MarsRovers

**Plateau**

#### **Data**

* **width – int**
  + If the width **is less than or equal to 0,** throw a **IlligalArgumentException** with message: **"Invalid width size."**
* **height – int**
  + If the height **is less than or equal to 0,** throw a **IlligalArgumentException** with message: **"Invalid height size."**

#### **Behavior**

boolean isOccupied()

The **isOccupied()**check the position of every rover in the RoverRepository. If position that is check is the same with the position of any of the rovers, the method **return true** otherwise **false.**

#### **Constructor**

A **Plateau** should take the following values upon initialization:

**(int width, int height, RoverRepository rovers)**

**Rover**

#### **Data**

* **String – name**
  + Hold basic name: “Rover#” + number
* **ArrayDeque<Characters> – commands**
  + If the commands are not valid**,** throw a **IlligalArgumentException** with message: **"** **Allowed commands are: [L, R, M]."**
* **Plateau - plateau**
* **Position - position**
  + If the positions are outside the Plateau**,** throw a **IlligalArgumentException** with message: **"Rover is outside the Plateau."**
  + If the position is inside the Plateau, but there is another Rover on that position, throw a **IlligalArgumentException** with message: **"** **Position {otherRoverX otherRoverY} already occupied by another rover."**
* **Orientation – orientation**
  + If the orientation is not valid**,** throw a **IlligalArgumentException** with message: **"Invalid orientation for the {roverName}."**

#### **Behavior**

void move()

The **move()** method change the position of every rover when called. It changes the position based on the orientation they are facing. Before seting the “oldPosition” to the new one, it checks if the “newPosition” is inside the Plateau boundaries(**throws IlligalArgumentException(“Rover cannot go outside the plateau boundaries.”))** as well as checking if another rover is at that position **(throws IlligalArgumentException(“Position {newPositionX newPositionY} already occupied by another rover.”))**.

void rotate()

The **rotate()** method change the orientation of the rover when called. It changes the orientation based on the command it is called with.

Example:

orientation = **N,** command = **L** -> orientation = **W**

orientation = **S**, command = **L** -> orientation = **E**

orientation = **W**, command = **R** -> orientation = **N**

orientation = **E**, command = **R** -> orientation = **S**

#### **Constructor**

A **Rover** should take the following values upon initialization:

**(Plateau plateau, int coordinateX, int coordinateY, String orientation, String commands)**

**Position**

#### **Data**

* **x – int**
  + If the **x** **is less than 0 or greater that Plateau width,** throw a **IlligalArgumentException** with message: **"** **Invalid X Coordinate. Outside Plateau bounds."**
* **y – int**
  + If the **y** **is less than 0 or greater that Plateau width,** throw a **IlligalArgumentException** with message: **"** **Invalid Y Coordinate. Outside Plateau bounds."**

#### **Behavior**

boolean isInsidePlateau()

The **isInsidePlateau()** method checks the position is inside the boundaries of the Plateau.

Position changePosition(Orientation orientation)

The **changePosition()** method change the position of the rover when called. It changes the position based on the orientation it is facing.

Example:

oldPosition = {3, 3}

orientation = **N**  -> newPosition = **{3, 4}**

orientation = **W** -> newPosition = **{2, 3}**

orientation = **S** -> newPosition = **{3, 2}**

orientation = **E** -> newPosition = **{4, 3}**

#### **Constructor**

A **Position** should take the following values upon initialization:

**(int x, int y)**

**ControllerImpl**

### Data

Plateau - Plateau  
rovers – RoversRepository

### Commands

There are several **commands**, which control the **business** **logic** of the **application**.

**CreatePlatea Command**

Parameters:

* width – int
* height – int

**Functionality**

If Plateau is already initialized, we **throw an IllegalArgumentException** with **the following message:**

**"Plateau already created!"**

If Plateau receives valid parameters it will be initialized successfully and the method **return**  
**the following message:**

**“Successfully created Plateau with width: {width}, height: {height}.”**

**AddRover Command**

Parameters:

* x – int
* y – int
* orientation – String
* commands - String

**Functionality**

If Plateau is not already initialized, we **throw an NullPointerException** with **the following message:**

**"Plateau is not created!"**

If Plateau is already created, and we give valid parameters to method, we create the Rover and add it to the RoverRepository**,** and **return the following message:**

**“Successfully added %s.., {roverName}”**

**ProcessCommands Command**

**Functionality**

If RoverRepository is empty, we **throw an IllegalArgumentException** with **the following message:**

**"No rovers added yet."**

If RoverRepository is not empty, we iterate through all rovers and execute it’s commands and the method **return** **the following message:**

**“Successfully processed commands for all rovers.”**

**Report Command**

**Functionality**

If RoverRepository is empty, we **throw an IllegalArgumentException** with **the following message:**

**"No rovers added yet."**

If RoverRepository is not empty, we iterate through all rovers and get information about each rover, and the method **return** **the following message:**

**Output format for each rover is:**

**“- {roverName}**

**-- Location {roverCoordinateX roverCoordinateX roverOrientation}”**

### Input / Output

Below, you can see the **format** in which **each command** will be given in the input:

* **CreatePlateau** **{width} {height}**
* **AddRover {coordinateX} {coordinateY} {Orientation} {Commands}**
* **ProcessCommands**
* **Report**
* **Exit**

#### **Output**

Print the output from each command when issued. If an exception is thrown during any of the commands' execution, print the exception message.

#### **Examples**

|  |
| --- |
| **Input** |
| **CreatePlateau 5 5 AddRover 1 2 N LMLMLMLMM AddRover 3 3 E MMRMMRMRRM ProcessCommands Report**  **Exit** |
| **Output** |
| **Successfully created Plateau with width:5, height:5.**  **Successfully added Rover#0.**  **Successfully added Rover#1.**  **Successfully processed commands for all rovers.**  **- Rover#0**  **-- Location 1 3 N**  **- Rover#1**  **-- Location 5 1 E**  **Mission accomplished!** |

|  |
| --- |
| **Input** |
| **CreatePlateau 6 6 AddRover 2 2 E M AddRover 3 1 N M ProcessCommands Report**  **Exit** |
| **Output** |
| **Successfully created Plateau with width:6, height:6.**  **Successfully added Rover#0.**  **Successfully added Rover#1.**  **Position {3 2} already occupied by another rover.**  **- Rover#0**  **-- Location 3 2 E**  **- Rover#1**  **-- Location 3 1 N**  **Mission accomplished!** |

|  |
| --- |
| **Input** |
| **CreatePlateau 2 2 AddRover 0 0 N MMM  ProcessCommands Report**  **Exit** |
| **Output** |
| **Successfully created Plateau with width:2, height:2.**  **Successfully added Rover#0.**  **Invalid Y Coordinate. Outside Plateau bounds.**  **- Rover#0**  **-- Location 0 2 N**  **Mission accomplished!** |