Technical University of Denmark
DTU Informatics
Assoc. Prof. H. Baumeister

Spring 2016
Prog. Assignment Nr. 6
March 11, 2016

02161: Software Engineering 1

Note

This is an *advanced* exercise and takes more time.

Adding a persistency layer to the library application

The last step for the library application is to to add a persistency layer to the application. To keep the problem simple, we use a rather naive storage for the objects. Media and users are both stored each in their respective files `media.txt` and `users.txt`. Note that addresses don't get their own file; they are stored with the users. The user is represented as follows in `users.txt`

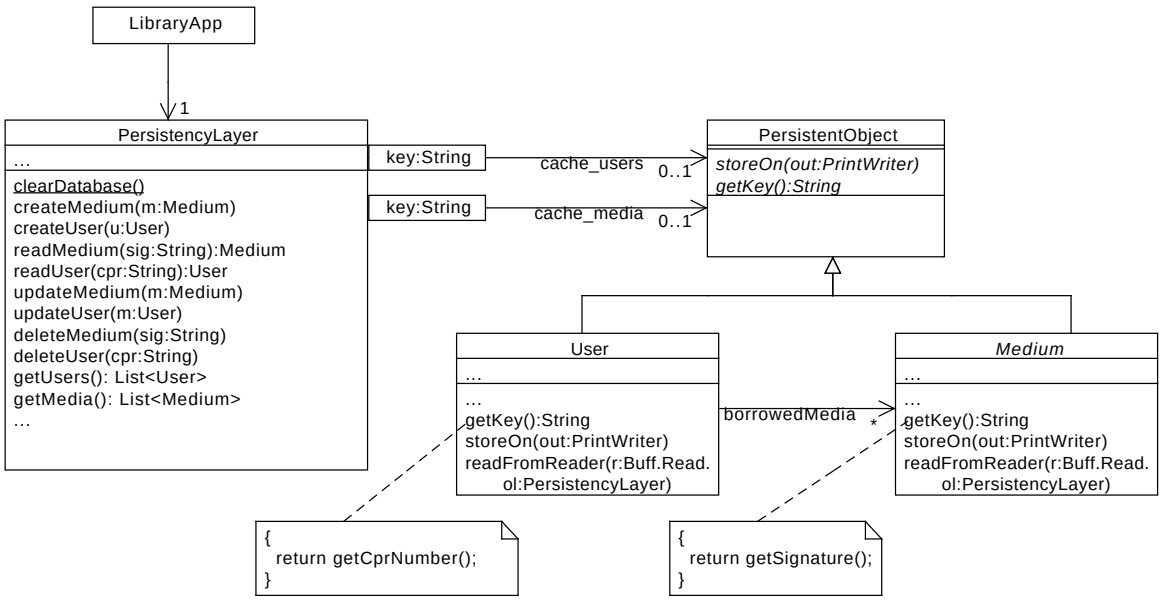
```
dtu.library.app.User
cpr-number
Some Name
a@b.dk
Kongevejen
2120
København
b01
c01
<empty line>
```

First, there is the name of the user class, then the values of the fields `cprNumber`, `name`, `email`. This is followed by the values of the address field `street`, `postNumber`, and `town`. Finally, this is followed by the signatures (one on each line) for the borrowed media (in the example `b01` and `c01`. Note that the fields are followed by an empty line as an end-of-record marker. The storage of media proceeds with a similar structure in `media.txt`, i.e. the name of the concrete medium class and then the fields (starting with the signature) and terminated with an empty line. For example:

```
dtu.library.app.Book
b01
some book author
some book title
Mar 13, 2011
<empty line>
```

Note that, in case that the borrow date is empty, the string "null" (without quotes) should be used. The basic operations of the persistency layer are based on the standard

operations of a database: create new records, read records, update records, and delete records (CRUD).



6.1 Implementation of a persistency layer

Implement the persistency layer based on the tests given in `http://www2.imm.dtu.dk/courses/02161/2015/files/library07.zip` (library07.zip contains the solutions to exercises 1—6 albeit without a full UI) and the diagram above.

- Note: The diagram above is the target design you should reach, but it does not make sense to implement the diagram immediately; instead, when implementing each of the tests, incrementally create the implementation that corresponds to the diagram
- createMedium/createUser adds the representation of a medium/user as described above to the file `media.txt`/users.txt using the operation storeOn of class Medium/class User
- readMedium/readUser reads the records in the file media.txt/users.txt until it finds the medium/user with the correct key (i.e. CPR number or signature)
  - The operations readFromReader in class User and Medium can be used to read the fields for a user and a medium from a buffered reader
- updateMedium(m:Medium)/updateUser(u:User) updates the files media.txt and users.txt by replacing the old record for m/u by the new record.
  - This is done by reading each record from the files and storing them (using storeOn) to new files. Finally, the new files are renamed to the old files (operation rename in class File).

- deleteMedium/deleteUser removes the record with the given signature / CPR number from the data files.
  - This is done similar to the update operations by copying all entries of the old files into new files (without the object to be deleted) and renaming the new files to the old files
- getUsers/getMedia creates a list of users / media from the records in the data files.

6.2 Use the library application together with the persistency layer

1) Refactor the library application to use the persistency layer.
   - Note that some tests needs to be adapted slightly, e.g. to use the clearDatabase operation of the persistency layer to ensure that the tests start with an empty database
2) Create additional tests to make sure that the right information is stored in the database, e.g. that when borrowing/returning a media, that then the data files are updated (i.e. by sending the updateMedium/updateUser message to the persistency layer).
   - There need to be tests to ensure that whenever attributes of users/media are changed that then the corresponding update operation of the persistency layer is called
     * Note that the cpr number of a user and a signature of a medium is read only and should not be able to change once set

More information on the implementation can be found on the Web page of the course.