

Innlevering 1 - Notat

Navn: Lars Erling Westbye Dahl

Dato: 31.01.2016

Forutsetninger for å kunne teste løsningen: **Ingen**

Kommentarer til eget resultat:

- Kunne benyttet en CountdownLatch til å vente med å starte trådene etter at 5 kunde-tråder er opprettet.
- Kunne lagt mer tilrette for testing, selv om jeg forsøkte å følge TDD-prinsippene. Synes testing var en stor utfordring i denne oppgaven, så dette er noe jeg må jobbere videre med.
- Lite optimal bruk av HashMap'et *reservations* i metoden `getCustomerByCar()`, der jeg må finne *key by value*. Dette er kun for å få skrevet ut navnet på vedkommende de forskjellige bilene er utleid til. Kom ikke på andre og bedre løsninger.

Spørsmål:

- Ønsker litt input i forhold til testing og testing av tråder. Det var dette jeg slet med under løsning av oppgaven.

Beskrivelse av løsning

CarRentalApplication

Her ligger main-metoden.

Antall kunder er bestemt av en konstant, *CUSTOMERS*. Det hentes i dette tilfellet inn 10 kundenavn fra bruker hvor disse blir lagt i en ArrayList. Når halvparten(5) kundenavn er lagt inn kalles det på metoden `startThreadPool()` med 0 og antall kunder delt på to (5) som parametere. Det opprettes da en ExecutorService med en fixed thread pool med plass til 5 kundertråder. En for løkke oppretter kunde-objektene og tråder utifra listen med navn, og i tillegg settes det hvilken bilutleietjeneste kundene skal benytte.

CarRental

Implementerer interfacet `CarRentalService` med metodene `rentACar()` og `returnACar()` for å gjøre løsningen skalerbar.

Klassen har konstanter for hva slags registreringsprefix bilene skal ha, i dette tilfellet «RF», og antall biler - `DEFAULT_CAR_POOL = 5`.

Metoden `init()` oppretter bil-objektene og legger dem i en `ArrayList`. Denne skriver også ut status for utleiebilene når den er fullført.

Metoden `rentACar()` kalles på av kunde-trådene. Denne har en lock, slik at ikke flere kunder prøver å leie samme bil samtidig (race condition). Når kunde-tråden kommer seg inn i metoden kalles det på `await()` dersom det ikke er biler tilgjengelig. Tråden blir da stående og vente til en bil blir returnert. Deretter kalles det på `findAvailableCar()` som returnerer et bil-objekt. Når kunden får tildelt en bil legges det inn i en `HashMap reservations`, hvor kunde er nøkkel og bil er verdi. Deretter skrives det ut status for utleiebilene og det returneres `true`. Dersom det er ingen bil tilgjengelig vil metoden skrive ut status om at kunde har forsøkt å leie bil, men det var ingen tilgjengelig. Metodens returtype brukes hovedsakelig til testing. I `finally`-blokken blir status for utleiebilene skrevet ut igjen og låsen blir låst opp.

Metoden `returnACar()` kalles også på av kunde-trådene. Denne har også en lock, slik som `rentACar()` for å unngå race conditions. Deretter brukes kunde-objektet til å hente ut bilen vedkommende har leid i `reservations` og sette status på den til tilgjengelig. Reservasjonen, eller bindingen mellom kunde- og bilobjektet blir slettet fra `reservations` og det blir skrevet ut at kunden har returnert bilen. Når en bil er gjort tilgjengelig signaliseres det til ventende tråder. I `Finally`-blokken blir det skrevet ut status for utleiebilene og låsen blir låst opp.

Customer

Selve kunde-objektet. Har navn, `CarRentalService` og tids-variabler og konstanter i attributt-feltet. `Customer` implementerer `Runnable`, slik at objekter av denne typen kan kjøres som egen tråd.

I `run()`-metoden starter en evig `while`-løkke som kaller på metodene `rentACar()` med et vilkårlig intervall. Default-verdi er 1-10 sekunder.

Dersom en annen kunde-tråd har låst `rentACar()`-metoden vil gjeldende kunde-tråd måtte vente på at metoden blir låst opp.

Dersom `rentACar()` returnerer `true` vil tråden sove mellom 1-3 sekunder (default) før den kaller på `returnACar()`. Hvis det returneres `false`, vil tråden prøve å leie bil igjen etter 1-10 sekunder.

Tids-intervallene kan settes av metodene `setSleepTimeHire()` og `setSleepTimeReturn()`.

Car

Bil-objektet. Har variabler for registreringsnummer og tilgjengelighet i

attributt-feltet samt settere og gettere for disse.