



Cégep de Saint-Hyacinthe
Département d'informatique

Programmation orientée objet

420-2DP-HY

(3-3-3)

Les énumérations

(Version 1.2)

1 heure

Préparé par

Martin Lalancette

Comprendre les éléments suivants:

- Qu'est-ce qu'une énumération?
- Déclaration
- Portée
- Utilisation

Table des matières

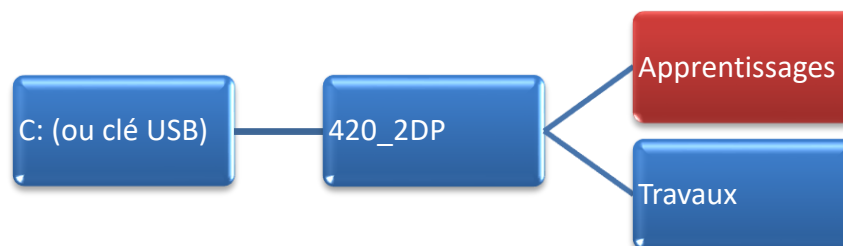
Introduction.....	3
Préparation de base	3
Qu'est-ce qu'une énumération?	3
Déclaration d'une énumération	5
Utilisation d'un type énumération	6
Conversion (« casting ») de la valeur d'une variable de type énumération	7
Utilisation d'une énumération avec une condition (if, while, do...while)	7
Utilisation d'une énumération dans une expression (switch).....	8
Bibliographie.....	9

Introduction

Cette séquence a pour but de vous initier aux notions de base nécessaires à l'introduction de la programmation. Nous commencerons par énoncer les éléments théoriques appuyés d'exemples simples et faciles à reproduire. Afin d'axer l'attention sur la compréhension de ces notions, il y aura des exercices à faire tout au long de cette séquence.

Préparation de base

Pour bien suivre les instructions qui vont être mentionnées tout au long des séquences d'apprentissage, une préparation de base s'impose. Il est important de créer un répertoire de travail (sur votre C: ou clé USB). Voici une suggestion d'arborescence:



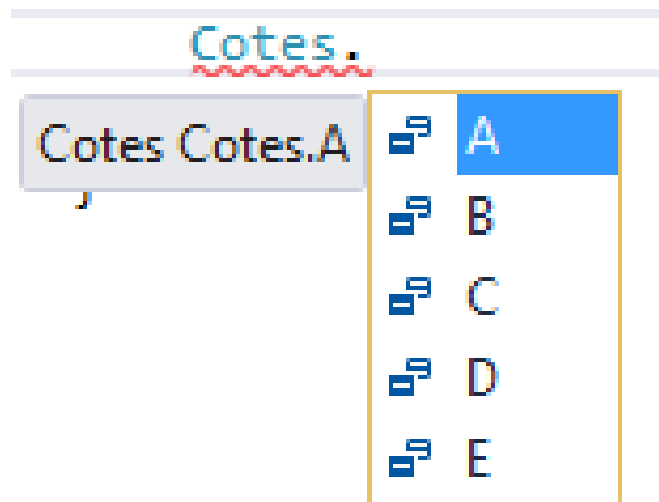
Préparation : S'assurer d'avoir créé l'arborescence ici haut mentionnée sur votre C ou votre clé USB. Copier ce document dans le répertoire en rouge ici haut mentionné.

Qu'est-ce qu'une énumération?

En langage de programmation, ***une énumération est un type de données contenant un ensemble de valeurs nommées***. Ces valeurs peuvent être appelées : éléments, membres ou énumérateurs. Les noms attribués à ces énumérateurs sont habituellement des identifiants qui se comporteront comme des constantes. Autre expression: Type énuméré.

À savoir :

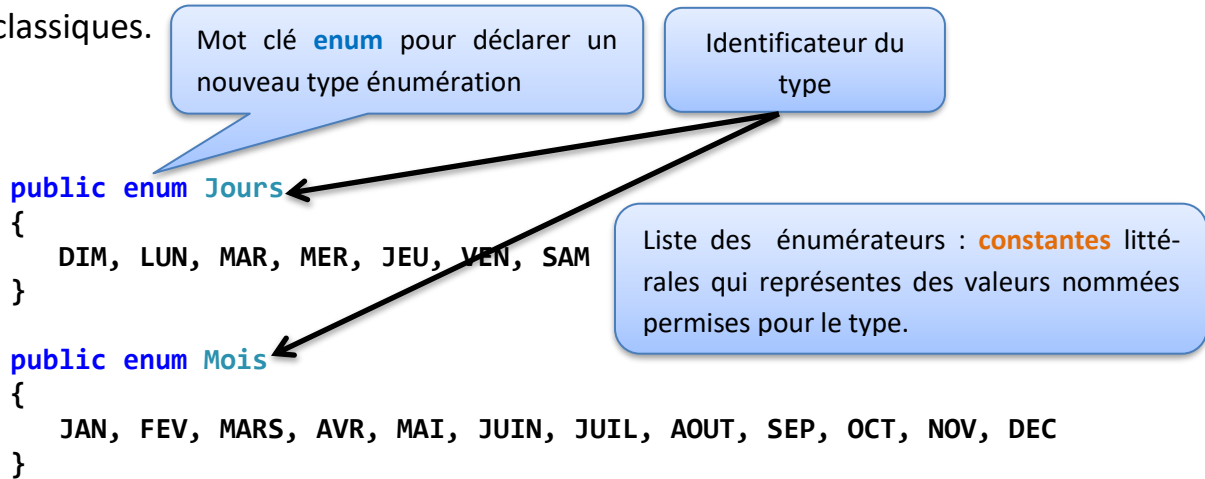
- Le mot clé **enum** est utilisé pour déclarer une énumération, c'est-à-dire un type distinct constitué d'un ensemble de constantes nommées que l'on appelle « liste d'énumérateurs » (MSDN, 2013).
- Généralement, il vaut mieux définir directement un *enum* dans un espace de noms afin que toutes les classes de cet espace puissent y accéder aussi aisément. Toutefois, un enum peut également être imbriqué dans une classe ou un struct (MSDN, 2013).
- Un type énumération possède un identificateur unique.
- Par défaut, le premier énumérateur a la valeur 0, et chaque énumérateur suivant a la valeur n+1 (MSDN, 2013).



Préparation : Créer la solution **Les énumérations** et le projet de type console **Exercice01**.

Déclaration d'une énumération

Voici maintenant comment déclarer une énumération avec des exemples classiques.



Une valeur entière est associée à chaque énumérateur. Par défaut, la valeur 0 est attribuée au 1^{er} énumérateur. Les énumérateurs se verront attribuer des incréments de 1. Exemple :

```

public enum Jours
{
    DIM, // = 0
    LUN, // = 1
    MAR, // = 2
    MER, // = 3
    JEU, // = 4
    VEN, // = 5
    SAM  // = 6
}

```

Valeurs entières par défaut.

Exercice 1. : Définir les énumérations **Jours** et **Mois**. Déclarer une variable de type **Jours** nommée **jours** et une variable **mois** de type **Mois**.

Nous pouvons aussi attribuer une valeur entière à chacun des énumérateurs de la liste. Les valeurs n'ont pas à être contiguës, ascendantes ou descendantes. Exemples :

```
public enum Jours
{
    DIM = 1,
    LUN, // = 2
    MAR, // = 3
    MER, // = 4
    JEU, // = 5
    VEN, // = 6
    SAM // = 7
}

public enum Cotes
{
    E = 50,
    D = 60,
    C = 70,
    B = 80,
    A = 90
}

public enum Jeton
{
    BLANC = 1,
    ROUGE = 5,
    BLEU = ROUGE * 2,
    VERT = ROUGE * 5,
    NOIR = VERT * 4
}
```

Une valeur entière peut aussi être attribuée au moyen d'un calcul avec les opérateurs standards.

À savoir :

- Une valeur entière doit être du type **byte**, **sbyte**, **short**, **ushort**, **int**, **uint**, **long** ou **ulong** (sauf char).
- Une valeur réelle devra être explicitement convertie en valeur entière.
- Un champ ou une propriété auto-implémentée est **initialisé par défaut avec le 1er énumérateur** de la liste des énumérateurs.

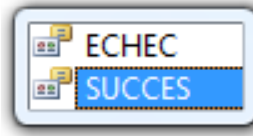
Utilisation d'un type énumération

Avant d'utiliser le type énumération, il y a quelques consignes à savoir:

- Pas d'instanciation (donc aucun new)
- Pas de static
- Les énumérateurs sont des littéraux (= identificateurs de constantes)
- Les valeurs associées sont de types intégraux (entiers, ou calculs simples dont le résultat est un entier)
- L'accès aux énumérateurs est fait par l'entremise de l'identificateur de l'énumération. Ex. **Enumeration.ENUMERATEUR** ou **Classe.EnumerationImbriquee.ENUMERATEUR**

- Les variables sont déclarées avec ce type. L'accès à un énumérateur commence par l'identificateur de l'énumération. Exemple:

```
Cotes cotePierre = Cotes.A; // 90
Cotes coteJean   = Cotes.B; // 80
Cotes coteJacques = Cotes.C; // 70
Etats resultat = Etats.
```



Conversion (« casting ») de la valeur d'une variable de type énumération

Vous pouvez convertir le contenu d'une variable de type énumération vers une variable de type numérique en utilisant l'opération de « casting ».

Exemple :

```
Cotes cotePierre = Cotes.B;
int iCotePierre = 0;
byte byCotePierre = 0;
float fCotePierre = 0;

// Conversion énumérateur vers numérique
iCotePierre = (int)cotePierre;
byCotePierre = (byte)cotePierre;
fCotePierre = (float)cotePierre;

// Conversion numérique vers énumérateur
cotePierre = (Cotes)iCotePierre;
cotePierre = (Cotes)byCotePierre;
cotePierre = (Cotes)fCotePierre;
```

Utilisation d'une énumération avec une condition (if, while, do...while)

```
Cotes cotePierre = Cotes.A; // 90
Cotes coteJean   = Cotes.B; // 80
Cotes coteJacques = Cotes.C; // 70
Etats resultat = Etats.SUCCES;

// Différents tests logiques
if( cotePierre == Cotes.A ) { /* vrai...*/ }
if( cotePierre >= Cotes.B ) { /* vrai...*/ }
if( cotePierre == coteJean ) { /* faux...*/ }
if( coteJacques.ToString() == "C" ) { /* vrai...*/ }
if( coteJacques.ToString() == "c" ) { /* faux...*/ }
```

Utilisation d'une énumération dans une expression (switch)

```
1.  // Sélection selon une expression
2.  switch( cotePierre)
3.  {
4.      case Cotes.E:
5.          resultat = Etats.ECHEC;
6.          break;
7.      case Cotes.D:
8.      case Cotes.C:
9.      case Cotes.B:
10.     case Cotes.A:
11.         resultat = Etats.SUCCES;
12.         break;
13. }
```

Une façon facile d'autogénérer le code d'un *switch* à partir d'une énumération est d'utiliser un « snippet ». Voici comment procéder :

1. Il faut au préalable déclarer une variable du type de l'énumération.
2. Saisir le mot **sw** et faire **deux fois Tabulation**.
3. Saisir la variable associée et faire **Tabulation**.
4. Faire **Entrée**, pour générer le code complet.

Exercice 2. : Définir une énumération qui traite des breuvages. Voici les valeurs possibles : COKE, SEVENUP, JUS_POMMES, THE_GLACE, LAIT. Demander à l'utilisateur de saisir une valeur entre 1 et 5 et afficher l'énumérateur associé.

Exercice 3. : Définir une énumération qui associe les noms des doigts en sachant qu'une main contient : POUCE, INDEX, MAJEUR, ANNULAIRE, AURICULAIRE. Demander les infos à l'utilisateur.

Exercice 4. : Définir une énumération qui contient les trois couleurs primaires soit : ROUGE, VERT, BLEU. Demander à l'utilisateur de saisir une valeur entre 1 et 3 et afficher l'énumérateur associé.

Bibliographie

Mastriani, R. (2013, 03 04). PowerPoint - Enums. Saint-Hyacinthe, Québec, Canada.

MSDN. (2013, 01 01). *enum (référence C#)*. Consulté le 03 04, 2013, sur MSDN:
[http://msdn.microsoft.com/fr-fr/library/vstudio/sbbt4032\(v=vs.110\).aspx](http://msdn.microsoft.com/fr-fr/library/vstudio/sbbt4032(v=vs.110).aspx)