Description des méthodes: (pseudo-code)

* classe_LOC :
- input = filestream d'un file .java
1) for each Line :
    if Line.isempty : continue
    else : counter ++
2) return: counter

* methode_LOC :
- input = ~~filestream~~ outputstream d'une méthode
1) for each Line:
    if Line.isEmpty: continue
    else: counter++
2) return: counter

* classe_CLOC:
- input = filestream d'un file .java
1) for each line
    for each character: *
        if (character_i == '/'):
            i++
            if (character_i == '/'): single-line comment
                count ++;
                break
            else if (character_i == '*'): mutti-lines comment
                multi-line = TRUE
                count++;
                break

*
if(multiLine):
    if(char_i == '*' &&
        char_i+1 == '*'):
            multiLine = false
            count++
            break
    else if (!eof):
        count++
        break
    else: continue

    endfor
endfor ~~continue~~

return: count

* methode_CLOC :
- input = outputstream d'une methode.
VOIR * classe_CLOC.

* classe_DC :
- input = file stream d'un file .java.
1) classe_CLOC = classe_CLOC(input).
2) classe_LOC = classe_LOC(input).
3) classe_DC = 1)/2)
return : classe_DC.

* methode_DC :
VOIR * classe_DC.

* get Classes :
- input = chemin absolu du dossier contenant les files.
1) new fileManager()
2) fileManager.getClassesFileStreams
3) return : classesFileStreams Array.

* get Methodes
1) for each classFileStream in self.classes :
2) for each line in the fileStreams
if (line.contains("(){" || "()" + EOL) :
if (line[i-1][last2] == "*/") :
while (line[i-1].trim()[0][1] != "/*"
i--
i++
startIndex = i
bracketStack = new Stack()
while (bracketStack.isEmpty && i <= startIndex) :
self.methods[j][k] = line[i]
for each character in line[i] :

```
if (char == '{') bracketStack.push(1)
else if (char == '}') bracketStack.pop()
i++
```

* JavaAnalyzer : (constructeur)
- input = chemin absolu du dossier.
1) self.chemin = ~~arg~~ input.
2) self.classes = getClasses(self.chemin)
3) self.methods = getMethods
// TODO: get classes' and methods' names.

* produceCSV :
1) produce classes.csv avec attributs de la classe + méthodes d'analyse.
2) pareil avec les méthodes