SCD-Lab

lab#3

Name: Anas Altaf

Roll.no: 22f-3639

Answers:

Task-1

```
package T_1;
import java.util.*;
class Task {
	private String tName;
	private String tId;
	Task(String tId, String tName) {
		this.tName = tName;
		this.tId = tId;
	}
	void setName(String tName) {
		this.tName = tName;
	}
	void setId(String tId) {
		this.tId = tId;
	}
	String getName() {
		return tName;
	}
	String getId() {
		return tId;
	}
}
class Employee {
	private String name;
	private String empId;
	public ArrayList<String> tasksList = new ArrayList<>();
	Employee(String name, String empId) {
		this.name = name;
		this.empId = empId;
	}
	boolean indexOutOfRange(int index) {
```

```
		if (tasksList.isEmpty()) {
			System.out.println("Empty List");
			return true;
		}
		if (index >= tasksList.size() || index < 0) {
			System.out.println("Index out of the Range");
			return true;
		}
		return false;
	}
	void addTask(String taskName) {
		tasksList.add(taskName);
	}
	void updateTask(int index, String taskName) {
		if (indexOutOfRange(index)) {
			return;
		}
		tasksList.add(index, taskName);
	}
	void deleteTask(int index) {
		if (indexOutOfRange(index)) {
			return;
		}
		tasksList.remove(index);
	}
	void displayTask(int index) {
		if (indexOutOfRange(index)) {
			return;
		}
		// System.out.println("Task at Index: " + index);
		System.out.println("Task Name: " + tasksList.get(index));
	}
	void displayAll() {
		if (tasksList.isEmpty()) {
			System.out.println("Empty List");
			return;
		}
		int index = 0;
		for (String task : tasksList) {
			// System.out.println("Task at Index: " + index);
			System.out.println("Task Name: " + tasksList.get(index));
			index++;
		}
	}
	void isExist(String taskName) {
		for (String task : tasksList) {
			int index = 0;
			if (task.trim() == taskName.trim()) {
				System.out.println("Task Exists at Index: " + index++);
```

```
                                        return;
                                }
                        }
                        System.out.println("Task does not Exist");
                }
                public void removeDuplicateTasks() {
                        ArrayList<String> checkingTasks = new ArrayList<>();
                        for (String task : tasksList) {
                                boolean isDuplicate = false;
                                for (String check : checkingTasks) {
                                        if (task == check) {
                                                isDuplicate = true;
                                                break;
                                        }
                                }
                                if (!isDuplicate) {
                                        checkingTasks.add(task);
                                }
                        }
                        tasksList = checkingTasks;
                }
                void SortTasks() {
                        tasksList.sort(null);
                }
}
public class Task_1 {
        public static void main(String[] args) {
                Employee emp = new Employee("Ali Baba", "E001");
                for (int i = 0; i < 10; i++) {
                        emp.addTask("Task" + i);
                }
                // Insert at Index
                emp.tasksList.add(4, "Task 1000");
                emp.displayTask(4);
                System.out.println("Tasks Sorted");
                // check if task exists
                emp.isExist("Task 1000");
                // Adding Null Values
                emp.tasksList.add(null);
                System.out.println("Null added");
                // Clearing the List
                emp.tasksList.clear();
                System.out.println("List Cleared");
                // Adding more tasks
                emp.addTask("Task 3");
                emp.addTask("Task 5");
                emp.addTask("Task 4");
                emp.addTask("Task 4");
                // Sorting List
```

```
                emp.SortTasks();
                emp.displayAll();
                System.out.println("Tasks Sorted");
                // Remove Duplicates
                emp.removeDuplicateTasks();
                emp.displayAll();
                System.out.println("Duplicates Removed");
        }
}
```

Output:

```
Task Name: Task 1000
Tasks Sorted
Task Exists at Index: 0
Null added
List Cleared
Task Name: Task 3
Task Name: Task 4
Task Name: Task 4
Task Name: Task 5
Tasks Sorted
Task Name: Task 3
Task Name: Task 4
Task Name: Task 5
Duplicates Removed
```

Task-2:

```
package T_2;
import java.util.*;
class Inventory {
        private HashMap<String, Integer> stockMap = new HashMap<>();
        void addItem(String sku, Integer stock) {
                if (sku == null || stock == null) {
                        System.out.println("SKU or stock value cannot be null");
                        return;
                }
                stockMap.put(sku, stock);
        }
        void updateStock(String sku, Integer stock) {
                if (sku == null || stock == null) {
                        System.out.println("SKU or stock value cannot be null");
                        return;
                }
```

```
                if (stockMap.containsKey(sku)) {
                        stockMap.put(sku, stock);
                } else {
                        System.out.println("SKU not found");
                }
        }
        void removeItem(String sku) {
                if (sku == null) {
                        System.out.println("SKU cannot be null");
                        return;
                }
                if (stockMap.remove(sku) == null) {
                        System.out.println("SKU not found");
                }
        }
        void retrieveStock(String sku) {
                if (sku == null) {
                        System.out.println("SKU cannot be null");
                        return;
                }
                Integer stock = stockMap.get(sku);
                if (stock == null) {
                        System.out.println("SKU not found");
                } else {
                        System.out.println("Stock for SKU " + sku + ": " + stock);
                }
        }
        void displayAllStocks() {
                if (stockMap.isEmpty()) {
                        System.out.println("Inventory is empty");
                        return;
                }
                for (String sku : stockMap.keySet()) {
                        System.out.println("SKU: " + sku + ", Stock: " + stockMap.get(sku));
                }
        }
}
public class Task_2 {
        public static void main(String[] args) {
                Inventory inv = new Inventory();
                inv.addItem("stockUpperCase", 100);
                inv.addItem("stockLowerCase", 200);
                inv.addItem(null, 50);
                inv.addItem("anSKU", null);
//Updating
                inv.updateStock("stockUpperCase", 150);
                inv.updateStock("invalidSKU", 50);
                inv.updateStock(null, 50);
//Retrieving
```

```
                inv.retrieveStock("stockUpperCase");
                inv.retrieveStock("stockLowerCase");
                inv.retrieveStock(null);
                inv.removeItem("stockUpperCase");
                inv.removeItem(null);
                // Displaying
                inv.displayAllStocks();
        }
}
```

Output:

```
SKU or stock value cannot be null
SKU or stock value cannot be null
SKU not found
SKU or stock value cannot be null
Stock for SKU stockUpperCase: 150
Stock for SKU stockLowerCase: 200
SKU cannot be null
SKU cannot be null
SKU: stockLowerCase, Stock: 200
```

Task-3:

```
package T_3;
import java.util.*;
class UserManager {
        private HashSet<String> usernames = new HashSet<>();
        void addUser(String username) {
                if (username == null || username.isEmpty()) {
                        System.out.println("Invalid username");
                        return;
                }
                if (usernames.contains(username)) {
                        System.out.println("Username already taken");
                } else {
                        usernames.add(username);
                        System.out.println("Username added: " + username);
                }
        }
        void removeUser(String username) {
                if (username == null || username.isEmpty()) {
                        System.out.println("Invalid username");
```

```
                        return;
                }
                if (usernames.remove(username)) {
                        System.out.println("Username removed: " + username);
                } else {
                        System.out.println("Username not found");
                }
        }
        void verifyUsername(String username) {
                if (username == null || username.isEmpty()) {
                        System.out.println("Invalid username");
                        return;
                }
                if (usernames.contains(username)) {
                        System.out.println("Username is taken");
                } else {
                        System.out.println("Username is available");
                }
        }
}
public class Task_3 {
        public static void main(String[] args) {
                UserManager um = new UserManager();
//Use cases
                um.addUser("Bazurg");
                um.addUser("New User");
                um.addUser("");
                um.addUser(null);
//verifiication
                um.verifyUsername("New User");
                um.verifyUsername("");
                um.verifyUsername(null);
//Removing
                um.removeUser("Bazurg");
                um.removeUser("");
                um.removeUser(null);
        }
}
```

output:

```
Username added: Bazurg
Username added: New User
Invalid username
Invalid username
Username is taken
Invalid username
```

```
Invalid username
Username removed: Bazurg
Invalid username
Invalid username
```

Task-4:

Output: