* Enum is a class
* Set of constants
* We can define our own datatype
* Enums are used when we know all the values at compile time.
* According to the Java naming conventions, we should have all constants in capital letters. So, we have enum constants in capital letters.
* The Java enum constants are static and final implicitly. It is available since JDK 1.5.
* we can define an enum either inside the class or outside the class.
* Java Enum internally inherits the Enum class, so it cannot inherit any other class, but it can implement many interfaces. We can have fields, constructors, methods, and main methods in Java enum.
* First line inside enum should be list of constants & then other things like methods,constructor,variable.
* Enums are used in switch statements.
* In switch statement we can use integer constant, character constants and enums.
* Enum can contain only concrete methods i.e no abstract methods.
* Enum can not be inherited coz enum is set of constants and inheritance means you modify existing things
* Constructor of enum type is private. If you don't declare private compiler internally creates private constructor.

### Can we create the instance of Enum by new keyword?

|  |
| --- |
| No, because it contains private constructors only. |

we are using value(), valueOf(), and ordinal() methods of Java enum.

1. **class** EnumExample1{
2. //defining enum within class
3. **public** **enum** Season { WINTER, SPRING, SUMMER, FALL }
4. //creating the main method
5. **public** **static** **void** main(String[] args) {
6. //printing all enum
7. **for** (Season s : Season.values()){
8. System.out.println(s);
9. }
10. System.out.println("Value of WINTER is: "+Season.valueOf("WINTER"));
11. System.out.println("Index of WINTER is: "+Season.valueOf("WINTER").ordinal());
12. System.out.println("Index of SUMMER is: "+Season.valueOf("SUMMER").ordinal());
13. //ordinal method return index starting from 0
14. }}