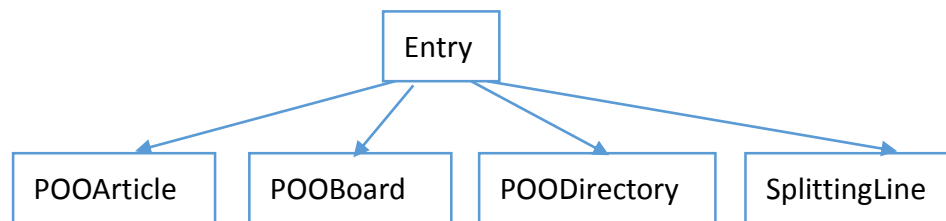


OOP HW3

B99902008 資工三 李朋臻

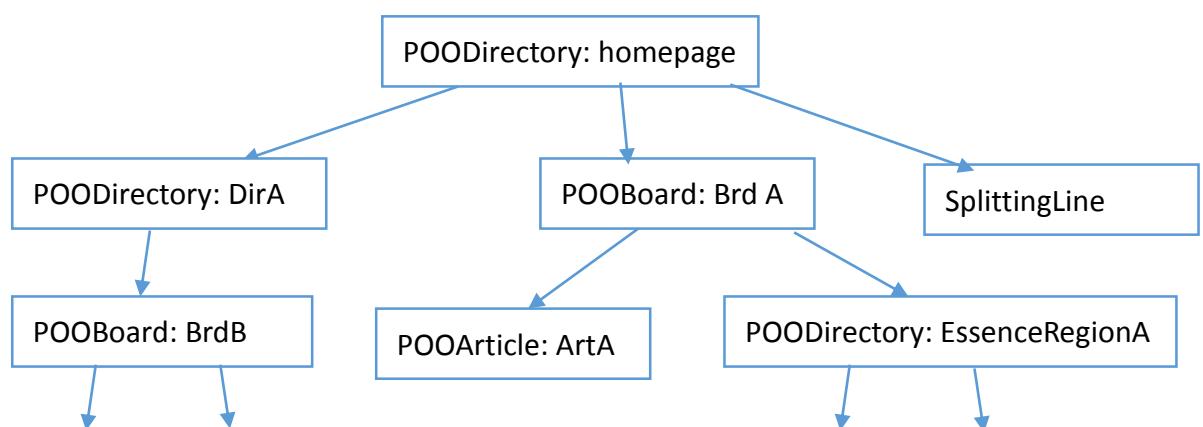
(1) The relations between classes

There are 7 classes in this work, Demo, Entry, POOArticle, POOBoard, POODirectory, SplittingLine, and 'User. Demo and User are not the key points of this design. Demo is a Unix-like command line class to demonstrate the whole work, and User' will hold some basic information. Relations between the rest five classes can be considered from two perspectives. We can observe them in the view of a programmer. Then it will be something like this:



The 'arrow' means that there is an inheriting relation. For example, POOArticle inherits Entry. That is, POOArticle is an Entry.

The other perspective is to observe them in view of a user. Then it will be something like this:



The 'arrow' means that there is a containing relation. That is, POODirectory can has a POODirectory, a POOBoard, and a SplittingLine but not a POOArticle. POOBoard can has a POOArticle and a special POODirectory serving as an essence region. POOArticle can not has anything. But where is Entry? Entry serves as a super type for Demo to traverse through this system.

(2) Advantages

It will be easy to understand that POOArticle, POOBoard, POODirectory and SplittingLine are Entry. Besides, considering the subtyping approach, since all

these four classes inherit Entry, Entry can serve as a super type. And this will make things easier for any other programmer who wants to traverse between these four classes. Also, it will be easy to maintain such programs. Because programmers only need to watch out for a list of Entry instead of lists of the four classes.

(3) Disadvantage

Because of using Entry to serve as a super type to hold all these four classes, it will somehow be a little bothering when we try access a specific member function that Entry doesn't have. We have to do either type-casting or keeping a temp reference which is not of the type Entry.

(4) Git

Because I use Eclipse to program, I only have to
"rm -r src", "cp -r [my working directory/src] src"
"git add src", "git commit -m 'blabla' "

So, in fact, I don't need to modify the file .gitignore

But I append "*.class" to .gitignore

(5) Bonus

In addition to the methods required by the spec, I implement several different features for POOBoard and POOArticle. While being on the page that is a POOBoard,

Focus: focus [pos]

We can mark an article on the indicated position of the board to be focused. Then, if the list methods of such article will add 'm' before the evaluation count.

Essence Region: Z

With the Demo program, you can press 'Z' to enter the essence region of the board. Such region is automatically generated while calling its constructor, and each board can only have exactly one essence region.

Add Essence: essence [source pos] [destination pos]

We can add an article on the board to the board of its essence region and mark the article focused through the positions you type. And if you delete the article of the board, there is still a copy in its essence region. This feature is implemented through clone(), which is a method of POOArticle that overrides the one of Object.

Transcript: transcript [source pos] [destination path]

We can transcript an article on the indicated position of the board to another board through the given path. Similar to Add_Essence, deleting the original article will not delete the other.

Besides, I add several convenient Unix-like commands on the Demo program to make it easier to use. The most important one is 'cd'. This command is almost the same as the one on linux. For example, you can type "cd ../../ntu/csie/notice" and you can also cd into an article to read it. And the command always allow immediate argument or delayed argument. For example, "push good" means to push an article with short comment "good." You can also type "push" and wait for hint message to tell you what argument(s) you should type next. Also, there is a 'help page'. Type "h" or "help" or any wrong command will show that page.

These are some screenshots for 'add essence':

After login as username = ta, and cd ./Favorite/OOP/HW3:

```
{HomePage}/{Favorite}/{OOP}/<HW3>/#> push excellent
```

ta pushes this work

```
*****
Author  : 008
Title   : HW3
Info    : ID 000, Evaluation 2
-----
```

This is a good homework

```
Push   : good
Push   : excellent
```

Push message is shown here, the above is written by another user

```
{HomePage}/{Favorite}/{OOP}/<HW3>/#> return
```

You can type "return" or "cd .."

```
*****
Board OOP, 2 items
-----
Pos   Eval  ID   Title  Author
000   2     000  HW3    008
001   0     001  HW?    nobody
*****
```

```
{HomePage}/{Favorite}/{OOP}/#> essence 0 0
```

Add article at pos 0 to the board at pos 0 of the essence region

Then ta boo HW? and type “z” to check the essence region

```
*****
Board OOP, 2 items
-----
Pos   Eval   ID    Title  Author
000   m2     000   HW3    008
001   -1     001   HW?    nobody
*****

{HomePage}/{Favorite}/{OOP}/#> z
*****
Directory Essence, 1 items
-----
000 Board      Awesome
*****

{HomePage}/{Favorite}/{OOP}/{Essence}/#> cd Awesome
*****
Board Awesome, 1 items
-----
Pos   Eval   ID    Title  Author
000   2      000   HW3    008
*****

{HomePage}/{Favorite}/{OOP}/{Essence}/{Awesome}/#> cd HW3
*****
Author : 008
Title  : HW3
Info   : ID 000, Evaluation 2
-----
```

HW3 is now focused

Get into the essence region

The board at pos 0 is called Awesome

Article in essence region is not focused as default

You can type “cd HW3” or “read 0”

Ps: {Favorite} means that it's a directory named Favorite.

[OOP] means that it's a board named OOP.

<HW3> means that it's an article named HW3.