Portfolio Project I

This portfolio is about object-oriented programming and software engineering with focus on how to implement classical OOP models and understand, apply, and ensure software requirements. You can work on the portfolio in groups but be aware that the exam is individual, and you must be able to answer questions related to all parts of your portfolio. If you work in groups the members of the group must be named (with name and study number) on the first page, and only one of the group members need to submit on Moodle.

The submission should contain a pdf document for the five parts below and a zip file of source code for parts 3 and 4.
The overall submission of this part is limited to 5 pages plus the source code.
The maximum group size is 4.
The portfolio can be written in Danish or English.
Submission deadline for this part: March 9th 2023 at 12 noon via Moodle
Submission deadline for the full portfolio: May 4th at 10AM via eksamen.ruc.dk
The final submission of the full portfolio is limited to 48,000 characters, including spaces.
Groups should only submit one version via Moodle of this part. All students must submit the final version of the full portfolio via eksamen.ruc.dk on May 4th

Description

You are going to develop an object-oriented tool for a shipping organization which deals with three different types of vessels: container vessels, tankers and RoRo vessels. All vessels have a name. They will also all carry cargo and a container vessel can carry a specific number of containers. The tankers have 10 compartments which each have the same capacity and which each can be filled to some level. Capacity is measured in cubic meters. The RoRo vessels has a total amount of lane meters and can carry cars and trucks, each car is 8 meter long and each truck is 30 meters.

The assignment consists of the following parts

1. Make a Kanban board for this assignment
2. Make a class diagram for the different kinds of vessels
3. Implement classes to represent these vessels
4. Specify and run unit tests for these classes
5. Specify a use case diagram for a future extension of the system

Part 1

The first part of the portfolio is to create a Kanban board for this portfolio. Your task is to start using Kanban as soon as possible, following the flow of adding tasks to the backlog, prioritize the tasks, do the task breakdown, solve tasks and validating the solutions. Your Kanban process can be done on paper, whiteboards, on GitHub, or whatever you like.

Part 2

Make a class diagram for the different kinds of vessels. Specify which type of connection there are between the different classes. Use standard UML notation for the class diagram.

Part 3

Implement classes to represent these vessels. The implementation should use inheritance and polymorphism where appropriate.

Provide constructors for relevant classes and methods **`loadingCargo()`** and **`loadFraction()`** for all vessels. The method **`loadingCargo()`** should take two parameters: a number and a string. For a container vessel it should be a number of containers and the string "TEU", for tankers the string should be a number between 1 and 10 as a string, for RoRo it should be either "car" or "truck". The method **`loadFaction()`** should return a number between 0 and 1 to indicate how much of the capacity is used. For tankers a compartment is assumed to be in use if it is not empty since it then cannot be used for other products.

Part 4

Provide unit test for valid usage of the classes and the two methods done in part 3. Provide unit tests for invalid usage of the Tanker class.

Part 5

In a later version (not part of this assignment) we want to extend the system with assistance to find vessels that can be used to carry given cargo to a destination. The shipping department will keep track of the position of their vessels and future destinations. The shipping department will also plan future departures of vessels. The freight department provide transportation of cargo to customers and will search for vessels that can be used for a given shipment.

Specify use case diagram for this extension of the system