

INF Z10 COURS THEORIQUE N° 12

Initiation à la programmation

Généralités

Tout logiciel que vous utilisez a été "**programmé**" par des êtres humains.

En effet, les actions qui s'exécutent selon les ordres de l'utilisateur correspondent en fait à des **lignes de programme** qui ont été écrites dans ce but par des "programmeurs" ou "développeurs".

On peut dire que l'utilisation d'un ordinateur correspond à un double phénomène de communication.

Il y a communication de l'utilisateur vers la machine par l'intermédiaire de l'interface de Windows ou d'un logiciel quelconque, à l'aide des périphériques d'entrée comme le clavier ou la souris.

Il y a aussi communication de la machine vers l'utilisateur par l'intermédiaire de l'interface à l'aide des périphériques de sortie comme le moniteur.

Mais auparavant il y a eu un autre phénomène de communication entre le programmeur et la machine par l'intermédiaire d'un langage spécifique : le **langage de programmation**.

En fait on peut restreindre tout ce cheminement en une communication entre le programmeur et l'utilisateur final.

L'ordinateur apparaît alors plus comme un "**médiateur**" que comme une machine pensante.

Dans un sens général communiquer nécessite un langage (parlé, écrit, par signes ...)

Langage : ensemble de **caractères** et de **symboles** assemblés selon des **règles** strictes, dans le but de communiquer.

Langages naturels : celui des hommes, des animaux, des plantes ?

Langages artificiels : utilisés pour simplifier la communication (pictogrammes) et surtout en **programmation informatique**.

Ils sont formellement définis par des "grammaires" (**INF Z18** : Informatique et linguistique)

Programmer signifie écrire dans un langage de programmation informatique une suite d'instructions, organisée en algorithme dans un but précis, exécutable par un ordinateur.

Langages de programmation

- **Langage machine**

Pour communiquer, l'homme utilise sa langue maternelle ou une autre qui lui a été apprise.

La machine elle, ne connaît qu'un seul langage : le langage binaire qui est un assemblage de **0** et de **1 (bits)**.

C'est le seul langage compréhensible par les circuits intégrés constituant le "cerveau" de la machine (le microprocesseur).

Il est pratiquement impossible de programmer directement en langage binaire.

En effet qui peut retenir ce que signifie un assemblage de bits comme : **00101110010101111001101110001**

Il faut donc posséder un "traducteur" entre ces deux langages très différents.

Le **langage d'assemblage** (ou Assembleur) permet de développer des programmes proches des instructions de base d'un microprocesseur.

Il est toutefois complexe à mettre en œuvre. C'est du domaine des spécialistes.

Un exemple :

```
Message DB 'Bonjour' , '$' ; range 'Bonjour' dans une zone mémoire
MOV AH,09h ; charge dans le registre A l'instruction d'affichage
MOV DX,OFFSET Message ; charge le Message dans registre de données DX
INT 21h ; appel interruption 21 qui provoquera l'affichage
```

Ligne N° 2 en hexadécimal : **B4 09**

Même ligne N° 2 en binaire : **1011 0100 0000 1001**

Langages évolués

Des langages proches du langage naturel ont été développés pour faciliter la programmation.

Les lignes d'instructions sont alors plus compréhensibles par un non-spécialiste.

Ces lignes d'instructions (**code source**) sont ensuite soit exécutées ligne après ligne par un **interpréteur** soit traduites en langage machine par un **compilateur** avant l'exécution.

Quelques exemples typiques de langages évolués :

- **LISP** (1960) programmation fonctionnelle. Utilise essentiellement des listes et des arbres binaires.
- **PASCAL** (1969) programmation structurée procédurale. Très prisé pour ses qualités pédagogiques.

Le bout de programme précédent écrit en Pascal : **Begin Write ('Bonjour') End.**

- **PROLOG** (1973) programmation logique. Créé à Luminy. Utilisé en intelligence artificielle.
- **C** (1972) et **C++** (1983) : programmation de logiciels.
- **VISUAL BASIC** (1987) programmation graphique événementielle. Pour s'initier à la programmation.
- **JAVA** : le plus récent, portable, voisin du C++, utilisé par les développeurs internet.

Les 3 derniers sont des langages objets ou orientés **objets**.

Méthode de programmation

Ces informations sont données pour montrer la complexité du développement d'un logiciel. On ne les prendra pas en compte en initiation à la programmation.

- Spécification des besoins des futurs utilisateurs.
- Spécifications fonctionnelles : comment satisfaire aux besoins.
- Conception générale : division du logiciel en programmes.
- Conception détaillée : algorithme le plus adapté pour chaque programme.
- Assemblage des différents programmes.
- Codage de chaque module à l'aide du langage le plus adapté.
- Validation et qualification.

La **conception** est beaucoup plus importante que le **codage** qui peut être sous-traité dans le cas de gros logiciels.

Les macro-commandes

Un logiciel est un ensemble de programmes couvrant les besoins des utilisateurs dans un domaine de connaissances ou de productivité définis. Traitement de texte, Tableur, SGBD, didacticiel, CDRom culturel, jeux ...

Chaque fonction d'un logiciel est un petit programme qui réalise une ou plusieurs actions bien précises.

Centrage d'un paragraphe, encadrement de cellule, enregistrement, impression...

Pour obtenir un effet désiré on peut avoir à enchaîner **plusieurs actions**.

On peut **automatiser** l'ensemble de ces actions en créant une **macro-commande**.

- **Macros prédéfinies :**

Exemple : dans **Word** les formules de politesse sont des macros qui se trouvent dans le menu **Insertion** sous-menu **Insertion automatique...**

- **Enregistrer une macro :**

Par exemple dans **Word** on peut enregistrer une séquence d'instructions :

Pour cela : menu **Outils - Macro - Nouvelle macro...**

Ne pas oublier d'arrêter l'enregistrement (**soit** par menu, **soit** par une barre d'outils spécifiques).

Un exemple d'enregistrement de macro est traité en **TP** dans **Word (Codage)**.

- **Programmer une macro :**

Une macro est une suite d'instructions en **Visual Basic pour Application (VBA)**.

Sa syntaxe générale est :

```

Sub Nom_Macro()
  Instruction 1
  Instruction 2
  .....
End Sub

```

On peut la programmer directement dans l'environnement de développement de **VBA**.
 Un exemple de macro à taper soi-même est traité en **TP** dans **Word (RedTitle)**.

Quelques notions sur le langage VB

Caractéristiques principales du langage

- Ancien **BASIC** (**B**eginner's **A**ll purpose **S**ymbolic **I**nstruction **C**ode)
- Programmation par **objets** (briques logicielles)
- Programmation **graphique** (fenêtres, icônes, menus, souris...)
- Programmation **évènementielle** (sollicitations : souris, clavier, autre évènement...)

Structure générale

- Les objets manipulés sont appelés des **contrôles** (bouton de commande, zone d'affichage, conteneur d'image, etc.)
- L'interface utilisateur créée est fenêtrée. Une fenêtre est appelée une **feuille** (Form). Une feuille est elle-même un contrôle. Au lancement ⇒ feuille de démarrage.
- Chaque contrôle peut réagir à des **événements** qui lancent des suites d'instructions codées en BASIC.

Les contrôles : propriétés et événements

Contrôles et propriétés

Un objet (contrôle) peut posséder un grand nombre de propriétés par exemple sur sa forme, sa couleur, sa position dans la feuille, sa visibilité, etc.

La plus importante est la propriété **Name** qui donne un nom interne au contrôle. Ce nom permet de référencer le contrôle. Il ne faut pas confondre cette propriété avec la propriété **Caption** qui est seulement une légende affichée sur ou dans l'objet utilisé.

Syntaxiquement le nom d'un objet est séparé du nom d'une de ses propriétés par un point.

objet . propriété = valeur

Exemple 1 : Etiquette.CouleurDeFond = Bleu (c'est un exemple non-VB)

Exemple 2 : Affichage.Caption = "Bonjour"

Exemple 3 : Image.Visible = True

Exemple 4 : Quitter.Enabled = False

Contrôles et événements

Le code d'un événement associé à un contrôle forme une **procédure** évènementielle dont la syntaxe générale est

```

Sub NomContrôle_Evénement()
  Instruction 1
  Instruction 2
  .....
End Sub

```

Exemple le plus simple :

```

Sub Quitter_Click()
  End
End Sub

```