Technical University of Denmark	Spring 2015
DTU Informatics	Prog. Assignment Nr. 4
Assoc. Prof. H. Baumeister	February 2, 2015

02161: Software Engineering 1

Working with other media besides book

Up to now, the library application is only suited to deal with books, but not CDs, DVDs and journals. The next step will be to introduce CDs into the system and how to borrow them. CDs have a different behaviour from books, as they can only be borrowed for 1 week and not 4 weeks as with books. In addition, the fine need to be payed for overdue books is 40 kroner instead of just 20 kroner for books. Thus this week exercises contains in

a) Adding the possibility to borrow CDs in addition to books

b) Making sure that CDs are already overdue after one week

c) Adding fines for overdue books to the library application

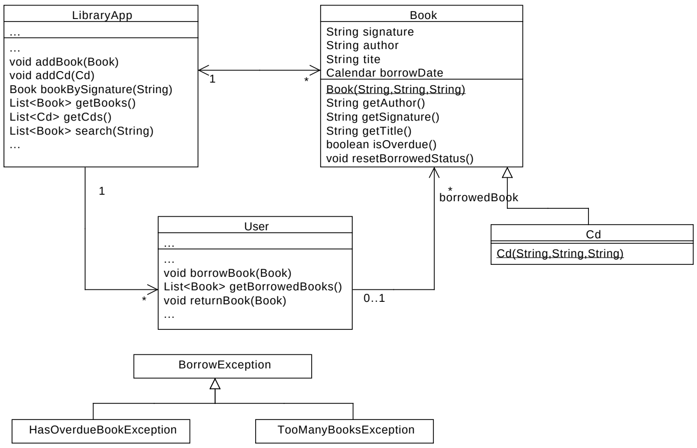
4.1 Add CDs to the library

- Create a test similar to the method testAdminAddBook() in class TestAddBook to test that the adminstrator can add a CD to the library application. That is, the test should 1) successfully log in as adminstrator 2) create a new CD with signature, title, and author 3) call addCd(aMedium) to add the CD to the library application 4) Check via getCds() that the CD is included in the library and has signature, title, and author set correctly.
- The test class should be called TestAddCd
- Create the missing class Cd, such that the test passes

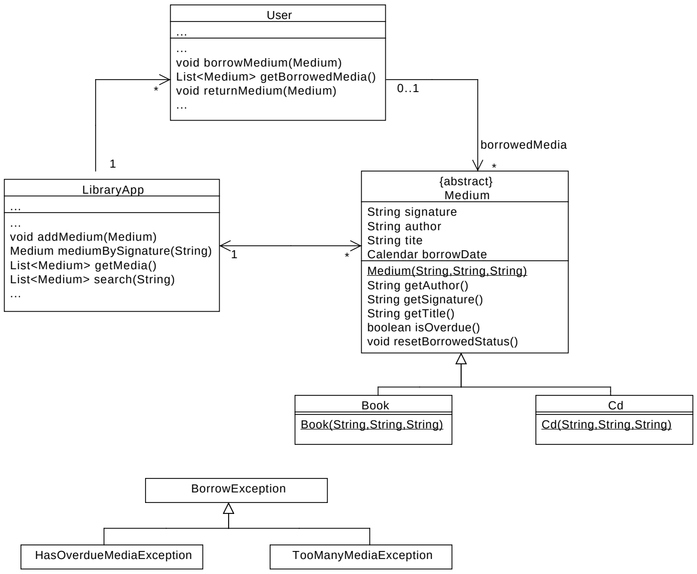
4.2 Refactor LibraryApp

After implementing the tests for adding CDs to the library, one finds out that the current interface for the library application mentions books and CDs explicitly, although they should work for books as well as CDs, i.e. in principle, they work for the more general concept of medium (plural media). Thus, the goal is to **refactor** the application to cope with general media instead of just books

A possible class diagram for starting the refactoring (depends on how you developed your application)



The class diagram after refactoring



Steps

1. Rename the operations and fields in LibraryApp and also any possible local variables and parameters from containg book/books to medium/media; Run all tests; fix any errors in the tests

   - Use the rename command of Eclipse (e.g. menu refactor::rename in the code view), which automatically also renames all uses of the method/field/variable

2. Rename Book to Medium; Run all tests

   - Use the rename command of Eclipse, which automatically also renames all uses of the class

3. Make Medium an abstract class

  - A medium should not be instantiated, i.e. only for books (and later CDs) it should be possible to create objects

4. Fix the compile time errors in the tests by replacing Medium with Book

   - This means, among others, to change the tests so that they create books again instead of just media (due to the renaming of Book to Medium)

   4a. Use Eclipse to create a new class Book as subclass of Medium

   4b. Use Eclipse to create the missing constructor for Book; Run all tests

5. Make Cd a subclass of Medium (if it is not already).

6. Redefine methods addCd and getCds to use addMedium and getMedia: This looks as follows:

```
public void addCd(Cd aCd) { addMedium(aCd); }
public List<Medium> getCds() { getMedia(); }
```

   Note the change of the return type of getCds from List¡Cd¿ to List¡Medium¿

   Don't forget to run the tests

   6a. Select addCd and getCds in turn and select Refactor::inline in the code view. This will then replace addCd and getCs with their bodies addMedium and getMedia. Run the tests.

4.3 Overdue CDs

- Add tests (and implementation) for overdue CDs. Note that a CD is already overdue after one week. Create tests similar to method testOverdueBook() in class TestOverdue.

- The test class should be named TestOverdueCd.

- Note: again a test for testing that it is not possible to borrow a CD if one has some media that is overdue should not be necessary, unless you did something special for CDs only in your implementation, so that the code you had previously has CD specific cases. If this is a case, you should add a test similar to testBorrowWithOverdueBooks in class TestOverdue

4.4 Fines for overdue media

- Add tests and implementation for recording and paying fines for overdue media. A fine occures, when a media is overdue (20 kroner for books and 40 kroner for CDs). Here it is assumed that the fine is paid when returning the book. Test the following scenarios

- Scenario a): Borrow a book and don't return it in time.

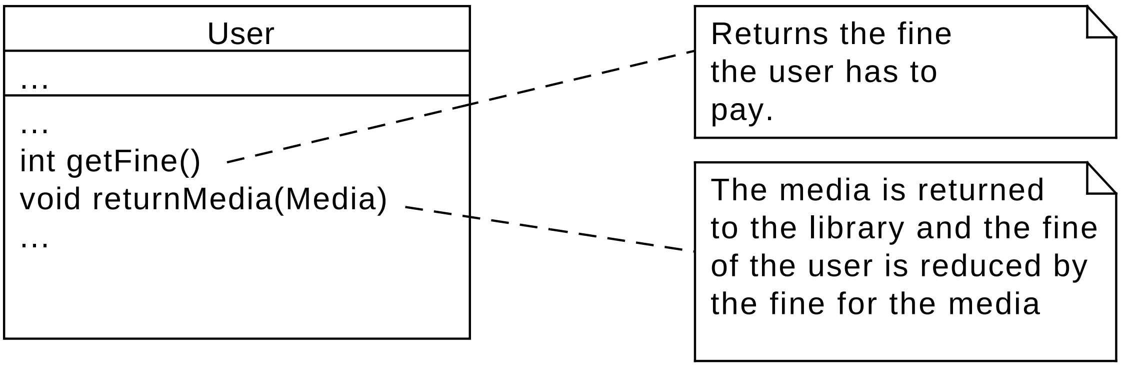
  - Borrow a book

  - Set the current time to 4 weeks plus 1 day
  - Check that the fine the user has to pay is 20 kroner
  - Return the book
  - Check that now the user has no fines anymore

- Scenario b): The user borrows two books and does not return them in time; check that the fine is 40 kroner
  - The user borrows two books and does not return them in time
  - Check that the fine is 40 kroner
  - Return one book
  - Check that the fine is now only 20 kroner
  - Return the other book
  - Check that there is no fine anymore

- Scenario d): Check that the fine for overdue CDs is 40 kroner
  - borrow a CD and do not return it in time
  - check that the fine the user has to pay is 40 kroner

- Note that the tests should use the following interface of user:


User
...
...
int getFine()
void returnMedia(Media)
...
Returns the fine the user has to pay.
The media is returned to the library and the fine of the user is reduced by the fine for the media