

Lab sheet 2: The World of Zuul

Setter: Verena Rieser

Please use the code in **lab3/zuulRefactored** for the following exercises.

Back Command and JUnit test

Add a *back* Command

- This command does not have a second word. Entering the back command takes the player into the previous room he/she was in.
- You should account for the cases where the user accidentally entered a second word and also if there is no previous room.

(1M)

Test your new command using a pre-defined JUnit test.

- Uncomment the last two lines in **TestBack.java** and try to run it.
- For now you have change the visibility of the **goBack()** method to **protected**. (NB: Later on you'll learn a way to work around this.)
- Describe how the JUnit code works by adding comments above each line of code.

(1M)

Refactoring Code

Add a Player Class

- Refactor your project to introduce a **Player** class. A Player object should store (at least) the current room of the player and the player's name.
- Add the Player to the Game:
You need to make sure that the Player is initialized correctly and set into the first Room.
Modify the **printWelcome()** method that to inform the Player about his current room.
Modify the **goRoom()** method. Make sure you'll notify the player about his moves.

(2M)

References:

- Lars Vogel. JUnit - Tutorial. Version 2.3, 2012.
- <http://www.vogella.com/articles/JUnit/article.html>
- <https://github.com/kentbeck/junit/wiki>