

STRUCTURE GENERALE D'UN PROGRAMME

(1) déclaration du programme

(2) déclaration des variables

(3) définition des procédures et fonctions

(4) programme principal

(1) syntaxe : **program** nom;

(2) syntaxe :

var i, k, n: **integer**;

x, a, b: **real**;

test: **boolean**;

liste: **array**[1..10] of

integer;

mat: **array**[1..3, 1..3] of **real**;

commentaire :

i, k, n sont de type entier ; ils prennent leurs valeurs dans $[-32768, 32767]$

x, a, b de type réel.

variable de type booléen ; sa valeur est **true** ou **false**.

variable de type tableau ; celle-ci prend sa valeur dans $\mathbb{N}^{10}$.

idem ; mat prend sa valeur dans l'ensemble des matrices carrées d'ordre 3 à coefficients réels.

(3) la définition des procédures et fonctions est identique à celle du programme : déclaration de la procédure ou de la fonction, puis déclaration des variables (locales), puis **begin** instructions **end** ;. La structure d'un programme simple est : (1), (2), (4), celle d'un programme structuré : (1), (2), (3), (4).

(4) syntaxe : **BEGIN** instructions **END**..

Les deux règles d'or pour bien commencer en turbo-pascal :

Une variable doit être déclarée en (2).

Une variable doit être initialisée en (4). Il y a deux moyens possibles :

- pour les variables d'entrée : **par lecture** ;

exemples :

writeln('écrire les valeurs de a et b');

readln(a, b);

writeln('écrire dix nombres entiers');

for i:=1 to 10 do read(liste[i]);

commentaires :

Quand le programme s'exécute, on voit apparaître à l'écran : écrire les valeurs de a et b.

les deux valeurs alors écrites par l'utilisateur sont affectées aux variables a et b.

Les dix nombres écrits par l'utilisateur sont affectés successivement à liste[1], liste[2], ... , liste[10].

- pour les variables de sortie : **par affectation**.

exemples :

test:=false;

x:=(a+b)/2;

commentaires :

a et b ont été initialisés.