*==========*

*Dt : 1/11/2022*

**imp*

*defining and raising User defined Exceptions:*

*step-1 : The user defined class must be extended from "java.lang.Exception" class.*

*step-2 : The user defined class must be declared with parameterized Constructor with String as parameter and this constructor will pass message to the ParentClass(java.lang.Exception)*

*step-3 : declare program-statements in try block*

*step-4 : define Exception Condition*

*step-5 : when Exception Condition is true then create object for User defined class and pass exception-msg as parameter while object creation process.*

*step-6 : Use "throw" keyword and throw the object reference onto catch block.*

*step-7 : display exception-msg from the catch block.*

---

***Ex-program :***

***IComparable.java***

```
package test;
public interface IComparable {
    public abstract int compare(int x,int y);
}
```

***DemoException1.java(MainClass)***

```
package maccess;

import test.*;

import java.util.*;

public class DemoException1 extends Exception

{

        public DemoException1(String msg)

        {

                super(msg);

        }

        public static void main(String[] args)

        {

    Scanner s = new Scanner(System.in);

    try

    {

          System.out.println("Enter the value1:");

       int v1 = s.nextInt();//Exception for NonInteger value

       System.out.println("Enter the value2:");
```

```
int v2 = s.nextInt();//Exception for NonInteger value
if(v1<=0 || v2<=0)//Exception Condition
{
    DemoException1 de = new DemoException1("Invalid values");
    throw de;//throwing object reference onto catch block
}
if(v1==v2)//Exception Condition
{
    DemoException1 de = new DemoException1("Equal Values..");
    throw de;//throwing object reference onto catch block
}
System.out.println("====Choice====");
System.out.println("1.GreaterValue\n2.SmallerValue");
System.out.println("Enter the Choice:");
switch(s.nextInt())//Exception for NonInteger value
{
case 1:
   //GreaterValue class as LambdaExpression
    IComparable gv = (int x,int y)->
                {
                        if(x>y) return x;
                        else return y;
                };
    int r1 = gv.compare(v1,v2);
```

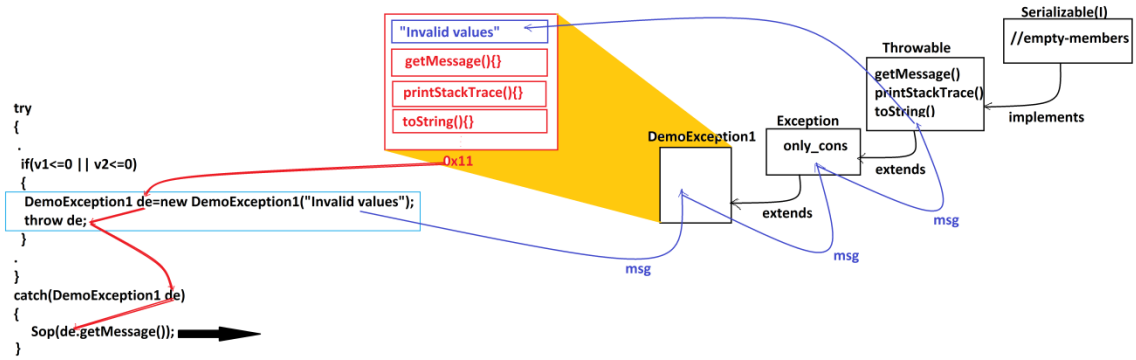
```
        System.out.println("GreaterValue:"+r1);
        break;
    case 2:
        //SmallerValue class as LambdaExpression
        IComparable sv = (int x,int y)->
                        {
                                if(x<y) return x;
                                else return y;
                        };
                int r2 = sv.compare(v1,v2);
                System.out.println("SmallerValue:"+r2);
        break;
    default:
        System.out.println("Invalid Choice...");
    }//end of switch

}//end of try
catch(InputMismatchException ob)//PreDefined Exception
{
        System.out.println("Enter only Integer values...");
}
catch(DemoException1 de)//user defined Exception
{
        System.out.println(de.getMessage());
```

```
    }
    finally
    {
        s.close();
    }
        }
}
```

==========

***Diagram:***




==========

***Ex-program:***

***Convert Employee details application in to Exception Handling process.***

***Note:***

***=>EmpId must 4 Characters(Alpha numeric)***

***=>EmpDesg must be in SE or TE or ME***

***=>bSal must min 12000/-***

*Designation.java*

```
package test;
public class Designation {
    public boolean verify(String desg) {
     return switch(desg) {
     case "SE":yield true;
     case "TE":yield true;
     case "ME":yield true;
     default : yield false;
     };
    }
}
```

*DemoMethods2.java(MainClass)*

```
package maccess;

import java.util.*;

import test.Designation;

public class DemoException2 extends Exception

{

  public DemoException2(String msg)

  {

       super(msg);

  }

       public static void main(String[] args)

       {

        Scanner s = new Scanner(System.in);

        try

        {

                System.out.println("Enter the EmpId:");

                String eId = s.nextLine();
```

```
if(eId.length()!=4)//Exception condition
{
        DemoException2 de = new DemoException2("Invalid empId");
        throw de;
}
System.out.println("Enter the empName:");
String eName = s.nextLine();
System.out.println("Enter the empDesg:");
String desg = s.nextLine().toUpperCase();
Designation dg = new Designation();
boolean k = dg.verify(desg);
if(!k)//Exception Condition
{
        DemoException2 de = new DemoException2("Invalid desg");
        throw de;
}
System.out.println("Enter the bSal:");
int bSal = s.nextInt();
if(bSal<12000)//Exception condition
{
        DemoException2 de = new DemoException2("Invalid bSal");
        throw de;
}
float totSal = bSal+(0.93F*bSal)+(0.63F*bSal);
```

```
    System.out.println("====Employee Details===");
    System.out.println("EmpId:"+eId);
    System.out.println("EmpName:"+eName);
    System.out.println("EmpDesg:"+desg);
    System.out.println("EmpBSal:"+bSal);
    System.out.println("EmpTotSal:"+totSal);

  }//end of try
  catch(DemoException2 de)
  {
        System.out.println(de.getMessage());
  }
  catch(InputMismatchException ime)
  {
        System.out.println("Enter only Integer value..");
  }
  finally
  {
    s.close();
  }

  }

}
```

***==========***

***Assignment:***

***Convert Student Application into Exception Handling process.***

***==========***

****imp***

***define "Throwable"?***

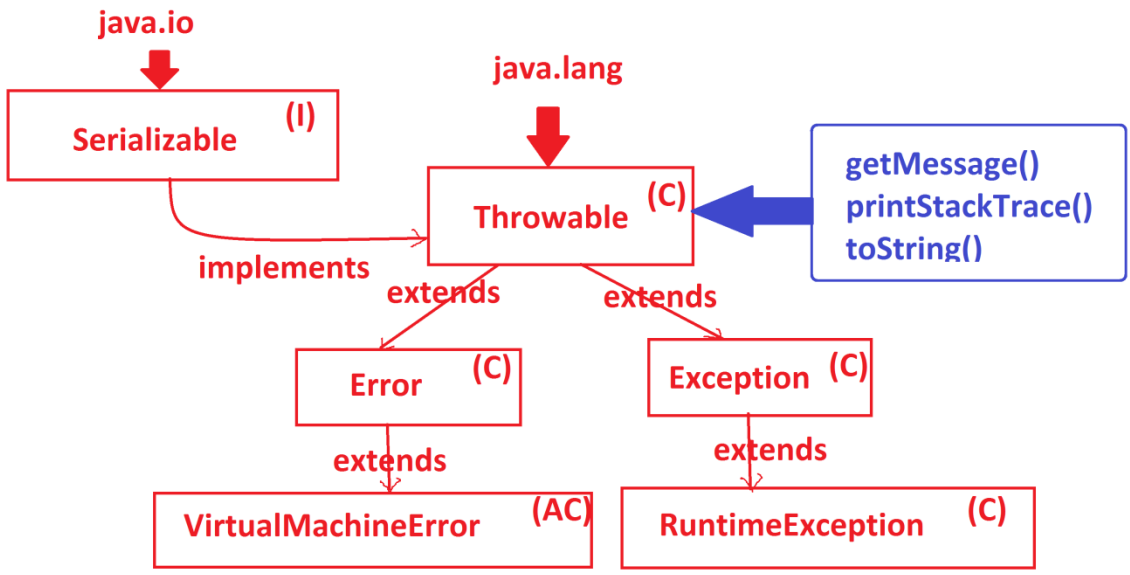
***=>"Throwable" is a class from java.lang package and which is root of exception handling process.***

***=>This "Throwable" class is extended into the following SubClasses:***

***1.Error class***

***2.Exception class***

***Hierarchy of "Throwable" class:***




***==========***