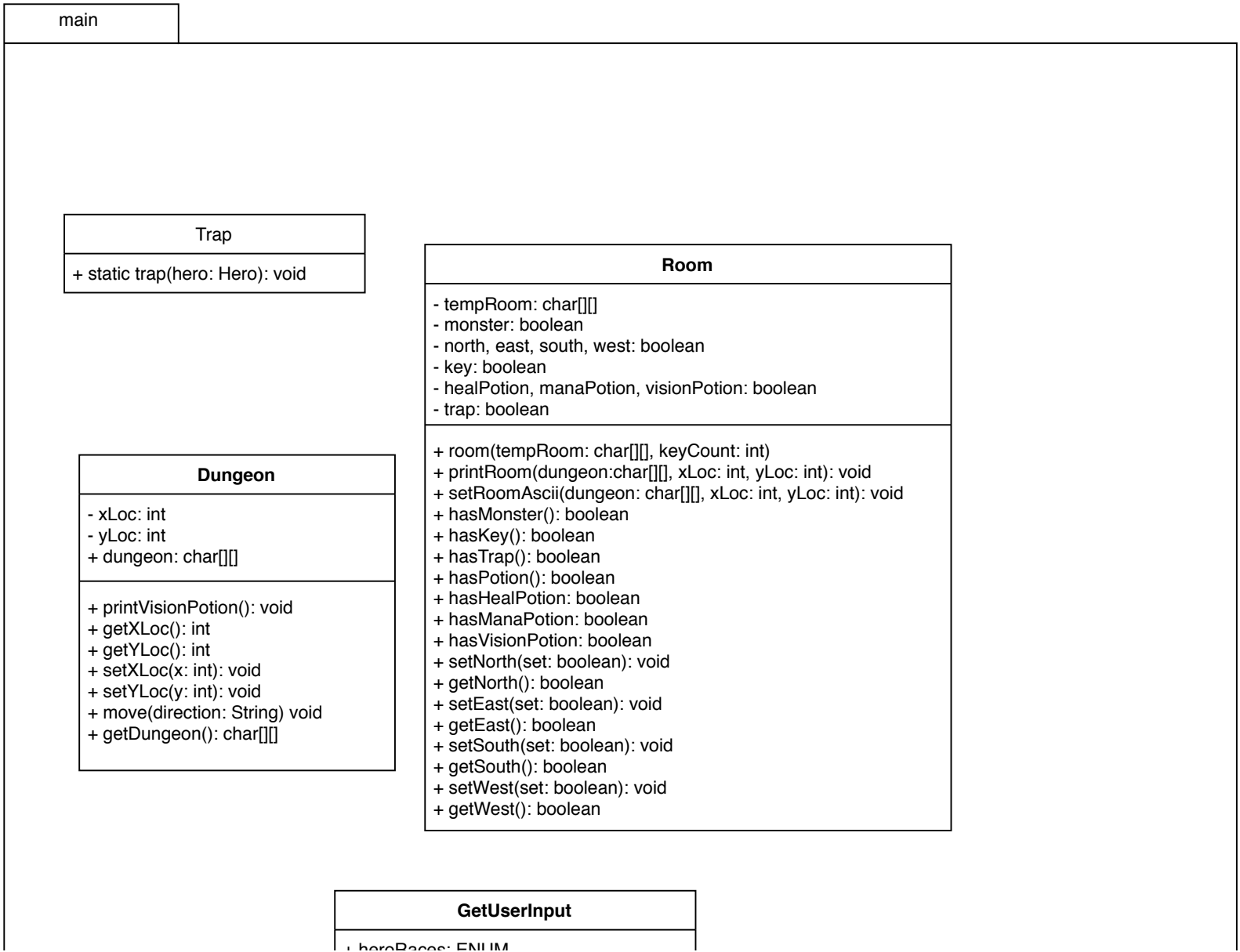
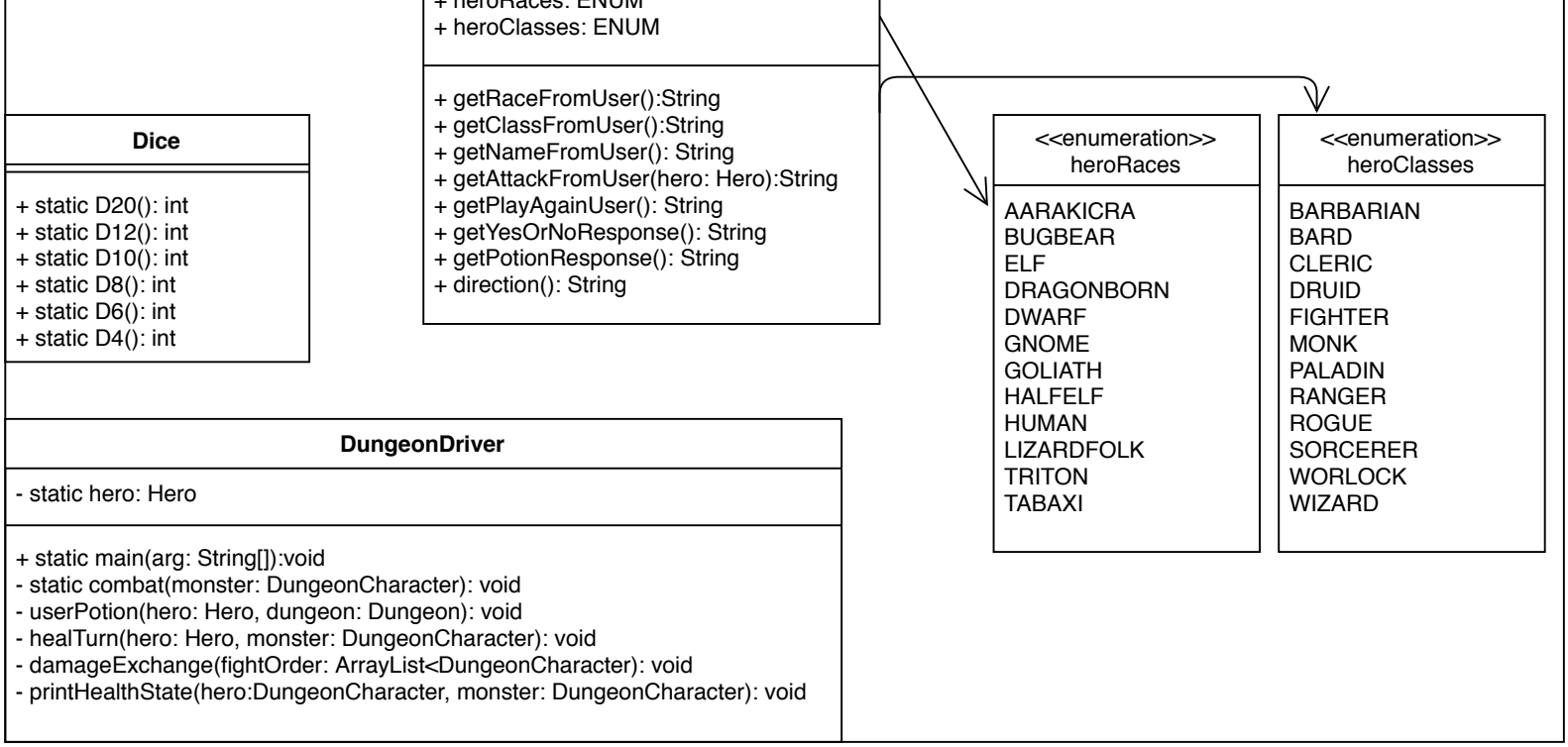
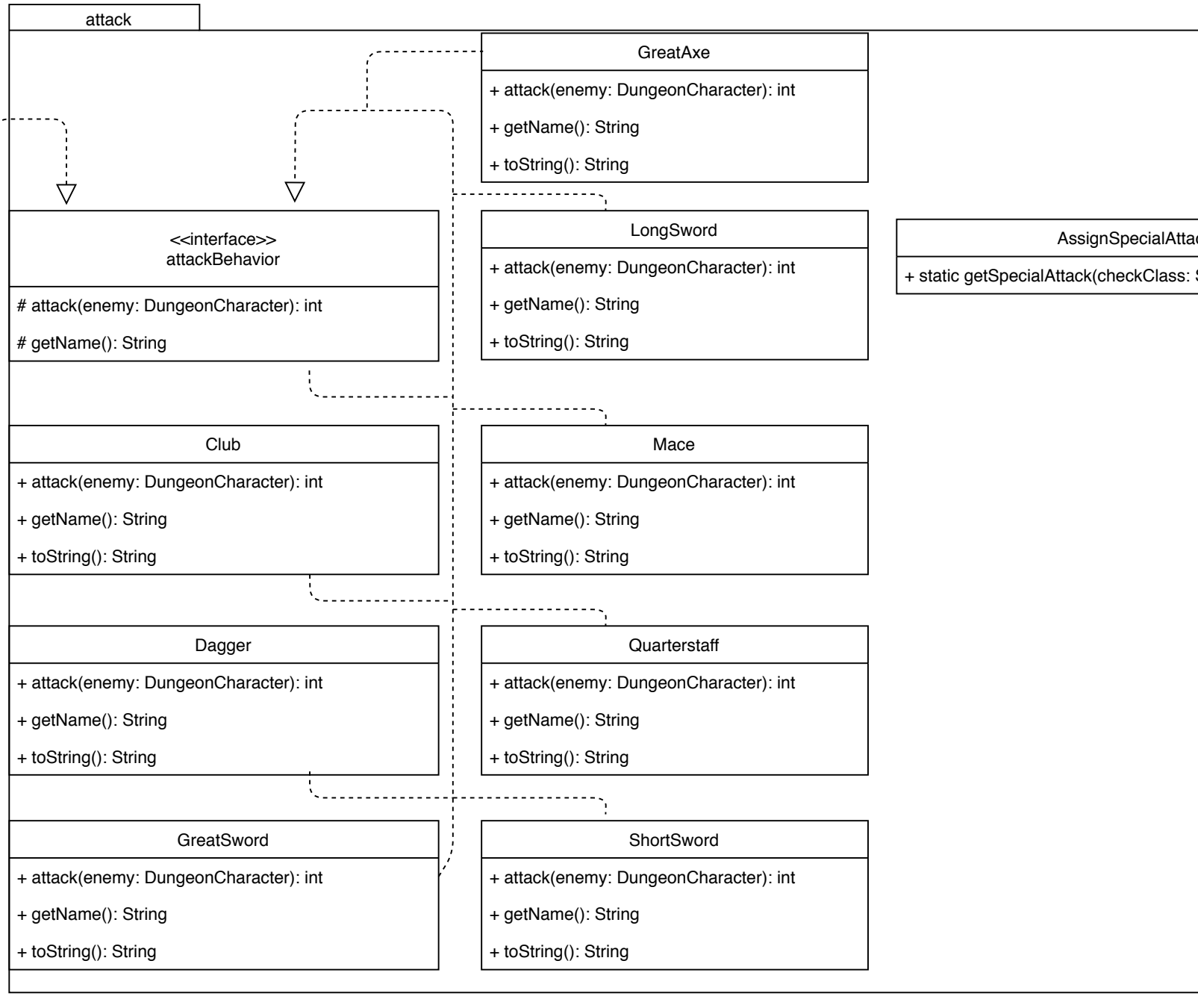
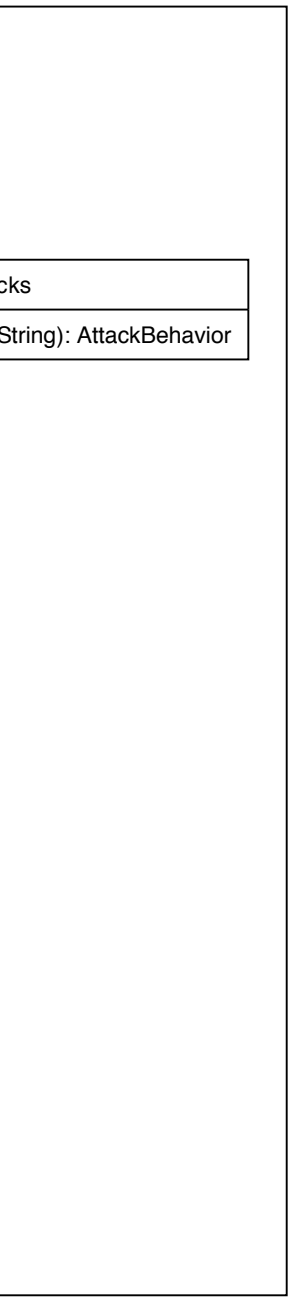
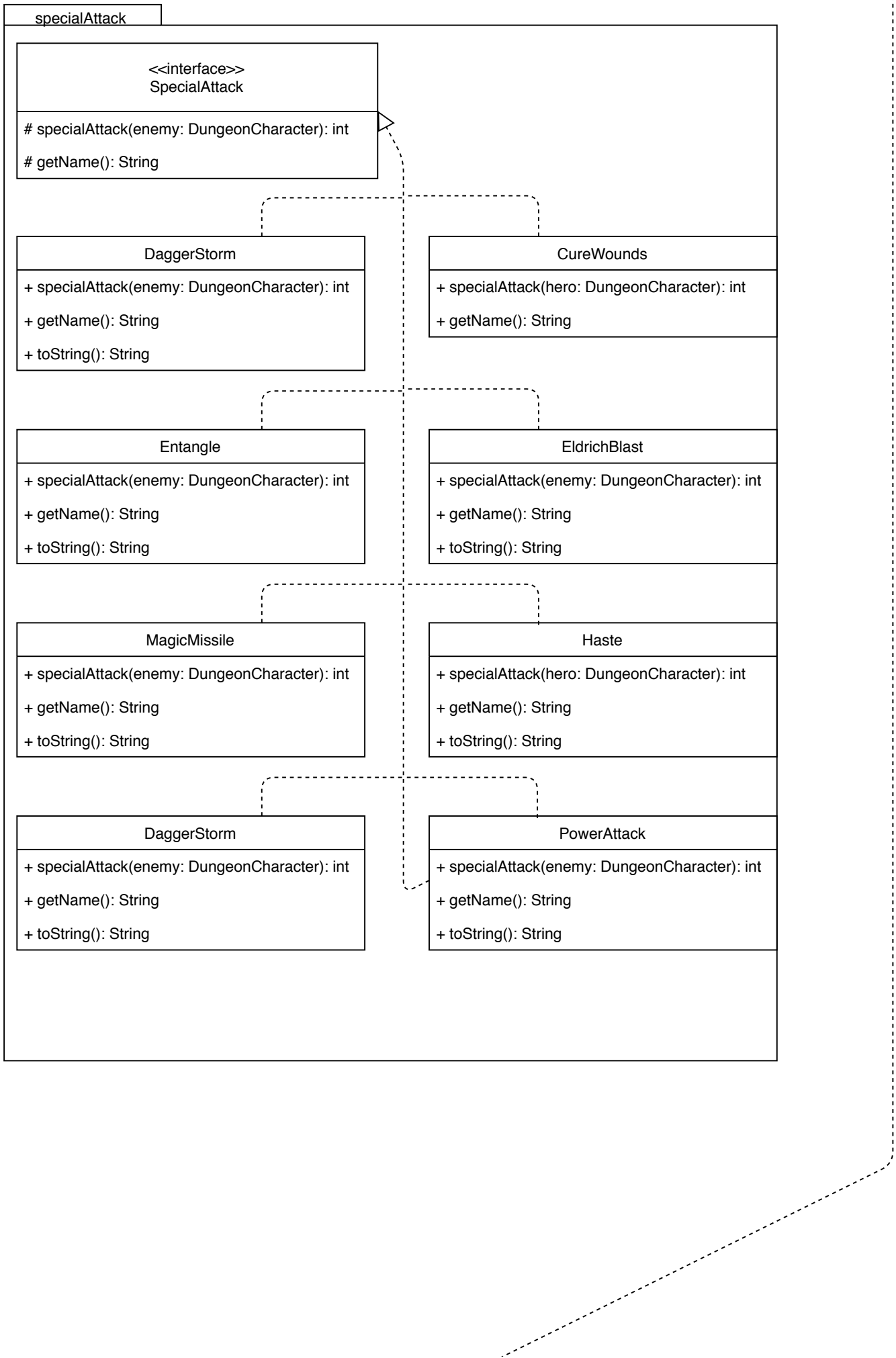
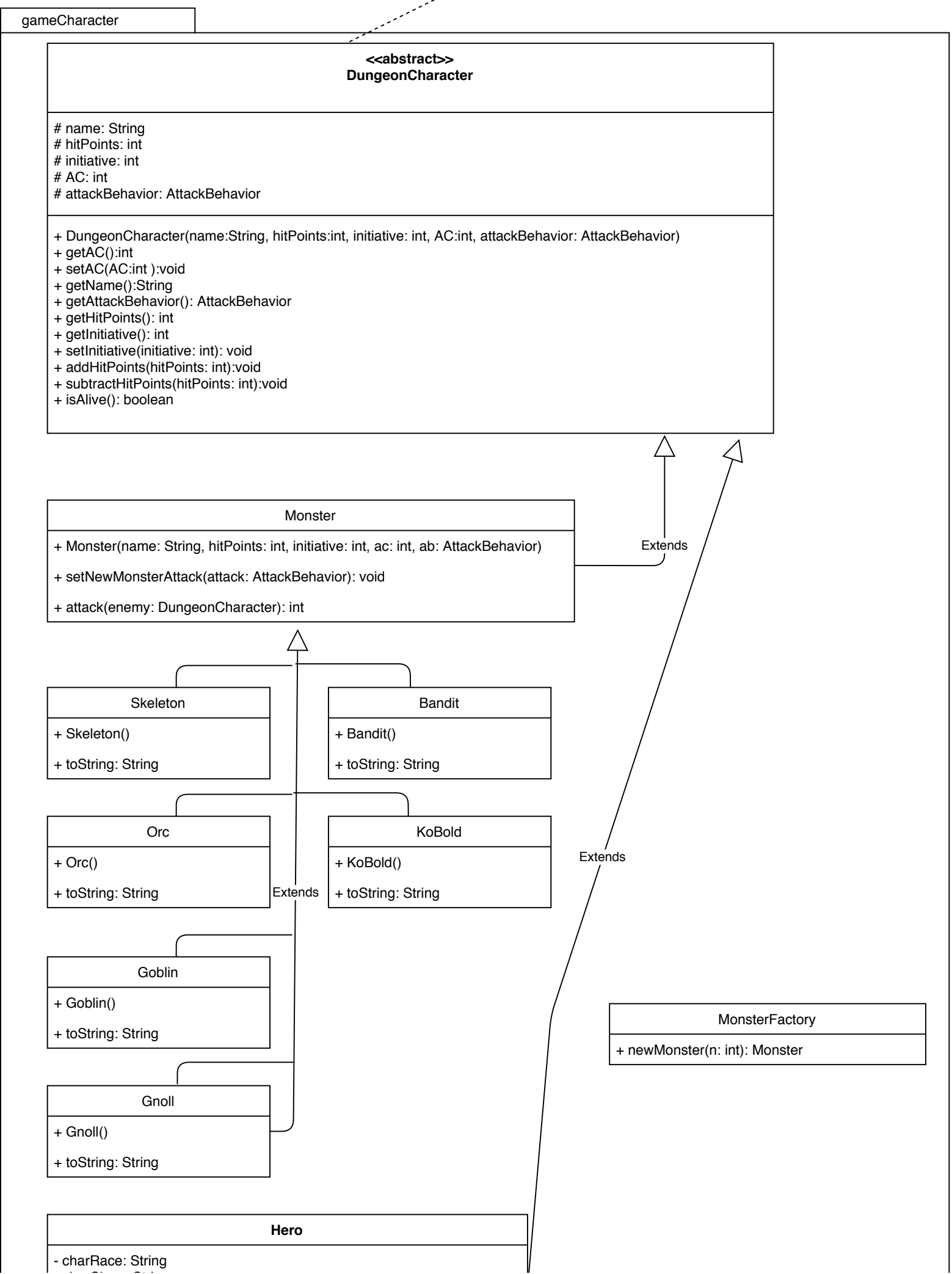

main
Trap
+ static trap(hero: Hero): void
Room
- tempRoom: char[][]
- monster: boolean
- north, east, south, west: boolean
- key: boolean
- healPotion, manaPotion, visionPotion: boolean
- trap: boolean
+ room(tempRoom: char[][], keyCount: int)
+ printRoom(dungeon:char[][], xLoc: int, yLoc: int): void
+ setRoomAscii(dungeon: char[][], xLoc: int, yLoc: int): void
+ hasMonster(): boolean
+ hasKey(): boolean
+ hasTrap(): boolean
+ hasPotion(): boolean
+ hasHealPotion: boolean
+ hasManaPotion: boolean
+ hasVisionPotion: boolean
+ setNorth(set: boolean): void
+ getNorth(): boolean
+ setEast(set: boolean): void
+ getEast(): boolean
+ setSouth(set: boolean): void
+ getSouth(): boolean
+ setWest(set: boolean): void
+ getWest(): boolean
Dungeon
- xLoc: int
- yLoc: int
+ dungeon: char[][]
+ printVisionPotion(): void
+ getXLoc(): int
+ getYLoc(): int
+ setXLoc(x: int): void
+ setYLoc(y: int): void
+ move(direction: String) void
+ getDungeon(): char[][]
GetUserInput
+ heroRaces: ENUM

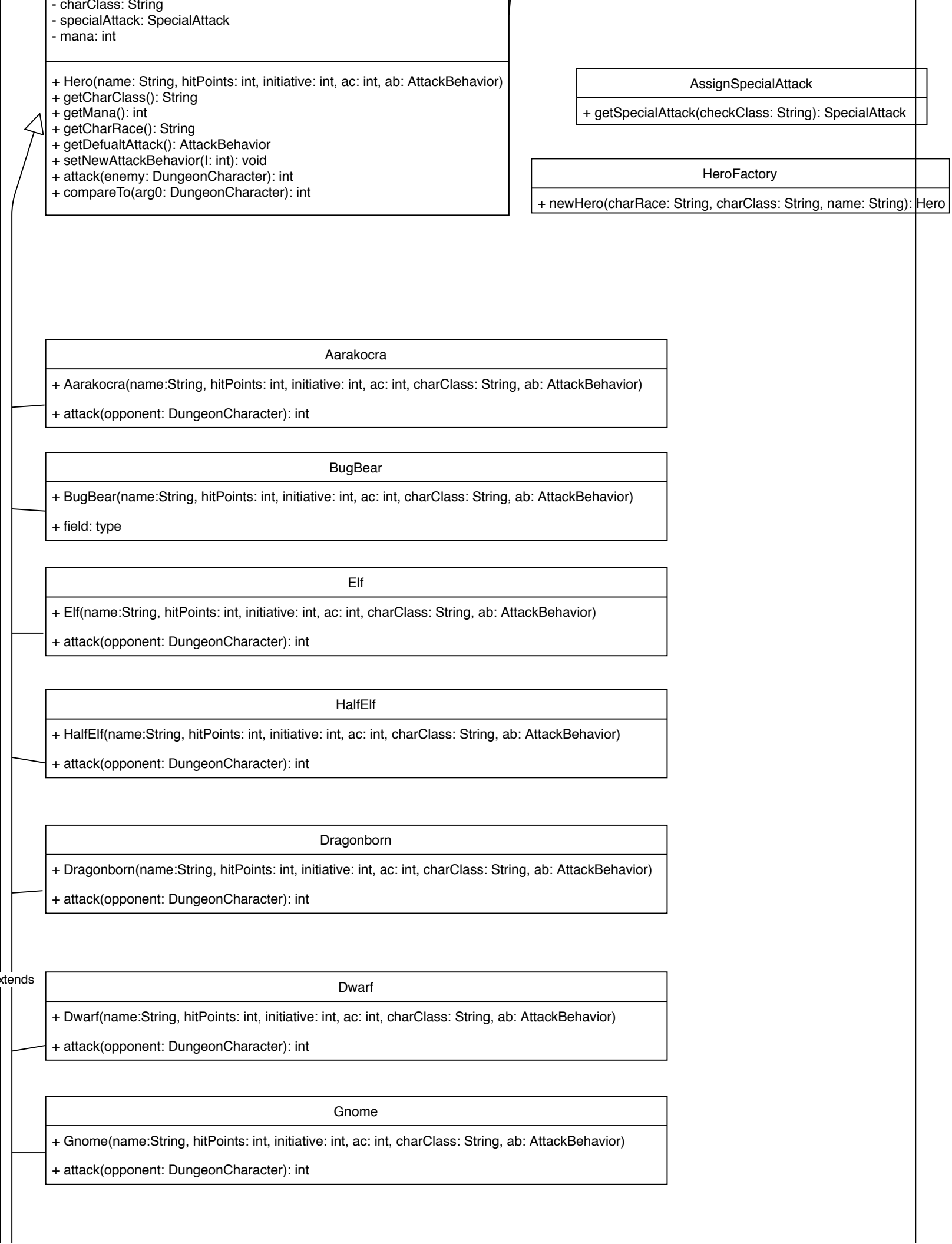

+ heroRaces: ENUM
+ heroClasses: ENUM
+ getRaceFromUser():String
+ getClassFromUser():String
+ getNameFromUser(): String
+ getAttackFromUser(hero: Hero):String
+ getPlayAgainUser(): String
+ getYesOrNoResponse(): String
+ getPotionResponse(): String
+ direction(): String
Dice
+ static D20(): int
+ static D12(): int
+ static D10(): int
+ static D8(): int
+ static D6(): int
+ static D4(): int
<<enumeration>>
heroRaces
AARAKICRA
BUGBEAR
ELF
DRAGONBORN
DWARF
GNOME
GOLIATH
HALFELF
HUMAN
LIZARDFOLK
TRITON
TABAXI
<<enumeration>>
heroClasses
BARBARIAN
BARD
CLERIC
DRUID
FIGHTER
MONK
PALADIN
RANGER
ROGUE
SORCERER
WORLOCK
WIZARD
DungeonDriver
- static hero: Hero
+ static main(arg: String[]):void
- static combat(monster: DungeonCharacter): void
- userPotion(hero: Hero, dungeon: Dungeon): void
- healTurn(hero: Hero, monster: DungeonCharacter): void
- damageExchange(fightOrder: ArrayList<DungeonCharacter): void
- printHealthState(hero:DungeonCharacter, monster: DungeonCharacter): void

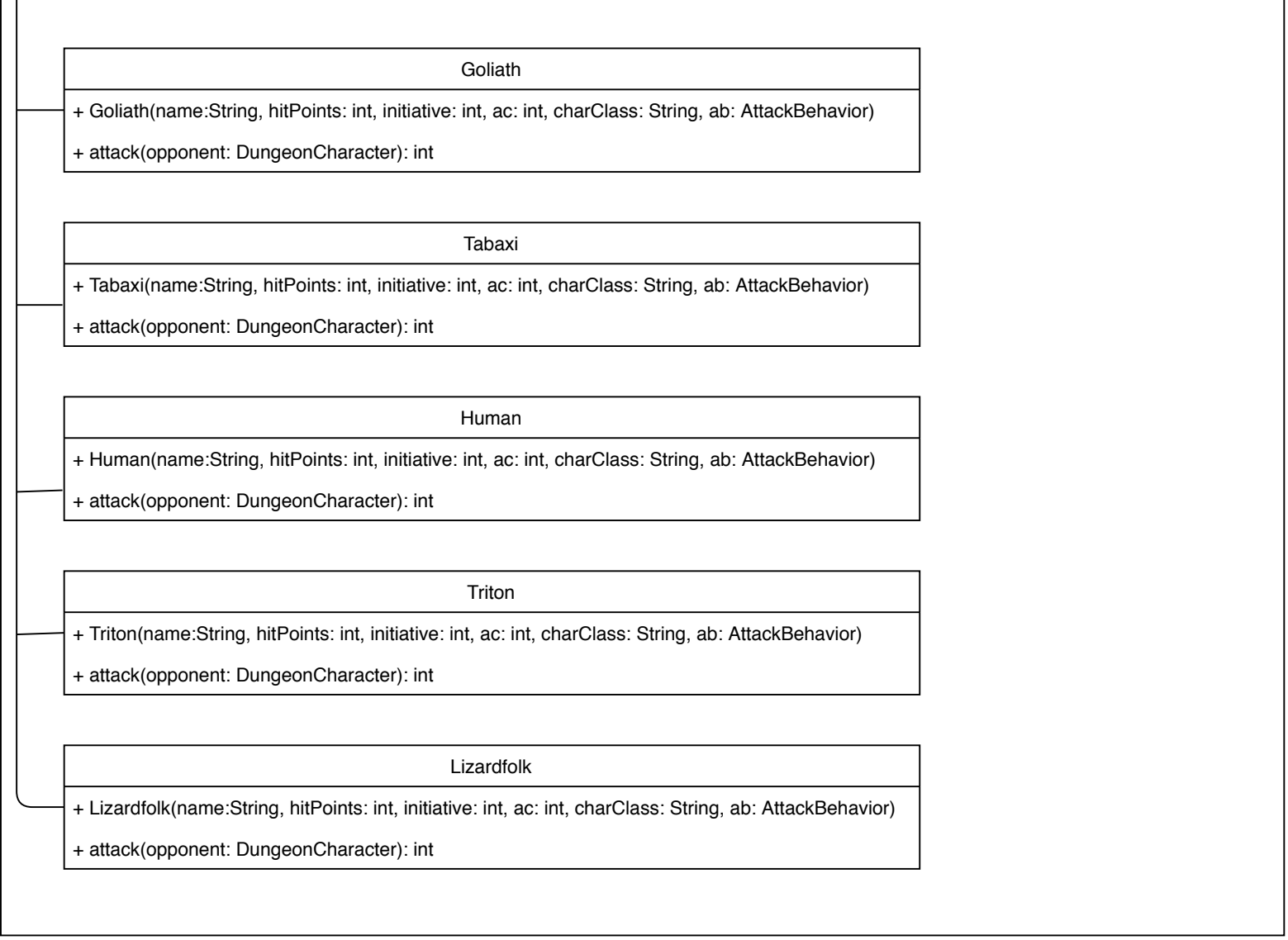

attack
GreatAxe
+ attack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
<<interface>>
attackBehavior
attack(enemy: DungeonCharacter): int
getName(): String
LongSword
+ attack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
AssignSpecialAttac
+ static getSpecialAttack(checkClass:
Club
+ attack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
Mace
+ attack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
Dagger
+ attack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
Quarterstaff
+ attack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
GreatSword
+ attack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
ShortSword
+ attack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String


cks
String): AttackBehavior


specialAttack
<<interface>>
SpecialAttack
specialAttack(enemy: DungeonCharacter): int
getName(): String
DaggerStorm
+ specialAttack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
CureWounds
+ specialAttack(hero: DungeonCharacter): int
+ getName(): String
Entangle
+ specialAttack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
EldrichBlast
+ specialAttack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
MagicMissile
+ specialAttack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
Haste
+ specialAttack(hero: DungeonCharacter): int
+ getName(): String
+ toString(): String
DaggerStorm
+ specialAttack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String
PowerAttack
+ specialAttack(enemy: DungeonCharacter): int
+ getName(): String
+ toString(): String


gameCharacter
<<abstract>>
DungeonCharacter
name: String
hitPoints: int
initiative: int
AC: int
attackBehavior: AttackBehavior
+ DungeonCharacter(name:String, hitPoints:int, initiative: int, AC:int, attackBehavior: AttackBehavior)
+ getAC():int
+ setAC(AC:int):void
+ getName():String
+ getAttackBehavior(): AttackBehavior
+ getHitPoints(): int
+ getInitiative(): int
+ setInitiative(initiative: int): void
+ addHitPoints(hitPoints: int):void
+ subtractHitPoints(hitPoints: int):void
+ isAlive(): boolean
Monster
+ Monster(name: String, hitPoints: int, initiative: int, ac: int, ab: AttackBehavior)
+ setNewMonsterAttack(attack: AttackBehavior): void
+ attack(enemy: DungeonCharacter): int
Extends
Skeleton
+ Skeleton()
+ toString: String
Bandit
+ Bandit()
+ toString: String
Orc
+ Orc()
+ toString: String
KoBold
+ KoBold()
+ toString: String
Extends
Extends
Goblin
+ Goblin()
+ toString: String
Gnoll
+ Gnoll()
+ toString: String
MonsterFactory
+ newMonster(n: int): Monster
Hero
- charRace: String


- charClass: String
- specialAttack: SpecialAttack
- mana: int
+ Hero(name: String, hitPoints: int, initiative: int, ac: int, ab: AttackBehavior)
+ getCharClass(): String
+ getMana(): int
+ getCharRace(): String
+ getDefualtAttack(): AttackBehavior
+ setNewAttackBehavior(I: int): void
+ attack(enemy: DungeonCharacter): int
+ compareTo(arg0: DungeonCharacter): int
AssignSpecialAttack
+ getSpecialAttack(checkClass: String): SpecialAttack
HeroFactory
+ newHero(charRace: String, charClass: String, name: String): Hero
Aarakocra
+ Aarakocra(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ attack(opponent: DungeonCharacter): int
BugBear
+ BugBear(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ field: type
Elf
+ Elf(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ attack(opponent: DungeonCharacter): int
HalfElf
+ HalfElf(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ attack(opponent: DungeonCharacter): int
Dragonborn
+ Dragonborn(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ attack(opponent: DungeonCharacter): int
xtends
Dwarf
+ Dwarf(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ attack(opponent: DungeonCharacter): int
Gnome
+ Gnome(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ attack(opponent: DungeonCharacter): int


Goliath
+ Goliath(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ attack(opponent: DungeonCharacter): int
Tabaxi
+ Tabaxi(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ attack(opponent: DungeonCharacter): int
Human
+ Human(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ attack(opponent: DungeonCharacter): int
Triton
+ Triton(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ attack(opponent: DungeonCharacter): int
Lizardfolk
+ Lizardfolk(name:String, hitPoints: int, initiative: int, ac: int, charClass: String, ab: AttackBehavior)
+ attack(opponent: DungeonCharacter): int