МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет

им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации

Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине

Поддержка и тестирование программных модулей

Тема: «Тестирование методом белого ящика»

Работу выполнил

Студент гр.4332

Салахов К. И.

Принял

Преподаватель

Шумилкин А. О.

Казань 2024

**Цель работы**

Изучить возможность создания автоматических тестов, для модульного тестирования.

**Задание на лабораторную работу**

1. Взять заранее разработанную программу с реализованными методами авторизации и регистрации.

2. Запустить и проверить работоспособность программы.

3. Протестировать метод регистрации и авторизации.

**Результат выполнения работы**

Добавляем проект модульного теста (Рисунок 1).

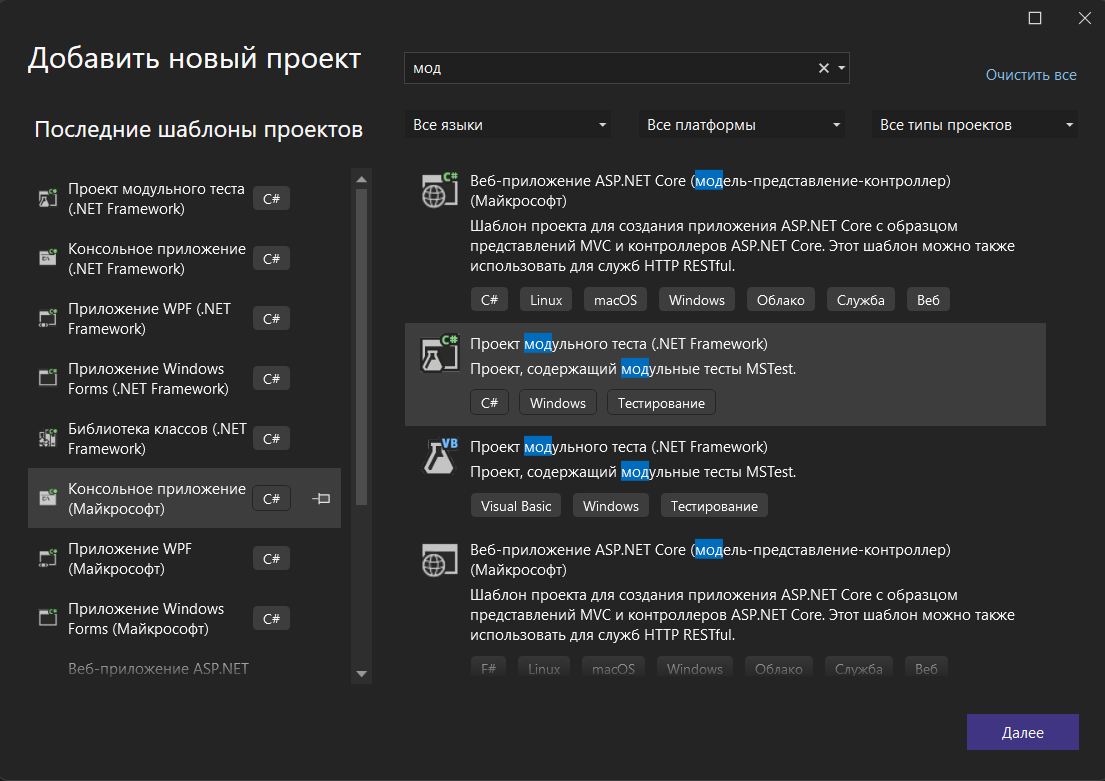


Рисунок 1 – Проект модульного текста

Добавляем ссылку на наш проект (Рисунок 2).

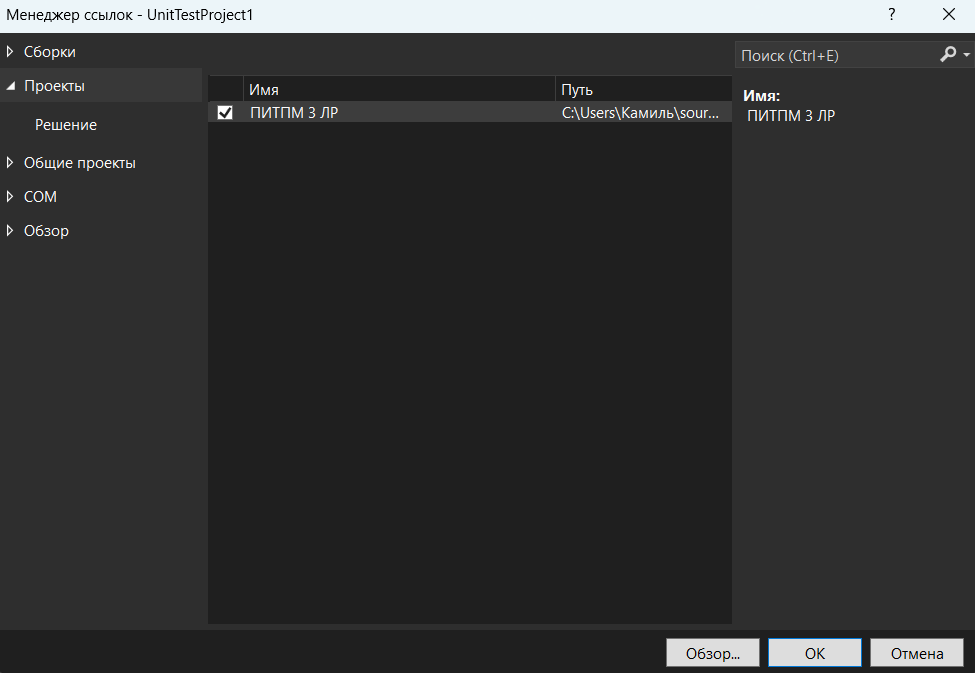


Рисунок 2 – Ссылка на проект

Создаем тесты для проверки методов авторизации и регистрации (Рисунок 2).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2 – Результаты тестов

Изменяем методы для теста и специально имитируем провал теста (Рисунок 3).

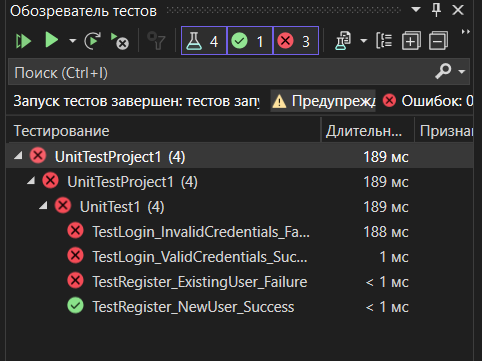


Рисунок 3 – Имитируем провал теста

**Листинг**

Разработанная программа с авторизацией и регистрацией:

Program.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ПИТПМ3

{

class Program

{

static void Main(string[] args)

{

RegAndLog regAndLog = new RegAndLog();

while (true)

{

Console.WriteLine("1. Регистрация");

Console.WriteLine("2. Авторизация");

Console.WriteLine("3. Выход");

Console.Write("Выберите действие: ");

string choice = Console.ReadLine();

switch (choice)

{

case "1":

Console.Write("Введите имя пользователя: ");

string regUsername = Console.ReadLine();

Console.Write("Введите пароль: ");

string regPassword = Console.ReadLine();

regAndLog.Register(regUsername, regPassword);

break;

case "2":

Console.Write("Введите имя пользователя: ");

string loginUsername = Console.ReadLine();

Console.Write("Введите пароль: ");

string loginPassword = Console.ReadLine();

regAndLog.Login(loginUsername, loginPassword);

break;

case "3":

return;

default:

Console.WriteLine("Неверный выбор. Попробуйте снова.");

break;

}

}

}

}

}

User.cs:  
 using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ПИТПМ3

{

public class User

{

public string Username { get; set; }

public string Password { get; set; }

public User(string username, string password)

{

Username = username;

Password = password;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

RegAndLog:

namespace ПИТПМ3

{

public class RegAndLog

{

private List<User> users = new List<User>();

public bool Register(string username, string password)

{

if (users.Exists(u => u.Username == username))

{

Console.WriteLine("Пользователь с таким именем уже существует.");

return false;

}

users.Add(new User(username, password));

Console.WriteLine("Регистрация прошла успешно.");

return true;

}

public bool Login(string username, string password)

{

User user = users.Find(u => u.Username == username);

if (user != null && user.Password == password)

{

Console.WriteLine("Авторизация прошла успешно.");

return true;

}

Console.WriteLine("Неверное имя пользователя или пароль.");

return false;

}

}

}

UnitTest1:

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using ПИТПМ3;

namespace UnitTestProject1

{

[TestClass]

public class UnitTest1

{

private RegAndLog regAndLog;

[TestInitialize]

public void Setup()

{

regAndLog = new RegAndLog();

}

[TestMethod]

public void TestRegister\_NewUser\_Success()

{

bool result = regAndLog.Register("newUser", "password");

Assert.IsTrue(result);

}

[TestMethod]

public void TestRegister\_ExistingUser\_Failure()

{

regAndLog.Register("existingUser", "password");

bool result = regAndLog.Register("existingUser", "password");

Assert.IsFalse(result);

}

[TestMethod]

public void TestLogin\_ValidCredentials\_Success()

{

regAndLog.Register("validUser", "validPassword");

bool result = regAndLog.Login("validUser", "validPassword");

Assert.IsTrue(result);

}

[TestMethod]

public void TestLogin\_InvalidCredentials\_Failure()

{

regAndLog.Register("invalidUser", "password");

bool result = regAndLog.Login("invalidUser", "wrongPassword");

Assert.IsFalse(result);

}

}

}

UnitTest1 (с неудачными тестами):

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using ПИТПМ3;

namespace UnitTestProject1

{

[TestClass]

public class UnitTest1

{

private RegAndLog regAndLog;

[TestInitialize]

public void Setup()

{

regAndLog = new RegAndLog();

}

[TestMethod]

public void TestRegister\_NewUser\_Success()

{

bool result = regAndLog.Register("newUser", "password");

Assert.IsTrue(result); // Правильно: ожидаем, что регистрация пройдет успешно

}

[TestMethod]

public void TestRegister\_ExistingUser\_Failure()

{

regAndLog.Register("existingUser", "password");

bool result = regAndLog.Register("existingUser", "password");

Assert.IsTrue(result); // Ошибка: ожидаем, что регистрация пройдет успешно

}

[TestMethod]

public void TestLogin\_ValidCredentials\_Success()

{

regAndLog.Register("validUser", "validPassword");

bool result = regAndLog.Login("validUser", "wrongPassword"); // Ошибка: используем неправильный пароль

Assert.IsTrue(result); // Ошибка: ожидаем, что авторизация пройдет успешно

}

[TestMethod]

public void TestLogin\_InvalidCredentials\_Failure()

{

regAndLog.Register("invalidUser", "password");

bool result = regAndLog.Login("invalidUser", "password"); // Ошибка: используем правильный пароль

Assert.IsFalse(result); // Ошибка: ожидаем, что авторизация провалится

}

}

}