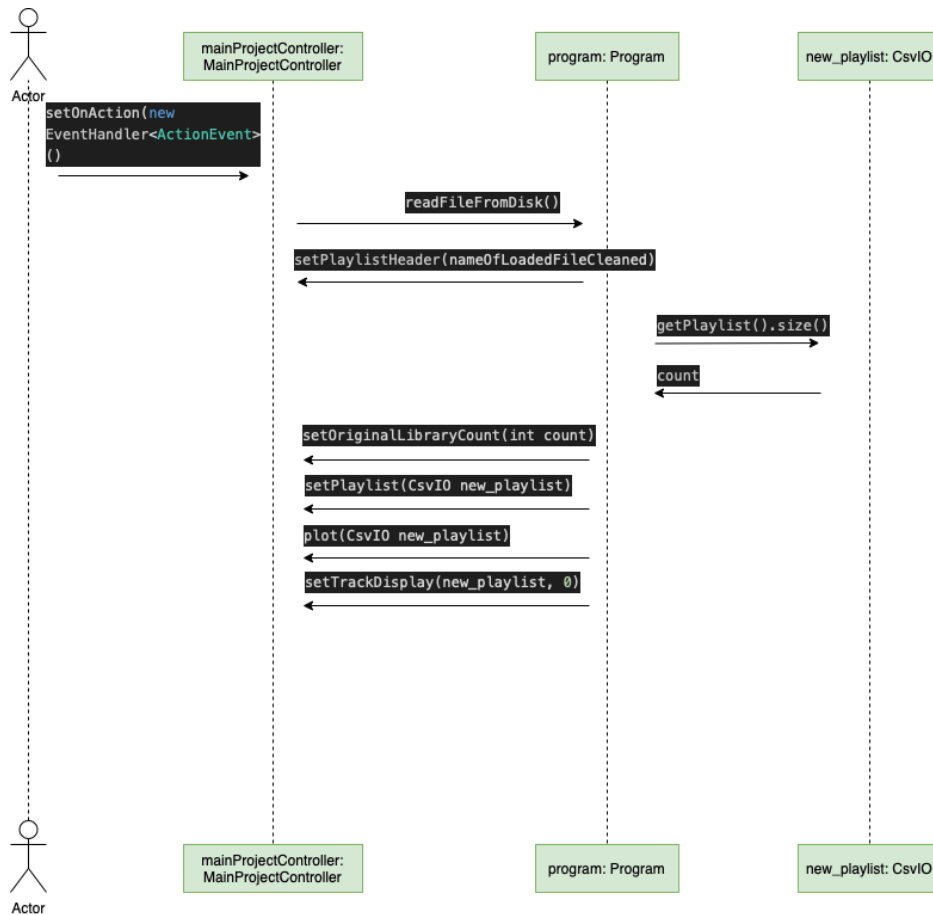


Programmet startes ved å importere: «top50.csv». file -> import
Mappen ligger i «ritmoProject» mappen med resten av koden

Applikasjonen er et analyseverktøy for musikk. I programmet kan du laste opp en csv-formatert datafil som inneholder data om ulike sanger og deres nøkkelparametere. Brukeren vil få en oversikt over datafilen i et listeformat. Fra spillelisten kan brukeren velge to sanger ved å trykke på sangen i spillelisten. Når to sanger er valgt vil brukeren få en oversikt over nøkkel-parameterne knyttet til hver av sangene. Samtidig vil den euklidske avstanden mellom sangene vises på skjermen. Brukeren vil også få innsikt i avstanden mellom de valgte sangene og resten av sangene i datasettet. Denne sammenhengen demonstreres i et spredningsdiagram. Her vil x-verdien til sangene som er plottet være avstanden til sang 1, og y-verdiene være avstand til sang 2. Brukeren kan sortere sanger basert på sangenes parametere. Tall parametere sorteres i synkende rekkefølge, og ulike typer navn sorteres alfabetisk. Sanger som brukeren vil huske kan markeres som «liket», og skrives til fil.

Applikasjonens hensikt er å hjelpe brukeren sortere å filtrere sanger. Et bruker scenario kan være følgende: En dj skal spille et show, han er nødt til å sette sammen en spilleliste. Han har allerede en spilleliste med mange sanger som han har fra før, men han trenger bare en delmengde av disse til dette showet. Han bruker sorteringsfunksjonen i applikasjonen for å finne sangene med høyest «Danceability», deretter velger han en referanse sang. Han bruker spredningsplottet og distansen mellom valgte sanger for å finne sangene som ligner mest på referansen. Han «liker» disse sangene, og eksporterer den likte listen til en datafil. Nå har han en liste over sanger som passer godt sammen til showet.



Dette er et sekvensdiagram som beskriver metodekallene som skjer når en sang velges fra spillelisten i brukergrensesnittet. Dette er en viktig del av programmet, fordi disse metodekallene henter sangen som er trykket og oppdaterer delen av brukergrensesnittet som viser hvilken sang som er valg, og spredningsplottet til de valgte sangene.

Interface og delegering er sentralt i programmet. En hoved funksjonalitet i programmet er å beregne avstanden mellom sanger. Dette brukes både når avstanden til de valgte sangene skal plottes, og når avstanden mellom to valgte sanger skal vises for brukeren. For å gjøre denne beregningen er det laget et Java-grensesnitt kalt «Distance». Det finnes ulike matematiske modeller for å beregne avstanden mellom to punkter i rommet. I dette programmet er bare euklidsk avstand som brukes, derfor implementerer «EuclideanDistance» dette grensesnittet.

Ettersom programmet følger MVC – modellen delegeres beregnings-logikken i kontroller - klassen til klassen «Program». Programmet tar bruk av grunnleggende relasjoner imellom klasser. «Track» som er klassen som representerer en sang og «CsvIO» som inneholder en spilleliste med sanger, har en en-til-mange relasjon. Lesing av fil gjøres med «Lambda Streams». Programmet bruker tilleggs biblioteket «java.io.File» for å behandle filer og bruker .streams() å lese filen linje for linje. Med lambdautrykk deler klassen dataen på «,», ettersom .csv er filformatet som leses.

Programmet følger MVC-modellen. Modell klassene er tiltenkt å ta for seg all logikk i appen, bortsett fra logikken som henger sammen med formatering av FXML filene. Modell klassene er {EuclideanDistance, CsvIO, LikedToText, Program, Selected, Track}. Under «resources» mappen ligger tre FXML filer {r_main, playlistEl, songParameters}. «r_main.fxml» lastes inn som «primaryStage» i ExampleProjectApp klassen. I MainProjectController lastes «playlistEl.fxml» og «songParameters.fxml». Først nevnte inneholder formateringen for spilleliste oversikten. Den andre inneholder formateringen for oversiktene for de to valgte sangene. Disse klassene står for View- delen i programmet. Programmet har tre kontrollerklasser for hver av FXML klassene. «ExampleProjectController» er kontrolleren for hoved grensesnittet. Denne klassen setter verdier i FXML elementer som vises på hovedsiden, samtidig som den inneholder Lytter funksjoners som delegerer oppgaver til program klassen «Program». «PlaylistController» og «SongParameterController» gjør tilsvarende for «playlistEl» og «songParameters».

Jeg har skrevet to tester som tester sentrale egenskaper ved programmet.

EuclideanDistanceTest og ProgramTest, førstnevnte tester klassen som tar for seg utregningen av euklidsk avstand mellom to sanger. Og den andre tester metoden som sorterer en liste. Metoden for avstanden mellom to sanger brukes i metoden for å plote spredningsdiagrammet og for å oppdatere FXML elementet som vises fram til brukeren. Jeg har valgt å begrense meg til disse, både fordi de inneholder nøkkelfunksjonaliteter i programmet og for å begrense oppgaven. Ideelt sett burde jeg laget en test som tester lesing til fil, altså CsvIO klassen. Den har også en viktig funksjonalitet og burde testes. Filtrerings klasse burde også testes, fordi på lik linje med sortering funksjonen er den viking.

