

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Кемеровский государственный университет»
Новокузнецкий институт (филиал)

Факультет информатики, математики и экономики

Кафедра информатики и вычислительной техники им. В. К. Буторина

В. Е. Шехтман, О. И. Новоселова

БАЗЫ ДАННЫХ

*Методические указания к выполнению курсовой работы
для обучающихся по направлениям подготовки*

*09.03.01 Информатика и вычислительная техника
направленность (профиль) Автоматизированные системы обработки
информации и управления*

*09.03.03 Прикладная информатика
направленность (профиль) Прикладная информатика в экономике*

Новокузнецк
2019

УДК [378.147.85:004.65](072)
ББК 74.484(2Рос-4Кем)я73+32.972.134я73

Ш54

Ш54 «Базы данных. Методические указания к выполнению курсовой работы» : метод. указ (текст. электрон. изд.)/ В.Е. Шехтман, О.И. Новоселова ; Новокузнец. ин-т (фил.) Кемеров. гос. ун-та – Новокузнецк: НФИ КемГУ, 2019. – 30 с.

Приводятся методические указания к выполнению курсовой работы по дисциплине «Базы данных»: задание на курсовую работу, примерные темы, требования к структуре, оформлению и содержанию всех разделов курсовой работы; примеры оформления отдельных разделов пояснительной записки.

Методические указания предназначены для студентов очной, очно-заочной и заочной форм обучения направления подготовки 09.03.01 «Информатика и вычислительная техника» профиля «Автоматизированные системы обработки информации и управления» и 09.03.03 «Прикладная информатика» профиля «Прикладная информатика в экономике» факультета информатики, математики и экономики.

Рекомендовано
на заседании кафедры
информатики и вычислительной техники
им. В. К. Буторина
17 октября 2019 года.
Заведующий кафедрой



А. В. Маркидонов

Утверждено
методической комиссией факультета
информатики, математики и экономики
14 ноября 2019 года.
Председатель методкомиссии



Г.Н. Бойченко

УДК [378.147.85:004.65](072)
ББК 74.484(2Рос-4Кем)я73+32.972.134я73

© Шехтман В.Е., Новоселова О. И., 2019
© Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Кемеровский государственный университет»,
Новокузнецкий институт (филиал), 2019

Текст представлен в авторской редакции

Содержание

Введение.....	3
1. Цель и задачи курсовой работы	3
2. Задание на курсовую работу	3
3. Структура курсовой работы	5
4. Оформление пояснительной записки к курсовой работе	6
5. Тематика курсовых работ	6
6. Защита курсовой работы и её оценка	9
Список использованных источников	10
Приложение А Пример титульного листа пояснительной записки к курсовой работе	12
Приложение Б Пример инфологической модели базы данных	13
Приложение В Пример даталогической модели базы данных	13
Приложение Г Фрагмент SQL-скрипта для создания базы данных ветеринарной клиники	14
Приложение Д Фрагмент оформления основной части пояснительной записки к КР.....	16

Введение

Методы и понятия баз данных являются фундаментом построения информационных и информационно-управляющих систем самого широкого профиля. Они лежат в основе грандиозных комплексов программ, позволяющих управлять предприятиями и промышленными установками; но без них не обходятся и информационные системы, с которыми человек сталкивается на каждом шагу – в системах связи, на транспорте, в больницах... Выполнение курсовой работы по дисциплине Базы данных преследует цель закрепление знаний, умений и навыков в проектировании, разработки и использовании баз данных.

1. Цель и задачи курсовой работы

Целью курсовой работы по дисциплине Базы данных является закрепление знаний и навыков по проектированию структуры БД, ее инфологической и даталогической модели, создания БД с помощью SQL - скриптов, а также работа с созданной БД с помощью разработанного программного настольного или web –приложения.

Задачи курсовой работы (КР):

1. Проектирование БД для выбранной предметной области.
2. Создание БД средствами SQL со всеми необходимыми объектами.
3. Разработка приложения на языке ООП или на языке PHP для манипулирования данными созданной БД.
4. Оформление пояснительной записки к курсовой работе. Пример в [приложении Д](#).

2. Задание на курсовую работу

В процессе выполнения задания студент должен:

1. Выбрать предметную область, на описание которой будет направлена работа.
 - 1.1 Описать предметную область на содержательном уровне (на русском языке).
 - 1.2 Объяснить цели разработки, требования к функциональности, т.е. перечислить функции (задачи), которые должно реализовывать будущее приложение (основные вопросы, на которые оно должно отвечать).

2. Разработать для этой предметной области модель “сущность-связь” (инфологическую модель). Обосновать мощност и обязательность или необязательность связей, выбор ключей. Пример в [приложении Б](#).
3. Преобразовать инфологическую модель в даталогическую. Пример в [приложении В](#).
 - 3.1 Обосновать выбор того или иного способа реализации схем отношений и связей между ними, выбор типов для атрибутов отношений.
 - 3.2 Позаботиться об отсутствии в даталогической модели избыточности и аномалий изменения и удаления. Для этого обосновать уровень нормализации соответствующих отношений.
 - 3.3 Обеспечить ссылочную целостность и целостность на уровне таблиц, записей, атрибутов, задав первичные и внешние ключи, ограничения других типов.
 - 3.4 Реализовать модель в виде физической базы данных. Указать последовательность команд SQL для создания базы данных со всеми необходимыми объектами (см. пример скрипта в [приложении Г](#)). В качестве СУБД можно использовать Microsoft Access, Microsoft SQL Server или другую по согласованию с преподавателем.
4. Спроектировать интерфейс пользователя.
5. Разработать, основываясь на проекте базы данных, полученном в ходе выполнения п.п.1.-4., программное приложение (например, на Delphi, C# или PHP) для ввода, редактирования нормативно-справочной информации и фактических данных, а также для построения отчетов.

Желательно иметь несколько типов пользователей для разделения прав доступа к функциям приложения. Каждый из пользователей может идентифицироваться своим именем и паролем.
6. Требования к программному приложению:
 - 6.1 приложение должно иметь стандартную архитектуру (главная форма, модуль данных, универсальная форма просмотра-редактирования нормативно-справочной информации с возможностью формирования твердой копии справочников, формы ввода фактов и формы построения отчетов);
 - 6.2 в качестве интерфейса доступа к данным следует использовать Microsoft ADO. Человеко-машинный интерфейс может быть любым. При необходимости, следует реализовать форму просмотра-редактирования справочника иерархической структуры. (Более подробно в [5]);
 - 6.3 желательно использовать в явном виде транзакции в случае, когда требуется выполнить сложную составную операцию (например, перевод

денег с одного счета на другой в банке выполняется с помощью нескольких SQL-запросов, которые должны выполняться полностью либо, при невозможности этого, изменения должны быть полностью отменены, несмотря на их частичное выполнение).

7. Оформить пояснительную записку к КР.

3. Структура курсовой работы

Курсовая работа содержит программную часть и пояснительную записку.

Программная часть представляется в электронном виде и демонстрируется при защите курсовой работы. Пояснительная записка представляется в электронном виде на занятиях в компьютерном классе и бумажном варианте при публичной защите курсовой работы.

Пояснительная записка должна содержать:

- ❖ титульный лист;
- ❖ реферат;
- ❖ введение;
- ❖ анализ предметной области и проектирование базы данных;
- ❖ обоснование выбранных технологий выполнения курсовой работы (языка программирования, среды разработки);
- ❖ разработка программного кода;
- ❖ заключение;
- ❖ список использованных источников;
- ❖ приложения (модели проектируемой БД, пример SQL-скриптов код программы с комментариями).

Пример оформления титульного листа пояснительной записки к КР приведен в [приложении А](#).

Реферат – это характеристика пояснительной записки и конспективное изложение хода проделанной работы. Реферат содержит: сведения о количестве страниц, иллюстраций, таблиц, приложений, использованных источников; перечень ключевых слов.

Введение содержит в сжатой форме все положения, обоснованию которых посвящена работа: актуальность темы, характеристику цели и перечень задач, необходимых для ее реализации, краткая характеристика используемых методов ИТ, характеристика результатов работы и их практическую ценность.

Содержание содержит названия всех частей пояснительной записки с номерами страниц, с которых они начинаются.

В **основной части** пояснительной записки к КР излагается ход работы, обосновываются и формулируются ее промежуточные и конечные результаты. В основной части пояснительной записки должны быть представлены:

- ❖ ER-диаграмма (инфологическую модель) с обсуждением особенностей и обоснованием конкретных решений;
- ❖ даталогическая модель данных с описанием степени нормализации;
- ❖ скрипт на языке SQL для создания базы данных “с нуля”;
- ❖ описание физической структуры приложения (можно с использованием UML-диаграммы компонентов);
- ❖ описание интерфейса пользователя (меню, экраны) – по сути, краткую инструкцию по работе с приложением;
- ❖ основные SQL-запросы, реализующие ввод сведений о фактах, возникающих в предметной области (в рамках транзакции, если необходимо);
- ❖ основные SQL-запросы, лежащие в основе отчетов, что обычно требует использования агрегатных функций для получения укрупненных характеристик положения дел в предметной области (в целом, по категориям товаров или услуг, видам автомобилей или заболеваний, временным диапазонам, и т.д.).

Текст основной части пояснительной записки должен делиться на разделы, подразделы, пункты и подпункты

По объему КР должна быть не менее 25-30 страниц печатного текста. Краткое изложение основных разделов пояснительной записки приведено [в приложении Д](#).

4. Оформление пояснительной записки к курсовой работе.

Текст пояснительной записки к курсовой работе оформляется по правилам, изложенным в [8]. Листы пояснительной записки должны быть сшиты с помощью скоросшивателя. Электронные версии программы, пояснительной записки, презентации доклада должны быть переданы на электронном носителе преподавателю.

5. Тематика курсовых работ.

1. Разработка базы данных и программного приложения для подсчета домашних расходов.

2. Разработка базы данных и программного приложения для учета регистрации в ГАИ.
3. Разработка базы данных и программного приложения для учета компьютерных запчастей.
4. Разработка базы данных и программного приложения тэгового файлового менеджера.
5. Разработка базы данных и программного приложения для расчета заработной платы.
6. Разработка веб-сервера ЖКХ.
7. Разработка приложения для работы с базой данных по теме «Клининговое агентство».
8. Разработка базы данных и программного приложения «Электронная регистратура».
9. Разработка модуля таблица рейтинга студентов для электронно-образовательной среды «Портфолио» НФИ КемГУ.
10. Разработка базы данных и программного приложения для учета регистрации в ГАИ.
11. Разработка приложения для работы с базой данных по теме «Электронный журнал».
12. Разработка базы данных и программного приложения для ведения электронного ежедневника.
13. Разработка базы данных и программного приложения для студенческой поликлиники.
14. Разработка базы данных и программного приложения для кассы кинотеатра.
15. Разработка базы данных и программного приложения учета продаж.
16. Разработка базы данных и программного приложения для управления складом.
17. Разработка базы данных и программного приложения агентства недвижимости.
18. Разработка базы данных и программного приложения рекламного агентства.
19. Разработка базы данных и программного приложения для оплаты коммунальных услуг.
20. Разработка базы данных и программного приложения для отдела кадров.
21. Разработка базы данных и программного приложения ветеринарной клиники.
22. Разработка базы данных и программного приложения для типовой средней школы.

- 23.Разработка базы данных и программного приложения информационной службы института.
- 24.Разработка базы данных и программного приложения справочной аптек города.
- 25.Разработка базы данных и программного приложения для библиотеки института.
- 26.Разработка базы данных и программного приложения для налоговой инспекции.
- 27.Разработка базы данных и программного приложения работы гостиницы.
- 28.Разработка базы данных и программного приложения автосалона.
- 29.Разработка базы данных и программного приложения службы занятости населения.
- 30.Разработка базы данных и программного приложения туристического агентства.

6. Защита курсовой работы и её оценка

Защита курсовой работы включает предоставление пояснительной записки к курсовой работе в бумажной копии и публичной защите, состоящей из доклада и презентации. В ходе публичной защиты оценивается способность студента представить свою работу, отвечать на вопросы. При оценке пояснительной записки в бумажной форме проверяется полнота наполнения, правильность решения поставленных задач, грамотность формулировок, правильность оформления документации в соответствии с ЕСПД.

Балльно-рейтинговая система оценивания курсовой представлена в таблице ниже.

Критерии оценивания	Баллы
1. Описание предметной области и формулировка функциональных требований к приложению	0-10
2. Разработка инфологической и даталогической моделей базы данных. Оценка степени нормализации БД.	0-10
3. Разработка SQL скрипта создания БД	0-10
4. Разработка структуры приложения и интерфейса пользователя	0-10
5. Поддержка нормативно-справочной информации и ввода/корректировки данных в приложении	0-10
6. Обеспечение разделения прав пользователей, применение транзакций для работы в многопользовательском режиме и обеспечения корректности хранимой информации	0-10
7. Наличие возможности формирования отчетности	0-10
8. Оформление пояснительной записки	0-10
9. Защита курсовой (раскрытие содержания темы, структура и качество доклада, оперирование профессиональной терминологией, ответы на вопросы)	0-20

Перевод баллов из 100-балльной шкалы в числовой и буквенный эквивалент

Сумма баллов по курсовой	Оценка	Буквенный эквивалент
0-50	2	неудовлетворительно
51-65	3	удовлетворительно
66-85	4	хорошо
86 - 100	5	отлично

Список использованных источников

1. Нестеров, С.А. Базы данных : учебник / С.А. Нестеров. – М. : Издательство Юрайт, 2016. – 230 с. – ISBN 978-5-9916-6427-1 – URL:
<http://www.biblio-online.ru/viewer/B5E199E0-F0B1-4B55-AF98-9B7BC4841BCC>
(дата обращения: 19.10.2019). – Текст : электронный
2. Базы данных. В 2-х кн. Кн. 2. Распределенные и удаленные базы данных: Учебник / В.П. Агальцов. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 272 с.: ил.; 60х90 1/16. - (Высшее образование). (переплет) ISBN 978-5-8199-0394-0, 2000 экз. URL:
<http://znanium.com/bookread.php?book=372740>
(дата обращения: 19.10.2019). – Текст : электронный
3. ГОСТ 19.101-77 Единая система программной документации (ЕСПД). Виды программ и программных продуктов. [Электронный ресурс]. – Введ. 1980-01-01. –Электрон. дан. – М.: Стандартиформ, 2010. – 2 с. – (Межгосударственный стандарт). – URL:
<http://docs.cntd.ru/document/1200007627>,
(дата обращения: 19.10.2019). – Текст : электронный
4. Леоненков, А. В. Самоучитель UML. [Электронный ресурс]: Самоучитель / Леоненков А.В., - 2-е изд., перераб. и доп. -

- СПб:БХВ-Петербург, 2015. - 418 с. ISBN 978-5-9775-1216-9 - URL:
<http://znanium.com/bookread2.php?book=939591>
(дата обращения: 19.10.2019). – Текст : электронный
5. Информационные аналитические системы [Электронный ресурс] : учебник / Т. В. Алексеева, Ю. В. Амириди, В. В. Дик и др.; под ред. В. В. Дика. - М.: МФПУ Синергия, 2013. - 384 с. - (Университетская серия). - ISBN 978-5-4257-0092-6.
URL: <http://znanium.com/bookread.php?book=451186>
(дата обращения: 19.10.2019). – Текст : электронный
6. Шехтман, В. Е. Базы данных, SQL и все такое [Текст]: курс лекций / В. Е. Шехтман : НФИ КемГУ . –Новокузнецк , 2006. - 239 с.
7. Буч, Г. Язык UML. [Электронный ресурс] : Руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон. - 2-е изд.: Пер. с англ. Н. Мухин. – М.: ДМК Пресс, 2008. - 496 с.: ил. - ISBN 5-94074-334-X. - URL: <http://znanium.com/bookread2.php?book=409294>
(дата обращения: 19.10.2019). – Текст : электронный
8. Жибинова, И. А. Правила оформления учебных работ студентов [Текст] : учебно-метод. указ. / И. А. Жибинова, Е. А. Аракелян, О. В. Соколова, Ю. Н. Соина-Кутищева ; НФИ КемГУ. – Новокузнецк, 2018. – 124 с.

Приложение А Пример титульного листа пояснительной записки к курсовой работе

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ И
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Новокузнецкий институт (филиал)
федерального государственного бюджетного образовательного
учреждения высшего образования
«Кемеровский государственный университет»**

Факультет информатики, математики и экономики

Кафедра информатики и вычислительной техники им. В. К. Буторина

Хожаинов Дмитрий Иванович
гр. ИВТ-15

**РАЗРАБОТКА БАЗЫ ДАННЫХ И ПРИЛОЖЕНИЯ «ВЕТЕРИНАРНАЯ
КЛИНИКА»**

Курсовая работа
по дисциплине «Базы данных»

по направлению подготовки 09.03.01 Информатика и вычислительная
техника
направленность (профиль) подготовки «Автоматизированные системы
обработки информации управления»

Проверил:
доцент
В.Е. Шехтман

Общий балл: _____

Оценка: _____

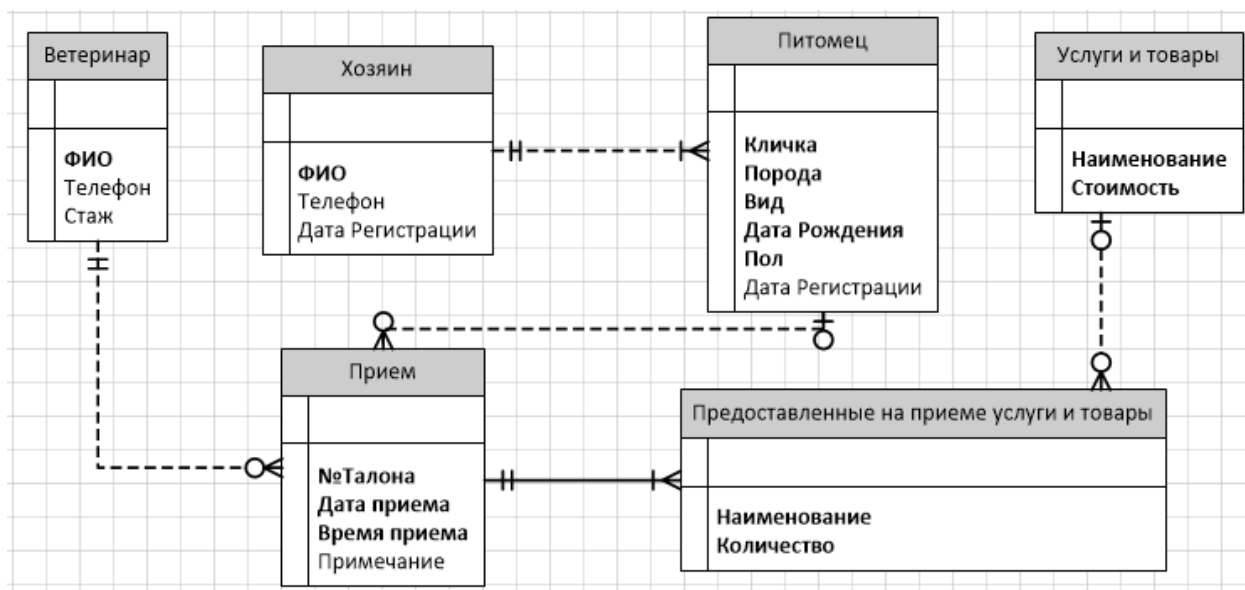
_____ подпись

« ____ » _____ 20__ г.

Новокузнецк 20 _____

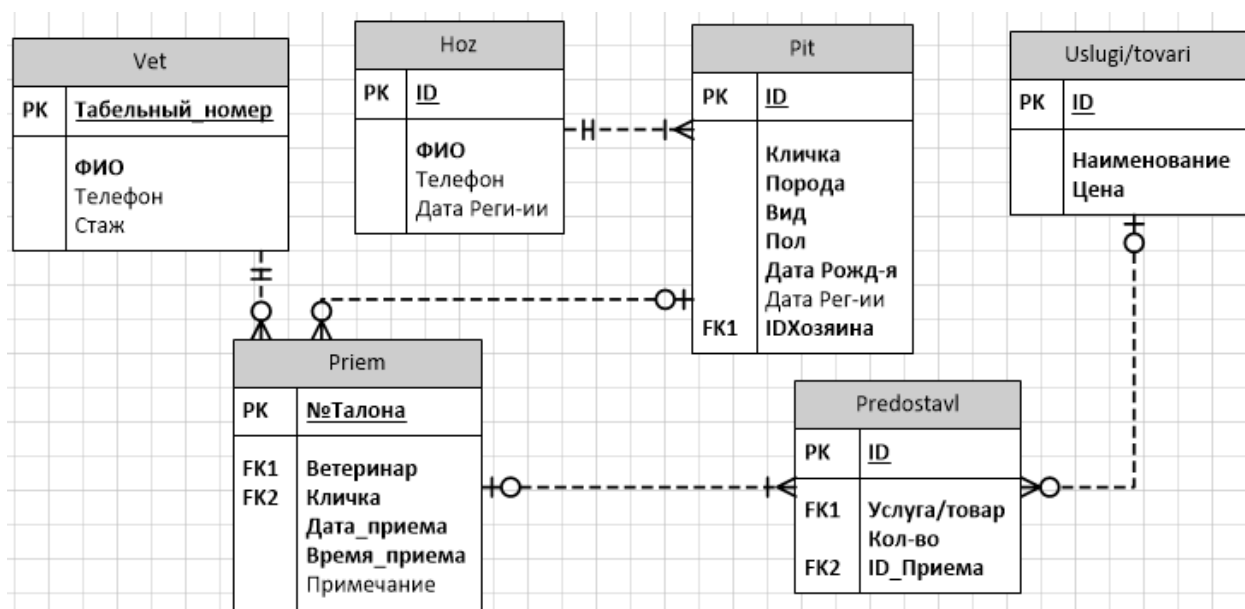
Приложение Б Пример инфологической модели базы данных

предметной области “Ветеринарная клиника”.



Приложение В Пример даталогической модели базы данных

предметной области “Ветеринарная клиника”. (СУБД MS SQL 2008).



Приложение Г Фрагмент SQL-скрипта для создания базы данных ветеринарной клиники

```
CREATE TABLE Hoz (--Таблица с данными о хозяине
    ID          AutoIncrement,
    [Дата_Рег-ии]    DateTime,
    [Телефон]       Text(255) WITH COMP,
    [ФИО]           Text(255) WITH COMP NOT NULL,
    CONSTRAINT PrimaryKey PRIMARY KEY (ID)
);

CREATE TABLE Pit (--Таблица с данными о питомце
    ID          AutoIncrement,
    [Вид]       Text(255) WITH COMP NOT NULL,
    [Дата_Рег-ии]    DateTime,
    [Дата_Рожд-я]    DateTime NOT NULL,
    [Кличка]       Text(255) WITH COMP NOT NULL,
    [Пол]         Text(255) WITH COMP NOT NULL,
    [Порода]       Text(255) WITH COMP NOT NULL,
    [Хозяин]       Long NOT NULL,
    CONSTRAINT PrimaryKey PRIMARY KEY (ID)
);

CREATE TABLE Predostavl (--Таблица с данными о предоставленных на приеме
услугах
    ID          AutoIncrement,
    [ID_Приема]    Long NOT NULL,
    [Кол-во]       Long NOT NULL,
    [Услуга/товар]    Long NOT NULL,
    CONSTRAINT PrimaryKey PRIMARY KEY (ID)
);

CREATE TABLE Priem (--Таблица с данными о приеме
    [№Талона]      AutoIncrement,
    [Ветеринар]     Long NOT NULL,
    [Время_приема]    DateTime NOT NULL,
    [Дата_приема]     DateTime NOT NULL,
    [Кличка]        Long NOT NULL,
```

```

[Примечание]      Text(255) WITH COMP,
CONSTRAINT PrimaryKey PRIMARY KEY ([№Талона])
);

CREATE TABLE Uslugi/tovari (--Таблица слуг и товаров
ID                AutoIncrement,
[Наименование]    Text(255) WITH COMP NOT NULL,
[Цена]            Long NOT NULL,
CONSTRAINT PrimaryKey PRIMARY KEY (ID)
);

CREATE TABLE Vet (--Таблица с данными о ветеринарах
[Стаж]            Long,
[Табельный_номер] AutoIncrement,
[Телефон]         Text(255) WITH COMP,
[ФИО]             Text(255) WITH COMP NOT NULL,
CONSTRAINT PrimaryKey PRIMARY KEY ([Табельный_номер])
);

ALTER TABLE Pit
    ADD CONSTRAINT HozPit FOREIGN KEY ([Хозяин]) REFERENCES Hoz;

ALTER TABLE Predostavl
    ADD CONSTRAINT PriemPredostavl FOREIGN KEY ([ID_Приема])
REFERENCES Priem;

ALTER TABLE Predostavl
    ADD CONSTRAINT Uslugi/tovariPredostavl FOREIGN KEY ([Услуга/товар])
REFERENCES Uslugi/tovari;

ALTER TABLE Priem
    ADD CONSTRAINT PitPriem FOREIGN KEY ([Кличка]) REFERENCES Pit;

ALTER TABLE Priem
    ADD CONSTRAINT VetPriem FOREIGN KEY ([Ветеринар]) REFERENCES Vet;

CREATE INDEX PitKlichka ON Pit ([Кличка]);

```

Приложение Д Фрагмент оформления основной части пояснительной записки к КР

Разработка оконного приложения

Описание интерфейсной части программы

Программа может использовать непосредственно самим ветеринаром или сотрудником регистратуры. После обслуживания клиента, ветеринар должен внести изменения в список предоставленных на приеме услуг и товаров. Или передать сведения в регистратуру.

Приложение представляет собой исполняемый файл, при запуске которого на экран выводится главная форма.



Рисунок 1 - Стартовое окно приложения

Главная форма включает:

Кнопку «База», открывающую форму для просмотра базы данных клиентов и ветеринаров.

Кнопку «Работа с записями», которая, после нажатия на нее, порождает

форму, позволяющую записывать клиента на прием, выбирая ветеринара, дату и время приема.

Кнопку «История приемов», которая открывает форму для просмотра истории приемов и поиска конкретного приема по хозяину, питомцу и дате, а так же регистрации предоставленных услуг и товаров клиенту.

Кнопку «Справка», выводящую краткую справочную информацию о программе.

Кнопку «Выход».

Окно справки, содержит три вкладки:

«Инструкция».

«О программе».

«Об авторе».

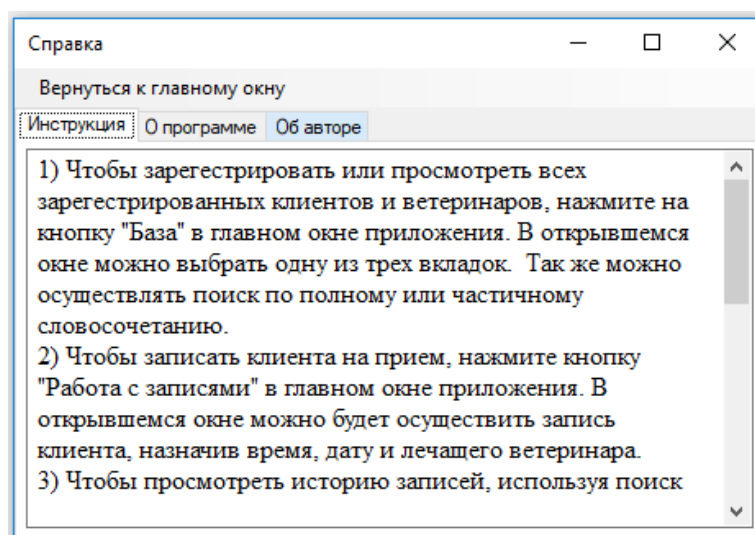


Рисунок 2 - Краткая инструкция по использованию программы

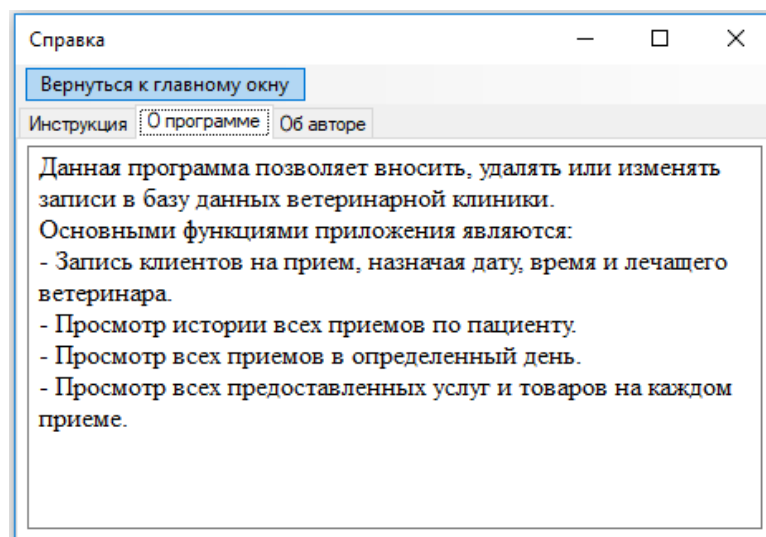


Рисунок 3 - Краткая информация о назначении программы

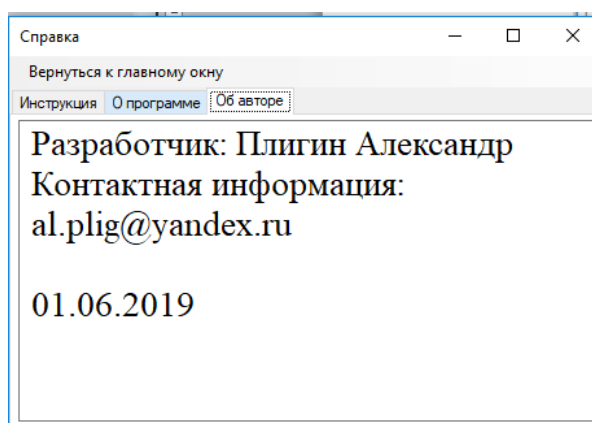


Рисунок 4 - Краткие сведения об авторе

На большинстве форм присутствует панель навигации, для управления базой данных.

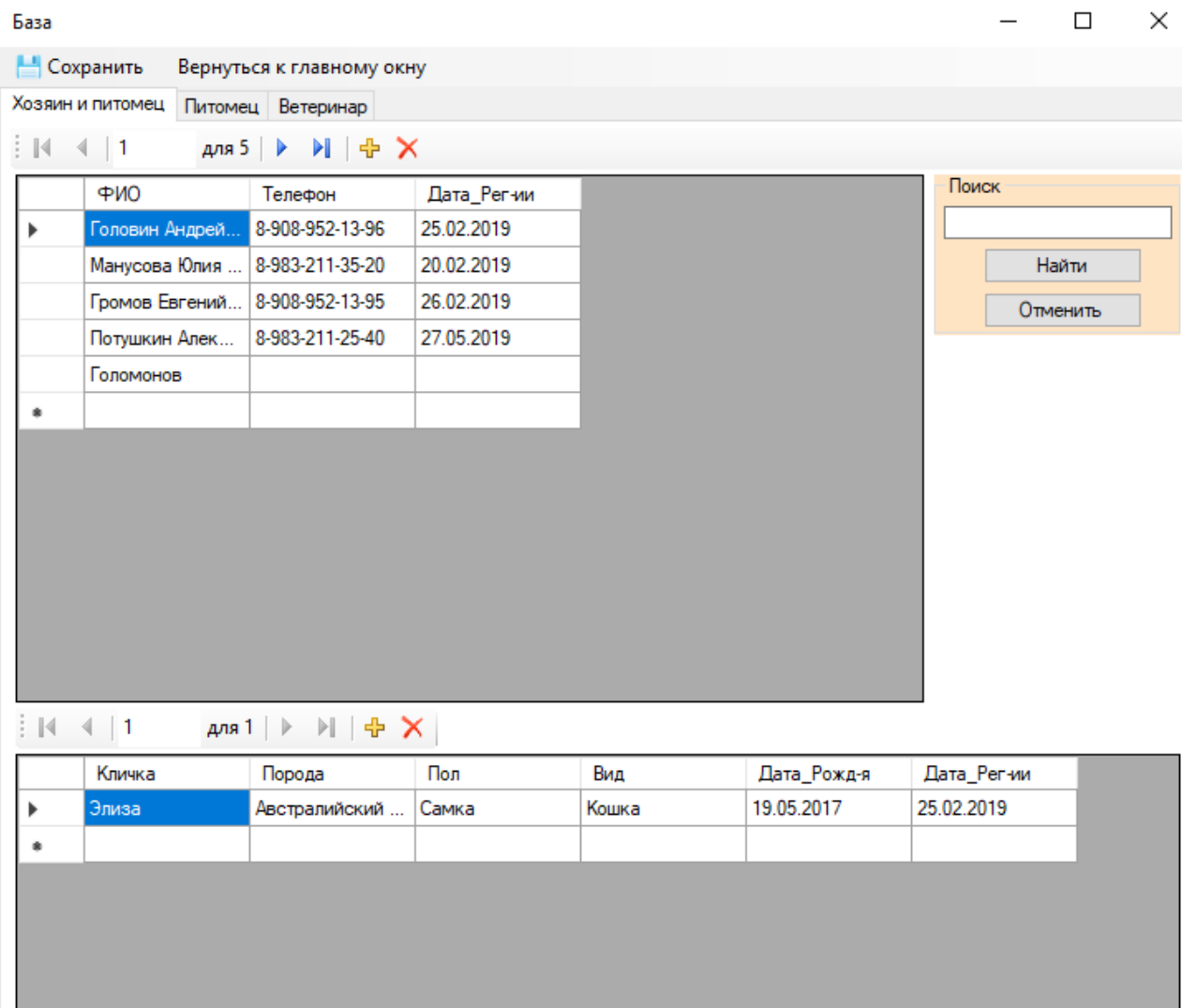


Рисунок 5 - Вкладка "Хозяин и питомец" окна БД

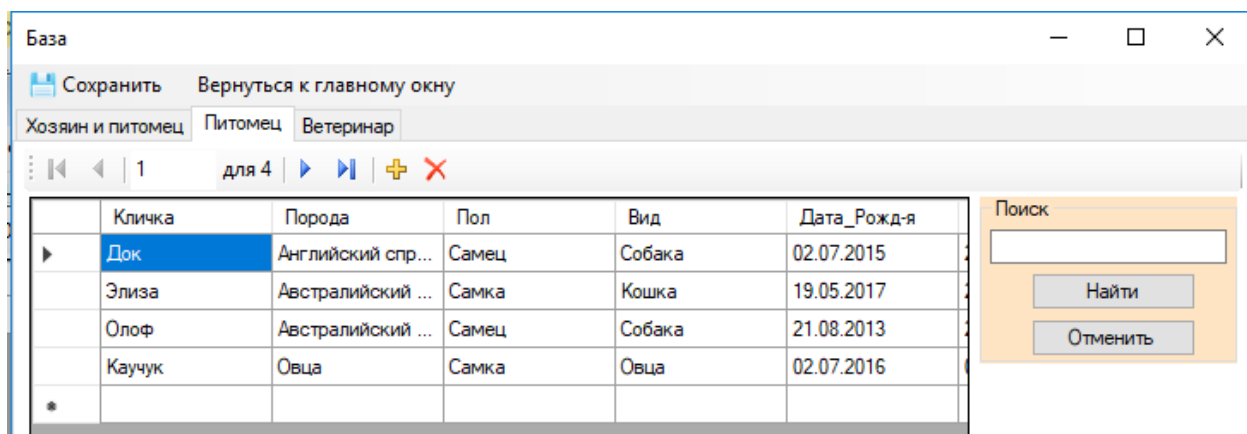


Рисунок 6 - Вкладка "Питомец" окна БД

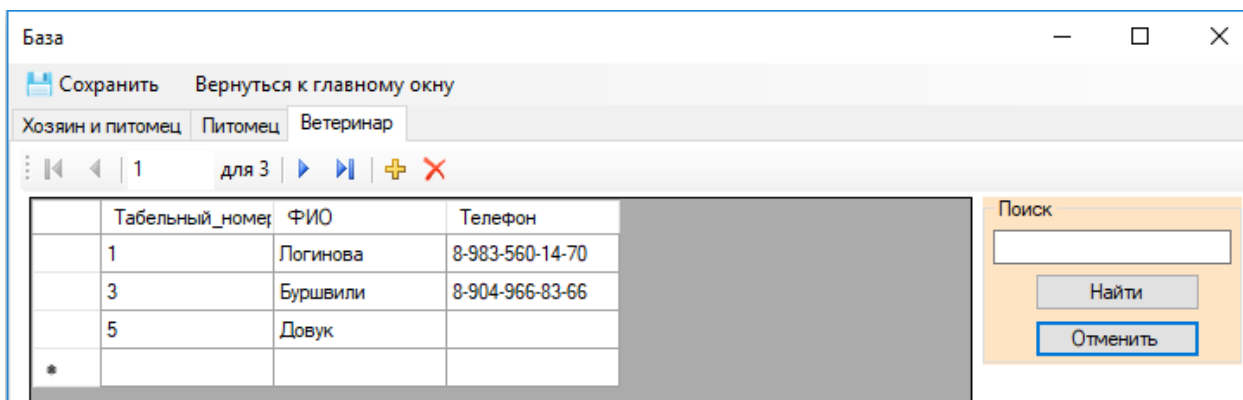


Рисунок 7 - Вкладка "Ветеринар" окна БД

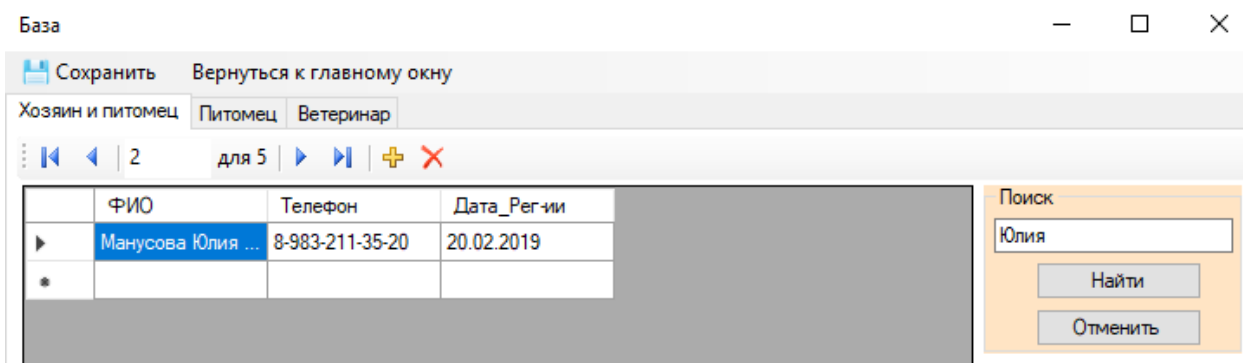


Рисунок 8 - Результат использования поиска

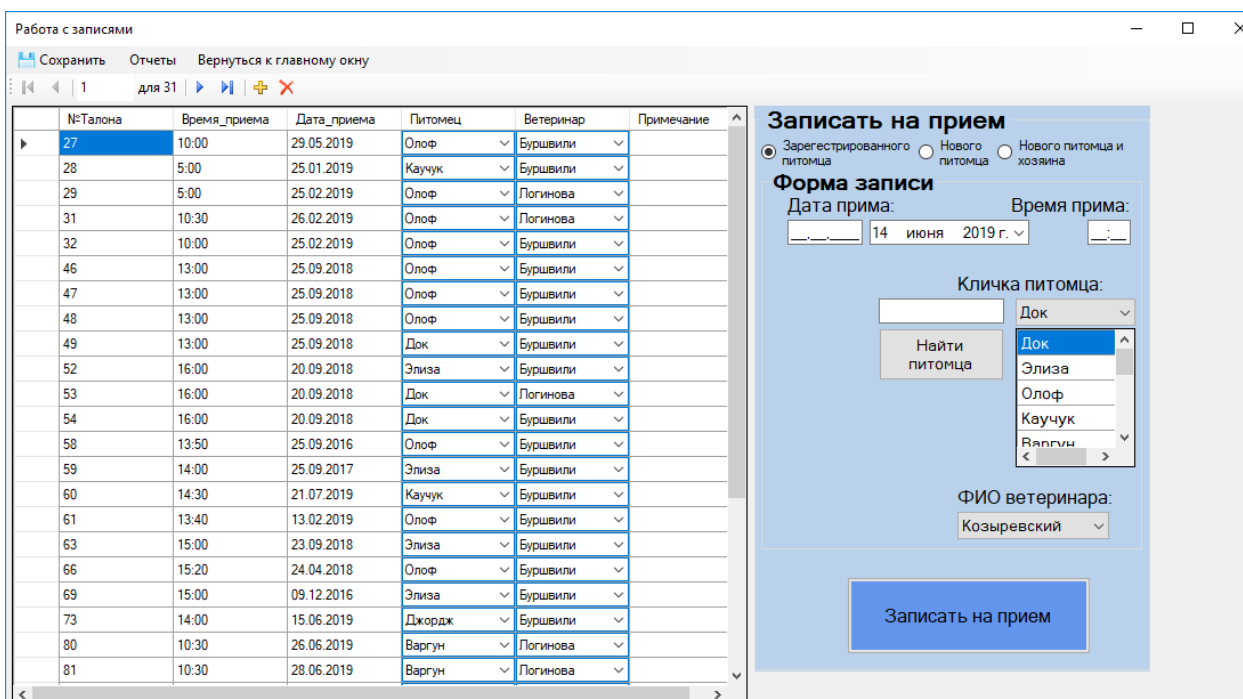


Рисунок 9 - Окно записи на прием

Записать на прием

☒ Зарегистрированного питомца
 ☐ Нового питомца
 ☐ Нового питомца и хозяина

Форма записи

Дата приема: 14 июня 2019 г.
 Время приема:

Кличка питомца:
 Док

ФИО ветеринара: Козыревский

Записать на прием

☐ Зарегистрированного питомца
 ☒ Нового питомца
 ☐ Нового питомца и хозяина

Форма записи

Кличка питомца:
 Хозяин питомца: Головин Андрей Вита.

Вид:

Порода:

Пол:

Дата рождения:
 Дата приема: 14 июня 2019 г.
 Время приема:

ФИО ветеринара: Козыревский

Записать на прием

☐ Зарегистрированного питомца
 ☐ Нового питомца
 ☒ Нового питомца и хозяина

Форма записи

Кличка питомца:
 ФИО хозяина:

Вид:
 Телефон:

Порода:
 Дата приема: 14 июня 2019 г.

Пол:
 Время приема:

Дата рождения:
 ФИО ветеринара: Козыревский

Рисунок 10 - Варианты записи на прием

Отчеты

Вернуться к главному окну

О выручке по каждому ветеринару за все время

О ежедневной выручке за все время

Рисунок 11 - Кнопки для формирования отчетов в текстовый документ

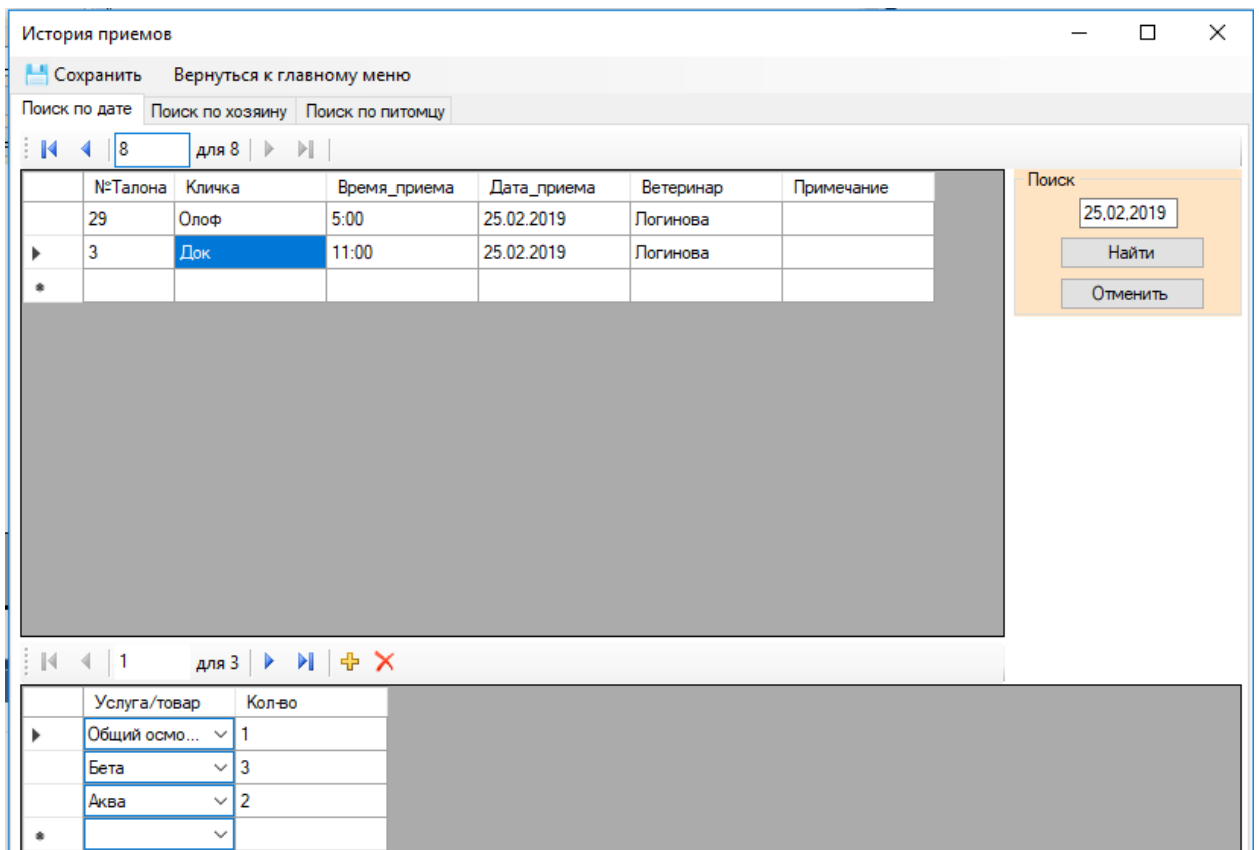


Рисунок 12 - Результат поиска приема по дате

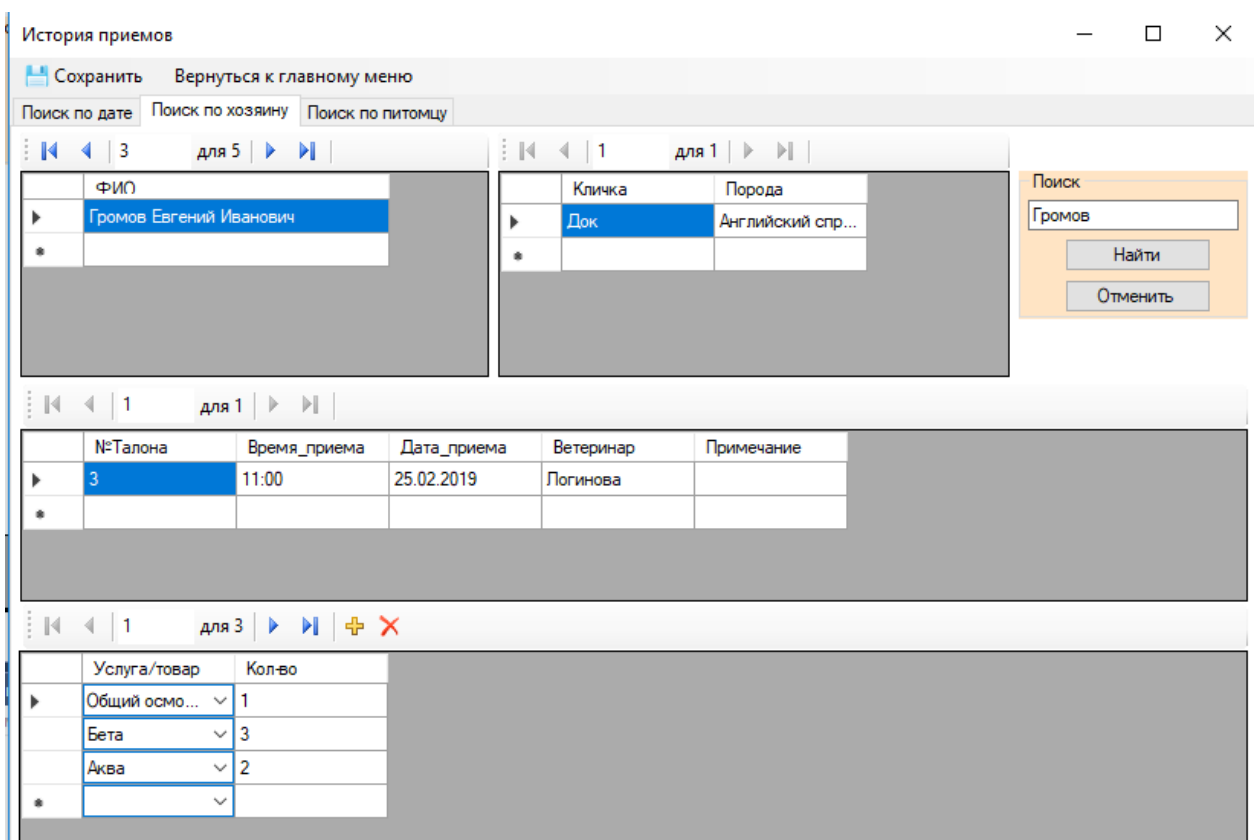


Рисунок 13 - Результат поиска приема по хозяину

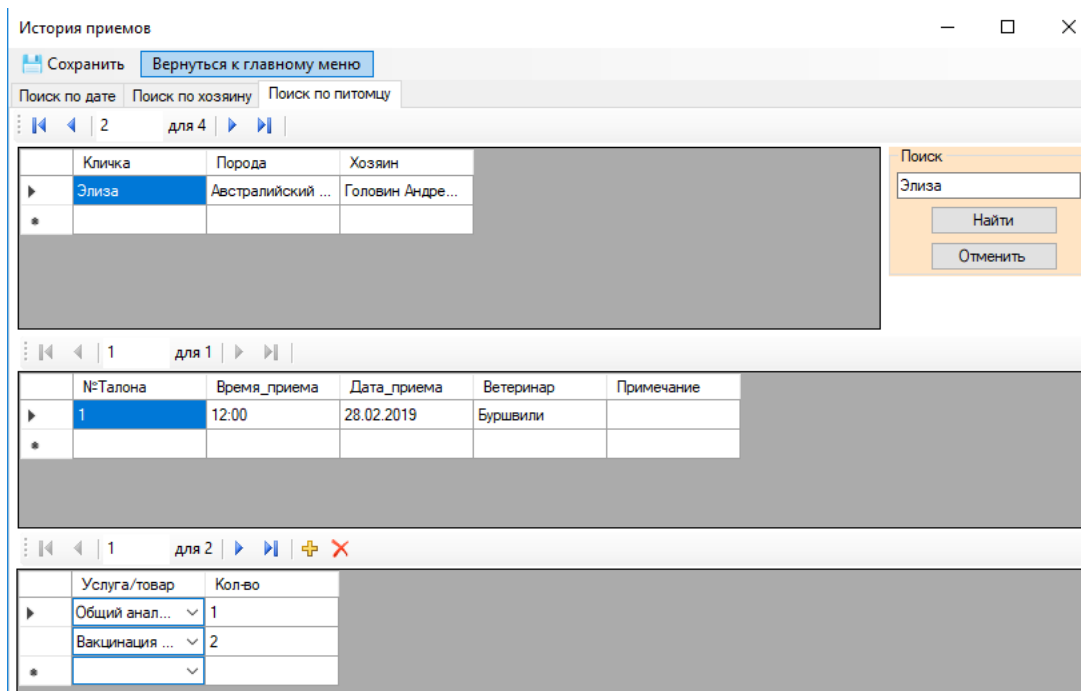


Рисунок 14- Результат поиска приема по кличке питомца

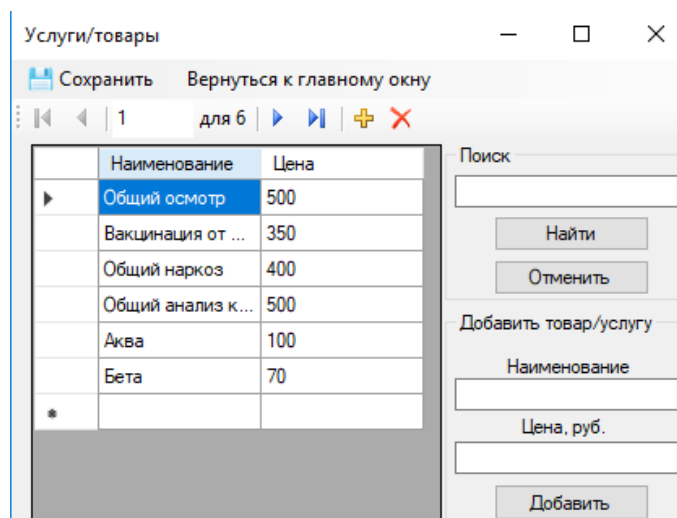


Рисунок 15 - Окно редактирования услуг и товаров

Краткое описание программной части работы

Файл Правка Формат Вид Справка

ОТЧЕТ О ВЫРУЧКЕ ПО КАЖДОМУ ВЕТЕРИНАРУ ЗА ВСЕ ВРЕМЯ

ФИО ветеринара	Выручка за все время (руб.)
Логинава Елизавета Андреевна	2810
Буршвили Олег Николаевич	10100
Козыревский Леонид Игнатьевич	4839

Рисунок 16 - Отчет о выручке каждого ветеринара

SQL – запрос для первого отчета:

```
SELECT vet.ФИО, vir.Выручка AS Выручка_за_все_время FROM
(SELECT pr.Ветеринар, SUM(итог1) AS Выручка FROM (SELECT SUM(итог)
AS итог1, ID_Приема FROM (SELECT ([Кол-во]*Цена) AS итог, * FROM
[Uslugi/tovari] AS u INNER JOIN Predostavl AS p ON p.[Услуга/товар] = u.ID)
GROUP BY ID_Приема) AS z INNER JOIN Priem AS pr ON
pr.[№Талона]=z.ID_Приема GROUP BY pr.Ветеринар) AS vir INNER JOIN
vet ON vet.[Табельный_номер] = vir.Ветеринар;
```

Файл	Правка	Формат	Вид	Справка
ОТЧЕТ О ЕЖЕДНЕВНОЙ ВЫРУЧКЕ ЗА ВСЕ ВРЕМЯ				
День	Количество приемов за день	Выручка за день (руб.)		
25.01.2019	1	1250		
25.02.2019	3	7810		
28.02.2019	1	1200		
07.03.2019	1	500		
29.05.2019	1	2150		
16.06.2019	2	4839		

Рисунок 17 - Отчет о ежедневной выручке

SQL – запрос для второго отчета:

```
SELECT p.Дата_Приема AS День, Count(p.[Дата_Приема]) AS [Кол-
во_приемов_за_день], Sum(t.vir) AS Выручка_за_день FROM (SELECT
[ID_Приема], SUM(Цена*[Кол-во]) AS vir FROM Predostavl AS pr INNER
JOIN [Uslugi/tovari] AS u ON u.ID=pr.[Услуга/товар] GROUP BY
[ID_Приема]) AS t INNER JOIN Priem AS p ON t.[ID_Приема] = p.[№Талона]
GROUP BY p.Дата_Приема;
```

Пример использования в программе транзакций для одной, двух и трех таблиц (проведение транзакций происходит при событии нажатия на кнопку с текстом «Записать на приём/Добавить питомца и записать на приём/Добавить клиента и записать на прием» в окне «Работа с записями»):

```
private void button1_Click(object sender, EventArgs e)
{
    //Создаем соединение
    OleDbConnection conn = new OleDbConnection();
    conn.ConnectionString = Properties.Settings.Default.VKConnectionString;
    conn.Open();
    OleDbCommand c = conn.CreateCommand();
    //Создаем транзакцию
    c.Transaction = conn.BeginTransaction(System.Data.IsolationLevel.ReadCommitted);

    if (radioButton3.Checked == true)
    {
        string dt = maskedTextBox1.Text.Replace(',', '.');
        string vr = maskedTextBox2.Text;
        try
        {
            c.CommandText = "INSERT INTO [Priem]([Время_приема], [Дата_приема],
Кличка, Ветеринар) VALUES('" + vr + "', '" + dt + "', '" + comboBox4.Text + "', '" +
comboBox3.Text + "')";
            c.ExecuteNonQuery();

            //Подтверждаем транзакцию
            c.Transaction.Commit();
            MessageBox.Show("Запись добавлена!", "Сохранение",
MessageBoxButtons.OK);
            this.priemTableAdapter.Fill(this.vKDataSet.Priem);
        }
        catch (Exception ex)
        {
            //Отклоняем транзакцию
            c.Transaction.Rollback();
            MessageBox.Show("Запись добавить не удалось из-за ошибки:\n\r" +
ex.Message, "Ошибка сохранения данных", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        finally
        {
            conn.Close();
        }
    }

    if (radioButton1.Checked == true)
    {
        string dt = maskedTextBox5.Text.Replace(',', '.');
        string vr = maskedTextBox4.Text;
        string dr = maskedTextBox3.Text.Replace(',', '.');
```

```

try
{
    c.CommandText = "INSERT INTO [Pit]([Кличка], [Порода], Пол, Вид,
[Дата_Рожд-я], [Дата_Рег-ии], Хозяин) VALUES('\" + textBox3.Text + \"\", \"\" + textBox7.Text
+ \"\", \"\" + comboBox5.Text + \"\", \"\" + textBox8.Text + \"\", \"\" + dr + \"\", \"\" +
Convert.ToString(DateTime.Now.ToShortDateString()) + \"\", \"\" + comboBox6.Text + \"\")";
    c.ExecuteNonQuery();
    c.CommandText = "SELECT @@IDENTITY";
    int pid = Convert.ToInt32(c.ExecuteScalar());

    c.CommandText = "INSERT INTO [Priem]([Время_приема], [Дата_приема],
Кличка, Ветеринар) VALUES('\" + vr + \"\", \"\" + dt + \"\", \"\" + pid + \"\", \"\" + comboBox8.Text + \"\")";
    c.Parameters.AddWithValue("Кличка", SqlDbType.Int);
    c.Parameters["Кличка"].Value = pid;
    c.ExecuteNonQuery();
    //Подтверждаем транзакцию
    c.Transaction.Commit();
    MessageBox.Show("Питомец зарегистрирован на данного хозяина. Запись
добавлена!", "Сохранение", MessageBoxButtons.OK);
    this.priemTableAdapter.Fill(this.vKDataSet.Priem);
    this.pitTableAdapter.Fill(this.vKDataSet.Pit);
    this.hozTableAdapter.Fill(this.vKDataSet.Hoz);
}
catch (Exception ex)
{
    //Отклоняем транзакцию
    c.Transaction.Rollback();
    MessageBox.Show("Запись добавить не удалось из-за ошибки:\n\r" +
ex.Message, "Ошибка сохранения данных", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
finally
{
    conn.Close();
}
}

if (radioButton2.Checked == true)
{
    string dt = maskedTextBox6.Text.Replace('.', '');
    string vr = maskedTextBox7.Text;
    string dr = maskedTextBox8.Text.Replace('.', '');

    try
    {
        c.CommandText = "INSERT INTO [Hoz]([ФИО], [Телефон], [Дата_Рег-ии])
VALUES('\" + textBox6.Text + \"\", \"\" + textBox11.Text + \"\", \"\" +
Convert.ToString(DateTime.Now.ToShortDateString()) + \"\")";
        c.ExecuteNonQuery();
        c.CommandText = "SELECT @@IDENTITY";
        hid = Convert.ToInt32(c.ExecuteScalar());
    }
}

```

```

        c.CommandText = "INSERT INTO [Pit]([Кличка], [Порода], Пол, Вид,
[Дата_Рожд-я], [Дата_Рег-ии], Хозяин) VALUES('\" + textBox10.Text + "\", '\" + textBox5.Text
+ "\", '\" + comboBox12.Text + "\", '\" + textBox4.Text + "\", '\" + dr + "\", '\" +
Convert.ToString(DateTime.Now.ToShortDateString()) + "\", '\" + hid + \"')";
        c.Parameters.AddWithValue("Хозяин", SqlDbType.Int);
        c.Parameters["Хозяин"].Value = hid;
        c.ExecuteNonQuery();
        c.CommandText = "SELECT @@IDENTITY";
        pid = Convert.ToInt32(c.ExecuteScalar());

        c.CommandText = "INSERT INTO [Priem]([Время_приема], [Дата_приема],
Кличка, Ветеринар) VALUES('\" + vr + "\", '\" + dt + "\", '\" + pid + "\", '\" + comboBox10.Text + \"')";
        c.Parameters.AddWithValue("Кличка", SqlDbType.Int);
        c.Parameters["Кличка"].Value = pid;
        c.ExecuteNonQuery();
        //Подтверждаем транзакцию
        c.Transaction.Commit();
        MessageBox.Show("Хозяин и питомец зарегистрированы. Запись добавлена!",
"Сохранение", MessageBoxButtons.OK);
        this.priemTableAdapter.Fill(this.vKDataSet.Priem);
        this.pitTableAdapter.Fill(this.vKDataSet.Pit);
        this.hozTableAdapter.Fill(this.vKDataSet.Hoz);
    }
    catch (Exception ex)
    {
        //Отклоняем транзакцию
        c.Transaction.Rollback();
        MessageBox.Show("Запись добавить не удалось из-за ошибки:\n\"r\" +
ex.Message, "Ошибка сохранения данных", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    finally
    {
        conn.Close();
    }
}
}

```

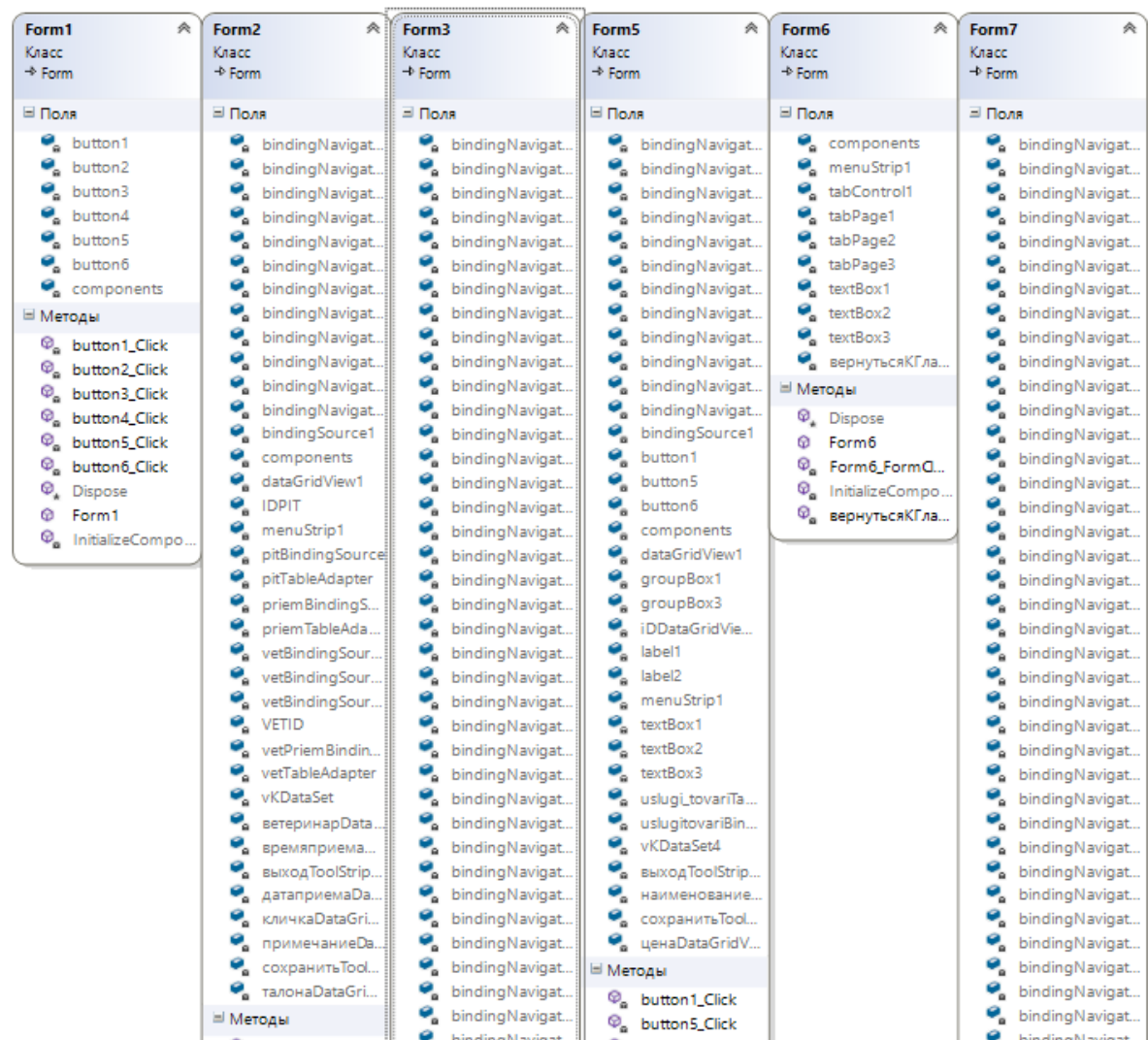


Рисунок 18 - Фрагмент диаграммы классов приложения

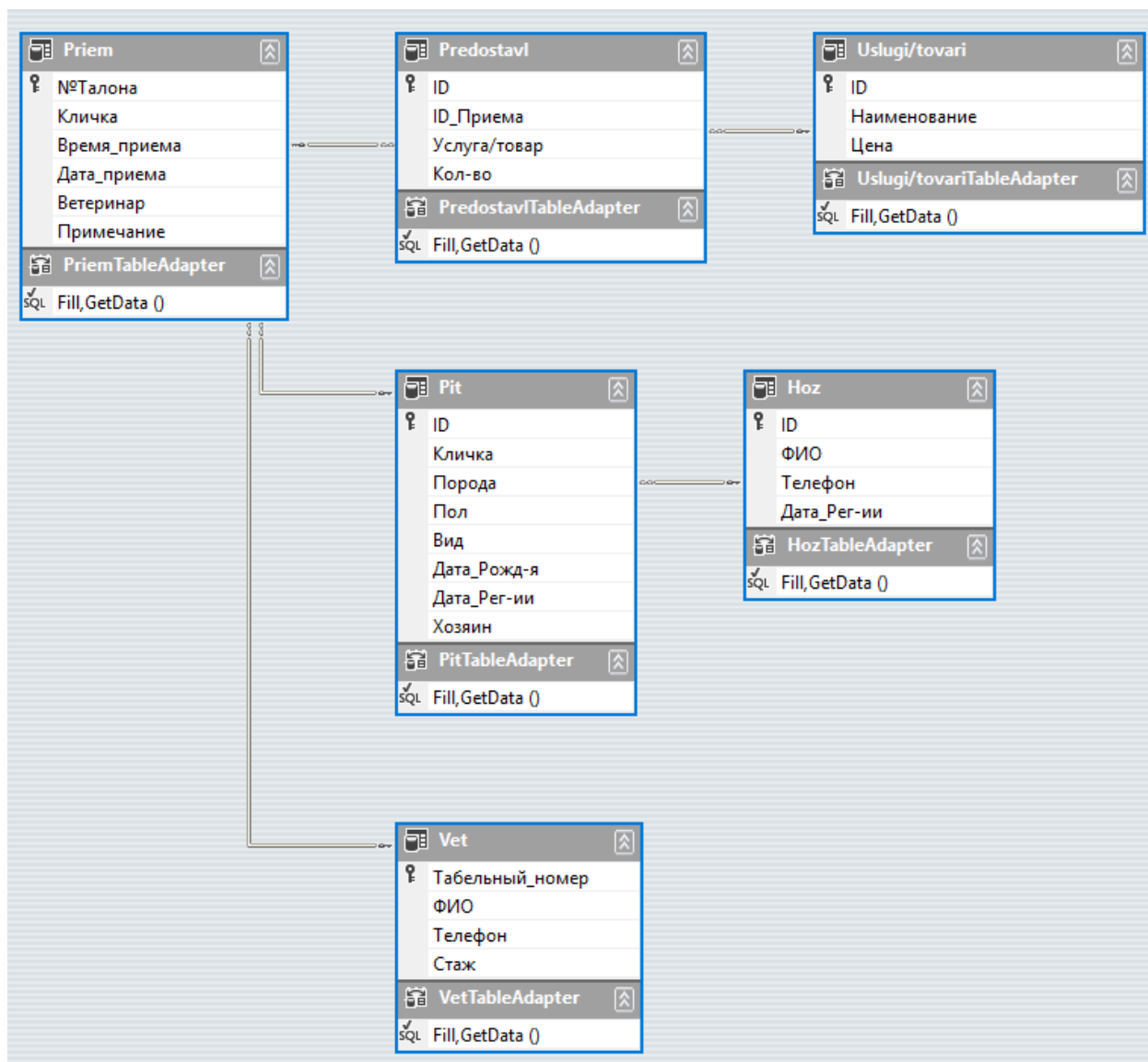


Рисунок 19 – Диаграмма классов на языке UML