**Innlevering 1 – PG4100 – Anita T. Ilieva**

Det første jeg gjorde var å lage de forskjellige klassene ut fra klassediagram eksempelet i oppgaveteksten. Deretter skrev jeg metoder og attributter som klassene skulle ha. Klassen Car inneholder opplysninger om bilen ( regnr og hvem den er utleid til). I CarRentalApp opprettes trådene med Scanner input og ExecutorService.

Customer klassen implementerer Runnable. Metoden run() skal kjøre i en evig loop siden kunder skal kunne leie og levere inn biler i det uendelige. Kundene begynner ikke å leie før 5 tråder har blitt opprettet. Har brukt CountdownLatch (5) for å få det til. Trådene kaller på await() og venter til countDown metoden har telt ned til 0 og så starter de å kjøre. Await metoden kalles i run metoden mens countDown metoden har jeg lagt i for løkka der trådene opprettes ( I CarRentalApp klassen).

Har også brukt semaphore.acquire() og semaphore.release() i Customer klassen. Når kunder kjører run(), prøver de å aquire() (få) en tillatelse fra semaphore(som har 5 tillatelser). Hvis de får en tillatelse ( a permit), venter de 1-10 sekunder før de leier en bil og deretter venter de i 1-3 sekunder før de leverer bilen tilbake. Når bilen er levert igjen , kalles semaphore.release() og loop iterasjonen starter på nytt og ny kunde prøver å få ny tillatelse.

CarRental klassen oppretter en liste av Car objekter og har 4 metoder: rentCar, returnCar og statusUpdates, getCars. I rentCar og returnCar metodene bruker jeg locks for å sikre at ingen tråder kan leie inn samme bil samtidig. Når en tråd kaller på rentCar metoden så låser den den til seg helt til den har leid bilen og da blir lock låst opp.

**Eventuelle forutsetninger for å kunne teste løsningene.**

Ingen forutsetninger

**Kommentarer til eget resultat. Kunne noe vært gjort bedre?**

Jeg er fornøyd med løsningen jeg har laget. Kunne sikkert klart å lage flere tester men synes det var litt vanskelig. Likevel har jeg har prøvd å lage noen tester. Når jeg kjører testene får jeg ikke noe feilmelding på noen av dem men der igjen er jeg usikker på de er laget på en “riktig” måte.

**Eventuelle spørsmål du trenger svar på for å komme videre.**

Testing av metoder er det jeg synes var vanskeligst for min del. Siden vi ikke har hatt så mye om det måtte jeg prøve å finne nyttig informasjon fra nettet.

**Eventuelle referanser til hvilke kilder som er brukt.**

<http://www.tutorialspoint.com/java/java_multithreading.html>

<http://tutorials.jenkov.com/java-concurrency/index.html>

<https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/CountDownLatch.html>

<http://stackoverflow.com/questions/26421021/mocking-getter-setter-method-of-interface-with-mockito>

<http://2min2code.com/articles/mockito_intro/creating_mock_object>

Forelesningspdf – Threads 1, Threads 2, Threads 3