**Command-line Video Store Exercise**



**Preparation**

- Clone project: https://github.com/ramonafensterer/commandline videostore.git

- Ensure that code is valid: run the tests ;)

**Exercise 1**

Create a new method in the class MovieRepository, with the signature

public List<Movie> getMoviesWithCondition(List<Movie> movies, Predicate<? super Movie> predicate){},

which returns the list of Movie objects that fulfill the condition specified as predicate.

To check your implementation please activate the test MovieRepositoryTest, by deleting the @Ignore annotation.

**Exercise 2**

Add a new line to the footer displayed after a rental, that makes a recommendation for another movie to rent.

The rule is:

* if the majority of the movies that the customer rented are produced before the year 2000, the recommendation consists of the oldeest movie existing in the database, that he did not yet rent and that was produced before year 2000.
* if the majority of the movies that the customer rented are produced after th year 2000, the recommendation consists of the newest movie existing in the database, that he did not yet rent and that was produced after the year 2000.
* if the result of the search for the recommendation is empty, don’t display any extra footer line.
* use lambdas and streams … and make sure the tests are green (delete the comment from the MainTest to check your implementation)

**Exercise 3**

Refactor the printRentalRecord() method in the Console.java class to use a method of the interface List. Also, instead of lambdas, use method references. You can check your implementation by uncommenting the test testQuoteLine from the MainTest.

**Hint:** a method needs to be overridden in Rental.java

**Exercise 4**

In order to be able to provide the customer with an invoice, add some information to the file invoice.txt, which will then be printed. At the end of the file, a famous movie quote should be listed. This line will be chosen randomly from the file quotes.cvs.

