**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ВТ**

**отчет**

**по лабораторной работе №2**

**по дисциплине «Программирование в среде .Net»**

**Тема: РАЗРАБОТКА СЛОЯ БИЗНЕС-ЛОГИКИ ПРИЛОЖЕНИЯ**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6306 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Бохан Ю.А. |
|  |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Пешехонов К. А. |

Санкт-Петербург

2020

**Цель работы:**

Реализовать слой бизнес-логики приложения.

**Задание:**

1) Написать слой бизнес-логики приложения.

2) Покрыть слой бизнес-логики приложения тестами.

Для разработки было выбрано приложение по ведению базы команд Формулы 1.

**Код программы:**

Примеры кода программы даны для сущности «гонщик», другая сущность («команда») реализованы похожим образом.

1) Пример сервиса (RiderCreateService)

- интерфейс IRiderCreateService

using System.Threading.Tasks;  
using Formula1.Domain;  
using Formula1.Domain.Models;  
  
namespace Formula1.BLL.Contracts  
{  
 public interface IRiderCreateService  
 {  
 Task<Rider> CreateAsync(RiderUpdateModel rider);  
 }  
}

- класс RiderCreateService

using System.Threading.Tasks;  
using Formula1.BLL.Contracts;  
using Formula1.DataAccess.Contracts;  
using Formula1.Domain;  
using Formula1.Domain.Models;  
  
namespace Formula1.BLL.Implementation  
{  
 public class RiderCreateService : IRiderCreateService  
 {  
 private IRiderDataAccess RiderDataAccess { get; }  
 private IF1TeamGetService F1TeamGetService { get; }  
  
 public RiderCreateService(IRiderDataAccess riderDataAccess, IF1TeamGetService f1TeamGetService)  
 {  
 RiderDataAccess = riderDataAccess;  
 F1TeamGetService = f1TeamGetService;  
 }  
  
 public async Task<Rider> CreateAsync(RiderUpdateModel rider)  
 {  
 await F1TeamGetService.ValidateAsync(rider);  
 return await RiderDataAccess.InsertAsync(rider);  
  
 }  
 }  
}

2) Пример теста (RiderCreateServiceTests)

using System;  
using System.Threading.Tasks;  
using Formula1.BLL.Contracts;  
using Formula1.BLL.Implementation;  
using Formula1.DataAccess.Contracts;  
using Formula1.Domain;  
using Formula1.Domain.Models;  
using FluentAssertions;  
using Moq;  
using NUnit.Framework;  
using Ploeh.AutoFixture;  
  
namespace Formula1.BLL.Tests  
{  
 public class RiderCreateServiceTests  
 {  
 [Test]  
 public async Task CreateAsync\_DepartmentValidationSucceed\_CreatesRider()  
 {  
 // Arrange  
 var rider = new RiderUpdateModel();  
 var expected = new Rider();  
   
 var f1TeamGetService = new Mock<IF1TeamGetService>();  
 f1TeamGetService.Setup(x => x.ValidateAsync(rider));  
  
 var riderDataAccess = new Mock<IRiderDataAccess>();  
 riderDataAccess.Setup(x => x.InsertAsync(rider)).ReturnsAsync(expected);  
   
 var riderGetService = new RiderCreateService(riderDataAccess.Object, f1TeamGetService.Object);  
   
 // Act  
 var result = await riderGetService.CreateAsync(rider);  
   
 // Assert  
 result.Should().Be(expected);  
 }  
   
 [Test]  
 public async Task CreateAsync\_F1TeamValidationFailed\_ThrowsError()  
 {  
 // Arrange  
 var fixture = new Fixture();  
 var rider = new RiderUpdateModel();  
 var expected = fixture.Create<string>();  
   
 var f1TeamGetService = new Mock<IF1TeamGetService>();  
 f1TeamGetService  
 .Setup(x => x.ValidateAsync(rider))  
 .Throws(new InvalidOperationException(expected));  
   
 var riderDataAccess = new Mock<IRiderDataAccess>();  
  
 var riderGetService = new RiderCreateService(riderDataAccess.Object, f1TeamGetService.Object);  
   
 // Act  
 var action = new Func<Task>(() => riderGetService.CreateAsync(rider));  
   
 // Assert  
 await action.Should().ThrowAsync<InvalidOperationException>().WithMessage(expected);  
 riderDataAccess.Verify(x => x.InsertAsync(rider), Times.Never);  
 }  
 }  
}

**Выводы:**

В процессе выполнения лабораторной работы был реализован слой бизнес-логики приложения в среде .NET, а также были реализованы тесты для этого слоя. Были получены навыки по написанию бизнес-логики приложения и его тестированию.