Logique de programmation

1. Algorithmique et programmation

1.1 Algorithmique

L'algorithmique est la science des algorithmes.

Un algorithme est une suite ordonnée d'instructions qui indique la démarche à suivre pour résoudre un problème. Un algorithme ne doit contenir que des instructions compréhensibles par celui qui devra l'exécuter.

1,2 Définition d'un programme

Suite organisée d'instructions élémentaires, l'ordinateur les exécutera les unes après les autres, séquentiellement.

Un programme a UN début et UNE fin.

Langages informels pour la réalisation de la logique (algorithme) :

Pseudo-code

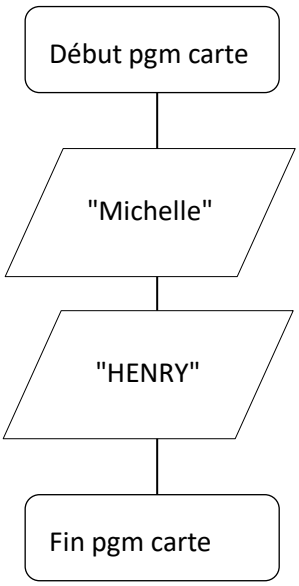
Début de pgm carte de visite

Afficher "Michelle"

Afficher 'HENRY'

Fin du pgm

Ordinogramme



Un bon programme est fiable (robuste), économe, clair et documenté.

1.3 La programmation

L'ordinateur ne comprend que des instructions élémentaires exprimées en langage machine.

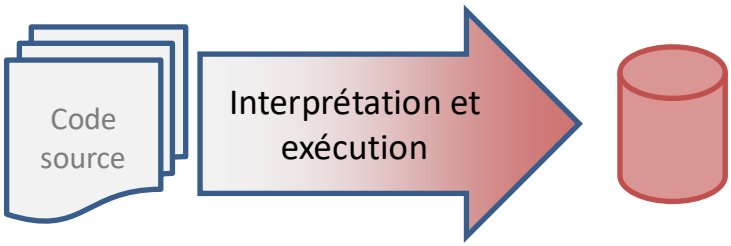
1.4 Langage de programmation

Un langage de programmation est composé d'un ensemble de mots-clés, de règles très précises indiquant comment on peut assembler ceux-ci pour former des « phrases » (syntaxe). Ils permettent au programmeur de programmer dans un langage plus « naturel » que le langage machine. Ils sont aussi des programmes « traducteur », se chargeant de traduire les instructions en langage binaire (séquence de 0 et de 1), seul langage compris de l'ordinateur, lui permettant ainsi d'exécuter celles-ci.

Compilateur et interpréteur

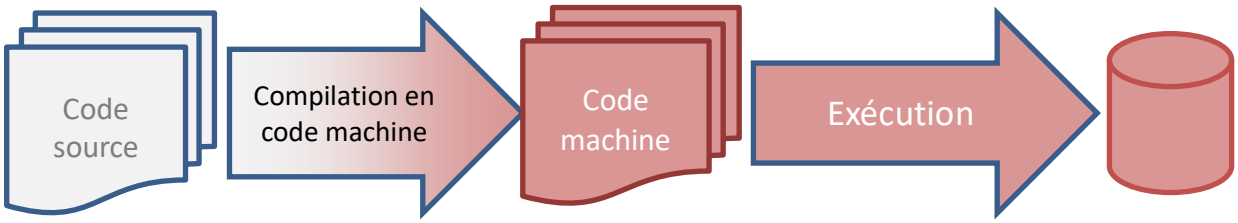
Il existe deux types de langages de programmation qui diffèrent par leur méthode de traduction en langage machine : interpréteurs et des compilateurs.

Les interpréteurs :



Les interprètes traduisent ET exécutent les instructions les unes après les autres.

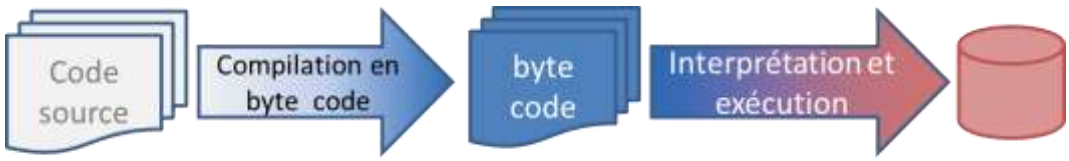
Les compilateurs :



Les compilateurs traduisent toutes les instructions du programme en langage machine. Une fois que la traduction est terminée, l'exécution peut se réaliser.

Comparatif des deux principes : avantages et inconvénients ?

Solutions hybrides



2. Variables, constantes, types, partie déclarative

2.1 Variables

Une variable est une valeur représentée par un identificateur (nom significatif). Sa valeur peut évoluer au cours de l'exécution du programme.

ex :

nom ← "Henry"

somme ← 2

RAM

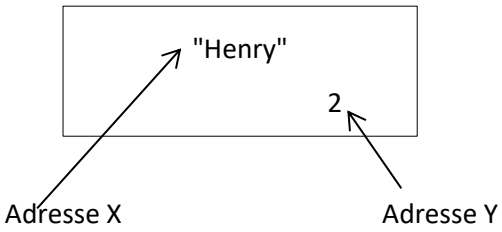


Table d'allocation

identificateur	adresse
nom	x
somme	y

Afficher nom => recherche dans la table d'allocation, si l'identificateur est trouvé alors il y aura affichage de la valeur stockée à l'adresse référencée.

2.2 Constantes

Une constante est une valeur représentée par un identificateur (nom significatif). Sa valeur ne peut évoluer au cours de l'exécution du programme.

Ex :

Début pgm compta

var montant : N ← 0

cste TVA : N ← 0.21 // seule assignation possible et obligatoire, dans la déclaration

… // calculs divers mentionnant la constante TVA en lieu et place de la valeur littérale

// En cas de changement de valeur de la tva, le programmeur ne devra modifier que la partie déclarative.

2.3 Types de données

L Logique ou booléen

N Nombre

C Caractère(s)

2.4 Partie déclarative

Section en début de programme qui regroupe toutes les déclarations de variables et de constantes.

Représentation ordinogramme

var montant : N ← 0

cste TVA : N ← 0.21

3. Affectation

Ou comment attribuer une valeur à une variable ou constante.

3.1 Assignation

C'est le programmeur qui attribue la valeur d'une variable vis une instruction du programme.

variable (à gauche) ← valeur (ou expression à droite)

ex :

var somme : N ←0

somme ← 5

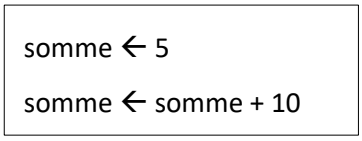
somme ← somme + 10

Afficher somme

somme
0
5
15

La table des valeurs montre la progression des valeurs d'une ou de plusieurs variables. Ell est très utile pour le tersting du code.

Représentation ordinogramme :



3.2 Lecture

Le programmeur, via une instruction de saisie, permet à l'utilisateur de rentre une valeur qui sera assignée à la variable.

Ex :

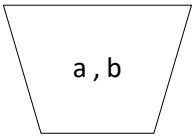
var a , b : N ← 0

…

Saisir a

Saisir b // Saisir a,b autorisé

Représentation ordinogramme :



Il est impossible de proposer une constante dans ce type d'instruction.

4. Expressions booléennes

4.1 Type Logique (ou booléen)

Valeurs possibles : true, false (ou T, F)

4.2 Expression booléenne

Une expression dont le type de résultat est logique.

Ex : 15 < 3

NB : les deux opérandes doivent être de même type

Opérateurs relationnels : < <= > >= = <> (ou !=)

Rappel opérateurs mathématiques : + - / * mod (%)

Le modulo est le reste de la division entière.

Ex : 5 mod 2 // résultat = 1

Priorité des opérateurs et rôle des parenthèses.

Opérateur de concaténation : +

Ex Mot← "Bon"+"jour" // "Bonjour"

Remarque : les algorithmes de comparaison :

Basé sur l'ASCII : 'A' < 'a' et 'Z' < 'a'

Basé sur Unicode : famille des a < famille des e … et au sein d'une même famille, ordre plus logique, comme 'a' < 'A'.

4.2 Opérateurs booléens (ou logiques)

Ex : 15 < 3 AND 27 = 24 + 3

4.2.1 AND

Table de vérité du AND		
1er opérande	2e opérande	Résultat
T	T	T
T	F	F
F	T	F
F	F	F

4.2.2 OR

Table de vérité du OR		
1er opérande	2e opérande	Résultat

T	T	T
T	F	T
F	T	T
F	F	F

4.2.3 NOT

Table de vérité du NOT	
Opérande	Résultat
T	F
F	T

4.2.4 Priorités

( )

NOT

AND

OR

5. Alternative

La structure alternative permet d'effectuer une séquence d'instructions si une condition est remplie et d'en effectuer une autre si celle-ci ne l'est pas.

L'exécution de l'alternative commence par l'évaluation de la condition (vraie ou fausse) suivie de l'exécution de la séquence d'instructions associées au résultat de la condition.

Syntaxe

SI condition (*expression booléenne)*
ALORS
bloc d'instructions
SINON
bloc d'instructions
FIN du sI

Lorsque le résultat de l'évaluation de l'expression booléenne est :

- VRAI : seules les instructions du bloc ALORS sont exécutées
- FAUX : seules les instructions du bloc SINON sont exécutées

Représentation

Pseudo - code

```
1. Debut lireTintin
2.
3.   VAR age :N <- 0
4.
5.  Afficher "âge : "
6.  Saisir age
7.  SI age >= 7 ET age <= 77
8.    ALORS
9.      Afficher "vous lisez Tintin"
10.  SINON
11.     Afficher " pas de Tintin"
12.   FINSI
13.Fin lireTintin
```

Ordinogramme

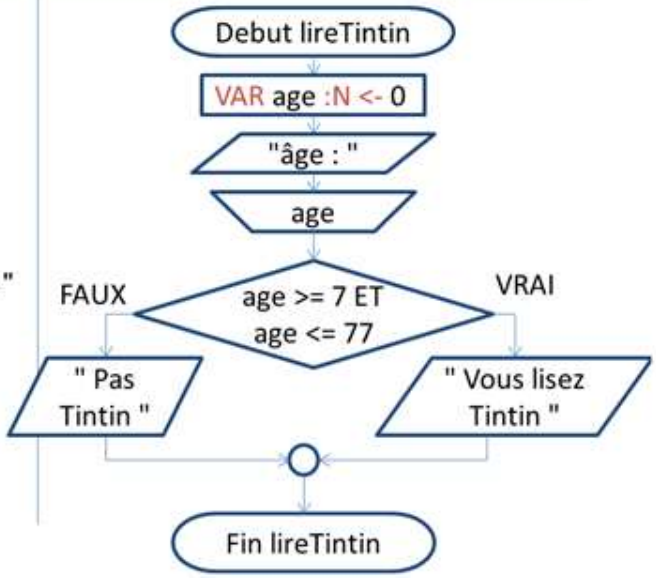


Table de valeurs

```
1.  Debut lireTintin
2.
3.   VAR age :N <- 0
4.
5.  Afficher "âge : "
6.  Saisir age
7.  SI age >= 7 ET age <= 77
8.    ALORS
9.      Afficher
          "vous lisez Tintin"
10.   SINON
11.     Afficher
          " pas de Tintin"
12. FINSI
13.Fin lireTintin
```

N°	age	Ecran / clavier
1	?	
2	?	
3	0	
4	0	
5	0	-> âge :
6	80	<- 80
7	80	
10	80	
11	80	->pas de Tintin
12	80	
13	?	

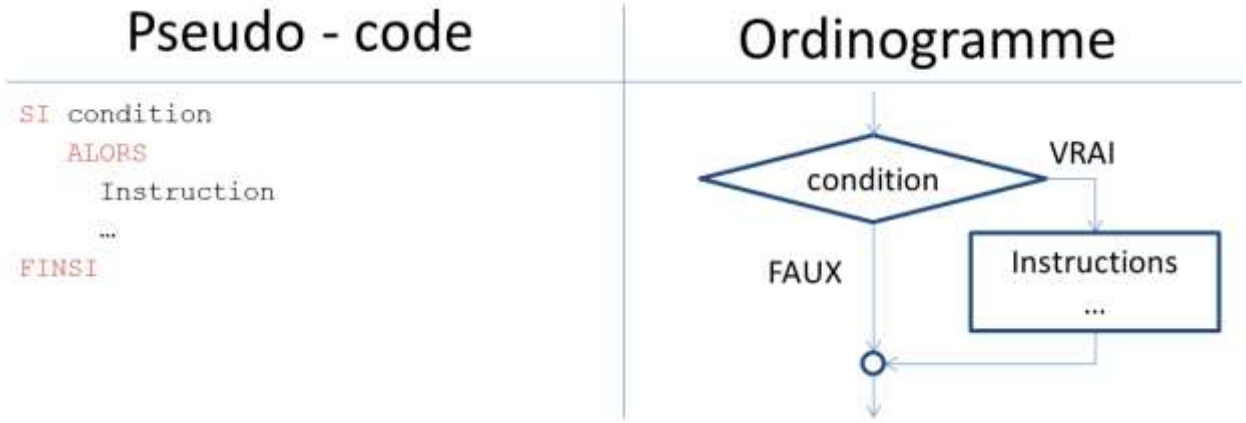
N°	age	Ecran / clavier
1	?	
2	?	
3	0	
4	0	
5	0	-> âge :
6	35	<- 35
7	35	
8	35	
9	35	->vous lisez Tintin
12	35	
13	?	

5.1 Alternative sans sinon (facultatif)

Lorsque aucune instruction n'est à exécuter quand la condition est fausse, on n'indiquera pas le sinon.

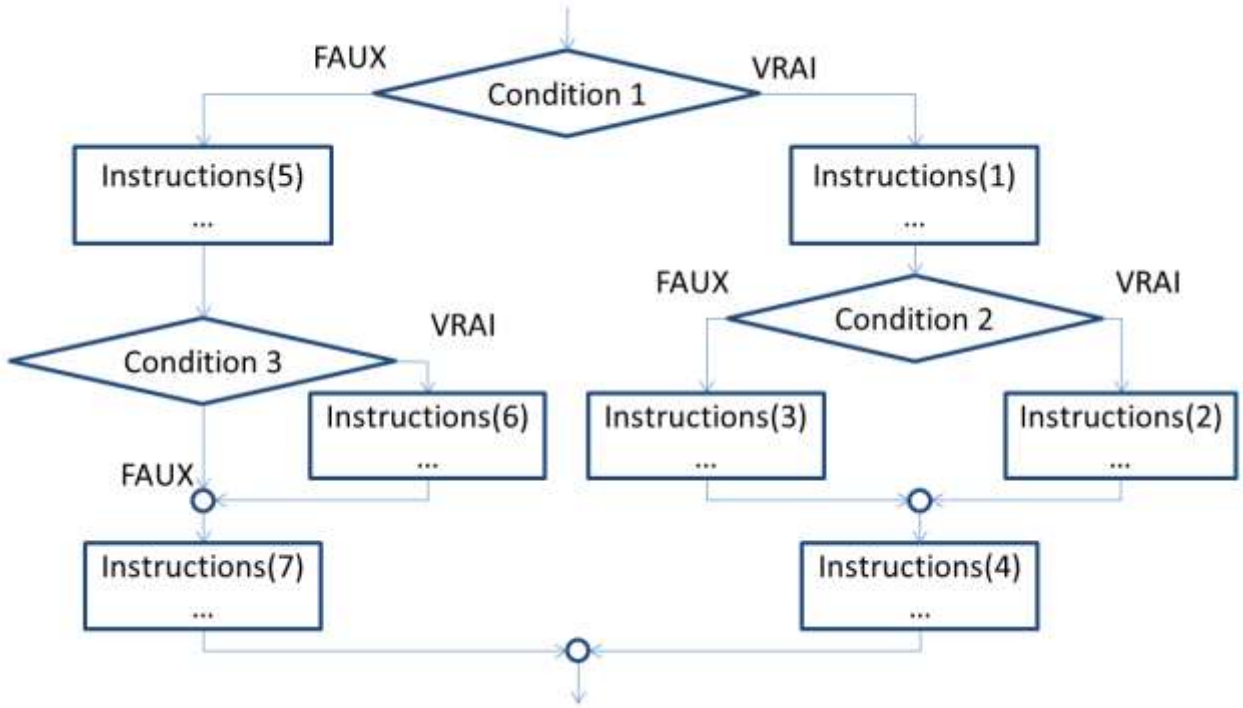
Syntaxe

SI condition
ALORS
bloc d'instructions
FINSI



5.2 Alternatives imbriquées

Ordinogramme



Pseudo-code

SI *condition 1*
ALORS
instructions(1)
SI *condition 2*
ALORS
instructions(2)
SINON
instructions(3)
Fin du si
instructions(4)
SINON
instructions(5)
SI *condition 3*
ALORS
instructions(6)
Fin du si
instructions(7)
Fin du si

6. Alternative composée

La structure alternative composée permet d'effectuer une suite d'instructions en fonction de la valeur d'une variable.

L'exécution de l'alternative commence par une recherche du bloc d'instructions liées à la valeur de la variable (appelée sélecteur) et se poursuit en exécutant les instructions de ce bloc.

Syntaxe de l'alternative composée

Cas où *variable (N ou C)*
valeur 1 :
bloc d'instructions
v*aleur 2 :*
bloc d'instructions
…
Sinon
bloc d'instructions
Fin du cas où

- Seules les instructions se trouvant dans le cas qui a la même valeur que l'expression numérique seront exécutées.
- Si aucun cas n'a la valeur de l'expression, ce sont les instructions du sinon qui sont exécutées (aucune si le sinon, qui est facultatif), n'est pas présent.
- L'alternative composée est une simplification d'écriture d'une alternative imbriquée.

Représentation

Pseudo - code

```
Debut ChoixBoisson

  VAR choix:N <- 0

 Afficher "Voulez-vous : 1 de l'eau, 2
    du jus, 3 du vin"
 Saisir choix
 CAS OU choix
    1:
      Afficher "de l'eau très bon choix"
    2:
      Afficher "du jus bravo"
    3:
      Afficher "du vin choix judicieux"
    Sinon
      Afficher "Vous avez raison de ne
    pas boire"
    FIN CAS OU
Fin ChoixBoisson
```

Ordinogramme