# 目　录

## 一、实验内容

基于作业 1，完成如下任务

（1）在一元计算和二元计算接口中重载 calculate 方法，实现浮点型数据的加减乘除；

（2）对于整数除法和浮点型数的除法，考虑分母为零的情况

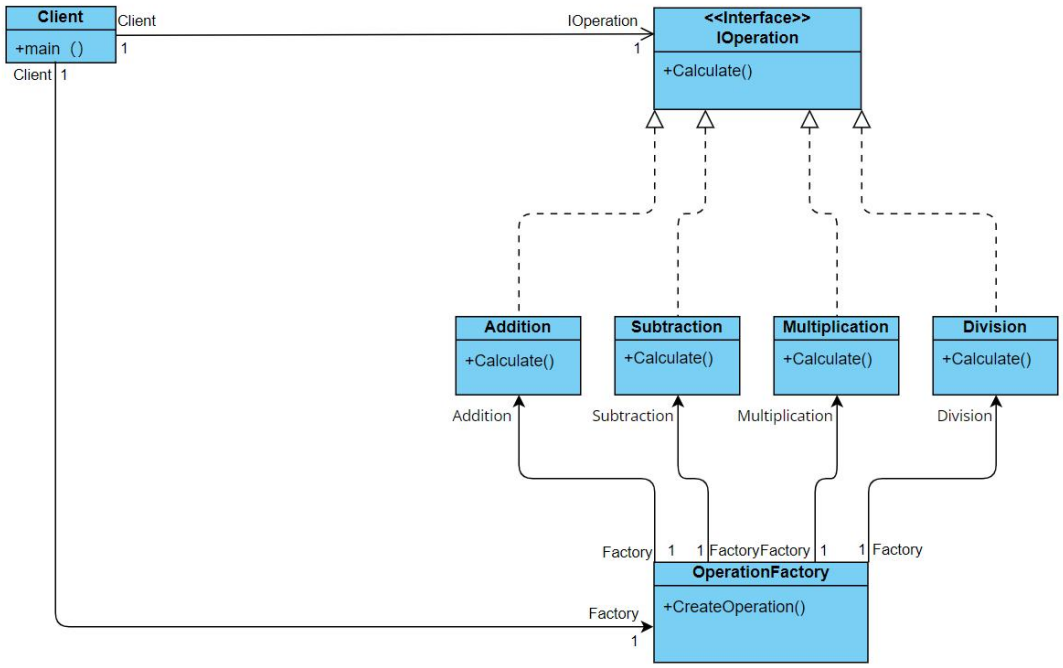
（3）基于 AAA 原则对于一元计算模块、二元计算模块和日志读写模块进行单元测试

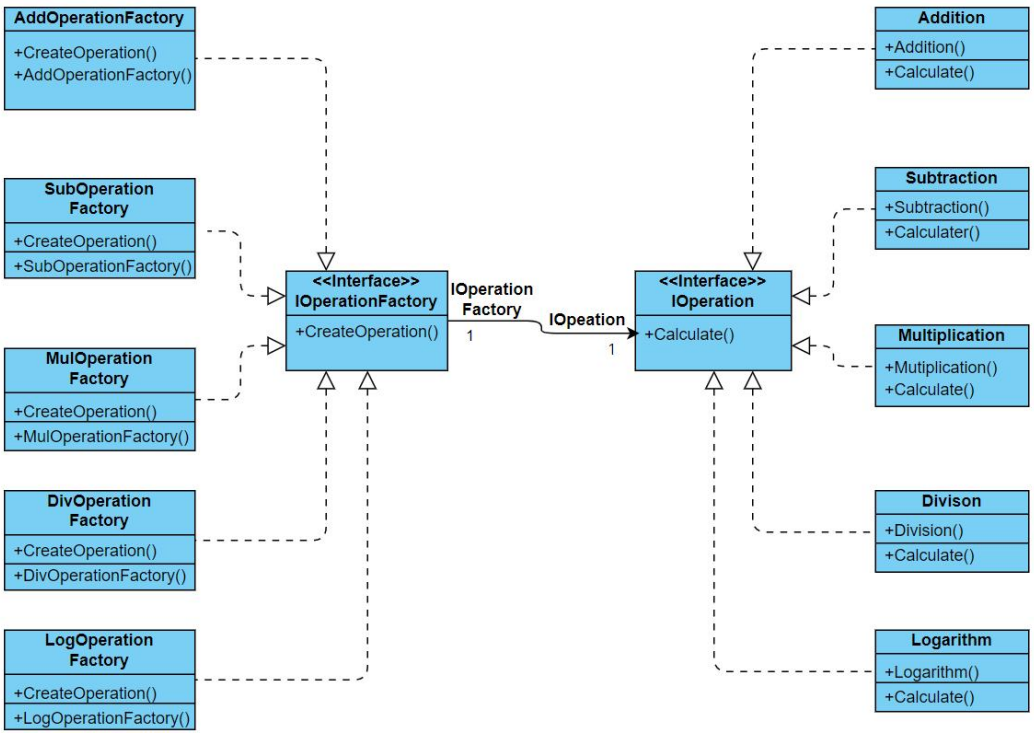
（4）参考 windows 自带计算器实现计算器的 UI 界面，在 UI 界面中使用之前实现的一元计算机和二元计算库

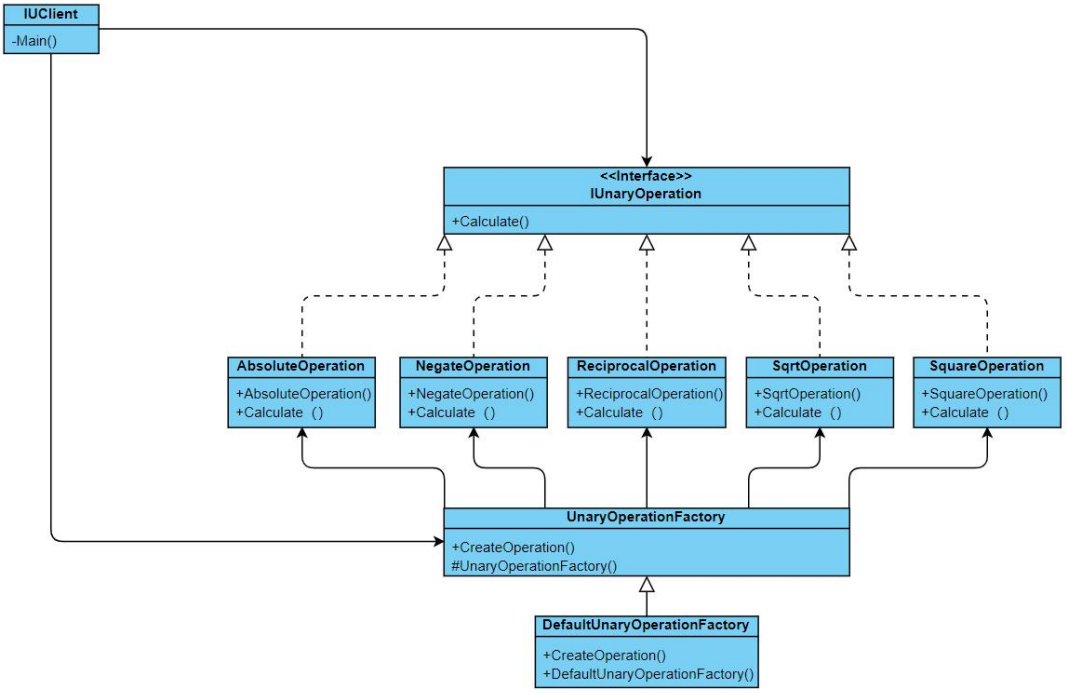
实现计算逻辑

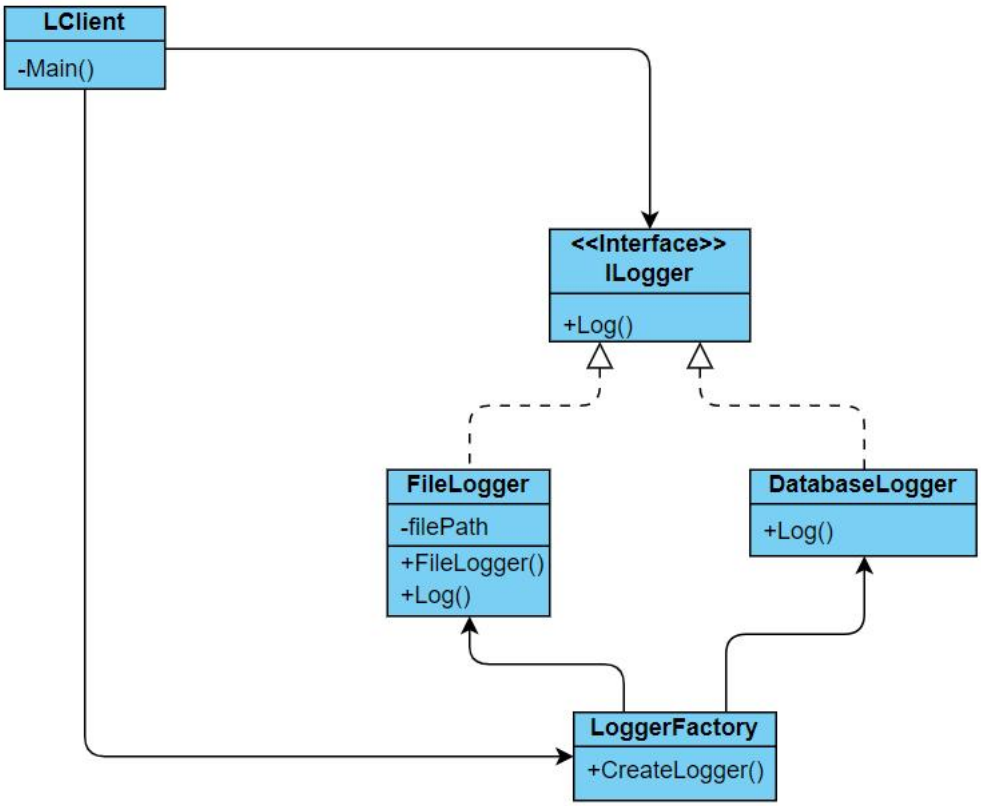

## 二、类图及代码结构

1. 类图

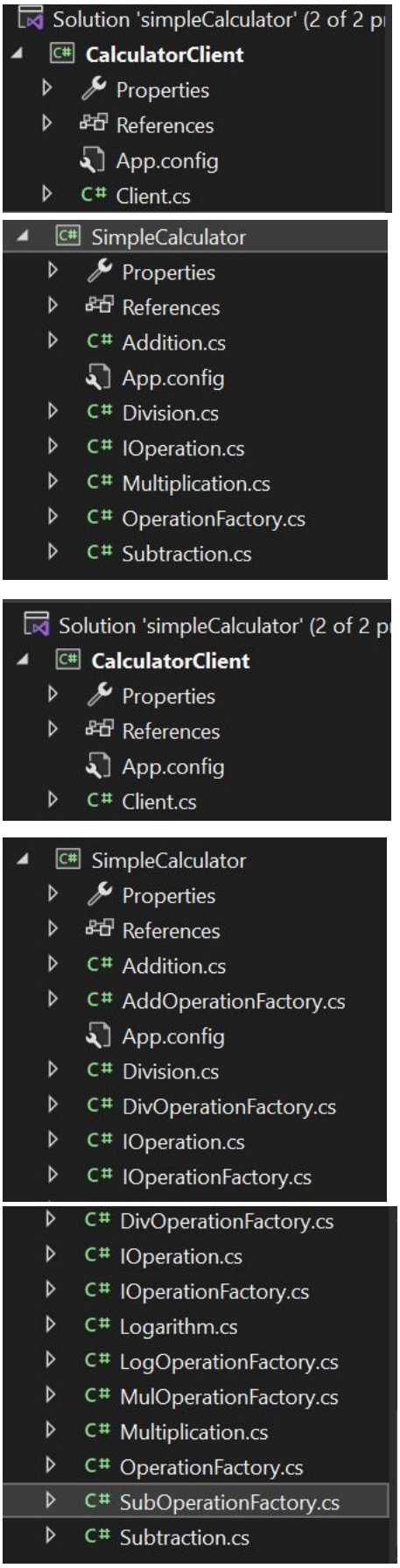

Client
+main ()
Client
1
Client
1
IOperation
1
<<Interface>>
IOperation
+Calculate()
Addition
+Calculate()
Subtraction
+Calculate()
Multiplication
+Calculate()
Division
+Calculate()
Addition
Subtraction
Multiplication
Division
Factory
1
1
FactoryFactory
1
1
Factory
OperationFactory
+CreateOperation()
Factory
1

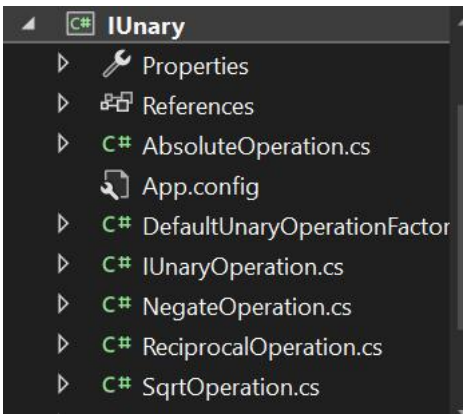

AddOperationFactory
+CreateOperation()
+AddOperationFactory()
SubOperation Factory
+CreateOperation()
+SubOperationFactory()
MulOperation Factory
+CreateOperation()
+MulOperationFactory()
DivOperation Factory
+CreateOperation()
+DivOperationFactory()
LogOperation Factory
+CreateOperation()
+LogOperationFactory()
<<Interface>>
IOperationFactory
+CreateOperation()
IOperation Factory
1
IOpeation
1
<<Interface>>
IOperation
+Calculate()
Addition
+Addition()
+Calculate()
Subtraction
+Subtraction()
+Calculater()
Multiplication
+Mutiplication()
+Calculate()
Divison
+Division()
+Calculate()
Logarithm
+Logarithm()
+Calculate()

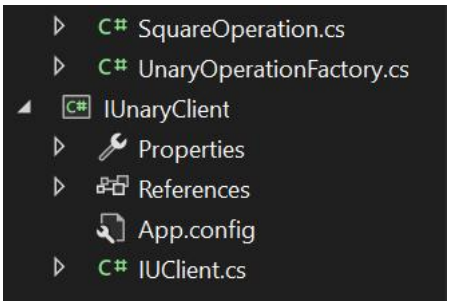

IUClient
-Main()
<<Interface>>
IUnaryOperation
+Calculate()
AbsoluteOperation
+AbsoluteOperation()
+Calculate ()
NegateOperation
+NegateOperation()
+Calculate ()
ReciprocalOperation
+ReciprocalOperation()
+Calculate ()
SqrtOperation
+SqrtOperation()
+Calculate ()
SquareOperation
+SquareOperation()
+Calculate ()
UnaryOperationFactory
+CreateOperation()
#UnaryOperationFactory()
DefaultUnaryOperationFactory
+CreateOperation()
+DefaultUnaryOperationFactory()

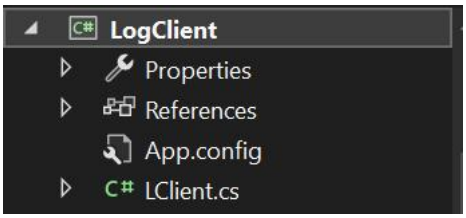

LClient
-Main()
<<Interface>>
ILogger
+Log()
FileLogger
-filePath
+FileLogger()
+Log()
DatabaseLogger
+Log()
LoggerFactory
+CreateLogger()

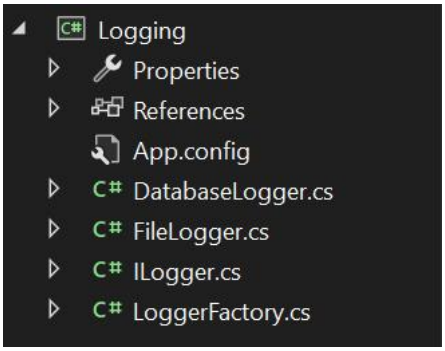
## 2. 代码结构


Solution 'simpleCalculator' (2 of 2 p
CalculatorClient
Properties
References
App.config
Client.cs
SimpleCalculator
Properties
References
Addition.cs
App.config
Division.cs
IOperation.cs
Multiplication.cs
OperationFactory.cs
Subtraction.cs
Solution 'simpleCalculator' (2 of 2 p
CalculatorClient
Properties
References
App.config
Client.cs
SimpleCalculator
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
Division.cs
DivOperationFactory.cs
IOperation.cs
IOperationFactory.cs
DivOperationFactory.cs
IOperation.cs
IOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
OperationFactory.cs
SubOperationFactory.cs
Subtraction.cs



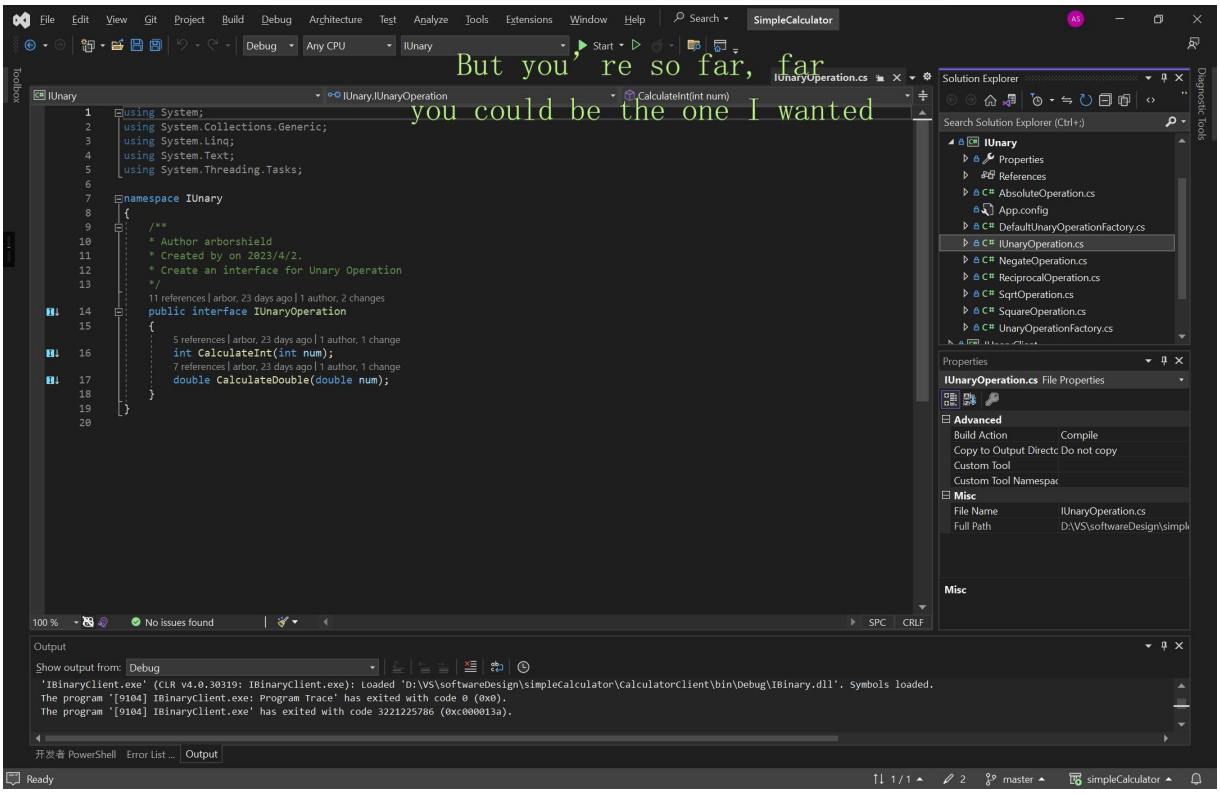



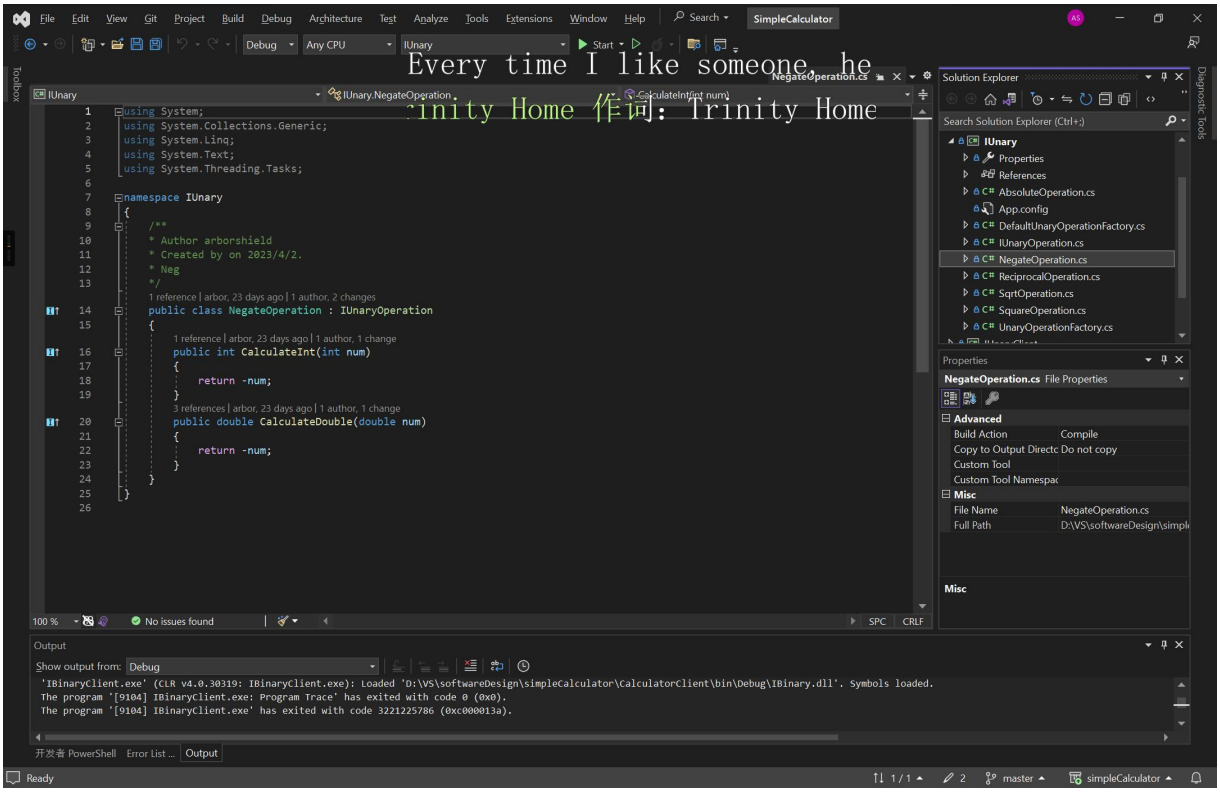



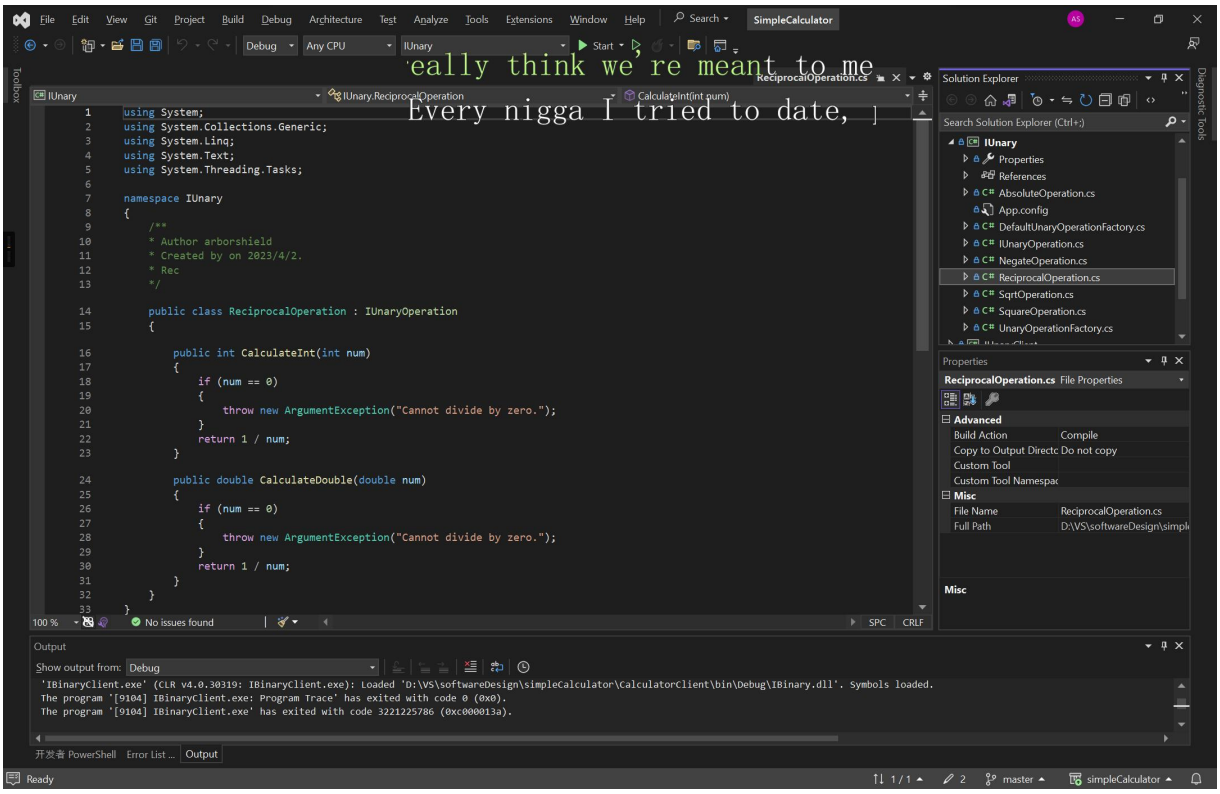

## 三、代码实现

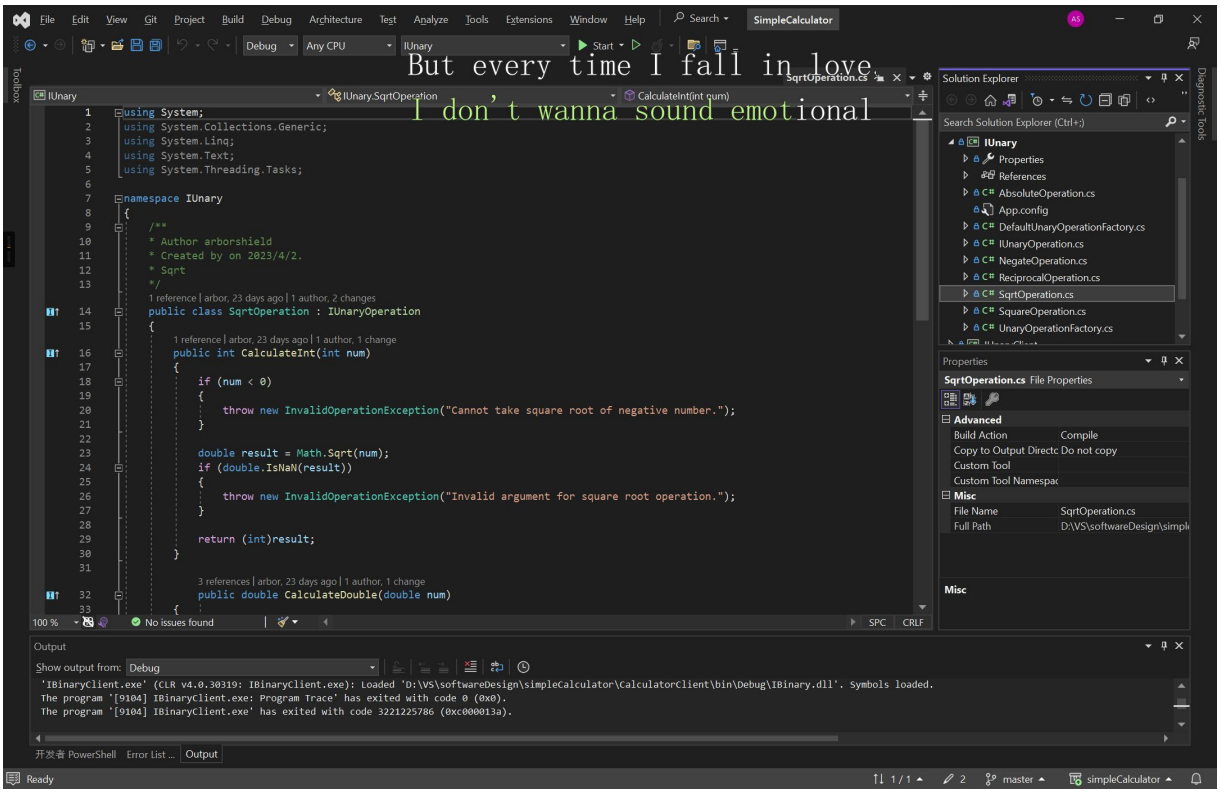
（1）在一元计算和二元计算接口中重载 calculate 方法，实现浮点型数据的加减乘除；

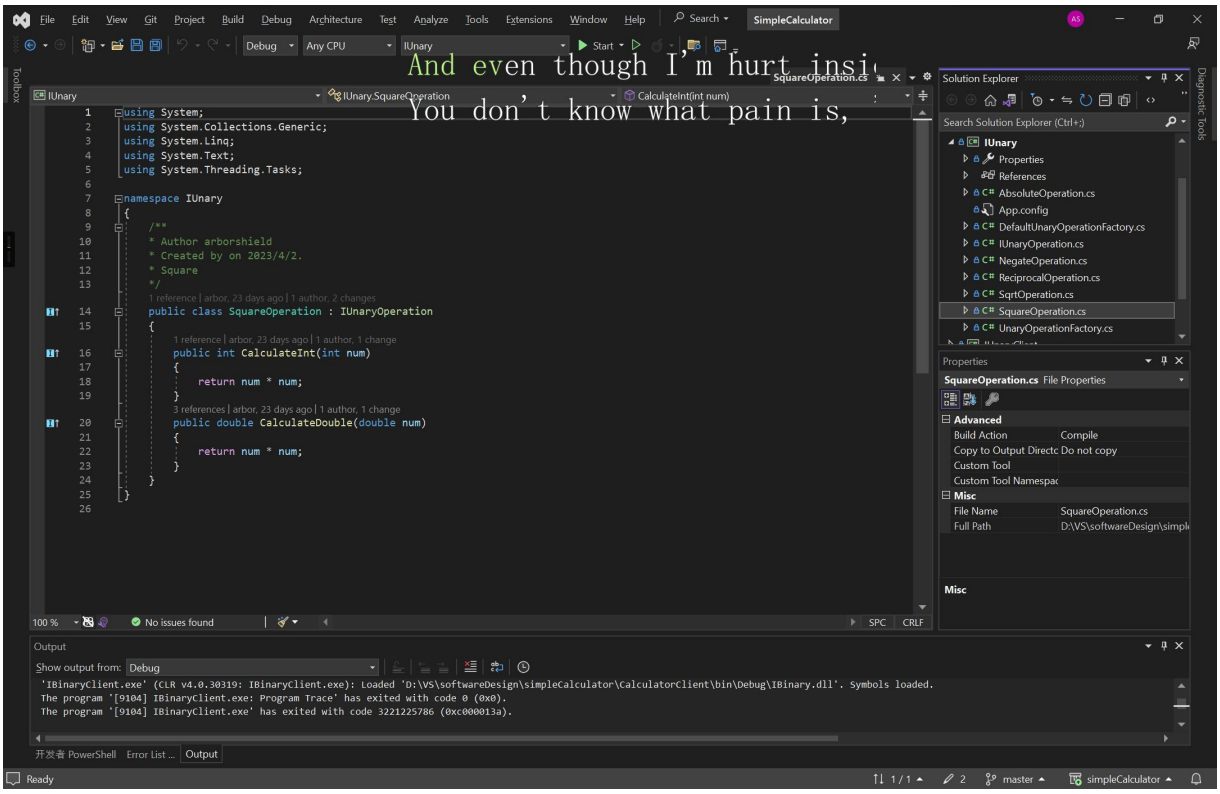
一元：


But you're so far, far
you could be the one I wanted

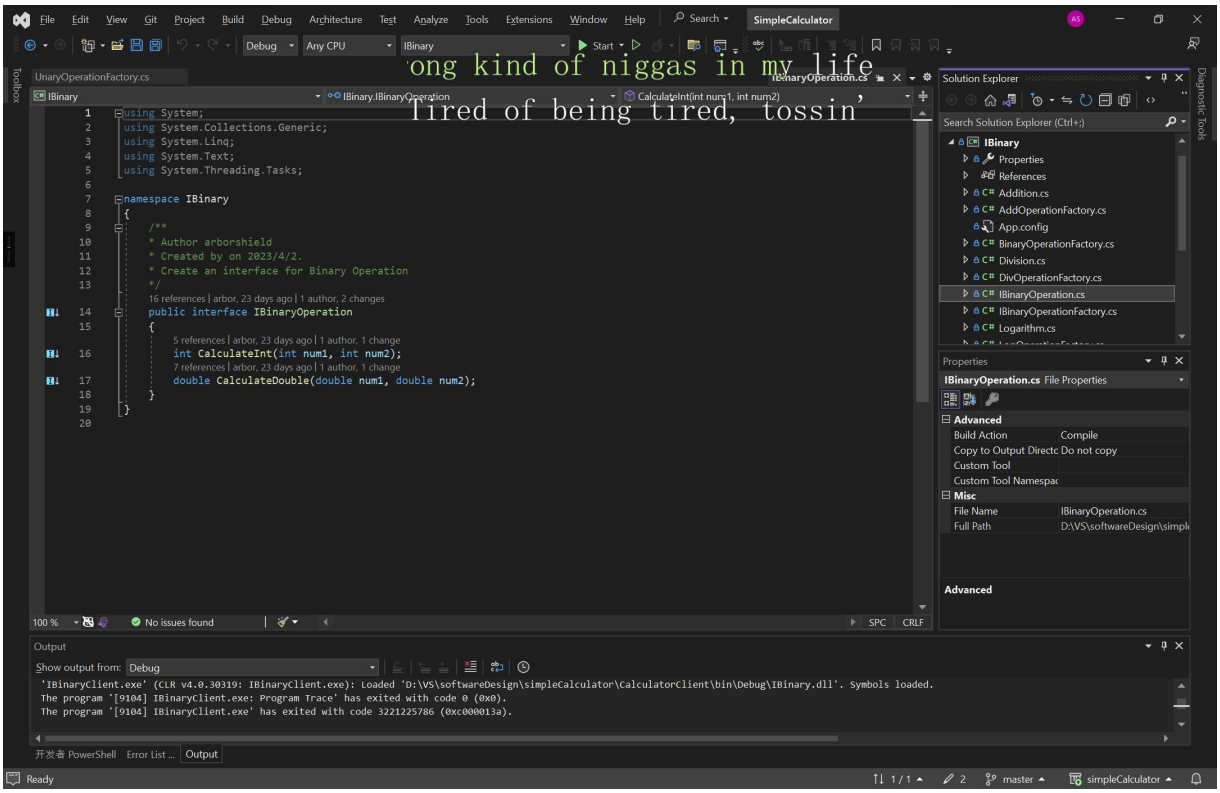

Every time I like someone, he
Trinity Home 作词: Trinity Home

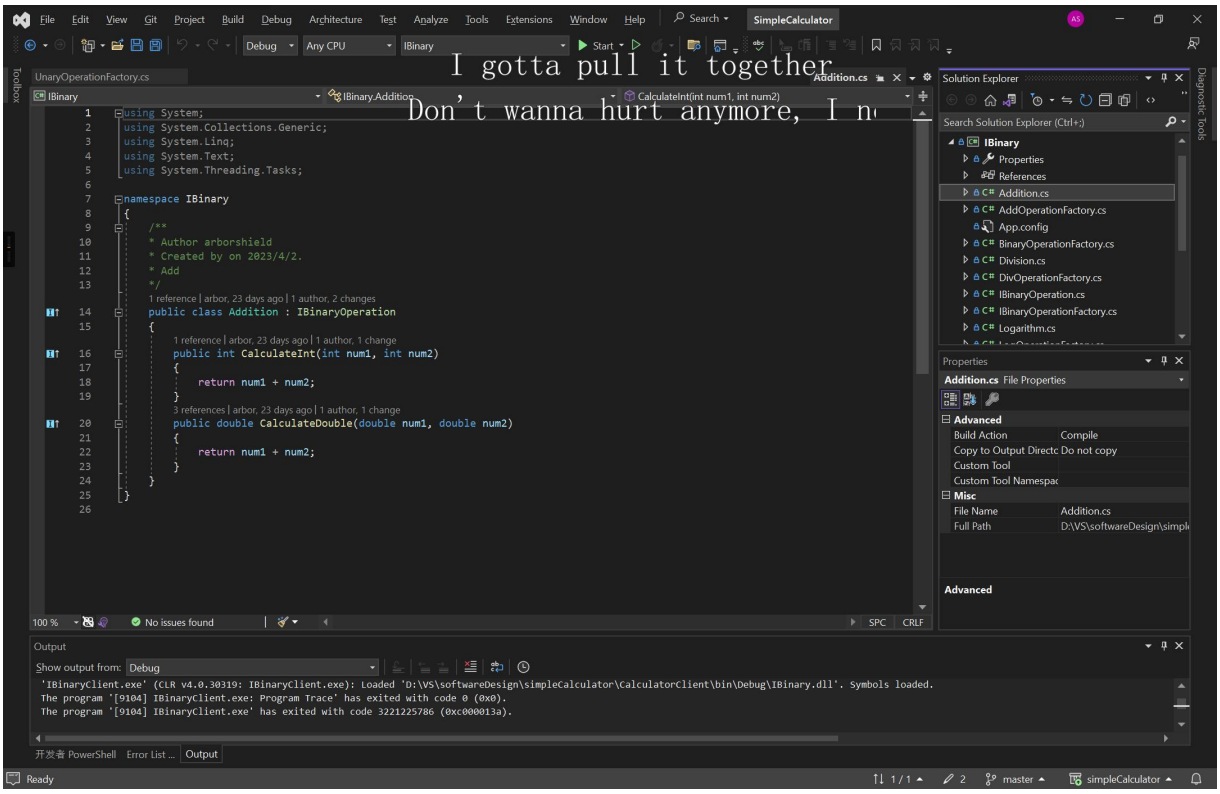

'eally think we're meant to me
Every nigga I tried to date,

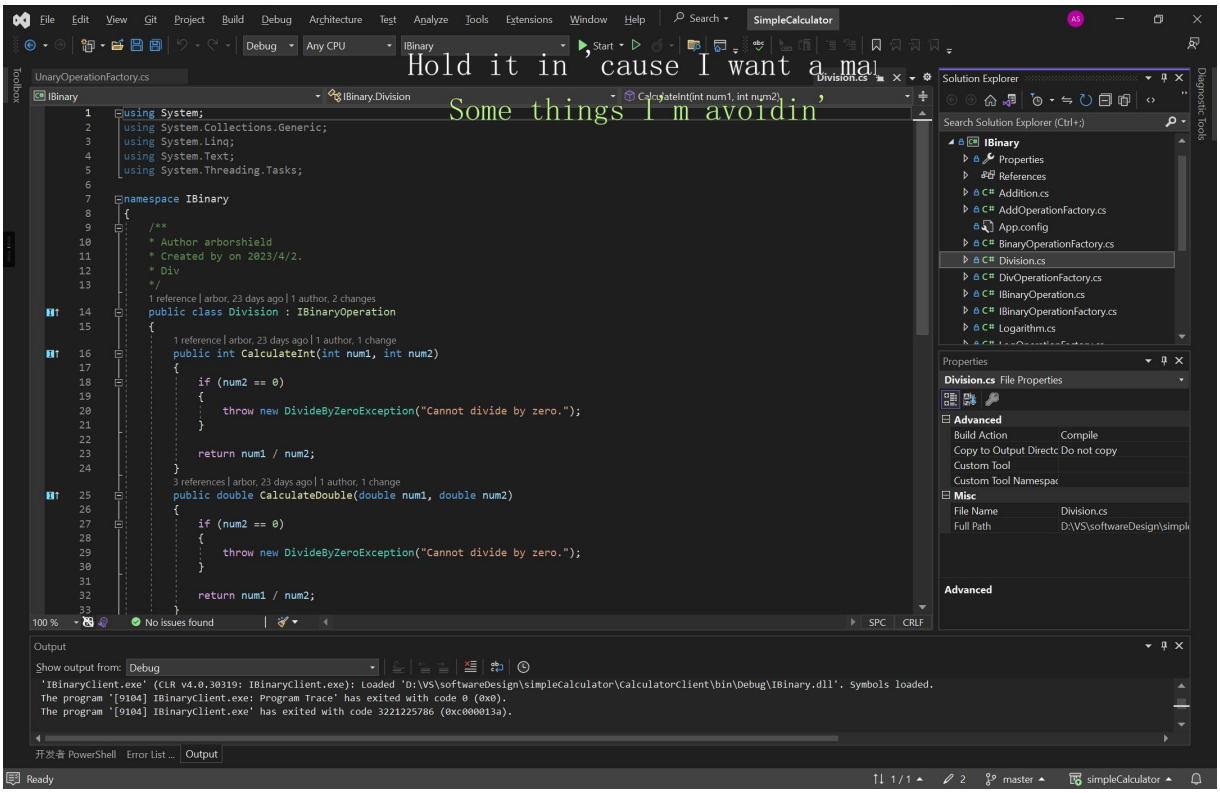

But every time I fall in love,
I don't wanna sound emotional


And even though I'm hurt insi
You don't know what pain is,

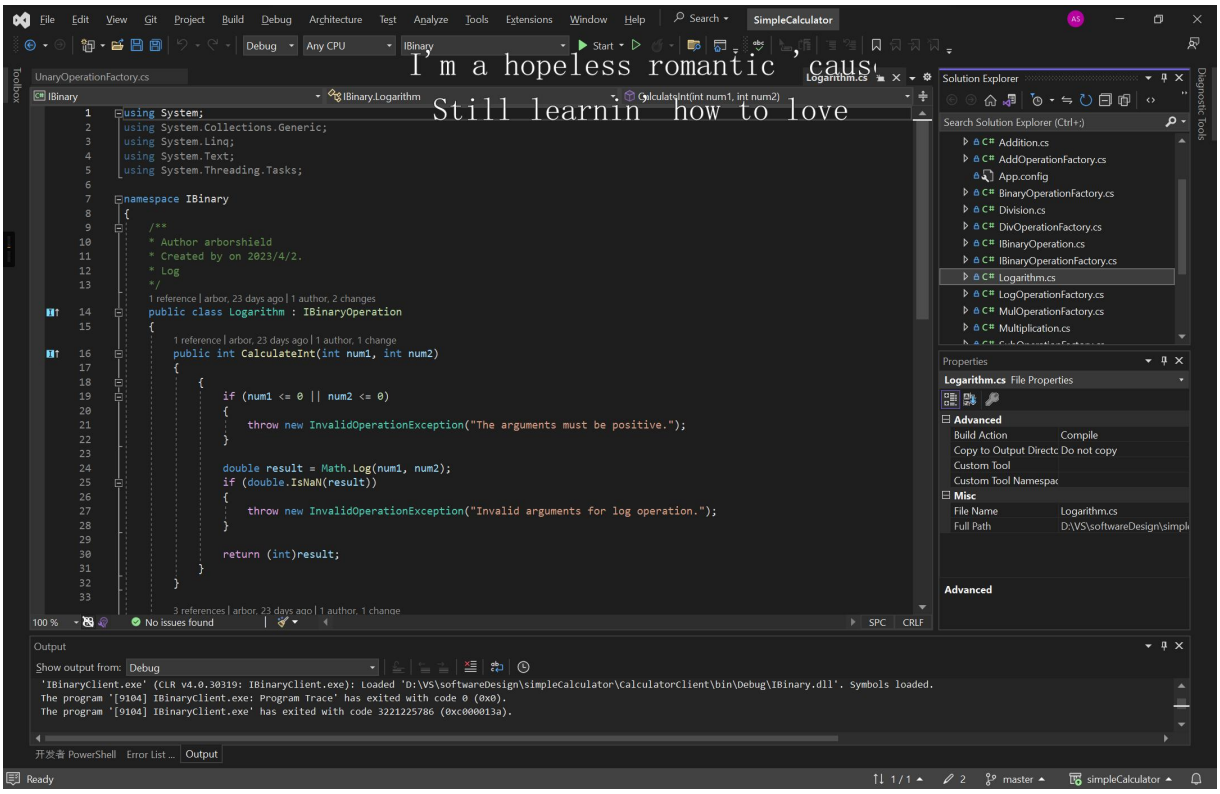
二元：


ong kind of niggas in my life
Tired of being tired, tossin'

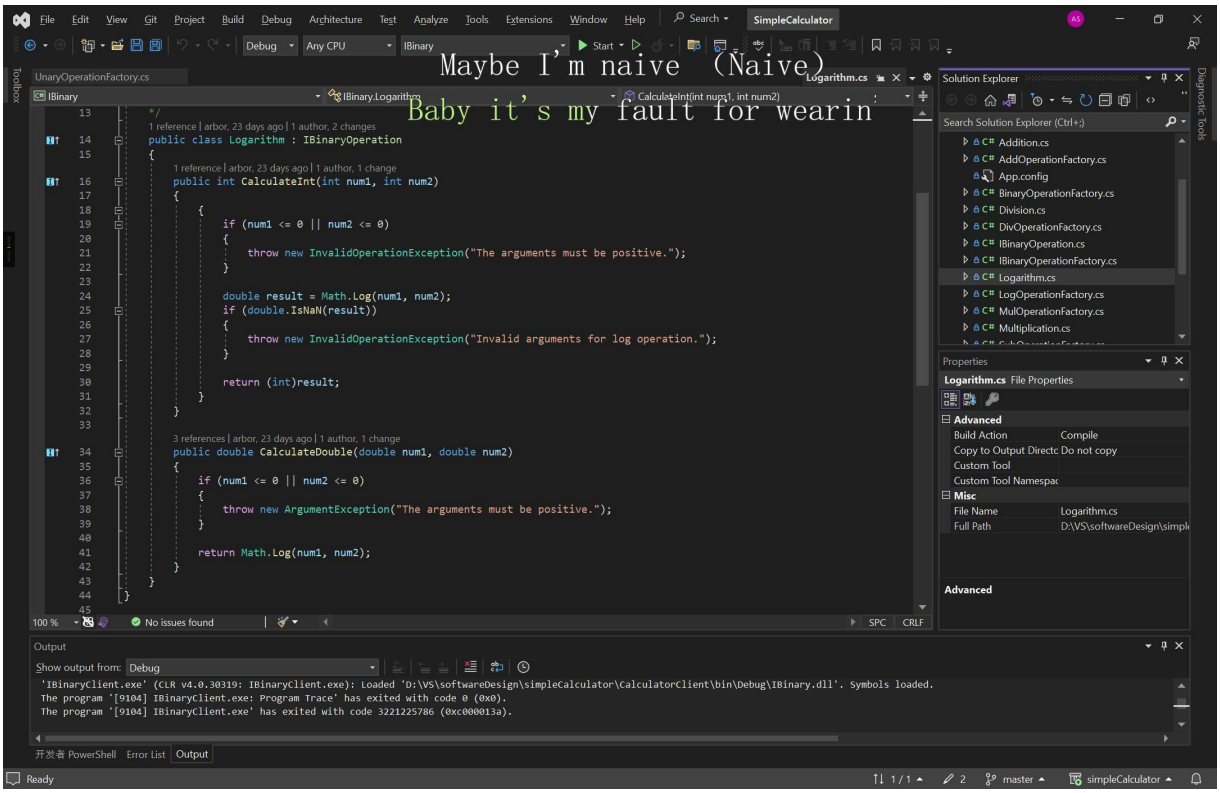

I gotta pull it together
Don't wanna hurt anymore, I n
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace IBinary
{
/**
* Author arborshield
* Created by on 2023/4/2.
* Add
*/
1 reference | arbor, 23 days ago | 1 author, 2 changes
public class Addition : IBinaryOperation
{
1 reference | arbor, 23 days ago | 1 author, 1 change
public int CalculateInt(int num1, int num2)
{
return num1 + num2;
}
3 references | arbor, 23 days ago | 1 author, 1 change
public double CalculateDouble(double num1, double num2)
{
return num1 + num2;
}
}
}
Solution Explorer
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
Addition.cs File Properties
Advanced
Build Action Compile
Copy to Output Directc Do not copy
Custom Tool
Custom Tool Namespac
Misc
File Name Addition.cs
Full Path D:\VS\softwareDesign\simpl
Advanced
No issues found
Output
Show output from: Debug
'IBinaryClient.exe' (CLR v4.0.30319: IBinaryClient.exe): Loaded 'D:\VS\softwareDesign\simpleCalculator\CalculatorClient\bin\Debug\IBinary.dll'. Symbols loaded.
The program '[9104] IBinaryClient.exe: Program Trace' has exited with code 0 (0x0).
The program '[9104] IBinaryClient.exe' has exited with code 3221225786 (0xc000013a).
Ready
master
simpleCalculator

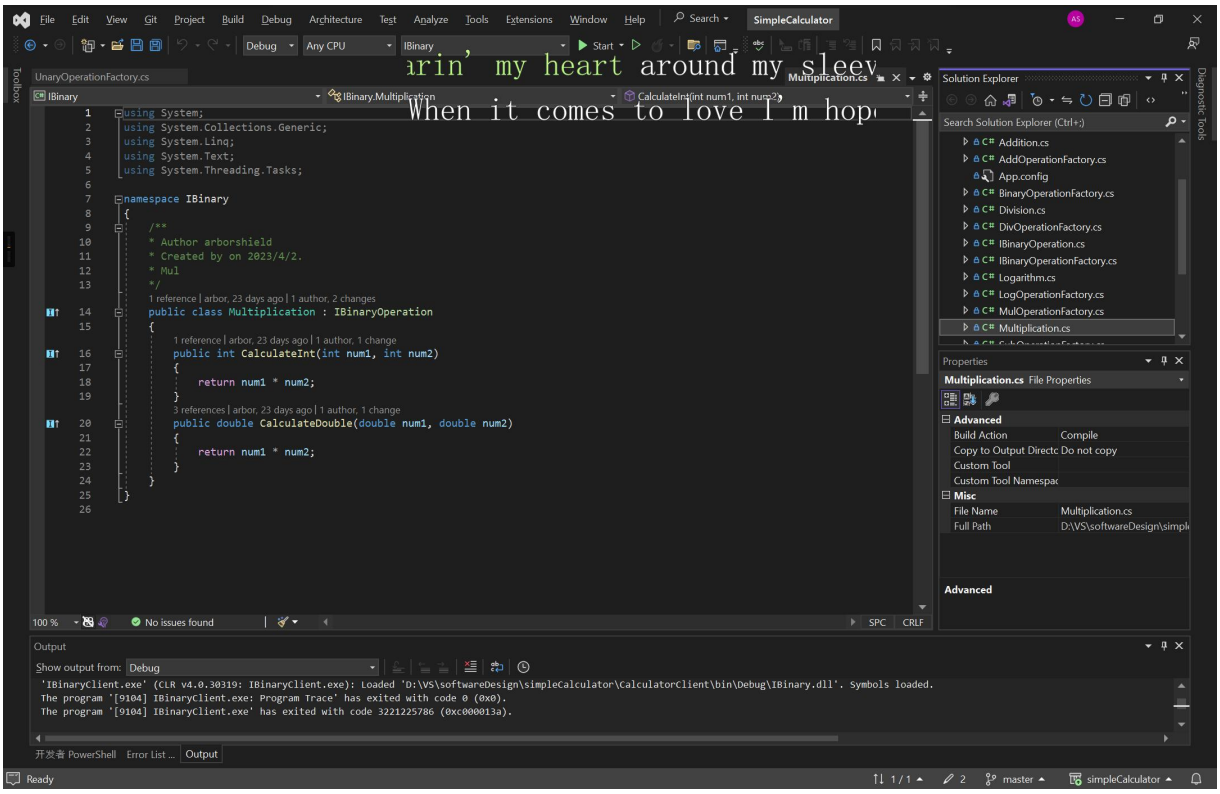

Hold it in 'cause I want a ma
Some things I'm avoidin
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace IBinary
{
/**
* Author arborshield
* Created by on 2023/4/2.
* Div
*/
1 reference | arbor, 23 days ago | 1 author, 2 changes
public class Division : IBinaryOperation
{
1 reference | arbor, 23 days ago | 1 author, 1 change
public int CalculateInt(int num1, int num2)
{
if (num2 == 0)
{
throw new DivideByZeroException("Cannot divide by zero.");
}
return num1 / num2;
}
3 references | arbor, 23 days ago | 1 author, 1 change
public double CalculateDouble(double num1, double num2)
{
if (num2 == 0)
{
throw new DivideByZeroException("Cannot divide by zero.");
}
return num1 / num2;
}
Solution Explorer
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
Division.cs File Properties
Advanced
Build Action Compile
Copy to Output Directc Do not copy
Custom Tool
Custom Tool Namespac
Misc
File Name Division.cs
Full Path D:\VS\softwareDesign\simpl
Advanced
No issues found
Output
Show output from: Debug
'IBinaryClient.exe' (CLR v4.0.30319: IBinaryClient.exe): Loaded 'D:\VS\softwareDesign\simpleCalculator\CalculatorClient\bin\Debug\IBinary.dll'. Symbols loaded.
The program '[9104] IBinaryClient.exe: Program Trace' has exited with code 0 (0x0).
The program '[9104] IBinaryClient.exe' has exited with code 3221225786 (0xc000013a).
Ready
master
simpleCalculator

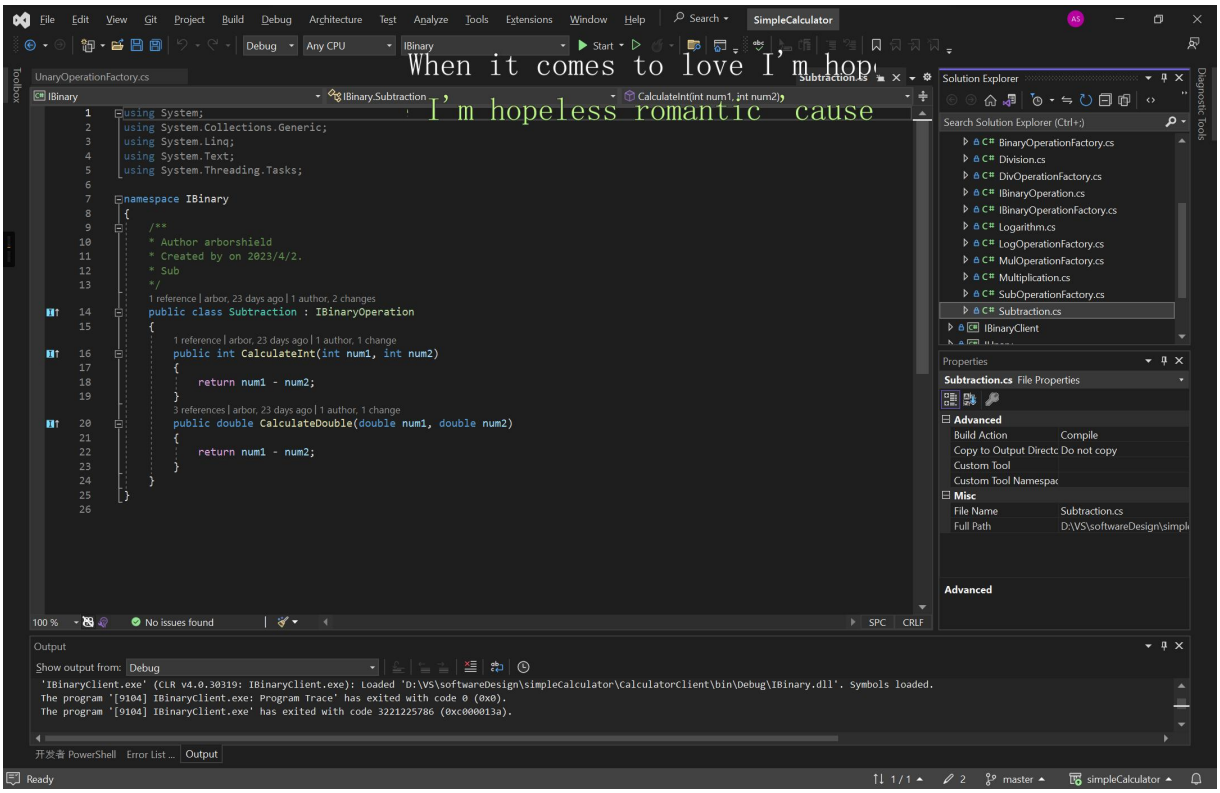
I'm a hopeless romantic 'caus
Still learnin' how to love



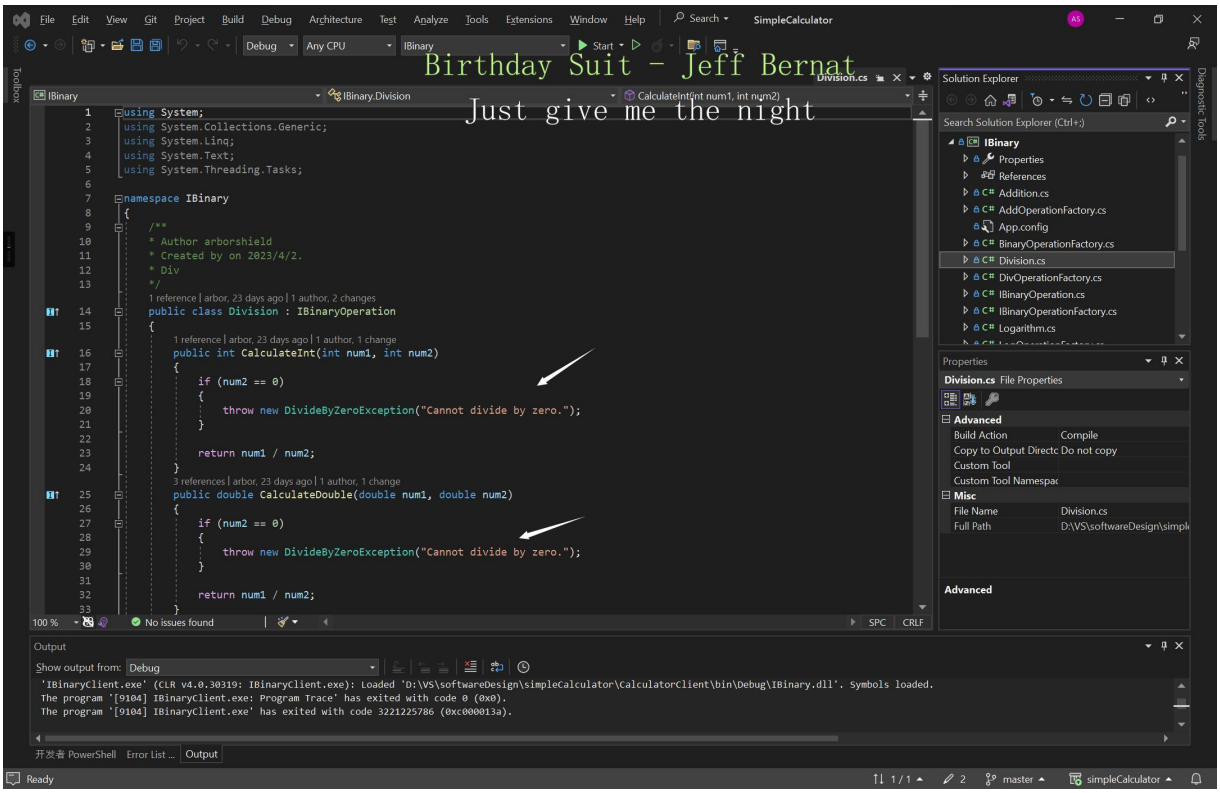

Maybe I'm naive (Naive)
Baby it's my fault for wearin





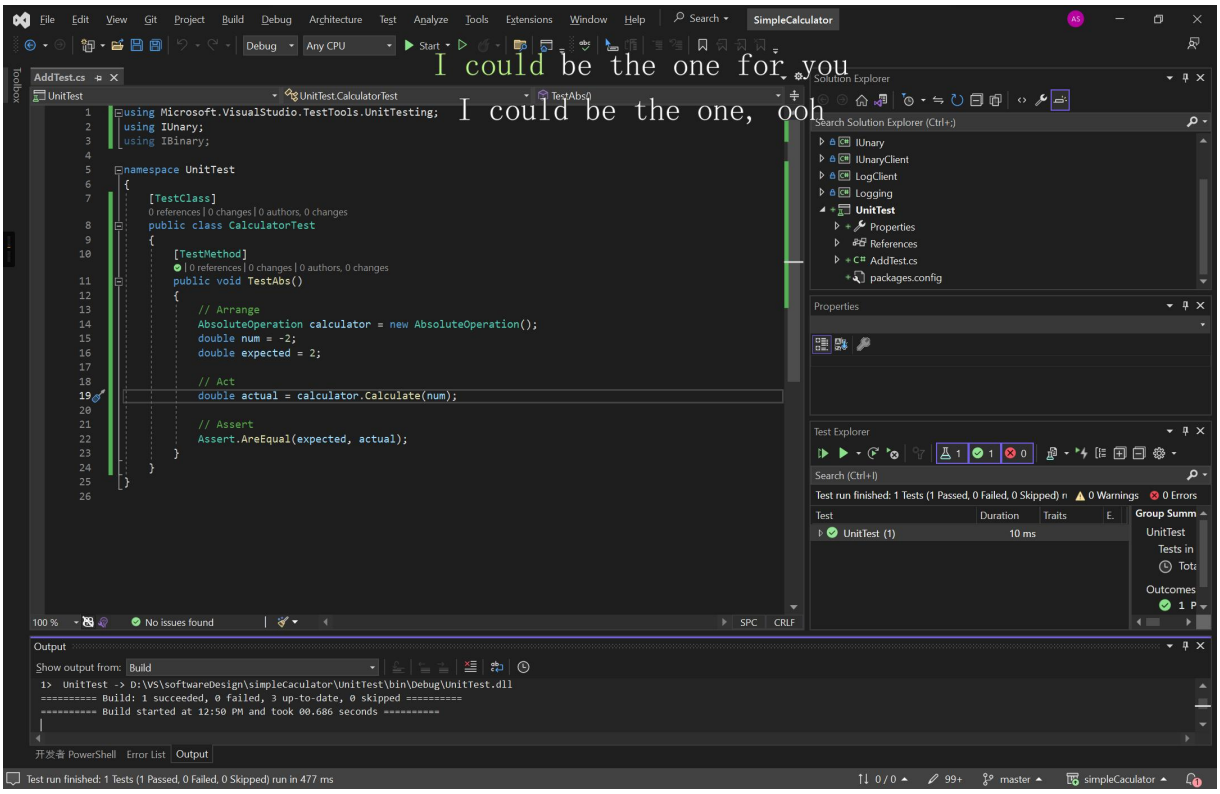



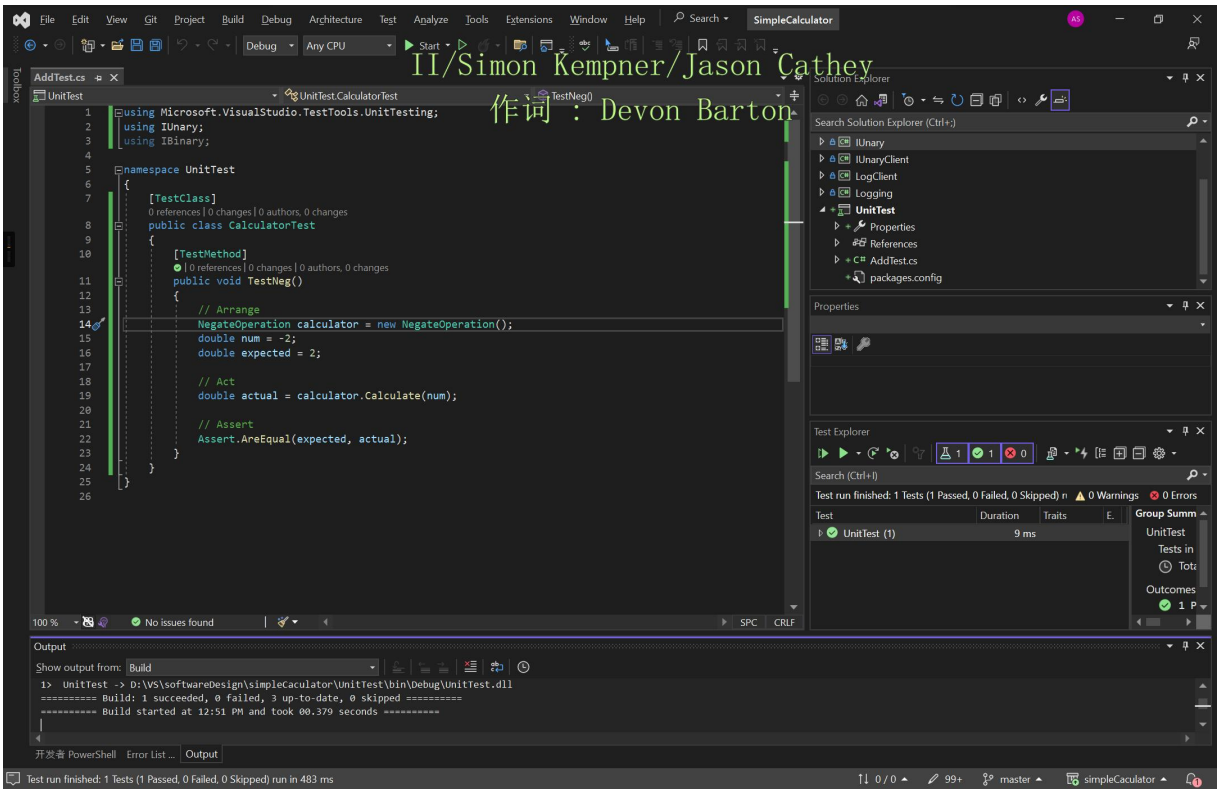

（2）对于整数除法和浮点型数的除法，考虑分母为零的情况

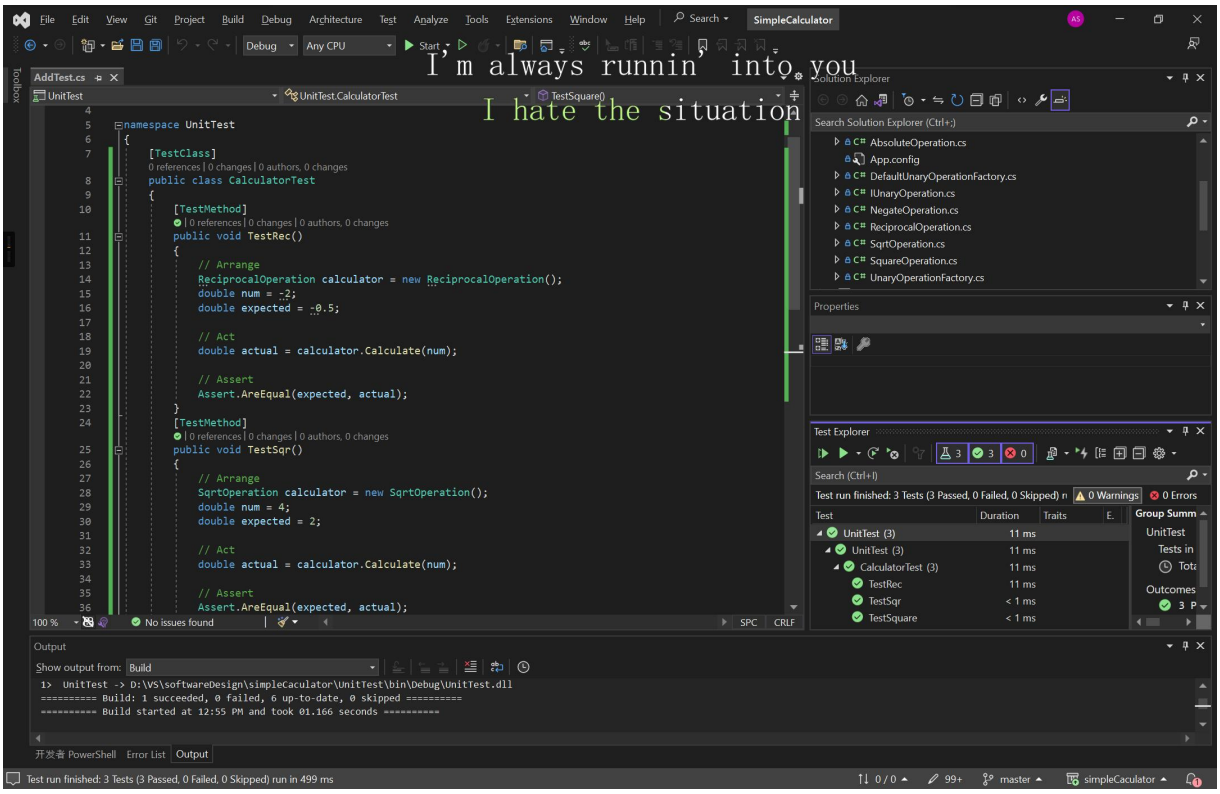


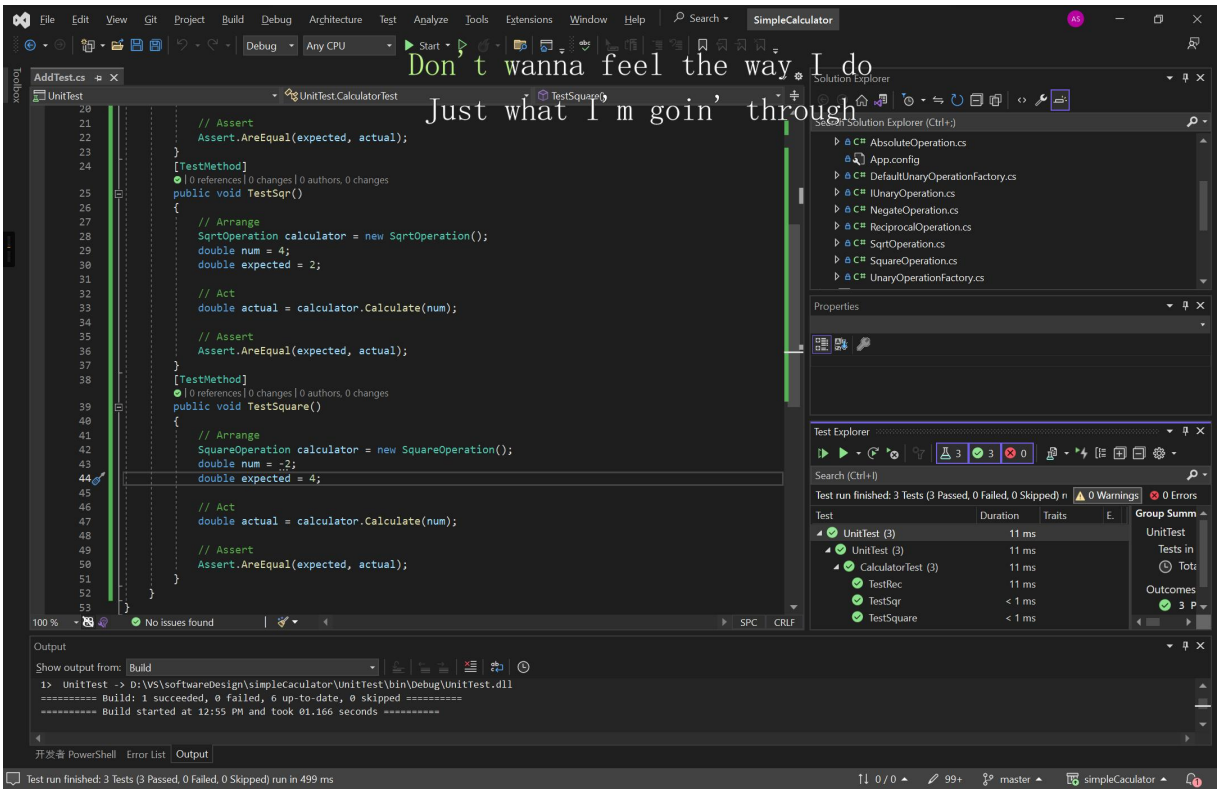

（3）基于 AAA 原则对于一元计算模块、二元计算模块和日志读写模块进行单元测试

一元：

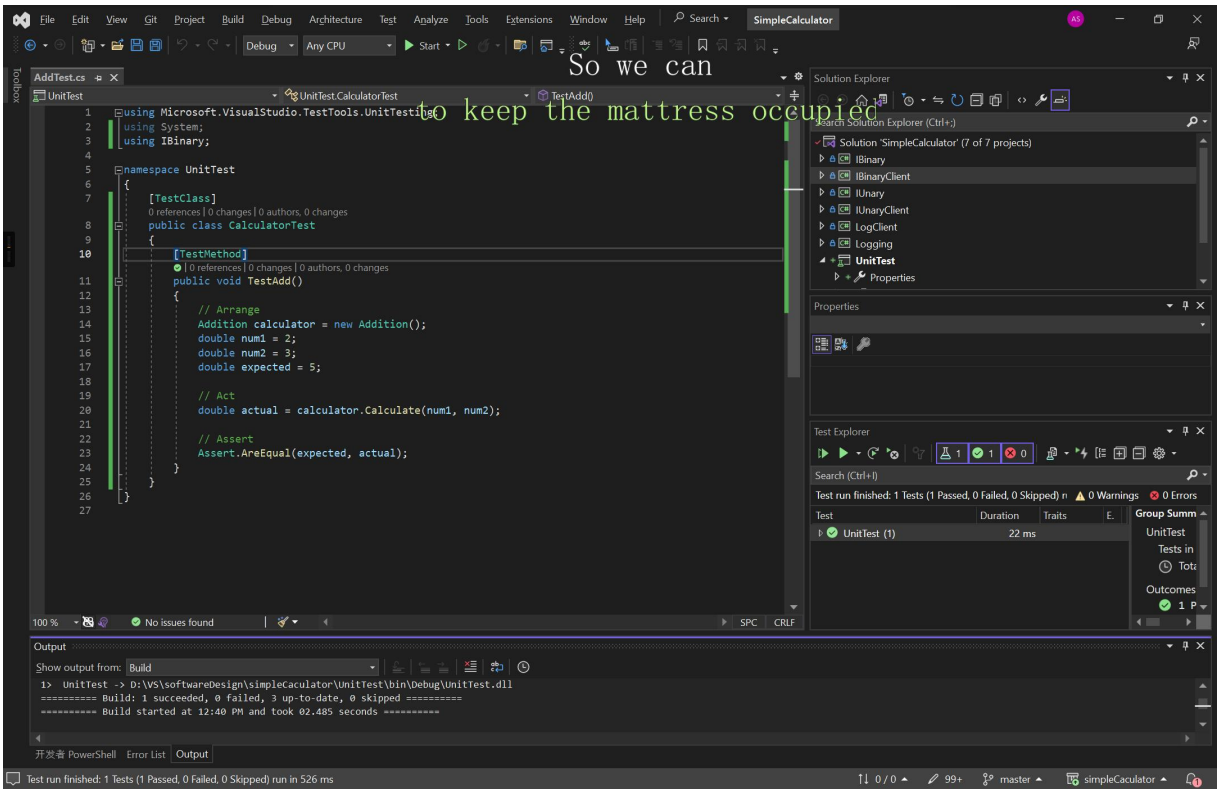

I could be the one for you
I could be the one, ooh
using Microsoft.VisualStudio.TestTools.UnitTesting;
using IUnary;
using IBinary;
namespace UnitTest
{
[TestClass]
public class CalculatorTest
{
[TestMethod]
public void TestAbs()
{
// Arrange
AbsoluteOperation calculator = new AbsoluteOperation();
double num = -2;
double expected = 2;
// Act
double actual = calculator.Calculate(num);
// Assert
Assert.AreEqual(expected, actual);
}
}
}
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 477 ms

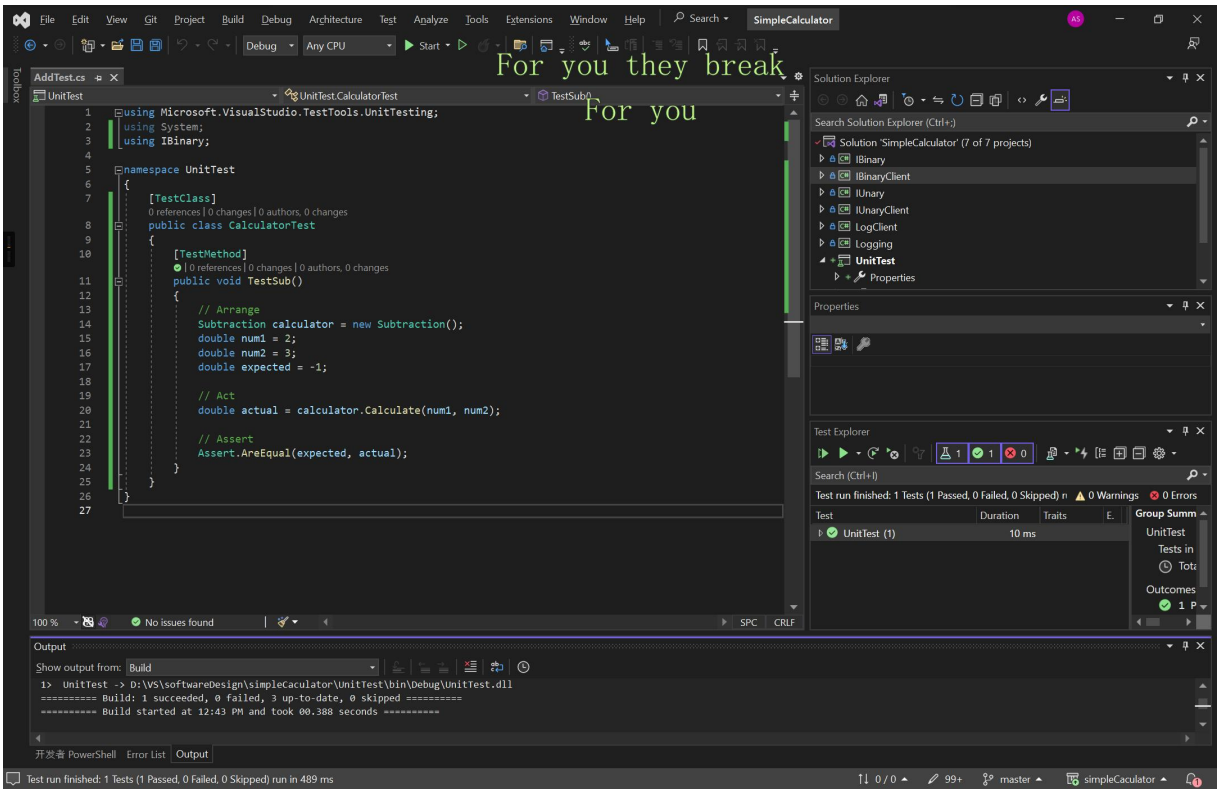

II/Simon Kempner/Jason Cathey
作词：Devon Barton
using Microsoft.VisualStudio.TestTools.UnitTesting;
using IUnary;
using IBinary;
namespace UnitTest
{
[TestClass]
public class CalculatorTest
{
[TestMethod]
public void TestNeg()
{
// Arrange
NegateOperation calculator = new NegateOperation();
double num = -2;
double expected = 2;
// Act
double actual = calculator.Calculate(num);
// Assert
Assert.AreEqual(expected, actual);
}
}
}
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 483 ms

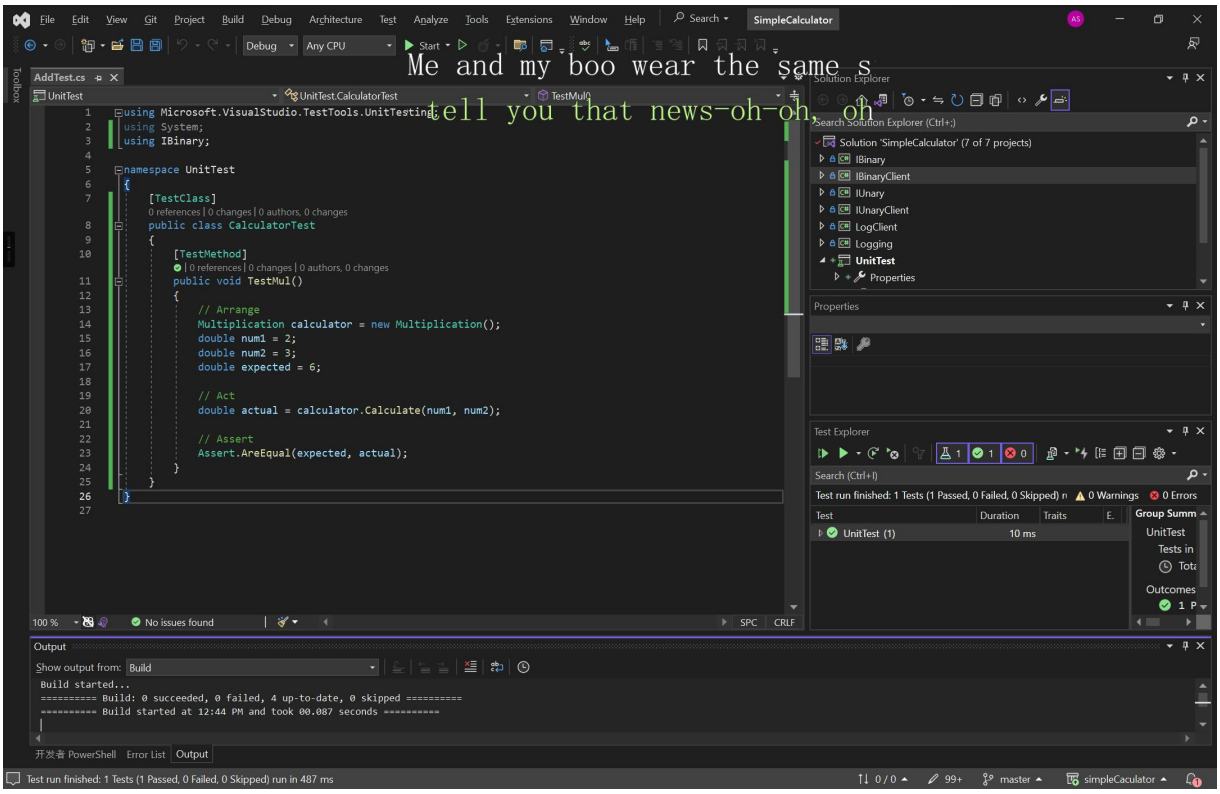

I'm always runnin' into you
I hate the situation

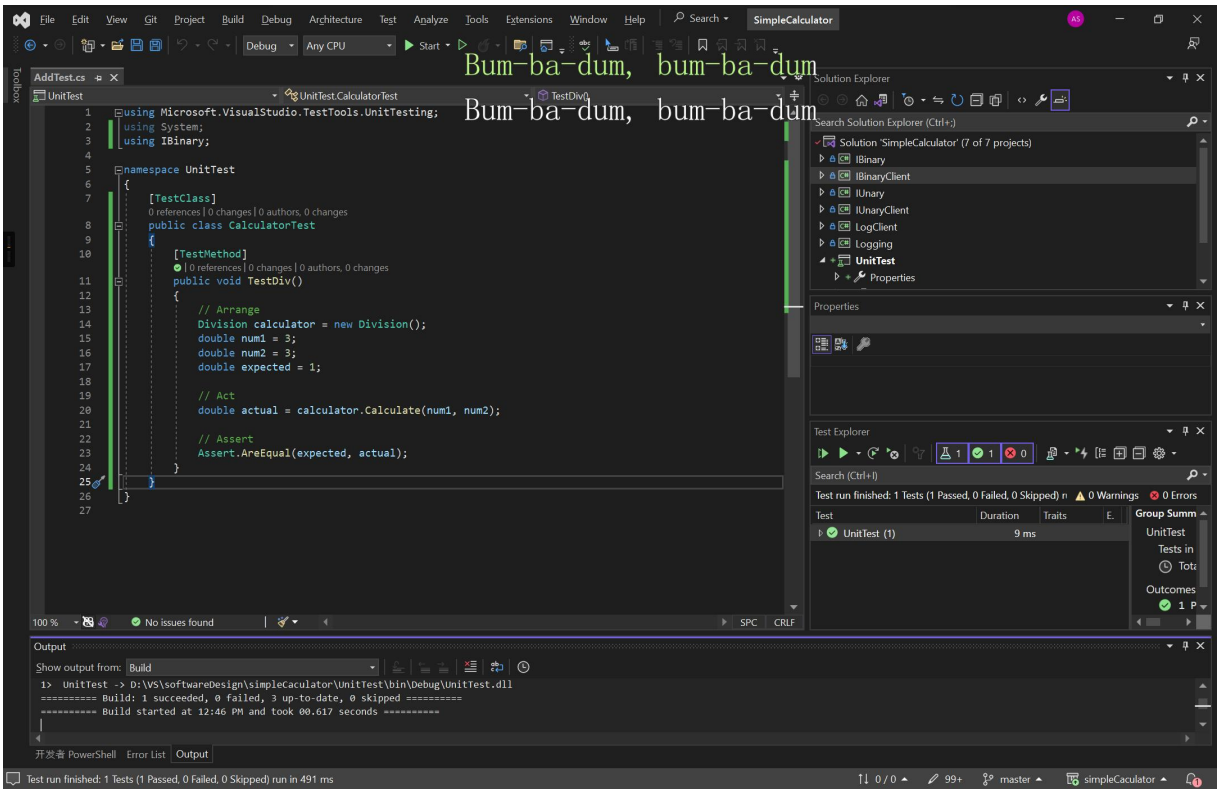

Don't wanna feel the way I do
Just what I'm goin' through

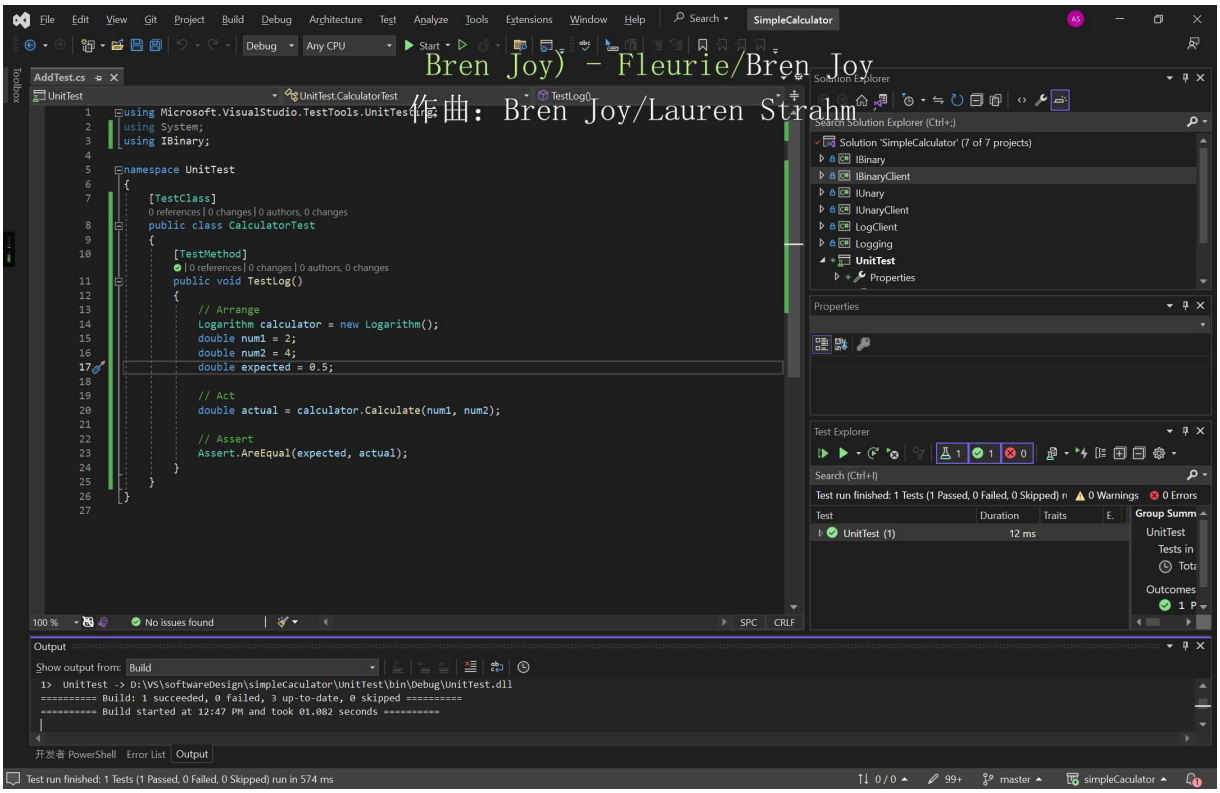
二元：


So we can
to keep the mattress occupied

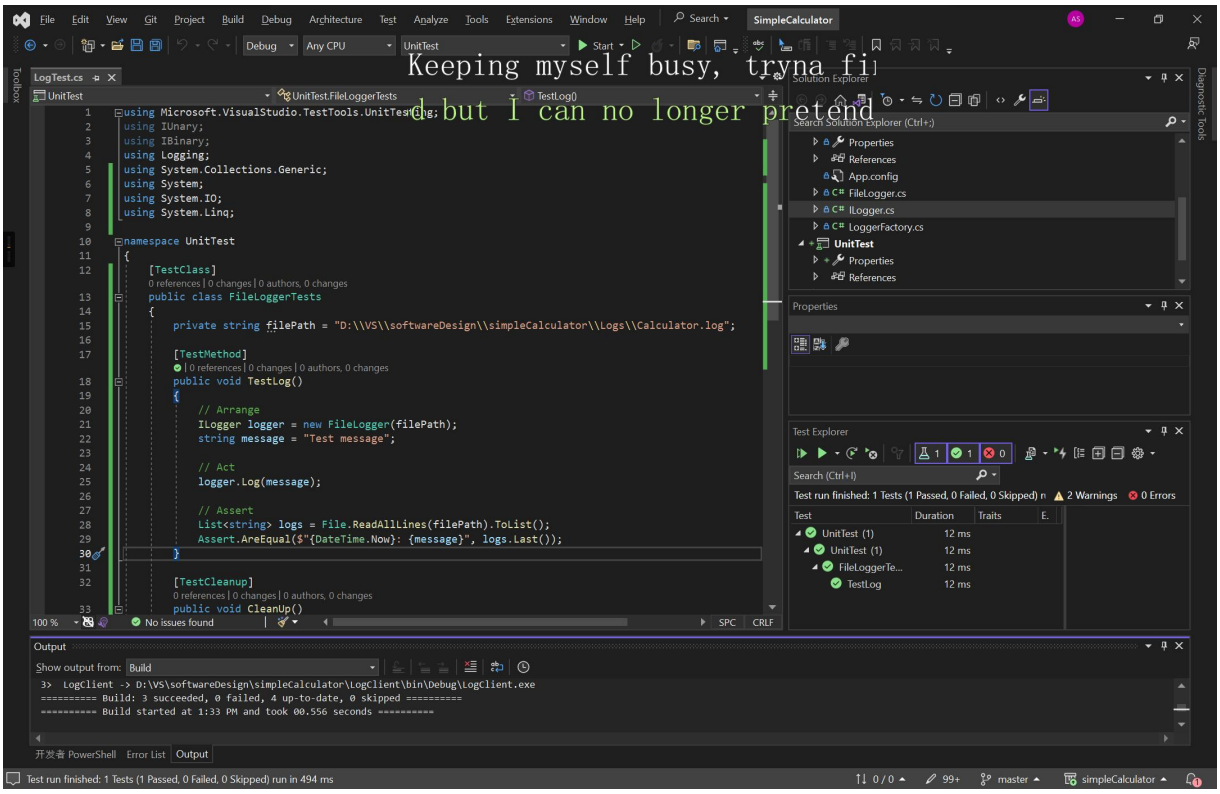

For you they break
For you

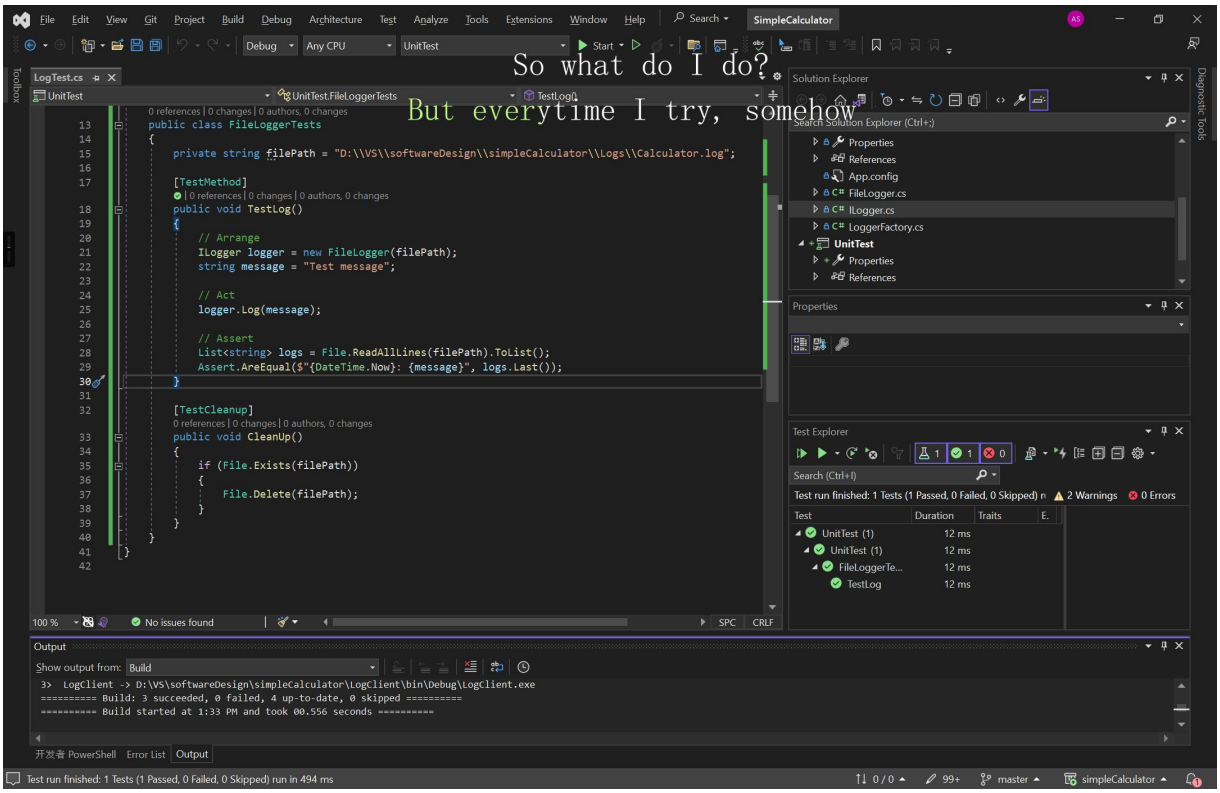

Me and my boo wear the same s
tell you that news-oh-oh, oh
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System;
using IBinary;
namespace UnitTest
{
[TestClass]
public class CalculatorTest
{
[TestMethod]
public void TestMul()
{
// Arrange
Multiplication calculator = new Multiplication();
double num1 = 2;
double num2 = 3;
double expected = 6;
// Act
double actual = calculator.Calculate(num1, num2);
// Assert
Assert.AreEqual(expected, actual);
}
}
}
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 487 ms


Bum-ba-dum, bum-ba-dum
Bum-ba-dum, bum-ba-dum
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System;
using IBinary;
namespace UnitTest
{
[TestClass]
public class CalculatorTest
{
[TestMethod]
public void TestDiv()
{
// Arrange
Division calculator = new Division();
double num1 = 3;
double num2 = 3;
double expected = 1;
// Act
double actual = calculator.Calculate(num1, num2);
// Assert
Assert.AreEqual(expected, actual);
}
}
}
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 491 ms

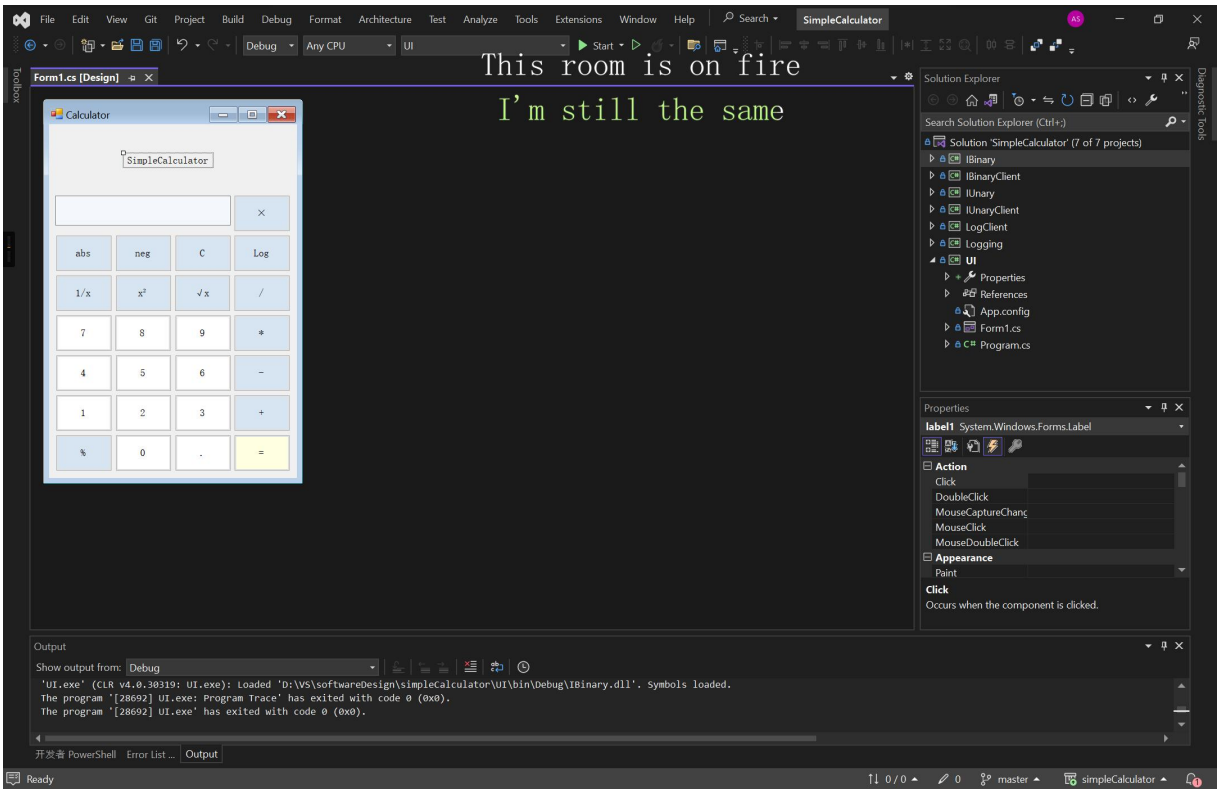

Bren Joy) - Fleurie/Bren Joy
作曲：Bren Joy/Lauren Strahm
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System;
using IBinary;
namespace UnitTest
{
[TestClass]
public class CalculatorTest
{
[TestMethod]
public void TestLog()
{
// Arrange
Logarithm calculator = new Logarithm();
double num1 = 2;
double num2 = 4;
double expected = 0.5;
// Act
double actual = calculator.Calculate(num1, num2);
// Assert
Assert.AreEqual(expected, actual);
}
}
}
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 574 ms

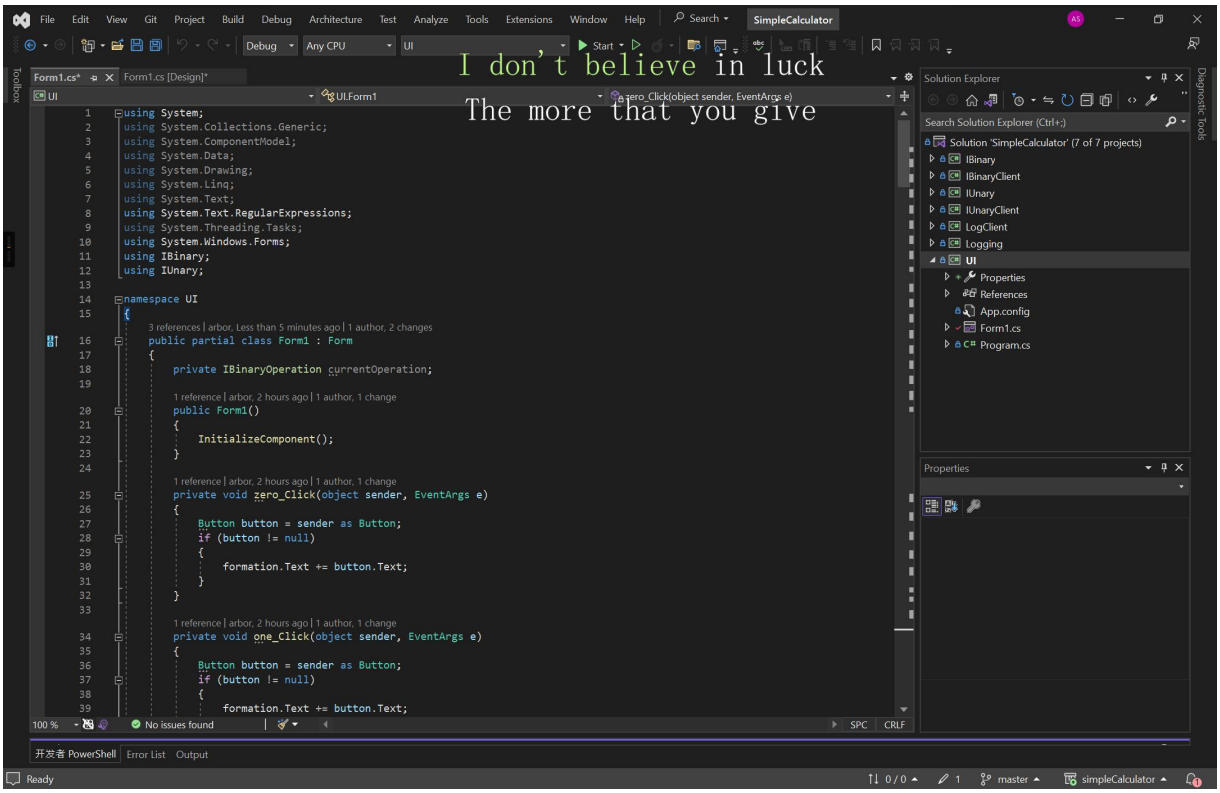
日志：

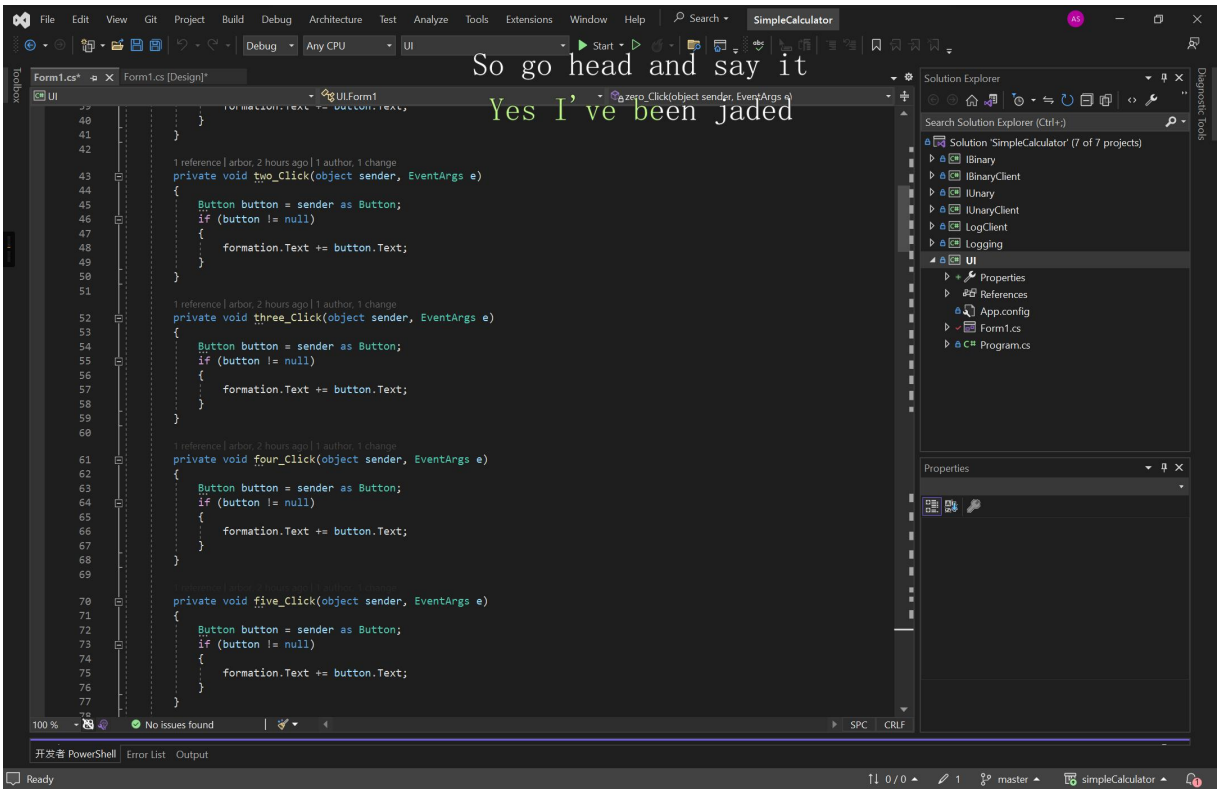


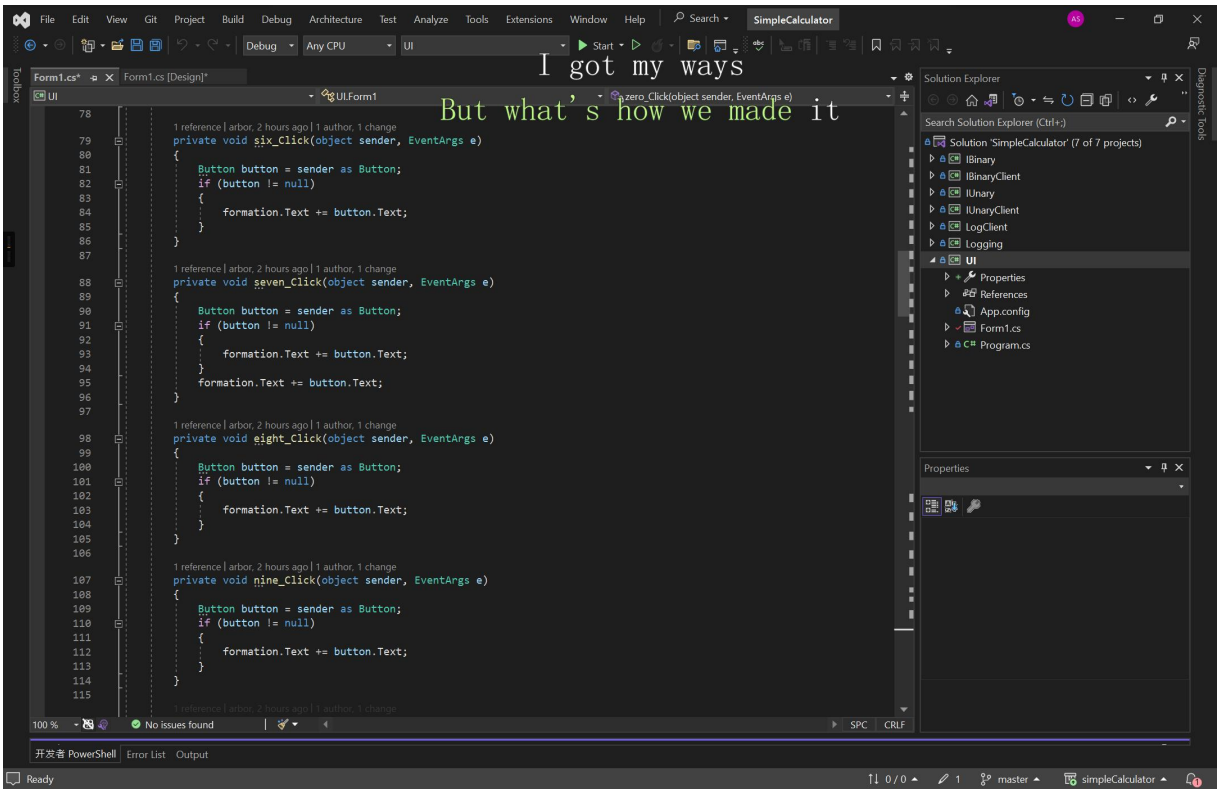



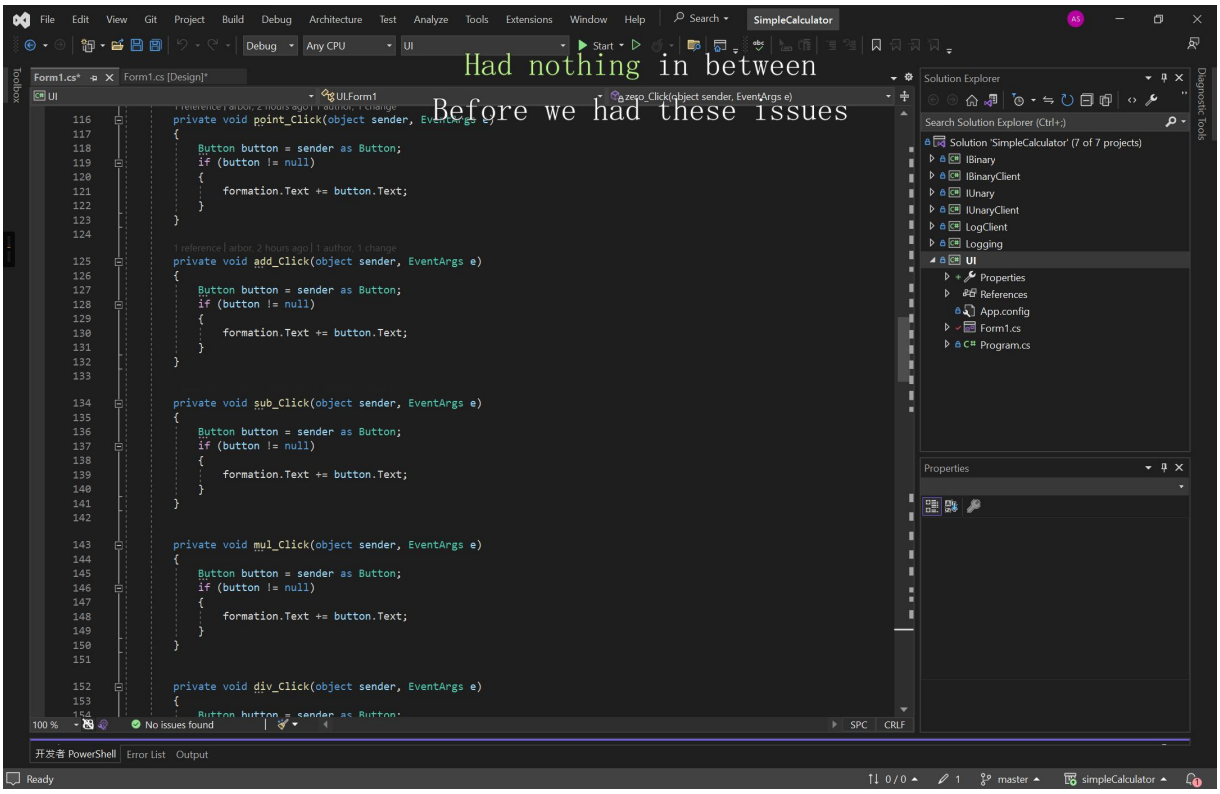

（4）参考 windows 自带计算器实现计算器的 UI 界面，在 UI 界面中使用之前实现的一元计算机和二元计算库

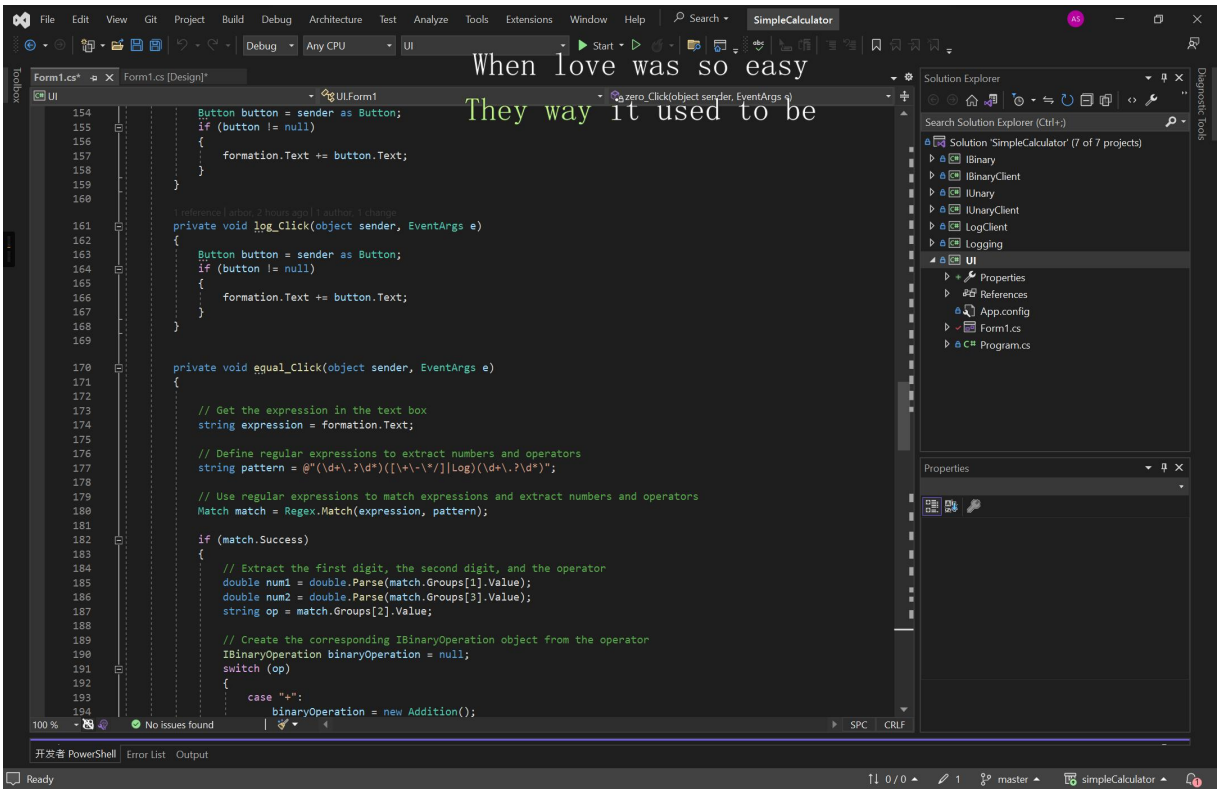

This room is on fire
I'm still the same

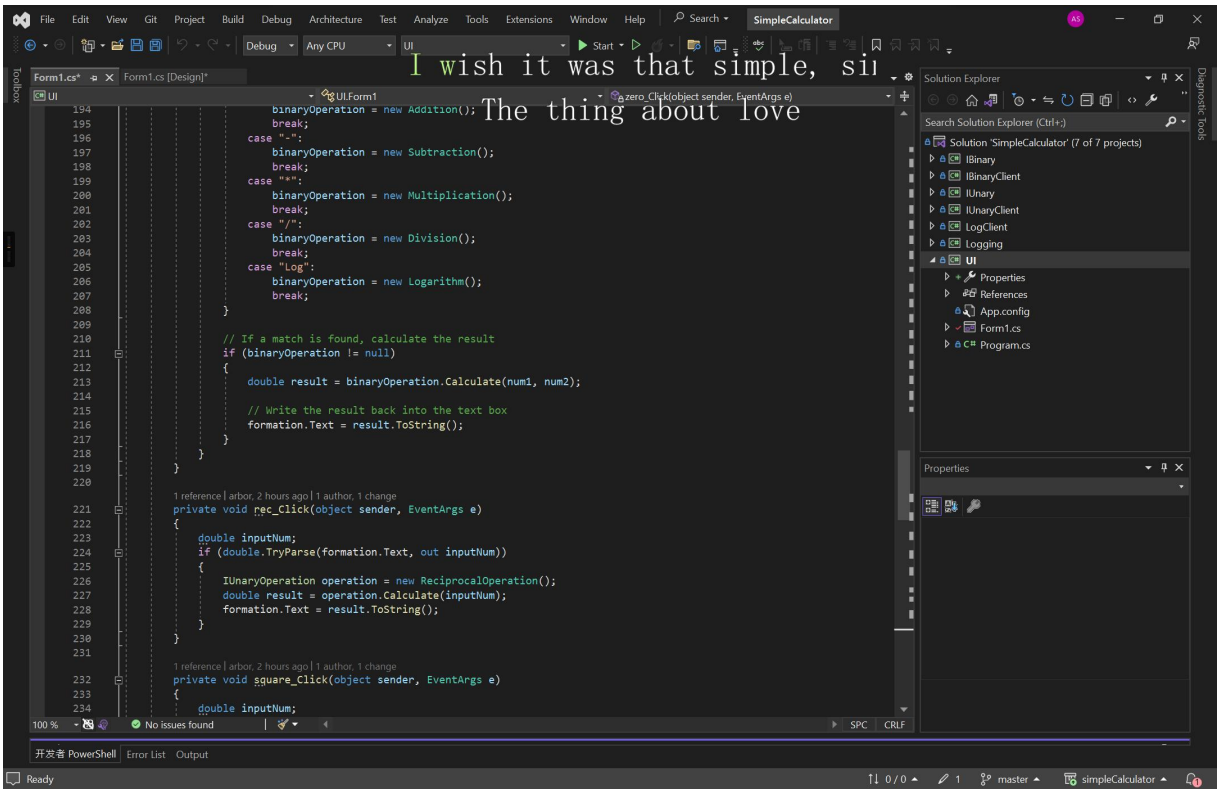

I don't believe in luck
The more that you give

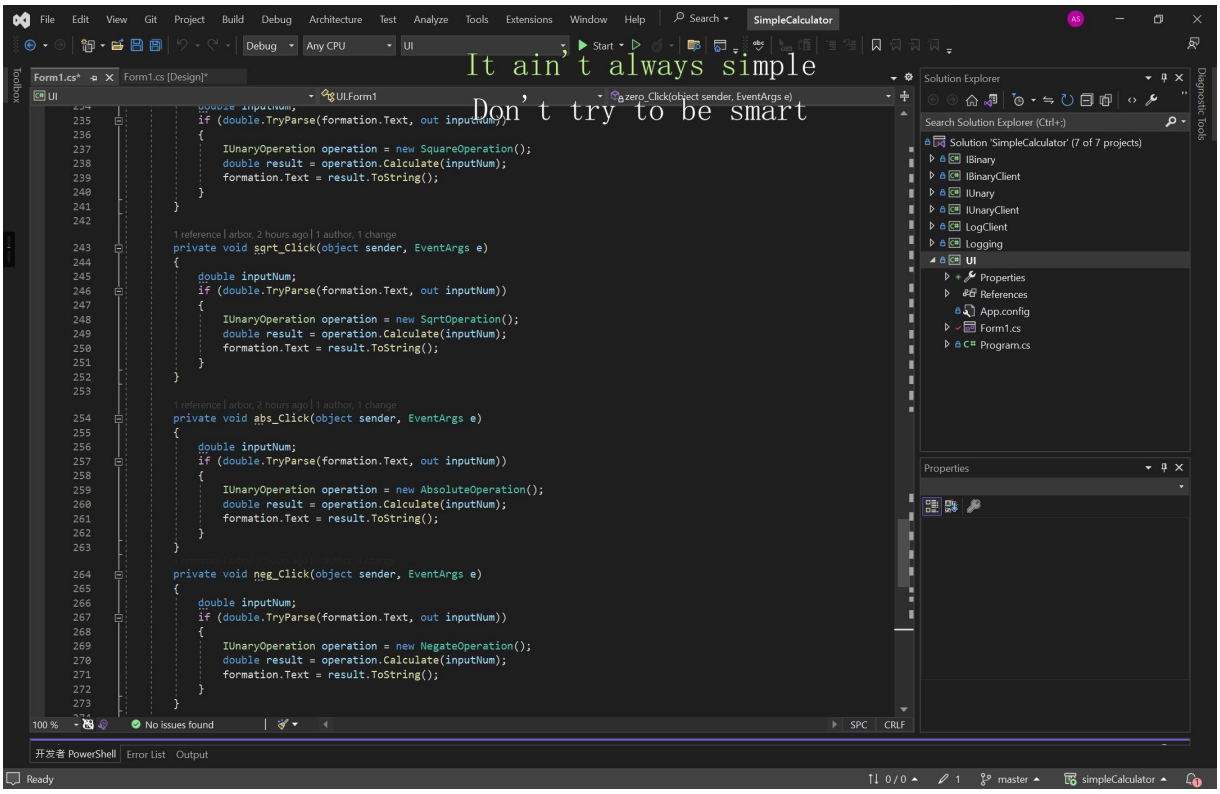

So go head and say it
Yes I've been jaded

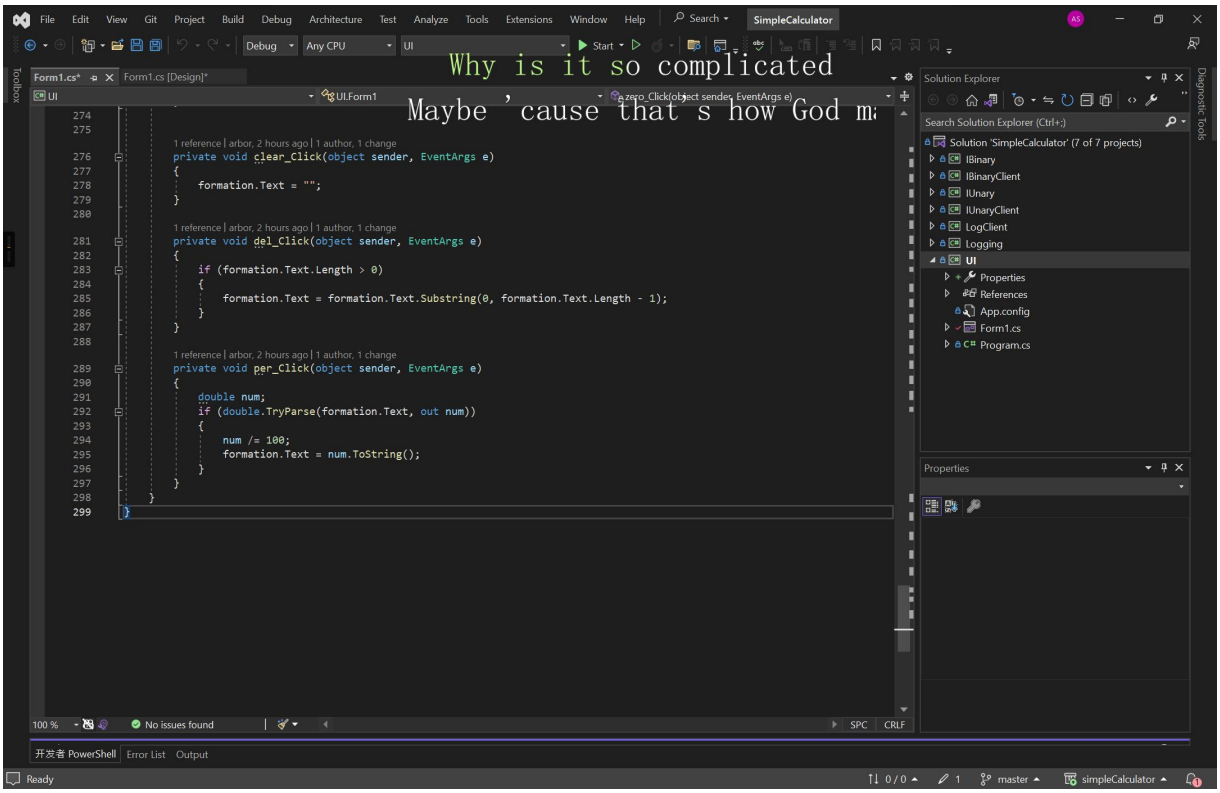

I got my ways
But what's how we made it


Had nothing in between
Before we had these issues

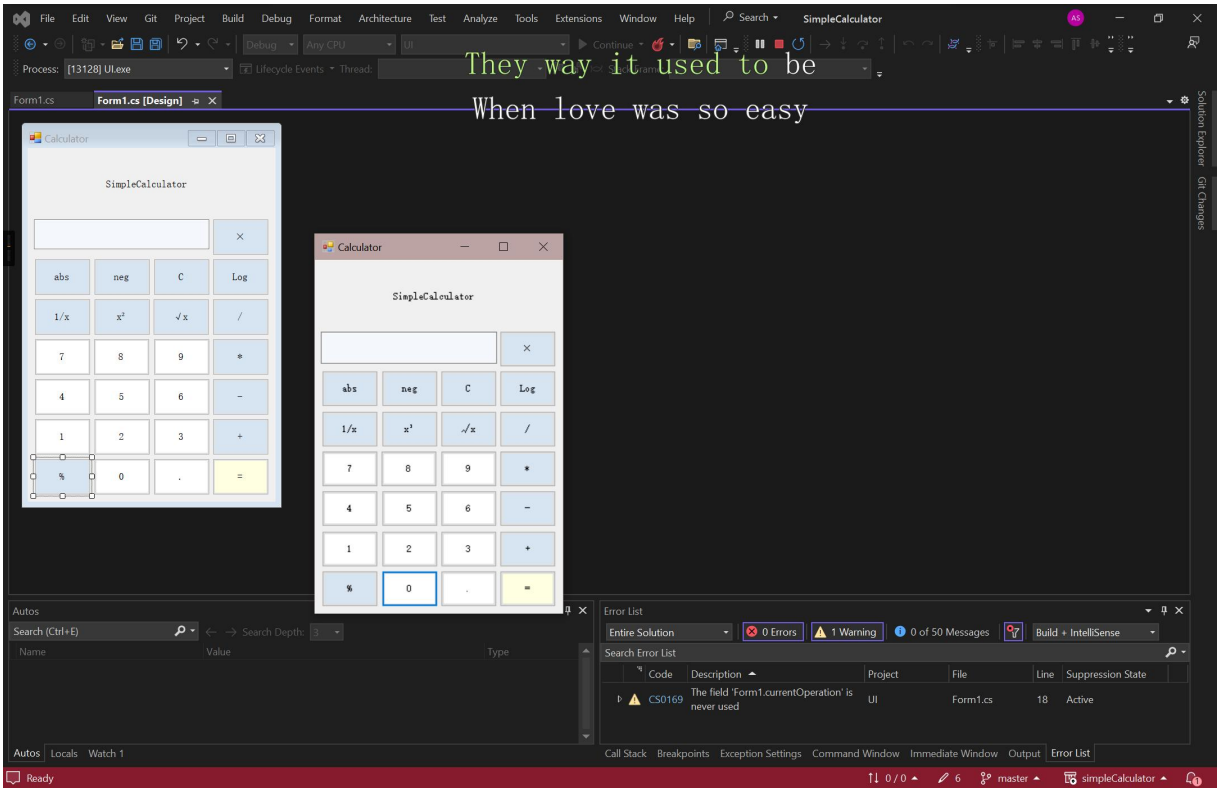

When love was so easy
They way it used to be

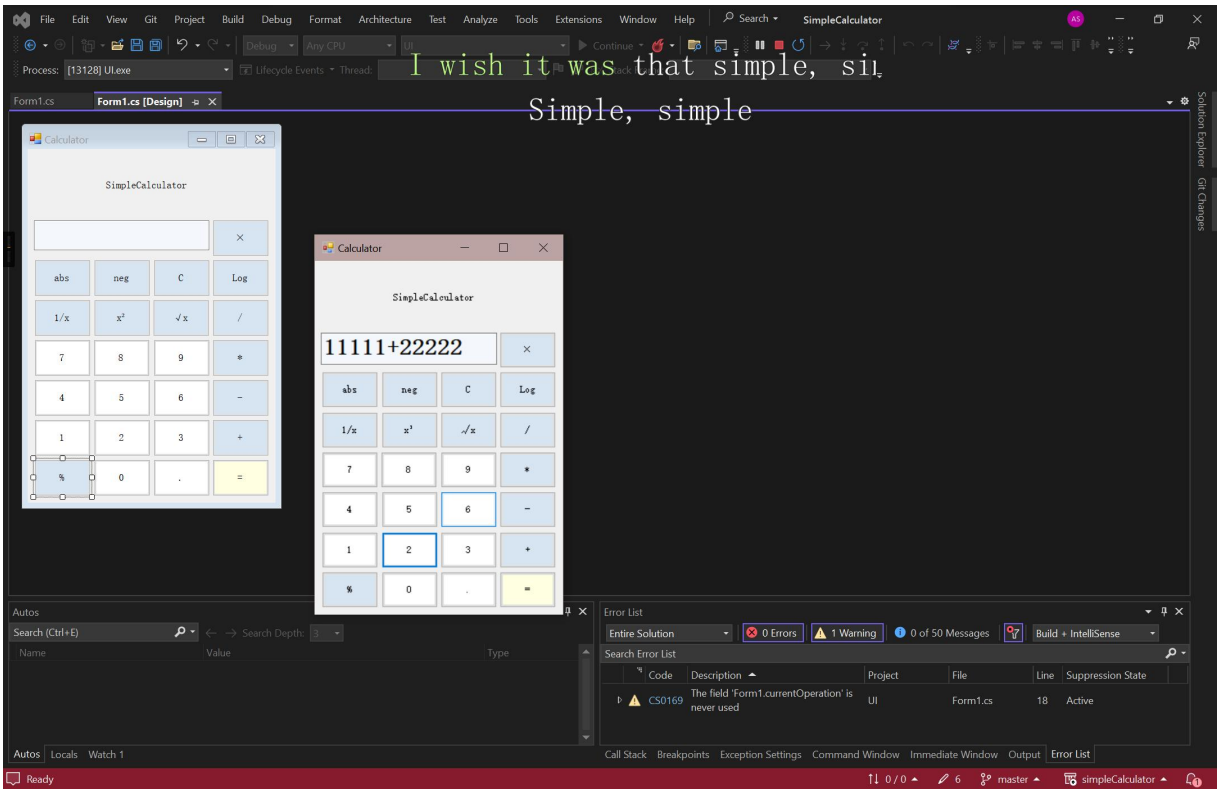

I wish it was that simple, sii
The thing about love

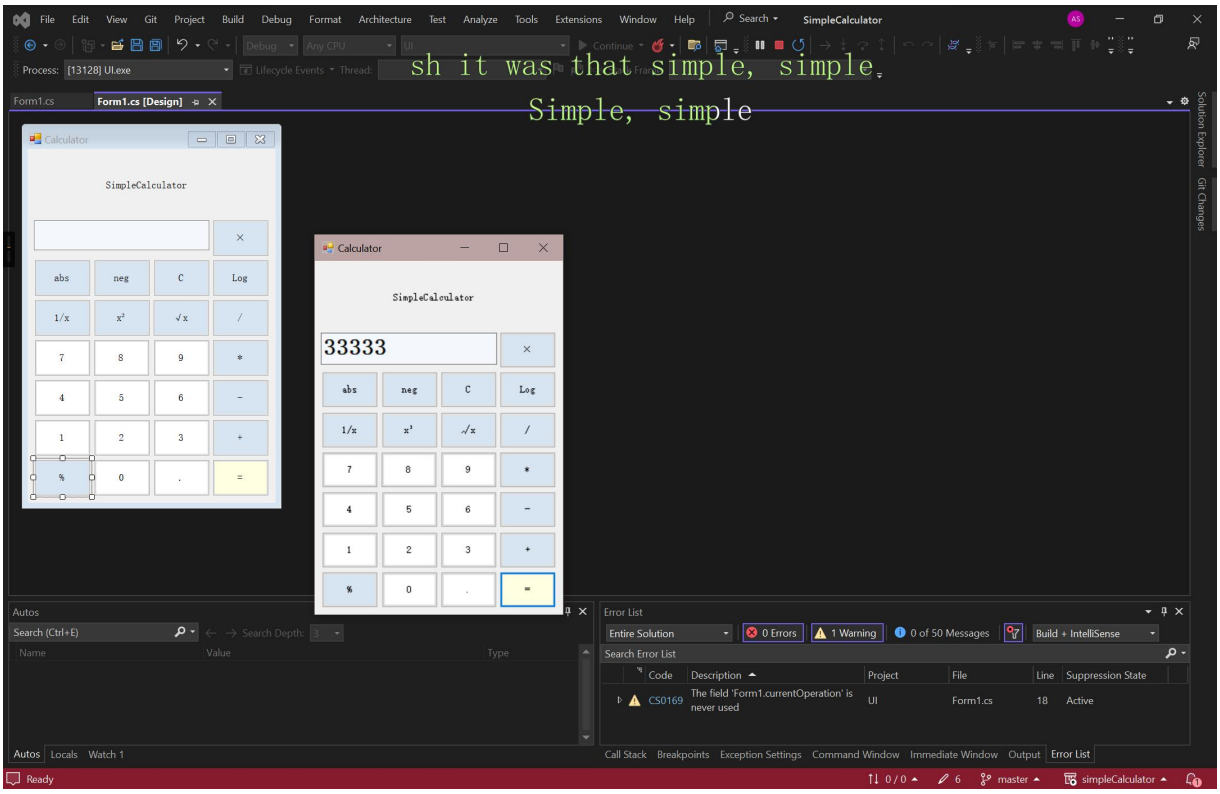

It ain't always simple
Don't try to be smart

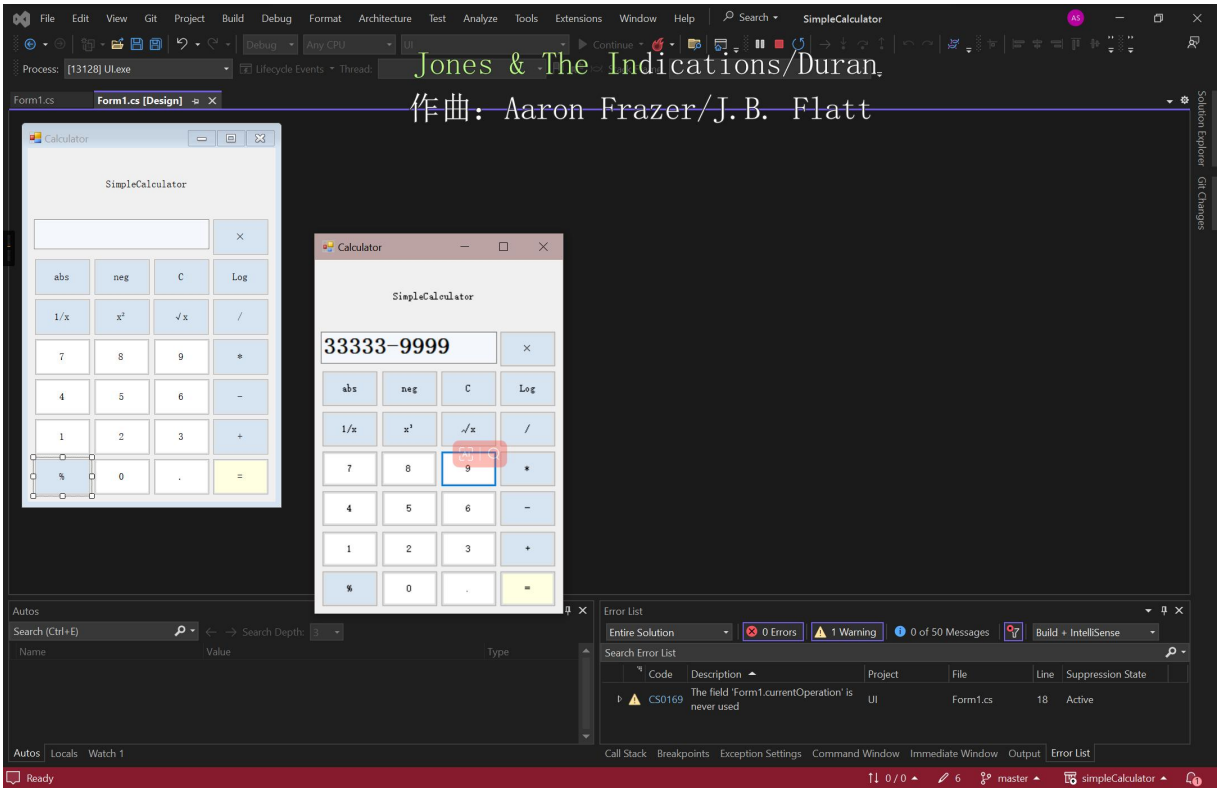

Why is it so complicated
Maybe 'cause that's how God m

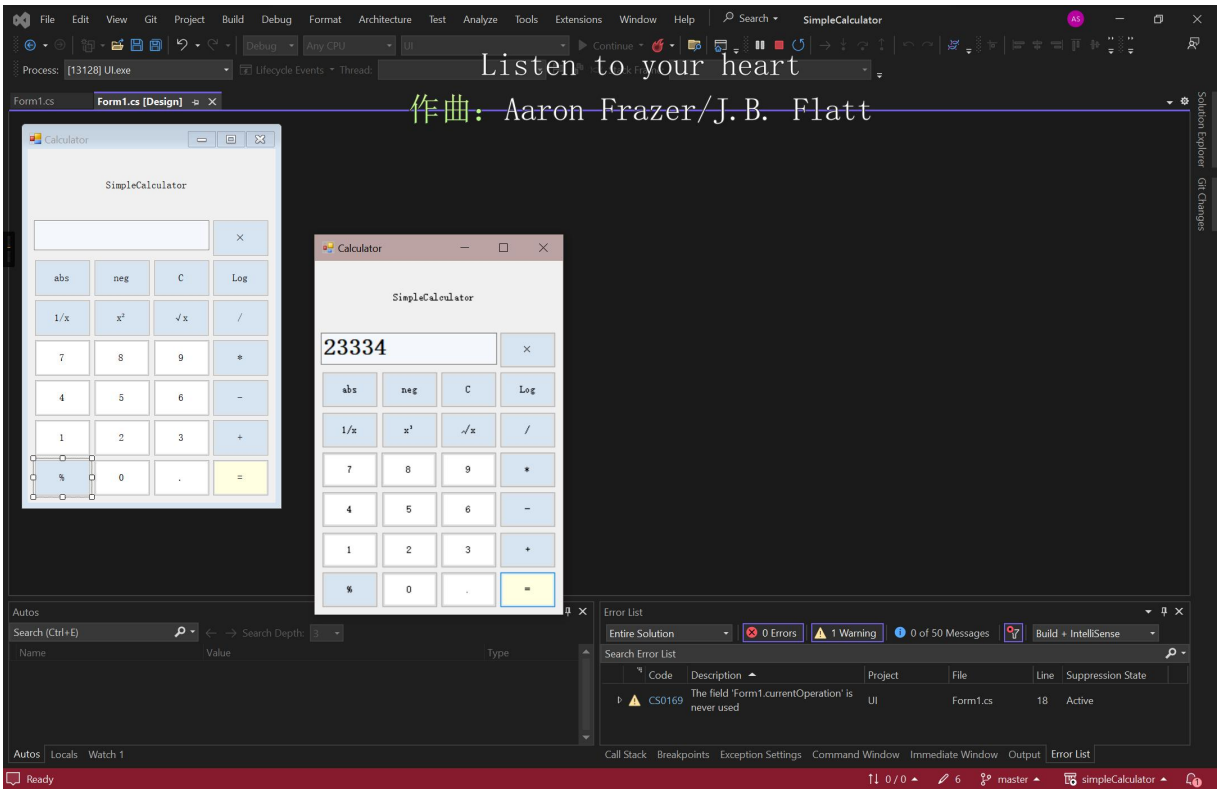
# 四、运行效果

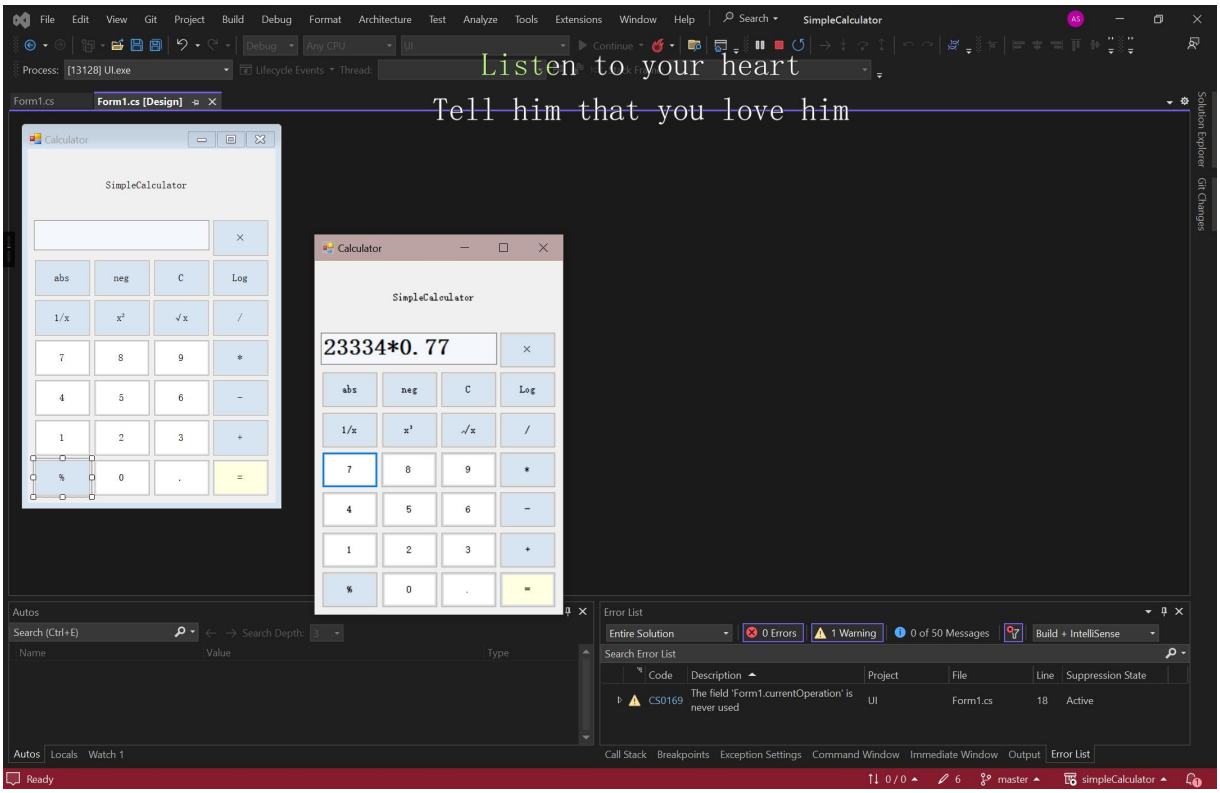

They way it used to be
When love was so easy
SimpleCalculator

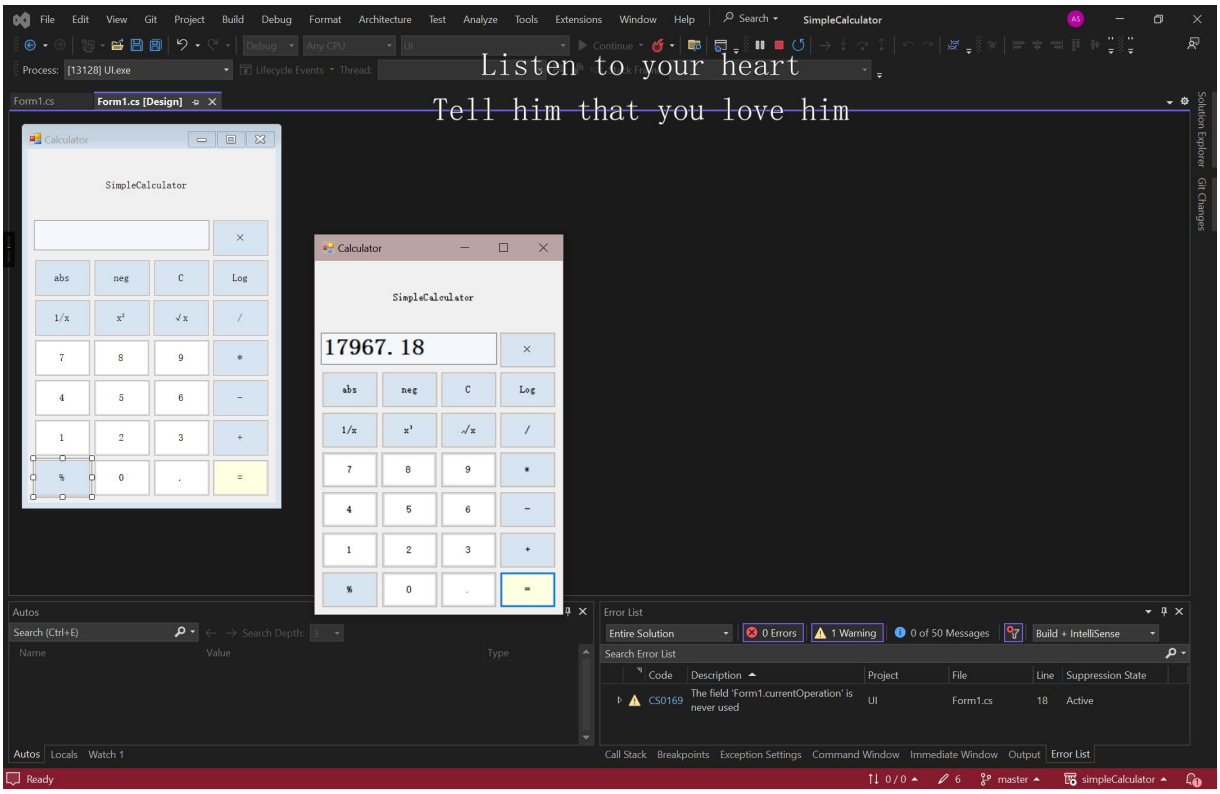

I wish it was that simple, sir
Simple, simple
SimpleCalculator
11111+22222

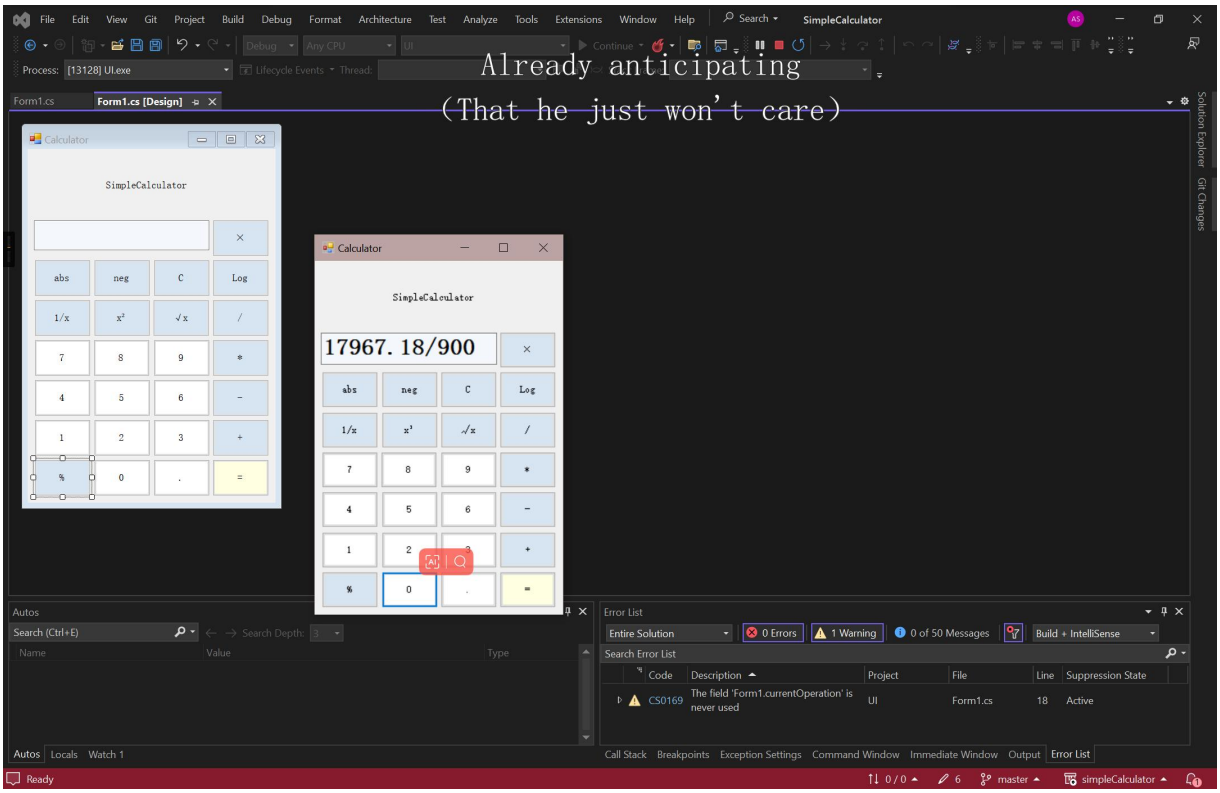

sh it was that simple, simple。
Simple, simple
SimpleCalculator
33333

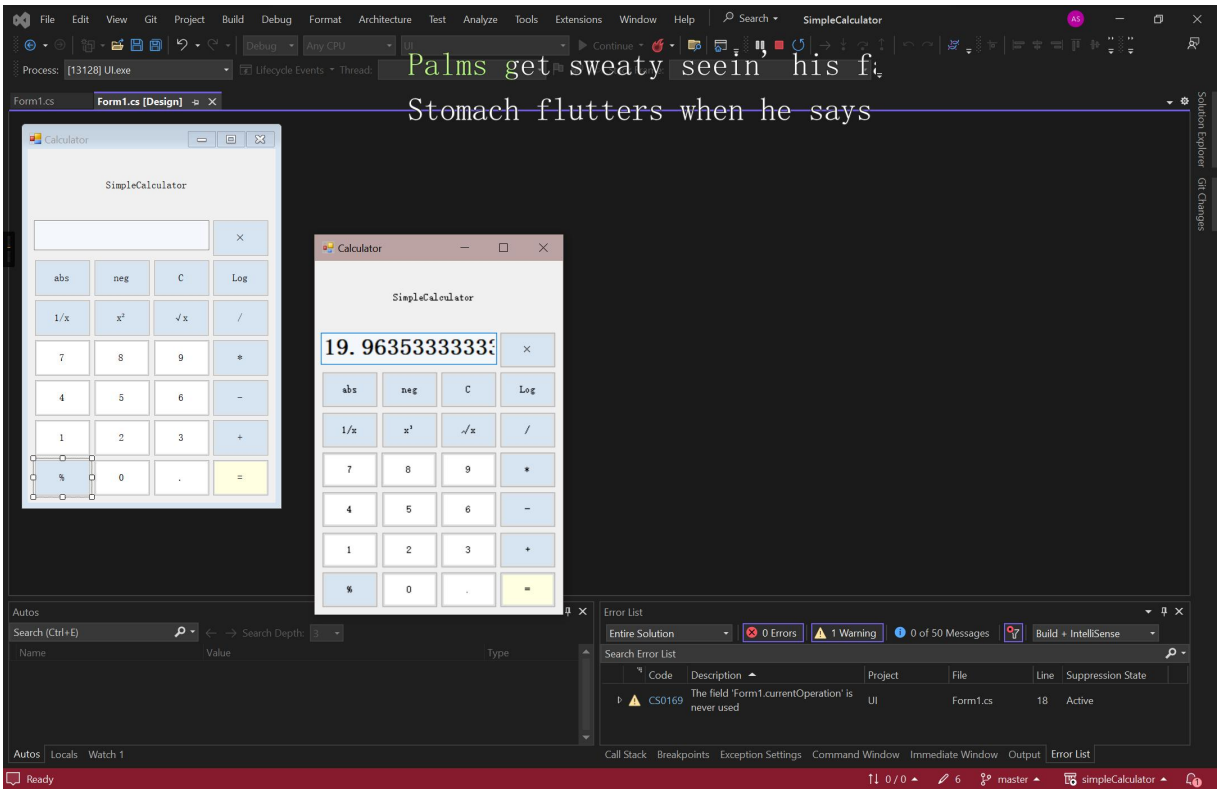

Jones & The Indications/Duran。
作曲：Aaron Frazer/J.B. Flatt
SimpleCalculator
33333-9999

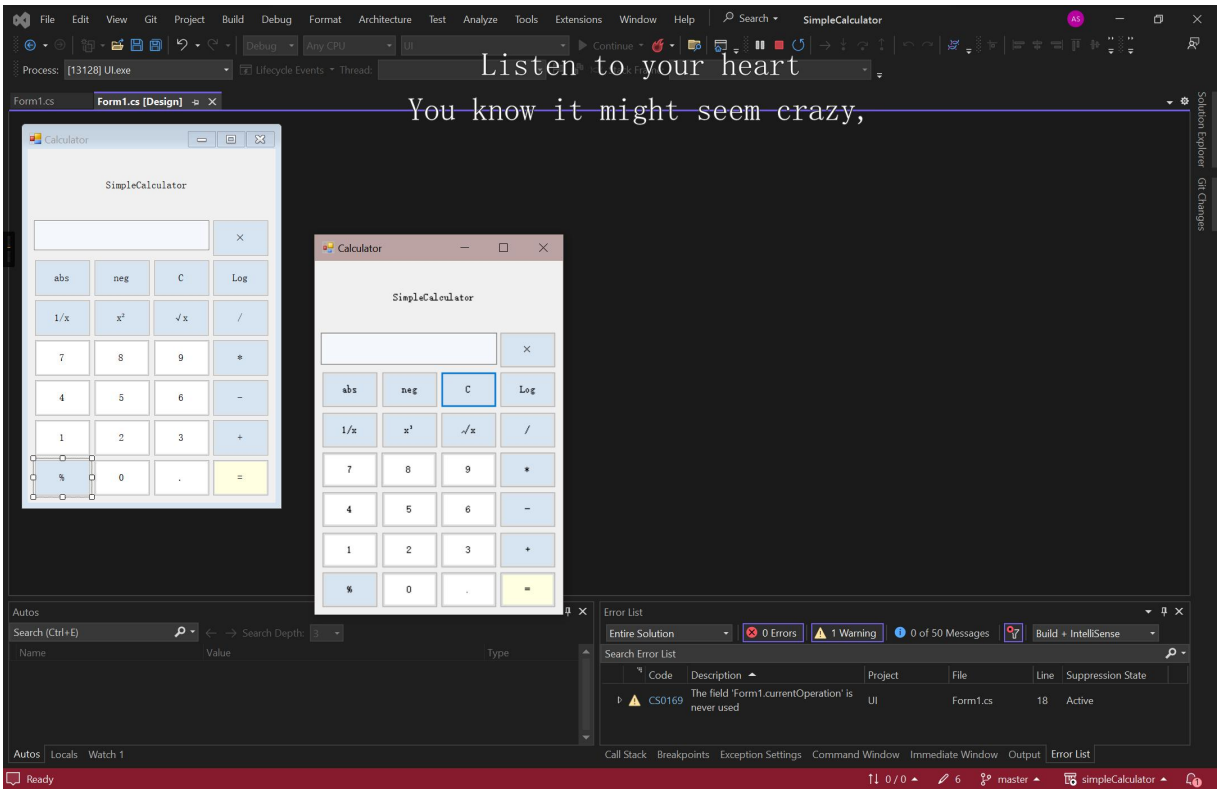

Listen to your heart
作曲：Aaron Frazer/J.B. Flatt
SimpleCalculator
23334

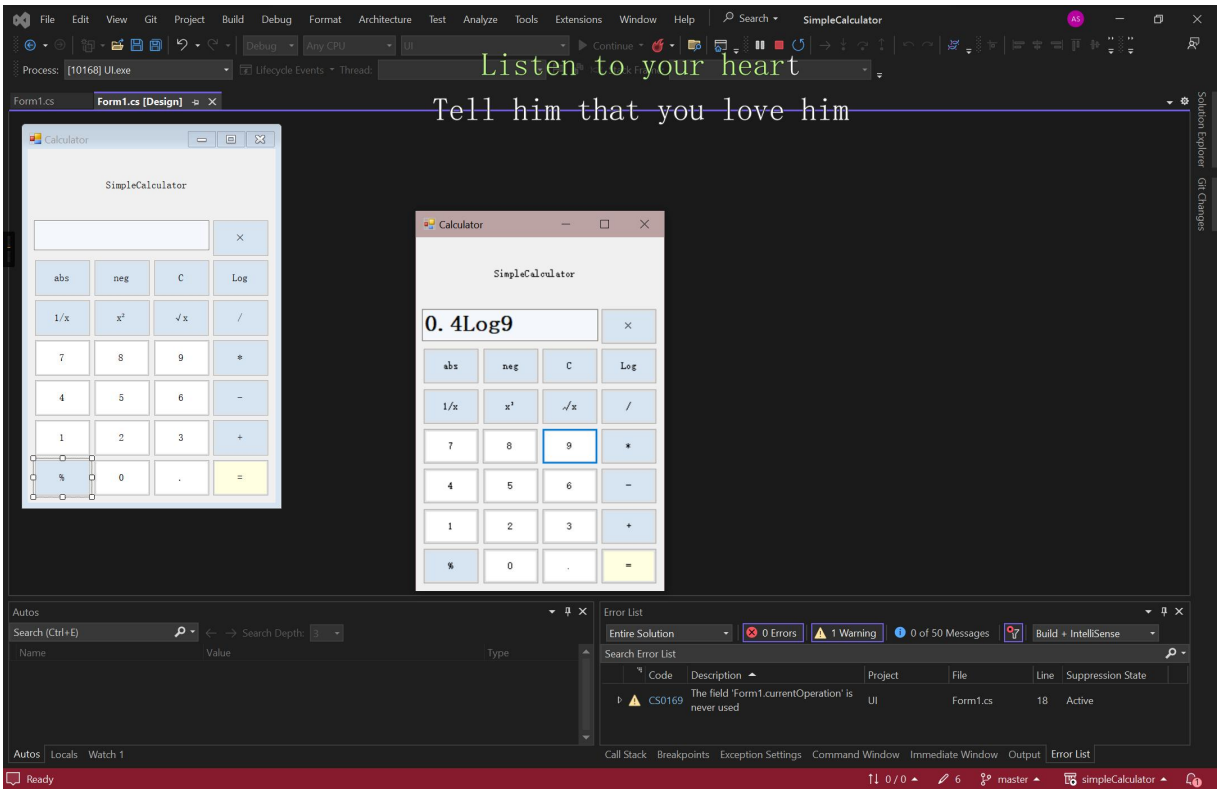

Listen to your heart
Tell him that you love him
SimpleCalculator
23334*0.77

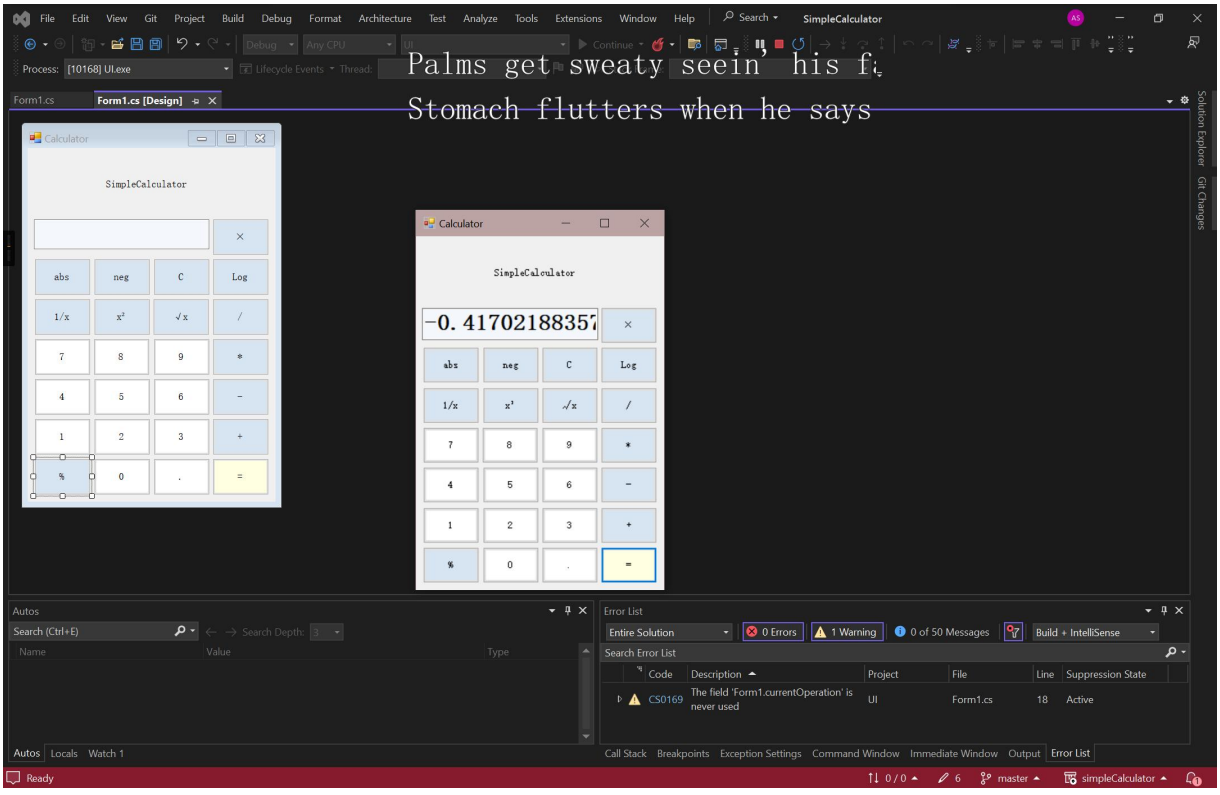

Listen to your heart
Tell him that you love him
SimpleCalculator
17967. 18

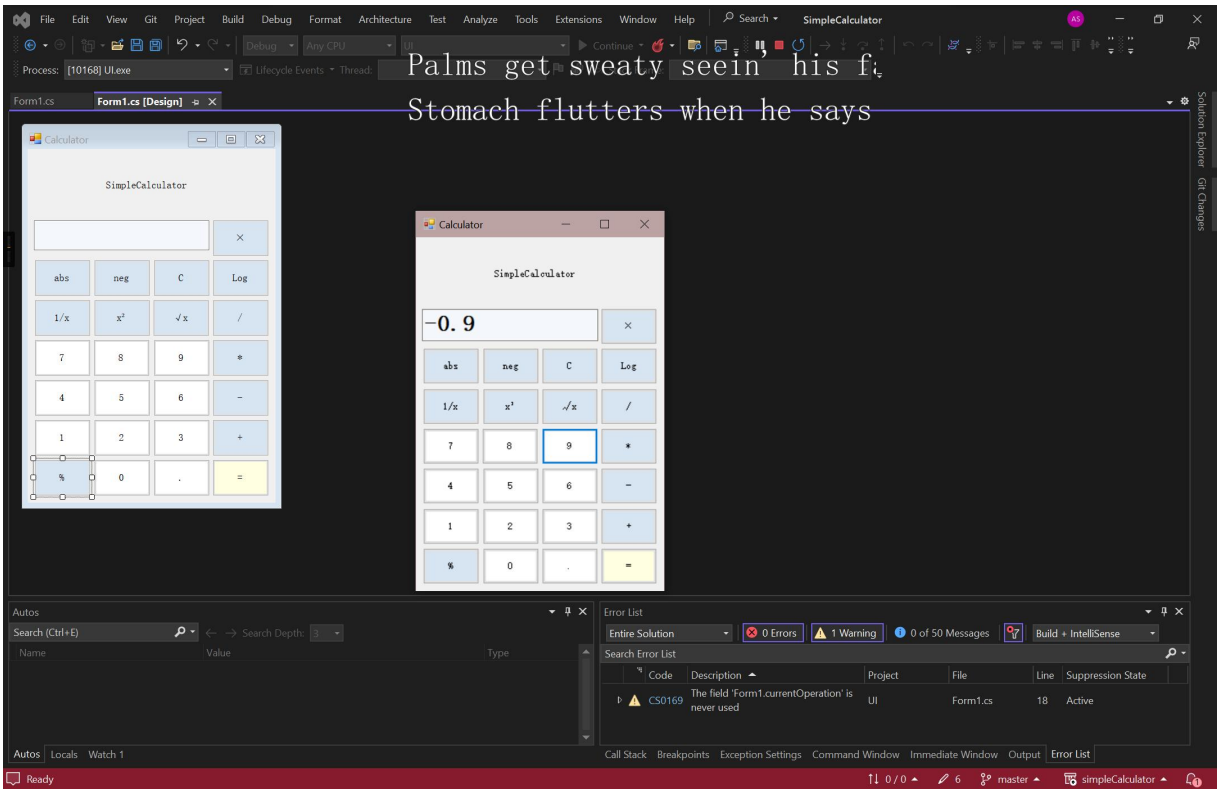

Already anticipating
(That he just won't care)
SimpleCalculator
17967. 18/900

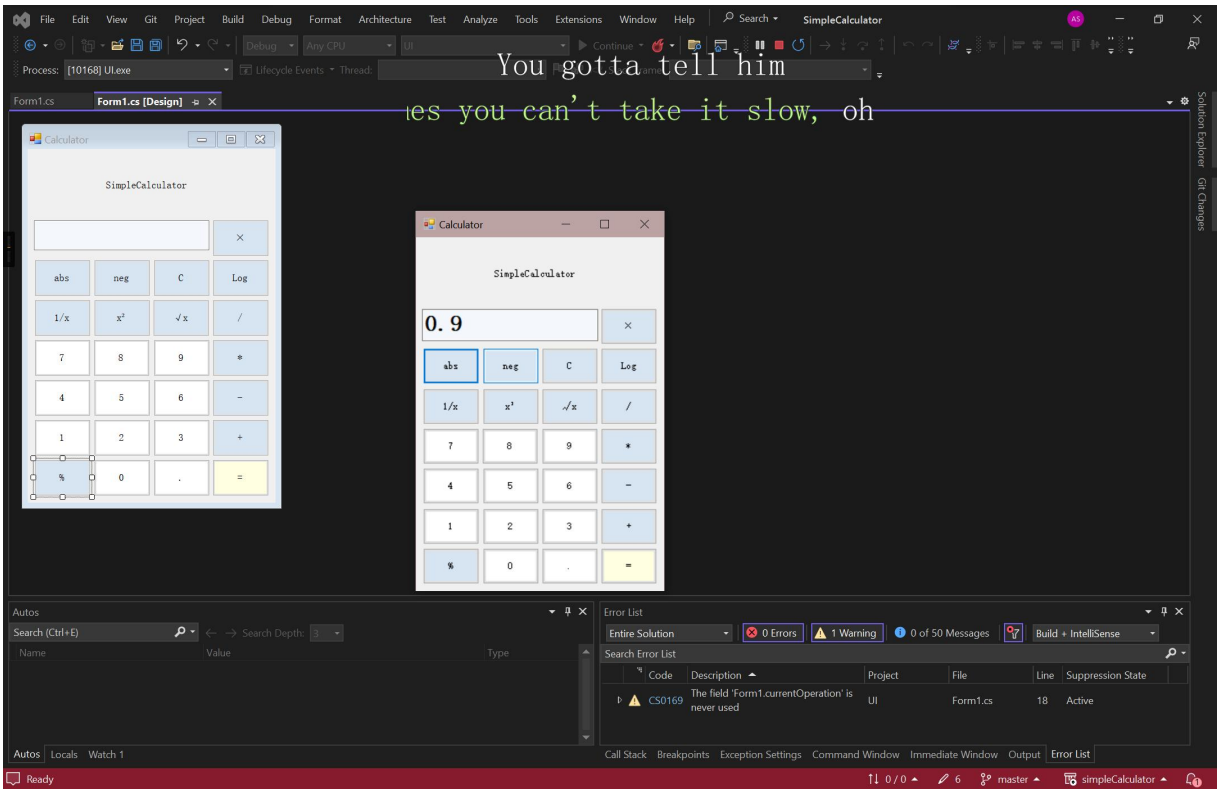

Palms get sweaty seein' his f
Stomach flutters when he says
SimpleCalculator
19. 96353333333
CS0169
The field 'Form1.currentOperation' is never used

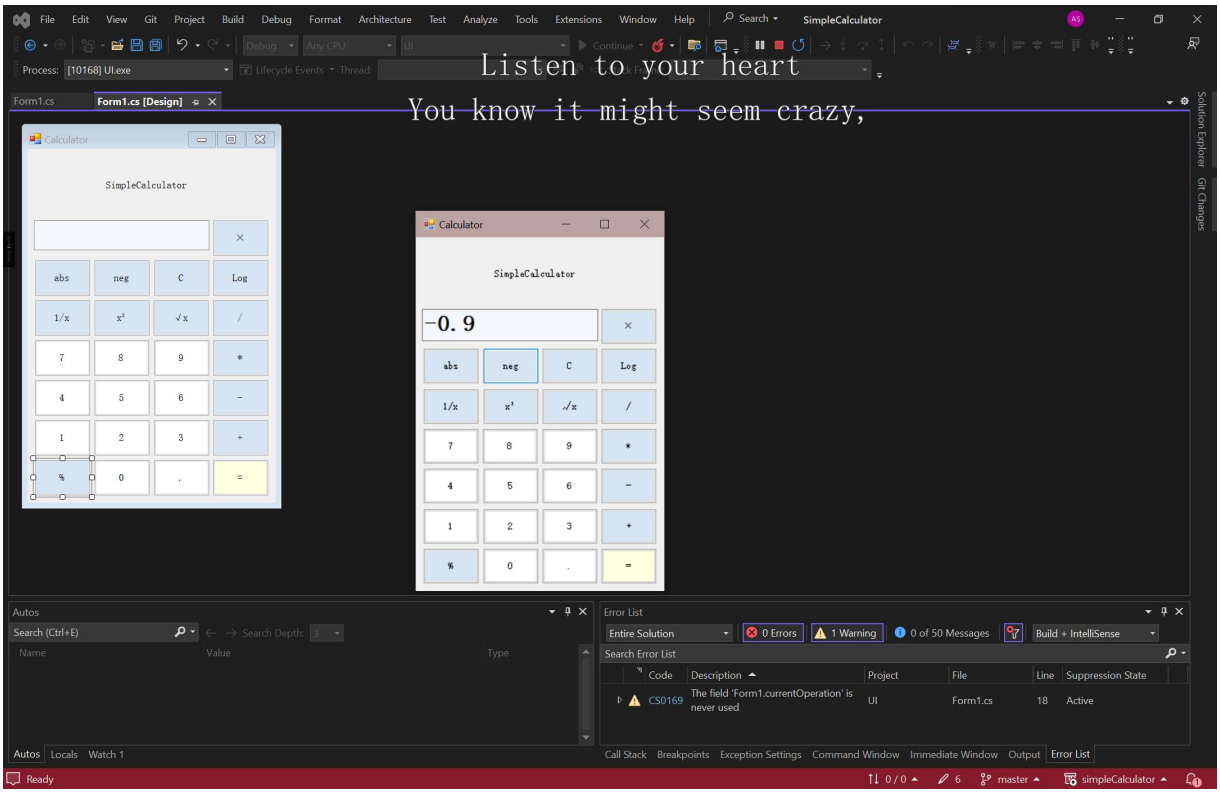

Listen to your heart
You know it might seem crazy,
SimpleCalculator
CS0169
The field 'Form1.currentOperation' is never used

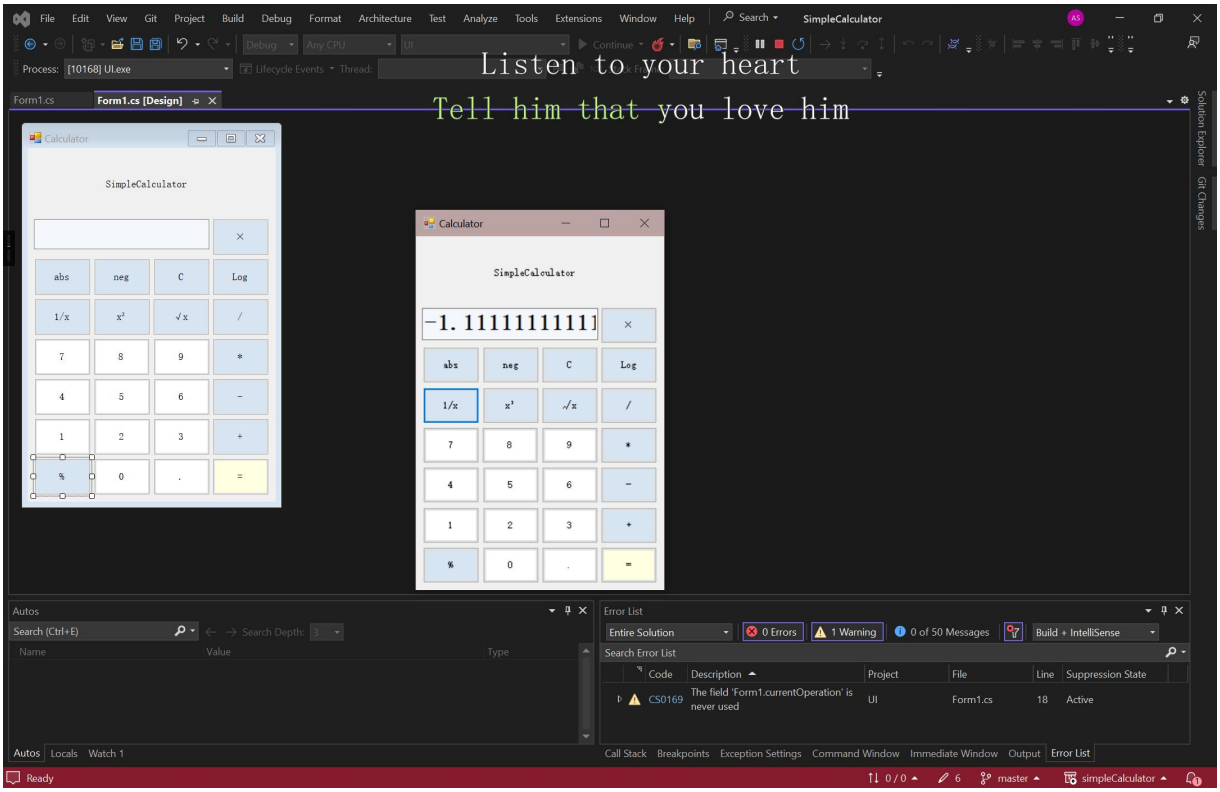

Listen to your heart
Tell him that you love him
SimpleCalculator
0. 4Log9

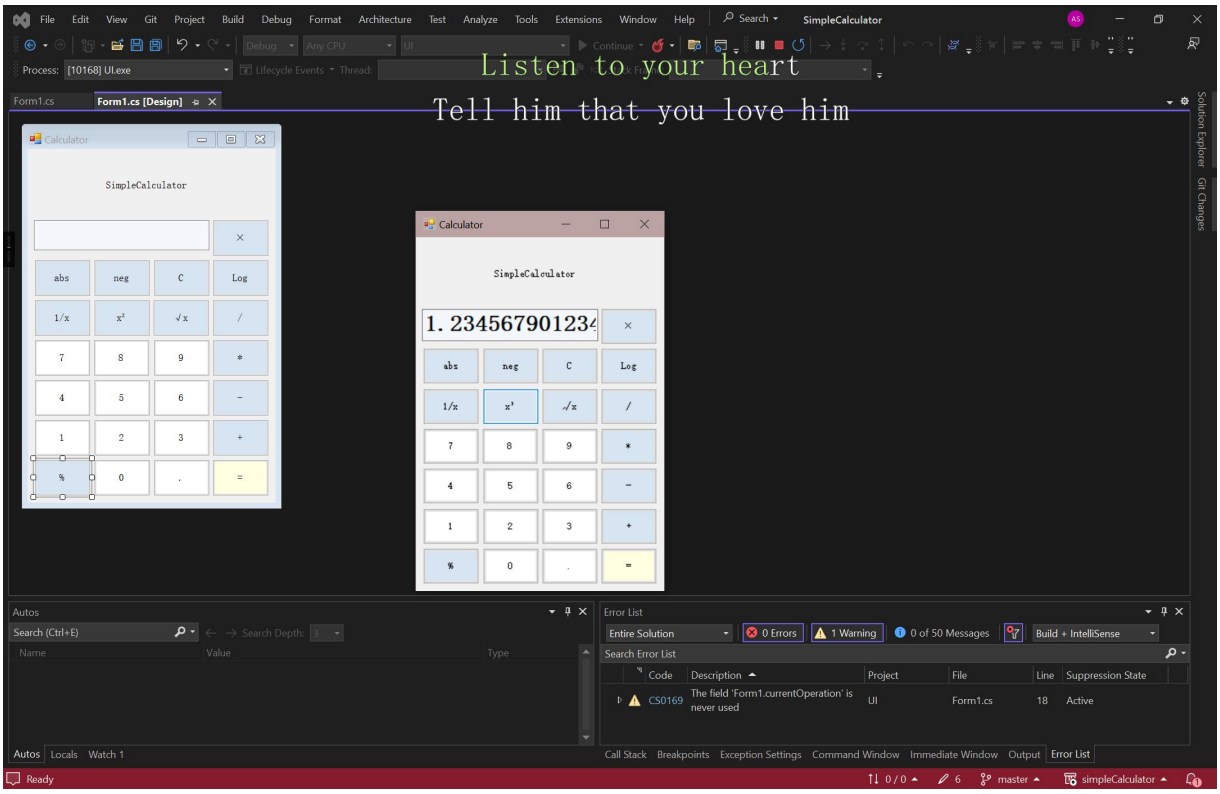

Palms get sweaty seein' his f
Stomach flutters when he says
SimpleCalculator
-0. 41702188357

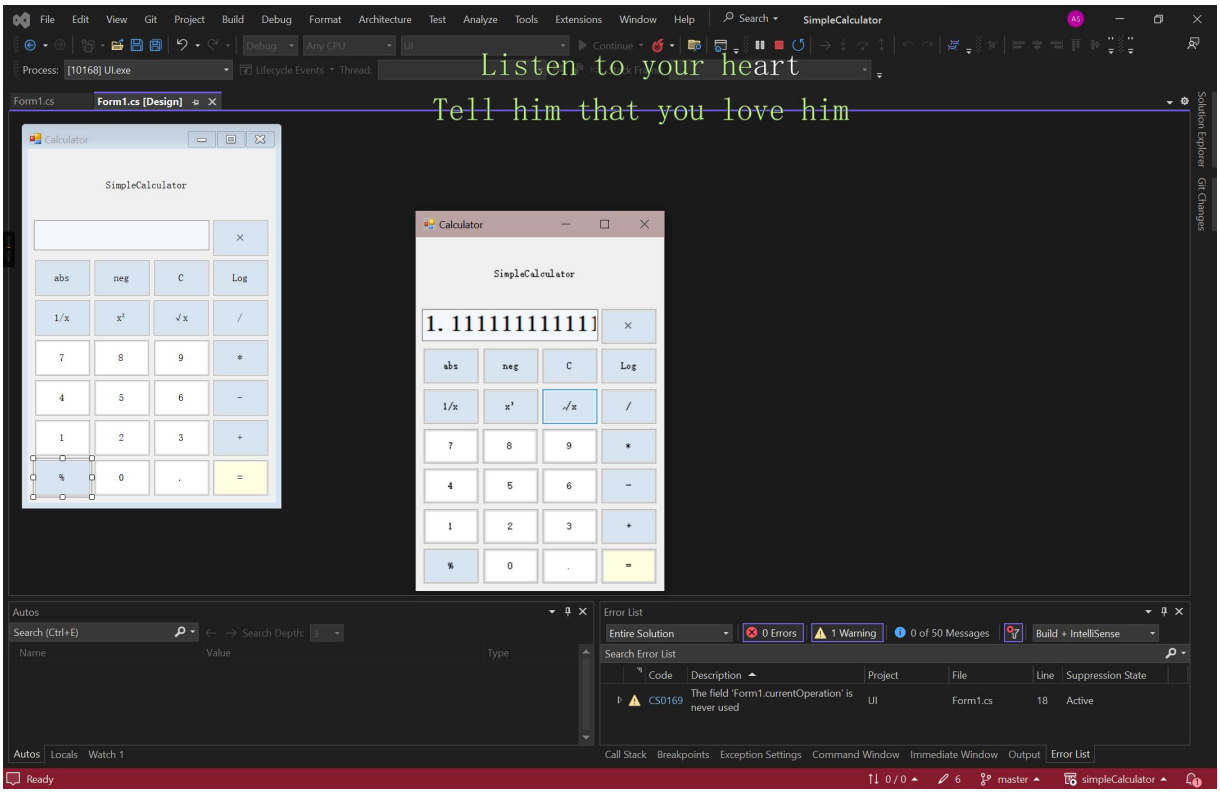

Palms get sweaty seein' his f
Stomach flutters when he says
SimpleCalculator
-0. 9
CS0169
The field 'Form1.currentOperation' is never used

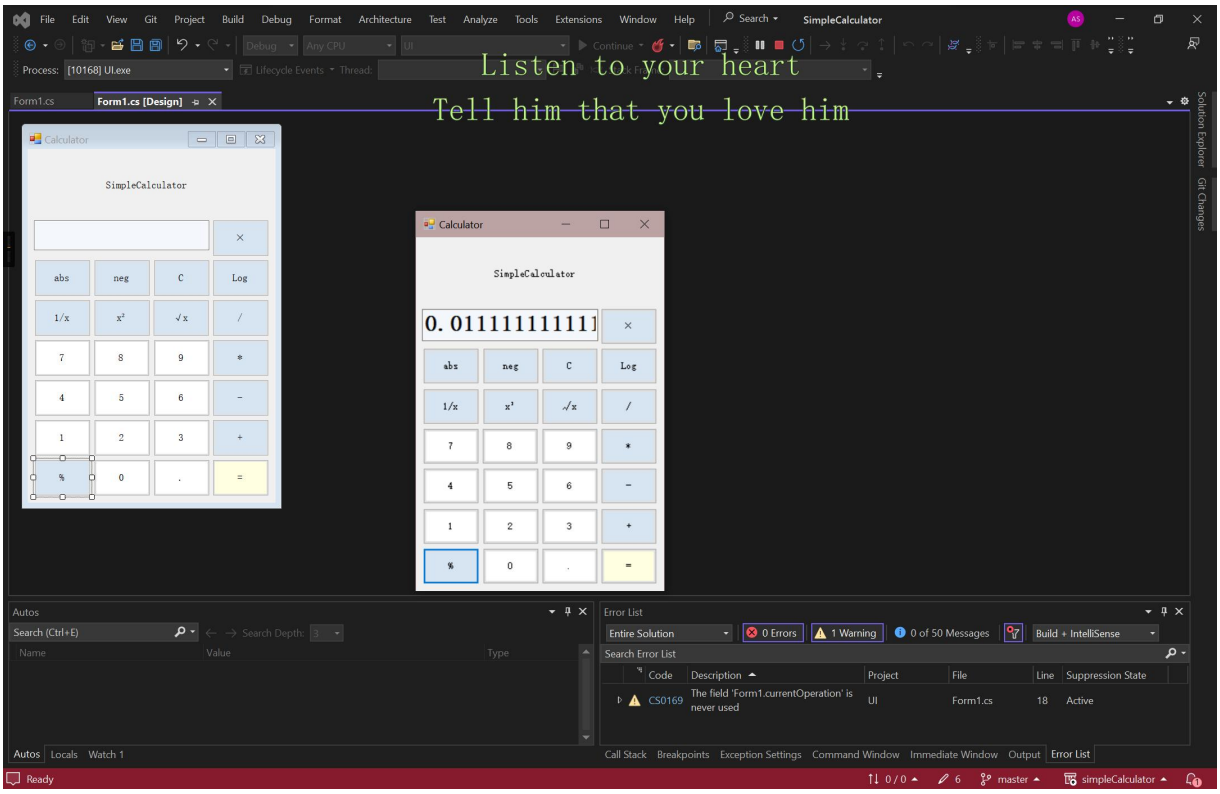

You gotta tell him
es you can't take it slow, oh
SimpleCalculator
0. 9
CS0169
The field 'Form1.currentOperation' is never used

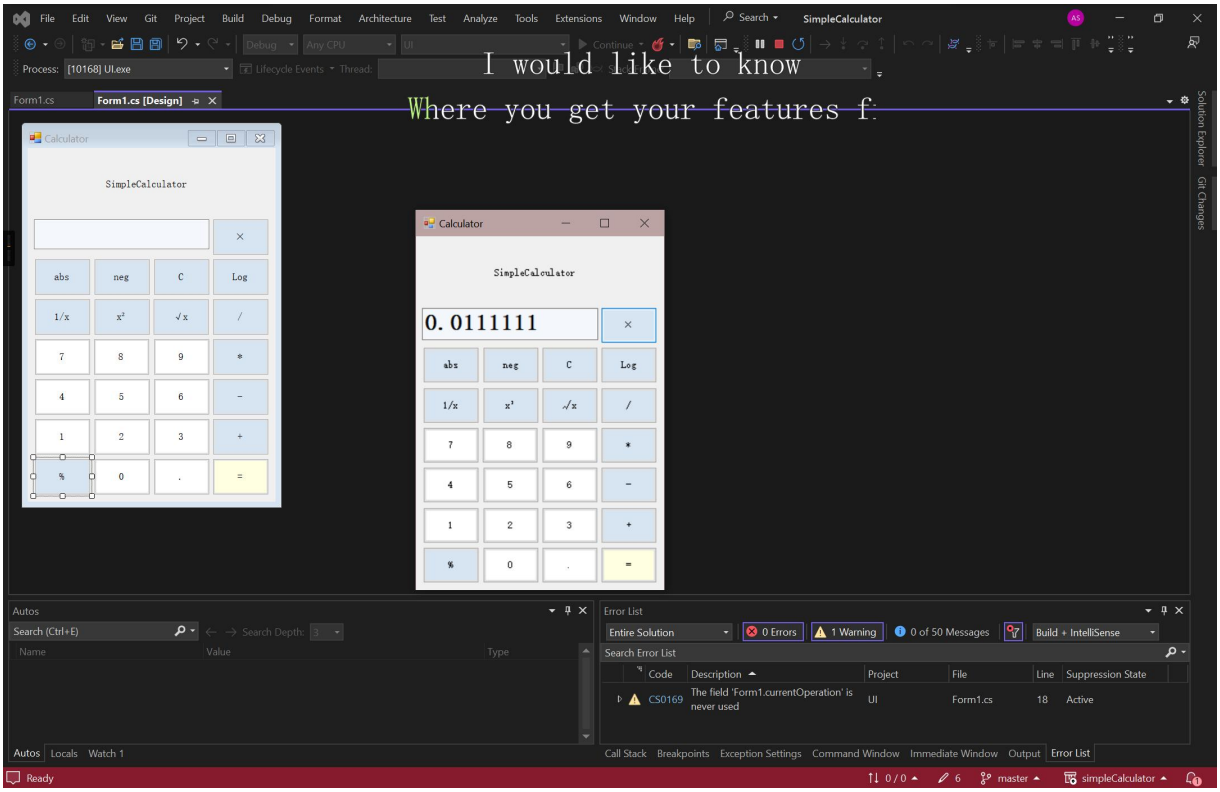

Listen to your heart
You know it might seem crazy,
SimpleCalculator
−0. 9

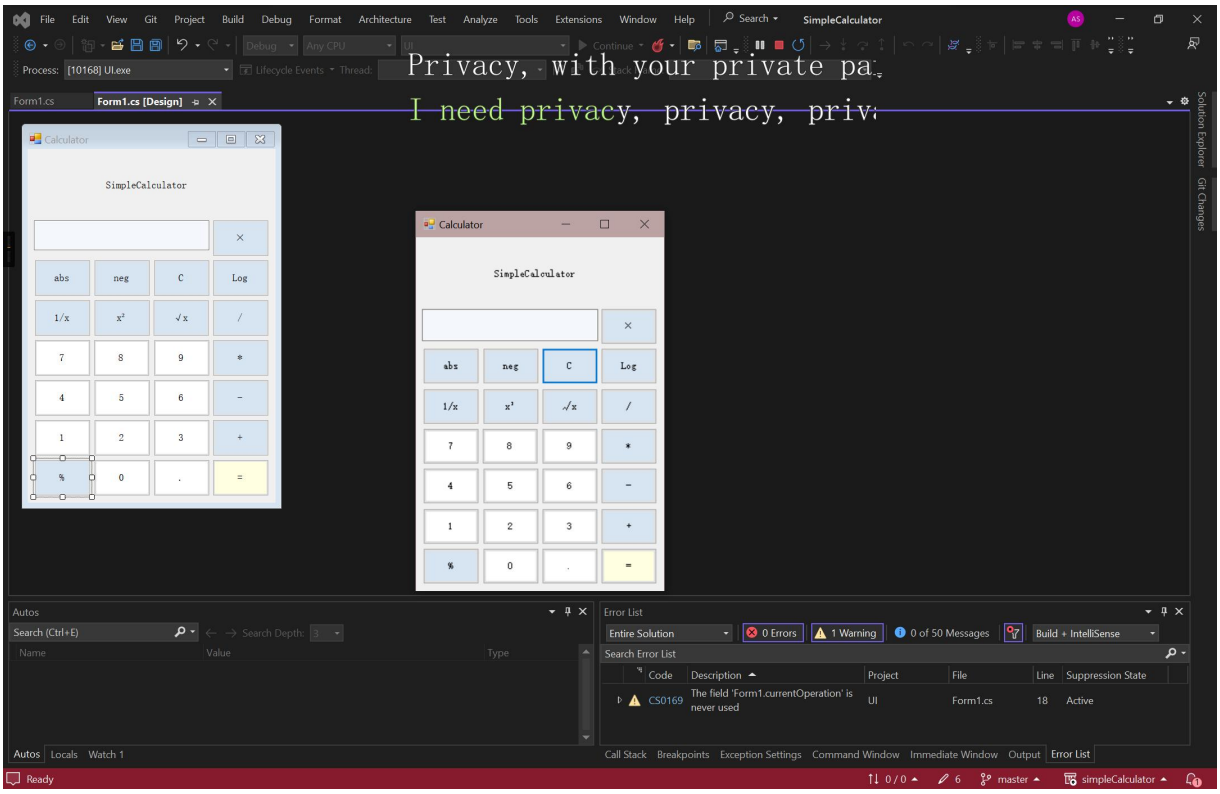

Listen to your heart
Tell him that you love him
SimpleCalculator
−1. 1111111111


Listen to your heart
Tell him that you love him
SimpleCalculator
1. 23456790123
SimpleCalculator


Listen to your heart
Tell him that you love him
SimpleCalculator
1. 11111111111
SimpleCalculator


Listen to your heart
Tell him that you love him
SimpleCalculator
0. 011111111111


I would like to know
Where you get your features f:
SimpleCalculator
0. 0111111




## 五、实验心得

在这次实验中，我通过编写一个简单的计算器程序来学习了 C#编程语言和 Windows 窗体应用程序开发。我学习了如何使用接口和面向对象编程概念来设计程序，并使用正则表达式和异常处理来实现程序的一些核心功能。此外，我还学习了一些基本的 Windows 窗体控件，并使用它们来构建您的计算器用户界面。

在实验的过程中，我遇到了一些问题，例如如何使用正则表达式匹配字符串中的操作符、如何在用户输入不合法时进行异常处理等等。通过解决这些问题，我对 C#编程语言和 Windows 窗体应用程序开发的理解和掌握程度得到了提高。

总的来说，这次实验为我提供了一个良好的学习机会，让我更深入地了解了 C#编程语言和 Windows 窗体应用程序开发。