

CHAPITRE 1

RECONNAITRE LA STRUCTURE D'UN ALGORITHME

1. Définition d'un algorithme

2. Objets informatiques (variable, constante, type)

3. Structure d'un algorithme



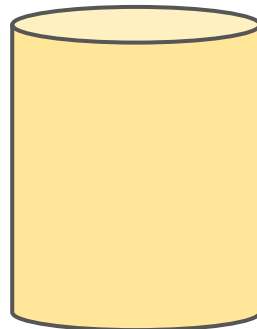
01 – STRUCTURE D'UN ALGORITHME

Objets informatiques (variable, constante, type)

Un algorithme manipule des objets (données) pour obtenir un résultat

Un objet est composé de :

- Un identificateur : Il s'agit du nom unique qui désigne cet objet.
- Un type : Il sert à déterminer la nature de l'objet qu'il soit simple (entier, caractère, etc.) ou composé (tableau,...), en particulier les valeurs possibles de l'objet, la taille mémoire réservée à l'objet et les opérations primitives applicables à l'objet.
- Une valeur : détermine le contenu unique de l'objet



Un objet



Un objet composé d'un identificateur



Un objet composé d'un identificateur + type



Un objet composé d'un identificateur + type + valeur

Composition d'un objet informatique

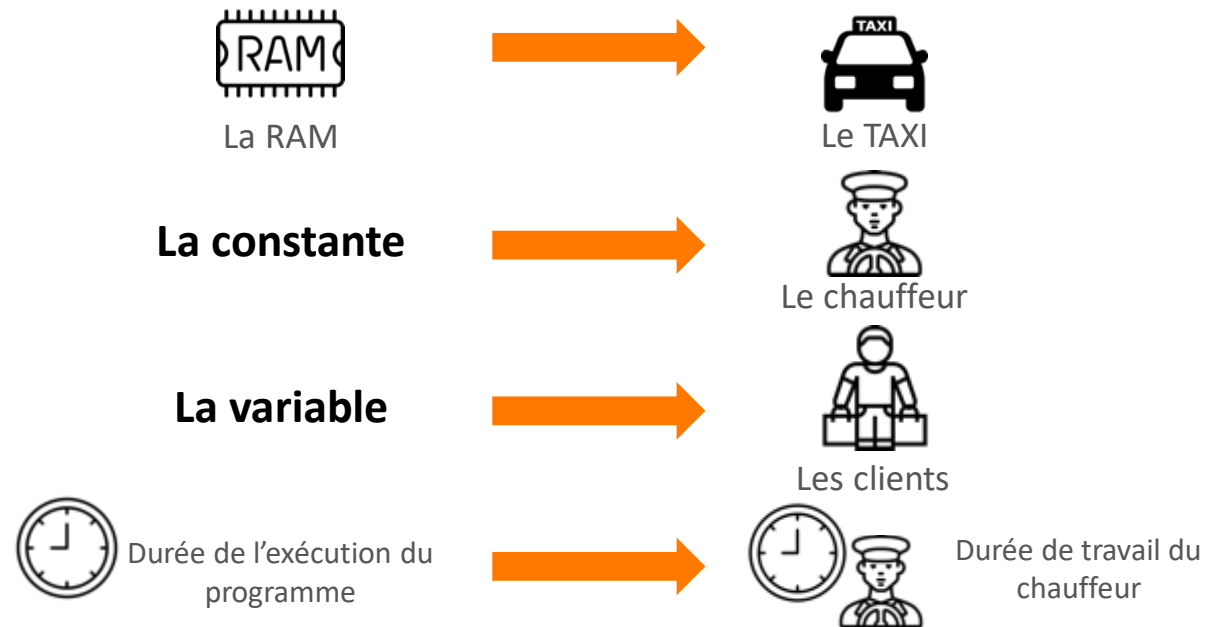
01 – STRUCTURE D’UN ALGORITHME

Objets informatiques (variable, constante, type)

Les objets sont de deux natures: les constantes et les variables

- Une **constante** est un objet dont l'état **reste inchangé** durant toute l'exécution d'un programme. On ne peut jamais modifier sa valeur et celle-ci doit donc être précisée lors de la définition de l'objet
- Une **variable** est un objet dont le contenu (sa valeur) **peut être modifié** par une action

Exemple:



Types des objets informatiques

01 – STRUCTURE D’UN ALGORITHME

Objets informatiques (variable, constante, type)



Types des objets

- À chaque variable utilisée dans le programme, il faut associer un type qui permet de définir :
 - L'ensemble des valeurs que peut prendre la variable
 - L'ensemble des opérations qu'on peut appliquer sur la variable

Les principaux types utilisés en algorithmique sont :

- Entier
- Réel
- Caractère
- Chaîne de caractères
- Logique ou booléen

Type entier

- Une variable est dite entière si elle prend ses valeurs dans \mathbb{Z} (ensemble des entiers relatifs)

Elle peut supporter les opérations suivantes :

Opération	Notation
Addition	+
Soustraction	-
Multiplication	*
Division entière	div
Modulo (reste de la division)	mod
Puissance	^

Exemples:

$$13 \text{ div } 5 = 2$$
$$13 \text{ mod } 5 = 3$$

01 – STRUCTURE D’UN ALGORITHME

Objets informatiques (variable, constante, type)



Type réel ou décimal

Il existe plusieurs types de réels représentant chacun un ensemble particulier de valeurs prises dans \mathbb{R} (ensemble des nombres réels)

Il existe deux formes de représentations des réels :

- La forme usuelle avec le point comme symbole décimal :

Exemples :

-3.2467 2 12.7 +36.49

- La notation scientifique selon le format aEb , où : a est la mantisse, qui s'écrit sous une forme usuelle, b est l'exposant représentant un entier relatif :

Exemples :

347 = 3.47E2 = 0.347E+3 = 3470E-1

Les opérations définies sur les réels sont :

Opération	Notation
Addition	+
Soustraction	-
Multiplication	*
Division (réelle)	/
Puissance	^

01 – STRUCTURE D’UN ALGORITHME

Objets informatiques (variable, constante, type)



Type caractère

- Un caractère peut appartenir au domaine des chiffres de "0" à "9", des lettres (minuscules et majuscules) et des caractères spéciaux ("*", "/", "{", "\$", "#", "%" ...)
- Un caractère sera toujours noté entre des guillemets.
- Le caractère espace (blanc) sera noté " "
- Les opérateurs définis sur les données de type caractère sont :

Opération	Notation
Égal	=
Différent	#
Inférieur	<
Inférieur ou égal	<=
Supérieur	>
Supérieur ou égal	>=

- La comparaison entre les caractères se fait selon leur codes ASCII : Le code ASCII est une norme informatique de codage de caractères, dans laquelle chaque caractère alphabétique, numérique ou spécial est représenté par un nombre binaire sur 7 bits (une chaîne composée de sept 0 ou 1).

Exemple :

" " < "0" < "1" < "A" < "B" < "a" < "b" < "{"

01 – STRUCTURE D’UN ALGORITHME

Objets informatiques (variable, constante, type)



Type logique ou booléen

- Une variable logique ne peut prendre que les valeurs "Vrai" ou "Faux"
- Elle intervient dans l'évaluation d'une condition
- Les principales opérations définies sur les variables de type logique sont : la négation (NON), l'intersection (ET) et l'union (OU)
- L'application de ces opérateurs se fait conformément à la table de vérité suivante :

A	B	NON (A)	A et B	A ou B
Vrai	Vrai	Faux	Vrai	Vrai
Vrai	Faux	Faux	Faux	Vrai
Faux	Vrai	Vrai	Faux	Vrai
Faux	Faux	Vrai	Faux	Faux

Table de vérité des opérateurs logiques

01 – STRUCTURE D’UN ALGORITHME

Objets informatiques (variable, constante, type)



Expressions

- Ce sont des combinaisons entre des variables et des constantes à l'aide d'opérateurs
- Elles expriment un calcul (expressions arithmétiques) ou une relation (expressions logiques)

Les expressions arithmétiques:

Ce sont des expressions simples construites avec des opérateurs arithmétiques et des constantes ou des références à des cellules.

Exemple: $x * 53.4 / (2 + \text{Pi})$

- L'ordre selon lequel se déroule chaque opération de calcul est important
- Afin d'éviter les ambiguïtés dans l'écriture, on se sert des parenthèses et des relations de priorité entre les opérateurs arithmétiques :

Priorité	Opérateurs
1	- Signe négatif (opérateur unaire)
2	() Parenthèses
3	^ Puissance
4	* Et / Multiplication et division
5	+ et – addition et soustraction

En cas de conflit entre deux opérateurs de même priorité, on commence par celui situé le plus à gauche

Ordre de priorité des opérateurs arithmétiques

01 – STRUCTURE D’UN ALGORITHME

Objets informatiques (variable, constante, type)



Expressions

Les expressions logiques:

- Ce sont des expressions de type booléen, c'est à dire des expressions pouvant prendre la valeur vrai ou faux.
- Elles sont formées à partir des combinaisons entre des variables et des constantes à l'aide des opérateurs relationnels et logiques.
- Les opérateurs relationnels sont (=, <, <=, >, >=, #) et les opérateurs logiques sont (NON, ET, OU, etc.)
- On utilise les parenthèses et l'ordre de priorité entre les différents opérateurs pour résoudre les problèmes de conflits

Priorité	Opérateur
1	NON
2	ET
3	OU

Opérateurs logiques

Priorité	Opérateur
1	la parenthèse () la plus interne
2	>
3	>=
4	<
5	<=
6	=
7	#

Opérateurs relationnels

Exemple :

$$5 + 2 * 6 - 4 + (8 + 2 ^ 3) / (2 - 4 + 5 * 2) = 15$$

On commence par calculer ce qui est entre les parenthèses : $(8 + 2 ^ 3) = 16$ et $(2 - 4 + 5 * 2) = 2$

L'expression deviant : $5 + 2 * 6 - 4 + 8 = 15$.

01 – STRUCTURE D'UN ALGORITHME

Objets informatiques (variable, constante, type)



Déclaration d'une variable

- Toute variable utilisée dans un programme doit être l'objet d'une déclaration préalable
- En pseudo-code, la déclaration de variables est effectuée par la forme suivante :

Var liste d'identificateurs : type

Exemple:

Var
i, j, k : Entier x, y : Réel
OK: Booléen
C1, C2 : Caractère

Déclaration d'une constante

- En pseudo-code, la déclaration des constantes est effectuée par la forme suivante :
- Par convention, les noms de constantes sont en majuscules
- Une constante doit toujours recevoir une valeur dès sa déclaration

Const identificateur=valeur : type

Exemple: Const PI=3.14 : réel

Pour calculer la surface des cercles, la valeur de Pi est une constante mais le rayon est une variable

CHAPITRE 1

RECONNAITRE LA STRUCTURE D'UN ALGORITHME

1. Définition d'un algorithme
2. Objets informatiques (variable, constante, type)

3. Structure d'un algorithme



01 – STRUCTURE D'UN ALGORITHME

Structure d'un algorithme



Structure d'un algorithme

<NOM_ALGORITHME>

Const

Const1 = val1 : type

Const2 = val2 : type

.....

Var

v1 : type

v2 : type

.....

Début

Instruction 1

Instruction 2

.....;

Fin

Liste des constantes

Liste des variables

Corps de l'algorithme

Cercle

Const

Pi = 3.14

Var

r, p, s : Réel

Début

Écrire ("Entrer le rayon du cercle : ")

Lire (r)

$p := 2 * \text{Pi} * r$

$s := \text{Pi} * r^2$

Écrire ("Périmètre = ", p)

Écrire ("Surface = ", s)

Fin