TD 3
Evalusation d'une expression algébrique postfixée

TAD
Semestre 2

Une pile Liste particulière dont les elements sont gérés selon la politique LIFO (Last In - First Out)
Exemple une pile de livre

1 Spéficication algorithmique du Type Abstrait de Données Pile[T]

```
1  -- construire une pile vide p
2  procedure creerPileVide (sortie p <Pile[T]);
3
4  -- retourne VRAI si la pile p est vide
5  fonction estVide (entree p <Pile[T]>)
6    retourne <Booleen>;
7
8  -- renvoie le dernier element de p
9  -- Necessite Pile non vide
10 fonction dernier (entree p <Pile[T]>)
11   retourne <T>
12   declenche pileVide;
13
14 -- Ajoute e au debut de la pile p
15 -- Necessite pile non pleine
16 procedure empiler (maj p <Pile[T]>, entree e <T>)
17   declenche pilePleine;
18
19 -- depile une pile p
20 -- Necessite pile non vide
21 procedure depiler(maj p <Pile[T]>)
22   declenche pileVide;
```

2 Utilisation du Type Abstrait de Données Pile[T]

Exemple $ab+$.

- empiler la valeur de a (a)
- empiler la valeur de b (b)
- dépiler a et b (+)
- empiler le résultat de $a+b$ (+)

Exemple $de*ab+/c-$.

- empiler la valeur de d (d)
- empiler la valeur de e (e)
- dépiler d et e (*)

- empiler le résultat de $d \times e$ (*)
- empiler la valeur de a (a)
- empiler la valeur de b (b)
- dépiler pour obtenir a et b (+)
- empiler la valeur de $a + b$ (/)
- dépiler pour obtenir de * et ab+ (/)
- empiler le résultat $(d \times e)/(a + b)$
- Empiler la valeur de c
- dépiler les trois opérandes de -
- empiler la valeur de $(d \times e)/(a + b) - c$

```
-- calcul la valeur d'une expression
lire le premier caractere;
tantque le caractere /= '.' faire
  si caractere est une operande alors
    empiler la valeur de l'operande;
  sinon
    depiler pour obtenir le deuxieme operande;
    depiler pour obtenir le premier operande;
    empiler le resultat de l'operateur sur les deux operandes;
  fin si;
  lire le caractere suivant;
fin tantque;
```

Listing 1 – Algorithme général

```
fonction operation(entree o1 <Entier>,
           entree c  <Entier>,
           entree o2 <Entier>)
            retourne <Entier>;

programme calculerValExpression
glossaire
  c <caractere>;
  o1 <Entier>;
  o2<Entier>;
  p <Pile[Entier]>;
debut
  creerPile(p);
  lire(c);
  tantque c /= '.' faire
    si operande(c) alors
      empiler(p,valeur(c));
    sinon
      o2 <- dernier(p);
      depiler(p);
      o1 <- dernier(p);
      depiler(p);
      empiler(p, operation(o1, c, o2));
    fin si;
    lire(c);
  fin tantque;
  ecrire(dernier(p));
  depiler(p);
fin
```

```
34 fonction operation(entree o1 <Entier>, entree c <Entier>, entree o2 <Entier
       >)
35   retourne <Entier>
36 debut
37   si c = '+' alors
38     retourner(o1 + o2);
39   fin si;
40   si c = '-' alors
41     retourner(o1 - o2);
42   fin si;
43   si c = '*' alors
44     retourner(o1*o2);
45   fin si;
46   si c = '/' alors
47     retourner(o1/o2);
48   fin si;
49 fin
```

Listing 2 – Programme