

Introduction à l'algorithme

Un ordinateur est une machine électronique programmable servant au traitement de l'information. Il peut être assimilé à un système produisant des résultats à partir :

- d'informations fournies ;
- de méthode de résolution permettant de traiter ces informations.

Les informations constituent des données et les méthodes de résolution aident à construire les algorithmes.

Un **algorithme** représente l'enchaînement des actions à réaliser nécessaires à la résolution d'un problème.

Exemple : déterminer le montant à payer pour l'achat de deux produits dont on connaît les prix. Le taux de TVA de 19,6 %.

Programme Facture

CONST taux_TVA = 0,196
VAR prix1, prix2, totalHT, totalTTC, TVA : réel

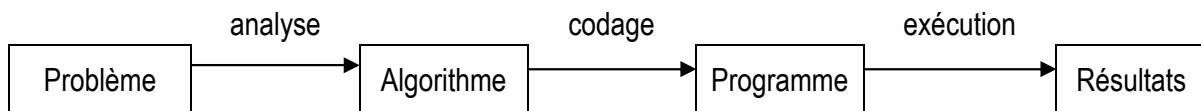
Début

```
| Afficher (« Entrer le prix d'une 1er produit : »)
| Saisir (prix1)
| Afficher (« Entrer le prix du 2ème produit : »)
| Saisir (prix2)
| totalHT ← prix1 + prix2
| TVA ← totalHT * taux_TVA
| totalTTC ← totalHT + TVA
| Afficher (« Le total à payer est de », totalTTC)
```

Fin

Pour obtenir des résultats, la description des données et les algorithmes doivent être codés sous forme de programmes interprétables par l'ordinateur.

Un **programme** est un enchaînement d'instruction, écrit dans un langage de programmation, exécutée par un ordinateur, permettant de traiter un problème et de renvoyer des résultats. Il représente la traduction un algorithme à l'aide d'un langage de programmation.



L'ordinateur est une machine totalement dénuée d'intelligence. Un programme exécute les instructions bien précises c'est-à-dire celle que le programmeur lui a donnée. Des erreurs ou des fantaisies lors de son exécution ne proviennent pas de l'ordinateur mais d'une erreur de conception.

Langage de programmation

Un ordinateur ne connaît que le système d'énumération binaire. Un langage utilisant le système binaire s'appelle langage machine.

Pour écrire des programmes sous des formes accessibles, on a développé dans les années 50, les langages d'assemblage.

Introduction algorithme

Cependant, un programme écrit en langage d'assemblage n'est plus directement exécutable par la machine. Il doit être traduit en un programme équivalent en langage machine. Cette opération de traduction s'effectue grâce à un autre programme appelé assembleur.

Le langage d'assemblage présente un inconvénient : il reste lié à l'ordinateur pour lequel il a été écrit car chaque famille de processeurs possède son propre langage d'assemblage. Il est difficile à utiliser car il nécessite de bonnes connaissances sur le fonctionnement des processeurs.

C'est pourquoi furent conçus les langages de programmation dits évolués plus compréhensibles et plus lisibles par l'homme.

Un langage de programmation est défini par des règles d'écriture des règles de construction que doivent respecter les programmes. La difficulté, pour le programmeur, consiste à respecter ses règles imposées.

On distingue plusieurs types de langage :

■ Langage de programmation procédurale

La programmation procédurale implique l'écriture de l'exécution des instructions les unes à la suite des autres (ordre séquentiel).

- **Fortran** (1957) : applications scientifiques ;
- **Cobol** (1960) : applications de légion du (ou de calcul, beaucoup d'édition) ; encore très présent dans le secteur des banques et des assurances (applications grands systèmes) ; menacée par l'essor de la micro-informatique et des nouveaux langages (la migration se fera cependant progressivement)
- **Basic** (1964) : langage destiné à être utilisé par des non informaticiens
- **Pascal** (1969) : langage scientifique ; peut utilisé en industrie
- **C** (1972) : conçues pour l'écriture du système UNIX

■ Langages orientés objets

La programmation orientée objet est basée sur l'écriture et la réutilisation de composants (objets de classes objets).

- **ADA** : applications scientifiques et de gestion, applications en temps réel (aéronautique, aérospatiale...)
- **C++** : évolution du C ; très utilisé dans l'industrie
- **Java** (1995) : conçues pour développer des l'applications fonctionnant indépendamment surtout type de processeur et de systèmes d'exploitation d'où le terme de machine virtuelle est Java (JVM) ; bien adapté pour les applications Web.

■ Langages de programmation événementielle

La programmation événementielle permet le développement d'applications à interfaces graphiques.

- Visual Basic
- Visual C++
- Visual J++
- C# (C sharp)
- DLPHI
- C++ builder
- J++ builder

Un programme doit être écrit en langage d'évolué à l'aide d'un éditeur de texte. Le programme ainsi obtenu est appelé **programmes sources** où code source.

Pour être utilisable, ce programme doit être traduit en langage machine. Le programme issu de cette traduction s'appelle **programme objet**.

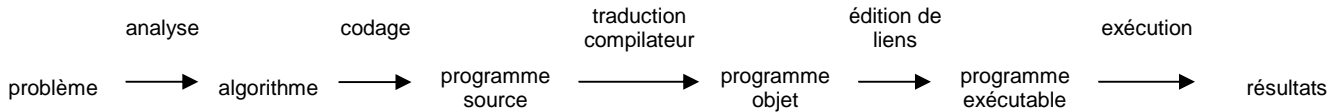
La traduction du programme source en programme objet s'appelle la **compilation** et est réalisée par un programme spécialisé appelé **compilateur**.

La compilation permet de détecter les erreurs dues à un non-respect de la syntaxe du langage.

Introduction algorithme

Un programme est très souvent composé de plusieurs parties (appelés et modules) qui devront être reliés entre elles pour former un fichier exécutable. C'est le rôle de **l'éditeur de liens** (linker) en queue de relier les différents modules d'un programme afin d'obtenir un programme exécutable.

Enfin pour être exécuté, un programme exécutable doit être chargé en mémoire centrale grâce **aux chargeurs**.



La création d'un programme nécessite donc plusieurs outils spécifiques qui sont souvent intégrés au sein d'un environnement de développement (**EDI : environnement de développement intégré**). Un environnement de développement est composé de l'ensemble des programmes qui permettent de créer des programmes exécutables (éditeurs, compilateur, « linker » et chargeurs) à ses programmes indispensables s'ajoutent des bibliothèques de fonction (mathématiques, graphique....) et un outil de débogage (programme de détection d'erreurs).

Les erreurs de logique et d'exécutions ne sont pas détectables à la compilation mais visibles uniquement lors du fonctionnement du programme exécutable. Les identifier et puis les corriger n'est pas toujours facile pour de grosses applications. Le débogueur permet de les localiser plus facilement.

Parmi les fonctions offertes le trouve :

- l'exécution pas à pas (instructions par instructions) du programme qui permet de suivre l'évolution de la valeur des données de celui-ci.
- L'utilisation de traceurs ou questions de variables qui affichent les valeurs et/ou adressent de certaines variables.
- Le placement de points d'arrêt permettant l'arrêt momentané d'un programme au cours de son exécution.

Remarque 1 : le programme source doit être sauvegardé avant la première compilation ou avant la première exécution.

Remarque 2 : à chaque modification du programme source, il devra être compilé ou que les modifications sont prises en compte.

*Remarque 3 : il existe un autre qui le traducteur de certains langages évolués : **l'interpréteur**.*

Ce programme traduit les instructions du programme source l'une après l'autre et en déclenche l'exécution au fur et à mesure. L'interpréteur ne fournit pas un programme objet écrit en langage machine car la traduction d'une instruction est perdue dès qu'elle a été exécutée. En conséquence, les programmes compilés traduits une fois pour toutes sont plus rapides à exécuter que les programmes interprétés ou la traduction est effectuée à chaque fois.

Les autres langages

- Langages de scripts

✓ SQL (Structured Query Language) signifie langage de requête structuré

Il permet de rechercher des informations dans des bases de données relationnelles. IBM a réalisé un prototype de ce langage et la société ORACLE, leader mondial des bases de données, a commercialisé à 1979 la première version. Depuis, l'ISO (International Standards Organization) a standardisé une première fois le langage 1987 puis en 1992 (SQL92) et plus récemment en 1999 (SQL3 ou SQL99).

✓ JavaScript

JavaScript est un langage développé par Netscape et Sun pour améliorer les fonctionnalités des pages Web écrites en HTML (langage à balises). Des scripts écrits en JavaScript sont directement incorporés au code HTML et interprétés par le navigateur. Le langage interprété et nécessite donc pas de compilateur.

- Langage à balises

Ces langages dits à balises servent à mettre en forme des documents électroniques pour être après utilisés en intranet, extranet ou pour la GEIDE (gestion de documents électroniques). Les données sont enregistrées entre des « balises ». Il ne s'agit pas vraiment de langages, mais de format pour structurer les données qui pourront être exploitées à partir d'un simple navigateur.

- ✓ HTML (HyperText Markup Language)

HTML n'est pas vraiment un langage de programmation; c'est une technique pour coder des pages à l'aide de commandes de mise en forme. Ces dernières sont ensuite interprétées par le navigateur (browser en anglais) et sont affichés à l'écran.

- ✓ XML

XML permet l'enregistrement, la gestion de la communication d'informations dans un format précis.

Un document XML est utilisable à partir d'un simple navigateur récent.

- Langage de programmation Web

- ✓ PHP
- ✓ ASP

Pourquoi passer par l'étape de l'algorithme ?

Nous pouvons effectivement nous poser cette question et pensait que l'on peut écrire directement le programme source à l'aide d'un langage de programmation.

Cependant, l'utilisation d'un langage de programmation présente des inconvénients suivants :

- La possibilité d'exécution immédiate diminue la productivité du programmeur toujours tenté d'essayer, sans trop réfléchir, de construire son programme un peu au hasard. Cette attitude se prolonge en cas d'erreur car il paraît plus facile de faire une modification et de réessayer plutôt que d'analyser la cause de l'erreur. On met alors beaucoup de temps à fabriquer un programme qui d'erreurs en erreurs et de modifications en modifications s'écartent généralement d'une solution efficace due à une bonne analyse préalable.
- Un langage de programmation est presque toujours orienté vers la résolution d'une famille particulière de problèmes (COBOL pour la gestion, C pour le calcul scientifique...). S'il est vrai que l'on peut presque tout faire avec tous les langages, on peut le faire plus ou moins facilement. Il est donc souhaitable de n'avoir aucun outil favori afin de faire une analyse claire et efficace.
- Par ailleurs, il faut savoir que le développement d'un logiciel commercial (traitement de texte, jeux...) peut demander le travail de plusieurs dizaines d'informaticiens pendant plusieurs mois ou années (on parle de programmes nécessitant par exemple 25 hommes-an). Chaque informaticien doit connaître sa tâche bien précisément et le code qu'il écrira devra s'imbriquer avec ceux écrits par ses collègues. Une bonne réflexion sur le problème à résoudre passe par la conception et l'écriture d'un algorithme résolvant le problème posé. Le passage par l'algorithmique est autant de temps gagné lors de l'écriture du programme source même si cela n'apparaît pas toujours lors de l'écriture de petits programmes de quelques dizaines de lignes.
- Les algorithmes sont indépendants des langages de programmation : les analystes programmeurs réfléchissent sur la spécification du problème et la méthode pour le résoudre, ils sont généralement réutilisables et beaucoup sont dans le domaine public.