

RAPPORT ANALYSE DU PROJET

INTRODUCTION

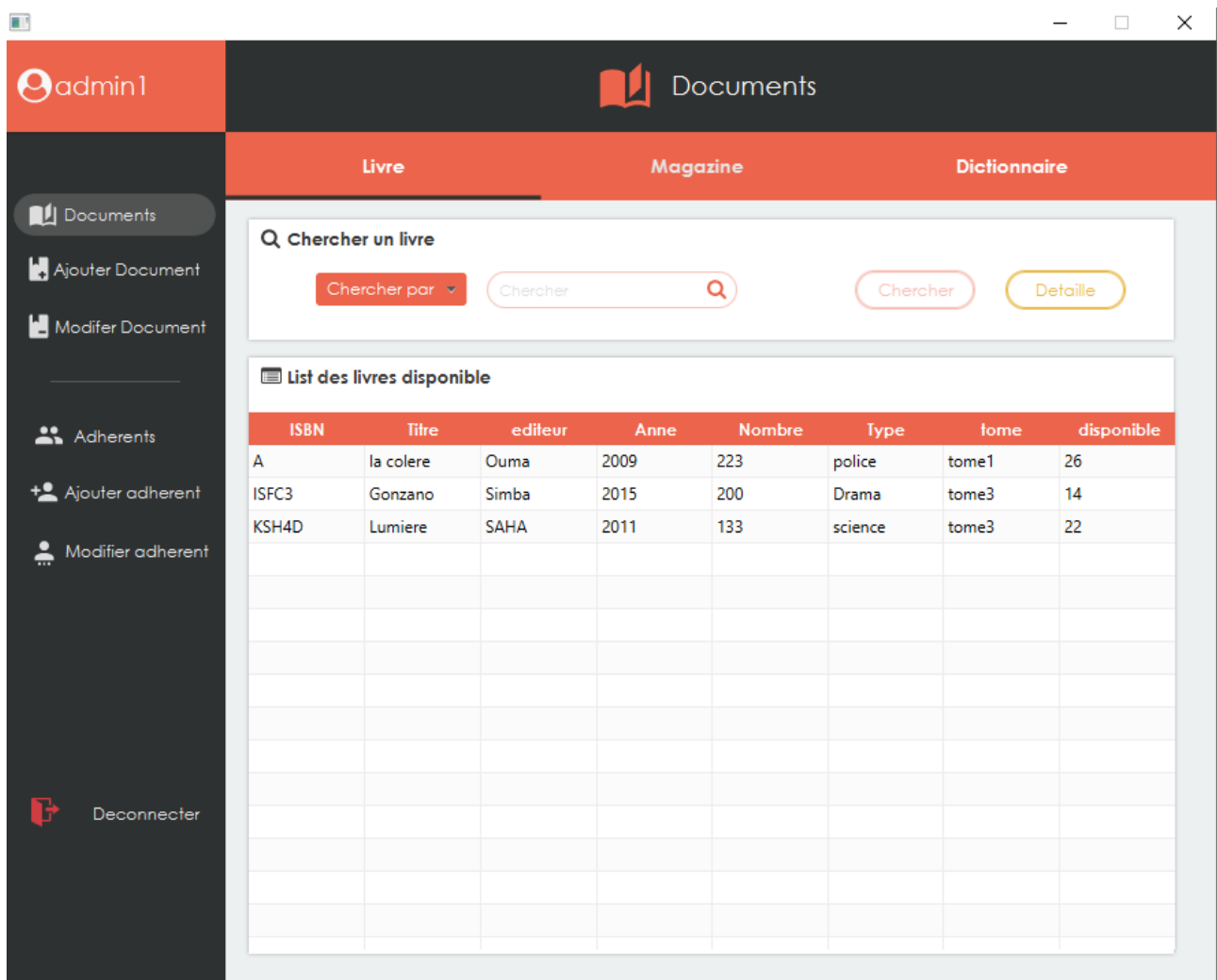
Personne ne peut douter aux jours d'aujourd'hui que l'informatique est une révolution fondamentale et innovante qui touche considérablement la vie humaine. En effet, loin d'être un phénomène effervescent, ou une tendance passagère, l'informatique est exploitée dans tous les aspects de la vie. Aucun domaine n'est resté à l'abri de cette politique qui est utilisé par des entreprises pour satisfaire les besoins de ces consommateurs, à cet effet de nombreux produits logiciels respectant des besoins fonctionnels de l'utilisateur et misent en production s'avèrent coûteuse pour ce dernier au niveau de la maintenance. Le problème qui y découle est celui de savoir comment analyser et retrouver tout éventuel problème dans une application?

Dans l'optique d'apporter une solution à cette interrogation, nous évoluerons suivant cette approche : de prime abord, nous présenterons le projet choisi, ensuite nous ferons l'analyse proprement dite du projet.

Présentation du projet choisi

En vue de mettre en pratique les notions d'évaluation de produit logiciel nous avons utilisé une **Application de Gestion d'une bibliothèque** libre et accessible sur **github**

interface administrateur de l'application



ANALYSE DU PROJET

Pour analyser notre projet, nous nous sommes référé aux outils d'analyse de projet Open source et avons utilisé **cloc**, **sloccount**, **O3SMeasures** chacun ayant des particularités.

Cloc

Permet de compter les différences de lignes physiques de code source dans le fichiers donnés (fichiers zip, tar, et tout projet git) et / ou récursivement sous le répertoires donnés. Il est entièrement écrit en Perl, en utilisant uniquement des modules de la distribution standard. cloc compte les lignes vierges, les lignes de commentaires et les lignes physiques du code source dans de nombreux langages de programmation et produits le résultat sous plusieurs format.

nous l'avons installer sur notre distribution linux en utilisant la commande **sudo apt-get install -y cloc**. Nous avons accès a son manuelle d'utilisation

avec **man cloc**

Analyse et sauvegarde du rapport dans le fichier **Analyse_cloc**

```
(base) kouam@kouam-HP-ProBook-4530s:~/Bureau/Fac M1/Gelo$ cloc --report-file=Analyse_cloc Tpe-GL-Group4-M1-20-21/
83 text files.
56 unique files.
45 files ignored.
Wrote Analyse_cloc
(base) kouam@kouam-HP-ProBook-4530s:~/Bureau/Fac M1/Gelo$ ls
Analyse_cloc      cours-02-gl-M1.pptx  cours-04-gl-M1.pptx  FILE  Tpe-GL-Group4-M1-20-21
cours-01-gl-M1.pptx  cours-03-gl-M1.pptx  cours-05-gl-M1.pptx  New
(base) kouam@kouam-HP-ProBook-4530s:~/Bureau/Fac M1/Gelo$ cat Analyse_cloc
github.com/AlDanial/cloc v 1.74  T=0.35 s (111.4 files/s, 24083.1 lines/s)
-----
Language          files      blank      comment      code
-----
Java               25         2053         313         5797
CSS                14          34           5          231
-----
SUM:              39         2087         318         6028
-----
(base) kouam@kouam-HP-ProBook-4530s:~/Bureau/Fac M1/Gelo$
```

sloccount

Nous permet d'estimer même l'effort, le temps et le coût de développement d'un produit logiciel. Sans options, il utilisera le modèle de base **COCOMO**, qui effectue ces estimations uniquement à partir du nombre de lignes de code.

Nous l'avons installer sur notre distribution linux en utilisant la commande **sudo apt-get install -y sloccount**. Nous avons accès a son manuelle d'utilisation avec **man sloccount**

Analyse et sauvegarde du rapport dans le fichier **Analyse_sloccount**

```

Fichier  Édition  Affichage  Rechercher  Terminal  Aide
(base) kouam@kouam-HP-ProBook-4530s:~/Bureau/Fac M1/Gelo$ sloccount Tpe-GL-Group4-M1-20-21/ > Analyse_sloccount.txt
(base) kouam@kouam-HP-ProBook-4530s:~/Bureau/Fac M1/Gelo$ cat Analyse_sloccount.txt
Have a non-directory at the top, so creating directory top_dir
Adding /home/kouam/Bureau/Fac M1/Gelo/Tpe-GL-Group4-M1-20-21//Analyse_cloc to top_dir
Adding /home/kouam/Bureau/Fac M1/Gelo/Tpe-GL-Group4-M1-20-21//Fichier de discussion.txt to top_dir
Creating filelist for bin
Adding /home/kouam/Bureau/Fac M1/Gelo/Tpe-GL-Group4-M1-20-21//build.fxbuild to top_dir
Adding /home/kouam/Bureau/Fac M1/Gelo/Tpe-GL-Group4-M1-20-21//rapport_Kouam1.docx to top_dir
Creating filelist for src_Adherents
Creating filelist for src_Documents
Creating filelist for src_application
Have a non-directory at the top, so creating directory src_top_dir
Adding /home/kouam/Bureau/Fac M1/Gelo/Tpe-GL-Group4-M1-20-21//src/image.jpg to src_top_dir
Categorizing files.
Finding a working MD5 command....
Found a working MD5 command.
Computing results.

SLOC    Directory      SLOC-by-Language (Sorted)
5423    src_application  java=5423
231     src_Documents   java=231
143     src_Adherents   java=143
0       bin             (none)
0       src_top_dir     (none)
0       top_dir         (none)

Totals grouped by language (dominant language first):
java:      5797 (100.00%)

Total Physical Source Lines of Code (SLOC)                = 5,797
Development Effort Estimate, Person-Years (Person-Months) = 1.27 (15.19)
  (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                        = 0.59 (7.03)
  (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule) = 2.16
Total Estimated Cost to Develop                           = $ 171,004
  (average salary = $56,286/year, overhead = 2.40).
SLOccount, Copyright (C) 2001-2004 David A. Wheeler
SLOccount is Open Source Software/Free Software, licensed under the GNU GPL.
SLOccount comes with ABSOLUTELY NO WARRANTY, and you are welcome to
redistribute it under certain conditions as specified by the GNU GPL license;
see the documentation for details.
Please credit this data as "generated using David A. Wheeler's 'SLOccount'."
(base) kouam@kouam-HP-ProBook-4530s:~/Bureau/Fac M1/Gelo$

```

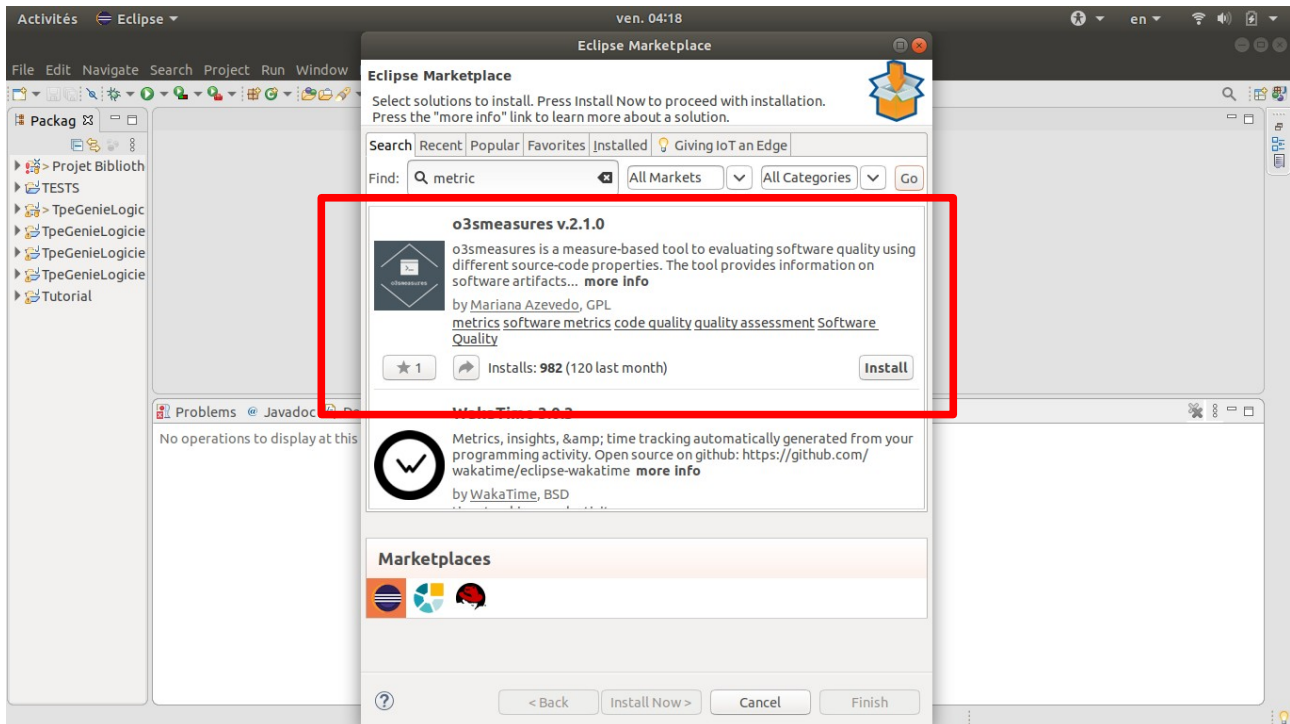
O3SMeasures

c'est d'outil principal d'analyse que nous avons utilisé. Il est basé sur des mesures pour évaluer la qualité d'un logiciel à l'aide de différentes propriétés de code source. L'outil fournit des informations sur les artefacts logiciels en utilisant les 15 mesures les plus citées dans les études empiriques pour étudier la qualité interne des logiciels au cours de la dernière décennie + 11 mesures de soutien (telles que le nombre de classes, le nombre de références de méthodes, le nombres d'attributs etc.).

Installation et utilisation

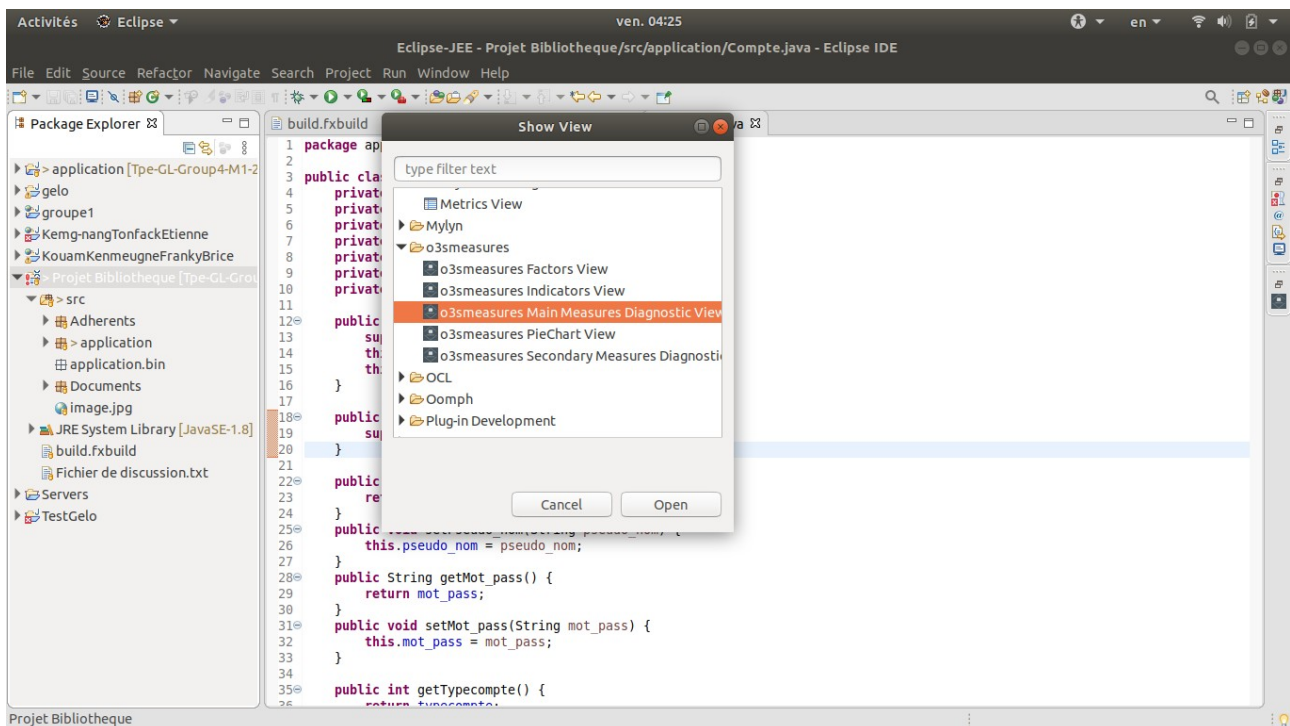
O3SMeasures s'installe comme plugin dans Eclipse

installation

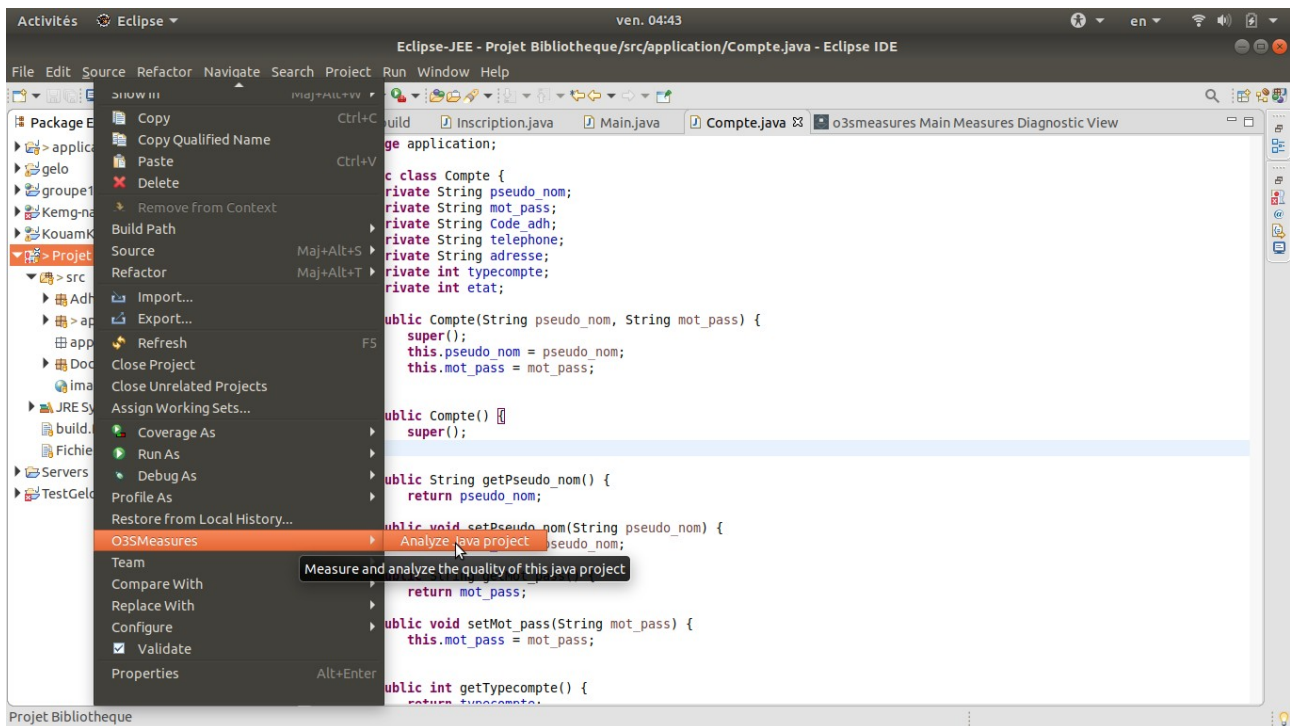


Utilisation

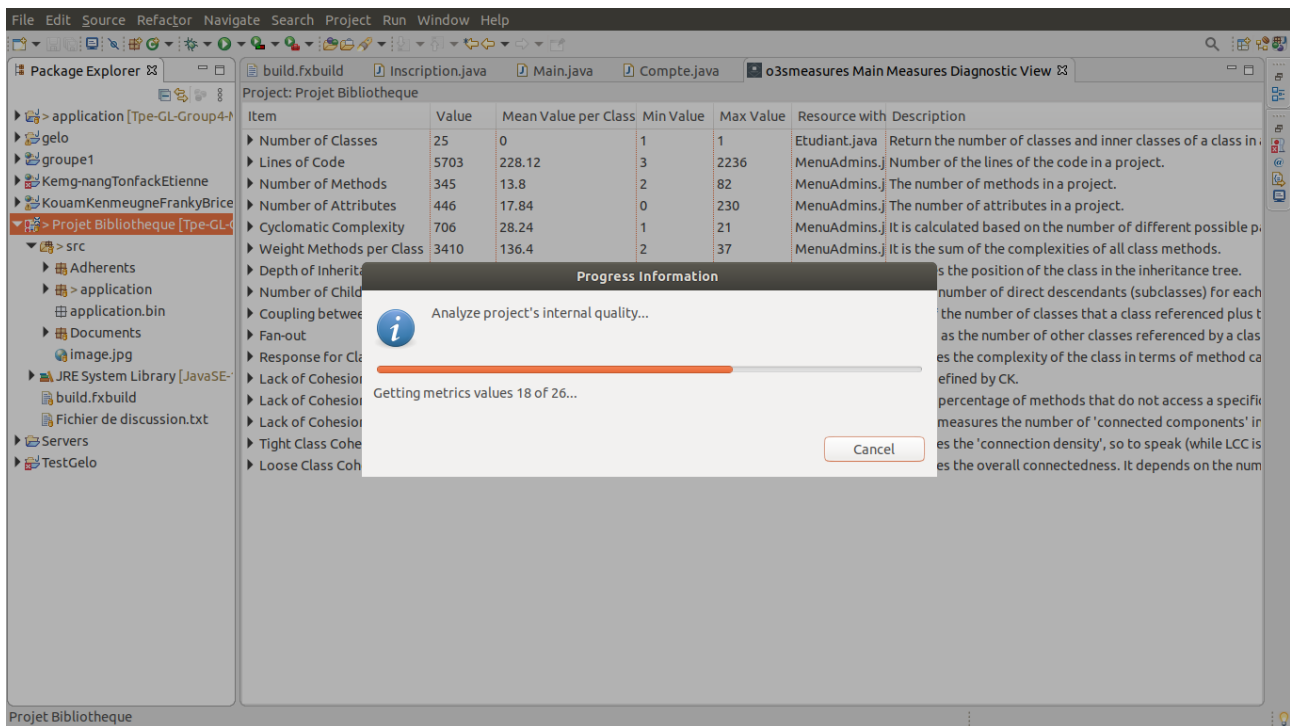
- création de la vue



- lancement de l'analyse



- Analyse en cours



- Résultat analyse

Item	Value	Mean Value per Class	Min Value	Max Value	Resource with	Description
Number of Classes	25	0	1	1	Etudiant.java	Return the number of classes and inner classes of a class in
Lines of Code	5703	228.12	3	2236	MenuAdmins.j	Number of the lines of the code in a project.
Number of Methods	345	13.8	2	82	MenuAdmins.j	The number of methods in a project.
Adherents	34	1.36	6	16	Adherent.java	
Documents	54	2.16	6	27	Document.java	
application	257	10.28	2	82	MenuAdmins.j	
Number of Attributes	446	17.84	0	230	MenuAdmins.j	The number of attributes in a project.
Cyclomatic Complexity	706	28.24	1	21	MenuAdmins.j	It is calculated based on the number of different possible p
Adherents	34	1.36	1	1		
Documents	54	2.16	1	1		
application	618	24.72	1	21	MenuAdmins.j	
Weight Methods per Class	3410	136.4	2	37	MenuAdmins.j	It is the sum of the complexities of all class methods.
Depth of Inheritance Tree	30	1.2	0	2	Etudiant.java	Provides the position of the class in the inheritance tree.
Number of Children	6	0.24	0	3	Adherent.java	It is the number of direct descendants (subclasses) for each
Coupling between Objects	91	3.64	1	30	DocumentBD.j	Total of the number of classes that a class referenced plus t
Fan-out	25	1	1	1	Etudiant.java	Defined as the number of other classes referenced by a clas
Response for Class	735	29.4	2	276	MenuAdmins.j	Measures the complexity of the class in terms of method ca
Lack of Cohesion of Metho	237	9.48	0	72	MenuAdmins.j	LCOM defined by CK.
Lack of Cohesion of Metho	11.146	0.446	0	0.978	MenuAdmins.j	It is the percentage of methods that do not access a specif
Lack of Cohesion of Metho	445	17.8	0	230	MenuAdmins.j	LCOM4 measures the number of 'connected components' in
Tight Class Cohesion	1.692	0.068	0	0.215	Inscription.jav	Measures the 'connection density', so to speak (while LCC is
Loose Class Cohesion	1.691	0.068	0	0.215	Inscription.jav	Measures the overall connectedness. It depends on the num
Adherents	0.469	0.019	0.067	0.134	Etudiant.java	
Documents	0.363	0.015	0.023	0.143	Magazine.java	
application	0.859	0.034	0	0.215	Inscription.jav	

- Exportation des résultats

Item	Value	Mean Value per Class	Min Value	Max Value	Resource with	Description
Number of Classes	25	0	1	1	Etudiant.java	Return the number of classes and inner classes of a class in
Adherents	4	0	1	1	Etudiant.java	
Documents	4	0	1	1	Etudiant.java	
application	17	0	1	1	Etudiant.java	
Lines of Code	5703	228.12	3	2236	MenuAdmins.j	Number of the lines of the code in a project.
Number of Methods	345	13.8	2	82	MenuAdmins.j	The number of methods in a project.
Number of Attributes	446	17.84	0	230	MenuAdmins.j	The number of attributes in a project.
Cyclomatic Complexity	706	28.24	1	21	MenuAdmins.j	It is calculated based on the number of different possible p
Weight Methods per Class	3410	136.4	2	37	MenuAdmins.j	It is the sum of the complexities of all class methods.
Depth of Inheritance Tree	30	1.2	0	2	Etudiant.java	Provides the position of the class in the inheritance tree.
Number of Children	6	0.24	0	3	Adherent.java	It is the number of direct descendants (subclasses) for each
Coupling between Objects	91	3.64	1	30	DocumentBD.j	Total of the number of classes that a class referenced plus t
Adherents	4	0.16	1	1	Etudiant.java	
Documents	4	0.16	1	1	Etudiant.java	
application	83	3.32	1	30	DocumentBD.j	
Fan-out	25	1	1	1	Etudiant.java	Defined as the number of other classes referenced by a clas
Response for Class	735	29.4	2	276	MenuAdmins.j	Measures the complexity of the class in terms of method ca
Lack of Cohesion of Metho	237	9.48	0	72	MenuAdmins.j	LCOM defined by CK.
Lack of Cohesion of Metho	11.146	0.446	0	0.978	MenuAdmins.j	It is the percentage of methods that do not access a specif
Lack of Cohesion of Metho	445	17.8	0	230	MenuAdmins.j	LCOM4 measures the number of 'connected components' in
Tight Class Cohesion	1.692	0.068	0	0.215	Inscription.jav	Measures the 'connection density', so to speak (while LCC is
Loose Class Cohesion	1.691	0.068	0	0.215	Inscription.jav	Measures the overall connectedness. It depends on the num

- Contenu finale de notre répertoire

```
kouam@kouam-HP-ProBook-4530s: ~/Bureau/Fac M1/Gelo/Tpe-GL-Group4-M1-20-21
Fichier Édition Affichage Rechercher Terminal Aide
(base) kouam@kouam-HP-ProBook-4530s:~/Bureau/Fac M1/Gelo/Tpe-GL-Group4-M1-20-21$
ls
Analyse_cloc 'Fichier de discussion.txt' '\Projet Bibliotheque.xml'
bin '\Projet Bibliotheque.csv' rapport_Kouam1.docx
build.fxbuild '\Projet Bibliotheque_main.xml' src
(base) kouam@kouam-HP-ProBook-4530s:~/Bureau/Fac M1/Gelo/Tpe-GL-Group4-M1-20-21$
```

