

NFA035 – Bibliothèques et patterns : Entrée/Sorties 2

S. Rosmorduc

May 6, 2016

Exercice 1

Question 1

On considère un langage de programmation utilisé pour gérer une calculatrice. Il comporte les instructions suivantes:

- **ADD** *valeur* : ajoute une valeur, qui est a priori un entier ;
- **MULT** *valeur* : multiplie par une valeur, qui est a priori un entier ;
- **CLEAR** efface le calcul courant (remet le résultat à 0);
- **PRINT** affiche le résultat du calcul.

Les instructions opèrent sur le résultat actuel du calcul, qui commence à 0.

Ainsi, la séquence :

```
CLEAR ADD 3 ADD 5
MULT 2 PRINT
```

va afficher le résultat 16.

Écrire une méthode java qui prendra comme argument un `Reader` et qui sera capable d'exécuter un programme écrit dans ce petit langage.

La structure de l'algorithme de lecture sera la suivante:

```
tant que tok.hasNextToken() :
    si c'est_une_instruction_ADD:
        tok.next() // on saute le ADD
        // on lit le nombre (ou on lève une erreur)
        // on termine en avançant
    sinon_si_c'est_une_instruction_MULT :
        l'analyser_et_la_traiter
    .....
    sinon
        lever_une_exception_pour_signaler_un_token
        inattendu.
    fin_si
```

Question 2

On ajoute une nouvelle commandes au langage:

- **SET VAR** : donne à la variable `VAR` la valeur actuelle du calcul ;

Ensuite, les commandes `ADD` et `MULT` sont modifiées: elles peuvent prendre comme argument un nombre, mais aussi un nom de variable (auquel cas, la valeur ajoutée ou multipliée sera celle de la variable).

Pour stocker les valeurs des variables, on suggère une `HashMap`.