PROG2070 MidTerm

Name: Aryan Gajjar

Unit Testing code and screenshorts

A computer screen shot of a computer screen

AI-generated content may be incorrect.

Code:

using NUnit.Framework;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace StudentRegistration\_Midterm\_AryanGajjar

{

// name: Aryan Gajjar email: agajjar2211@conestogac.on.ca student no.: 8972211

[TestFixture]

public class registerationTests

{

private Registration \_registration;

[SetUp]

public void Setup()

{

\_registration = new Registration();

}

[Test]

public void RegisterName\_ValidName\_ReturnsName()

{

// Arrange

string name = "Aryan Gajjar";

// Act

string result = \_registration.RegisterName(name);

// Assert

Assert.That(result, Is.EqualTo(name));

}

[Test]

public void RegisterName\_NullOrEmpty\_ThrowsArgumentNullException()

{

// Arrange & Act & Assert

Assert.Throws<ArgumentNullException>(() => \_registration.RegisterName(null));

Assert.Throws<ArgumentNullException>(() => \_registration.RegisterName(""));

}

[Test]

public void RegisterName\_TooLong\_ThrowsArgumentException()

{

// Arrange

string longName = new string('G', 36);

// Act & Assert

Assert.Throws<ArgumentException>(() => \_registration.RegisterName(longName));

}

[Test]

public void RegisterAge\_ValidAge\_ReturnsAge()

{

// Arrange

int age = 25;

// Act

int result = \_registration.RegisterAge(age);

// Assert

Assert.That(result, Is.EqualTo(age));

}

[Test]

public void RegisterAge\_Negative\_ThrowsArgumentOutOfRangeException()

{

// Arrange & Act & Assert

Assert.Throws<ArgumentOutOfRangeException>(() => \_registration.RegisterAge(-1));

}

[Test]

public void RegisterAge\_TooHigh\_ThrowsArgumentOutOfRangeException()

{

// Arrange & Act & Assert

Assert.Throws<ArgumentOutOfRangeException>(() => \_registration.RegisterAge(100));

}

[Test]

public void RegisterGroupSize\_ValidSize\_ReturnsSize()

{

// Arrange

int groupSize = 10;

// Act

int result = \_registration.RegisterGroupSize(groupSize);

// Assert

Assert.That(result, Is.EqualTo(groupSize));

}

[Test]

public void RegisterGroupSize\_ZeroOrNegative\_ThrowsArgumentOutOfRangeException()

{

// Arrange & Act & Assert

Assert.Throws<ArgumentOutOfRangeException>(() => \_registration.RegisterGroupSize(0));

Assert.Throws<ArgumentOutOfRangeException>(() => \_registration.RegisterGroupSize(-1));

}

[Test]

public void RegisterGroupSize\_TooHigh\_ThrowsArgumentOutOfRangeException()

{

// Arrange & Act & Assert

Assert.Throws<ArgumentOutOfRangeException>(() => \_registration.RegisterGroupSize(18));

}

}

}

Git log screenshot and link