* GameObject
  + GameObject(String XML): takes an XML Block
    - Set private String \_XMLBlock to XML parameter
    - Takes XML block’s name, sets \_name to that
    - Takes XML block’s description, sets \_description to that
    - Calls initialize
  + name(): returns the name of a GameObject when called. This is a String.
  + Description(): [Human readable] examines its internal state and describes itself to the user when called. There may be functionality in this method that can change the description. This is String
    - i.e. a description of a light or room can change if it is lit or not.
  + Verbs(): this is a list of verb commands that can be used on GameObjects. Specifically what verbs can be used on an object. Go example Go Shoggoth is not a valid verb use.
  + Inventory(): List of GameObject, an actual list of all GameObjects in the GameObject we are calling for the program to enumerate through.
  + doVerb(String): Takes input of what verb is being performed on this object (possibly with verb modifiers) and performs whatever functionality is necessary for that verb. Return a human readable description of the results of the action
  + private think(): this function examines the all parts of the game state this Object has access to and performs whatever actions this Object needs to perform this cycle.
  + Private initialize(): performs the following actions
    - Scans the objects XML block
    - Takes the XML block’s Inventory list
    - For each entry in the XML block’s \_inventory list, finds or spawns the corresponding named GameObject, then adds that GameObject to this GameObjects \_inventory
    - Takes the XML block’s Command List, for each entry in the Command List, adds the Command entries verb to this Objects verb list and links it to the corresponding GameCommand
    - Load the XML block’s Think subsection and connects this GameObjects think function to the corresponding global Think function.
    - Load the XML block’s Description subsection and set the \_description to that String
  + Private \_inventoryList: List of all GameObjects that are the inventory of this GameObject
  + Private name: name of a GameObject
  + Private \_description: the base static description of a GameObject in its default state
  + Private \_verbs: list of all GameCommands and the verbs that activate them
  + Private String \_XMLBlock: holds this objects XML block in case it needs to refer to it without having to reload the entire XML file
* RoomObject
  + Exits(): enumerates through the list of the all GameObjects and builds a list of all Objects that are DoorObjects.
  + Creatures(): List of CreatureObjects, ennumeratres through the list of all GameObjects and builds a list of all Objects that are CreatureObjects .
  + @override private initialize():
    - Calls GameObjects initialize()
    - loads the XML block’s Exits list,
    - For each entry in the XML’s Exit List
      * it will find the named RoomObject that corresponds to the Exit’s Destination tag
      * Spawn a new DoorObject
      * Insert the new DoorObject into this object’s \_inventory
      * set the new DoorObject’s destination to the exit’s RoomObject
      * set the DoorObject’s status to the RoomObject’s status
      * If the Exit entry has a DoorObject subsection, load this into the new DoorObject’s XMLblock and call the DoorObject’s initialize() function.
* ExitObject
  + Destination(): returns the RoomObject on the other side of this exit
  + Private int \_status; is a variable that determines a doors state
    - i.e. 1 can be locked, 2 can be open, 3 can be jammed, and 4 could be there is a Shoggoth in the way
  + private \_destination; RoomObject is a variable that determines where a door will lead to and therefore what description to give the player next and set their location.
* MobileObject
  + @override Private think(): override to have a much more elaborate function
    - Primarily it has to look at its path
  + @override Inventory(): String,
  + Private \_location: returns the RoomObject that this Object is currently in
  + Private \_path: List of RoomObject that this MobileObject will travel through as it moves. Used by think().
  + Private \_pathindex: entry in the \_path list the MobileObject is currently on.
* ItemObject
* CreatureObject
  + Equipped(): when called will print out a list of ItemObjects that are currently equipped to the character.
  + useEquipment(): when called will perform an action with the equipment to effect another GameObject.
  + @override think(): (optional)
  + Private \_pathBack: List of RoomObjects (optional)
  + Private \_target: CreatureObject (optional)
* MonsterObject
  + @override think(): add ability to eat
* CharacterObject
  + @override think(): will call Client Object’s input and parse that input
    - Coordinates with the Client Object
* GameCommand
  + verbName(): returns the name of the verb as it would typed by the client to perform that verb
  + modifiers(): Secondary parameters for a verb (at, in, with, through) that alter the verbs function or GameObject
  + doVerb(): performs that verbs action with the modifiers provided.
* Client
  + Get File path: reads in a user inputted file path
  + Load XML file: using a do/while loop using Java’s XML interpreter
  + Read Client input: (do Loop) exit condition of the loop will be an exit command
  + Private String \_filePath: a blank variable that the file path will be set to in order to load it.