

Algorithmique Structure de Données (simplifié)

lundi 2 septembre 2013

11:04

A.S.D.

Algorithmique Structure de Données

Un algorithme est la description de la logique d'un besoin de gestion adapté au fonctionnement d'un ordinateur.

Cette suite se traduit par une suite d'instructions élémentaires :

Exemple : *Exemple d'algorithme*

Pour fonctionner un algorithme doit contenir uniquement des instructions compréhensibles par celui qui va l'exécuter.

L'écriture algorithmique est un travail de programmation à visé universelle.

Un algorithme ne dépend pas du langage auquel il est implanté, ni de la machine qui exécutera le programme.

Un algorithme traduit dans un langage lisible par l'ordinateur donne un programme qui par la suite devra être exécuter pour effectuer un traitement en 3 phases :

- Préparation du traitement
- Le traitement
- L'édition des résultats

Pour rédiger un algorithme il y a plusieurs représentations, un organigramme (logigramme), pseudo-code, ...

Le pseudo-code est donc construit à partir d'instruction de base :

Exemple : Instruction 1

Instruction 2

Et le traitement est entouré de "début" et "fin".

I) *Exemple d'algorithme*

Algorithme EleveAuCarre

Variable : UnNombre, SonCarre : entier

Début

| Afficher ("Saisir un nombre à élever au carré")

| Saisir (UnNombre)

| SonCarre ← UnNombre * UnNombre

| Afficher ("est", SonCarre)

Fin

II) *Les variables*

Un programme à besoin en permanence de stocker les valeurs (saisie par l'utilisateur, des résultats de calcul) pour cela il utilise des variables et ces dernières peuvent être de différents types.

Pour déclarer une variable, on utilise le mot variable ou le mot déclarer, ce mot est suivit de l'identificateur de variable et de son type :

Variable <liste d'identificateurs> : type

Ou

Déclarer <liste d'identificateurs> : type

Lorsque que l'on déclare une variable il faut automatiquement préciser le type d'information que l'on met dedans.

Il existe différents types :

- Numériques (entier / décimal)
- Alphanumérique
- Chaîne
- Booléen
- Date/heures

I) *L'affectation*

Ceci consiste à attribuer une valeur à une variable (exemple : Val ← 50).

/! Une instruction d'un instruction d'affectation ne modifie que ce qui est à gauche de la flèche. **/!**

II) *Les opérateurs*

- Les opérateurs numériques : + ; - ; / ; * ; mod () ou %
- Les opérateurs de chaîne : &
- Les opérateurs de comparaison : = ; < > ; < ; <= ; > ; >=
- Les opérateurs logiques : ET ; OU ; XOR

III) *Les entrées et les sorties*

L'instruction **SAISIR** permet de placer en mémoire les informations saisies par l'utilisateur.

L'instruction **AFFICHER** permet de visualiser les instructions en mémoire.

IV) *Les structures conditionnelles*

Exemple :

- La structure **SI** ne permet de distinguer que deux cas, nous avons souvent plus de deux cas et pour cela nous allons utiliser ce qui s'appelle des **SI** imbriqués.

Variable nombre : entier

Debut

| Ecrire (« Entrez un nombre »)

| Lire nombre

| Si nombre < 0 alors

| | Ecrire (« votre nombre est négatif »)

| | Sinon

| | Si nombre > 0 alors

| | | Ecrire (« votre nombre est positif »)

| | | Sinon

| | | Ecrire (« Votre nombre est 0 »)

| | FinSi

| FinSi

Fin