

# Arbeidskrav 1: PGR100 høst 2017

## Frist:

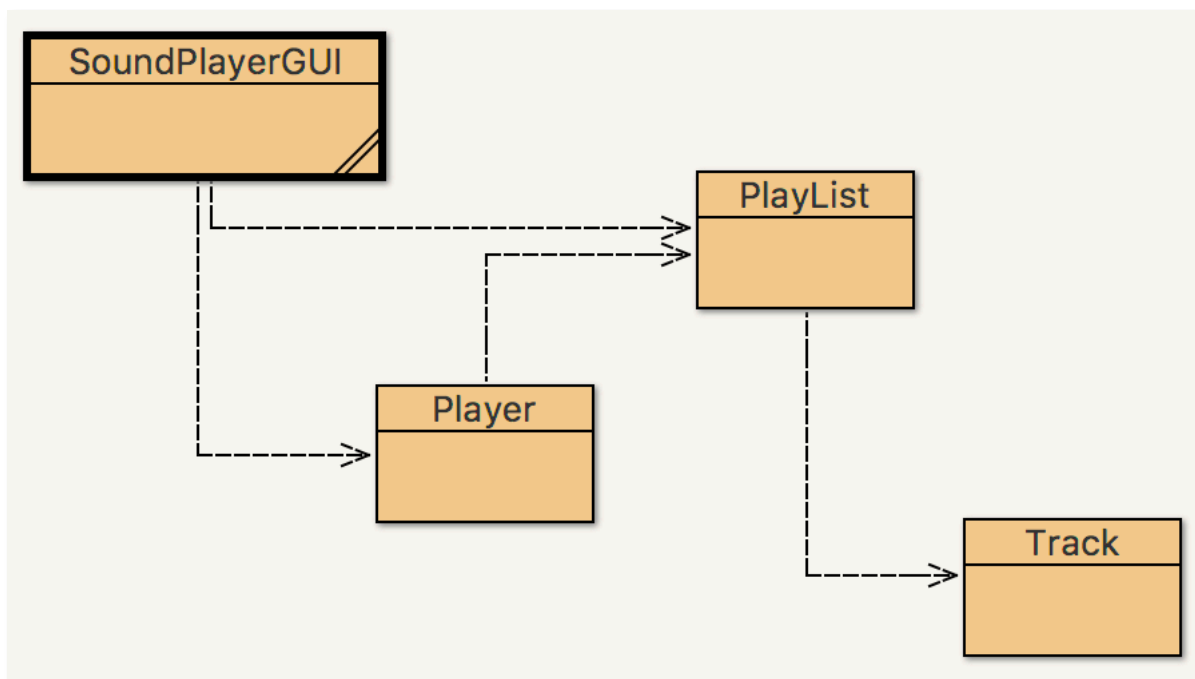
Søndag 15.10 innen 23:59. Leveres i It's Learning. IKKE LEVER I SISTE LITEN!

## Mål:

Lage en lydavspiller. Deler av løsningen er allerede implementert. I ITL finner du prosjektet du skal ta utgangspunkt i (**kentSound**). Navnet på prosjektet viser at dette er en oppgave som er utviklet og benyttet ved University of Kent av pensumbokens forfattere: Michael Kölling og David Barnes.

## Utgangspunkt:

Når du åpner prosjektet vil du se klassene i figuren nedenfor.



Gode nyheter! Tre av klassene er allerede laget for deg! Du trenger bare å implementere en av dem: klassen **Player**. Her er en kort beskrivelse av de tre klassene du ikke behøver å endre.

### SoundPlayerGUI

Denne klassen representerer GUI (Graphical User Interface) i applikasjonen. Den tegner opp hovedvinduet, menyer, knapper og alt det andre en bruker vil se og interagere med. Men, det skjer ikke så mye når dette brukergrensesnittet benyttes nå. Hvis en bruker klikker en knapp, så videreføres bare informasjonen om knappetrykket til **Player**-klassen.

### Track

Denne klassen representerer et spillbart lydspor. Sjekk ut hvordan den er laget: Den har metoder som **play**, **stop** etc. Du skal kalle disse metodene for å spille lydsporene i applikasjonen.

### Playlist

Denne representerer en liste over lydspor som kan spilles. En spilleliste er laget ved å spesifisere en mappe på harddisken med noen lydfiler i. (By default en mappen som heter audio inne i prosjektmappen). Listen vil automatisk laste inn alle lydfiler som den finner i den mappen (wav, au og AIFF formater)

## Player

Det er er i denne klassen du skal skrive avspillerens kildekode. Og ja, du må nok rette opp i noen initielle feil i denne før du setter i gang.

## Oppgaver:

Denne innleveringen krever at du gjør følgende: Implementere de metode-stubbene (metodene som mangler innhold) i **Player**. Da skal applikasjonen fungere slik:

- Når en bruker velger et spor i listen skal sporinformasjon (spornavn og varighet) vises i grensesnittet.
- Når en bruker klikker Play skal sporet begynner å spille. Stopp skal stoppe sporet.
- Når en bruker klikker Play uten å ha valgt et spor skal ingenting skje. (Det samme gjelder for Stop.)
- Når Play klikkes mens et spor fortsatt spiller skal det spillende sporet stoppe og deretter spille fra start.
- Når et annet spor blir valgt i listen mens et spor spilles skal lydavspillingen stoppe.
- **getTrackName** og **getTrackInfo** skal returnere en tom streng (en streng med lengde 0) hvis ingen spor er valgt.
- Du må fullføre alle manglende kommentarer (klasse- og metode-kommentarer).

For å oppnå dette, må du gjøre bruk av **playlist**-variabelen i **Player**-klassen til å hente spor, og **Track**-metodene for å starte og stoppe avspilling av sporene. For eksempel, linjen

```
currentTrack = playlist.getTrack (2);
```

vil hente spor nummer 2 fra spillelisten og lagre den i en variabel kalt currentTrack. Test koden godt. Programmet ditt skal aldri vise en runtime error.

**Tips:** Det første du bør gjøre er å få prosjektet ditt til å kompilere. Gjør bare de nødvendige endringene i koden for å få det til. Når koden kompilerer, forsøk å starte applikasjonen. Du starter applikasjonen ved å lage et objekt som representerer brukergrensesnittet. Programmet vil naturlig nok ikke i utgangspunktet fungere slik tiltenkt. Men det vil være et godt første steg å se at du har et brukergrensesnitt du kan bruke for å teste at løsningen din fungerer.

Det er først når du har en kildekode som kompilerer og du har klart å starte applikasjonen at du kan begynne å implementere etterspurt funksjonalitet. Ikke start med å skrive masse kode. Gjør små endringer og pass på at dine endringer også kompilerer.

## Ekstraoppgaver (ikke nødvendig for å få godkjent):

Implementer tre ekstra metoder:

- en metode som heter **getNumberOfTracksPlayed** som returnerer antall spor som har blitt spilt (dvs. som har blitt startet) siden spilleren ble startet. Returverdien skal være en **int**.
- en metode som heter **getTotalPlayedTrackLength** som returnerer den totale spilletiden av alle spor som har blitt spilt (eller, for å være mer presis: begynte å spille. Vi bryr oss ikke om de ble spilt helt ferdig). Resultatet er en **int**.
- en metode som heter **averageTrackLength** som returnerer den gjennomsnittlige lengden på de spilte låtene. Resultatet er av typen **double**.

Når du har implementert disse metodene åpner du **SoundPlayerGUI**, finner metoden som heter **showAbout**, og tar bort kommenteringen av fire linjer med kode som angitt i kommentaren

merket "CHALLENGE TASK " i kildekoden. Lydspillerens «About»-boks (Help / About ...) vil da kalle dine metoder for å vise denne informasjonen.

## Flere ekstraoppgaver (kun for moro skyld):

Legg til en knapp merket Statistics til grensesnittet. Hvis denne knappen klikkes, åpnes en dialog som viser statistikk-opplysningene fra den forrige oppgaven (antall spor, total og gjennomsnittlig spilletid). Tips: I utgangspunktet kan du la klikk på denne knappen medføre at den eksisterende «About»-boksen dukker opp. Hvis du vil gå litt videre med oppgaven, så lar du informasjonen dukke opp i et eget dialog-vindu som bare viser statistikk, ikke navnet og versjonsinformasjonen.

## Vurdering:

Oppgaven kan godkjennes på 2 måter:

1: Oppgave er løst som beskrevet :

- Kildekoden blir vurdert til bestått.

Eller

2: *Dokumentasjon på innsats:*

Du klarte *ikke* å løse oppgaven, men dokumenterer::

- Hvordan du har jobbet med oppgaven
- Hvor mye tid du har brukt på oppgaven?
- Hvilke hjelpemidler du har brukt
- Hvilken hjelp du mener du trenger for å komme videre med din løsning

Det legges i vurderingen vekt på:

- at klasser / metoder er skrevet i tråd med beskrivelsen i oppgaven
- ryddig kode (luft, innrykk)
- standard navngiving av klasser, metoder og variabler
- fornuftige datatyper
- klargjørende kommentering av kode der det er naturlig
- at klasser / metoder virker som tiltenkt
- at det ikke gjentas kode, men kaller på metoder der det er naturlig

Innleveringen teller ikke inn på endelig karakter, men må bestås for å kunne gå opp til eksamen.

## Levering:

Oppgaven leveres som en zip fil på It's Learning.

Oppgaven er individuell.

## Lykke til!