

Leçon 1

**La structure de contrôle conditionnelle simple**

I-Définition:

\* La structure de contrôle conditionnelle permet à un programme de modifier son traitement en fonction d'une condition.

\* Il existe trois formes d'instructions conditionnelles:

- Forme simple
- Forme généralisée.
- Forme à choix

II-La structure de contrôle conditionnelle simple :

**A-La forme réduite:**

**1-Définition:**

Une structure de contrôle conditionnelle est dite à forme simple réduite lorsque le traitement dépend d'une condition. Si la condition est évaluée à « vrai », le traitement est exécuté.

**2-Vocabulaire et syntaxe:**

Analyse	Algorithme	Pascal
[Init] <b>Si</b> condition <b>Alors</b> Instruction 1 Instruction 2 ..... Instruction N <b>FinSi</b>	..... {Init} <b>Si</b> condition <b>Alors</b> Instruction 1 Instruction 2 ..... Instruction N <b>FinSi</b>	..... ; {Init} <b>IF</b> condition <b>THEN</b> <b>Begin</b> Instruction_1; Instruction_2; .....; Instruction_N; <b>End;</b>

**Remarque :** [Init] est une séquence d'instructions qui contiendra les éventuelles initialisations.

**B-La forme alternative:**

**1-Définition:**

Une structure de contrôle conditionnelle est dite à forme alternative lorsque le traitement dépend d'une condition à deux états: Si la condition est évaluée à « vrai », le premier traitement est exécuté; traitement dépend d'une condition à deux états: Si la condition est évaluée à « faux », le second traitement est exécuté.

**2-Vocabulaire et syntaxe:**

Analyse	Algorithme
[Init] <b>Si</b> condition <b>Alors</b> Instruction 1 de TR1 Instruction 2 de TR1 ..... Instruction m de TR1 <b>Sinon</b> Instruction 1 de TR2 Instruction 2 de TR2 ..... Instruction n de TR2 <b>FinSi</b>	..... {Init} <b>Si</b> condition <b>Alors</b> Instruction 1 de TR1 Instruction 2 de TR1 ..... Instruction m de TR1 <b>Sinon</b> Instruction 1 de TR2 Instruction 2 de TR2 ..... Instruction n de TR2 <b>FinSi</b>

**Pascal**

```
..... ; {Init}
IF condition THEN
  Begin
  Instruction_1_de_TR1;
  Instruction_2_de_TR1;
  .....;
  Instruction_m_de_TR1;
  End
  ELSE
  Begin
  Instruction_1_de_TR2;
  Instruction_2_de_TR2;
  .....;
  Instruction_n_de_TR2;
  End;
```

**Remarque:** On ne met pas « ; » après le END qui précède le ELSE car la structure conditionnelle n'est pas encore terminée.

Leçon 2

**La structure de contrôle conditionnelle généralisée**

I-Définition:

Une structure de contrôle conditionnelle est dite généralisée lorsqu'elle permet de résoudre des problèmes comportant plus de deux traitements en fonction des conditions. L'exécution d'un traitement entraîne automatiquement la non exécution des autres traitements.

**2-Vocabulaire et syntaxe:**

Analyse & Algorithme	Pascal
[Init] <b>Si</b> condition 1 <b>Alors</b> Traitement 1 <b>Sinon</b> <b>Si</b> condition 2 <b>Alors</b> traitement 2 <b>Sinon</b> <b>Si</b> condition 3 <b>Alors</b> traitement 3 ..... <b>Sinon</b> <b>Si</b> condition N-1 <b>Alors</b> traitement N-1 <b>Sinon</b> traitement N <b>FinSi</b>	..... ; {Init} <b>IF</b> condition_1 <b>THEN</b> Traitement_1 <b>ELSE IF</b> condition_2 <b>THEN</b> traitement_2 <b>ELSE IF</b> condition_3 <b>THEN</b> traitement_3 ..... <b>ELSE IF</b> condition_N-1 <b>THEN</b> traitement_N-1 <b>ELSE</b> traitement_N, ..... ;

**Remarque:** il est préférable de mettre les événements les plus probables en premier lieu.

- Chaque traitement peut comporter une ou plusieurs instructions.

Leçon 3

**La structure de contrôle conditionnelle à choix**

I-Définition: Une structure de contrôle conditionnelle est dite à choix lorsque le traitement dépend de la valeur que prendra le sélecteur, Ce scalaire doit être de type scalaire(entier ou caractère).

**2-Vocabulaire et syntaxe:**

Analyse & Algorithme	Pascal
[Init] <b>selon</b> sélecteur <b>Faire</b> Valeur 1 : Action 1 Valeur 2 : Action 2-1 Action 2-2 Action 2-n Valeur 3 : Action 3  Valeur 4,valeur 6, valeur 8: Action4 Valeur 5,valeur 7,valeur 9 :Action 5 Valeur 10 .. Valeur 19 :Action 6  ..... Valeur N : Action N <b>Sinon</b> Action R <b>FinSelon</b>	<b>Case</b> selecteur <b>OF</b> Valeur_1 : Action_1 ; Valeur_2 : <b>Begin</b> Action_2_1 ; Action_2_2 ; ..... Action_2_n ; <b>End ;</b> Valeur_3 : Action_3 ; Valeur_4, valeur_6, valeur_8 :Action_4 ; Valeur_5, valeur_7, valeur_9 : Action_5 ; Valeur_10 .. Valeur_19 : Action_6 ; ..... Valeur_N : Action_N ; <b>ELSE</b> Action_R ; <b>END;</b>