DO SELECT CODING QUESTION

QUESTION

In **Java**, we can use more than one catch block with the try block. Generally, multiple catch block is used to handle different types of exceptions, which means each catch block is used to handle different types of exceptions.

If you use multiple catch blocks for the same type of exception, then it will give you a compile-time error because **Java does not allow you to use multiple catch block for the same type of exception**. A catch block is always preceded by the try block.

Write a program to demonstrate Multiple Exceptions.

**Input Format**

* The Main() method has already been implemented, which will pass values for the variables: **String1**, **String2** and **Operator**.

**IMPORTANT:**

* If you want to test your program you can implement a **Main()** function given in the stub and you can use **RUN CODE** to test your Main() provided you have made valid function calls with valid data required.

**Tasks:**

In the function **handleException**(Activity a):

* Using try-catch block check that the value of either **string1**or **string2**variable is **null**, then throw appropriate exception for **NullPointerException** and return "**Null values found**".
* Using try-catch block check if the value of **operator**variable is not equal to "+" or "-" then throw and return the default exception(**Exception**) with the operator as parameter the return operator. Example if operator = "/" throw the **Exception(a.operator)**and return the operator: "/"**.**
* If no exception is found return "**No Exception Found**".

In the function **operate**(Activity a):

* perform the string operations, using switch statement and return the correct value.

**Specifications:**

**class** **Activity**:

    data fields:

      String string1

      String string2

      String **operator**

    Implement a parameterized constructor to initialize all the instance variables

**class** **Source**:

method definitions:

handleException(Activity a): implement **try**-**catch** blocks **and** **throw** different exceptions as described in above.

**return** type: String

visibility: **public**

operate(Activity a): implement **switch** statement to calculate Result based on value of Operator

**return** type: String

visibility: **public**

You have to implement the following methods under Source class:

* **handleException (Activity a)** - In this function you have to check for exceptions.
* **operate (Activity a)** - this function should implement the string operation between **String1** and **String2** for the operator **Operator**.
* If **operator = +**, concat the strings **String1** and **String2**.
* **e.g.** for **String1 = "hello"**and **String2 = "world"**, then **Result** = **"helloworld"**
* If **operator = -**, replace the contents of **String2** in **String1** with empty string.
* **e.g.** If **String1 = "helloworld"** and **String2 = "world"**, then **Result = "hello"**

IMPORTANT:

* If you want to test your program, you can implement a **main()** function given in the stub and you can use **RUN CODE** to test your main() provided you have made valid function calls with valid data required.

**ALLOWED TECHNOLOGIES**

* Java 8

**SOLUTION:**

**package**com.training.assignment;

**class**Activity{

//Implement Activity class here..

String string1;

String string2;

String operator;

Activity (String string1, String string2, String operator){

**this**.string1 = string1;

**this**.string2 = string2;

**this**.operator = operator;

}

**publicclass** Source {

//implement the two function given in description in here...

**public** String handleException (Activity a){

**try** {

**if**(a.string1.equals(**null**) || a.string2.equals(**null**)) {

**thrownew**NullPointerException();

}

**if**(!a.operator.equals("+") && !a.operator.equals("-")) {

**thrownew** Exception(a.operator);

}

} **catch** (NullPointerExceptionne) {

**return**"Null values found";

} **catch**(Exception e) {

**return**e.getMessage();

}

**return**"No Exception Found";

}

**public** String doOperation (Activity a){

String string1=a.string1.trim();

String string2= a.string2.trim();

String operator= a.operator.trim();

String result="";

**switch**(operator){

**case**"+":

// result=string1.concat(string2);

result=string1 + string2;

**break**;

**case**"-":

result=string1.replaceAll(string2,"");

**break**;

**default**:

**break**;

}**return**result;

}

**publicstaticvoid**main(String args[]) **throws** Exception {

//Write your own main to check the program....

}

}}