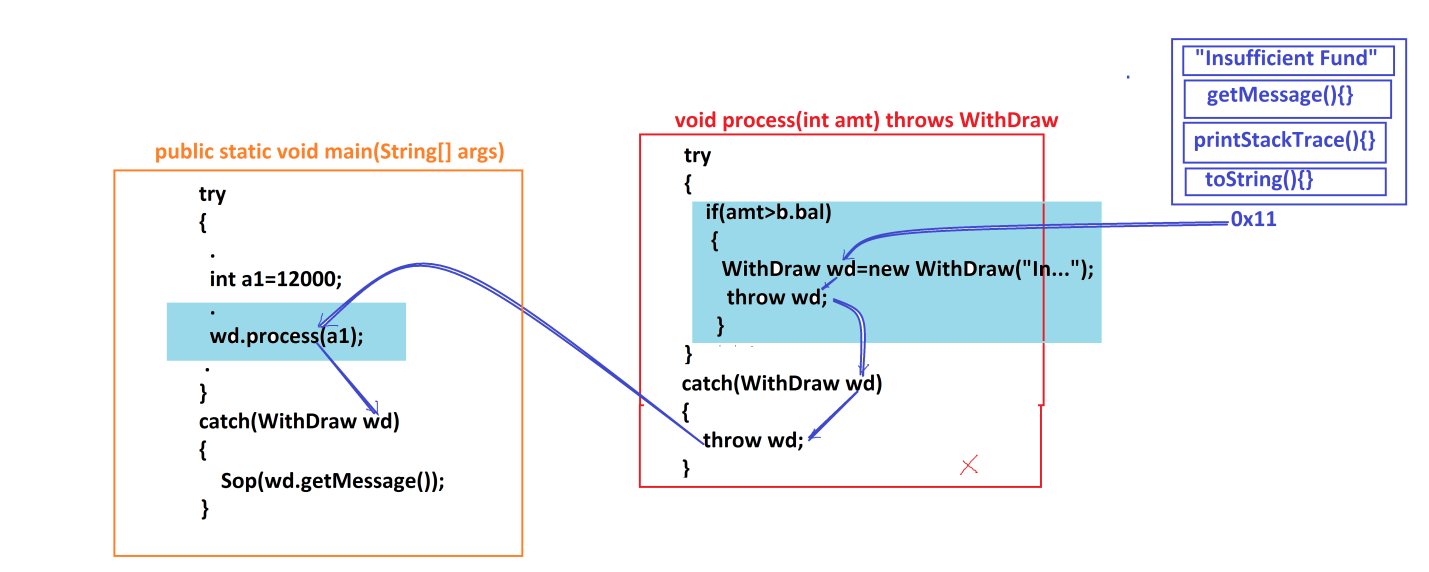
***Dt : 2/11/2022***

***Diagram:***

******

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

***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 exception.***

***(ii)throws:***

***=>"throws" keyword added with method signature and raise the exception***

***at method\_call,in this process the exception is ignored from current running***

***method.***

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

***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.***

***Note:***

***=>In Exception re-throwing process the object reference is moved 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.***

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

***\*imp***

***Handling Multiple Exceptions:***

***=>we use the following two ways to handle Multiple Exceptions:***

***(i)Using multiple catch blocks declared to a try block***

***syntax:***

***try***

***{***

***Exception1;***

***Exception2;***

***}***

***catch(Exception1 ob1)***

***{***

***//msg***

***}***

***catch(Exception2 ob2)***

***{***

***//msg***

***}***

***(ii)From Java7 version onwards we can use single catch block to handle***

***multiple exceptions***

***syntax:***

***try***

***{***

***Exception1;***

***Exception2;***

***}***

***catch(Exception1 | Exception2 | ... ob)***

***{***

***//msg***

***}***

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

***\*imp***

***define try-with-resource statement?***

***=>try-with-resource statement introduced by Java7 version and in which***

***we declare resources with try.***

***syntax:***

***try(resource1;resouce2;...)***

***{***

***//statements***

***}***

***Ex:***

***try(Scanner s = new Scanner(System.in);)***

***{***

***//statements***

***}***

***Advantage:***

***=>In try-with-resource statement the resources are closed automatically,***

***which means "no need use finally block to close resources".***

***Note:***

***=>catch is optional block in try-with-resource statement.***

***=>The resources which are declared with try must be implementations of***

***"java.lang.AutoCloseable" interface.***

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

***\*imp***

***define Enhanced try-with-resource statement?***

***=>Enhanced try-with-resource statement introduced by Java9 version and***

***in which the resources are declared outside the try and resource reference***

***variables are declared with try.***

***syntax:***

***resource1;resouce2;...***

***try(res1\_var;res2\_var;...)***

***{***

***//statements***

***}***

***Ex:***

***Scanner s = new Scanner(System.in);***

***try(s;)***

***{***

***//statements***

***}***

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

***Assignment1:(Solution)***

***Convert BankTransaction application with anonymous InnerClasses into***

***Exception Handling process.***

***Note:***

***=>In the process handling exception in Anonymous InnerClasses we handle***

***"java.lang.Exception" directly,because the class names are not available.***

***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 Exception;***

***}***

***DemoException5.java(MainClass)***

***package maccess;***

***import java.util.\*;***

***import test.\*;***

***@SuppressWarnings("serial")***

***public class DemoException5 extends Exception***

***{***

***public DemoException5(String msg) {***

***super(msg);***

***}***

***public static void main(String[] args) {***

***try(Scanner s = new Scanner(System.in);)//Java7***

***{***

***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***

***{***

***DemoException5 de = new DemoException5("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***

***{***

***DemoException5 de = new DemoException5("Invalid amt");***

***throw de;***

***}***

***Transaction wd2 = new Transaction()***

***{***

***public void process(int amt) throws Exception***

***{***

***try***

***{***

***if(amt>b.bal)//Exception condition***

***{***

***Exception wd = new Exception("Insuffiecient 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 Successfull...");***

***}//end of try***

***catch(Exception wd)***

***{***

***throw wd;//re-throwing Exception***

***}***

***}***

***};***

***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))***

***{***

***DemoException5 de = new DemoException5("Invalid amt");***

***throw de;***

***}***

***Transaction dp = new 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...");***

***}***

***};***

***dp.process(a2);***

***break xyz;***

***default:***

***System.out.println("Invalid Choice...");***

***break xyz;***

***}//end of switch***

***}//end of try***

***catch(Exception de)***

***{***

***System.out.println(de.getMessage());***

***if(de.getMessage().equals("Invalid PinNo"))***

***{***

***count++;***

***if(count==3)***

***{***

***System.out.println("Transaction blocked...");***

***break xyz;***

***}***

***}//end of if***

***else***

***{***

***break xyz;***

***}***

***}***

***}//end of loop***

***}//try-with-resource***

***}***

***}***

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

***Assignment2:***

***Convert BankTransaction application with LambdaExpression into***

***Exception Handling process.***

***Note:***

***=>In the process handling exception in Lambda Expressions we handle***

***"java.lang.Exception" directly,because the No class names and No method***

***names.***

***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 Exception;***

***}***

***DemoException6.java(MainClass)***

***package maccess;***

***import java.util.\*;***

***import test.\*;***

***@SuppressWarnings("serial")***

***public class DemoException6 extends Exception***

***{***

***public DemoException6(String msg) {***

***super(msg);***

***}***

***public static void main(String[] args) {***

***Scanner s = new Scanner(System.in);***

***try(s;)//Java9***

***{***

***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***

***{***

***DemoException6 de = new DemoException6("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***

***{***

***DemoException6 de = new DemoException6("Invalid amt");***

***throw de;***

***}***

***Transaction wd2 = (int amt)->***

***{***

***try***

***{***

***if(amt>Transaction.b.bal)//Exception condition***

***{***

***Exception wd = new Exception("Insuffiecient fund");//Para\_con\_call***

***throw wd;***

***}***

***System.out.println("Amt withDrawn:"+amt);***

***Transaction.b.bal=Transaction.b.bal-amt;***

***System.out.println("Balance amt:"+Transaction.b.getBalance());***

***System.out.println("Transaction Successfull...");***

***}//end of try***

***catch(Exception wd)***

***{***

***throw wd;//re-throwing Exception***

***}***

***};***

***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))***

***{***

***DemoException6 de = new DemoException6("Invalid amt");***

***throw de;***

***}***

***Transaction dp = (int amt)->***

***{***

***System.out.println("Amt deposited:"+amt);***

***Transaction.b.bal=Transaction.b.bal+amt;***

***System.out.println("Balance amt:"+Transaction.b.getBalance());***

***System.out.println("Transaction Completed...");***

***};***

***dp.process(a2);***

***break xyz;***

***default:***

***System.out.println("Invalid Choice...");***

***break xyz;***

***}//end of switch***

***}//end of try***

***catch(Exception de)***

***{***

***System.out.println(de.getMessage());***

***if(de.getMessage().equals("Invalid PinNo"))***

***{***

***count++;***

***if(count==3)***

***{***

***System.out.println("Transaction blocked...");***

***break xyz;***

***}***

***}//end of if***

***else***

***{***

***break xyz;***

***}***

***}***

***}//end of loop***

***}//try-with-resource***

***}***

***}***

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