

Algorithmique

I. Déclarations

1) Constantes

CONST <Nom> = <Valeur>

2) Variables

VAR <Nom> : <Type> [longueur]

Type peut-être : car (1), ch.car [n], réel, entier, booléen.

Nom peut-être Var1, Var2, Var3...

3) Structures

STRUCTURE <Nom>

<Var1> : <Type> [longueur]

<Var2> : <Type> [longueur]

FIN STRUCTURE

L'intérêt est de créer un template réutilisable pour la déclaration d'objet similaires.

II. Instructions de base

1) Affichage

AFFICHER("Texte")

La concaténation de plusieurs chaînes se fait avec ,

2) Saisie

SAISIR(Variable)

L'affectation de la saisie à la variable est automatique.

3) Affectation

<Var_dest> ← <Var_orig>

III. Structures de contrôle

1) Séquentielle

Action 1

Action 2

2) Conditionnelle + Alternative

SI <condition> ALORS

Action1

SINON

Action2

FINSI (aka FSI)

3) Itérative

TANT QUE <condition> FAIRE

Action

FTQUE

REPETER

Action

JUSQU'A <condition>

IV. Parcours d'un tableau 2D (n lignes)

...

cpt_x ← 1

TANT QUE (cpt_x ← nb_lignes) FAIRE

cpt_y ← 1

TANT QUE (cpt_y ← nb_colonnes) FAIRE

... tableau[cpt_x, cpt_y] ...

cpt_y ← cpt_y + 1

FTQUE

cpt_x ← cpt_x + 1

FTQUE

...

V. Procédure

1) Définition

Une procédure comporte dans son appel les paramètres de données (envoyés) et de résultats (retournés).
 Quand un seul paramètre résultat et renvoyé, il vaut mieux utiliser une fonction.

2) Déclaration

PROCEDURE <Nom> (d <données_1> : <Type>, ... , r <result_1> : <Type>, ...)

3) Appel

<Nom> (<données_1>, ... , <result_1>, ...)

VI. Fonction

1) Définition

Une fonction comporte plusieurs paramètres d'appel mais une seule valeur de retour.

2) Déclaration

FONCTION <Nom> (d <données_1> : <Type>, ...) : <Type>

3) Appel

<result> ← <Nom> (<données_1>, ...)

VII. Fichiers

1) Organisation séquentielle

Les enregistrements sont implantés consécutivement et comportent un identifiant.

2) Organisation séquentielle indexée

Un index permet l'accès direct à l'enregistrement voulu.

3) Organisation relative

Les enregistrements sont triés et accessibles directement par son rang : $(n^{\circ}-1) * \text{taille}$.

4) Accès séquentiel

PRENDRE(<Filename>) (aka LIRE)

METTRE(<Filename>) (aka ECRIRE)

5) Accès direct

PRENDRE(<Filename>, CLE = <Clef>) (aka LIRE)

METTRE(<Filename>, CLE = <Clef>) (aka ECRIRE)