Due Date: 08.12.2025 23:55

CENG211 – Programming Fundamentals
Homework #3

In this homework, you are expected to implement a simple **"Sliding Penguins Puzzle Game App"** in Java.
The submitted homework should fulfill the following concepts of:

- Lists and ArrayLists
- Interfaces
- Abstract Classes
- Inheritance
- Polymorphism
- Enumerations

In this application, you are required to create an application that **simulates a puzzle game involving 3 penguins on an icy terrain**. The terrain can be described as a **10 by 10 grid** of equal-sized squares. The icy surface of the terrain causes the penguins to **slide** on them. There are **hazards** across the terrain that the penguins should be wary of. There are also different types of **food** scattered across the terrain.

The user of the game is **assigned a random penguin** at the start of the game. The goal of the game is to direct the penguin across the terrain so that it collects the **highest possible amount of food** while avoiding the hazards and competing with the other 2 penguins. Each penguin has **4 turns** to move across the terrain. They can move by choosing a direction: ***Up, Down, Left, Right***. At the end of the 4 turns, the penguin with the **food that weighs the most wins the game**. Assume that penguins are able to carry as much food as possible while also being able to measure their weight.

The game should start inside the **SlidingPuzzleApp main class**. The main method of the **SlidingPuzzleApp** class should only **initialize an IcyTerrain** object. The **IcyTerrain** class should **contain the TerrainGrid**. You must utilize List and ArrayLists correctly to represent the TerrainGrid. The IcyTerrain class can also possess menu-related operations as well.

At the start of the game **three penguins should be randomly generated** and one of them is randomly assigned to the player. Each penguin should be placed onto an **edge square** location of the grid. No two penguins can occupy the same square at any point during the game.

Then, **15 hazards are randomly generated** and placed onto the grid. The hazards and penguins cannot occupy the same space as well.

Lastly, **20 food items are placed** onto the grid randomly. Food items and hazards cannot occupy the same square. However, food items can temporarily exist with penguins on the same square, before being taken by the penguins (not shown on the menu since it is instantaneous). When a penguin **reaches a food item**, the penguin always stops on that square.

All penguins, all food items, and all hazards are considered to be ***ITerrainObjects***. All penguins and some hazards tend to **slide on the ice until they hit another object** on their path or fallen of the side of the grid. Once these slidable objects start moving, they keep going in the same linear direction for most of the cases mentioned below. The IcyTerrain grid is **surrounded by water** on all sides.

There are ***4 types of penguins***. When randomly generating penguins, each penguin has an **equal chance** of becoming one of them. So, it is possible to have all 3 penguins of the same type since repetitions are allowed. Give penguins the following names when generating them: ***P1, P2, and P3***.

Each penguin type has **unique action/ability** they can use **once** during the game. During each turn they are given a chance to use their unique action. Remember that all penguins can *slide(),* but they can have different behaviors when sliding. Also, when a penguin slides into another one, the one that was sliding stops at the square right before the collision. The one that was stationary starts sliding instead. *(Simply imagine that movement transfers from one penguin to another.)*

The ***penguin types*** are:

- KingPenguin: When sliding they can choose to **stop at the fifth square** they slide into. If the direction they choose has less than five free squares, this ability is still considered used.
- EmperorPenguin: When sliding they can choose to **stop at the third square** they slide into. If the direction they choose has less than three free squares, this ability is still considered used.
- RoyalPenguin: Before they start sliding, they can **choose to safely move into an adjacent square** (only horizontally and vertically). It is possible to accidentally step out of the grid and fall into water while using this ability (or other similar accidents).
- RockhopperPenguin: Before they start sliding, they can prepare to **jump over one hazard** in their path. This ability is not used automatically and can be wasted if there are no hazards in their path. Also, they can only jump to an empty square. If the square they are about to land is not empty, they fail to jump and collide with the hazard, nevertheless.

There are ***5 types of food items***. Each food item randomly receives a **food type** and a **weight** between *1-5 units*. The chances of each item's type and weight assignment are equal. Assume that food types are independent from weights. Food types are as follows: ***"Krill", "Crustacean", "Anchovy", "Squid", and "Mackerel".***

There are 4 types of ***IHazards***. Two of them can slide on ice while the others cannot. No hazards can move by itself. In rare cases where a sliding hazard comes across a food item, that food item is removed from the game and that hazard continues its movement. In rare cases where a sliding hazard collides into another penguin (that the current turn doesn't belong to), the sliding hazard is stopped in its tracks while the other penguin is unaffected.

When randomly generating hazards, each hazard has an **equal chance** of becoming one of the 4 types. These are the ***4 hazard types***:

- LightIceBlock: This ice block **starts moving in the transmitted direction** after a penguin or another sliding hazard collides into it. Moving penguins or other sliding hazards stop on the square which they were at the moment of collision. A sliding LightIceBlock can fall from the edges. The **colliding penguin is temporarily stunned** due to hitting the ice block and their next turn is automatically skipped.
- HeavyIceBlock: This ice block **cannot be moved**. Anything that collides with it stops in its tracks. The colliding penguin **loses the lightest food item** they are carrying as a penalty. If the penguin is not carrying any food item, it is unaffected.
- SeaLion: A penguin that hits a **SeaLion bounces from the SeaLion and starts sliding in the opposite direction**. Meanwhile, the penguin's movements are transmitted to the SeaLion. So, the SeaLion starts sliding in the initial direction of the penguin. After collision, both animals can fall from the edges and collide into anything else in the IcyTerrain. If a LightIceBlock collides with a SeaLion, the LightIceBlock's movement is transmitted to SeaLion and LightIceBlock stops moving. Then, SeaLion starts sliding.

- HoleInIce: Anything that slides into a HoleInIce, **falls into it**. Naturally, HoleInIce cannot move or slide. If a penguin falls into a HoleInIce, that **penguin is removed from the game**. However, any food items collected are still counted at the end of the game (also applied when falling from the edges). If a LightIceBlock or a SeaLion slides into a HoleInIce, those sliding objects fall to the hole and plug it. When a HoleInIce is plugged, any sliding objects can pass through it without any issues.

The penguins **compete to collect the food items**. However, they do **not know the weight** of the food items before acquiring them. When a penguin falls into the water, its remaining turns are **automatically skipped**. However, their collected food items are still weighed at the end of the game. There can also be **extreme scenarios where a penguin has no valid move**. In these situations, you do not need to check what can be done by the penguin. Instead, the penguin is forced to move in any direction. Even if they end up falling into water or staying at the same square, their turn is still considered to be used.

When simulating the game, the **actions of the non-player penguins should be randomly decided**. Each opposing penguin should aim to choose a direction that will not cause them to end up in the water or come across hazards. So, when randomly choosing an action, they **prioritize one of the 4 directions that directly lead them to food items**. Otherwise, they choose to go in the direction of a hazard except HoleInIce. Only if they have no other choice, they choose to fall into the water.

Each non-player penguin also has a ***30% chance of using their special action*** in each turn. KingPenguins and EmperorPenguins will **simply follow their usual rules**. It is okay if they waste their special action. RoyalPenguins will use their **single step in a random direction** that does not lead them to a Hazard or falling to water, unless they have no other choice. RockhopperPenguins are an exception to 30% chance of using their action. The first time they decide to move in the direction of a hazard, they **will automatically use their action**. Lastly, the order of turns is simply P1-P2-P3.

It is okay if a penguin ends up in an **unwinnable situation** at the start of the game (or during the game). **Our goal is to simply simulate this game.** At the end of the game, each penguin's **total food weights are calculated**. Then, the game leaderboard is announced with the total food amount of each penguin ordered from top to bottom. Naturally, there can be draws at the end of the game as well.

During menu operations, incorrect inputs will simply cause the **game to ask for the input again** until a correct input is entered. A detailed example scenario output is shown below alongside a **legend with the shorthand notations of each object**. You should aim to design your game menus to be almost identical with the example menu below.

Legend for Icy Terrain Grid Menu Notations:

Penguins: `P1, P2, P3`

Food items: `Kr` (Krill), `Cr` (Crustacean), `An` (Anchovy), `Sq` (Squid), `Ma` (Mackerel)

Hazards: `LB` (LightIceBlock), `HB` (HeavyIceBlock), `SL` (SeaLion), `HI` (HoleInIce)

Plugged HoleInIce: `PH`

Example Scenario Output:

```
Welcome to Sliding Penguins Puzzle Game App. An 10x10 icy terrain grid is being generated.

Penguins, Hazards, and Food items are also being generated. The initial icy terrain grid:

-------------------------------------------------------------
| Kr  |     | P1  |     |     | LB  |     | Cr  |     | HB  |
-------------------------------------------------------------
|     | Ma  | SL  |     | Cr  |     | LB  |     |     |     |
-------------------------------------------------------------
| HB  |     | Sq  |     |     | Ma  |     | HI  | Kr  |     |
-------------------------------------------------------------
|     |     |     | SL  | HI  |     |     |     |     |     |
-------------------------------------------------------------
| HI  |     |     |     |     | An  |     |     | Sq  |     |
-------------------------------------------------------------
|     | An  |     | Kr  |     |     | Sq  |     | LB  | P2  |
-------------------------------------------------------------
|     |     | HB  |     | Kr  |     |     |     | An  | SL  |
-------------------------------------------------------------
|     |     | Cr  |     |     |     |     |     |     |     |
-------------------------------------------------------------
|     | An  |     |     |     |     | HB  |     | Cr  |     |
-------------------------------------------------------------
| P3  |     |     | LB  | HI  | Kr  |     |     | Cr  | Ma  |
-------------------------------------------------------------

These are the penguins on the icy terrain:
- Penguin 1 (P1): Rockhopper Penguin
- Penguin 2 (P2): Emperor Penguin ---> YOUR PENGUIN
- Penguin 3 (P3): Royal Penguin


*** Turn 1 – P1:

P1 does NOT to use its special action.
P1 chooses to move to the LEFT.
P1 takes the Krill on the ground. (Weight=5 units)
New state of the grid:

-------------------------------------------------------------
| P1  |     |     |     |     | LB  |     | Cr  |     | HB  |
-------------------------------------------------------------
|     | Ma  | SL  |     | Cr  |     | LB  |     |     |     |
-------------------------------------------------------------
| HB  |     | Sq  |     |     | Ma  |     | HI  | Kr  |     |
-------------------------------------------------------------
|     |     |     | SL  | HI  |     |     |     |     |     |
-------------------------------------------------------------
| HI  |     |     |     |     | An  |     |     | Sq  |     |
-------------------------------------------------------------
|     | An  |     | Kr  |     |     | Sq  |     | LB  | P2  |
-------------------------------------------------------------
|     |     | HB  |     | Kr  |     |     |     | An  | SL  |
-------------------------------------------------------------
|     |     | Cr  |     |     |     |     |     |     |     |
-------------------------------------------------------------
|     | An  |     |     |     |     | HB  |     | Cr  |     |
-------------------------------------------------------------
| P3  |     |     | LB  | HI  | Kr  |     |     | Cr  | Ma  |
-------------------------------------------------------------


*** Turn 1 – P2 (Your Penguin):

Will P2 use its special action? Answer with Y or N --> Y
Which direction will P2 move? Answer with U (Up), D (Down), L (Left), R (Right) --> U
P2 stops at an empty square using its special action.
New state of the grid:

-------------------------------------------------------------
| P1  |     |     |     |     | LB  |     | Cr  |     | HB  |
-------------------------------------------------------------
|     | Ma  | SL  |     | Cr  |     | LB  |     |     |     |
-------------------------------------------------------------
| HB  |     | Sq  |     |     | Ma  |     | HI  | Kr  | P2  |
-------------------------------------------------------------
|     |     |     | SL  | HI  |     |     |     |     |     |
-------------------------------------------------------------
| HI  |     |     |     |     | An  |     |     | Sq  |     |
-------------------------------------------------------------
|     | An  |     | Kr  |     |     | Sq  |     | LB  |     |
-------------------------------------------------------------
|     |     | HB  |     | Kr  |     |     |     | An  | SL  |
-------------------------------------------------------------
|     |     | Cr  |     |     |     |     |     |     |     |
-------------------------------------------------------------
|     | An  |     |     |     |     | HB  |     | Cr  |     |
-------------------------------------------------------------
| P3  |     |     | LB  | HI  | Kr  |     |     | Cr  | Ma  |
-------------------------------------------------------------


*** Turn 1 – P3:

P3 chooses to USE its special action.
P3 moves one square to the RIGHT.
P3 chooses to move UPWARDS.
P3 takes the Anchovy on the ground. (Weight=1 units)
New state of the grid:

-------------------------------------------------------------
| P1  |     |     |     |     | LB  |     | Cr  |     | HB  |
-------------------------------------------------------------
|     | Ma  | SL  |     | Cr  |     | LB  |     |     |     |
-------------------------------------------------------------
| HB  |     | Sq  |     |     | Ma  |     | HI  | Kr  | P2  |
-------------------------------------------------------------
|     |     |     | SL  | HI  |     |     |     |     |     |
-------------------------------------------------------------
| HI  |     |     |     |     | An  |     |     | Sq  |     |
-------------------------------------------------------------
|     | An  |     | Kr  |     |     | Sq  |     | LB  |     |
-------------------------------------------------------------
|     |     | HB  |     | Kr  |     |     |     | An  | SL  |
-------------------------------------------------------------
|     |     | Cr  |     |     |     |     |     |     |     |
-------------------------------------------------------------
|     | P3  |     |     |     |     | HB  |     | Cr  |     |
-------------------------------------------------------------
|     |     |     | LB  | HI  | Kr  |     |     | Cr  | Ma  |
-------------------------------------------------------------


*** Turn 2 – P1:

P1 will automatically USE its special action.
P1 chooses to move DOWNWARDS.
P1 jumps over HB in its path.
P1 falls into the water due to HI in its path.
*** P1 IS REMOVED FROM THE GAME!

New state of the grid:

-------------------------------------------------------------
|     |     |     |     |     | LB  |     | Cr  |     | HB  |
-------------------------------------------------------------
|     | Ma  | SL  |     | Cr  |     | LB  |     |     |     |
-------------------------------------------------------------
| HB  |     | Sq  |     |     | Ma  |     | HI  | Kr  | P2  |
-------------------------------------------------------------
|     |     |     | SL  | HI  |     |     |     |     |     |
-------------------------------------------------------------
| HI  |     |     |     |     | An  |     |     | Sq  |     |
-------------------------------------------------------------
|     | An  |     | Kr  |     |     | Sq  |     | LB  |     |
-------------------------------------------------------------
|     |     | HB  |     | Kr  |     |     |     | An  | SL  |
-------------------------------------------------------------
|     |     | Cr  |     |     |     |     |     |     |     |
-------------------------------------------------------------
|     | P3  |     |     |     |     | HB  |     | Cr  |     |
-------------------------------------------------------------
|     |     |     | LB  | HI  | Kr  |     |     | Cr  | Ma  |
-------------------------------------------------------------
```

```
*** Turn 2 – P2 (Your Penguin):


...
...
(GAME CONTINIES UNTIL THE END OF THE FOURTH TURN.)
...
...




***** GAME OVER *****
***** SCOREBOARD FOR THE PENGUINS *****

* 1st place: P2 (Your Penguin)
  |---> Food items: Kr (3 units), Sq (2 units), An (5 units)
  |---> Total weight: 10 units

* 2nd place: P3
  |---> Food items: An (1 units), An (2 units), Kr (4 units), Sq (1 units)
  |---> Total weight: 8 units

* 3rd place: P1
  |---> Food items: Kr (5 units)
  |---> Total weight: 5 units
```

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 "d" and "D" 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. 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.

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_HW3

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

   G05_CENG211_HW3.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.