

Ministère de l'enseignement supérieur et de la recherche
scientifique

Université Abderrahmane MIRA de Bejaia

Faculté de Technologie
Département de Technologie



Structure des ordinateurs et applications

Présentation

- ✓ Dr. DJAFRI Ghani
- ✓ ghani.djafri@univ-bejaia.dz



À propos du cours

- **Crédits:** 02
- **Coefficients:** 02

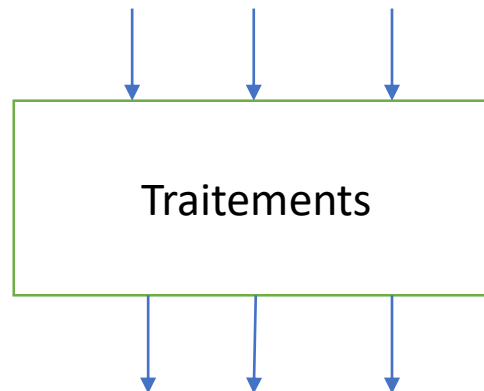
Notion d'algorithme et de programme

Notion d'algorithme et de programme

- Concept d'un algorithme

Un Algorithme est une **séquence d'instructions ordonnées**, qui permet de **résoudre un problème**. Il reçoit des données en entrée, les traite selon les instructions, et fournit des données en sortie pour obtenir le résultat souhaité.

Ensemble de données (en entrée)

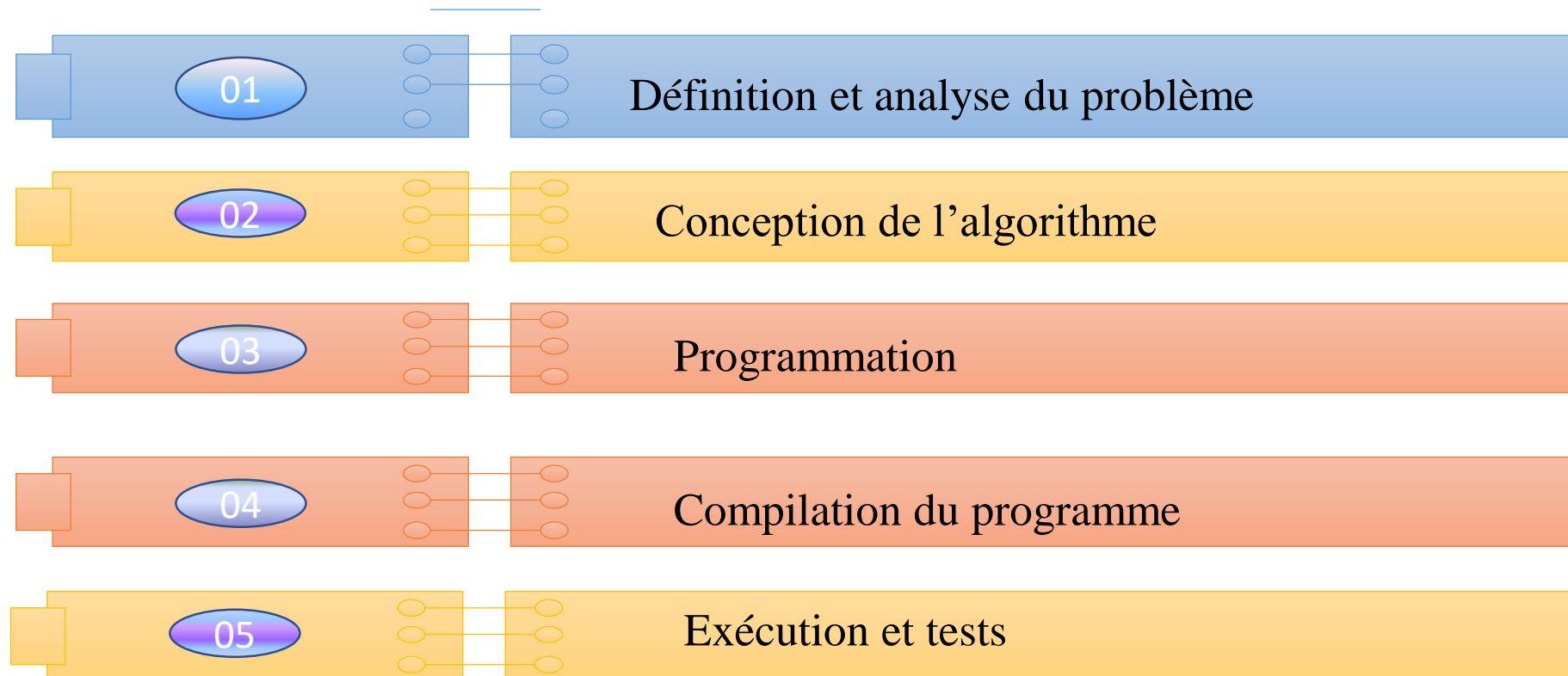


Ensemble de données (en sortie)

Notion d'algorithme et de programme

- Les étapes de résolution d'un problème

La résolution d'un problème par un algorithme se fait selon plusieurs étapes logiques et ordonnées.
Voici les étapes principales à suivre :



Notion d'algorithme et de programme

○ Structure d'un algorithme/programme

Un algorithme/programme se compose de trois parties principales : L'entête, la partie déclarative et le corps de l'algorithme/programme:

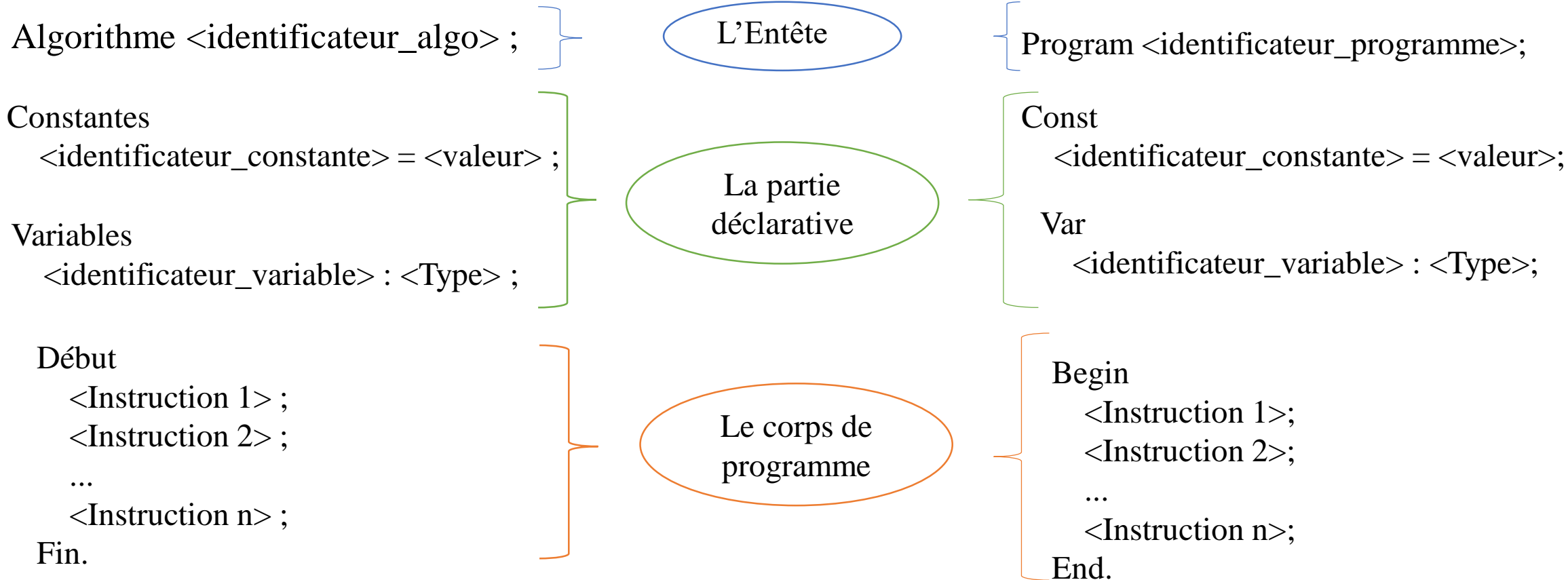
- **L'Entête:** c'est la première ligne de l'algorithme. Elle permet de donner un nom à l'algorithme à travers un identificateur;
- **La partie déclarative :** c'est la section où l'on déclare toutes les données que l'algorithme va utiliser;
- **Le corps de programme/algorithme :** c'est l'ensemble des instructions qui font fonctionner l'algorithme.

Notion d'algorithme et de programme

Structure d'un algorithme/program

Algorithme

Pascal



Notion d'algorithme et de programme

○ L'Entête

L'entête sert à donner un nom à l'algorithme en utilisant un identificateur. Ce dernier est précédé par le mot clé “**Algorithme**”. Alors qu'est-ce qu'un identificateur ?

Identificateur : Un identificateur est une chaîne de caractère qui permet de donner un nom unique à un programme (algorithme), une constante ou une variable. Cette chaîne doit commencer soit par un caractère alphabétique ou par un tiret du 8 (_) et ne peut contenir que des caractères alphanumériques. Aussi, les mots réservés (mots-clés) d'un langage de programmation ne peuvent être utilisés comme identificateurs.

Identificateur valide	Identificateur non valide
Commencer soit par un caractère alphabétique ou par un tiret du 8 (_)	Commencer soit par un caractère spécial ou par un caractère numérique
	Contenir un caractère special &,%,\$,#, @,?,-:.,.+
Contenir que des caractères alphanumériques	(mots-clés) réservés au langage pascal: begin, end, program, var, const, real, integer, char, if, then, else, while, for, do, repeat

Notion d'algorithme et de programme

○ La partie déclarative

La partie déclarative sert à déclarer les différentes données que l'algorithme utilise (Constantes, variables,.. etc.). Une donnée non déclarée et utilisée par le programme engendre une erreur lors de la compilation. Alors qu'est-ce qu'une variable et qu'est-ce qu'une constante ?

- **Constantes** c'est une donnée qui ne change jamais pendant l'exécution du programme;
- **Variables** :est un espace mémoire qui contient une valeur pouvant être modifiée pendant l'exécution du programme.

Constantes
Const PI = 3.14; Constante réelle.
Const MAX = 10; Constante entière.
Const cc = 'a'; Constante caractère.
Const ss = 'algo'; Constante chaîne de caractère.
Const b1 = true; Constante booléenne.
Const b2 = false; Constante booléenne.

Algorithme	Pascal	Valeurs
Entier	Integer	Représente l'ensemble {..., 4, 3, 2, 1, 0, 1, 2, 3, 4, ...}
Réel	Real	Représente les valeurs numériques fractionnels et avec des virgule fixes (ou flottante)
Booléen	Boolean	Représente les deux valeurs <i>TRUE</i> et <i>FALSE</i> .
Caractère	Char	Représente tous les caractères imprimable.
Chaîne de caractères	String	Une séquence d'un ou plusieurs caractères

Notion d'algorithme et de programme

○ Le corps de programme/algorithme

Le corps d'un algorithme/programme est constitué d'un ensemble d'actions / instructions ordonnées de manière séquentielle et logique. Ces instructions se divisent en quatre types distincts: Lecture , *Écriture*, *Affectation*, *Structures de contrôle*(*Structure de Test alternatif simple / double*, *Structure* répétitives (itérative)).

Remarques

- **Une instruction** spécifie une opération à exécuter dans un algorithme ou un programme(lire une valeur, effectuer un calcul, afficher un résultat).
- Les instructions se situent entre les mots-clés Début (Begin) et Fin (End.).
- Les instructions sont séparées par des ; et sont exécutées séquentiellement, c'est-à-dire l'une après l'autre, depuis le Begin jusqu'au End.

Notion d'algorithme et de programme

- Le corps de programme(Types d'instructions)

1. Instructions d'Entrées/Sorties (Lecture / Écriture)

Entrées (Lecture)

Une instruction d'entrée est une instruction qui permet d'introduire (saisir) une valeur à l'Algorithme à travers le clavier.

La syntaxe d'une lecture

Algorithme	Pascal
Lire (id_var) ;	read (id_var) ; Readln (id_var) ; {Lecture avec retour à la ligne.}
Lire (id_var1, id_var2, .., id_varN) ; {Lecture de plusieurs variables en même temps.}	Read (id_var1, id_var2, id_var3, ..., id_varN) ;

Remarque : Il est important de noter que l'instruction de lecture concerne uniquement les variables, **on peut pas lire** des constantes ou des valeurs(nombres).

Notion d'algorithme et de programme

- Le corps de programme(Types d'instructions)

1. Instructions d'Entrées/Sorties (Lecture / Écriture)

Sorties (Écriture))

L'écriture est une instruction qui permet d'afficher, à l'écran, des données. Ces dernières peuvent être un message, une valeur(nombre), la valeur d'une variable, une constante et même le résultat d'un calcul.

La syntaxe d'une écriture

Algorithme	Pascal
Écrire (id_var) ;	Write (id_var) ; Writeln (id_var) ; {Écriture avec retour à la ligne.}
Écrire (' Ceci est un message ! ') ; {Affichage du message : Ceci est un message ! à l'écran.}	Write (' Ceci est un message ! ') ;

Pascal
write ('Bonjour'); write(a, b, c); write(5+2); write(a+b-c); write(5<2);

Notion d'algorithme et de programme

- Le corps de programme(Types d'instructions)

Remarque :

□ En Pascal, il est possible de **commenter** un programme, il suffit d'écrire les commentaires entre accolades { } ou de mettre deux slaches // avant le commentaire.

Par **exemple** : {Ceci est un commentaire} ou // Ceci est un commentaire.

Le **commentaire** n'est pas pris en compte à la compilation. Il sert à rendre le programme plus clair à la lecture, à noter des remarques, etc.

Notion d'algorithme et de programme

- Le corps de programme(Types d'instructions)

2. Instruction d'affectation)

Affectation : l'affectation est une instruction qui permet de donner ou modifier la valeur d'une variable. La valeur affectée peut être une constante, un nombre, une expression,

La syntaxe d'une affectation

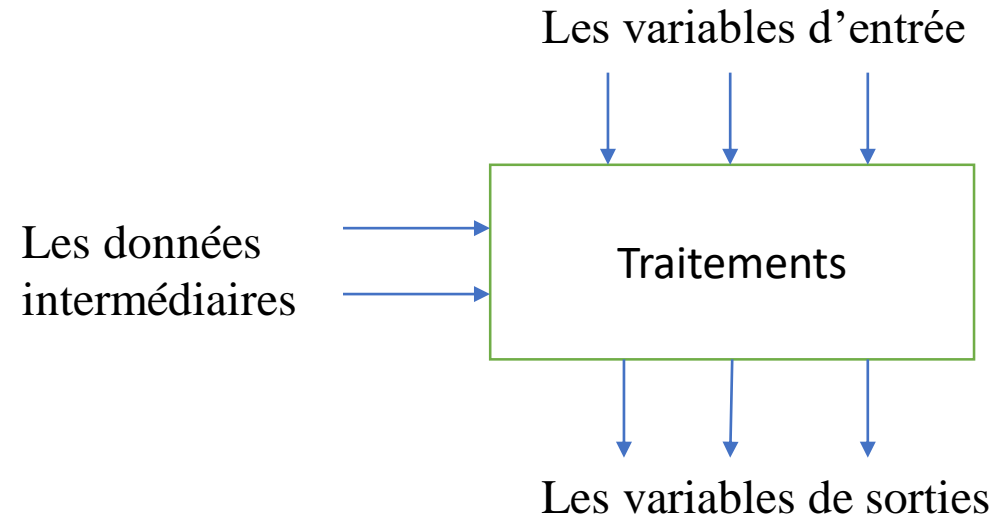
Algorithme	Pascal
a ← b ; {a : DOIT être une variable. b peut être une valeur, une constante, une variable ou une expression.}	a :=b;

Remarque : Remarque : Les deux côtés d'une affectation doivent être du même type sauf pour le type entier qui peut être stocké dans un réel car l'ensemble des réels inclut l'ensemble des entiers.

Algorithme	Pascal	Signification
<i>a ← 5</i> <i>b ← a+5</i> <i>sup ← a>b</i>	<i>a:=5;</i> <i>b:=a+5;</i> <i>sup:=a>b;</i>	Mettre la valeur 5 dans la variable a Mettre la valeur de l'expression a+5 dans la variable B a>b donne un résultat booléen, donc sup est une variablebooléenne

Notion d'algorithme et de programme

○ Schéma générale d'un algorithme



- Entrées (Inputs) : les données initiales nécessaires pour démarrer le traitement. Ce sont généralement les **variables lues** avec l'instruction Lire;
- **Traitement (Process)** : ce sont les opérations ou calculs effectués sur les données d'entrée
- **Les données intermédiaires** ce sont des valeurs utilisées pendant le traitement, comme des variables ou des constantes, qui ne sont ni lues ni affichées directement ;
- **Sorties (Outputs)** : ce sont les résultats produits par l'algorithme, généralement affichés à l'écran avec l'instruction Écrire.

Exemple 1: (Conversion d'Euros en Dinars)

Écrire un programme en Pascal qui permet de convertir un montant en euros en dinars.

Le programme devra :

1. Demander à l'utilisateur de saisir le montant en euros;
2. Utiliser un taux de conversion fixe (par exemple : 1 Euro = 16 Dinars);
3. Calculer et afficher le montant correspondant en dinars..

Exemple 2: (Calcul de la moyenne de trois nombres)

Écrire un programme Pascal qui :

1. Lit trois nombres réels A, B, C;
2. Calcule la moyenne $M = (A + B + C) / 3$;
3. Affiche la moyenne.

Exemple 1	Exemple 2
<pre>Program EuroEnDinar; const k=16; { 1 Euro = 16 Dinars } var E: real; D: real; begin write('Entrez le montant en Euros : '); readln(E); D := E *K; writeln(' Euro(s) = ', E:0:2, 'Dinar(s)=',D:0:2); end.</pre>	<pre>Program MoyenneTroisNombres; var A, B, C: real; M: real; Begin // Lecture des trois nombres write('Entrez le nombre A : '); readln(A); write('Entrez le nombre B : '); readln(B); write('Entrez le nombre C : '); readln(C); // Calcul de la moyenne M := (A + B + C) / 3; // Affichage de la moyenne writeln('La moyenne est : ', M:0:2); end.</pre>