

CHAPITRE

Chapitre 3

Les structures de contrôle conditionnelles

Objectifs :

- Résoudre des problèmes faisant appel aux structures de contrôle conditionnelles.
- Choisir la forme adéquate des structures de contrôle conditionnelles.
- Présenter les solutions sous forme d'un algorithme puis d'un programme.

Plan du chapitre :

Leçon 1 :
La structure conditionnelle simple

Leçon 2 :
La structure conditionnelle généralisée

Leçon 3 :
La structure conditionnelle à choix



Leçon 1

La structure de contrôle conditionnelle simple

Objectifs spécifiques :

- Résoudre des problèmes faisant appel aux structures de contrôle conditionnelles simples.
- Présenter les solutions sous forme d'un algorithme puis d'un programme.

Plan de la leçon :

I. Introduction

II. La structure conditionnelle simple

II.1. La forme réduite

II.2. La forme alternative

Retenons

Exercices

Leçon 1

La structure de contrôle conditionnelle simple

L'art est une certaine disposition, accompagnée de règle vraie, capable de produire; le défaut d'art, au contraire, est une disposition à produire accompagnée de règle fausse.

ARISTOTE

I. Introduction

Vous avez vu en 3^{ème} année que les structures de contrôle conditionnelles permettent à un ordinateur de prendre des décisions. En effet, les opérations à accomplir peuvent varier dans un algorithme selon les données fournies.

Par exemple :

- Aviser la date de validité d'un logiciel en version démonstration.
- Calculer le montant net d'une facture en fonction des remises accordées.
- Déclencher une alarme en fonction de l'état d'un détecteur.
- Donner les propriétés d'un nombre donné (pair, impair, premier ...).
- Evaluer des sorties en utilisant des fonctions logiques.
- Déterminer la nature d'un caractère tapé au clavier.

Il existe trois types de structures de contrôle conditionnelles :

- la structure conditionnelle simple
- la structure conditionnelle généralisée
- la structure conditionnelle à choix

Dans cette leçon, nous expliquerons comment écrire des algorithmes répondant à la structure de contrôle conditionnelle simple.

II. La structure de contrôle conditionnelle simple

II.1 La forme réduite

II.1.1. Vocabulaire et syntaxe

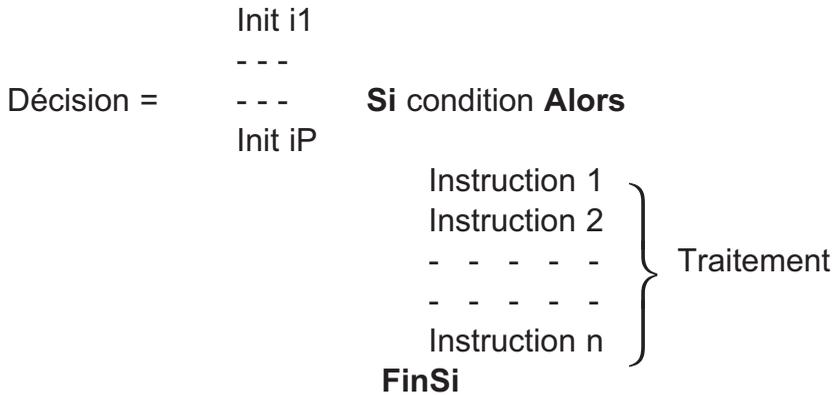
Au niveau de l'analyse

Décision = [Initialisation]	Si condition	Alors	Instruction 1	}	Ensemble d'instructions exécutées si la condition est valide
			Instruction 2		
			- - - - -		
			- - - - -		
			Instruction n		

FinSi

Remarques :

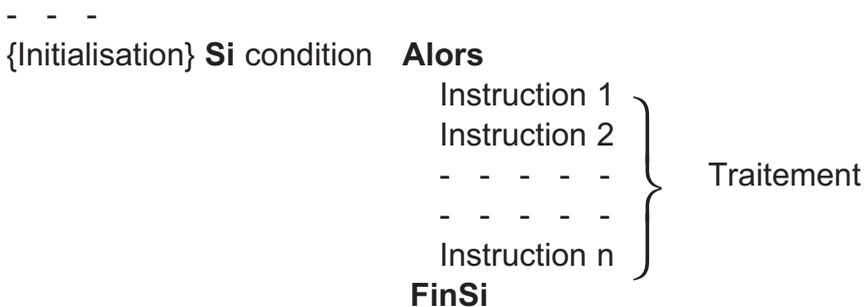
- L'initialisation et le traitement sont généralement des séquences formées de plusieurs instructions. Nous obtenons donc l'écriture suivante :



- La **condition** est une expression logique qui prend soit la valeur « VRAI » soit la valeur « FAUX ».
- Si la condition est « VRAI », les instructions entre Alors et FinSi sont exécutées.
- La condition peut être une condition composée de plusieurs autres propositions logiques liées par les opérateurs booléens.

Exemple : Not (n>2) OU (x>3) ET (x<50)

Au niveau de l'algorithme



Traduction en Pascal

```

- - - ;
Inst1; InstP; {le corps d'initialisation}
IF condition THEN
    BEGIN
        Instruction i1;
        Instruction i2;
        - - - - ;
        - - - - ;
        Instruction in;
    } END;
    } Traitement
- - - ;
    
```

Activité 1

On se propose d'écrire un programme intitulé REGULATEUR qui affiche le message "Régulateur déclenché" et produit un bip sonore si la température n'est pas comprise entre 18 °C et 24 °C.

Analyser ce problème et en déduire l'algorithme correspondant.

NB. On rappelle que CHR(7) produit un bip sonore.

Analyse

Nom = REGULATEUR		
S	L.D.E.	O.U.
3	Résultat = Ecrire (Bip,Message)	Bip Message Temp
2	(Bip,Message) = [Bip ← "", Message ← "Régulateur non déclenché"] Si (Temp < 18) ou (Temp > 24) Alors Message ← " Régulateur déclenché" Bip ← CHR(7) FinSi	
1	Temp = Donnée ("Entrer la valeur de la température : ")	
4	Fin REGULATEUR	

Tableau de déclaration des objets

Objet	Type / Nature	Rôle
Bip	Caractère	Produit un bip sonore
Message	Chaîne de caractères	Sauvegarde l'état de l'alarme
Temp	Entier	Saisie la température

Algorithme

0) Début REGULATEUR

1) Écrire ("Entrer la valeur de la température : "), Lire (Temp)

2) [Bip ← "", Message ← "Régulateur non déclenché"]

 Si (Temp < 18) ou (Temp > 24) Alors

 Message ← "Régulateur déclenché"

 Bip ← chr(7)

 FinSi

3) Ecrire (Bip,Message)

4) Fin REGULATEUR

II.1.2. 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.

Activité 2

Traduire l'algorithme REGULATEUR en Pascal.

Traduction en Pascal

```
PROGRAM REGULATEUR;
USES WINCRT;
VAR
  Temp : INTEGER;
  Message : STRING;
  Bip : CHAR;

BEGIN
WRITE ('Entrer la valeur de la température : ');
READLN(Temp);
Bip := '';
Message := 'Régulateur non déclenché';
IF (Temp < 18) OR (Temp > 24)
  THEN
    BEGIN
      Message := 'Régulateur déclenché';
      Bip := Chr(7);
    END;
WRITE(Bip,Message);
END.
```

Activité 3

On se propose d'écrire un programme intitulé CLASSEMENT qui saisit 2 chaînes de caractères puis les affiche en commençant par la plus longue. Si elles sont de même longueur la première saisie sera affichée en premier.

Questions :

- 1) Analyser ce problème et en déduire l'algorithme correspondant.
- 2) Traduire l'algorithme CLASSEMENT en Pascal.

Tableaux de déclaration des objets

Nom = CLASSEMENT		
S	L.D.E.	O.U.
4	Résultat = Ecrire (Chaine1, Chaine2)	Chaine1 Chaine2
3	(Chaine1, Chaine2) = [Chaine1 ← Ch1, Chaine2 ← Ch2] Si (LONG(Ch2) > LONG(Ch1)) Alors Chaine1 ← Ch2 Chaine2 ← Ch1 FinSi	Ch1 Ch2
2	Ch2 = Donnée ("Entrer la Chaîne 2 : ")	
1	Ch1 = Donnée ("Entrer la Chaîne 1 : ")	
5	Fin CLASSEMENT	

Tableau de déclaration des objets

Objet	Type / Nature	Rôle
Chaine1	Chaîne de Caractères	Sauvegarde la plus petite chaîne
Chaine2	Chaîne de Caractères	Sauvegarde la plus longue chaîne
Ch1	Chaîne de Caractères	Saisie la première chaîne
Ch2	Chaîne de Caractères	Saisie la deuxième chaîne

Algorithme

- 1) Début CLASSEMENT
- 2) Ecrire ("Entrer la Chaîne 1 : "), Lire (Ch1)
- 3) Ecrire ("Entrer la Chaîne 2 : "), Lire (Ch2)
- 4) [Chaine1 ← Ch1, Chaine2 ← Ch2] Si (LONG(Ch2) > LONG(Ch1)) Alors
 Chaine1 ← Ch2
 Chaine2 ← Ch1
 FinSi
- 4) Ecrire (Chaine1, " ", Chaine2)
- 5) Fin CLASSEMENT

Traduction en Pascal

```
PROGRAM CLASSEMENT;
USES WINCRT;
VAR
    Chaine1, Chaine2, Ch1, Ch2 : STRING;
```

```

BEGIN
  WRITE ('Entrer la Chaîne 1 : '); READLN(Ch1);
  WRITE ('Entrer la Chaîne 2 : '); READLN(Ch2);
  Chaine1 := Ch1;
  Chaine2 := Ch2;

  IF (LONG(Ch2) > LONG(Ch1)) THEN
    BEGIN
      Chaine1 := Ch2;
      Chaine2 := Ch1;
    END;

  WRITE (Chaine1, ' ', Chaine2);
END.

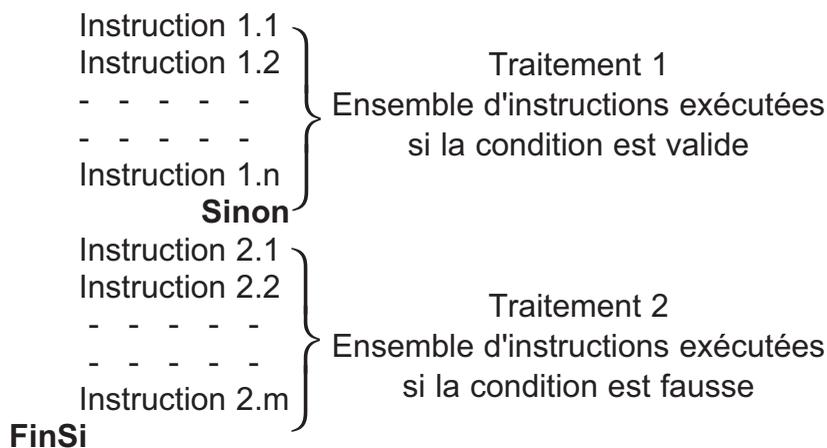
```

II.2 La forme alternative

II.2.1. Vocabulaire et syntaxe

Au niveau de l'analyse et de l'algorithme

Décision = [Initialisation] **Si** condition **Alors**



Remarques :

- Lorsque l'évaluation de la condition produit la valeur :
VRAI seules les instructions du traitement 1 sont exécutées.
FAUX seules les instructions du traitement 2 sont exécutées.
- Chaque traitement peut comporter une ou plusieurs instructions.
- Les instructions du traitement 1 sont délimitées entre « Alors » et « Sinon » et celles du traitement 2 le sont entre « Sinon » et « Finsi ».

Traduction en Pascal

```

- - - ;
Inst1; InstP;  {le corps d'initialisation}
IF condition THEN
    BEGIN
        Instruction i11;
        Instruction i12;
        - - - - ;
        - - - - ;
        Instruction i1n;
    END;
        ELSE
            BEGIN
                Instruction i21;
                Instruction i22;
                - - - - ;
                - - - - ;
                Instruction i2m;
            END;
- - - ;

```

} Traitement 1

} Traitement 2

Remarque :

On ne met pas de ";" après le END qui précède le ELSE car la structure conditionnelle n'est pas encore terminée.

Activité 4

Un interrupteur de courant électrique a deux états ON-OFF (1, 0). Selon le montage, un circuit électrique à deux interrupteurs peut réaliser les deux fonctions logiques ET, OU.

Faire une analyse, écrire un algorithme puis la traduction en Pascal du programme intitulé FN_LOGIQUE_OU qui, suivant l'état des interrupteurs K et L, affiche l'état de la lampe LED.

Le principe d'une porte logique OU peut se résumer comme suit : Deux entrées et une sortie. Suivant le niveau logique appliqué aux entrées (niveau 0 ou 1) la sortie sera 0 ou 1.

Fonction	symbole	Table de vérité	Circuit électrique																		
OU (OR)		<table border="1"> <thead> <tr> <th colspan="3">Fonction OU (OR)</th> </tr> <tr> <th>Entrée K</th> <th>Entrée L</th> <th>Sortie</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Fonction OU (OR)			Entrée K	Entrée L	Sortie	0	0	0	0	1	1	1	0	1	1	1	1	
Fonction OU (OR)																					
Entrée K	Entrée L	Sortie																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	1																			

Analyse

Nom = FN_LOGIQUE_OU		
S	L.D.E.	O.U.
4	Résultat = Ecrire (LED)	LED
3	LED = [] Si (K=0) ET (L=0) Alors LED ← "Éteinte" Sinon LED ← "Allumée" FinSi	K L
2	L = Donnée ("Entrer l'état de l'interrupteur L : ")	
1	K = Donnée ("Entrer l'état de l'interrupteur K : ")	
5	Fin FN_LOGIQUE_OU	

Tableau de déclaration des objets

Objet	Type / Nature	Rôle
LED	Chaîne de Caractères	Sauvegarde l'état de la LED
K	Entier	Saisie l'état de l'interrupteur K
L	Entier	Saisie l'état de l'interrupteur L

Algorithme

- 0) Début FN_LOGIQUE_OU
- 1) Ecrire ("Entrer l'état de l'interrupteur K: "), Lire (K)
- 2) Ecrire ("Entrer l'état de l'interrupteur L : "), Lire (L)
- 3) Si (K=0) ET (L=0) Alors
LED ← " Éteinte "
Sinon
LED ← " Allumée "
FinSi
- 4) Ecrire (LED)
- 6) Fin FN_LOGIQUE_OU

Traduction en Pascal

```

PROGRAM FN_LOGIQUE_OU;
USES WINCRT;
VAR
  K,L: INTEGER;
  LED: STRING;

BEGIN
  WRITE ('Entrer l''état de l''interrupteur K : ');READLN(K);
  WRITE ('Entrer l''état de l''interrupteur L : ');READLN(L);
  IF (K=0)AND(L=0) THEN
    LED := 'Éteinte'
  ELSE
    LED := 'Allumée';

  WRITE (LED);
END.

```

II.2.2 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 ; si la condition est évaluée à « FAUX », le second traitement est exécuté.

Activité 5

Écrire un programme qui demande à l'utilisateur de saisir 4 valeurs réelles correspondants aux coordonnées de 2 vecteurs $\vec{u}(x_1, y_1)$ et $\vec{v}(x_2, y_2)$ et qui détermine et affiche si ces vecteurs sont orthogonaux entre eux ou pas.

Analyser ce problème et en déduire l'algorithme correspondant.

NB. Deux vecteurs sont orthogonaux si leur produit scalaire est égal à 0.

Pré-analyse

Soient $\vec{u}(x_1, y_1)$ et $\vec{v}(x_2, y_2)$ dans une base orthonormée de l'ensemble des vecteurs du plan.

\vec{u} perpendiculaire à \vec{v} équivaut à $x_1 \cdot x_2 + y_1 \cdot y_2 = 0$.

Analyse

Nom = ORTHOGONAL		
S	L.D.E.	O.U.
4	Résultat = Ecrire (Nature)	Nature
3	Nature= [] Si $x1*x2 + y1*y2=0$ Alors Nature←" u est perpendiculaire à v " Sinon Nature←" u n'est pas perpendiculaire à v " FinSi	x1 x2 y1 y2
2	x2,y2 = Donnée ("Entrer les coordonnés du vecteur v: ")	
1	x1,y1 = Donnée ("Entrer les coordonnés du vecteur u: ")	
5	Fin ORTHOGONAL	

Tableau de déclaration des objets

Objet	Type / Nature	Rôle
Nature	Chaîne	Nature des deux vecteurs
x1,x2,y1,y2	Réel	Coordonnées des vecteurs u et v

Algorithme

0) Début ORTHOGONAL

1) Ecrire ("Entrer les coordonnés du vecteur u: "), Lire(x1,y1)

2) Ecrire ("Entrer les coordonnés du vecteur v: "), Lire(x2,y2)

3) Si $x1*x2 + y1*y2=0$

 Alors

 Nature←" u est perpendiculaire à v "

 Sinon

 Nature←" u n'est pas perpendiculaire à v "

 FinSi

4) Ecrire (Nature)

5) Fin ORTHOGONAL

Activité 6

Traduire l'algorithme ORTHOGONAL en Pascal.

Traduction en Pascal

```

PROGRAM ORTHOGONAL;
USES WINCRT;
VAR
  Nature:STRING;
  X1,y1,X2,Y2:REAL;

BEGIN
  WRITE('Entrer les coordonnées du vecteur u: ');READLN(x1,y1);
  WRITE('Entrer les coordonnées du vecteur u: ');READLN(x2,y2);

  IF x1*x2 + y1*y2=0
    THEN
      Nature := 'u est perpendiculaire à v'
    ELSE
      Nature := 'u n''est pas perpendiculaire à v';

  WRITE (Nature);
END.

```

Retenons

La structure de contrôle conditionnelle simple est utilisée pour définir au maximum deux résultats dépendant de l'évaluation d'une seule condition.

- 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.
- Une structure de contrôle conditionnelle a une forme alternative si suivant la valeur d'une condition, on exécute soit un traitement 1 soit un traitement 2.

La structure de contrôle conditionnelle simple	Au niveau de l'analyse et de l'algorithme	Au niveau de Pascal
La forme réduite	[Initialisation] Si condition Alors Traitement FinSi	- - -; If condition then Traitement;
La forme alternative	[Initialisation] Si condition Alors Traitement1 Sinon Traitement2 FinSi	- - -; If condition then Traitement1; Else Traitement 2;

Exercices

Pour chacun des exercices suivants, effectuer une analyse, dresser un algorithme et donner la traduction en Pascal.

Exercice 1

Lire un nombre et afficher sa racine carrée si elle existe.

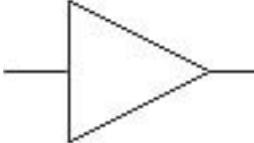
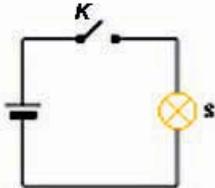
Exercice 2

Demander un caractère à l'utilisateur et tester si ce caractère est majuscule ou minuscule.

Exercice 3

Soit le tableau suivant :

Afficher l'état de la lampe S (allumée ou éteinte) suivant l'état de l'interrupteur K.

Fonction	symbole	Table de vérité	Circuit électrique						
OUI		<table border="1"> <thead> <tr> <th>Entrée K</th> <th>Sortie</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> </tr> </tbody> </table>	Entrée K	Sortie	0	1	0	1	
Entrée K	Sortie								
0	1								
0	1								

Exercice 4

Ordonner dans le sens décroissant, trois entiers saisis au clavier.

Exercice 5

Saisir une chaîne de caractères et vérifier si elle est composée de plusieurs mots.

Exercice 6

Lire l'année et afficher si elle est éventuellement bissextile.

NB. Si l'année n'est pas divisible par 4, l'année n'est pas bissextile.

Exercice 7

L'université décide d'attribuer un e-mail à ses étudiants selon le modèle suivant : Nom@Campus.net

Il y a cependant une restriction, si le nom excède sept caractères, il doit être tronqué.

Ainsi l'étudiant « MohamedAli » aura comme e-mail: Mohamed@campus.net

Demander le nom de l'étudiant et afficher son e-mail.

Exercice 8

On veut décrire le comportement d'un contrôleur pour une porte d'entrée par carte. Ce contrôleur peut recevoir une indication de lecture d'une carte autorisée (a) ou refusée (r).

Nous avons les contraintes suivantes:

- Si une carte est refusée, il faut sonner immédiatement.
- Lorsqu'une carte est autorisée, la porte s'ouvre.

On se propose de faire un programme intitulé **CONTROLEUR** qui lit l'état d'un lecteur de carte et affiche le message "Porte en cours d'ouverture" si la carte est autorisée, sinon produit un bip sonore si la carte est refusée.

Exercice 9

Un chef d'entreprise décide d'automatiser les horaires de début de travail, de pause et de fin de travail. Le système consiste à déclencher une alarme et à afficher un message à des heures bien précises de la journée.

On se propose de faire un programme intitulé **MINUTERIE** qui lit l'heure maintenue par le système d'exploitation et affiche le message "Reprise du travail" et produit deux bips sonores si l'horaire est équivalent à 7 Heures et 30 Minutes ou 14 Heures, sinon affiche le message "Arrêt du travail" et produit un seul bip sonore si l'horaire est équivalent à 12 Heures ou 17 Heures et 15 Minutes.

NB.

- On rappelle que l'écriture de **CHR(7)** produit un bip sonore.
- La procédure prédéfinie **GETTIME** renvoie l'heure maintenue par le système d'exploitation.

Syntaxe: **procedure GETTIME**(var Heure, Minute, Seconde, Sec100: Word);

Description: les intervalles des valeurs renvoyées sont 0..23 pour heure, 0..59 pour minute, 0..59 pour Seconde et Sec100(centièmes de secondes).

Exercice 10

Saisir un entier de trois chiffres et afficher si cet entier est :

- Cube ou non (la somme des cubes de ses chiffres est égale à lui même).
- Pair - impair
- Positif - négatif

Leçon 2

La structure de contrôle conditionnelle généralisée

Objectifs spécifiques

- Résoudre des problèmes faisant appel à la structure de contrôle conditionnelle généralisée
- Présenter les solutions sous forme d'un algorithme puis d'un programme

Plan de la leçon

I. Introduction

II. Vocabulaire et syntaxe

III. Définition

Retenons

Exercices

Leçon 2

La structure de contrôle conditionnelle generalisee

La logique est l'étude des conditions formelles de la vérité
 PIAGET

I. Introduction

Nous avons vu dans la leçon précédente qu'une structure conditionnelle simple conduit à deux alternatives uniquement.

Dans certains problèmes faisant appel à la structure de contrôle conditionnelle, deux situations ne suffisent pas, d'où l'intérêt de la structure de contrôle conditionnelle généralisée qui résout des problèmes dans lesquels plusieurs alternatives sont envisageables.

II. Vocabulaire et syntaxe

Au niveau de l'analyse et de l'algorithme

```
Résultat = [Initialisation ] Si condition 1 Alors traitement 1
                               Sinon Si condition 2 Alors traitement 2
                               Sinon Si condition 3 Alors traitement 3
                               - - - - -
                               Sinon Si condition n-1 Alors traitement n-1
                               Sinon traitement n
                               FinSi
```

Remarques :

- Lorsque l'évaluation de la condition 1 produit la valeur : **VRAI**, le traitement 1 sera exécuté puis l'exécution continue avec l'instruction qui suit le **Si** généralisé.
FAUX, la condition 2 sera évaluée, si elle produit la valeur : **VRAI**, le traitement 2 sera exécuté puis l'exécution continue avec l'instruction qui suit le **Si** généralisé.
FAUX, la condition 3 sera évaluée et ainsi de suite.

- Si aucune des N-1 premières conditions ne produit la valeur VRAI, c'est le traitement N qui sera exécuté.
- Chaque traitement peut comporter une ou plusieurs instructions.
- Si la partition des conditions est exhaustive, il existe une condition et une seule qui est vraie.
- Il est préférable de mettre les événements les plus probables en premier lieu.

Au niveau de Pascal

```
- - - ;
{Initialisation}
IF condition 1 THEN traitement 1
    ELSE IF condition 2 THEN traitement 2
    ELSE IF condition 3 THEN traitement 3
        - - -
    ELSE IF condition N-1 THEN traitement N-1
    ELSE traitement N ;
- - - ;
```

Activité 1

Effectuer une analyse, écrire un algorithme et faire la traduction en Pascal du programme intitulé NATURE_TRIANGLE, qui à partir de trois points donnés A, B et C détermine et affiche la nature du triangle (isocèle en A, isocèle en B, isocèle en C, équilatéral, quelconque).

Pré-analyse

Lorsqu'un triangle ABC est tel que $AC = AB$ (les deux côtés d'extrémité A sont égaux), alors on dit que le triangle est isocèle de sommet A. A est le sommet principal du triangle. Le côté [BC], opposé à A, est appelé base du triangle.

Si un triangle a trois côtés de même longueur, il est dit équilatéral.

Dans le plan cartésien, les points sont définis à l'aide de leurs coordonnées dites cartésiennes.

Soient deux points A et B dans le plan cartésien. On appelle (x_A, y_A) les coordonnées du point A et (x_B, y_B) les coordonnées du point B. La distance AB dans le plan vaut :

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Analyse

Nom = NATURE_TRIANGLE		
S	L.D.E.	O.U.
8	Résultat = Ecrire (Nature)	Nature
7	Nature = [] Si (AC=AB) et (AB=BC) Alors Nature←"Triangle équilatéral" Sinon Si AB=BC Alors Nature←"Triangle isocèle en B" Sinon Si AC=BC Alors Nature←"Triangle isocèle en C" Sinon Si AB= AC Alors Nature←"Triangle isocèle en A" Sinon Nature←"Triangle quelconque" FinSi	AB AC BC
6	BC← RacineCarré (Carré(Xc-Xb)+ Carré(Yc-Yb))	
5	AB← RacineCarré (Carré(Xb-Xa)+ Carré(Yb-Ya))	Xc
4	AC← RacineCarré (Carré(Xc-Xa)+ Carré(Yc-Ya))	Xb
3	Xc,Yc = Donnée ("Entrer les coordonnés du point C: ")	Yc
2	Xa,Yb = Donnée ("Entrer les coordonnés du point B: ")	Yb
1	Xa,Ya = Donnée ("Entrer les coordonnés du point A: ")	Xa Ya
9	Fin NATURE_TRIANGLE	

Tableau de déclaration des objets

Objet	Type/nature	Rôle
Nature	Chaîne	Sauvegarde la nature du triangle
AB,AC,BC	Réel	Longueur des côtés du triangle
Xc,Xb,Yc,Yb,Xa,Ya	Réel	Saisie les coordonnés des points A, B et C

Algorithme

- 0) Début NATURE_TRIANGLE
- 1) Ecrire ("Entrer les coordonnés du point A: "), Lire(Xa,Ya)
- 2) Ecrire ("Entrer les coordonnés du point B: "), Lire(Xb,Yb)
- 3) Ecrire ("Entrer les coordonnés du point C: "), Lire(Xc,Yc)
- 4) AC←RacineCarré (Carré(Xc-Xa)+ Carré(Yc-Ya))
- 5) AB←RacineCarré (Carré(Xb-Xa)+ Carré(Yb-Ya))
- 6) BC←RacineCarré (Carré(Xc-Xb)+ Carré(Yc-Yb))

```

7) Si (AC=AB) et (AB=BC) Alors
    Nature←"Triangle équilatéral"
Sinon Si AB=BC Alors
    Nature←"Triangle isocèle en B"
Sinon Si AC=BC Alors
    Nature←"Triangle isocèle en C"
Sinon Si AB = AC Alors
    Nature←"Triangle isocèle en A"
Sinon
    Nature←"Triangle quelconque"

FinSi
8) Ecrire (Nature)
9) Fin NATURE_TRIANGLE

```

Traduction en Pascal

```

PROGRAM NATURE_TRIANGLE;
USES WINCRT;
VAR
    Nature:STRING;
    AC,AB,BC:REAL;
    Xc,Xb,Xa,Yb,Yc,Ya:REAL;

BEGIN
WRITE('Entrer les coordonnés du point A: '); READLN(Xa,Ya);
WRITE('Entrer les coordonnés du point B: '); READLN(Xb,Yb);
WRITE('Entrer les coordonnés du point C: '); READLN(Xc,Yc);
AC:= SQRT(SQR(Xc-Xa)+ SQR(Yc-Ya));
AB:= SQRT(SQR(Xb-Xa)+ SQR(Yb-Ya));
BC:= SQRT(SQR(Xc-Xb)+ SQR(Yc-Yb));
IF (AC=AB)and(AB=BC)THEN
    Nature:='Triangle équilatéral'
ELSE IF AB=BC THEN
    Nature:='Triangle isocèle en B'
ELSE IF AC=BC THEN
    Nature:='Triangle isocèle en c'
ELSE IF AC=AB THEN
    Nature:='Triangle isocèle en A'
ELSE
    Nature:='Triangle quelconque';
WRITE(Nature);
END.

```

III. 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.

Activité 2

Dans le but de déterminer la nature d'une solution chimique on utilise un pH_mètre.

Une solution peut être :

- Acide fort si le pH est inférieur à 2
- Acide faible si le pH est entre 2 et 6
- Neutre si le pH est égal à 7
- Base faible si le pH est entre 8 et 12
- Base forte si le pH est supérieur à 13

Pré-analyse

On constate que le pH_mètre ne peut prendre que des valeurs entières, on pourra le considérer comme une variable de type entier.

La solution chimique peut être Acide, Neutre ou Basique.

Analyse

Nom = SOLUTION		
S	L.D.E.	O.U.
3	Résultat = Ecrire (Nature)	Nature pH
2	Nature= [] Si (pH<2) Alors Nature←"Acide fort" Sinon Si pH<7 Alors Nature←"Acide faible" Sinon Si pH=7 Alors Nature←"Neutre" Sinon Si pH<13 Alors Nature←"Base faible" Sinon Nature←"Base forte" FinSi	
1	pH = Donnée ("Entrer le pH de la solution: ")	
4	Fin SOLUTION	

Tableau de déclaration des objets

Objet	Type/nature	Rôle
Nature	Chaîne	Sauvegarde de la nature de la solution
pH	Entier	Saisie le pH d'une solution

Algorithme

```

0) Début SOLUTION
1) pH = Donnée ("Entrer le pH de la solution: ")
2) Si pH<2 Alors
    Nature←"Acide fort"
    Sinon Si pH<7 Alors
        Nature←"Acide faible"
    Sinon Si pH=7 Alors
        Nature←"Neutre"
    Sinon Si pH<13 Alors
        Nature←"Base faible"
    Sinon
        Nature←"Base forte"
    FinSi
3) Ecrire (Nature)
4) Fin SOLUTION
    
```

Retenons

La structure de contrôle conditionnelle généralisée offre la possibilité de résoudre des problèmes comportant plus de deux traitements. Une fois qu'une condition est évaluée et qu'un traitement est exécuté les autres ne le seront pas. Cette économie sur les tests se répercute sur le temps d'exécution du programme.

	Au niveau de l'analyse et de l'algorithme	Au niveau de Pascal
La structure de contrôle conditionnelle généralisée	[Initialisation] Si cond1 Alors trait1 Sinon Si cond2 Alors trait2 Sinon Si cond3 Alors trait3 - - - - - Sinon Si cond n-1 Alors trait n-1 Sinon trait n FinSi	- - - ; IF cond THEN trait1 ELSE IF cond2 THEN trait2 ELSE IF cond2 THEN trait3 - - - - - ELSE IF cond n-1 THEN trait n-1 ELSE trait n ;

Exercices

Pour chacun des exercices suivants, effectuer une analyse, écrire un algorithme et faire la traduction en Pascal.

Exercice 1

Lire le nom du port de communication et afficher le débit correspondant.

Port E/S	Débit
Blue tooth	1 Mbits/S
Infrarouge	12 Mbps
IEEE1394	100 - 400 Mbits/s
USB1	12 Mbits/s
USB2	480 Mbits/s

Exercice 2

Écrire un programme qui affiche ce menu :

1. Effectuer une multiplication
2. Effectuer une addition
3. Effectuer une soustraction
4. Effectuer une division

Puis saisit un réel x_1 , le choix de l'opération et un deuxième réel x_2 , puis affiche le résultat de l'application de l'opérateur choisi sur x_1 et x_2 .

prévoir un message d'erreur au cas où une entrée incorrecte est faite.

Exercice 3

Saisir deux entiers x et y , le premier en décimal et le deuxième en binaire et vérifier si x est supérieur, inférieur ou égal à y puis affiche le résultat.

Exercice 4

Écrire un programme Pascal intitulé **PROTOCOLE** qui lit le nom d'un service internet, puis affiche le protocole correspondant.

Exercice 5

Le conseil scientifique d'une institution effectue un vote pour décider de l'achat de matériels informatiques.

On vous demande d'écrire un programme qui affiche la décision à prendre par le conseil sachant qu'elle est :

« Reportée si le pourcentage des neutres est strictement supérieur à 50%, sinon elle est « acceptée » si le pourcentage des favorables est strictement supérieur à celui des défavorables et « refusée » dans le cas contraire.

Exercice 6

Lire le type d'un fichier et afficher les extensions possibles.

Exemple :

Type de fichier : Image

Extensions possibles : BMP / TIFF / GIF

Leçon 3

La structure de contrôle conditionnelle à choix

Objectifs spécifiques :

- Résoudre des problèmes faisant appel à la structure de contrôle à choix
- Présenter les solutions sous forme d'un algorithme puis d'un programme Pascal

Plan de la leçon :

I. Introduction

II. Vocabulaire et syntaxe

III. Définition

Retenons

Exercices

Leçon 3

La structure de contrôle conditionnelle à choix

Est nécessaire la proposition qui, étant vraie, n'est pas capable d'être fausse, ou bien celle qui en est capable, mais qui est telle que les circonstances extérieures s'opposent à ce qu'elle soit fausse.

LA PLEIADE

I. Introduction

Si la structure conditionnelle généralisée est une structure à plusieurs traitements en fonction des conditions, la structure conditionnelle à choix est en revanche un choix multiple lorsque plusieurs voies sont possibles selon les différentes valeurs que peut contenir une variable de type scalaire (sélecteur).

Cette structure rend le code source du programme plus lisible.

Une combinaison entre la structure conditionnelle généralisée et la structure conditionnelle à choix multiple est possible.

Activité 1

Pour un entier N donné, lequel de ces nombres est multiple de 5 : N , $N + 1$, $N + 2$, $N + 3$ et $N + 4$.

Effectuer une analyse, écrire un algorithme et faire la traduction en Pascal du programme intitulé MULTIPLE5, qui saisit un entier N et détermine lequel parmi ces nombres est multiple de 5.

Pré-analyse

Pour vérifier que le premier entier N est multiple de cinq, il suffit de montrer que le reste de la division entière de N par 5 vaut 0.

Si le reste est différent de 0 il suffit d'ajouter à N la valeur $(5 - \text{reste})$ pour que l'entier $(N + 5 - \text{reste})$ soit multiple de 5.

Analyse

Nom = MULTIPLE5		
S	L.D.E.	O.U.
4	Résultat = Ecrire (M)	M
3	M = [] Selon R Faire 0 : M ← "N" 1 : M ← "N +4" 2 : M ← "N +3" 3 : M ← "N +2" 4 : M ← "N +1" Fin Selon	R N
2	R ← N MOD 5	
1	N = Donnée ("Entrer un entier N : ")	
4	Fin MULTIPLE5	

Tableaux de déclaration des objets

Objet	Type/nature	Rôle
M	Chaîne de Caractères	Sauvegarde du résultat
R	Réel	Contient le reste
N	Réel	Saisit de l'entier N

Algorithme

```

0) Début MULTIPLE5
1) Ecrire ("Entrer un entier N : "), Lire(N)
2) R ← N MOD 5
3) Selon R Faire
    0 : M ← "N"
    1 : M ← "N +4"
    2 : M ← "N +3"
    3 : M ← "N +2"
    4 : M ← "N +1"
    Fin Selon
4) Ecrire (M )
5) Fin MULTIPLE5
    
```

Traduction en Pascal

```

PROGRAM MULTIPLE5;
USES WINCRT;
VAR
    N,R : INTEGER;
    M: STRING;
    
```

```

BEGIN
WRITE ('Entrer un entiere N : '); READLN(N);
R:=N MOD 5;
CASE R OF
  0 : M:='N';
  1 : M:='N+4';
  2 : M:='N+3';
  3 : M:='N+2';
  4 : M:='N+1';
END;
WRITE (M);
END.

```

II. Vocabulaire et syntaxe

Au niveau de l'analyse et de l'algorithme

[Initialisation] **Selon** sélecteur **Faire**

Valeur 1 : Traitement 1

Valeur 2 : Traitement 2

Valeur 3 : Traitement 3

Valeur 4, Valeur 5, Valeur 6 : Traitement 4

Valeur 7 .. Valeur 10 : Traitement 5

Valeur N : Traitement N

Sinon traitement R

Fin Selon

Commentaire :

- Le sélecteur doit être de type scalaire discret et non réel.
- Plusieurs valeurs séparées par des virgules, auront un même traitement.
- Un intervalle de valeurs peut entraîner un même traitement.
- Un traitement peut comporter une ou plusieurs instructions.
- Si la valeur du sélecteur est différente des valeurs proposées alors le traitement de la clause **Sinon** sera exécuté.
- La partie **Sinon** est facultative.

Traduction en Pascal

```

- - - ;
CASE selecteur OF
  Valeur 1 : Traitement 1;
  Valeur 2 : Traitement 2;
  Valeur 3 : Traitement 3;
  Valeur 4, Valeur 5, Valeur 6 : Traitement 4;
  Valeur 7 .. Valeur 10 : Traitement 5;
  - - -
  Valeur N : Traitement N;
ELSE traitement R;
END;

```

III. Définition

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

Activité 2

Effectuer une analyse, écrire un algorithme et faire la traduction en Pascal du programme intitulé JOUR, qui saisit une date sous la forme jj/mm/aaaa (chaîne de 10 caractères) où jj représente le jour, mm le mois, aaaa l'année et détermine le jour de la semaine.

Parmi plusieurs possibilités, on peut utiliser la congruence de Zeller dont le résultat est un entier compris entre 0 et 6. (0 signifie samedi, 1 dimanche, 2 lundi, 3 mardi, 4 mercredi, 5 jeudi, et 6 vendredi).

Formule de Zeller :

$$J = [jj + \text{ENT}(2,6 * M - 0,2) + (aaaa \bmod 100) + \text{ENT}((aaaa \bmod 100)/4) + \text{ENT}((aaaa \text{ div } 100)/4) - 2 * (aaaa \text{ div } 100)] \bmod 7.$$

Où : - jj est le jour du mois.

- M est le mois en utilisant la numération Romaine (1→mars, 2→avril, ... 10→décembre, 11→janvier, ..).
- aaaa représente une année donnée en commençant par 1582.

Exemple : pour la date du 10/11/2006, J = 5 alors le jour de la semaine est le vendredi.

Pré-analyse

Formule de Zeller :

$$J = [jj + \text{ENT}(2,6 * M - 0,2) + (aaaa \bmod 100) + \text{ENT}((aaaa \bmod 100)/4) + \text{ENT}((aaaa \text{ div } 100)/4) - 2 * (aaaa \text{ div } 100)] \bmod 7.$$

Où - ENT désigne la partie entière.

- JJ correspond à un numéro du jour du mois mm de l'année aaaa.
- M représente le rang du mois avec la numération Romaine.
- (aaaa div 100) et (aaaa mod 100) sont mis respectivement pour avoir les deux premiers et les deux derniers chiffres de l'année.

Les jours de la semaine sont numérotés selon leur rang, sauf que le samedi correspond à zéro.

Exemple :

Le 20 août 1970 était un jeudi, car J est égal à 96 et le reste de la division par 7 est 5.

Voici le détail des calculs :

$$J = 20 + [2,6(8 + 1)] + [19/4] - 38 + 70 + [70/4]$$

$$J = 20 + 23 + 4 - 38 + 70 + 17 = 96$$

Si J est négatif, on fait $(7n + k)$ où n est choisi de façon à ce que la valeur de l'expression varie de 0 à 6. Par exemple, si $J = -23$, on fait $28 - 23 = 5$.

Analyse

Nom = JOUR		
S	L.D.E.	O.U.
12	Résultat = Ecrire (Jour)	Jour
11	Jour = [] Selon J Faire 0 : Jour ← "Dimanche" 1 : Jour ← "Lundi" 2 : Jour ← "Mardi" 3 : Jour ← "Mercredi" 4 : Jour ← "Jeudi" 5 : Jour ← "Vendredi" 6 : Jour ← "Samedi" SiNon Jour ← "ERREUR" Fin Selon	J JJ M aaaa mm an E mois day date
10	J = [] Si J < 0 Alors J ← -J mod 7 J ← 7 - J FinSi	
9	$J \leftarrow (JJ + \text{TRONC}((2.6 * M) - 0.2) + (aaaa \bmod 100) + \text{TRONC}((aaaa \bmod 100) / 4) + \text{TRONC}((aaaa \text{ div } 100) / 4) - 2 * (aaaa \text{ div } 100)) \bmod 7$	
8	Selon mm Faire 3 : M ← 1 4 : M ← 2 5 : M ← 3 6 : M ← 4 7 : M ← 5 8 : M ← 6 9 : M ← 7 10 : M ← 8 11 : M ← 9 12 : M ← 10 1 : M ← 11 2 : M ← 12 Fin Selon	
7	Valeur(An,aaaa,e)	
6	Valeur(mois,mm,e)	
5	Valeur(day,jj,e)	
4	an ← Sous-chaîne(Date,7,4)	
3	mois ← Sous-chaîne(Date,4,2)	
2	day ← Sous-chaîne(Date,1,2)	
1	date = Donnée ("Entrer une date : ")	
13	Fin JOUR	

Tableaux de déclaration des objets

Objet	Type / Nature	Rôle
J	Entier	Calcul du jour par la formule de Zeller
JJ	Entier	Contient le jour
mm	Entier	Contient le mois
aaaa	Entier	Contient l'année
M	Entier	Contient le mois en Romain
E	Entier	Erreur
Jour	Chaîne de caractères	Sauvegarde du jour
date	Chaîne de caractères	Saisie de la date
day	Chaîne de caractères	Contient le jour
mois	Chaîne de caractères	Contient le mois
an	Chaîne de caractères	Contient l'année

Algorithme

0) Début JOUR

1) Ecrire ("Entrer une date: "), Lire(Date)

2) day ← Sous-chaîne(Date,1,2)

3) mois ← Sous-chaîne(Date,4,2)

4) an ← Sous-chaîne(Date,7,4)

5) Valeur(day,jj,e)

6) Valeur(mois,mm,e)

7) Valeur(an,aaaa,e)

8) Selon mm Faire

3 : M ← 1

4 : M ← 2

5 : M ← 3

6 : M ← 4

7 : M ← 5

8 : M ← 6

9 : M ← 7

10 : M ← 8

11 : M ← 9

12 : M ← 10

1 : M ← 11

2 : M ← 12

Fin Selon

9) $J \leftarrow (JJ + \text{TRONC}((2.6 * M + 1) + (aaaa \bmod 100)) + \text{TRONC}((aaaa \bmod 100) / 4) + \text{TRONC}((aaaa \text{ div } 100) / 4) - 2 * (aaaa \text{ div } 100)) \bmod 7$

10) Si $J < 0$ Alors

$J \leftarrow -J \bmod 7$

$J \leftarrow 7 - J$

FinSi

11) Selon J Faire

0 : Jour ← "Dimanche"

1 : Jour ← "Lundi"

2 : Jour ← "Mardi"

3 : Jour ← "Mercredi"

4 : Jour ← "Jeudi"

5 : Jour ← "Vendredi"

6 : Jour ← "Samedi"

SiNon Jour ← "ERREUR"

Fin Selon

12) Ecrire (Jour)

13) Fin JOUR

Traduction en Pascal

PROGRAM JOUR;

USES WINCRT;

VAR

jour, date, day, mois, an: **STRING**;

jj, mm, aaaa, M, e, J: **INTEGER**;

BEGIN

WRITE('Entrer une date: '); **READLN**(date);

day:=**COPY**(date, 1, 2);

mois:= **COPY**(date, 4, 2);

an:= **COPY**(date, 7, 4);

Val(day, jj, e);

Val(mois, mm, e);

Val(an, aaaa, e);

CASE mm **OF**

3 : M :=1;

4 : M :=2;

5 : M :=3;

6 : M :=4;

7 : M :=5;

8 : M :=6;

9 : M :=7;

10: M :=8;

11: M :=9;

12: M :=10;

1 : M :=11;

2 : M :=12;

END;

Exercices

Pour chacun des exercices suivants, effectuer une analyse, écrire un algorithme et faire la traduction en Pascal.

Exercice 1

Écrire un programme intitulé TOUCHE, qui affiche selon le cas, la nature du caractère (consonne, voyelle, chiffre ou symbole) correspondant à une touche saisie. On considère que le clavier est verrouillé en minuscule.

Exercice 2

Lire un code d'ADN sous forme d'une chaîne de trois caractères, puis déterminer et afficher le code d'ARN correspondant. Sachant que le code d'ADN utilise les lettres A, T, et G et le code d'ARN est obtenu par la correspondance suivante : A→U, T→A, C→G et G→C.

Exercice 3

Saisir le prix unitaire d'un produit et la quantité commandée, affiche le prix à payer, sachant que :

- la remise est de 3% si le montant net est compris entre 1000 D et 3000 D
- la remise est de 5% si le montant net est compris entre 3001 D et 5000 D
- la remise est de 7% si le montant net est supérieur à 5000 D

Exercice 4

Effectuer le cryptage d'un caractère C donnée en utilisant le principe de cryptage suivant :

- si le caractère est une voyelle alors la fonction $f(x)=X-1$
 - si le caractère est une consonne alors la fonction $f(x)=X-2$
 - si le caractère est un chiffre alors la fonction $f(x)=X+1$
 - si le caractère est un symbole alors la fonction $f(x)=X+2$
- X étant le code ASCII du caractère.

Exercice 5

Afficher le jour selon un nombre entré par l'utilisateur (1 = LUNDI, 2 = MARDI, etc.). Votre programme devra détecter les jours erronés.

Exemple :

Entrez un jour : 1

Vous avez choisi LUNDI

Exercice 6

Dans cet exercice, nous nous intéressons aux ordinaux abrégés en anglais, ou le nombre est écrit en chiffres. Les premiers sont :

« 1st », « 2nd », « 3rd », « 4th », etc. (abbreviation de « first, second, third, fourth, ... »).

Pour déterminer le suffixe, on regarde le dernier chiffre du nombre :

- si c'est 1, on ajoute le suffixe est -st
- si c'est 2, le suffixe est -nd
- si c'est 3, le suffixe est -rd
- sinon le suffixe est -th.
- si l'avant-dernier chiffre du nombre est 1, le suffixe est toujours -th.

Écrire un programme qui lit un nombre, et qui affiche l'ordinal anglais abrégée correspondant.

Exercice 7

Écrire un programme intitulé JOURS qui pour une année et un mois donnée affiche le nombre de jours de ce mois.