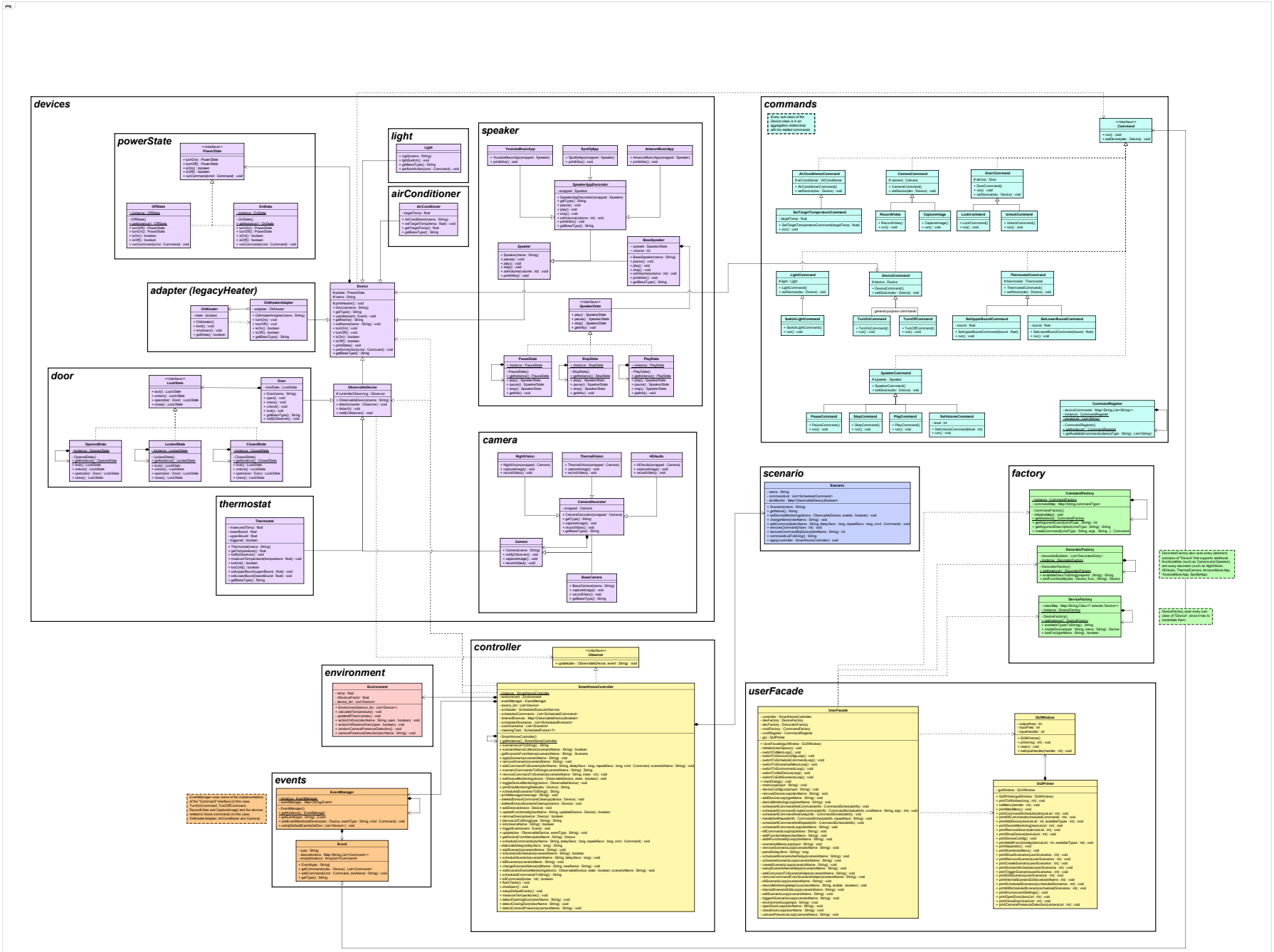
pkg
devices
powerState
<<interface>>
PowerState
+ turnOn() : PowerState
+ turnOff() : PowerState
+ isOn() : boolean
+ isOff() : boolean
+ runCommand(cmd : Command) : void
OffState
- instance : OffState
- OffState()
- getInstance() : OffState
+ turnOn() : PowerState
+ turnOff() : PowerState
+ isOn() : boolean
+ isOff() : boolean
+ runCommand(cmd : Command) : void
OnState
- instance : OnState
- OnState()
- getInstance() : OnState
+ turnOn() : PowerState
+ turnOff() : PowerState
+ isOn() : boolean
+ isOff() : boolean
+ runCommand(cmd : Command) : void
light
Light
+ Light(name : String)
+ lightSwitch() : void
+ getBaseType() : String
+ performAction(cmd : Command) : void
airConditioner
AirConditioner
- targetTemp : float
+ AirConditioner(name : String)
+ setTargetTemp(temp : float) : void
+ getTargetTemp() : float
+ getBaseType() : String
speaker
YoutubeMusicApp
+ YoutubeMusicApp(wrapped : Speaker)
+ printInfo() : void
SpotifyApp
+ SpotifyApp(wrapped : Speaker)
+ printInfo() : void
AmazonMusicApp
+ AmazonMusicApp(wrapped : Speaker)
+ printInfo() : void
SpeakerAppDecorator
- wrapped : Speaker
+ SpeakerAppDecorator(wrapped : Speaker)
+ getType() : String
+ pause() : void
+ play() : void
+ stop() : void
+ setVolume(volume : int) : void
+ printInfo() : void
+ getBaseType() : String
Speaker
+ Speaker(name : String)
+ pause() : void
+ play() : void
+ stop() : void
+ setVolume(volume : int) : void
+ printInfo() : void
BaseSpeaker
- volume : int
+ BaseSpeaker(name : String)
+ pause() : void
+ play() : void
+ stop() : void
+ setVolume(volume : int) : void
+ printInfo() : void
+ getBaseType() : String
adapter (legacyHeater)
OldHeater
- state : boolean
+ OldHeater()
+ boot() : void
+ shutdown() : void
+ getState() : boolean
OldHeaterAdapter
- adaptee : OldHeater
+ OldHeaterAdapter(name : String)
+ turnOn() : void
+ turnOff() : void
+ isOn() : boolean
+ isOff() : boolean
+ getBaseType() : String
Device
name : String
printInfo() : void
+ Device(name : String)
+ getName() : String
+ update(event : Event) : void
+ getName() : String
+ setName(name : String) : void
+ turnOn() : void
+ turnOff() : void
+ isOn() : boolean
+ isOff() : boolean
+ printInfo() : void
+ performAction(cmd : Command) : void
+ getBaseType() : String
<<interface>>
SpeakerState
+ play() : SpeakerState
+ pause() : SpeakerState
+ stop() : SpeakerState
+ getInfo() : void
PauseState
- instance : PauseState
- PauseState()
+ getInstance() : PauseState
+ play() : SpeakerState
+ pause() : SpeakerState
+ stop() : SpeakerState
+ getInfo() : void
StopState
- instance : StopState
- StopState()
+ getInstance() : StopState
+ play() : SpeakerState
+ pause() : SpeakerState
+ stop() : SpeakerState
+ getInfo() : void
PlayState
- instance : PlayState
- PlayState()
+ getInstance() : PlayState
+ play() : SpeakerState
+ pause() : SpeakerState
+ stop() : SpeakerState
+ getInfo() : void
door
<<interface>>
LockState
+ lockCo() : LockState
+ unlockCo() : LockState
+ open(door : Door) : LockState
+ close() : LockState
Door
- lockState : LockState
+ Door(name : String)
+ open() : void
+ close() : void
+ unlock() : void
+ lock() : void
+ getBaseType() : String
+ notifyObserver() : void
ObservableDevice
controlleObserving : Observer
+ ObservableDevice(name : String)
+ attach(master : Observer) : void
+ detach() : void
+ notifyObserver() : void
OpenedState
- instance : OpenedState
- OpenedState()
+ getInstance() : OpenedState
+ lock() : LockState
+ unlock() : LockState
+ open(door : Door) : LockState
+ close() : LockState
LockedState
- instance : LockedState
- LockedState()
+ getInstance() : LockedState
+ lock() : LockState
+ unlock() : LockState
+ open(door : Door) : LockState
+ close() : LockState
ClosedState
- instance : ClosedState
- ClosedState()
+ getInstance() : ClosedState
+ lock() : LockState
+ unlock() : LockState
+ open(door : Door) : LockState
+ close() : LockState
thermostat
Thermostat
- measuredTemp : float
- upperBound : float
- lowerBound : float
- triggered : boolean
+ Thermostat(name : String)
+ getTemperature() : float
+ notifyObserver() : void
+ measureTemperature(temperature : float) : void
+ boilCold() : boolean
+ boilCold() : boolean
+ setUpperBound(upperBound : float) : void
+ setLowerBound(lowerBound : float) : void
+ getBaseType() : String
camera
NightVision
+ NightVision(wrapped : Camera)
+ captureImage() : void
+ recordVideo() : void
ThermalVision
+ ThermalVision(wrapped : Camera)
+ captureImage() : void
+ recordVideo() : void
HDAudio
+ HDAudio(wrapped : Camera)
+ captureImage() : void
+ recordVideo() : void
CameraDecorator
- wrapped : Camera
+ CameraDecorator(wrapped : Camera)
+ getType() : String
+ captureImage() : void
+ recordVideo() : void
+ getBaseType() : String
Camera
+ Camera(name : String)
+ notifyObserver() : void
+ captureImage() : void
+ recordVideo() : void
BaseCamera
+ BaseCamera(name : String)
+ captureImage() : void
+ recordVideo() : void
+ getBaseType() : String
commands
Every sub-class of the Device class is in an aggregation relationship with his related commands.
<<interface>>
Command
+ run() : void
+ setDevice(dev : Device) : void
AirConditionerCommand
airConditioner : AirConditioner
+ AirConditionerCommand()
+ setDevice(dev : Device) : void
CameraCommand
camera : Camera
+ CameraCommand()
+ setDevice(dev : Device) : void
DoorCommand
device : Door
+ DoorCommand()
+ run() : void
+ setDevice(dev : Device) : void
SetTargetTemperatureCommand
- targetTemp : float
+ SetTargetTemperatureCommand(targetTemp : float)
+ run() : void
RecordVideo
+ RecordVideo()
+ run() : void
CaptureImage
+ CaptureImage()
+ run() : void
LockCommand
+ LockCommand()
+ run() : void
UnlockCommand
+ UnlockCommand()
+ run() : void
LightCommand
light : Light
+ LightCommand()
+ setDevice(dev : Device) : void
DeviceCommand
device : Device
+ DeviceCommand()
+ setDevice(dev : Device) : void
ThermostatCommand
thermostat : Thermostat
+ ThermostatCommand()
+ setDevice(dev : Device) : void
SwitchLightCommand
+ SwitchLightCommand()
+ run() : void
general purpose commands
TurnOnCommand
+ TurnOnCommand()
+ run() : void
TurnOffCommand
+ TurnOffCommand()
+ run() : void
SetUpperBoundCommand
- bound : float
+ Set(upperBoundCommand(bound : float)
+ run() : void
SetLowerBoundCommand
- bound : float
+ SetLowerBoundCommand(bound : float)
+ run() : void
SpeakerCommand
speaker : Speaker
+ SpeakerCommand()
+ setDevice(dev : Device) : void
PauseCommand
+ PauseCommand()
+ run() : void
StopCommand
+ StopCommand()
+ run() : void
PlayCommand
+ PlayCommand()
+ run() : void
SetVolumeCommand
- level : int
+ SetVolumeCommand(level : int)
+ run() : void
CommandRegister
- deviceCommands : Map<String,List<String>>
- instance : CommandRegister
- instlist : List<String>
- CommandRegister()
+ getInstance() : CommandRegister
+ getAvailableCommands(deviceType : String) : List<String>
scenario
Scenario
- name : String
- commandList : List<ScheduledCommand>
- devMonitor : Map<ObservableDevice,Boolean>
+ Scenario(name : String)
+ getName() : String
+ setDeviceMonitoring(device : ObservableDevice, enable : boolean) : void
+ changeName(newName : String) : void
+ addCommand(devName : String, delaySecs : long, repeatSecs : long, cmd : Command) : void
+ removeCommand(index : int) : void
+ removeCommandToScenario(scenarioName : String) : int
+ removeCommandToFromScenario(index : int)
+ applyController(controller : SmartHomeController) : void
factory
CommandFactory
- instance : CommandFactory
- commandMap : Map<String,CommandType>
- CommandFactory()
+ initializeMap() : void
+ getInstance() : CommandFactory
+ getAvailableCommandsOfType(type : String) : int
+ getAvailableCommandDescriptionsOfType : String) : String
+ createCommand(cmdType : String, args : String...) : Command
DecoratorFactory
- decoratorBuilders : List<DecoratorEntry>
- instance : DecoratorFactory
- DecoratorFactory()
+ getInstance() : DecoratorFactory
+ getAvailableDecoratorsToWrap(wrapped : String) : String
+ addFunctionality(dev : Device, func : String) : Device
DeviceFactory
- classMap : Map<String,Class<? extends Device>>
- instance : DeviceFactory
- DeviceFactory()
+ getInstance() : DeviceFactory
+ availableTypesToBuild() : String
+ create(deviceType : String, name : String) : Device
+ isInstTypeName(name : String) : boolean
DecoratorFactory also uses every (abstract) subclass of "Device" that supports additional functionality (such as SpeakerAppDecorator) and every decorator (such as NightVision, HDAudio, ThermalCamera, AmazonMusicApp, YoutubeMusicApp, SpotifyApp).
class of "Device", since it has to instantiate them.
controller
<<interface>>
Observer
+ update(dev : ObservableDevice, event : String) : void
SmartHomeController
- instance : SmartHomeController
- environment : Environment
- eventManager : EventManager
- device_list : List<Device>
- scheduler : ScheduledExecutorService
- scheduledCommands : List<ScheduledCommand>
- listenedDevices : Map<ObservableDevice,Boolean>
- scheduledScenarios : List<ScheduledScenario>
- userScenarios : List<Scenario>
- cleaningTask : ScheduledFuture<?>
+ getInstance() : SmartHomeController
+ addScenarioToCommand(scenarioName : String) : void
+ scenarioNameCollision(scenarioName : String) : boolean
+ getScenarioFromNameInscenarioName : String) : Scenario
+ applyScenario(scenarioName : String) : void
+ removeScenario(scenarioName : String) : void
+ addCommandToScenario(scenarioName : String, devName : String, repeatSecs : long, cmd : Command, scenarioName : String) : void
+ addCommandToScenario(scenarioName : String) : String
+ removeCommandToScenario(scenarioName : String, index : int) : void
+ setDeviceMonitoring(device : ObservableDevice, state : boolean) : void
+ toggleDeviceMonitoring(device : ObservableDevice) : void
+ printDeviceMonitoringStatus(dev : Device) : String
+ printDeviceMonitoring() : void
+ printMessage(message : String) : void
- deleteDeviceCommandOnwards(device : Device) : void
- deleteDeviceScenarioOnwards(device : Device) : void
- addDevice(device : Device) : void
+ updateFunctionality(devName : String, updatedDevice : Device) : boolean
+ removeDevice(device : Device) : boolean
+ getDevice(name : String) : Device
+ setAliveName(name : String) : boolean
+ triggerEvent(event : Event) : void
+ update(dev : ObservableDevice, eventType : String) : void
+ getDeviceFromNameInDevName : String) : Device
+ scheduleCommand(devName : String, delaySecs : long, repeatSecs : long, cmd : Command) : void
- elaborateDelaySeks(seks : long) : String
+ addScenarioToSchedule(scenarioName : String) : void
+ isScenarioScheduled(scenarioName : String) : boolean
+ scheduleScenario(scenarioName : String, delaySecs : long) : void
+ isScenario(scenarioName : String) : void
+ changeScenarioName(oldName : String, newName : String) : void
+ setup(dev : ObservableDevice, state : ObservableDevice, state : boolean, scenarioName : String) : void
+ scheduleSetCommandToWrap() : String
+ initCommand(index : int) : boolean
+ flushTasks() : void
+ shutdown() : void
+ setupDefaultEvents() : void
+ measureTemperature() : void
+ setup(devName : String, scenarioName : String) : void
+ detectClosing(doorUserName : String) : void
+ detectCameraPresence(cameraName : String) : void
environment
Environment
- temp : float
- influenceFactor : float
- device_list : List<Device>
+ Environment(device_list : List<Device>)
+ calculateTemperature() : void
+ updateAllThermostats() : void
+ setDoorOpen(doorUserName : String, open : boolean) : void
+ actionOnNeighbourDoor(open : boolean) : void
+ randomCameraPresenceDetection() : void
+ cameraPresenceDetection(devName : String) : void
events
EventManager
- instance : EventManager
- eventMap : Map<String,Event>
- EventManager()
+ getInstance() : EventManager
+ getEvent(type : String) : Event
+ addEventWhileDevice(dev : Device, eventType : String, cmd : Command) : void
+ setupDefaultEvents(devDev : List<Device>) : void
Event
- type : String
- deviceActions : Map<String,List<Command>>
- emptyInstance : ArrayList<Command>
+ Event(type : String)
+ getCommands(dev : Device) : List<Command>
+ addCommand(cmd : Command, devName : String) : void
+ getType() : String
EventManager uses some of the implementations of the "Command" interface (in this case, TurnOnCommand, TurnOffCommand, SetVolumeCommand) to interact with the devices related to these commands (in this case, OldHeaterAdapter, AirConditioner and Camera)
userFacade
UserFacade
- controller : SmartHomeController
- cmdFactory : CommandFactory
- decFactory : DecoratorFactory
- cmdFactory : CommandFactory
- cmdRegister : CommandRegister
- gui : GUIPrinter
+ UserFacade(guiWindow : GUIWindow)
- initializeUserSpace() : void
- switchToMainLoop() : void
- switchToDevicesLoop() : void
- switchToCommandsMenuLoop() : void
- switchToScenarioMenuLoop() : void
- switchToEnvironmentLoop() : void
- switchToAddDevicesLoop() : void
- switchToEditScenariosLoop() : void
- initCmdLoop() : void
- startLoop(loop : String) : void
- deviceConfigLoop(input : String) : void
- removeDevicesLoop(devName : String) : void
- addDevicesLoop(devName : String) : void
- devicMonitoringLoop(devName : String) : void
- scheduleAllCommandsAndCommandsInfo : CommandScheduleInfo) : void
- scheduleCommandAndDelayInfo : CommandScheduleInfo, cmdName : String, args : int) : void
- scheduleCommandAndDelayInfo : CommandScheduleInfo) : void
- handleAskRepeatInfo : CommandScheduleInfo, repeatInput : String) : void
- scheduleCommandAndRepeatInfo : CommandScheduleInfo) : void
- scheduleCommandLoop(devName : String) : void
- editCommandLoop(input/line : String) : void
- addFunctionsHelper(devName : String) : void
- addFunctionsLoop(devName : String) : void
- scenariosMenuLoop(input : String) : void
- removeScenarioLoop(scenarioName : String) : void
- parseDelay(time : String) : long
- scheduleAllScenariosAllDelays(scenarioName : String) : void
- scheduleScenarioAndDelayInfo(scenarioName : String) : void
- createScenarioLoop(scenarioName : String) : void
- setupScenarioNameHelper(scenarioNameIn : String) : void
- addCommandToScenarioHelper(scenarioNameIn : String) : void
- removeCommandFromScenarioHelper(scenarioNameIn : String) : void
- editScenarioLoop(scenarioNameIn : String) : void
- devicMonitoringHelper(scenarioNameIn : String, enable : boolean) : void
- addDelayToScenarioHelper(scenarioNameIn : String) : void
- editScenarioLoop(scenarioNameIn : String) : void
- triggerAScenarioLoop(scenarioNameIn : String) : void
- environmentLoop(input : String) : void
- openDoorLoop(doorName : String) : void
- closeDoorLoop(doorName : String) : void
- cameraPresenceLoop(cameraName : String) : void
GUIWindow
- outputArea : int
- inputField : int
- inputHandler : int
+ GUIWindow()
+ print(msg : int) : void
+ clear() : void
+ setInputHandler(handler : int) : void
GUIPrinter
- guiWindow : GUIWindow
+ GUIPrinter(guiWindow : GUIWindow)
+ printMainMenu() : void
+ setMainHandler(h : int) : void
+ printMainMenu() : void
- printCommandSchedulerDevicesList : int) : void
- printAllCommandsInScheduledCommands : int) : void
- printAllDevicesDetailed(list : int, availableTypes : int) : void
- printDevicesMonitoringDetailed(list : int) : void
- printDevicesMonitoringSummary(list : int) : void
- printDevicesDetailed(list : int) : void
- printDevicesConfig() : void
- printAllAvailableDevicesList : int, availableTypes : int) : void
- printSeparator() : void
- printScenarioMenu() : void
- printScenariosLoop(userScenarios : int) : void
- printRemoveScenariosLoop(userScenarios : int) : void
- printCreateScenariosLoop(userScenarios : int) : void
- printScenarioSchedulerLoop(userScenarios : int) : void
- printTriggerScenariosLoop(userScenarios : int) : void
- printEditScenariosLoop(userScenarios : int) : void
- printAddCommandToScenarioLoop(scenarios : int) : void
- printAddDelayToScenarioLoop(scenarioNameIn : int) : void
- printAllScheduledScenariosLoop(scheduledScenarios : int) : void
- printEnvironmentDisplay() : void
- printOpenDoorsLoop(doorList : int) : void
- printCloseDoorsLoop(doorList : int) : void
- printCameraPresenceDetectionLoop(cameraList : int) : void