Star Exam

Captain's log, star date 201511090944. We have found an alien life-form unknown by the Stelar Federation. They call themselves Candennarians. They look friendly but their technology is far more advanced than ours. The overmatch us in number and attack power. They say they wish our cooperation to create a new program to be projected in the hollodeck. This program is a combat simulation that will help with the training of new recruits joining the Candennarian army.

We have selected you, second lieutenant, because of your object-oriented programming skills, to successfully resolve this situation and start as soon as possible diplomatic actions with the Candennarians and manage their introduction inside the Stelar Float.

Lieutenant La Forge has been working together with the alien team and prepared for you the following report. This report contains the critical points to develop for the simulation program. We require that you provide us with:

A working simulator. You must develop the code for the entire simulator. The developed code must be readable and working.

A set of unit tests for the simulator. You must ensure the quality of the simulator by writing unit tests for every functionality.

A description of the design. You must write a report including the design decisions you made while developing. Class diagrams and object diagrams will be accepted as part of this report.

Remember also the following:

Keep your code readable. Use comments and good indentation. Avoid long methods, use several small methods instead. Choose good names for variables, methods, and classes.

Avoid branching/select clauses. Use polymorphism where it is possible.

Avoid repeated code. Use wisely inheritance and delegation.

1 La Forge's Report

The developped simulation consists in spaceships facing each other. A spaceship can be destroyed when receiving an impact from an enemy spaceship. Also, spaceships can attack cities.

Every spaceship is inhabited by its crew and commanded by a unique person, its captain. The captain is in charge of the ship. Each spaceship contains several sub-systems that keep it up and running e.g., the direction system, the motor, the communication system. Each sub-system is managed by one or more members of the crew. We know how many crew members a sub-system requires at least to keep working. Another important information we know about each sub-system is it's manufacturer. We can model the manufacturer for the means of this simulation as a *String*.

We know from the crew members their age and that they are trained by one or more manufacturer before joining a ship. There, crew members learn to handle the sub-systems of the manufacturer that formed them. Additionally, crew members can learn to handle specific sub-systems when they are in the ship. For example, crew member Spock lives in spaceship SZ-141 and was formed by manufacturers Acme, Beta and Omega. He also learnt from his experience in the ship how to handle the refrigeration sub-system. Then, Spock knows how to handle all sub-systems from manufacturers Acme, Beta and Omega and also the refrigeration sub-system of the ship.

Each crew member is assigned to maximum one sub-system. They can be, however, not assigned to any sub-system. We need to be able to assign a free crew member to a sub-system, re-assign it from a sub-system to another sub-system and un-assign him.

A ship also counts with several combat devices. From each of the combat devices we know its manufacturer, its weight, its pursuit points (that are useful to avoid attacks) and its damage points. When a ship uses a combat device, the device is consumed i.e., the device is lost after usage. For this simulation we will include the following kind of combat devices:

Photon torpedos. They are one of the most powerful attack devices. The damage it causes is equals to the number of photons it contains. Its number of pursuit points are the weight of the torpedo times the maximum degree of instantaneous deviation. This maximum degree is specified for each torpedo.

Neutrino torpedos. The pursuit points of these torpedos are obtained by adding up (a) the adaptability to the environment of the torpedo (which is the same for every torpedo) and (b) a plus that is know for each individual torpedo. The damage level of these torpedos results from multiplying its weight 15 times and add to it a minimum damage that is the same for all torpedos.

Faisers. They do not have pursuit points. They always irradiate 230 points of damage.

Additionally, to avoid unfortunate situations, each ship counts with several evasive maneuvers to avoid enemy attacks. Each maneuver is limited to work with attacks from certain manufacturers, and has an associated affectiveness level that is measured in points. A maneuver avoids an attack from a device if the pursuit points of the attacker device are less that the effectiveness level of the maneuver, and also the manufacturer of the attack is one of the manufacturers that the maneuver can handle. A ship can avoid an attack if at least one of its maneuvers can avoid the attack. If that happens, the ship does not receive any damage from the attack.

Finally, in every spaceship there are three indexes that have to be monitored at all times:

1. **Shield level.** Each ship starts with the maximum shield level: 100. When it is attacked, that level goes down. A ship with a shield level is below zero is destroyed
2. **Number of available combat devices.** A ship can have a maximum of a 100 combat devices. When it performs an attack with a device, the device is consumed. A device can also be damaged (that is indicated for each device). Damaged devices are not available for attack.
3. **Percentage of working sub-systems.** A working subsystem is a sub-system that is managed by at least the amount of crew members that it needs. Calculate the percentage as (working sub-systems * 100 / sub-systems).

2 Development Requirements

1. Basic Statistics.
   (a) Obtain the number of non-working sub-systems in a ship.
   (b) Obtain the set of the oldest crew members for each of the sub-systems of a ship. That is, the oldest of the motor sub-system, the oldest of the communications sub-system, etc.
   (c) Obtain the robustness level of a spaceship: That is the mean of the three indexes explained above (shield level, number of available combat devices, percentage of working sub-systems).
   (d) Know if a spaceship is lazy. A spaceship is lazy if it has more unassigned crew members than crew members assigned to a sub-system.
   (e) Know if a spaceship is well organized. A well organized ship has all its sub-systems working and there is no crew member assigned to a sub-system that it cannot handle.
2. Moving Crew Members.

(a) Evacuate a ship. Evacuating a ship consists in making all crew members abandon the ship. The captain only abandons the ship if he considers that the ship does not require any heroic action. A ship requires an heroic action if the captain is ready to sacrifice himself and the robustness of the ship is lower than 5.

(b) Incorporate a group of crew members to the ship. Each member is destined to a sub-system they can handle and that is not working. Crew members that cannot be assigned using this criteria are left unnasigned.

3. Avoiding Attacks.

   (a) Know if a maneuver can avoid a combat device.

   (b) Know if a ship can avoid a combat device.

4. Receiving an impact from an enemy.

   (a) Inform the captain that *the ship received critical damage*. When the captain receives this message, he checks if the robustness of the ship is less than the minimal he would expect. In that case, he orders the evacuation of the ship.

   (b) Receive an impact of a certain points of damage. The ship lowers its shield by the amount specified by the damage. Afterwards, if the shield level is less than 20, we should inform the captain that *the ship received critical damage*.

5. Attack with a ship another ship or city. When a ship attacks a target, it will always use its best combat device for the attack. That is, the combat device that may generate more damage. When a ship is attacked with a device, it will avoid it if it can. If it can not, it will receive an impact of a point for each 15 damage made by the device. When a city is attacked by a device, 200 citizens die from the attack. For each city we know its population. Take into account that once the attack is done, the combat device should be discarded and not used any more.