

# HashMap, Scanner (Bruker Input) & Random

## Oppgaver

### Beskrivelse

I denne oppgavesamlingen skall du lage et lite musikkbiblotek som kan legge til, hente ut og late som den spiller av ulike album. I tillegg skall du senere i oppgavene lage en interaktiv "AI" som kan ta imot tekst kommandoer fra brukeren ved hjelp av Scanner (new Scanner(System.in)), og reagere på disse kommandoene ved å utføre handlinger i musikkbibloteket. Disse oppgavene er ment som en øvelse på lagring og uthenting av objekt i HashMap, bruk av Scanner for å hente inn og reagere på bruker-input, samt bruk av Random for å gjøre "tilfeldige" valg i programmet.

1. Lag en klasse som heter **MusicLibrary** som har en HashMap musicCollection som lagrer objekt av typen <String, Album>
2. Lag klassen Album som har feltene: String title, String artist, int year, og boolean hasPlayed (samt getters og setters for disse), samt en konstruktør som tar som parameter String title, og int year og setter disse.
3. Legg til metoden **addAlbum** i **MusicLibrary** klassen som tar et Album objekt som input, og legger den inn i musicCollection hashmap med title som key og album objektet som value.
4. Legg til metoden **fillMusicLibrary** i **MusicLibrary** klassen som legger til noen album i musicCollection ved hjelp av addAlbum metoden.
5. Legg til metoden **getAlbum** i **MusicLibrary** klassen som tar album tittel som input og returnerer Album objektet fra musicCollection som tilsvarer den tittelen. Metoden skall ikke være case-sensitiv, slik at den finner riktig album uansett om man bruker store eller små bokstaver i tittel. (Hint: Bruk .toLowerCase() eller toUpperCase() som del av en eller flere metoder)

6. Legg til en metode **play** I MusicLibrary som tar inn en album tittel og printer ut "Playing [Album title goes here] – Year: [Year goes here]" dersom albummet finnes I musicCollection hashmap.
7. Legg til en metode **albumCount** I **MusicLibrary** klassen som returnerer en integer for antall album som finnes I musicCollection hashmap
8. Legg til metoden **playRandom** I **MusicLibrary** som bruker en intans av Random (java.util.Random) (random-number-generator) som plukker ut en tilfeldig album tittel fra musicCollection hashmap og spiller den av via play() metoden I MusicCollection.
9. Lag klassen **MusicAI** som har et felt kalt userInput av typen Scanner som kan lese user input, og et felt musicLibrary av tyen MusicLibrary som holder en instans av MusicLibrary. Konstruktøren til MusicAI klassen skall opprette en ny instans av MusicLibrary klassen og lagre denne I feltet musicLibrary. Deretter skall den kalle fillMusicLibrary metoden I MusicLibrary klassen fra oppg. 4 ( **musicLibrary.fillMusicLibrary()** ) slik at den blir fylt med album.

10. Lag en metode **search** i **MusicLibrary** klassen som tar en String som parameter og returnerer alle album i musicCollection hashmap som har titler som inneholder Stringen som er tatt inn som parameter. Resultatet skall ikke være case-sensitiv, slik at dersom vi har albumet “Super Album ABC”, og man søker på “super”, så skall vi fortsatt få “Super Album ABC” som resultat.

11. Legg til metoden **receiveCommand** i **MusicAI** som bruker userInput Scanneren (fra oppg 5) , og en if-else til å sjekke input den mottar og gjør ulike ting basert på input. Metoden skall respondere på følgende kommandoer:

Deloppgave	Kommando	Beskrivelse
11 a)	“how many albums do I have?”	Printer ut hvor mange album som finnes i MusicLibrary (bruk albumCount metoden fra oppg. 7)
11 b)	“play something”	Spiller av et tilfeldig album i MusicLibrary (bruk playRandom() metoden fra oppg. 8)
11 c)	“play”	Spørr om følgende: “Which album?”, og spiller av det albumet brukeren spesifiserer ved hjelp av play() metoden fra oppg. 6
11 d)	“play like”	Spørr om følgende: “Like what?”, og søker på album som ligner det brukeren spesifiserer ved hjelp av search() metoden fra oppg. 10, og spiller av det første albumet den finner, dersom den finner. Dersom den ikke finner noe skall den printe ut “Could not find (det brukeren søkte på her)”

12. Gjør en endring i **receiveCommand** metoden fra oppg 11. slik at den ikke blir case-sensitive, slik at dersom man for eksempel gir den kommandoen “PLay” vil den fortsatt tolke denne som “play” osv.

13. Legg til en metode `findNewerThan(int year)` I `MusicCollection` som returnerer alle album som har årstall nyere enn tallet som er tatt inn som parameter.

14. Legg til en `lastPlayed ArrayList` I `MusicLibrary` klassen som lagrer objekt av typen `Album`. Denne skall holde en liste over de siste spilte albummene.

14 a) Endre `play()` metoden I `MusicLibrary` slik at denne legger inn albummet som spilles av I `lastPlayed` listen.

14 b) Legg til en metode I `MusicLibrary` kalt `printLastPlayed(int amount)` som printer ut X antall siste spilte album (det antallet som er spesifisert av tallet som er tatt inn som parameter). Albummene skall printes ut I rekkefølgen Sist spilt til først spilt, altså sist spilt, så nest sist spilt, så nest nest sist spilt osv...