

Rapport de Laboratoire n°1

Manuel d'utilisation

Ce premier laboratoire nous a permis de créer un module permettant la construction d'une suite chaînée. Cette suite doit respecter différents critères de taille, de méthode de calcul et de valeurs de départ. Cette chaîne est ensuite enregistrée dans un fichier de type texte, en compagnie des principales informations sur sa construction.

Voici les informations permettant la construction de la chaîne :

- String chemin : chemin vers le nom du fichier où est enregistrée la suite
- String op : opérateur à utiliser (« addition », « soustraction », « multiplication », « division », ces opérateurs sont définis dans la classe « Operateurs »)
- Int val1 : première valeur de la chaîne
- Int val2 : deuxième valeur de la chaîne
- Int taille : taille de la chaîne (doit être inférieur ou égal à 10)
- Boolean étatVide : variable pour exiger que la liste soit vide

La classe « SuiteChaine » permet de construire la chaîne et de lui appliquer différentes fonctions d'ajout, de modification ou de suppression d'éléments. Cette classe est utilisée dans la classe « Main » qui permet de commander les différentes actions de l'application.

Par défaut, la classe « Main » contient uniquement la création de la « SuiteChaine ». D'autres fonctions sont disponibles mais leur exécution est commentée. A l'utilisateur de les décommenter s'il souhaite les utiliser.

La « SuiteChaine » est composée d'éléments créés à l'aide de la classe « Element ». Ils sont composés d'une valeur et d'un pointeur vers l'élément suivant. Un élément est ajouté à la suite courante à l'aide de la fonction « add() ».

Voici la liste des méthodes disponibles pour interagir avec une suite chaînée créée :

- public void display() // affiche le contenu de la chaîne
- public static String saveToFile(String path, Element firstElement, String op, int index, int size) // enregistre le contenu de la chaîne dans un fichier
- public void add(Element newElement) // ajouter un élément à la fin de la liste
- public void removeAt(int position) // supprime l'élément à la position donnée
- public void removeItem(Element element) // supprime l'élément
- public void setAt(Element element, int position) // modifie un élément à la position donnée
- public Element getAt(int position) // retourne l'élément situé à la position donnée
- public int getSize() // retourne la taille de la liste
- public void reset() // remet la chaîne à zéro
- public boolean isValid() throws Exception // vérifie que les paramètres d'entrées sont valides pour la construction de la chaîne
- public boolean isEmpty() // vérifie si la liste est vide

Manuel de Conception

Dans le rapport de conception ci-présent, nous allons expliquer les choix qui ont été fait lors du développement de cette application.

- Pour l'ensemble des nombres et des opérations composants notre application, nous avons choisi le type entier *int* plutôt qu'un type *float* ou *double*. Ce choix permet de faciliter la construction des opérateurs qui ont été codés (plutôt que d'utiliser le +, -, * et /). La définition de ces opérateurs se trouve dans la classe Opérateurs.
- La classe Main permet de commander l'application.
- L'interface ISuiteChaine qui définit l'ensemble des méthodes de gestion d'une Suite Chainée.
- La classe SuiteChaine permet la construction d'un objet SuiteChaine composée des informations nécessaires, passés en paramètres lors de l'appel du constructeur : chemin du fichier, les deux valeurs initiales, l'opérateur choisi, la taille de la suite ainsi qu'un paramètre permettant de connaître l'état du fichier, vide par défaut.
- La classe Elements sert à la construction de la SuiteChaine. Un élément est composé d'une valeur ainsi qu'un lien vers l'élément suivant, nul par défaut. (pour le dernier élément de la liste). La classe dispose de setter et getter pour gérer ces paramètres. A cela s'ajoute une fonction « add » pour ajouter un élément à la fin d'une liste.