

Table des matières

Préambule	2
Variables locales	2
Règle de nommage	2
Déclaration	2
Déclaration du programme principal	2
Opérateurs	2
Structure de contrôle conditionnelle	3
Une alternative	3
Deux alternatives	3
Plusieurs alternatives	3
Structure de contrôle itérative	3
Tantque ... Faire	3
Faire ... Tantque	3
Répéter ... Jusqu'à	3
Boucle	4
Structure de contrôle CAS	4
Fonctions	4
Procédures	4
Fonctions d'entrées/sorties standards	5
Tableaux statiques	5
Structures	5
Interruptions	5

Préambule

- Notre pseudo langage est sensible à la casse.
- Une instruction se termine par un passage à la ligne

Variables locales

Règle de nommage

Un libellé de variable est construit comme suit :

- Il peut contenir des lettres non accentuées
- Il peut contenir des chiffres
- Il ne contient aucun autre symbole
- Il commence impérativement par une lettre minuscule
- Pour plus de clarté, quand ce libellé est formé de plusieurs mots, on commence les nouveaux mots par une lettre majuscule.

Déclaration

Variables

NomVariable1, NomVariable2, ... : TypeVariable

....

Déclaration du programme principal

Programme nomProgramme

Liste des fonctions et des procédures s'il y a lieu

Variables

....

Début

....

Fin

Opérateurs

- Affectation : := ou ←
- Comparaison : <, >, <=, >=, =, ≠
- Booléen : ET, OU, NON
- Numériques : +, -, *, / (division réelle), div (division entière), mod (modulo)
- Chaîne : & (concaténer)

Structure de contrôle conditionnelle

Une alternative

Si condition **Alors**

...

FinSi

Deux alternatives

Si condition **Alors**

...

Sinon

...

FinSi

Plusieurs alternatives

Si condition **Alors**

...

SinonSi ... Alors

...

Sinon

...

FinSi

Structure de contrôle itérative

Tantque ... Faire

Tantque condition **Faire**

....

FinTantque

Faire ... Tantque

Faire

....

Tantque condition

Répéter ... Jusqu'à

Repéter

...

Jusquaceque condition

Boucle

Pour compteur **de** debut **a** fin [**pas** valeur] **Faire**

...

Finpour

Structure de contrôle CAS

Selon expression **Faire**

Cas valeur1 :

....

Cas valeur 2 :

...

...

Défaut :

FinSelon

Fonctions

Fonction NomFonction (****** nomParametre1 : TypeParametre, ****** nomParametre2 : TypeParametre,...) : TypeSortie

Variables

...

Début

...

...

Fin

****** peut valoir :

- Rien ce qui signifie paramètre en entrée
- E ce qui signifie paramètre en entrée
- S ce qui signifie paramètre en sortie
- ES ce qui signifie paramètre en entrée et en sortie

On utilise le mot-clé **retourner** suivi d'une expression pour terminer l'exécution de la fonction et renvoyer le résultat.

Procédures

Fonction NomFonction (****** nomParametre1 : TypeParametre, ****** nomParametre2 : TypeParametre,...)

Variables

...

Début

...

Fin

** peut valoir :

- Rien ce qui signifie paramètre en entrée
- E ce qui signifie paramètre en entrée
- S ce qui signifie paramètre en sortie
- ES ce qui signifie paramètre en entrée et en sortie

Fonctions d'entrées/sorties standards

Les fonctions s'appellent lire et écrire.

Pour la fonction écrire, on utilise l'opérateur & pour concaténer.

Pour lire, on utilise la virgule comme séparateur

Tableaux statiques

Unidimensionnel : NomTableau[debut : fin] : Tableau de typeVariable. Si debut n'est pas mentionné alors debut vaut 1.

Biidimensionnel : NomTableau[debut1 : fin1, debut2 : fin2] : Tableau de typeVariable. Si debut1 ou debut2 alors ils valent 1.

On peut étendre à d'autres dimensions.

On accède à une case d'un tableau unidimensionnel : nomTableau[noCase].

On accède à une case d'un tableau bidimensionnel : nomTableau[noCase1, noCase2].

Syntaxe idem pour les autres dimensions.

Structures ou enregistrements

Enregistrement nomEnregistrement

nomChamps1 : TypeChamps

nomChamps2 : TypeChamps

...

FinEnregistrement

Interruptions

Stop pour sortir d'une structure de contrôle

Sortir pour sortir d'un programme ou d'une procédure.