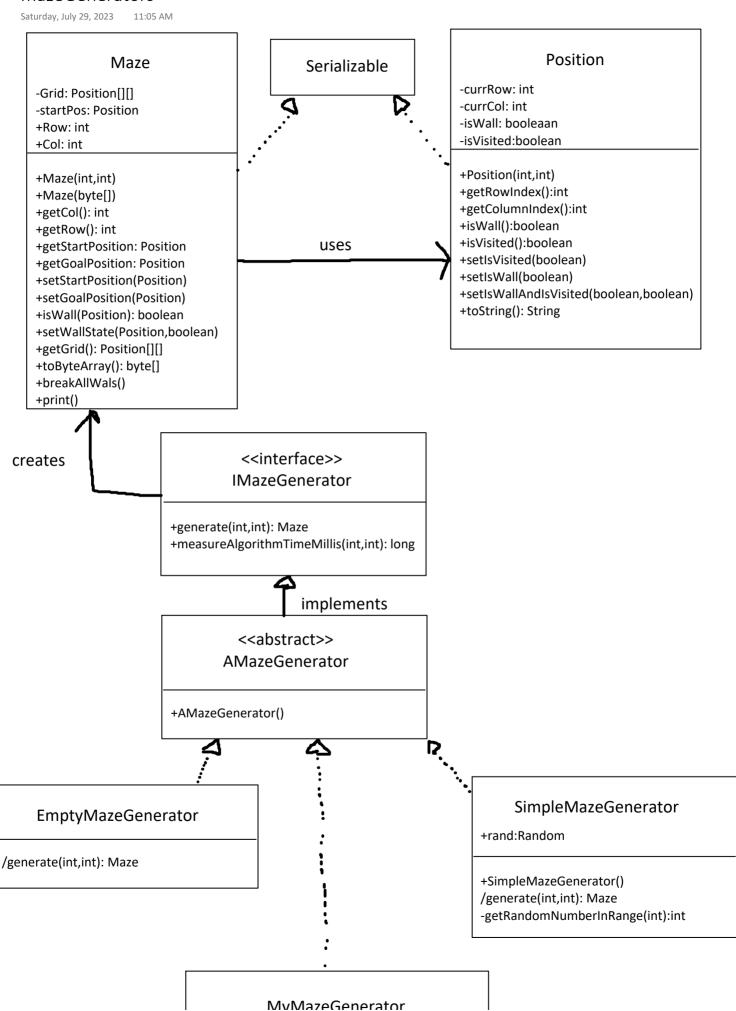
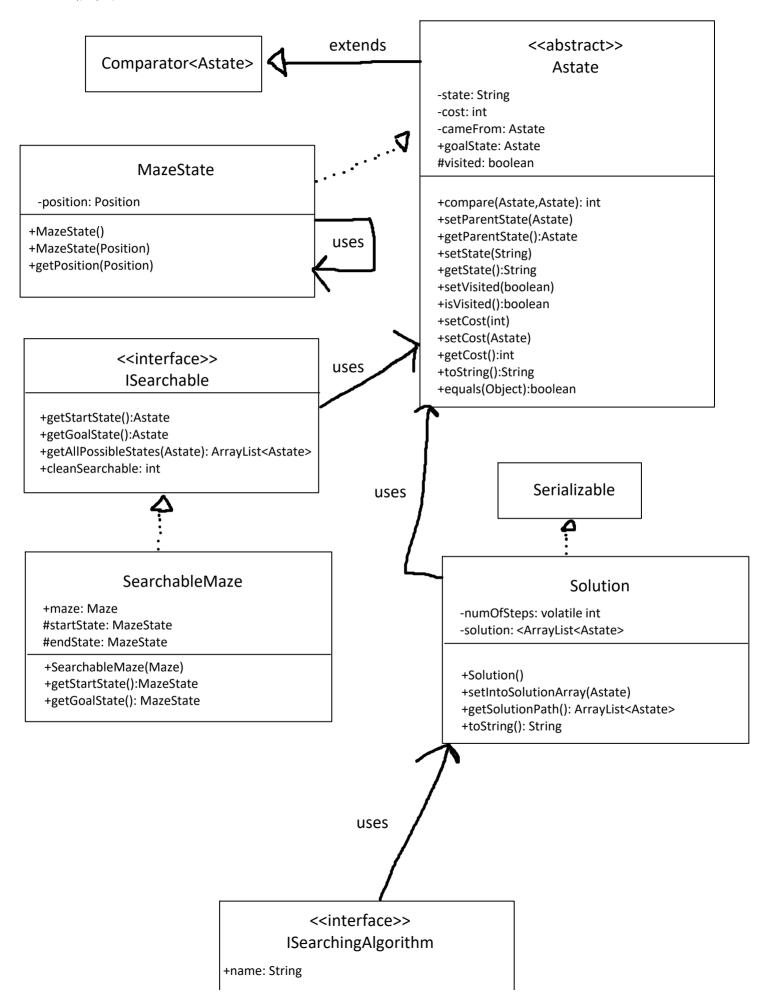
mazeGenerators

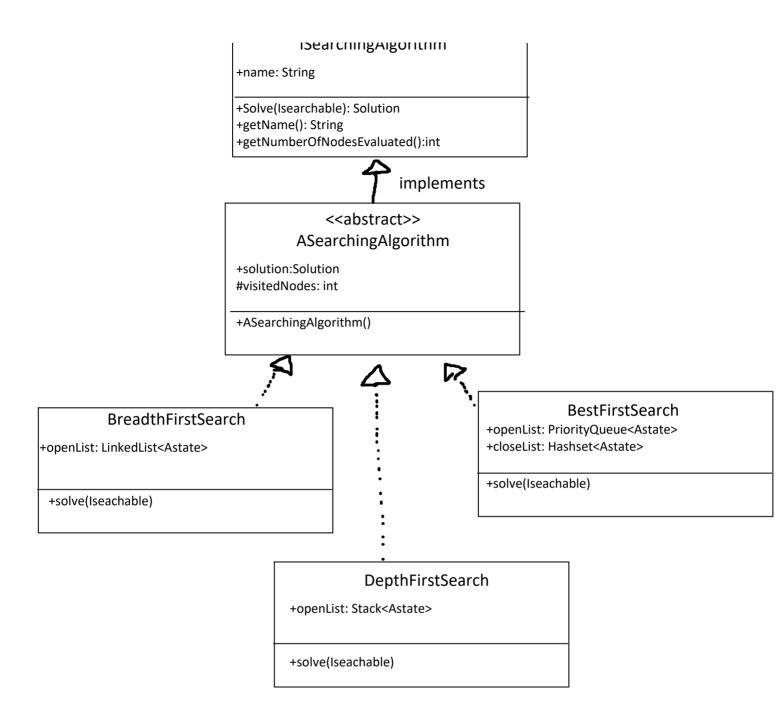


My Maze Generator

/generate(int,int): Maze -createDefaultMaze(Maze)

- -createPassage(Position,Poistion,Position[][])
- -getRandomoOption(Position,Position[][],int,int)





Client

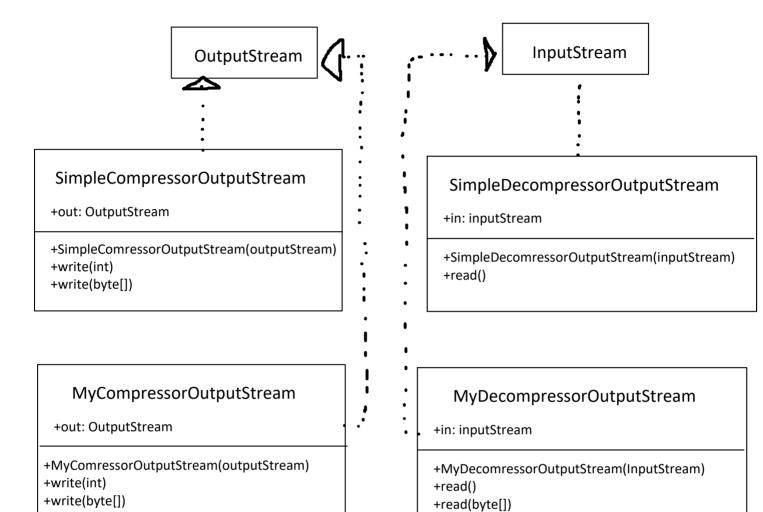
Saturday, July 29, 2023 12:42 PM

<<Interface>>
IClientStrategy

+clientStrategy(InputStream,OutputStream)

Client

- -serverip: InetAddress
- -severport: int
- -strategy: IClientStrategy
- +Client(InetAddress,int,IclientStrategy)
- +communicateWithServer()
- +clientStrategy(InputStream,OutputStream)



Configurations

-instance: Configurations-searchAlgorithm: String-numberOfThreads: String-generatorAlgorithm: String

-props: Properties

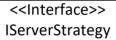
-Configurations()

+getInstance(): Configurations

+getSearchAlgorithm(): ISearchingAlgorithm

+getNumberOfThreads(): int

+getGeneratorAlgorithm(): IMazeGenerator



+applyStrategy(InputStream,OutputStream

:

Server Strategy Solve Search Problem

-numOfSolution: static AtomicInteger

-findIfSolutionExist(Maze): Solution
-writeMazeSolutionToFile(Maze,Solution)

Runnable



Server

-listeningIntervalsMS: int

-port: int

-stop: volatile boolean -strategy: IServerStrategy -threadpool: ExecutorService -ServerThread: Thread

-RunStrategy(Anonymous Class)

+Server(int,int,IServerStrategy)

+start() +stop() +run()

ServerStrategyGenerateMaze

-getCompressMaze(Maze): byte[]

Packages Connections

Saturday, July 29, 2023 1:31 PM

