**Teori**

**Oppgave 1 - Ord og begreper**

* **JavaFX**

JavaFX er en software for håndtering av data og fremstillingen av disse. Den gir god støtte for animasjoner, grafikk og CSS hvor grensesnittet lages med XML-baserte filer. I tillegg er JavaFX plattformuavhengig. JavaFX har mange tilgjengelige elementer, blant annet button, checkbox, textfield, label, textarea, datepicker, togglebutton etc.

* **Anonym Klasse**

En anonym klasse er en klasse som ikke er blitt gitt noe navn og som både blir laget og instansiert i en og samme kodesnutt. En anonym klasse må alltid implementere et interface eller arve/hente fra en abstrakt klasse. Likevel benytter man ikke nøkkelordene extends eller implements for å lage en anonym klasse, men derimot «new interface-eller-klassenavn () { kode }».

* **MVC**

MVC står for Model View Controller, og er en måte å dele opp logikken i programmet vårt. Da deler vi logikken opp i tre deler, Model, View og Controller. Model representerer dataen som vi behandler i programmet, View er brukergrensesnittet, altså presentasjonen av dataen, og Controlleren er selve kodelogikken i programmet som kommuniserer mellom model og view.

* + **Model**

Model-en er klassene som brukes i prosjektet, og det er i klassene det blir oppgitt hva slags data som skal håndteres. For eksempel String navn, etternavn, int alder etc. Vanligvis henter vi data ut fra en database eller lignende, men her vil vi istedet simulerer dette ved at vi bruker egne klasser til å hente ut data fra. Her benytter vi en datahandler for å legge til/oppgi disse dataene til kassene.

* + **View**

View-er er brukergrensesnittet, altså det brukeren ser. View-et vet ingenting om model-en eller logikken i programmet, den er kun et view som skal vise deg noe. View gir beskjed til controlleren når en knapp blir trykket eller en handling blir utført men vet derimot ikke noe om hva som skal gjøres når det skjer.

* + **Controller**

Controlleren kommuniserer mellom model og view. Den henter ut data fra modellen og sender det videre til view-et hvor dataen skal vises. Controlleren får beskjed når noe trykkes på eller gjøres i view-et. Controlleren gjør noe basert på disse handlingene den får beskjed om, for eksempel oppdatere model-en eller vise nye data. Controlleren er den eneste i systemet som vet om «alle» klassene.

**Oppgave 2 - Kodesammenligning**

På denne oppgaven har jeg og Ole Henrik Johansen sammenlignet implementasjonene våre.

I ArrayList<Rolle> hentRollebesetning()-implementasjonene våre har vi begge en for-løkke med en if-setning inni som sjekker at ikke det er noen verdier for roller som er tom(null). Vi har begge gjort det slik for at det ikke skal gå an å liste ut tomme roller. Eneste forskjellene vi har er navna på array-listen vår med episoder.

I leggTilMangeRoller(ArrayList<Rolle> flereRoller)-implementasjonen vår har vi begge benyttet oss av addAll-funksjonen til å legge rollene. Dette har vi gjort for å enkelt legge til alle rollene i array-listen flereRoller. Her var det i grunn ingen forskjell. Begge har brukt enkle og logiske navn på både array-listene rolleListe og flereRoller.