

Lexique informatique : tout ce qu'il faut savoir sur Pascal

ECE3 Lycée Carnot

24 novembre 2009

Structure d'un programme :

```
PROGRAM nomduprogramme ;  
USES wincrt ;  
VAR nomvariable : type ;  
BEGIN  
instructions ;  
END.
```

Les commentaires éventuels se mettent entre accolades : {Ce programme ne fonctionne pas.}

Types de variables les plus fréquents :

Les seuls types utilisés à ce jour sont **real** pour les nombres réels, **integer** pour les petits entiers, **longint** pour les entiers plus gros, **char** pour les caractères (un seul caractère) et **string** pour une chaîne de caractères (un bloc de texte donc). On croisera aussi **boolean** pour des variables de test, qui ne peuvent prendre que deux valeurs : **true** ou **false**.

Procédures d'entrée-sortie :

WriteLn('texte',variable) ; affiche à l'écran tous les arguments placés entre apostrophes, et la valeur des variables qui ne sont pas placés entre apostrophes. Les différents éléments à afficher doivent être séparés par une virgule.

ReadLn(variable) ; stocke la valeur tapée par l'utilisateur dans la variable spécifiée.

Règles de syntaxe de base :

Toutes les instructions doivent être suivies d'un ;

L'affectation d'une valeur à une variable se fait via **variable := valeur** ; (le simple = étant réservé aux tests).

Syntaxe des boucles :

IF test THEN instruction1 ELSE instruction 2 ; (pas de ; avant le ELSE, qui est par ailleurs facultatif : si on ne met pas de ELSE, il ne se passe rien si la condition n'est pas vérifiée).

```
FOR i :=1 TO n DO  
BEGIN  
instruction1 ; ... ; instructionk ;
```

END; (le **BEGIN** et le **END** sont facultatifs dans le cas où on n'effectue qu'une instruction à chaque passage dans la boucle; le 1 et le n peuvent être remplacés par n'importe quel entier).

```
WHILE test DO  
BEGIN  
instruction1; ...; instructionk;  
END;
```

```
REPEAT instruction1; ...; instructionk;  
UNTIL test;
```

Opérations booléennes

Lorsqu'on veut effectuer (dans une instruction conditionnelle ou une boucle **WHILE** ou **REPEAT**) un test faisant intervenir plusieurs conditions, on dispose des opérations logiques suivantes (mettez des parenthèses partout, c'est plus prudent) :

(**test1**) **AND** (**test2**) sera vrai seulement si **test1** et **test2** sont vérifiés.
(**test1**) **OR** (**test2**) sera vrai dès que **test1** ou **test2** est vérifié.
NOT (**test1**) sera vrai si **test1** est faux (rarement utilisé).

Tableaux

Une variable de type tableau (par exemple **VAR t : ARRAY[1..10] OF real**; permet de stocker plusieurs variables d'un même type dans un tableau aux cases numérotées (de 1 à 10 dans notre exemple). La variable se trouvant dans la case numéro *i* du tableau est notée *t*[*i*]. On ne peut pas effectuer d'opérations simultanées sur toutes les cases d'un tableau, d'où emploi constant de boucles **FOR**.

Fonctions

Une fonction se déclare dans l'en-tête, en même temps que les variables. Une déclaration de fonction est un mini-programme pouvant contenir ses propres variables et autres instructions. Elle est nécessairement délimitée par un **BEGIN**; et un **END**; et doit contenir une ligne de la forme *f* := pour définir la fonction. L'en-tête de déclaration d'une fonction ressemble à :

```
FUNCTION nomfonction (var1 : type1 ; var2 : type2 ... ; vark : typek) : typeresultat ;
```

Probabilités

Un programme faisant intervenir le simulateur aléatoire doit contenir l'instruction **Randomize**; . La fonction prédéfinie **random** tire des nombres aléatoires dans l'intervalle [0; 1], la fonction **random(n)** tire un entier aléatoire compris entre 0 et *n* - 1.

Programmes classiques

Pour finir, trois programmes dont vous croiserez la structure suffisamment souvent pour que ça vaille le coup de les connaître sur le bout des doigts. Tout d'abord, un calcul de suite récurrente, ici le calcul du terme d'indice *n* (choisi par l'utilisateur) de la suite définie par $u_0 = 3$ et $u_{n+1} = \frac{u_n^2}{3} - 2$:

```
PROGRAM suite ;  
USES winCRT ;
```

```

VAR i,n : integer; u : real;
BEGIN
WriteLn('Choisissez la valeur de n');
ReadLn(n);
u :=3;
FOR i :=1 TO n DO
u :=u*u/3-2;
WriteLn(u);
END.

```

Un deuxième qui calcule $\sum_{k=1}^{k=?} \frac{1}{k^2}$ en s'arrêtant quand $\frac{1}{k^2}$ devient plus petit qu'un réel e choisi par

l'utilisateur :

```

PROGRAM somme;
USES wincrt;
VAR k : longint; e,s : real;
BEGIN
WriteLn('Choisissez la valeur de e');
ReadLn(e);
k :=0; s :=0;
REPEAT
k :=k+1;
s :=s+1/(k*k);
UNTIL 1/(k*k) < e;
WriteLn(s);
END.

```

Et un dernier simulant une loi géométrique de paramètre $\frac{1}{6}$, c'est-à-dire effectuant des lancers de dé jusqu'à tomber sur la face numérotée 6 :

```

PROGRAM geometrique;
USES wincrt;
VAR a,n : integer;
BEGIN
Randomize;
n := 0;
REPEAT a := random(6);
n := n+1;
UNTIL a := 5;
WriteLn(n);
END.

```