Assignment 2

Deadline: Friday, 16 October 2020 at 6pm

First of all, we ask you to create a new folder for this assignment in your GitHub repository.

Exercise 1 (7 points)

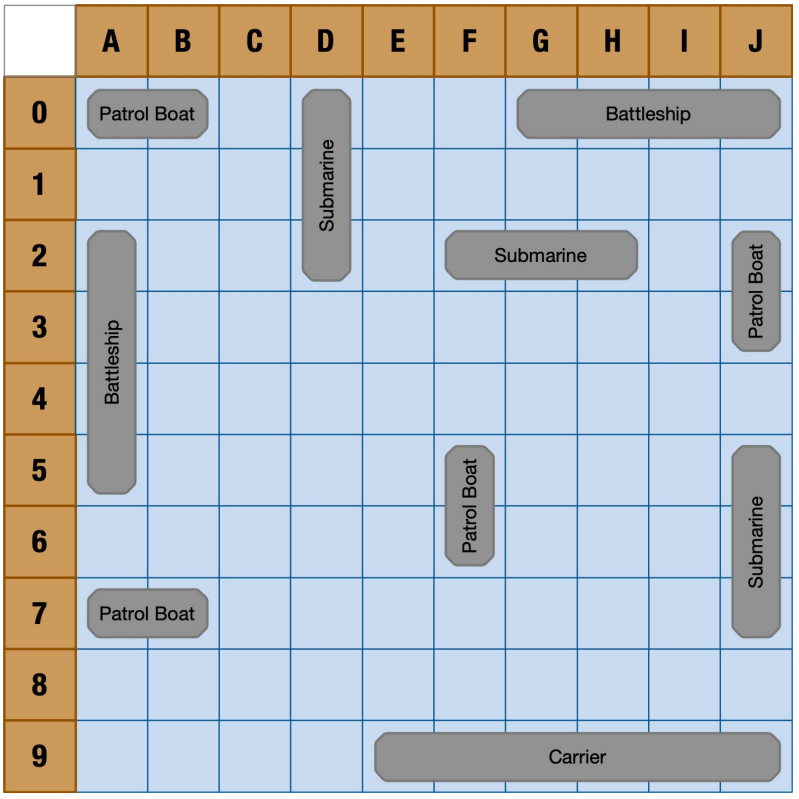
This exercise is based on the famous 2D Battleship game[1]. In particular, in this exercise, you are going to implement the placement of the boats in the gaming board, based on the user's text input, and display a text-based version of the board. See examples below.

You must use the concepts of encapsulation and interfaces you learned in class to design your code. You should make sure that your code does not violate the encapsulation principles and decouple the interface of a class from its implementation, but be careful that classes do not implement unnecessary interfaces.

The game should feature:

- A board (10x10 grid). Each block in the board is associated with a letter (on the horizontal axis) and a number (on the vertical axis). So, the top right corner must be block A1, as shown in the following figure:
- A Fleet with the following boats:

Type	**Size (in blocks)**	**Number of boats**
Carrier	6	1
Battleship	4	2
Submarine	3	3
Patrol boat	2	4



The game should start by asking the player where they want to place each boat in their fleet, then the game should print a textual version of the board.

Considering the figure above, it would correspond to the following example in your implementation of the game (the user input is in blue):

```
Welcome to Battleship!!

Please enter the position of your Carrier:         E9 J9
Please enter the position of your BattleShip 1:    A2 A5
Please enter the position of your BattleShip 2:    G0 J0
Please enter the position of your Submarine 1:     D0 D2
Please enter the position of your Submarine 2:     F2 H2
Please enter the position of your Submarine 3:     J5 J7
Please enter the position of your Patrol boat 1:   A0 B0
Please enter the position of your Patrol boat 2:   A7 B7
Please enter the position of your Patrol boat 3:   F5 F6
Please enter the position of your Patrol boat 4:   J2 J3

Your board is:

   |[A][B][C][D][E][F][G][H][I][L]
---|------------------------------
[0]|[P][P][ ][S][ ][ ][B][B][B][B]
[1]|[ ][ ][ ][S][ ][ ][ ][ ][ ][ ]
[2]|[B][ ][ ][S][ ][S][S][S][ ][P]
[3]|[B][ ][ ][ ][ ][ ][ ][ ][ ][P]
[4]|[B][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[5]|[B][ ][ ][ ][ ][P][ ][ ][ ][S]
[6]|[ ][ ][ ][ ][ ][P][ ][ ][ ][S]
[7]|[P][P][ ][ ][ ][ ][ ][ ][ ][S]
[8]|[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
```

```
[9]|[ ][ ][ ][ ][C][C][C][C][C][C]
```

During the input of the boats' positions, the game must check that the boat can be placed in the user-specified position. If the position entered by the user is invalid (e.g., because overlapping with another boat), the game prints "the specified input is invalid" and asks the user to re-enter the position.

A position is invalid if:

1. The position is (partially) occupied by another boat (two boats can not overlap);

```
Please enter the position of your Carrier: B3 G3
Please enter the position of your BattleShip 1: C2 C5
The specified input is invalid
Please enter the position of your BattleShip 1:
```

In this example, the position of BattleShip 1 overlaps with the position of Carrier, so it is invalid.

2. It contains a block that does not exist on the board (e.g., block W1)

```
Please enter the position of your Carrier: I10 N10
Specified input is invalid
Please enter the position of your Carrier:
```

In this example, the carrier is placed partly out of the board, so the position is invalid.

3. The position does not respect the size of the ship

```
Please enter the position of your Carrier: E10 J10
Specified input is invalid
Please enter the position of your Carrier:
```

In this example, the user tries to place the carrier over 10 blocks, but the carrier size is only 6 blocks. So the positioning is invalid.

Please, consider other corner cases and handle them accordingly. For instance, the position of the ship should be provided as capitalized characters and negative numbers are not valid.

Exercise 2 (3 points)

Consider the project from Assignment 1 and draw a UML class diagram comprising all the classes in the following folder:

> Groups 1-9:
> https://github.com/cryptomator/cryptomator/tree/develop/main/launcher/src/main/java/org/cryptomator/launcher

Groups 10-18:
https://github.com/LibrePDF/OpenPDF/tree/master/pdf-swing/src/main/java/com/lowagie/rups/model

Groups 19-27:
https://github.com/pmd/pmd/tree/master/pmd-core/src/main/java/net/sourceforge/pmd/util/viewer/model

Groups 28-36:
https://github.com/JodaOrg/joda-time/tree/master/src/main/java/org/joda/time/tz

Groups 37-46:
https://github.com/kotcrab/vis-ui/tree/master/ui/src/main/java/com/kotcrab/vis/ui/util/adapter

You can use a tool to help you draw the class diagram, but you must **not** automatically generate the class diagram.

Write a description of the class diagram (if needed) to further describe some of your choices. Address these points in the **ANSWERS.md** file in the assignment folder.

Important notes

Write all your answers in the **ANSWERS.md** file. Do not use a file with a different name. Also, please ensure that it is clear which answers belong to each exercise of the assignment.

Please note that no commits can be added to the assignment folder on GitHub after the deadline. Any commit added after the deadline **will not** be counted in the evaluation of the assignment.

You can **not** automatically generate the class diagram