**ProjectA**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

/// <summary>

/// PROG2070 – Quality Assurance: Winter 2017

/// Assignment 03

/// Yoonsuk Cho Mar 14, 2017

/// </summary>

namespace ProjectA

{

/// <summary>

/// main class of assignment 3

/// Project A

/// </summary>

class Program

{

/// <summary>

/// program start main method

/// using TimeConversion class

/// it will convert time unit

/// this program will not check a time unit and time result

/// </summary>

/// <param name="args">parameters of start method</param>

static void Main(string[] args)

{

bool doCont = true;

while (doCont)

{

double inputTime = 0.0d;

Console.WriteLine("=========================================");

Console.WriteLine("====== Time Converter ProjectA ======");

Console.WriteLine("=========================================");

Console.WriteLine();

Console.Write("Enter a time: ");

string timeStr = Console.ReadLine();

if (Double.TryParse(timeStr, out inputTime))

{

Console.Write("Enter a time unit (convert from): ");

string convertFrom = Console.ReadLine();

Console.Write("Enter a time unit (convert to): ");

string convertTo = Console.ReadLine();

Console.WriteLine();

try

{

double convertedValue = TimeConversion.Convert(inputTime,

convertFrom,

convertTo);

Console.WriteLine("{0}({1}) converts to {2}({3}).",

timeStr,

convertFrom,

convertedValue,

convertTo);

}

catch (ArgumentException ae)

{

Console.WriteLine(ae.Message);

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

else

{

Console.WriteLine("Invalid number input.");

}

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Press Enter to continue, or any other key to exit.");

if (Console.ReadKey().Key == ConsoleKey.Enter)

{

doCont = true;

Console.Clear();

}

else

{

doCont = false;

Console.Clear();

Console.WriteLine();

Console.WriteLine(" =================");

Console.WriteLine(" BYE!!!");

Console.WriteLine(" =================");

Console.ReadKey();

}

}

}

}

}

**ProjectA**

**TimeConversion.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

/// <summary>

/// PROG2070 – Quality Assurance: Winter 2017

/// Assignment 03

/// Yoonsuk Cho Mar 14, 2017

/// </summary>

namespace ProjectA

{

/// <summary>

/// Time Conversion class of assignment 3

/// </summary>

public static class TimeConversion

{

/// <summary>

/// get converted time value

/// from convertFrom time unit to convertTo time unit

/// with ModifyInputStub(fixed return) and GetMultiplierStub(fixed return)

/// </summary>

/// <param name="value">numeric time value</param>

/// <param name="convertFrom">time unit convert From</param>

/// <param name="convertTo">time unit convert to</param>

/// <returns></returns>

public static double Convert(double value,

string convertFrom,

string convertTo)

{

convertFrom = ModifyInputStub(convertFrom);

convertTo = ModifyInputStub(convertTo);

double multiplier = GetMultiplierStub(convertFrom, convertTo);

return value / multiplier;

}

/// <summary>

/// modify time unit

/// from various input time unit

/// to specific time unit sign for multiply

/// this method will return fixed value

/// </summary>

/// <param name="input">input time unit</param>

/// <returns>specific time unit sign</returns>

private static string ModifyInputStub(string input)

{

return "seconds";

}

/// <summary>

/// get multiplier factor

/// using convertFrom time unit to convertTo time unit

/// this method will return fixed value

/// </summary>

/// <param name="convertFrom">convert From time unit</param>

/// <param name="convertTo">convert To time unit</param>

/// <returns></returns>

private static double GetMultiplierStub(string convertFrom,

string convertTo)

{

return 1.0d;

}

}

}

**ProjectA.Tests**

**TimeConversionATest.cs**

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using NUnit.Framework;

/// <summary>

/// PROG2070 – Quality Assurance: Winter 2017

/// Assignment 03

/// Yoonsuk Cho Mar 14, 2017

/// </summary>

namespace ProjectA.Tests

{

/// <summary>

/// main class of assignment 3

/// Project A - Test Program

/// This class tests Program class of Project A

/// </summary>

[TestClass]

public class TimeConversionATest

{

/// <summary>

/// This method tests equal case of

/// Convert method With ModifyInputStub And GetMultiplierStub method Test

/// of TimeConversion class

/// </summary>

/// <param name="time">input time value</param>

/// <param name="convertFrom">time unit convert from</param>

/// <param name="convertTo">time unit convert to</param>

/// <param name="expectedValue">expected return time value</param>

///

// input: 1.0, hours, seconds, return: 1.

[TestCase(1.0, "Hours", "Seconds", 1)]

// input: 36, SSS, HHH, return: 36.

[TestCase(36, "SSS", "HHH", 36)]

public void ConvertAreEqualTest(double time,

string convertFrom,

string convertTo,

double expectedValue)

{

double returnValue = TimeConversion.Convert(time, convertFrom, convertTo);

NUnit.Framework.Assert.AreEqual(returnValue, expectedValue);

}

/// <summary>

/// This method tests not equal case of

/// Convert method With ModifyInputStub And GetMultiplierStub method Test

/// of TimeConversion class

/// </summary>

/// <param name="time">input time value</param>

/// <param name="convertFrom">time unit convert from</param>

/// <param name="convertTo">time unit convert to</param>

/// <param name="expectedValue">expected return time value</param>

///

// input: 1.0, hours, seconds, return: 1.

[TestCase(1.0, "Hours", "Seconds", 3600)]

// input: 36, HHH, DDD, return: 36.

[TestCase(36, "HHH", "DDD", 3)]

public void ConvertAreNotEqualTest(double time,

string convertFrom,

string convertTo,

double expectedValue)

{

double returnValue = TimeConversion.Convert(time, convertFrom, convertTo);

NUnit.Framework.Assert.AreNotEqual(returnValue, expectedValue);

}

}

}

**ProjectB**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

/// <summary>

/// PROG2070 – Quality Assurance: Winter 2017

/// Assignment 03

/// Yoonsuk Cho Mar 14, 2017

/// </summary>

namespace ProjectB

{

/// <summary>

/// main class of assignment 3

/// Project B

/// </summary>

class Program

{

/// <summary>

/// program start main method

/// using TimeConversion class

/// it will convert time unit

/// this program will check only a time unit

/// </summary>

/// <param name="args">parameters of start method</param>

static void Main(string[] args)

{

bool doCont = true;

while (doCont)

{

double inputTime = 0.0d;

Console.WriteLine("=========================================");

Console.WriteLine("====== Time Converter ProjectB ======");

Console.WriteLine("=========================================");

Console.WriteLine();

Console.Write("Enter a time: ");

string timeStr = Console.ReadLine();

if (Double.TryParse(timeStr, out inputTime))

{

Console.Write("Enter a time unit (convert from): ");

string convertFrom = Console.ReadLine();

Console.Write("Enter a time unit (convert to): ");

string convertTo = Console.ReadLine();

Console.WriteLine();

try

{

double convertedValue = TimeConversion.Convert(inputTime,

convertFrom,

convertTo);

Console.WriteLine("{0}({1}) converts to {2}({3}).",

timeStr,

convertFrom,

convertedValue,

convertTo);

}

catch (ArgumentException ae)

{

Console.WriteLine(ae.Message);

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

else

{

Console.WriteLine("Invalid number input.");

}

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Press Enter to continue, or any other key to exit.");

if (Console.ReadKey().Key == ConsoleKey.Enter)

{

doCont = true;

Console.Clear();

}

else

{

doCont = false;

Console.Clear();

Console.WriteLine();

Console.WriteLine(" =================");

Console.WriteLine(" BYE!!!");

Console.WriteLine(" =================");

Console.ReadKey();

}

}

}

}

}

**ProjectB**

**TimeConversion.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProjectB

{

/// <summary>

/// Time Conversion class of assignment 3

/// </summary>

public static class TimeConversion

{

/// <summary>

/// get converted time value

/// from convertFrom time unit to convertTo time unit

/// with ModifyInput and GetMultiplierStub(fixed return)

/// </summary>

/// <param name="value">numeric time value</param>

/// <param name="convertFrom">time unit convert From</param>

/// <param name="convertTo">time unit convert to</param>

/// <returns></returns>

public static double Convert(double value,

string convertFrom,

string convertTo)

{

convertFrom = ModifyInput(convertFrom);

convertTo = ModifyInput(convertTo);

double multiplier = GetMultiplierStub(convertFrom, convertTo);

return value / multiplier;

}

/// <summary>

/// modify time unit

/// from various input time unit

/// to specific time unit sign for multiply

/// </summary>

/// <param name="input">input time unit</param>

/// <returns>specific time unit sign</returns>

private static string ModifyInput(string input)

{

string returnValue = "";

switch (input)

{

case "seconds":

case "Seconds":

case "s":

case "S":

returnValue = "seconds";

break;

case "minutes":

case "Minutes":

case "m":

case "M":

returnValue = "minutes";

break;

case "hours":

case "Hours":

case "h":

case "H":

returnValue = "hours";

break;

case "days":

case "Days":

case "d":

case "D":

returnValue = "days";

break;

default:

throw new ArgumentException("Incorrect time unit");

}

return returnValue;

}

/// <summary>

/// get multiplier factor

/// using convertFrom time unit to convertTo time unit

/// this method will return fixed value

/// </summary>

/// <param name="convertFrom">convert From time unit</param>

/// <param name="convertTo">convert To time unit</param>

/// <returns></returns>

private static double GetMultiplierStub(string convertFrom,

string convertTo)

{

return 1.0d;

}

}

}

**ProjectB.Tests**

**TimeConversionBTest.cs**

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using NUnit.Framework;

/// <summary>

/// PROG2070 – Quality Assurance: Winter 2017

/// Assignment 03

/// Yoonsuk Cho Mar 14, 2017

/// </summary>

namespace ProjectB.Tests

{

/// <summary>

/// main class of assignment 3

/// Project B - Test Program

/// This class tests Program class of Project B

/// </summary>

[TestClass]

public class TimeConversionBTest

{

/// <summary>

/// This method tests equal case of

/// Convert method With ModifyInput And GetMultiplierStub method Test

/// of TimeConversion class

/// </summary>

/// <param name="time">input time value</param>

/// <param name="convertFrom">time unit convert from</param>

/// <param name="convertTo">time unit convert to</param>

/// <param name="expectedValue">expected return time value</param>

// input: 1.0, hours, seconds, return: 1.

[TestCase(1.0, "hours", "Seconds", 1)]

// input: 36, minutes, s, return: 36.

[TestCase(36, "minutes", "s", 36)]

public void ConvertAreEqualTest(double time,

string convertFrom,

string convertTo,

double expectedValue)

{

double returnValue = TimeConversion.Convert(time, convertFrom, convertTo);

NUnit.Framework.Assert.AreEqual(returnValue, expectedValue);

}

/// <summary>

/// This method tests not equal case of

/// Convert method With ModifyInput And GetMultiplierStub method Test

/// of TimeConversion class

/// </summary>

/// <param name="time">input time value</param>

/// <param name="convertFrom">time unit convert from</param>

/// <param name="convertTo">time unit convert to</param>

/// <param name="expectedValue">expected return time value</param>

// input: 1.0, hours, seconds, return: 3600.

[TestCase(1.0, "Hours", "Seconds", 3600)]

public void ConvertAreNotEqualTest(double time,

string convertFrom,

string convertTo,

double expectedValue)

{

double returnValue = TimeConversion.Convert(time, convertFrom, convertTo);

NUnit.Framework.Assert.AreNotEqual(returnValue, expectedValue);

}

/// <summary>

/// This method tests Exception case of

/// Convert method With ModifyInput method Test

/// of TimeConversion class

/// This checks Exception type and message

/// </summary>

/// <param name="time">input time value</param>

/// <param name="convertFrom">time unit convert from</param>

/// <param name="convertTo">time unit convert to</param>

/// <param name="expectedMessage">expected Message of Exception</param>

[TestCase(1.0, "HHH", "Seconds", "Incorrect time unit")]

public void ConvertThrowsExcpetionTest(double time,

string convertFrom,

string convertTo,

string expectedMessage)

{

// the ArgumentException we expect thrown from the TimeConversion.Convert method

var exception = NUnit.Framework.Assert.Throws<ArgumentException>(() =>

TimeConversion.Convert(time, convertFrom, convertTo));

// now we can test the exception itself

NUnit.Framework.Assert.That(exception.Message == expectedMessage);

}

}

}

**Project C**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

/// <summary>

/// PROG2070 – Quality Assurance: Winter 2017

/// Assignment 03

/// Yoonsuk Cho Mar 14, 2017

/// </summary>

namespace ProjectC

{

/// <summary>

/// main class of assignment 3

/// Project C

/// </summary>

class Program

{

/// <summary>

/// program start main method

/// using TimeConversion class

/// it will convert time unit

/// </summary>

/// <param name="args">parameters of start method</param>

static void Main(string[] args)

{

bool doCont = true;

while (doCont)

{

double inputTime = 0.0d;

Console.WriteLine("=========================================");

Console.WriteLine("====== Time Converter ProjectC ======");

Console.WriteLine("=========================================");

Console.WriteLine();

Console.Write("Enter a time: ");

string timeStr = Console.ReadLine();

if (Double.TryParse(timeStr, out inputTime))

{

Console.Write("Enter a time unit (convert from): ");

string convertFrom = Console.ReadLine();

Console.Write("Enter a time unit (convert to): ");

string convertTo = Console.ReadLine();

Console.WriteLine();

try

{

double convertedValue = TimeConversion.Convert(inputTime,

convertFrom,

convertTo);

Console.WriteLine("{0}({1}) converts to {2}({3}).",

timeStr,

convertFrom,

convertedValue,

convertTo);

}

catch (ArgumentException ae)

{

Console.WriteLine(ae.Message);

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

else

{

Console.WriteLine("Invalid number input.");

}

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Press Enter to continue, or any other key to exit.");

if (Console.ReadKey().Key == ConsoleKey.Enter)

{

doCont = true;

Console.Clear();

}

else

{

doCont = false;

Console.Clear();

Console.WriteLine();

Console.WriteLine(" =================");

Console.WriteLine(" BYE!!!");

Console.WriteLine(" =================");

Console.ReadKey();

}

}

}

}

}

**ProjectC**

**TimeConversion.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

/// <summary>

/// PROG2070 – Quality Assurance: Winter 2017

/// Assignment 03

/// Yoonsuk Cho Mar 14, 2017

/// </summary>

namespace ProjectC

{

/// <summary>

/// Time Conversion class of assignment 3

/// </summary>

public static class TimeConversion

{

/// <summary>

/// get converted time value

/// from convertFrom time unit to convertTo time unit

/// with ModifyInput and GetMultiplier

/// </summary>

/// <param name="value">numeric time value</param>

/// <param name="convertFrom">time unit convert From</param>

/// <param name="convertTo">time unit convert to</param>

/// <returns></returns>

public static double Convert(double value,

string convertFrom,

string convertTo)

{

convertFrom = ModifyInput(convertFrom);

convertTo = ModifyInput(convertTo);

double multiplier = GetMultiplier(convertFrom, convertTo);

return value / multiplier;

}

/// <summary>

/// modify time unit

/// from various input time unit

/// to specific time unit sign for multiply

/// </summary>

/// <param name="input">input time unit</param>

/// <returns>specific time unit sign</returns>

private static string ModifyInput(string input)

{

string returnValue = "";

switch (input)

{

case "seconds":

case "Seconds":

case "s":

case "S":

returnValue = "seconds";

break;

case "minutes":

case "Minutes":

case "m":

case "M":

returnValue = "minutes";

break;

case "hours":

case "Hours":

case "h":

case "H":

returnValue = "hours";

break;

case "days":

case "Days":

case "d":

case "D":

returnValue = "days";

break;

default:

throw new ArgumentException("Incorrect time unit");

}

return returnValue;

}

/// <summary>

/// get multiplier factor

/// using convertFrom time unit to convertTo time unit

/// </summary>

/// <param name="convertFrom">convert From time unit</param>

/// <param name="convertTo">convert To time unit</param>

/// <returns></returns>

private static double GetMultiplier(string convertFrom,

string convertTo)

{

double returnValue = 1.0d;

Dictionary<string, int> indexes = new Dictionary<string, int>();

indexes.Add("seconds", 0);

indexes.Add("minutes", 1);

indexes.Add("hours", 2);

indexes.Add("days", 3);

int[] multi = new int[] { 60, 60, 24 };

int startIdx = indexes[convertFrom];

int lastIdx = indexes[convertTo];

if (startIdx < lastIdx)

{

for (int i = startIdx; i < lastIdx; i++)

{

returnValue \*= multi[i];

}

}

else

{

for (int i = lastIdx; i < startIdx; i++)

{

returnValue /= multi[i];

}

}

return returnValue;

}

}

}

**ProjectC.Tests**

**TimeConversionCTest.cs**

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using NUnit.Framework;

namespace ProjectC.Tests

{

[TestClass]

public class TimeConversionCTest

{

/// <summary>

/// This method tests equal case of

/// Convert method With ModifyInput And GetMultiplierb method Test

/// of TimeConversion class

/// </summary>

/// <param name="time">input time value</param>

/// <param name="convertFrom">time unit convert from</param>

/// <param name="convertTo">time unit convert to</param>

/// <param name="expectedValue">expected return time value</param>

///

// input: 1.0, hours, seconds, return: 3600.

[TestCase(1.0, "hours", "Seconds", 3600)]

// input: 36, minutes, s, return: 2160.

[TestCase(36, "minutes", "s", 2160)]

// input: 4320, m, Days, return: 3.

[TestCase(4320, "m", "Days", 3)]

// input: 7200, S, Hours, return: 2.

[TestCase(7200, "S", "Hours", 2)]

public void ConvertAreEqualTest(double time,

string convertFrom,

string convertTo,

double expectedValue)

{

double returnValue = TimeConversion.Convert(time, convertFrom, convertTo);

NUnit.Framework.Assert.AreEqual(returnValue, expectedValue);

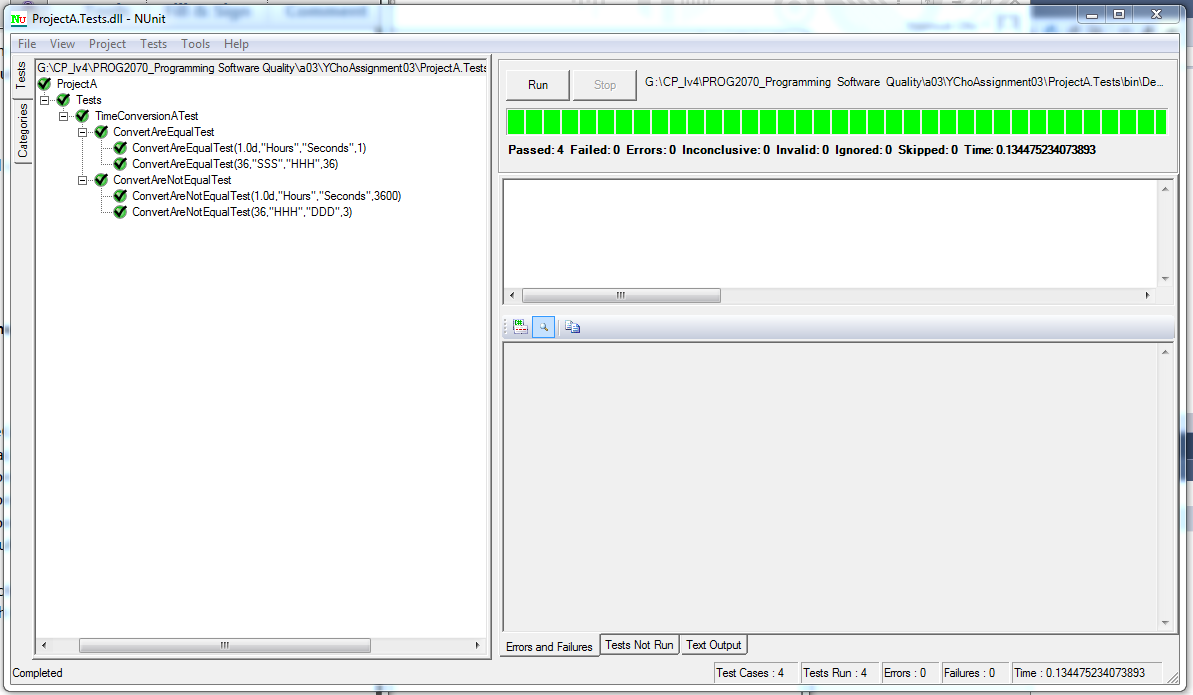
}

}

}

**Project A**

**Test Results**



Case 1. Equal test: Convert gets same time unit and same multiplier (1.0)

Input: 1.0, from “Hours” to “Seconds” then expected return: 1

Case 2. Equal test: Convert gets same time unit and same multiplier (1.0)

Input: 36, from “SSS” to “HHH” then expected return: 36

Case 3. Not Equal test: Convert gets same time unit and same multiplier (1.0)

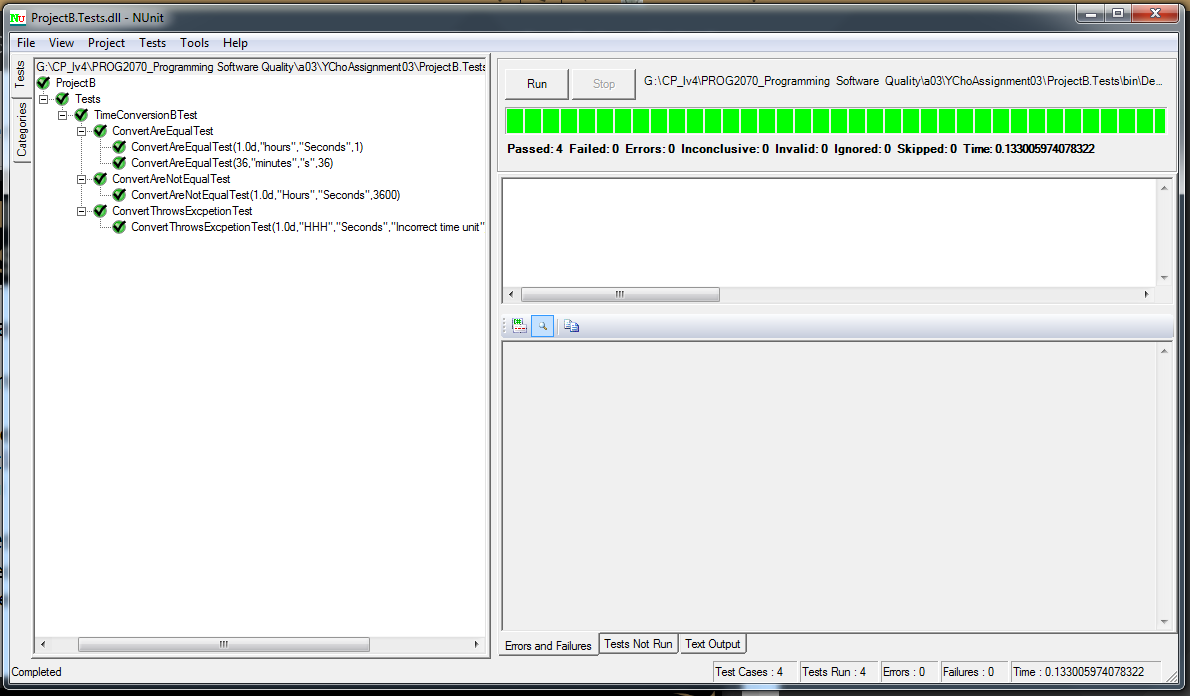
Input: 1.0, from “Hours” to “Seconds” then expected return: 3600

Case 4. Not Equal test: Convert gets same time unit and same multiplier (1.0)

Input: 36, from “HHH” to “DDD” then expected return: 3

**Project B**

**Test Results**



Case 1. Equal test: Convert gets proper time unit but same multiplier (1.0)

Input: 1.0, from “hours” to “Seconds” then expected return: 1

Case 2. Equal test: Convert gets proper time unit but same multiplier (1.0)

Input: 36, from “minutes” to “s” then expected return: 36

Case 3. Not Equal test: Convert gets proper time unit but same multiplier (1.0)

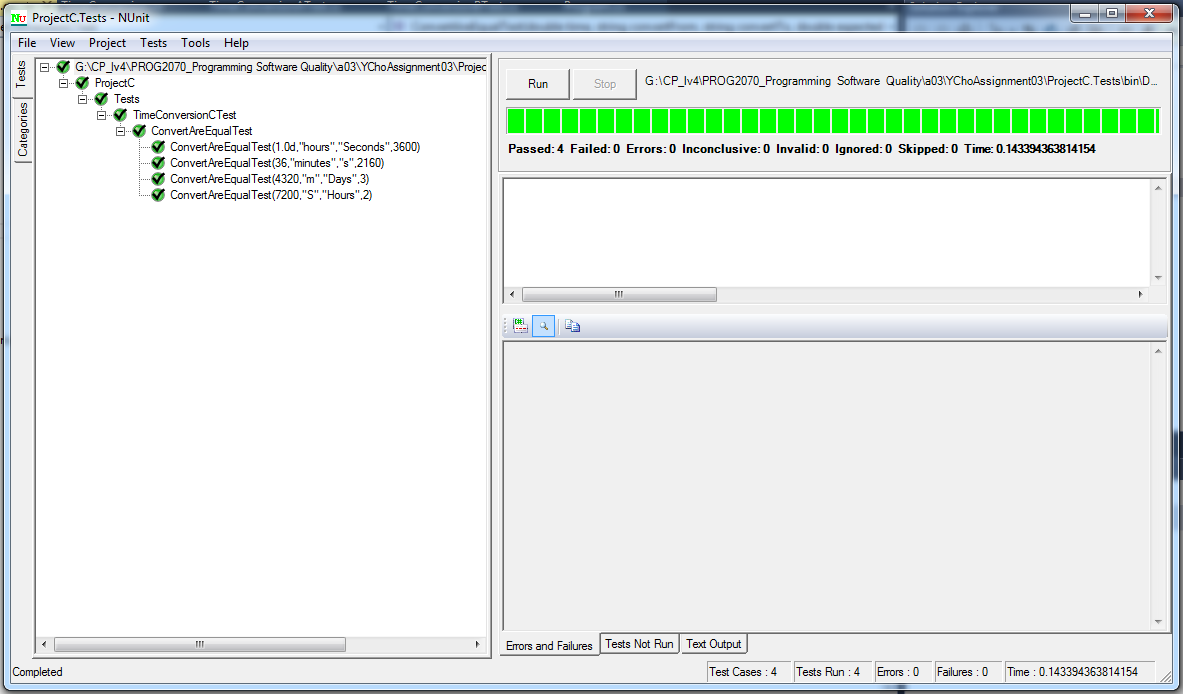
Input: 1.0, from “Hours” to “Seconds” then expected return: 3600

Case 4. Exception test: if put wrong input value in time unit, it will occur Exception

Input: 1, from “HHH” to “Seconds” then expected return Exception message: “Incorrect time unit”

**Project C**

**Test Results**



Case 1. Equal test: Convert gets proper time units and a correct multiplier

Input: 1.0, from “hours” to “Seconds” then expected return: 3600

Case 2. Equal test: Convert gets proper time units and a correct multiplier

Input: 36, from “minutes” to “s” then expected return: 2160

Case 3. Equal test: Convert gets proper time units and a correct multiplier

Input: 4320, from “m” to “Days” then expected return: 3

Case 4. Equal test: Convert gets proper time units and a correct multiplier

Input: 7200, from “S” to “Hours” then expected return: 2