

Fonctions et procédures

I) Notion de procédure et de fonction

Définition :

Une procédure ou une fonction est un bloc d'instruction ayant un début et une fin identifié par un nom.

II) Notion de paramètres

Cette méthode, lors de l'appel de procédure, il ne reste plus qu'à indiquer le nom des variables réellement concernées par l'échange (ces dernières sont appelées variables).

Exemple :

```
Procédure ECHANGE (Valeur1 : Nombre, Valeur2 : Nombre)
Var
    Temp : NOMBRE
Début
    Temp ← Valeur 1
    Valeur1 ← Valeur2
    Valeur2 ← Temp
Fin Procédure
```

Les paramètres sont des variables locales de communication entre le programme appelant et l'appelé. Ils permettent de préciser l'action de la procédure ou de la fonction.

Les paramètres sont assimilés à des variables locales, car ils ne sont visibles qu'à l'intérieur du bloc d'instruction.

III) Appel d'une procédure ou d'une Fonction

Une procédure ou une fonction est un bloc d'instruction nommé qui peut être appelé de n'importe quel endroit. Pour l'appeler, il suffit de spécifier au moment souhaiter son nom, cela déclenche l'exécution des instructions de la fonction ou de la procédure.

Exemple :

```
Programme Inverse Valeurs
Var
    A,B : Nombre /* Variable pour la saisie Clavier */
    Temps : Nombre /* Variable utilisée pour la permutation */
Début
    /*Saisir les deux variables Concernées */
    Afficher « Entrez deux valeurs »
    Saisir A,B

    /* Effectuer l'opération */
    Echange (A,B)
    Afficher A
    Afficher B
Fin
```

Remarque 1 : Une procédure ou une fonction peut ne pas avoir de paramètres, mais lorsque celle-ci a besoin pour son fonctionnement que le programme appelant lui transmette des informations il lui est nécessaire de déclarer dans l'entête de la fonction ou de la procédure ; le nom et le type des paramètres formels. Les paramètres effectifs à transmettre sont à placer dans l'instruction d'appel.

Remarque 2 : Lors de l'appel de la fonction ou de la procédure, les paramètres effectifs doivent être placés dans le même ordre que les paramètres formels associés.

Rappel : Il s'agit d'un passage de paramètres par position. Il faut le même nombre de paramètres effectifs que de paramètres formels.

IV) Procédures

Définition :

Une procédure est un bloc d'instruction qui ne retourne aucune valeur. Elle ne fait qu'exécuter des actions sans avoir à rendre compte au programme appelant.

1) *Déclaration de procédure*

Procédure *NomProcédure* (Paramètres Formels avec leur Types)

Var

Déclaration des variables locales

Débuts

[...]

Instructions

[...]

Fin

2) *Appel d'une procédure*

Il suffit de spécifier le nom de la procédure à appeler à l'endroit souhaité, avec éventuellement les paramètres effectifs. Si la procédure ne possède pas de paramètres on supprime les parenthèses lors de l'appel.

V) Fonctions

Définition :

Une fonction est un bloc d'instruction qui retourne obligatoirement une et une seule valeur résultat au programme appelant. Une fonction n'affiche jamais la réponse à l'écran car elle la renvoie simplement au programme appelant.

1) *Déclaration d'une Fonction*

Fonction *NomFonction* (Paramètres formels avec leur type) + type valeur retournée

Var

Déclaration des variables locales

Début

[...]

Instructions avec insertion de retour

[...]

Fin Fonction

Exercices :

Fonction recevant ...

Fonction Donne_Occurrence(La_Chaine : Tableau[20] de caractères, CarRech : Caractère) : Nombre
Var

NbOcc : NOMBRE /* Contient le nombre d'occurrence */
Indice : NOMBRE /* Variables pour parcourir la chaîne */
LG : NOMBRE /* Contient la longueur de la chaîne */

Début

LG ← 1
NbOcc ← 0
Tant que Indice ≤ LG faire
 Si LaChaine[Indice]=CarRech Alors
 NbOcc ← NbOcc + 1
 Fin Si
 Indice ← Indice + 1
Fin Tant que
Retourne NbOcc

Fin Fonction

2) *Instruction de Retour*

La syntaxe de l'instruction de retour sera : retourne expression

VI) Paramètres Formels

Les paramètres formels que l'on met en place lors de la définition d'une procédure ou d'une fonction doivent en fait être vu comme des variables locales, c'est à dire qu'ils sont visibles uniquement dans le bloc des instructions de la procédure ou de la fonction.

VII) Mécanisme de passage des paramètres

1) *Passage de paramètres par valeur*

C'est le mode de transmission par défaut, il y a copie de la valeur, des paramètres effectifs dans les variables locales issues des paramètres formels de la procédure ou de la fonction appelé. Dans ce mode, le contenu des paramètres effectifs ne peut pas être modifié par les instructions de la fonction ou de la procédure. Car nous ne travaillons pas directement avec la variable, mais sur une copie.

Procédure AjouteUn (LaValeur : NOMBRE)

Début

LaValeur ← LaValeur + 1
Afficher « Valeur dans procédure : » ; LaValeur

FinProcédure

Programme TestParamètre

Var

Val : NOMBRE

Début

Val ← 5
Afficher « Avant appel de la procédure : », Val
AjouteUn(Val)
Afficher « Après appel de la procédure : », Val

Fin

En exécutant cet algorithme nous obtenons à l'affichage :

Avant appel de la procédure : ligne 5

Après appel de la procédure : ligne 6

Ligne 7 : Début du programme principal

Ligne 8 & 9 : Déclaration d'une variable locale Val et réservation d'un emplacement mémoire de 2 octets

Ligne 11 : Affectation de la variable 5 à Val

Ligne 12 : Affiche le texte demandé

Ligne 13 : Appel de la procédure AjouteUn, on place en paramètre effectif le contenu de la variable Val, à savoir la valeur 5. Le programme poursuit son execution à partir de la Ligne 1

Ligne 1 : Création de la variable locale de nom de la valeur issue de la déclaration du paramètre formel et initialisation de cette dernière avec la valeur du paramètre effectif transmis lors de l'appel.

Ligne 3 : Ajout d'une unité à la Valeur LaValeur

Ligne 4 : Affiche le texte demandé

Ligne 5 : Les variables locales sont supprimées de la mémoires, la procédure se termine. Donc la variable locale LaValeur disparaît.

Le programme reprend son execution à la ligne 14

Ligne 14 : Affiche le texte demandé

Ligne 15 : La Variable Val est supprimée de la mémoire et la fonction prend fin.

Ce mode de transmission est le mode par défaut, il suffit d'employer le mot clef INT lors de la déclaration des paramètres formel de la procédure ou de la fonction. Cette indication se place devant la définition du type du paramètre.

Le passage par adresse

Il suffit de rajouter le mot clef IN ou OUT

Lorsque l'on met en place une procédure ou une fonction il n'est pas nécessaire d'avoir tout les paramètres formels en passage par valeur (Mode IN) ou en passage par adresse (Mode IN OUT) Il peut y avoir un mélange des deux mode de transmissions.

Exemple :
[à Recup]

