

COCKS VALENTINO

1. Nombre de classes

Chaque fois qu'un TypeDeclaration détecte une classe, elle est ajoutée dans la liste postClasses. Le nombre de classes est donc la taille de postClasses.

2. Nombre de lignes

Le code lit chaque fichier et fait :

```
nombreDeLignes += source.split("\n").length;
```

Cela ajoute le nombre de lignes du fichier dans la variable nombreDeLignes.

3. Nombre total de méthodes

Pour chaque classe trouvée, le programme prend noeud.getMethods().length. Ces valeurs sont additionnées pour obtenir le total.

4. Nombre total de packages

Le nom du package est ajouté dans l'ensemble nombreDePackages. Comme c'est un Set, il n'y a pas de doublons. La taille de nombreDePackages donne le nombre de packages.

5. Moyenne de méthodes par classe

La variable totalMethodes est divisée par le nombre de classes (postClasses.size()).

6. Moyenne de lignes par méthode

Pour chaque méthode, on calcule :

```
int ligneDebut = cu.getLineNumber(methode.getStartPosition());
```

```
int ligneFin = cu.getLineNumber(methode.getStartPosition() + methode.getLength());
```

Puis on fait ligneFin - ligneDebut + 1. Ces valeurs sont stockées dans postMethodes. Ensuite, la moyenne est totalLignesDeMethodes / nombreTotalDeMethodes.

7. Moyenne d'attributs par classe

Chaque classe donne noeud.getFields().length. Ces valeurs sont additionnées et divisées par le nombre de classes.

8. Top 10% des classes avec le plus de méthodes

postClasses est triée par getNombreDeMethodes(). On garde les 10% premières (au moins 1).

9. Top 10% des classes avec le plus d'attributs

Même logique que pour les méthodes, mais avec getNombreAttributs().

10. Classes dans les deux catégories

On met les noms des classes du top méthodes dans un Set. Puis on parcourt le top attributs pour voir lesquelles apparaissent aussi dans ce Set.

11. Classes avec plus de X méthodes

On parcourt postClasses et on affiche celles où getNombreDeMethodes() > X.

12. Top 10% des méthodes les plus longues

postMethodes est triée selon getNombreDeLignes(). Les 10% plus grandes sont affichées.

13. Méthode avec le plus de paramètres

On parcourt postMethodes. On compare chaque getNombreDeParametres(). La méthode qui a le plus de paramètres est gardée.