IN100 : Cours 7 – Les Fonctions

Sandrine Vial
sandrine.vial@uvsq.fr

Octobre 2013

Portée et Visibilité des variables

- Les variables :
  - Déclarées au sein d'un bloc
  - Utilisables uniquement dans le bloc dans lequel elles ont été déclarées
  - Durée de vie : celle du bloc

Les variables globales

- Les variables locales :
  - Déclarées au sein d'une fonction
  - Utilisables uniquement dans la fonction dans laquelle elles ont été déclarées
  - Durée de vie : celle de la fonction
- Les variables globales :
  - Déclarées en dehors de toute fonction
  - Utilisables dans n'importe quelle fonction
  - Durée de vie : celle du programme

Les variables globales

```
int a,b;
void echange()
{
    int c;
    write_int(a); write_text(" "); write_int(b); writeln();
    c = a;
    a = b;
    b = c;
    write_int(a); write_text(" "); write_int(b); writeln();
}

int main()
{

  init_graphics(600,300);
  a = 10;
  b = 20;
  write_int(a); write_text(" "); write_int(b); writeln();
  echange();
  write_int(a); write_text(" "); write_int(b); writeln();
  wait_escape();
  return 1;
}
```

Les variables globales

Avantage

On peut utiliser/modifier une variable globale n'importe où dans le code

Inconvénients

- On peut utiliser/modifier une variable globale n'importe où dans le code
- Les procédures sont moins réutilisables.

Une solution : les variables globales

Et si une variable globale porte le même nom qu'une variable locale ?

```
int a;

void modifie(int b)
{
   int a;

   b = b + 10;
   a = b * 2;
}

int main()
{
   int b;

   a = 10;
   b = 30;
   modifie(b);
   modifie(a);
}
```

Les Procédures et Les Fonctions

- Une procédure : prend des paramètres et effectue des calculs sur ces paramètres
- Une fonction : prend des paramètres, effectue des calculs sur ces paramètres et renvoie le résultat du calcul à la fonction appelante.

Les fonctions

Une fonction a des paramètres, peut avoir des effets sur les variables globales et renvoie le résultat d'un calcul sur les paramètres, les variables locales et les variables globales.

```
type Nom_fonction(paramètres)
{
....
}
```

Les fonctions : le corps

Le corps d'une fonction contient :

- Des déclarations de variables locales : utilisables uniquement dans le corps de la fonction.
- Des instructions : elles portent sur les variables locales, les variables globales et sur les paramètres.
- Une instruction de renvoi de résultat

Les fonctions : l'instruction `return`

- Permet de revnvoyer le résultat d'un calcul à la fonction appelante.
- Ce qui suit l'instruction `return` doit être du même type que dans la signature de la fonction.
- Cette instrution termine l'exécution de la fonction. Toutes les instructions suivantes ne sont pas exécutées.

Les fonctions : un exemple

```
float moyenne(int a, int b, int c)
{
   float s;
   s = (a + b + c)/3.0;
   return s;
}
int main()
{
   int a = 10;
   int b = 20;
   int c = 40;
   float d;

   d = moyenne(a,b,c);
   return 1;
}
```

Les fonctions : un exemple

Les fonctions sont évaluées :
elles peuvent être incluses dans un calcul arithmétique.

```
float moyenne(int a, int b, int c)
{
   float s;
   s = (a + b + c)/3.0;
   return s;
}
int main()
{
   int a = 10;
   int b = 20;
   int c = 40;
   float d;

   d = moyenne(a,b,c) + moyenne(45,67,89);
   return 1;
}
```

Les fonctions : un exemple

Les fonctions sont évaluées :
elles peuvent être passées en paramètres de fonctions.

```
int max(int a, int b)
{
   if (a > b)
      return a;
   else
      return b;
}
```

```
int min(int a, int b)
{
   if (a < b)
      return a;
   else
      return b;
}
```

```
int main()
{
   int a = 10;
   int b = 20;
   int c = 5;
   int  d;

   d = min(max(a,b),c);
   return 1;
}
```

Les fonctions : un exemple

Les fonctions sont évaluées :
elles peuvent s'appeler elles-mêmes : la récursivité.

```
int factorielle(int a)
{
   if (a  == 1)
      return 1;
   else
      return factorielle(a-1) * a;
}

int main()
{
   int a = 4;
   int d;

   d = factorielle(a);
   return 1;
}
```

La récursivité

- Une fonction peut se rappeler elle-même
- Préciser les cas d'arrêt de la récursivité
- Préciser le cas général

La récursivité : suite de fibonacci

- $F_1 = 1$ et $F_2 = 1$
- $F_n = F_{n-1} + F_{n-2}$

```
int fibo(int n)
{
   if (n==1 || n==2)
        return 1;
   else
        return fibo(n-1) + fibo(n-2);
 }
```

La récursivité : les tours de hanoi

- 3 piquets nommés A, B, C.
- Au début du jeu, tous les disques sont sur le piquet A.
- A la fin du jeu, tous les disques sont sur le piquet C.
- Les disques sont empilés par taille décroissante au début et à la fin.
- Règles :
  - On ne déplace qu'un seul disque à la fois.
  - Les disques doivent toujours être empilés par taille décroissante.

La récursivité : les tours de hanoi

```
void hanoi (int n, int a, int b, int c)
{
   if (n!=0)
   {
        hanoi(n-1,a,c,b);
        write_text(‘‘Déplacer un disque de’’);
            write_int(a); write_text(‘‘vers ’’);
            write_int(c); writeln();
        hanoi(n-1,b,a,c);
     }
}
```