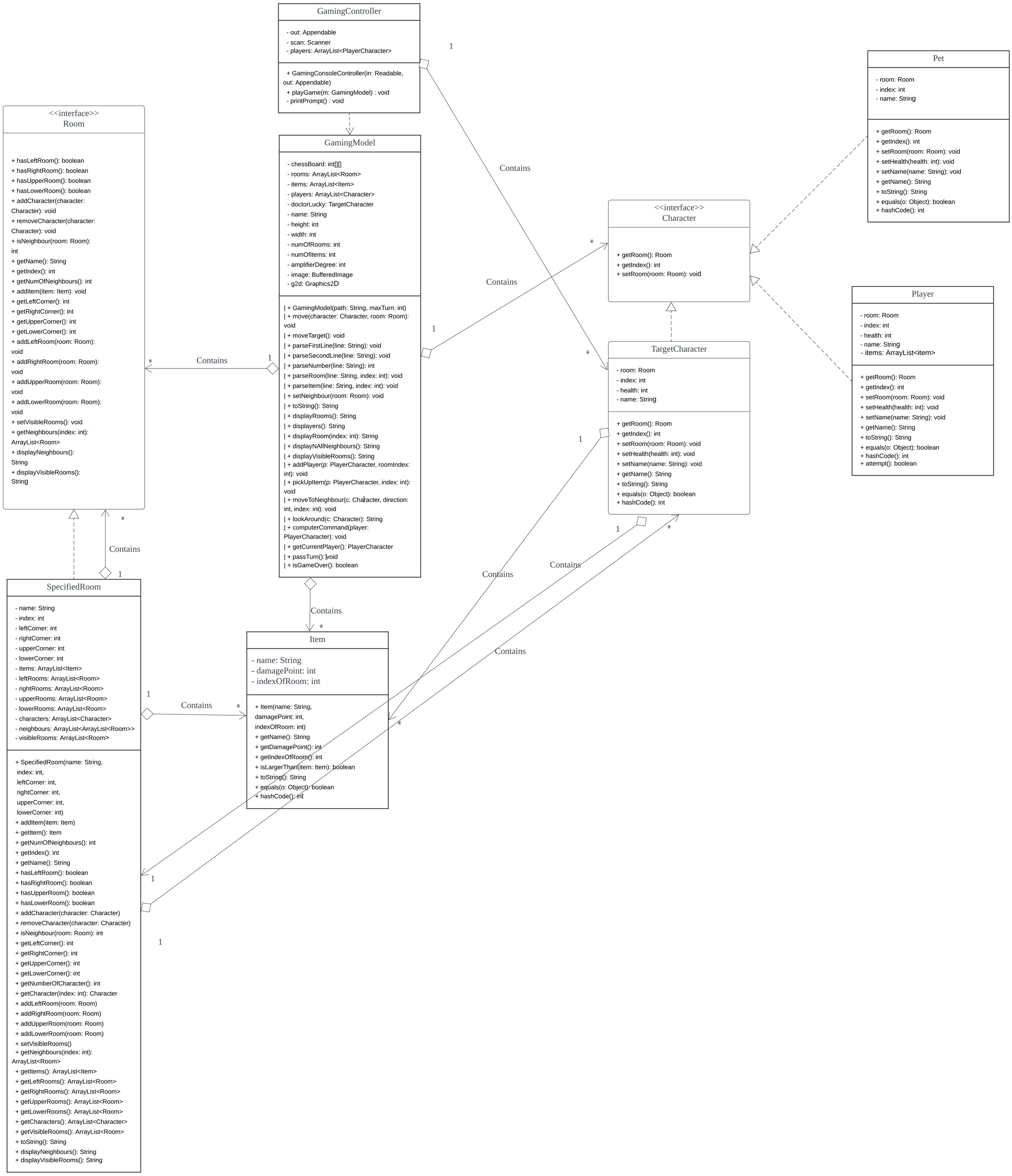
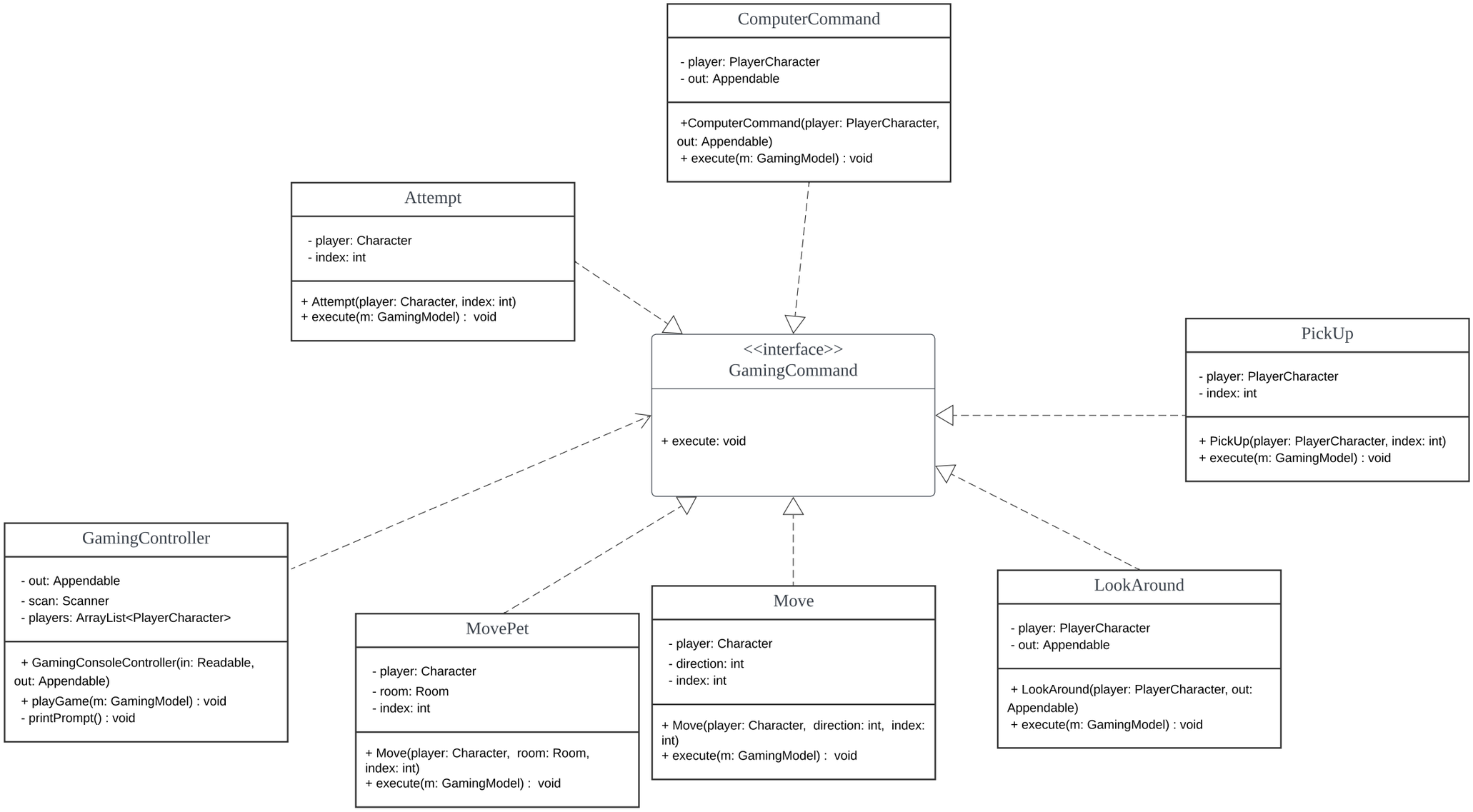

GamingController
- out: Appendable
- scan: Scanner
- players: ArrayList<PlayerCharacter>
+ GamingConsoleController(in: Readable, out: Appendable)
+ playGame(m: GamingModel) : void
- printPrompt() : void
GamingModel
- chessBoard: int[][]
- rooms: ArrayList<Room>
- items: ArrayList<Item>
- players: ArrayList<Character>
- doctorLucky: TargetCharacter
- name: String
- height: int
- width: int
- numOfRooms: int
- numOfItems: int
- amplifierDegree: int
- image: BufferedImage
- g2d: Graphics2D
| + GamingModel(path: String, maxTurn: int)
| + move(character: Character, room: Room): void
| + moveTarget(): void
| + parseFirstLine(line: String): void
| + parseSecondLine(line: String): void
| + parseNumber(line: String): int
| + parseRoom(line: String, index: int): void
| + parseItem(line: String, index: int): void
| + setNeighbour(room: Room): void
| + toString(): String
| + displayRooms(): String
| + displayers(): String
| + displayRoom(index: int): String
| + displayNAllNeighbours(): String
| + displayVisibleRooms(): String
| + addPlayer(p: PlayerCharacter, roomIndex: int): void
| + pickUpItem(p: PlayerCharacter, index: int): void
| + moveToNeighbour(c: Character, direction: int, index: int): void
| + lookAround(c: Character): String
| + computerCommand(player: PlayerCharacter): void
| + getCurrentPlayer(): PlayerCharacter
| + passTurn(): void
| + isGameOver(): boolean
<<interface>>
Room
+ hasLeftRoom(): boolean
+ hasRightRoom(): boolean
+ hasUpperRoom(): boolean
+ hasLowerRoom(): boolean
+ addCharacter(character: Character): void
+ removeCharacter(character: Character): void
+ isNeighbour(room: Room): int
+ getName(): String
+ getIndex(): int
+ getNumOfNeighbours(): int
+ addItem(item: Item): void
+ getLeftCorner(): int
+ getRightCorner(): int
+ getUpperCorner(): int
+ getLowerCorner(): int
+ addLeftRoom(room: Room): void
+ addRightRoom(room: Room): void
+ addUpperRoom(room: Room): void
+ addLowerRoom(room: Room): void
+ setVisibleRooms(): void
+ getNeighbours(index: int): ArrayList<Room>
+ displayNeighbours(): String
+ displayVisibleRooms(): String
<<interface>>
Character
+ getRoom(): Room
+ getIndex(): int
+ setRoom(room: Room): void
Pet
- room: Room
- index: int
- name: String
+ getRoom(): Room
+ getIndex(): int
+ setRoom(room: Room): void
+ setHealth(health: int): void
+ setName(name: String): void
+ getName(): String
+ toString(): String
+ equals(o: Object): boolean
+ hashCode(): int
Player
- room: Room
- index: int
- health: int
- name: String
- items: ArrayList<item>
+ getRoom(): Room
+ getIndex(): int
+ setRoom(room: Room): void
+ setHealth(health: int): void
+ setName(name: String): void
+ getName(): String
+ toString(): String
+ equals(o: Object): boolean
+ hashCode(): int
+ attempt(): boolean
TargetCharacter
- room: Room
- index: int
- health: int
- name: String
+ getRoom(): Room
+ getIndex(): int
+ setRoom(room: Room): void
+ setHealth(health: int): void
+ setName(name: String): void
+ getName(): String
+ toString(): String
+ equals(o: Object): boolean
+ hashCode(): int
SpecifiedRoom
- name: String
- index: int
- leftCorner: int
- rightCorner: int
- upperCorner: int
- lowerCorner: int
- items: ArrayList<Item>
- leftRooms: ArrayList<Room>
- rightRooms: ArrayList<Room>
- upperRooms: ArrayList<Room>
- lowerRooms: ArrayList<Room>
- characters: ArrayList<Character>
- neighbours: ArrayList<ArrayList<Room>>
- visibleRooms: ArrayList<Room>
+ SpecifiedRoom(name: String, index: int, leftCorner: int, rightCorner: int, upperCorner: int, lowerCorner: int)
+ addItem(item: Item)
+ getItem(): Item
+ getNumOfNeighbours(): int
+ getIndex(): int
+ getName(): String
+ hasLeftRoom(): boolean
+ hasRightRoom(): boolean
+ hasUpperRoom(): boolean
+ hasLowerRoom(): boolean
+ addCharacter(character: Character)
+ removeCharacter(character: Character)
+ isNeighbour(room: Room): int
+ getLeftCorner(): int
+ getRightCorner(): int
+ getUpperCorner(): int
+ getLowerCorner(): int
+ getNumberOfCharacter(): int
+ getCharacter(index: int): Character
+ addLeftRoom(room: Room)
+ addRightRoom(room: Room)
+ addUpperRoom(room: Room)
+ addLowerRoom(room: Room)
+ setVisibleRooms()
+ getNeighbours(index: int): ArrayList<Room>
+ getItems(): ArrayList<Item>
+ getLeftRooms(): ArrayList<Room>
+ getRightRooms(): ArrayList<Room>
+ getUpperRooms(): ArrayList<Room>
+ getLowerRooms(): ArrayList<Room>
+ getCharacters(): ArrayList<Character>
+ getVisibleRooms(): ArrayList<Room>
+ toString(): String
+ displayNeighbours(): String
+ displayVisibleRooms(): String
Item
- name: String
- damagePoint: int
- indexOfRoom: int
+ Item(name: String, damagePoint: int, indexOfRoom: int)
+ getName(): String
+ getDamagePoint(): int
+ getIndexOfRoom(): int
+ isLargerThan(item: Item): boolean
+ toString(): String
+ equals(o: Object): boolean
+ hashCode(): int
Contains
1
*


ComputerCommand
- player: PlayerCharacter
- out: Appendable
+ComputerCommand(player: PlayerCharacter, out: Appendable)
+ execute(m: GamingModel) : void
Attempt
- player: Character
- index: int
+ Attempt(player: Character, index: int)
+ execute(m: GamingModel) : void
<<interface>>
GamingCommand
+ execute: void
PickUp
- player: PlayerCharacter
- index: int
+ PickUp(player: PlayerCharacter, index: int)
+ execute(m: GamingModel) : void
GamingController
- out: Appendable
- scan: Scanner
- players: ArrayList<PlayerCharacter>
+ GamingConsoleController(in: Readable, out: Appendable)
+ playGame(m: GamingModel) : void
- printPrompt() : void
MovePet
- player: Character
- room: Room
- index: int
+ Move(player: Character, room: Room, index: int)
+ execute(m: GamingModel) : void
Move
- player: Character
- direction: int
- index: int
+ Move(player: Character, direction: int, index: int)
+ execute(m: GamingModel) : void
LookAround
- player: PlayerCharacter
- out: Appendable
+ LookAround(player: PlayerCharacter, out: Appendable)
+ execute(m: GamingModel) : void