# SIX Coding Exercise - SpaceX Dragon Rockets Repository

Instructions

Please provide the implementation of the SpaceX Dragon Rockets repository as a simple library. If you choose to use an LLM to help you, we expect you to explain how you used it.

Guidelines

- Keep it simple. Stick to the requirements (keep in mind edge cases)
- Use an in-memory store solution (just use collections to store information you might require)
- It should be a library, NOT REST API, Web Service or Microservice. Just a simple implementation, no need to use any frameworks
- Focus on quality, use Test-Driven Development (TDD), pay attention to OO design, clean code, and adherence to SOLID principles.
- Code the solution according to your standards, please share your solution with a link to the remote repository (Github, Gitlab, etc.) as we would like to see your progress – commit history is important!
- Add a README.md file where you can make notes of any assumptions or specifications you would like to mention about your solution or approach, including any use of AI.

SpaceX Dragon Rockets repository

You are working in SpaceX, and we would like you to develop a new rockets repository that shows the current status of SpaceX dragons.

The repository supports the following operations:

1. Add a new rocket. When the new rocket is added, its status should be “On ground” (it’s the initial score).

2. Assign rocket to mission. Rocket can be assigned only to 1 mission.

3. Change rocket status (see list possible statuses below)

4. Add a new mission. When the new mission is added, its status should be “Scheduled”)

5. Assign rockets to the mission (the mission can have multiple rockets)

6. Change mission status

7. Get a summary of missions by number of rockets assigned. Missions with the same number of rockets should be ordered in descending alphabetical order.

8. Statuses of rockets and mission

a. Rocket can have statuses:

i. “On ground” – initial status, where the rocket is not assigned to any mission

ii. “In space” – the rocket was assigned to the mission

iii. “In repair” – the rocket is due to repair, it implies “Pending” status of the mission

b. Mission can have statuses:

i. “Scheduled” – initial status, where no rockets are assigned

ii. “Pending” – at least one rocket is assigned and one or more assigned rockets are in repair

iii. “In Progress” – at least one rocket is assigned and none of them is in repair

iv. “Ended” – the final stage of the mission, at this point rockets should not be assigned anymore to a mission

Example

Having below data in the system:

- Mars – Scheduled – Dragons: 0
- Luna1 – Pending – Dragons: 2
  - Dragon 1 – On ground
  - Dragon 2 – On ground
- Double Landing – Ended – Dragons: 0

- Transit – In progress – Dragons: 3
  - Red Dragon – On ground
  - Dragon XL – In space
  - Falcon Heavy – In space
- Luna2 – Scheduled – Dragons: 0
- Vertical Landing – Ended – Dragons: 0

The summary of the data should look like:

- Transit – In progress – Dragons: 3
  - Red Dragon – On ground
  - Dragon XL – In space
  - Falcon Heavy – In space
- Luna1 – Pending – Dragons: 2
  - Dragon 1 – On ground
  - Dragon 2 – On ground
- Vertical Landing – Ended – Dragons: 0
- Mars – Scheduled – Dragons: 0
- Luna2 – Scheduled – Dragons: 0
- Double Landing – Ended – Dragons: 0