Java - 1ère année

Enumérations

Enumérations

- Rappel : définition d'une énumération :

```
public enum CouleurFeu {Rouge, Vert, Orange}
```

- Utilisations :

```
CouleurFeu couleur = CouleurFeu.Rouge;
if (couleur == CouleurFeu.Rouge) {…}
switch (couleur) {
  case Rouge: …
  case Vert: …
  case Orange: …
}
```

Enumération : où ?

- Utilisée de manière partagée : publique et dans son propre fichier.

```
// Fichier CouleurFeu.java
public enum CouleurFeu {Rouge, Vert, Orange}

// Fichier Feu.java
public class Feu {
  private CouleurFeu couleur;
  public CouleurFeu getCouleur();
  …
}
```

Enumération : où ?

- Pour un état interne : privée et interne à la classe concernée.

```
public class Door {
  private enum DoorState {Opened, Closed, Locked};
  private DoorState state;
  public void close () {
    if (state == DoorState.Opened) {
      state = DoorState.Closed;}}
  public void open() {…}
  public void lock() {…}
  public boolean isClosed() {
    return (state == DoorState.Closed);}
  public boolean isOpened() {…}
  public boolean isLocked() {…}
  … }
```

« Object » methods

- Un type énuméré possède les méthodes toString, equals et hasCode de haute qualité.
  - toString() : le nom de la constante telle que définie dans l'enum.
  - equals() : compare les constantes
- Exemple :

```
CouleurFeu couleur = CouleurFeu.Rouge;
System.out.println(couleur);
```

- Ce code affiche : `Rouge`

« Enum » methods

- Un type énuméré hérite de la classe java.lang.Enum
- Les méthodes toString, equals et hashCode y sont redéfinies
- Un type énuméré est aussi :
  - Serializable
  - Comparable : possède la méthode compareTo
- Autres méthodes : valueOf, ordinal et name

« Enum » methods

- valueOf :
  - méthode statique qui convertit une chaîne de caractères en constante correspondante du type énuméré
  - lève l'exception IllegalArgumentException si pas de correspondance
  - utile pour des saisies utilisateurs
- Exemple :

```
public static void main(String[] args) {

   CouleurFeu couleur;

   couleur = CouleurFeu.valueOf(args[0]);

   …

}
```

« Enum » methods

- ordinal() :
  - renvoie la position de la constante dans la déclaration du type énuméré
  - utile pour (re)définir un ordre sur les constantes
- name() :
  - chaîne de caractères
  - valeur de la constante telle qu'elle a été définie
  - valeur d'origine renvoyée par toString

Ensemble des valeurs

- values() :
  - Méthode statique
  - renvoie un tableau contenant toutes les constantes du type énuméré
- Exemple :

```
CouleurFeu[] couleurs = CouleurFeu.values();
for (CouleurFeu couleur : couleurs) {
  System.out.print(couleur + " ");
}
```

Donne : `Rouge Vert Orange`

Compléter une énumération

- redéfinition d'une méthode

```
public enum CouleurFeu {
   Rouge, Vert, Orange;
   @Override
   public String toString() {
      return super.toString().toLowerCase();
      // return CouleurFeu.values()[this.ordinal()]
      //      .name().toLowerCase();
   }
}

CouleurFeu couleur = CouleurFeu.Rouge;
System.out.prinln(couleur); // affiche : rouge
```

Compléter une énumération

- ajout d'attributs et de méthodes

```
public enum CouleurFeu {
  Rouge, Vert, Orange;

  private static CouleurFeu[] couleurs =
      CouleurFeu.values();

  public CouleurFeu getNext() {
      int position = this.ordinal();
      int nombreCouleurs = couleurs.length;
      return couleurs[(position+1)%nombreCouleurs];
  }
}
```

Compléter une énumération

```
public static void main(String[] args) {
   CouleurFeu couleur;
   couleur = CouleurFeu.valueOf(args[0]);
   System.out.println("Couleur entrée : " + couleur);
   System.out.println("Prochaine couleur : "
       + couleur.getNext());
}
```

Compléter une énumération

- Ajout d'un attribut d'objet :

```
public enum CouleurFeu {
   Rouge, Vert, Orange;
   private int duree;
   public int getDuree() {return duree;}
}
```

- Comment construire la valeur énumérée ?
- => un constructeur privé appelé pour chaque définition de constante

Compléter une énumération

```
public enum CouleurFeu {
  Rouge (35),           //
  Vert (30),            // appels du constructeur
  Orange (5);           //

  private int duree;

  // constructeur
  private CouleurFeu(int duree) {
    this.duree = duree;
  }

  public int getDuree() {return duree;}
}
```

Compléter une énumération

```
public static void main(String[] args) {
   CouleurFeu couleur;
   couleur = CouleurFeu.valueOf(args[0]);
   System.out.println("Couleur entrée : " + couleur);
   System.out.println("Durée couleur : "
       + couleur.getCouleur());
}
```