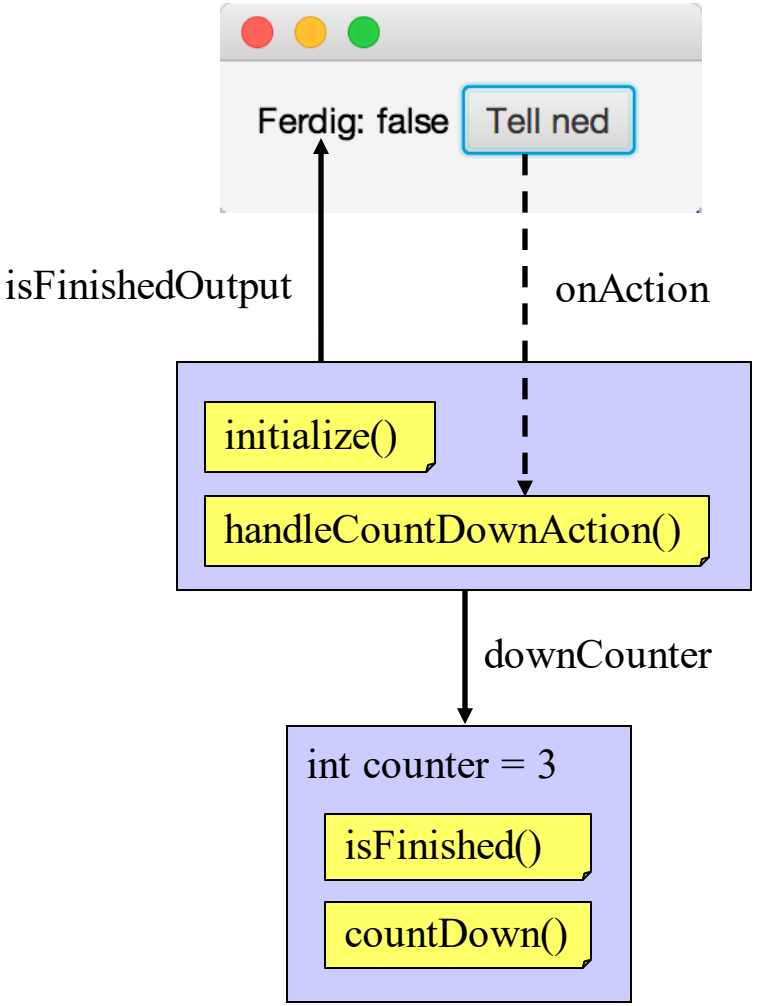
JavaFX-GUI

GUI med JavaFX (utvidet på gitlab)

```
<?import javafx.scene.control.Button?>
<?import javafx.scene.text.Text?>
<?import javafx.scene.layout.AnchorPane?>

<AnchorPane xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
    prefHeight="57.0" prefWidth="181.0"
    fx:controller="counter.DownCounterController">
    <Text fx:id="isFinishedOutput" text="Ferdig:" layoutX="14.0" layoutY="27.0"/>
    <Button onAction="#handleCountDownAction" text="Tell ned" layoutX="91.0" layout
</AnchorPane>
```

```
import javafx.fxml.FXML;
import javafx.scene.text.Text;

public class DownCounterController {

    DownCounter downCounter;

    @FXML Text isFinishedOutput;

    @FXML
    void initialize() {
        downCounter = new DownCounter(3);
        isFinishedOutput.setText("Ferdig: " + downCounter.isFinished());
    }

    @FXML public void handleCountDownAction() {
        downCounter.countDown();
        isFinishedOutput.setText("Ferdig: " + downCounter.isFinished());
    }
}
```

GUI med JavaFX



GUI med JavaFX

- Visuelle objekter (view)
  - FXML-filer beskriver GUI-struktur og –stil med tags og attributter ala HTML
  - ved kjøring så "oversettes" FXML-en til objekter av klasser tilsvarende XML-elementer (tags) og variabelverdier tilsvarende XML-attributter
- Koordinator (controller)
  - klasse angis i FXML-en med **fx:controller**-attributt
  - referanser til de visuelle objektene angis med **fx:id**-attributter og **@FXML**-annoterte felt med samme navn
  - aktivering av metoder med **onXXX="#metodenavn"**