Dt : 3/11/2022

**imp*

(ii)User defined checked exceptions:

=>The checked exceptions which are defined and raised by the programmer are known as User defined Checked exceptions.

=>We use the following steps to define and raise User defined Checked Exceptions:

step-1 : add "throws" keyword to user defined method signature and raise the exception at method call

Step-2 : declare "throw" keyword in catch block and perform "re-throwing" process.

Ex:(Convert BankTransaction application into Exception handling process)

Balance.java

```
package test;
public class Balance {
    public double bal=2000;
    public double getBalance() {
        return bal;
    }
}
```

CheckPinNo.java

```
package test;
```

```
public class CheckPinNo {
    public boolean verify(int pinNo) {
      return switch(pinNo) {
      case 1111 : yield true;
      case 2222 : yield true;
      case 3333 : yield true;
      default : yield false;
      };
    }
}
```

Transaction.java

```
package test;
public interface Transaction {
   public static final Balance b = new
Balance();
   public abstract void process(int
amt)throws WithDraw;
}
```

WithDraw.java

```
package test;
public class WithDraw extends
Exception implements Transaction
{
   public WithDraw() {}
   public WithDraw(String msg)
   {
```

```
        super(msg);
    }
     public void process(int amt)throws
WithDraw
     {
        try
        {
            if(amt>b.bal)//Exception
condition
            {
                WithDraw wd = new
WithDraw("Insufficient fund");

//Para_Con_Call
                throw wd;
            }
            System.out.println("Amt
withDrawn:"+amt);
            b.bal=b.bal-amt;
            System.out.println("Balance
amt:"+b.getBalance());

   System.out.println("Transaction
Completed...");
        }//end of try
        catch(WithDraw wd)
        {
            throw wd;//re-throwing
```

```
        }
    }
}
```

Deposit.java

```
package test;
public class Deposit implements
Transaction{
    public void process(int amt)
    {
        System.out.println("Amt
deposited:"+amt);
        b.bal=b.bal+amt;
        System.out.println("Balance
amt:"+b.getBalance());
        System.out.println("Transaction
completed...");
    }
}
```

DemoException4.java(MainClass)

```
package maccess;

import java.util.*;

import test.*;

public class DemoException4 extends Exception

{

        public DemoException4(String msg)
```

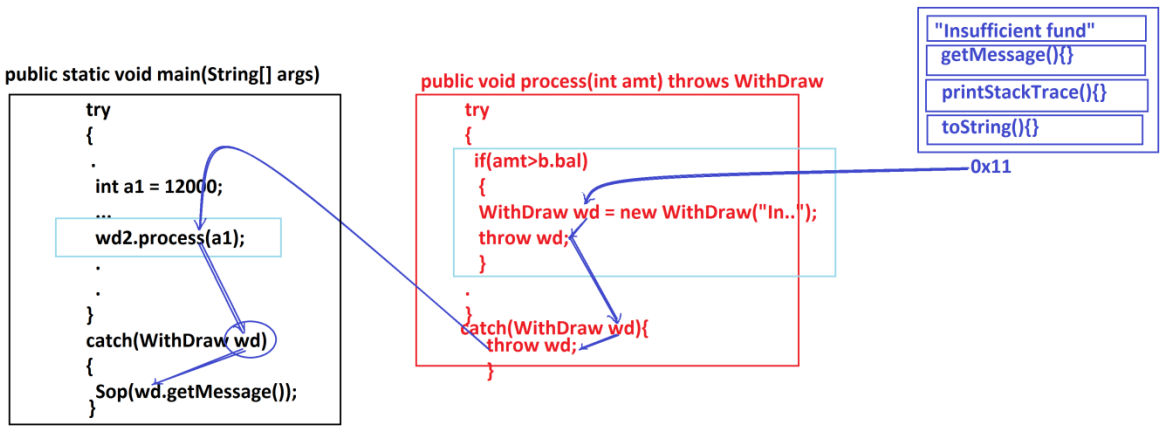
```
    {
            super(msg);
    }
    public static void main(String[] args)
    {
Scanner s = new Scanner(System.in);
int count=0;
xyz:
     while(true)
    {
            try
            {
                    System.out.println("Enter the pinNo:");
                    int pinNo = s.nextInt();
                    CheckPinNo cpn = new CheckPinNo();
                    boolean k = cpn.verify(pinNo);
                    if(!k)//Exception Condition
                    {
                            DemoException4 de = new DemoException4("Invalid pinNo");
                            throw de;
                    }
                    System.out.println("====Choice====");
                    System.out.println("1.WithDraw\n2.Deposit");
                    System.out.println("Enter the choice:");
                    switch(s.nextInt())
```

```
{
case 1:
        System.out.println("Enter the amt:");
        int a1 = s.nextInt();
        if(!(a1>0 && a1%100==0))//Exception Condition
        {
            DemoException4 de = new DemoException4("Invalid amt");
            throw de;
        }
        WithDraw wd2 = new WithDraw();//0_para_Con_call
        wd2.process(a1);//method Call
        break xyz;
case 2:
        System.out.println("Enter the amt:");
        int a2 = s.nextInt();
        if(!(a2>0 && a2%100==0))//Exception Condition
        {
            DemoException4 de = new DemoException4("Invalid amt");
            throw de;
        }
        Deposit dp = new Deposit();
        dp.process(a2);
        break xyz;
default:
        System.out.println("Invalid Choice...");
```

```
                    break xyz;
            }//end of switch
    }//end of try
    catch(DemoException4 de)
    {
            System.out.println(de.getMessage());
            if(de.getMessage().equals("Invalid pinNo"))
            {
                    count++;
            if(count==3)//Nested Simple if
            {
                    System.out.println("Transaction blocked...");
                    break xyz;
            }
            }//end of if
            else
            {
                    break xyz;
            }
    }
    catch(WithDraw wd)
    {
            System.out.println(wd.getMessage());
            break xyz;
    }
```

```
        }//end of loop
      }
}
```

Diagram:



==========

faq:

define "Exception re-throwing process"?

=>The process of declaring "throw" keyword in catch block and throwing the exception is known as "Exception re-throwing process".

=>In Exception re-throwing process the object reference is thrown to the catch block of try-block where method call is available.

faq:

define Exception Propagation?

=>In Exception re-throwing process the exception is moved from one method to another method is known as Exception Propagation.

==========

faq:

wt is the diff b/w

(i)throw

(ii)throws

(i)throw:

=>"throw" keyword is used to throw the object reference onto catch

block in the process of handling the exception.

(ii)throws:

="throws" keyword added with method signature and raises the exception

at method call.

==========

Assignment-1:

Convert BankTransaction application with Anonymous classes into

Exception handling process.

Assignment-2:

Convert BankTransaction application with LambdaExpressions into

Exception handling process.

==========