MonsterFactory

+ createCharacter(int):Monster
+ setAttack(int): IAttack

HeroFactory

+createCharacter(int): Hero
+SetAbility(int): IAbility

<<Abstract>> DungeonCharacter

name: String
hitPoints: int
attackSpeed: int
chanceToHit: double
damageMin: int
damageMax: int
numTurns: int

+ addHitPoints(int)
+ subtractHitPoints(int)
+ isAlive(): boolean
+ attack(DungeonCharacter)

Monster

chanceToHeal: double
minHeal: int
maxHeal: int
attack: IAttack

+ heal()
+ attack(DungeonCharacter)
+ subtractHitPoints(int)
+ setAttack(Attack)

Hero

chanceToBlock: double
numTurns: int
ability: IAbility

+ readName()
+ defend(): boolean
+ attack(DungeonCharacter)
+ performAbility(DungeonCharacter)
+subtractHitPoints(int)
+ battleChoices(DungeonCharacter)
+ setAbility(IAbility)

SorceressAbility

+ performAbility(DungeonCharacter x2)
+ attack(DungeonCharacter x2)
+ prompt():int

WarriorAbility

+ performAbility(DungeonCharacter x2)
+ attack(DungeonCharacter x2)
+ prompt():int

ThiefAbility

+ performAbility(DungeonCharacter x2)
+ attack(DungeonCharacter x2)
+ prompt():int

DwarfAbility

+ performAbility(DungeonCharacter x2)
+ attack(DungeonCharacter x2)
+ prompt():int

ClericAbility

+ performAbility(DungeonCharacter x2)
+ attack(DungeonCharacter x2)
+ prompt():int

GremlinAttack

+ attack(DungeonCharacter x2)

OgreAttack

+ attack(DungeonCharacter x2)

SkeletonAttack

+ attack(DungeonCharacter x2)

DarkLordAttack

+ attack(DungeonCharacter x2)

OrcAttack

+ attack(DungeonCharacter x2)

<<Interface>> IAttack

+ attack(DungeonCharacter x2)

<<Interface>> IAbility

+ performAbility(DungeonCharacter x2)
+ attack(DungeonCharacter x2)
+ prompt():int

This is dashed

MonsterAttackFactory

- HashMap<String, IAttack>: attacks

+ getAttack(String): IAttack

HeroAttackFactory

- HashMap<String, IAbilityk>: abilities

+ getAbility(String): IAbilities

Dungeon

+ main()
+chooseHeros(heroFactory, heroAttackFactory)
+generateMonsters(monsterFactory, monsterAttackFactory)
+battle(Hero[], Monster[])

Main contains all the elements needed to run