

Procédures et Paramètres formels [ss] Résumé de cours

Université de Haute Alsace

Unisciel 

algotprog 

Version 15 mai 2018

Table des matières

1	C++ - Résumé de cours	1
1.1	Schéma d'une procédure	1
1.2	Paramètres formels	1

1 C++ - Résumé de cours

1.1 Schéma d'une procédure



Schéma d'une procédure

```
void nomSsp(  
D1 d1, ..., // les données  
R1 & r1, ..., // les résultats  
M1 & m1, ...) // les modifiés  
{  
    // Corps de la procédure:  
    // sur une donnée D : passage par valeur  
    // sur un résultat R ou un modifié M : passage par référence (&)  
}
```



Type et mode

Chaque paramètre possède **son type** et **son mode** de passage.

1.2 Paramètres formels



Paramètres Entrants/Sortants/Mixtes

Les **paramètres entrants** ou *données* :

- Ont une **valeur** à l'entrée du module.
- Et seront **consultés** à l'intérieur du module.

Les **paramètres sortants** ou *résultats* :

- Ont une **valeur indéterminée** à l'entrée du module.
- Et seront **utilisables après l'appel** du module.

Les **paramètres mixtes** ou *modifiés* :

- Ont une **valeur à l'entrée** du module.
- Et seront éventuellement **modifiés à l'intérieur** de celui-ci.



Modes de transmission des paramètres

En programmation, il en existe principalement deux :

- **Par valeur** : le contenu des paramètres effectifs ne peut pas être modifié ni altéré par les instructions du module.
- **Par référence** (appelé aussi **par variable**) : toute modification de la variable paramètre est **reportée** sur la variable de l'appel.

Par conséquent, le mode de passage implémente :

- **Par valeur** : une communication **unidirectionnelle** (de l'appelant vers l'appelé).
- **Par référence** : une communication **bidirectionnelle**.



Arguments et paramètres associés

Ils ne sont pas obligés d'avoir le même nom, mais dans le cas d'un passage :

- **Par valeur** : Ils doivent avoir des **types compatibles**.
- **Par référence** : Ils doivent **absolument** avoir le **même type**.

Enfin seul le nom des paramètres est à utiliser dans le corps du module concerné.

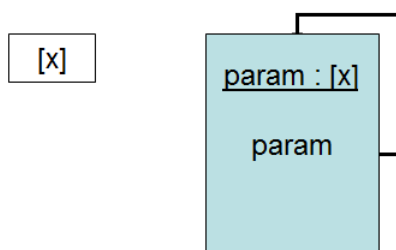


Passage par valeur v.s. référence

Prototype : ... f (T param) ;

Appel : ... f (x)

Les modifications effectuées à l'intérieur du module **ne sont pas répercutées à l'extérieur**.



Prototype : f (T& param) ;

Appel : ... f (x)

Les modifications effectuées à l'intérieur du module **sont répercutées à l'extérieur**.

