**Ex No: Exception and File Handling**

**Date:**

**AIM**

The aim of the provided Java programs is to implement the Exception and File Handling

**STEP 1: Null Pointer Exception Program (NullException3704.java):**

* Initialize a LocalTime and LocalDate and print them.
* Create an object of the ExceptionShow class.
* Call the Method function with a null parameter, causing a Null Pointer Exception.
* Handle the exception and print a message indicating the occurrence of a Null Pointer Exception.

**STEP 2: Array Out of Bounds Exception Program:**

* Initialize a LocalTime and LocalDate and print them.
* Create an integer array of size 8 and input values.
* Attempt to access elements beyond the array size in a loop.
* Handle the exception and print a message indicating the occurrence of an Array Out of Bounds Exception.

**STEP 3: Arithmetic Exception Program:**

* Attempt to perform a division operation where the divisor is zero.
* Handle the exception and print a message indicating the occurrence of an Arithmetic Exception.

**STEP 4: File Not Found Exception Program:**

* Attempt to read from an input file and write to an output file.
* Handle the exception and print a message indicating the occurrence of a File Not Found Exception.

**STEP 5: Stack Overflow Exception Program:**

* Call a recursive method that does not have a base case, causing a Stack Overflow Exception.
* Handle the exception and print a message indicating the occurrence of a Stack Overflow Exception.

**STEP 6: Email Exception Program:**

* Input an email and validate it using the emailVal method.
* If the email is invalid, throw a custom EmailException.
* Handle the exception and print a message indicating the occurrence of an Email Exception.

**STEP 7: File Example Program:**

* Open input and output streams for file handling.
* Read lines from the input file and print them to the console.
* Write the content of the input file to the output file.
* Close the streams.

**STEP 8: Number Format Exception Program:**

* Attempt to convert a string ("Bhumisvara") to an integer.
* Handle the exception and print a message indicating the occurrence of a Number Format Exception.

**STEP 9: Stack Overflow Program with Catching Exception:**

* Call a recursive method without a base case, causing a Stack Overflow Exception.
* Catch the exception and print a message indicating the occurrence of a Stack Overflow Exception.

**STEP 10:Stack Overflow Program with Try-Catch Block:**

* Call a recursive method without a base case, causing a Stack Overflow Exception.
* Catch the exception using a try-catch block and print a message indicating the occurrence of a Stack Overflow Exception.

**STEP 11: Number Format Exception Program with Finally Block:**

* Attempt to convert a non-numeric string to an integer.
* Handle the exception and print a message indicating the occurrence of a Number Format Exception.
* Use a finally block to print a message indicating the end of the program.

1. **Null pointer Exception:**

**PROGRAM**

import java.time.LocalDate;

import java.time.LocalTime;

public class NullException3704

{

public static void main(String[] args)

{

System.out.println("Bhumisvara");

System.out.println("2021503704");

LocalTime lt=LocalTime.now();

LocalDate ld=LocalDate.now();

System.out.println(lt);

System.out.println(ld);

ExceptionShow obj=new ExceptionShow();

Method(null);

}

public static void Method(ExceptionShow obj)

{

try

{

obj.example(obj);

}

catch(NullPointerException e)

{

e.printStackTrace();

}

finally

{

System.out.println("NULL POINTER EXECUTION");

}

}

}

class ExceptionShow

{

public String show()

{

return "Exception string executed";

}

public void example(ExceptionShow obj) throws NullPointerException

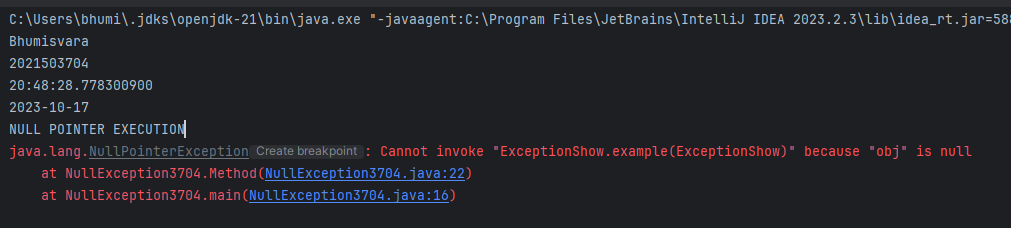
{

obj.show();

}

}

OUTPUT



1. **Array out of bound**

**Program.**

import java.time.LocalDate;

import java.time.LocalTime;

import java.util.\*;

public class Bhumisvara {

public static void main(String[] args) {

System.out.println("Bhumisvara");

System.out.println("2021503704");

LocalTime lt=LocalTime.now();

LocalDate ld=LocalDate.now();

System.out.println(lt);

System.out.println(ld);

int[] x=new int[8];

Scanner sc=new Scanner(System.in);

for(int i=0;i<8;i++)

{

x[i]=sc.nextInt();

}

try

{

for(int i=0;i<10;i++)

{

System.out.println(x[i]);

}

}

catch(Exception e)

{

e.printStackTrace();

}

finally

{

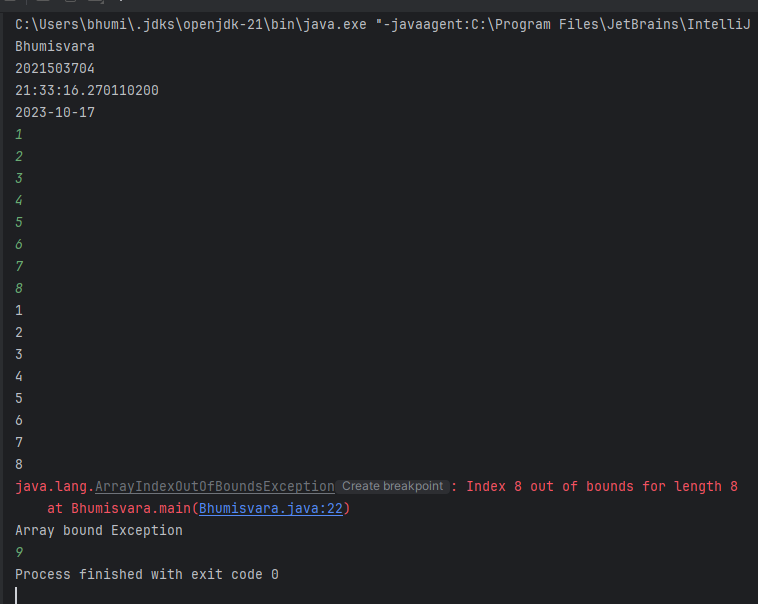
System.out.println("Array bound Exception");

}

}

}

**OUTPUT.**



1. **Arithmetic Exception**

**Program**

public class Bhumisvara {

public static void main(String[] args) {

System.out.println("Bhumisvara");

System.out.println("2021503704");

try

{

int a=10;

int b=0;

int c=a/b;

}

catch(Exception e)

{

e.printStackTrace();

}

finally

{

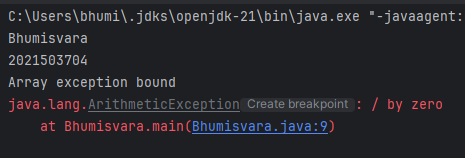
System.out.println("Array exception bound");

}

}

}

**OUTPUT**



1. **File not found Exception**

**Program**

import java.io.\*;

public class Bhumisvara

{

public static void main(String[] args) throws IOException

{

System.out.println("Bhumisvara");

System.out.println("2021503704");

try{

FileInputStream in=new FileInputStream("File.txt");

FileOutputStream out=new FileOutputStream("Output.txt");

int i=0;

while((i=in.read())!=-1)

{

out.write(i);

}

}

catch(FileNotFoundException e)

{

e.printStackTrace();

}

finally

{

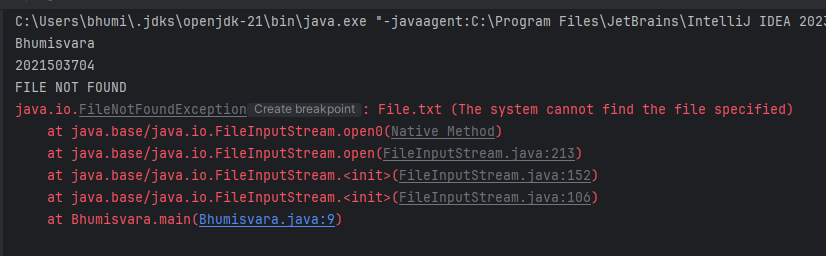
System.out.println("FILE NOT FOUND");

}

}

}

**OUTPUT**



1. **Stack overflow Exception**

**Program**

public class Bhumisvara

{

public static void main(String[] args)

{

System.out.println("Bhumisvara");

System.out.println("2021503704");

try

{ overflow(1);

}

catch(StackOverflowError e)

{

e.printStackTrace();

}

finally

{

System.out.println("STACKOVERFLOW");

}

}

public static void overflow(int n)

{

//System.out.println(n);

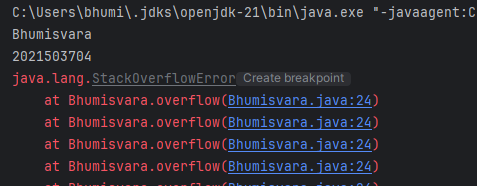
if(n>0)

overflow(n+1);

}

}

**OUTPUT**



1. **Email Exception**

**Program**

import java.util.Scanner;

public class Bhumisvara

{

public static void main(String[] args)

{

System.out.println("Bhumisvara");

System.out.println("2021503704");

System.out.print("Enter the Email:");

Scanner in=new Scanner(System.in);

String email=in.nextLine();

try

{

emailVal(email);

System.out.println("Valid Email");

}

catch(EmailException e)

{

e.printStackTrace();

}

finally

{

System.out.println("Email");

}

}

public static void emailVal(String email) throws EmailException

{

if(!email.contains("@")) {

throw new EmailException("Invalid Email @ not present");

}

}

}

class EmailException extends Exception

{

public EmailException(String s)

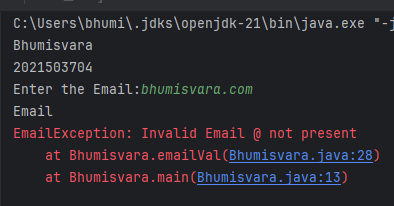
{

super(s);

}

}

**OUTPUT**



1. **File Example**

**Program**

import java.io.\*;

public class Bhumisvara

{

public static void main(String[] args) throws IOException

{

System.out.println("Bhumisvara");

System.out.println("2021503704");

FileInputStream in=new FileInputStream("F:\\input.txt");

FileOutputStream out=new FileOutputStream("F:\\output.txt");

int u=0;

InputStreamReader ir=new InputStreamReader(in,"UTF-8");

OutputStreamWriter or=new OutputStreamWriter(out,"UTF-8");

BufferedReader br=new BufferedReader(ir);

String s;

while ((s = br.readLine()) != null) {

System.out.println(s);

}

while(((u=in.read())!=-1))

{

out.write(u);

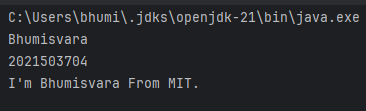
}

br.close();

}

}

**OUTPUT**



1. **Number Format Exception:**

**Program**

import java.io.\*;

public class Bhumisvara

{

public static void main(String[] args) {

System.out.println("Bhumisvara");

System.out.println("2021503704");

try

{

String s="Bhumisvara";

int i=Integer.parseInt(s);

}

catch(NumberFormatException e)

{

e.printStackTrace();

}

finally

{

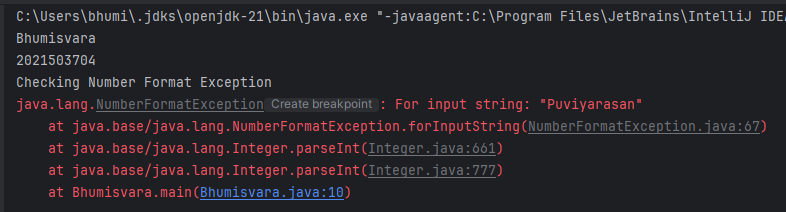
System.out.println("Checking Number Format Exception");

}

}

}

**OUTPUT**



1. **Stack Over Flow**

**Program**

import java.time.\*;

public class Bhumisvara {

public static void recursiveMethod() {

recursiveMethod(); // This will lead to a StackOverflowError

}

public static void main(String[] args) {

System.out.println("Bhumisvara");

System.out.println("2021503704");

try {

recursiveMethod();

} catch (StackOverflowError e) {

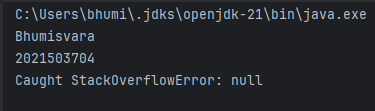
System.out.println("Caught StackOverflowError: " + e.getMessage());

}

}

}

**OUTPUT**



**RESULT**

Thus, The Exception and File Handling program has been successfully implemented.