LendItemManager

This assignment combines your knowledge in Programming in one “big” application.

You have already developed some of the needed functions and are asked to reuse them where possible.

LendItem

A LendItem represents something that one can lend (to a friend, for instance) and is used for maintaining a list of items one lends to different people (it’s like one entry in such a list). Just think of books, CDs, DVDs or whatever you lend to people (and tend to forget about). In addition to ES05 this LendItem also has an ID. For the fields consult the class diagram.

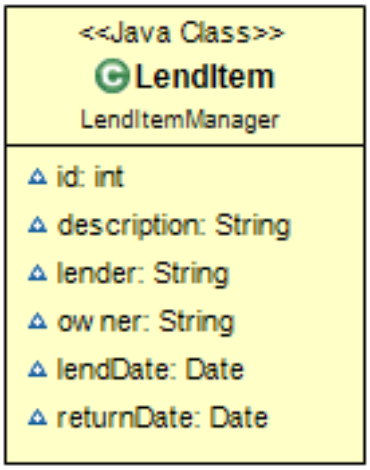

Figure 1: LendItem class diagram (LendItemManager)

Date

A Date represents a Calendar Date starting with the year 1582. For the fields consult the class diagram.

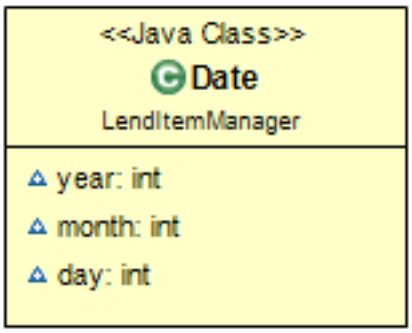

Figure 2: Date class diagram (LendItemManager)

default format for Date: `<yyyy>.<mm>.<dd>`

LendItem functions

In addition to the already implemented functions from the LendItems exercise add the following functions to LendItemManager.Main:

- `public static String lendItemString(LendItem it, int format)`

  as in ES05, except Full Format: `<ID> <description> <lender> <lend date> <return date> <owner>` where

  - ID is three digits wide, left aligned
  - description is exactly 15 characters wide, left aligned
  - lender and owner are exactly 10 characters wide, left aligned
  - dates are in their default format

- `public static String lendItemHeadings(int format)`

  creates the headings for tabular formatted display of LendItems. The format is controlled by parameter format (which has the same meaning as in lendItemString). This function returns a formatted string with the column header. Typical usage of this function is together with lendItemSeparator: This function prints the column headers for the desired format, then lendItemSeparator is called to print a separator line of correct length.

- `public static String lendItemSeparator(int format)`

  creates the separators for tabular formatted display of LendItems. This function returns a String of dashes (-) to separate the header from the data section in tabular display of LendItems.

- `public static String dateString(Date d)`

  As in ES05, except: If a null Date is passed to this function it returns `"<not set>"` (without quotes, but WITH brackets).

LendItemArrayList

A LendItemArrayList is part of a (somewhat abstract) data structure that (together with LendItemArrayList functions) supports position based operations on a collection of LendItems. This is an array-backed implementation of a list (designed with the java.util.List interface in mind). See the class diagram for an overview.

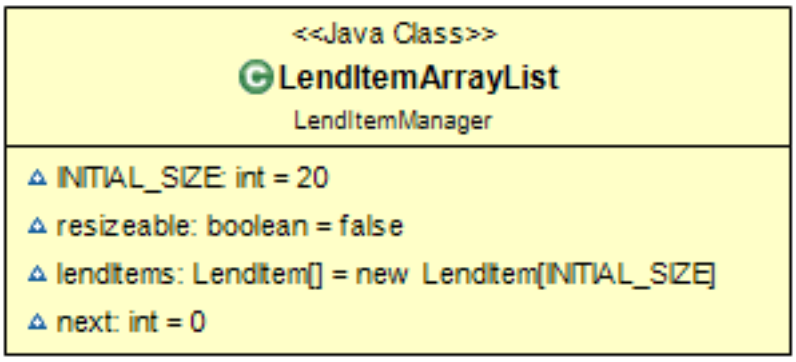

Figure 3: LendItemArrayList class diagram

Some explanation on the fields:

- `INITIAL_SIZE` the initial size of the list (the array). Initial size is 20.
- `resizeable` whether this list is resizeable. By default a list is not resizeable.
- `lendItems` the array of lenditems.
- `next` index of the next free slot in lendItems

LendItemArrayList functions

The LendItemArrayList functions are part of a (somewhat abstract) data structure that (together with LendItemArrayList) supports position based operations on a collection of LendItems. The operations include insertion, removal, retrieval, searching, and sorting for the LendItemArrayList. Implement the following functions in LendItemManager.Main:

- `public static boolean add(LendItemArrayList list, LendItem p)`

  adds a LendItem to the end of the list. A resizeable list doubles its capacity, when an item is to be added to a full list. A non-resizable will not accept another item.

  Returns true if the item was successfully added to the list, false otherwise.

- `public static LendItem remove(LendItemArrayList list, int n)`

  removes a LendItem at a specified (index) position.

  This functions returns the item removed from the list or null if no such item exists. This function leaves no gaps, that means all items after the removed item are shifted one position.

- `public static int list(LendItemArrayList list, int format)`

  Displays all items in the list in a specified tabular format.

  This functions prints a table of all items in the list including header and two horizontal separators. One to separate headers from the body and one after the last item (the body).

  The format (of header, separator and a single LendItem) is controlled by parameter format (which has the same meaning as in LendItemFunctions.lendItemString and its related functions).

  At the end of each line (in the body) the index position of the item is printed in parentheses. After the table a status message is printed: `<n> LendItem(s) in list, <n> free.` with `<n>` representing the actual values (without brackets).

  If the list is empty `List empty.` is printed.

  The function returns the number of items printed in the table.

- `public static void sort(LendItemArrayList list, int order)`

  Sorts the list in a specified order.

  The list is sorted with a stable sorting method.

  The sorting order is controlled by parameter order, which has the same meaning as method in LendItemFunctions.compare (use it!).

- `public static LendItemArrayList filterByDescription(LendItemArrayList list,String desc)`

  Filters a list returning only those items that match a specified description.

  This function scans all items in the list and returns a list of those and only those items whose description contains desc.

- `public static int findByID(LendItemArrayList list, int id)`

  Finds an item in a list by id.

  returns the index of the first occurrence of a lend item with id or -1 if no such item is in the list.

LendItemManager.Main

This class is the application that allows the user to manage Lend items. All LendItem functions and LendItemArrayList functions are in this class. In the main() method implement a simple, text based menu-driven console application with a sortable and searchable list of lend items and multiple display formats. There are no test cases for the main method, instead you create a short demo video (see below for details).

Figure 4: Functions in Main

Requirements

- Menu: The following operations must be selectable:

```
1) list
2) add
3) remove
4) sort
5) filter
6) set format
0) quit
```

- Menu: The programm runs until `quit` is selected
- Add: (interactively) scans a lend item and adds it to the list. (see [[LendItems]] for scanning!)

```
your choice: 2
description: Master of Puppets (CD)
lender: you
owner: me
year: 2015
month: 11
day: 19
1 item added.
```

- List: prints all items currently in the list in the currently selected format (full format by default)

```
 ID description     lender     lend date return date owner
-----------------------------------------------------------------
 19 B_description   Gustav_07  1956.12.16 <not set> null       (00)
 24 B_description   Gustav_01  1985.12.16 <not set> null       (01)
  4 C_description   Gustav_07  1968.02.09 <not set> null       (02)
  9 C_description   Gustav_01  1997.02.16 <not set> null       (03)
 14 C_description   Gustav_04  1926.01.16 <not set> null       (04)
 17 E_description   Gustav_02  1984.03.16 <not set> null       (05)
 22 E_description   Gustav_05  1913.02.16 <not set> null       (06)
  2 F_description   Gustav_02  1996.05.22 <not set> null       (07)
  7 F_description   Gustav_05  1926.04.17 <not set> null       (08)
 12 F_description   Gustav_09  1955.04.16 <not set> null       (09)
 15 H_description   Gustav_07  1912.06.16 <not set> null       (10)
 20 H_description   Gustav_00  1941.05.16 <not set> null       (11)
 25 H_description   Gustav_03  1971.04.16 <not set> null       (12)
  5 I_description   Gustav_00  1954.07.02 <not set> null       (13)
 10 I_description   Gustav_03  1983.06.16 <not set> null       (14)
 18 K_description   Gustav_05  1970.08.16 <not set> null       (15)
 23 K_description   Gustav_08  1999.07.16 <not set> null       (16)
  3 L_description   Gustav_05  1982.10.15 <not set> null       (17)
  8 L_description   Gustav_08  1911.09.16 <not set> null       (18)
 13 L_description   Gustav_01  1941.08.16 <not set> null       (19)
 26 Master of Puppe you        2015.11.19 <not set> me         (20)
 16 N_description   Gustav_09  1998.10.16 <not set> null       (21)
 21 N_description   Gustav_03  1927.10.16 <not set> null       (22)
  6 O_description   Gustav_03  1940.12.23 <not set> null       (23)
 11 O_description   Gustav_06  1969.11.16 <not set> null       (24)
-----------------------------------------------------------------
25 LendItem(s) in list, 15 free.
```

- Remove: scans an ID and removes the lend item with that ID.

```
enter ID of LendItem to be removed: 999
LendItem not found (ID 999).
```

```
enter ID of LendItem to be removed: 1
1 LendItem (ID=1) removed.
```

- Sort: sorts all items with a stable sorting algorithm. Several sorting options are available.

```
available sort options:
1) by lend date
2) by return date
3) by by lender
4) by owner
0) by description
```

see the list sorted by return date:

```
 ID description     lender     lend date return date owner
-------------------------------------------------------------------
  8 L_description   Gustav_08  1911.09.16 <not set> null       (00)
 15 H_description   Gustav_07  1912.06.16 <not set> null       (01)
 22 E_description   Gustav_05  1913.02.16 <not set> null       (02)
 14 C_description   Gustav_04  1926.01.16 <not set> null       (03)
  7 F_description   Gustav_05  1926.04.17 <not set> null       (04)
 21 N_description   Gustav_03  1927.10.16 <not set> null       (05)
  6 O_description   Gustav_03  1940.12.23 <not set> null       (06)
 20 H_description   Gustav_00  1941.05.16 <not set> null       (07)
 13 L_description   Gustav_01  1941.08.16 <not set> null       (08)
  5 I_description   Gustav_00  1954.07.02 <not set> null       (09)
 12 F_description   Gustav_09  1955.04.16 <not set> null       (10)
 19 B_description   Gustav_07  1956.12.16 <not set> null       (11)
  4 C_description   Gustav_07  1968.02.09 <not set> null       (12)
 11 O_description   Gustav_06  1969.11.16 <not set> null       (13)
 18 K_description   Gustav_05  1970.08.16 <not set> null       (14)
 25 H_description   Gustav_03  1971.04.16 <not set> null       (15)
  3 L_description   Gustav_05  1982.10.15 <not set> null       (16)
 10 I_description   Gustav_03  1983.06.16 <not set> null       (17)
 17 E_description   Gustav_02  1984.03.16 <not set> null       (18)
 24 B_description   Gustav_01  1985.12.16 <not set> null       (19)
  2 F_description   Gustav_02  1996.05.22 <not set> null       (20)
  9 C_description   Gustav_01  1997.02.16 <not set> null       (21)
 16 N_description   Gustav_09  1998.10.16 <not set> null       (22)
 23 K_description   Gustav_08  1999.07.16 <not set> null       (23)
 26 Master of Puppe you        2015.11.19 <not set> me         (24)
-------------------------------------------------------------------
25 LendItem(s) in list, 15 free.
```

- Filter: scans a search string and lists all lend items that contain that search string in their description. The result list is displayed in tabular format.

```
enter description: L_de
displaying matches:

 ID description     lender     lend date return date owner
-------------------------------------------------------------------
```

```
  3 L_description   Gustav_05  1982.10.15 <not set> null       (00)
  8 L_description   Gustav_08  1911.09.16 <not set> null       (01)
 13 L_description   Gustav_01  1941.08.16 <not set> null       (02)
-----------------------------------------------------------------
3 LendItem(s) in list, 17 free.
```

- Formatting options: When a specific format is selected, *all* output uses that format. Default format is full format. The following formatting options are available:

```
available options:
1) full format
2) short format
3) csv format
```

- General:
  Your program shall be user friendly, in the sense that it tells the user what was selected, what was performed (including success/failure information).
- Video: create a video with your favorite screen recording software and demonstrate fulfillment of **all** requirements.
  - Technical requirements for the video:
    - 1024x768
    - max 3min runtime
    - size <=70MB
    - playable with VLC
  - Content of the vidoe:
    - demonstrate that all test cases are passed
    - demostrate fulfillment of all (functional) requirements using good test data (for instance to demonstrate correct sorting)
    - point out and explain the section in your code that was hardest for you.

Grading

Necessary conditions:

- runnable submission (no compile errors)
- LendItemArrayListTest, LendItemArrayListTestFunctions passed
- video uploaded

If all necessary conditions are fulfilled, credits are awarded according to the following table.

Feature	Credits	Available tests
menu	2	
list full format	1	
list short format	1	
list csv format	1	
add (unit tests)	5	3
add (in application/video)	2 (valid data/invalid data)	
remove (unit tests)	5	5
remove (in application/video)	2 (existing entry/non-existing entry)	
sort (unit tests)	5	6
sort (in application/video)	3 (all options)	

Feature	Credits	Available tests
filter (unit tests)	5	5
filter (in application/video)	2	
findById (unit tests)	2	5
set format	1	
hardest code	1	
explanations (in video)	1	

Note: Make sure your video's duration is *no longer* than 3 min. Anything demonstrated after those 3 minutes will not be awarded any credit.

Credits for unit tests are calculated based on pass percentage of overall tests per method - passed tests/ available tests *100 (percent)

Demo data

You may use the following code snippet to create (somewhat) useful demo entries.

```
for (int i = 0; i < 25; i++) {
    LendItem li = new LendItem();
    li.id = nextID++;
    li.description = String.format("%c_description", ((int) (i
            * Math.PI * 10000)) % 15 + 'A');
    li.lender = String.format("Gustav_%02d", ((int) (i
            * Math.PI * 1000000)) % 10);
    li.lendDate = new Date();
    li.lendDate.year = 2010 - ((int) (i * Math.PI * 100)) % 100;
    li.lendDate.month = ((int) (i * Math.PI * 1000000)) % 12 + 1;
    li.lendDate.day = ((int) (i * Math.PI * 100000000)) % 28 + 1;
    LendItemArrayListFunctions.add(list, li);
}
```