1. Modify the above code to handle the exception that may arise using try catch block?

public class TryCatchExample {

public static void main(String args[]){

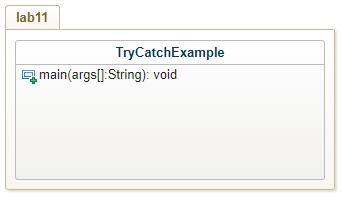
int data = 50/0;

System.*out*.println(“Rest of the code”);

}

}

**Class diagram:**



**Program:**

**package** lab11;

**public** **class** TryCatchExample {

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

{

**try** {

**int** data = 50/0;

}

**catch**(ArithmeticException e)

{

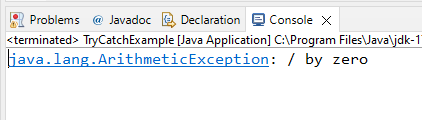
System.***out***.println(e);

}

}

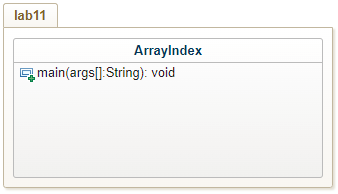
}

**Output:**

****

1. Write a program that meets the following requirements creates an array with 100 randomly chosen integers prompts the user to enter the index of the array and then displays the corresponding element value if the specified index is out of bounds, display the message out of bounds. (ArrayIndexOutOfBoundsException)

**Class diagram:**

****

**Program:**

**package** lab11;

**public** **class** ArrayIndex {

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

{

**int** arr[] = **new** **int**[100];

**try** {

System.***out***.println(arr[100]);

}

**catch**(ArrayIndexOutOfBoundsException e){

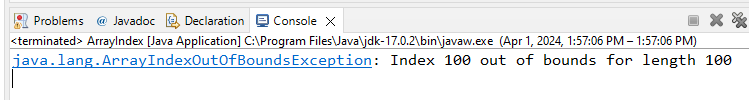
System.***out***.println(e);

}

}

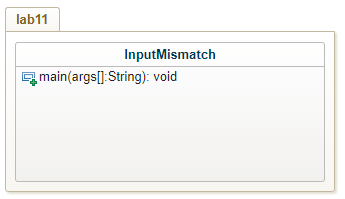
}

**Output:**



1. Write a program that prompts the user to read two integers and display their sum your program should prompt the user to read the number again if the input is incorrect.(InputMismatchException).

**Class diagram:**

****

**Program:**

**package** lab11;

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

**public** **class** InputMisMatch {

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

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

**int** a = 0;

**int** b = 0;

**boolean** c = **false**;

**do**

{

System.***out***.println("Enter a,b numbers:");

**try** {

a = in.nextInt();

b = in.nextInt();

c = **false**;

}

**catch**(InputMismatchException e){

System.***out***.println(e);

c = **true**;

in.nextLine();

}

}**while**(c);

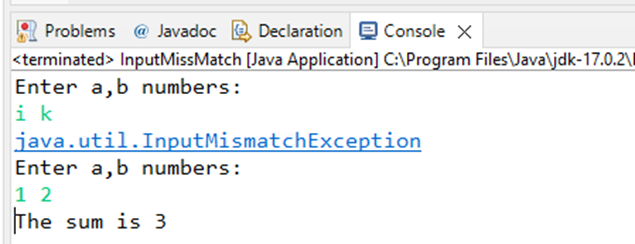
System.***out***.println("The sum is "+(a+b));

in.close();

}

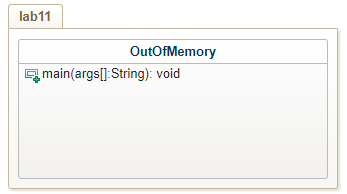
}

**Output:**

****

1. Write a program that cause the JVM to throw an OutOfMemory Error and catches and handles this error.

**Class diagram:**

****

**Program:**

**package** lab11;

**public** **class** OutOfMemory {

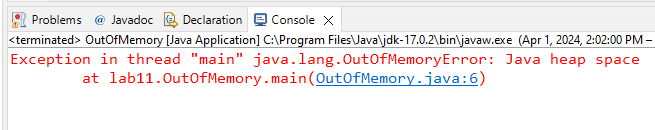
**public** **static** **void** main(String[] args) **throws** Exception{

Integer[] array = **new** Integer[1000000000];

}

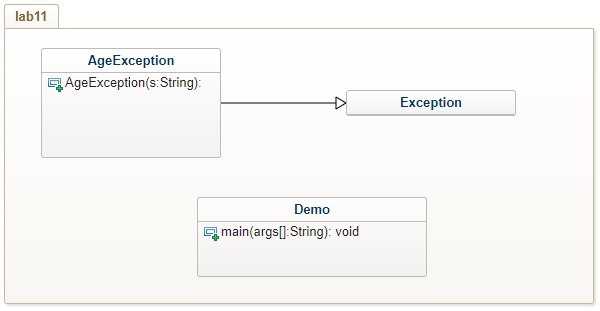
}

**Output:**

****

1. Election committee wants to check whether the voter is eligible to vote or not. The person can vote if his age is greater than 18. Help the Election committee by developing a code which arises exception if the voter age is less than 18 then print the exception and “VOTER IS NOT ELIGIBLE TO VOTE” otherwise print “NOT ELIGIBLE TO VOTE”.

**Class diagram:**

****

**Program:**

**package** lab11;

**public** **class** AgeException **extends** Exception {

**public** AgeException(String s) {

**super**(s);

}

}

**package** lab11;

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

**public** **class** Demo {

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

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

System.***out***.print("Enter your age: ");

**int** age = in.nextInt();

**try** {

**if**(age<=18)

**throw** **new** AgeException("VOTER IS NOT ELIGIBLE TO VOTE");

**else**

System.***out***.println("VOTER IS ELIGIBLE TO VOTE");

}

**catch**(AgeException a) {

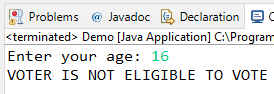
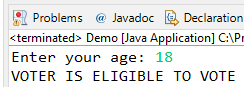
System.***out***.println(a.getMessage());

}

}

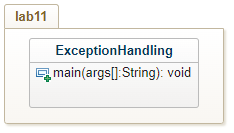
}

**Output:**

** **

1. Develop a Java program that uses multiple catch statements to handle different types of exceptions test it with scenarios that will throw

* Number format
* Null Pointer Exception
* Array Index Out of Bounds



**package** lab11;

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

**public** **class** ExceptionHandling {

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

String text = **null**;

**int**[] numbers = **new** **int**[5];

**try** {

// Test for NumberFormatException

**int** convertedValue = Integer.*parseInt*("hello"); // Trying to convert a string to an integer

// Test for NullPointerException

System.***out***.println(text.length()); // Trying to access length of a null object

// Test for ArrayIndexOutOfBoundsException

numbers[10] = 20; // Trying to access an element outside the array bounds

} **catch** (NumberFormatException e) {

System.***out***.println("NumberFormatException: " + e.getMessage());

} **catch** (NullPointerException e) {

System.***out***.println("NullPointerException: " + e.getMessage());

} **catch** (ArrayIndexOutOfBoundsException e) {

System.***out***.println("ArrayIndexOutOfBoundsException: " + e.getMessage());

} **finally** {

System.***out***.println("This block always executes regardless of exceptions.");

}

}

}

