

Lycée Secondaire Raccada



Chapitre 3

Les sous programmes

4^{ème} SC

Enseignant: Sami MEJRI

2015-2016

La modularité

Mise en situation :usine de fabrication des voitures

La décomposition d'un problème en des sous problèmes indépendants et de tailles réduits est appelée la **décomposition modulaire** c'ad la décomposition d'un problème en **modules** (fonction ou procédures).

Le but de la programmation modulaire:

- Faciliter la résolution
- Éviter les redondances (répétitions)
- Réutiliser le même sous programme plusieurs fois
- La lisibilité des gros programmes

1- Les fonctions

1- Définition

Une fonction est un sous programme qui retourne une seule valeur d'un seul type simple (entier, réel, caractère, booléen ou chaîne) qui sera celui de la fonction.

2- Déclaration

Analyse et Algorithme

DEF FN Nom_Fonction (**Liste des paramètres formels**) : type du résultat

Instructions de la fonction

Nom_Fonction ← résultat

Fin NomFonction

Pascal

```
Function NomFonction ( Liste des paramètres formels ) : type du résultat ;  
{ Déclarations locales } ;  
Begin  
Instructions de la fonction ;  
Nom_Fonction := résultat ;  
End ;
```

3- Application

Ecrire un programme qui calcule la puissance de deux entiers x et y.

1. Analyse

Analyse du programme principal

Résultat = Ecrire (x, " à la puissance ", y, " = ", **Power(x,y)**)

X=donnée (" Donner x ")

Y=donnée (" Donner y ")

Analyse de la fonction POWER

DEF FN POWER (x : entier ; y : entier) : **entier**

Résultat = P

[P ← 1]

Pour i de 1 à y faire

P ← P * x

Fin pour

Power ← P

Fin Power

2. Algorithmme

Algorithmme de la fonction POWER

DEF FN POWER(x : entier ; y : entier) : entier

P ← 1

Pour i de 1 à y faire

P ← P * x

Power ← P

Fin power

Algorithmme du programme principal

o) Début Puissance

1) Ecrire (" Donner x ") Lire(x)

2) Ecrire (" Donner y ") Lire(y)

3) Ecrire(x, " puissance ", y, " = ", power(x,y))

4) Fin Puissance

3. T.D.O

Objet	Type / Nature	Rôle
P	Entier	Puissance à calculer
i	Entier	Compteur
Power	Fonction	Fonction calcul puissance
x,y	Entier	Entiers donnés

4. Traduction en Pascal

```
program powerxy;
uses wincrt;
var
  x,y:integer;
function power( x : integer ; y : integer):
integer;
var
  i: integer;
  p: integer;
begin
  p := 1;
  for i := 1 to y do
    p := p * x;
  power := p;
end;

begin
  writeln('donner x'); readln(x);
  writeln('donner y'); readln(y);
  writeln(x, ' Puissance ',y,' = ',power(x,y));
end.
```

4- Remarques

- Les paramètres formels d'une fonction sont toujours passés par valeur.
- Le type de résultat retourné par une fonction doit être simple (le type tableau n'est pas permis).
- Une fonction appelée est manipulée comme une valeur.
Exemple : $X := X + \text{Somme}(5, 6)$;
- Il est nécessaire d'affecter le résultat du calcul de la fonction au nom de la fonction, obligatoirement avant la fin du bloc.
- la partie déclaration locale ou seront déclarés les objets locaux de la fonction.

II- LES PROCEDURES

1- Définition

Les procédures sont des sous programmes qui peuvent avoir plusieurs résultats.

**Merci pour votre
attention**

Chapitre 4

Les algorithmes avancés