

Cours Algorithmme : Conditions

Les Commentaires

- Pour faciliter la lecture de l'algorithme, on peut ajouter des commentaires.
- Pour écrire un commentaire, on utilise le symbole `//`, ce qui permet à l'algorithme d'ignorer cette ligne lors de son exécution.

LES TESTS: LES STRUCTURES CONDITIONNELLES

Tests: Instructions Conditionnelles

- Les instructions conditionnelles servent à n'exécuter une instruction ou une séquence d'instructions que si une condition est vérifiée.
- Une condition représente une comparaison entre valeurs de même type.
- Cette condition retourne une valeur booléenne.

Tests: Instructions Conditionnelles

Si (condition) **alors**
instruction ou suite
d'instructions1

Sinon
instruction ou suite
d'instructions2

Finsi

- La condition ne peut être que vraie ou fausse
- Si la condition est vraie, se sont les instructions1 qui seront exécutées
- Si la condition est fausse, se sont les instructions2 qui seront exécutées
- La condition peut être une condition simple ou une condition composée de plusieurs conditions

Tests: Instructions Conditionnelles

- La partie Sinon n'est pas obligatoire, quand elle n'existe pas et que la condition est fausse, aucun traitement n'est réalisé.
- On utilisera dans ce cas la forme simplifiée suivante:

Si (condition) alors

instruction ou suite d'instructions1

Finsi

Conditions Composées

- Certains problèmes exigent parfois de formuler des conditions qui ne peuvent pas être exprimées sous une forme simple.
- Une condition composée est une condition formée de plusieurs conditions simples reliées par des opérateurs logiques ET, OU, OU exclusif (XOR) et NON
- Exemples :
 - x compris entre 2 et 6 :
 - $(x > 2) \text{ ET } (x < 6)$
 - n divisible par 3 ou par 2 :
 - $(n \% 3 = 0) \text{ OU } (n \% 2 = 0)$
- L'évaluation d'une condition composée se fait avec les tables de vérité.

Tests: Instructions Conditionnelles

- La condition est composée de trois éléments :
 - Une valeur
 - Un opérateur de comparaison
 - Une autre valeur

- Les opérateurs de comparaison sont :

Égal à	=
Différents de	<>
Plus petit, plus grand	< ; >

Exemple 1

- Ecrire un algorithme qui permet d'afficher la valeur absolue d'un entier.

Exemple 1: Solution

- Ecrire un algorithme qui permet d'afficher la valeur absolue d'un entier.

Algorithme valeur_absolue;

Variable x : réel;

Début

Ecrire("Entrez un réel: ");

Lire (x);

Si ($x < 0$) alors

 Ecrire ("la valeur absolue de ", x, "est:", -x);

Sinon

 Ecrire ("la valeur absolue de ", x, "est:", x);

Finsi

Fin

Exemple 2

- Ecrire un algorithme qui demande un nombre entier à l'utilisateur, puis le teste et affiche s'il est divisible par 3.

Exemple 2: Solution

Algorithme Divisible_par3 ;

Variable n : entier;

Début

Ecrire ("Entrez un entier: ");

Lire (n);

Si ($n \% 3 = 0$) alors

 Ecrire (n, " est divisible par 3");

Sinon

 Ecrire (n, " n'est pas divisible par 3");

Finsi

Fin

Tests Imbriqués

- Les tests peuvent avoir un degré quelconque d'imbrications, où deux possibilités restent insuffisantes.

```
SI (condition1) ALORS
  instructions
  SINON SI (condition2) ALORS
    instructions
    SINON SI (condition3)
      ALORS
        instructions
      SINON
        Instructions
      FIN SI
    FIN SI
  FIN SI
```

```
Si (condition1) alors
  Si (condition2) alors
    instructionsA
  Sinon
    instructionsB
  Finsi
Sinon
  Si (condition3) alors
    instructionsC
  Finsi
Finsi
```

Exemple 1

- Ecrire un algorithme qui permet de vérifier la nature d'un réel saisi au clavier.

Exemple 1 : Solution

Algorithme Type;

Variable n : réel;

Début

Ecrire ("entrez un nombre: ");

Lire (n);

Si ($n < 0$) alors

 Ecrire("Ce nombre est négatif");

Sinon

 Si ($n = 0$) alors

 Ecrire ("Ce nombre est nul");

 Sinon

 Ecrire ("Ce nombre est positif");

 Finsi

Finsi

Fin

Exemple 2

- Afficher « Reçu avec mention » si une note est supérieure ou égale à 12, « Passable » si elle est supérieure à 10 et inférieure à 12, et « Insuffisant » dans tous les autres cas

Exemple 2: Solution

Algorithme Classification;

Variable note : entier ;

Début

Ecrire ("note: ") ;

Lire (note) ;

Si (note >= 12) Alors

 écrire ("Reçu avec mention") ;

 Sinon Si (note >= 10) alors

 écrire ("Passable") ;

 Sinon

 écrire ("Insuffisant") ;

 Finsi

Finsi

Fin

Exemple 3: Quel résultat affiche cet algorithme?

Variable age : Entier;

Début

Écrire("Entrez l'âge de l'enfant:");

Lire(age);

Si(age >= 12) alors

 écrire("catégorie Cadet");

 SinonSi (age >= 10) alors

 écrire ("Catégorie Minime");

 SinonSi (age >= 8) alors

 écrire ("Catégorie pupille");

 SinonSi (age >= 6) alors

 écrire ("Catégorie Poussin");

 FinSi

 FinSi

 FinSi

FinSi

Fin

Sélection à choix multiple

- On utilise la structure conditionnelle « SELON ».
- Elle est souvent utilisée pour le cas des menus.

Selon (choix)

Début

```
cas 1 : instruction 1;  
      break;
```

```
...
```

```
cas p : instruction p;  
      break;
```

```
default : instruction;
```

FinSelon

À Suivre