Started on	Tuesday, 2 January 2024, 7:07 AM
State	Finished
Completed on	Tuesday, 2 January 2024, 7:42 AM
Time taken	35 mins 8 secs

Minimum eligibility to apply voter id is

1. Age should be above 18 years old
2. Belongs to the Country-India.

Write a program to approve or display suitable exceptions whenever a person tries to apply for a voter id.

Create two exception InvalidAgeForVoterIdException and InvalidCountryForVoterIdException to handle above scenarios.

Create Main class to get Name, Age and Country from user , check the data and approve or throw suitable exception.

Note : Refer sample output for exact statement and format.

Input Format :

Name
Age
Country

Output Format :

Approved or suitable exception

Note : Sample test cases for exact exception statement

Sample testcases :

Testcase 1 Input

Guru
33
India

Testcase 1 Output

Approved

Testcase 2 Input

Guru
2
India

Testcase 2 Output

InvalidAgeForVoterIdException: Age should be more than 18 years old

Testcase 3 Input

Guru
-3
95

Testcase 3 Output

InvalidAgeForVoterIdException: Invalid age

Testcase 4 Input

Guru
33
China

Testcase 4 Output

InvalidCountryForVoterIdException: Country must be India

Testcase 5 Input

Guru
33
India143

Testcase 5 Output

InvalidCountryForVoterIdException: Invalid country

For example:

Input	Result
Guru 33 India	Approved
Guru 2 India	InvalidAgeForVoterIdException: Age should be more than 18 years old
Guru -3 India	InvalidAgeForVoterIdException: Invalid age
Guru 33 China	InvalidCountryForVoterIdException: Country must be India
Guru 33 India143	InvalidCountryForVoterIdException: Invalid country

Answer: (penalty regime: 0 %)

```java
import java.util.*;
class InvalidAgeForVoterIdException extends Exception
{
    public InvalidAgeForVoterIdException(String str)
    {
        super(str);
    }
}
class InvalidCountryForVoterIdException extends Exception
{
    public InvalidCountryForVoterIdException(String str)
    {
        super(str);
    }
}
public class VoterIdValidation
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        String name = sc.nextLine();
        int age = sc.nextInt();
        sc.nextLine();
        String country = sc.nextLine();
        int flag=0;

        try
        {
            if(age>=0 && age<=18){
                throw new InvalidAgeForVoterIdException("Age should be more than 18 years old");
            }
            else if(age<0)
            {
                throw new InvalidAgeForVoterIdException("Invalid age");
            }
            for(char i:country.toCharArray())
            {
                if(i>='0' && i<='9')
                {
```

```
40              flag=1;
41              break;
42          }
43      }
44      if(flag==1)
45      {
46          throw new InvalidCountryForVoterIdException("Invalid country");
47      }
48      else if(flag==0)
49      {
50          if(!(country.equals("India")))
51          {
52              throw new InvalidCountryForVoterIdException("Country must be India");
```

	Input	Expected	Got	
✔	Guru 33 India	Approved	Approved	✔
✔	Guru 2 India	InvalidAgeForVoterIdException: Age should be more than 18 years old	InvalidAgeForVoterIdException: Age should be more than 18 years old	✔
✔	Guru -3 India	InvalidAgeForVoterIdException: Invalid age	InvalidAgeForVoterIdException: Invalid age	✔
✔	Guru 33 China	InvalidCountryForVoterIdException: Country must be India	InvalidCountryForVoterIdException: Country must be India	✔
✔	Guru 33 India143	InvalidCountryForVoterIdException: Invalid country	InvalidCountryForVoterIdException: Invalid country	✔

Passed all tests! ✔

Exception handling is the process of responding to the occurrence, during computation, of exceptions – anomalous or exceptional conditions requiring special processing – often changing the normal flow of program execution. (Wikipedia)

Java has built-in mechanism to handle exceptions. Using the *try* statement we can test a block of code for errors. The *catch* block contains the code that says what to do if exception occurs.

This problem will test your knowledge on try-catch block.

You will be given two integers x and y as input, you have to compute x/y. If x and y are not 32 bit signed integers or if y is zero, exception will occur and you have to report it. Read sample Input/Output to know what to report in case of exceptions.

Sample Input 0:

```
10
3
```

Sample Output 0:

```
3
```

Sample Input 1:

```
10
Hello
```

Sample Output 1:

```
java.util.InputMismatchException
```

Sample Input 2:

```
10
0
```

Sample Output 2:

```
java.lang.ArithmeticException: / by zero
```

Sample Input 3:

```
23.323
0
```

Sample Output 3:

```
java.util.InputMismatchException
```

For example:

Input	Result
10 3	3
10 Hello	java.util.InputMismatchException
10 0	java.lang.ArithmeticException: / by zero
23.323 0	java.util.InputMismatchException

Answer: (penalty regime: 0 %)

```
1  import java.util.*;
2  public class Exception1
3  {
4      public static void main(String[] args)
5      {
6          Scanner sc = new Scanner(System.in);
7          try
8          {
```

```
 9              int x = sc.nextInt();
10              int y = sc.nextInt();
11              System.out.println(x/y);
12          }
13          catch(Exception e)
14          {
15              System.out.println(e);
16          }
17      }
18  }
```

	Input	Expected	Got	
✔	10 3	3	3	✔
✔	10 Hello	java.util.InputMismatchException	java.util.InputMismatchException	✔
✔	10 0	java.lang.ArithmeticException: / by zero	java.lang.ArithmeticException: / by zero	✔
✔	23.323 0	java.util.InputMismatchException	java.util.InputMismatchException	✔

Passed all tests! ✔