ASSIGNMENT

Rupesh Bhochhibhoya

Contents

Week 2 Assignment: Enhancing Functionality with Arrays and ArrayLists 1

Problem Overview and Objectives 1

Instructions 1

Expected Output 2

Deliverables 2

Week 2 Assignment: Enhancing Functionality with Arrays and ArrayLists

Problem Overview and Objectives

In Week 2, you will build upon the foundational classes created in Week 1 by implementing key functionalities using Arrays and ArrayLists. This assignment will help you understand how to manage collections of data effectively within the Cybernetic Organ Management System (COMS). The goal is to add and refine functionality in the Patient and OrganInventory classes while learning how to manipulate data using Arrays and ArrayLists.

Instructions

Total Points: 40

1. Refactor the Patient Class to Use an Array for Installed Organs (10 points)
   - Update the Patient class from Week 1 to manage installed organs using an array (CyberneticOrgan[]), with a fixed maximum capacity of 5 organs.
   - Modify the addOrgan(CyberneticOrgan organ) method to:
     - Add the organ to the array if there is space available.
     - Ensure the method handles cases where the array is full by returning an appropriate message.
   - Implement the getOrganList() method to return an ArrayList<CyberneticOrgan> containing the installed organs.
   - Refactor existing methods to work with the array and handle cases where no organs are installed.
2. Enhance the OrganInventory Class with an ArrayList (12 points)
   - Refactor the OrganInventory class to use an ArrayList<CyberneticOrgan> for storing organs, allowing dynamic resizing as more organs are added.
   - Implement the addOrgan(CyberneticOrgan organ) method to add an organ to the inventory, ensuring that the inventory dynamically grows as needed.
   - Implement the getOrganList() method to return the list of all organs in the inventory.
   - Add a method removeOrgan(String model) that searches for and removes an organ by its model name. Return a confirmation message if the organ is successfully removed, or an appropriate message if it is not found.
3. Implement Search Functionality in OrganInventory (10 points)
   - Implement a method searchOrganByFunctionality(String functionality) in the OrganInventory class that returns an ArrayList<CyberneticOrgan> of organs matching the specified functionality.
   - Use a loop to iterate over the ArrayList, adding matching organs to the result list before returning it.
   - Ensure the method handles cases where no matching organs are found by returning an empty list or a message indicating no matches.
4. Add Sorting to the OrganInventory Class (8 points)
   - Implement a sortOrgansByModel() method in the OrganInventory class that sorts the organs in alphabetical order based on their model name.

- Use an appropriate sorting algorithm (e.g., Collections.sort() with a custom comparator or implement your own sorting logic).
- Return the sorted list as an ArrayList<CyberneticOrgan>.

Expected Output

By the end of this assignment: You should create a simple test scenario in your Main class when run printing following in your console.

```
> java com.cyberorgansystem.Main

Adding organs to inventory...
Added CyberHeartX1 to inventory.
Added CyberEyeV2 to inventory.

Adding organs to patient John Doe...
Added CyberHeartX1 to John Doe's installed organs.

Listing installed organs for John Doe:
- CyberHeartX1: Pumps blood

Searching for organs with functionality 'Enhanced vision'...
Found 1 organ(s):
- CyberEyeV2: Enhanced vision

Sorting organs by model name in inventory...
Sorted organs:
- CyberEyeV2
- CyberHeartX1
```

Deliverables

- Code Submission: Submit your code files via the course's submission portal. You can either provide a link to your GitHub repository or upload the project as a ZIP file.