



## Ensemble de valeurs fini

---

Exemple: les jours de la semaine

LUNDI , MARDI, MERCREDI, JEUDI,  
VENDREDI, SAMEDI, DIMANCHE

XH

1



## Rappel sur les constantes dans une classe, exemple

```
public class MaClasse {  
    public static final double PI = 3.14;  
    public static final int ANNEE_EN_COURS = 1970;  
}  
  
public class UnTest {  
    public static void main( String [] args ) {  
        double d = MaClasse.PI; //Affectation  
        System.out.println( MaClass.PI );//affichage  
        ...  
    }  
}
```

XH ...}}

2



## Remarques sur les constantes

- Le nom des constantes est en majuscule avec un séparateur underscore
- Avec le modificateur « static », pas besoin d'instancier la classe
- Avec le modificateur « final », la valeur ne peut plus être changée
- Avec le modificateur « public », la constante devient ~globale à l'application
- Résumé: public final static → constante globale à l'application

XH

3



## Ensemble de valeurs fini avec une classe

```
Public class UnTest {  
    public static final int LUNDI =1;public static final int MARDI =2;  
    public static final int MERCREDI =3;public static final int JEUDI =4;  
    public static final int VENDREDI =5;public static final int SAMEDI =6;  
    public static final int DIMANCHE =7;  
    public static final String[] joursAff = {  
        "indefini", "lundi", "mardi", "mercredi", "jeudi" , "vendredi"};  
    public static void main( String [] args ) { ...  
        int leJour = JEUDI;  
        if( leJour == MARDI ) { ...  
            switch( leJour )  
            case LUNDI: ...  
                sysout.println(« Aujourd'hui : » + leJour); //NOK  
                sysout.println(« Aujourd'hui : » + joursAff[ leJour ]); //MIEUX
```

XH

4



## Ensemble de chaînes

```
Public interface EnumJours {  
    String premierJour=« lundi », dernierJour=« dimanche » ;  
}  
//utilisation  
Public class UnTest implements EnumJours {  
    public static void main( String [] args ) { ...  
        String leJour = dernierJour ; //OK  
        if( leJour.equals(premierJour) ) { ... //OK  
        switch( leJour ) //marche depuis Java 7  
  
        sysout.println(« Aujourd'hui : » + leJour); //OK
```

XH

5



## Problème

- il n'y a pas vraiment de contrôle sur la valeur!!

XH

6



XH

7



## Utilisation de l'interface pour avoir des constantes globales

---

XH

8



## Propriétés dans l'interface

- On peut déclarer des champs dans une interface !
- Ce sont obligatoirement des constantes globales.
  - Implicitement, elles ont les modificateurs **public final static**

//exemple

```
public interface MesConstantesGlobales {  
  
    double PI = 3.14;}  
  
XH
```

9



## Propriétés dans l'interface (2)

- L'accès aux constantes globales peut se faire en préfixant par le nom de l'interface (~ propriété de classe).

```
public interface MesConstantesGlobales {  
    double PI = 3.14;  
}  
  
public class UnTest{  
    public static void main( String [] args ) {  
        double surface = MesConstantesGlobales.PI * ...  
        System.out.println(MesConstantesGlobales.PI);  
        ...}}  
  
XH
```

10



## Propriétés dans l'interface (3)

- Si vous implémentez l'interface, vous n'avez pas besoin de préfixer (héritage) → simplification d'écriture

```
Public interface MesConstantesGlobales {  
    double PI = 3.14;  
}  
  
Public class UnTest implements MesConstantesGlobales{  
    public static void main( String [] args ) {  
        double surface = PI * ...  
        System.out.println(PI);  
    }  
}
```

11



## Ensemble de valeurs fini

Exemple: les jours de la semaine

LUNDI , MARDI, MERCREDI, JEUDI,  
VENDREDI, SAMEDI, DIMANCHE

XH

12



## Ensemble de valeurs fini avec une interface

```
Public interface EnumJoursDeLaSemaine {  
  
    int LUNDI =1, MARDI =2, MERCREDI=3, JEUDI=4, VENDREDI=5,  
    SAMEDI=6, DIMANCHE=7;  
  
    String[] joursAff ={"indefini", "lundi", "mardi", "mercredi",  
    "jeudi" , "vendredi"};  
  
    }//utilisation  
  
Public class UnTest implements EnumJoursDeLaSemaine{  
  
    public static void main( String [] args ) { ...  
  
        int leJour = JEUDI;  
  
        if( leJour == MARDI ) { ...  
  
            switch( leJour )  
  
                case LUNDI: ...  
  
                sysout.println(« Aujourd'hui : » + leJour); //NOK  
  
                xH sysout.println(« Aujourd'hui : » + joursAff[ leJour ]); //MIEUX
```

13



## Ensemble de chaînes

```
Public interface EnumJours {  
  
    int premierJour=« lundi », dernierJour=« dimanche » ;  
  
    }  
  
    //utilisation  
  
Public class UnTest implements EnumJours {  
  
    public static void main( String [] args ) { ...  
  
        String leJour = dernierJour ; //OK  
  
        if( leJour.equals(premierJour) ) { ... //OK  
  
        switch( leJour ) //Ne marche pas !!!  
  
        xH sysout.println(« Aujourd'hui : » + leJour); //OK
```

14



## Problème

- il n'y a pas vraiment de contrôle sur la valeur!!
  - La valeur fait-elle partie de l'ensemble fini de constantes ?

XH

15



## Les enumérations JAVA5

---

il y a un contrôle sur la valeur !!!!

XH

16





## Le mot réservé enum

- Voir exemples et démo sur le site ftp habituel
- TP voir le pdf de Delannoy

XH

17



### Exemple

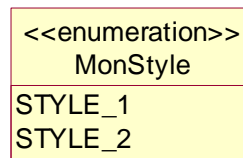
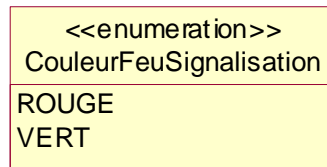
```
public class Demo3Enum {  
    enum CouleurFeu { VERT, ROUGE }  
  
    public static void main(String[] args) {  
        CouleurFeu feu = CouleurFeu.VERT; //la variable  
  
        switch(feux) {  
            case VERT:  
                System.out.println("Vous pouvez avancer !");  
                break;  
            case ROUGE:  
                System.out.println("Arrêtez vous !");  
                break;  
            default:  
                System.out.println("Pb : cas impossible");  
                throw new IllegalArgumentException("cas  
impossible");  
                //break;  
        }  
    }  
}
```

XH

18



## Représentation des énumération en UML



XH

19



# END



XH

20