

LOG3430 -Travail pratique 1

Codage d’une application

Auteurs

Thierno Barry 1550237

Tomas Costanzo 1578677

Automne 2016

20 Septembre 2016

Dans le prochain laboratoire, nous allons developper un module qui permet la construccion d’une liste chaînée. Ensuite, nous allons developper des methodes pour executer les operations suivantes : union, intersection, difference, symmetric difference, is subset, is superset. De plus, nous allons fournir des méthodes pour la gestion de la liste chaînée. Un manuel d’utilisation sera inclut dans ce rapport pour approfondit sur le comportement de l’application. Finalement, nous allons expliquer comment nous avons concu les algoritmes.

# Manuel d’utilisation

## Comment créer une liste chaînée

Pour créer une liste chaînée il faut exécuter cette commande :

**SuiteChainee("is superset", liste1, liste2);**

Où "is superset" est l’opération qu’on désire effectuer sur la liste1 et la liste2. L’objet retourné sera de type SuiteChainnee. Les opérations supportées sont les suivantes :

* union
* intersection
* difference
* symmetric
* difference
* is subset
* is superset

## Comment obtenir les resultats

Une fois que l’objet SuiteChainee est créé, le résultat est accessible par l’instruction suivante :

**test\_suite.get();**

Cette méthode retourne un objet de type Chaine. Cet objet pourra être imprime sur la console avec la méthode print();

# Manuel de conception

Diagramme de classes :



IChaine est une interface, elle possède donc une liste de toutes les fonctions qui seront implémentées dans la classe Chaine. La classe suiteChainee est celle avec laquelle nous allons interagir. Elle possède comme attribut une liste de type Chaine.

../../../../Downloads/Untitled%20Diagram2.png

La classe Utils possède une seule méthode, cette méthode prends un ArrayList et enlevé tous les valeurs qui sont dupliquées, ensuite il retourne la nouvelle liste.

# Annexes