

1

Les variables

L'écriture d'un programme est une opération complexe qui requiert de nombreuses étapes. Le plus important est de comprendre l'objectif final et de le respecter. Pour cela, il est souvent préférable de décomposer le traitement souhaité en une succession d'opérations plus petites et plus simples. Un algorithme est constitué de la suite de ces opérations élémentaires. Elles devront être décrites avec précision dans un ordre cohérent. Pour pouvoir représenter cette suite d'opérations, nous allons introduire un langage algorithmique et une présentation précise.

Structure d'un algorithme

Les algorithmes ont pour vocation de nous faire réfléchir, mais pas de s'exécuter sur un ordinateur : pour cela, il sera nécessaire de traduire l'algorithme dans un langage de programmation. L'algorithme décrit sur papier un traitement : il est nécessaire d'en simuler le déroulement.

Définition

Algorithme

Un algorithme est une suite d'opérations élémentaires permettant d'obtenir le résultat final déterminé à un problème.

Propriété d'un algorithme

Un algorithme, dans des conditions d'exécution similaires (avec des données identiques) fournit toujours le même résultat.

Notre langage structure un algorithme en deux parties.

- La première ligne indique le nom de l'algorithme.
- La deuxième partie, le traitement, située entre les mots clés `Debut` et `Fin`, contient le bloc d'instructions.

Définition

Bloc d'instructions

Un bloc d'instructions est une partie de traitement d'un algorithme, constituée d'opérations élémentaires situées entre `Debut` et `Fin` ou entre accolades.

La structure d'un algorithme est la suivante :

```
Algorithme nom-de-l'algorithme // partie en-tête
Debut                               // partie traitement
    bloc d'instructions;
Fin
```

Chaque ligne comporte une seule instruction. L'exécution de l'algorithme correspond à la réalisation de toutes les instructions, ligne après ligne, de la première à la dernière, dans cet ordre.

Définition

Commentaires

Les commentaires sont des explications textuelles inscrites dans l'algorithme par le programmeur à la suite des deux caractères `//`. Ils ne sont pas exécutés : ils sont invisibles au moment de l'exécution de l'algorithme.

Ces commentaires seront utiles aux programmeurs qui veulent comprendre ou modifier l'algorithme. La complexité des algorithmes impose de les commenter judicieusement : ni trop, ni trop peu, toujours de manière utile.

Écrivons l'algorithme permettant d'afficher "bonjour tout le monde". Pour cela, nous avons besoin d'une fonction `ecrire("phrase")` permettant d'écrire dans le résultat la phrase écrite entre parenthèses. Soit l'algorithme suivant :

```
Algorithme algo-bonjour
Debut
    écrire("bonjour tout le monde");
Fin
```

Pour résoudre un problème, nous imaginons souvent plusieurs solutions. Dans notre exemple, l'algorithme suivant conviendra également, sachant que les lignes du traitement seront toutes exécutées de la première à la dernière, dans cet ordre.

```
Algorithme algo-bonjour2
Debut
    écrire("bonjour");
    écrire("tout le monde");
Fin
```

Mais comme l'exécution de ces deux algorithmes fournit un résultat identique, il est souvent préférable de choisir le plus simple (ici, le premier).

Les données

Déclaration et utilisation des variables

La plupart des problèmes nécessitent le traitement de valeurs : certaines sont données dans l'énoncé, d'autres sont le résultat des calculs issus de l'exécution de l'algorithme. Une troisième catégorie de valeurs intermédiaires nous servira pour calculer le résultat à partir des données : nous les introduisons dans le chapitre suivant.

Les valeurs, pour pouvoir être manipulées, sont stockées dans des variables.

Définition

Variable

Une variable désigne un emplacement mémoire qui permet de stocker une valeur. Une variable est définie par :

- un nom unique qui la désigne ;
- un type de définition unique ;
- une valeur attribuée et modifiée au cours du déroulement de l'algorithme.

Dans tous les cas, les variables utilisées au cours de l'exécution de l'algorithme sont déclarées immédiatement après le nom de l'algorithme. Il suffit d'indiquer le nom de la variable suivi de son type, séparés par deux points « : ».

La syntaxe

La structure d'un algorithme (déclarant une variable nommée `indice` et de type entier) est alors la suivante :

```
Algorithme nom-de-l'algorithme // partie en-tête
variables: indice: entier // partie des déclarations des variables

Debut // partie traitement
    bloc d'instructions;
Fin
```

La partie supplémentaire, placée nécessairement avant le bloc `Debut – Fin`, décrit les variables à déclarer pour arriver au résultat. Ici, une variable nommée `indice` de type entier a été déclarée, et pourra donc être initialisée et utilisée dans le bloc d'instructions.

Le nom – le type – la valeur

Définition

Nom d'une variable – identifiant d'une variable

Le nom d'une variable permet de l'identifier de manière unique au cours de l'algorithme.