

Programmation Pascal/Procédures et fonctions

[< Programmation Pascal](#)

Une procédure ou une fonction est une sous-routine, en général appelée plusieurs fois. Ceci permet de regrouper un code utilisé plusieurs fois, de diviser le programme en sous-programmes.

Différences procédure/fonction

Procédures et fonctions acceptent des paramètres, seule une fonction retourne une valeur. La procédure n'en retourne pas.

Déclaration d'une procédure

La déclaration d'une procédure se fait avec le mot clé `procedure` suivi du nom de la procédure, et de la liste des paramètres entre parenthèses. Cette liste de paramètres est déclarée avec la même syntaxe que les variables.

Ensuite, les variables locales sont déclarées, puis les instructions sont comprises dans un bloc `begin` ...
`end;`

Par exemple :

```
procedure NumeroFoisDix(  
numero : integer ) ;  
var j : integer ;  
begin  
    j := numero * 10 ;  
    println('j = ', j) ;  
end ;
```

Déclaration d'une fonction

La syntaxe est similaire à celle d'une procédure, excepté que le mot clé utilisé est `function` et que le type de retour doit être déclaré.

Par exemple :

```
function NumeroPlusCinq(  
numero : integer ) :  
integer ;  
var j : integer ;  
begin  
    j := numero + 5 ;  
    NumeroPlusCinq := j ;  
end ;
```

Retourner une valeur se fait en affectant celle-ci au nom de la fonction.

Interface et implémentation dans un module

Dans un module, la partie

`implentation` contient le code des

sous-routines en utilisant la syntaxe des paragraphes précédant. La partie `interface` ne reprend que la première ligne.

Par exemple :

```
unit MonModule;  
  
interface  
  
  procedure NumeroFoisDix(  
    numero : integer ) ;  
  function NumeroPlusCinq(  
    numero : integer ) :  
    integer ;
```

implementation

```
procedure NumeroFoisDix(  
numero : integer ) ;  
var j : integer ;  
begin  
    j := numero * 10 ;  
    println("j = ", j) ;  
end ;  
  
function NumeroPlusCinq(  
numero : integer ) :  
integer ;  
var j : integer ;  
begin  
    j := numero + 5 ;  
    NumeroPlusCinq := j ;  
end ;
```

```
end
```

```
end.
```

Ordre de déclaration

La compilation en une passe implique que chaque fonction ou procédure appelée doit être définie avant. L'ordre des sous-routines est donc important.

L'exemple suivant produit une erreur de compilation :

```
procedure FaireFaire;  
begin  
    println('Procédure qui  
sous-traite l''action');
```

```
        Faire;  
end;  
  
procedure Faire;  
begin  
    println('Procédure qui  
fait l''action');  
end;
```

L'ordre correct est le suivant :

```
procedure Faire;  
begin  
    println('Procédure qui  
fait l''action');  
end;
```



```
procedure FaireFaire;  
begin  
    println('Procédure qui  
sous-traite l'action');  
    Faire;  
end;
```

Cette restriction ne s'applique pas aux sous-routines publiques d'un même module, car elles sont pré-déclarées dans la partie interface.

Il existe des cas où deux procédures s'appellent mutuellement. Aucune ne peut être située avant/après l'autre. Dans ce cas, il faut en prédéclarer une avec le mot clé `forward`.

Par exemple :

```
procedure  
GrandCalcul(n:integer);  
forward;
```

```
procedure  
PetitCalcul(n:integer);  
begin  
    if n>100 then  
GrandCalcul(n);  
    else begin  
        ...  
    endif;  
end;
```

```
procedure
```

```
GrandCalcul(n:integer);  
begin  
    if n<=100 then  
PetitCalcul(n);  
    else begin  
        ...  
    endif;  
end;
```

Récupérée de

« https://fr.wikibooks.org/w/index.php?title=Programmation_Pascal/Procédures_et_fonctions&oldid=564679 »

Dernière modification il y a 12 moi...

Le contenu est disponible sous licence CC BY-SA
3.0 sauf mention contraire.