TD Info n°1 : Introduction à la programmation et à PASCAL

ECE3 Lycée Carnot

1er octobre 2010

Nous y voila, premier TD d'informatique de l'année, vous allez enfin savoir ce qui vous attend dans cette matière qui occupe une place un peu particulière en prépa commerciale. Mettons déjà une première chose au point : il ne s'agit pas d'apprendre à manipuler un quelconque outil ou logiciel informatique, mais bel et bien de se concentrer sur la pratique de la programmation, et plus particulièrement de la programmation appliquée aux mathématiques. D'où le paragraphe suivant :

Qu'est-ce que la programmation ?

Le principe de base de la programmation est simple : faire faire à une machine des choses qui nous demanderaient trop de temps ou de calculs. La programmation est de nos jours présente à peu près partout autour de nous : un avion en pilotage automatique, ou le jeu video dernier cri, fonctionnent à grands coups de programmes. La machine, qui a le grand avantage d'avoir des limitations beaucoup moins contraignantes qu'un être humain pour ce qui est de la capacité de calcul, a toutefois un gros défaut, celui de ne pas avoir de cerveau. Pour lui faire faire ce qu'on veut, il est donc essentiel de décomposer le travail en une suite d'instructions suffisamment élémentaires pour pouvoir être effectuées de façon mécanique par la machine. Un langage de programmation, c'est donc en gros la chose suivante : une liste de commandes relativement basiques et compréhensibles par la machine, à partir desquelles nous pourrons imaginer des algorithmes et construire des programmes permettant de faire des choses plus complexes. Exemple idiot pour vous donner une idée de ce que ça signifie : si votre machine sait multiplier deux nombres, vous pouvez écrire un programme permettant d'élever un nombre au carré (il suffit de le multiplier par lui-même).

Outre la liste des instructions disponibles, il existe une autre donnée inhérente au langage de programmation que l'apprenti programmeur se doit de maitriser : la syntaxe. En effet, la machine, décidémment très bête, ne comprendra vos instructions que si elles respectent scrupuleusement des règles de syntaxe très précises. C'est un peu comme si vous aviez en face de vous quelqu'un qui ne comprend pas une phrase sous prétexte que vous avez mal accordé le verbe, ou même qu'il manque un signe de ponctuation ou une majuscule. C'est incontestablement le côté le plus rebutant de la programmation au début : on a l'impression de ne jamais arriver à faire un programme sans erreurs de syntaxe...

Un programme, à quoi ça ressemble ?

L'écriture d'un programme se déroule en trois phases :

- l'écriture proprement dite, où le programmeur tape ses instructions à la suite les unes des autres, dans le langage adéquat (le langage PASCAL est un des plus simples qui soient, vous avez de la chance).
- la compilation, où la machine relit votre programme en vérifiant la syntaxe. Si elle trouve une erreu, elle vous le signalera (en essayant de vous expliquer quelle est l'erreur, mais ce n'est pas toujours très compréhensible). S'il n'y a pas d'erreur, votre programme est prêt à être exécuté, ce qui ne signifie absolument pas qu'il va faire ce que vous voulez (la machine ne peut pas deviner à quoi est censé servir votre programme !).

- l'exécution : vous faites tourner le programme, la machine exécute vos instructions et, habituellement, vous donne un résultat. Il peut aussi (hélas) y avoir des problèmes pendant cette étape : si vous demandez par exemple à l'intérieur du programme à faire une division par un nombre qui se trouve être égal à zéro, le programme va compiler (l'opération de division est écrite correctement), mais va renvoyer une erreur à l'exécution.

Quant au programme proprement dit, c'est une suite de lignes de texte ayant la structure suivante :

- une ligne d'en-tête qui annonce le nom du programme et ressemble à ceci :
  PROGRAM nomduprogramme ;
- une zone de déclarations, où le programmeur doit annoncer tout ce qu'il va utiliser à l'intérieur du programme (c'est un peu comme la douane, vous devez déclarer tout ce qui se trouve dans votre programme). Pour l'instant, nous nous conterons de déclarer de temps à autre des variables. Par exemple, si on veut écrire un programme résolvant les équations du second degré, les calculs vons faire intervenir des nombres notés a, b, c et Δ. Il faut préciser en début de programme que nous utiliserons des variables appelées a, b, c et Δ, et également dire qu'il s'agira de nombres réels, ce qu'on fait de la façon suivante :
  VAR a, b, c, delta : real ;
  Le « real » (réel) en fin de ligne est ce qu'on appelle le type de la variable, sa déclaration est obligatoire. Nous verrons un peu plus tard quels types de variables on peut définir en Pascal, et les quelques subtilités que ces histoires de variables et de types entrainent. Pour l'instant, on ne travaillera qu'avec des « real ».
- enfin, le corps du programme, qui débute nécessairement par un **BEGIN** et se conclut par un **END.** (oui, oui, avec un . derrière, si vous l'oubliez, Pascal va râler), et qui contient entre ces deux mots-clés les instructions du programme, séparées par des **;** (là encore, ces ; sont indispensables, tout comme ils le sont après l'en-tête et la déclaration des variables).

Quelques instructions histoire de pouvoir écrire nos premiers programmes

On ne va provisoirement utiliser que trois types d'instructions dans nos programmes :

- WriteLn() (ou Write, qui fait la même chose sans sauter de ligne, et n'est donc à peu près jamais utilisée pour des raisons de lisibilité) sert à écrire quelque chose à l'écran lors de l'exécution. Si on veut écrire du texte, il faut le mettre entre apostrophes, sinon Pascal croit que vous voulez afficher à l'écran la *valeur* d'une variable dont le nom est le texte en question.
  WriteLn('youhou') ; va ainsi écrire youhou à l'écran
  WriteLn(youhou) ; va afficher la valeur de la variable youhou (et s'il n'y a pas de variable s'appelant youhou, ce qui est au fond assez probable, Pascal va râler).
- ReadLn() (ou Read) permet de « lire » quelque chose que l'utilisateur tape à l'écran, et de l'affecter à une variable (précisée dans la parenthèse).
  ReadLn(a) ; va ainsi attendre que l'utilisateur tape une valeur puis appuye sur Entrée, et va affecter cette valeur à la variable a.
- l'instruction d'affectation est (de loin) la plus utilisée à l'intérieur d'un programme. Elle permet de stocker une valeur dans une des variables définies en début de programme. En gros, une variable est une case dans la mémoire de la machine sur laquelle vous avez mis une étiquette (le nom de la variable), et lui affecter une valeur revient à mettre cette valeur dans la case en question. Vous pouvez ensuite réutiliser cette valeur via son nom de variable. Les affectations en Pascal se font à l'aide de la syntaxe nomdevariable := valeur
  a :=7/2 ; va ainsi stocker la valeur $\frac{7}{2}$ dans la variable qui s'appelle a.
  a :=2*b ; va stocker dans la variable a le double de la valeur actuellement dans la variable b.