

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. 1

Выполнил: студент 2 курса, группы ИП-014

Обухов Артём Игоревич

Проверил: ст. преподаватель

Милешко Антон Владимирович

Новосибирск, 2022 г.

РГР. Часть 1

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 - 17

15. Дартс (<https://dartsdb.com>, <https://americandartsdatabase.com/>)

Часть 1

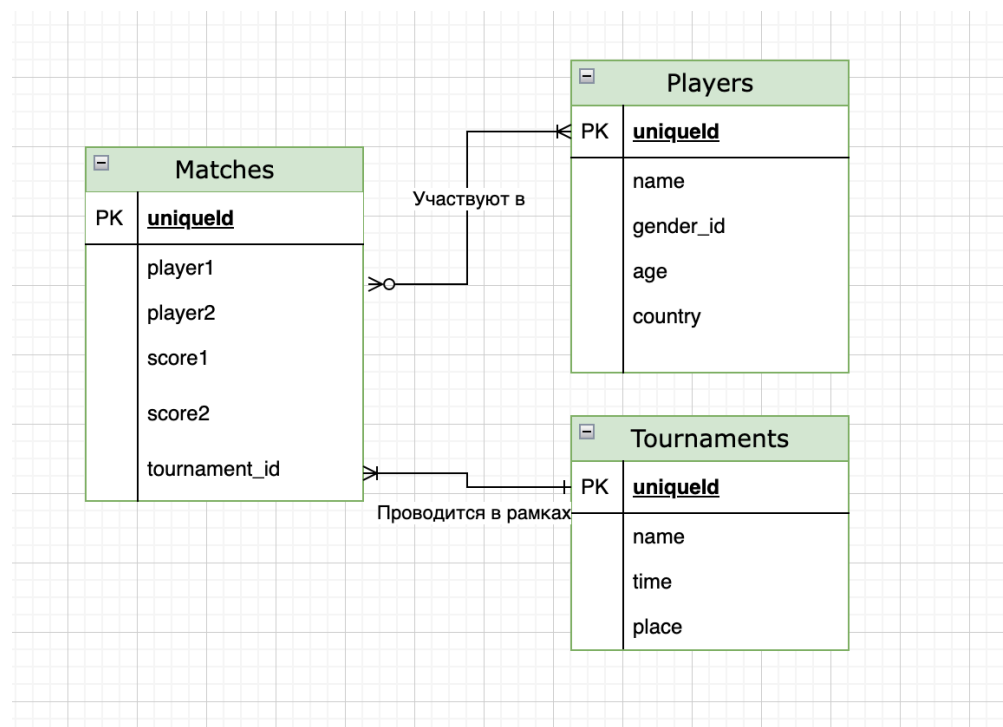
Предметная область

Тема: Дартс

1. Players
 - 1.1. Имя
 - 1.2. Пол
 - 1.3. Возраст
 - 1.4. Страна
2. Matches
 - 2.1. Игрок 1
 - 2.2. Игрок 2
 - 2.3. Счет первого игрока
 - 2.4. Счет второго игрока
 - 2.5. Турнир
3. Tournaments
 - 3.1. Имя
 - 3.2. Время
 - 3.3. Место

ER-диаграмма

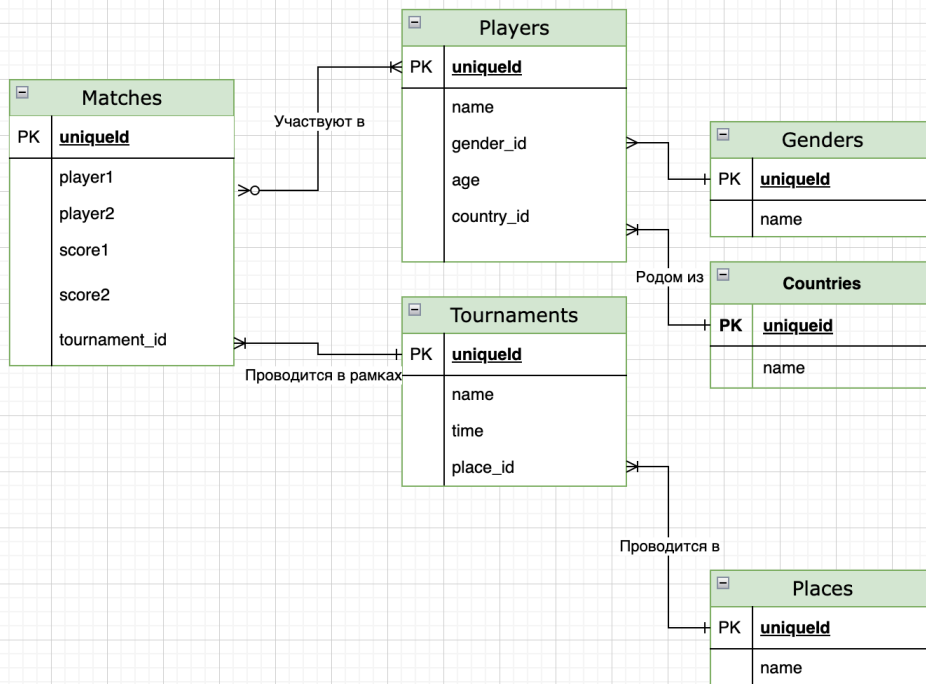
Сущности: Players(Игроки), Matches(Результаты игр), Tournaments(Мероприятия, в пределах которых проводятся раунды),



Часть 2

Реляционная модель

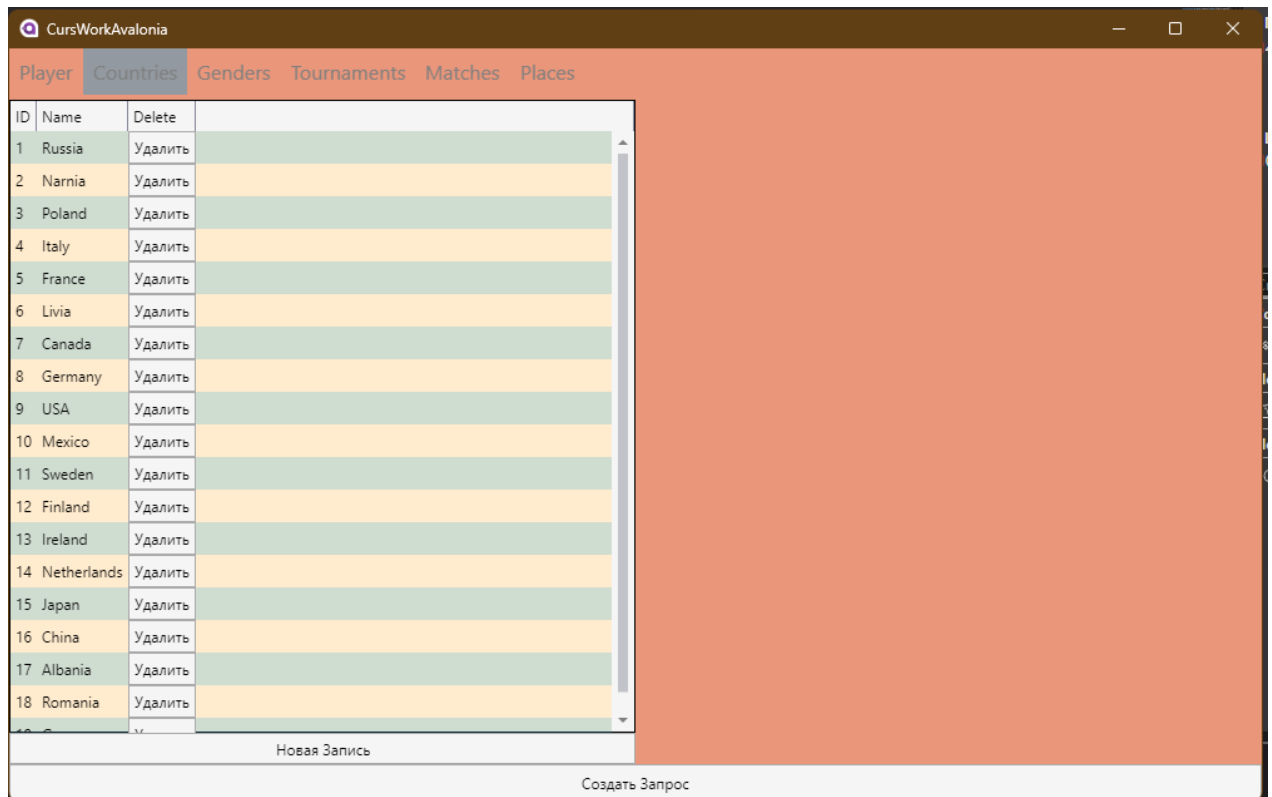
1. players
 - 1.1. id (PK)
 - 1.2. name (STRING)
 - 1.3. gender_id (Внешний ключ)
 - 1.4. country_id (Внешний ключ)
 - 1.5. age (INTEGER)
2. matches
 - 2.1. id (PK)
 - 2.2. player1_id (Внешний ключ)
 - 2.3. player2_id (Внешний ключ)
 - 2.4. score1 (INTEGER)
 - 2.5. score2 (INTEGER)
 - 2.6. tournament_id (Внешний ключ)
3. tournaments
 - 3.1. id (PK)
 - 3.2. name (STRING)
 - 3.3. time (DATETIME)
 - 3.4. place_id (Внешний ключ)
4. genders
 - 4.1. id (PK)
 - 4.2. name (STRING)
5. countries
 - 5.1. id (PK)
 - 5.2. name (STRING)
6. places
 - 6.1. id(PK)
 - 6.2. name (STRING)



Часть 3

Проработка визуального интерфейса

Основное окно:



Для отображения списка таблиц использует **<TabControl>**

Чтобы отобразить содержимое таблиц внутри **<TabControl>** используется элемент **<DataGrid>**

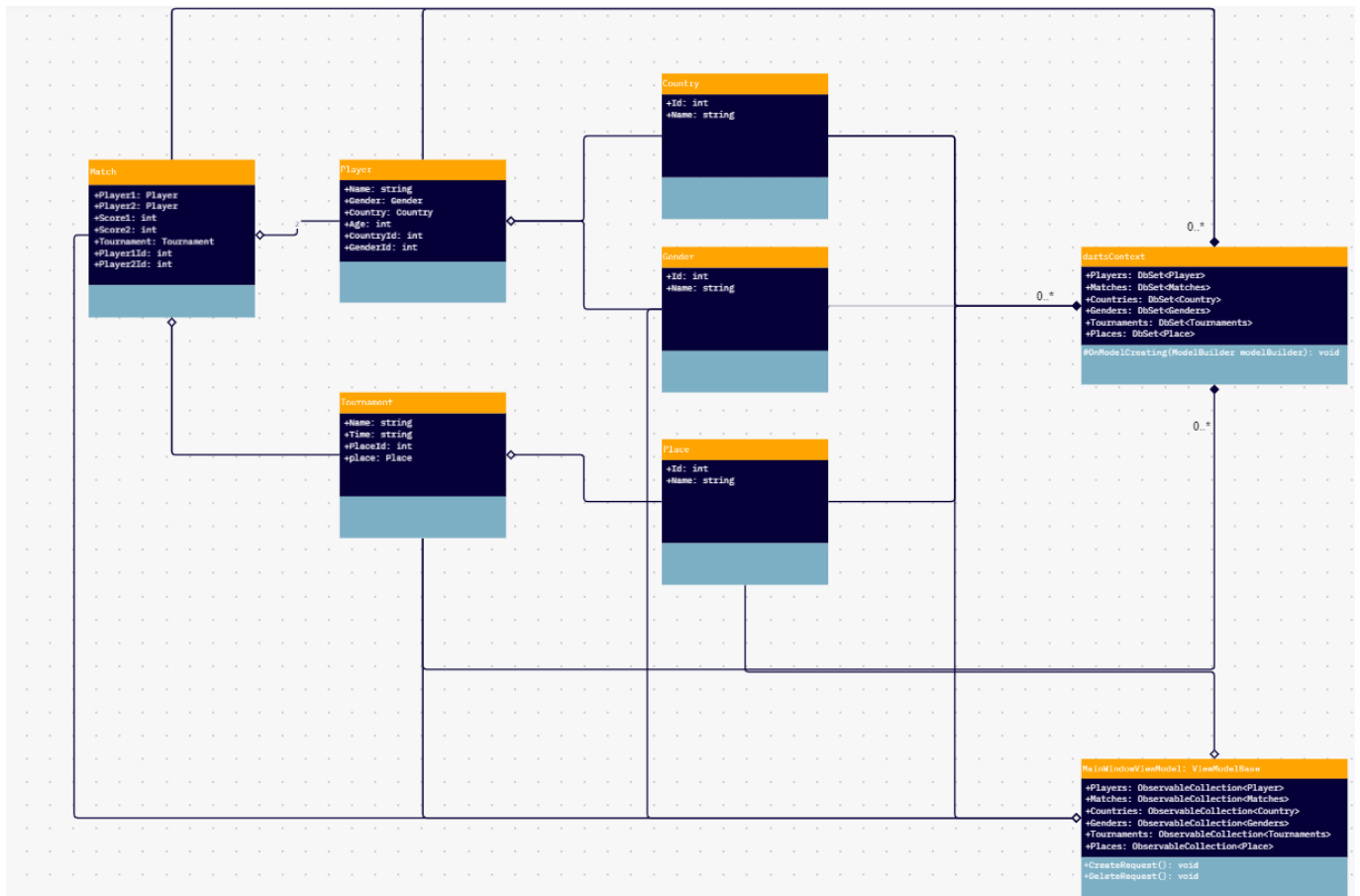
Окно запросов:

The screenshot shows a window titled "запросы" (queries) with a tab labeled "1". The main area is a query editor with a light gray background. It contains a "SELECT" clause followed by a dropdown menu showing "1", and a "FROM" clause followed by a dropdown menu showing "1". Below the query editor, there are three buttons: "Удалить" (Delete), "Создать" (Create), and "Запустить" (Run). The window has a standard Windows-style title bar with minimize, maximize, and close buttons.

Для выбора операторов, полей таблиц и таблиц используется **<ComboBox>** с записанными заранее значениями полей.

Часть 4

Создание диаграммы классов



Классы:

1. Country
 - 1.1. public string name
2. Gender
 - 2.1. public string name
3. Place
 - 3.1. public string name
4. Player
 - 4.1. public string name
 - 4.2. public Gender gender
 - 4.3. public Country country
 - 4.4. public int age
5. Match
 - 5.1. public Player player1
 - 5.2. public Player player2
 - 5.3. public int score1
 - 5.4. public int score2

- 5.5. public Tournament tournament
- 6. Tournament
 - 6.1. public string name
 - 6.2. public DateTime date
 - 6.3. public Place place
- 7. DbContext – класс, который позволяет работать с базой данных через модели. Каждый DbSet<Model> представляет собой сущности в БД
 - 7.1. public DbSet<Player> Players
 - 7.2. public DbSet<Match> Matches
 - 7.3. public DbSet<Country> Countries
 - 7.4. public DbSet<Gender> Genders
 - 7.5. public DbSet<Tournament> Tournaments
 - 7.6. public DbSet<Place> Places
 - 7.7. protected void OnModelCreating(ModelBuilder modelBuilder)
- 8. MainWindowViewModel – здесь представлены все основные данные и методы для работы с View, каждая ObservableCollection – наблюдаемая коллекция, содержимое которых будет отображаться в окне программы, также они непосредственно связаны с моделями из DbContext
 - 8.1. public ObservableCollection<Player> Players
 - 8.2. public ObservableCollection <Match> Matches
 - 8.3. public ObservableCollection <Country> Countries
 - 8.4. public ObservableCollection <Gender> Genders
 - 8.5. public ObservableCollection <Tournament> Tournaments
 - 8.6. public ObservableCollection <Place> Places
 - 8.7. public void SQLRequestOpen()
 - 8.8. public void SQLRequestRun()

Часть 5

Реализация основного окна приложения