HACETTEPE UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

BBM104

Name-Surname : Mustafa Çağrı KORKMAZ

Student ID : 2200356833

E-Mail : b2200356833@cs.hacettepe.edu.tr

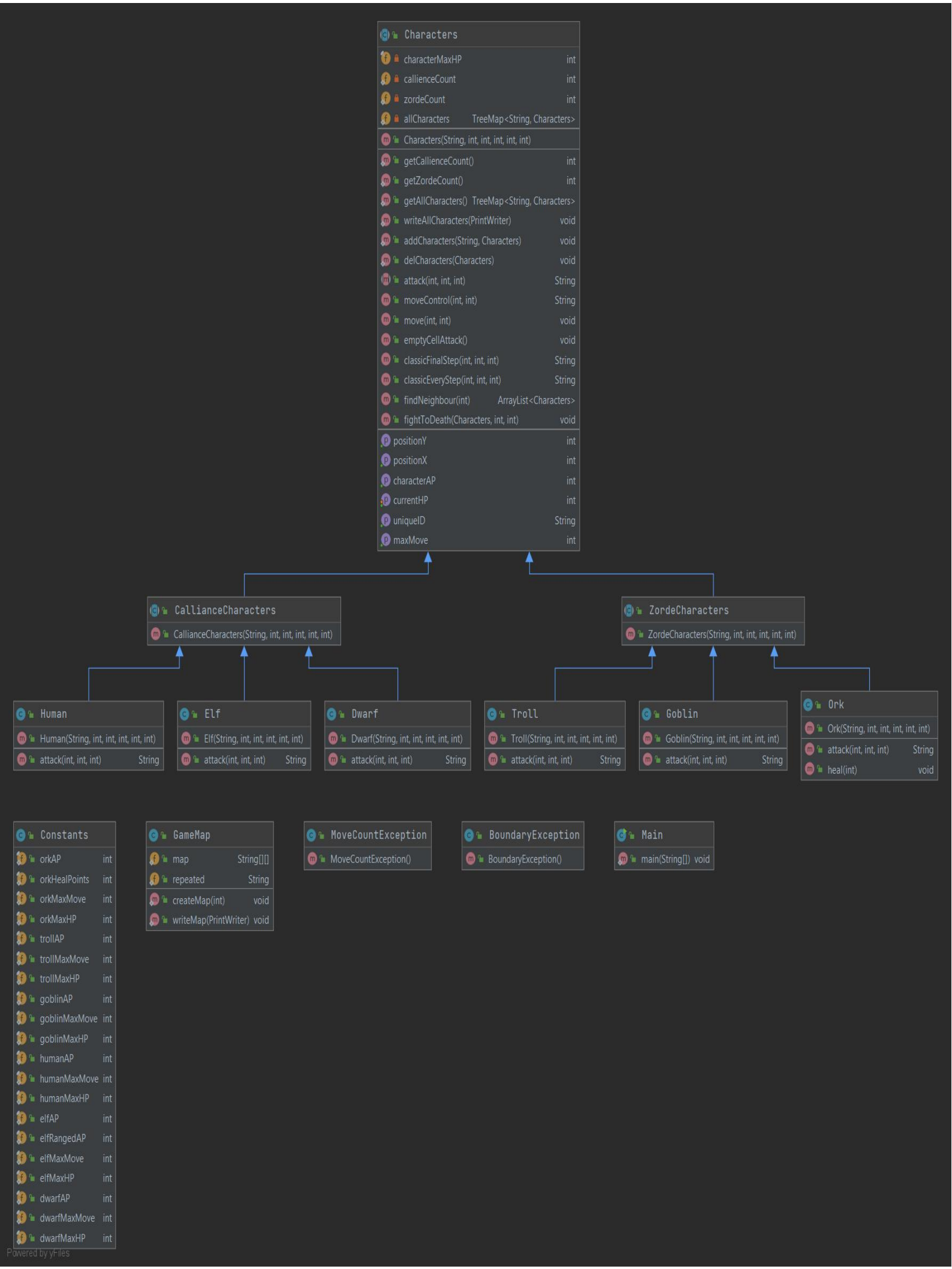
Subject : Programming Assignment 3

PROBLEM:

There are two teams that are enemies to each other as Zorde and Calliance, these teams have their own special characters. Dwarf, Human, and Elf are characters of Callience while Ork, Goblin, and Troll are characters of Zorde. All characters have their special and constant properties such as maxMove, attack type, etc. According to their constant properties, we attack the enemy characters, change their current location, and other inconstant properties such as current HP. Also if there is a custom error such as MoveStepException and BoundryException while the character is moving on the cells of the 2D game board according to the commands, we should print this error on the output file. End of the commands, the team that has no character on the board loses the game.

SOLUTION:

I have tried to write a dynamic code as I can. Firstly, I have created Characters,CallianceCharacters(extends Characters) and ZordeCharacters(extends Characters) abstract classes.Then, I have created the classes of all types of characters as a subclass of their team classes. Moreover, I have created a static allCharacters TreeMap that can contain all types of character classes in the Characters abstract class. When I was reading the initials.txt I have created and added every character to the allCharacters map. All constants properties have been received from Constants.java file and attained to the related fields of the characters. Secondly, I have created attack abstract method in the Characters class. Also, I have created classicFinalStep, classicEveryStep, emptyCellAttack methods in Characters class for characters that have not a special skill on their moving or attacking. I have overridden attack method in the every character classes in according to the their move and attack type.Also, if character has a special skill, I have coded this skills in the character own class.Therefore, I have used polymorphism in here. Lastly, when I was reading commands and writing output files , I have received all character objects from allCharacter TreeMap that contains Characters objects and these objects have moved and attacked according to the their overridden attack method. My code is as flexible as I can, adding new characters, new methods and changing constant properties are very easy.Therefore, making an update is not a big problem. In this assignment I have learned how can I write dynamic and flexible code, and using polymorphism,custom exceptions.



Characters:

All common methods are in this abstract class. All characters that added to the game are kept in static allCharacters TreeMap of this class.

CallianceCharacters:

This class is super abstract class of the Calliance characters and extends form Characters. If you want to add more fields or methods for only Calliance characters you can add them to this abstract class in the future.

ZordeCharacters:

This class is super abstract class of the Zorde characters and extends form Characters. If you want to add more fields or methods for only Zorde characters you can add them to this abstract class in the future.

Human:

This class extends CallianceCharacters abstract class and holds methods for Human.

Elf:

This class extends CallianceCharacters abstract class and holds methods for Elf.

Dwarf:

This class extends CallianceCharacters abstract class and holds methods for Dwarf.

Ork:

This class extends ZordeCharacters abstract class and holds methods for Ork.

Goblin:

This class extends ZordeCharacters abstract class and holds methods for Goblin.

Troll:

This class extends ZordeCharacters abstract class and holds methods for Troll.

Constants:

This class holds constant properties of the characters as public value.

GameMap:

This class has two public static method for creating and writing game map.

MoveCountException:

This exception is given when move count on the command line is not same with the maxMove of the character.

BoundaryException:

This exception is given when the character tries to go out of the game map.

Main:

All file reading and writing operations are processed in this class.

REFERENCES

https://www.geeksforgeeks.org/

https://stackoverflow.com/

https://www.udemy.com/course/java-the-complete-java-developer-course/

Lecture Notes Of BBM102