

Ministère de L'Éducation

**Direction Générale des Programmes et de la Formation
Continue**

**Conventions d'enseignement de l'analyse
et de l'algorithmique**

(Toutes sections)

A. L'analyse descendante

1. Principe

L'approche descendante consiste à établir une liste de définitions selon les règles suivantes :

- **R1** : Commencer impérativement par le(s) **résultat(s)**.
- **R2** : Chaque objet apparaissant dans une définition et qui n'est pas défini, doit faire l'objet, lui aussi, d'une définition, il devient à son tour, un résultat (intermédiaire) à définir.
- **R3** : Arrêter ce processus une fois que tous les objets utilisés soient définis.

Pour définir un objet, il est nécessaire de vérifier à chaque fois, que les propriétés suivantes sont vérifiées :

- **P1** : La définition de l'état d'un objet ne se fait qu'une seule fois.
- **P2** : Un objet est défini si et seulement si, tous les objets intermédiaires nécessaires à sa définition ont été à leur tour définis.

N.B. :

- **Il est possible de définir plusieurs objets à la fois.**
- **Lors de la définition d'un objet, il est possible que d'autres objets se définissent.**

B. Les syntaxes des structures algorithmiques

1. Les opérations élémentaires simples

a. L'opération d'entrée

En Analyse	En Algorithmique
Nom_objet=Donnée ("Message ")	Écrire ("Message"), Lire(Nom_objet)

b. L'opération de sortie

En Analyse	En Algorithmique
Écrire ("Message",Nom_objet, Expression)	Écrire ("Message",Nom_objet, Expression)

c. L'opération d'affectation

En Analyse	En Algorithmique
Nom_objet ← Expression	Nom_objet ← Expression

2. Les structures de contrôle conditionnelles

a. La structure de contrôle conditionnelle simple

En Analyse	En Algorithmique
Nom_objet= [Initialisation]	<i>Initialisation</i>
Si Condition Alors	Si Condition Alors
Traitement1	Traitement1
Sinon	Sinon
Traitement2	Traitement2
FinSi	FinSi

b. La structure de contrôle conditionnelle généralisée

En Analyse	En Algorithmique
Nom_objet= [Initialisation]	<i>Initialisation</i>
Si Condition 1 Alors Traitement 1	Si Condition 1 Alors Traitement 1
Sinon Si Condition 2 Alors Traitement 2	Sinon Si Condition 2 Alors Traitement 2
Sinon Si Condition 3 Alors Traitement 3	Sinon Si Condition 3 Alors Traitement 3
Sinon	Sinon

.....
.....
Sinon si condition N-1 Alors Traitement N-1	Sinon si condition N-1 Alors Traitement N-1
Sinon Traitement N	Sinon Traitement N
FinSi	FinSi

c. La structure de contrôle conditionnelle à choix multiples

En Analyse	En Algorithmique
Nom_objet= [<i>Initialisation</i>]	<i>Initialisation</i>
Selon sélecteur Faire	Selon sélecteur Faire
Valeur1 : Traitement 1	Valeur1 : Traitement 1
Valeur 2, Valeur 3, Valeur 4 : Traitement 2	Valeur 2, Valeur 3, Valeur 4 : Traitement 2
Valeur 5 .. Valeur 6 : Traitement 3	Valeur 5 .. Valeur 6 : Traitement 3
.....
.....
Valeur N-1: Traitement N-1	Valeur N-1: Traitement N-1
Sinon Traitement N	Sinon Traitement N
Fin Selon	Fin Selon

3. Les structures de contrôle itératives

a. La structure de contrôle itérative complète

En Analyse	En Algorithmique
Nom_objet= [<i>Initialisation</i>]	<i>Initialisation</i>
Pour VC de D à F Faire	Pour VC de D à F Faire
Traitement	Traitement
Fin Pour	Fin Pour

b. La structure de contrôle itérative à condition d'arrêt

• Première formulation

En Analyse	En Algorithmique
Nom_objet= [Initialisation]	<i>Initialisation</i>
Répéter	Répéter
Traitement	Traitement
Jusqu'à condition(s)	Jusqu'à condition(s)

• Deuxième formulation

En Analyse	En Algorithmique
Nom_objet= [Initialisation]	<i>Initialisation</i>
Tant que Condition(s) Faire	Tant que Condition(s) Faire
Traitement	Traitement
Fin Tant que	Fin Tant que

4. Les modules

a. Les Fonctions

• L'appel(En analyse et algorithmique)

Nom_objet ← FN Nom_fonction(pe₁, pe₂, ... , pe_n)

• L'entête de la définition(En analyse et algorithme)

DEF FN Nom_fonction(pf₁:type₁, pf₂:type₂, ... , pf_n:type_n) : type_fonction

b. Les Procédures

• L'appel en analyse

Nom_objet=PROC Nom_procédure((pe₁, pe₂, ... , pe_n))

• L'appel en algorithme

PROC Nom_procédure((pe₁, pe₂, ... , pe_n))

• L'entête de la définition(En analyse et algorithme)

DEF PROC nom_procédure(pf₁:type₁, pf₂:type₂, ... , pf_n:type_n)