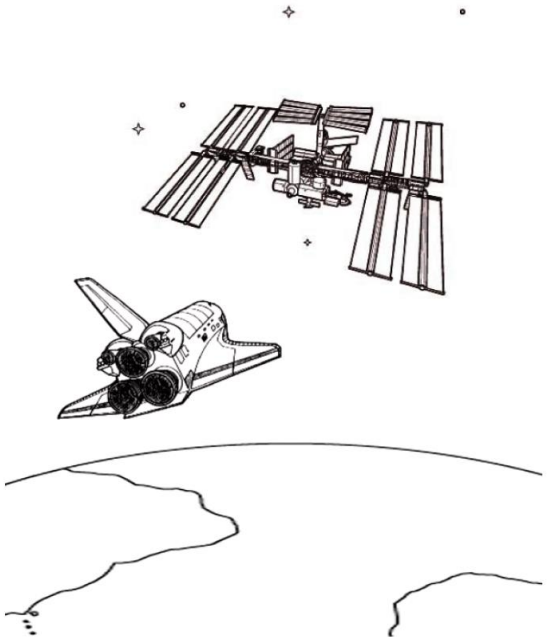
Stack and Queue Assessment

You have been asked to design a computer system to help with the transport of material taken to the international space station.

The material is stored in containers. These are flown to the international space station and unloaded and stored in three different areas depending on cargo type.

Due to the nature of space flight and the design of the space station you will need to implement a stack and a queue.



On Earth the supplies are loaded into pods, items are placed inside a container one on top of another until the container is full. At the other end of the journey containers can be emptied by removing items called pods one at a time starting with the top one. The shuttle carries 2 containers with room for 9 pods.

container1 is emptied first then container2.

container1 always has supplies when going to the space station, on rare occasions container2, although transported may be empty.

There are 3 different types of cargo that is taken to the international space station these are Food, Technology and Personal.

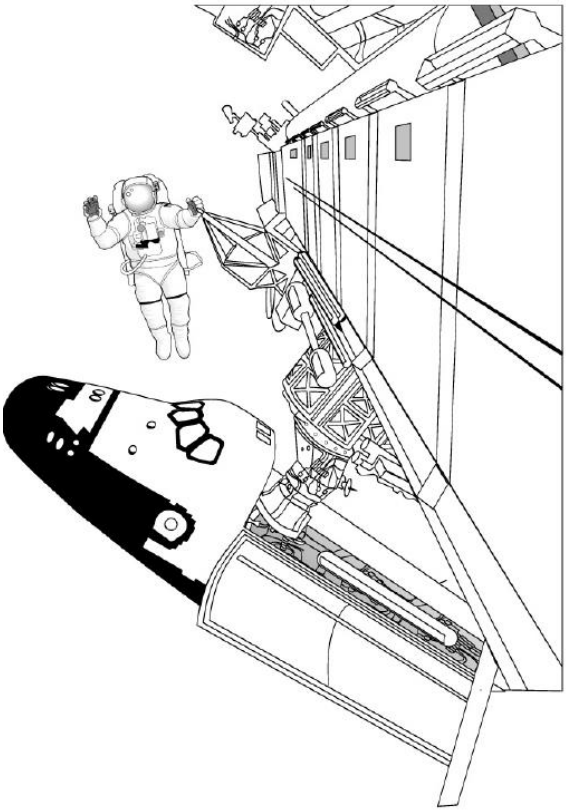
Stack and Queue Assessment

The items are marked with a letter identifying the type of cargo and a 2 digit number. So F14 would contain food and P22 would be personal and T45 would be a Technology pod. Each podmust have a unique number. You cannot have two items marked F14 for example.

When the shuttle reaches the space station it docks with it and the first container is emptied by taking the top pod off of container 1, this is moved through to the loading bay of the space station via a connecting corridor so the first item in will be the first item out at the other side. The operating procedure means that the pod must be fully emptied into the corridor before the items from the container can be stored in the loading bay. Only once a container is emptied into the corridor can the supplies be transferred to the loading bay (use arraylist for the bays). This is done by taking the first item out and taking it to the correct loading bay.

The loading bay contains 3 areas one for each cargo type. These are Food, Technical and personal. When a container comes out of the corridor it is piled up in the relevant bay.

- ***Bay1 [technical]***
- ***Bay2 [personal]***
- ***Bay3 [food]***



Stack and Queue Assessment

The excess material from the previous mission has been removed and returned to Earth. Therefore you can assume the bays are empty before you load the new cargo.

After each loading bay is filled report the number of items stored in BAY1, BAY2 and BAY3

Create a solution implementing the **Stack** and **Queue** Abstract Data Types from the Java util library for the above scenario. Read the data from a file

Use a String to hold the item ID.

Test the program with the following two data runs. Include a test runs of your own.

Data run 1

Pod one F11, F34, P22, T56, F16, T77, P12, F41, T22.

Pod two P19, F39, T92, T36, P36, T75, P15, F48, F88.

Data run 2

Pod one T11, T34, P22, F56, T77, P12, F41, T11.

Pod two EMPTY

If tests are successful run the program with the 7 files found in the missions folder in the teams. Think of theses as 7 separate missions.

Supply printout of the stack code queue code used to implement the solution.

A printout to show the stack and queue at various stages.

- Show the state of the stack and queue before the pods are loaded.
- Show the state of the stack and queue when shuttle reaches space station.
- Show the state of the stack and queue after half the stack has been unloaded.
- Show the state of the stack and queue after the stack has been unloaded.
- Show the state of the stack and queue after the loading bay is used.

You now have to use the following files instead of the Java util library files.

- **ListNode.java**
- **ListQueue.java**
- **ListStack.java**
- **ArrayQueue.java**
- **ArrayStack.java**
- **Queue.java**
- **Stack.java**
- **UnderFlowException.java**

Using the files above remove the java stack and queue library files and replace these files instead. Your program should still run with minimum modification. Retest to ensure that the program still works.

Supply printout of the stack code queue code used to implement the solution. Print the files that you have written not the ones supplied.

A printout to show the stack and queue at various stages.

- Show the state of the stack and queue before the pods are loaded.
- Show the state of the before launch
- Show the state of the stack and queue when shuttle reaches space station.
- Show the state of the stack and queue after the loading bay is used.