Les instructions conditionnelles

Programmation de base sur PC

Plan du cours

- L’instruction if
- L’instruction else
- La structure d’un switch
- Exercices 2

L’instruction if-else

Les instructions conditionnelles

- Nous allons régulièrement avoir besoin de faire des opérations en fonction d'un résultat précédent.
- Par exemple, lorsque nous nous connectons à une application avec notre nom d’utilisateur et notre mot de passe, par la suite on souhaite afficher l’écran principal.
- Il s'agit de ce que l'on appelle une condition. Elle est évaluée lors de l'exécution et en fonction de son résultat (vrai ou faux) nous ferons telle ou telle chose.

Les opérateurs de comparaison

Opérateur	Description
==	Egalité
!=	Différence
>	Supérieur à
<	Inférieur à
>=	Supérieur ou égal
<=	Inférieur ou égal
&&	ET logique
||	OU logique
!	Négation

- Une condition se construit grâce à des **opérateurs de comparaison**. On dénombre plusieurs opérateurs de comparaisons.
- Voici, à gauche, un tableau avec les plus courants.

L’instruction « if »

- L’instruction if permet d’exécuter du code si une condition est vraie (if = si en anglais).
- L’instruction optionnelle « else » exécutera du code si le « if » est faux

```
if( condition )
{
  //exécuter le code si VRAI
}
else
{
  //exécuter le code si FAUX
}
```

- De cette façon, un seul des deux blocs peut être exécuté à la fois, selon que la condition soit vérifiée ou non.

Écrire une condition

Question

En initialisant les variables *a*, *b*, *lettre* et *car* de la façon suivante :

```
int a = 3, b = 5 ;
char lettre = 'i', car = 'j' ;
```

examinez si les conditions suivantes sont vraies ou fausses :

```
(a != b)
(a + 2 == b)
(a + 8 < 2 * b)
(lettre <= car)
(lettre == 'w')
```

Réponse

La condition `(a != b)` est vraie car 3 est différent de 5.

La condition `(a + 2 == b)` est vraie car 3 + 2 vaut 5.

La condition `(a + 8 < 2 * b)` est fausse car 3 + 8 est plus grand que 2 * 5.

La condition `(lettre <= car)` est vraie car le caractère `'i'` est placé avant `'j'` dans l'ordre alphabétique.

La condition `(lettre == 'w')` est fausse car le caractère `'i'` est différent du caractère `'w'`.

Écrire une condition

Question

En initialisant les variables `x`, `y`, `z` et `r` de la façon suivante :

```
int x = 3, y = 5, z = 2, r = 6 ;
```

examinez si les conditions suivantes sont vraies ou fausses :

```
(x < y) && (z < r)
(x > y) || (z < r)
!(z < r)
```

Réponse

Sachant que la condition `(x < y) && (z < r)` est vraie si les deux expressions `(x < y)` **et** `(z < r)` sont toutes les deux vraies et devient fausse si l'une des deux expressions est fausse, l'expression donnée en exemple est vraie. En effet `(3 < 5)` est vraie et `(2 < 6)` est vraie.

Sachant que la condition `(x > y) || (z < r)` est vraie si l'une des expressions `(x > y)` **ou** `(z < r)` est vraie et devient fausse si les deux expressions sont fausses, l'expression donnée en exemple est vraie car `(3 > 5)` est fausse, mais `(2 < 6)` est vraie.

Sachant que la condition `!(z < r)` est vraie si l'expression `(z < r)` est fausse et devient fausse si l'expression est vraie, alors l'expression donnée en exemple est fausse car `(2 < 6)` est vraie.

Mise-en-oeuvre

Programmez l'algorithme suivant permettant de trouver le plus grand de deux nombres entiers.

```
int first, second, max;

Console.WriteLine("Veuillez entrer le premier nombre entier");
first = int.Parse(Console.ReadLine());
Console.WriteLine("Veuillez entrer le deuxième nombre entier");
second = int.Parse(Console.ReadLine());

if(first > second)
{
  max = first;
} else
{
  max = second;
}

Console.WriteLine("La plus grande valeur est " + max);
```

Des if-else imbriqués

- Il est possible d'imbriquer les if les uns dans les autres

Imbrications d'`if else`	Représentation du choix arborescent

```
if (Condition 1)
{
  if (Condition 2)
  {
   instruction A
  }
  else
  {
   instruction B
  }
}
else
{
  instruction C
}
```

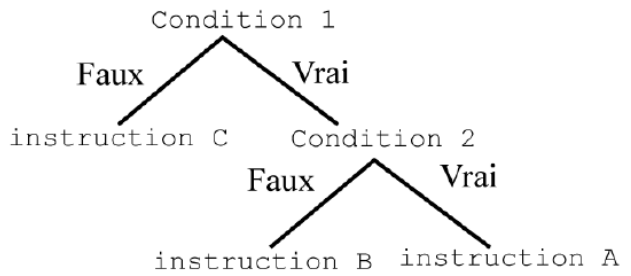

Figure 3-1

Des if-else imbriqués

- Une instruction « if » peut ne pas contenir de « else », voir condition 2

```
if (Condition 1)
{
  if (Condition 2)
  {
    if (Condition 3)
    {
      instruction A
    }
    else
    {
      instruction B
    }
  }
}
else
{
  instruction C
}
```

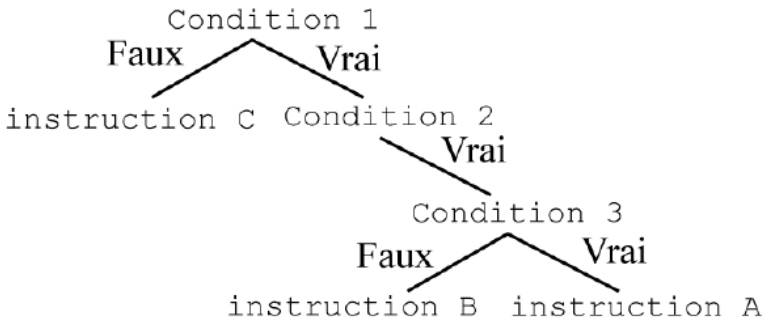

Figure 3-2

Mise-en-oeuvre

Programmez l'algorithme suivant permettant d'afficher la parité.

```
int input;
Console.WriteLine("Veuillez entrer un chiffre ou un nombre entier :");
input = int.Parse(Console.ReadLine());
if (input < 10)
{
        if (input % 2 == 0)
                Console.WriteLine(input + " est un CHIFFRE PAIR");
        else
                Console.WriteLine(input + " est un CHIFFRE IMPAIR");

else
{
        if (input % 2 == 0)
                Console.WriteLine(input + " est un NOMBRE PAIR");
        else
                Console.WriteLine(input + " est un NOMBRE IMPAIR");
}
```

L’instruction Switch

On imbrique à l'infinie ?

- Que faire s'il y a plusieurs choix possibles ? Imbriquer, imbriquer, imb... ?

```
String month;
if (value == 1)
{
  month = "janvier";
}
else
{
  if (value == 2)
  {
    month = "février";
  }
  else
  {
    if (value == 3)
    {
      month = "mars";
    }
```

```
    else {
      if (value == 4)
      {
        month = "avril";
      }
      else {
        if (value == 5)
        {
          month = "mai";
        }
        else
        {
          //[...] Sans fin...
        }
      }
    }
  }
}
```

Exemple du switch

- Nous allons plutôt utiliser la structure du « switch »

```
int value = 2;

switch (value)
{
  case 1:
    // Une ou plusieurs instructions quand value = 1
    break;
  case 2:
  case 3:
    // Une ou plusieurs instructions quand value = 2 ou 3
    break;
  default:
    // Une ou plusieurs instructions pour les autres valeurs
  break;
}
```

Fonctionnement du « switch »

- La variable « value » est évaluée
- Le programme recherche l'étiquette correspondant dans les instructions **`case`**
  - Si le programme trouve une étiquette correspondant au contenu de la variable « value », il exécute la ou les instructions qui suivent l'étiquette, jusqu'à rencontrer le mot-clé **`break`**
  - S'il n'existe pas d'étiquette correspondant à valeur , alors le programme exécute les instructions de l'étiquette **`default`** .
- S'il n'y a pas de `break` pour une étiquette donnée, le programme exécute les instructions de l'étiquette suivante également

Mise-en-oeuvre

Écrivez un programme qui demande un numéro de mois et qui affiche le nombre de jour de ce mois. Présumons que le mois de février est toujours de 28 jours.

La sortie à la console doit être exactement comme suit :

```
Entrez un numéro de mois [1 à 12] : 3
Le mois #3 contient 31 jours
```

```
Entrez un numéro de mois [1 à 12] : 13
Numéro de mois invalide
```

DÉFI: Tentez maintenant de prendre en compte les années bisextilles en demandant aussi l'année du mois de février voulu.

Exercices