UNITED NATIONS MANAGEMENT

/ 05 - 2025

Huy James Viên Ha

Data Structures & Object Oriented Programming

TABLE OF CONTENTS

United Nations Management for Data Structures & Object Oriented Programming

DESCRIPTION!

This is a small made-up system to manage United Nations operations in an effort to promote their original goal of maintaining international peace and security in the world.

PROJECT **FEATURES**

STAFF.java

SUBCLASSES**:**
Intern.java (+ University)
Diplomat.java (+ Country of Representation)
Administrator.java (+ Department)

MANAGES STAFF WITHIN MISSIONS

Abstract **displayDetails()** where each subclass implements its own @Override
compareTo(Staff staff) sorts by staff name

MISSIONS.java

SUBCLASSES**:**
Humanitarian.java (+ Aid Items)
Peacekeeping.java (+ Troop Count)

Implementation of *Reportable.java* interface for all displayDetails()

JUnit 5 unit testing for all non-void methods

MANAGES MISSION FUNCTION

Abstract **displayDetails**() where each subclass implements its own @Override
addStaff(Staff staff) assigns a Staff object to the mission
removeStaff(Staff staff) removes a Staff object from the mission
generateReport(Mission mission) creates a CSV file with missions attributes (+ subclass attributes with instanceof)
readReport(String path) reads aforementioned CSV file
MissionComparator inner class sorts by id, budget, priority level, and objective (default) respectively

PROJECT **FEATURES**

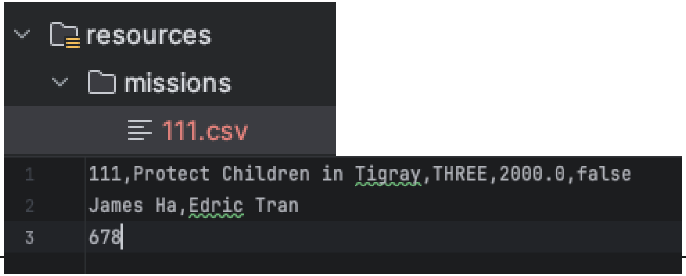
MISSIONCONTROL.java

MANAGES MISSIONS

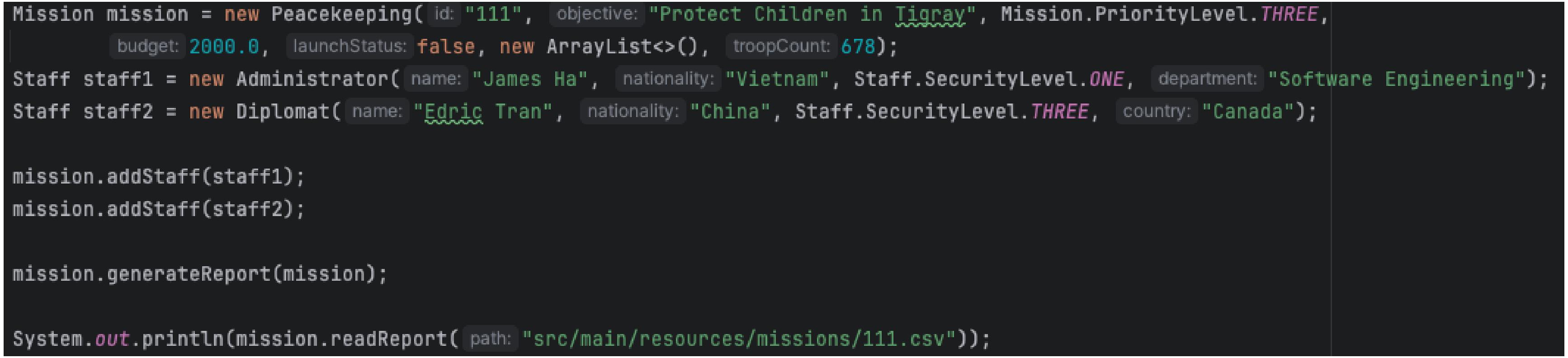
addMission(Mission mission) adds a mission to system list of missions
terminateMission(Mission mission) removes aforementioned mission
launchMission(Mission mission) triggers generateReport(Mission mission) and restricts addStaff(Staff staff) & removeStaff(Staff staff)
FindMissionById(String missionId) uses stream to return mission according to ID, is a launchMission(Mission mission) helper

SCREENSHOTS

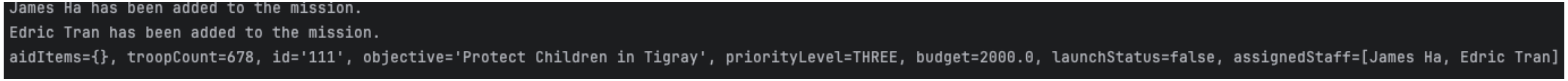
addStaff(Staff staff), generateReport(Mission mission), readReport(String path)
Usage of list, Read/Write Text IO



```
111,Protect Children in Tigray,THREE,2000.0,false
James Ha,Edric Tran
678
```



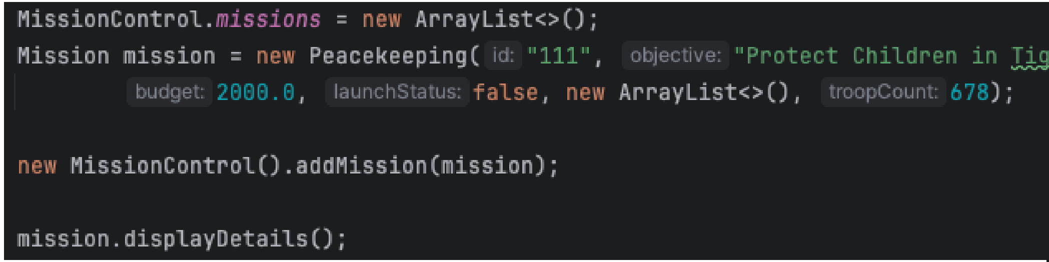
```
Mission mission = new Peacekeeping( id: "111",  objective: "Protect Children in Tigray", Mission.PriorityLevel.THREE,
         budget: 2000.0,  launchStatus: false, new ArrayList<>(),  troopCount: 678);
Staff staff1 = new Administrator( name: "James Ha",  nationality: "Vietnam", Staff.SecurityLevel.ONE,  department: "Software Engineering");
Staff staff2 = new Diplomat( name: "Edric Tran",  nationality: "China", Staff.SecurityLevel.THREE,  country: "Canada");

mission.addStaff(staff1);
mission.addStaff(staff2);

mission.generateReport(mission);

System.out.println(mission.readReport( path: "src/main/resources/missions/111.csv"));
```



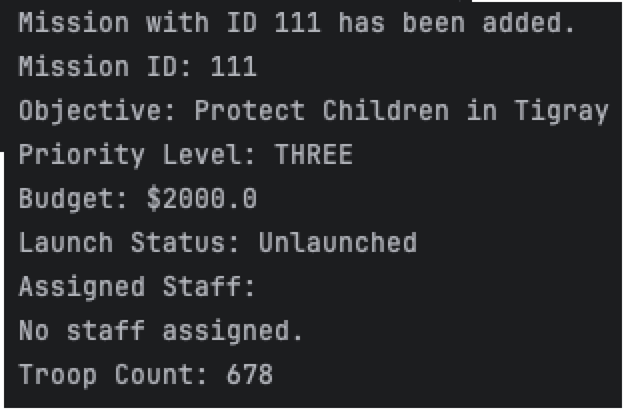
```
James Ha has been added to the mission.
Edric Tran has been added to the mission.
aidItems={}, troopCount=678, id='111', objective='Protect Children in Tigray', priorityLevel=THREE, budget=2000.0, launchStatus=false, assignedStaff=[James Ha, Edric Tran]
```

SCREENSHOTS

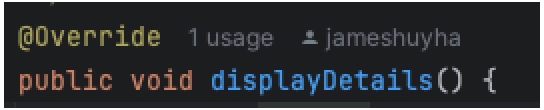
addMission(), displayDetails(), Reportable
Method Overriding, Runtime Polymorphism, Interface



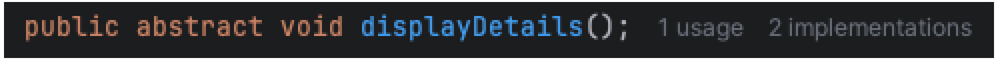
```
MissionControl.missions = new ArrayList<>();
Mission mission = new Peacekeeping( id: "111",  objective: "Protect Children in Tigray", Mission.PriorityLevel.THREE,
         budget: 2000.0,  launchStatus: false, new ArrayList<>(),  troopCount: 678);

new MissionControl().addMission(mission);

mission.displayDetails();
```



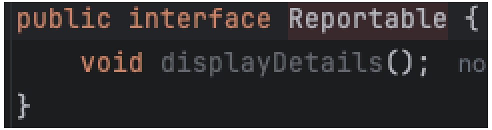
```
Mission with ID 111 has been added.
Mission ID: 111
Objective: Protect Children in Tigray
Priority Level: THREE
Budget: $2000.0
Launch Status: Unlaunched
Assigned Staff:
No staff assigned.
Troop Count: 678
```



```
@Override  1 usage  jameshuyha
public void displayDetails() {
```



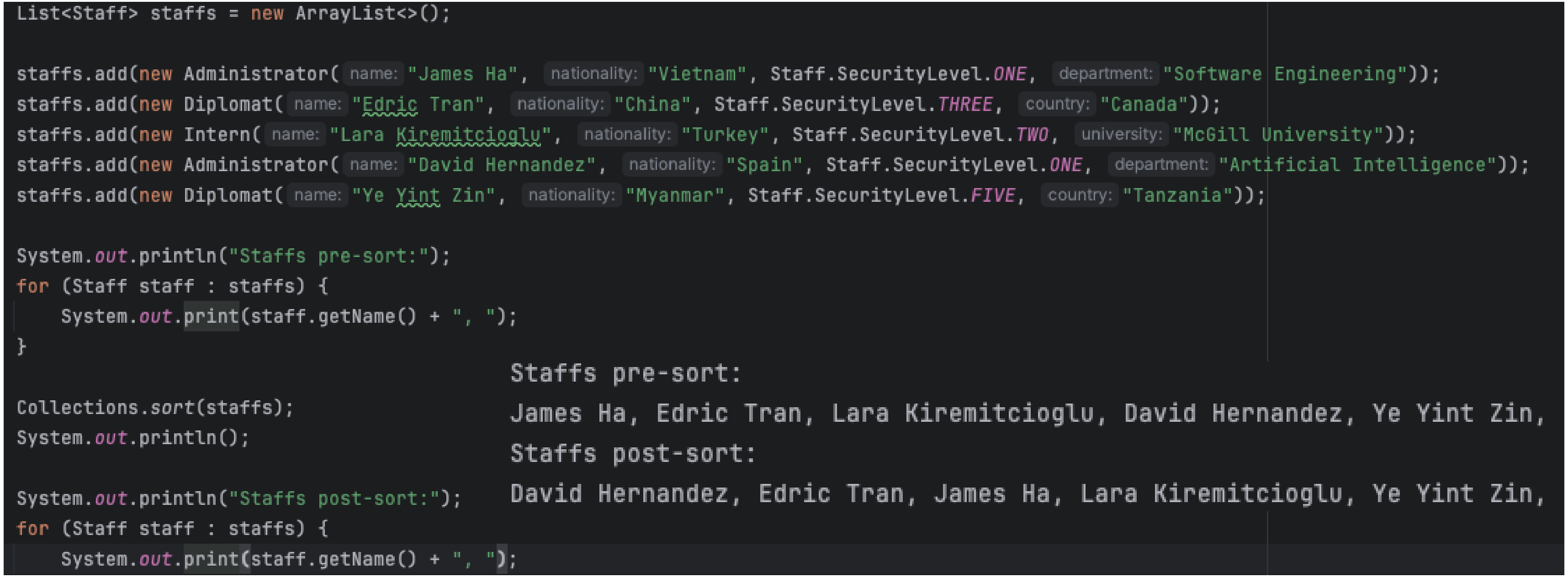
```
public abstract void displayDetails();  1 usage  2 implementations
```



```
public interface Reportable {
    void displayDetails();  no
}
```

SCREENSHOTS

compareTo(Staff staff) according to name
Comparable

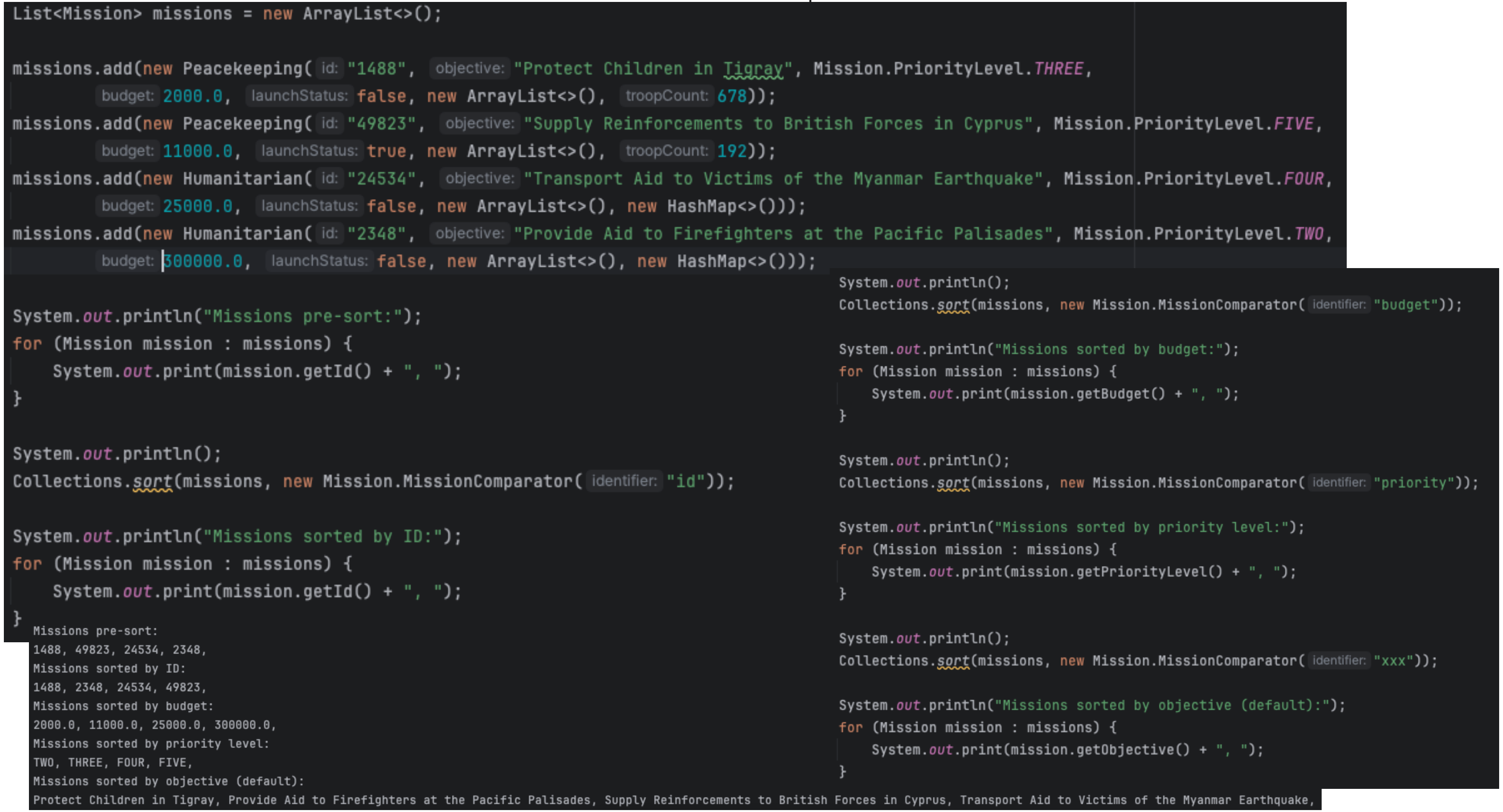


```
List<Staff> staffs = new ArrayList<>();

staffs.add(new Administrator( name: "James Ha",  nationality: "Vietnam", Staff.SecurityLevel.ONE,  department: "Software Engineering"));
staffs.add(new Diplomat( name: "Edric Tran",  nationality: "China", Staff.SecurityLevel.THREE,  country: "Canada"));
staffs.add(new Intern( name: "Lara Kiremitcioglu",  nationality: "Turkey", Staff.SecurityLevel.TWO,  university: "McGill University"));
staffs.add(new Administrator( name: "David Hernandez",  nationality: "Spain", Staff.SecurityLevel.ONE,  department: "Artificial Intelligence"));
staffs.add(new Diplomat( name: "Ye Yint Zin",  nationality: "Myanmar", Staff.SecurityLevel.FIVE,  country: "Tanzania"));

System.out.println("Staffs pre-sort:");
for (Staff staff : staffs) {
    System.out.print(staff.getName() + ", ");
}

Collections.sort(staffs);
System.out.println();

System.out.println("Staffs post-sort:");
for (Staff staff : staffs) {
    System.out.print(staff.getName() + ", ");
```

```
Staffs pre-sort:
James Ha, Edric Tran, Lara Kiremitcioglu, David Hernandez, Ye Yint Zin,
Staffs post-sort:
David Hernandez, Edric Tran, James Ha, Lara Kiremitcioglu, Ye Yint Zin,
```

SCREENSHOTS

MissionComparator inner class according to ID, budget, priority level, and objective (default)
Comparator



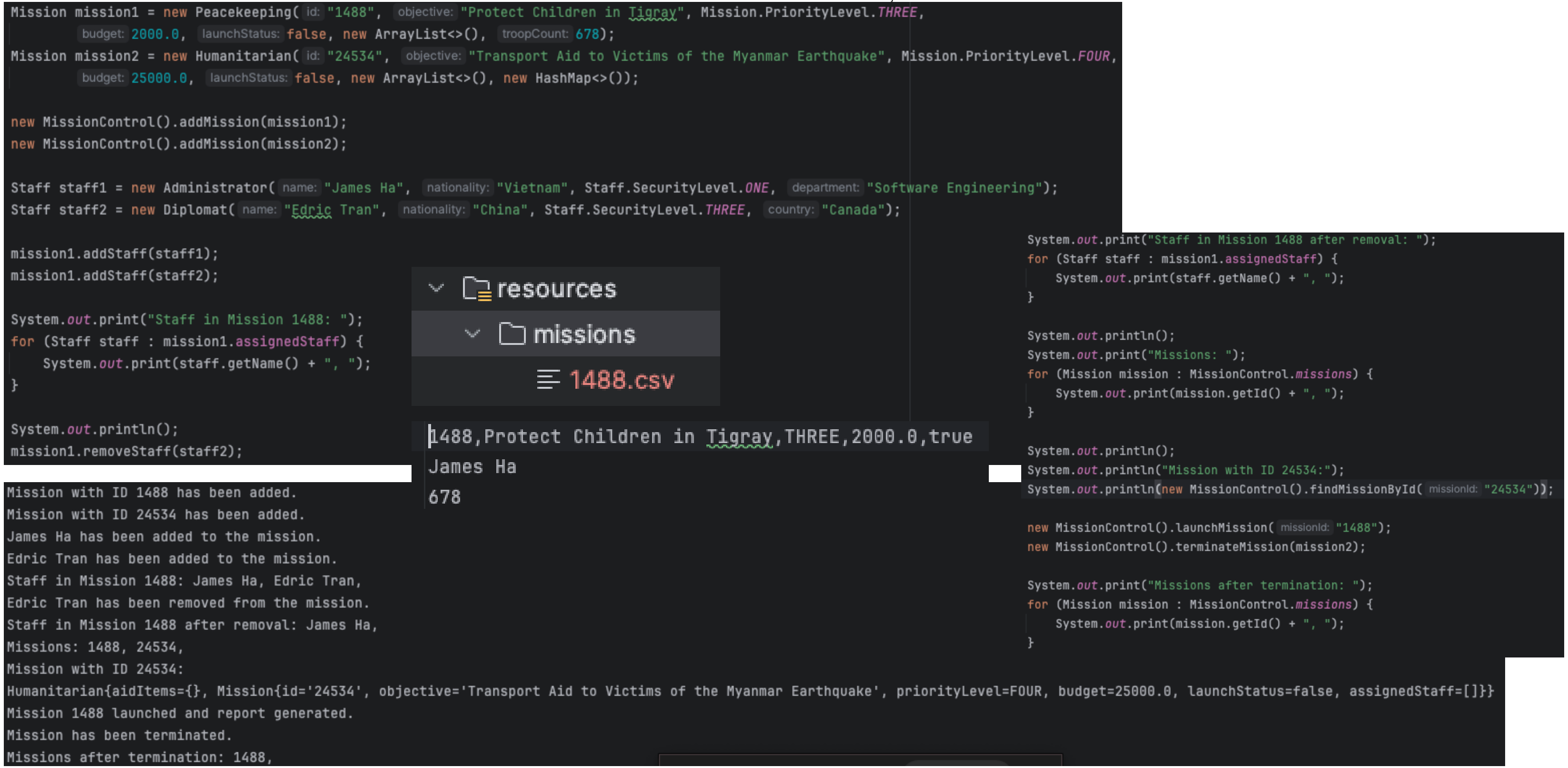
```java
List<Mission> missions = new ArrayList<>();

missions.add(new Peacekeeping( id: "1488", objective: "Protect Children in Tigray", Mission.PriorityLevel.THREE,
        budget: 2000.0, launchStatus: false, new ArrayList<>(), troopCount: 678));
missions.add(new Peacekeeping( id: "49823", objective: "Supply Reinforcements to British Forces in Cyprus", Mission.PriorityLevel.FIVE,
        budget: 11000.0, launchStatus: true, new ArrayList<>(), troopCount: 192));
missions.add(new Humanitarian( id: "24534", objective: "Transport Aid to Victims of the Myanmar Earthquake", Mission.PriorityLevel.FOUR,
        budget: 25000.0, launchStatus: false, new ArrayList<>(), new HashMap<>()));
missions.add(new Humanitarian( id: "2348", objective: "Provide Aid to Firefighters at the Pacific Palisades", Mission.PriorityLevel.TWO,
        budget: 300000.0, launchStatus: false, new ArrayList<>(), new HashMap<>()));

System.out.println("Missions pre-sort:");
for (Mission mission : missions) {
    System.out.print(mission.getId() + ", ");
}

System.out.println();
Collections.sort(missions, new Mission.MissionComparator( identifier: "id"));

System.out.println("Missions sorted by ID:");
for (Mission mission : missions) {
    System.out.print(mission.getId() + ", ");
}
```

```java
System.out.println();
Collections.sort(missions, new Mission.MissionComparator( identifier: "budget"));

System.out.println("Missions sorted by budget:");
for (Mission mission : missions) {
    System.out.print(mission.getBudget() + ", ");
}

System.out.println();
Collections.sort(missions, new Mission.MissionComparator( identifier: "priority"));

System.out.println("Missions sorted by priority level:");
for (Mission mission : missions) {
    System.out.print(mission.getPriorityLevel() + ", ");
}

System.out.println();
Collections.sort(missions, new Mission.MissionComparator( identifier: "xxx"));

System.out.println("Missions sorted by objective (default):");
for (Mission mission : missions) {
    System.out.print(mission.getObjective() + ", ");
}
```

```
Missions pre-sort:
1488, 49823, 24534, 2348,
Missions sorted by ID:
1488, 2348, 24534, 49823,
Missions sorted by budget:
2000.0, 11000.0, 25000.0, 300000.0,
Missions sorted by priority level:
TWO, THREE, FOUR, FIVE,
Missions sorted by objective (default):
Protect Children in Tigray, Provide Aid to Firefighters at the Pacific Palisades, Supply Reinforcements to British Forces in Cyprus, Transport Aid to Victims of the Myanmar Earthquake,
```

SCREENSHOTS

addMission(Mission mission), addStaff(Staff staff), removeStaff(Staff staff), findMissionById(String missionId), launchMission(String missionId), generateReport(Mission mission), terminateMission(Mission mission)
Write Text IO, Lambda



```
Mission mission1 = new Peacekeeping( id: "1488",  objective: "Protect Children in Tigray", Mission.PriorityLevel.THREE,
         budget: 2000.0,  launchStatus: false, new ArrayList<>(),  troopCount: 678);
Mission mission2 = new Humanitarian( id: "24534",  objective: "Transport Aid to Victims of the Myanmar Earthquake", Mission.PriorityLevel.FOUR,
         budget: 25000.0,  launchStatus: false, new ArrayList<>(), new HashMap<>());

new MissionControl().addMission(mission1);
new MissionControl().addMission(mission2);

Staff staff1 = new Administrator( name: "James Ha",  nationality: "Vietnam", Staff.SecurityLevel.ONE,  department: "Software Engineering");
Staff staff2 = new Diplomat( name: "Edric Tran",  nationality: "China", Staff.SecurityLevel.THREE,  country: "Canada");

mission1.addStaff(staff1);
mission1.addStaff(staff2);

System.out.print("Staff in Mission 1488: ");
for (Staff staff : mission1.assignedStaff) {
    System.out.print(staff.getName() + ", ");
}

System.out.println();
mission1.removeStaff(staff2);
```

```
1488,Protect Children in Tigray,THREE,2000.0,true
James Ha
678
```

```
System.out.print("Staff in Mission 1488 after removal: ");
for (Staff staff : mission1.assignedStaff) {
    System.out.print(staff.getName() + ", ");
}

System.out.println();
System.out.print("Missions: ");
for (Mission mission : MissionControl.missions) {
    System.out.print(mission.getId() + ", ");
}

System.out.println();
System.out.println("Mission with ID 24534:");
System.out.println(new MissionControl().findMissionById( missionId: "24534"));

new MissionControl().launchMission( missionId: "1488");
new MissionControl().terminateMission(mission2);

System.out.print("Missions after termination: ");
for (Mission mission : MissionControl.missions) {
    System.out.print(mission.getId() + ", ");
}
```

```
Mission with ID 1488 has been added.
Mission with ID 24534 has been added.
James Ha has been added to the mission.
Edric Tran has been added to the mission.
Staff in Mission 1488: James Ha, Edric Tran,
Edric Tran has been removed from the mission.
Staff in Mission 1488 after removal: James Ha,
Missions: 1488, 24534,
Mission with ID 24534:
Humanitarian{aidItems={}, Mission{id='24534', objective='Transport Aid to Victims of the Myanmar Earthquake', priorityLevel=FOUR, budget=25000.0, launchStatus=false, assignedStaff=[]}}
Mission 1488 launched and report generated.
Mission has been terminated.
Missions after termination: 1488,
```

CHALLENGES

1

MISSIONCONTROL

Had to implement an unplanned class to manage Missions, since I realized I was adding staff to nonexistent Missions that I wasn't creating

2

METHOD LOCATION

Methods, such as generateReport() and readFromFile() had to be renamed and moved around, because of difficulties visualizing their locations while doing Deliverable One

3

UNIMPLEMENTED

DataManager.java for reading and writing Text IO for mission reports was merged with Mission.java

assignTo(Mission mission) in Staff.java was a duplicate of addStaff(Staff staff) in Mission.java and was thus unnecessary, deleted

4

GIT

Difficulty setting up and working with Git and GitHub on outdated MacOS Catalina

5

TEST CASES

Difficulty creating JUnit test cases without implementing methods, particularly with visualizing output of reading CSV

LEARNING OUTCOMES

1

GIT

Usage of Git and GitHub for future programming projects, README.md

2

CREATIVE DIRECTION

Using my own creativity and ideas to apply to my programming, not having to follow a specific rulebook like for our assignments & exams

3

TIME

Projects take considerable time, planning (UML diagram) and breaks (commits) in between to be successful

4

UNITED NATIONS

Understood the basic systems of management for the United Nations and for most international organizations

5

TEXT IO

How to write & read Text IO according to varying and new situations that I create (e.g. reading a map (aidItems))

6

NEW TOOLS

findFirst() & orElse(null) for lambda expressions
assertTrue & assertFalse for Assertions

THANKS!

THANKS FOR THIS SEMESTER :)

You are a wonderful teacher, and I hope to see more of you in the future. Have a wonderful summer, Mr. Wang!

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon** and infographics & images by **Freepik**