

## Chapitre2 :Procédure et fonction

### I) Procédure/Fonction :

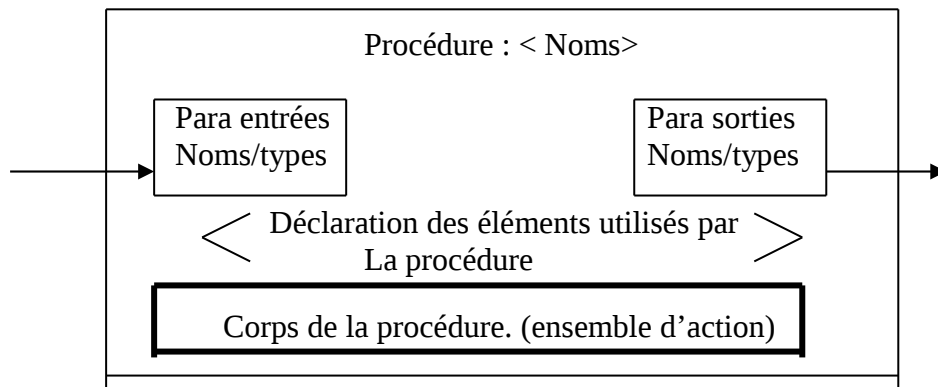
A) Un algorithme est une solution a une classe de problèmes.  
Un algorithme peut être décomposer en procédures.

B) Une procédure peut être décomposer en plusieurs procédures :

L'algorithme est donc la procédure principale.

On dira d'une procédure qui résous un classe de problème et contenant ou non d'autres procédures, qu'elle est une procédure principale.

Toute procédure possède : un nom, des entrées/sorties, des éléments locaux qu'elle utilisera, un corps qui montre ce qu'elle fait.



#### Schéma type :

Procedure Nom ( para ) ;

- *déclaration des éléments.*
- *Qu'utilise cette procédure*

Begin

- *Corps de la procédure.*
- *Ensemble d'actions, ce que fait cette procédure.*
- *Et comment elle le fait*

end ; *de la procédure*

### II) Règles d'écriture de la procédure :

Le nom d'une procédure est soumis à la règles de **construction des identifications** :

- 1) Une identification doit commencer par une lettre.
- 2) On a autant de caractère que l'on veut (alphanumérique).
- 3) Le nom choisi doit être significatif.

## A) Procédure sans paramètres :

Soit la procédure principale ( Algorithme ) qui décrit le dessin de l'arbre et de la maison. D'un point de vue de l'agencement de la procédure, on peut avoir le schéma suivant :

Procédure principale → Procédure arbre.  
→ Procédure maison.

### Schéma :

Procédure Principale :

Procédure Arbre ;

Begin

*Séquence dessin arbre ;*

end ;

Procédure maison ;

Begin

*Séquence dessin maison ;*

end ;

Begin

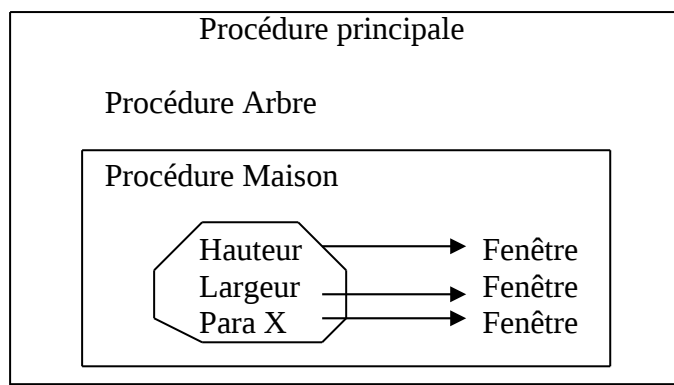
*Utilisation de la procédure arbre*  
*Utilisation de la procédure maison*

end .

### 1) Identification versus Mots réservés :

Un mot est dit réservé dans un langage, si son utilisation dans ce langage est prédéfini.

## B) Procédure avec paramètres :



### Ecriture de la procédure maison :

Procédure maison ;

Procédure fenêtre ( const Hauteur : dimension ;  
const Largeur : dimension ) ;

Begin

*La séquence de dessin de la fenêtre  
En fonction des paramètres hauteur et largeur  
Utiliser comme de valeurs constantes.*

End ;

Begin

Fenêtre ( 10,10 ) ;

Fenêtre ( 12,5 ) ;

Fenêtre (14,18 ) ;

End.

### **C) Paramètres formels Versus paramètres effectifs :**

Paramètres formels : Identification (noms) du Para.

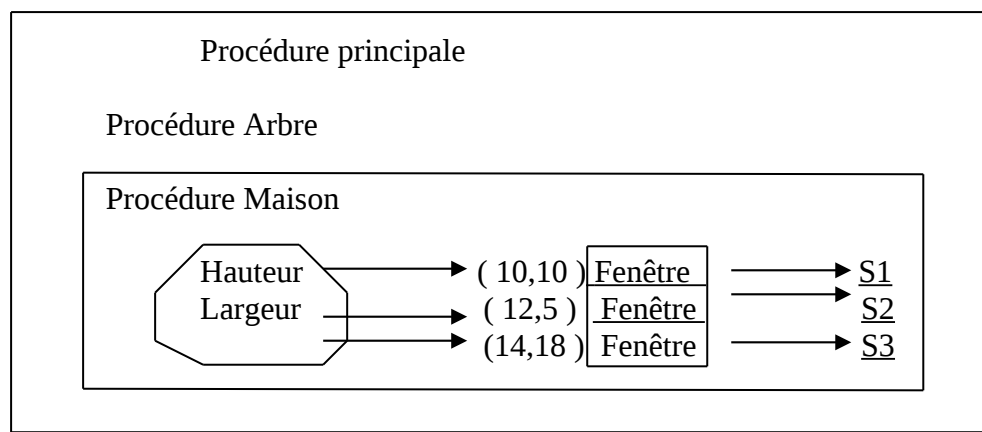
Lors de l'écriture. (Hauteur, largeur).

Paramètre effectifs : Information qu'on place à la bonne position lors de l'appel

De la procédure. Cette info se substitue au paramètre formel de la même position. (fenêtre ( 12,5).

### **D) Procédure avec paramètres en sortie :**

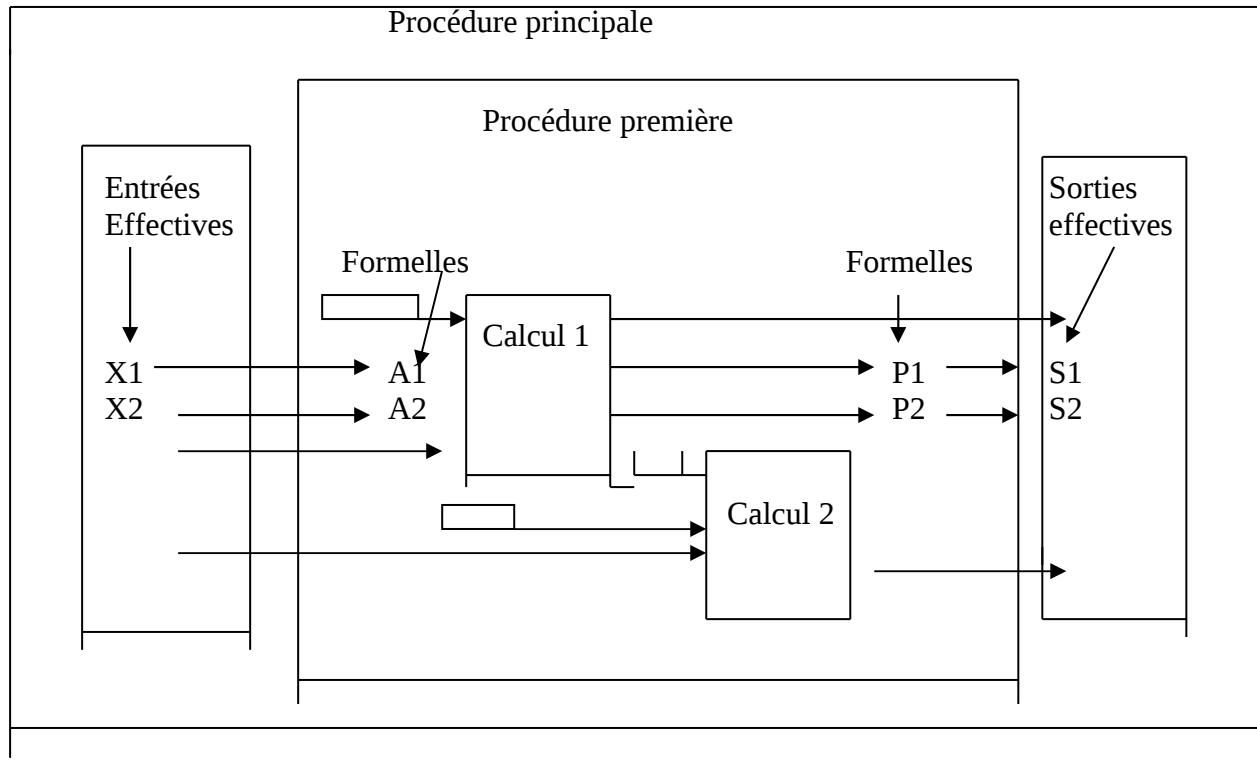
Ex : calcul de la surface de la fenêtre.



S1, S2, S3 doivent être de même nature. Ces valeurs sont réutilisables.

Procédure fenêtre ( const Hauteur : dimension ;  
const Largeur : dimension ;  
var Surface : dimension ; )

### III) Règle d'écriture de procédure :



#### Concepts :

- Les boîtes sont des procédures, on peut avoir des procédures dans une procédure.
- Il y a 2 formes de paramètres :
  - formels dans la déclaration de la procédure
  - Effectifs au moment de l'appel de la procédure.
- Chaque paramètre a un mode : entrée ou sortie
- En effectif, une entrée coïncide avec une valeur, en sortie à une variable.
- Chaque formel doit être typé, le paramètre effectif coïncide avec le typage.
- Dans une procédure, l'agencement des appels de procédures, c'est des actions, qu'elle utilise et détermine son fonctionnement et ce qu'elle calcule.

#### Notion de Types :

On appelle type de données un ensemble de valeur, un nom prédéfini, un ensemble d'opérations qui vont avec.

Exemples de types : Entier relatifs Z ,Entiers Réels R Réel, V/F B Booléen.

### 1) Forme général d'une procédure :

Procédure Nom ;

*Déclaration :*

- *Identification des constantes*
- *Types*
- *Variables*

*Enoncé des autre procédures*

Begin

*Corps de la P, avec toutes les actions, et les appels aux procédures.*

end ;

### 2) Déclaration de paramètres Formels :

- Chaque paramètre formel a un nom
- Chaque paramètre doit être typé
- A chaque paramètre ,on associe un mode.
- On peut regrouper les para en fonction de leur type et mode.

Soit la procédure qui contrôle l'impôt sur le revenu de la manière suivante, si l'impôt, déjà calculé est inférieur à 400F alors l'impôt est nul.

On a tendance a écrire cette procédure avec 2 paramètres :

Procédure Contrôle (const Impôt : réel  
var Impôt : réel) ;

Dans ce cas de figure, on a à faire à un seul para entrée / sortie.

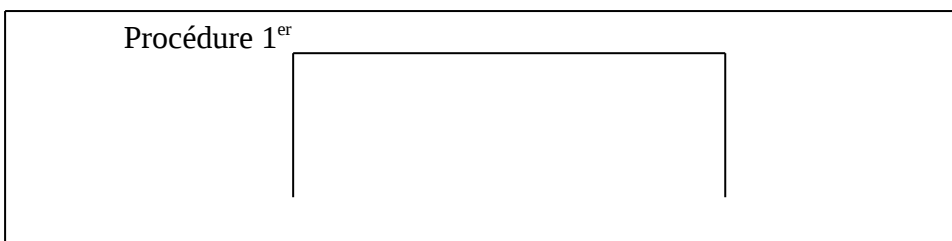
**Variable** : une variable est un récipient qui n'admet des valeur que d'un type donné, on l'utilise pour mémoriser des valeurs.

### 3) Déclaration de Variables :

- Chaque variable doit être nommée.
- On n'utilise pas deux fois le même nom que ce soit P. ou V ou E.
- Chaque variable doit être déclarée dans la procédure qui l'utilise et utilisée seulement dans cette procédure.
- Chaque variable doit être typée, pour les bonnes valeurs données.

### 4) Paramètre d'appel ( effectif ) :

L'appel d'une procédure doit avoir lieu dans le corps de la Procédure qui la contient. On peut utiliser les Paramètres formels de la grande procédure comme paramètres effectif de la petite.



		Procédure 2 <sup>nd</sup>			
Hauteur : réel	→	Haut : réel	Surf : R	→	Surface : réel
Valeur locale	→	Larg : réel	Variable : R	→	Récipient : réel

---

Schéma final :

Procédure 1<sup>er</sup> (const hauteur : réel ;  
var Surface : réel ) ;

var INTERNE : réel

Procédure 2<sup>nd</sup> ( const haut, larg :réel ;  
var Surf, Valeur : réel ) ;

Begin  
*On ne doit pas utiliser INTERNE*  
end;

Begin

Seconde ( Hauteur,10, surf, INTERNE ) ;  
Seconde ( Hauteur, 2\* HAUTEUR , surf, INTERNE ) ;

etc....

End.