

Séminaire d'algorithmique: Cours 3

Informatique Différenciée - EISTI - ING 1

Ecole Internationale des Sciences du Traitement de l'Information

Définition :

Suite d'instructions réalisant une certaine tâche, à laquelle on donne un nom pour qu'on puisse l'appeler ultérieurement

Syntaxe :

```
procedure nom(liste de paramètres formels)  
Variables locales  
Instructions  
finprocedure
```

Pourquoi

- ▶ Traitements répétitifs (longues applications) :
ex : saisie d'une réponse et son traitement
 - ▶ Lisibilité
 - ▶ Maintenance
- ▶ Programmation modulaire :
 - ▶ Modules uniques : briques
 - ▶ Application : assemblage de briques
- ▶ Vérification et tests

Comment

- ▶ Procédure principale : corps du programme

```
programme nom  
debut  
    Instructions  
fin
```

- ▶ Sousprocédure : suite d'instruction en dehors de la procédure principale
- ▶ Appel des sousprocédures explicitement en utilisant leur nom et en fixant les valeurs des paramètres formels (entre parenthèses) qui deviennent des paramètres effectifs

```
nom(liste des paramètres effectifs)
```

Exemple

```
procedure afficherMenu()  
  ecrire(" Pour accéder au service commercial ,  
        tapez 1")  
  ecrire(" Pour accéder au service technique ,  
        tapez 2")  
  ecrire(" Pour quitter , tapez 3")  
finprocedure
```

Exemple (suite)

```
procedure traiterMenu(entier numero)
si numero=1 alors
    commercial()
sinon si numero=2 alors
    technique()
sinon
    quitter()
finprocedure
```

```
programme test
variables resultat: entier
debut
    afficherMenu()
    lire(resultat)
    traiterMenu(resultat)
fin
```

Définition

Suite d'instructions nommée (comme les procédures) qui retourne une valeur à la procédure ou fonction appelante.

Syntaxe

```
fonction nom(liste de parametres): type de  
    retour  
    Variables locales  
    Instructions  
    retourner ...  
finfonction
```

Exemple

```
fonction calculCarre(nb:entier):entier
  variables tmp:entier
  tmp ← nb*nb
  retourner tmp
finfonction
```

```
programme test
variables x:entier
debut
  x ← calculCarre(5)
  ecrire(x)
fin
```


Exemples selon le type

► Numérique :

```
ent(n: reel) // partie entière  
alea(n: entier) // nombre aléatoire entre 0 et n  
sqrt(n: entier) // racine carrée de n  
sin(x: reel)  
cos(x: reel)
```

► Texte :

```
// nombre de caractères:  
longueur(c: chaîne)  
/ sous chaîne entre n1 et n2:  
extrait(c: chaîne, n1: entier, n2: entier)  
// valeur de c dans la table ascii:  
ascii(c: chaîne)  
// caractère ascii associé à n:  
caractere(n: entier)
```

Types de paramètres

- ▶ Paramètres en entrée (lecture) : E
- ▶ Paramètres en sortie (écriture) : S
- ▶ Paramètres en entrée-sortie (lecture-écriture) : ES

Exemple

```
procedure divisionEntiere(E a:entier ,E b:entier , S  
    quotient:entier , S reste: entier )  
procedure inversion(ES s:chaine)
```

Variables locales

- ▶ Déclarées au sein d'une procédure ou d'une fonction
- ▶ Visibles uniquement par cette procédure/fonction

Variables globales

- ▶ Alternative aux arguments pour communiquer entre procédures
- ▶ Visibles par tout le programme

Règles à respecter

1. Objectifs

- ▶ Architecture claire
- ▶ Modules concis et spécifiques
- ▶ Modules testables et modifiables de manière indépendante

2. Mise en oeuvre

- ▶ Limiter au maximum les variables globales
- ▶ Adapter les modules aux unités fonctionnelles de votre application
- ▶ Bien séparer les fonctions (valeur retournée) des procédures

Découpage d'un problème

Fonctions ou procédures

- ▶ Déterminer les fonctions ou procédures adéquates pour un problème donné
- ▶ Découper son algorithme pour le rendre léger, lisible et performant
- ▶ Ensemble de tâches constituant l'analyse fonctionnelle

Exemple

TP3 et cours "découpage d'un problème"