

Introduction de `enum`

## 8.1 TYPES ÉNUMÉRÉS

# Problème

- Écrire un programme qui permet d'afficher différents messages en fonction de l'état d'une porte (qui peut être *ouverte*, *fermée*, *cassée* ou dans un état *inconnu*).
- Comment coder cela ?

# Solution 1

- Coder l'état avec une **constante entière**

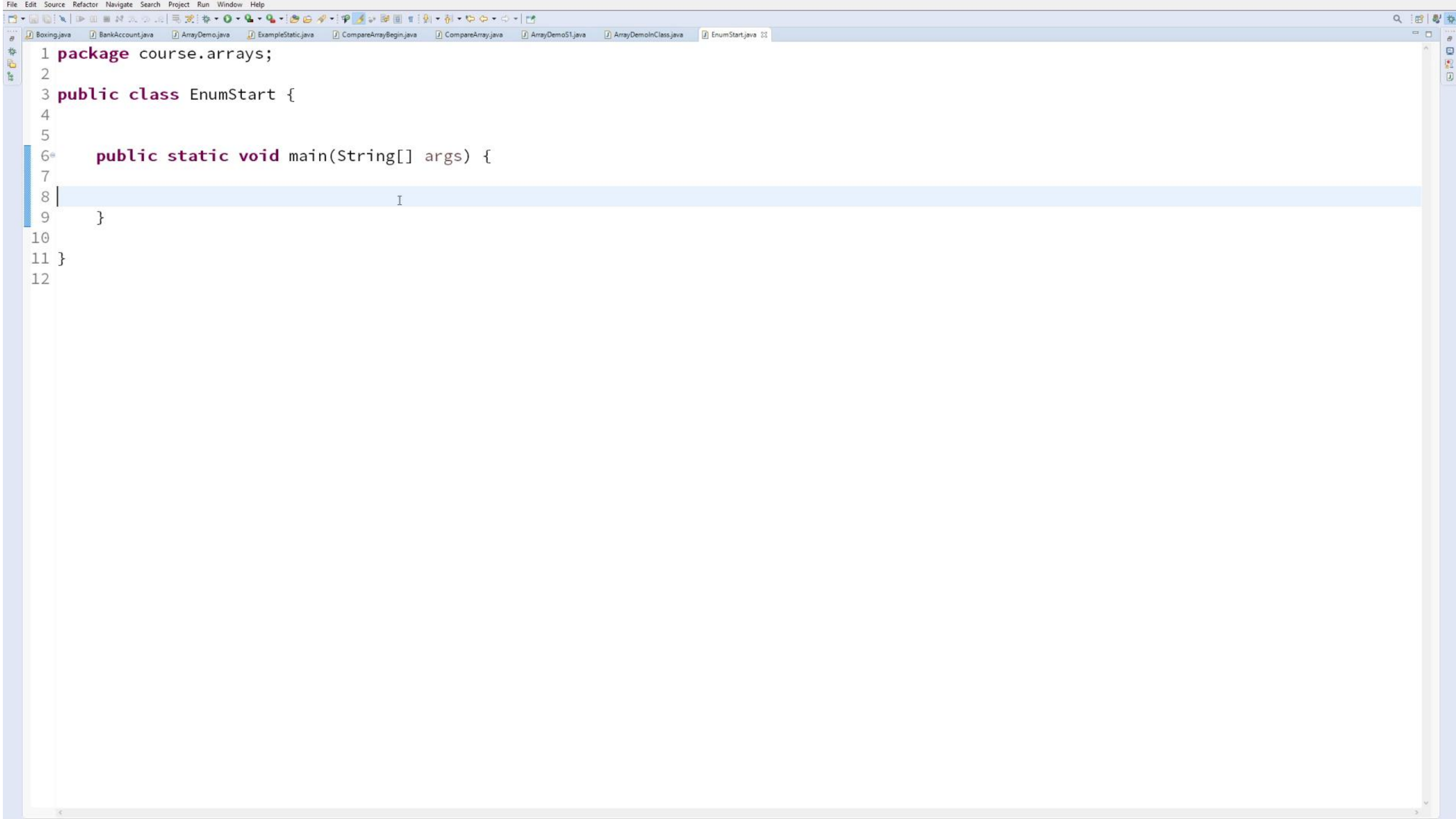
```
public final int OPEN = 0;
public final int CLOSED = 1;
public final int BROKEN = 2;
public final int UNKNOWN = 3;
int doorState = OPEN;

if(doorState == CLOSED){
    ...
}
```

## Solution 2

- Coder l'état avec un *String*

```
String doorState = "OPEN";  
if(doorState.equalsTo("OPEN")){  
    ...  
}  
  
if(doorState.equalsTo("CLOSED")){  
    ...  
}  
...
```



# Solution à utiliser

- Un **type énuméré**

Permettent de stocker des énumérations **finies**, c'est-à-dire contenant un nombre fixe (constant) d'éléments

Types énumérés

# Exemples

# Syntaxe

- Syntaxe de déclaration :


**enum** *identif*ier {*VAL1*, *VAL2*, ...};

- *Exemple* :

```
enum Sizes {S, M, L, XL, XXL};
```



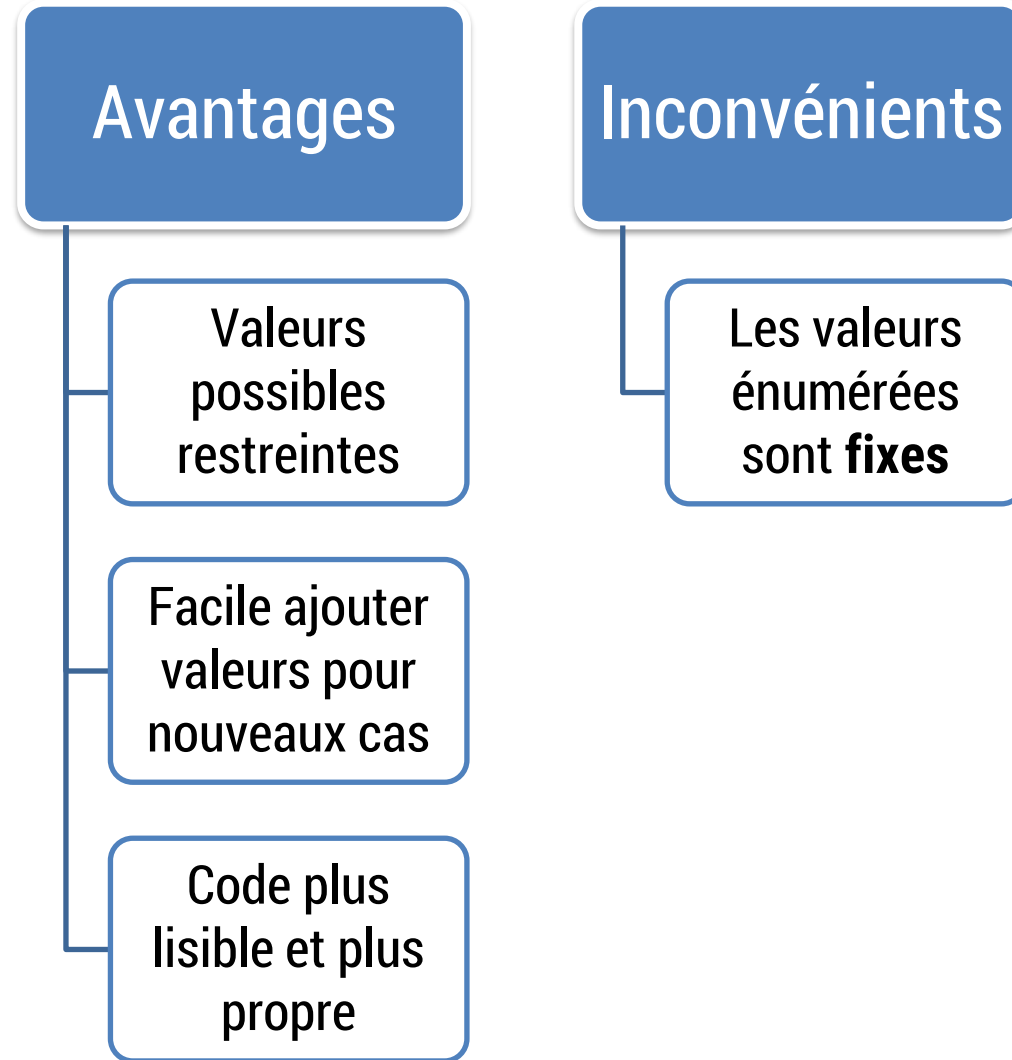
# Les types énumérés (1)



```
public class EnumDemo{
    enum State {OPEN, BROKEN};

    public static void main(String args[]){
        State doorState = State.OPEN;
        switch(doorState){
            case OPEN:
                System.out.println("The door is open");
                break;
            case BROKEN :
                System.out.println("The door is destroyed");
                break;
            default:
                System.out.println("Door state is unknown");
                break;
        }
    }
}
```

# Les types énumérés (2)



# Exercice

- Déclarer des types énumérés pour :
  - Arôme de glace
  - Races de chat