```
using System;

namespace backendDay7
{

    3 references
    public class CustomDivideByZeroException : Exception
    {
        1 reference
        public CustomDivideByZeroException(string message) : base(message) { }
    }

    0 references
    internal class Program
    {
        2 references
        public class InputHelper
        {
            2 references
            public static int CheckInput(string prompt)
            {
                int value;
                bool valid;
                do
                {
                    Console.Write(prompt);
                    valid = int.TryParse(Console.ReadLine(), out value);
                    if (!valid)
                    {
                        Console.WriteLine("Please input a valid integer.");
                    }
                } while (!valid);
                return value;
            }
        }
```

```
1 reference
public class ProcessNumbers
{
    1 reference
    public static void DivideNumbers(int numerator, int divisor)
    {
        if (divisor == 0)
        {
            throw new CustomDivideByZeroException("Error: Division by zero is not allowed.");
        }

        int quotient = numerator / divisor;
        int remainder = numerator % divisor;

        Console.WriteLine($"\nResult:");
        Console.WriteLine($"Quotient: {quotient}");
        if (remainder != 0)
        {
            Console.WriteLine($"Remainder: {remainder}");
        }
    }
}
```

```
0 references
static void Main(string[] args)
{
    int successCount = 0;
    int failCount = 0;
    string useAgain;

    do
    {
        Console.WriteLine("\n=======\tDIVISION PROGRAM\t=======");
        try
        {
            int numerator = InputHelper.CheckInput("Enter numerator (integer): ");
            int divisor = InputHelper.CheckInput("Enter divisor (integer): ");
            ProcessNumbers.DivideNumbers(numerator, divisor);
            successCount++;
        }
        catch (CustomDivideByZeroException ex)
        {
            Console.WriteLine(ex.Message);
            failCount++;
        }
        catch (Exception ex)
        {
            Console.WriteLine("An unexpected error occurred: " + ex.Message);
            failCount++;
        }

        Console.Write("\nDo you want to divide another set of numbers? (yes/no): ");
        useAgain = Console.ReadLine()?.Trim().ToLower();

        while (useAgain != "yes" && useAgain != "no")
        {
            Console.Write("Invalid input. Please type 'yes' or 'no': ");
            useAgain = Console.ReadLine()?.Trim().ToLower();
        }

    } while (useAgain == "yes");

    Console.WriteLine($"\nSummary:");
    Console.WriteLine($"Successful attempts: {successCount}");
    Console.WriteLine($"Failed attempts: {failCount}");
    Console.WriteLine("Thank you for using the division program!");
}
}
}
```

OUTPUT

```
======= DIVISION PROGRAM        =======
Enter numerator (integer): 54
Enter divisor (integer): 2

Result:
Quotient: 27

Do you want to divide another set of numbers? (yes/no): yes

======= DIVISION PROGRAM        =======
Enter numerator (integer): 32
Enter divisor (integer): 0
Error: Division by zero is not allowed.

Do you want to divide another set of numbers? (yes/no): yes

======= DIVISION PROGRAM        =======
Enter numerator (integer): 32
Enter divisor (integer): 23

Result:
Quotient: 1
Remainder: 9

Do you want to divide another set of numbers? (yes/no): no

Summary:
Successful attempts: 2
Failed attempts: 1
Thank you for using the division program!

D:\NICO\WPH\BACKEND\backendDay7\bin\Debug\net8.0\backendDay7.exe (process 15280) exited with code 0 (0x0).
Press any key to close this window . . .
```

ERRO HANDLING

```
======= DIVISION PROGRAM        =======
Enter numerator (integer):
Please input a valid integer.
Enter numerator (integer): s
Please input a valid integer.
Enter numerator (integer): bi
Please input a valid integer.
Enter numerator (integer): sd
Please input a valid integer.
Enter numerator (integer): 2
Enter divisor (integer): 3

Result:
Quotient: 0
Remainder: 2

Do you want to divide another set of numbers? (yes/no): bs
Invalid input. Please type 'yes' or 'no': asd
Invalid input. Please type 'yes' or 'no': ,
Invalid input. Please type 'yes' or 'no': 1
Invalid input. Please type 'yes' or 'no': no

Summary:
Successful attempts: 1
Failed attempts: 0
Thank you for using the division program!
```

REFLECTION

In my program, I separated the logic for division by putting it in its own method called DivideNumbers inside the ProcessNumbers class. I did this to keep my code more organized and avoid mixing everything in the Main() method. At first, I actually forgot to add the try-catch block, so when I tried dividing by zero, the program crashed. After realizing that, I added proper exception handling in the Main method to catch errors and display a clear message when something goes wrong, like dividing by zero.

I also created a custom exception called CustomDivideByZeroException, which is thrown when the divisor is zero. This made the error more meaningful and specific to the situation, instead of using the default system exception. For input validation, I used a method that checks if the user's input is a valid integer using int.TryParse. If it isn’t, the program tells the user and asks again until a valid number is entered, which helps avoid random errors from invalid input.

To let the user try again, I used a do-while loop, and I added another loop to make sure they type only "yes" or "no" when asked if they want to try again. This makes the program feel more controlled and user-friendly. I also displayed messages that clearly show the result, like the quotient and remainder, or explain if there was an error. One challenge I faced was learning how to make a custom exception and place the error handling in the right spot, but once I figured it out, the program worked much better. Overall, I learned a lot about structuring code properly, validating input, and handling exceptions to make the program more reliable.