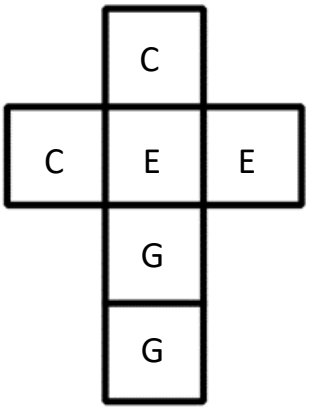
**Due Date: 26.12.2025 23:55**

CENG211 – Programming Fundamentals
Homework #4

In this homework, you are expected to implement a simple **"Box Top Side Matching Puzzle App"** in Java.
The submitted homework should mainly fulfill the following concepts:

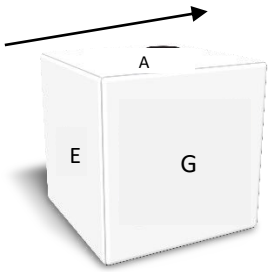
- Collections
- Inheritance
- Interfaces & Abstract Classes
- Polymorphism
- Generics
- Exception Handling
- Inner Classes

In this application, you will simulate **a puzzle game app involving an 8x8 grid of cubic boxes**. All 6 surfaces of each box have a large letter stamped on them. There are **8 possible letters** in total: A, B, C, D, E, F, G, H. Each box cannot have the same letter stamped on its surfaces **more than twice**. For example, a box cannot have the following letters on its 6 sides: B, D, D, D, F, G. However, the following combination is possible: C, C, E, E, G, G.




The goal of the game is to **manipulate the boxes so that a specified letter appears on the top side of the greatest number of boxes**. The game starts by randomly selecting the **target letter** from A to H. The game lasts for **5 turns**, and the player must make the **most optimal choices** at each turn to maximize the number of boxes with the target letter on their top sides. **Each turn has 2 stages:**

Edge box selection: The player selects **one of the boxes from the edge of the grid**. Then, the player **rolls the selected box** inward into the grid, causing every other box in the same row or column to roll as well. Assume there is a special mechanism that rolls each box in the same direction together. You can imagine this action as similar to a **domino-effect**. The movement of one box **affects the other boxes in the same direction** unless a special case occurs (described below). If the selected box is in a **corner**, the player is given the choice of rolling in either direction, and must select one. When rolling the boxes, it is easier to imagine them as dice. ***For example,*** if the top side of the box below is A and it is rolled to the right, the letter E arrives on the top side of the box.




Box opening and SpecialTool usage: The player must select a box **that was rolled in the previous stage** by specifying its position on the grid. Grid positions are shown using the following IDs: rows are named from *R1 to R8*, and columns from *C1 to C8*. An example position is "*R2-C4*". The box in the selected position is then opened to **acquire a SpecialTool** from it. Assume that a box can be opened from any of its sides as long as it was moved in the previous stage. After a box is opened, its contents are **marked as empty** and its lid is closed, so that its top-side letter remains unchanged. Note that some boxes may **already be empty from the start**. If a SpecialTool is acquired from the box, it must be **used immediately**.

There are **3 types of Boxes**. They are marked on the grid with special markers. At the start of the game each box is generated randomly:

- RegularBox: A **simple box** that may or may not contain a SpecialTool. It has a 75% chance of containing a SpecialTool. A RegularBox has an 85% chance of being generated.
- UnchangingBox: A box that **cannot be re-stamped** with any other letter on any of its sides. Once it is generated, no letters on its sides can be changed by any SpecialTool (mentioned below). They are **guaranteed** to contain a SpecialTool and have a 10% chance of being generated.
- FixedBox: A box that cannot be rolled in any direction. Its **top side stays the same at all times**. Since it cannot be moved, **it prevents any actions from being passed through** it. In other words, it **stops the domino-effect** from being transmitted past it. If an edge box is also a FixedBox and it is chosen during the first stage of a turn, an UnmovableFixedBoxException is thrown and that turn is wasted. They can **never contain** a SpecialTool and are marked as empty from the start. They have a 5% chance of being generated.

There are **5 types of SpecialTools**. At the start of the game, they are randomly placed inside the boxes. They are the main tools the player uses to change the letters on the top sides of the boxes. After the game starts, the rule requiring no more than 2 of the same letters per box no longer applies. The player can freely change letters by re-stamping them using SpecialTools:

- PlusShapeStamp: A tool that re-stamps **5 boxes (in a plus shape) to the target letter**. The app asks for a center box through the menu, and the selected center box along with its upper, lower, left, and right neighbors are stamped.
- MassRowStamp: A tool that **re-stamps all boxes in an entire row** to the target letter. The app asks the player to select a row.
- MassColumnStamp: A tool that **re-stamps all boxes in an entire column** to the target letter. The app asks the player to select a column.
- BoxFlipper: A tool that **flips a box upside down**. The chosen box's top side becomes its new bottom side, and vice versa. The app asks the player to select a specific box location.
- BoxFixer: A tool that **replaces a box with an identical FixedBox** copy. If it has a tool inside, it is removed from the game. The app asks the player to select a specific box location.

Each tool has a 15% chance of being generated inside RegularBoxes. Thus, RegularBoxes have a 25% chance of being empty. For UnchangingBoxes, the chance of each tool increases to 20%. For FixedBoxes, the chance is 0%.

If an opened box is empty, an EmptyBoxException is thrown and that turn is wasted. If a tool is found after opening a box, the player should take that tool as their acquiredTool and use it with the useTool() method. The same useTool() method should be called directly on the object regardless of its type. (**Hint:** This is where you are going to use **Generics**. acquiredTool can be of type "T". Naturally, type "T" should extend SpecialTool.) After calling the method in this way, you are free to receive and handle inputs as you prefer.

When the boxes are rolled during the first stage of a turn, they are all rolled using their own roll(…) method with a directional parameter. The directional parameter corresponds to the grid's positioning. For example, the upper side of the grid corresponds to rolling upwards. How you handle directional parameters is up to you. The most important part here is to **simulate the domino-effect of the boxes instead of manually manipulating each box individually**. (Hint: The roll method may need more than one parameter.)

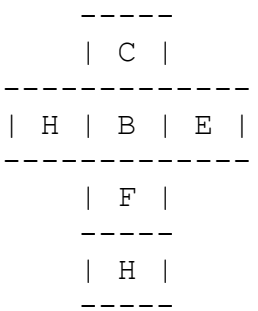
The game should start inside the **BoxPuzzleApp main class**. The main method of the BoxPuzzleApp class should only **initialize a BoxPuzzle** object. The BoxPuzzle class should **contain the BoxGrid**. BoxGrid **possesses 64 boxes**. Most of the menu operations should be handled **inside an inner class** within the BoxPuzzle class.

You should choose **an appropriate data structure from the Collections** framework to use inside the BoxGrid class. **At the top of that file, explain your reasoning for this choice in a few sentences inside a comment block.** Mark the start of the comment block with *"ANSWER TO COLLECTIONS QUESTION:"* to make it more visible.

Trying to flip a FixedBox using a BoxFlipper will throw an UnmovableFixedBoxException and the turn will be wasted. Trying to fix a FixedBox using a BoxFixer will throw a BoxAlreadyFixedException and the turn will be wasted.

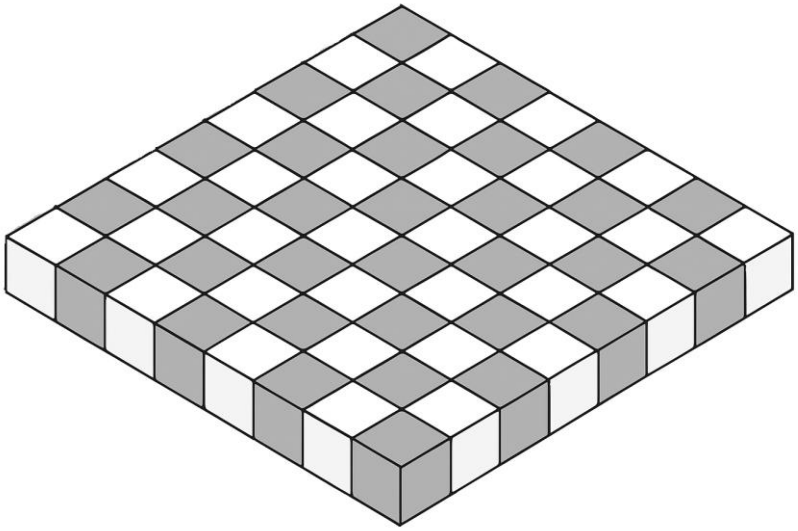
An UnchangingBox can be restamped using relevant SpecialTools directly or indirectly. However, its top side **letter remains the same afterwards**.

The player has an **additional ability at the start of each turn**. They are given the choice of selecting one box from the grid and **observing all its surfaces** to make a better move during their turn. The player may choose not to use this ability. In case the player chooses to use this ability, they are expected to input the box location. Afterwards, surfaces (and the stamped letters on them) are **shown in a cube diagram similar to the one shown below**. The middle position of the box diagram must correspond to the top side of the selected box. In this example, the top side has the letter "B".



During menu operations, an incorrect box location input will simply cause the **game to ask for the input again** until a correct input is entered. The same conditions also apply for any other incorrect menu input other than ones that cause the aforementioned exceptions. ***For example,*** when the player tries to open a box that has not been rolled on the same turn, the menu will simply ask for the input again.

**At the end of the game,** the number of boxes with the target letter should be printed along with a *SUCCESS* message. The goal of the app is to simulate this program. It is acceptable even if you end up with 0 boxes showing the target letter. In an extreme scenario where no moves can be made, the game may end earlier than 5 turns with a *FAILURE* message. For example, this situation can occur when all edge (and corner) boxes are fixed. Alternatively, this situation is possible when there is no available box for a SpecialTool to be used.



**<u>Legend for Box Grid Menu Notations:</u>**

Stamped letters: `A, B, C, D, E, F, G, H`

RegularBox: `R`

FixedBox: `X`

UnchangingBox: `U`

The mark for an empty box that have been opened already (or it is a FixedBox): `O`

The mark for a RegularBox or an UnchangingBox that have not been opened: `M` (Mystery)

An example unopened RegularBox with stamped letter E: `| R-E-M |`

An example opened empty RegularBox with stamped letter B: `| R-B-O |`

An example empty FixedBox with stamped letter A: `| X-A-O |`

An example unopened UnchangingBox with stamped letter F: `| U-F-M |`

## Example Scenario Output:

```
Welcome to Box Top Side Matching Puzzle App. An 8x8 box grid is being generated.

Your goal is to maximize the letter "D" on the top sides of the boxes.

The initial state of the box grid:

        C1      C2      C3      C4      C5      C6      C7      C8
    ---------------------------------------------------------------
R1  | R-E-M | R-B-M | R-A-M | R-B-M | R-E-M | U-H-M | R-A-M | R-F-M |
    ---------------------------------------------------------------
R2  | R-F-M | R-B-M | R-C-M | R-H-M | R-B-M | R-H-M | R-B-M | R-H-M |
    ---------------------------------------------------------------
R3  | R-F-M | R-B-M | R-B-M | R-B-M | R-D-M | R-D-M | R-D-M | X-E-O |
    ---------------------------------------------------------------
R4  | U-H-M | R-D-M | R-E-M | R-H-M | R-G-M | R-E-M | R-C-M | R-A-M |
    ---------------------------------------------------------------
R5  | R-A-M | R-A-M | R-F-M | U-A-M | R-G-M | R-E-M | R-B-M | R-G-M |
    ---------------------------------------------------------------
R6  | R-C-M | R-B-M | R-D-M | R-B-M | R-E-M | R-C-M | R-E-M | R-A-M |
    ---------------------------------------------------------------
R7  | R-D-M | R-G-M | R-F-M | X-D-O | R-H-M | R-F-M | R-F-M | R-B-M |
    ---------------------------------------------------------------
R8  | R-D-M | R-D-M | R-E-M | R-C-M | R-E-M | R-A-M | R-D-M | U-C-M |
    ---------------------------------------------------------------

=====> TURN 1:
---> Do you want to view all surfaces of a box? [1] Yes or [2] No?  1

Please enter the location of the box you want to view:  R1-C2     (Can also accept "1-2")
    -----
    | C |
-------------
| H | B | D |
-------------
    | F |
    -----
    | H |
    -----

---> TURN 1 - FIRST STAGE:
Please enter the location of the edge box you want to roll: R5-C3
INCORRECT INPUT: The chosen box is not on any of the edges. Please reenter the location: R1-C1

The chosen box can be rolled to either [1] right or [2] downwards: 1
The chosen box and any box on its path have been rolled to the right. The new state of the box grid:

        C1      C2      C3      C4      C5      C6      C7      C8
    ---------------------------------------------------------------
R1  | R-B-M | R-H-M | R-A-M | R-E-M | R-A-M | U-G-M | R-G-M | R-D-M |
    ---------------------------------------------------------------
R2  | R-F-M | R-B-M | R-C-M | R-H-M | R-B-M | R-H-M | R-B-M | R-H-M |
    ---------------------------------------------------------------
R3  | R-F-M | R-B-M | R-B-M | R-B-M | R-D-M | R-D-M | R-D-M | X-E-O |
    ---------------------------------------------------------------
R4  | U-H-M | R-D-M | R-E-M | R-H-M | R-G-M | R-E-M | R-C-M | R-A-M |
    ---------------------------------------------------------------
R5  | R-A-M | R-A-M | R-F-M | U-A-M | R-G-M | R-E-M | R-B-M | R-G-M |
    ---------------------------------------------------------------
R6  | R-C-M | R-B-M | R-D-M | R-B-M | R-E-M | R-C-M | R-E-M | R-A-M |
    ---------------------------------------------------------------
R7  | R-D-M | R-G-M | R-F-M | X-D-O | R-H-M | R-F-M | R-F-M | R-B-M |
    ---------------------------------------------------------------
R8  | R-D-M | R-D-M | R-E-M | R-C-M | R-E-M | R-A-M | R-D-M | U-C-M |
    ---------------------------------------------------------------

---> TURN 1 - SECOND STAGE:
Please enter the location of the box you want to open: R2-C2
INCORRECT INPUT: The chosen box was not rolled during the first stage. Please reenter the location:
R1-C2

The box on location R1-C2 is opened. It contains a SpecialTool --> BoxFlipper

Please enter the location of the box to use this SpecialTool: R1-C2
```

The chosen box on location R1-C2 has been flipped upside down. The new state of the box grid:

| | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
|---|---|---|---|---|---|---|---|---|
| R1 | R-B-M | R-D-O | R-A-M | R-E-M | R-A-M | U-G-M | R-G-M | R-D-M |
| R2 | R-F-M | R-B-M | R-C-M | R-H-M | R-B-M | R-H-M | R-B-M | R-H-M |
| R3 | R-F-M | R-B-M | R-B-M | R-B-M | R-D-M | R-D-M | R-D-M | X-E-O |
| R4 | U-H-M | R-D-M | R-E-M | R-H-M | R-G-M | R-E-M | R-C-M | R-A-M |
| R5 | R-A-M | R-A-M | R-F-M | U-A-M | R-G-M | R-E-M | R-B-M | R-G-M |
| R6 | R-C-M | R-B-M | R-D-M | R-B-M | R-E-M | R-C-M | R-E-M | R-A-M |
| R7 | R-D-M | R-G-M | R-F-M | X-D-O | R-H-M | R-F-M | R-F-M | R-B-M |
| R8 | R-D-M | R-D-M | R-E-M | R-C-M | R-E-M | R-A-M | R-D-M | U-C-M |

=====> TURN 2:
---> Do you want to view all surfaces of a box? [1] Yes or [2] No? 2

Continuing to the first stage...

---> TURN 2 – FIRST STAGE:
Please enter the location of the edge box you want to roll: R1-C8

The chosen box can be rolled to either [1] left or [2] downwards: 2
The chosen box and any box on its path have been rolled downwards until a FixedBox has been reached.
The new state of the box grid:

| | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
|---|---|---|---|---|---|---|---|---|
| R1 | R-B-M | R-D-O | R-A-M | R-E-M | R-A-M | U-G-M | R-G-M | R-A-M |
| R2 | R-F-M | R-B-M | R-C-M | R-H-M | R-B-M | R-H-M | R-B-M | R-B-M |
| R3 | R-F-M | R-B-M | R-B-M | R-B-M | R-D-M | R-D-M | R-D-M | X-E-O |
| R4 | U-H-M | R-D-M | R-E-M | R-H-M | R-G-M | R-E-M | R-C-M | R-A-M |
| R5 | R-A-M | R-A-M | R-F-M | U-A-M | R-G-M | R-E-M | R-B-M | R-G-M |
| R6 | R-C-M | R-B-M | R-D-M | R-B-M | R-E-M | R-C-M | R-E-M | R-A-M |
| R7 | R-D-M | R-G-M | R-F-M | X-D-O | R-H-M | R-F-M | R-F-M | R-B-M |
| R8 | R-D-M | R-D-M | R-E-M | R-C-M | R-E-M | R-A-M | R-D-M | U-C-M |

---> TURN 2 – SECOND STAGE:
Please enter the location of the box you want to open: R2-C8

The box on location R1-C2 is opened. It contains a SpecialTool --> PlusShapeStamp

Please enter the location of the box to use this SpecialTool: R5-C5

Top sides of the chosen box (R5-C5) and its surrounding boxes have been stamped to letter “D”. The new state of the box grid:

| | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
|---|---|---|---|---|---|---|---|---|
| R1 | R-B-M | R-D-O | R-A-M | R-E-M | R-A-M | U-G-M | R-G-M | R-A-M |
| R2 | R-F-M | R-B-M | R-C-M | R-H-M | R-B-M | R-H-M | R-B-M | R-B-O |
| R3 | R-F-M | R-B-M | R-B-M | R-B-M | R-D-M | R-D-M | R-D-M | X-E-O |
| R4 | U-H-M | R-D-M | R-E-M | R-H-M | R-D-M | R-E-M | R-C-M | R-A-M |
| R5 | R-A-M | R-A-M | R-F-M | U-A-M | R-D-M | R-D-M | R-B-M | R-G-M |
| R6 | R-C-M | R-B-M | R-D-M | R-B-M | R-D-M | R-C-M | R-E-M | R-A-M |
| R7 | R-D-M | R-G-M | R-F-M | X-D-O | R-H-M | R-F-M | R-F-M | R-B-M |

```
R8  | R-D-M | R-D-M | R-E-M | R-C-M | R-E-M | R-A-M | R-D-M | U-C-M |
    ---------------------------------------------------------------


=====> TURN 3:
---> Do you want to view all surfaces of a box? [1] Yes or [2] No?  2

Continuing to the first stage...

---> TURN 3 - FIRST STAGE:
Please enter the location of the edge box you want to roll: R7-C8

The chosen box is automatically rolled to left until a FixedBox has been reached. The new state of
the box grid:

        C1      C2      C3      C4      C5      C6      C7      C8
    ---------------------------------------------------------------
R1  | R-B-M | R-D-O | R-A-M | R-E-M | R-A-M | U-G-M | R-G-M | R-A-M |
    ---------------------------------------------------------------
R2  | R-F-M | R-B-M | R-C-M | R-H-M | R-B-M | R-H-M | R-B-M | R-B-O |
    ---------------------------------------------------------------
R3  | R-F-M | R-B-M | R-B-M | R-B-M | R-D-M | R-D-M | R-D-M | X-E-O |
    ---------------------------------------------------------------
R4  | U-H-M | R-D-M | R-E-M | R-H-M | R-D-M | R-E-M | R-C-M | R-A-M |
    ---------------------------------------------------------------
R5  | R-A-M | R-A-M | R-F-M | U-A-M | R-D-M | R-D-M | R-B-M | R-G-M |
    ---------------------------------------------------------------
R6  | R-C-M | R-B-M | R-D-M | R-B-M | R-D-M | R-C-M | R-E-M | R-A-M |
    ---------------------------------------------------------------
R7  | R-D-M | R-G-M | R-F-M | X-D-O | R-D-M | R-H-M | R-G-M | R-A-M |
    ---------------------------------------------------------------
R8  | R-D-M | R-D-M | R-E-M | R-C-M | R-E-M | R-A-M | R-D-M | U-C-M |
    ---------------------------------------------------------------

---> TURN 3 - SECOND STAGE:
Please enter the location of the box you want to open: R7-C7

BOX IS EMPTY! Continuing to the next turn...


=====> TURN 4:

…
…
…


=====> TURN 5:
---> Do you want to view all surfaces of a box? [1] Yes or [2] No?  2

Continuing to the first stage...

---> TURN 5 - FIRST STAGE:
Please enter the location of the edge box you want to roll: R3-C8

The chosen box is automatically rolled to left.
HOWEVER, IT IS FIXED BOX AND CANNOT BE MOVED. Continuing to the next turn...


******** GAME OVER ********

The final state of the box grid:

        C1      C2      C3      C4      C5      C6      C7      C8
    ---------------------------------------------------------------
R1  | R-B-M | R-D-O | R-A-M | R-E-M | R-A-M | U-G-M | R-G-M | R-A-M |
    ---------------------------------------------------------------
R2  | R-F-M | R-B-M | R-C-M | R-H-M | R-B-M | R-H-M | R-B-M | R-B-O |
    ---------------------------------------------------------------
R3  | R-F-M | R-B-M | R-B-M | R-B-M | R-D-M | R-D-M | R-D-M | X-E-O |
    ---------------------------------------------------------------
R4  | U-H-M | R-D-M | R-E-M | R-H-M | R-D-M | R-E-M | R-C-M | R-A-M |
    ---------------------------------------------------------------
R5  | R-A-M | R-A-M | R-F-M | U-A-M | R-D-M | R-D-M | R-B-M | R-G-M |
    ---------------------------------------------------------------
R6  | R-C-M | R-B-M | R-D-M | R-B-M | R-D-M | R-C-M | R-E-M | R-A-M |
    ---------------------------------------------------------------
```

```
R7  | R-D-M | R-G-M | R-F-M | X-D-O | R-D-M | R-H-M | R-G-M | R-A-M |
    ---------------------------------------------------------------
R8  | R-D-M | R-D-M | R-E-M | R-C-M | R-E-M | R-A-M | R-D-M | U-C-M |
    ---------------------------------------------------------------

THE TOTAL NUMBER OF TARGET LETTER "D" IN THE BOX GRID --> 16

The game has been SUCCESSFULLY completed!
```

## Important Notes:

1. You can use standard **java.io** packages to read files. You are also free to use anything inside **java.util** packages (including Collections). Do NOT use other 3rd party libraries.
2. You are expected to **write clean, readable, and tester-friendly** code. Please try to maximize reusability and prevent redundancy in your methods.
3. To increase the readability of the code, you are expected to **comment on** your code as much as possible. You can simply use Copilot to generate comments as well. However, it is up to you to make sure that the comments generated are correct. **Point deductions will be applied in case there are not enough comments.**
4. You should **adhere to object-oriented principles as much as possible**. For example, you should place your files inside a proper package structure instead of putting everything into a single package.
5. We advise you to **plan how to write your code beforehand** since you have access to generative AI tools like Copilot.
6. **The inputs should not be case-sensitive.** For example, inputs "R1-C1" and "r1-c1" should both be accepted.
7. **Your menus are expected to be almost the same as the given example scenario**. Naturally, some situations are not shown in the example. Your menus should also handle them while not straying too far from the menu style shown in the example. **Significant point deductions will be applied if difficulties arise while navigating your menus.**
8. **You are expected to fulfil each of required concepts at the start of the file. Significant point deductions will be applied if any concept is unfulfilled.**
9. **When throwing exceptions, you should always handle them. The application should never terminate due to an exception.**
10. **We provided you with hints regarding some of the interfaces and classes across this document.** However, keep in mind that we did not show every single one in this document. You should think and decide design choices related to these yourselves.
11. **Lastly, you should be able to determine the appropriate use of interfaces versus abstract classes when representing objects such as boxes.**

## Assignment Rules:

1. In this lecture's homework, cheating is not allowed. If cheating has been detected, the homework will be graded as 0 and there will be no further discussion on this.

2. You are expected to submit your homework in groups. Therefore, only one of you is sufficient to submit your homework.
3. Make sure you export your homework as a Visual Studio Code Java project. You can use other IDEs as well; however, you must test if it **can be executed** in Visual Studio Code. It is a good idea to check your exported project on another group member's PC.
4. Submit your homework through Microsoft Teams.
5. Your exported Java Project should have the following naming format with your assigned group ID (which will be announced on MS Teams) as given below:

   **G05_CENG211_HW4**

   Also, the zip folder that your project is in should have the same name.

   **G05_CENG211_HW4.zip**
6. Please beware that if you do not follow the assignment rules for exporting and naming conventions, **you will lose points**.
7. Please be informed that your submissions may be anonymously used in software testing and maintenance research studies. Your names and student IDs will be replaced with non-identifying strings. If you do not want your submissions to be used in research studies, please inform the instructor (Dr. Tuğlular) via e-mail.