HACETTEPE UNIVERSITY

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

Name and Surname:	Mustafa Aşkın Pınar
Identity Number:	20724964
Course:	BBM104 Introduction to Programming Laboratory II
Experiment:	Assignment 3
Subject:	Polymorphism, Exceptions
Due Date:	07.05.2021
Advisor:	Bahar Gezici
E-mail:	b20724964@cs.hacettepe.edu.tr
Main Program:	Main.java

PROBLEM

There is a game called Zorde and Calliance where two sides fight each other. Each side has 3 different types of characters. The game is played on a board (just like chess). During the progress of the game, each side makes moves in turns. The aim of the game is to kill every opponent character.

SOLUTION

"Commands" class read the commands.txt and initials.txt files. It has read commands as a 2d string array (called *commands*). This class contains some methods. These methods were created to execute commands given in commands.txt and initials.txt.

The operation was done by calling the Commands class in the "Main" class. Traveled over *commands* with for loop. Then, the data on the *commands* was compared with the methods and the operation was made.

Data Structure

I created twelve different classes. Which are;

- Main.java
- Constants.java
- Reader.java → Interface

  Has one method;
  - FileReader()
- Commands.java → implements Reader.java

  Has ten methods;
  - findZorde()
  - findCalliance()
  - closeOutputFile()
  - createOutputFile()
  - createBoard()
  - setBoard()
  - getBoard()
  - checkInitial()
  - createChars()
  - printBoard()
- Calliance.java
- Dwarf.java → extends Calliance.java
- Elf.java → extends Calliance.java
- Human.java → extends Calliance.java
- Zorde.java
- Ork.java → extends Zorde.java
- Troll.java → extends Zorde.java
- Goblin.java → extends Zorde.java

Explanation of my code

1-(commands.java)

- findZorde() → Returns the character at the given point coordinate.
- findCalliance() → Returns the character at the given point coordinate.
- createOutputFile() → Creates an output file with the given name.
- closeOutputFilr() → Close the created output file.
- createBoard → Creates the board according to the given in initials.txt.
- setBoard() → Updates the playing field.
- getBoard() → Returns the board.
- checkInitial() → Control the initials.txt
- createChars() → Create characters in initials.txt according to their group..
- printBoard() → Print board and some information in output file

2-(any char *.java)

- move() → Regulates and controls the movements of the character. Updates the situations caused by the character's movement. This method is arranged separately for each.

3-(Zorde.java and Calliance.java)

- getHP() → Return character’s HP.
- getFullHp() → Return character’s first HP.
- setHP() → Update character’s HP.
- getAP() → Return character’s AP.
- getMaxMove() → Return character’s max move.
- getName() → Return character’s name.
- setName() → Set character’s name .
- getColumn() → Return character’s x axes .
- getRow() → Return character’s y axes.
- setXY() → Update character’s coordinate.
- setXYFirst() → Set character’s first coordinate.

Advantages

In general, the fact that the object classes are in one place and the method classes are in one place increases the readability of the cod. The variable and method names I use do not require much explanation. I tried not to import the library as much as possible. All operations were made using simple methods. You can understand what operations have been done just by looking at the Main class. As polymorphism; I called my org, troll, and goblin operations using the force class. This is the case for Calliance characters.

Disadvantages

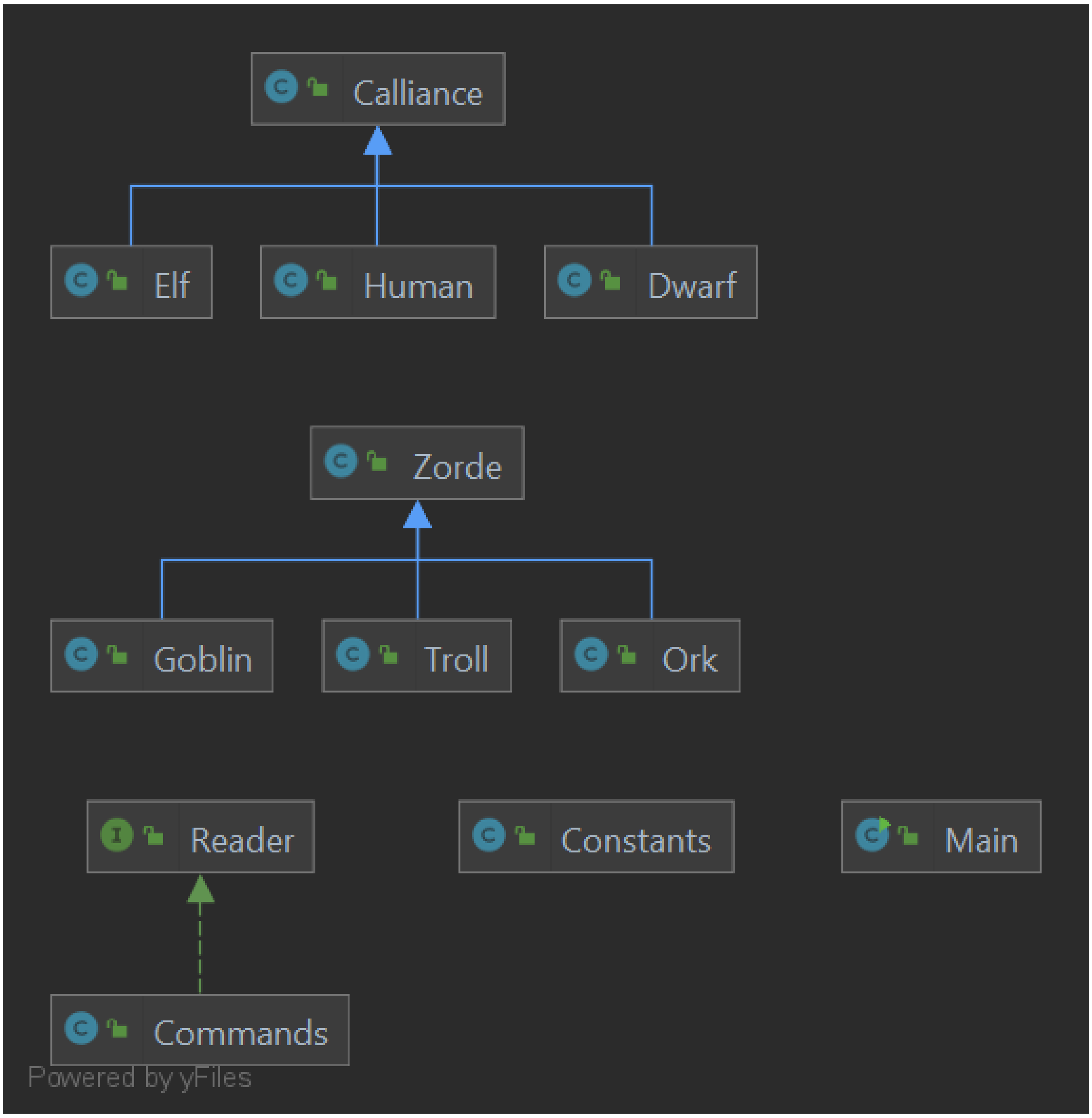
If I had made a class above Zorde and Calliance as a structure, my transactions would be easier. I did not want some methods of the Zorde and Calliance classes to be accessed directly. For this reason, I did not write some set methods. However, that required me to override most of the methods in Zorde and Calliance. I did not create a separate class for exceptions.

References

1. https://www.stacoverflow.com
2. https://www.geeksforgeeks.org

Diagrams

Class Diagrams



Note: I added the other diagrams to the file as jpg. because if I added it here the resolution would be lower.