

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

Студент гр. 6306

Жукаускайте А.В.

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 кинотеатров города.

Код программы:

Примеры кода программы даны для сущности «сеанс», другие сущности («фильм» и «кинотеатр») реализованы похожим образом.

1) Пример сервиса (ScreeningCreateService)

- интерфейс IScreeningCreateService

```
public interface IScreeningCreateService
{
    Task<Screening> CreateAsync(ScreeningUpdateModel screening);
}
```

- класс ScreeningCreateService

```
public class ScreeningCreateService : IScreeningCreateService
{
    private IScreeningDataAccess ScreeningDataAccess { get; }
    private IMovieGetService MovieGetService { get; }
    private ICinemaGetService CinemaGetService { get; }

    public ScreeningCreateService(IScreeningDataAccess
        screeningDataAccess, ICinemaGetService cinemaGetService,
        IMovieGetService movieGetService)
    {
        ScreeningDataAccess = screeningDataAccess;
        MovieGetService = movieGetService;
        CinemaGetService = cinemaGetService;
    }

    public async Task<Screening> CreateAsync(ScreeningUpdateModel screening)
    {
        await MovieGetService.ValidateAsync(screening);
        await CinemaGetService.ValidateAsync(screening);

        return await ScreeningDataAccess.InsertAsync(screening);
    }
}
```

2) Пример теста (ScreeningCreateServiceTests)

```

public class ScreeningCreateServiceTests
{
    [Test]
    public async Task CreateAsync_DepartmentValidationSucceed_CreatesScreening()
    {
        // Arrange
        var screening = new ScreeningUpdateModel();
        var expected = new Screening();

        var cinemaGetService = new Mock<ICinemaGetService>();
        cinemaGetService.Setup(x => x.ValidateAsync(screening));

        var movieGetService = new Mock<IMovieGetService>();
        movieGetService.Setup(x => x.ValidateAsync(screening));

        var screeningDataAccess = new Mock<IScreeningDataAccess>();
        screeningDataAccess.Setup(x => x.InsertAsync(screening)).ReturnsAsync(expected);

        var screeningGetService = new ScreeningCreateService(screeningDataAccess.Object,
            cinemaGetService.Object, movieGetService.Object);

        // Act
        var result = await screeningGetService.CreateAsync(screening);

        // Assert
        result.Should().Be(expected);
    }

    [Test]
    public async Task CreateAsync_CinemaValidationFailed_ThrowsError()
    {
        // Arrange
        var fixture = new Fixture();
        var screening = new ScreeningUpdateModel();
        var expected = fixture.Create<string>();

        var cinemaGetService = new Mock<ICinemaGetService>();
        cinemaGetService
            .Setup(x => x.ValidateAsync(screening))
            .Throws(new InvalidOperationException(expected));

        var movieGetService = new Mock<IMovieGetService>();
        movieGetService.Setup(x => x.ValidateAsync(screening)).Throws(
            new InvalidOperationException(expected));

        var screeningDataAccess = new Mock<IScreeningDataAccess>();

        var screeningGetService = new ScreeningCreateService(screeningDataAccess.Object,
            cinemaGetService.Object, movieGetService.Object);

        // Act
        var action = new Func<Task>(() => screeningGetService.CreateAsync(screening));

        // Assert
        await action.Should().ThrowAsync<InvalidOperationException>().WithMessage(expected);
        screeningDataAccess.Verify(x => x.InsertAsync(screening), 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.