

Introduction

Welcome to Mission Mars! With growing waste on Earth and climate change, our current mission is to make Mars a habitable planet. Our initial plan is to make Mars inhabitable by planting vegetation and breeding some cattle. To help us make Mars inhabitable our space robots will perform some activities with the help of some space movers. However, beware of some martian animals present in the terrain. They can't hurt the space robots but don't want their territory claimed by foreign entities. They try to destroy the vegetation and the cattle. Let's try and make Mars habitable.



Happy Coding everyone!

Entities

To understand the mission better, let's look at the entities. Once we have looked at the entities, we will describe actions to be performed on or by the entities later.

Entities

- Space Robots – Space robots are here to make Mars habitable by performing different actions. They can move around but you need to give them directions to move.
- Space Rovers – Space Rovers can also move around when they are given directions, but they can only perform specific actions.
- Vegetation – We want to make Mars habitable by planting some plants and trees for food and making air breathable by recycling oxygen and carbon dioxide. Some plants/trees may grow whereas some trees are already grown and cannot grow further. Some plants can bear fruits/vegetables.
- PlantType – These don't produce anything but make the environment breathable. Plant Type Categories - Rose, Lily, Eucalyptus
- VegetableType – These plants/trees produce fruits/vegetables. Categories - Potatoes, Tomatoes, Onions, Apple, Banana
- Earth Animals – We want to rear some cattle like cows, goats, and sheep. There are dogs to overlook the cattle. Cattle produce something like Milk or Wool. Dogs don't produce anything but guard the cattle.
- Martian Animals - However, there are some Martian animals present as well. They destroy the vegetation and want to harm animals.
- Terrain – The Martian terrain can have some rocks. Rocks could be just plain rocks or mineral rocks. Space rovers can collect these minerals and destroy other rocks.

Though it is not a physical entity, there exists a logical entity that represents an overall measure of habitability. Every time we plant vegetation, breed cattle or collect minerals the habitability should go up. Every time a Martian animal harms the cattle or vegetation the habitability should go down.

Note that we can have more than one quantity of each entity i.e. we can have 1 or more space robots/rovers/plants/animals etc.

Symbols

Different entities are represented by different symbols.

Symbols that represent the Martian land and the space entities are given below

- . represents a valid Martian land
- # defines the boundary of the Martian map
- Z represents a space robot
- X represents a space rover
- H – Martian Animal Heebie
- J – Martian Animal Jeebie

Symbols that represent the entities that are planted/bred/found in the Martian Land map. You should use the name given against each in UPPERCASE in your Habitability Status.

- P – POTATO
- T – TOMATO
- O – ONION
- R – ROSE
- L – LILY
- A – APPLE
- B – BANANA
- E – EUCALYPTUS
- C – COW
- G – GOAT
- S – SHEEP
- D – DOG
- @ – ROCK
- * – MINERAL

Exception Handling

Exceptions are the inability of the program to handle normal scenarios. We should raise appropriate exceptions when our program cannot function normally and handle them accordingly.

- **FileNotFoundException** – If the file is not found then File IO Classes in Java throws this exception. You should also throw this exception and terminate the program. This is a Java Exception that you should use and need not create it by yourself.
- **InvalidFileNotFoundException** – If not all rows in the file are of the same length, then throw a custom exception named InvalidFileNotFoundException and terminate the program.
- **UnknownEntityException** – This should be thrown during the handling of files. If we try to load from the file and find any invalid entities, we should throw this custom exception.
- **InvalidLocationException** – If you go outside the Martian space or hit the boundary while moving, you should throw a custom exception InvalidLocationException and go back to the main menu.

We will discuss these further with use cases in the specifications.

Martian Land Map: File Handlin

A Martian Land Map is shown.

- Load a file given in command-line arguments.
- Load from a default file.
- Load from a named file.

-E.....E.....
R.....P.....E.....E.....
R.....CC.....ZK.....TT.....
R.....GD.....LR.....

possible scenarios in this

Main Menu

Main Menu

The main menu is printed after the file is loaded successfully.

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

The menu items are discussed in a different order so that you can gather a better understanding.

Menu Items 1 (Move Space Robot) & 2 (Move Space Rover)

There will always be at least 1 Space Robot and 1 Space Rover be present in all the maps loaded.

On selecting any of these we can make the space robot or space rover move. This will lead to the move menu. The move menu is discussed in another section in detail. However, there could be more than one space robot or space rover present. You need to select one to move one. See the sample output here.

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 1
There are 2 Space Robot Found. Select
(1) for Space Robot at position (13, 2)
(2) for Space Robot at position (9, 3)
```

The positions are (x,y) coordinates. The top left corner is specified as (0,0).

Tip: This is similar to specifying array indices. The array indices start from 0:0. The first index specifies a y because it represents rows that goes vertically, and the second index specify an x coordinate representing column growing horizontally.

With the default map loaded, you can see there are two Space Robots (2)

```
#####
#.....P.....E...#
#..cc.....ZE.....TT...#
#..GB.....Z.....LR...#
#####
```

- In the third row (i.e. y coordinate is 2) and 14th column (i.e. x coordinate 13)
- In the fourth row (i.e. y coordinate is 3) and 10th column (i.e. x coordinate 9)

In this case you can see there is only one Rover and no Martian animal. If we select menu item 2 to move the Space Rover below sample output (without grammatical consideration for ease of implementation) should be shown

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 2
There are 1 Space Rover Found. Select
(1) for Space Rover at position (14, 2)
```

Once you have selected a number for space robots/space rover/martian animal in menu 3, you can show the move menu like this.

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 3
SpaceRover can move in the following directions
(1) to move north,
(2) to move west,
(3) to move east,
(4) to move south,
(5) to move north-west,
(6) to move south-west,
(7) to move north-east,
(8) to move south-east,
(9) to go back to main menu
Please enter a direction.
```

Menu item 3 (Move Martian Animal)

In this case, assume that at least one Space Robot and one Space Rover are present in the various maps loaded but there may or may not be a Martian animal present in those maps. In case no Martian Animal Found and option 3 is selected then you must print an error and return to the main menu.

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 3
No martian animal found to move.
```

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

In case a martian animal is found you can further load the move menu and make the martian animal move. Move menu is discussed in a separate section.

Menu item 4 (Print Habitability Status of current map loaded)

Once the map is loaded you may or may not get some habitability stats based on the entities in the map. This will also lead you to a score. Also, when your space entities perform some actions, it will increase or decrease your habitability score as well. You must keep track of it and update the HabitabilityStatus for various actions performed. For menu item 4, you must print the current updated habitability status.

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 4
Habitability Status
```

```
#####
PGMXTD = 1
CALYPTUS = 2
LEAF = 1
ROSE = 1
COW = 2
GOAT = 1
BEE = 1
BEEATO = 2
```

```
Total Habitability Score: 37
```

```
Please enter
```

```
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 4
```

Enter a Filename for saving Martian Land Map

```
> resources/v1.out
```

```
Terminating the mission for now. See you next time.
```

Output of v1.out

```
#####
#.....P.....E...#
#..cc.....ZE.....TT...#
#..GB.....Z.....LR...#
#####
```

The output of the habitability log file is up to you to maintain. You must keep appending the log and then show the output in menu item 5.

In case you cannot create a file you must print these errors and exit-

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 4
```

Enter a filename for saving Martian Land Map

```
> 
```

Cannot create file for Martian Land Map.

Terminating the mission for now. See you next time.

Or in case of habitability log file

```
Cannot create file for Habitability Status Log.
```

In case you found the file but are unable to write to the file, print these errors and exit

```
Cannot write Martian Land Map to the file.
```

```
Or
```

```
Cannot write Habitability Status Log to a file.
```

Menu Item 5 (Print Habitability Status from the logs)

Since you can load and update the same Martian Map or different Martian Maps with each run of the program, we want to save the habitability status each time just to record our log of work. This file can be loaded either

- A command line argument is passed

```
$ java v1.java
$ java MarsHabitatApplication --f resources/v1.in --l resources/habitability.log
```

- A default file named habitability.log is found in the resources folder.

Once that is found you can print the logs with the number of runs mentioned. Here is a sample output.

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 4
```

Program Run 1

Habitability Status

```
#####
CALYPTUS = 2
LEAF = 1
ROSE = 1
COW = 2
GOAT = 1
BEE = 1
BEEATO = 2
```

```
Total Habitability Score: 35
```

Program Run 2

Habitability Status

```
#####
CALYPTUS = 2
LEAF = 1
ROSE = 1
COW = 2
GOAT = 1
BEE = 1
BEEATO = 2
```

```
Total Habitability Score: 37
```

Please enter

```
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 4
```

Enter a filename for saving Martian Land Map

```
> 
```

Cannot create file for Martian Land Map.

Terminating the mission for now. See you next time.

Or in case of habitability log file

```
Cannot create file for Habitability Status Log.
```

In case you found the file but are unable to write to the file, print these errors and exit

```
Cannot write Martian Land Map to the file.
```

```
Or
```

```
Cannot write Habitability Status Log to a file.
```

Menu Item 5 (Print Habitability Status from the logs)

Since you can load and update the same Martian Map or different Martian Maps with each run of the program, we want to save the habitability status each time just to record our log of work. This file can be loaded either

- A command line argument is passed

```
$ java v1.java
$ java MarsHabitatApplication --f resources/v1.in --l resources/habitability.log
```

- A default file named habitability.log is found in the resources folder.

Once that is found you can print the logs with the number of runs mentioned. Here is a sample output.

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 4
```

Program Run 1

Habitability Status

```
#####
CALYPTUS = 2
LEAF = 1
ROSE = 1
COW = 2
GOAT = 1
BEE = 1
BEEATO = 2
```

```
Total Habitability Score: 35
```

Program Run 2

Habitability Status

```
#####
CALYPTUS = 2
LEAF = 1
ROSE = 1
COW = 2
GOAT = 1
BEE = 1
BEEATO = 2
```

```
Total Habitability Score: 37
```

Please enter

```
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 4
```

Enter a filename for saving Martian Land Map

```
> 
```

Cannot create file for Martian Land Map.

Terminating the mission for now. See you next time.

Or in case of habitability log file

```
Cannot create file for Habitability Status Log.
```

In case you found the file but are unable to write to the file, print these errors and exit

```
Cannot write Martian Land Map to the file.
```

```
Or
```

```
Cannot write Habitability Status Log to a file.
```

Menu Item 5 (Print Habitability Status from the logs)

Since you can load and update the same Martian Map or different Martian Maps with each run of the program, we want to save the habitability status each time just to record our log of work. This file can be loaded either

- A command line argument is passed

```
$ java v1.java
$ java MarsHabitatApplication --f resources/v1.in --l resources/habitability.log
```

- A default file named habitability.log is found in the resources folder.

Once that is found you can print the logs with the number of runs mentioned. Here is a sample output.

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 4
```

Program Run 1

Habitability Status

```
#####
CALYPTUS = 2
LEAF = 1
ROSE = 1
COW = 2
GOAT = 1
BEE = 1
BEEATO = 2
```

```
Total Habitability Score: 35
```

Program Run 2

Habitability Status

```
#####
CALYPTUS = 2
LEAF = 1
ROSE = 1
COW = 2
GOAT = 1
BEE = 1
BEEATO = 2
```

```
Total Habitability Score: 37
```

Please enter

```
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 4
```

Enter a filename for saving Martian Land Map

```
> 
```

Cannot create file for Martian Land Map.

Terminating the mission for now. See you next time.

Or in case of habitability log file

```
Cannot create file for Habitability Status Log.
```

In case you found the file but are unable to write to the file, print these errors and exit

```
Cannot write Martian Land Map to the file.
```

```
Or
```

```
Cannot write Habitability Status Log to a file.
```

Menu Item 5 (Print Habitability Status from the logs)

Since you can load and update the same Martian Map or different Martian Maps with each run of the program, we want to save the habitability status each time just to record our log of work. This file can be loaded either

- A command line argument is passed

```
$ java v1.java
$ java MarsHabitatApplication --f resources/v1.in --l resources/habitability.log
```

- A default file named habitability.log is found in the resources folder.

Once that is found you can print the logs with the number of runs mentioned. Here is a sample output.

```
Please enter
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 4
```

Program Run 1

Habitability Status

```
#####
CALYPTUS = 2
LEAF = 1
ROSE = 1
COW = 2
GOAT = 1
BEE = 1
BEEATO = 2
```

```
Total Habitability Score: 35
```

Program Run 2

Habitability Status

```
#####
CALYPTUS = 2
LEAF = 1
ROSE = 1
COW = 2
GOAT = 1
BEE = 1
BEEATO = 2
```

```
Total Habitability Score: 37
```

Please enter

```
(1) to move Space Robot
(2) to move Space Rover
(3) to move Martian animals
(4) to print the current habitability stats
(5) to print the old habitability stats
(6) to exit
```

```
> 4
```

Enter a filename for saving Martian Land Map

```
> 
```

Cannot create file for Martian Land Map.

Terminating the mission for now. See you next time.

Or in case of habitability log file

```
Cannot create file for Habitability Status Log.
```

In case you found the file but are unable to write to the file, print these errors and exit

```
Cannot write Martian Land Map to the file.
```

```
Or
```

```
Cannot write Habitability Status Log to a file.
```


Habitability Statistics

Habitability Statistics

Habitability Stats maintains how habitable you have made Mars. Habitability Stats comprises of two things -

- A list of entities on the Martian Land Map and their corresponding count as shown below
- A total habitability score

Habitability Status

=====

```
POTATO = 1
MINERAL = 1
EUCALYPTUS = 2
LILY = 1
ROSE = 1
COW = 2
GOAT = 1
DOG = 2
TOMATO = 2
```

Total Habitability Score: 41

Habitability Score: Potato + Mineral + Eucalyptus + Lily + Rose + Cow + Goat + Dog + Tomato = $1 \times 2 + 1 \times 2 + 2 \times 2 + 1 \times 2 + 1 \times 2 + 2 \times 5 + 1 \times 5 + 2 \times 5 + 2 \times 2 = 41$.



Note that if there is a mineral or rock present on the Martian land map, they don't add to the score until the Space Rover takes some action on it. During initial map load, the habitability score then should be 39 ie no score for mineral added. However, if you are loading a habitability.log file or if the space rover has already collected the mineral then it should be 41.

The name of the entities are mentioned under the Symbols. Habitability score changes as you perform some actions.

How the habitability score changes -

When the map is loaded and if it has some entities like vegetation/cattle, it means it was previously set up to be habitable. You will add a habitability score. Otherwise, with some actions that Space Robot/Rover/Martian Animals take, you will update the habitability score.

- When a plant/veg like a Lily/ Rose/Eucalyptus/Tomato/Potato/Apple/Banana/Onion is planted, increase the score by 2.
- When an earth animal is added including the dog, increase the score by 5.
- When a Space Robot waters a plant or feeds a cattle, the habitability score goes up by 1
- When a Space Rover destroys a rock score increases by 1. When a Space rover collects a mineral, the score goes up by 2. **Note that if there is a mineral or rock present on the Martian land map, they don't add to the score until the Space Rover takes some action on it.**
- When a Martian animal eats a plant, the score decreases by 2
- When a Martian Animal kills and eats a cattle (Sheep/Goat/Cow) then the score decreases by 5.
 - If a Martian Animal kills a dog, the score decreases by 5.
 - If a martian Animal is killed by a dog, the score increases by 7.

Guidelines

Some guidelines for your code

- Enums- When you have different types of an entity, you might want to create a single class and represent different types using Enums.
- Inheritance - Entities that have similar types can be divided into parent and child classes using inheritance. You should write down the common functionality in the parent class and override or add different functionality as per specifications in child classes.
- Interfaces - Interfaces help you define the contracts/functionality that a class should follow. When you want to use same method in different classes which are unrelated, use interfaces. When you have same functionality in similar classes use inheritance.
- Polymorphism - Sometimes same method may have different functionality based on different parameters/data. You must use polymorphism in that case.
- Generics <Optional> - You may or may not use Generics to handle some common functionality between different classes.
- Exception Handing - There are various kinds of exceptions present in the specifications. Try and implement the exceptions the right way.
- File Handling - You may want to create different classes or interfaces for classes handling file read/write. Make sure to catch the exceptions that occurs with File Handling.
- Collections - Collections are great way to store data in a compact manner. Use ArrayList instead of Arrays. Make use of HashMaps if necessary.
- Encapsulation - It is important you write the right method in the right classes. This gains you marks. Encapsulate the data properties and action taken on them in a class.
- Make things private where necessary. If inheritance is used, make things protected. Only getters/setters and methods should be public wherever needed.
- Packages - similar classes are packaged together. For eg - you can package interfaces together or enums together in a package.