Énumération

Arnaud Labourel arnaud.labourel@univ-amu.fr

15 mars 2022

Section 1

Les énumérations

Énumérations en Java (1/2)

En programmation, une *énumération* est un type spécial qui permet de définir des variables pouvant prendre des valeurs parmi un ensemble prédéfini de constantes.

Il est possible de définir des énumérations en Java grace au mot-clé `enum` :

```
enum Suit{
  SPADES,
  HEARTS,
  DIAMONDS,
  CLUBS;
}
```

Énumérations en Java (2/2)

```
enum Suit {
  SPADES, HEARTS, DIAMONDS, CLUBS;
}
```

Une énumération est une classe avec des éléments prédéfinis et statiques.

On peut donc tester directement l'égalité avec l'opérateur égal car il n'y qu'un objet et donc qu'une référence qui correspond à chaque valeur possible.

```
Suit suit = Suit.SPADES;
/* ... */
if (suit == Suit.SPADES)
{
  /* .... */
}
```

Définition d'attributs, de méthodes et d'un constructeur

```
public enum Suit {
  SPADES("Pique", "Pi"), HEARTS("Cœur", "Co"),
  DIAMONDS("Carreau", "Ca"), CLUBS("Trèfle", "Tr");

  private final String frenchName;
  private final String frenchSymbol;

  Suit(String name, String symbol) {
    this.frenchName = name;
    this.frenchSymbol = symbol;
  }
  public String frenchName() { return frenchName; }
  public String frenchSymbol() { return frenchSymbol; }
}
```

Classe `Enum`

Toutes les énumération de java étendent la classe `Enum`.

Quelques méthodes utiles :

- `String name()` : retourne le nom de la constante tel que déclaré dans l'`enum`.
- `int ordinal()` : retourne la position de la constante dans la déclaration de l'`enum` (commençant par 0).

Tout `enum` définit aussi une méthode statique `values()` renvoyant un tableau contenant les constantes dans l'ordre de leurs déclarations.

Exemple d'utilisation d'énumération (1/2)

```
public static void main(String[] args) {
  Suit[] values = Suit.values();
  for (Suit suit : values)
  System.out.printf("Le symbole de %s est %s.\n",
          suit.frenchName(), suit.frenchSymbol());
}
```

Affichage :

```
Le symbole de Pique est Pi.
Le symbole de Cœur est Co.
Le symbole de Carreau est Ca.
Le symbole de Trèfle est Tr.
```

Exemple d'utilisation d'énumération (2/2)

```
public static void main(String[] args) {
  for (Suit suit : Suit.values())
  System.out.printf("The position of %s is %d.\n",
                    suit.name(), suit.ordinal());
}
```

Affichage :

```
The position of SPADES is 0
The position of HEARTS is 1
The position of DIAMONDS is 2
The position of CLUBS is 3
```

Mot-clé `switch` et `enum`

Supposons qu'on est une `enum` pour les jour de la semaine :

```
public enum Day
{
    SUNDAY,
    MONDAY,
    TUESDAY,
    WEDNESDAY,
    THURSDAY,
    FRIDAY,
    SATURDAY;
}
```

Mot-clé switch et enum

On peut utiliser le mot-clé `switch` pour traiter des cas différents :

```
public enum Day{
  public void dayIsLike() {
    switch (this) {
      case MONDAY:
        System.out.println("Mondays are bad."); break;
      case FRIDAY:
        System.out.println("Fridays are better."); break;
      case SATURDAY:
      case SUNDAY:
        System.out.println("Weekends are best."); break;
      default:
        System.out.println("Midweek days are so-so.");
        break;
    }
```

Mot-clé switch et enum (nouvelle version)

Depuis java 14, on peut écrire de façon équivalente :

```
public enum Day{
  public void dayIsLike() {
    switch (this) {
      case MONDAY
        -> System.out.println("Mondays are bad.");
      case FRIDAY
        -> System.out.println("Fridays are better.");
      case SATURDAY, SUNDAY
        -> System.out.println("Weekends are best.");
      default
        -> System.out.println("Midweek days are so-so.");
    }
  }
}
```