**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | колледж |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Специалист по ИС и программированию |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по лабораторной работе №9**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Конфигурационное тестирование | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Тестирование информационных систем |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Масевич Екатерина Александровна |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-311 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев Иван Валерьевич |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2025 г.**

**Лабораторная работа №9. «Конфигурационное тестирование»**

**Цель работы:**

Цель работы: изучение конфигурационного тестирования ПО. Результатом работы является отчет, в котором должны быть приведены исходные коды программы, конфигурационноготестирования ПО.

Язык С#

**Пример 1: Тестирование конфигурации с использованием xUnit**

В этом примере мы создаём тест, который проверяет, правильно ли загружается конфигурация из файла appsettings.json.

using Microsoft.Extensions.Configuration;

using Xunit;

public class ConfigurationTests

{

private readonly IConfiguration \_configuration;

public ConfigurationTests()

{

var builder = new ConfigurationBuilder()

.AddJsonFile("appsettings.json");

\_configuration = builder.Build();

}

[Fact]

public void TestDatabaseConnectionString()

{

var connectionString = \_configuration.GetConnectionString("DefaultConnection");

Assert.Equal("YourExpectedConnectionString", connectionString);

}

}

##### Пример 2: Мокирование конфигурации с использованием Moq

В этом примере мы используем библиотеку Moq для создания макета интерфейса IConfiguration для тестирования класса, который зависит от конфигурации.

using Moq;

using Xunit;

public class MyServiceTests

{

private readonly Mock<IConfiguration> \_mockConfiguration;

private readonly MyService \_service;

public MyServiceTests()

{

\_mockConfiguration = new Mock<IConfiguration>();

\_mockConfiguration.Setup(m => m["MySetting"]).Returns("TestValue");

\_service = new MyService(\_mockConfiguration.Object);

}

[Fact]

public void TestMySetting()

{

var result = \_service.GetSetting();

Assert.Equal("TestValue", result);

}

}

public class MyService

{

private readonly IConfiguration \_configuration;

public MyService(IConfiguration configuration)

{

\_configuration = configuration;

}

public string GetSetting()

{

return \_configuration["MySetting"];

}

}

##### Пример 3: Тестирование с использованием конфигурационного файла

Этот пример показывает, как можно использовать конфигурационный файл для определения источника данных в тестах.

using System.Configuration;

using Xunit;

public class DataSourceTests

{

[Fact]

public void TestDataSource()

{

var dataSource = ConfigurationManager.AppSettings["DataSource"];

Assert.NotNull(dataSource);

Assert.Equal("ExpectedDataSource", dataSource);

}

}

### Заключение

Эти примеры демонстрируют, как можно использовать различные подходы к конфигурационному тестированию в C#.