目　录

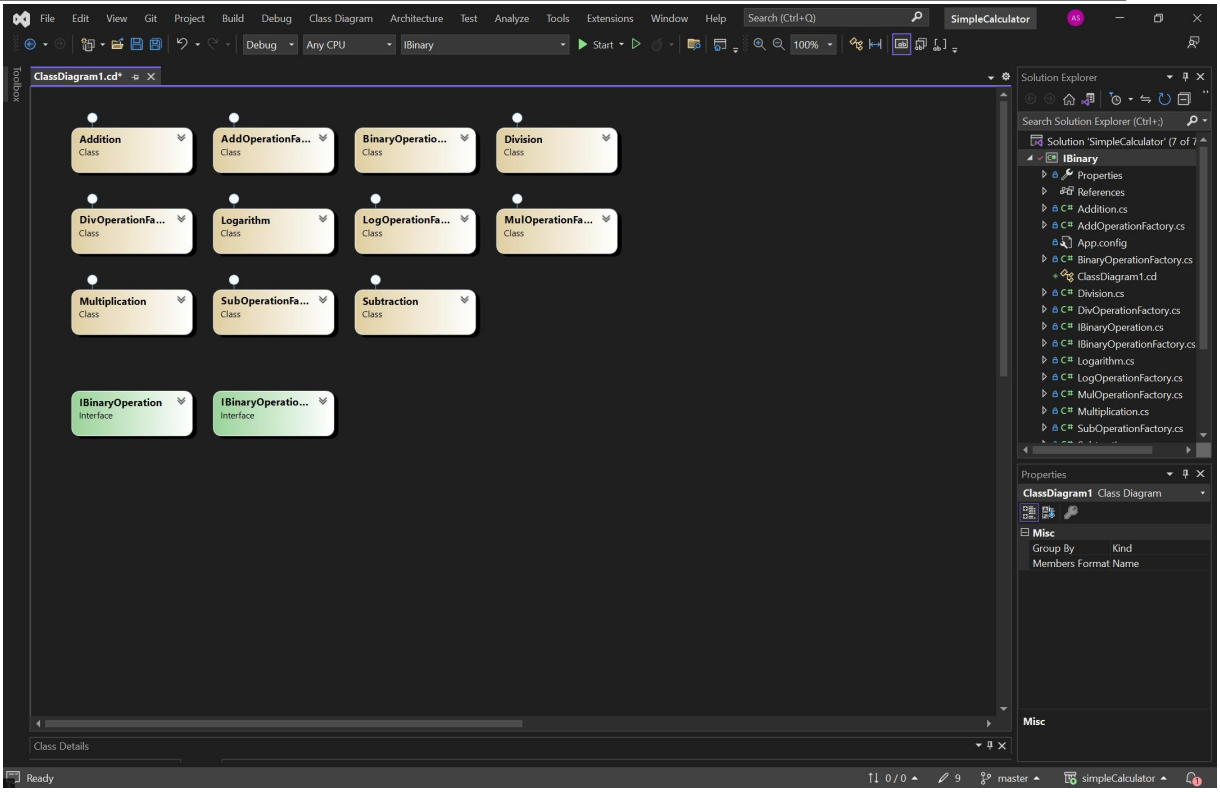
一、实验内容

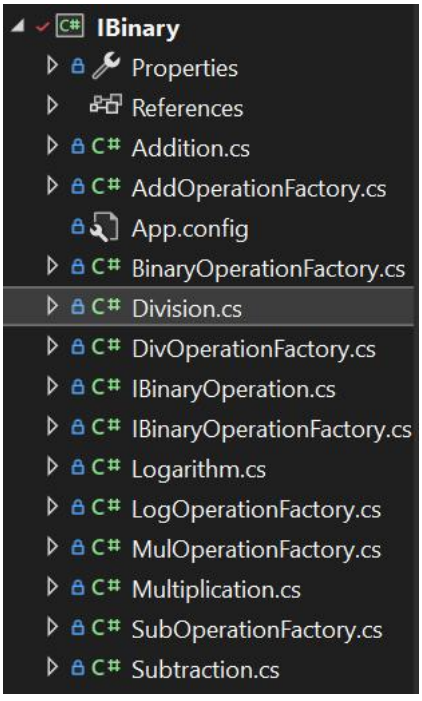
基本环境包括 IDE 开发工具 Visual Studio、数据库 Sql Server Express LocalDb、单元测试框架 MsTest、文档 Word，Orm 数据访问框架 Simple.Data 和 EF 框架，IOC 框架 Autofac 等。

序号	内容	具体要求	考核比例
1	计算器的领域模型 1	采用**工厂方法模式**实现计算器**二元计算领域模型**，实现加、减、乘、除、对数 $\log_a x$、指数 x^y 等二元运算	10
2	对于领域模型 1 的单元测试	对领域模型 1 进行单元测试	5
3	计算器的领域模式 2	采用工厂方法模式实现计算器**一元计算领域模型**，实现取反、求 1/x、$2\sqrt{\ }$、x^2、10^x、$\lvert x\rvert$、log、ln、exp 等一元运算	10
4	对于领域模型 2 的单元测试	对领域模型 2 进行单元测试	5
5	计算器的 UI 界面	使用**配置文件**与**反射机制**实现二元计算工厂对象和一元计算工厂对象的实例化	10
6	实现日志读写模块	将操作日志**写入文本文档、使用 ADO.NET 写入数据库、使用 Simple.Data 和 EF 框架将计算器操作日志写入数据库**	10
7	数据库日志读写模块单元测试	对于日志读写的多种不同的实现，对其进行单元测试	10
8	在 UI 项目中增加日志读写功能	在 UI 项目中使用日志读写模块实现日志读写功能，使用 IOC（Autofac）实现日志读写对象的实例化	10
9	MEF 实现日志读写功能	重构 UI 项目和日志读写模块项目，**使用 MEF 实现日志读写功能（类似实验 2 的第 2 部分）**	15
10	实现计算器主题设置和切换	采用**抽象工厂模式**设计计算器主题功能，定义至少两种主题，每种主题下，计算器的特性是不同的，比如背景图，字体颜色，字体等不相同。	15

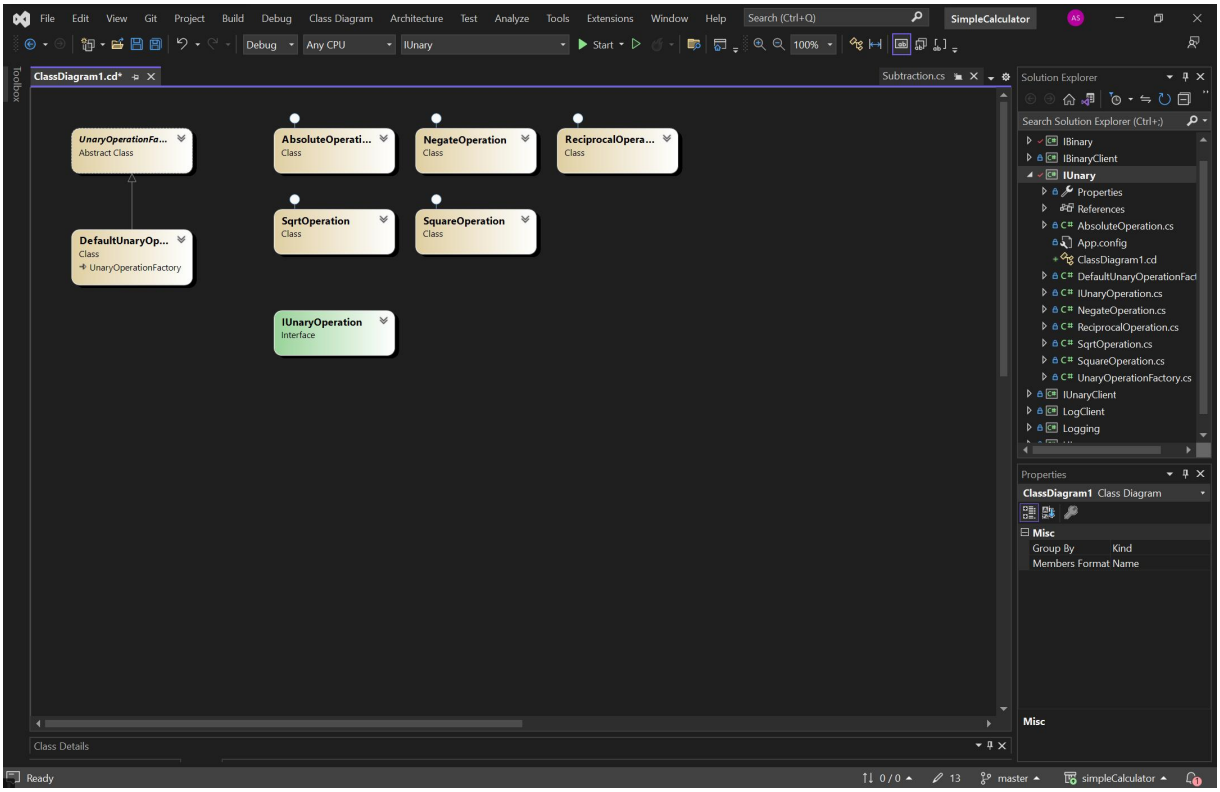
二、类图及代码结构

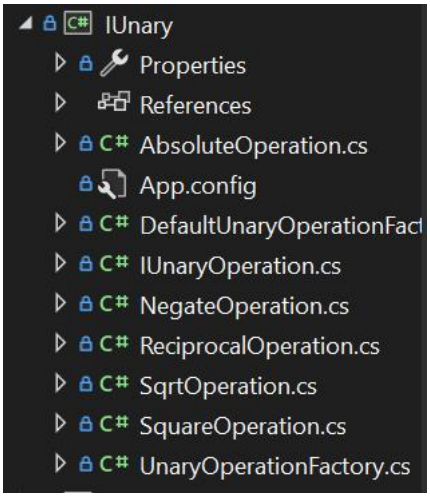
1	计算器的领域模型 1	采用**工厂方法模式**实现计算器**二元计算领域模型**，实现加、减、乘、除、对数 $\log_a x$、指数 x^y 等二元运算



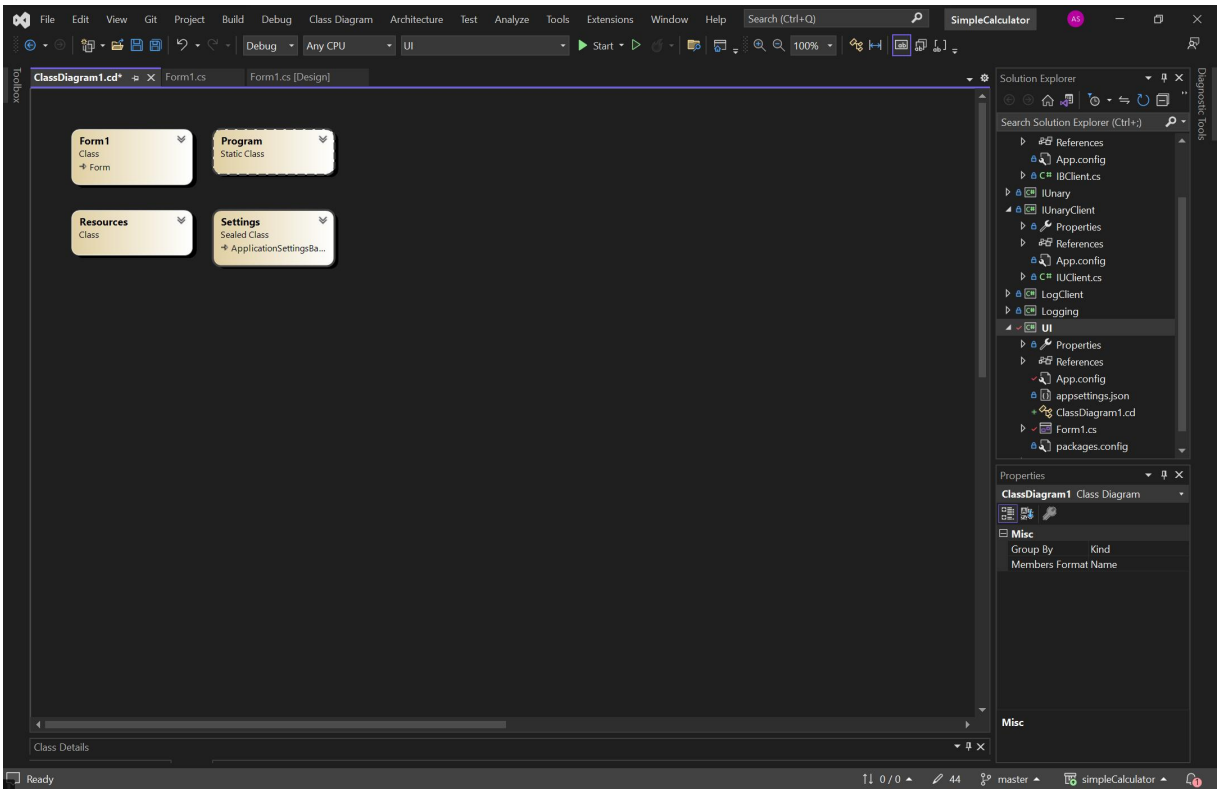


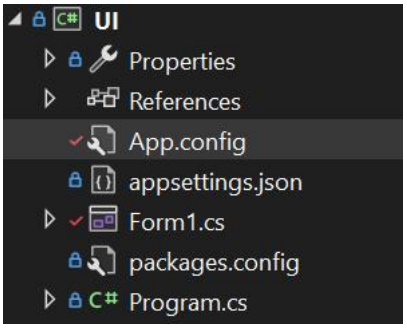
3	计算器的领域模式 2	采用工厂方法模式实现计算器**一元计算领域模型**，实现取反、求 1/x、$2\sqrt{\ }$、x^2、10^x、|x|、log、ln、exp 等一元运算



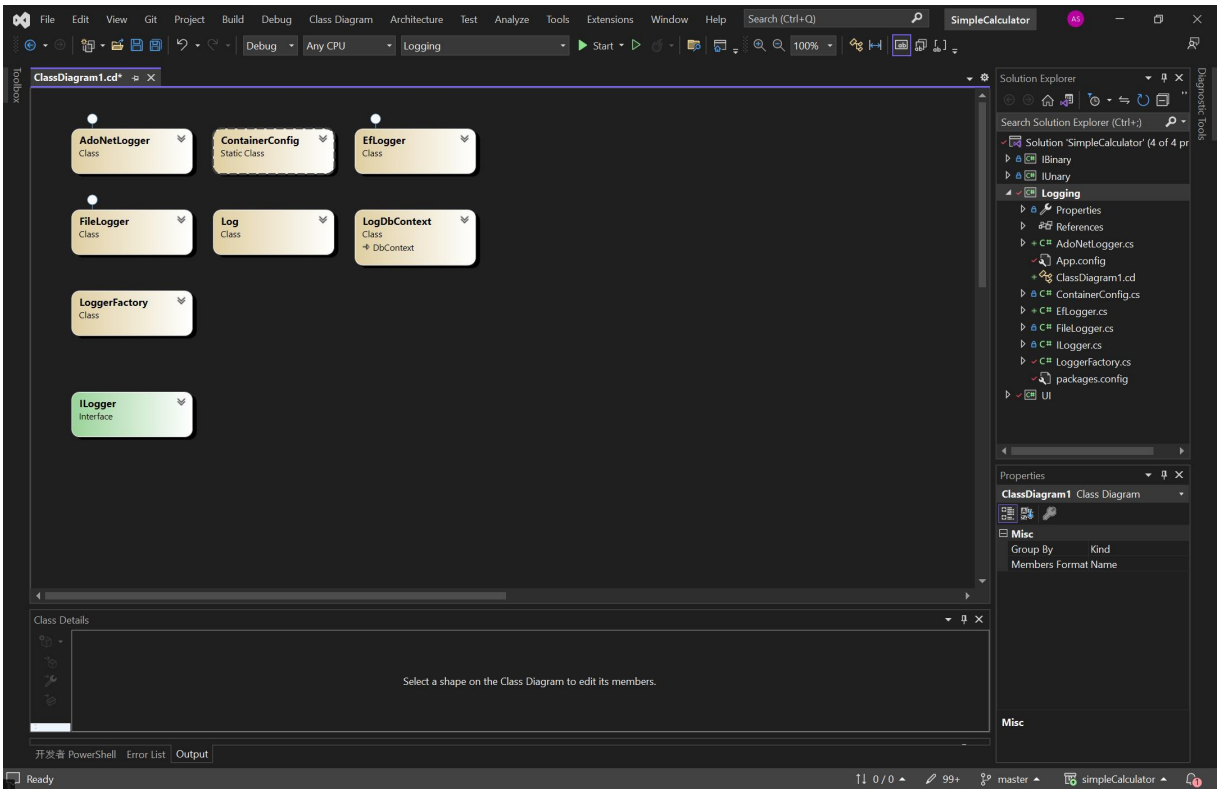


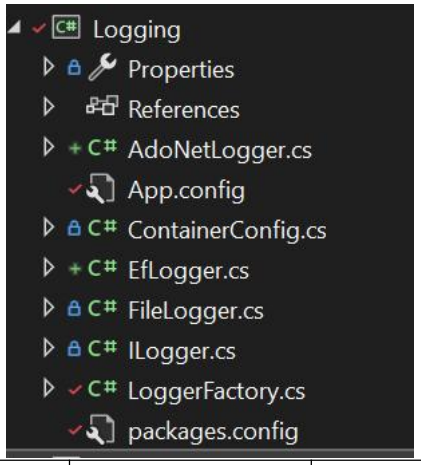
5	计算器的 UI 界面	使用**配置文件**与**反射机制**实现二元计算工厂对象和一元计算工厂对象的实例化



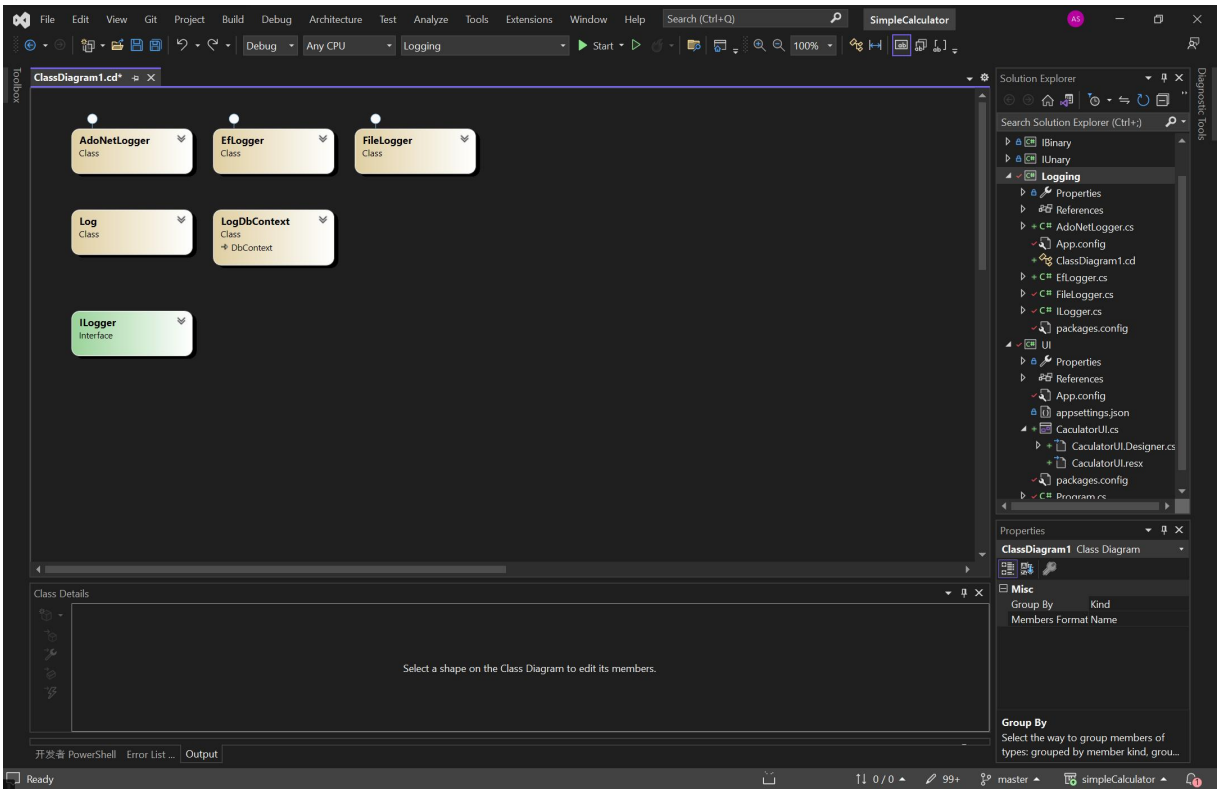


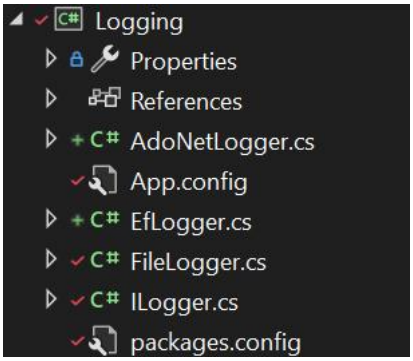
6	实现日志读写模块	将操作日志**写入文本文档、使用 ADO.NET 写入数据库、使用 Simple.Data 和 EF 框架将计算器操作日志写入数据库**
8	在 UI 项目中增加日志读写功能	在 UI 项目中使用日志读写模块实现日志读写功能，使用 IOC（Autofac）实现日志读写对象的实例化



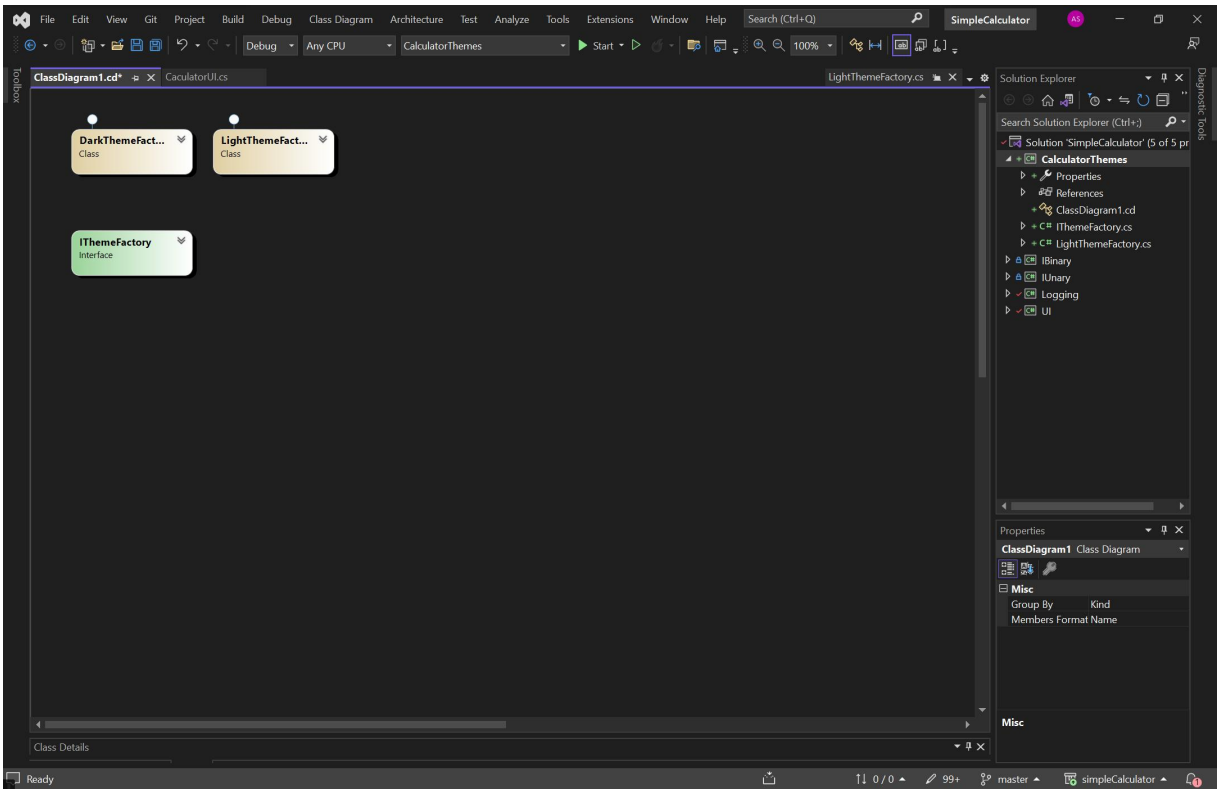


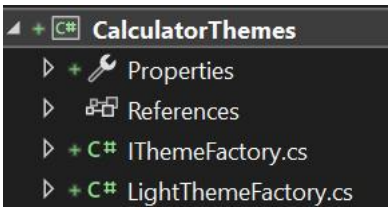
9	MEF 实现日志读写功能	重构 UI 项目和日志读写模块项目，**使用 MEF 实现日志读写功能（类似实验 2 的第 2 部分）**





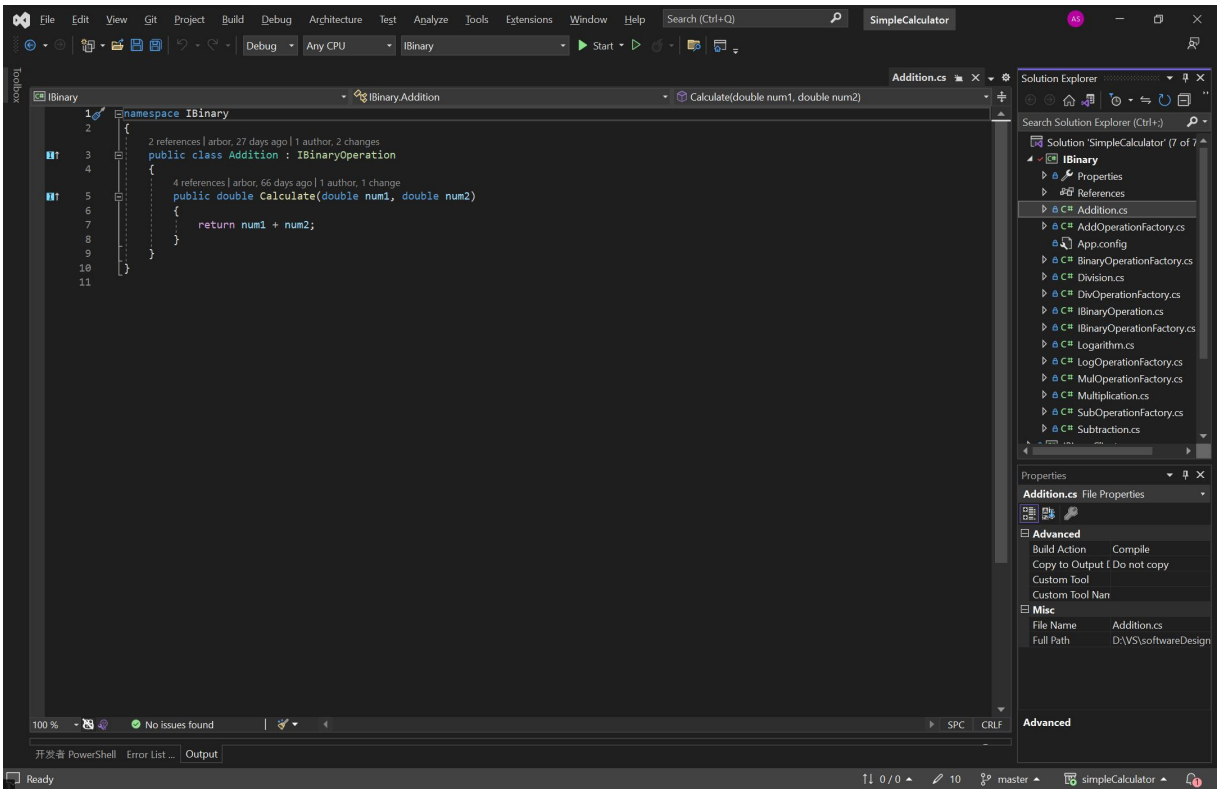
10	实现计算器主题设置和切换	采用**抽象工厂模式**设计计算器主题功能，定义至少两种主题，每种主题下，计算器的特性是不同的，比如背景图，字体颜色，字体等不相同。

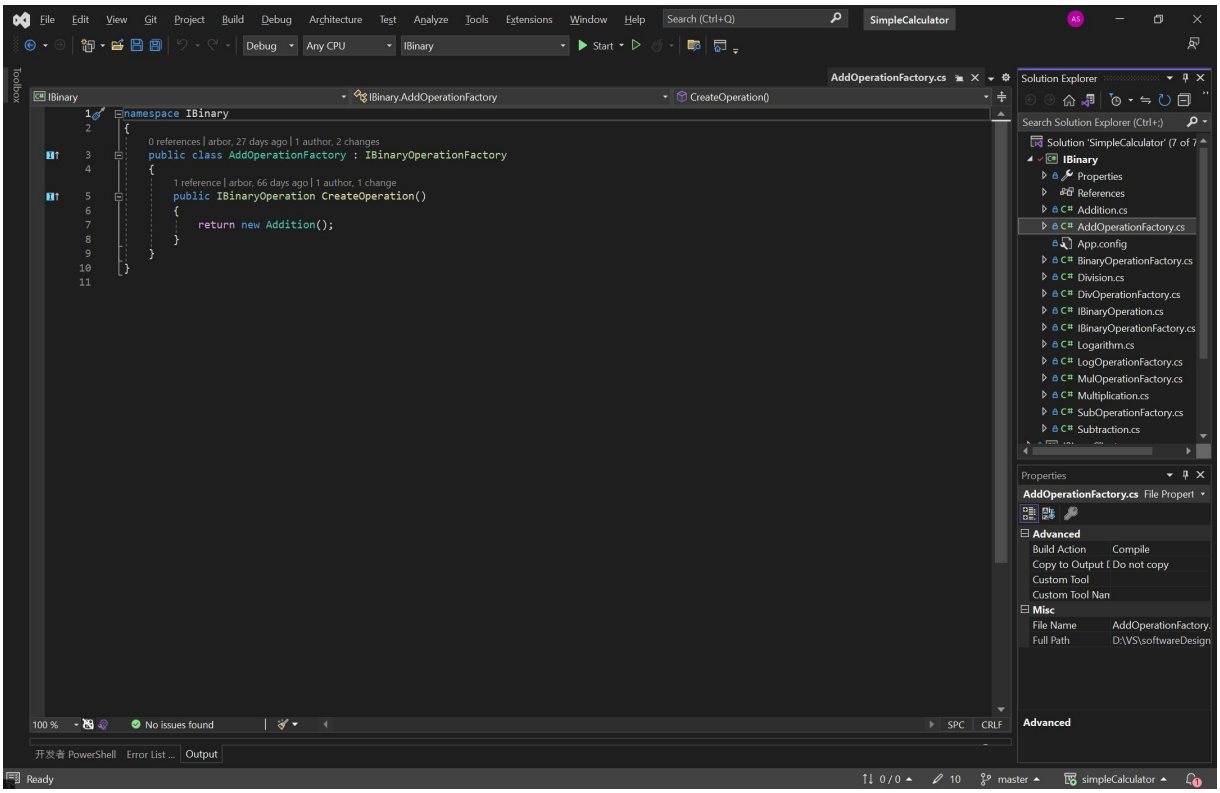


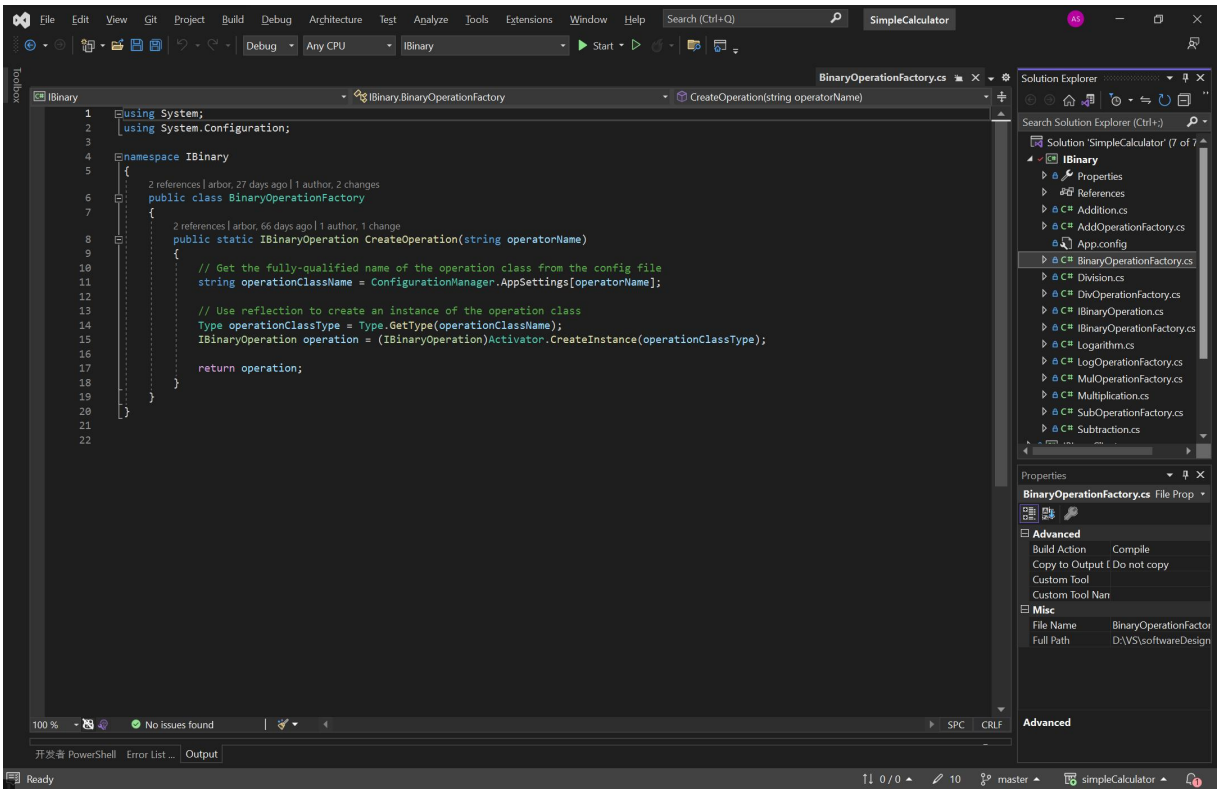


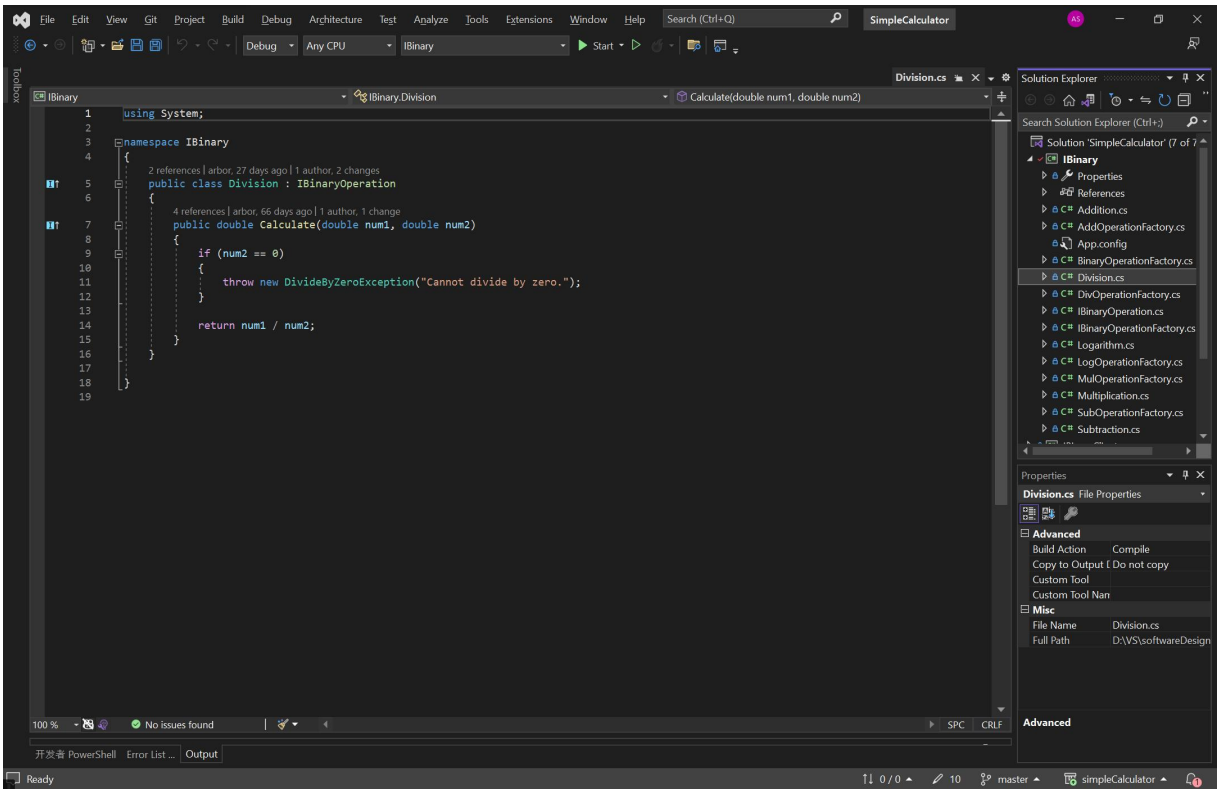
三、代码实现

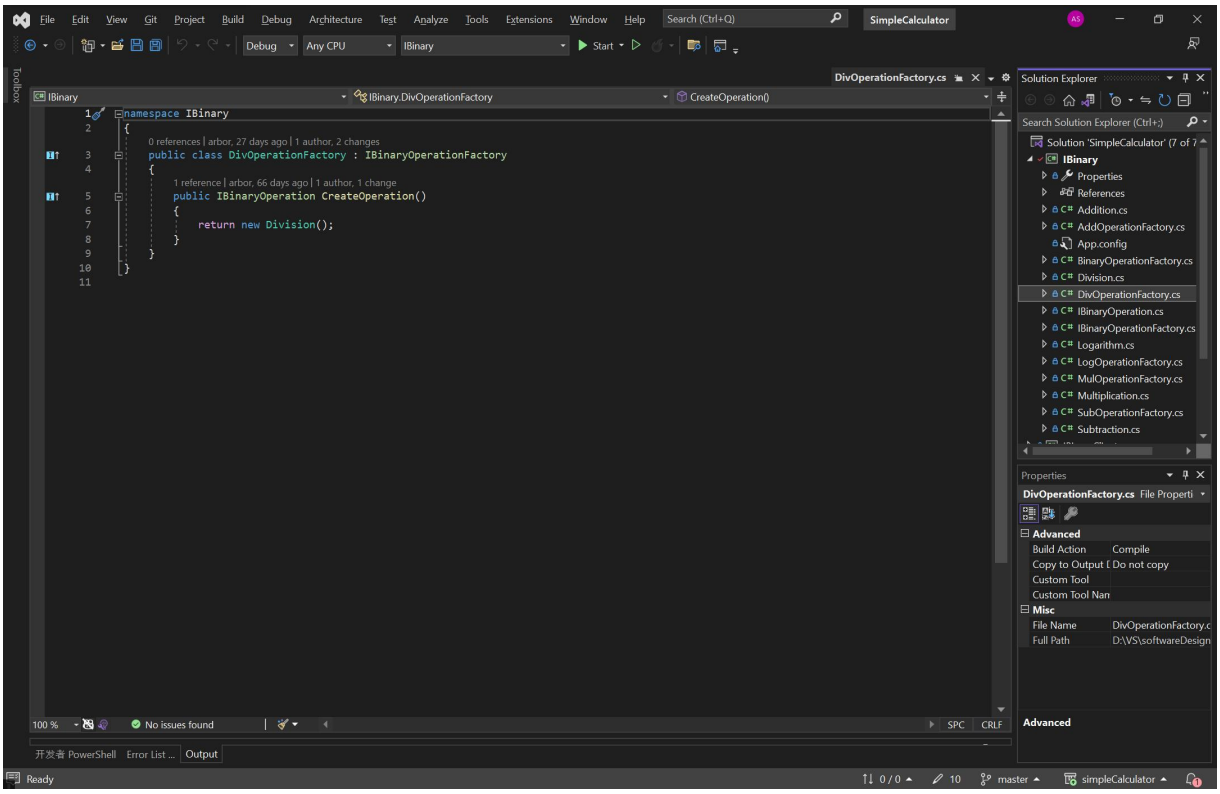
1	计算器的领域模型 1	采用**工厂方法模式**实现计算器**二元计算领域模型**，实现加、减、乘、除、对数 $\log_a x$、指数 x^y 等二元运算

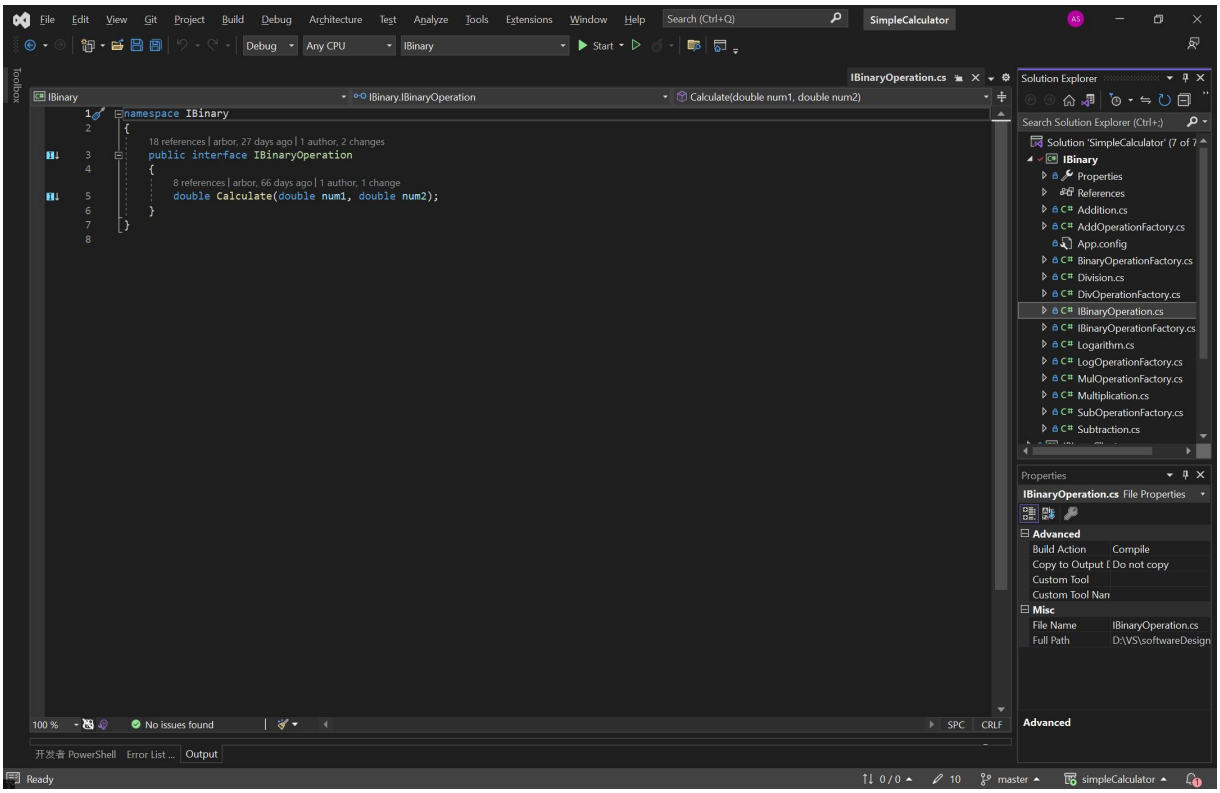

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
Addition.cs
IBinary
IBinary.Addition
Calculate(double num1, double num2)
namespace IBinary
{
2 references | arbor, 27 days ago | 1 author, 2 changes
public class Addition : IBinaryOperation
{
4 references | arbor, 66 days ago | 1 author, 1 change
public double Calculate(double num1, double num2)
{
return num1 + num2;
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
Addition.cs File Properties
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nam
Misc
File Name Addition.cs
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator

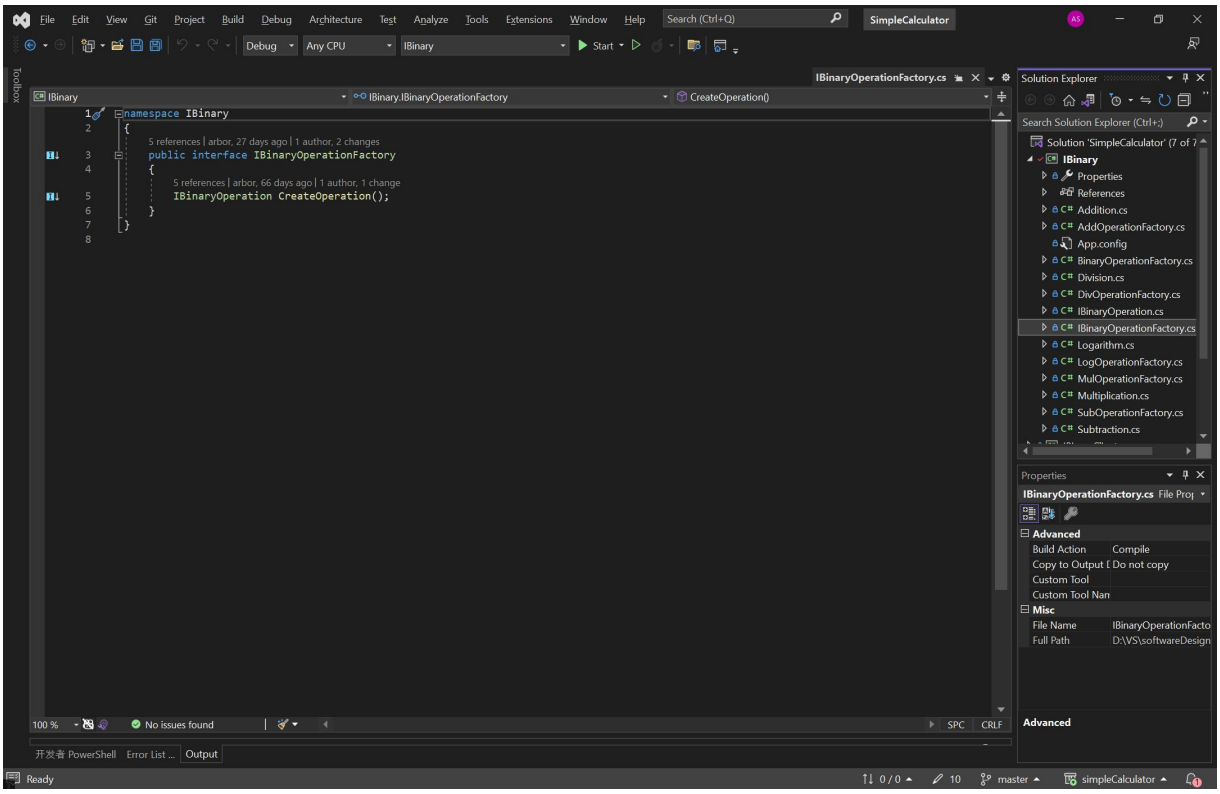

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
AddOperationFactory.cs
IBinary
IBinary.AddOperationFactory
CreateOperation()
namespace IBinary
{
0 references | arbor, 27 days ago | 1 author, 2 changes
public class AddOperationFactory : IBinaryOperationFactory
{
1 reference | arbor, 66 days ago | 1 author, 1 change
public IBinaryOperation CreateOperation()
{
return new Addition();
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
AddOperationFactory.cs File Propert
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nam
Misc
File Name AddOperationFactory.
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator

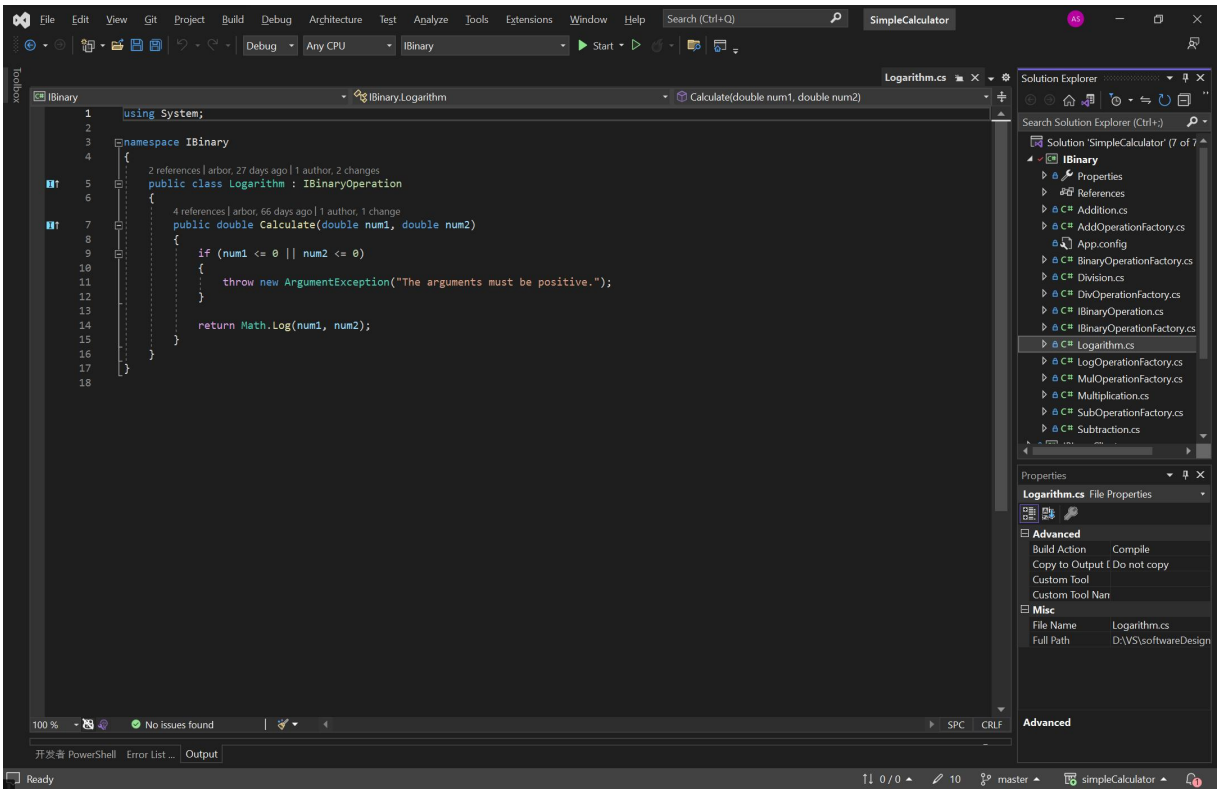

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
BinaryOperationFactory.cs
IBinary
IBinary.BinaryOperationFactory
CreateOperation(string operatorName)
using System;
using System.Configuration;
namespace IBinary
{
2 references | arbor, 27 days ago | 1 author, 2 changes
public class BinaryOperationFactory
{
2 references | arbor, 66 days ago | 1 author, 1 change
public static IBinaryOperation CreateOperation(string operatorName)
{
// Get the fully-qualified name of the operation class from the config file
string operationClassName = ConfigurationManager.AppSettings[operatorName];
// Use reflection to create an instance of the operation class
Type operationClassType = Type.GetType(operationClassName);
IBinaryOperation operation = (IBinaryOperation)Activator.CreateInstance(operationClassType);
return operation;
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
BinaryOperationFactory.cs File Prop
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name BinaryOperationFactor
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator

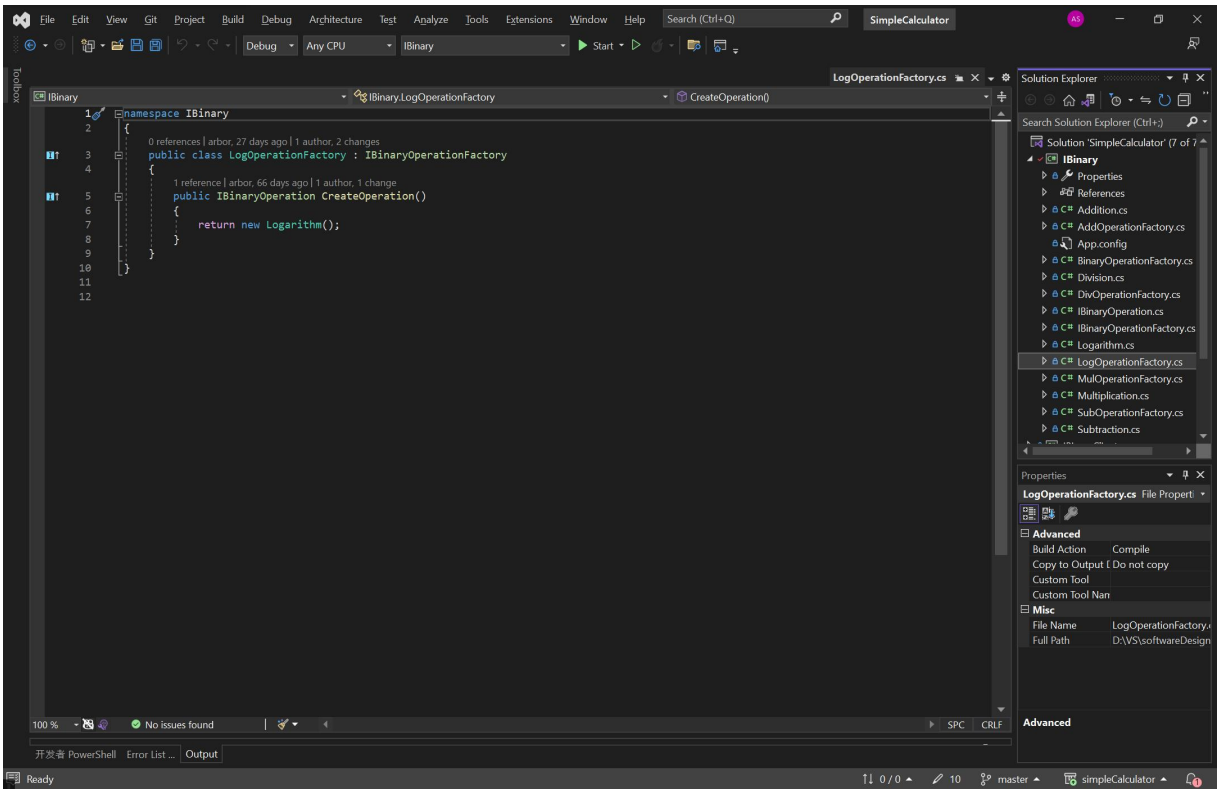

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
Division.cs
IBinary
IBinary.Division
Calculate(double num1, double num2)
using System;
namespace IBinary
{
2 references | arbor, 27 days ago | 1 author, 2 changes
public class Division : IBinaryOperation
{
4 references | arbor, 66 days ago | 1 author, 1 change
public double Calculate(double num1, double num2)
{
if (num2 == 0)
{
throw new DivideByZeroException("Cannot divide by zero.");
}
return num1 / num2;
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
Division.cs File Properties
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name Division.cs
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator

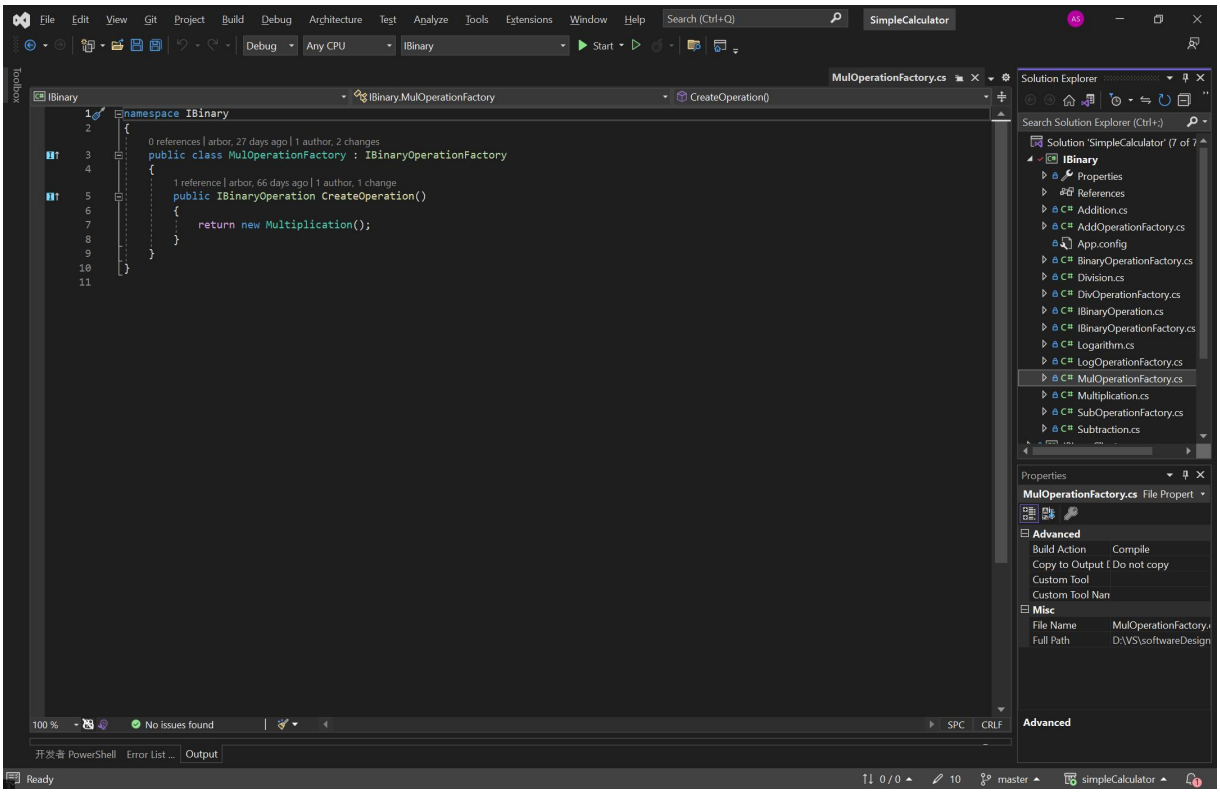

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
DivOperationFactory.cs
IBinary
IBinary.DivOperationFactory
CreateOperation()
namespace IBinary
{
0 references | arbor, 27 days ago | 1 author, 2 changes
public class DivOperationFactory : IBinaryOperationFactory
{
1 reference | arbor, 66 days ago | 1 author, 1 change
public IBinaryOperation CreateOperation()
{
return new Division();
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
DivOperationFactory.cs File Properti
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nam
Misc
File Name DivOperationFactory.c
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator

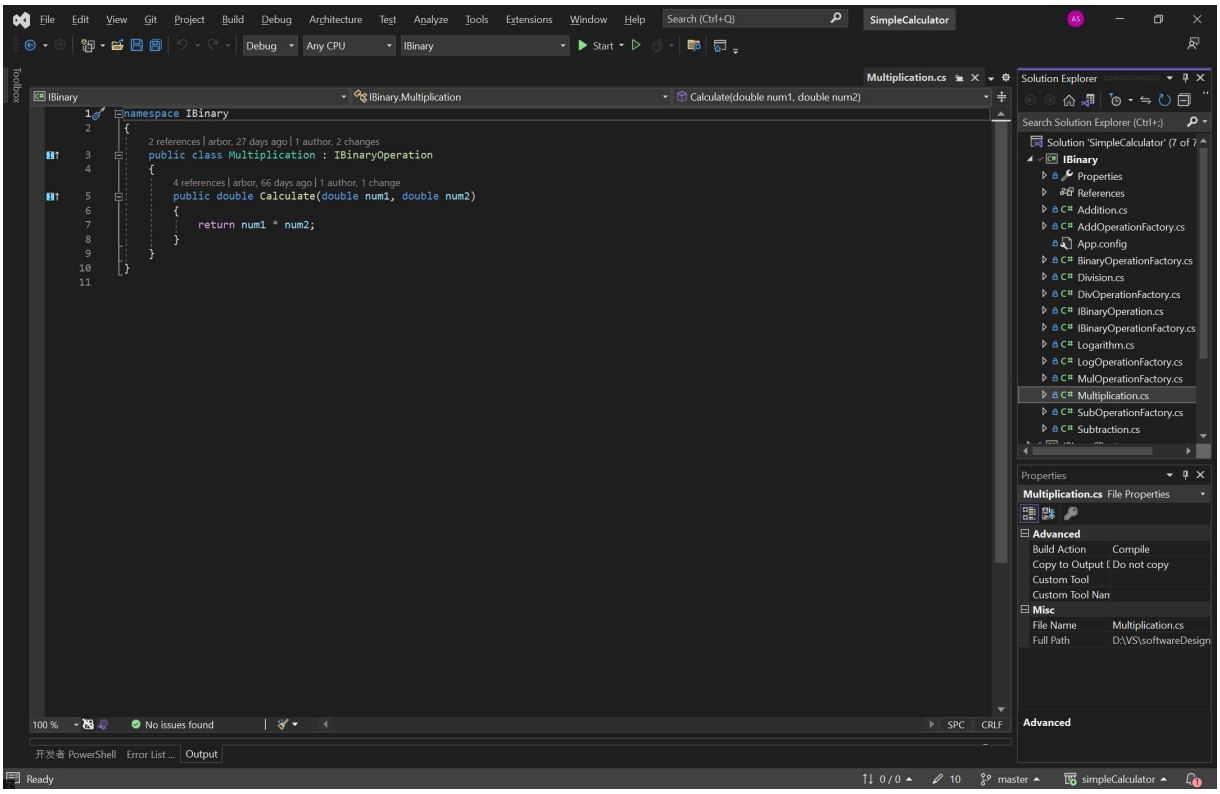

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
IBinaryOperation.cs
IBinary
IBinary.IBinaryOperation
Calculate(double num1, double num2)
namespace IBinary
{
18 references | arbor, 27 days ago | 1 author, 2 changes
public interface IBinaryOperation
{
8 references | arbor, 66 days ago | 1 author, 1 change
double Calculate(double num1, double num2);
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
IBinaryOperation.cs File Properties
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nam
Misc
File Name IBinaryOperation.cs
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator

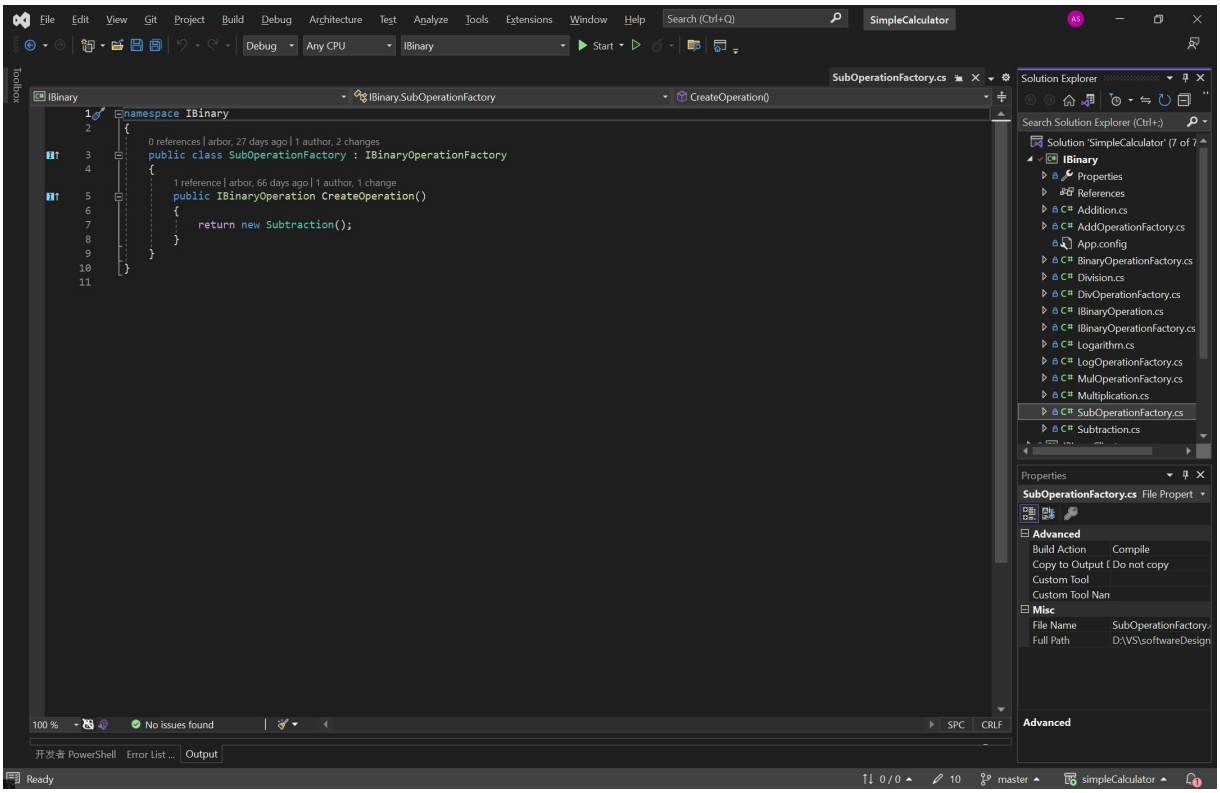

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
IBinaryOperationFactory.cs
IBinary
IBinary.IBinaryOperationFactory
CreateOperation()
namespace IBinary
{
5 references | arbor, 27 days ago | 1 author, 2 changes
public interface IBinaryOperationFactory
{
5 references | arbor, 66 days ago | 1 author, 1 change
IBinaryOperation CreateOperation();
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
IBinaryOperationFactory.cs File Prop
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nam
Misc
File Name IBinaryOperationFacto
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator

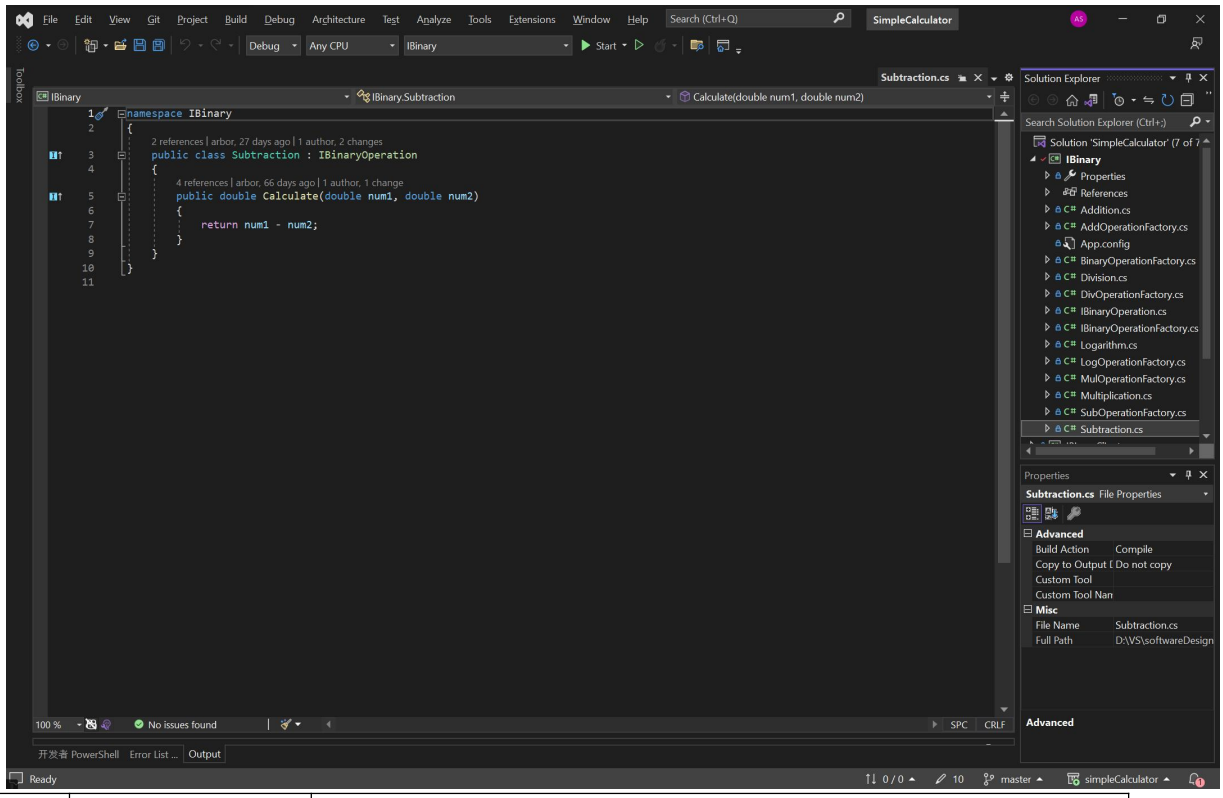

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
Logarithm.cs
IBinary
IBinary.Logarithm
Calculate(double num1, double num2)
using System;
namespace IBinary
{
2 references | arbor, 27 days ago | 1 author, 2 changes
public class Logarithm : IBinaryOperation
{
4 references | arbor, 66 days ago | 1 author, 1 change
public double Calculate(double num1, double num2)
{
if (num1 <= 0 || num2 <= 0)
{
throw new ArgumentException("The arguments must be positive.");
}
return Math.Log(num1, num2);
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
Logarithm.cs File Properties
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nam
Misc
File Name Logarithm.cs
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator


File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
LogOperationFactory.cs
IBinary
IBinary.LogOperationFactory
CreateOperation()
namespace IBinary
{
0 references | arbor, 27 days ago | 1 author, 2 changes
public class LogOperationFactory : IBinaryOperationFactory
{
1 reference | arbor, 66 days ago | 1 author, 1 change
public IBinaryOperation CreateOperation()
{
return new Logarithm();
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
LogOperationFactory.cs File Properti
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name LogOperationFactory.
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator

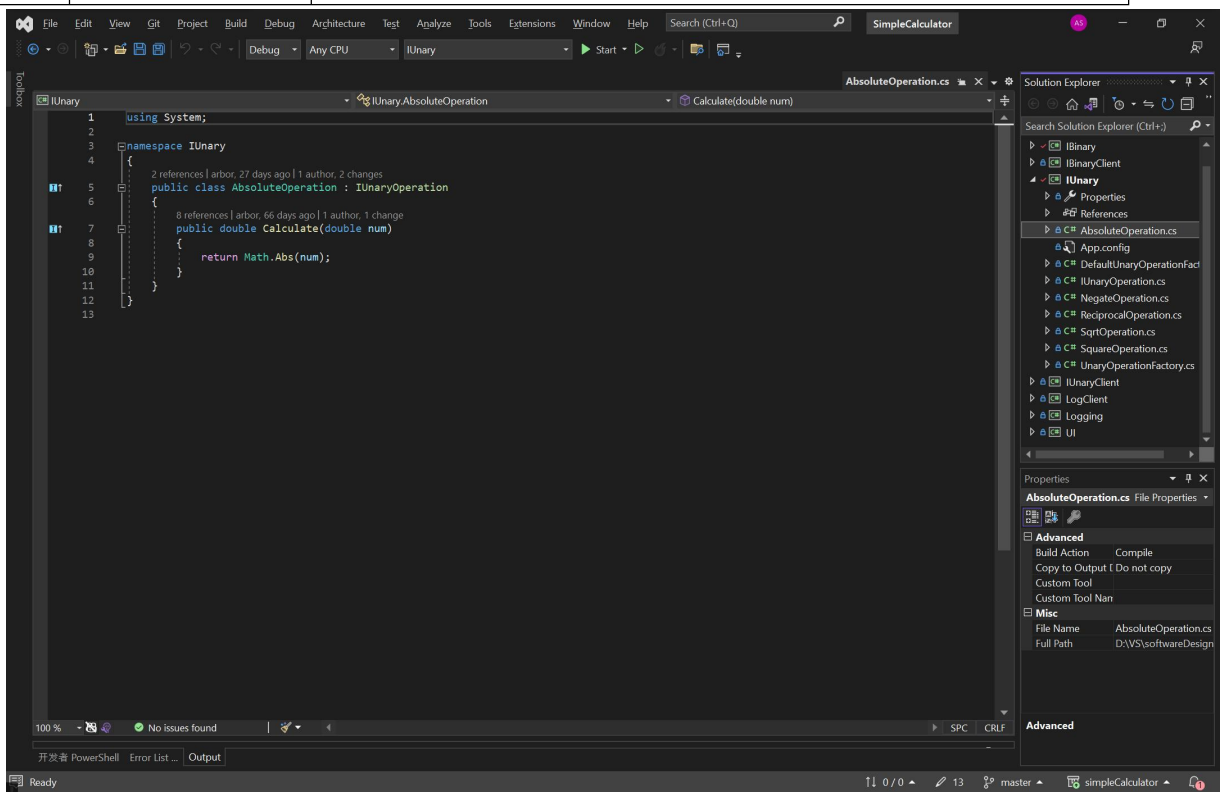

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
MulOperationFactory.cs
IBinary
IBinary.MulOperationFactory
CreateOperation()
namespace IBinary
{
0 references | arbor, 27 days ago | 1 author, 2 changes
public class MulOperationFactory : IBinaryOperationFactory
{
1 reference | arbor, 66 days ago | 1 author, 1 change
public IBinaryOperation CreateOperation()
{
return new Multiplication();
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
MulOperationFactory.cs File Propert
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name MulOperationFactory.
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator

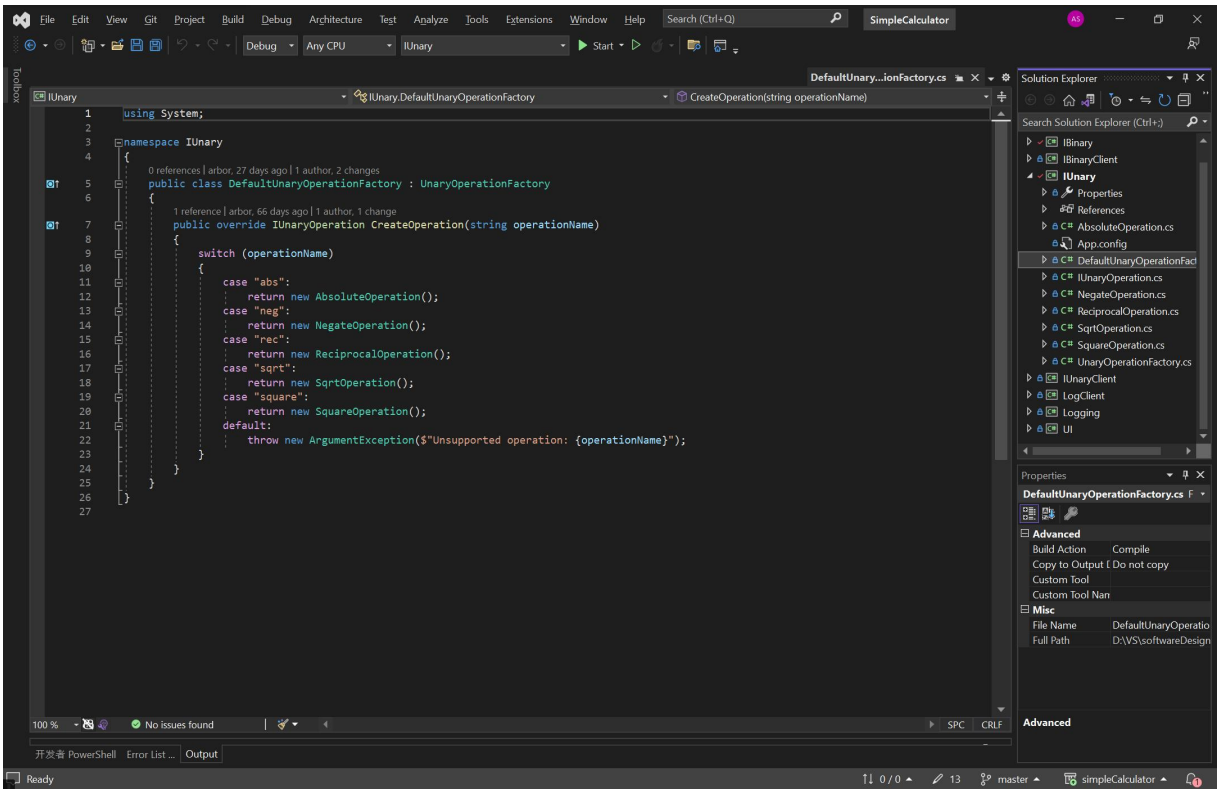

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
Multiplication.cs
IBinary
IBinary.Multiplication
Calculate(double num1, double num2)
namespace IBinary
{
2 references | arbor, 27 days ago | 1 author, 2 changes
public class Multiplication : IBinaryOperation
{
4 references | arbor, 66 days ago | 1 author, 1 change
public double Calculate(double num1, double num2)
{
return num1 * num2;
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
Multiplication.cs File Properties
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name Multiplication.cs
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator

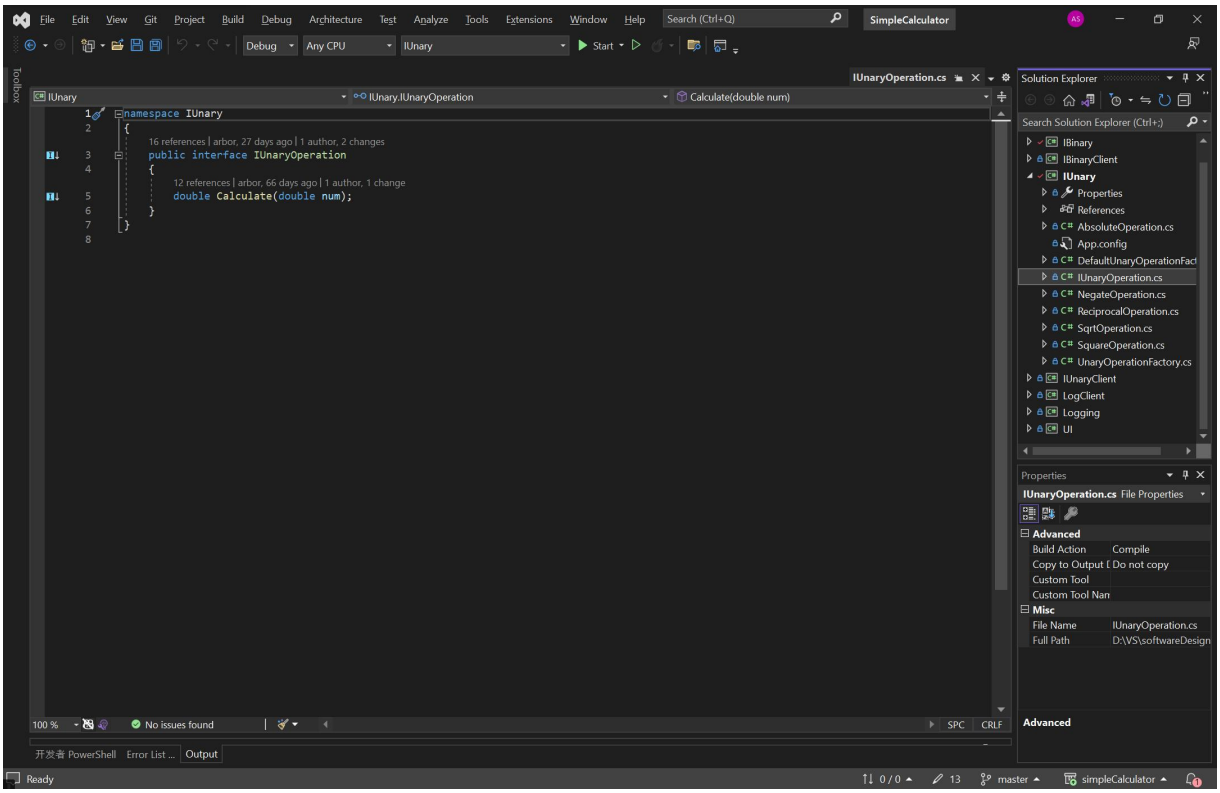

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IBinary
Start
SubOperationFactory.cs
IBinary
IBinary.SubOperationFactory
CreateOperation()
namespace IBinary
{
0 references | arbor, 27 days ago | 1 author, 2 changes
public class SubOperationFactory : IBinaryOperationFactory
{
1 reference | arbor, 66 days ago | 1 author, 1 change
public IBinaryOperation CreateOperation()
{
return new Subtraction();
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (7 of 7
IBinary
Properties
References
Addition.cs
AddOperationFactory.cs
App.config
BinaryOperationFactory.cs
Division.cs
DivOperationFactory.cs
IBinaryOperation.cs
IBinaryOperationFactory.cs
Logarithm.cs
LogOperationFactory.cs
MulOperationFactory.cs
Multiplication.cs
SubOperationFactory.cs
Subtraction.cs
Properties
SubOperationFactory.cs File Propert
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name SubOperationFactory.
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
10
master
simpleCalculator

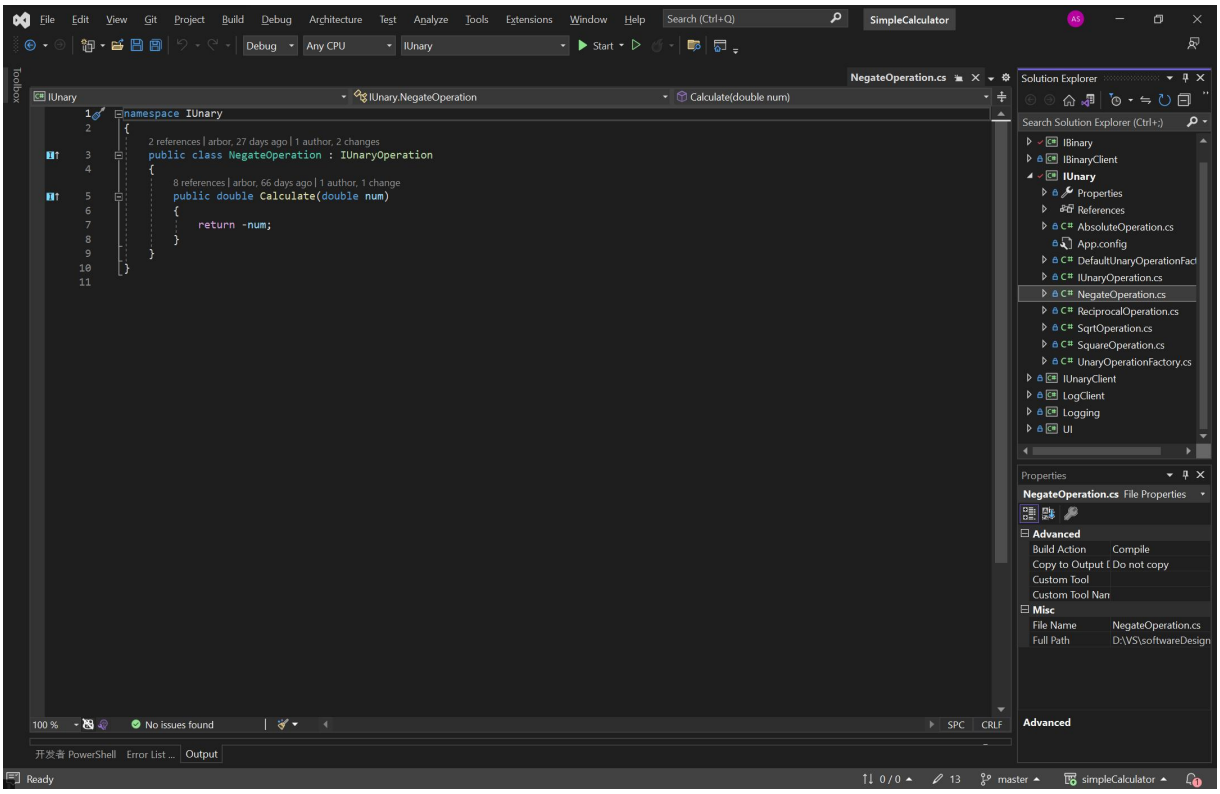


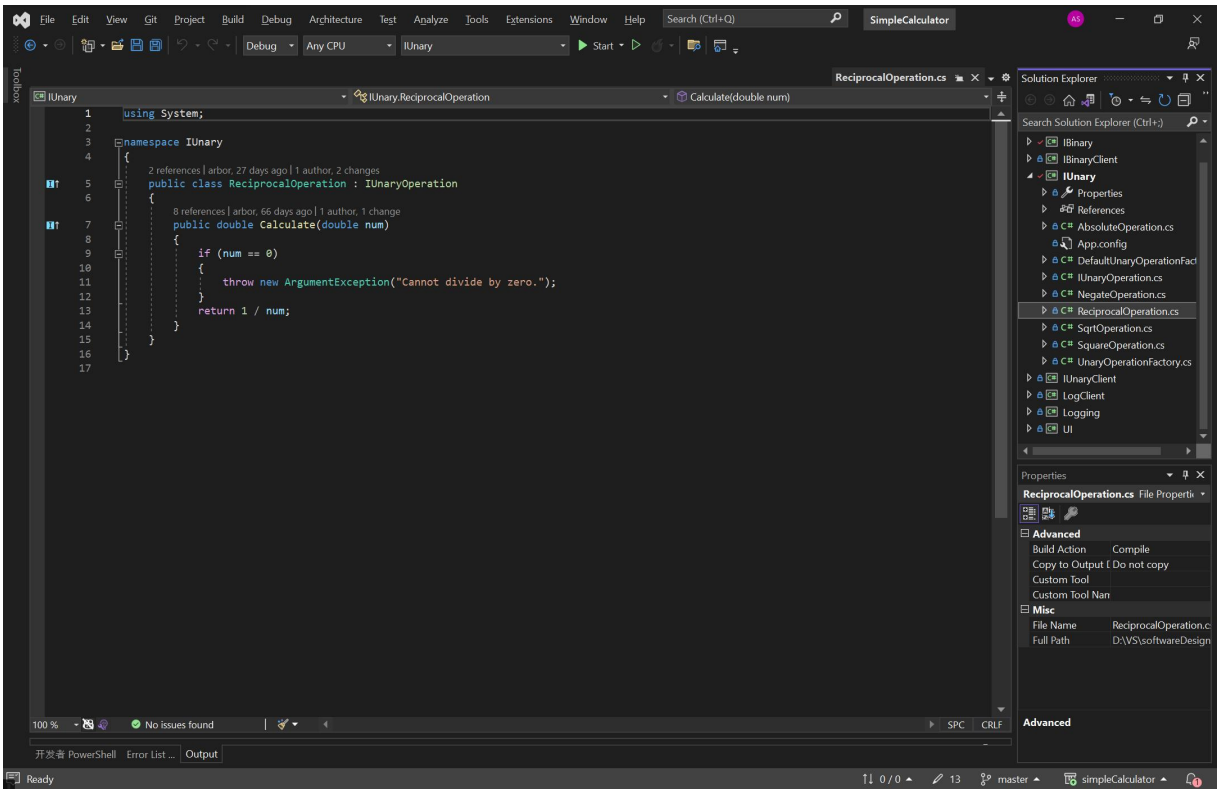
3	计算器的领域模式 2	采用工厂方法模式实现计算器**一元计算领域模型**，实现取反、求 1/x、$2\sqrt{\ }$、x^2、10^x、|x|、log、ln、exp 等一元运算

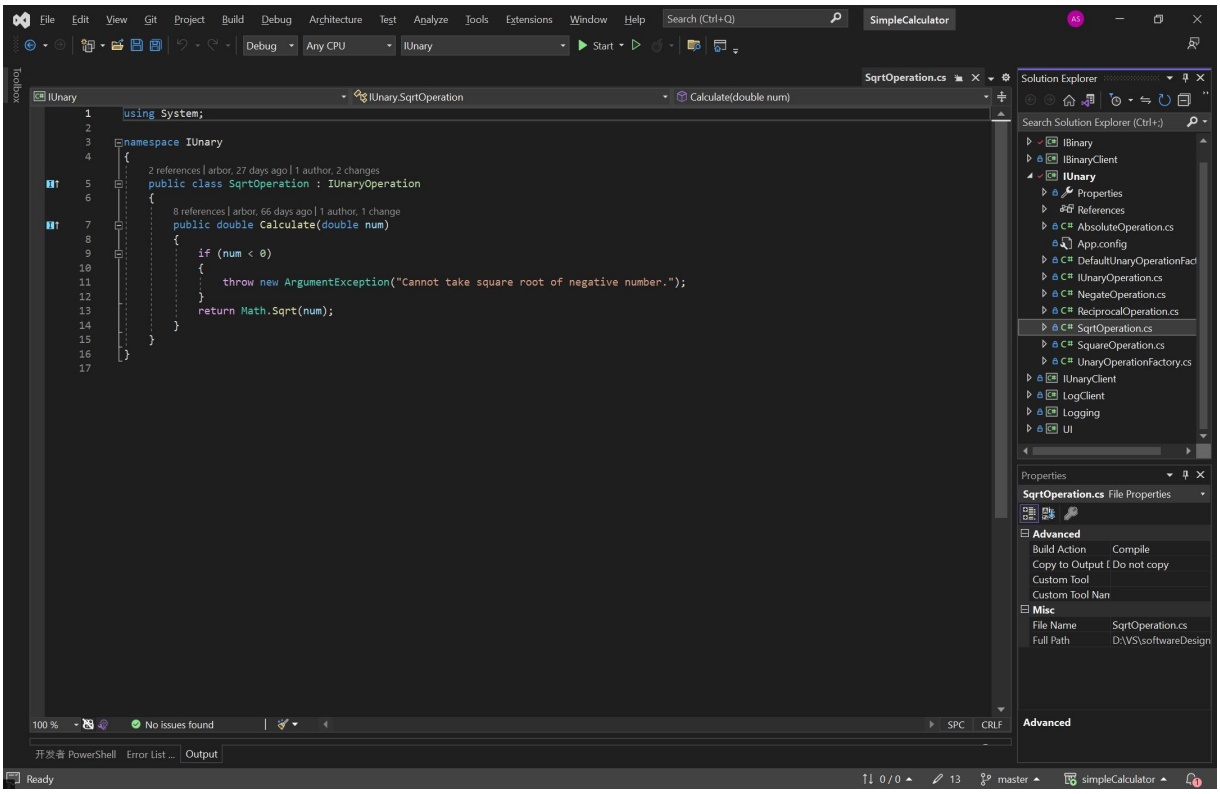


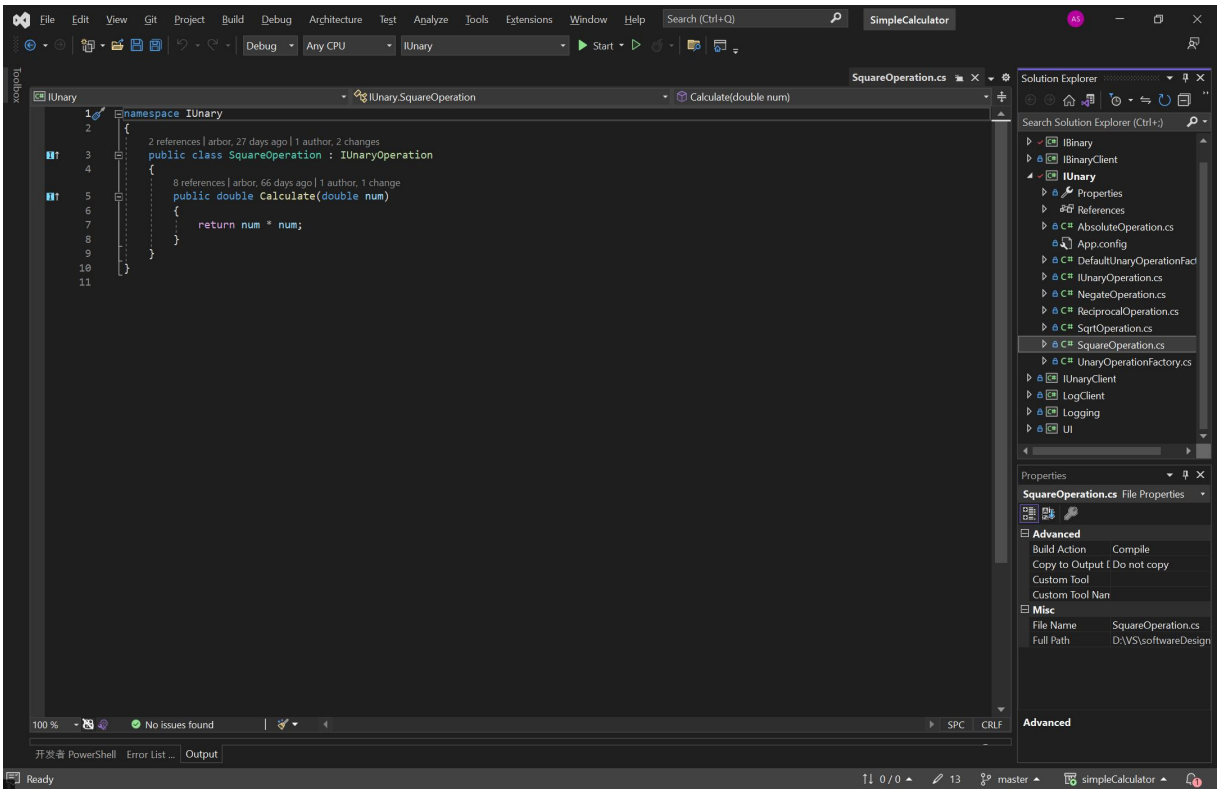

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IUnary
Start
DefaultUnary...ionFactory.cs
IUnary
IUnary.DefaultUnaryOperationFactory
CreateOperation(string operationName)
using System;
namespace IUnary
{
0 references | arbor, 27 days ago | 1 author, 2 changes
public class DefaultUnaryOperationFactory : UnaryOperationFactory
{
1 reference | arbor, 66 days ago | 1 author, 1 change
public override IUnaryOperation CreateOperation(string operationName)
{
switch (operationName)
{
case "abs":
return new AbsoluteOperation();
case "neg":
return new NegateOperation();
case "rec":
return new ReciprocalOperation();
case "sqrt":
return new SqrtOperation();
case "square":
return new SquareOperation();
default:
throw new ArgumentException($"Unsupported operation: {operationName}");
}
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
IBinary
IBinaryClient
IUnary
Properties
References
AbsoluteOperation.cs
App.config
DefaultUnaryOperationFac
IUnaryOperation.cs
NegateOperation.cs
ReciprocalOperation.cs
SqrtOperation.cs
SquareOperation.cs
UnaryOperationFactory.cs
IUnaryClient
LogClient
Logging
UI
Properties
DefaultUnaryOperationFactory.cs F
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name DefaultUnaryOperatio
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
13
master
simpleCalculator

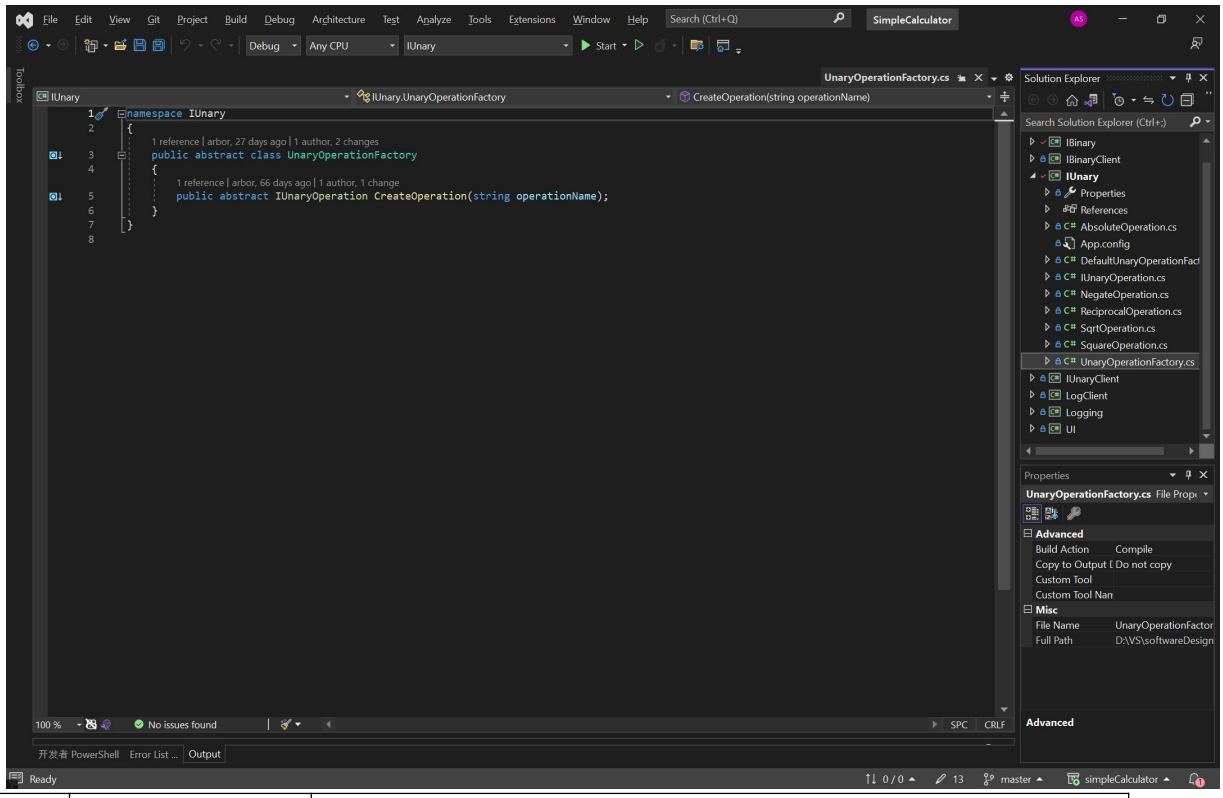

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IUnary
Start
IUnaryOperation.cs
IUnary
IUnary.IUnaryOperation
Calculate(double num)
namespace IUnary
{
16 references | arbor, 27 days ago | 1 author, 2 changes
public interface IUnaryOperation
{
12 references | arbor, 66 days ago | 1 author, 1 change
double Calculate(double num);
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
IBinary
IBinaryClient
IUnary
Properties
References
AbsoluteOperation.cs
App.config
DefaultUnaryOperationFac
IUnaryOperation.cs
NegateOperation.cs
ReciprocalOperation.cs
SqrtOperation.cs
SquareOperation.cs
UnaryOperationFactory.cs
IUnaryClient
LogClient
Logging
UI
Properties
IUnaryOperation.cs File Properties
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name IUnaryOperation.cs
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
13
master
simpleCalculator


File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IUnary
Start
NegateOperation.cs
IUnary
IUnary.NegateOperation
Calculate(double num)
namespace IUnary
{
2 references | arbor, 27 days ago | 1 author, 2 changes
public class NegateOperation : IUnaryOperation
{
8 references | arbor, 66 days ago | 1 author, 1 change
public double Calculate(double num)
{
return -num;
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
IBinary
IBinaryClient
IUnary
Properties
References
AbsoluteOperation.cs
App.config
DefaultUnaryOperationFact
IUnaryOperation.cs
NegateOperation.cs
ReciprocalOperation.cs
SqrtOperation.cs
SquareOperation.cs
UnaryOperationFactory.cs
IUnaryClient
LogClient
Logging
UI
Properties
NegateOperation.cs File Properties
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name NegateOperation.cs
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
13
master
simpleCalculator

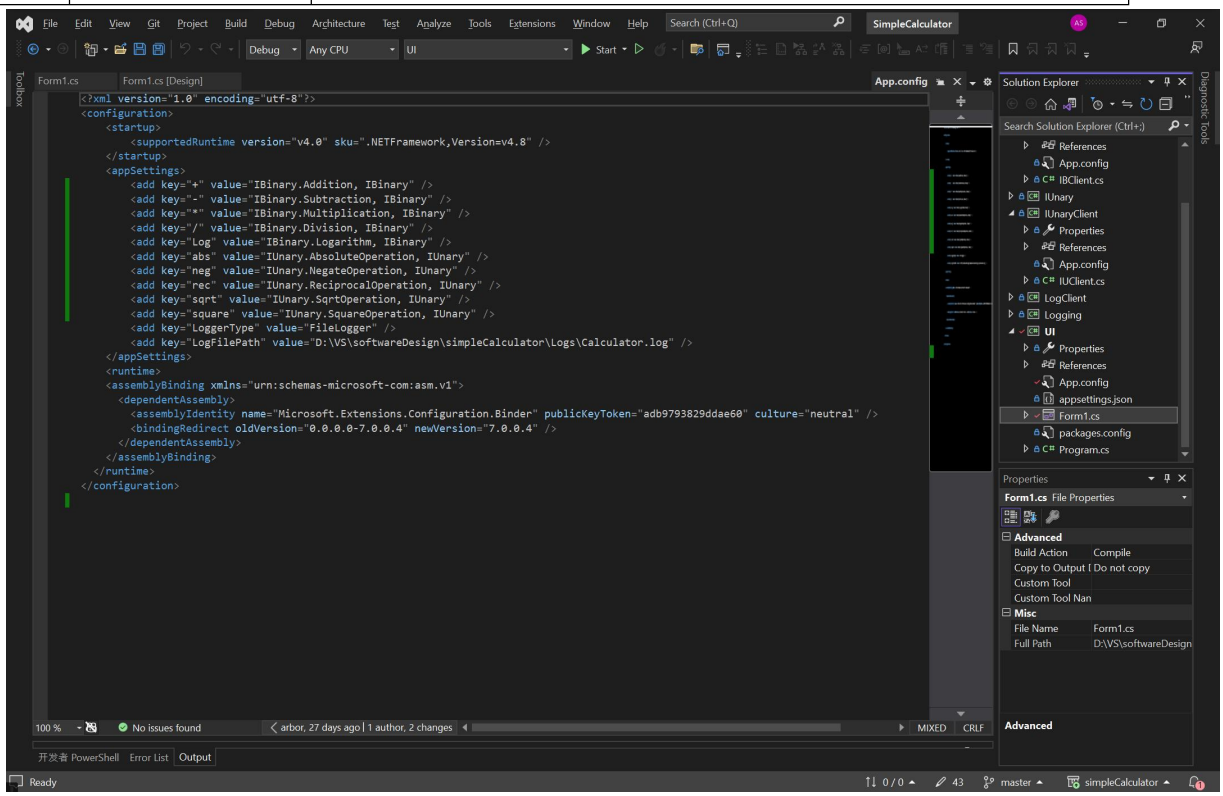

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
IUnary
Start
ReciprocalOperation.cs
IUnary
IUnary.ReciprocalOperation
Calculate(double num)
using System;
namespace IUnary
{
2 references | arbor, 27 days ago | 1 author, 2 changes
public class ReciprocalOperation : IUnaryOperation
{
8 references | arbor, 66 days ago | 1 author, 1 change
public double Calculate(double num)
{
if (num == 0)
{
throw new ArgumentException("Cannot divide by zero.");
}
return 1 / num;
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
IBinary
IBinaryClient
IUnary
Properties
References
AbsoluteOperation.cs
App.config
DefaultUnaryOperationFact
IUnaryOperation.cs
NegateOperation.cs
ReciprocalOperation.cs
SqrtOperation.cs
SquareOperation.cs
UnaryOperationFactory.cs
IUnaryClient
LogClient
Logging
UI
Properties
ReciprocalOperation.cs File Properti
Advanced
Build Action Compile
Copy to Output [Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name ReciprocalOperation.c
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
13
master
simpleCalculator

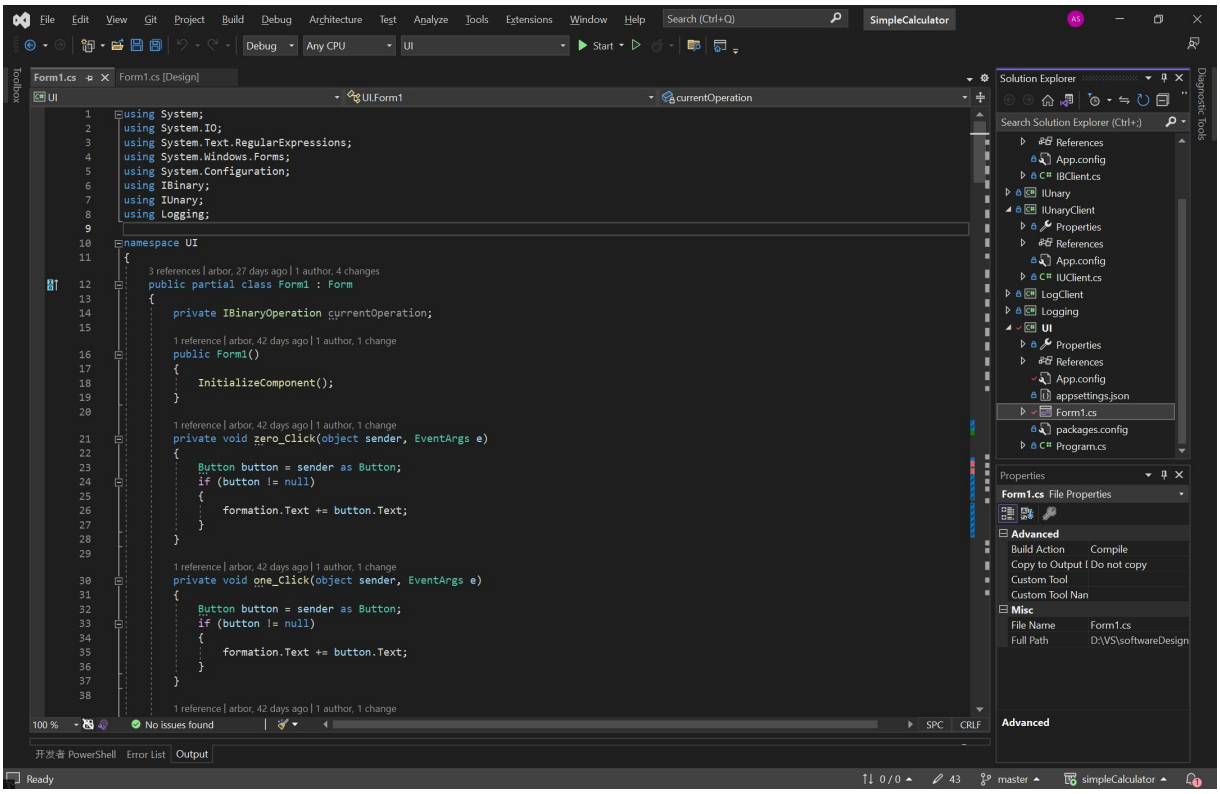

SqrtOperation.cs
IUnary.SqrtOperation
Calculate(double num)
using System;
namespace IUnary
{
2 references | arbor, 27 days ago | 1 author, 2 changes
public class SqrtOperation : IUnaryOperation
{
8 references | arbor, 66 days ago | 1 author, 1 change
public double Calculate(double num)
{
if (num < 0)
{
throw new ArgumentException("Cannot take square root of negative number.");
}
return Math.Sqrt(num);
}
}
}
Solution Explorer
SqrtOperation.cs File Properties
Advanced
Build Action Compile
Misc
File Name SqrtOperation.cs
Full Path D:\VS\softwareDesign
No issues found
Ready
simpleCalculator


SquareOperation.cs
IUnary.SquareOperation
Calculate(double num)
namespace IUnary
{
2 references | arbor, 27 days ago | 1 author, 2 changes
public class SquareOperation : IUnaryOperation
{
8 references | arbor, 66 days ago | 1 author, 1 change
public double Calculate(double num)
{
return num * num;
}
}
}
Solution Explorer
SquareOperation.cs File Properties
Advanced
Build Action Compile
Misc
File Name SquareOperation.cs
Full Path D:\VS\softwareDesign
No issues found
Ready
simpleCalculator

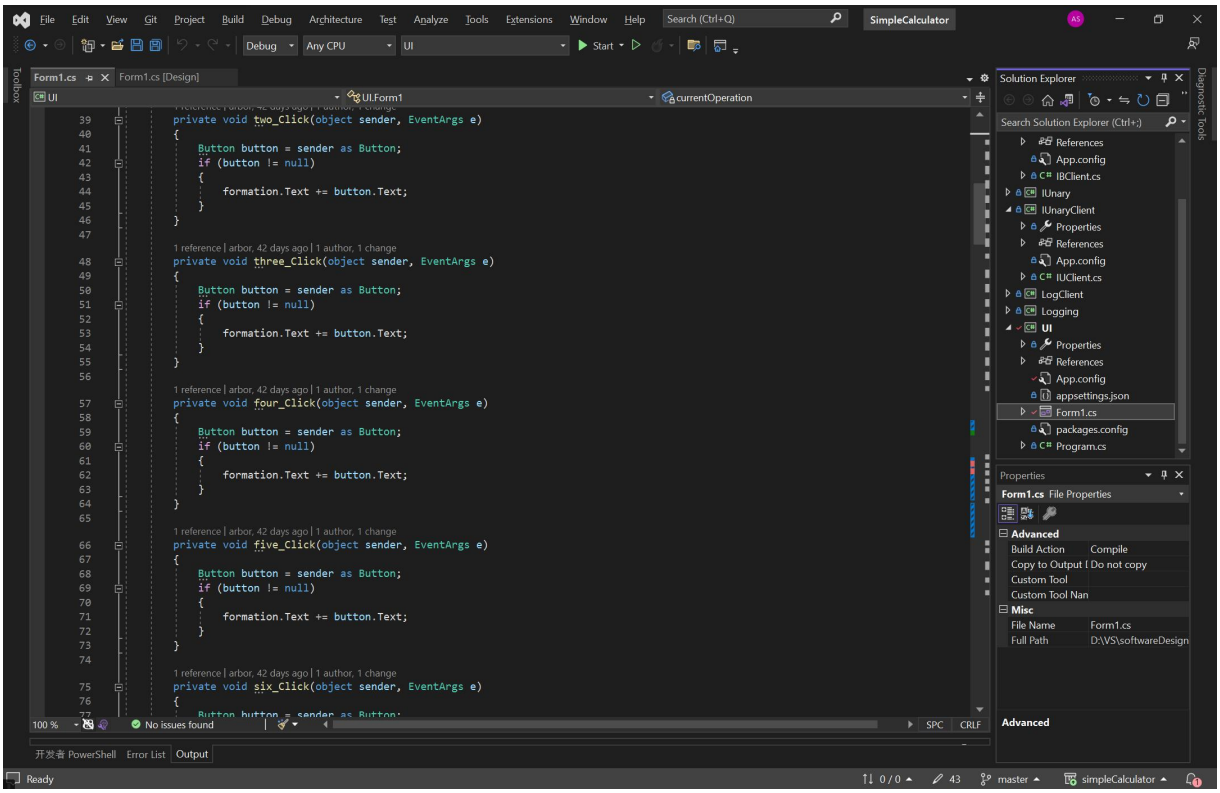


5	计算器的 UI 界面	使用**配置文件**与**反射机制**实现二元计算工厂对象和一元计算工厂对象的实例化

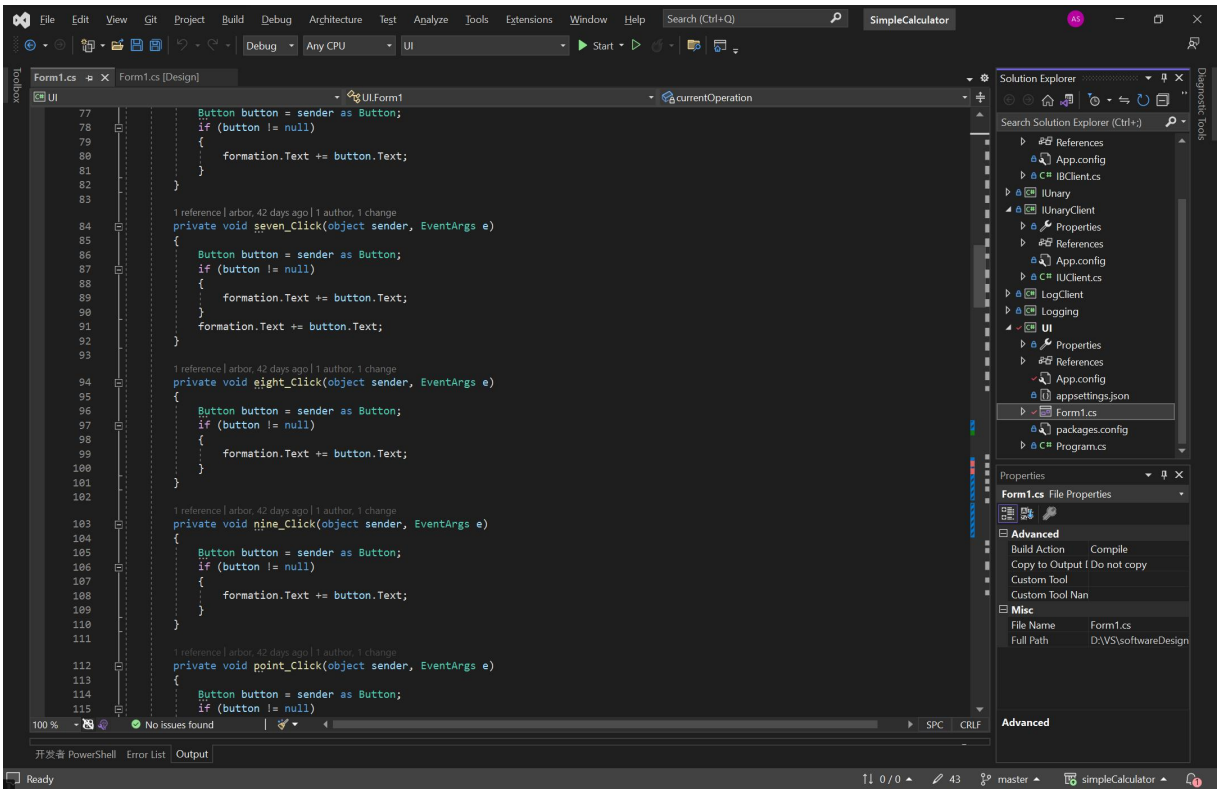




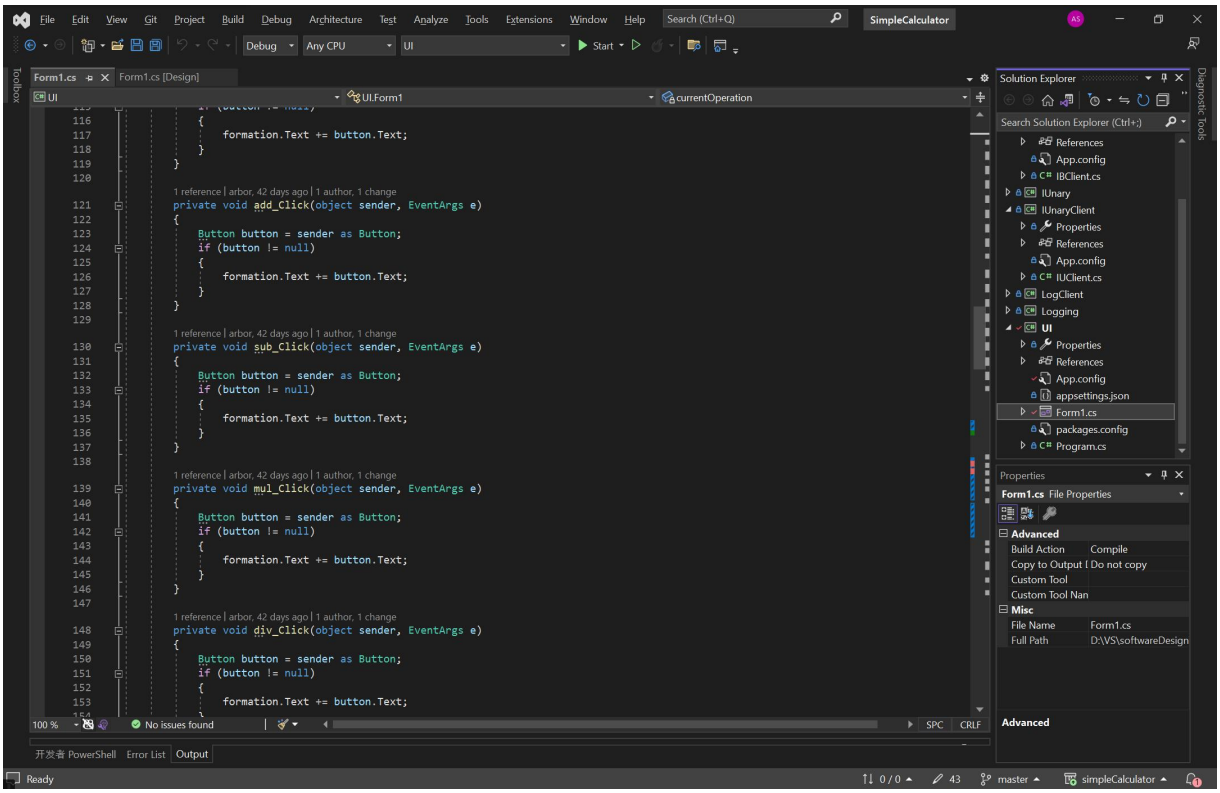
```
using System;
using System.IO;
using System.Text.RegularExpressions;
using System.Windows.Forms;
using System.Configuration;
using IBinary;
using IUnary;
using Logging;

namespace UI
{
    3 references | arbor, 27 days ago | 1 author, 4 changes
    public partial class Form1 : Form
    {
        private IBinaryOperation currentOperation;

        1 reference | arbor, 42 days ago | 1 author, 1 change
        public Form1()
        {
            InitializeComponent();
        }

        1 reference | arbor, 42 days ago | 1 author, 1 change
        private void zero_Click(object sender, EventArgs e)
        {
            Button button = sender as Button;
            if (button != null)
            {
                formation.Text += button.Text;
            }
        }

        1 reference | arbor, 42 days ago | 1 author, 1 change
        private void one_Click(object sender, EventArgs e)
        {
            Button button = sender as Button;
            if (button != null)
            {
                formation.Text += button.Text;
            }
        }

        1 reference | arbor, 42 days ago | 1 author, 1 change
```



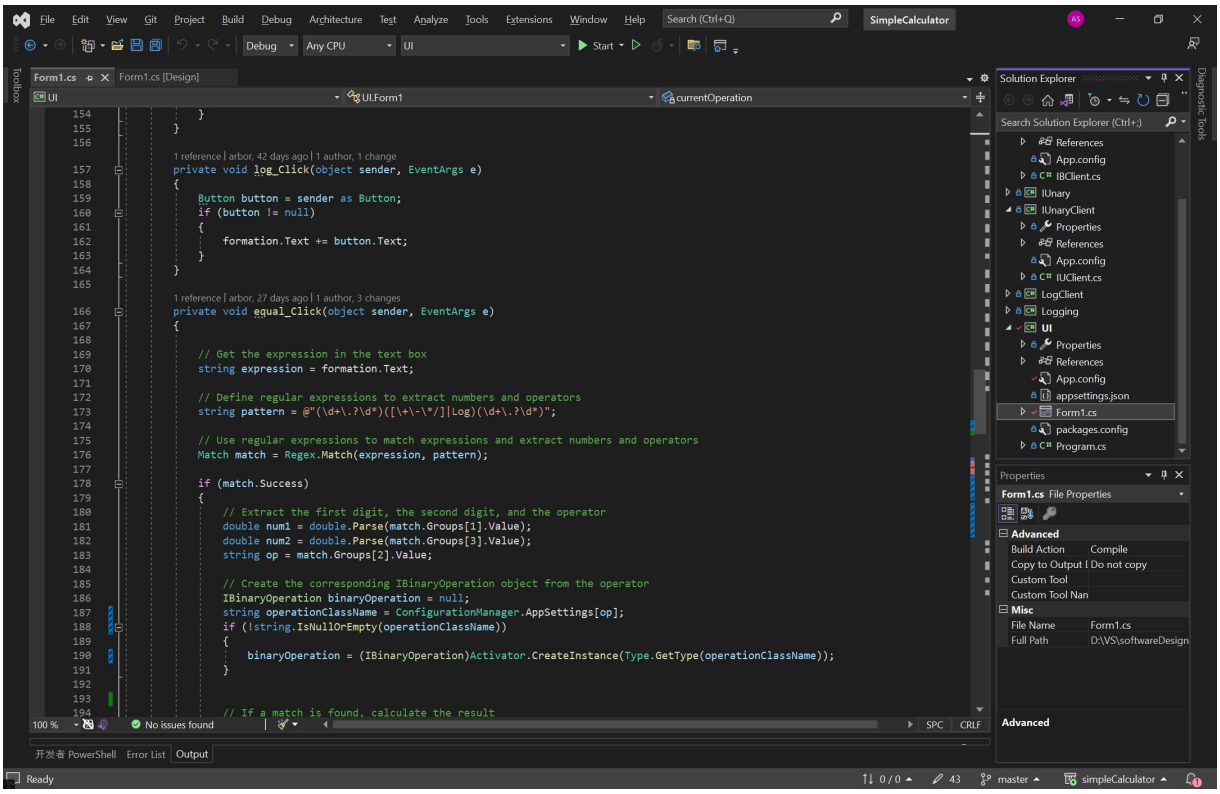
```
        private void two_Click(object sender, EventArgs e)
        {
            Button button = sender as Button;
            if (button != null)
            {
                formation.Text += button.Text;
            }
        }

        1 reference | arbor, 42 days ago | 1 author, 1 change
        private void three_Click(object sender, EventArgs e)
        {
            Button button = sender as Button;
            if (button != null)
            {
                formation.Text += button.Text;
            }
        }

        1 reference | arbor, 42 days ago | 1 author, 1 change
        private void four_Click(object sender, EventArgs e)
        {
            Button button = sender as Button;
            if (button != null)
            {
                formation.Text += button.Text;
            }
        }

        1 reference | arbor, 42 days ago | 1 author, 1 change
        private void five_Click(object sender, EventArgs e)
        {
            Button button = sender as Button;
            if (button != null)
            {
                formation.Text += button.Text;
            }
        }

        1 reference | arbor, 42 days ago | 1 author, 1 change
        private void six_Click(object sender, EventArgs e)
        {
            Button button = sender as Button;
```



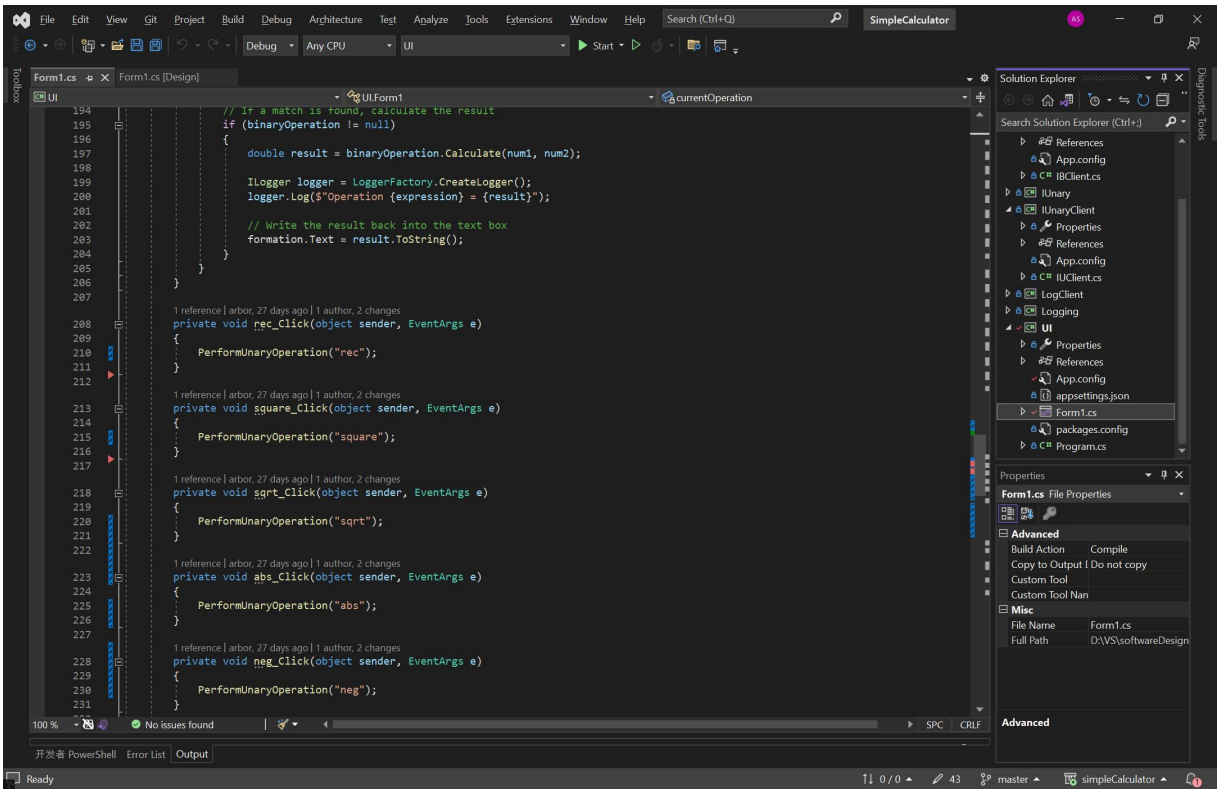
```
        Button button = sender as Button;
        if (button != null)
        {
            formation.Text += button.Text;
        }
    }

    1 reference | arbor, 42 days ago | 1 author, 1 change
    private void seven_Click(object sender, EventArgs e)
    {
        Button button = sender as Button;
        if (button != null)
        {
            formation.Text += button.Text;
        }
        formation.Text += button.Text;
    }

    1 reference | arbor, 42 days ago | 1 author, 1 change
    private void eight_Click(object sender, EventArgs e)
    {
        Button button = sender as Button;
        if (button != null)
        {
            formation.Text += button.Text;
        }
    }

    1 reference | arbor, 42 days ago | 1 author, 1 change
    private void nine_Click(object sender, EventArgs e)
    {
        Button button = sender as Button;
        if (button != null)
        {
            formation.Text += button.Text;
        }
    }

    1 reference | arbor, 42 days ago | 1 author, 1 change
    private void point_Click(object sender, EventArgs e)
    {
        Button button = sender as Button;
        if (button != null)
```



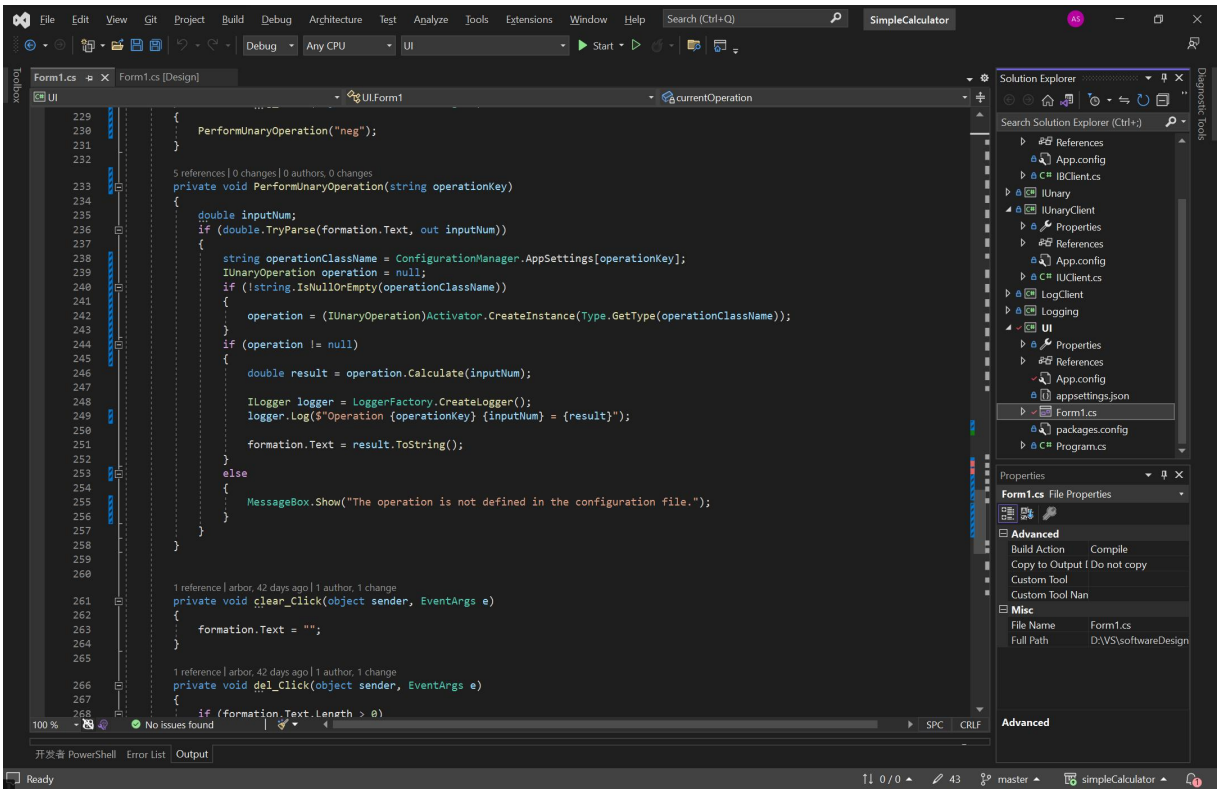
```
        {
            formation.Text += button.Text;
        }
    }

    1 reference | arbor, 42 days ago | 1 author, 1 change
    private void add_Click(object sender, EventArgs e)
    {
        Button button = sender as Button;
        if (button != null)
        {
            formation.Text += button.Text;
        }
    }

    1 reference | arbor, 42 days ago | 1 author, 1 change
    private void sub_Click(object sender, EventArgs e)
    {
        Button button = sender as Button;
        if (button != null)
        {
            formation.Text += button.Text;
        }
    }

    1 reference | arbor, 42 days ago | 1 author, 1 change
    private void mul_Click(object sender, EventArgs e)
    {
        Button button = sender as Button;
        if (button != null)
        {
            formation.Text += button.Text;
        }
    }

    1 reference | arbor, 42 days ago | 1 author, 1 change
    private void div_Click(object sender, EventArgs e)
    {
        Button button = sender as Button;
        if (button != null)
        {
            formation.Text += button.Text;
```



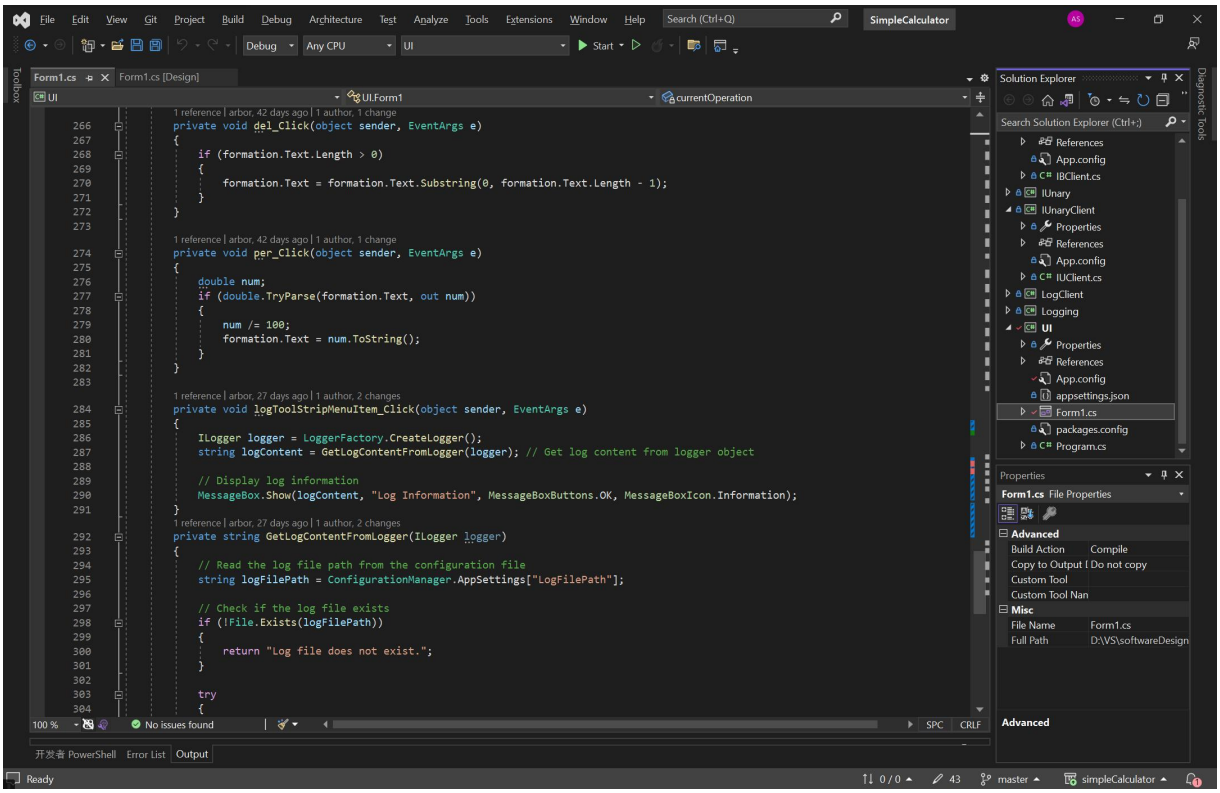
```
        }
    }

    1 reference | arbor, 42 days ago | 1 author, 1 change
    private void log_Click(object sender, EventArgs e)
    {
        Button button = sender as Button;
        if (button != null)
        {
            formation.Text += button.Text;
        }
    }

    1 reference | arbor, 27 days ago | 1 author, 3 changes
    private void equal_Click(object sender, EventArgs e)
    {

        // Get the expression in the text box
        string expression = formation.Text;

        // Define regular expressions to extract numbers and operators
        string pattern = @"(\d+\.?\d*)([\+\-\*/]|Log)(\d+\.?\d*)";

        // Use regular expressions to match expressions and extract numbers and operators
        Match match = Regex.Match(expression, pattern);

        if (match.Success)
        {
            // Extract the first digit, the second digit, and the operator
            double num1 = double.Parse(match.Groups[1].Value);
            double num2 = double.Parse(match.Groups[3].Value);
            string op = match.Groups[2].Value;

            // Create the corresponding IBinaryOperation object from the operator
            IBinaryOperation binaryOperation = null;
            string operationClassName = ConfigurationManager.AppSettings[op];
            if (!string.IsNullOrEmpty(operationClassName))
            {
                binaryOperation = (IBinaryOperation)Activator.CreateInstance(Type.GetType(operationClassName));
            }


            // If a match is found, calculate the result
```



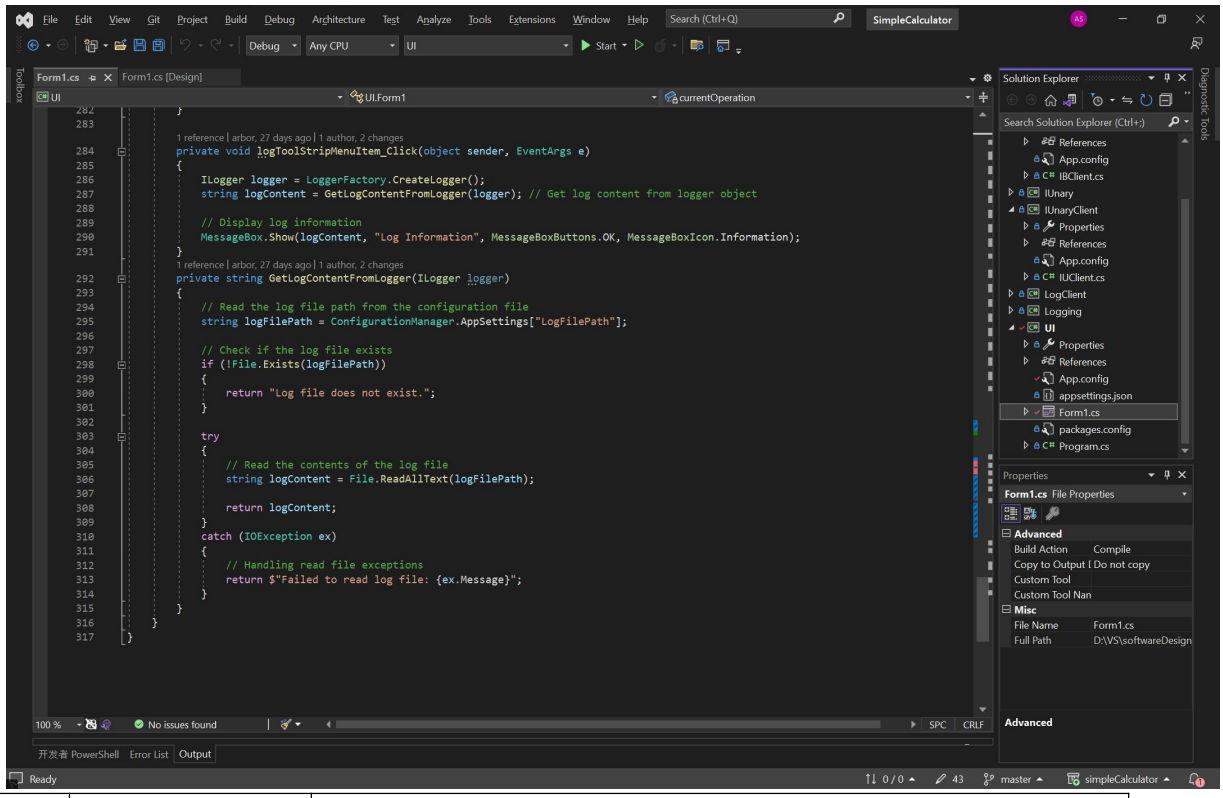
```
            // If a match is found, calculate the result
            if (binaryOperation != null)
            {
                double result = binaryOperation.Calculate(num1, num2);

                ILogger logger = LoggerFactory.CreateLogger();
                logger.Log($"Operation {expression} = {result}");

                // Write the result back into the text box
                formation.Text = result.ToString();
            }
        }
    }

    1 reference | arbor, 27 days ago | 1 author, 2 changes
    private void rec_Click(object sender, EventArgs e)
    {
        PerformUnaryOperation("rec");
    }

    1 reference | arbor, 27 days ago | 1 author, 2 changes
    private void square_Click(object sender, EventArgs e)
    {
        PerformUnaryOperation("square");
    }

    1 reference | arbor, 27 days ago | 1 author, 2 changes
    private void sqrt_Click(object sender, EventArgs e)
    {
        PerformUnaryOperation("sqrt");
    }

    1 reference | arbor, 27 days ago | 1 author, 2 changes
    private void abs_Click(object sender, EventArgs e)
    {
        PerformUnaryOperation("abs");
    }

    1 reference | arbor, 27 days ago | 1 author, 2 changes
    private void neg_Click(object sender, EventArgs e)
    {
        PerformUnaryOperation("neg");
    }
```



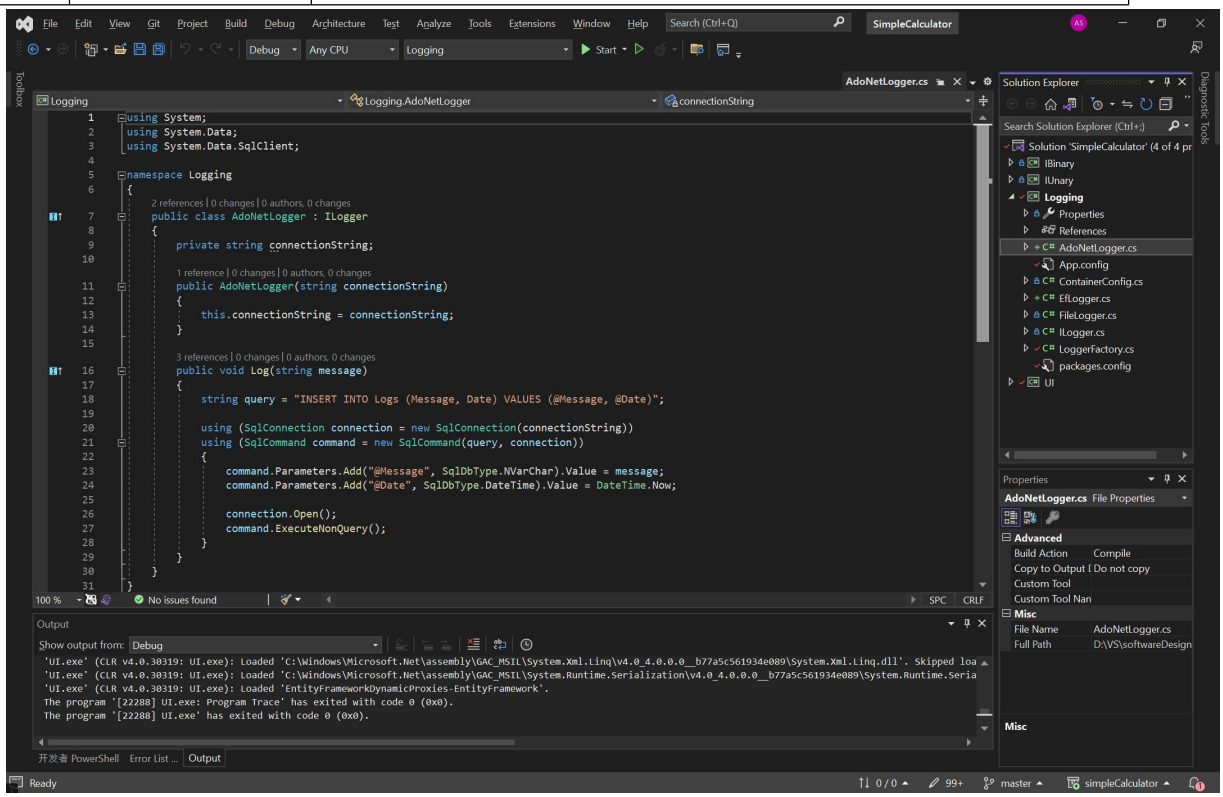
```
        {
            PerformUnaryOperation("neg");
        }

        5 references | 0 changes | 0 authors, 0 changes
        private void PerformUnaryOperation(string operationKey)
        {
            double inputNum;
            if (double.TryParse(formation.Text, out inputNum))
            {
                string operationClassName = ConfigurationManager.AppSettings[operationKey];
                IUnaryOperation operation = null;
                if (!string.IsNullOrEmpty(operationClassName))
                {
                    operation = (IUnaryOperation)Activator.CreateInstance(Type.GetType(operationClassName));
                }
                if (operation != null)
                {
                    double result = operation.Calculate(inputNum);

                    ILogger logger = LoggerFactory.CreateLogger();
                    logger.Log($"Operation {operationKey} {inputNum} = {result}");

                    formation.Text = result.ToString();
                }
                else
                {
                    MessageBox.Show("The operation is not defined in the configuration file.");
                }
            }
        }


        1 reference | arbor, 42 days ago | 1 author, 1 change
        private void clear_Click(object sender, EventArgs e)
        {
            formation.Text = "";
        }

        1 reference | arbor, 42 days ago | 1 author, 1 change
        private void del_Click(object sender, EventArgs e)
        {
            if (formation.Text.Length > 0)
```

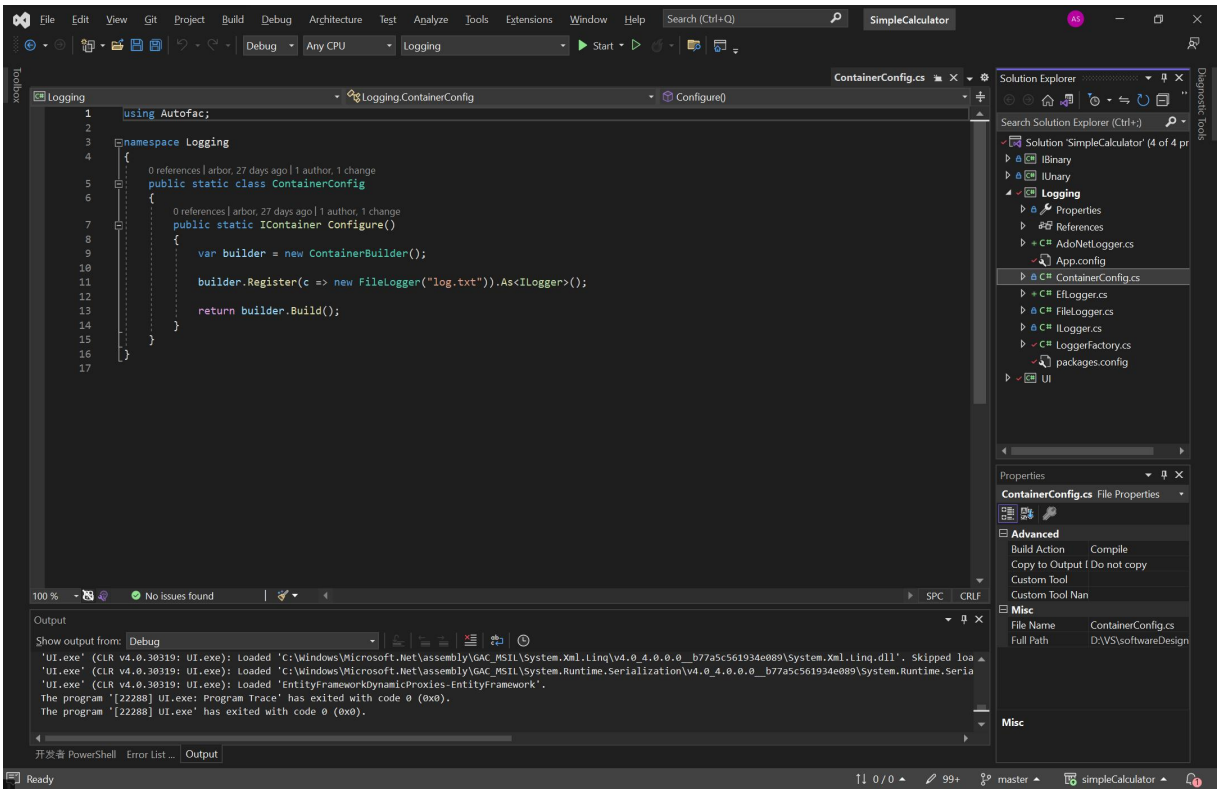


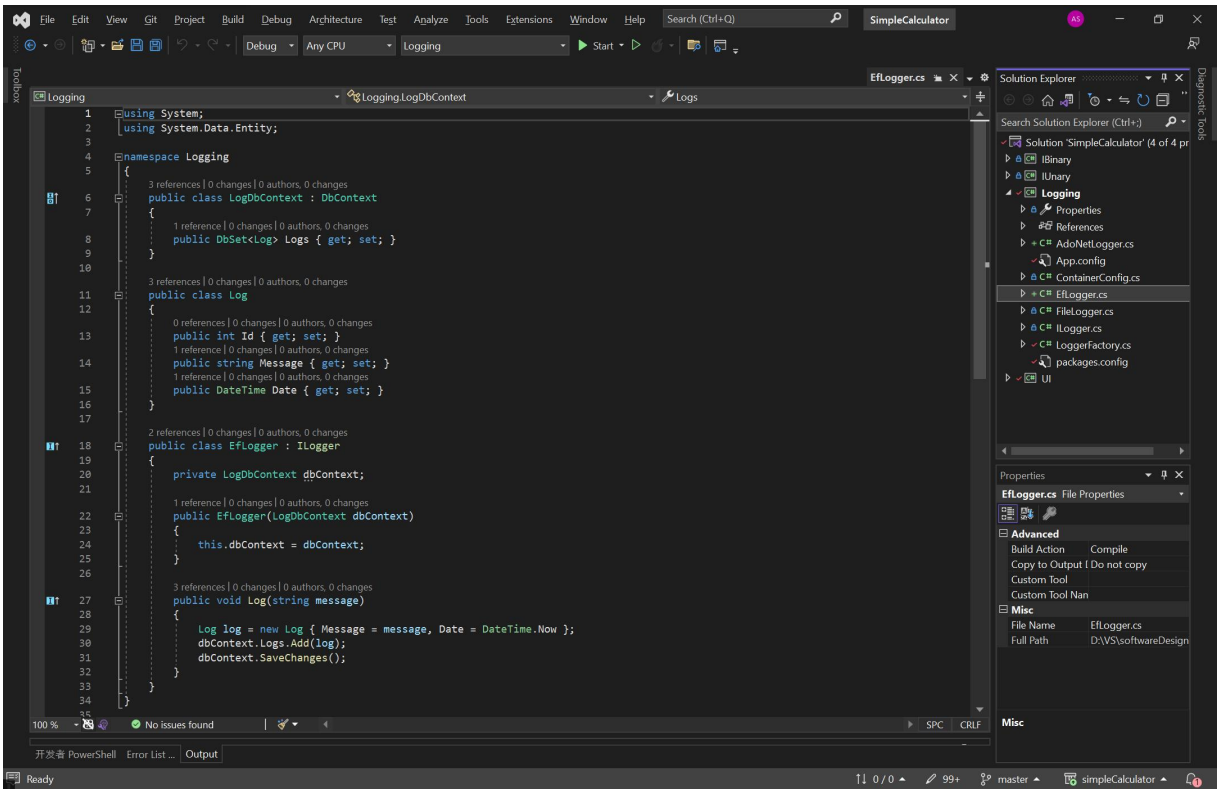
```
        1 reference | arbor, 42 days ago | 1 author, 1 change
        private void del_Click(object sender, EventArgs e)
        {
            if (formation.Text.Length > 0)
            {
                formation.Text = formation.Text.Substring(0, formation.Text.Length - 1);
            }
        }

        1 reference | arbor, 42 days ago | 1 author, 1 change
        private void per_Click(object sender, EventArgs e)
        {
            double num;
            if (double.TryParse(formation.Text, out num))
            {
                num /= 100;
                formation.Text = num.ToString();
            }
        }

        1 reference | arbor, 27 days ago | 1 author, 2 changes
        private void logToolStripMenuItem_Click(object sender, EventArgs e)
        {
            ILogger logger = LoggerFactory.CreateLogger();
            string logContent = GetLogContentFromLogger(logger); // Get log content from logger object

            // Display log information
            MessageBox.Show(logContent, "Log Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        1 reference | arbor, 27 days ago | 1 author, 2 changes
        private string GetLogContentFromLogger(ILogger logger)
        {
            // Read the log file path from the configuration file
            string logFilePath = ConfigurationManager.AppSettings["LogFilePath"];

            // Check if the log file exists
            if (!File.Exists(logFilePath))
            {
                return "Log file does not exist.";
            }

            try
            {
```

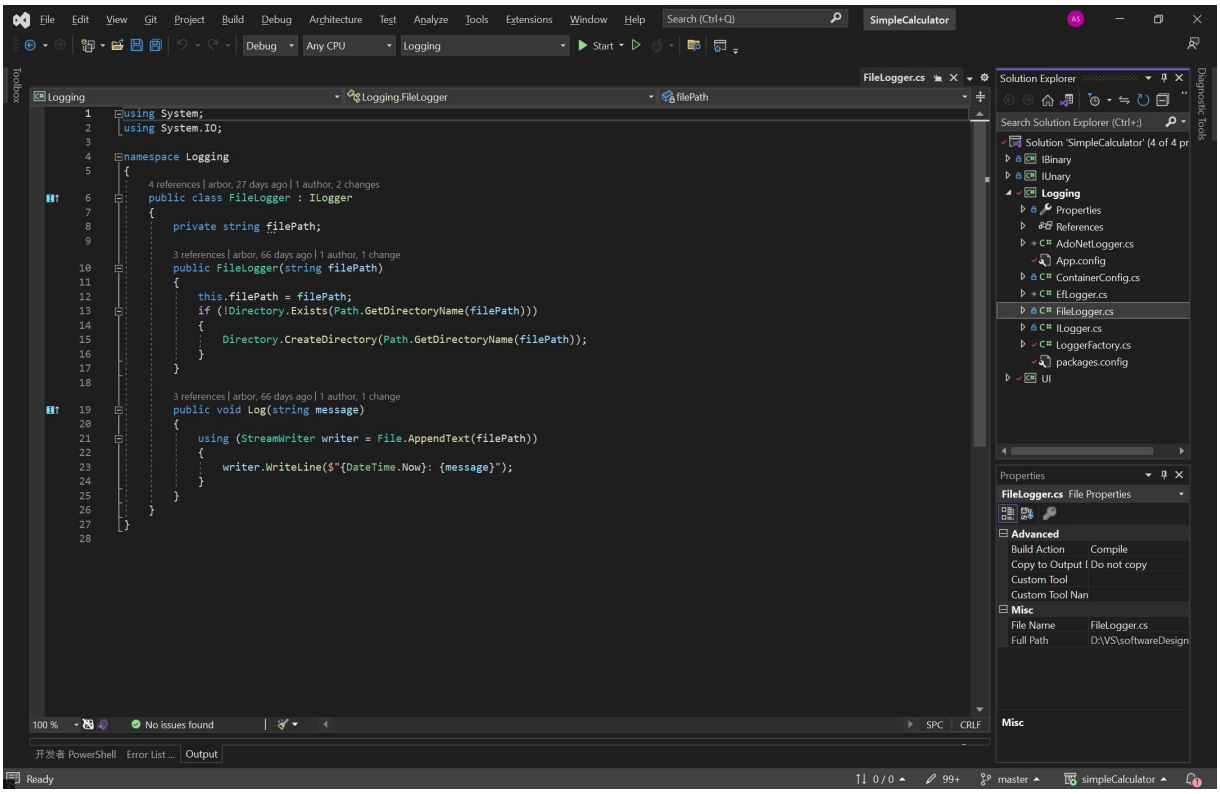


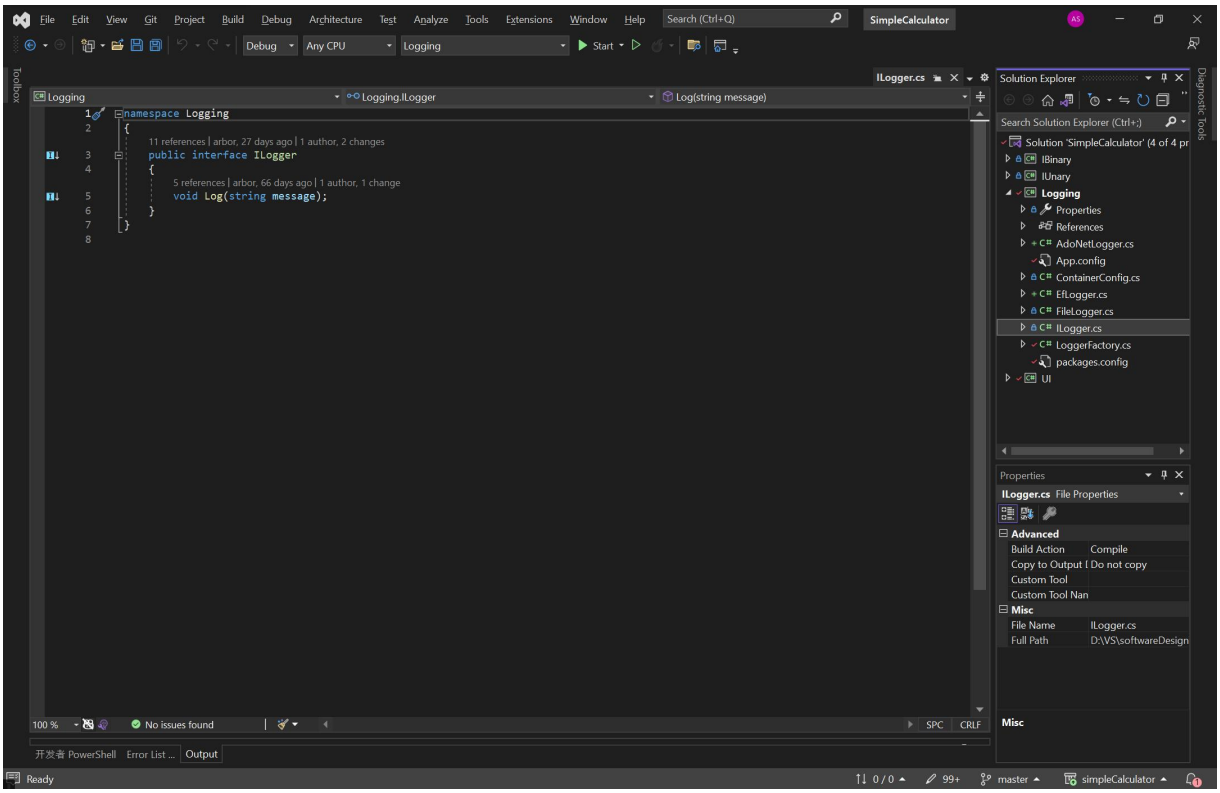
6	实现日志读写模块	将操作日志**写入文本文档、使用 ADO.NET 写入数据库、使用 Simple.Data 和 EF 框架将计算器操作日志写入数据库**
8	在 UI 项目中增加日志读写功能	在 UI 项目中使用日志读写模块实现日志读写功能，使用 IOC（Autofac）实现日志读写对象的实例化

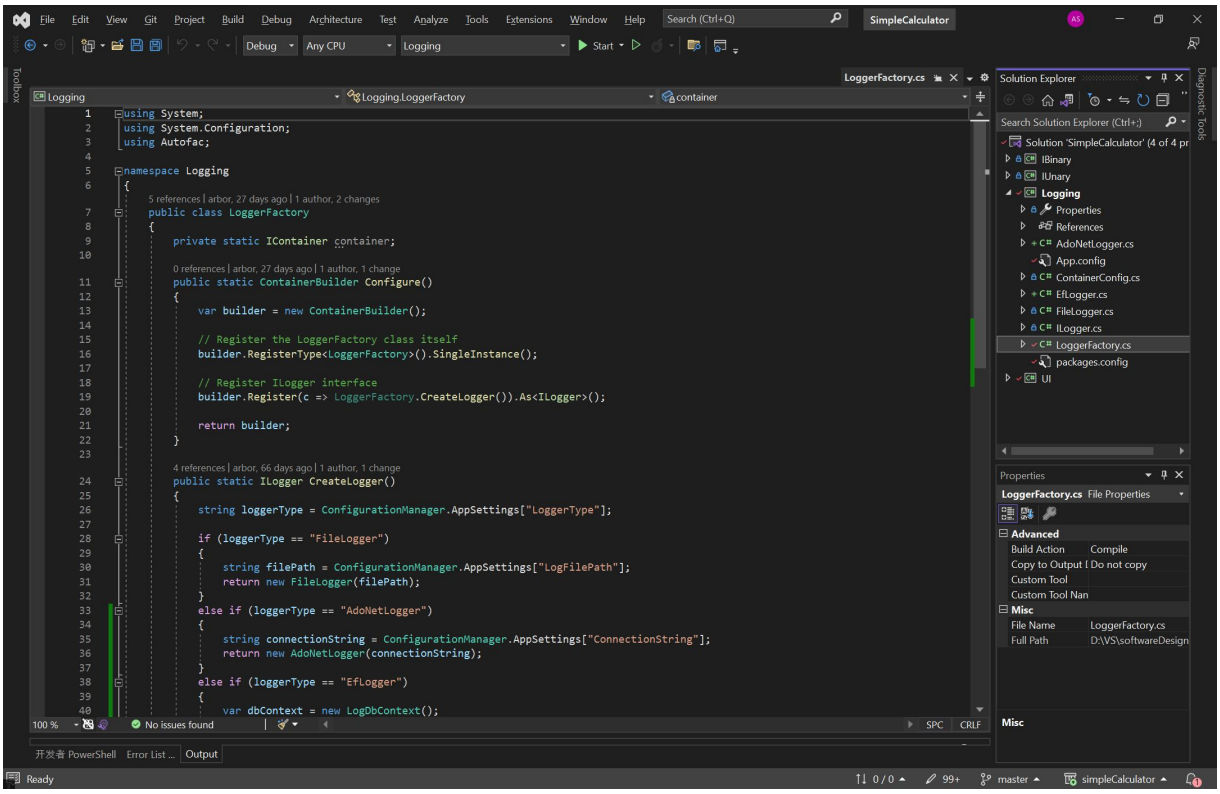


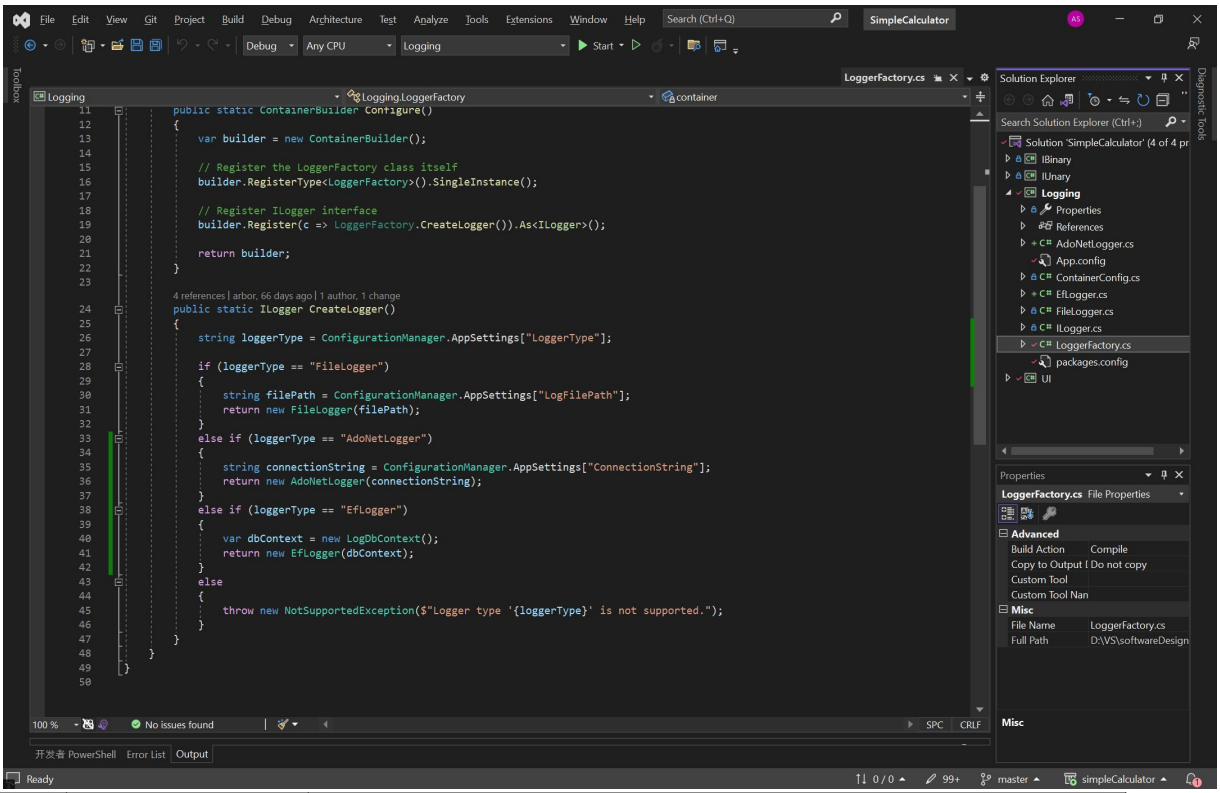

ContainerConfig.cs
Logging
Logging.ContainerConfig
Configure()
using Autofac;
namespace Logging
{
public static class ContainerConfig
{
public static IContainer Configure()
{
var builder = new ContainerBuilder();
builder.Register(c => new FileLogger("log.txt")).As<ILogger>();
return builder.Build();
}
}
}
Solution Explorer
Logging
Properties
References
AdoNetLogger.cs
App.config
ContainerConfig.cs
EfLogger.cs
FileLogger.cs
ILogger.cs
LoggerFactory.cs
packages.config
UI
Output
Debug
'UI.exe' (CLR v4.0.30319: UI.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Xml.Linq\v4.0_4.0.0.0__b77a5c561934e089\System.Xml.Linq.dll'. Skipped loa
'UI.exe' (CLR v4.0.30319: UI.exe): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Runtime.Serialization\v4.0_4.0.0.0__b77a5c561934e089\System.Runtime.Seria
'UI.exe' (CLR v4.0.30319: UI.exe): Loaded 'EntityFrameworkDynamicProxies-EntityFramework'.
The program '[22288] UI.exe: Program Trace' has exited with code 0 (0x0).
The program '[22288] UI.exe' has exited with code 0 (0x0).
Ready


EfLogger.cs
Logging
Logging.LogDbContext
Logs
using System;
using System.Data.Entity;
namespace Logging
{
public class LogDbContext : DbContext
{
public DbSet<Log> Logs { get; set; }
}
public class Log
{
public int Id { get; set; }
public string Message { get; set; }
public DateTime Date { get; set; }
}
public class EfLogger : ILogger
{
private LogDbContext dbContext;
public EfLogger(LogDbContext dbContext)
{
this.dbContext = dbContext;
}
public void Log(string message)
{
Log log = new Log { Message = message, Date = DateTime.Now };
dbContext.Logs.Add(log);
dbContext.SaveChanges();
}
}
}
Solution Explorer
Properties
Ready

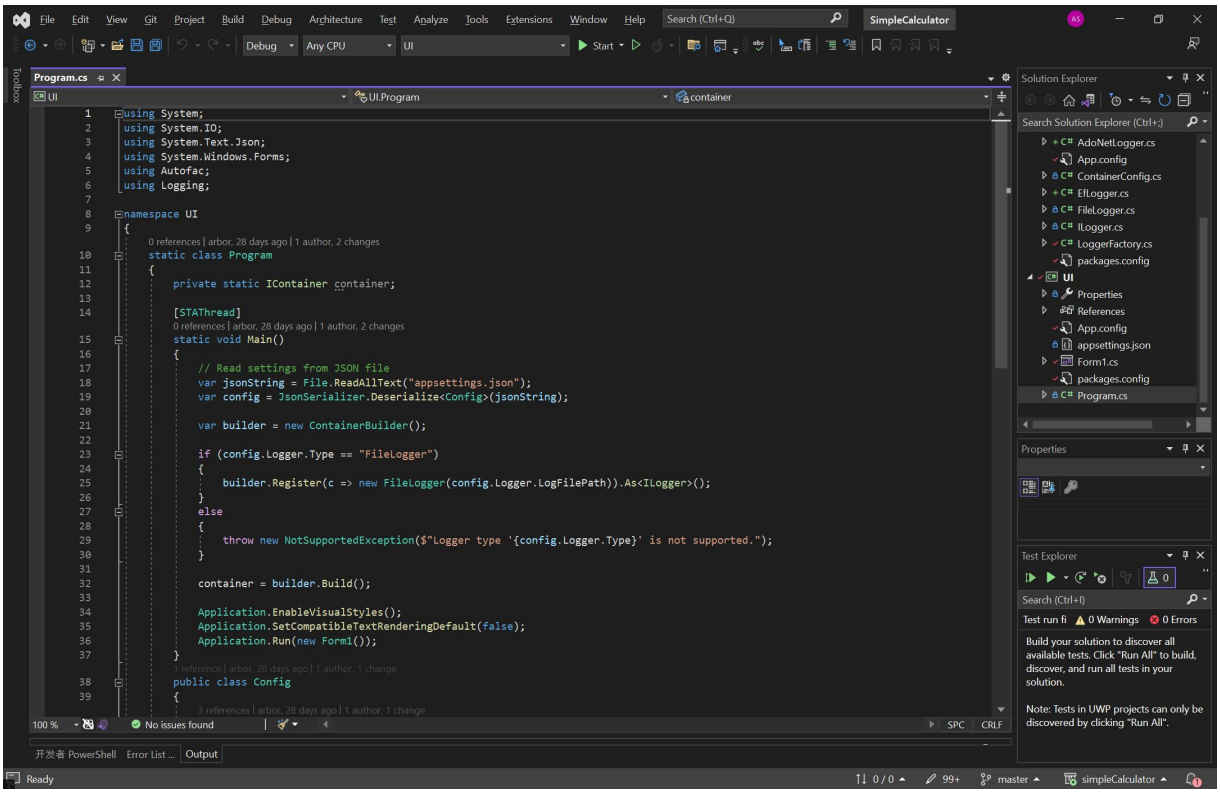

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
Logging
Start
FileLogger.cs
Logging
Logging.FileLogger
filePath
using System;
using System.IO;
namespace Logging
{
4 references | arbor, 27 days ago | 1 author, 2 changes
public class FileLogger : ILogger
{
private string filePath;
3 references | arbor, 66 days ago | 1 author, 1 change
public FileLogger(string filePath)
{
this.filePath = filePath;
if (!Directory.Exists(Path.GetDirectoryName(filePath)))
{
Directory.CreateDirectory(Path.GetDirectoryName(filePath));
}
}
3 references | arbor, 66 days ago | 1 author, 1 change
public void Log(string message)
{
using (StreamWriter writer = File.AppendText(filePath))
{
writer.WriteLine($"{DateTime.Now}: {message}");
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (4 of 4 pr
IBinary
IUnary
Logging
Properties
References
AdoNetLogger.cs
App.config
ContainerConfig.cs
EfLogger.cs
FileLogger.cs
ILogger.cs
LoggerFactory.cs
packages.config
UI
Properties
FileLogger.cs File Properties
Advanced
Build Action Compile
Copy to Output I Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name FileLogger.cs
Full Path D:\VS\softwareDesign
Misc
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
99+
master
simpleCalculator

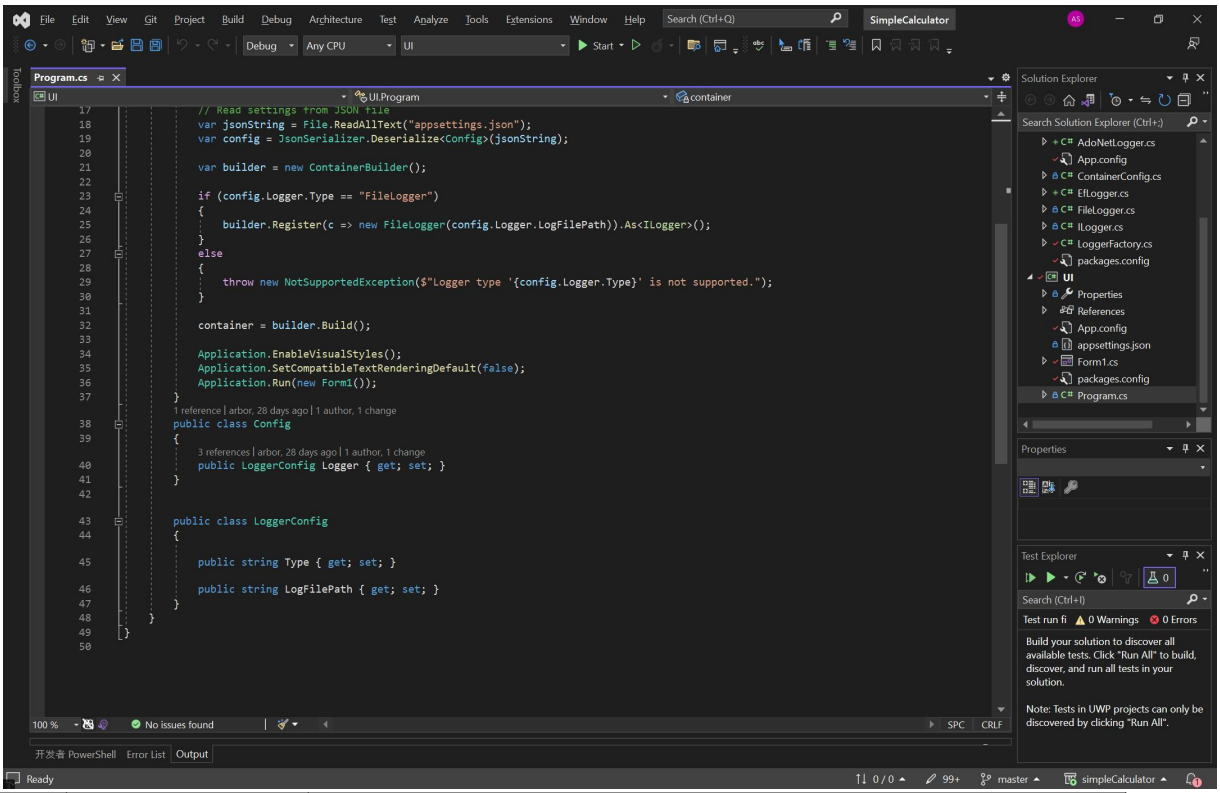

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
Logging
Start
ILogger.cs
Logging
Logging.ILogger
Log(string message)
namespace Logging
{
11 references | arbor, 27 days ago | 1 author, 2 changes
public interface ILogger
{
5 references | arbor, 66 days ago | 1 author, 1 change
void Log(string message);
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (4 of 4 pr
IBinary
IUnary
Logging
Properties
References
AdoNetLogger.cs
App.config
ContainerConfig.cs
EfLogger.cs
FileLogger.cs
ILogger.cs
LoggerFactory.cs
packages.config
UI
Properties
ILogger.cs File Properties
Advanced
Build Action Compile
Copy to Output I Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name ILogger.cs
Full Path D:\VS\softwareDesign
Misc
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
99+
master
simpleCalculator



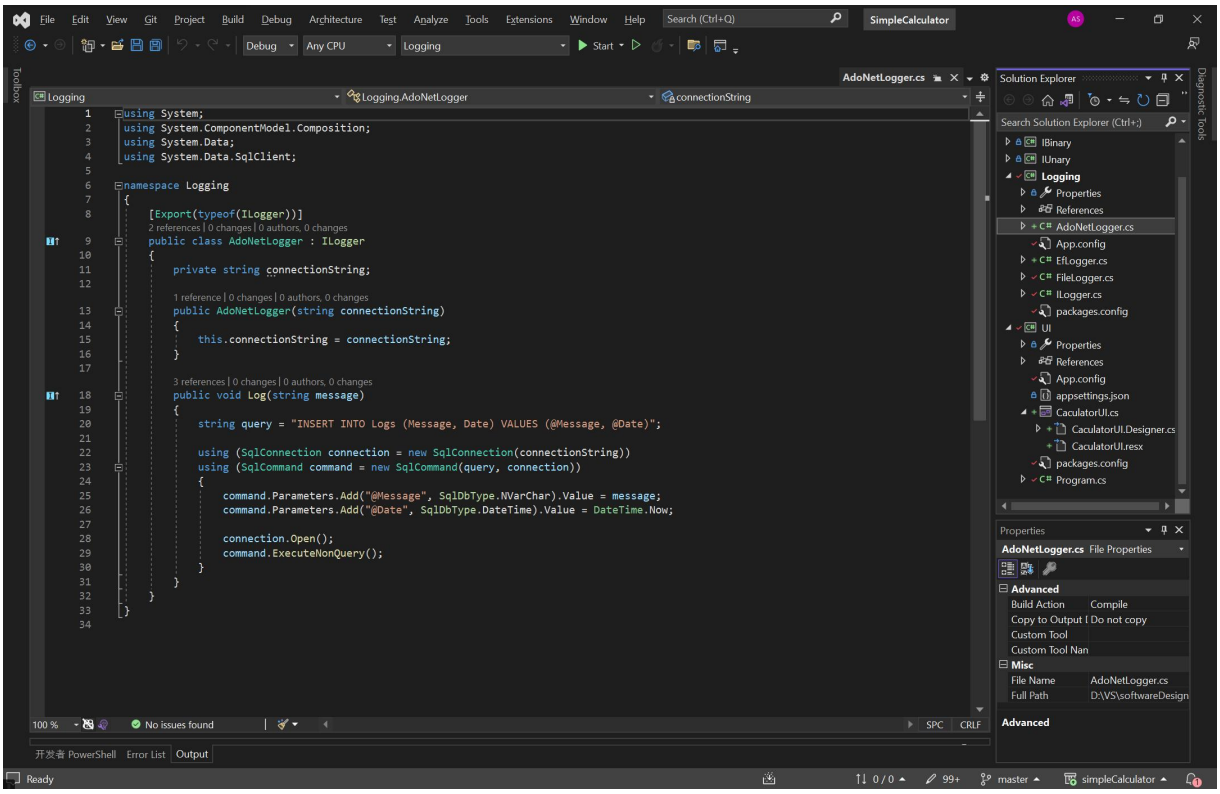


8	在 UI 项目中增加日志读写功能	在 UI 项目中使用日志读写模块实现日志读写功能，使用 IOC（Autofac）实现日志读写对象的实例化

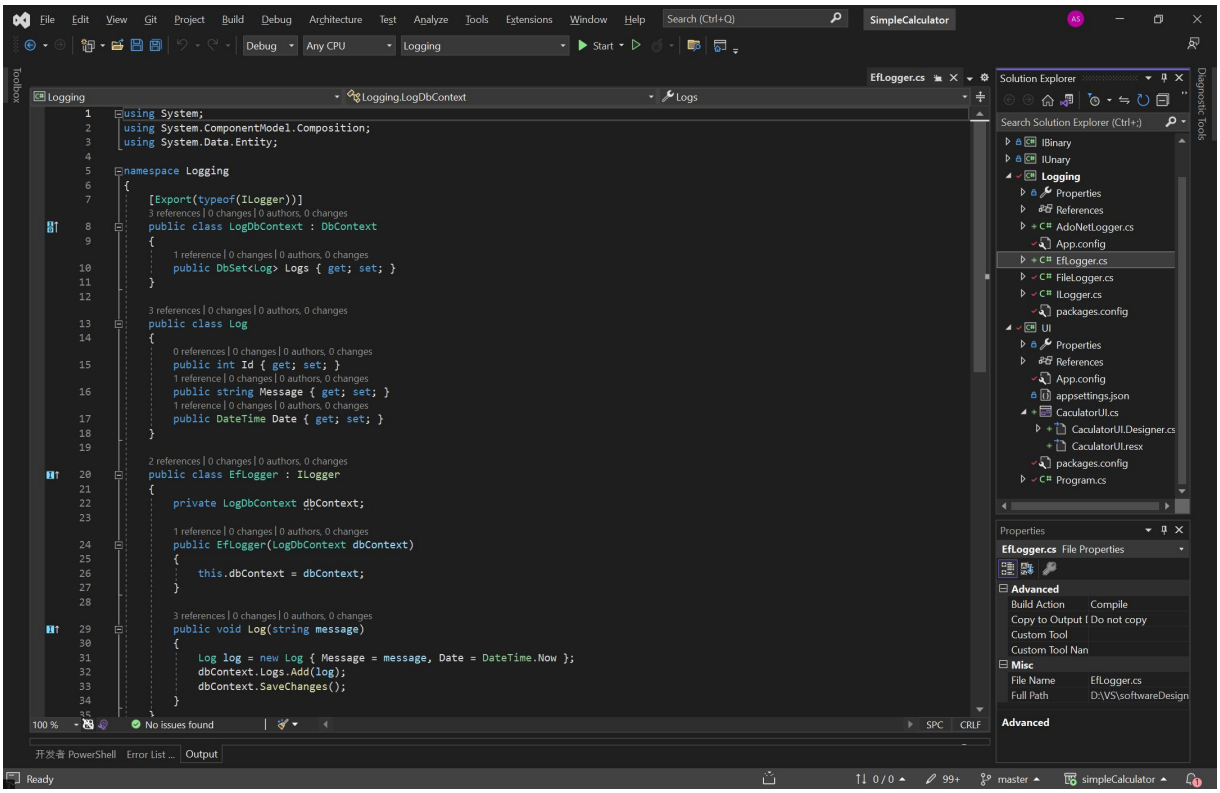




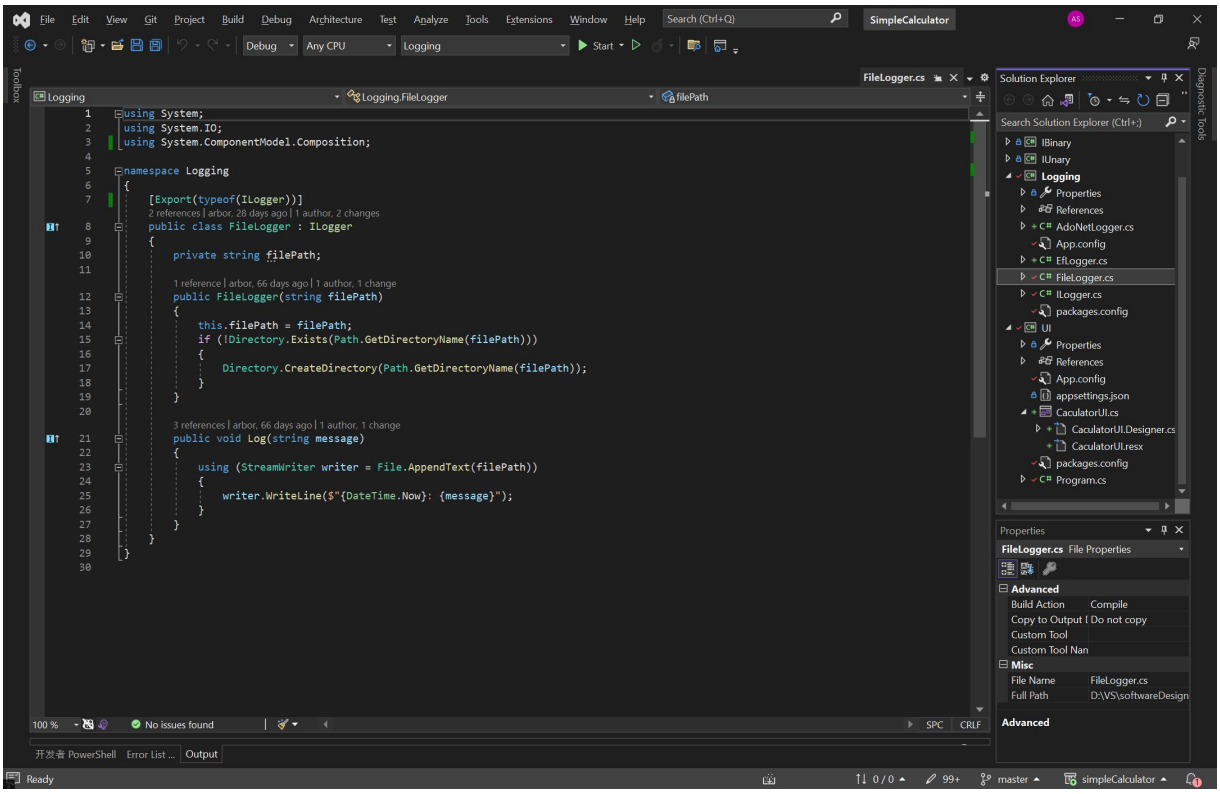
9	MEF 实现日志读写功能	重构 UI 项目和日志读写模块项目，**使用 MEF 实现日志读写功能（类似实验 2 的第 2 部分）**

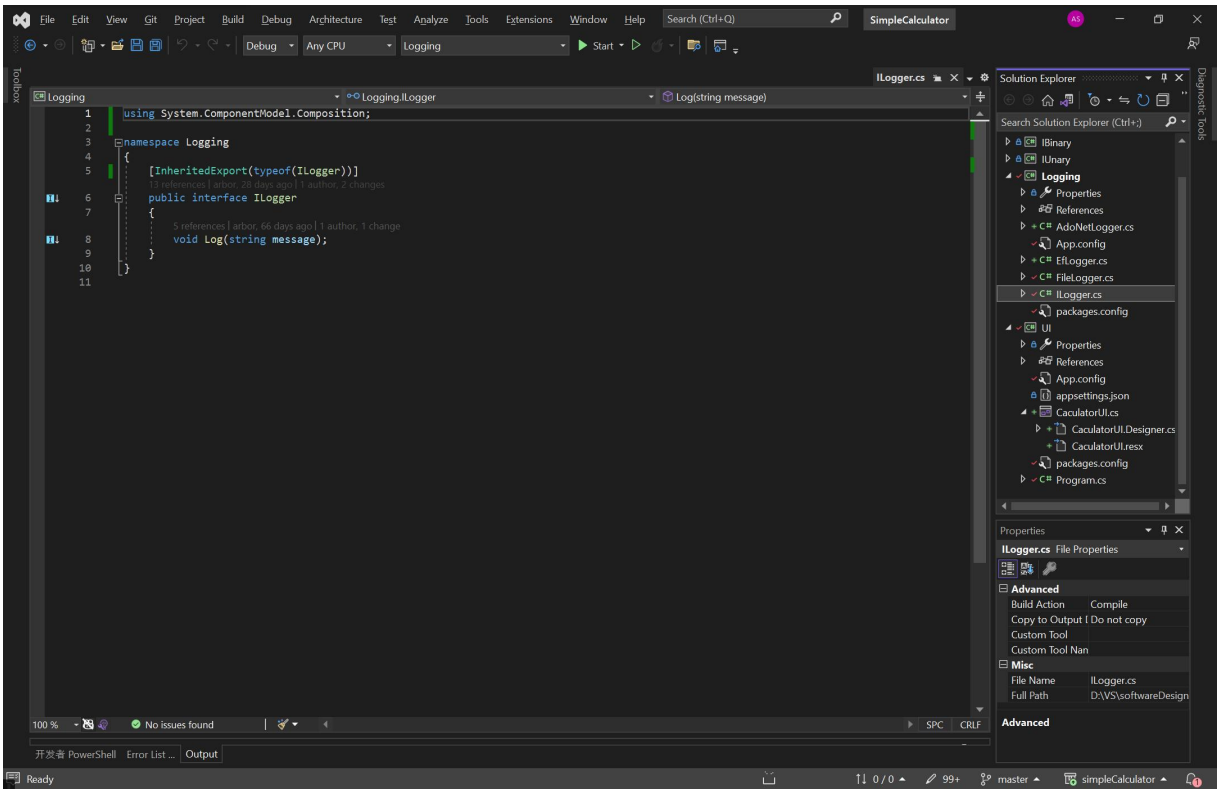


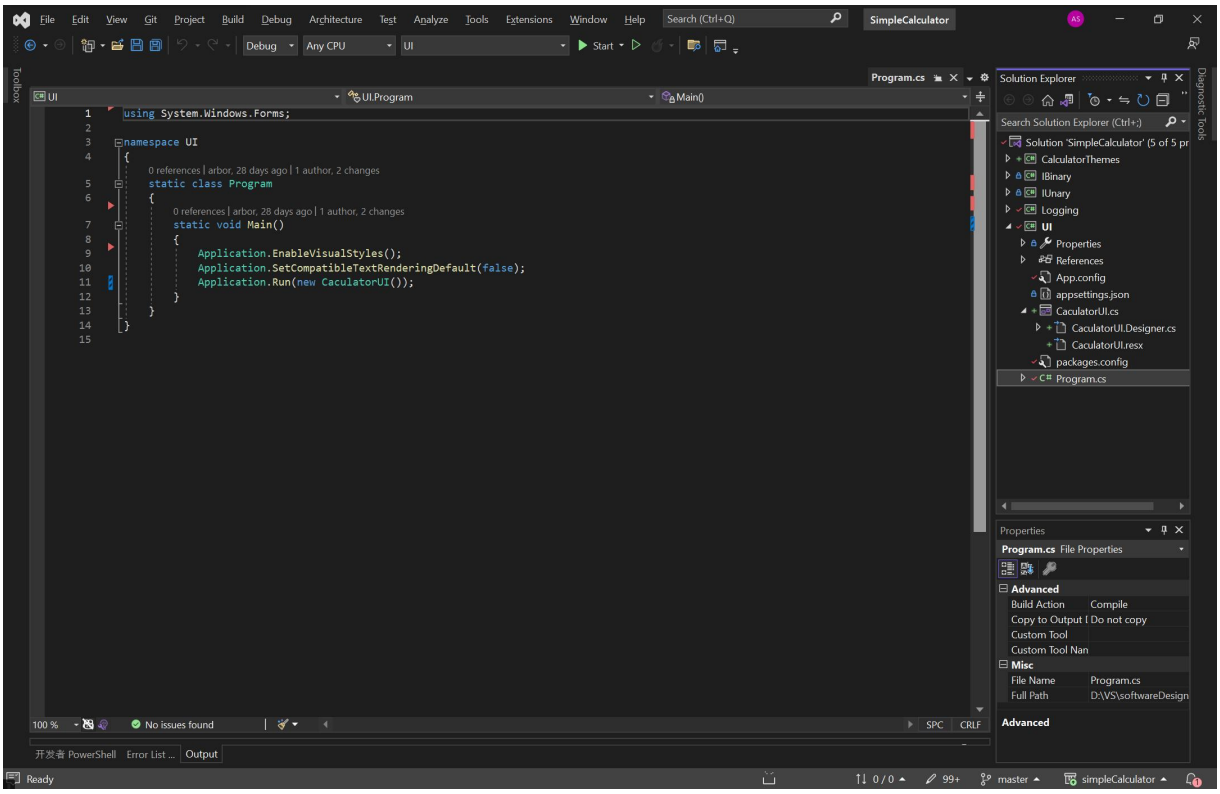
```using System;
using System.ComponentModel.Composition;
using System.Data;
using System.Data.SqlClient;

namespace Logging
{
    [Export(typeof(ILogger))]
    2 references | 0 changes | 0 authors, 0 changes
    public class AdoNetLogger : ILogger
    {
        private string connectionString;

        1 reference | 0 changes | 0 authors, 0 changes
        public AdoNetLogger(string connectionString)
        {
            this.connectionString = connectionString;
        }

        3 references | 0 changes | 0 authors, 0 changes
        public void Log(string message)
        {
            string query = "INSERT INTO Logs (Message, Date) VALUES (@Message, @Date)";

            using (SqlConnection connection = new SqlConnection(connectionString))
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                command.Parameters.Add("@Message", SqlDbType.NVarChar).Value = message;
                command.Parameters.Add("@Date", SqlDbType.DateTime).Value = DateTime.Now;

                connection.Open();
                command.ExecuteNonQuery();
            }
        }
    }
}
```

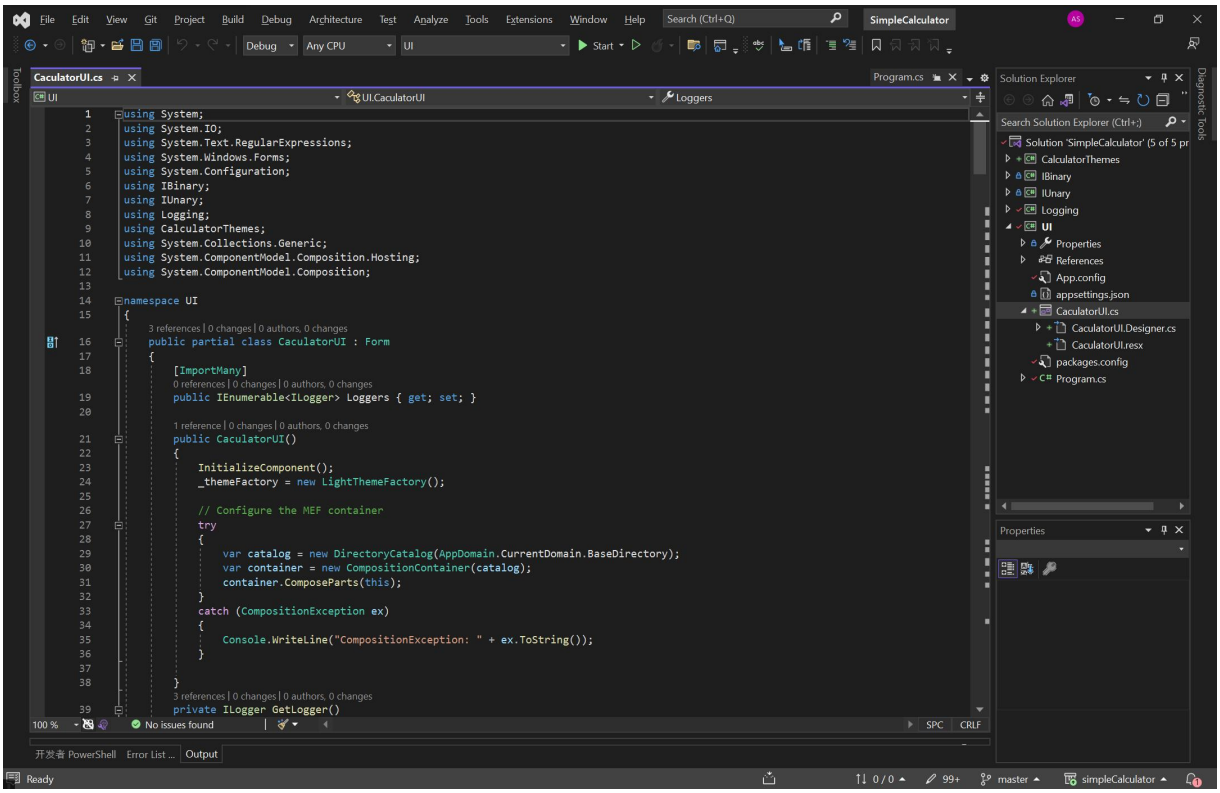


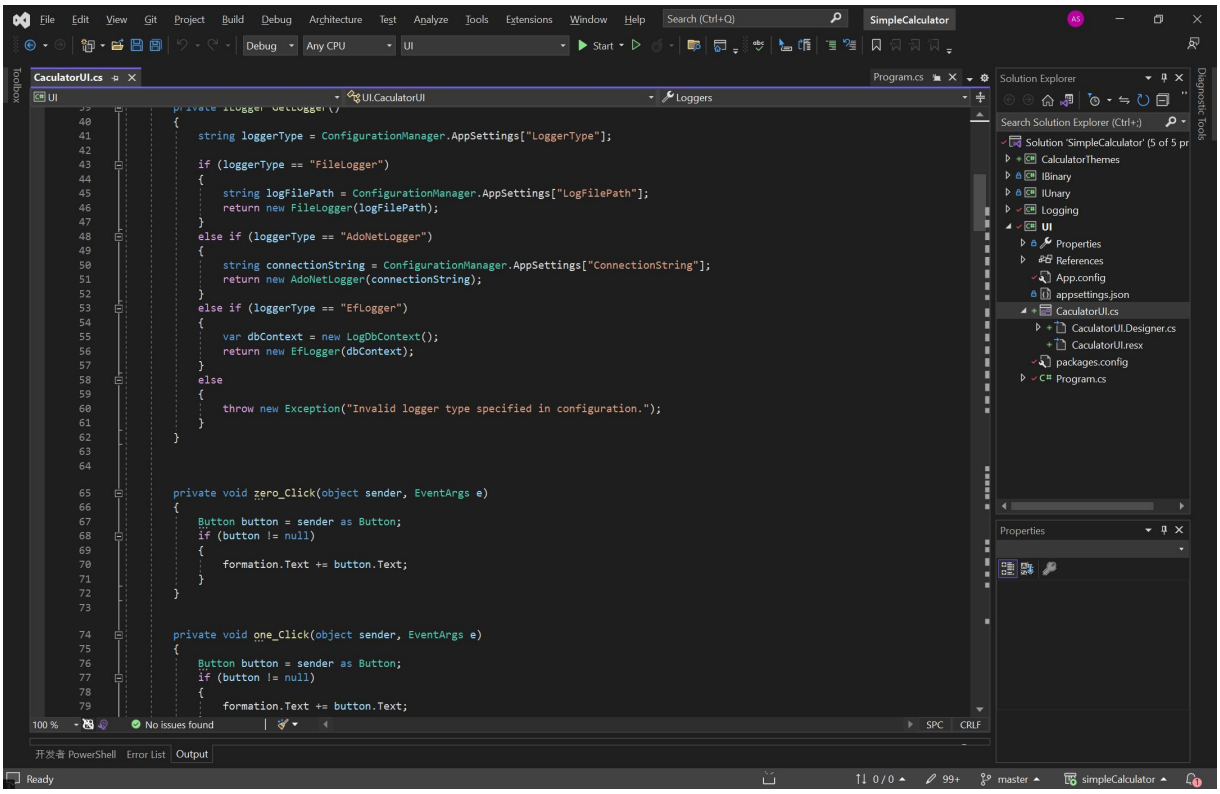
```using System;
using System.ComponentModel.Composition;
using System.Data.Entity;

namespace Logging
{
    [Export(typeof(ILogger))]
    3 references | 0 changes | 0 authors, 0 changes
    public class LogDbContext : DbContext
    {
        1 reference | 0 changes | 0 authors, 0 changes
        public DbSet<Log> Logs { get; set; }
    }

    3 references | 0 changes | 0 authors, 0 changes
    public class Log
    {
        0 references | 0 changes | 0 authors, 0 changes
        public int Id { get; set; }
        1 reference | 0 changes | 0 authors, 0 changes
        public string Message { get; set; }
        1 reference | 0 changes | 0 authors, 0 changes
        public DateTime Date { get; set; }
    }

    2 references | 0 changes | 0 authors, 0 changes
    public class EfLogger : ILogger
    {
        private LogDbContext dbContext;

        1 reference | 0 changes | 0 authors, 0 changes
        public EfLogger(LogDbContext dbContext)
        {
            this.dbContext = dbContext;
        }

        3 references | 0 changes | 0 authors, 0 changes
        public void Log(string message)
        {
            Log log = new Log { Message = message, Date = DateTime.Now };
            dbContext.Logs.Add(log);
            dbContext.SaveChanges();
        }
    }
}
```


File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
Logging
Start
FileLogger.cs
Logging
Logging.FileLogger
filePath
using System;
using System.IO;
using System.ComponentModel.Composition;
namespace Logging
{
[Export(typeof(ILogger))]
2 references | arbor, 28 days ago | 1 author, 2 changes
public class FileLogger : ILogger
{
private string filePath;
1 reference | arbor, 66 days ago | 1 author, 1 change
public FileLogger(string filePath)
{
this.filePath = filePath;
if (!Directory.Exists(Path.GetDirectoryName(filePath)))
{
Directory.CreateDirectory(Path.GetDirectoryName(filePath));
}
}
3 references | arbor, 66 days ago | 1 author, 1 change
public void Log(string message)
{
using (StreamWriter writer = File.AppendText(filePath))
{
writer.WriteLine($"{DateTime.Now}: {message}");
}
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
IBinary
IUnary
Logging
Properties
References
AdoNetLogger.cs
App.config
EfLogger.cs
FileLogger.cs
ILogger.cs
packages.config
UI
Properties
References
App.config
appsettings.json
CaculatorUI.cs
CaculatorUI.Designer.cs
CaculatorUI.resx
packages.config
Program.cs
Properties
FileLogger.cs File Properties
Advanced
Build Action Compile
Copy to Output I Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name FileLogger.cs
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
99+
master
simpleCalculator

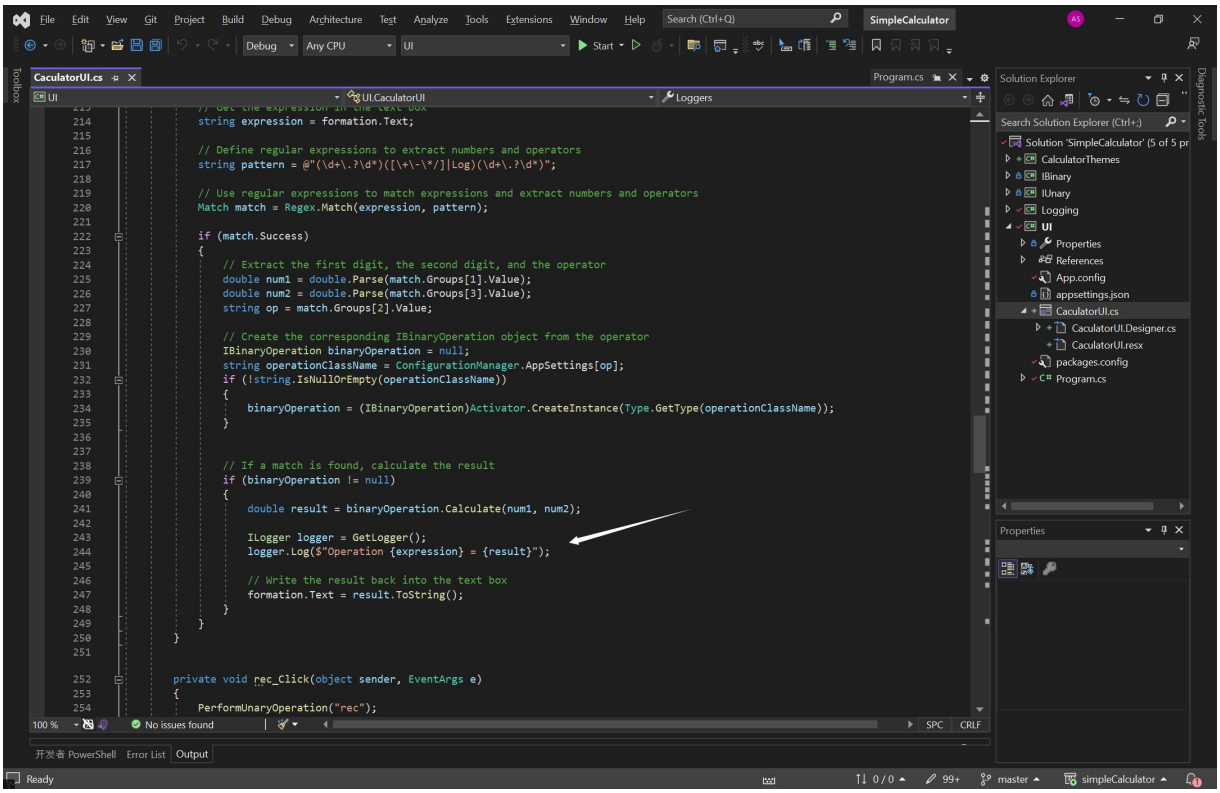

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SimpleCalculator
Debug
Any CPU
Logging
Start
ILogger.cs
Logging
Logging.ILogger
Log(string message)
using System.ComponentModel.Composition;
namespace Logging
{
[InheritedExport(typeof(ILogger))]
13 references | arbor, 28 days ago | 1 author, 2 changes
public interface ILogger
{
5 references | arbor, 66 days ago | 1 author, 1 change
void Log(string message);
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
IBinary
IUnary
Logging
Properties
References
AdoNetLogger.cs
App.config
EfLogger.cs
FileLogger.cs
ILogger.cs
packages.config
UI
Properties
References
App.config
appsettings.json
CaculatorUI.cs
CaculatorUI.Designer.cs
CaculatorUI.resx
packages.config
Program.cs
Properties
ILogger.cs File Properties
Advanced
Build Action Compile
Copy to Output I Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name ILogger.cs
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
99+
master
simpleCalculator


File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
SimpleCalculator
Debug
Any CPU
UI
Start
Program.cs
UI.Program
Main()
using System.Windows.Forms;
namespace UI
{
0 references | arbor, 28 days ago | 1 author, 2 changes
static class Program
{
0 references | arbor, 28 days ago | 1 author, 2 changes
static void Main()
{
Application.EnableVisualStyles();
Application.SetCompatibleTextRenderingDefault(false);
Application.Run(new CaculatorUI());
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (5 of 5 pr
CalculatorThemes
IBinary
IUnary
Logging
UI
Properties
References
App.config
appsettings.json
CaculatorUI.cs
CaculatorUI.Designer.cs
CaculatorUI.resx
packages.config
Program.cs
Properties
Program.cs File Properties
Advanced
Build Action Compile
Copy to Output I Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name Program.cs
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
99+
master
simpleCalculator

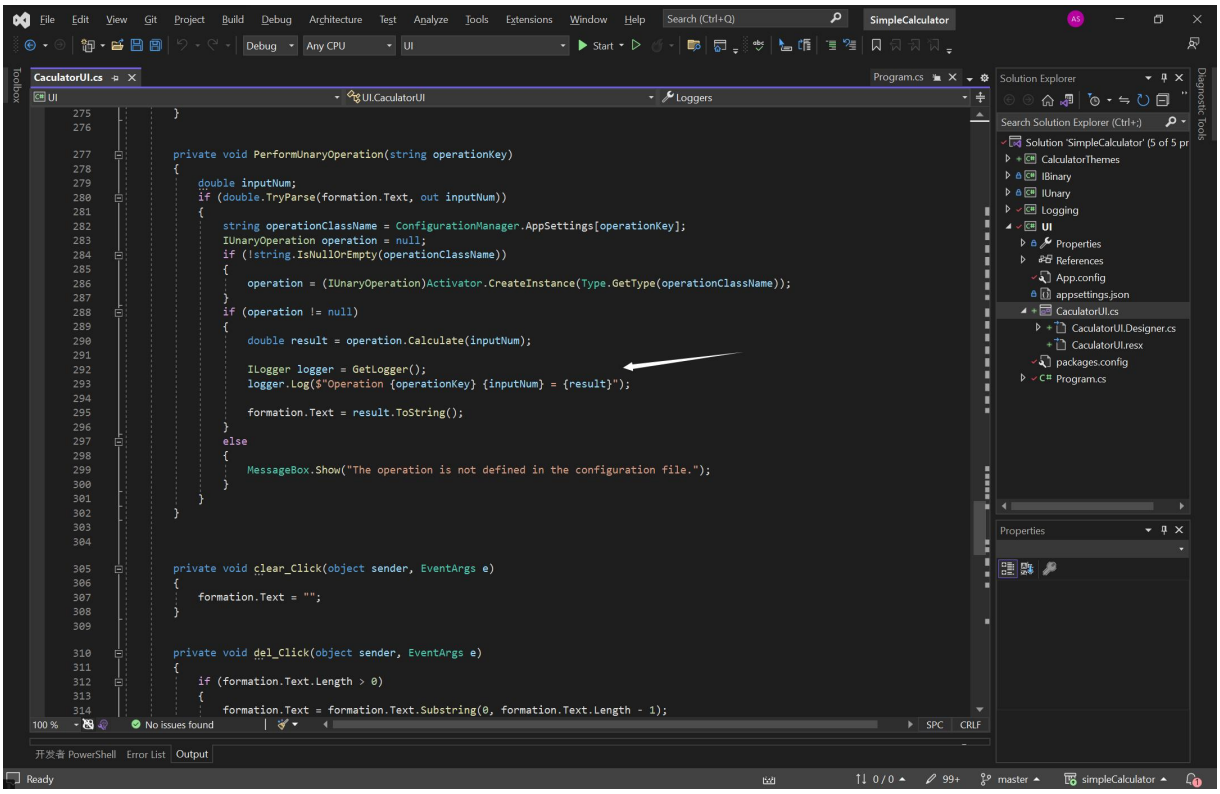

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
SimpleCalculator
Debug
Any CPU
UI
Start
CaculatorUI.cs
Program.cs
UI
UI.CaculatorUI
Loggers
using System;
using System.IO;
using System.Text.RegularExpressions;
using System.Windows.Forms;
using System.Configuration;
using IBinary;
using IUnary;
using Logging;
using CalculatorThemes;
using System.Collections.Generic;
using System.ComponentModel.Composition.Hosting;
using System.ComponentModel.Composition;
namespace UI
{
3 references | 0 changes | 0 authors, 0 changes
public partial class CaculatorUI : Form
{
[ImportMany]
0 references | 0 changes | 0 authors, 0 changes
public IEnumerable<ILogger> Loggers { get; set; }
1 reference | 0 changes | 0 authors, 0 changes
public CaculatorUI()
{
InitializeComponent();
_themeFactory = new LightThemeFactory();
// Configure the MEF container
try
{
var catalog = new DirectoryCatalog(AppDomain.CurrentDomain.BaseDirectory);
var container = new CompositionContainer(catalog);
container.ComposeParts(this);
}
catch (CompositionException ex)
{
Console.WriteLine("CompositionException: " + ex.ToString());
}
}
3 references | 0 changes | 0 authors, 0 changes
private ILogger GetLogger()
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (5 of 5 pr
CalculatorThemes
IBinary
IUnary
Logging
UI
Properties
References
App.config
appsettings.json
CaculatorUI.cs
CaculatorUI.Designer.cs
CaculatorUI.resx
packages.config
Program.cs
Properties
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
99+
master
simpleCalculator



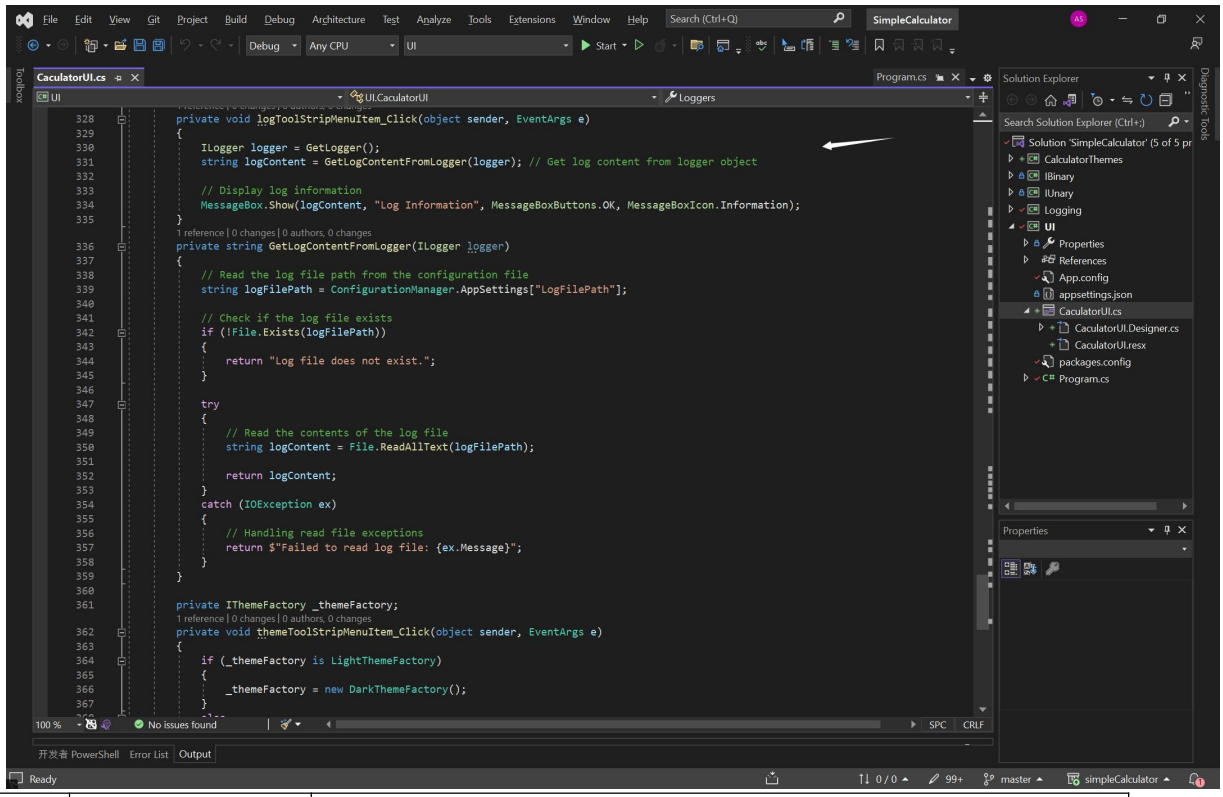
```
        {
            string loggerType = ConfigurationManager.AppSettings["LoggerType"];

            if (loggerType == "FileLogger")
            {
                string logFilePath = ConfigurationManager.AppSettings["LogFilePath"];
                return new FileLogger(logFilePath);
            }
            else if (loggerType == "AdoNetLogger")
            {
                string connectionString = ConfigurationManager.AppSettings["ConnectionString"];
                return new AdoNetLogger(connectionString);
            }
            else if (loggerType == "EfLogger")
            {
                var dbContext = new LogDbContext();
                return new EfLogger(dbContext);
            }
            else
            {
                throw new Exception("Invalid logger type specified in configuration.");
            }
        }


        private void zero_Click(object sender, EventArgs e)
        {
            Button button = sender as Button;
            if (button != null)
            {
                formation.Text += button.Text;
            }
        }


        private void one_Click(object sender, EventArgs e)
        {
            Button button = sender as Button;
            if (button != null)
            {
                formation.Text += button.Text;
```



```
            string expression = formation.Text;

            // Define regular expressions to extract numbers and operators
            string pattern = @"(\d+\.?\d*)([\+\-\*/]|Log)(\d+\.?\d*)";

            // Use regular expressions to match expressions and extract numbers and operators
            Match match = Regex.Match(expression, pattern);

            if (match.Success)
            {
                // Extract the first digit, the second digit, and the operator
                double num1 = double.Parse(match.Groups[1].Value);
                double num2 = double.Parse(match.Groups[3].Value);
                string op = match.Groups[2].Value;

                // Create the corresponding IBinaryOperation object from the operator
                IBinaryOperation binaryOperation = null;
                string operationClassName = ConfigurationManager.AppSettings[op];
                if (!string.IsNullOrEmpty(operationClassName))
                {
                    binaryOperation = (IBinaryOperation)Activator.CreateInstance(Type.GetType(operationClassName));
                }


                // If a match is found, calculate the result
                if (binaryOperation != null)
                {
                    double result = binaryOperation.Calculate(num1, num2);

                    ILogger logger = GetLogger();
                    logger.Log($"Operation {expression} = {result}");

                    // Write the result back into the text box
                    formation.Text = result.ToString();
                }
            }
        }


        private void rec_Click(object sender, EventArgs e)
        {
            PerformUnaryOperation("rec");
```

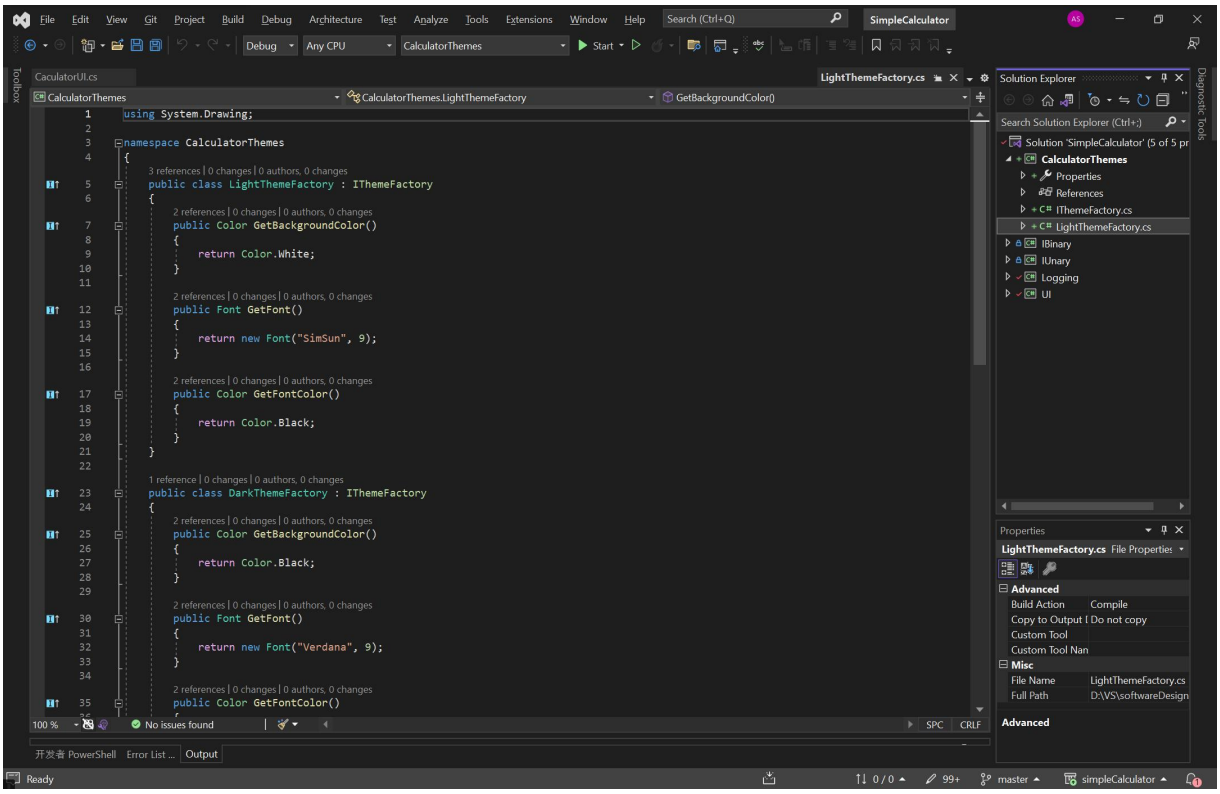


```
        }

        private void PerformUnaryOperation(string operationKey)
        {
            double inputNum;
            if (double.TryParse(formation.Text, out inputNum))
            {
                string operationClassName = ConfigurationManager.AppSettings[operationKey];
                IUnaryOperation operation = null;
                if (!string.IsNullOrEmpty(operationClassName))
                {
                    operation = (IUnaryOperation)Activator.CreateInstance(Type.GetType(operationClassName));
                }
                if (operation != null)
                {
                    double result = operation.Calculate(inputNum);

                    ILogger logger = GetLogger();
                    logger.Log($"Operation {operationKey} {inputNum} = {result}");

                    formation.Text = result.ToString();
                }
                else
                {
                    MessageBox.Show("The operation is not defined in the configuration file.");
                }
            }
        }


        private void clear_Click(object sender, EventArgs e)
        {
            formation.Text = "";
        }


        private void del_Click(object sender, EventArgs e)
        {
            if (formation.Text.Length > 0)
            {
                formation.Text = formation.Text.Substring(0, formation.Text.Length - 1);
```

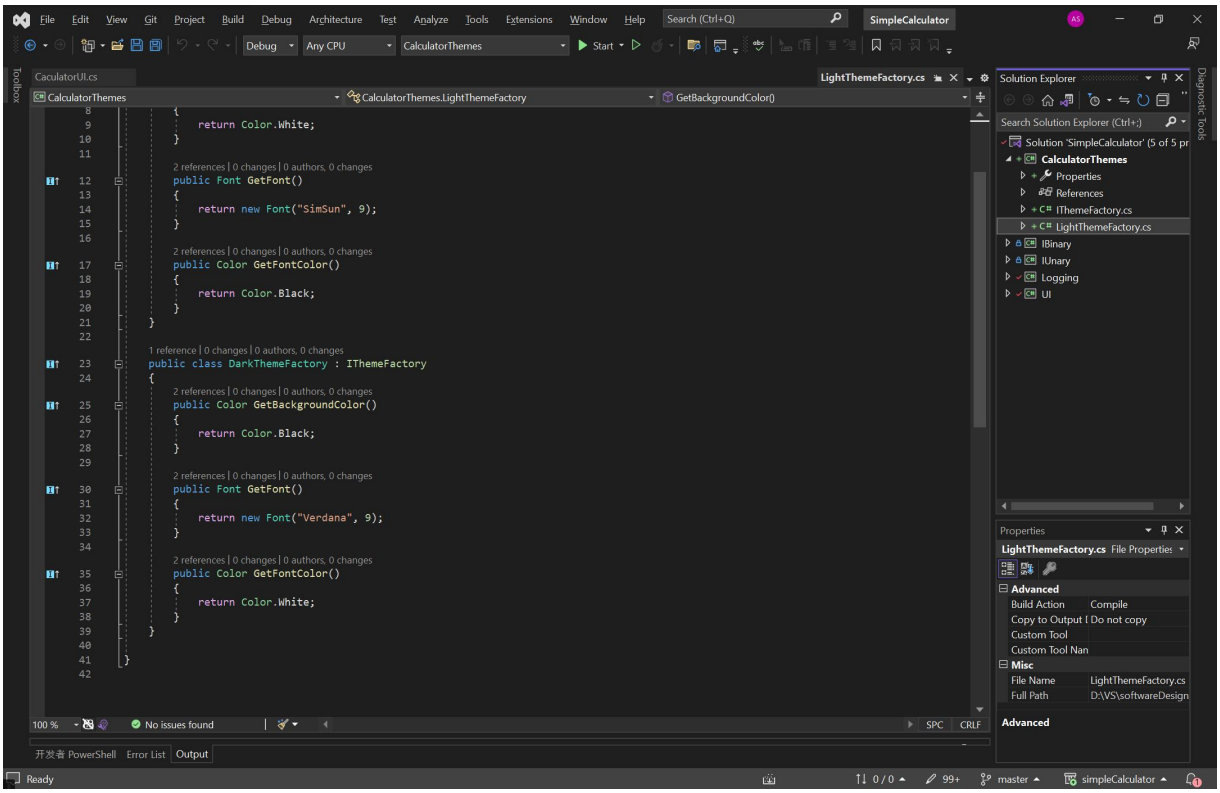


```
        private void logToolStripMenuItem_Click(object sender, EventArgs e)
        {
            ILogger logger = GetLogger();
            string logContent = GetLogContentFromLogger(logger); // Get log content from logger object

            // Display log information
            MessageBox.Show(logContent, "Log Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        1 reference | 0 changes | 0 authors, 0 changes
        private string GetLogContentFromLogger(ILogger logger)
        {
            // Read the log file path from the configuration file
            string logFilePath = ConfigurationManager.AppSettings["LogFilePath"];

            // Check if the log file exists
            if (!File.Exists(logFilePath))
            {
                return "Log file does not exist.";
            }

            try
            {
                // Read the contents of the log file
                string logContent = File.ReadAllText(logFilePath);

                return logContent;
            }
            catch (IOException ex)
            {
                // Handling read file exceptions
                return $"Failed to read log file: {ex.Message}";
            }
        }

        private IThemeFactory _themeFactory;
        1 reference | 0 changes | 0 authors, 0 changes
        private void themeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (_themeFactory is LightThemeFactory)
            {
                _themeFactory = new DarkThemeFactory();
            }
```

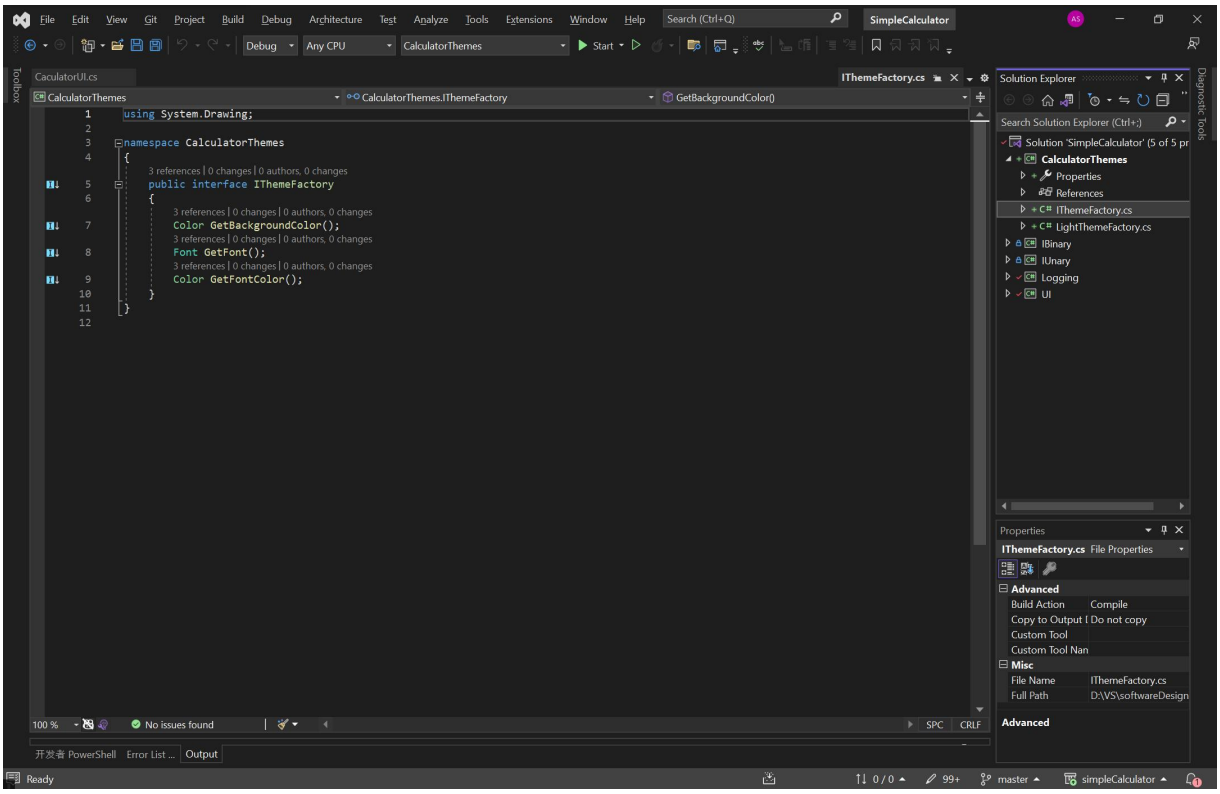
10	实现计算器主题设置和切换	采用**抽象工厂模式**设计计算器主题功能，定义至少两种主题，每种主题下，计算器的特性是不同的，比如背景图，字体颜色，字体等不相同。

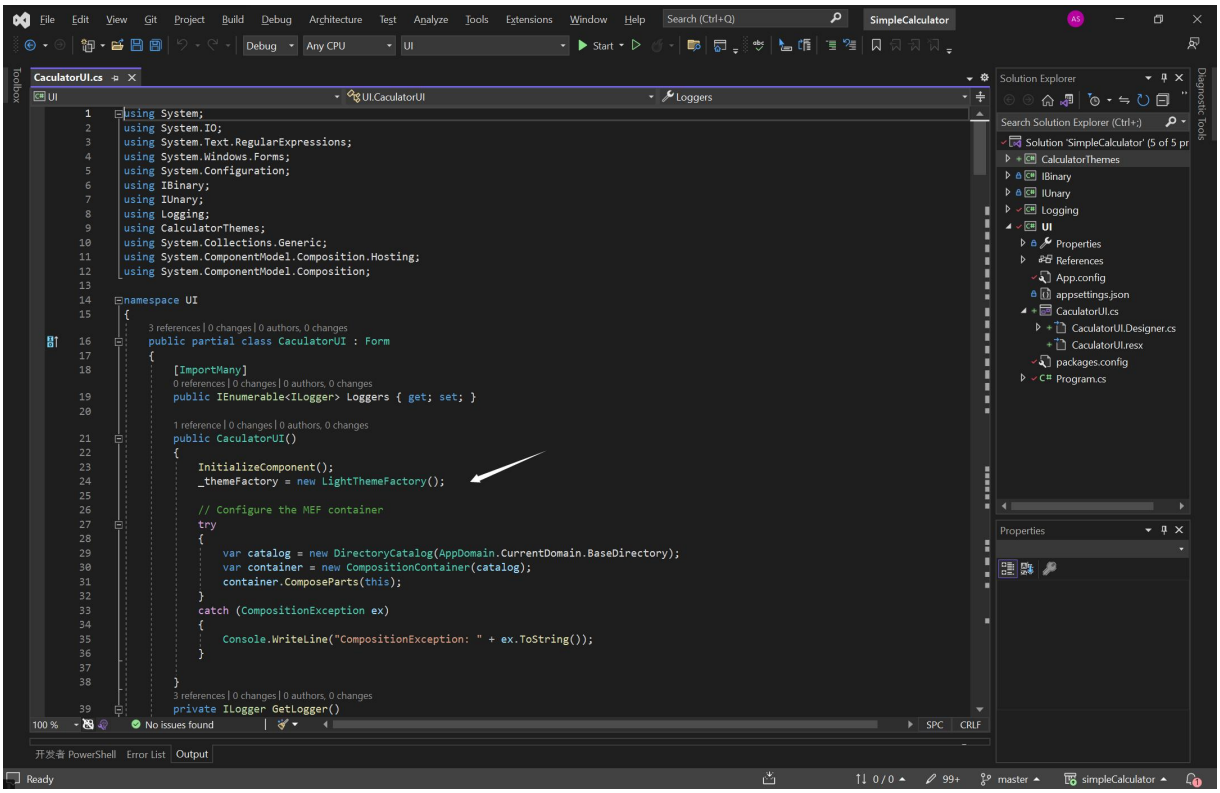


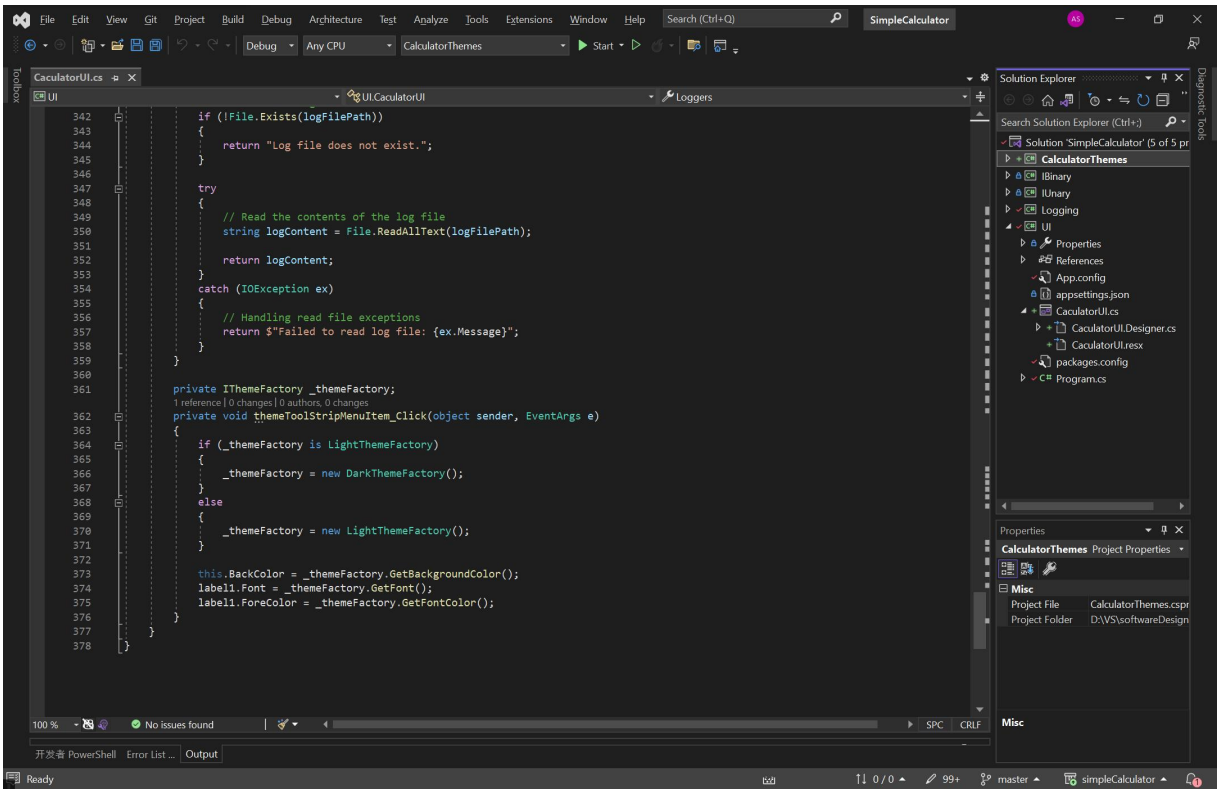
```
using System.Drawing;

namespace CalculatorThemes
{
    public class LightThemeFactory : IThemeFactory
    {
        public Color GetBackgroundColor()
        {
            return Color.White;
        }

        public Font GetFont()
        {
            return new Font("SimSun", 9);
        }

        public Color GetFontColor()
        {
            return Color.Black;
        }
    }

    public class DarkThemeFactory : IThemeFactory
    {
        public Color GetBackgroundColor()
        {
            return Color.Black;
        }

        public Font GetFont()
        {
            return new Font("Verdana", 9);
        }

        public Color GetFontColor()
        {
```



```
        {
            return Color.White;
        }

        public Font GetFont()
        {
            return new Font("SimSun", 9);
        }

        public Color GetFontColor()
        {
            return Color.Black;
        }
    }

    public class DarkThemeFactory : IThemeFactory
    {
        public Color GetBackgroundColor()
        {
            return Color.Black;
        }

        public Font GetFont()
        {
            return new Font("Verdana", 9);
        }

        public Color GetFontColor()
        {
            return Color.White;
        }
    }
}
```

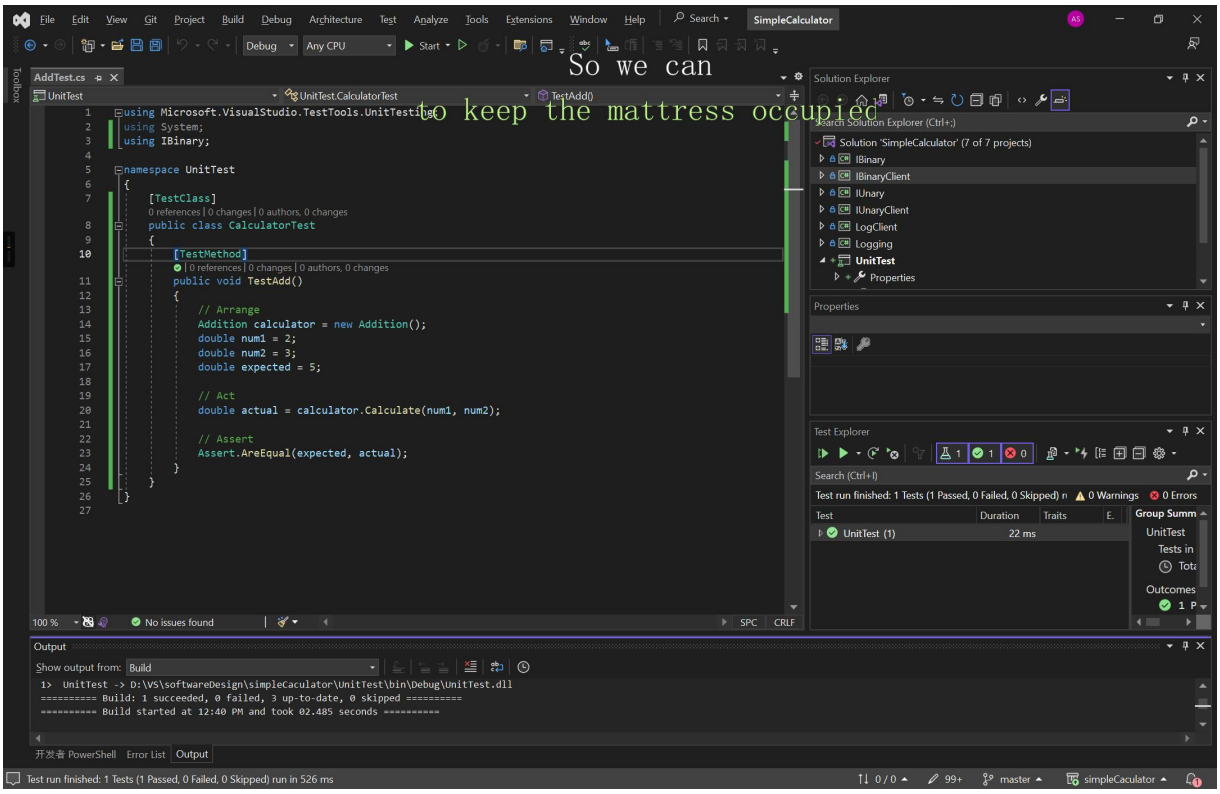

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
SimpleCalculator
Debug
Any CPU
CalculatorThemes
Start
IThemeFactory.cs
CalculatorThemes.IThemeFactory
GetBackgroundColor()
using System.Drawing;
namespace CalculatorThemes
{
3 references | 0 changes | 0 authors, 0 changes
public interface IThemeFactory
{
3 references | 0 changes | 0 authors, 0 changes
Color GetBackgroundColor();
3 references | 0 changes | 0 authors, 0 changes
Font GetFont();
3 references | 0 changes | 0 authors, 0 changes
Color GetFontColor();
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (5 of 5 pr
CalculatorThemes
Properties
References
IThemeFactory.cs
LightThemeFactory.cs
IBinary
IUnary
Logging
UI
Properties
IThemeFactory.cs File Properties
Advanced
Build Action Compile
Copy to Output I Do not copy
Custom Tool
Custom Tool Nan
Misc
File Name IThemeFactory.cs
Full Path D:\VS\softwareDesign
Advanced
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List ...
Output
Ready
0 / 0
99+
master
simpleCalculator

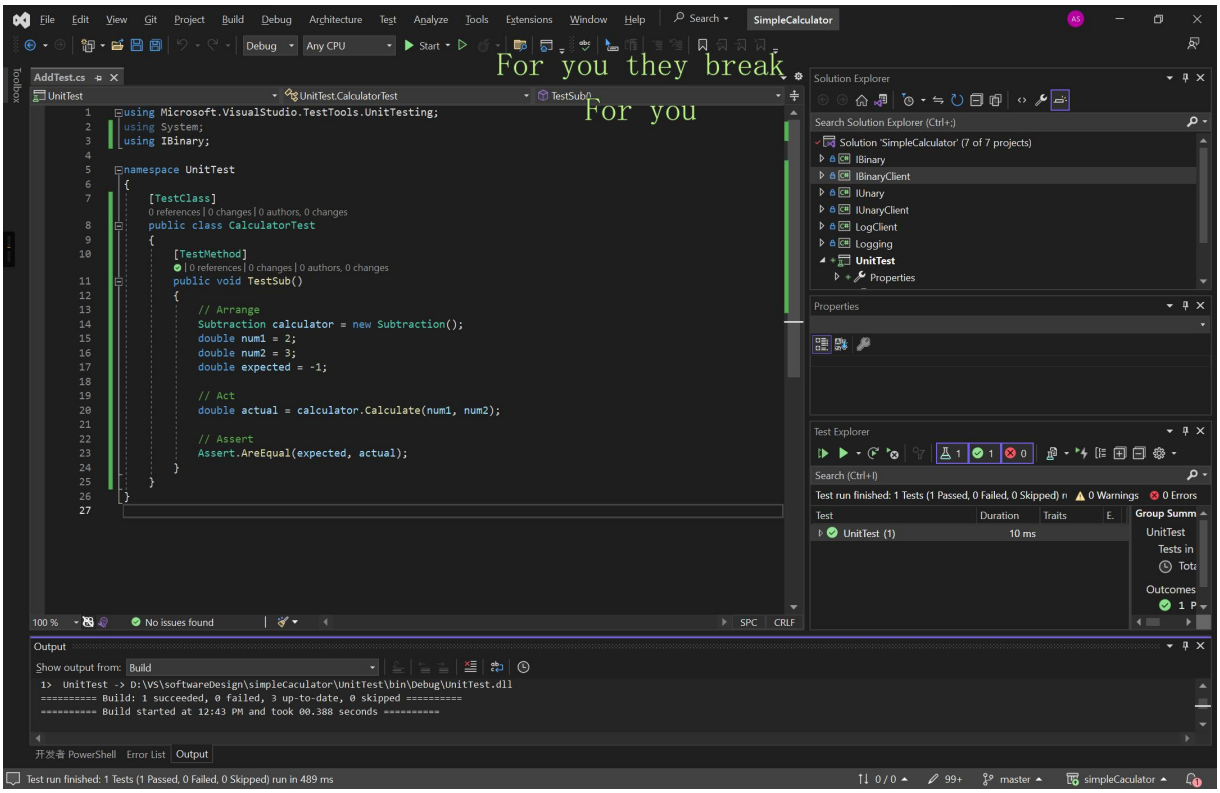

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
SimpleCalculator
Debug
Any CPU
UI
Start
CaculatorUI.cs
UI
UI.CaculatorUI
Loggers
using System;
using System.IO;
using System.Text.RegularExpressions;
using System.Windows.Forms;
using System.Configuration;
using IBinary;
using IUnary;
using Logging;
using CalculatorThemes;
using System.Collections.Generic;
using System.ComponentModel.Composition.Hosting;
using System.ComponentModel.Composition;
namespace UI
{
3 references | 0 changes | 0 authors, 0 changes
public partial class CaculatorUI : Form
{
[ImportMany]
0 references | 0 changes | 0 authors, 0 changes
public IEnumerable<ILogger> Loggers { get; set; }
1 reference | 0 changes | 0 authors, 0 changes
public CaculatorUI()
{
InitializeComponent();
_themeFactory = new LightThemeFactory();
// Configure the MEF container
try
{
var catalog = new DirectoryCatalog(AppDomain.CurrentDomain.BaseDirectory);
var container = new CompositionContainer(catalog);
container.ComposeParts(this);
}
catch (CompositionException ex)
{
Console.WriteLine("CompositionException: " + ex.ToString());
}
}
3 references | 0 changes | 0 authors, 0 changes
private ILogger GetLogger()
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SimpleCalculator' (5 of 5 pr
CalculatorThemes
IBinary
IUnary
Logging
UI
Properties
References
App.config
appsettings.json
CaculatorUI.cs
CaculatorUI.Designer.cs
CaculatorUI.resx
packages.config
Program.cs
Properties
100 %
No issues found
SPC
CRLF
开发者 PowerShell
Error List
Output
Ready
0 / 0
99+
master
simpleCalculator

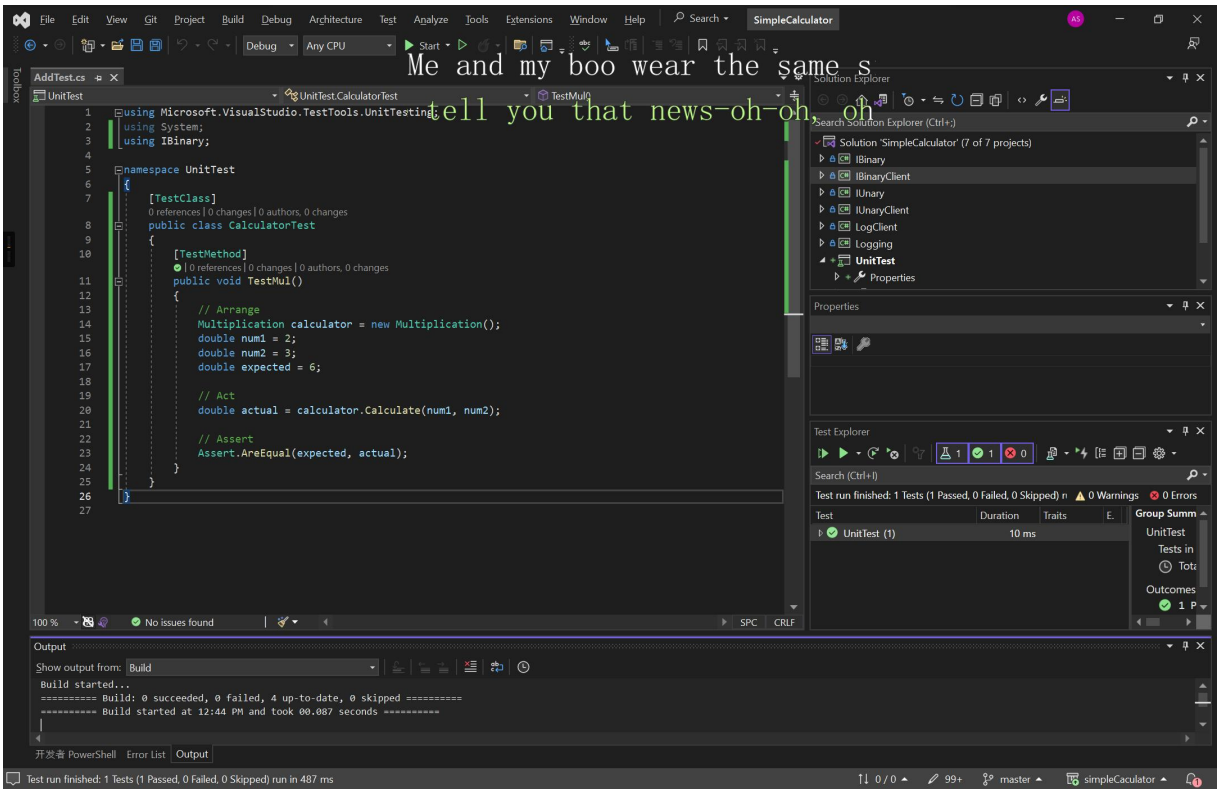


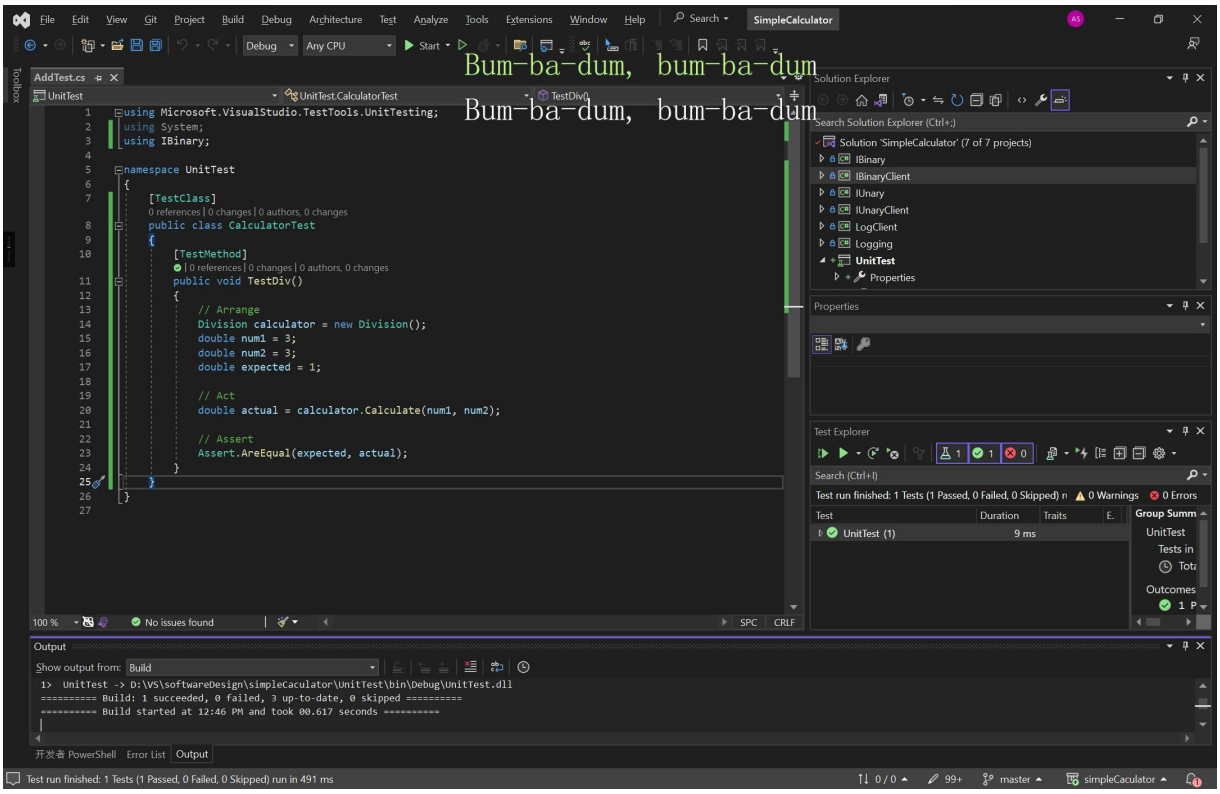
四、运行效果

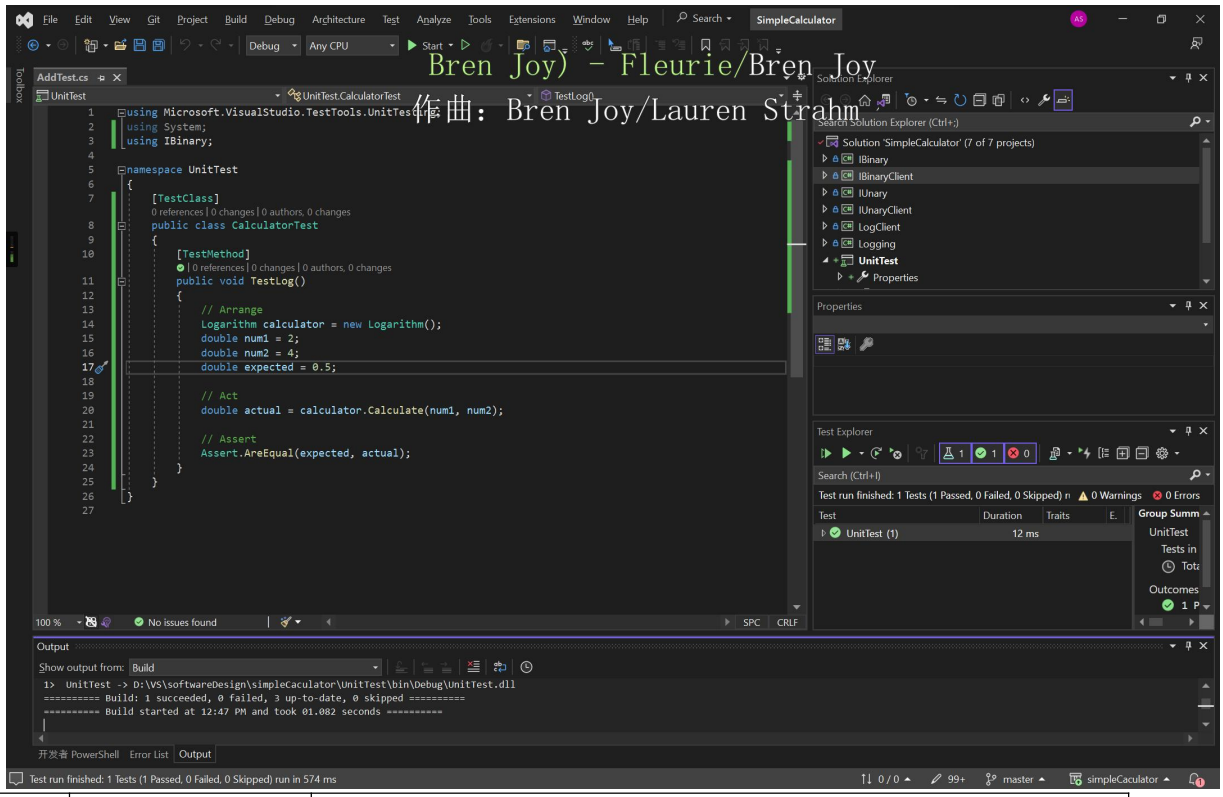
2	对于领域模型 1 的单元测试	对领域模型 1 进行单元测试


So we can
to keep the mattress occupied

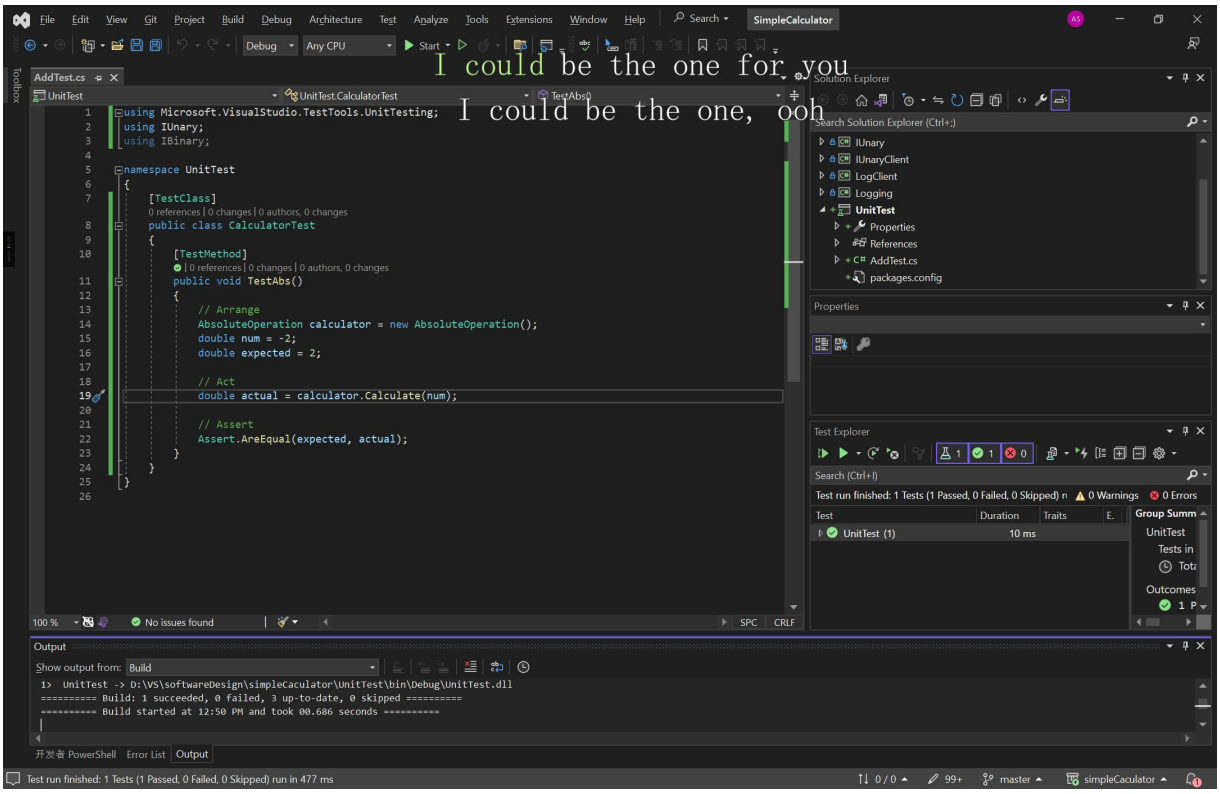

For you they break
For you

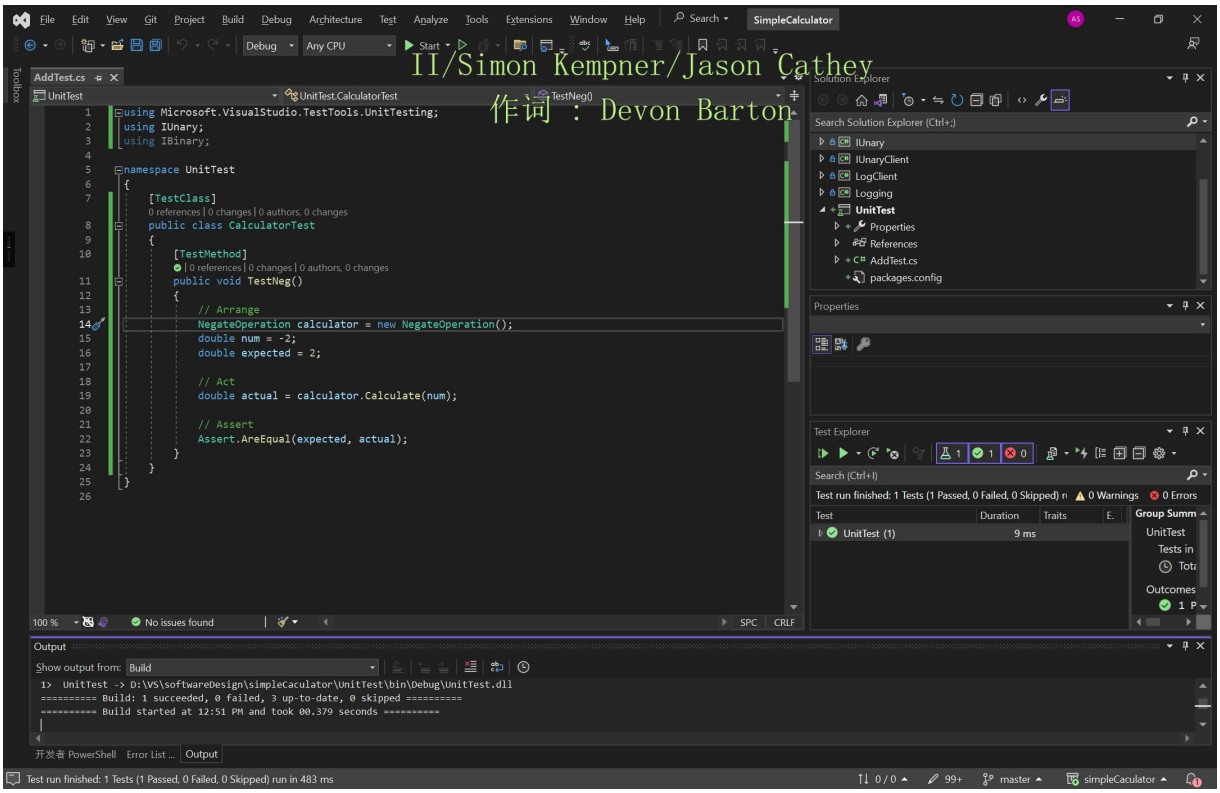

Me and my boo wear the same s
tell you that news-oh-oh, oh
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System;
using IBinary;
namespace UnitTest
{
[TestClass]
public class CalculatorTest
{
[TestMethod]
public void TestMul()
{
// Arrange
Multiplication calculator = new Multiplication();
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

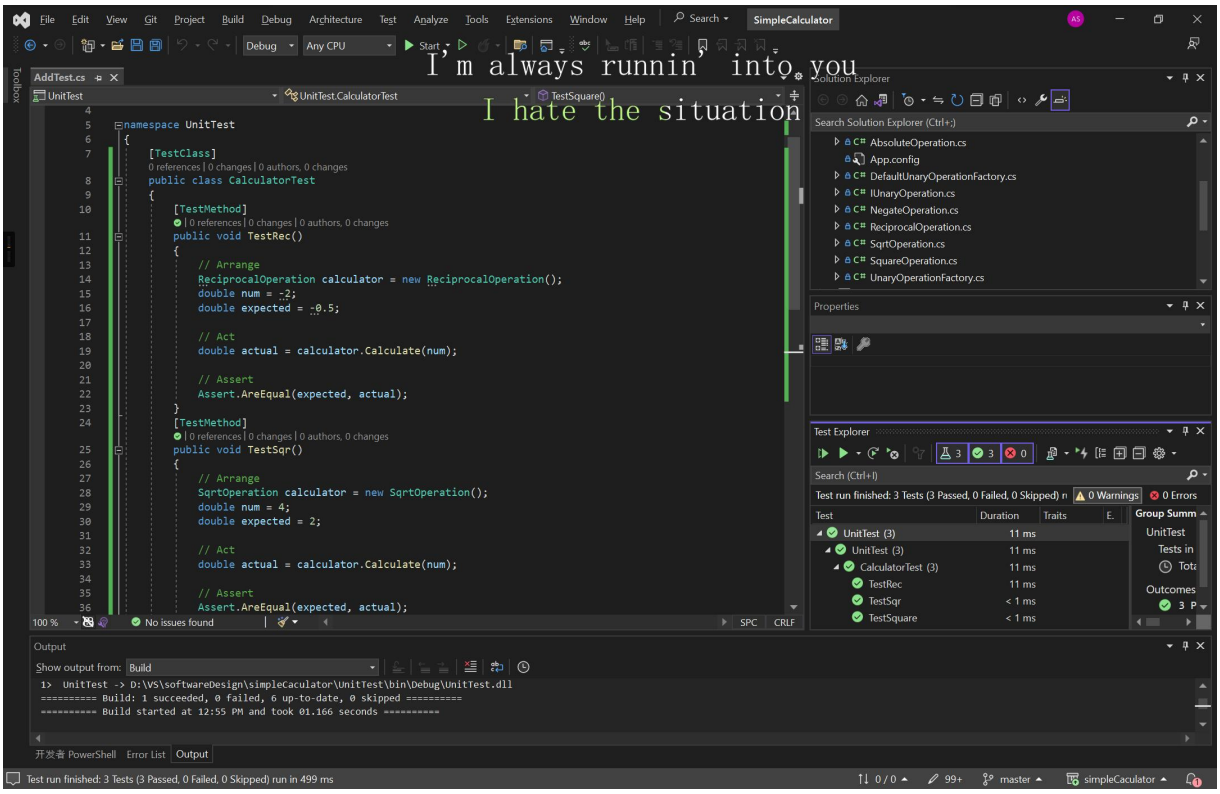


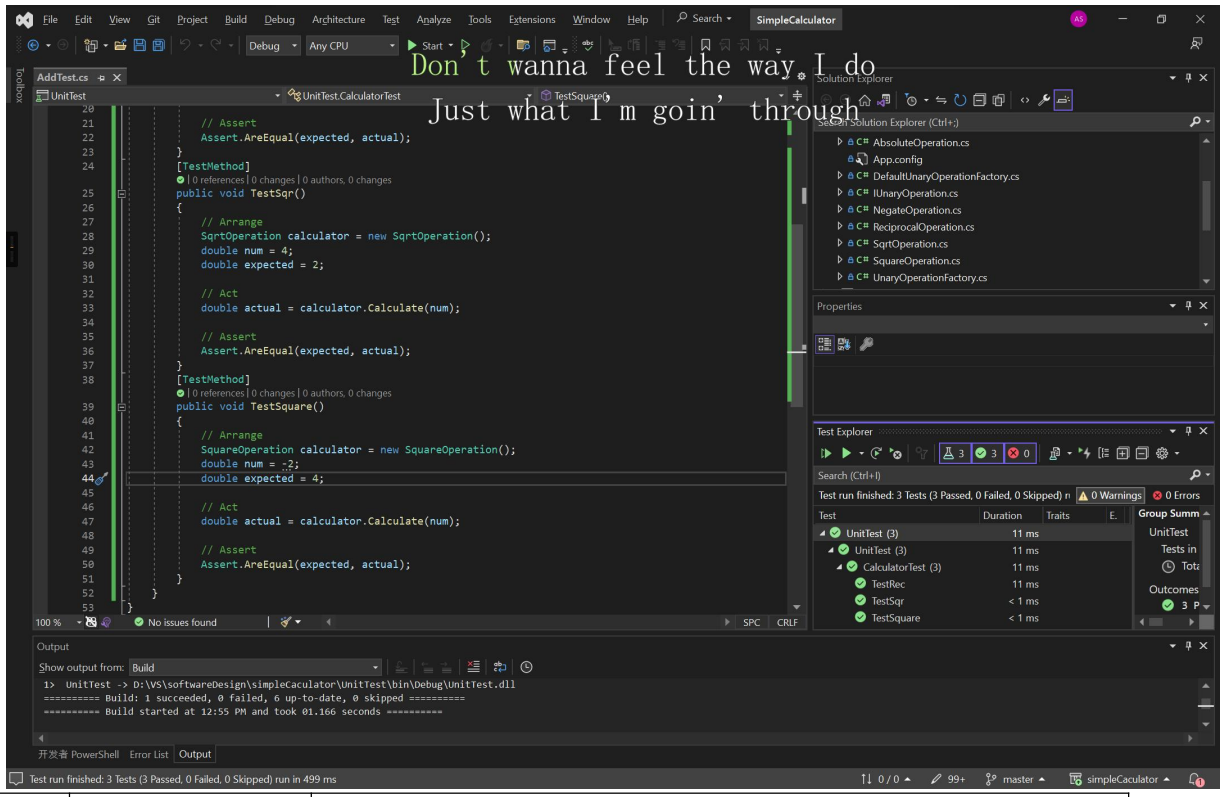


4	对于领域模型 2 的单元测试	对领域模型 2 进行单元测试

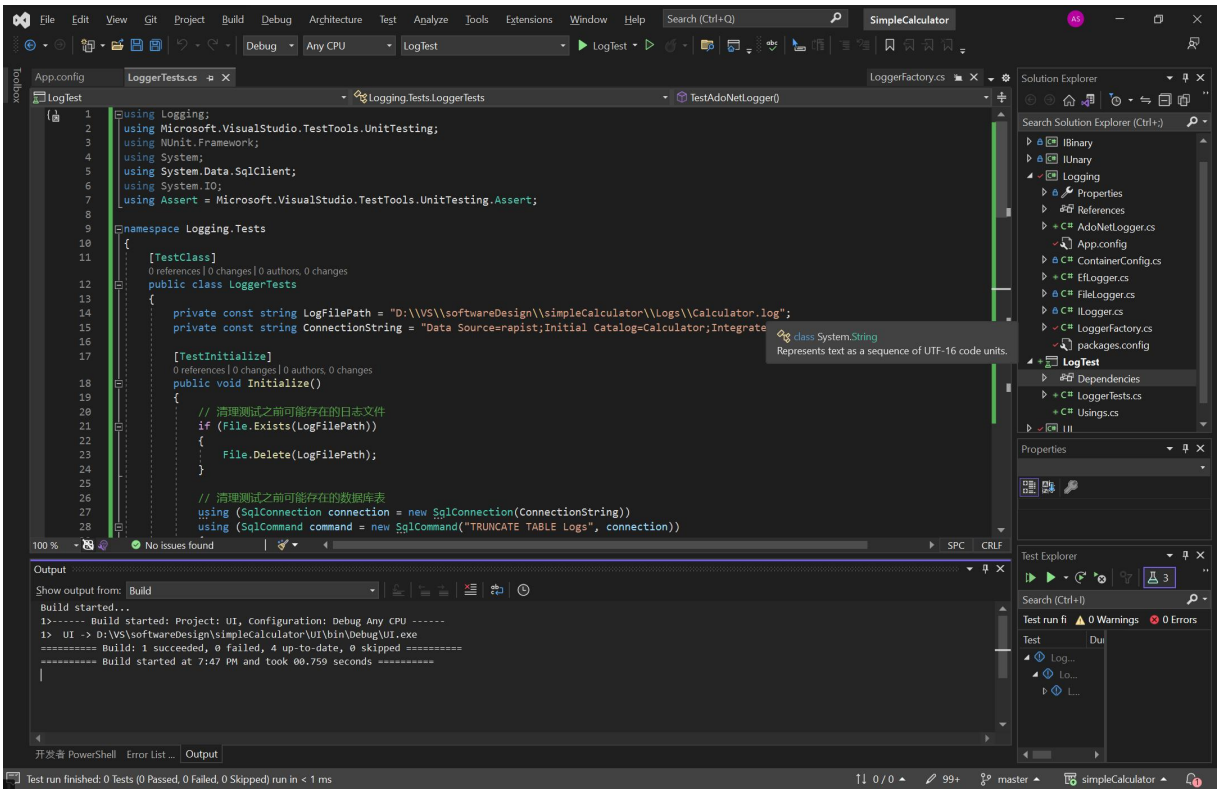

I could be the one for you
I could be the one, ooh
AddTest.cs
UnitTest.CalculatorTest
TestAbs()
Solution Explorer
Search Solution Explorer (Ctrl+;)
IUnary
IUnaryClient
LogClient
Logging
UnitTest
Properties
References
AddTest.cs
packages.config
Properties
Test Explorer
Search (Ctrl+I)
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) n
0 Warnings
0 Errors
Test
Duration
Traits
UnitTest (1)
10 ms
Group Summ
UnitTest
Tests in
Outcomes
100 %
No issues found
SPC
CRLF
Output
Show output from: Build
1> UnitTest -> D:\VS\softwareDesign\simpleCaculator\UnitTest\bin\Debug\UnitTest.dll
========== Build: 1 succeeded, 0 failed, 3 up-to-date, 0 skipped ==========
========== Build started at 12:50 PM and took 00.686 seconds ==========
开发者 PowerShell
Error List
Output
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 477 ms
master
simpleCaculator


II/Simon Kempner/Jason Cathey
作词：Devon Barton
AddTest.cs
UnitTest.CalculatorTest
TestNeg()
Solution Explorer
Search Solution Explorer (Ctrl+;)
IUnary
IUnaryClient
LogClient
Logging
UnitTest
Properties
References
AddTest.cs
packages.config
Properties
Test Explorer
Search (Ctrl+I)
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) n
0 Warnings
0 Errors
Test
Duration
Traits
UnitTest (1)
9 ms
Group Summ
UnitTest
Tests in
Outcomes
100 %
No issues found
SPC
CRLF
Output
Show output from: Build
1> UnitTest -> D:\VS\softwareDesign\simpleCaculator\UnitTest\bin\Debug\UnitTest.dll
========== Build: 1 succeeded, 0 failed, 3 up-to-date, 0 skipped ==========
========== Build started at 12:51 PM and took 00.379 seconds ==========
开发者 PowerShell
Error List ...
Output
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 483 ms
master
simpleCaculator





7	数据库日志读写 模块单元测试	对于日志读写的多种不同的实现，对其进行单元测试



五、课设心得

在完成这个课设的过程中，我获得了许多宝贵的经验和心得。这个课设要求我设计并实现了一个计算器应用程序，其中包括基本的四则运算以及日志记录和主题设置功能。以下是我对这次课设的总结心得：

首先，通过这个课设，我深入理解了面向对象设计的重要性和实践。通过将功能模块划分为不同的类和接口，我能够更好地组织和管理代码。我使用了抽象工厂模式和依赖注入来实现灵活的日志记录和主题设置功能。这种面向对象的设计方法使得代码更加模块化、可扩展和可维护。

其次，我学会了使用常见的设计模式来解决实际问题。在这个课设中，我使用了抽象工厂模式来实现主题设置功能，这样可以轻松地切换和扩展不同的主题。我还使用了依赖注入来解耦具体的日志记录器实现和使用它的类。这些设计模式帮助我更好地组织代码，并提供了灵活性和可扩展性。

第三，我体验到了测试驱动开发（TDD）的价值。在实现计算器的各个功能模块时，我首先编写了单元测试，然后逐步实现功能以使测试通过。这种迭代的开发方式使得代码更加可靠和健壮，并减少了出错的可能性。通过编写单元测试，我能够更好地理解和验证代码的行为，确保每个模块都按预期工作。

最后，我学会了合理规划项目的结构和组织代码。通过将不同的功能模块分成不同的命名空间和文件夹，我可以更好地管理和维护代码。同时，使用适当的命名和注释可以提高代码的可读性和可理解性。这种良好的项目结构和代码组织可以减少开发过程中的混乱，并帮助其他开发人员更轻松地理解和参与项目。

总的来说，这个课设是一个很好的实践机会，让我将所学的知识应用到一个具体的项目中。通过设计和实现一个完整的计算器应用程序，我不仅加深了对面向对象设计原则和设计模式的理解，还提高了自己的编码能力和解决问题的能力。我相信这些经验和技能将对我未来的软件开发工作产生积极的影响。