

Java training

Enums - creation, usage, constructors, methods

Session overview

- **Enums**
 - Creation
 - Usage
 - Constructors
 - Methods
- **Hands-on**

Enums

- An **enum** - enumeration of values (constants)

```
public enum CompassDirections {  
    NORTH, EAST, SOUTH, WEST;  
}
```

- **Usage** - representing a **fixed set of constants**

Creation

- Enums values can be created as:
 - Simple entries: MONDAY, TUESDAY, WEDNESDAY
 - Entries with values: MONDAY(1), TUESDAY(2), WEDNESDAY(3)

- Creating entries with values:

```
enum WeekDay {  
    MONDAY(1), TUESDAY(2), ...;    // specified before the constructor  
  
    private int day;    // + getter for it  
    WeekDay(int day) { // must be private or package-private (default)  
        this.day = day;  
    }  
}
```

Usage

- As static and final values:

```
WeekDay monday = WeekDay.MONDAY;  
System.out.println(monday);           // output: MONDAY  
System.out.println(monday.getDay());  // output: 1
```

- Using the enum's static `.valueOf(String name)` method: **+ Hands-on**

```
WeekDay tuesday = WeekDay.valueOf("TUESDAY");
```

- Retrieving an enum value as an index of the static `.values()` method:

```
WeekDay wednesday = WeekDay.values[2];
```

Benefits

- **Type safety** - a variable of enum type cannot be assigned to another type

- Usage in `switch` statements:

```
WeekDay stockDay = productService.getProcessingDay();
switch (stockDay) {
    case WeekDay.FRIDAY:
        // code for doing the processing on Friday
        break;
    //...
}
```

- Can be compared using the `==` operator

Methods in enum values

Enum *values* can have methods defined in them:

```
enum WeekDay {  
    MONDAY(1) {  
        public void process() {  
            System.out.println("Monday processing");  
        }  
    }  
  
    public void abstract process();  
}
```

+ Hands-on

Methods in enums

Enums can also contain methods:

```
enum WeekDay {  
    MONDAY(1);  
  
    public String processCurrentDay() {  
        if (this.day == 1) {  
            System.out.println("Processing for Monday...");  
        }  
    }  
}
```


Q&A session

1. You ask, I answer
2. I answer, you ask