Projet 2019 ASD2

E.Epoy,T.Bersoux, Gr. 481D

Table des matières

1 Introduction

Afin de se familiariser avec l'utilisation des structures de données telles que les files, listes, et les tables associatives, ainsi que les notions de calcul de coût temporel, ce mini projet propose d'implémenter un langage très simple permettant de déclarer, afficher et incrémenter des variables. Ces variables sont placées dans des blocs. Un bloc ne peut pas modifier les variables des autres blocs, mais peut y accéder et les incrémenter. A la fermeture d'un bloc, les variables sont listées par ordre chronologique de déclaration.

2 Travail

2.1 SDA Bloc

Un bloc est une structure contenant des variables sous forme de chaînes, associées à leur valeur respectives. Un bloc possède les fonctions suivantes :

- declarer(chaîne c, entier v) : retourne entier : Permet d'ajouter dans le bloc variable c et sa valeur v. Elle retourne 0 dans le cas nominal, et 1 si la variable existe déjà dans le bloc.
- est_dans(chaîne c) : retourne booléen : retourne vrai si la variable c est dans le bloc.
- lecture(chaîne c) : : renvoie entier renvoie la valeur numérique associée à la variable c.
  Pré-condition : c est présente dans le bloc.
- modifier(chaîne c,entier v) : modifie en v la valeur de la variable c.
  Pré-condition : c est présente dans le bloc.
- listeVariables() : liste les variables du bloc par ordre chronologique d'ajout.

2.2 SDC Bloc

On met de côté (jusqu'à question 5) la fonction listeVariables().

L'implémentation de Bloc utilise une table associative. Les clés sont les noms d'une variable, et la valeur associée est la valeur de cette variable.

Cela permet d'avoir un coût temporel en $\Theta(1)$ en moyenne (Constant amorti), pour la fonction de lecture, puisque lecture(c) se contente d'utiliser la fonction find(c) de la table associative, fonction qui a un coût moyen constant. Les autres fonctions utilisent aussi insert() et end(), qui ont un coût constant en moyenne et constant (respectivement).

2.3 SDA StrucBloc (Structure de blocs)

Une structure de bloc est un ensemble de blocs, définis les uns dans les autres. Deux variables de deux blocs différents peuvent avoir le même nom sans avoir la même valeur. Cette structure possède les fonctions suivantes :

- afficher(chaîne c) : affiche la valeur associée à la variable c. Affiche un message d'erreur si c est non trouvée.
- ajouterBloc() : ajoute un nouveau bloc dans la structure de blocs.
- retirerBloc() : retire le bloc ouvert en dernier de la structure de blocs.
- declarer(chaîne c, entier v) : retourne entier : appelle la fonction "declarer" du bloc ouvert en dernier. Retourne 0 si la déclaration s'est bien passée, 1 sinon.
- incrementer(chaîne c) : retourne entier : incrémente la valeur associée à la variable c. Retourne 0 si l'incrémentation s'est bien passée, 1 sinon.
- listeVariables() : appelle la fonction "listeVariables()" du dernier bloc ajouté.

2.4 SDC StrucBloc

L'implémentation de StrucBloc utilise une liste de blocs. Les ajouts et suppressions se font en tête : le bloc ouvert en dernier est donc en tête, cela permet de le manipuler directement avec la fonction list.front().