Enums, Vars, Vars, and Ellipsis

Josh Wilcox (jw14g24)

April 28, 2025

Table of Contents

Enums

The need for Enums

- Enums are a means of modelling a variable with a discrete, finite number of constant values
- Are a good way of replacing multiple related `public static final` constant variables into one organized structure
  - Instead of having separate constants, they're grouped logically together
- Removes various problems with static constants:
  - Not typesafe: Regular constants can be mixed up with other integer values
    - E.g., accidentally using a month number where a day of week is expected
  - Brittle: Changing constant values can break code that depends on specific integer values
    - Recompilation is required everywhere the constant is used
  - Uninformative: Plain constants don't provide any context about their purpose or valid values
    - Just seeing the number 1 doesn't tell you if it means Monday, January, or something else

Enum to the rescue

- Enums alone are more informative and type safe - meaning methods requiring an enum can only accept that kind of enum
- Errors can be found at compile time rather than at runtime

Extra Features of Enum

values() method

- Returns the possible values of an enum as an array
- Allows looping etc
- `ScienceSchool[] allSchools = ScienceSchool.values();`

ordinal() method

- Returns the rank of a particular enum as an integer
  - The rank is where position it appears in the original enum

```
1  public class ordinal {
2          enum ScienceSchool {ORC,CHEM,ENG,ECS}
3          enum Faculty{FEPS,FAH, FELS, FM, FSS}
4
5          public static void main(String[] args) {
6                  ScienceSchool x=ScienceSchool.ORC;
7                  ScienceSchool y=ScienceSchool.ENG;
8                  System.out.println("ORDINAL for "+x+" = "+x.ordinal());
9                  System.out.println("ORDINAL for "+y+" = "+y.ordinal());
10                 for(Faculty f:Faculty.values()) {
11                         System.out.println("ORDINAL for "+f+" = "+f.ordinal());
12                 }
13         }
14 }
15
16 // ORDINAL for ORC = 0
17 // ORDINAL for ENG = 2
18 // ORDINAL for FEPS = 0
19 // ORDINAL for FAH = 1
20 // ORDINAL for FELS = 2
21 // ORDINAL for FM = 3
22 // ORDINAL for FSS = 4
```

Enums in Java

How are Java enums unique?

- Each enum type is the instantiation of an *anonymous* **inner class**
- Enum Types are therefore a mechanism for implementing the **Singleton Programming Pattern** - Software Engineering Principle
- It also means you can enhance an enum to give it extra behaviour - a characteristic unique to Java

More on the Singleton Programming Pattern

- Restricts the instantiation of a class to one ”single” instance
- Useful when exactly one object is needed to coordinate actions across the system
- A singleton is a form of global variable

Enums are instances of anonymous nested classes

`public enum Country{UK,FRANCE,GERMANY,ITALY}` is equivalent to:

```
1 static public abstract class Country{
2   private Country(){};
3   public static final Country UK=new Country(){};
4   public static final Country FRANCE=new Country(){};
5   public static final Country GERMANY=new Country(){};
6   public static final Country ITALY=new Country(){};
7 }
```