

Chapitre 1 : Instructions algorithmiques de base

Laurent Debize

BTS SIO 1A - Algorithmique appliquée

Cinquième partie V

Respect et importance d'une charte de code

Une charte de code en deux mots

- Lisible et clair
- Rien d'officiel mais des habitudes (qui peuvent être fixées par chaque entreprise)
- Pas obligatoire mais indispensable pour la compréhension d'un programme

Le cartouche complet

C'est un commentaire qui se trouve en en-tête, il doit contenir au minimum :

- le nom du programme ou du projet
- le but
- l'auteur
- la date de création

On peut y ajouter :

- la liste des modules présents (on le verra plus tard)
- la liste des variables globales, leur rôle
- toutes les dates de révision (versions) du programme

Le cartouche complet

Pour le moment, on se contentera de ces informations :

```
/* programme nomDuProgramme
* but : but du programme en clair
* auteur : nom de l'auteur
* date : date de création du programme
*/
```

Les indentations

Que sont les indentations ?

- Ce sont des décalages qui permettent de mieux repérer chaque bloc ainsi que les imbrications entre chaque bloc
- Chaque bloc doit être décalé vers la droite
- Les décalages doivent être toujours identiques (une tabulation, 2 espaces ou 3 espaces)

Qu'est-ce qu'un bloc ?

- Le contenu du programme principal
- le contenu d'un 'alors'
- d'un 'sinon'
- d'un 'tantque'
- d'un 'repeter'
- d'un 'pour'...

Les indentations

Exemple

```
...
tantque condition1
  si condition2 alors
    repeter
      ...
      ...
    jusqu'à condition3
  finsi
fintantque
...
```

Les indentations

Il existe des variantes en programmation suivant les langages et les IDE, par exemple pour le positionnement des accolades en C++.

Exemple 1

```
if (condition)
{
    ...
}
else
{
    ...
}
```

Exemple 2

```
if (condition) {
    ...
} else{
    ...
}
```

Mais on reste libre d'utiliser celle que l'on préfère.

Les commentaires

Premier commentaire : le cartouche de début de programme.

Dans le reste du programme, il est possible, et très conseillé, d'ajouter des commentaires à chaque fois que cela est nécessaire.

Prenez l'habitude d'écrire la ligne de commentaire AVANT même d'écrire le bloc de code correspondant. Il y a 3 avantages à cette pratique :

- cela vous aide à réfléchir au contenu du code
- vous n'aurez pas à revenir sur votre code par la suite pour ajouter les commentaires
- vous aurez plus de facilités à relire et comprendre votre code en cas de besoin

Les noms des variables

Il est bien de respecter certaines règles de nommage :

- les noms doivent être **explicites et compréhensibles**
- on peut repérer le **type** de variable dans le nom (en utilisant les 3 premières lettres du nom)

Les 3 premières lettres peuvent être en minuscule, suivi du nom explicite commençant par une majuscule.

Exemple :

```
intResultat // variable de type entier qui contiendra un résultat  
strBinaire // variable de type chaîne qui contiendra un binaire  
strRep // variable de type chaîne qui contiendra une réponse
```

Remarques

- Eviter les accents, cédilles...
- Utiliser des copier/coller dès que l'on veut taper le nom de la variable évite bien des soucis...

Les noms des variables

Pour des petites applications ou des applications ne manipulant pas d'objets, les noms peuvent rester simples et juste explicites.