

## Question: Submarine Mission Validation System

You are tasked with developing a Java application to validate and process submarine mission details. The system must ensure that only valid submarines with authorized missions and safe operational parameters are allowed.

Requirements:

### 1. Custom Exception:

Implement a custom exception class `InvalidSubmarineException` to handle all validation errors related to submarine operations.

### 2. Utility Class:

Create a class `MarineUtil` with the following methods:

- `boolean validateSubmarineId(String submarineId)` throws `InvalidSubmarineException`  
Validates that the submarine ID follows the format DS-XXX (where X is a digit).
- `boolean validateMissionType(String missionType)` throws `InvalidSubmarineException`  
Validates that the mission type is one of: "Research", "Salvage", or "Maintenance".
- `boolean validateDiveDepth(int depth)` throws `InvalidSubmarineException`  
Validates that the dive depth is between 100 and 10,000 meters (inclusive).
- `boolean validatePressureRating(int depth, int rating)` throws

`InvalidSubmarineException`

Validates that the pressure rating is at least equal to the number of kilometers in the dive depth (rounded up).

- `double calculateOxygenHours(String missionType, int depth)`  
Calculates the oxygen duration (in hours) based on mission type and depth.

The oxygen duration is calculated using the following formula:

$$\text{depper} = (0.4 \times \text{depth}) / 100$$

$$\text{depred} = 150 - (\text{depper} \times 150) / 100$$

$$\text{oxygenHours} =$$

$$\text{depred} - 3 \quad \text{if missionType} = \text{"Salvage"}$$

$$\text{depred} - 2 \quad \text{if missionType} = \text{"Maintenance"}$$

$$\text{depred} - 4 \quad \text{if missionType} = \text{"Research"}$$

### 3. User Interface:

Implement a `UserInterface` class with a main method that:

- Prompts the user to enter mission details in the format:  
SubmarineID:MissionType:DiveDepth:PressureRating  
(e.g., DS-309:Research:5000:6)
- Validates each input using the methods in `MarineUtil`.
- If all validations pass, displays the submarine details and calculated oxygen duration.
- If any validation fails, catches the `InvalidSubmarineException` and displays the error message.

Sample Input and Output:

Sample Input 1:

Enter mission details

DS-309:Research:5000:6

Sample Output 1:

Submarine ID: DS-309

Mission Type: Research

Target Depth: 5000 meters

Pressure Rating: 6

Calculated Oxygen Duration: 116.0 hours

Sample Input 2:

Enter mission details

DS-123:Exploration:2000:3

Sample Output 2:

Error: Unauthorized mission: Exploration

Sample Input 3:

Enter mission details

DS-12A:Research:5000:6

Sample Output 3:

Error: Invalid submarine ID: DS-12A

Sample Input 4:

Enter mission details

DS-309:Maintenance:12000:10

Sample Output 4:

Error: Dangerous depth: 12000 m

Sample Input 5:

Enter mission details

DS-309:Salvage:9000:5

Sample Output 5:

Error: Insufficient pressure rating 5(required: 9 )