

Seminaropplegg uke 12

Tema:

Hendelseshåndtering GUI, JavaFX

- JavaFX
 - Application
 - Stage Og Scene
 - Noder
 - EventHandlerler - neste uke.

Application

- Alle applikasjoner vi lager arver fra `javafx.application.Application` (vis gjerne Java-dokumentasjonssidene underveis)
- Vi `@Override` den abstrakte metoden `start(Stage stage)` med ønsket funksjonalitet for vår applikasjon
- Når vi skal kjøre applikasjonen kaller vi bare metoden `launch()` i main-metoden vår (launch arves fra `Application`).
 - Når vi launcher en `Application` oppretter vi samtidig en uavhengig applikasjonstråd.

Stage

- Når vi launcher en applikasjon opprettes det et `Stage`-objekt som blir sendt med som referanse og som vi kan bruke i vår definisjon av `start(Stage stage)`.
- `Stage` er hovedbeholderen for GUI-et og inneholder blant annet informasjon om max størrelse på vinduet som skal vises frem, tittel og `Scene`-objektet vi vil vise.
- For å vise vinduet til brukeren må vi kalle på `stage` sin `show()`-metode.

Scene

- `Scene`-objektet inneholder blant annet et eller flere layout-objekter (`GridPane`, `VBox` osv.), og kan også registrere hendelser og styre "kameraet".
- Vi kan instansiere et `Scene`-objekt ferdig med layout, eller legge til dette senere.

Noder

- Kort sagt: Alt du putter i en Scene.
 - Layout (panes)
 - Button, Label, TextField osv
 - Kontroll av events
 - En del noder kan ha "barnenoder" - for eksempel kan man ofte "nøste" panes (dette må ikke forveksles med arv)
 - Vi legger til barn under en node ved å hente ut "barnelisten" med `getChildren()`, for deretter å kalle på listas `add(nyNode)` eller `addAll(nyNode1, nyNode2, nyNode3, ...)`

Panes

- Flere å velge mellom, avhengig av hva vi trenger (tegn gjerne opp eksempler underveis).
- VBox/HBox: Legger alt i en vertikal eller horisontal linje
- GridPane: Lager et rutenett
- StackPane: Gjør det enkelt å legge elementer oppå hverandre fra bakerst til forrest
- Se dokumentasjon for flere alternativer

Kodeeksempel: HelloWorld

(Trenger kun å vise frem dette, da det alt er gjennomgått i forelesning)

```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.text.Font;
import javafx.scene.text.Text;

public class Hallo extends Application {
    @Override
    public void start(Stage teater) {
        Text hilsen = new Text("Hallo, alle sammen!");
        hilsen.setY(40);
        hilsen.setFont(new Font(40));
        Pane kulisser = new Pane();
        kulisser.getChildren().add(hilsen);
        Scene scene = new Scene(kulisser);
        teater.setTitle("Velkommen til Java FX");
        teater.setScene(scene);
    }
}
```

```

        teater.show();
    }
    public static void main(String[] args) {
        Application.launch(args);
    }
}

```

Oppgaver:

Oppgave 1

Lag et GridPane med 9 ruter (3*3), og fyll hver rute med tall slik at det ser slik ut:

1	2	3
4	5	6

Hint:

Benytt metoden `.add(Node element, int kolonnenr, int radnr)` på GridPanet for å legge til ting.

Benytt klassen "Text" for å legge inn tallene, denne er en subklasse av Node. Konstruktører som kan benyttes til å opprette instanser av "Text": (ligger i oppgavesettet).

Løsning:

(med StackPane, nevnt kort at denne er nyttig for å legge elementer oppå hverandre). Vis gjerne med kun Text først og legg deretter til Rectangle og StackPane.

```

//import ...

public class EnkeltRutenett extends Application {
    public void start(Stage stage) {

        GridPane pane = new GridPane();

        int teller = 1;
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {

                Text text = new Text(""+teller);
                text.setFont(new Font(30));
                Rectangle rect = new
Rectangle(40, 40, Color.SKYBLUE);
                rect.setStroke(Color.BLACK);

                StackPane cell = new StackPane();
                cell.getChildren().addAll(rect,
text);

                pane.add(cell, j, i);

                teller++;
            }
        }

        Scene scene = new Scene(pane);
        stage.setScene(scene);
        stage.setTitle("Rutenett");
        stage.show();
    }

    public static void main(String[] args) {
        Application.launch(args);
    }
}

```

```
}  
}
```

Oppgave 2, Digital klokke

I denne oppgaven skal du lage en digital klokke! For å telle benytter vi en indre klasse “Sekundteller”, som er en egen tråd som oppdaterer klokkeslettet hvert sekund. Du skal fylle inn den manglende koden her:

```
import javafx.application.Application;  
import javafx.stage.Stage;  
import javafx.scene.Scene;  
import javafx.scene.layout.Pane;  
import javafx.scene.text.Font;  
import javafx.scene.text.Text;  
  
import java.time.LocalTime;  
  
public class Klokke extends Application {  
    Text hilsen = new Text(naa());  
  
    class SekundTeller extends Thread {  
        // En egen tråd som oppdaterer klokkeslettet hvert sekund.  
  
        public void run() {  
            boolean slutt = false;  
            while (! slutt) {  
                try {  
                    sleep(1000);  
                } catch (InterruptedException e) { slutt = true; }  
  
                //1. Sett ny tekst/nytt klokkeslett i “hilsen”-variablen, FYLL INN:  
                //(Hint: se paa hvordan “hilsen” er opprettet..)  
  
            }  
        }  
    }  
  
    @Override  
    //3. FYLL INN METODENAVN OG EVT PARAMETRE:  
    public void _____ {  
  
        //3. SETT Y-aksen til teksten til å være 100.  
  
        //4. SETT FONT, opprett font med parameter 100
```

```

        Pane kulisser = new Pane();

//5. LEGG TIL "hilsen" i kulisser

//6. Opprett en scene med kulisser..

//7. SETT tittle på "teateret":

//8. Sett scenen og vis den!

//9. Opprett og start en "SekundTeller"-tråd:

    }

    private static String naa() {
        // Hva er klokken nå? Svaret er på formen "12:34:56".
        LocalTime t = LocalTime.now();
        return String.format("%02d:%02d:%02d",
            t.getHour(), t.getMinute(), t.getSecond());
    }

    public static void main(String[] args) {
        //10. Start applikasjonen

    }
}

```

Oppgave 2, Digital klokke, Løsning:

```

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.text.Font;
import javafx.scene.text.Text;

```

```

import java.time.LocalTime;

public class Klokke extends Application {
    Text hilsen = new Text(naa());

    class SekundTeller extends Thread {
        // En egen tråd som oppdaterer klokkeslettet hvert sekund.

        public void run() {
            boolean slutt = false;
            while (! slutt) {
                try {
                    sleep(1000);
                } catch (InterruptedException e) { slutt = true; }

                hilsen.setText(naa());
            }
        }
    }

    @Override
    public void start(Stage teater) {
        hilsen.setY(100);
        hilsen.setFont(new Font(100));

        Pane kulisser = new Pane();
        kulisser.getChildren().add(hilsen);

        Scene scene = new Scene(kulisser);

        teater.setTitle("Klokken er");
        teater.setScene(scene);
        teater.show();

        new SekundTeller().start();
    }

    private static String naa() {
        // Hva er klokken nå? Svaret er på formen "12:34:56".

        LocalTime t = LocalTime.now();
        return String.format("%02d:%02d:%02d",
            t.getHour(), t.getMinute(), t.getSecond());
    }

    public static void main(String[] args) {
        Application.launch(args);
    }
}

```

}