

Informatique

I - introduction : algorithmique & programmation

a Algorithmique

Étude de résolution de problèmes par la mise en oeuvre de suites d'opérations élémentaires selon un processus défini aboutissant à une solution.

Résolution en un certain nbs d'étapes d'un problème défini.

b Le programme -> le code

Un programme informatique est la traduction d'un algorithme dans un langage défini, appelé **langage de programmation**.

On obtient donc du code source.

c Quelques langages de programmation

Le Lua, le C, le Basic, le Fortran, ...

En 1970 apparaît la notion la notion d'“objet” informatique, puis en 1980/90 apparaît la **P.O.O** (programmation orienté objet) qui a révolutionné les principes de programmations.

Différents objets :

- Dim a As Integer
- Dim btnTest As Button

Déclarations de variables (objets), ils se déclarent et s'utilisent avec un “.” derrière leur nom (a.blablaba) et avec Visual Studio, tout sera objet.

Quelques langages objets : C++, PHP, Java

II Les outils de réalisation d'un algorithme

a Les variables

Ce sont des valeurs (numériques ou alphanumérique) stockées provisoirement en mémoire (RAM).

RAM (Random Access Memory) : mémoire à accès très rapide

Un variable est une partie de la mémoire. Le programme va associer le nom de la variable à l'adresse de stockage de celle-ci.

On veut stocker une valeur pour effectuer un traitement informatique
On déclare une variable dédiée en choisissant un nom (si possible explicite), par exemple “prixProd”
L’outil de programmation va associer “prixProd” à une adresse en mémoire.
L’accès à la donnée se fait ainsi par le nom de la variable, bien moins indigeste qu’une adresse en RAM : 0x8001007C.

b Types de variables/Type de données

- entier (Integer, Long)
- réel (Single, Double)
- booléen (Boolean)
- chaîne de caractère (texte) (String)

Il existe d’autres types : Date, Byte, Currency ...

c Les opérateurs

Les 4 opérateurs +, -, *, / et exposant ^, l’opérateur Mod (modulo), l’opérateur & (A = “infor” et B = “Matique” => A&B = “inforMatique”)

Et aussi les opérateurs de comparaison =, <, >, <=, >=, <>

Les opérateurs dits logiques “ET” et “OU”

d l’affectation et la comparaison

Ne pas confondre l’affectation <- : A <- 5 et la comparaison (=) : A = 5

En basic, on va utiliser =, quand on va affecter ou comparer la variable : = sert dans les deux cas.

En algorithmique, on va utiliser systématiquement la flèche <- pour affecter une valeur à une variable

Les étapes de réalisation d’un programme

a L’analyse fonctionnelle

Travail à réaliser par l’ordinateur

Information dont on dispose (données) -> traitement à effectuer -> Résultat à obtenir

b Conception de l’algo

lister les variables dont on va avoir besoin (pour stocker) les données de départ, les résultats, ...). Cette liste n’est pas définitive, elle peut (c’est souvent le cas) évoluer en cours de réalisation de l’algo

c La mise au point (de l’algo au programme, puis du programme au résultat)

Traduire l’algorithme (coder / implémenter) dans le langage de programmation choisi -> Corriger les éventuelles erreurs de syntaxe jusqu’à ce que le programme puisse lancer.

Test du programme

Entrée des données -> Vérifier la conformité des résultats avec les résultats attendus

Les erreurs pendant l’exécution

- Exécution interrompue (le programme “plante” -> message d’erreur)
- Exécution “sans fin” (erreur de conception dans une boucle - condition de sortie jamais réalisé)
- Exécution “ok” mais résultats faux (erreurs de conception, quelque part dans l’algorithme)

Les opérations élémentaires

Opération de base que peut réaliser un ordinateur

a Description

Lire, Ecrire, Tester, Calculer, Affecter, Répéter => 6 opérations élémentaires

b Caractéristiques

Lire- Ecrire
Calculer Affecter
Rupture de séquentialité

Utilisation des variables

a variable isolées et tableaux de données

Les variables peuvent être : - isolées : 1 variable -> 1 information
- groupées sous forme de tableau : l'indice de la première case est toujours 0, accès en lect/écriture (tab(2) <- 10)

b Compteurs et accumulateurs

Rôle :

utilisés pour :

- comptabiliser le nombre de répétitions d'un événement
- totaliser les différences valeurs prises par une variable

Forme :

Nouvelle valeur de la variable <- ancienne valeur + 1 ou autre valeur