Eric Wallace

Exercise8.1- The application is a game that takes commands from a user to go in either east, west, or south. Based on the selection the return is a message describing the new location with a display of which directional moves are available. The other commands it accepts are help which, returns a message about being lost in a university with the available command and the last command it quit which ends the program. There are three rooms connected with a fourth room inside the computer lab.

Exercise8.2-

Game- provides the user interface by starting and stopping, taking user inputs and guiding the user.

Parser- parses the user input to extract keywords to recognize valid commands.

CommandWords- validates the commands entered.

Room- defines the rooms that are in the game.

Exercise8.3-

Bounty Hunter. The user raids a house to collect a fugitive for a bounty.

Exercise8.4- Edited the title to bob-the-bounty-hunter and changed the rooms.

Exercise8.5- Did it, works.

Exercise8.6- Added the accessor method for descriptions and replaced the old code.

Exercise8.7- Did it.

Exercise8.8- Did it.

Exercise8.9- The keySet method of HashMap returns a Set containing all keys of the HashMap

Exercise8.10- the getExitString method checks for exits the user can input, then returns a string listing those available.

Exercise8.11- Did it.

Exercise8.13- It moves on to the Room we moved into and shows the description and exits from the new room.

Exercise8.14- Did it.

Exercise8.15- Did it.

Exercise8.16- Did it, works.

Exercise8.17- Yes, the processCommand method has to be updated with added commands.

Exercise8.18- Did it.

Exercise8.19- It is a design patter that programmers use to divide a program up into three units. The model is the central part of the program where the data is stored and organized, the view is the part that interfaces with users and gets input from them. The controller then receives a command from the view and manipulates the model and returns it back to the user. The model keeps the program simple to minimize the coupling which is what we were implementing in the project.

Exercise8.20- Did it.

Exercise8.21- It should be set when constructing the room. The Item class. If a room doesn't have an item it will just print a blank string.

Exercise8.22- Did it.

Exercise8.23- Did it.

Exercise8.25- After typing back twice in a row the user goes back to the previous room and then back to the room the command was typed in and if you continue it will go back and forth between them. Made sense to me.

Exercise8.26- Did it.

Exercise8.27- Maybe test the commands, move through all rooms and test moving backwards.

Exercise8.28- Did it.

Exercise8.29- Did it.

Exercise8.30- Did it.

Exercise8.31- Did it.

Exercise8.32- Did it.

Exercise8.33- Did it.

Exercise8.34- It works.

Exercise8.35- Did it.

Exercise8.36- Did it. Only need to change the CommandWords class

Exercise8.37- It still says to type "help"

Exercise8.38-

public enum Direction

{

NORTH, EAST, SOUTH, WEST

}

Exercise8.39- Did it. No, had to update the look method in Game class, and put it in the processCommand method as well.

Exercise8.40- It automatically calls the new word for "Help"

Exercise8.41- Added a 5 turn limit to reach the computer office.

Exercise8.42- Did it.

Exercise8.43- Did it.

Exercise8.44- Did it.

Exercise8.45- Did it.

Exercise8.46- Did it.

Exercise8.47- Did it.

Exercise8.48- Did it.

Exercise8.49- Added it. I have errors to fix.

Exercise9.1- It behaved as expected, the ArrayList grew as I added comments.

Exercise9.2- The showInfo printed correctly.

Exercise9.3- It works correctly.

Exercise9.4- It rejects attempt to put a repeat comment from an author.

Exercise9.5- It works correctly.

Exercise9.6- It accepts 0 as a rating but prevents 6.

Exercise9.7- Balance is correctly calculated.

Exercise9.8- the findMostHelpfulComment method worked.

Exercise9.9- It throws exception for an empty list.

Exercise9.10- It's fixed.

Exercise9.11-

-Positive tests: valid comments, showing info about comments, boundary checking valid ratings, voting comments, finding the most helpful comment in a non-empty comment list

-Negative tests: adding multiple comments by same author, checking invalid rating boundaries, trying to find the most helpful comment in an empty list.

Exercise9.12- Ran the test and saw the correct dialogue box.

Exercise9.13- Did it.

Exercise9.14- CommentTest, setUp and tearDown

Exercise9.15- Did it.

Exercise9.16- Did it.

Exercise9.17- Clicking show source takes you to the line causing the error.

Exercise9.18- Did it.

Exercise9.19- Did it.

Exercise9.20- Ran some more tests.

Exercise9.21- I made the changes but hit an error.

Exercise9.22- All tests passed with result of 7 and 5.

Exercise9.23- The addition results varied when it should always have the same answer of 7.

Exercise9.24- It produced -13 as a result.

Exercise9.25- I walked through the steps in the debugger.

Exercise9.26-

numberPressed(4): 4 ; 3 ; '+'

equals(); 7; 0; '+'

getDisplayValue: 7; 0; '+'

Exercise9.27- Yes

Exercise9.28- It will impact the previousOperator field.

Exercise9.29- Fixed the clear method to reset the previousOperator.

Exercise9.30- displayValue has the correct value. The result was 6.

Exercise9.32- Yes, it helps.

Exercise9.33- I think it is about right.

Exercise9.34- It gives detailed insight to the processes being executed through each method. Makes it easier to debug. Using print statements will help identify where problems exist.

Exercise9.35- Did it.

Exercise9.37- Did it.

Exercise9.38- Finding the logical errors.