ZOLTAN SZTRELCSIK

Budapest/Hungary

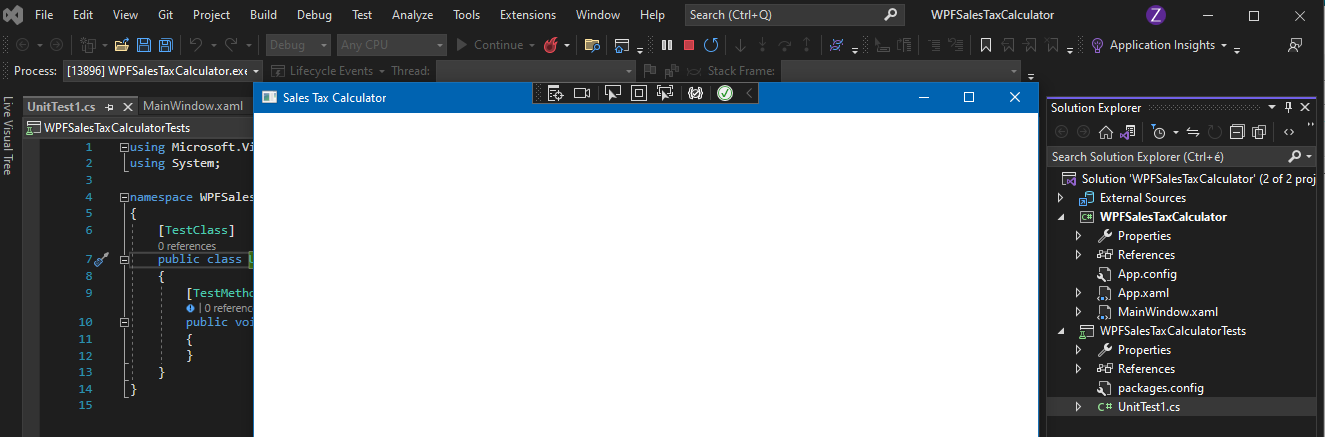
szzoltan@gmx.net

szzoltan@gmx.net

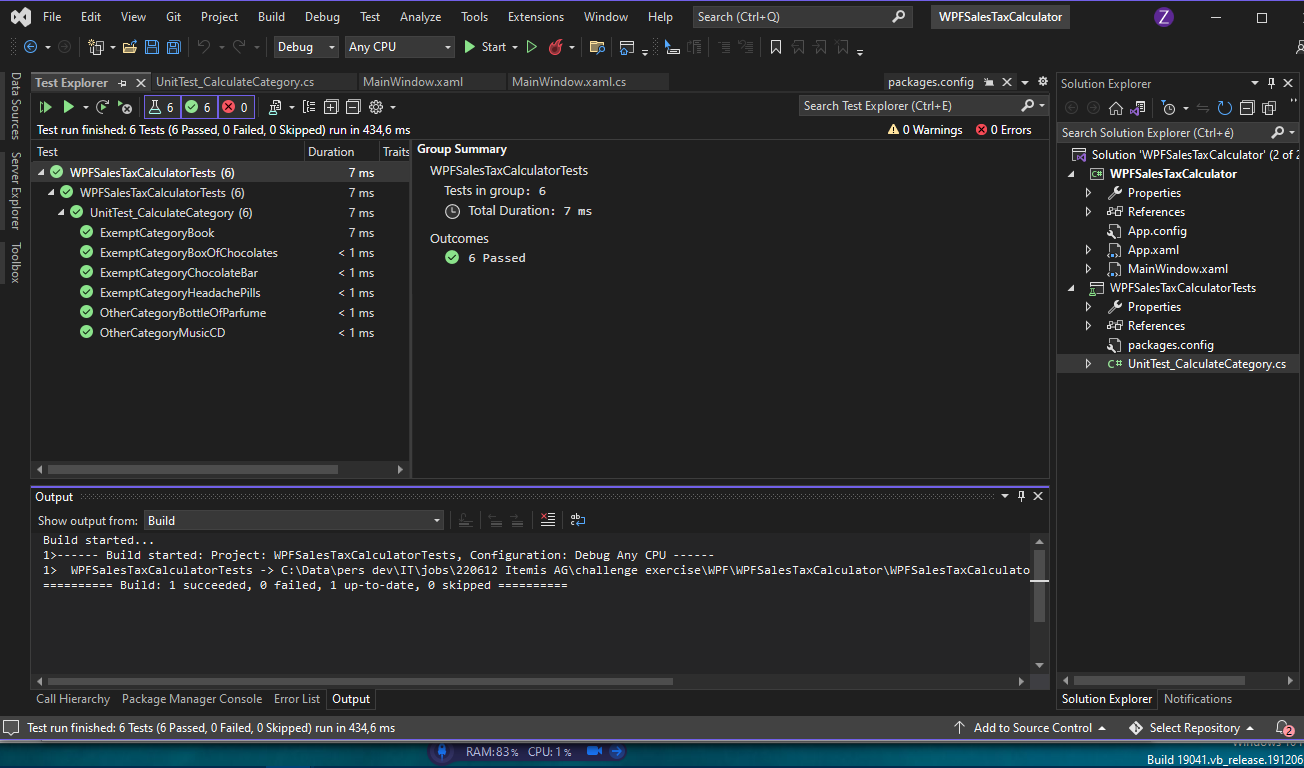
# **Development process of the Sales Tax Calculator WPF application**

In this document you find a number of images I created during the development process. They will give you an overview of how and in which sequence the application was developed. I used test-driven development (TDD), therefore in many cases the test cases for the individual method were created in the first step, which was followed by the design of the method itself.

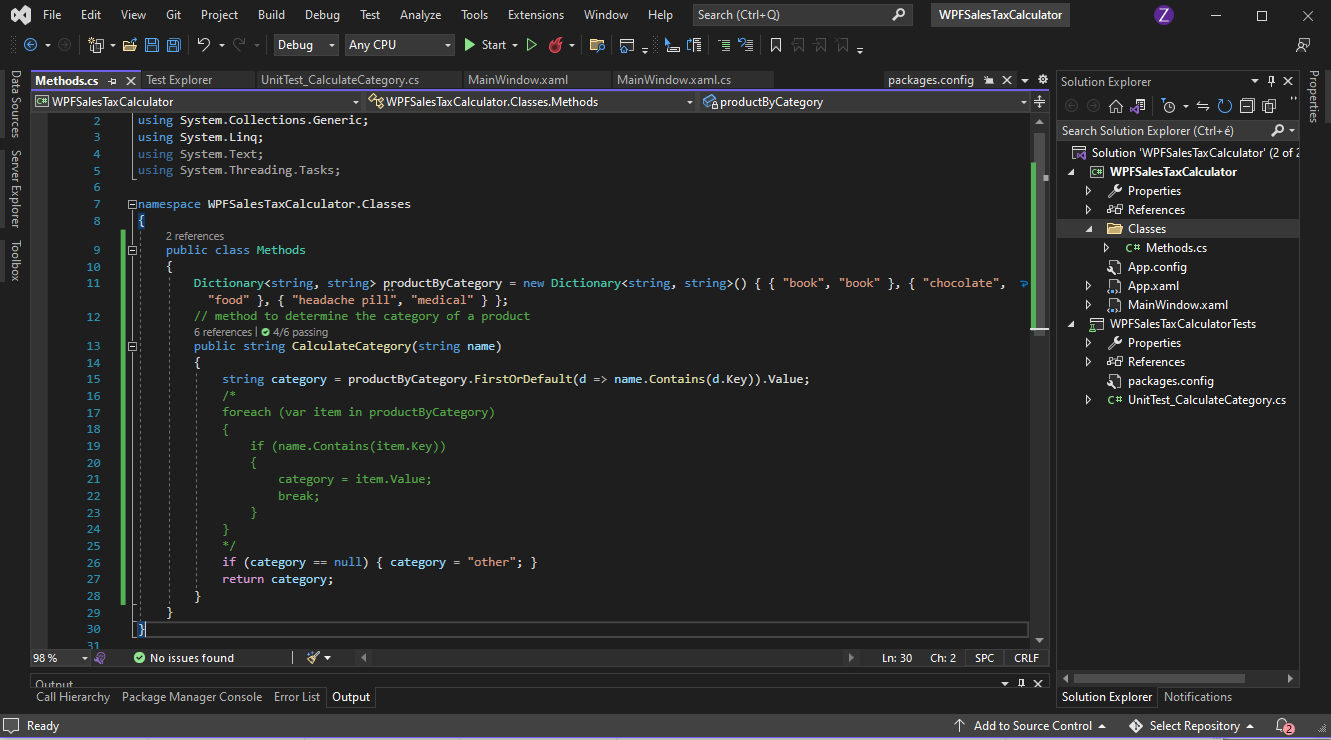
Default WPF project template



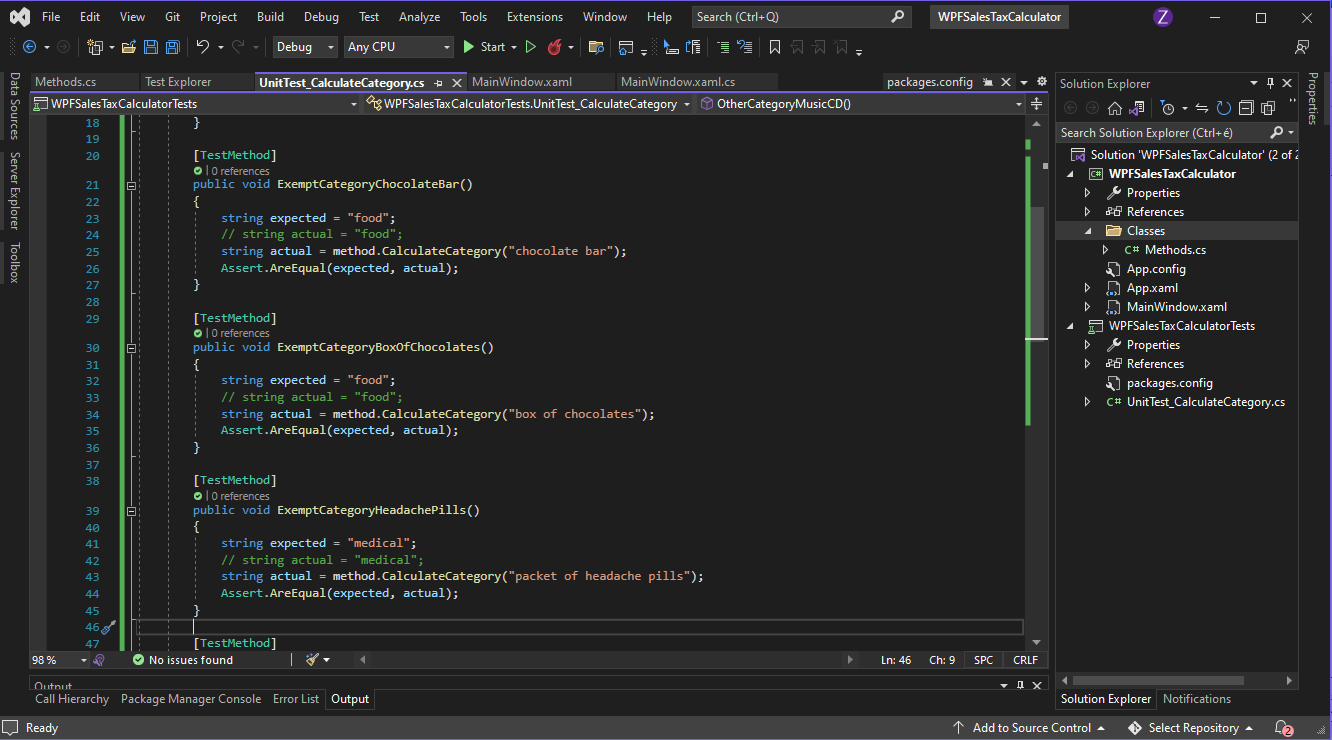
Test cases for CalculateCategory method



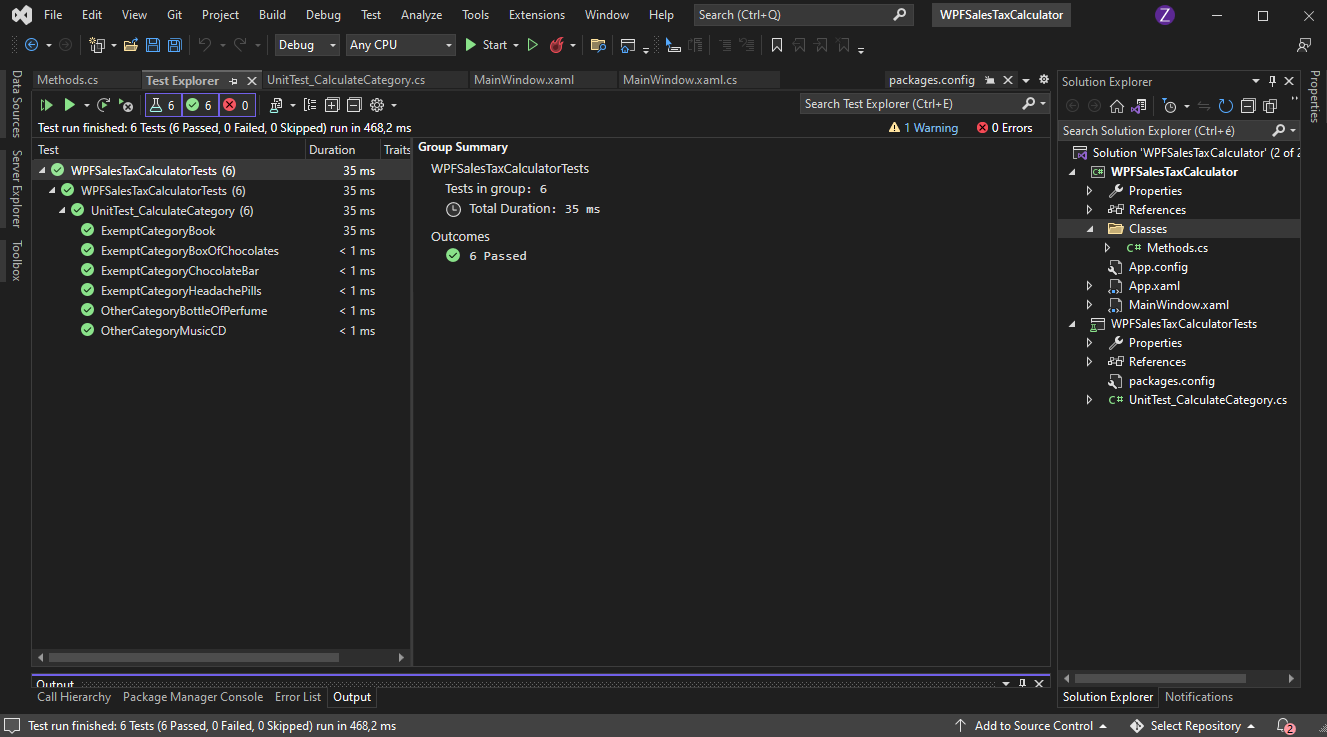
Create new CalculateCategory method to test



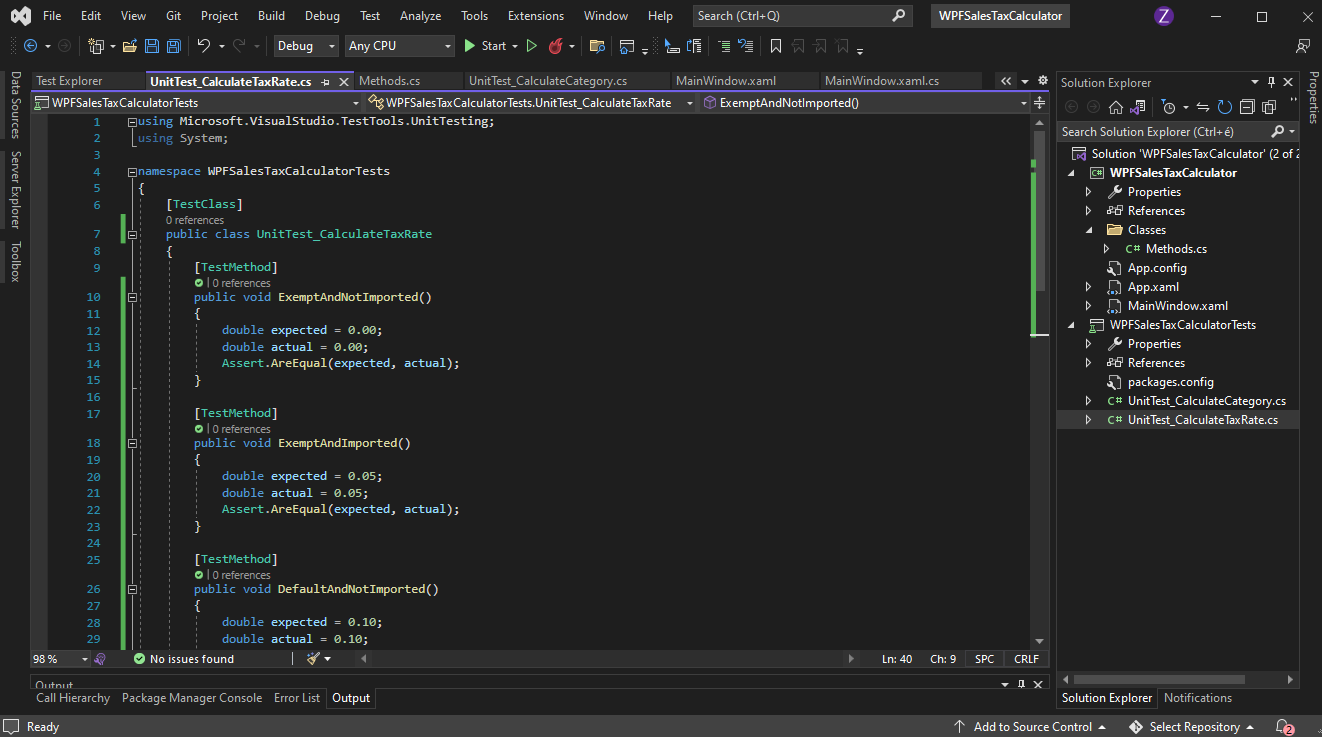
Change actual value of test methods to returned values of CalculateCategory method



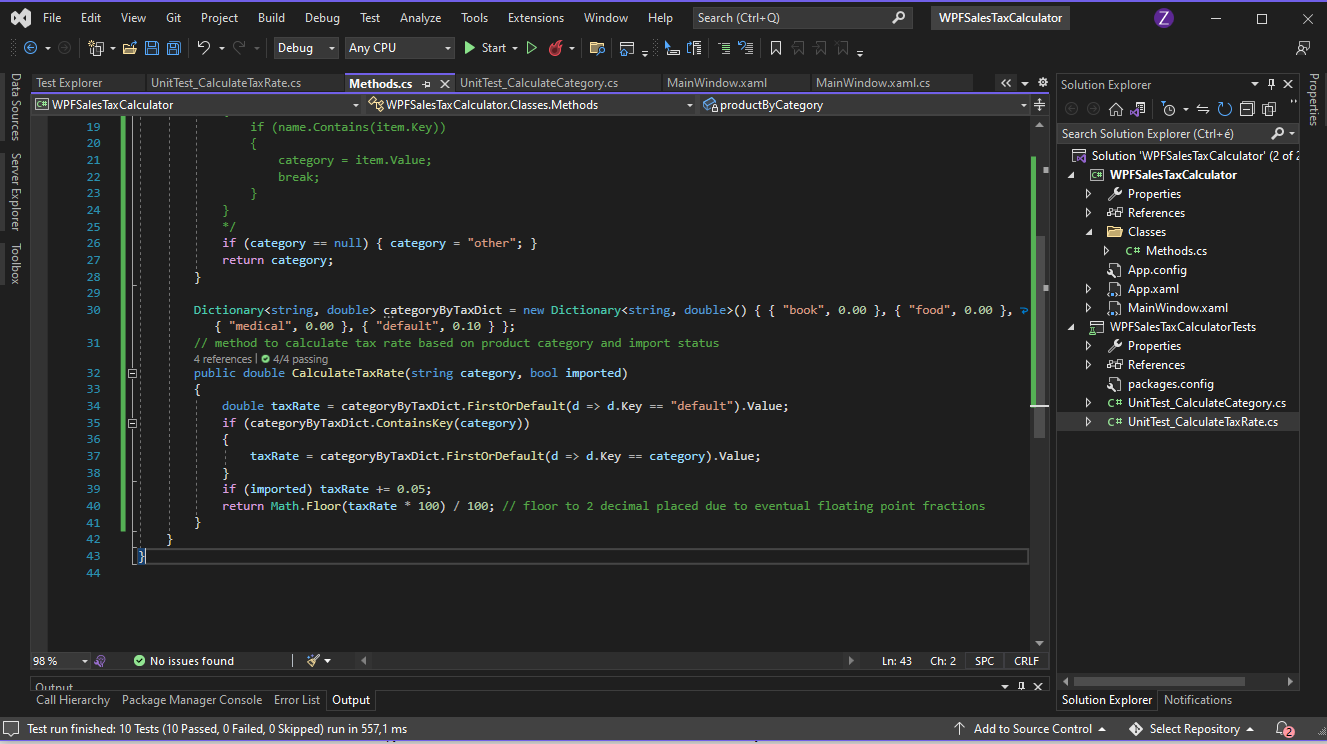
Successful tests for CalculateCategory method



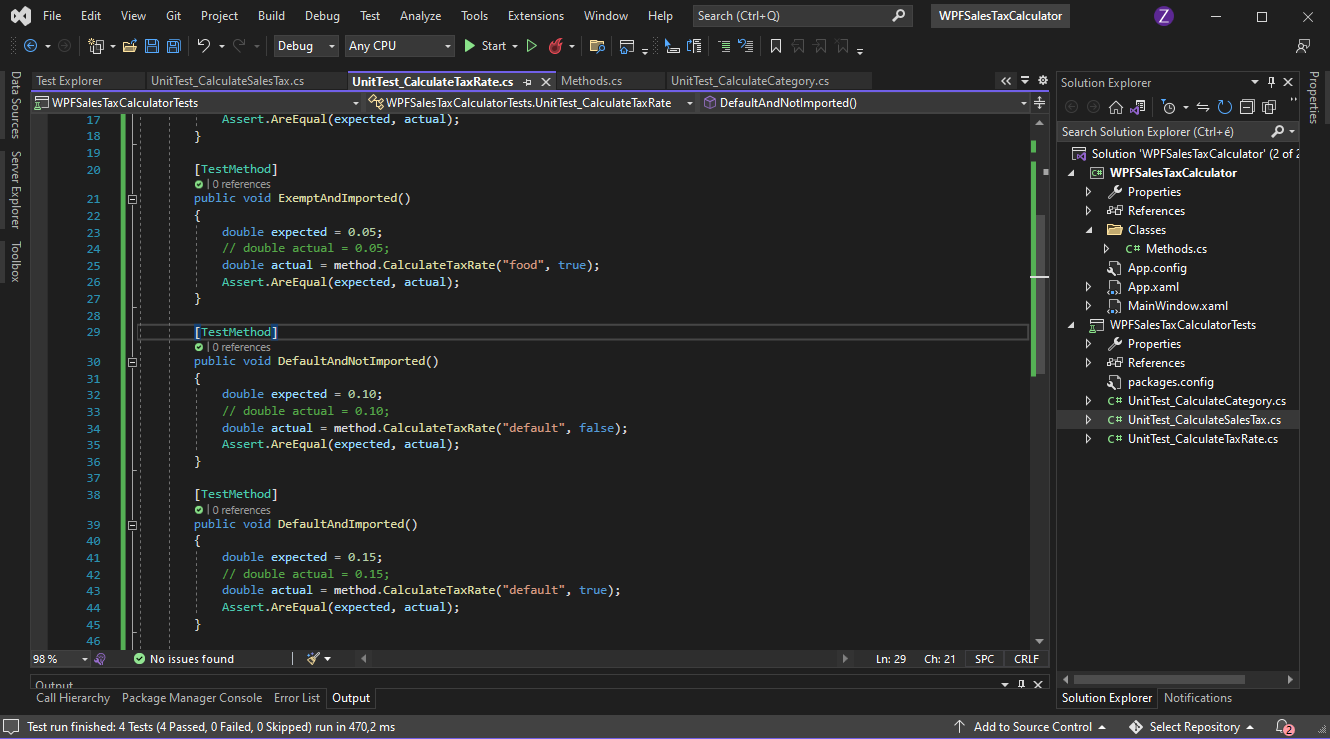
Create initial test methods for CalculateTaxRate method



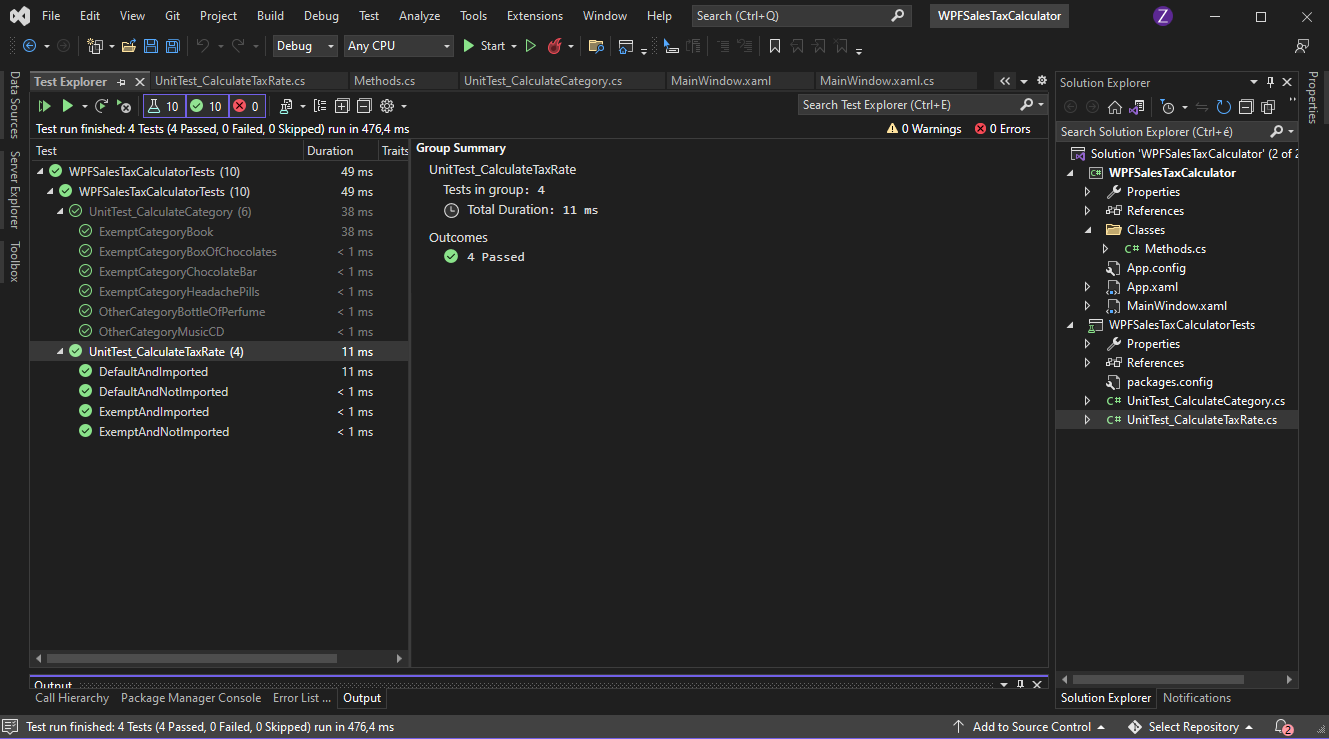
Create new CalculateTaxRate method to test:



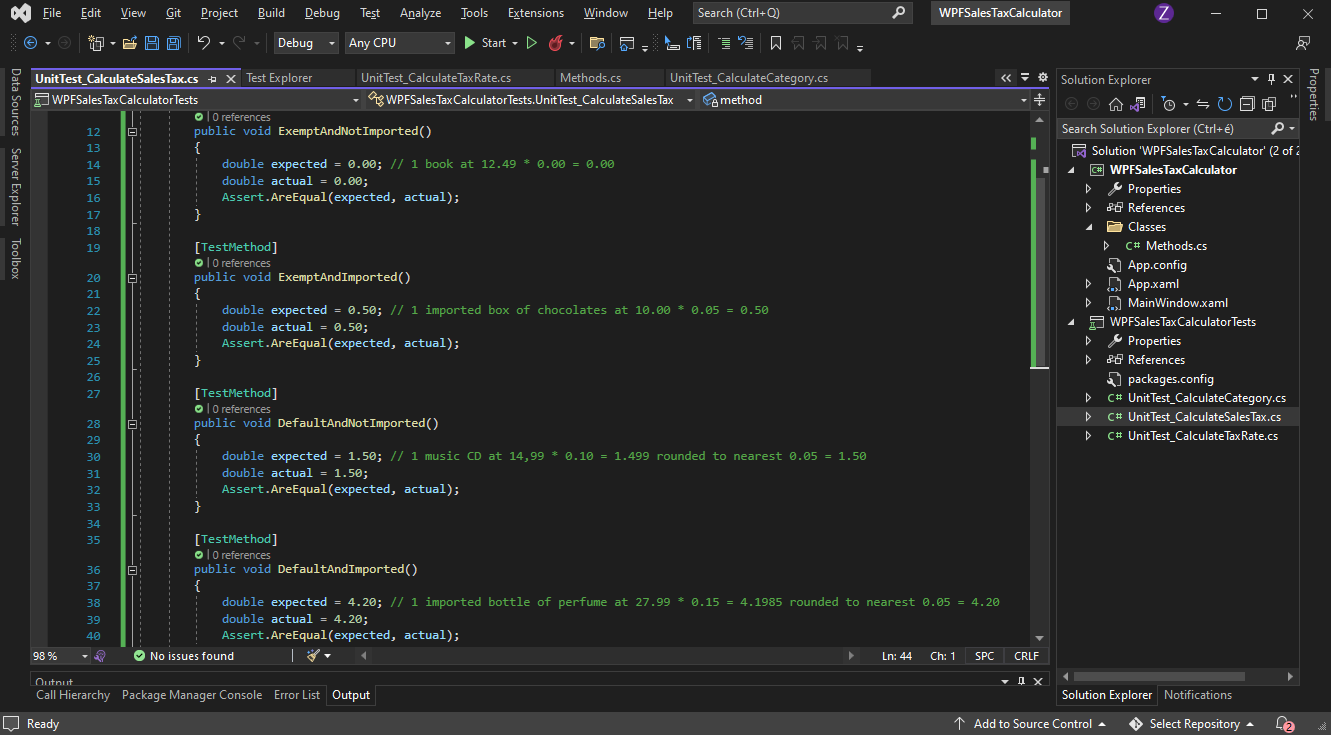
Change actual value of test methods to returned values of CalculateTaxRate method



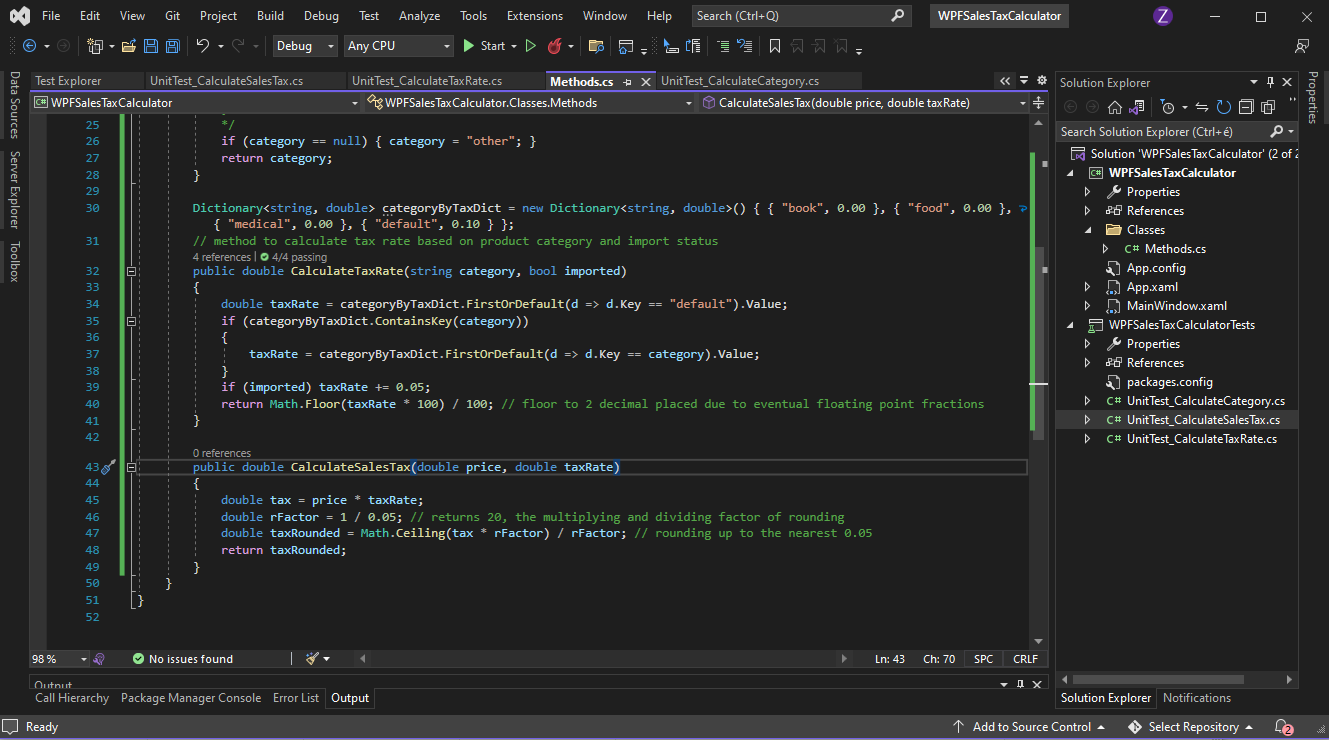
Successful tests for CalculateTaxRate method



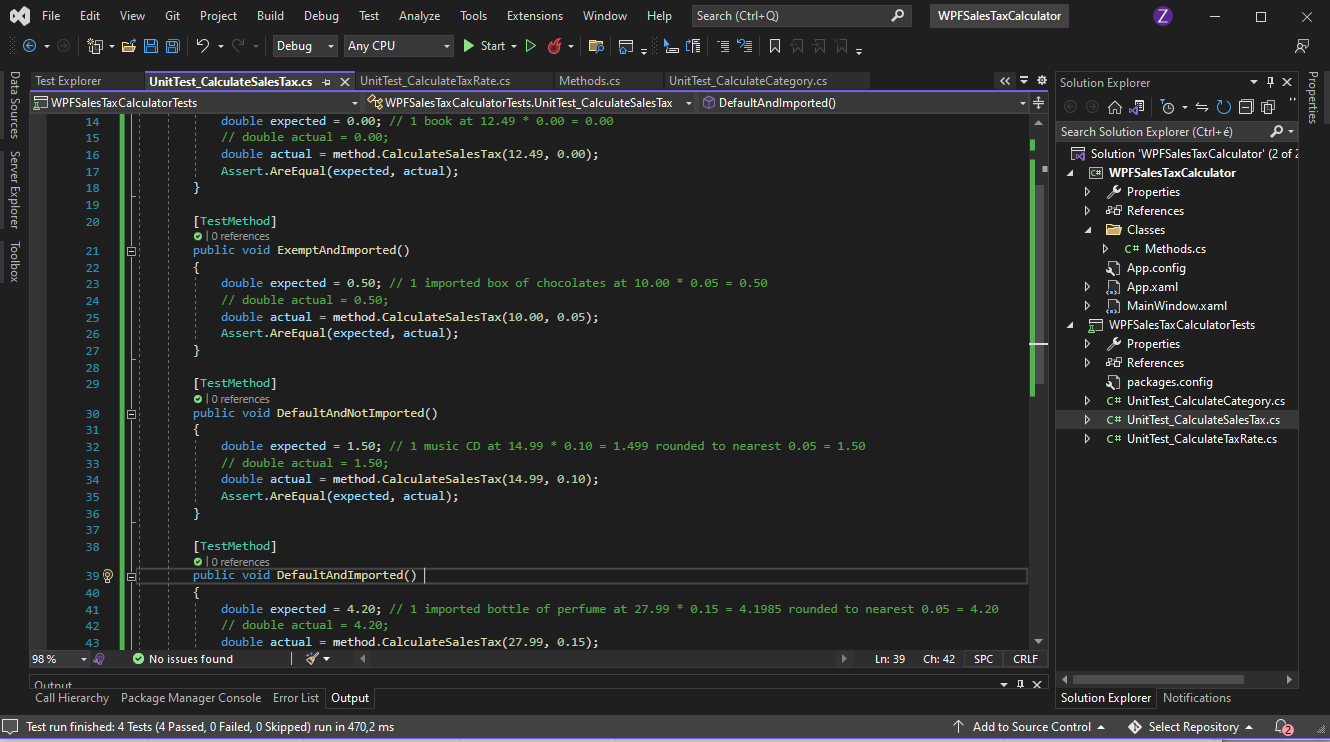
Create text methods for CalculateSalesTax method



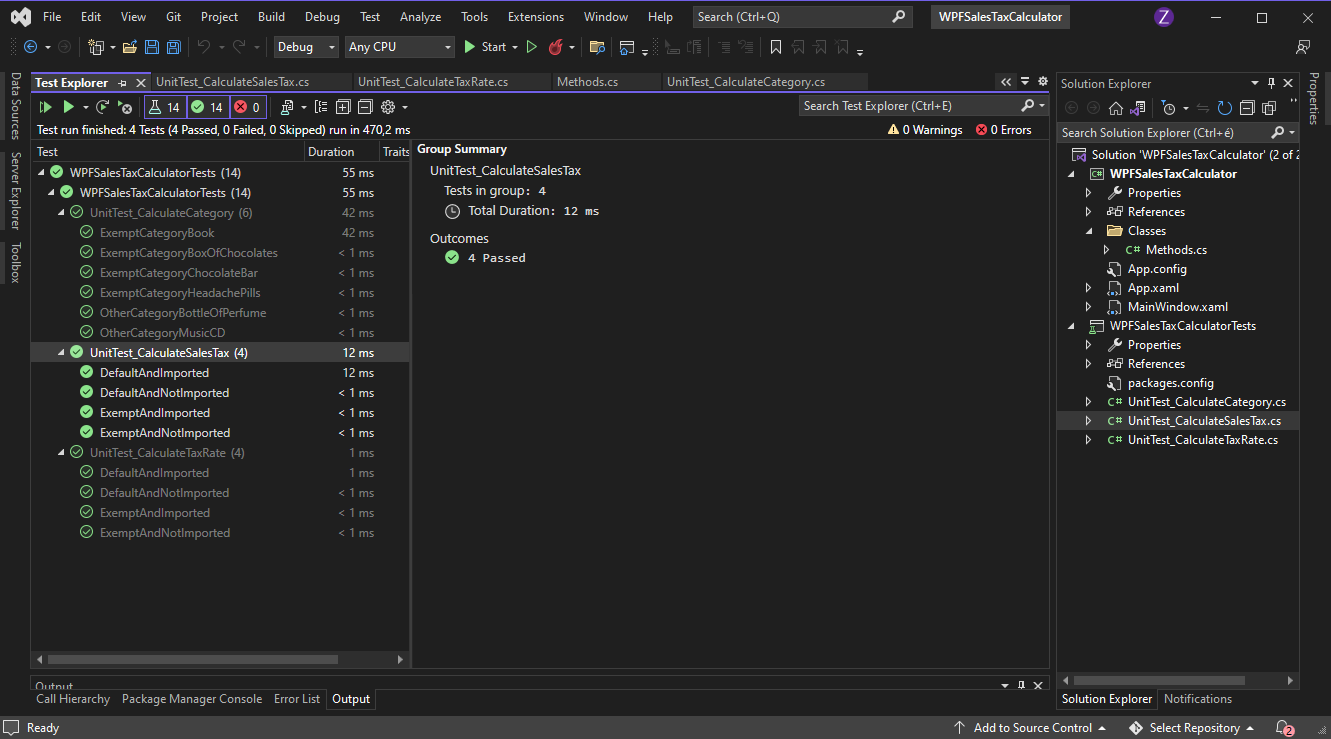
Create new CalculateSalesTax method to test:



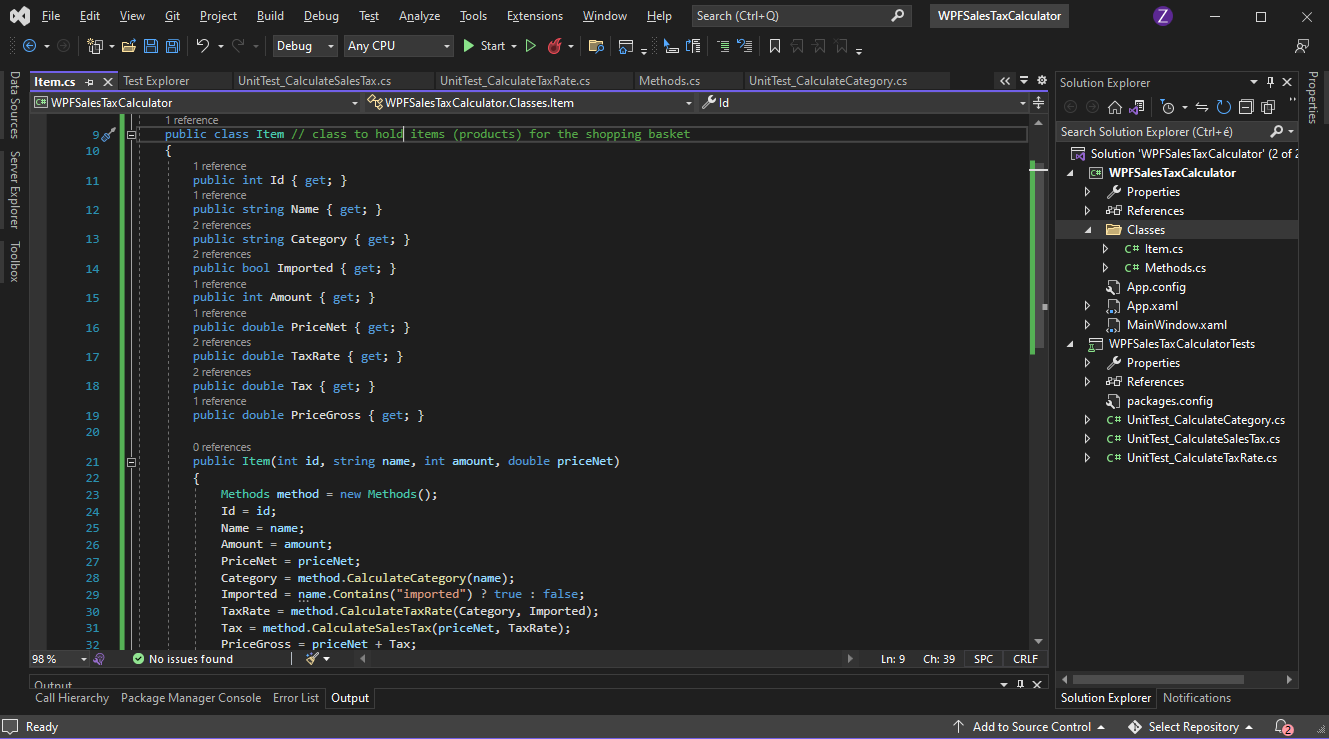
Change actual value of test methods to returned values of CalculatesSalesTax method



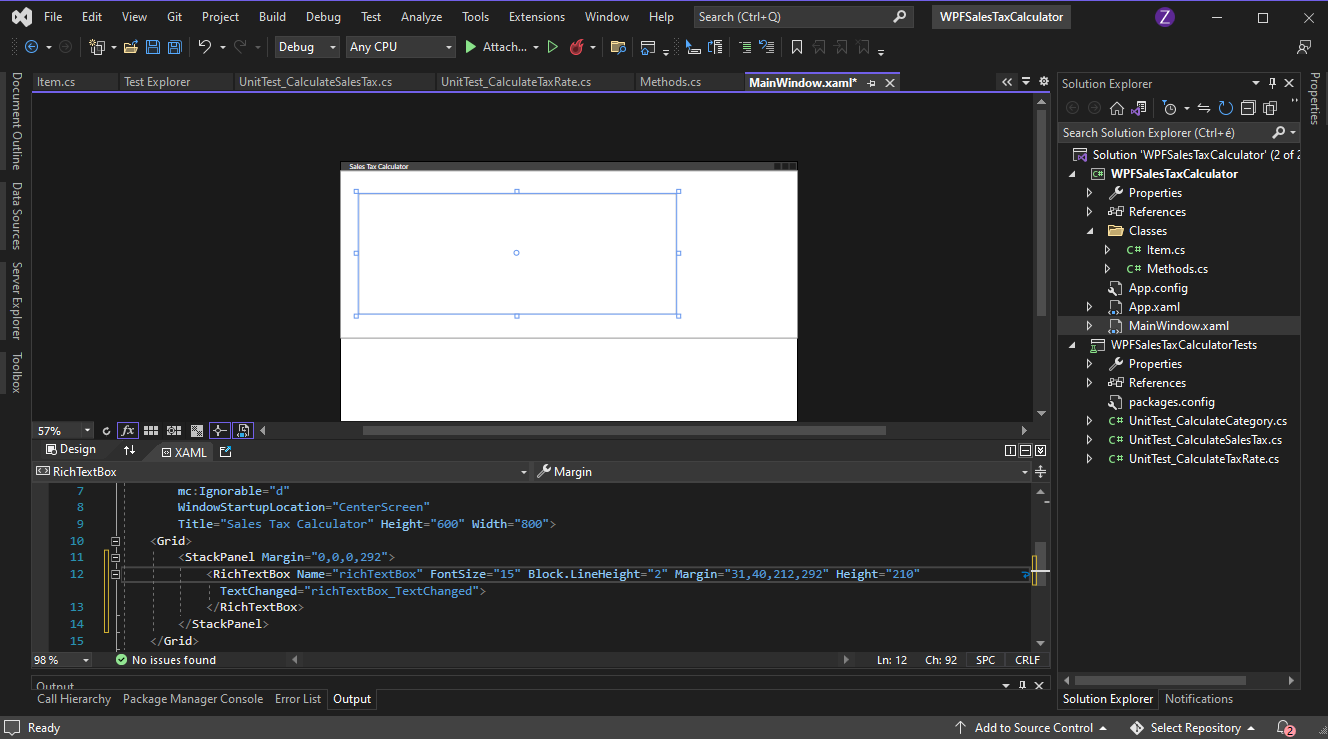
Successful tests for CalculateSalesTax method



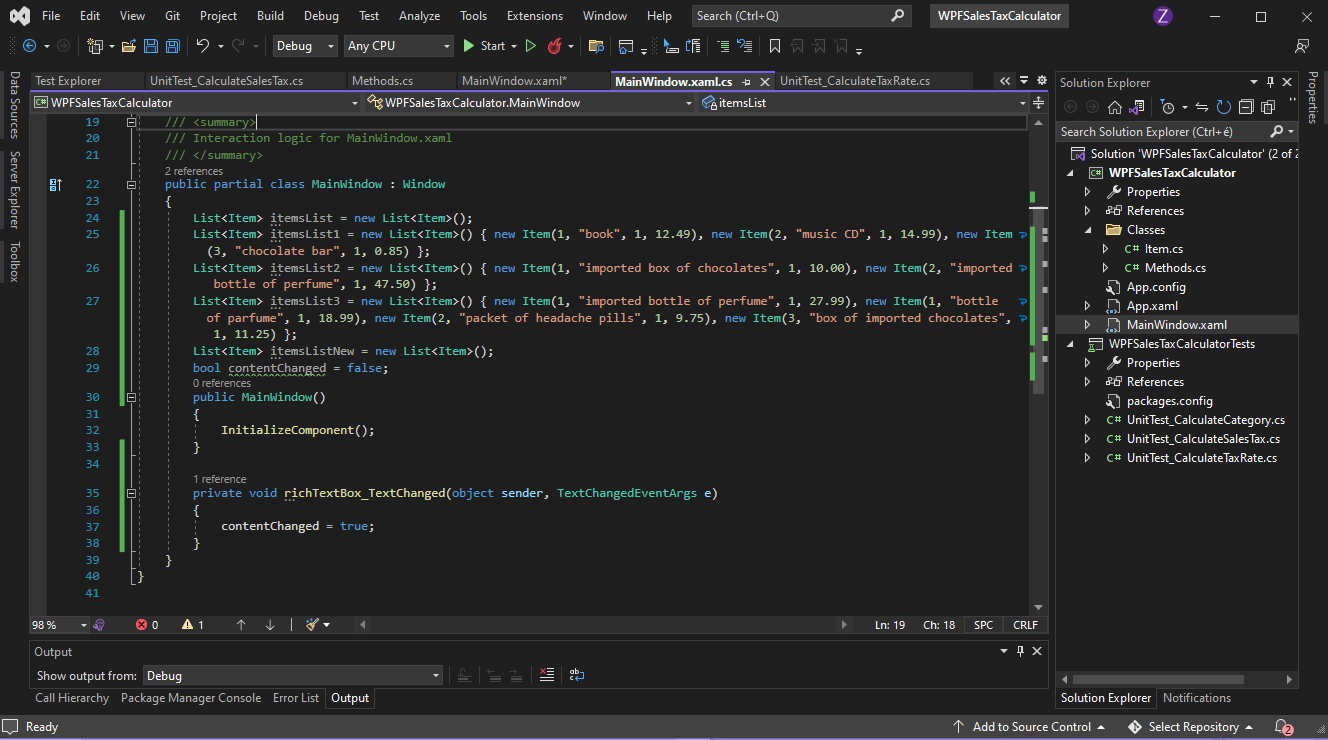
Add Items class (this holds all information for any product in the shopping basket)



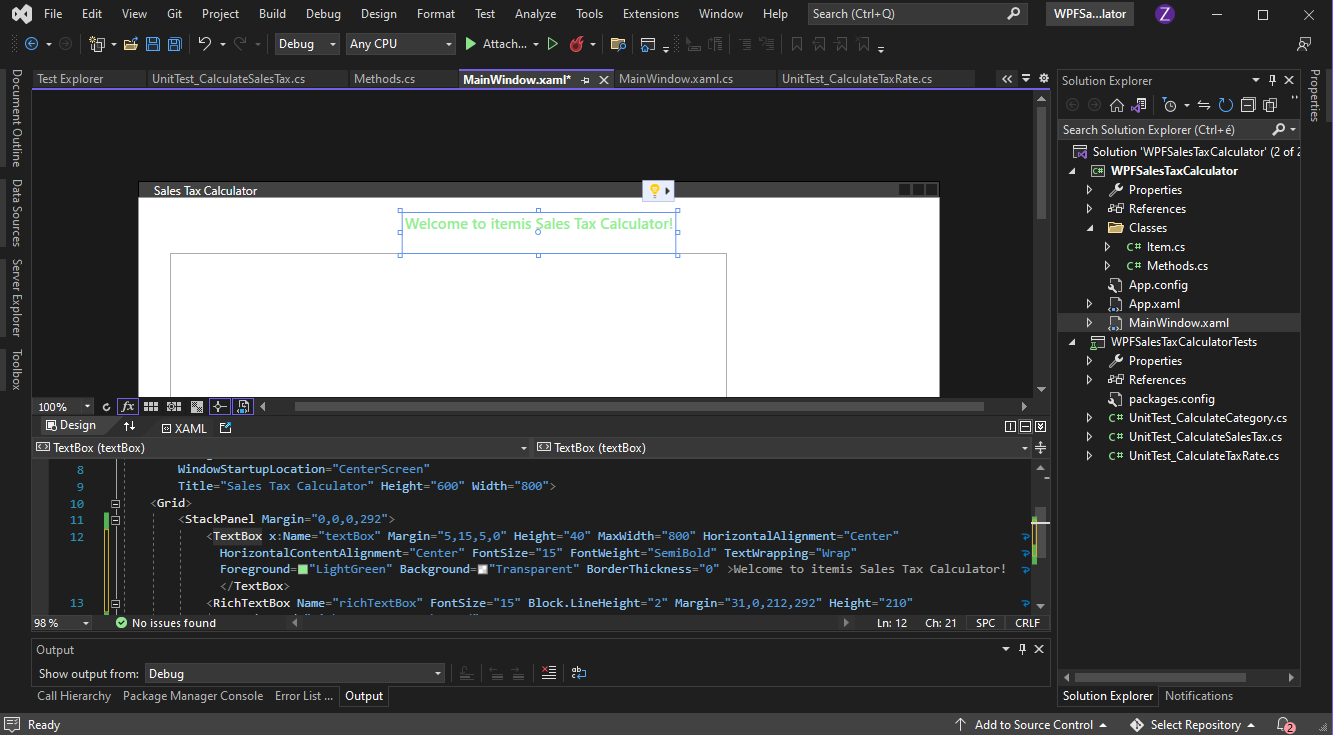
Add RichTextBox control (this is where the user can enter the content of the shopping basket)



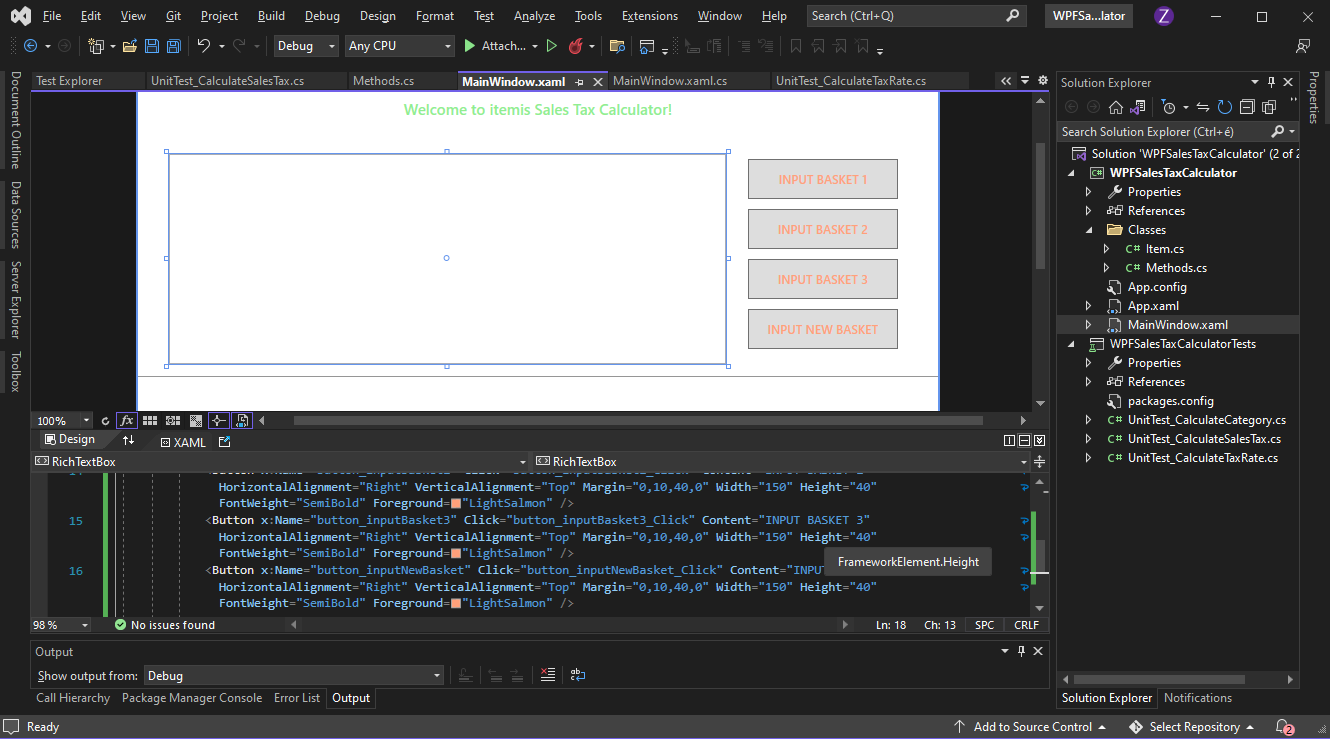
Add TextChanged event and lists (including the items of the sample shopping baskets)



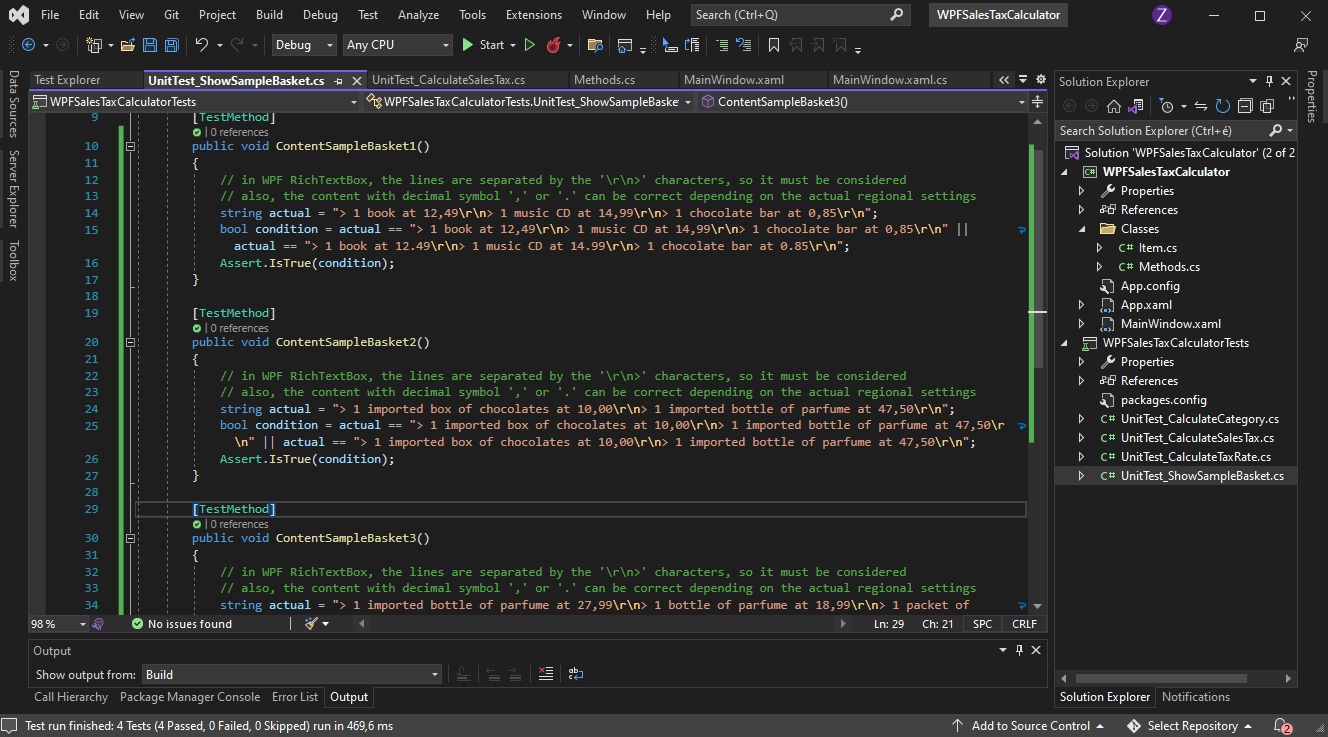
Add TextBox (this displays useful messages for the user for the different functions)



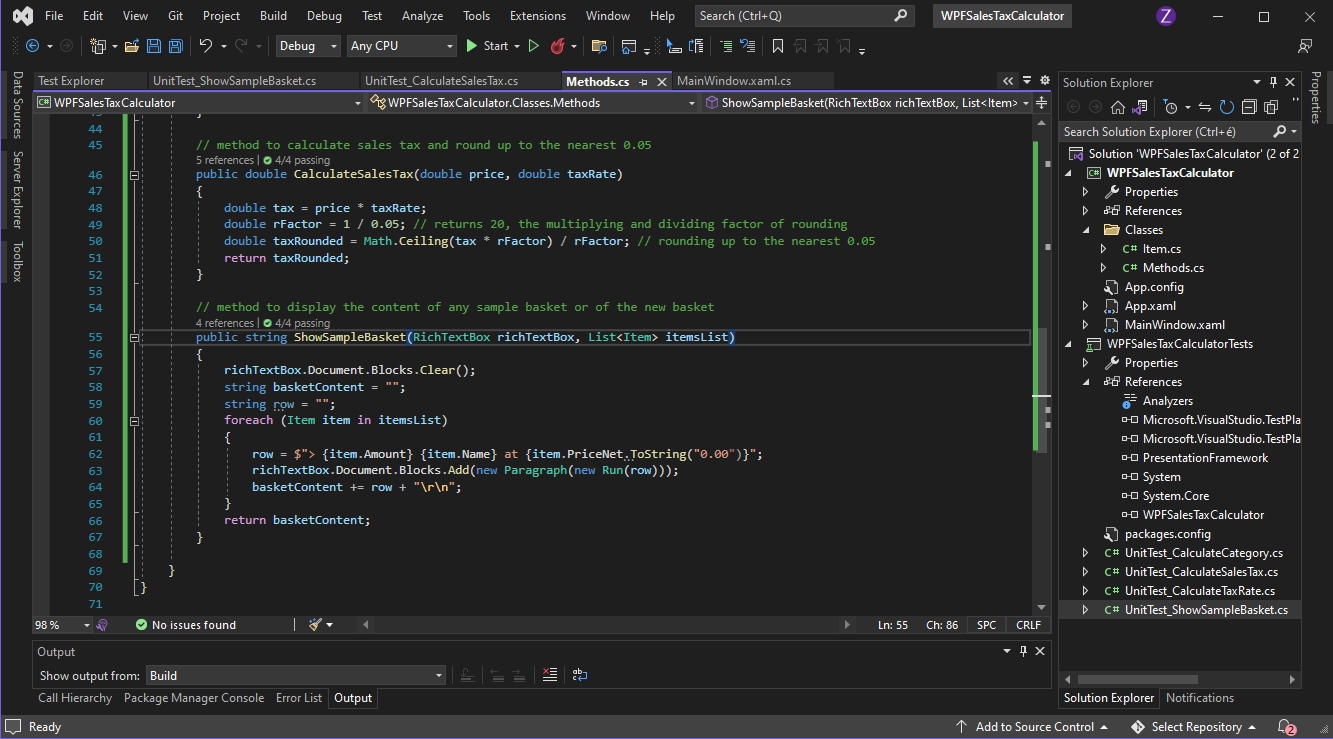
Add input buttons



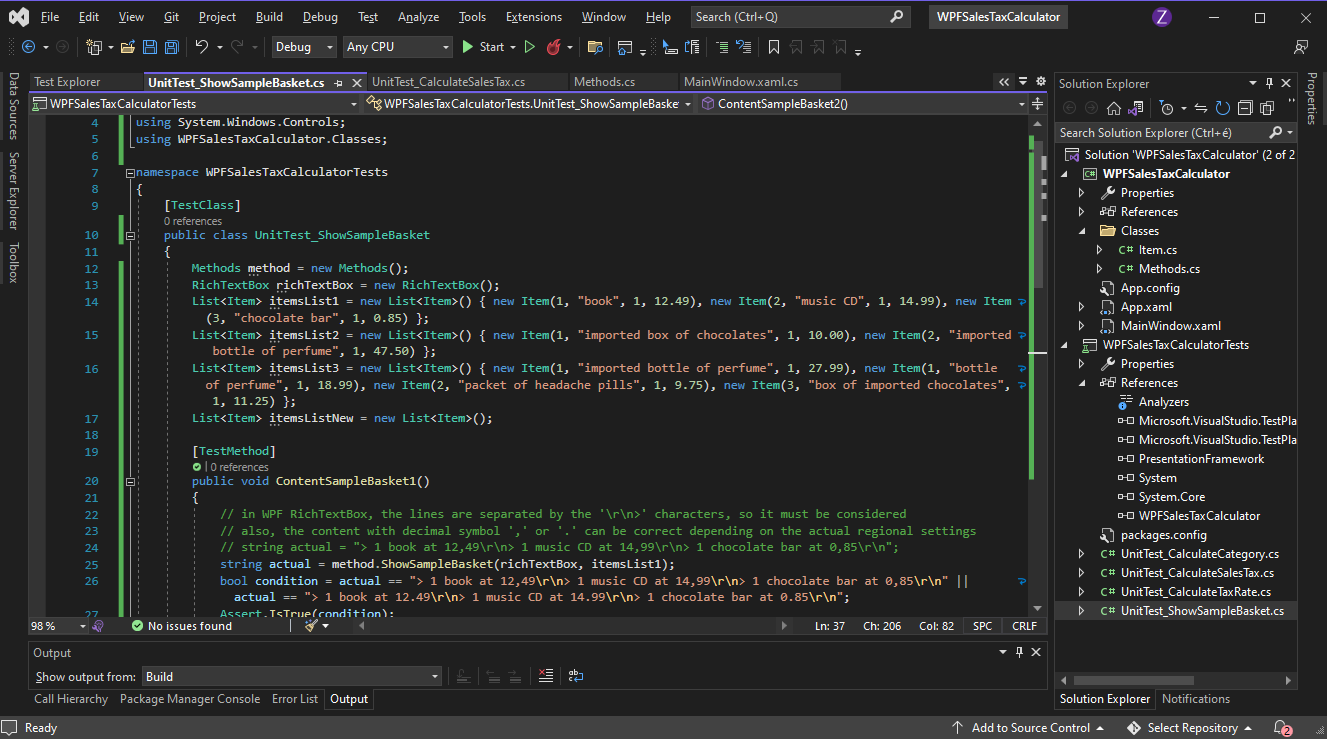
Create initial test methods for ShowSampleBasket method



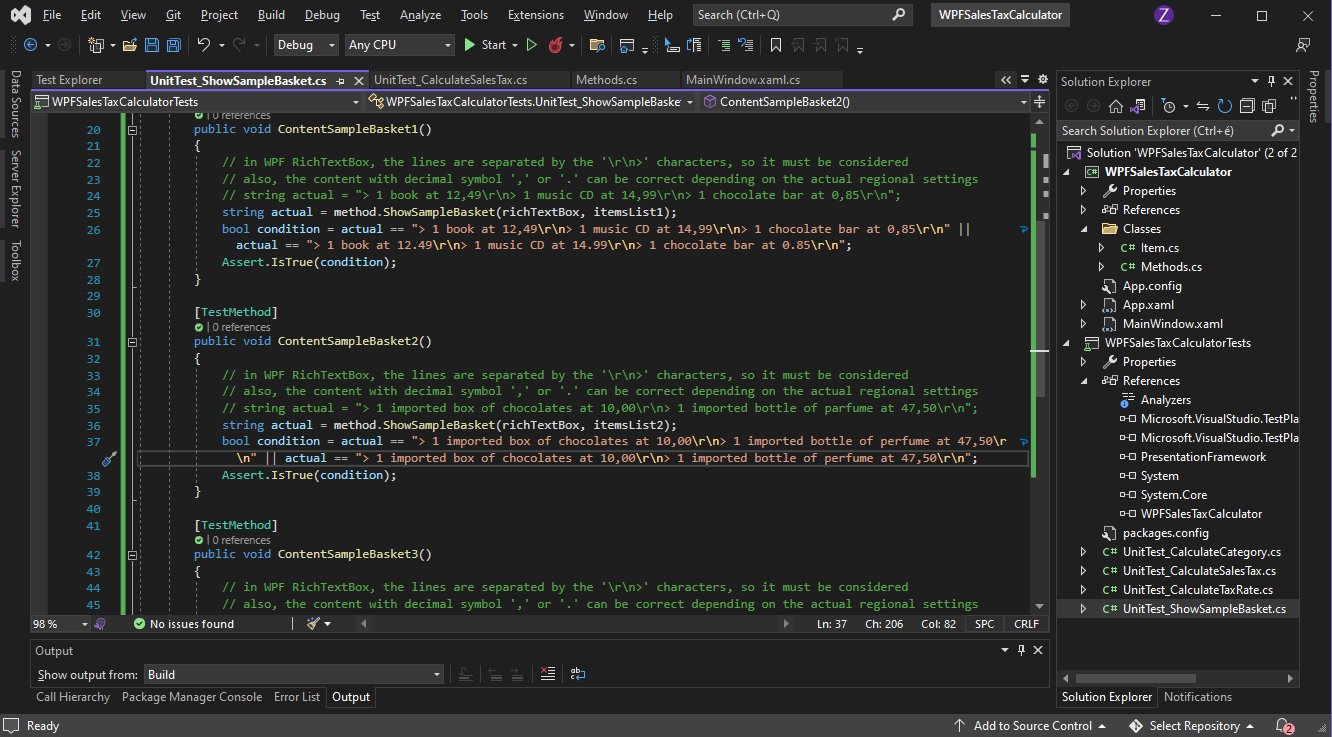
Create new ShowSampleBasket method



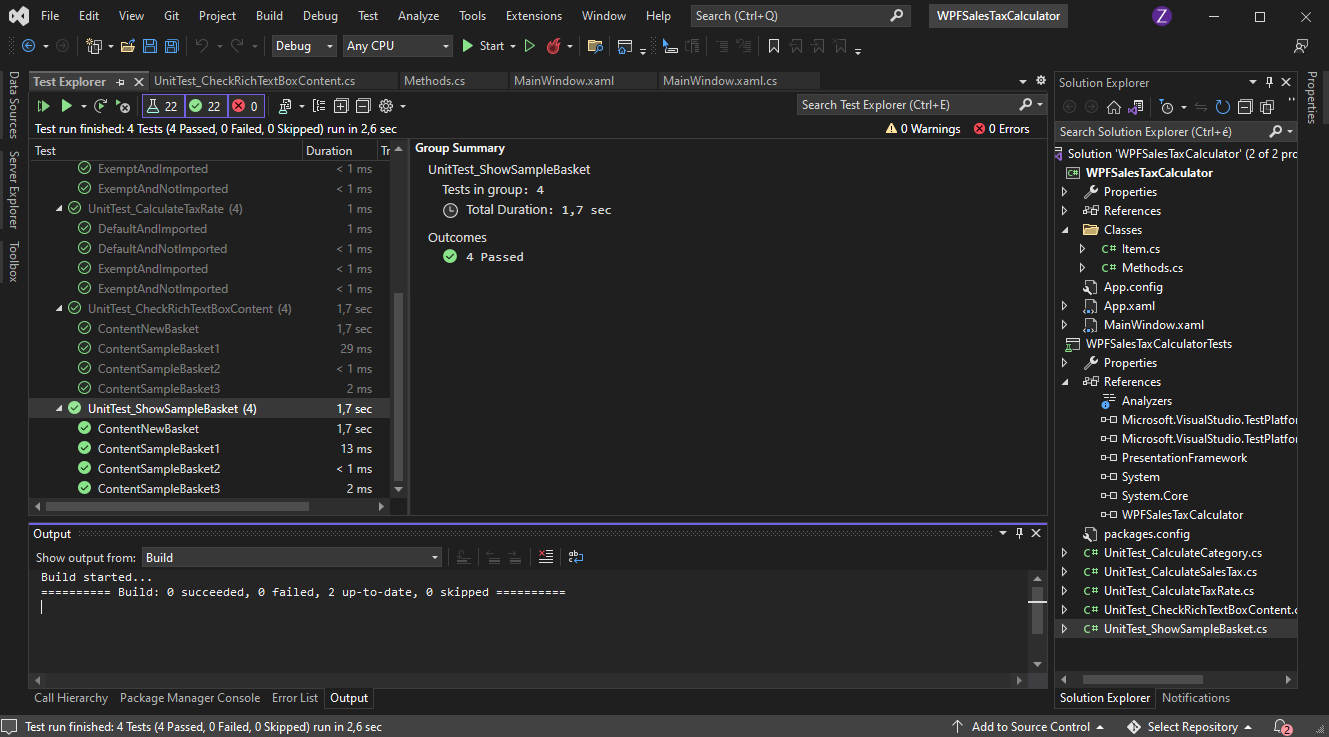
Add RichTextBox and lists to the test class used by the ShowSampleBasket method as parameters



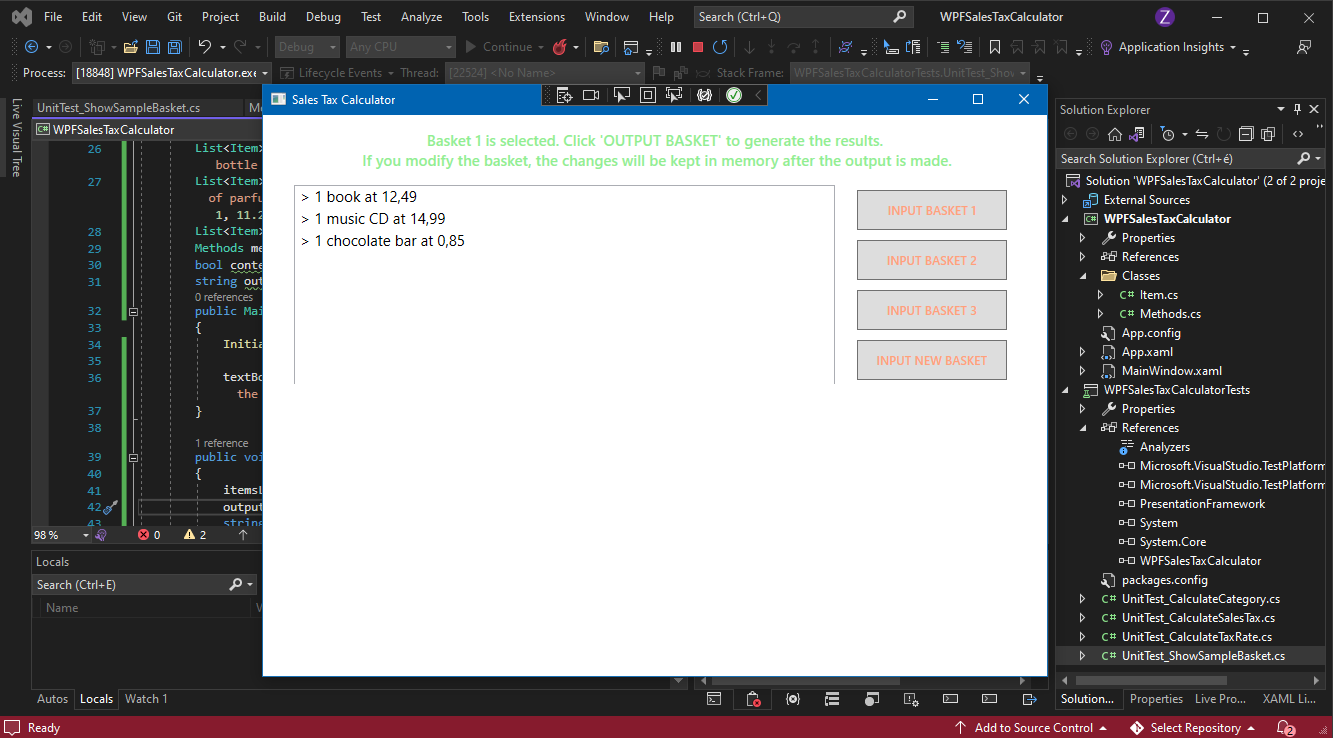
Change actual value of test methods to returned values of ShowSampleBasket method



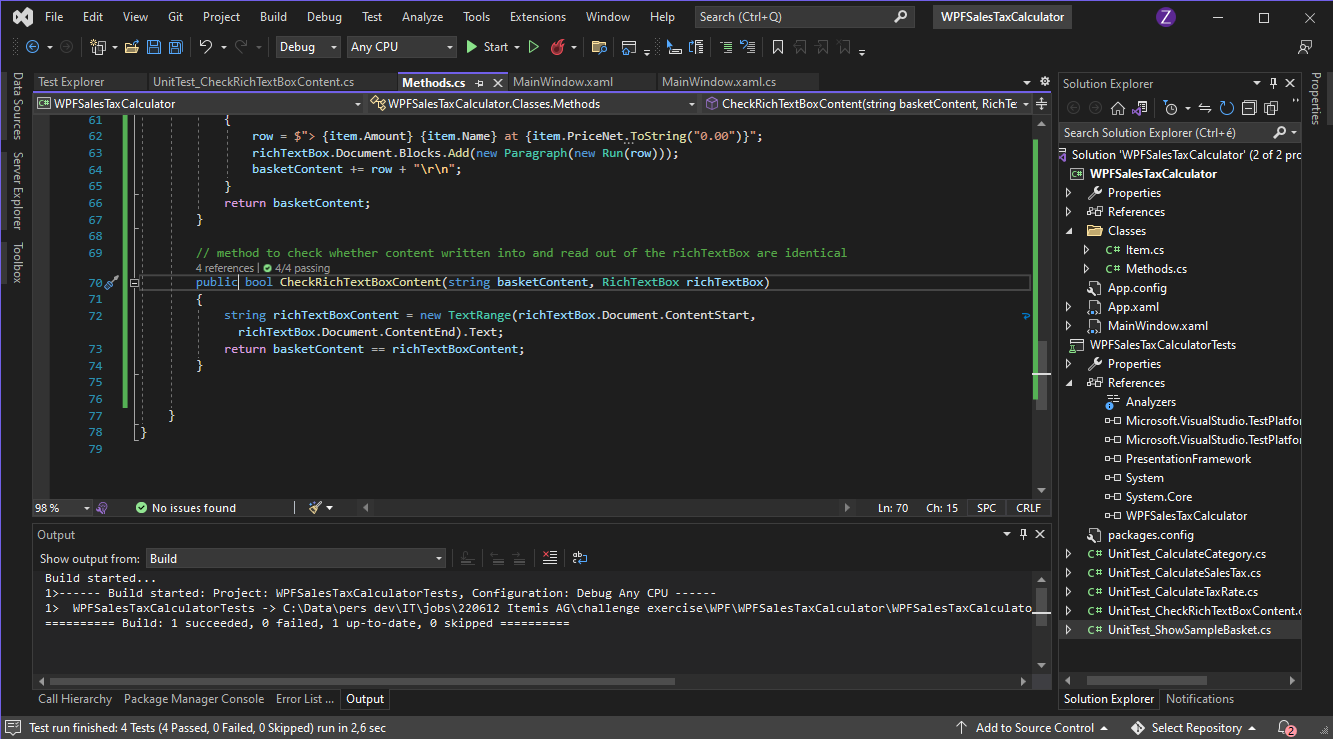
Successful tests for ShowSampleBasket method



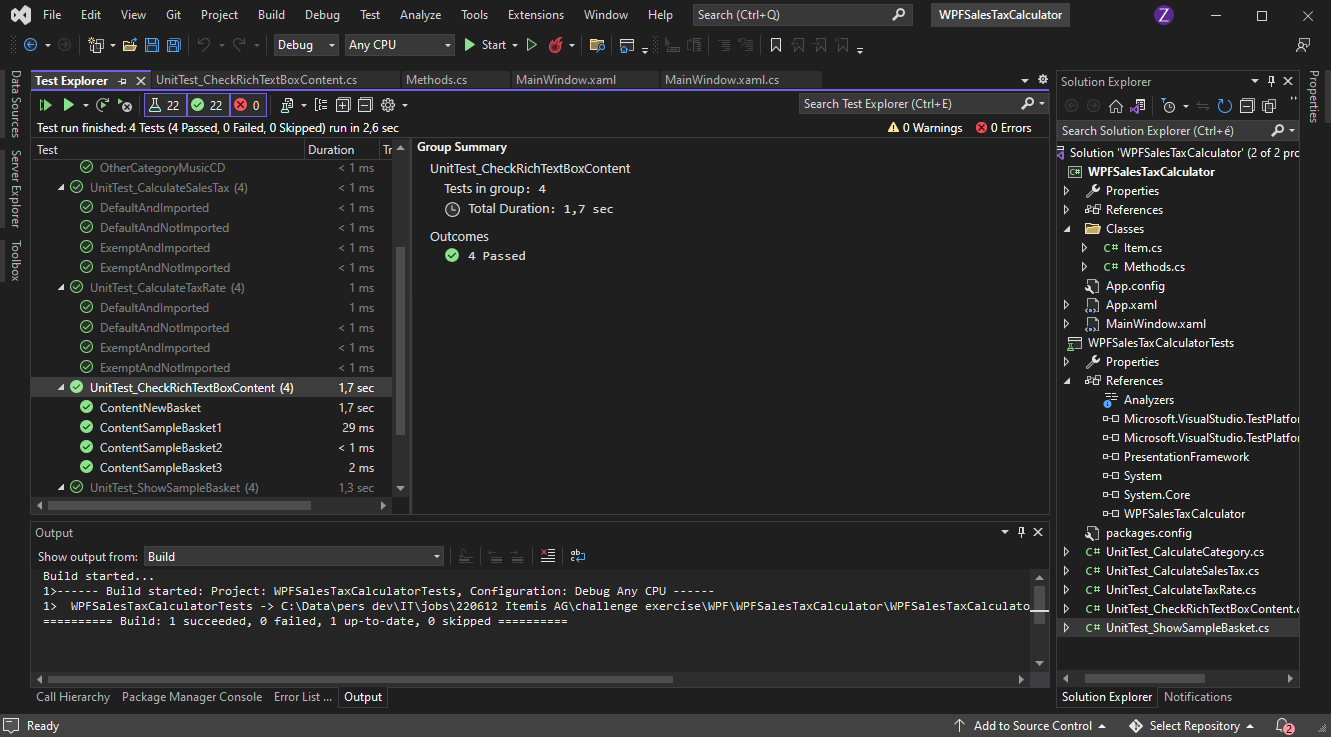
Display content of sample basket 1



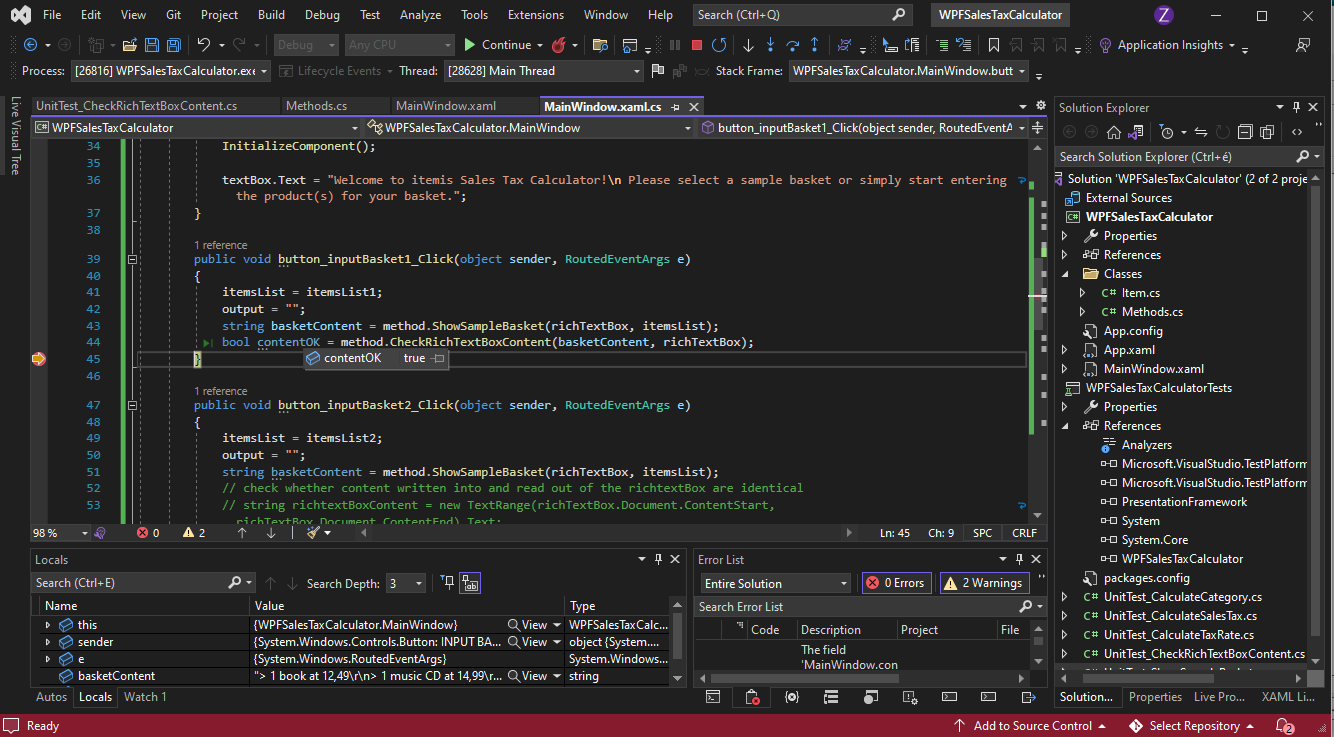
Add new CheckRichTextBoxContent method to test



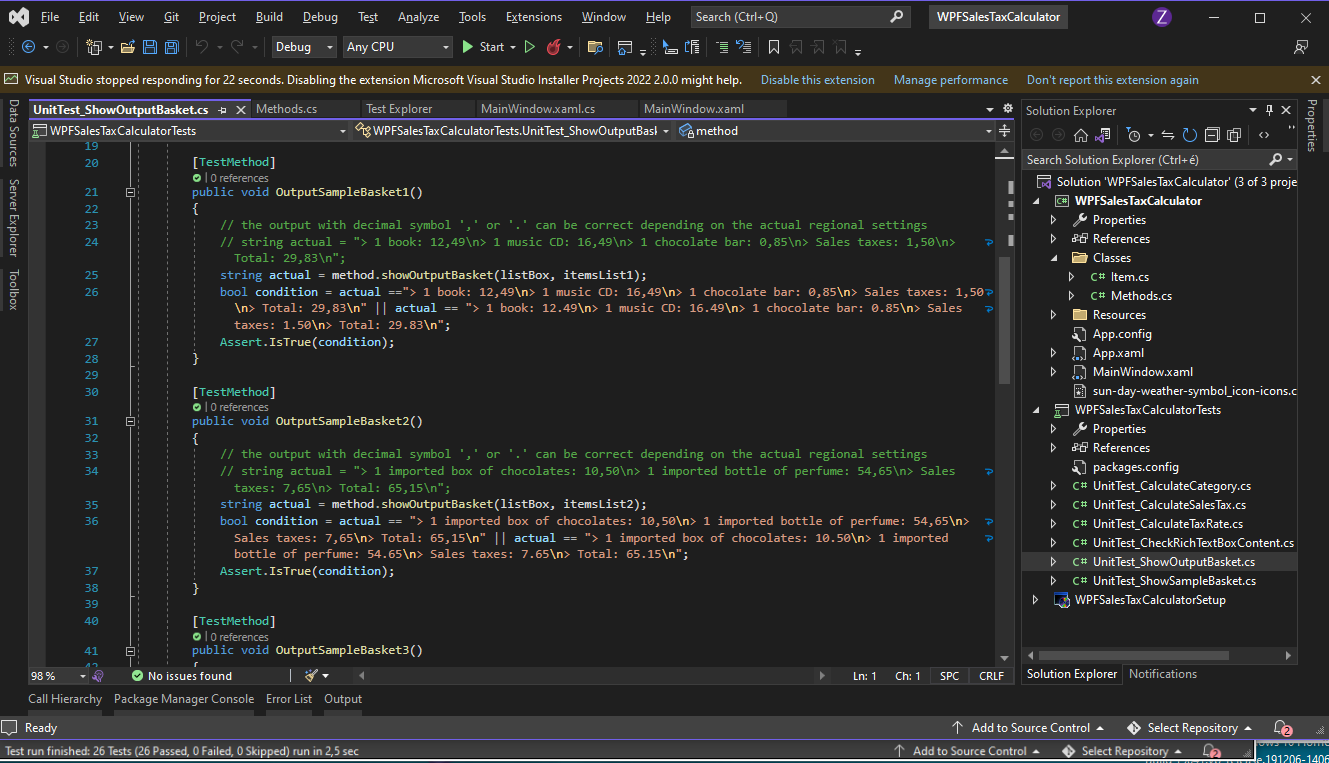
Successful tests for CheckRichTextBoxContent method



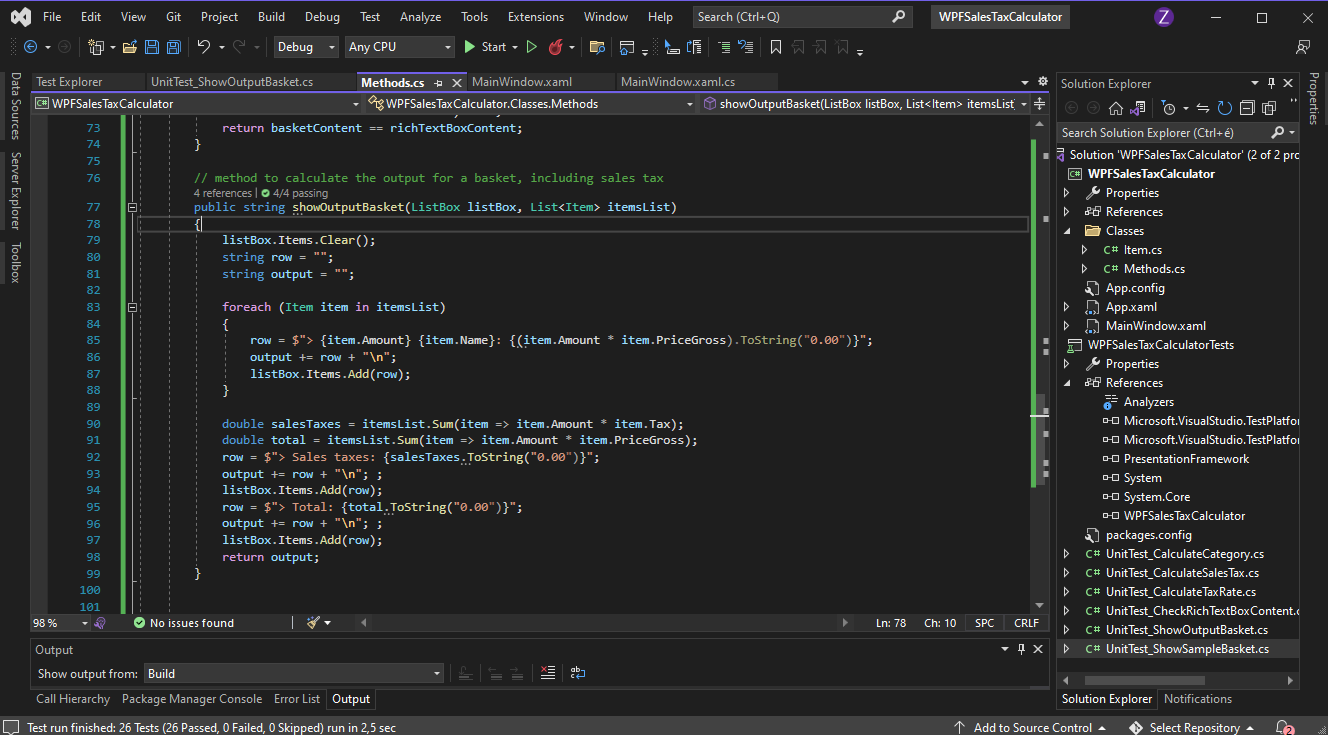
Implementing and checking result of CheckRichTextBoxContent method in project



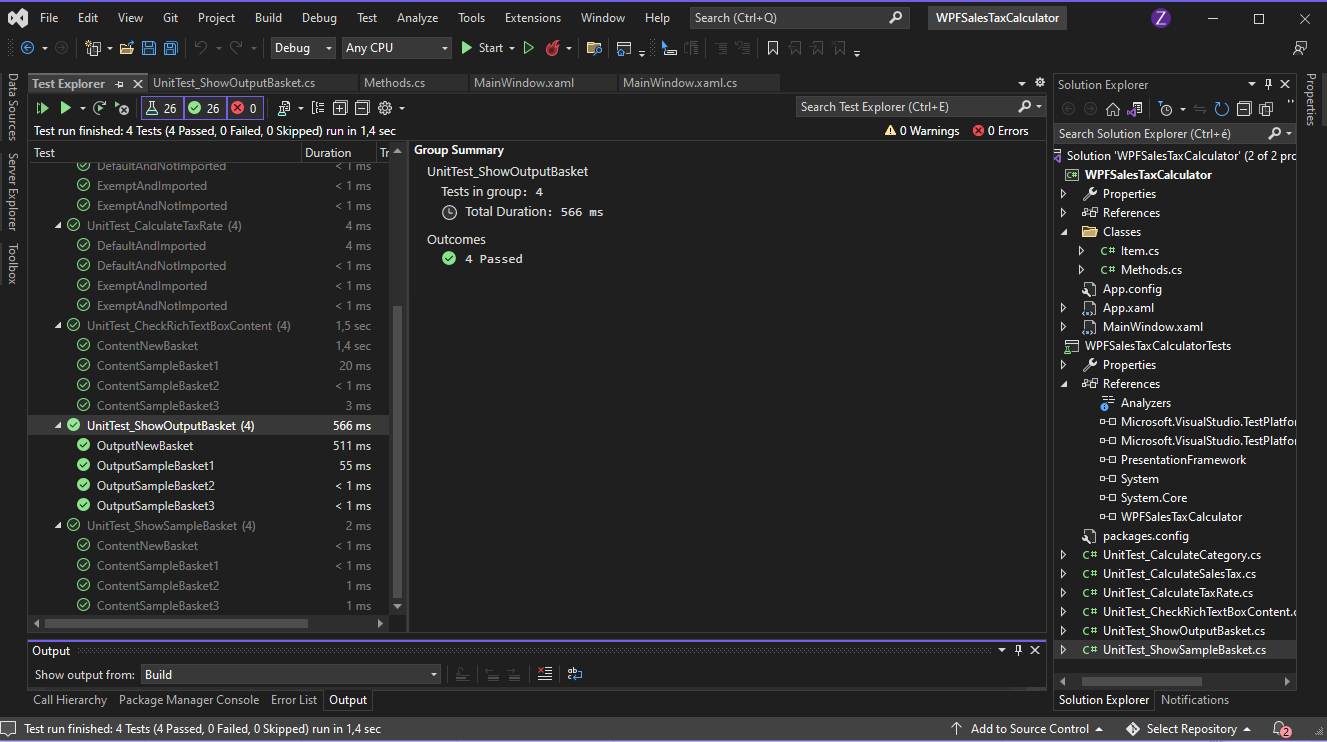
Create initial test methods for ShowOutputBasket method (screenshot was made afterwards)



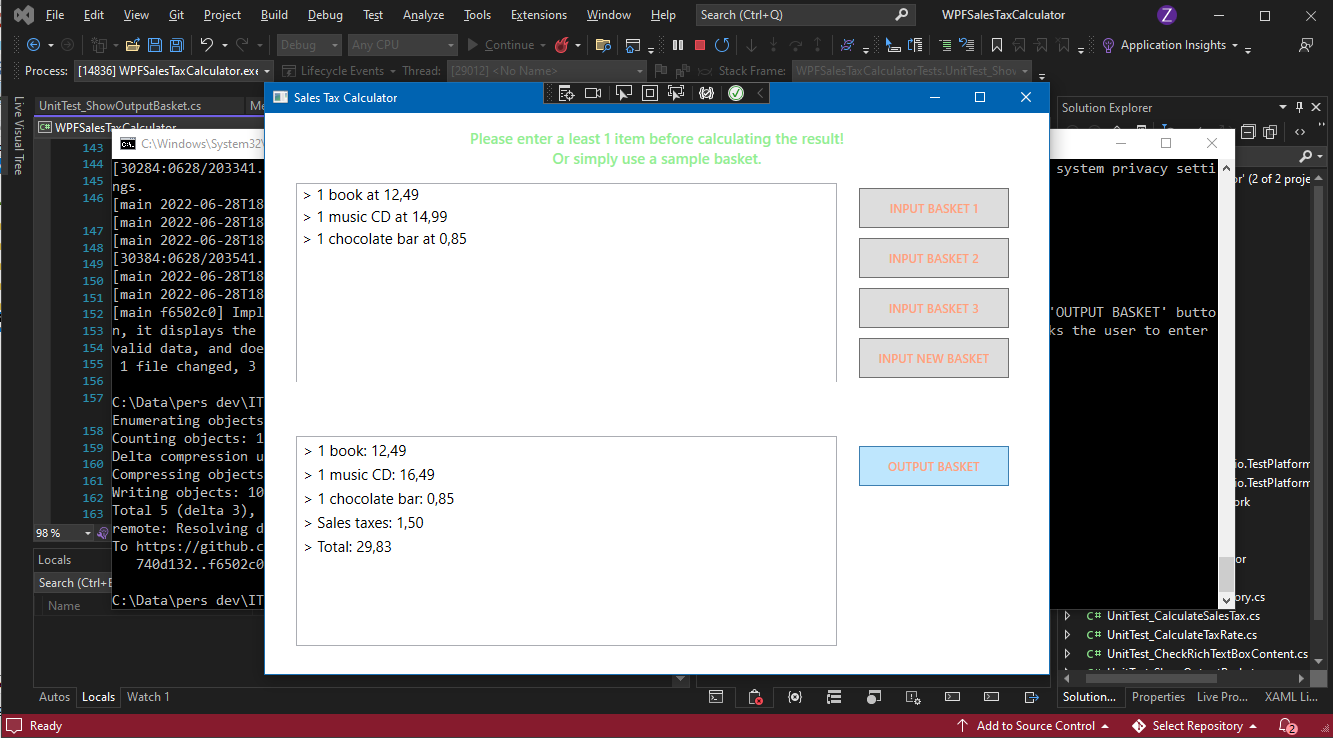
Create new ShowOutputBasket method to test



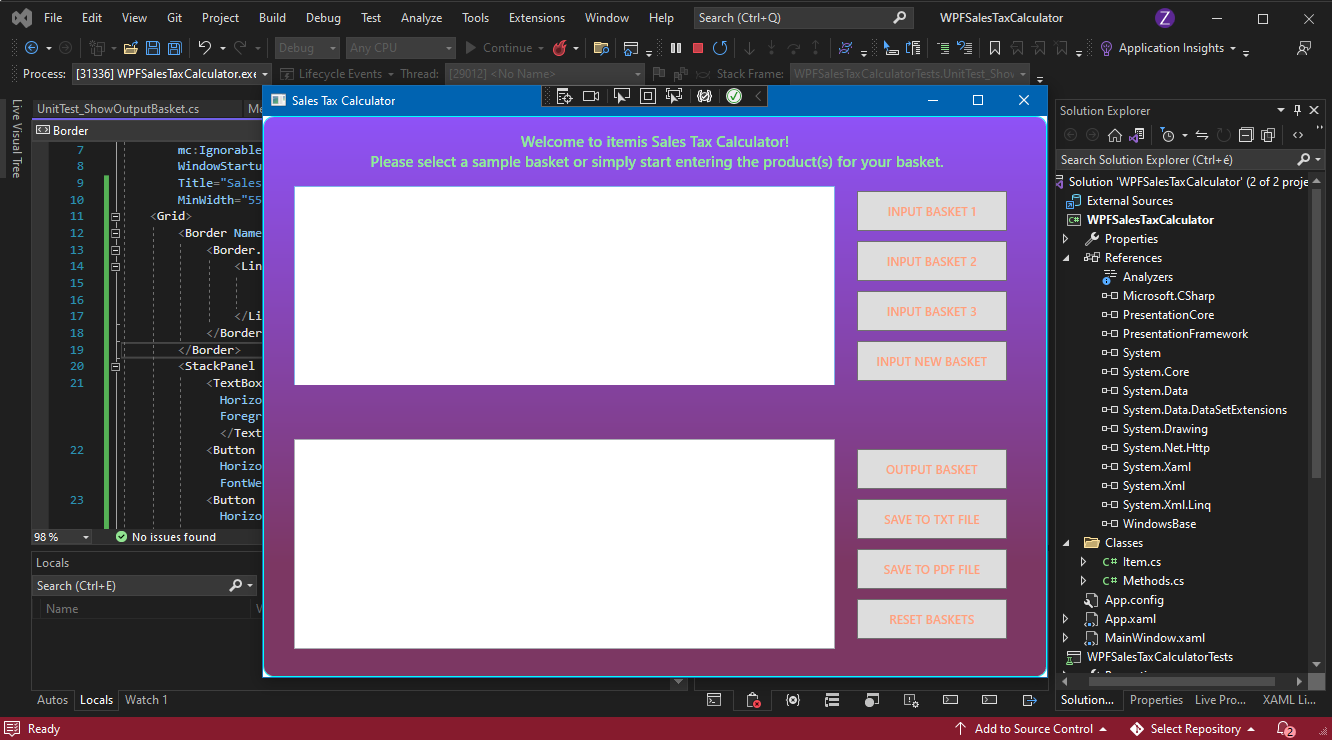
Successful tests for ShowOutputBasket method



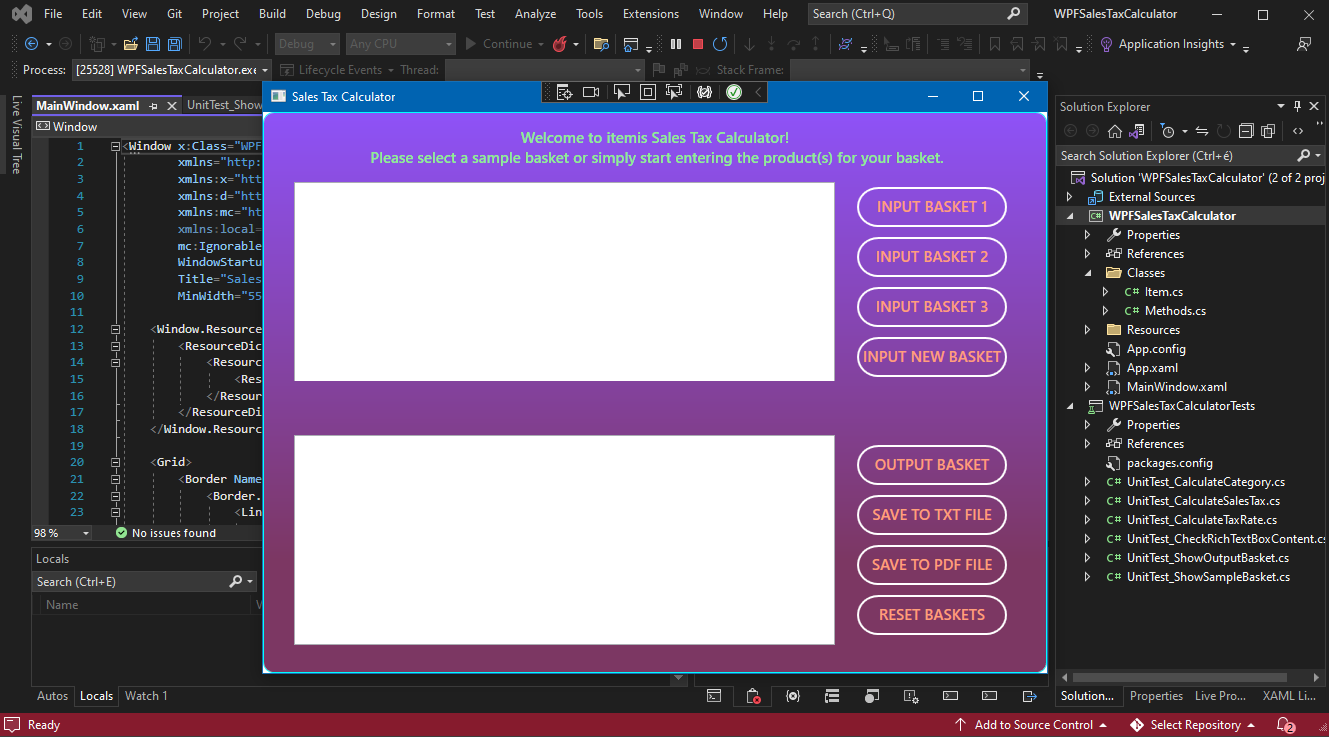
Calculate correct output for sample basket 1



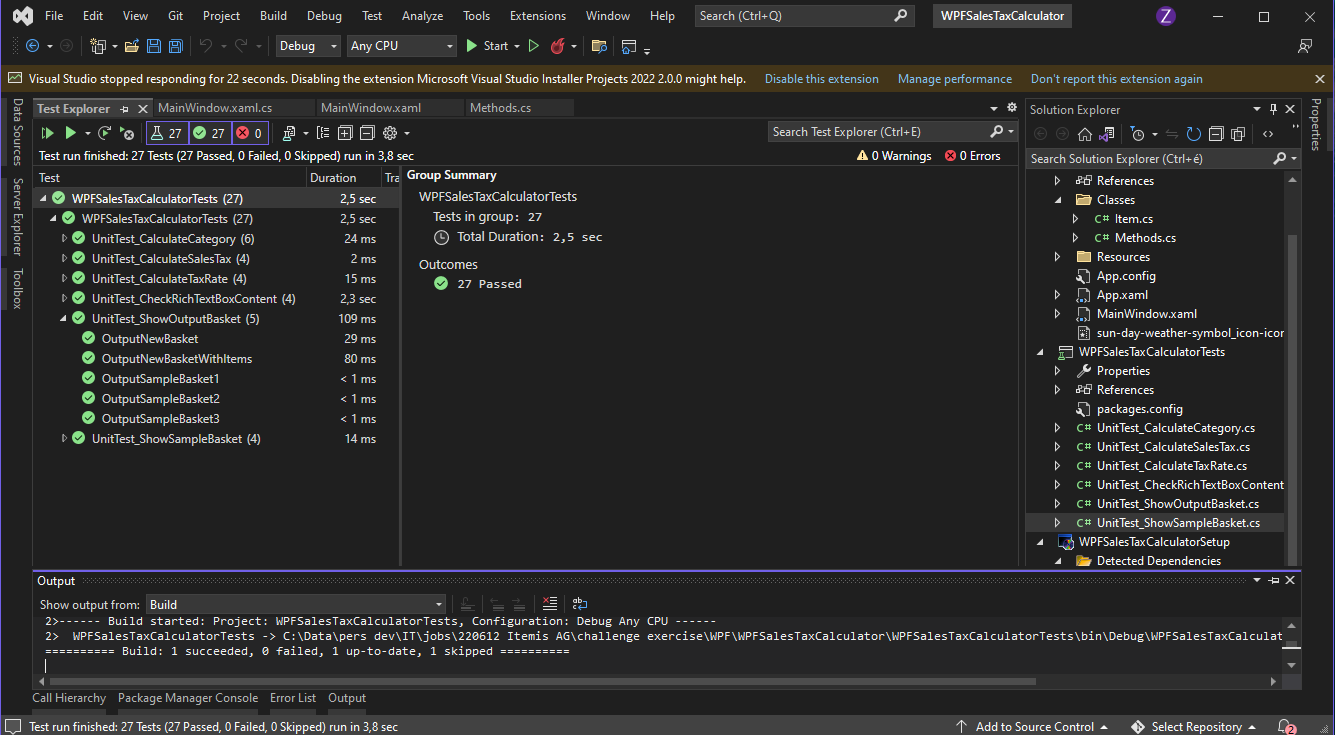
Add further buttons, and a border with background color design



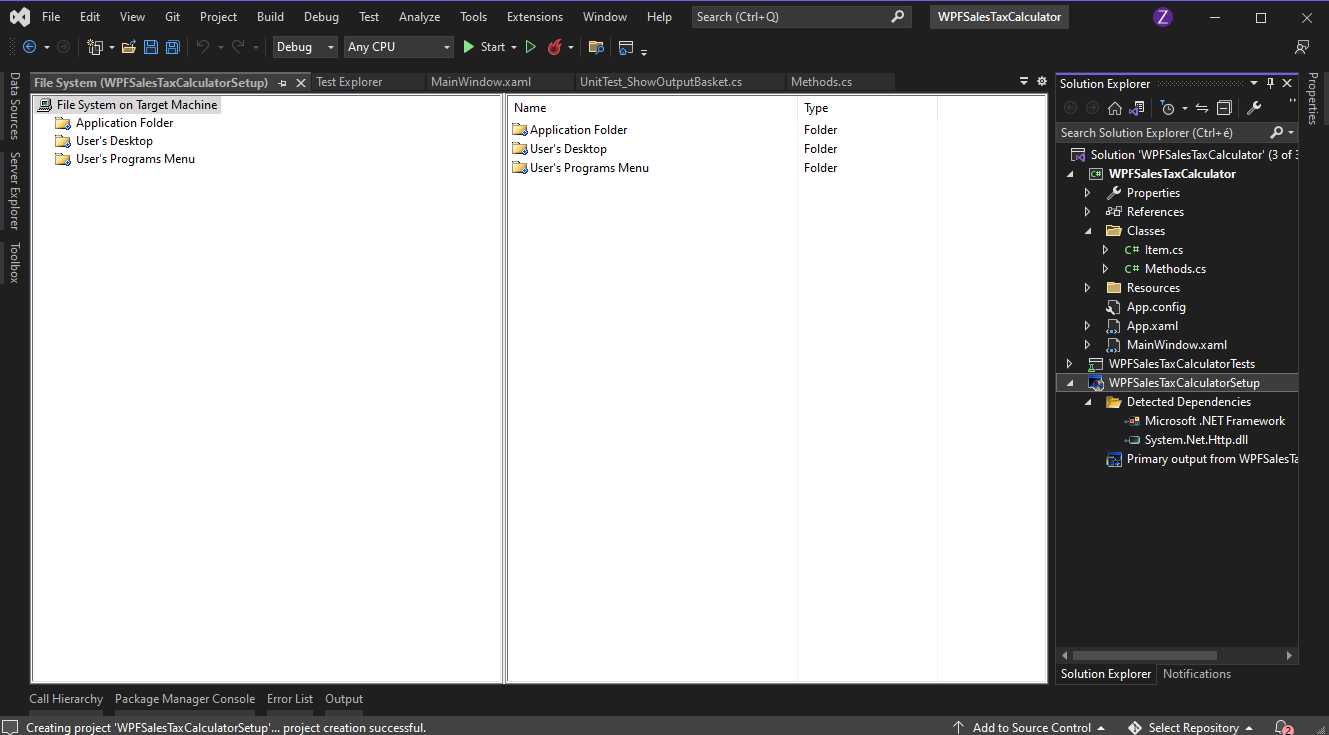
Add button design with rounded corners:



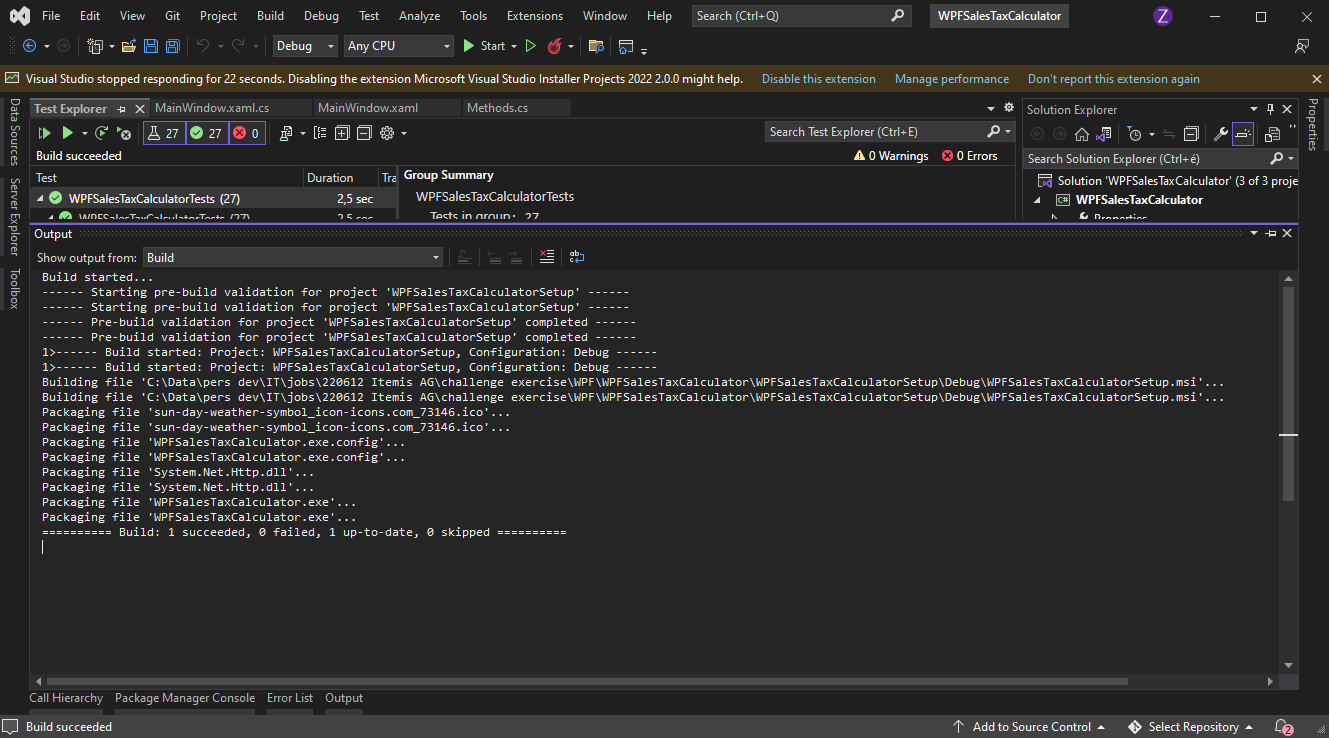
All 27 unit tests successful



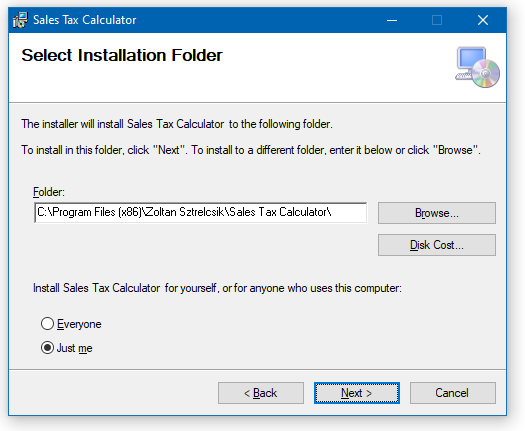
Add Windows Installer Project 2022



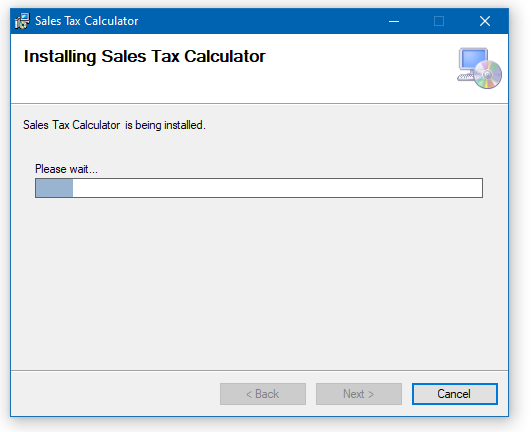
After configuration, build the setup project:



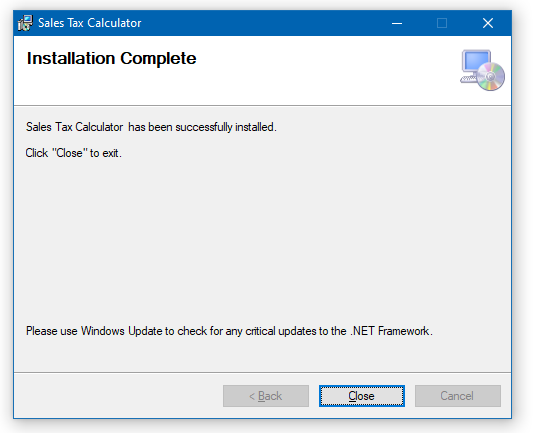
Installation screen



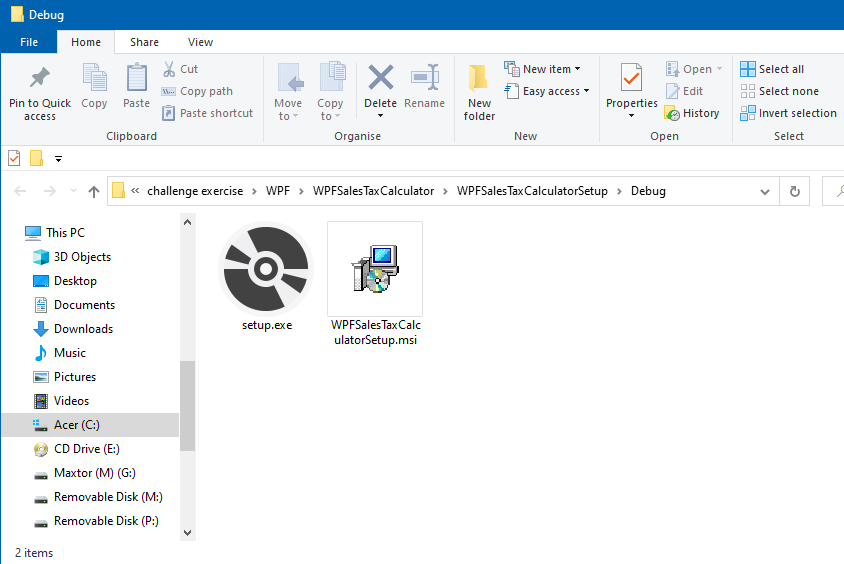
Installation progress



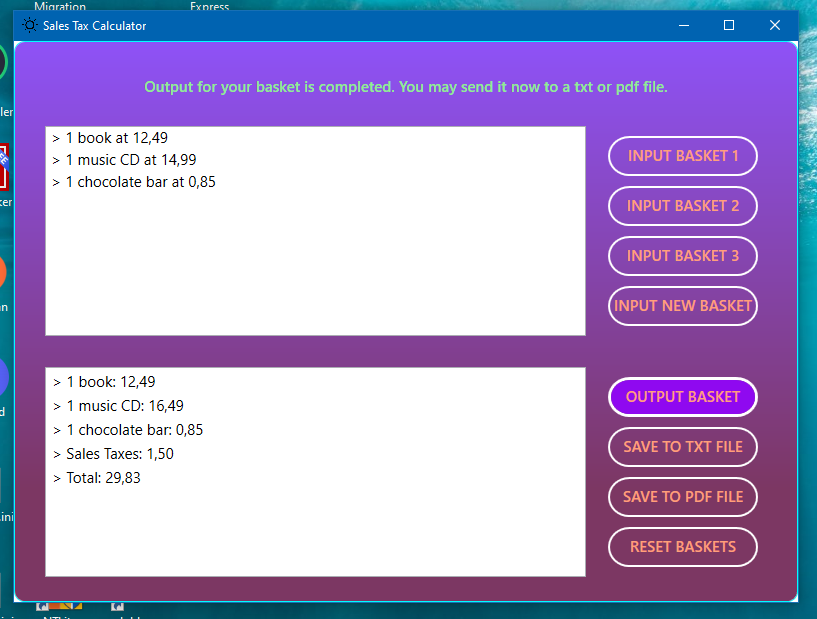
Installation complete



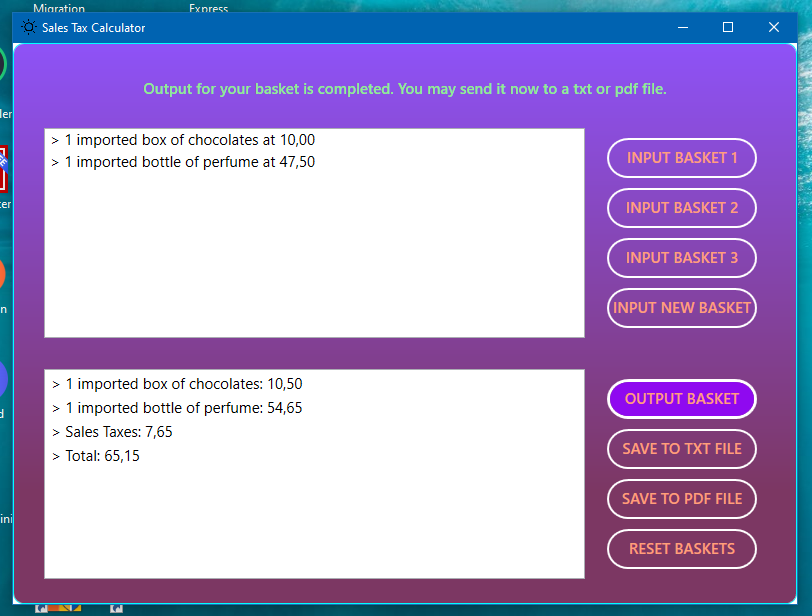
Installation files



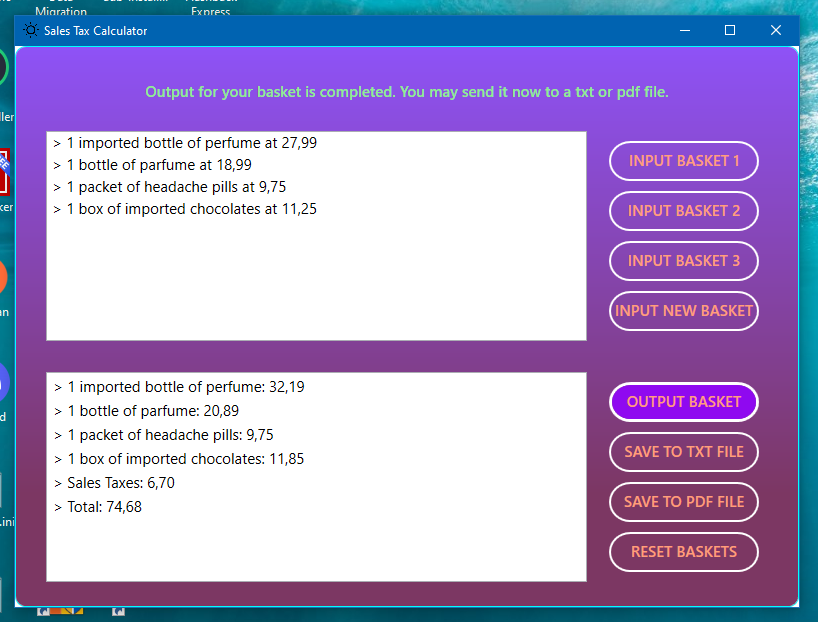
Output for sample basket 1



Output for sample basket 2



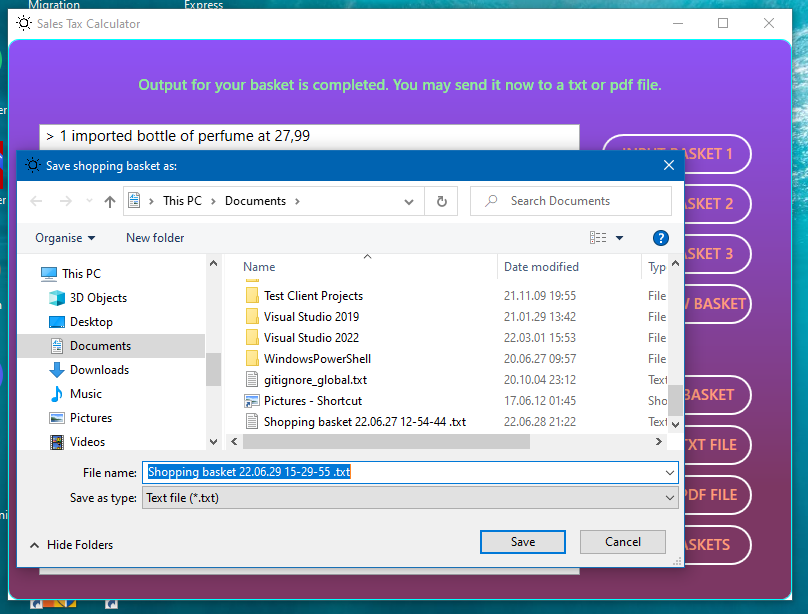
Output for sample basket 3



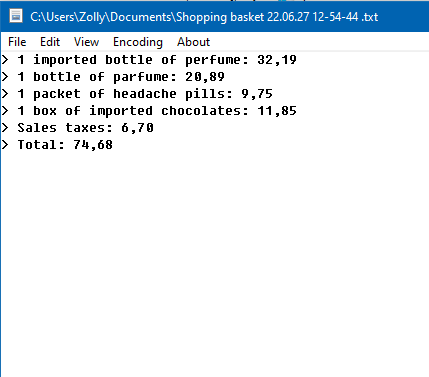
Output for a new basket (with various amount of products)



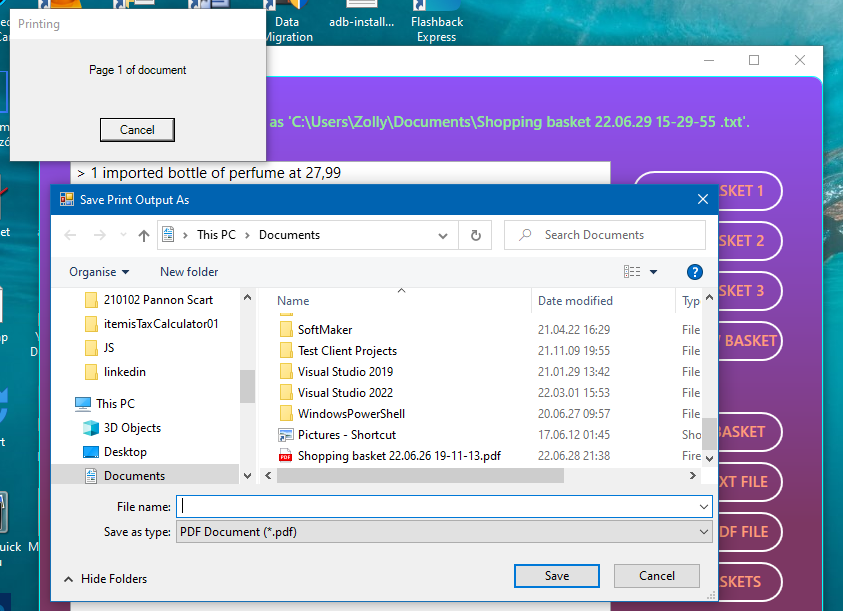
Save to file dialog:



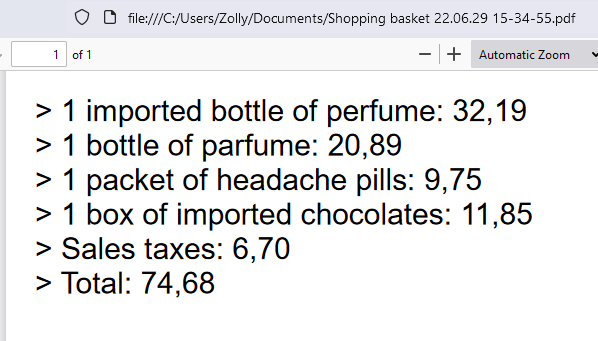
Content of txt file output for sample basket 3



Save to pdf file dialog



Content of pdf file output for sample basket 3



Best Regards,

Zoltan Sztrelcsik