IntegerDivisionByZeroException

Workshop #01

Flutter Developer Bootcamp

# **Purpose**

This workshop demonstrates how to handle exceptions in Dart. It shows how to use a try block to wrap code that might throw an exception and a catch block to handle that exception gracefully.

In this specific case, if you run this code, it will print the exception message "IntegerDivisionByZeroException", indicating that an attempt was made to divide by zero, which is not allowed.

# **Problem**

In this Dart workshop, there will be an integer division operation (x ~/ y), but if y is set to 0, it will trigger an IntegerDivisionByZeroException. This exception is caught in a catch block, where it's printed out. This demonstrates how to handle errors gracefully during program execution. If there are no values assigned to x, give it any value, then run the code to see the error handling message.

# **How to Solve**

1. Checkout the workshop from Git Repo:

git clone -b <user-branch> <repo-URL>

2. Open the root folder inside VS Code

3. Open the root folder in terminal

4. Run the command dart run filename.dart

5 set any value to x

# **You Will Achieve**

When you complete this workshop you will learn the following:

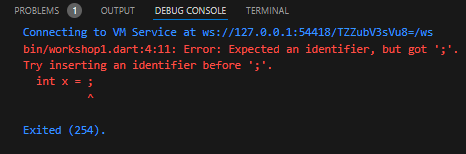
void main(List<String> args): This is the entry point of the Dart program. It defines a function named main that takes a list of strings (args) as input.

* int x = 5;: Declares an integer variable x and assigns it the value 5.
* int y = 0;: Declares an integer variable y and assigns it the value 0.
* try { ... }: Starts a try block, indicating that the code inside it might throw an exception.
* int xy = x ~/ y;: Performs integer division between x and y. The ~/ operator is used for integer division. Since y is 0, this line would normally throw an IntegerDivisionByZeroException.
* print(xy);: Prints the result of the integer division. However, this line will not be executed if an exception occurs in the previous line.
* } on IntegerDivisionByZeroException catch (e) { ... }: Specifies that if an IntegerDivisionByZeroException occurs within the try block, it should be caught, and the code inside this catch block should be executed. In this case, it simply prints the exception e.

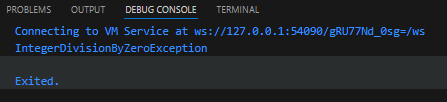
So, when you run this code, instead of crashing with a division by zero error, it will catch that error and print it out. This allows the program to gracefully handle the error without crashing.

# **Screenshots**

## **Before implementation (without value)**



## **After implementation (With value)**



# **How to submit your workshop**

Push your project back to the same git branch using command:

<command name>

# **Happy Coding!**