Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №5

По дисциплине: «Современные платформы программирования»

Выполнил:

Студент 3 курса

Группы ПО-4

Иваненко И. Л.

Проверил:

Монтик Н. С.

Брест 2022

Цель работы: освоить приемы тестирования кода на примере использования библиотеки NUnit и Moq

# Общее задание

На основе многооконного приложения, реализованного в предыдущем проекте, реализовать тесты для нескольких классов

# Тестируемый класс:

using BL.Interfaces;  
using DAL.Interfaces;  
using DAL.Models;  
  
namespace BL.Implementations;  
  
public class StudentManager : IStudentManager  
{  
 private readonly IStudentRepository \_studentRepository;  
 private readonly IGroupRepository \_groupRepository;  
 private readonly ISubjectRepository \_subjectRepository;  
  
 public StudentManager(  
 IStudentRepository studentRepository,   
 IGroupRepository groupRepository,   
 ISubjectRepository subjectRepository)  
 {  
 \_studentRepository = studentRepository;  
 \_groupRepository = groupRepository;  
 \_subjectRepository = subjectRepository;  
 }  
  
 public string CreateStudent(string name, string groupId)  
 {  
 string id = Guid.NewGuid().ToString();  
 Group group = \_groupRepository.FindById(groupId);  
  
 var student = new Student  
 {  
 Id = id,  
 Name = name,  
 GroupId = groupId,  
 Group = group  
 };  
   
 \_studentRepository.AddStudent(student);  
 return id;  
 }  
  
 public IEnumerable<Student> GetAllStudents()  
 {  
 IEnumerable<Student> students = \_studentRepository.GetAllStudents().ToList();  
 return students;  
 }  
  
 public Student FindById(string id)  
 {  
 Student student = \_studentRepository.FindById(id);  
 return student;  
 }  
  
 public void AddSubject(string studentId, string subjectId)  
 {  
 Subject subject = \_subjectRepository.FindById(subjectId);  
 \_studentRepository.AddSubject(studentId, subject);  
 }  
  
 public void Delete(string id)  
 {  
 \_studentRepository.Delete(id);  
 }  
}

# Тесты

using System;  
using System.Collections.Generic;  
using System.Linq;  
using Autofac.Extras.Moq;  
using BL.Implementations;  
using DAL.Interfaces;  
using DAL.Models;  
using Moq;  
using NUnit.Framework;  
  
namespace Tests;  
  
[TestFixture]  
public class Tests  
{  
 [Test]  
 public void CreateStudent\_Success\_ReturnsValidGuid()  
 {  
 using AutoMock mock = AutoMock.GetLoose();  
 mock.Mock<IGroupRepository>()  
 .Setup(x => x.FindById(It.IsAny<string>()))  
 .Returns(new Group  
 {  
 Id = Guid.NewGuid().ToString()  
 });  
 var studentManager = mock.Create<StudentManager>();  
 string studentId = studentManager.CreateStudent("student", "group");  
 Assert.True(Guid.TryParse(studentId, out \_));  
 }  
  
 [Test]  
 public void GetAllStudents\_Success\_ReturnsAllStudentsFromRepository()  
 {  
 List<Student> sampleStudents = GetSampleStudents().ToList();  
 AutoMock mock = AutoMock.GetLoose();  
 mock.Mock<IStudentRepository>()  
 .Setup(x => x.GetAllStudents())  
 .Returns(sampleStudents);  
 var studentManager = mock.Create<StudentManager>();  
 Assert.AreEqual(sampleStudents.Count, studentManager.GetAllStudents().Count());  
 }  
  
 private IEnumerable<Student> GetSampleStudents()  
 {  
 return new List<Student>  
 {  
 new() {Id = Guid.NewGuid().ToString()},  
 new() {Id = Guid.NewGuid().ToString()},  
 new() {Id = Guid.NewGuid().ToString()},  
 new() {Id = Guid.NewGuid().ToString()}  
 };  
 }  
  
 [Test]  
 public void FindById\_Success\_ReturnsValidStudentFromRepository()  
 {  
 var sampleStudent = new Student {Id = Guid.NewGuid().ToString()};  
 using AutoMock mock = AutoMock.GetLoose();  
 mock.Mock<IStudentRepository>()  
 .Setup(x => x.FindById(It.IsAny<string>()))  
 .Returns(sampleStudent);  
 var studentManager = mock.Create<StudentManager>();  
 Student actualStudent = studentManager.FindById(sampleStudent.Id);  
 Assert.AreEqual(actualStudent.Id, sampleStudent.Id);  
 }  
  
 [Test]  
 public void AddSubject\_Success\_DoesNotThrow()  
 {  
 using AutoMock mock = AutoMock.GetLoose();  
 mock.Mock<ISubjectRepository>()  
 .Setup(x => x.FindById(It.IsAny<string>()))  
 .Returns(new Subject {Id = Guid.NewGuid().ToString()});  
 var studentManager = mock.Create<StudentManager>();  
 Assert.DoesNotThrow(() =>  
 studentManager.AddSubject(  
 Guid.NewGuid().ToString(), Guid.NewGuid().ToString()));  
 }  
  
 [Test]  
 public void Delete\_Success\_DoesNotThrow()  
 {  
 using AutoMock mock = AutoMock.GetLoose();  
 var studentManager = mock.Create<StudentManager>();  
 Assert.DoesNotThrow(() => studentManager.Delete(Guid.NewGuid().ToString()));  
 }  
}

# Результат выполнения тестов

Text

Description automatically generated

# Вывод

В ходе выполнения данной лабораторной работы были приобретены практические навыки создания тестов для классов с использованием NUnit, Moq и Autofac.Extensions.Moq.