**ENUM – Part\_02**

* **Enum vs inheritance:**

1. Every Enum is always direct child class of java.lang.Enum and hence our Enum can’t extend any other enum. (Because won’t provide support for multiple inheritance).
2. Every Enum is always final implicitly and hence for our Enum we can’t create child Enum.

Because of above reasons we can conclude inheritance concept not applicable for Enum explicitly and we can’t use “extends” keyword for Enum.

* **Example:**

enum X{

}

enum Y extends X{

}

// Invalid

enum X extends java.lang.Enum{

}

// Invalid

class X{

}

enum Y extends X{

}

// Invalid

enum X{

}

class Y extends X{

}

// Invalid – CE:

CE1: cannot inherit from final X

CE2: enum types are not extensible

Anyway an Enum can implement any number of interfaces

interface X{

}

enum Y implements X{

}

* **Conclusions of java.lang.Enum:**

Every enum in Java is the direct child class of java.lang.Enum and hence this class acts as a base class for all Java enums.

It is an abstract class and it is direct child class of java.lang.Object.

It implements Serializable and Comparable interfaces.

* **values() method:**

Every Enum implicitly contains values method, to list out all values present inside Enum.

Beer[] b = Beer.values();

values() not present in java.lang.Enum and java.lang.Object classes. enum keyword implicitly provides this method.

* **ordinal() method:**

Inside enum order of constants is important and we can represent this order by using ordinal value.

We can find ordinal value of enum constant by using ordinal()

public fina int ordinal();

Ordinal value is 0 based like array index.

Example:

enum Beer{

KF,KO,RC,FO;

}

class Test{

public static void main(String[] args){

Beer[] b = Beer.values();

for(Beer b1 : b){

System.out.println(b1+”….”+b1.ordinal());

}

}

}

Output:

KF….0

KO….1

RC….2

FO….3

* **Specialty of java Enum:**

In old languages enum we can take only constants. But in Java enum in addition to constants we can take methods, constructors, normal variables etc. Hence Java enum is more powerful than old languages Enum.

Even inside Java enum, we can declare main() and we can run enum class directly from command prompt.

Example:

enum Fish{

START,GUPPY,GOLD;

public static void main(String[] args){

System.out.println(“ENUM MAIN METHOD”);

}

}

javac Fish.java

java Fish

Output: ENUM MAIN METHOD

Note:

In addition to constants if we are taking any extra member like a method then list of constants should be in the first line and should ends with semi-colon ;

Example:

enum Fish{

STAR,GUPPY; // Semi-colon is mandatory

public void m1(){

}

} // Valid

enum Fish{

STAR,GUPPY

public void m1(){

}

}

// Invalid

enum Fish{

public void m1(){

}

STAR, GUPPY

}

Inside enum if we are taking any extra member like a method, compulsory the first line should contain list of constant at least semi-colon.

enum Fish{

public void m1(){

}

}

//Invalid

enum Fish{

;

public void m1(){

}

}

// Valid

Anyway an empty enum is a valid Java syntax.

enum Fish{

}