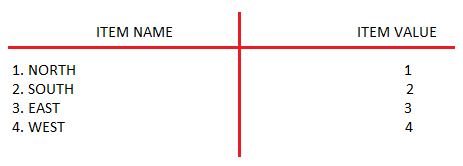
Write a Java program to print the following constant item name and its constant value.



class Menu

{

public final String ITEM\_NAME;

public final int VAL;

Menu(String i, int v)

{

this.ITEM\_NAME=i;

this.VAL=v;

}

public void display()

{

System.out.println( "\t"+ ITEM\_NAME+ "\t"+ VAL);

}

}

public class Sample

{

public static void main(String args[])

{

Menu m1=new Menu("SOUTH",0);

Menu m2=new Menu("NORTH",1);

Menu m3=new Menu("EAST",2);

Menu m4=new Menu("WEST",3);

m1.display();

m2.display();

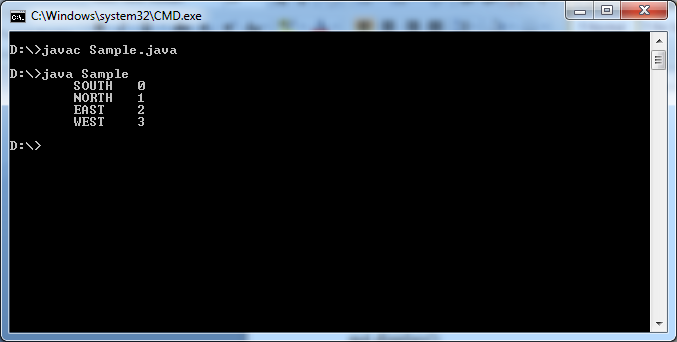
m3.display();

m4.display();

}

}

Output:



Before Java5, The java developer had to write code to display named constants and their values. This is burden to Java developer.

SunMicroSystem solved this problem by introducing “enum” concept in Java5.

**1.Enum :** Enum is a special “class” that represents group of named constants(unchangeable variables,like final variables) . To create enum in java 5, a new keyword called “enum” is introduced.

Syntax:

[Accessibility modifier] enum <enumname>

{

Only items names with “,” separator and with out data type.

;🡪The semicolon is an optional.

[ normal all 10 static and non-static members which u defined in normal class.] 🡪 This part is an optional.

}

* Enum can be easily used in switch.
* Enum can be defined inside a class and outside the class.
* Enum can have all 10 static and non-static members.
* We can not derive a subclass from enum , because it is final class.
* We can not instantiate enum using new keyword and constructor, It leads to CE. Because enum constructor is private.
* Named constants must be separated with “,” but not with semicolor”;”.
* After named constants,The semi colon is mandatory to place normal members(static and non-static members) in enum definition.
* If enum definition does not have static and non-static members then “;” is optional after the named constants.
* We can not declare explicit constructor as protected or public, because enum constructor is by default private.
* Inside Enum, we can define inner class, interface, enum.

**2.Example To Enum:-**

enum Month{

JAN,FEB

}

When you compile the enum , the compiler make some following changes in enum.

1. “enum” keyword is replaced by “final & class” keywords.

2. It extends the “java.lang.Enum”. The java.lang.Enum is abstract class. This Enum class implements Comparable & Serializable interfaces. So every custom enum also become Comparable & Serializable type.

3. The compiler creates “public static final Enum type variable” for each named constant. The variable name is same as constant name.

4. These enum type variables are initialized in static block with this enum class object by using (String,int) parameter constructor

🡪String parameter value is named constant name exactly as declared in its enum declaration.

* Int parameter value is its position in the list. The position number starts from “ZERO”.

5. The compiler add one more variable which is “$value”. It is private , static and final array. Its data type is current “enum type”.

6. The compiler places private parameterized constructor.

Private enum\_name(String name, String value)

{

Super(name,value);

}

The name and values are stored in java.lang.Enum class non-static variables “name” abd “ordinal” respectively.

7. The compiler add below two additional methods.

7.1. public static <enumtype> [] values()

This method returns all enum variable’s objects.

7.2. public static <enumtype> valueOf(String s)

Where parameter is named constant name in string form.

This method returns given enum object.

8. this class also inherits several methods from java.lang.Enum class

1. Public final String name()

Returns name of current enum exactly declared in its enum declaration.

1. Public final int orinal()

Returns current enum position.

**After compiler made changes in enum, The enum will be as**

Final class Month extends java.lang.Enum

{

Public static final Month JAN;

Public static final Month FEB;

Private static final Month[] $VALUES;

Static{

JAN=new Month(“JAN”,0);

FEB= new Month(“FEB”,1);

$VALUES=new Month[]{JAN,FEB};

}

Private Month(String name,int val)

{

Super(name,val);

}

Public static Month valueOf(String s){

Return (Month) Enum.valueOf(Month,s);

}

Public static Month[] values(){

Return (Month[])$VALUES.clone();

}

}

**Examples**

1.enum Direction

{

SOUTH,NORTH,EAST,WEST

}

public class Sample

{

public static void main(String args[])

{

Direction d[];

System.out.println("Directions are:");

d=Direction.values();

for(int i=0;i<d.length;i++)

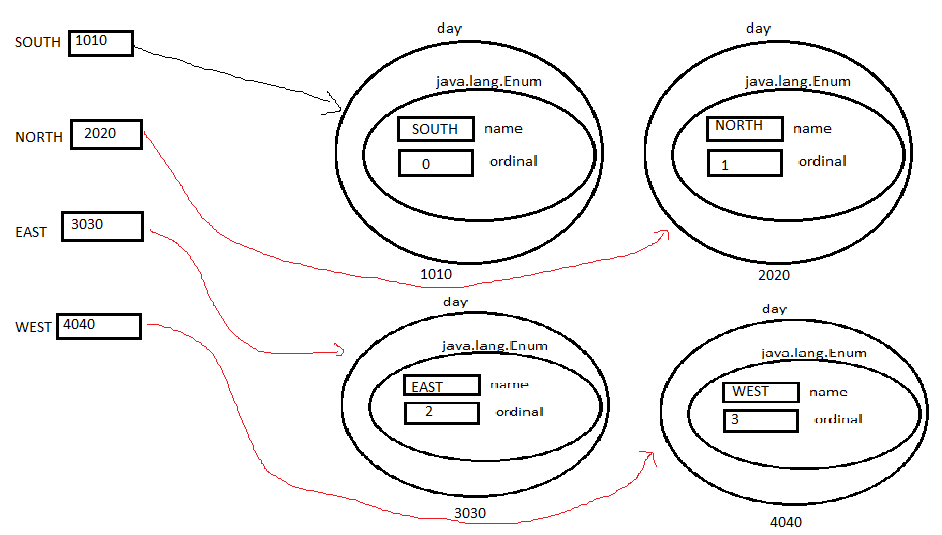
{

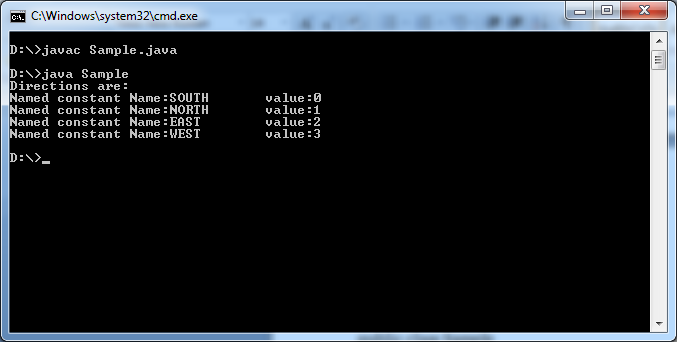
System.out.println("Namedconstant Name:"+d[i].name()+"\t"+"value:"+d[i].ordinal());

}

}

}





2.

enum TiffenItems

{

IDLY,DOSA

}

public class Sample

{

public static void main(String args[])

{

TiffenItems TI;

TI=TiffenItems.valueOf("IDLY");

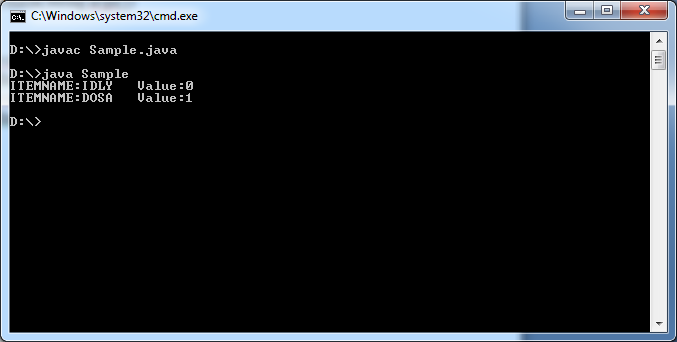
System.out.println("ITEMNAME:"+TI.name()+"\t"+"Value:"+TI.ordinal());

TI=TiffenItems.valueOf("DOSA");

System.out.println("ITEMNAME:"+TI.name()+"\t"+"Value:"+TI.ordinal());

}

}



**3.How to Assign values to named Constants in Enum:**

Syntax:

namedConstant(value);

Example: JAN(1),FEB(2).

To store these named constants values , we must follow below 3 rules in enum class.

1. Named constants must end with “;” to place variables, methods and constructors explicitly.
2. We must create non-static variable in enum with passed argument type.
3. We must define a parameterized constructor with argument type.
4. We must define a method to retrieve value of named constant.

Example:

enum TiffenItems

{

IDLY(30),DOSA(50);

int b;

TiffenItems(int x){

this.b=x;

}

public int getValue(){

return this.b;

}

}

public class Sample

{

public static void main(String args[])

{

TiffenItems TI;

TI=TiffenItems.valueOf("IDLY");

System.out.println("ITEMNAME:"+TI.name()+"\t"+"Ordinal:"+TI.ordinal()+" "+”Value:”+TI.getValue());

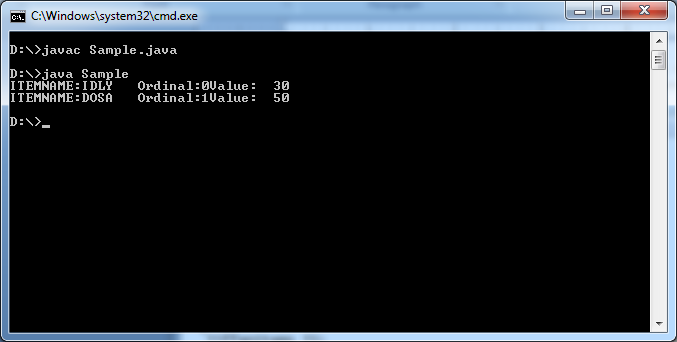
TI=TiffenItems.valueOf("DOSA");

System.out.println("ITEMNAME:"+TI.name()+"\t"+"Ordinal:"+TI.ordinal()+" "+”Value:”+TI.getValue());

}

}

Output:-



1. The following is valid .

Enum Color{

;// although you have not written named constants, the ; should be put to separate the members from empty named constant section.

Private Color()

{}

Public display(){}

}