

Федеральное агентство связи
Сибирский Государственный Университет Телекоммуникаций и
Информатики
СибГУТИ
Кафедра прикладной математики и кибернетики
РГР по дисциплине Визуальное программирование и человеко-машинное
взаимодействие.

Выполнил: студент 2 курса, группы ИП-014

Зайцев Д.Д.

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

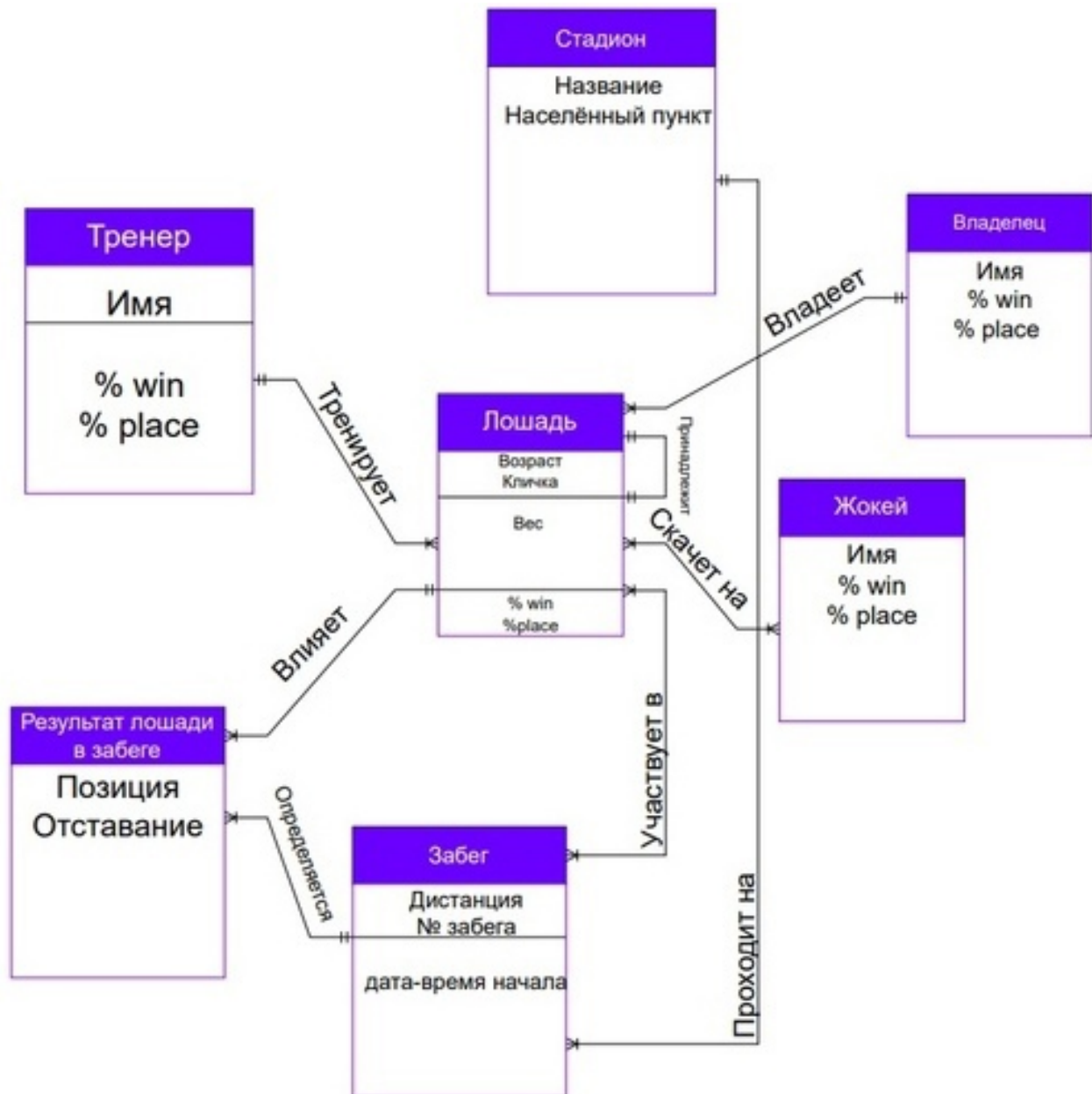
Вариант 9- Лошадиные скачки UK hurdles

Предметная область

1. Стадион
 - 1.1 Название
 - 1.2 Населенный пункт
2. Забег
 - 2.1 Дистанция
 - 2.2 Номер забега
 - 2.3 Дата время начала
3. Результат лошади в забеге
 - 3.1 Позиция
 - 3.2 Отставание
4. Тренер
 - 4.1 Имя
 - 4.2 % win
 - 4.3 % place
5. Лошадь
 - 5.1 Возраст
 - 5.2 Кличка
 - 5.3 Вес
 - 5.4 % win
 - 5.5 % place
6. Владелец
 - 6.1 Имя
 - 6.2 % win
 - 6.3 % place
7. Жокей
 - 7.1 Имя
 - 7.2 % win
 - 7.3 % place

Сущности: Стадион, Владелец, Тренер, Лошадь, Жокей, Забег, Результат лошади в забеге.

ER-диаграмма



Часть 2.

Реляционная модель.

1. Владелец

- 1.1. id(PK)
- 1.2. Имя(String)
- 1.3. % win(DOUBLE)
- 1.4. % place(DOUBLE)

2. Жокей

- 2.1. id(PK)
- 2.2. Имя(String)
- 2.3. % win(DOUBLE)
- 2.4. % place(DOUBLE)

3. Забег

- 3.1. Дистанция(DOUBLE)
- 3.2. № забега(PK)
- 3.3. Дата время начало(DATETIME)
- 3.4. Стадион id(Внешний ключ)

4. Лошадь

- 4.1. Возраст(INTEGER)
- 4.2. Кличка(String)
- 4.3. Вес(DOUBLE)
- 4.4. % win(DOUBLE)
- 4.5. % place(DOUBLE)
- 4.6. id(PK)
- 4.7. Тренер (Внешний ключ)
- 4.8. Жокей (Внешний ключ)
- 4.9. Владелец (Внешний ключ)

5. Результат лошадей в забеге

- 5.1. Позиция(INTEGER)
- 5.2. Отставание(DOUBLE)
- 5.3. Лошадь id(Внешний ключ)

5.4. Забег id(Внешний ключ)

5.5. id(PK)

6. Стадион

6.1. Населённый пункт(INTEGER)

6.2. Название(INTEGER)

6.3. id(PK)

7. Тренер

7.1. Имя(String)

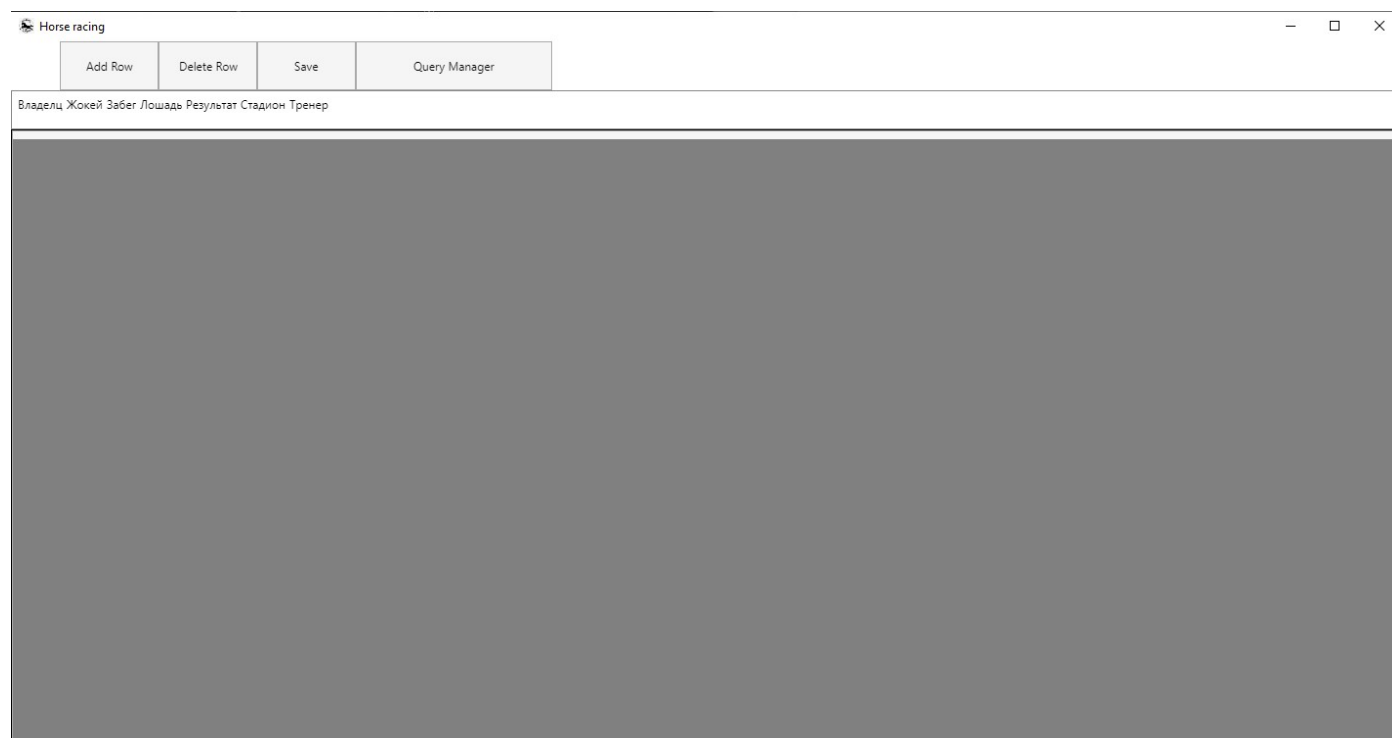
7.2. % win(DOUBLE)

7.3. % place(DOUBLE)

7.4. id(PK)

Часть 3.

Основное окно:

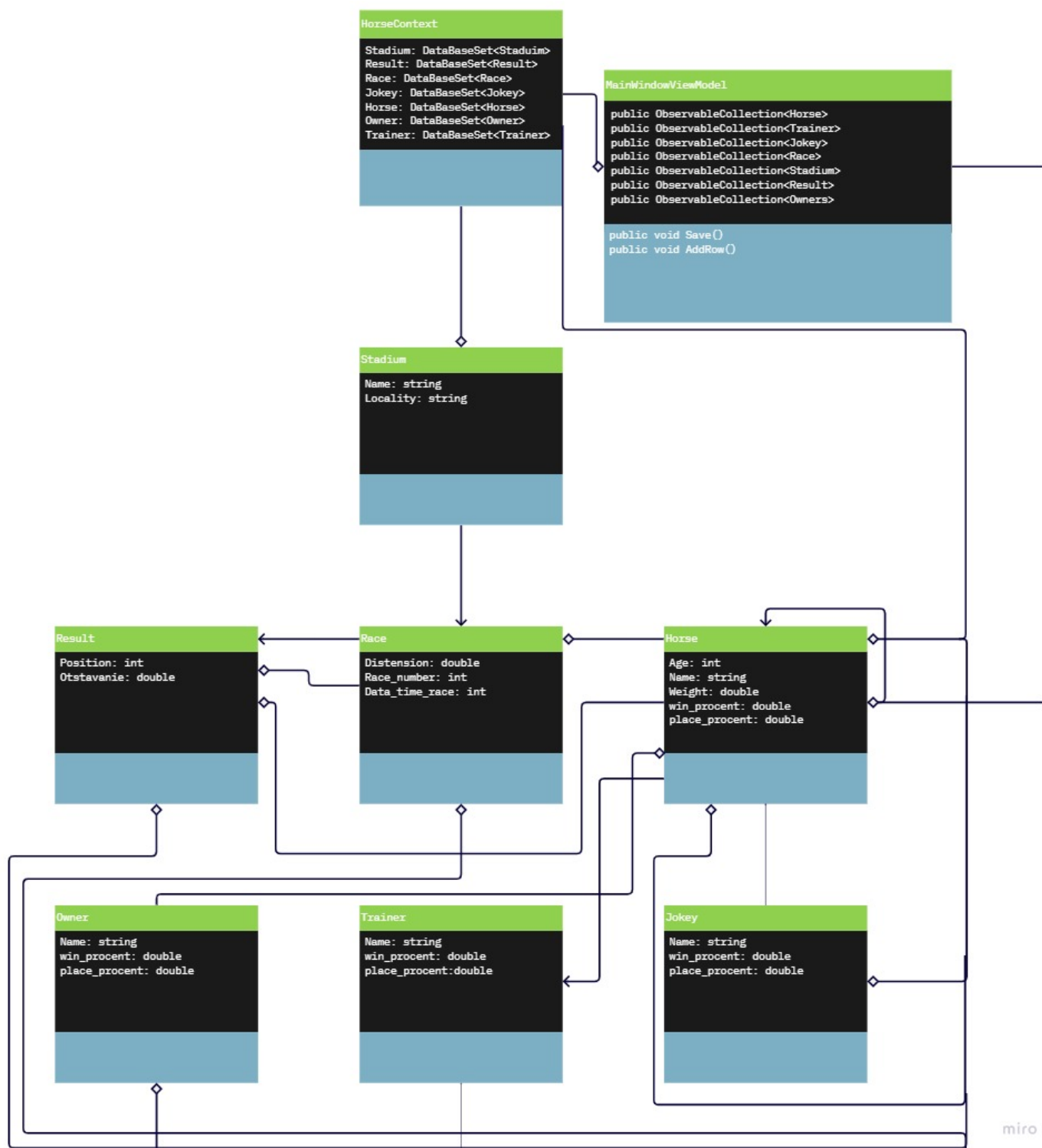


Меню навигации:

1. «Лошадь, Тренер, Жокей, Владелец, Забег, Результаты забегов, Стадион»— вкладки, при нажатии, которых будут отображены информация о каждой вкладке.
2. «Add Row» — добавление данных для записи в таблицу.
3. Query Manager — кнопка навигации, для перехода к интерфейсу работы с запросами.
4. «Delete Row» — кнопка, позволяющая удалить информацию из таблицы.
5. «Save» — Кнопка, позволяющая сохранить информацию таблицы.

Часть 4.

Создание диаграммы классов.



Классы:

1. Stadium

- 1.1. public string name
- 1.2. public string race

2. Result -

- 2.1. public int position
- 2.2. public double otstavanie

3. Race

- 3.1. public double distension
- 3.2. public int race_number
- 3.3. public int Data_Time_Race

4. Horse

- 4.1. public double weight
- 4.2. public double win_procent
- 4.3. public double place_procent
- 4.4. public int age
- 4.5. public string name

5. Owner

- 5.1. public double win_procent
- 5.2. public double place_procent
- 5.3. public string name

6. Trainer

- 6.1. public double win_procent
- 6.2. public double place_procent
- 6.3. public string name

7. Jokey

- 7.1. public double win_procent
- 7.2. public double place_procent
- 7.3. public string name

8. DataBaseContext- класс, который позволяет работать с базой данных через модели. DbSet<Model> представляет собой сущности в базе данных.

8.1. public DataBaseSet <Staduim>

8.2. public DataBaseSet <Result>

8.3. public DataBaseSet <Race>

8.4. public DataBaseSet <Jokey>

8.5. public DataBaseSet <Owner>

8.6. public DataBaseSet <Horse>

8.7. public DataBaseSet <Trainer>

9. MainWindowViewModel – здесь представлены основные данные и методы для работы с view, каждая ObservableCollection – коллекция, в которой содержимое будет отображаться в окне программы, которые связаны с моделями из DataBaseContext.

9.1. public ObservableCollection<

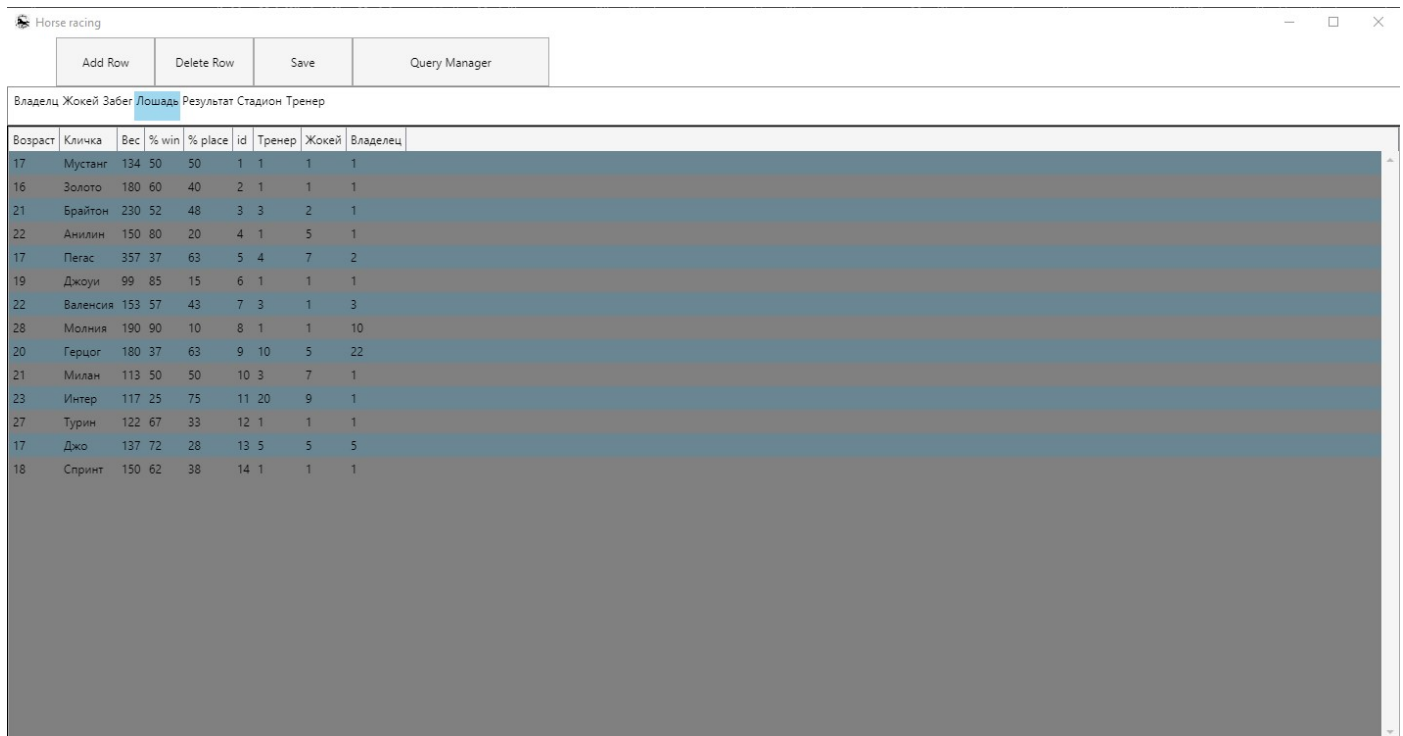
10. public void Save() – Функция сохранения данных в таблице.

11. public void AddRow() – функция добавления в таблицу данных.

Часть 5.

Реализация основного окна приложения

Было реализовано окно приложения, где представлены таблицы с базы данных с возможностью добавить и удалить запись



Часть 6.

Реализация менеджера запросов

Было реализовано окно приложения, в котором можно сделать запрос по определенным параметрам.

