

Développement Orienté Objet

Cours 6 : Interfaces, énumération



Albrecht Zimmermann

albrecht.zimmermann@unicaen.fr

**Un peu plus d'informations sur
les interfaces**

Variables d'une interface

- Sont toujours implicitement
 - `public`
 - `static`
 - `final`

```
interface Exemple{  
    int a=10;  
    public int a=10;  
    public static final int  
        a=10;  
    final int a=10;  
    static int a=0;  
}
```

Pourquoi qu'il n'y a pas de conflit de variables ?

Variables d'une interface

(2)

- Doivent toujours être initialisées
- → Ne peuvent pas recevoir d'autres valeurs

```
interface Exemple{  
    int a;  
}
```

- Peuvent être accédées comme toutes autres variables de classe

```
public class Test  
implements Exemple{  
    public static void  
    main(String args[]){  
        Exemple.a = 5;  
    }  
}
```

Variables d'une interface

(2)

- Doivent toujours être initialisées
- → Ne peuvent pas recevoir d'autres valeurs

```
interface Exemple{  
    int a;  
}
```

Erreur de
compilation

- Peuvent être accédées comme toutes autres variables de classe

```
public class Test  
implements Exemple{  
    public static void  
    main(String args[]){  
        Exemple.a = 5;  
    }  
}
```

Variables d'une interface

(2)

- Doivent toujours être initialisées

```
interface Exemple{  
    int a;  
}
```

Erreur de
compilation

- Peuvent être accédées comme toutes autres variables de classe

- → Ne peuvent pas recevoir d'autres valeurs

```
public class Test  
implements Exemple{  
    public static void  
    main(String args[]){  
        Exemple.a = 5;  
    }  
}
```

Erreur de
compilation

Méthodes d'une interface

- Peuvent être déclarées **default**
 - → peut avoir d'implémentation
- Peuvent être déclarées **static**

Méthodes d'une interface

- Peuvent être déclarées **default**
 - → peut avoir d'implémentation

Changer l'interface sans
exiger de changer les
classes qui l'implémentent

- Peuvent être déclarées **static**

Les énumérations

Quoi ?

- Collection de constantes
- Mot clé **enum**
- Enfant (implicite) de `java.lang.enum`
- Traduit en classe pendant la compilation
 - Suffix `.java`
 - Déclaration **public** (**protected**) **enum**
- Méthodes implicitement déclarées
 - P. ex. `public static values()`

<https://docs.oracle.com/javase/8/docs/api/java/lang/Enum.html>

Quoi ?

- Collection de constantes
- Mot clé **enum**
- Enfant (implicite) de `java.lang.enum`
- Traduit en classe pendant la compilation
 - Suffix `.java`
 - Déclaration **public** (**protected**) **enum**
- Méthodes implicitement déclarées
 - P. ex. `public static values()`

Ne peuvent pas hériter d'une
autre énumération
Peuvent étendre des interfaces

Pourquoi ?

Pourquoi ?

- Structuration
- Réutilisation
- Maintenabilité

Comment ?

```
public enum Planet{
```

```
    MERCURY, VENUS, EARTH,  
    MARS, JUPITER, SATURN,  
    URANUS, NEPTUNE
```

```
}
```

Comment ?

Planet.java

```
public enum Planet{  
    MERCURY, VENUS, EARTH,  
    MARS, JUPITER, SATURN,  
    URANUS, NEPTUNE  
}
```

Comment ? (2)

```
public enum Planet {
```

Peuvent être plus complexes

```
    MERCURY (3.303e+23, 2.4397e6),
```

```
    VENUS   (4.869e+24, 6.0518e6),
```

```
    EARTH   (5.976e+24, 6.37814e6),
```

```
    MARS     (6.421e+23, 3.3972e6),
```

```
    JUPITER  (1.9e+27,    7.1492e7),
```

```
    SATURN   (5.688e+26, 6.0268e7),
```

```
    URANUS   (8.686e+25, 2.5559e7),
```

```
    NEPTUNE  (1.024e+26, 2.4746e7);
```

```
...
```

```
}
```


Comment ? (3)

```
public enum Planet {
```

Besoin des noms d'attribut
Besoin d'un constructeur

```
...
```

```
private final double mass;
```

```
private final double radius;
```

```
Planet(double mass, double radius) {
```

```
    this.mass = mass;
```

```
    this.radius = radius;
```

```
}
```

Privé ou paquetage-privé,
automatiquement appelé

```
...
```

```
}
```

Comment ? (4)

```
public enum Planet {
```

```
...
```

```
    private double mass() { return mass; }
```

```
    private double radius() { return radius; }
```

```
    // universal gravitational constant ( $\text{m}^3 \text{kg}^{-1} \text{s}^{-2}$ )
```

```
    public static final double G = 6.67300E-11;
```

```
    double surfaceGravity() {
```

```
        return G * mass / (radius * radius);
```

```
    }
```

```
...
```

```
}
```

+ de méthodes possible
+ de variables possible

On peut même avoir une « main » !

Comment ? (5)

```
public enum Planet {  
...  
    VENUS (4.869e+24, 6.0518e6){  
        @Override  
        public Planet nearestInner(){  
            return MERCURY;  
        }  
        public Planet nearestOuter(){  
            return EARTH;  
        }  
    },  
    ...  
    public abstract Planet nearestInner();  
    public abstract Planet nearestOuter();  
    ...  
}
```

Comportements
spécifiques

Appel : Planet.EARTH.nearestInner()