# Enum

# Enum

1. Data type which **contains a fixed set of constants**.
2. Used for **days of the week** (SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, and SATURDAY) , **directions** (NORTH, SOUTH, EAST, and WEST), **season** (SPRING, SUMMER, WINTER, and AUTUMN or FALL), **colors** (RED, YELLOW, BLUE, GREEN, WHITE, and BLACK)
3. Enum constants are **static and final implicitly**
4. Enums are used to **create our own data type like classes**
5. enum can be **defined either inside the class or outside the class**.

# Defining Java Enum

**Syntax:**

**enum** Season { WINTER, SPRING, SUMMER, FALL }

(OR)

**enum** Season { WINTER, SPRING, SUMMER, FALL; }

**Defined outside class:**

```
enum Season { WINTER, SPRING, SUMMER, FALL }

class example2{
public static void main(String[] args) {

     }
}
```

**Defined inside class:**

```
class EnumExample3{

enum Season { WINTER, SPRING, SUMMER, FALL}

public static void main(String[] args) {

}
}
```

# Methods in Enum

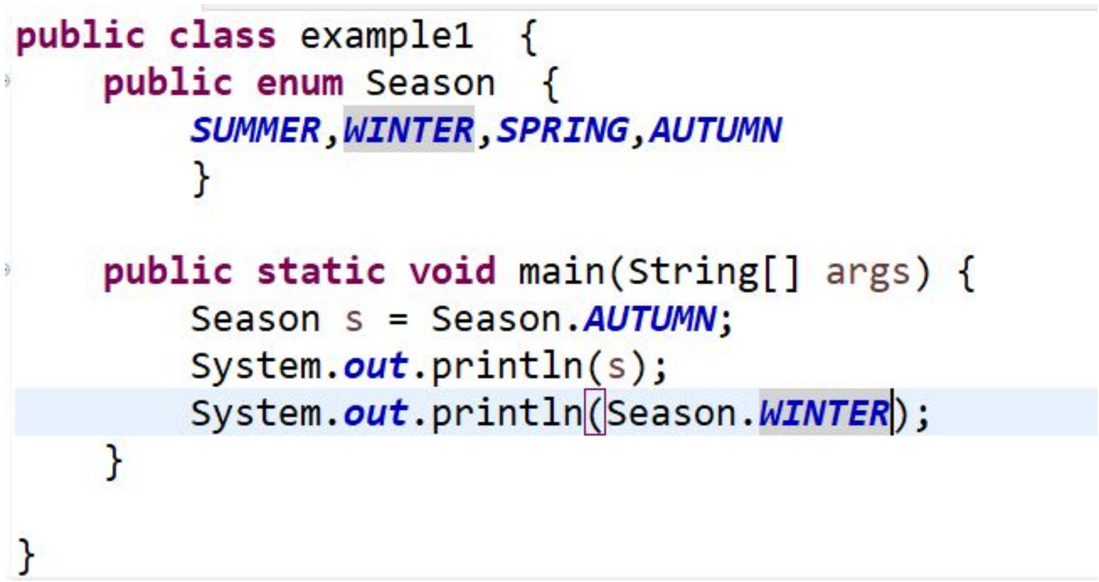
values() : returns an **array** containing all the **values** of the enum.

valueOf() : returns the **value** of given constant enum.

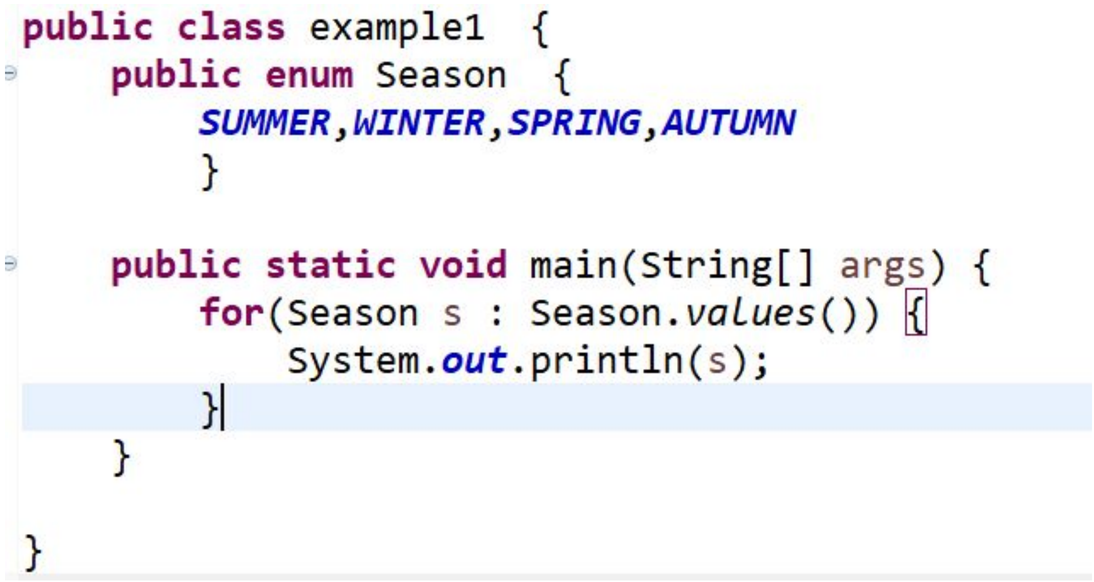
ordinal() : returns the **index** of the enum value.

value() : return the value of the given enum

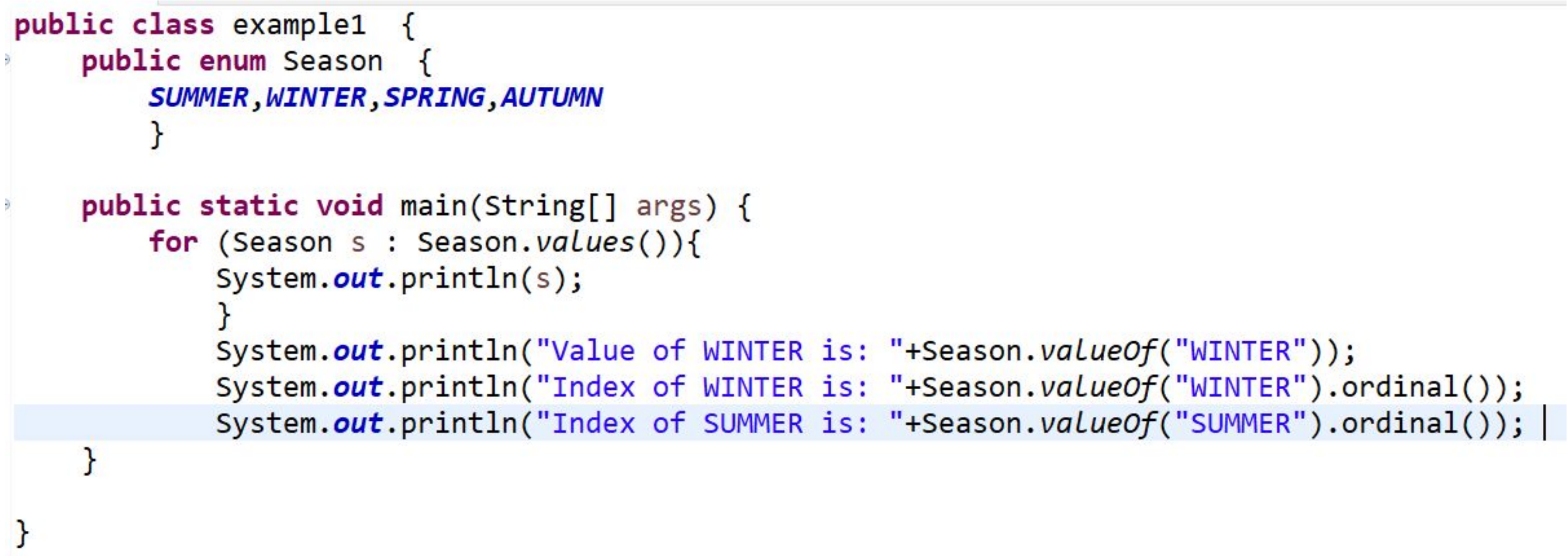
# Java Enum Methods



```java
public class example1  {
    public enum Season  {
        SUMMER,WINTER,SPRING,AUTUMN
        }

    public static void main(String[] args) {
        Season s = Season.AUTUMN;
        System.out.println(s);
        System.out.println(Season.WINTER);
    }

}
```

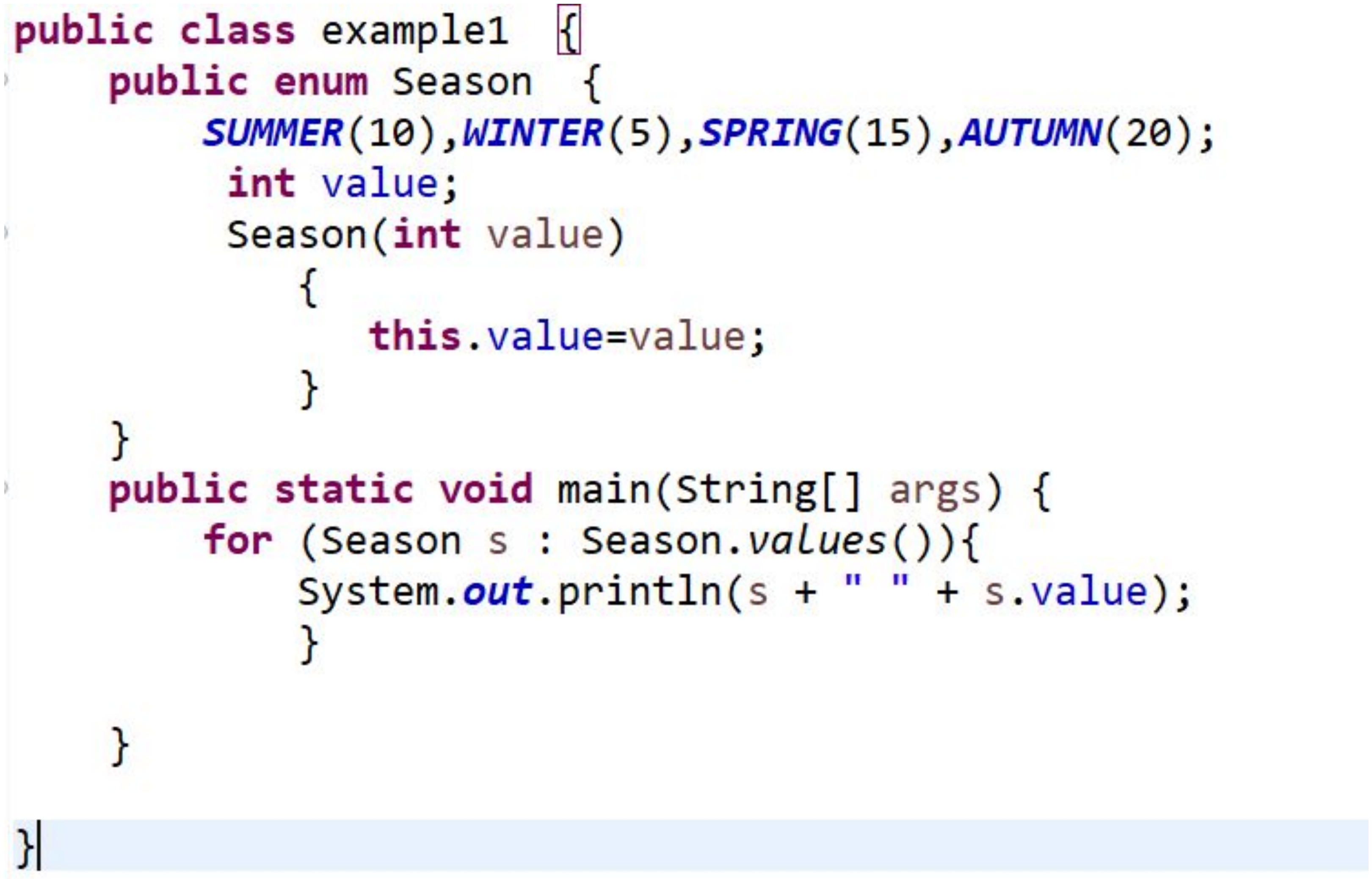


```java
public class example1  {
    public enum Season  {
        SUMMER,WINTER,SPRING,AUTUMN
        }

    public static void main(String[] args) {
        for(Season s : Season.values()) {
            System.out.println(s);
        }
    }

}
```



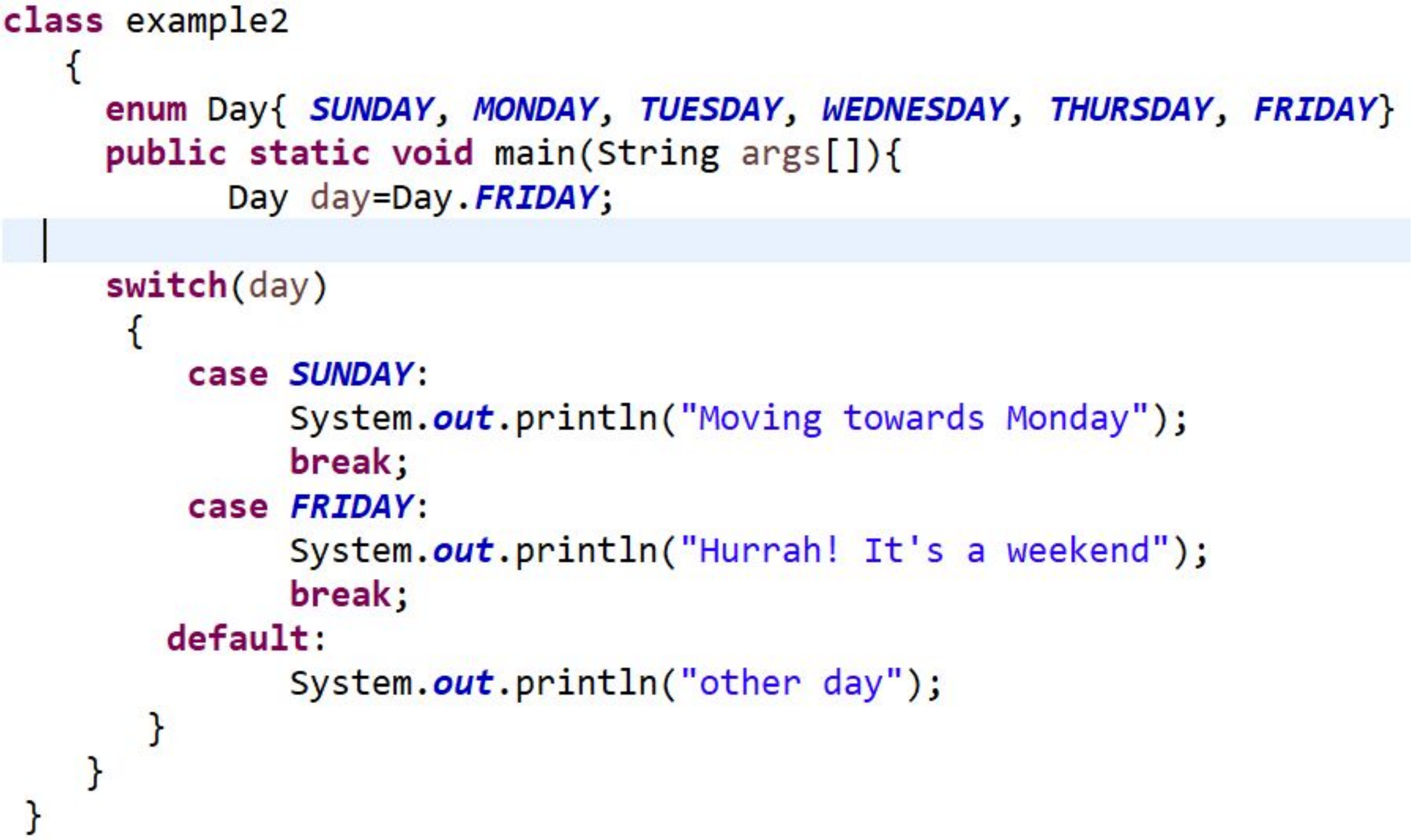
```java
public class example1  {
    public enum Season  {
        SUMMER,WINTER,SPRING,AUTUMN
        }

    public static void main(String[] args) {
        for (Season s : Season.values()){
            System.out.println(s);
            }
            System.out.println("Value of WINTER is: "+Season.valueOf("WINTER"));
            System.out.println("Index of WINTER is: "+Season.valueOf("WINTER").ordinal());
            System.out.println("Index of SUMMER is: "+Season.valueOf("SUMMER").ordinal());
    }

}
```

## Specifying initial value to the enum constants



```java
public class example1  {
    public enum Season  {
        SUMMER(10),WINTER(5),SPRING(15),AUTUMN(20);
         int value;
         Season(int value)
            {
               this.value=value;
            }
    }
    public static void main(String[] args) {
        for (Season s : Season.values()){
            System.out.println(s + " " + s.value);
            }

    }

}
```

# Enum in a switch statement



```
class example2
   {
     enum Day{ SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY}
     public static void main(String args[]){
           Day day=Day.FRIDAY;

     switch(day)
      {
         case SUNDAY:
              System.out.println("Moving towards Monday");
              break;
         case FRIDAY:
              System.out.println("Hurrah! It's a weekend");
              break;
        default:
              System.out.println("other day");
       }
    }
 }
```