

Initiation à la Programmation

<http://www.fil.univ-lille1.fr/Licence123>

Fonctions

Les fonctions

Une **fonction** est une suite d'instructions réalisant un **calcul**, et donc, dont l'évaluation (i.e. l'exécution) retourne une valeur.

Déclarer une fonction c'est donc **abstraire** un calcul en lui donnant un **nom**.

Exemple: la fonction de l'unité cartes `SommetCoeur(1)` retourne le booléen `true` s'il y a un coeur au sommet du tas 1 et `false` sinon.

Les fonctions

Une **fonction** a

- ✓ des entrées (ses paramètres)
- ✓ une sortie (la valeur renvoyée)

Les entrées et sorties ont un type (définissant le domaine des valeurs qu'elles peuvent prendre)

Exemple: la fonction `SommetCoeur` a un paramètre de type `TasPossible` et une sortie de type `Booléen`.

Analogie: en maths, les fonctions ont également des types, par exemple la fonction f définie par $f(n,m)=n+m$ a le type

$$\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

Les fonctions

Entête

```
function nom(const paramètre: type;  
    ...;  
    const paramètre: type ) : type retour;  
  
begin  
    Instructions;  
    nom := expression;  
end;
```

La valeur renvoyée par cette fonction
est la valeur de cette expression

Type de la valeur
renvoyée par cette
fonction

Attention: *expression* doit être de type *type retour*

Les fonctions

Exemple: déterminer si la carte au sommet d'un tas est de couleur rouge

```
function rouge(const tas:TasPossibles) : Boolean;  
begin  
    rouge := SommetCoeur(tas) or SommetCarreau(tas);  
end;
```

Expression booléenne



Les fonctions

Exemple: appel de la fonction rouge

SI: tas 1: 'P+T+C+K', autres tas vides

SF: tas 2: '[C+K]', tas 3: '[P+T]', autres tas vides

Appel à la
fonction rouge

```
if rouge(1) then
  begin
    deplacerSommet(1,2);
  end
else
  begin
    deplacerSommet(1,3);
  end;
```

Spécification des fonctions

Spécifier une fonction c'est:

- ✓ choisir un **identificateur** pour la nommer
- ✓ préciser le **nombre de paramètres, leur type et les nommer**
- ✓ indiquer le **rôle joué par ces paramètres**
- ✓ indiquer les **conditions d'utilisation** (vérifiées par les paramètres)
- ✓ indiquer le **type de la valeur qu'elle renvoie**
- ✓ et indiquer la **relation entre valeur renvoyée et paramètres**

Spécification des fonctions

- ✓ choisir un **identificateur** pour la nommer
- ✓ préciser le **nombre de paramètres, leur type et les nommer**
- ✓ indiquer le **rôle joué par ces paramètres**
- ✓ indiquer les **conditions d'utilisation** (vérifiées par les paramètres)
- ✓ indiquer le **type de la valeur qu'elle renvoie**
- ✓ et indiquer la **relation entre valeur renvoyée et paramètres**

```
function rouge(const tas:TasPossibles) : Boolean;  
begin  
    rouge := SommetCoeur(tas) or SommetCarreau(tas);  
end;
```

Exercice: donner la spécification de la fonction rouge

Les fonctions

```
function rouge(const tas:TasPossibles) : Boolean;  
begin  
    rouge := SommetCoeur(tas) or SommetCarreau(tas);  
end;
```

Exercice: Spécifier et écrire une version plus générique de cette fonction. C'est-à-dire une fonction DeCouleur qui a deux paramètres de types respectivement String et TasPossibles et qui retourne true si le numéro de tas passé en 2ème paramètre a une carte sur son sommet de la couleur passée en 1er paramètre. Elle retourne false dans le cas contraire.

Procédures vs. fonctions

caractéristique	fonction	procédure
Séparation déclaration/ appel (utilisation)	oui	oui
Paramètres (constants)	oui	oui
Conditions d'utilisation	oui	oui
Modifie l'environnement	normalement non	normalement oui
retourne un résultat	oui	non

Exercices

Exercice 1: Déterminez si, pour chacune des instructions suivantes de l'unité cartes, s'il s'agit d'une fonction ou d'une procédure. Écrivez les entêtes de ces procédures et fonctions.

InitTas, DeplacerSommet

CouleurSommet, Superieur, TasNonVide,

SommetTrefle, SommetCoeur, SommetPique, SommetCarreau

Exercices

Exercice 2 : Écrivez une fonction `TasTousVides` qui renvoie un booléen indiquant si tous les tas sont vides.

Exercice 3 :

- écrivez une fonction `TasMax` qui donne le numéro du tas dont le sommet a la carte de valeur la plus élevée parmi les deux tas passés en paramètres.
- écrivez une fonction `TasMax3` analogue à la précédente pour trois tas passés en paramètres. Même chose pour quatre tas ! (paramètres ou non ?)