Name: Bryan J Sanchez
ID: 802-07-8188
Section: 031

<u>Project 2: Operation Sahure UML Design</u>

iomanager package	main package	pyramid package

iomanager package

```
<<Java Class>>
Ⓖ InputReader
iomanager
```
- ⚲ InputReader()
- ⚲ getCaseList():ArrayList<Case>
- ⚲ parseMembers(String):PyramidNetwork
- ⚲ getDirectory():String

```
<<Java Class>>
Ⓖ OutputWriter
iomanager
```
- ⚲ OutputWriter()
- ⚲ saveFile(String,int,ArrayList<Scenario>):void
- ⚲ saveInvalidFile(String,String):void
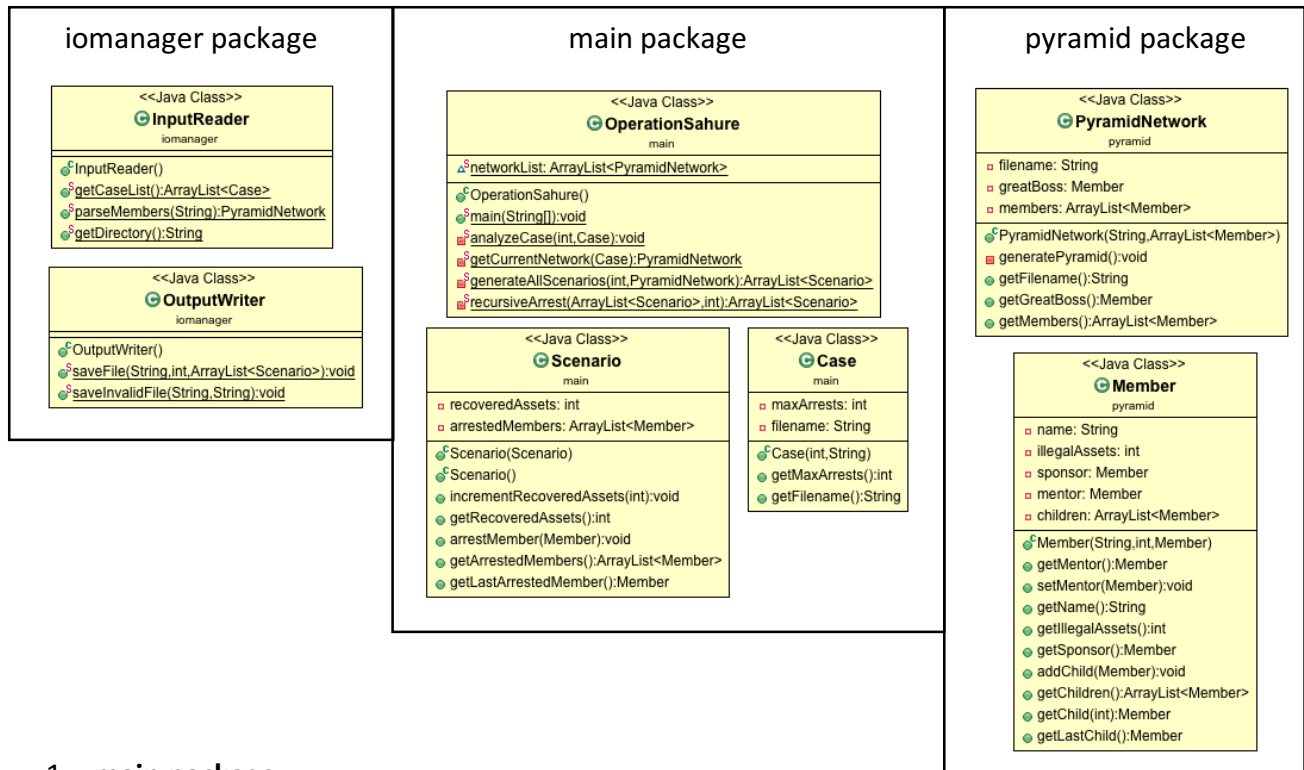
main package

```
<<Java Class>>
Ⓖ OperationSahure
main
```
- networkList: ArrayList<PyramidNetwork>
- ⚲ OperationSahure()
- ⚲ main(String[]):void
- analyzeCase(int,Case):void
- getCurrentNetwork(Case):PyramidNetwork
- generateAllScenarios(int,PyramidNetwork):ArrayList<Scenario>
- recursiveArrest(ArrayList<Scenario>,int):ArrayList<Scenario>

```
<<Java Class>>
Ⓖ Scenario
main
```
- recoveredAssets: int
- arrestedMembers: ArrayList<Member>
- ⚲ Scenario(Scenario)
- ⚲ Scenario()
- incrementRecoveredAssets(int):void
- getRecoveredAssets():int
- arrestMember(Member):void
- getArrestedMembers():ArrayList<Member>
- getLastArrestedMember():Member

```
<<Java Class>>
Ⓖ Case
main
```
- maxArrests: int
- filename: String
- ⚲ Case(int,String)
- getMaxArrests():int
- getFilename():String

pyramid package

```
<<Java Class>>
Ⓖ PyramidNetwork
pyramid
```
- filename: String
- greatBoss: Member
- members: ArrayList<Member>
- ⚲ PyramidNetwork(String,ArrayList<Member>)
- generatePyramid():void
- getFilename():String
- getGreatBoss():Member
- getMembers():ArrayList<Member>

```
<<Java Class>>
Ⓖ Member
pyramid
```
- name: String
- illegalAssets: int
- sponsor: Member
- mentor: Member
- children: ArrayList<Member>
- ⚲ Member(String,int,Member)
- getMentor():Member
- setMentor(Member):void
- getName():String
- getIllegalAssets():int
- getSponsor():Member
- addChild(Member):void
- getChildren():ArrayList<Member>
- getChild(int):Member
- getLastChild():Member

1. **main package**
   a. *OperationSahure*: Contains main method and is responsible for determining which scenarios return the maximum assets.
   b. *Case*: A case represents a specific police mission on a certain pyramid scheme in which they will arrest a certain number of its members. It consists of a network log file containing the pyramid structure data and the number of maximum arrests which can be made.
   c. *Scenario*: Since the police only has the resources to arrest a certain number of members, a scenario represents a specific selection of members that they will arrest. It consists of a list of arrested members and the total assets recovered by the police.
2. **iomanager package**
   a. *InputReader*: This class is responsible for managing all input files. It should read a file called input.txt which contains a list of cases to be evaluated.
   b. *OutputWriter*: This class is responsible for writing results to output files.
3. **pyramid package**
   a. *PyramidNetwork*: Represents the hierarchical structure of the members in a pyramid scheme.
   b. *Member*: Represents a member of the pyramid scheme.