

CENG211 – Programming Fundamentals
Homework #2

In this homework, you are expected to implement a “Digital Treasure Hunt” in Java.
You should fulfill the concepts of:

- Defining Classes
- Constructors, Getters & Setters
- ArrayList
- Inheritance

The Digital Treasure Hunt game is based on a 20x20 Map, where each cell contains a MapItem. All objects in the game, such as the Player, Booster, and Breaker, are types of MapItem. The Booster objects, which help the player by giving points or lives, include Coin, Diamond, and Treasure. The Breaker objects, which harm the player by reducing points or lives, include Mushroom and Frog. All these game elements inherit properties and behavior from MapItem, making them part of the same hierarchy.

The game's Map class has an ArrayList of ArrayLists where each inner ArrayList holds 20 MapItem objects representing the various elements on the Map, such as the player, a booster, and a breaker. The outer ArrayList holds 20 ArrayList objects. The structure of the Map class is given below partially.

```
public class Map {  
    private ArrayList<ArrayList<MapItem>> map;  
    ...  
}
```

Each MapItem is represented with a symbol on the Map. In the following table, each symbol is given. On the Map, there are 1 Booster, 10 Coins, 5 Diamonds, 2 Treasures, 1 Breaker, 5 Mushrooms, 2 Frogs, and a single Player.

Object	MapItem	Booster	Breaker	Coin	Diamond	Treasure	Mushroom	Frog	Player
Symbol	_	O	X	C	D	T	M	F	P

Additionally, each MapItem on the game's Map holds a MapPosition, which consists of x and y coordinates, along with a boolean value indicating whether the position is occupied or not. For example, if a Treasure is placed in the first ArrayList at the third element, its MapPosition will have x = 0 and y = 2 (since ArrayList indices start from 0). All of the MapItems are placed randomly on the map. A Coin, Frog, Diamond etc. cannot be placed on a MapPosition if it is occupied. A standard MapItem's occupancy status (in its MapPosition) is false. The subclass MapItems, such as Booster, Player, Coin, Diamond, etc., change their MapPosition's occupancy status to true.

In this game, the Player starts with 100 points and 2 lives. The Player moves randomly on the Map, meaning that the x and y coordinates are determined randomly with each move. When the Player lands on a MapPosition, that object interacts with the Player in the following way:

- MapItem adds 0 (zero) points and 0 lives
- Coin adds 5 points
- Diamond adds 10 points
- Treasure adds 1 live
- Mushroom deducts 20 points
- Frog deducts 1 live

- Booster has an ArrayList that contains a Coin, a Diamond, and a Treasure.
- Breaker has an ArrayList that contains a Mushroom and a Frog.
- If the Player lands on a MapPosition that is occupied by a Breaker or a Booster, it interacts with all of the objects in the ArrayList of the Breaker or the Booster.

The Player can continue moving as long as both the number of lives and points remain greater than 0. The number of lives and the points could not be negative values. When the Player moves from one MapPosition to another, the position they departed from becomes occupied by a MapItem.

In this game, each time the Player moves, the game Map is printed on the console. When the game is over, the highest score achieved during that play is also displayed. An example game map is given below. As a part of the game, each time the Player's points increase or decrease, a Score is created, recording the updated points, the departed MapPosition, and the landed MapPosition. This Score is then added to a Scoreboard. When the game ends, the Scoreboard is printed to a TXT file, with each Score printed on a new line.

```

_ D _ _ _ C _ D F _ _ _ _ _ D M _ C _ C _
C _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ M _ D P _ _ _ _ _ _ _
_ T _ _ _ C _ _ _ D _ _ _ _ _ _ _ _ _ M
_ _ X _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ F _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ O _ _ _ _ _ _ _ _
_ _ _ M _ _ _ C _ _ _ C _ _ _ _ _ _ C
_ _ _ _ _ _ _ _ _ _ _ _ _ C _ _ _ _ _
_ _ _ _ _ _ _ _ _ T _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ M _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

```

Important Notes:

1. Do NOT request inputs in your app. Printing the map (at each move) and the highest score will be enough.
2. You can use standard **java.io** packages to read files. Do NOT use other 3rd party libraries.
3. You should use **relative** paths (e.g. Files/sample.csv) instead of **absolute** paths (e.g. C:\\user\\eclipse-workspace\\MyProject\\Files\\sample.csv). Please be sure of it, otherwise there will be **no output** of your application and you certainly will **lose points**.
4. To support **Turkish characters**, you may need to change your project's text file encoding to UTF8: Right click on your project (in package explorer) → Properties → Text file encoding → Other → UTF8 → Apply.
5. You are expected to write clean, readable, and tester-friendly code. Please try to maximize reusability and prevent from redundancy in your methods.

References

Assignment Rules:

1. In this lecture's homework, there is no cheating allowed. If any cheating has been detected, they 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 will be able to submit your homework.
3. Make sure you export your homework as an 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.
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 the given below:

G05_CENG211_HW2

Also the **zip folder** that your project in should have the same name

G05_CENG211_HW2.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. Tuglular) via e-mail.