HACETTEPE UNIVERSITY COMPUTER SCIENCE

REPORT AND EXPLANATION OF BBM104 THIRD ASSIGNMENT

MELİH MERAL-21946438

Problem : In this experiment we have a lot of characters with same attributes. This characters have diffrent hp ,ap etc.We have some special attribute.We have some same methods for same parts.

Solution: I use inheritance and interface for this special attributes and same attributes. Important part is some charecters have special attributes and characters seperate in two teams so we need to seperate two oppenent and for special attributes we use diffrent methods for each other.For the same parts i use functions for clean and understandable code.I use loops for reading input files and writing output file.

Fighter class is super class of dwarf ,ork,troll,elf,human and goblin.It controls characters attributes.

Dwarf class control dwarfs attributes and return dwarfs constants.

Elf class control elfs attributes and control elfs special attributes of ranged attacks.

Ork class control orks attributes and orks special attributes of healing.

Human class control humans attributes and return humans constants

Goblin class control goblin attributes and return goblins constants.

Troll class control dwarfs attributes and return trolls constants.

Constants class is the owner of fighters constants and their attributes.

Main

~ count2 : int
~ count : int

~ printBoard(line : int, board : List, output : PrintWriter) : void
~ returner() : int
~ cleaner() : int
~ returner2() : int
~ cleaner2() : int
~ addBoard(lister : List<String>, int_line : int, board : List<String>) : void
~ one_moveboard(movement : List<String>, boardd : List<String>, int_line : int, output : PrintWriter, fighter_man : String, ally : List<String>, count : int, human_attack : List<String>, board : List<String>, movement : List<String>, dost : List<String>, enemy : List<String>, ally_list : List<String>, xx : int, D0 : Dwarf, O0 : Ork, E0 : Elf, H0 : Human, T0 : Troll, G0 : Goblin, D1 : Dwarf, O1 : Ork, E1 : Elf, H1 : Human, T1 : Troll, G1 : Goblin) : void
~ figtherHp(human_attackk : List<String>, moral : int, D0 : Dwarf, O0 : Ork, E0 : Elf, H0 : Human, T0 : Troll, G0 : Goblin, D1 : Dwarf, O1 : Ork, E1 : Elf, H1 : Human, T1 : Troll, G1 : Goblin, boardd : List<String>) : void
~ figtte_death(D0 : Dwarf, O0 : Ork, E0 : Elf, H0 : Human, T0 : Troll, G0 : Goblin, D1 : Dwarf, O1 : Ork, E1 : Elf, H1 : Human, T1 : Troll, G1 : Goblin, attackerHp : int, defenserHp : int, attackrHp : int, board : List<String>, index : int, index2 : int, fighter_man : String, defender : String, attacker : String) : void
~ figterHpprinter(fighter : List<String>, output : PrintWriter, D0 : Dwarf, O0 : Ork, E0 : Elf, H0 : Human, T0 : Troll, G0 : Goblin, D1 : Dwarf, O1 : Ork, E1 : Elf, H1 : Human, T1 : Troll, G1 : Goblin) : void
~ attack_list_edit(human_attack : List<String>, board : List<String>, movement : List<String>, int_line : int, ally : List<String>, enemy : List<String>, ally_list : List<String>) : void
+ main(args : String[]) : void

Dwarf

+ getfight : int
+ dwarfHp : int

+ lastHp() : void
+ Hp() : int
+ Ap() : int
+ maxMove() : int
+ setDwarfHp(dwarfHp : int) : void

Constants

+ dwarfMaxMove : int {readOnly}
+ dwarfAP : int {readOnly}
+ elfMaxMove : int {readOnly}
+ elfRangedAP : int {readOnly}
+ elfAP : int {readOnly}
+ humanMaxMove : int {readOnly}
+ humanAP : int {readOnly}
+ goblinMaxMove : int {readOnly}
+ goblinAP : int {readOnly}
+ trollMaxMove : int {readOnly}
+ trollAP : int {readOnly}
+ orkMaxMove : int {readOnly}
+ orkHealPoints : int {readOnly}
+ orkAP : int {readOnly}

<<interface>> Fighter

maxMove() : int
Ap() : int
Hp() : int

Ork

+ getfight : int
+ orkHp : int

+ lastHp() : void
+ Hp() : int
+ Ap() : int
+ maxMove() : int
+ Healpoints() : int

Elf

+ getfight : int
+ elfHp : int

+ lastHp() : void
+ Hp() : int
+ Ap() : int
+ maxMove() : int
+ rangedAp() : int

Goblin

+ getfight : int
+ goblinHp : int

+ lastHp() : void
+ Hp() : int
+ Ap() : int
+ maxMove() : int

Human

+ getfight : int
+ humanHp : int

+ lastHp() : void
+ Hp() : int
+ Ap() : int
+ maxMove() : int

Troll

+ getfight : int
+ trollHp : int

+ lastHp() : void
+ Hp() : int
+ Ap() : int
+ maxMove() : int

int