Mandatory Exercise – Lecture J2

02160 – Agile Object-Oriented Software Development

v. 2019-2-12

Instructions

It is mandatory to complete this exercise and to submit it the day before the next lecture (deadline is 19-02-2019 13:00). To deliver the exercise, upload **only** the `.java` files containing your solutions in DTU Inside.

It is possible to work in pairs but in this case it is necessary to clearly state that using a group submission. Additionally, during the next session, you might be asked to explain your solution. In case you are not able to properly explain the solution and answer related questions, the whole exercise will be considered as failed.

Exercise 1

Implement the hierarchy of classes and corresponding logic needed to have the "Car Collision Game" program (see `CarCollisionGame.java`) properly functioning. You are not allowed to alter in any way the provided `main` method.

Exercise 2

Implement the hierarchy of classes and corresponding logic needed to have the "Chess" program (see `Chess.java`) properly functioning. You are not allowed to alter in any way the provided `main` method. The program has to properly determine the validity of possible positions of pieces on a Chess board. We assume no other pieces are on the board. The moves to check correspond to moves of king, rook, bishop, knight, and pawn. For further information you can visit the Chess entry[1] on Wikipedia.

[1] See `https://en.wikipedia.org/wiki/Chess`.