Федеральное агентство связи

Сибирский Государственный Университет Телекоммуникаций и Информатики

СибГУТИ

Кафедра прикладной математики и кибернетики

РГР по дисциплине Визуальное программирование и человеко-машинное взаимодействие. Ч. 4

Выполнил: студент 2 курса, группы ИП-014

Альхимович Михаил Валерьевич  
Проверил: ст. преподаватель

Милешко Антон Владимирович

Новосибирск, 2022 г.

**РГР. Часть 4**

# Задание на РГР

Создать ПО для отображения и обработки статистических данных для определённого вида спорта. ПО должно включать 2 основных окна: окно отображающее таблицы БД со статистической информацией и результаты запросов к БД, переключение таблиц и результатов должно быть реализовано через вкладки; и окно для менеджера запросов к БД.

Первое окно должно давать возможность просматривать и изменять все таблицы БД, а также просматривать результаты запросов к БД. Должна иметься возможность удалить вкладки с результатами запросов, но не вкладки с таблицами. Также должна иметься возможность перейти к окну менеджера запросов.

Окно менеджера запросов должно предоставлять интерфейс для создания, сохранения, удаления, редактирования запросов. Созданные запросы должны отображаться в виде списка с названиями запросов, в который можно добавлять новые запросы, удалять, просматривать существующие. Для создания и редактирования запросов должен предоставляться визуальный интерфейс, а не язык запросов. Редактор запросов должен поддерживать операции выборки, соединения, группирования, подзапросы (в качестве подзапроса используются ранее сохранённые запросы).

Ход работы:

1. Исследование предметной области и создание ER диаграммы.
2. Перевод ER диаграммы в реляционную модель, создание и заполнение БД.
3. Проработка визуального интерфейса приложения
4. Создание диаграммы классов приложения
5. Реализация основного окна приложения
6. Реализация менеджера запросов
7. Тестирование и отладка

**Вариант задания - 1**

1. Футбол (клубы) ([https://www.whoscored.com/Statistic](https://www.whoscored.com/Statistics)s <https://www.soccerstats.com/> <https://fbref.com/en/>)

**Создание диаграммы классов приложения**

В первую очередь, стоит отметить, что архитектура приложения будет основана на архитектуре Avalonia MVVM (Model-View-ViewModel), с использованием кроссплатформенного XAML фреймворк для платформы .NET Avalonia.

Бизнес логика приложения, в паттерне первая буква “M” (**Model**). В данном слое описывается логика и основные задачи приложения. Взаимодействие с файловой системой, базой данных, API, описание сущностей системы и т.п. Часто общение с различными источниками данных, выделяют в отдельную под-часть (Services).

Интерфейс – в паттерне буква V (**View**) описывается с помощью языка разметки XAML.

ViewModel — связывающий слой между Model и View с помощью технологии привязки (Binding). Для понятия Binding, введем понятие свойства (Property) – изменяемое поле данных во ViewModel. Простыми словами, с помощью binding, все property, описанные в ViewModel доступны для View. Важным, также является **изменяемость** property – под этим следует, что любые изменения во View или Model о которых "узнает" ViewModel будут автоматически изменены в зависимости от того, откуда пришли изменение (ввод текстового поля, получение ответа от API и т.п)

**Описание слоев:**

**Model:**

Опишем здесь классы, которые являются объектами предметной области, которую мы моделируем. То есть классы, с помощью которых в слое ViewModel мы будем прописывать логику приложения.