

# Lab Practicum #4

## Unit Testing

### Background:

A unit test is a piece of code written by a developer that exercises a very small, specific area of functionality of the code being tested.

Usually a unit test exercises some particular method in a particular context. For example, you might add a large value to a sorted list, then confirm that this value appears at the end of the list. Or you might delete a pattern of characters from a string and then confirm that they are gone.

Unit tests are performed to prove that a piece of code does what the developer thinks it should do.

### Goal of the lab:

Consolidate theoretical knowledge by writing unit tests for .NET collection classes.

### Task description:

- 1) Open Example Solution, study written unit tests. Run them using NUnit.
- 2) Create new test solution in Visual Studio 2008 (using NUnit or Microsoft unit testing framework available in professional version)
- 3) Write 10 unit tests for one of the following .NET classes:
  - a. Stack<>
  - b. Queue<>
  - c. List<>
  - d. SortedList<,>
  - e. Dictionary<,>
  - f. LinkedList<>
  - g. Hashtable
  - h. ArrayList

- 4) Use following guidelines:

- a. Test the most common functions (Add, Remove, Sort, Contains)
- b. Test with boundary values (null, 0, -1, int.MaxValue)
- c. Test exceptional situations (NullReferenceException, ArgumentOutOfRangeException, etc)

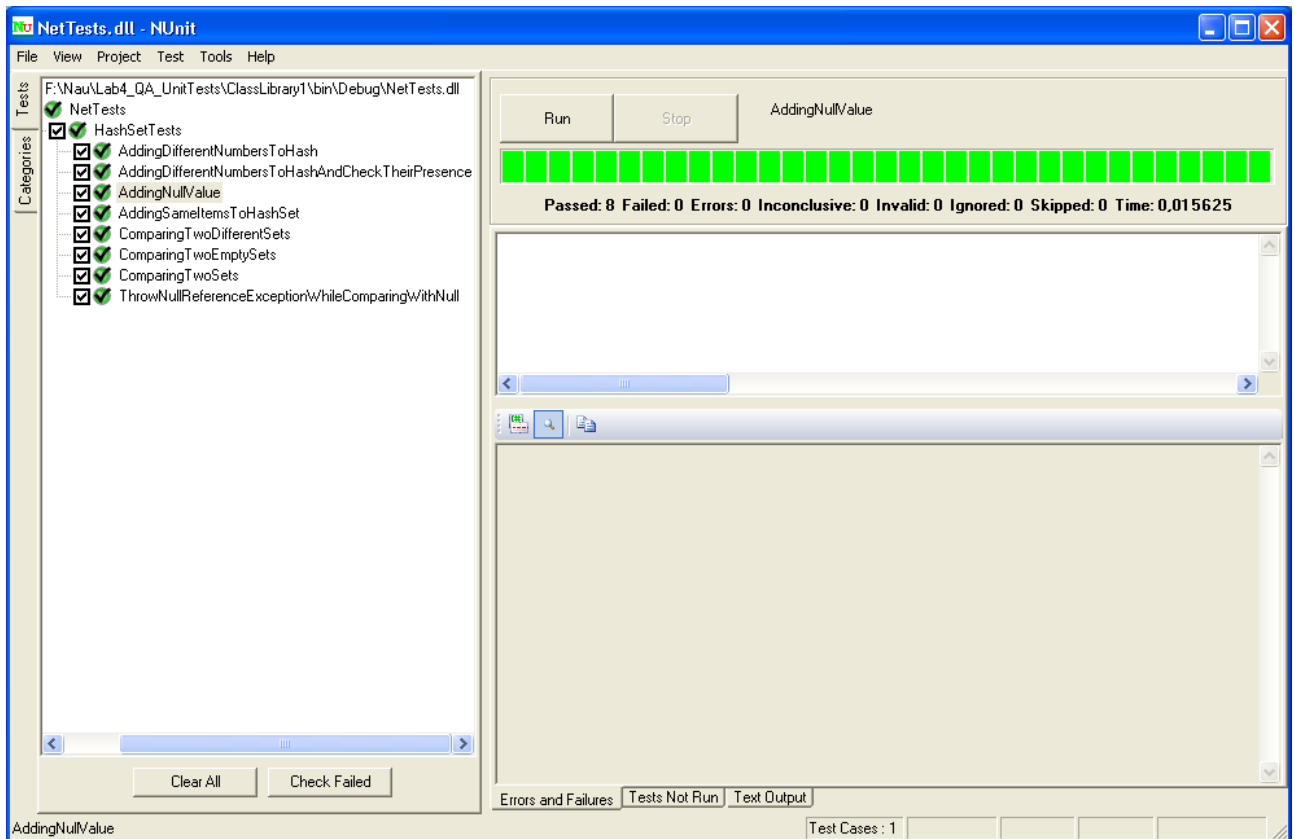
### **Outcome:**

- 1) Unit Tests project.
- 2) Test results and test description document

### **Input Materials:**

- 1) Andrew Hunt, David Thomas, Pragmatic Unit Testing In C# with NUnit,
- 2) <http://geosoft.no/development/unittesting.html>
- 3) [http://en.wikipedia.org/wiki/Unit\\_testing](http://en.wikipedia.org/wiki/Unit_testing)
- 4) <http://www.nunit.org/index.php>
- 5) Example Solution.

## Test results and test description



### Test ComparingTwoSets:

Tests **SetEquals** function from `HashSet<>` class. Ensure that two hash sets built from different arrays of numbers are equals as sets.