

Le Langage Java

1ère année

M. Bastreggi J. Beleho P. Bettens M. Codutti
A. Hallal C. Leruste D. Nabet N. Pettiaux A. Rousseau

Haute École de Bruxelles — École Supérieure d'Informatique

Année académique 2011 / 2012

Leçon 3

Les énumérations

- Le concept d'énumération
- Implémentation traditionnelle
- L'énumération en Java
- Attributs d'une énumération
- Méthodes d'une énumération

Présentation

Enumération \equiv ensemble fixe et petit de valeurs ayant un nom et sémantiquement liées

Exemples

- ▶ Saisons : Été, Automne, Hiver, Printemps
- ▶ Couleurs carte : Coeur, Carreau, Pique, Trèfle
- ▶ État civil : Célibataire, Marié, Divorcé, Veuf/Veuve
- ▶ ...

Implémentation traditionnelle

Traditionnellement implémenté via des constantes entières

Exemple :

```
final int SAISON_ETE = 1;  
final int SAISON_AUTOMNE = 2;  
final int SAISON_HIVER = 3;  
final int SAISON_PRINTEMPS = 4;
```

Implémentation traditionnelle

Pose de nombreux problèmes

- ▶ Pas de type \implies déclaration peu explicite
`int saison ;`
- ▶ Pas d'obligation d'utiliser les noms
`saison = 1; // Illisible mais valide`
- ▶ Pas de vérification de la valeur
`saison = 5; // Pas de sens mais accepté`
- ▶ Affichage non explicite
`System.out.println (saison); // Affiche un entier`

Un cas concret

Examinons la javadoc de `GregorianCalendar`

- ▶ Constantes définies : `HOUR`, `MONTH`, `DAY_OF_MONTH`, ...
- ▶ Utilisées dans la méthode `get()`

```
GregorianCalendar date = ...;  
date.get(GregorianCalendar.HOUR); // Pour obtenir les heures  
date.get(10); // Idem ligne ci-dessus  
date.get(100); // Compilation ok mais erreur à l'exécution !!!  
date.get(Integer.MAX_VALUE); // Idem !!!
```

Une vraie énumération

À partir de 1.5 : mot clé `enum`

- ▶ Définit un nouveau type de données
- ▶ **Exemple** :
`enum Saison {ETE, AUTOMNE, HIVER, PRINTEMPS}`
- ▶ Au même niveau que `class`
- ▶ Défini dans un fichier à part ou au sein d'une autre classe

Une vraie énumération

Exemple d'utilisation

```
public class Test {  
    enum Saison {ETE, AUTOMNE, HIVER, PRINTEMPS}  
    // Notez l'absence de ; !  
  
    public static void main(String args []) {  
        Saison saison = Saison.ETE;  
        System.out.println (saison); // Ecrit ETE  
        saison = 1; // Erreur de compilation;  
    }  
}
```


Une vraie énumération

Variante :

- ▶ Un fichier `Saison.java`

```
public enum Saison {ETE, AUTOMNE, HIVER, PRINTEMPS}
```

- ▶ Et un programme qui l'utilise

```
public class Test {  
    public static void main(String args []) {  
        Saison saison = Saison.ETE;  
        System.out.println (saison); // Écrit ETE  
        saison = 1; // Erreur de compilation;  
        if (saison == Saison.HIVER)  
            System.out.println ("A_gla_gla!");  
    }  
}
```

Test de la valeur

Le `switch` peut être utilisé avec une énumération

Exemple :

```
// Fichier Titre.java  
enum Titre {MONSIEUR, MADAME, MADEMOISELLE}
```

```
void afficherNom( Titre titre , String nom ) {  
    switch( titre ) {  
        // Remarquer l'absence du 'Titre.'  
        case MONSIEUR : System.out.println("M. " + nom); break;  
        case MADAME : System.out.println("Mme " + nom); break;  
        case MADEMOISELLE : System.out.println("Mlle " + nom); break;  
    }  
}
```

Itérer sur les valeurs

On peut obtenir un tableau de toutes les valeurs
(idéal pour itérer)

► **Exemple :**

```
Saison[] saisons = Saison.values();  
for(int i=0; i<saisons.length; i++) {  
    System.out.println ( saisons[ i ] );  
}
```

► Ou encore grâce au **foreach**

```
for( Saison saison : Saison.values() ) {  
    System.out.println ( saison );  
}
```

Attributs

On peut ajouter des **attributs** à une énumération

- ▶ Très proche d'une classe
- ▶ **Exemple** (fichier `Mois.java`) :

```
public enum Mois {  
    // Valeurs de l'enum  
    // Comme un appel de constructeur.  
    JANVIER(31), FEVRIER(28), MARS(31); // \e...  
    // Attribut  
    public final int NB_JOURS;  
    // Constructeur (pas public !)  
    private Mois(int nbJours) { NB_JOURS = nbJours; }  
}
```

Attributs

Utilisation des attributs

```
public class Test {  
    public static void main(String args[]) {  
        for( Mois mois : Mois.values() ) {  
            System.out.println (mois + " : " + mois.NB_JOURS);  
        }  
    }  
}
```

Méthodes

On peut aussi ajouter ses propres méthodes

► **Exemple**

```
public enum Mois {  
    JANVIER(31), FEVRIER(28), MARS(31);  
    private int nbJours;  
    private Mois(int nbJours) {  
        this.nbJours = nbJours;  
    }  
    public int getNbJours() {  
        return nbJours;  
    }  
}
```

Crédits

Ce document a été produit avec les outils suivants

- ▶ La distribution **Ubuntu** du système d'exploitation **Linux**
- ▶ **LaTeX** comme système d'édition
- ▶ La classe **Beamer** pour les transparents
- ▶ Les packages **listings**, **fancyvrb**, ...
- ▶ Les outils **make**, **rubber**, **pdfnup**, ...