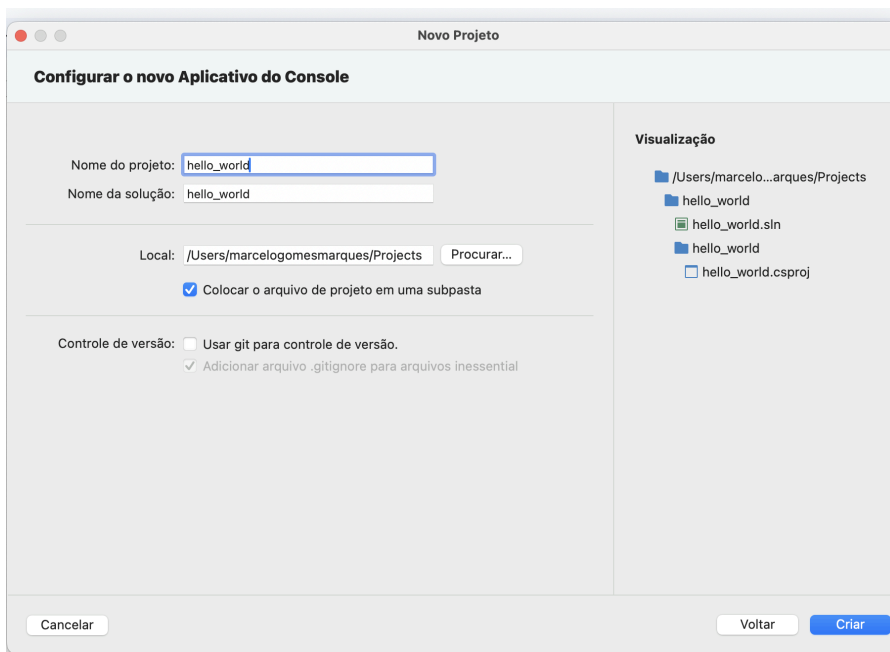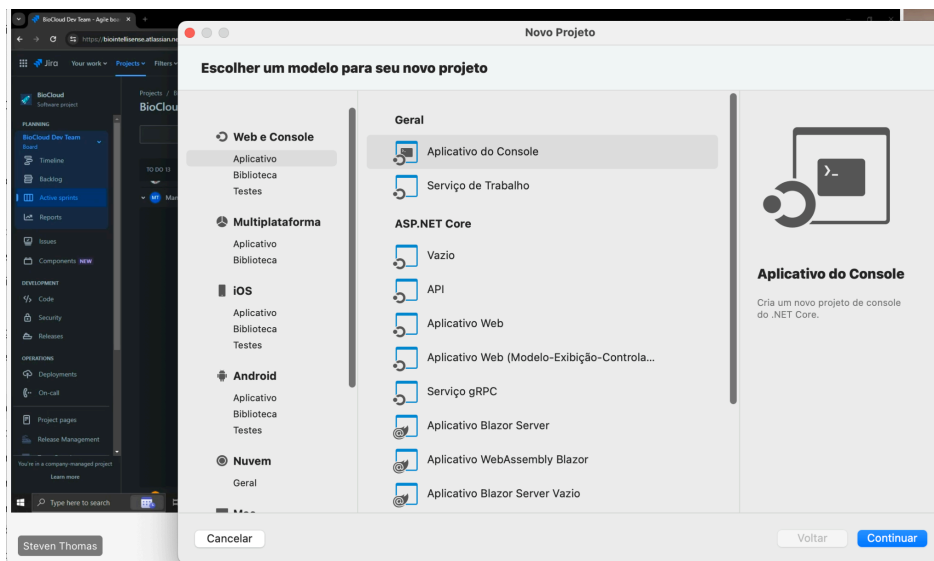# Exercise 1: Hello World Application

Statement:

Create a console application in C# that outputs "Hello World" to the console.

Topics to Practice:

Console applications, output to console

```
using System.Diagnostics;

[DebuggerDisplay($"{{{nameof(GetDebuggerDisplay)}(),nq}}")]
internal class Program
{
    private static void Main(string[] args)
    {
        Console.WriteLine("Hello, World!");
    }

    private string GetDebuggerDisplay()
    {
        return ToString();
    }
}
```

Terminal – hello_world

Hello, World!
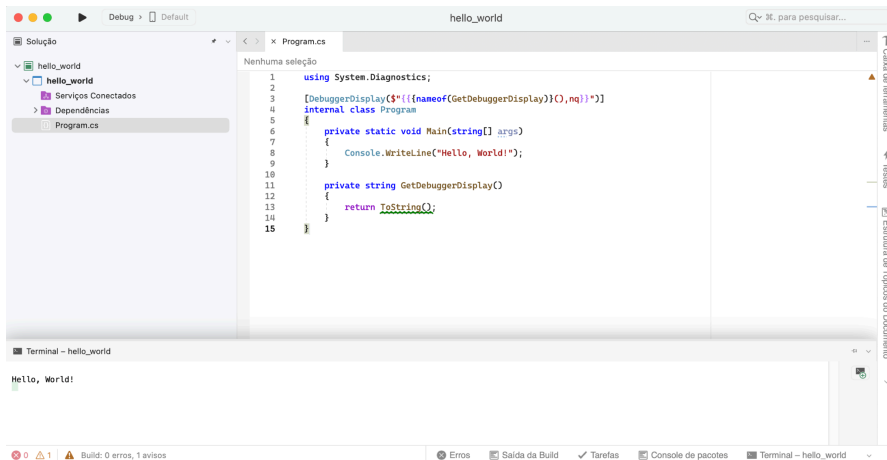
Exercise 2: Calculator for Basic Arithmetic Operations

Statement:

Develop a Calculator class in C# that supports basic arithmetic operations: addition, subtraction, multiplication, and division. This class should have a method for each operation, each taking two double parameters and returning the result as a double. Create a console application that uses this Calculator class, allowing the user to input two numbers and select an operation to perform with these numbers.

Topics to Practice:
Class, method, function, conditional statements, console input/output, exception handling

Unit Testing Task:
For each arithmetic operation in the Calculator class, write unit tests that cover:
Normal operations (e.g., 2 + 2, 4 - 2)
Edge cases (e.g., division by zero)

Error scenarios (if applicable)

CalculatorProgram > Debug > Default

Adoraríamos ouvir sobre sua experiência com o .NET Core no Visual Studio

Solução
simple_calculator (main)
CalculatorLibrary

Terminal – CalculatorProgram

CalculatorProgram.cs
CalculatorProgram

```
Console Calculator in C#
------------------------

Please, type the firt number, and then press Enter: a
This is not valid input. Please enter a valid number: 1.2
Please, type the second number, and then press Enter: 3.3
Choose an operator from the following list:
        a – Add
        s – Subtract
        m – Multiply
        d – Divide
Your option? a
Your result: 4.5

------------------------

Press 'n' to close the app, or press any other key to continue: 3

Please, type the firt number, and then press Enter: .2
Please, type the second number, and then press Enter: 7.1
Choose an operator from the following list:
        a – Add
        s – Subtract
        m – Multiply
        d – Divide
Your option? m
Your result: 1.42

------------------------

Press 'n' to close the app, or press any other key to continue: .

Please, type the firt number, and then press Enter: 3.4
Please, type the second number, and then press Enter: 2.1
Choose an operator from the following list:
        a – Add
        s – Subtract
        m – Multiply
        d – Divide
Your option? s
Your result: 1.3

------------------------

Press 'n' to close the app, or press any other key to continue:
```

CalculatorProgram > Debug > Default

Adoraríamos ouvir sobre sua experiência com o .NET Core no Visual Studio

Solução
simple_calculator (main)
CalculatorLibrary

Terminal – CalculatorProgram

CalculatorProgram.cs
CalculatorProgram

```
        s – Subtract
        m – Multiply
        d – Divide
Your option? m
Your result: 1.42

------------------------

Press 'n' to close the app, or press any other key to continue: .

Please, type the firt number, and then press Enter: 3.4
Please, type the second number, and then press Enter: 2.1
Choose an operator from the following list:
        a – Add
        s – Subtract
        m – Multiply
        d – Divide
Your option? s
Your result: 1.3

------------------------

Press 'n' to close the app, or press any other key to continue:

Please, type the firt number, and then press Enter: 1
Please, type the second number, and then press Enter: 0.0
Choose an operator from the following list:
        a – Add
        s – Subtract
        m – Multiply
        d – Divide
Your option? d
Error: Division by zero is not allowed.
This operation will result in a mathematical error.

------------------------

Press 'n' to close the app, or press any other key to continue:

Please, type the firt number, and then press Enter: 3.1
Please, type the second number, and then press Enter: 0.25
Choose an operator from the following list:
        a – Add
        s – Subtract
        m – Multiply
        d – Divide
Your option? d
Your result: 12.4

------------------------
```

WriterTest > PrintMessage()

```csharp
using Xunit.Abstractions;

namespace Calculator.UnitTests.Services
{
    public class WriterTest
    {
        private readonly TextWriter outputWriter;
        private readonly string message;

        public WriterTest(TextWriter outputWriter, string message)
        {
            this.outputWriter = outputWriter;
            this.message = message;
        }

        public void PrintMessage()
        {
            outputWriter.WriteLine(message);
        }
    }

    public class CalculatorTest
    {
        private readonly CalculatorLibrary.Calculator _calculator;
        private readonly ITestOutputHelper output;

        public CalculatorTest(ITestOutputHelper output)
        {
            _calculator = new CalculatorLibrary.Calculator();
            this.output = output;
        }

        [Theory]
        [InlineData(2.0, 2.0, "a", 4.0, "Add")]
        [InlineData(1.0, 1.0, "s", 0.0, "Subtract")]
        [InlineData(3.0, 2.0, "m", 6.0, "Multiply")]
        [InlineData(3.0, 2.0, "d", 1.5, "Divide")]
        [InlineData(3.0, 0.0, "d", double.NaN, "Error: Division by zero is not allowed.")]
        public void CalculatorDo(double num1, double num2, string op,
            double expected, string message)
        {
            var result = _calculator.DoOperation(num1, num2, op);

            Assert.Equal(expected, result);
            output.WriteLine(message);

            var content = new System.Text.StringBuilder();
            var writer = new StringWriter(content);
            var sut = new WriterTest(writer, message);

            sut.PrintMessage();

            var actualOutput = content.ToString();
            Assert.Equal(message + "\n", actualOutput);
        }
    }
}
```

CalculatorProgram > Debug > Default

simple_calculator

We'd love to hear about your .NET Core experience in Visual Studio for Mac.   Don't show again   Take a survey

Solution
simple_calculator (master)
Calculator.Tests
Connected Services
Dependencies – 4 updates
CalculatorTest.cs
Usings.cs
CalculatorLibrary
Connected Services
Dependencies
Calculator.Library.cs

CalculatorLibrary.cs CalculatorTest.cs CalculatorProgram.cs

```csharp
using Newtonsoft.Json;

namespace CalculatorLibrary
{
    public class Calculator
    {
        JsonWriter writer;

        public void Finish()
        {
            writer.WriteEndArray();
            writer.WriteEndObject();
        }
    }
}
```

Test Results
Successful Tests   Failed Tests   Inconclusive Tests   Tests not run   Output

[xUnit.net 00:00:00.00] xUnit.net VSTest Adapter v2.4.5+1cae2f33e (64-bit .NET 8.0.2)
[xUnit.net 00:00:00.19] Discovering: Calculator.Tests
[xUnit.net 00:00:00.21] Discovered: Calculator.Tests
[xUnit.net 00:00:00.21] Starting: Calculator.Tests
[xUnit.net 00:00:00.24] Finished: Calculator.Tests
 ⊘ Success 'simple_calculator.Calculator.Tests.Calculator.UnitTests.Services.CalculatorTest.CalculatorDo(num1: 3, num2: 0, op: "d", expected: NaN, message: "Error:
Division by zero is not allowed.")'
Console output:
Error: Division by zero is not allowed.

 ⊘ Success 'simple_calculator.Calculator.Tests.Calculator.UnitTests.Services.CalculatorTest.CalculatorDo(num1: 3, num2: 2, op: "d", expected: 1.5, message: "Divide")'
Console output:
Divide

 ⊘ Success 'simple_calculator.Calculator.Tests.Calculator.UnitTests.Services.CalculatorTest.CalculatorDo(num1: 2, num2: 2, op: "a", expected: 4, message: "Add")'
Console output:
Add

 ⊘ Success 'simple_calculator.Calculator.Tests.Calculator.UnitTests.Services.CalculatorTest.CalculatorDo(num1: 3, num2: 2, op: "m", expected: 6, message: "Multiply")'
Console output:
Multiply

 ⊘ Success 'simple_calculator.Calculator.Tests.Calculator.UnitTests.Services.CalculatorTest.CalculatorDo(num1: 1, num2: 1, op: "s", expected: 0, message: "Subtract")'
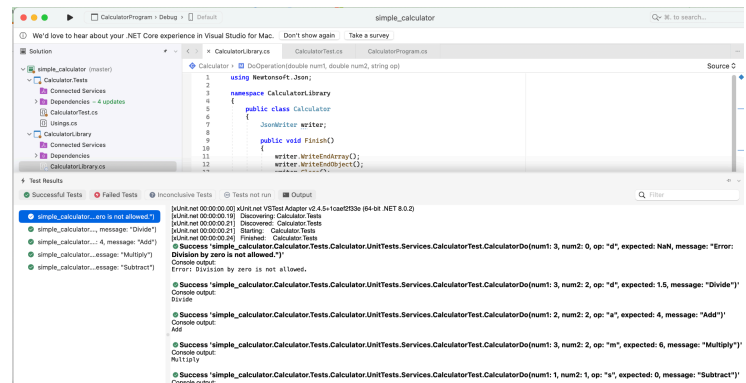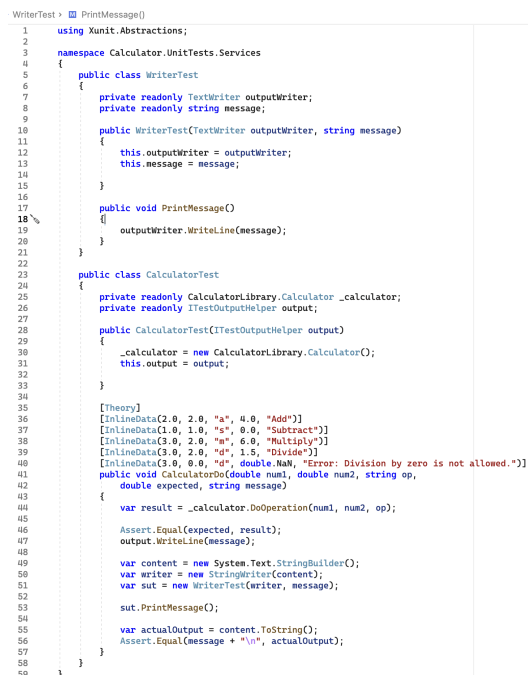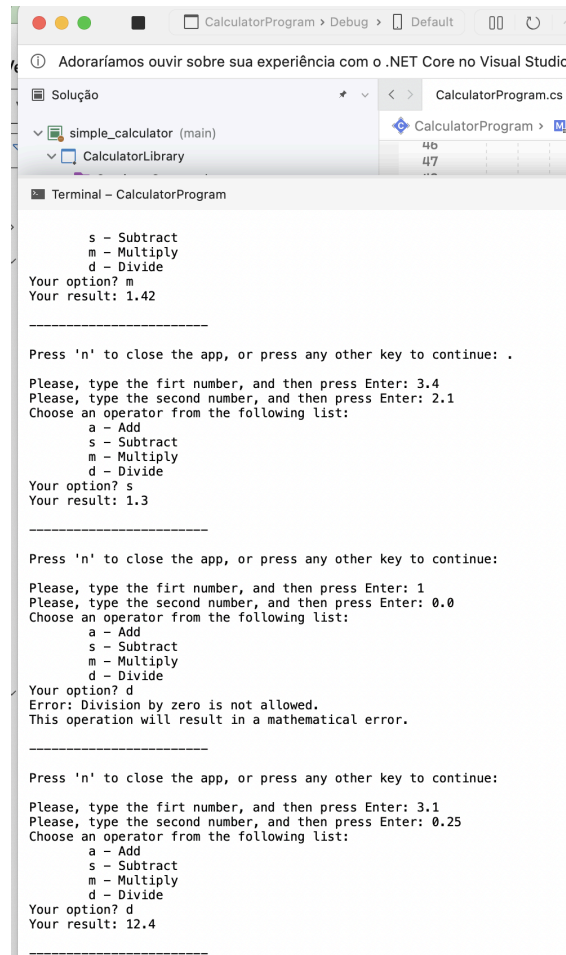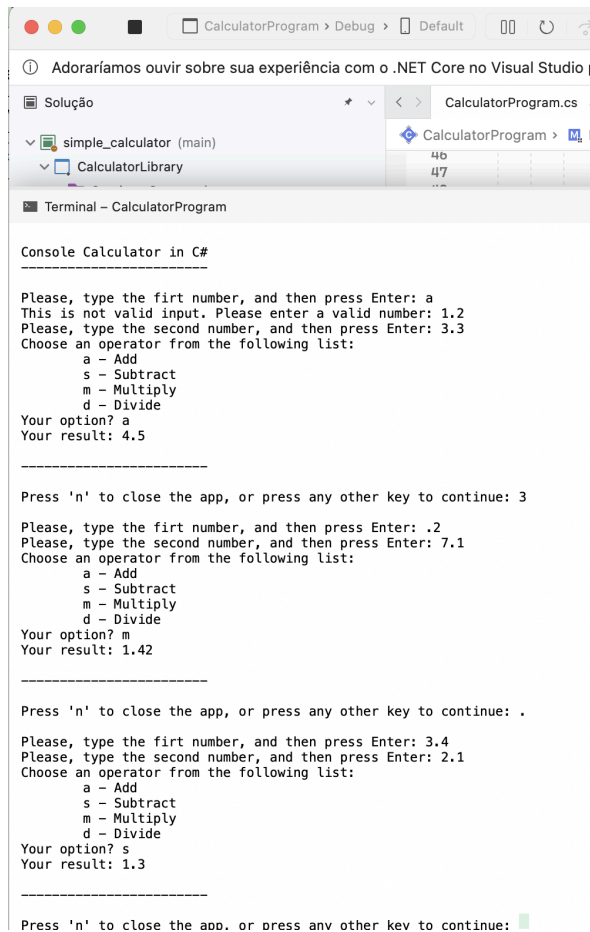Console output:

Exercise 3: A Console Application That Sorts a List of Numbers

Statement:

Implement a console application in C# that prompts the user to enter a series of numbers separated by commas. The application should then convert the input into a list of integers, sort them in ascending order, and print the sorted list to the console. Ensure the application gracefully handles any input errors, such as invalid numbers.

Topics to Practice:

Console input/output, string manipulation, arrays or lists, sorting algorithms or library sorting functions, error handling

Solution

⬛ SortsListNumbers (master)
  ⬛ SortsListNumbers
    📁 Connected Services

Program.cs   App.cs ×   UnitTestSortsListNumbers.cs

◆ App › Ⓜ SortList(string[] line)

1   namespace SortsListNumbersApp
2   {

Terminal — SortsListNumbers

Console Sort a List of Numbers C#

0, A, 9 3, 7
Oh no! You entered an element that is not an integer! Please review and try again.
  - Details: The input string 'A' was not in a correct format.

Press 'n' to close the app, or press any other key to continue:

0, 9, 3 5, 7
Oh no! You entered an element that is not an integer! Please review and try again.
  - Details: The input string '3 5' was not in a correct format.

Press 'n' to close the app, or press any other key to continue:

Oh no! You entered an element that is not an integer! Please review and try again.
  - Details: The input string '' was not in a correct format.

Press 'n' to close the app, or press any other key to continue:

Oh no! You entered an element that is not an integer! Please review and try again.
  - Details: The input string '' was not in a correct format.

Press 'n' to close the app, or press any other key to continue:

9, 6, 0, 7, 3, -1, 12, 4
Here the sorted list of the numbers:
-1, 0, 3, 4, 6, 7, 9, 12
Press 'n' to close the app, or press any other key to continue: