

Structures de base [bs] Résumé de cours

Université de Haute Alsace

Unisciel 

algotprog 

Version 12 mai 2018

Table des matières

1 alg - Résumé de cours	1
1.1 Le langage	1
1.2 Variables, types et valeurs	1
1.3 Déclarations	2
1.4 Structure générale	2
1.5 Interactions avec l'extérieur	3
1.6 Expressions algébriques	3
1.7 Affectation interne	4

1 alg - Résumé de cours

1.1 Le langage

((alg)) Identifiant

Séquence de lettres (A...Z, a...z) avec ou sans accents, de chiffres (0...9) ou du caractère souligné (_). Il doit commencer par une lettre ou un souligné.

((alg)) La casse

Le langage **n'est pas sensible** à la casse.
Ceci signifie que `cout`, `Cout` et `COU` réfèrent un même mot.

1.2 Variables, types et valeurs



Variable

Élément informatique qu'un algorithme peut manipuler.
Décrite par :

- Un **identifiant** unique qui la désigne.

- Un **type** qui définit de quel « genre » est l'information associée.
- Une **valeur** qui doit respecter le type.

((alg)) Types intégrés

- **Entier** : Pour les entiers relatifs \mathbb{Z}
- **Réel** : Pour les nombres réels (approchés) \mathbb{R}
- **Booléen** : Le domaine \mathbb{B} des booléens (vrai / faux)
- **Caractère** : Le domaine \mathbb{A} des caractères (alphanumériques et ponctuations)
- **Chaîne** : Le domaine \mathbb{T} des textes (suite de caractères)

((alg)) Littéraux

- **Entier** : Suite de chiffres éventuellement préfixé par un signe (+ ou -).
- **Réel** : S'écrit en notation décimale ou en notation scientifique.
- **Booléen** : Identifie le **Vrai** et le **Faux**.
- **Caractère** : Se place entre quotes (').
- **Chaîne** : Se place entre guillemets (").

1.3 Déclarations

((alg)) Déclaration de variables

```
Variable nomVar : TypeVar  
Variable nomVar1, nomVar2, ... : TypeVar
```

((alg)) Définition de constante

```
Constante nomConst <- expression
```

1.4 Structure générale

((alg)) Commentaire orienté ligne

```
... # rend le reste de la ligne non-exécutable
```

((alg)) Bloc

```
Début  
| instruction1  
| instruction2  
| ...  
Fin
```

((alg)) Structure générale

```

Algorithme nomAlgo
| déclaration_des_variables_et_constants
Début
| saisie_des_données
| instructions_utilisant_les_données_lues
| communication_des_résultats
Fin

```

1.5 Interactions avec l'extérieur**((alg)) Saisie de données**

```
Saisir(nomVar1, nomVar2, ..., nomVarN)
```

((alg)) Affichage de résultats

```

AfficherSS(expr1, expr2, ..., exprN) # SANS retour de ligne
Afficher(expr1, expr2, ..., exprN) # AVEC retour de ligne

```

1.6 Expressions algébriques**Expression, opérandes, opérateurs**

Éventuellement accompagnés de parenthèses, une **expression** est une séquence « bien formée » (au sens de la syntaxe) d'**opérandes** (valeurs littérales, variables ou expressions) et d'**opérateurs** destinée à l'évaluation.

((alg)) Opérateurs arithmétiques

Opérateur Mathématique	Signification	Équivalent Algorithmique
+	(unaire) valeur	+a
-	(unaire) opposé	-a
+	addition	a + b
-	soustraction	a - b
*	multiplication	a * b
/	division décimale	a / b
div	division entière	DivEnt(a,b)
mod	modulo (reste de la division entière)	Modulo(a,b)
^	élévation à la puissance	a ^ b

**Ordre de priorité des opérateurs arithmétiques**

Comme en mathématique :

1. Les opérateurs unaires (+, -) (priorité la plus élevée)

2. L'opérateur d'exponentiation (\wedge) (s'il existe)
3. Les opérateurs multiplicatifs ($*$, $/$, div , mod)
4. Les opérateurs additifs ($+$, $-$) (priorité la plus basse)

La règle d'associativité s'applique en cas d'ambiguïté entre opérateurs du même ordre de priorité.



Règle de promotion

Pour qu'une opération numérique binaire ($+$, $-$, $*$, $/$) puisse s'effectuer, il faut que ses deux opérandes soient du **même type** ou d'un **type compatible**. Lorsque ce n'est pas le cas, il y a **promotion** de l'opérande de type le plus faible vers le plus grand.



Fonctions mathématiques

Elles agissent sur des paramètres à valeurs réelles et donnent un résultat réel.

1.7 Affectation interne

((alg))

Affectation interne

```
nomVar <- expression
```