Université De Montpellier
Faculté Des Sciences

Niveau : Master 2

Module : Évolution et restructuration des logiciels

HAI913I

Rapport de TP N°1 Partie 2 : Analyse statique et dynamique

Réalisé par :

TOUIL Idriss

2022/2023

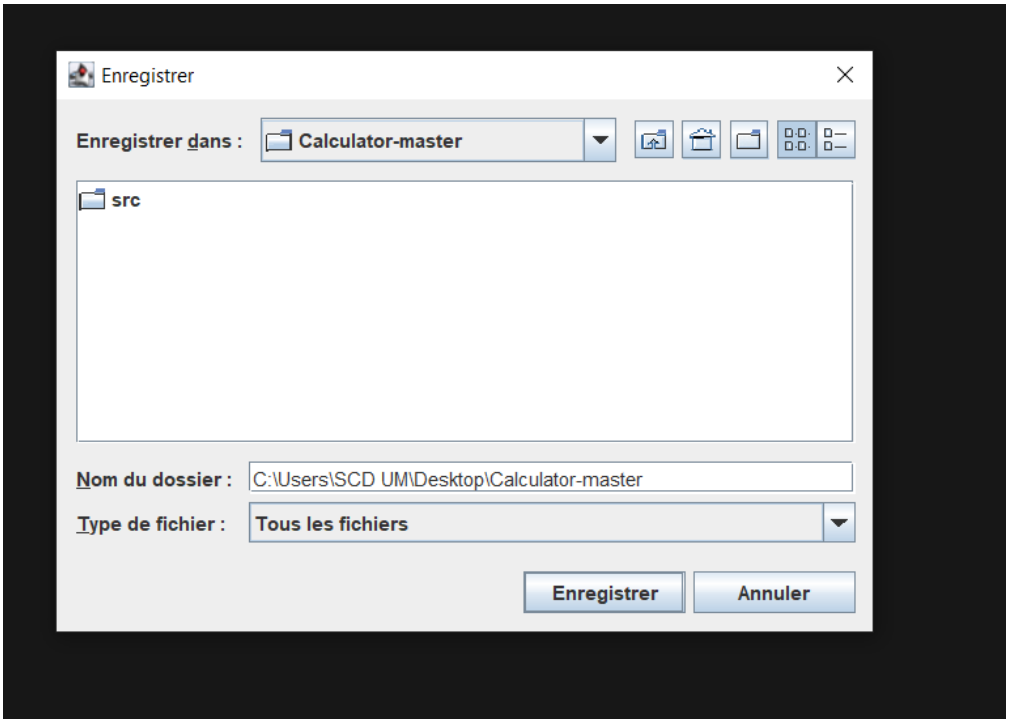
Lien du dépôt : https://github.com/idrisstouil/tp1_hai913i

Exercice1 : Calcul statistique pour une application OO

Pour lancer l'application après la récupération du code il faut faire un Run as Java Application sur la classe ParserStep2

Après l'exécution de l'application on obtient le menu sur la Figure suivante qui nous permet de sélectionner le Chemin du Projet:

Dans le projet un dossier Calculator-master-for-test qui est un projet aléatoire sur le quel j'ai fait le test du programme



Apres le choix du dossier on obtient l'affichage des informations statistiques sur la console et aussi le graph d'appel.

```
Console × Problems Debug Shell Coverage
ParserStep2 [Java Application] C:\Users\SCD UM\Desktop\openlogic-openjdk-11.0.16+8-windows-x64\bin\javaw.exe  (14 oct. 2022, 23:12:09) [pid: 14092]
Chemin du projet selectionné:
C:\Users\SCD UM\Desktop\Calculator-master\src
Q1 : Nombre de classes de l'application : 2
Q2 : Nombre de lignes de code de l'application : 662
Q3 : Nombre total de méthodes de l'application : 8
Q4 : Nombre total de packages de l'application : 1
Q5 : Nombre moyen de méthodes par classe : 4
Q6 : Nombre moyen de lignes de code par méthode : 6
Q7 : Nombre moyen d'attributs par classe : 9
Q8 : Les 10% des classes qui possèdent le plus grand nombre de méthodes :
        Calculator
Q9 : Les 10% des classes qui possèdent le plus grand nombre d'attributs :
        Calculator
Q10 : Les classes qui font partie en même temps des deux catégories précédentes :
        Calculator
Q11 : Les classes qui possèdent plus de X méthodes :
*Veuillez insérer la valeur de X :
1
        CalculatorTest
        Calculator
Q12 : Les 10% des méthodes qui possèdent le plus grand nombre de lignes de code (par classe) :
        Nom de la classe : CalculatorTest
                Nom de la methode : testCalc
        Nom de la classe : Calculator
                Nom de la methode : calc
Q13 : Le nombre maximal de paramètres par rapport à toutes les méthodes de l'application est de : 5
```

Exercice 2 : Construction du graphe d'appel d'une application

Question : 2.1

Sur la console on a aussi graphe d'appel de application avec les méthodes qui appels chaque méthode et aussi las méthodes appelés par la méthode

```
Construction du graphe d'appel de m'application

========================================
Methodes en entree : []
Methode Noeud : initCombo
Methodes en sortie : []
========================================

========================================
Methodes en entree : []
Methode Noeud : initBtn
Methodes en sortie : [add, addActionListener, setFocusable, setCursor, setFont, setBounds]
========================================

========================================
Methodes en entree : [testCalc]
Methode Noeud : calc
Methodes en sortie : [parseDouble, deriveFont, pow, getFont, setFont]
========================================

========================================
Methodes en entree : []
Methode Noeud : repaintFont
Methodes en sortie : [deriveFont, getFont, setFont]
========================================

========================================
Methodes en entree : []
Methode Noeud : main
Methodes en sortie : []
========================================

========================================
Methodes en entree : []
Methode Noeud : setUp
Methodes en sortie : []
```

Question 2.2 : Question optionnelle

On peut avoir le graph d'appel sous la forme d'une image généré dans les ressources du projet (src/main/resources/graph.png) tracer avec la dépendance JGraphT