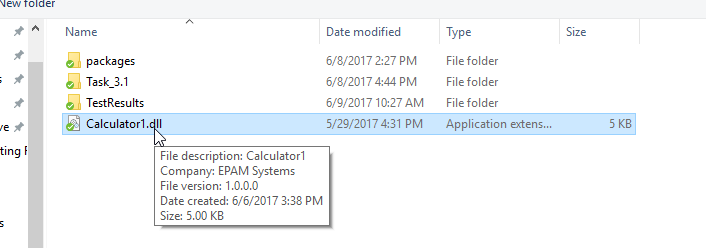
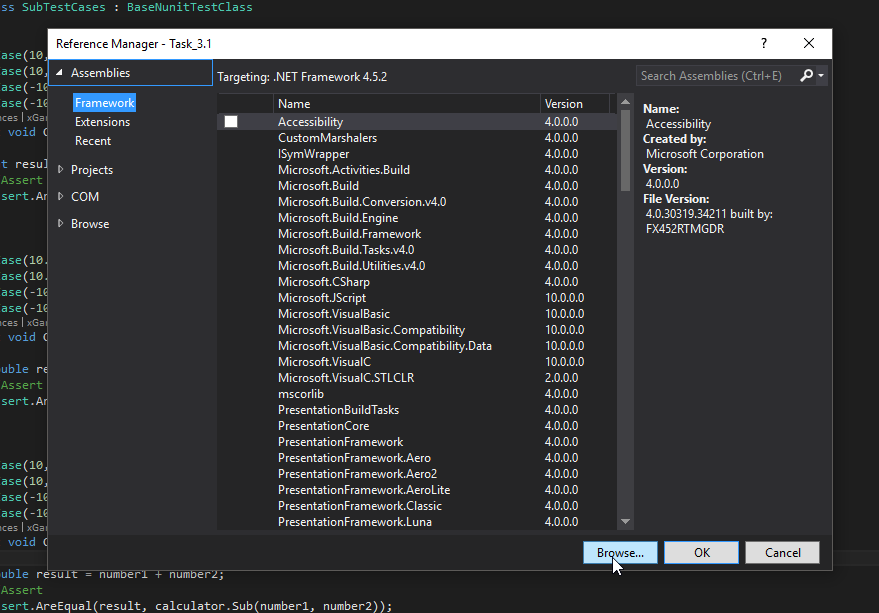
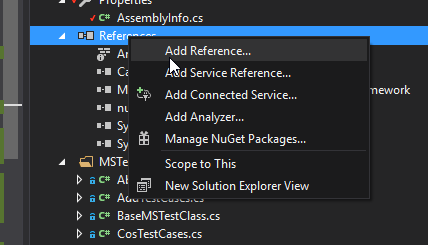
1. Add Calculator.dll to your project reference

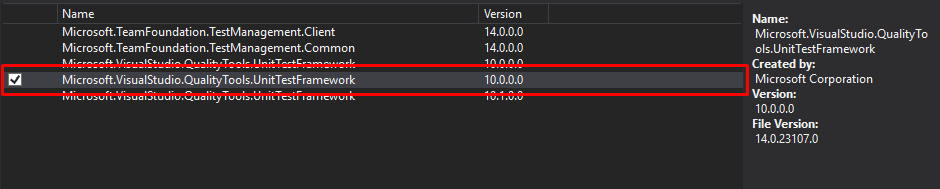


1. Write tests using Nunit and MSTest

To have ability write tests we should install needed packages (for NUnit) and add needed libraries.

* MSTest

To have ability use MSTest we should add following reference to project

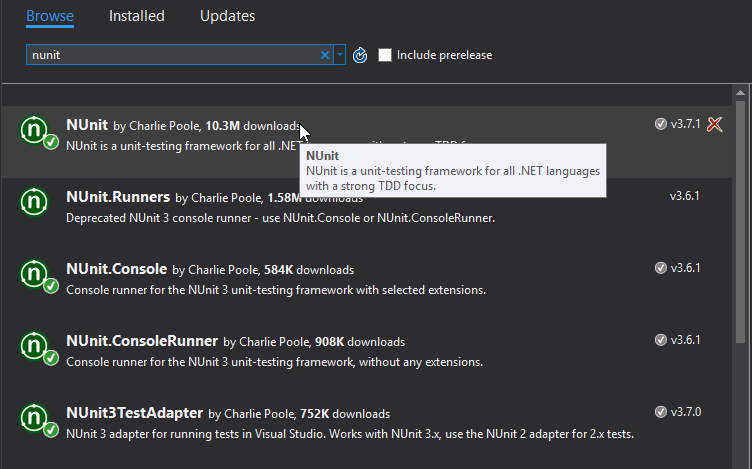
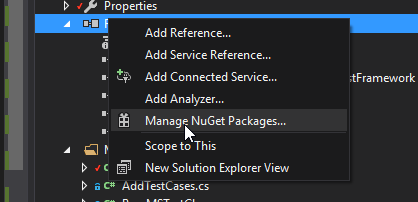


And also add library

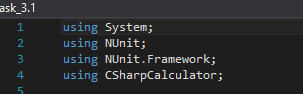


* NUnit

Install needed packages from NuGet

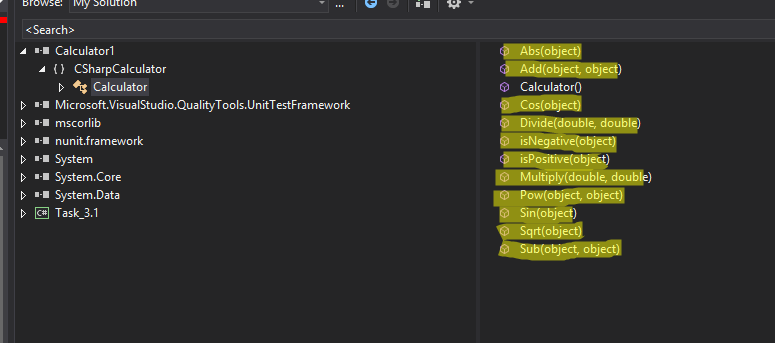


Add library

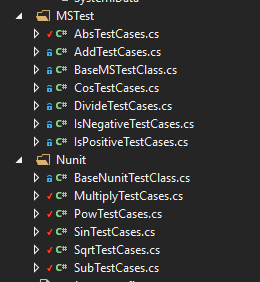


1. Each class should test only one method of the library

As we can see from object browser our Calculator1.dll contains 11 methods



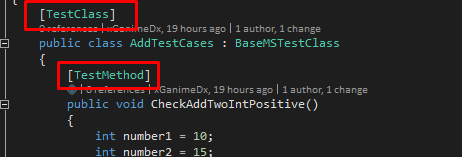
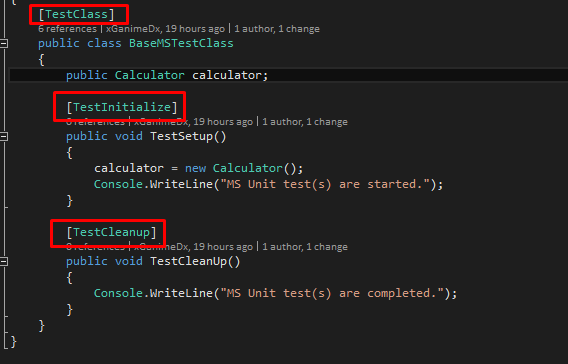
I separated test classes between MSTest and NUnit and wrote for each method separate class



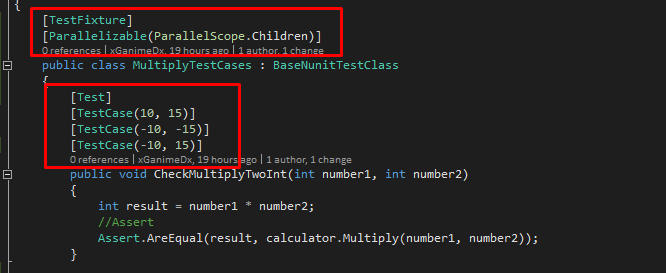
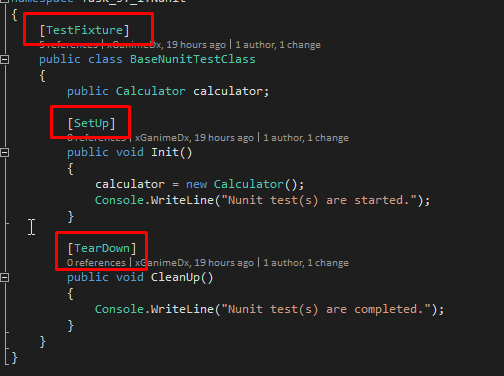
Base classes were created for simplicity creating tests.

1. Tests pre-conditions and post-conditions should be implemented with appropriate annotations

* MSTest

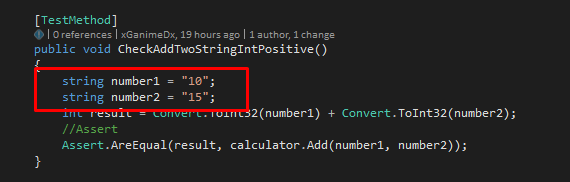
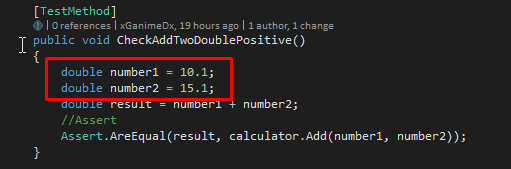


* NUnit

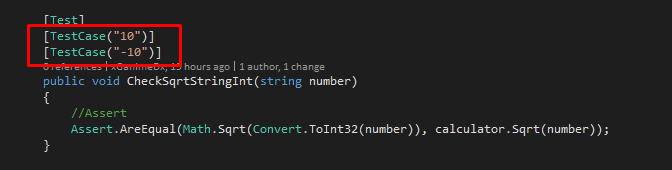
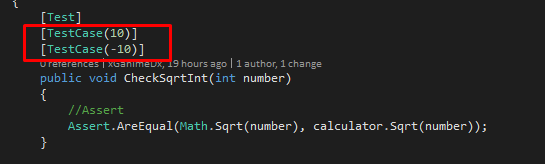


1. Tests should have ability to enter numbers as well as string numbers

* MSTest

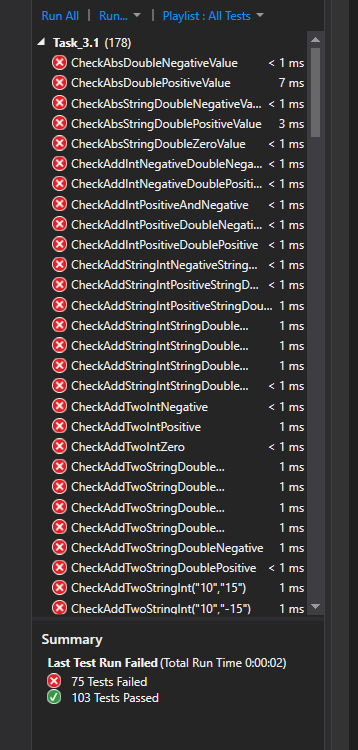
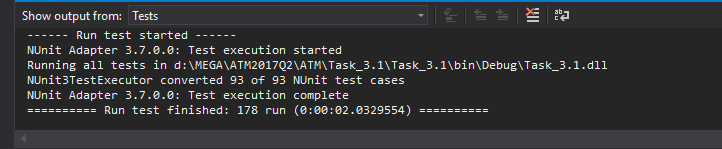


* NUnit



1. Run tests in Visual Studio

To run tests from VS we need to built our project, open Test Explorer and run needed tests



1. Run tests in command line (MSTest, NUnit-console) with specified output directory for results.

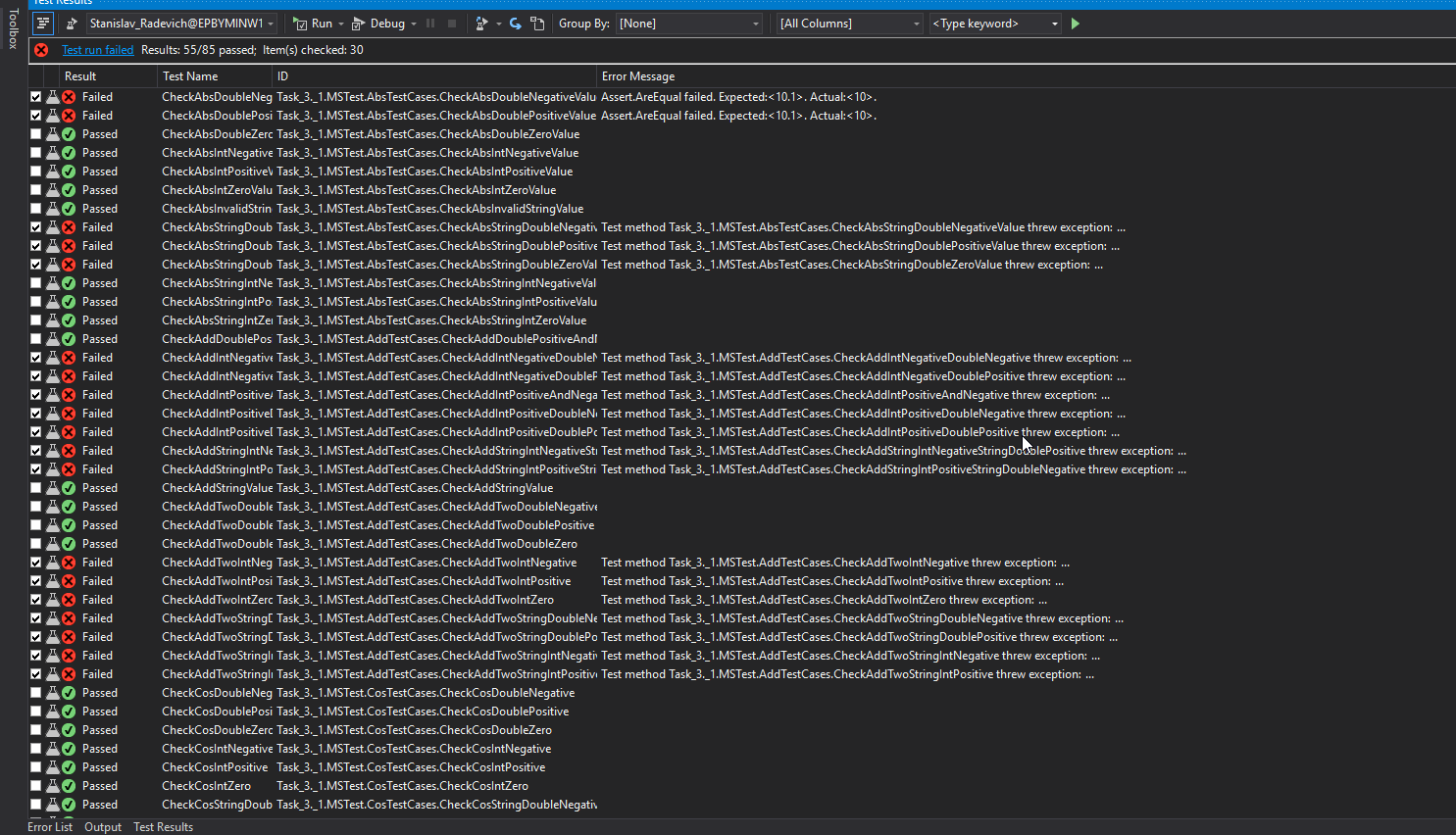
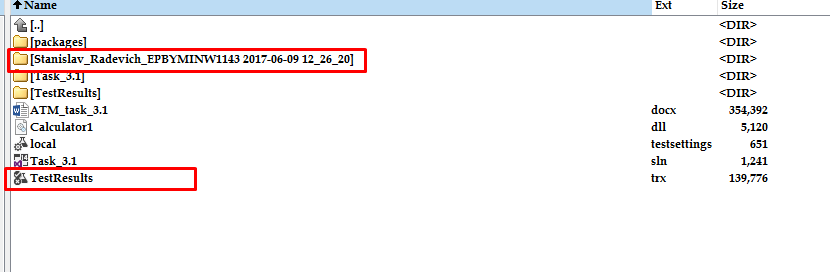
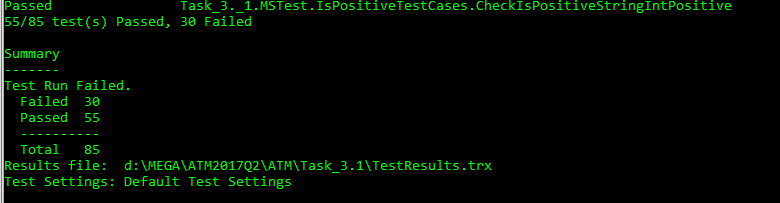
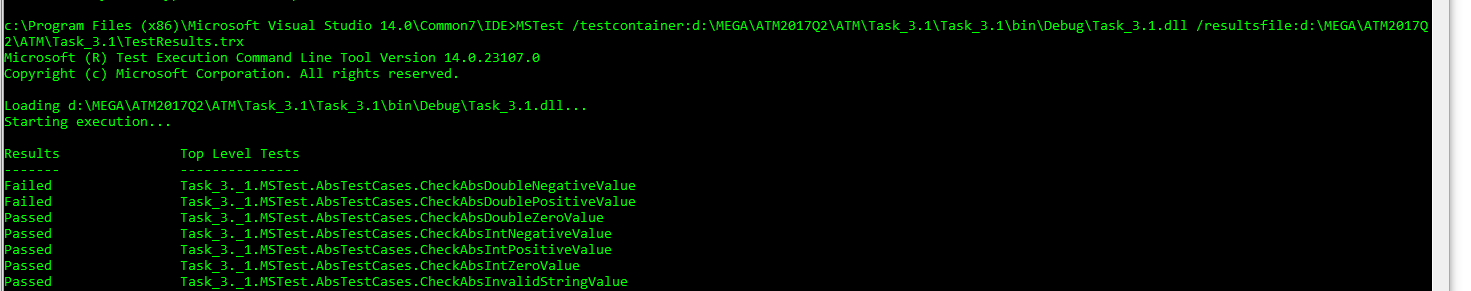
* MSTest

We should run CMD as administrator from (path can be different):

c:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\

And run following command:

MSTest /testcontainer:d:\MEGA\ATM2017Q2\ATM\Task\_3.1\Task\_3.1\bin\Debug\Task\_3.1.dll /resultsfile:d:\MEGA\ATM2017Q2\ATM\Task\_3.1\TestResults.trx



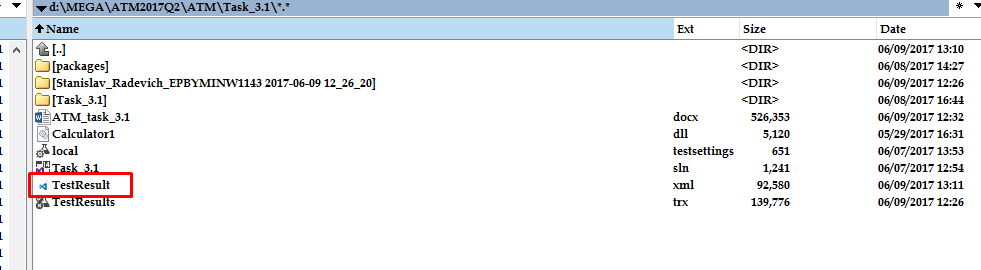
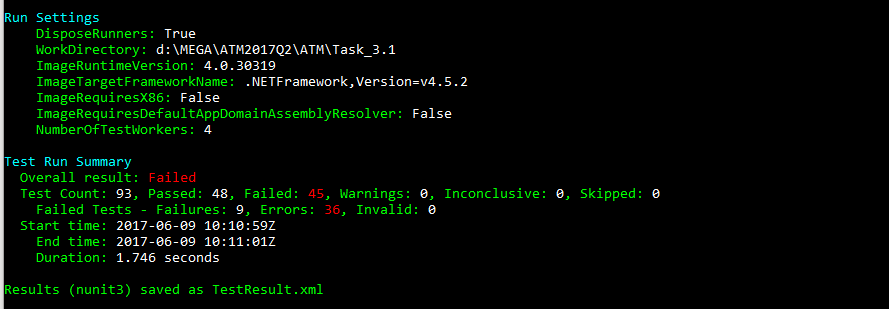
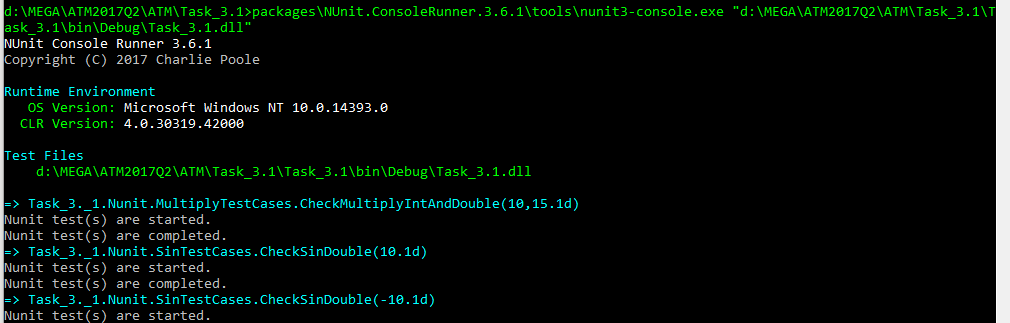
* NUnit

We should run CMD as administrator from (path can be different):

d:\MEGA\ATM2017Q2\ATM\Task\_3.1\

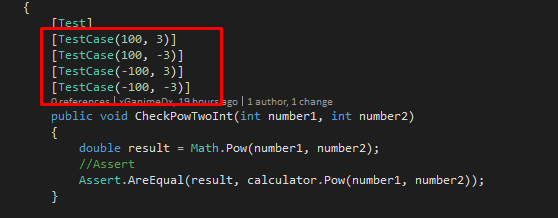
And run following command:

packages\NUnit.ConsoleRunner.3.6.1\tools\nunit3-console.exe "d:\MEGA\ATM2017Q2\ATM\Task\_3.1\Task\_3.1\bin\Debug\Task\_3.1.dll"



1. Create parametrized tests (data-driven approach)

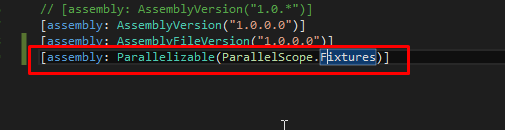
I’ve created DD tests using NUnit. For that we just need to add one more parameter to each test in which we want to use DD:



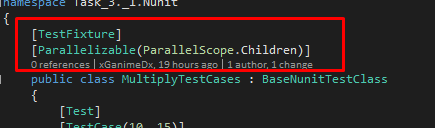
In this case test will use each pair of parameters one by one, so we can write one test, but check several values.

1. Configure and execute tests in parallel mode

I’ve run in parallel mode NUnit tests, for that we need add one string in AssemblyInfo.cs file



And add one more parameter to test classes



I use “Children” because I want to parallel tests from one TestFixture. Also we can use “Fixtures” to parallel TestFixture’s from different classes and etc.