14/02/2026: Enumerators

1. Enum objects can be directly used in switch. They are self qualified.

   ```
   Gender g=Gender.MALE;
   Switch(g){
       Case MALE: //
       Case FEMALE://
       Case OTHER://
   }
   ```
   If you type case Gender.MALE inside switch it will throw error

2. values() method is inbuild method of any enum. It will return all the enum object of particular enum

   ```
   enum Gender{
     MALE,FEMALE,OTHER
   }

   For(Gender g: Gender.values())
       System.out.println(g)
   ```

   It will print all the Gender object

   ```
   enum EmployeeType{
     FULLTIME(10),PARTTIME(20),CONTRACT(30);
     int value;
     String name;
   }
   ```
   Here value() returs only FULLTIME,PARTTIME,CONTRACT, not value and name

3. == and equals() method will compare 2 enum for equality
   ```
   If(Gender.MALE==Gender.FEMALE) //False
   If(Gender.MALE.equals(Gender.MALE))  //True
   ```

4. ordinal(), this method will return the positional value of enum object.
   ```
    enum Gender{
     MALE,FEMALE,OTHER
   }
   ```

MALE.ordinal() //0
FEMALE.ordinal()//1

If I change the order
enum Gender{
  OTHER, MALE,FEMALE
}
Now MALE.ordinal() is 1 not 0

That's why it should bot be used for business logic

5. compareTo(), it will returns either -, 0, + based on ordinal value comparison
   enum Gender{
     MALE,FEMALE,OTHER
   }

   Gender.MALE.compareTo(Gender.MALE) //0
   Gender.MALE.compareTo(Gender.OTHER) //-2
   Gender.FEMALE.compareTo(Gender.MALE) //1