Assignment - Equipment Sets and Loan System

[Weight: 15 marks out of the final mark of this course]

Deadline: Apr 22, 2022 (Friday of revision week)

For late submissions, 2% of your original marks will be deducted if you hand in 1-day late (i.e. on Apr 23), 25% for 2-days (i.e. on Apr 24), assignments handed in on or after Apr 25 will get zero mark.

Academic dishonesty is strictly prohibited. The principle concerns whether students get their deserved marks and do not intend to cause unfairness. Dishonesty also involves when one let others have a chance to copy his/her code,

Grading:

Students **must** obtain the following results in sequence:

Phase 1 (100% correct in PASS) ==> Phase 2 (100% correct in PASS) ==> Phase 3

- If you can finish Phase 1 with good programming styles + OO programming skills => up to B+
- If you can finish Phase 2 with good programming styles + OO programming skills => up to A-
- If you can finish Phase 3 with good programming styles + OO programming skills => up to A+

Various test cases are used for each phase (e.g. Phase 1: 1a.txt, 1b.txt, etc..). If you get partial correct, your work is still considered. E.g. If you can pass 1a.txt – 1b.txt only, your grade may be up to C+.

- For "Good Programming Styles", note that proper indentations, code-layout formatting, proper, meaningful naming, well-designed classes, methods, fields are more important than writing comments.
- During manual marking (Apr 23-30), selected students will be asked to meet me for discussion of your work.

Note:

Please apply what you learn from Lab08 - Lab10. You may reuse the code that you worked for Lab08 – Lab10. Reusing these code would not be considered as plagiarism.

Please first finish your program for Lab09, then modify and add the required functionalities for this Assignment.

Assignment Description

Extend the `Lab09-Q2` program to allow adding equipment, borrow equipment, and reserve equipment.

The club owns a collection of equipment. For any given equipment, multiple sets are available. The club members can borrow or reserve depending on availability of the equipment sets.

1. For proper modelling of the system, you should revise the existing classes and add new classes, including the following:

   - Add two classes, `Equipment` and `EquipmentSet`, for the equipment types and the sets of equipment correspondingly. For example, the club may provide 3D printers, google glasses, etc. Then there is one `Equipment` object to stand for 3D printer, and another `Equipment` object to stand for google classes. Suppose there are 3 sets of 3D printers and 5 sets of google glasses, then the `Equipment` object of 3D printer contains an `ArrayList` of three `EquipmentSet` objects for the three 3D printers and the `Equipment` object of google glasses contains an `ArrayList` of five `EquipmentSet` objects for the five sets of google glasses.

   - Each `Equipment` object should have the *Equipment code* (e.g. `E1`, `E2`), and the name (e.g. 3D printer, google glasses), as well as the `ArrayList` of `EquipmentSet` objects.

   - Each `EquipmentSet` object stores the status of the equipment set: available or being borrowed (the loan details), and a list of reservations (or *requests*). We label the equipment set using the code of the equipment appended with "_[*set number*]", like `E1_1`, `E1_2`, where `E1` is the equipment code, `E1_1` means the first set, `E1_2` means the second set, etc.

2. The basic command to borrow an equipment is `borrow [member ID] [equipment code]`. For example, "borrow 001 E1". It means that the member whose ID is 001 would like to borrow a set of the equipment E1 for a period of 7 days (which is the default loan period) starting from the current day. The program should check for the equipment sets of E1 one by one, starting from E1_1, then E1_2 and so on, to mark the equipment set for the member to borrow.

   As a variation, the borrow command also accepts the number of days that the member wants to borrow, instead of the default duration of 7 days. This will be covered in the advanced part of the assignment.

3. The advanced part of this assignment will deal with reservations of equipment. The command to request an equipment is `request [member ID] [equipment code] [start date] [number of days]`. For example, "request 001 E1 06-Mar-2022 3" means that the member whose ID is 001 would like to borrow a set of the equipment E1 for a period of from 06-Mar-2022 to 09-Mar-2022 inclusive. The program should check for the equipment sets of E1 one by one, starting from E1_1, then E1_2 and so on, to identify and mark the available equipment set which is free for borrowing during the period.

4. To guarantee the availability of equipment sets, the club does not allow any member to borrow two or more sets of the same equipment at the same time. That is, anyone who is currently borrowing an equipment set cannot borrow one more set of the same equipment; also the periods of borrowing and requests must not overlap.

5. For simplicity
   - There is no quota limit for loans or requests.
   - There is no restriction on the duration of the period of loans / requests.
   - We don't take care of returning equipment set in this assignment (though in reality it should take place) That is, you may assume that members return equipment on time so that we don't worry about missing the scheduled reservations (accepted requests).
   - As in Lab09, the `startNewDay` command advances the system date. The listing commands should ignore the passed borrowing/request records. However, for simplicity you may assume that the test cases will not come across such issue.

6. The names of command classes should start with "`Cmd`", eg. "`class CmdRegister`", "`class CmdListItems`"

7. We have already implemented the following commands in Lab9:

   i. `register` – Register new members of the club
   ii. `listMembers` – List the members and the count of loans and requests
   iii. `startNewDay` – Start a new day as the system date
   iv. `undo` – Undo one command
   v. `redo` – Redo one command

   For this assignment, you will revise the above methods and handle the following new commands:

   i. `create`: Create an equipment in the collection
   ii. `arrive`: Add a set of an equipment
   iii. `listEquipment`: List the equipment and the count of equipment sets together with the loans

   iv. `borrow`: A member borrows a set of an equipment for a duration starting from the current day
   v. `request`: A member requests to reserve an equipment for a period
   vi. `listMemberStatus`: For each member, list the loans and requests
   vii. `listEquipmentStatus`: For each equipment, list the equipment sets, and the loans and requests of the equipment sets

8. Sorting: in the outputs of listings,
   The ordering of loans and requests should be based on the start day of the concerned periods.
   The ordering of equipment should be based on the equipment codes.
   The ordering of members should be based on the member IDs.

9. You will need to add handling for the following error cases
   a) Member ID already in use (For registering member)
   b) Equipment code already in use (For creation of equipment)
   c) Member not found (For borrowing, requesting)
   d) Equipment not found (For borrowing, requesting, arrival of items)
   e) Member already borrow the same equipment (For borrowing)
   f) Borrow/request period overlaps with a current period (For borrowing/requesting)
   g) Insufficient equipment set to satisfy the command (For borrowing/requesting)
   h) Number of days of period is less than 1 (For borrowing/requesting)
   i) Invalid date format, date is before the current day (For startNewDay, requesting)
   j) Insufficient command arguments (For all commands, e.g. missing member id to register)
   k) Unknown command (Checking in `main()`. The program should terminate.)

   - Most of the above should be done by Exception Handling. You should name all Exception classes with prefix: "`Ex`", eg. "`ExItemIdInUse`", "`ExMemberNotFound`"

   - Please pay attention to the following:

Advantages of Using Exceptions (Week 09 Lecture exercise Q5) == Advantage 1. Separating Error-handling from "Regular" Code (** You should achieve in this assignment (most cases) !!) Advantage 2. Grouping and Differentiating Errors *(Not needed for assignment)* Advantage 3. Propagating Erros up the call stack (** MUST achieve in your assignment !!!)

10. The requirements in each phase and test cases for reference are given on Page 4.
    The given test cases and outputs are to show the functionalities that you need to implement. They also serve to specify the input and output formats.

    However, note that they do not test rigorously for the accuracy of your program.
    For example, your program should be able to handle various sequences of undo-redo of different undoable commands appropriately.
    If your program fails to work appropriately, then your grade will be affected due to incorrect solution (despite that you might have obtained 100% correct in PASS).

    To help verify that students have applied proper solution design, further test cases will be applied and are NOT disclosed before marking. These test cases will be posted after manual marking.

    Q: What if my program cannot pass these test cases just because of some minor problems, e.g. spacing or string spelling problems? I don't want to lose my marks just based on these non-technical issues.

    A: Helena will manually check the cases and will restore the score for you if the discrepancy is not related to problem solving or code design.

11. Concerns about efficiency: Assume that we have up to m=5000 members and n=50000 equipment sets. For modern computers, keeping these records, doing any linear time operations, and providing sorting/searching are not a problem. Therefore, please spend more time and effort on modelling the entities involved in the case study.

Grading and requirements by phases:
(See next page)

Grading and requirements by phases:

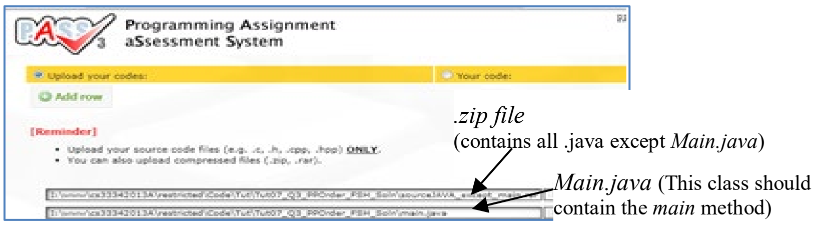
The table below lists the main requirements and test cases added in each phase. For command formats and required outputs, please refer to the styles in Lab08 Q2 to Lab10 Q1, and the contents in the given test cases and outputs on the course web.

Also required:
good programming styles +
OO programming skills
(Please refer to P. 1)

Phases	Execution of commands	Requirements of proper undo/redo	Requirements of Exception handling	Test case for reference (most can be found on courseweb *)	Maximum Grade
Phase 1					
Phase 1.1	Register member, List members	--	Member ID already in use (For register). Insufficient command arguments. Unknown commands	1a.txt	C
Phase 1.2	Start new day	--	The new day should be later than the current day. The format of the input day should be valid.	1b.txt	C+
Phase 1.3	Create new equipment Arrival of equipment set List equipment	--	Equipment code already in use (when creating equipment), equipment code not found (when adding equipment set). Insufficient command arguments.	1c.txt	B-
Phase 1.4	Borrow equipment (7 days loan) List member (revised) List equipment (revised)	--	Member not found, equipment not found. Insufficient command arguments.	1d.txt	B
Phase 1.5	--	undo/redo of Start new day Register member Create equipment Arrive equipment set	--	1e.txt	B+
Phase 2					
Phase 2.1	--	--	Member who is currently borrowing an equipment set cannot borrow one more set of the equipment. No equipment set is available.	2a.txt	A-
Phase 2.2	listMemberStatus and listEquipmentStatus	--	--	2b.txt	
Phase 2.3	--	[assorted]	--	2c.txt	
Phase 3					
Phase 3.1	Borrow equipment for some days Request equipment for a period	--	--	3a.txt	~A
Phase 3.2	--	[assorted]	--	2b.txt	~A+
Phase 3.3	--	--	Periods of borrowing and requests must not overlap. Assorted checking of command arguments.	3b.txt	

Submission:

Please submit them to PASS as shown below:


Programming Assignment aSsessment System

Upload your codes:

Your code:

Add row

[Reminder]

- Upload your source code files (e.g. .c, .h, .cpp, .hpp) **ONLY**.
- You can also upload compressed files (.zip, .rar).

.zip file
(contains all .java except *Main.java*)

Main.java (This class should contain the *main* method)

-- end --