Implementation Scenario:

Create a service which can initialize with different Tax Calculators depending on the customer type and calculate the tax for following two cases

* Get the Tax rates for a location
* Calculate the taxes for an order

# Implementation Detail:

Visual Studio Solution Contains Following Projects

**TaxCalculator**

This is a console application which is used to test the service methods. It has App.Config file with the settings for TaxJar API.  
  
**TaxCalculator.Common**

This is a class library project. Which has the interface and Models used by the Service and Client Projects.   
  
**TaxCalculator.Service**

Tax Service is a class library project. It has a Tax Service class which implements the ITaxService Interface. Services is initialized by passing the customer type (enum value) in the constructor for the service. Depending on the customer type service creates the client for a specific tax calculator.  
  
**TaxCalculator.TaxClients**

This is a class library project. It has the clients for different tax calculator types. Currently TaxJar is implemented with the two methods, HRBlock client is added with only class definition to show a different tax calculator.  
  
**TaxCalculator.Service.Tests**

This is a unit test project with tests for Tax Calculator Service.  
  
**TaxCalculator.TaxClients.Tests**

This is a unit test project with tests for Tax Calculator Clients.

UML Diagram for the project TaxCalculator\_UML.pdf is included in the solution folder.