CvP - Programming Assignment 4

- Deadline: December 7, at the start of the werkcollege.
- Submit your solution electronically via liacscvp2018@gmail.com
- Clearly state your name and student number your solution file.
- Your Java programs should be submitted together with a written report (.txt or .pdf) in which you explain your programs.
- This exercise comes with a Java program, called `BeerDrinkers.java`. To compile this program, run `javac BeerDrinkers.java`. This generates a `.class` file for each class in `BeerDrinkers.java`. To execute the compiled java program, run `java BeerDrinkers`.
- https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html, provides a tutorial on concurrency in Java.

Question Consider the *Readers/Writers Problem*: Two kinds of processes readers and writers share a database. Writers execute transactions that update the database; reader transactions access the database without modifying it. The database is assumed initially to be in a consistent state (i.e. one in which relations between data are meaningful). Each transaction, if executed in isolation, transforms the database from one consistent state to another. To preclude interference between transactions, a writer process must have exclusive access to the database. Assuming no writer is accessing the database, any number of readers may concurrently execute transactions.

Inspired by the example `BeerDrinkers.java` program, write two Java programs for the following problem:

(a) Write a Java program `RWP1.java` that implements the Readers/Writers problem using monitors. Write a report that explains your program.

(b) Write a Java program `RWP2.java` that implements the Readers/Writers problem using monitors and giving priority to the writers; i.e., if a writer wants to enter the database but has to wait, then no reader may enter. Of course, the writer still has to wait for the current readers in the database to finish their transactions. Write a report that explains your program.