**Лабораторная работа №2**

**Разработка модели предметной области с использованием ENTITY FRAMEWORK.**

**Цель работы.** Ознакомиться с возможностями ENTITY FRAMEWORK для моделирования предметной области, выборки, вставки, удаления и обновления данных объектов, связанных с таблицами базы данных заданной предметной области. Ознакомиться с особенностями трехслойной архитектуры приложений.

**Задание.**

* Разработать слой доступа к данным (Data Access Layer) в виде библиотеки классов для создаваемой системы в рамках выбранной предметной области.

Библиотека должна содержать:

* классы сущностей (Entities), моделирующие не менее чем три таблицы базы данных;
* интерфейсы и классы, реализующие паттерн **Репозиторий** для доступа к БД с использованием Entity Framework Core, интерфейсы должны быть **public**, а реализации **internal**;
* Разработать слой бизнес-логики (Buisiness Logic Layer) в виде библиотеки классов, со ссылкой на DAL, содержащей:
* объекты DTO для сущностей слоя DAL;
* классы, реализующие функцию трансляции данных из DTO в сущности и обратно
* классы (сервисы), реализующие операции с данными в рамках выбранной предметной области с использованием классов библиотеки DAL; обязательные операции: получение всех данных, добавление, удаление, поиск данных, удовлетворяющих заданному критерию, **валидация данных**; для выполнения запросов использовать LINQ;
* классы для реализации внедрения зависимостей.

# Разработать приложение WPF (слой UI), предоставляющее пользователю удобный интерфейс для работы с системой. Это приложение должно содержать ссылку на слой BLL и не должно содержать ссылку на DAL. Обязательна реализация паттерна MVVM. Программа не должна «ломаться».

На рисунке отображена структура приложения.

Приложение

WPF

(cлой UI)

Ядро приложения

(слой

BLL)

Слой доступа к да

ным

(DAL)

База данных

Эта часть будет

использоваться в

следующих

з

а

даниях

Критерии для поиска указаны в таблице 1. Если предметная область отличается от указанной таблицы, критерии для поиска выбрать аналогичные указанным в таблице самостоятельно.

**Порядок выполнения.** Для выполнения поставленной задачи нужно создать решение с тремя проектами: приложением WPF и двумя библиотеками классов (слой DAL и слой BLL).

Добавить в проект DAL пакеты:

* Microsoft.EntityFrameworkCore.SqlServer;
* Microsoft.EntityFrameworkCore.Tools.

Создать в этом слое классы моделей данных, класс контекста данных, интерфейсы репозиториев. Реализовать репозитории для каждой сущности данных с использованием Entity Framework. Например,

internal class IngredientRepository

: IRepository<Ingredient>

{

public ApplicationContext Database { get; set; }

public IngredientRepository(ApplicationContext db)

{

Database = db;

}

public void Create(Ingredient item)

{

Database.Ingredients.Add(item);

Database.SaveChanges();

}

public void Delete(Ingredient item)

{

Database.Ingredients.Remove(item);

Database.SaveChanges();

}

public void Update(Ingredient item)

{

Database.Ingredients.Update(item);

Database.SaveChanges();

}

public IQueryable<Ingredient> GetAll() =>

Database.Ingredients.AsQueryable().AsNoTracking();

}

Создать статический класс для конфигурирования внедрения зависимостей и в нем метод расширения типа IServiceCollection для добавления контекста данных и объектов репозиториев в коллекцию сервисов. Например,

public static class ConfigurationExtensions

{

public static void ConfigureDAL(

this IServiceCollection services,string connection)

{

services.AddDbContext<ApplicationContext>(

options => options.UseSqlServer(connection));

services.AddScoped<IRepository<Ingredient>,

IngredientRepository>();

services.AddScoped<IRepository<IngredientCategory>, IngredientCategoryRepository>();

…

}

}

Далее следует добавить в проект BLL пакеты:

* AutoMapper;
* AutoMapper.Extensions.Microsoft.DependencyInjection.

Добавить ссылку на библиотеку DAL.

Создать в этом слое классы объектов DTO, интерфейсы сервисов.

Например,

public class IngredientCategoryDTO

{

public int Id { get; set; }

public string Name { get; set; }

}

public interface IService<T> where T:IDTO

{

void Add(T item);

void Update(T item);

void DeleteById(int id);

T FindById(int id);

List<T> GetAll();

}

public interface IPizzaService:IService<PizzaDTO>

{

List<(IngredientDTO, int)> GetIngredients(int id);

}

Создать классы профилей для отображения моделей данных с помощью AutoMapper. Например,

class PizzaProfile:Profile

{

public PizzaProfile()

{

CreateMap<Pizza, PizzaDTO>().ReverseMap();

}

}

Реализовать сервисы, исходя из требуемого функционала.

Создать статический класс для конфигурирования внедрения зависимостей и в нем метод расширения типа IServiceCollection для добавления объекта AutoMapper и объектов сервисов в коллекцию. Например,

public static class ConfigurationExtensions

{

public static void ConfigureBLL(

this IServiceCollection services,string connection)

{

services.ConfigureDAL(connection);

services.AddAutoMapper(

typeof(IngredientCategoryProfile),

typeof(IngredientProfile),

typeof(OrderItemProfile),

…

);

services

.AddTransient<IService<IngredientCategoryDTO>,

IngredientCategoryService>();

services

.AddTransient<IService<IngredientDTO>,

IngredientService>();

services.AddTransient<IPizzaService,

PizzaService>();

…

}

}

Добавить в проект WPF ссылку на BLL. Разработать удобный пользовательский интерфейс и реализовать требуемый в задании функционал с использованием сервисов слоя BLL.

Таблица 1

|  |  |  |
| --- | --- | --- |
| Вариант | Система для разработки | Критерии для поиска |
| 1 | Система записи пациентов на прием в поликлинику | - по фамилии врача  - по фамилии пациента |
| 2 | Приложение фитнес-центра | - по виду тренировок  - по фамилии тренера |
| 3 | Автоматизированная система учета заявок станции техобслуживания | - по фамилии владельца авто  - по дате приема заявки |
| 4 | Автоматизированная система заказа такси | - по времени заказа  - по адресу клиента |
| 5 | Система продажи авиабилетов | - по номеру рейса  - по цене |
| 6 | Сайт туристического агентства | - по срокам поездки  - по региону путешествия |
| 7 | Интернет-магазин по продаже компьютерных комплектующих | - по наименованию комплектующих  - по цене |
| 8 | Сайт заведения общественного питания | - по названию блюда  - по категории блюда |
| 9 | Система продажи электронных железнодорожных билетов | - по пункту назначения  - по дате отправления |
| 10 | Система управления производством и реализацией мебели | - по наименованию изделия  - по дате выпуска продукции |
| 11 | Системапродажи театральных билетов | - по дате спектакля  - по цене билета |
| 12 | Приложение «Новостной портал» | - по теме новостей  - по дате публикации |
| 13 | Система записи клиентов салона красоты | - по виду услуги  - по фамилии мастера |
| 14 | Система заказа обедов в офис организации | - по дате заказа  - по названию организации |
| 15 | Система для автоматизации деятельности фирмы организации мероприятий | - по виду мероприятия  - по дате проведения |

**Приветствуется:**

1. Логирование в приложении, которое бы позволило просматривать запросы к БД, а также сведения о времени выполнения операций.
2. Создание unit-тестов

**Отчет должен содержать:**

* Структуру приложения в виде диаграммы классов.
* Исходные коды классов и интерфейсов.
* Результаты выполнения.

## Ссылки по теме

1. <https://docs.microsoft.com/en-us/dotnet/core/> - официальная документация по платформе .NET Core.
2. Official documentation for Entity Framework Core - <https://docs.microsoft.com/en-us/ef/core/>
3. Official repository with EF Core - <https://github.com/dotnet/efcore>
4. Руководство по EntityFrameworkCore - <https://metanit.com/sharp/entityframeworkcore/>
5. Entity Framework Core Tutorial - <https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>
6. Learn Entity Framework Core - <https://www.learnentityframeworkcore.com/>
7. <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15> - SQL Server technical documentation.

Библиотеки и пакеты:

1. Ninject – для внедрения зависимостей.
2. Automapper – для отображения в DTO.

## Литература

1. Entity Framework Core in Action 1st Edition - <https://www.amazon.com/Entity-Framework-Core-Action-Smith/dp/161729456X>
2. Технология LINQ на примерах. Практикум с использованием электронного задачника ProgrammingTaskbookfor LINQ - <https://www.ozon.ru/context/detail/id/32851534/>