

Java Programming 2-4: Exceptions and Assertions Practice Activities

Vocabulary Definitions:

1. 1. A certain kind of try statement that contains resources.
2. Try-with-resources
3. An invariant used to evaluate the assumptions of the class instances.
4. Assertion
5. Certain types of boolean statements that allow you to test specific aspects of your code.
6. Assertions
7. Key statement for handling exceptions in Java.
8. Try
9. An invariant that handles boolean statements to test internal values.
10. Assertion
11. An invariant that handles conditions in control flow statements.
12. Assertion
13. A statement that allows you to handle multiple exceptions.
14. Multi-catch
15. An optional addition to a try-catch statement that will always be executed.
16. Finally
17. Run-time errors that can be handled inside the program.
18. Exceptions
19. Try It/Solve It Tasks
19. 1. Exception Handling for the "Make Transaction" Button
20. Here's how you might handle exceptions for the "make transaction" button in a Java Swing application:

PROGRAM:

1. Assuming a button named "makeTransactionButton" and a method:

```
named makeTransaction() makeTransactionButton.addActionListener(e -> {  
    try {  
        makeTransaction();  
    } catch (Exception ex) {  
        JOptionPane.showMessageDialog(null, "An error occurred during transaction: "  
+ ex.getMessage());  
    }  
});
```

2. Create a Custom Exception Class Code:

```

public class MyException extends Exception {
    public MyException(String message) {
        super(message);
    }
}

```

3. Update Catch Statements to Use Custom Exception

Update JavaBank.java to use MyException instead of Exception. Here's how you might do it:

```

try {
    // code that might throw an exception
} catch (Exception e) {
    MyException newExc = new MyException("An unhandled error occurred!!");
    JOptionPane.showMessageDialog(null, newExc.getMessage());
}

```

4. Surround Method Calls with Try-Catch Statements Code:

```

try {
    createAccount(); // Example method
} catch (MyException ex) {
    JOptionPane.showMessageDialog(null, "Create Account Error: " +
        ex.getMessage());
}
try {
    makeTransaction(); // Example method
} catch (MyException ex) {
    JOptionPane.showMessageDialog(null, "Transaction Error: " + ex.getMessage());
}

```

5. Test the Custom Exception Comment out all other catch statements:

```

try {
    createAccount(); // Example method
} catch (Exception e) {
    MyException newExc = new MyException("An unhandled error occurred!!");
    JOptionPane.showMessageDialog(null, newExc.getMessage());
}
// Similarly for transaction method
try {
    makeTransaction(); // Example method
}

```

```
} catch (Exception e) {  
    MyException newExc = new MyException("An unhandled error occurred!!");  
    JOptionPane.showMessageDialog(null, newExc.getMessage());  
}
```

Result:

Enter incorrect data for both methods and verify that the custom exception message appears. Uncomment other catch statements when tests are complete.