**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ** **РОССИЙСКОЙ ФЕДЕРАЦИИ** **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»** **ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК**

**КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ**

**КУРСОВАЯ РАБОТА**

**по дисциплине «Информатика и программирование»**

**«Разработка приложения для автоматизация учета приема пациентов в поликлинике»**

Исполнитель:  
студентка группы 185-3  
Кенжетаева А.М.

Руководитель:  
старший преподаватель  
Павлова Е.А.

**Тюмень – 2019**

Оглавление

[Введение 3](#_Toc11821287)

[1. Постановка задач 4](#_Toc11821288)

[2. Описание алгоритмов и классов, методов 5](#_Toc11821289)

[3. Описание приложения 7](#_Toc11821290)

[4. Описание базы данных Access 12](#_Toc11821291)

[Заключение 16](#_Toc11821292)

[Список литературы 17](#_Toc11821293)

Введение

Одной из актуальных проблем любого вида человеческой деятельности является проблема обработки увеличивающегося потока информации.

Перспективным способом решения данной проблемы является автоматизация работы с информацией, в частности, создание компьютерных баз данных, позволяющих хранить, систематизировать и обрабатывать данные.

1. Постановка задач

Предметной областью базы данных является деятельность поликлиники.

Входная информация для БД: сведения о врачах (ФИО, специализация, кабинет, время работы), о пациентах (ФИО, дата рождения, номер полиса), а также сведения о приемах (дата, время, статус).

Целью создания базы данных является автоматизация учета информации о приеме пациентов.

Назначение проектируемой базы:

* хранение информации о врачах поликлиники;
* хранение информации о пациентах поликлиники;
* обновление и добавление информации;
* анализ информации по выборке (врачи, дата, статус);
* выдача итоговой информации для отчетов.

2. Описание алгоритмов и классов, методов

При разработке программы использовался язык C# для платформы .NET Framework 4.7.2 и интегрированная среда разработки Microsoft Visual Studio 2019.

Программа содержит несколько форм:

* Form1 – главная страница приложения, предназначенная для выбора необходимой базы;
* Doctors – форма для просмотра и редактирования базы данных врачей;
* Patients – форма для просмотра и редактирования базы данных пациентов;
* Reception - форма для просмотра и редактирования базы данных приемов;
* Docpat – форма с «подчиненной» для просмотра приемов определенного врача;
* InfoAll – форма для отображения обобщенной информации.

Алгоритмы:

* Алгоритм каскадного удаления (идентичен для форм «Врачи» и «Пациенты»).

Рассмотрим на примере «Врачи»:

При сохранении данных по врачам происходит проверка соответствия колонки «ФИО врача» из формы «Receptions» и колонки «ФИО» из «Doctors». Если врач не найден, прием удаляется из текстового документа, связанного с формой «Приемы».

* Алгоритм выборки для формы «InfoAll»:

На форме находятся 3 checkbox’а «По врачам», «По статусу», «По дате». При их выборе свойство Visible для всех строк DataGridView принимает положительное значение. Из соответствующего checkbox’а считывается параметр выбора и сравнивается с определенным столбцом. Если соответствие не установлено свойство Visible приравнивается к значению false – строка не отображается. Если выбрано несколько параметров выборки, то она выполняется поочередно: по врачам, по статусу, по дате.

Основные методы:

Таблица 1. Методы

|  |  |
| --- | --- |
| Метод | Предназначение |
| Для форм «Врачи», «Пациенты», «Приемы» | |
| private void Button7\_Click(object sender, EventArgs e) | Добавление нового врача в DataGridView с последующим сохранением в текстовый документ |
| private void Button4\_Click(object sender, EventArgs e) | Изменение данных по выделенному врачу в DataGridView с последующим сохранением в текстовый документ |
| private void DocSave() | Считывание построчно данных из DataGridView и сохранение в документ |
| private void DocShow() | Метод, обратный private void DocSave(). Считывание построчно данных их текстового документы и отображение в DataGridView |
| Для формы «Приемы» и «Обобщенная информация» | |
| private void ComboDoc() | Создание combobox’а для поиска врачей |
| private void ComboPat() | Создание combobox’а для поиска пациентов |

Все текстовые файлы представлены в формате txt.

* Doc – файл содержит информацию по врачам;
* Patients – файл содержит информацию по пациентам;
* Receptions – файл содержит информацию по приемам. На данные файла влияют изменения в файлах Doc и Patients.

В качестве разделителя для наглядности и удобства программного считывания используется знак решетка (#).

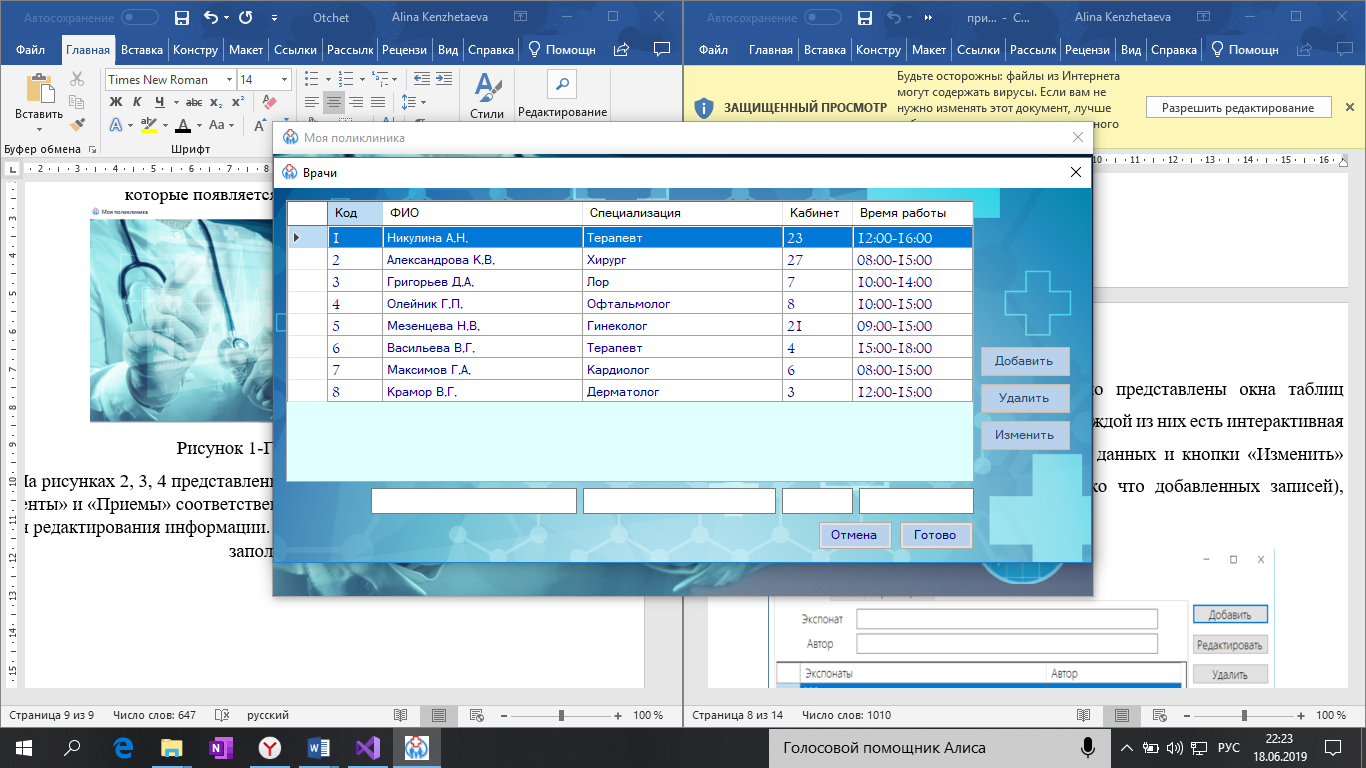
3. Описание приложения

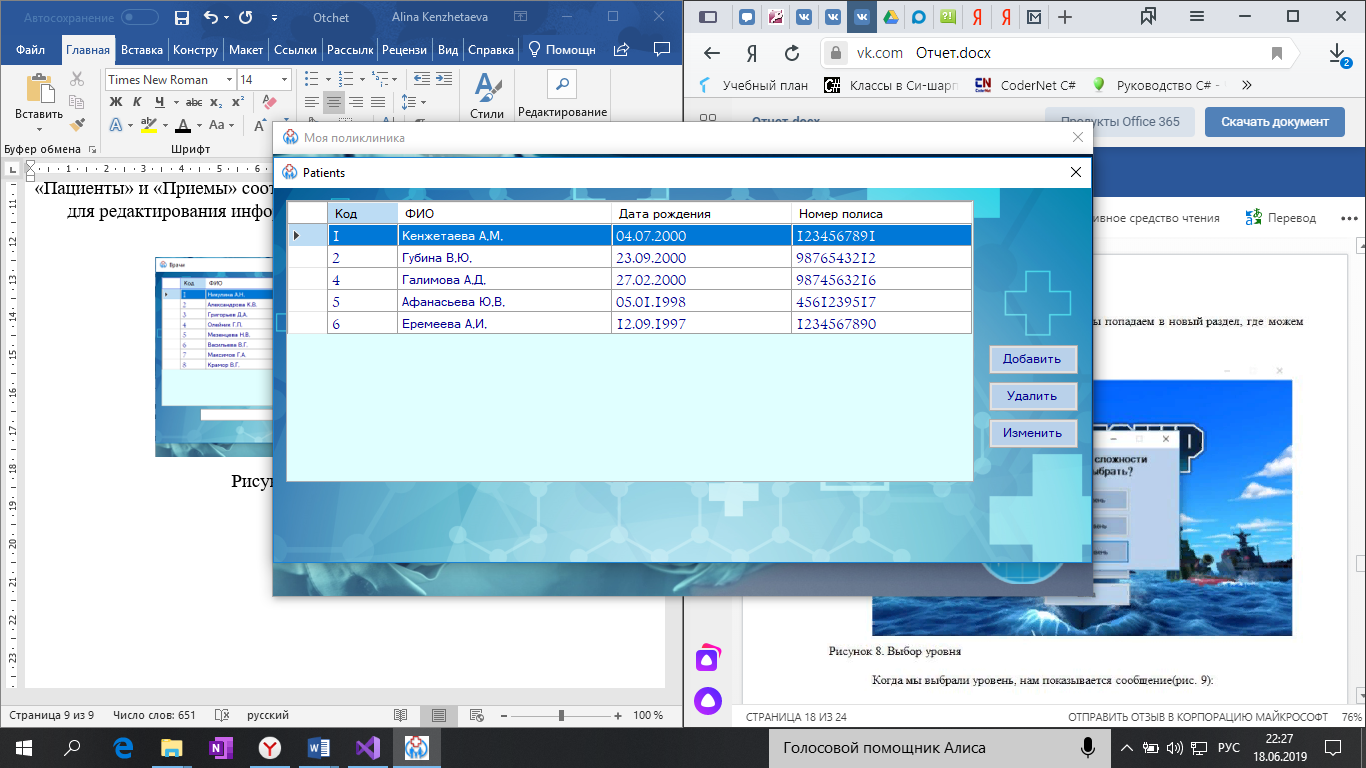
При запуске программы пользователю открывается главное меню, из которого он может перейти в другие разделы при помощи кнопок, при наведении на которые появляются подсказки (рис. 1).



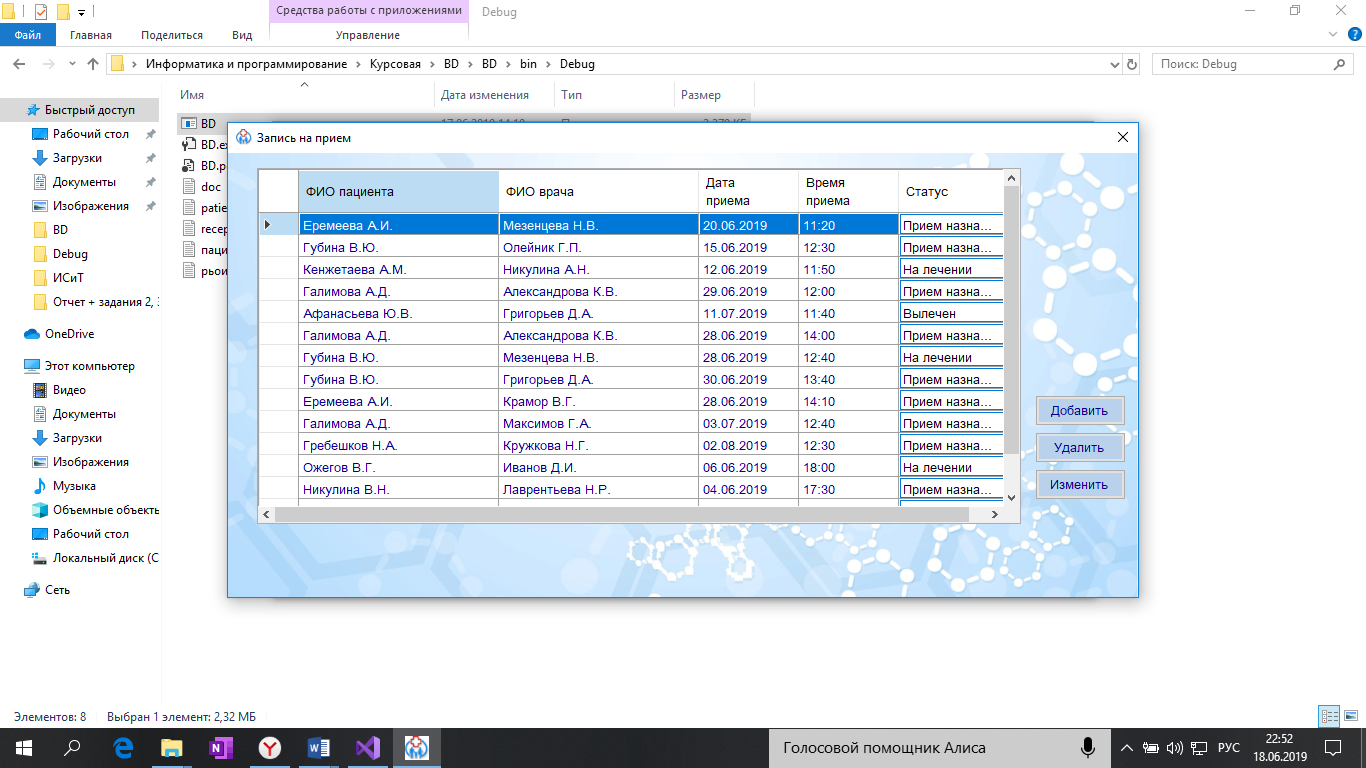
*Рисунок 1-Главное меню*

На рисунках 2, 3, 4 представлены таблицы базы данных «Врачи», «Пациенты» и «Приемы» соответственно.



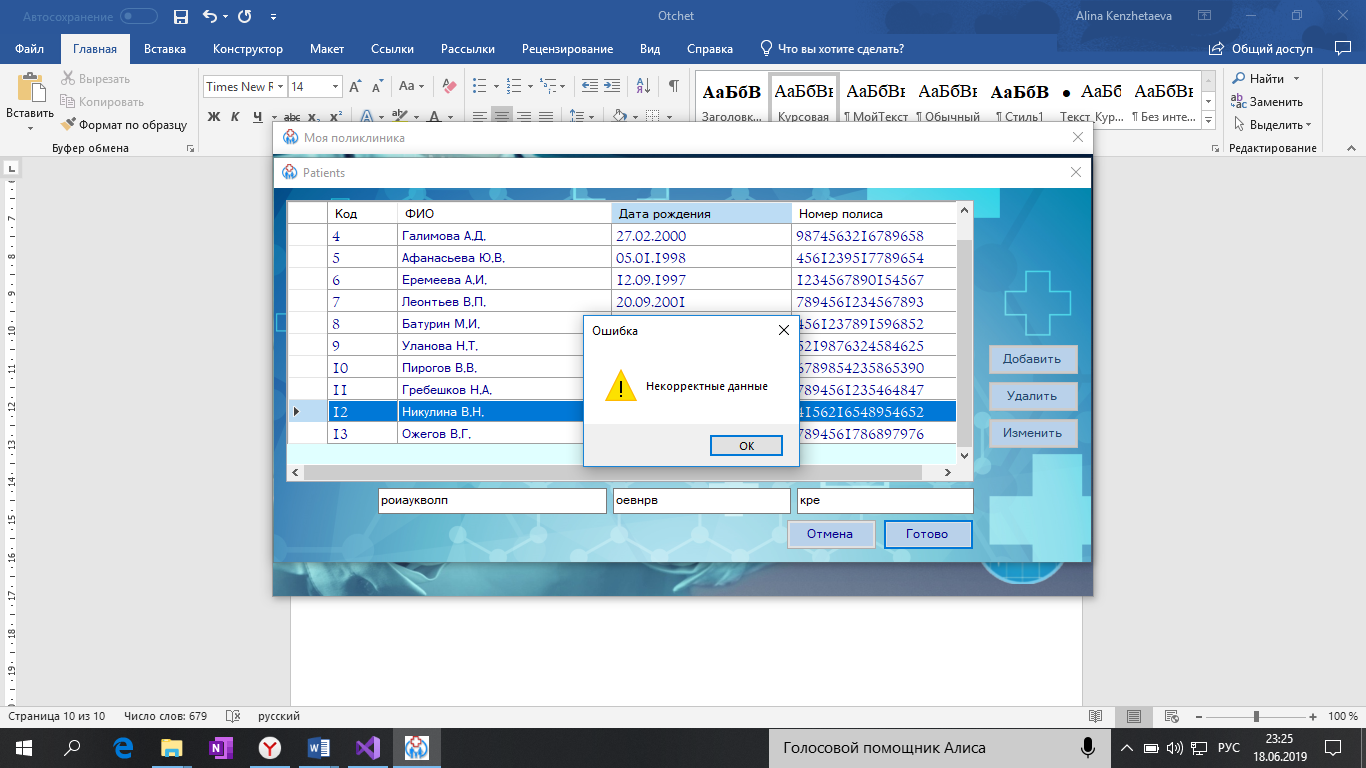
*Рисунок 2 – форма «Врачи»*

*Рисунок 3 – форма «Пациенты»*



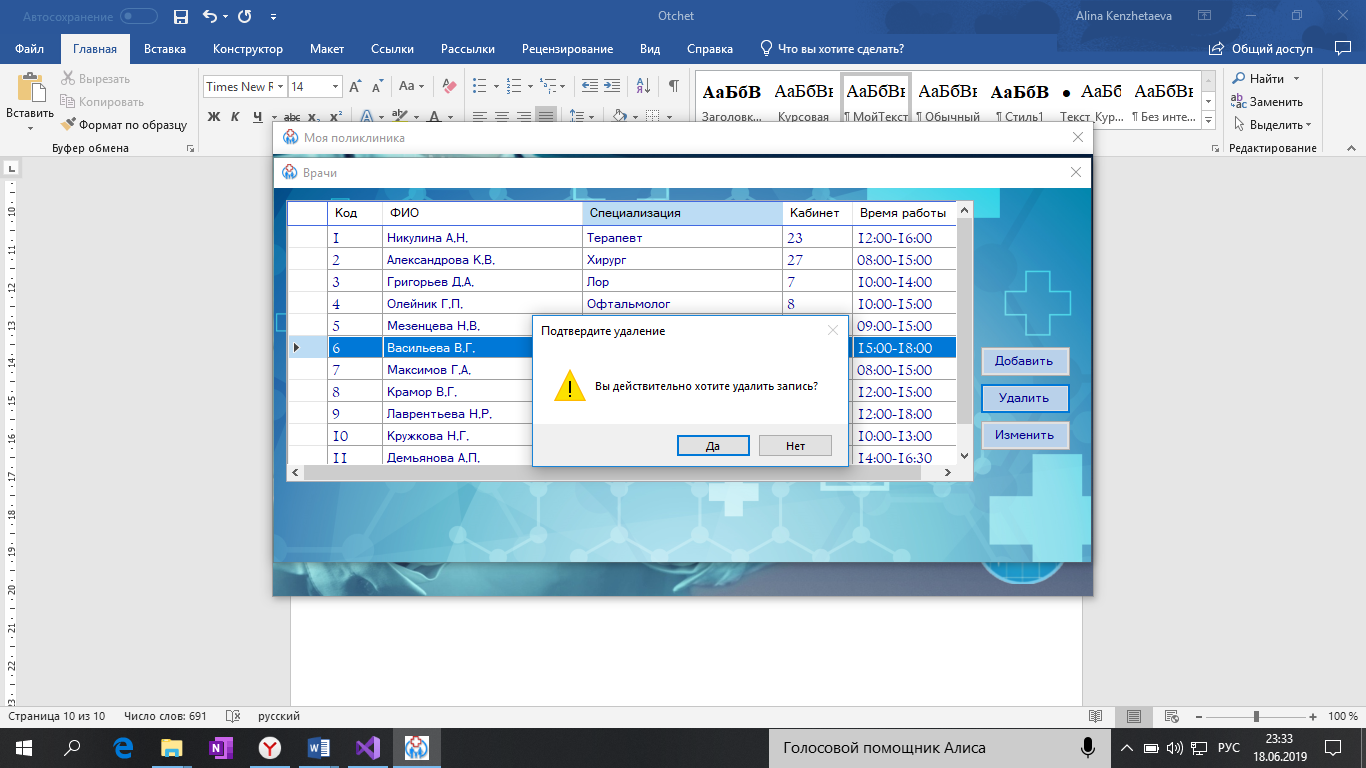
*Рисунок 4 – форма «Приемы»*

На каждой форме имеются кнопки для редактирования информации. При нажатии «Добавить» и «Изменить» появляются поля для заполнения и редактирования данных, другие кнопки становятся недоступны. Кнопка «Готово» проверяет введенные данные на корректность. При несоответствии появляется сообщение об ошибке (рис. 5)



*Рисунок 5 – редактирование данных таблицы*

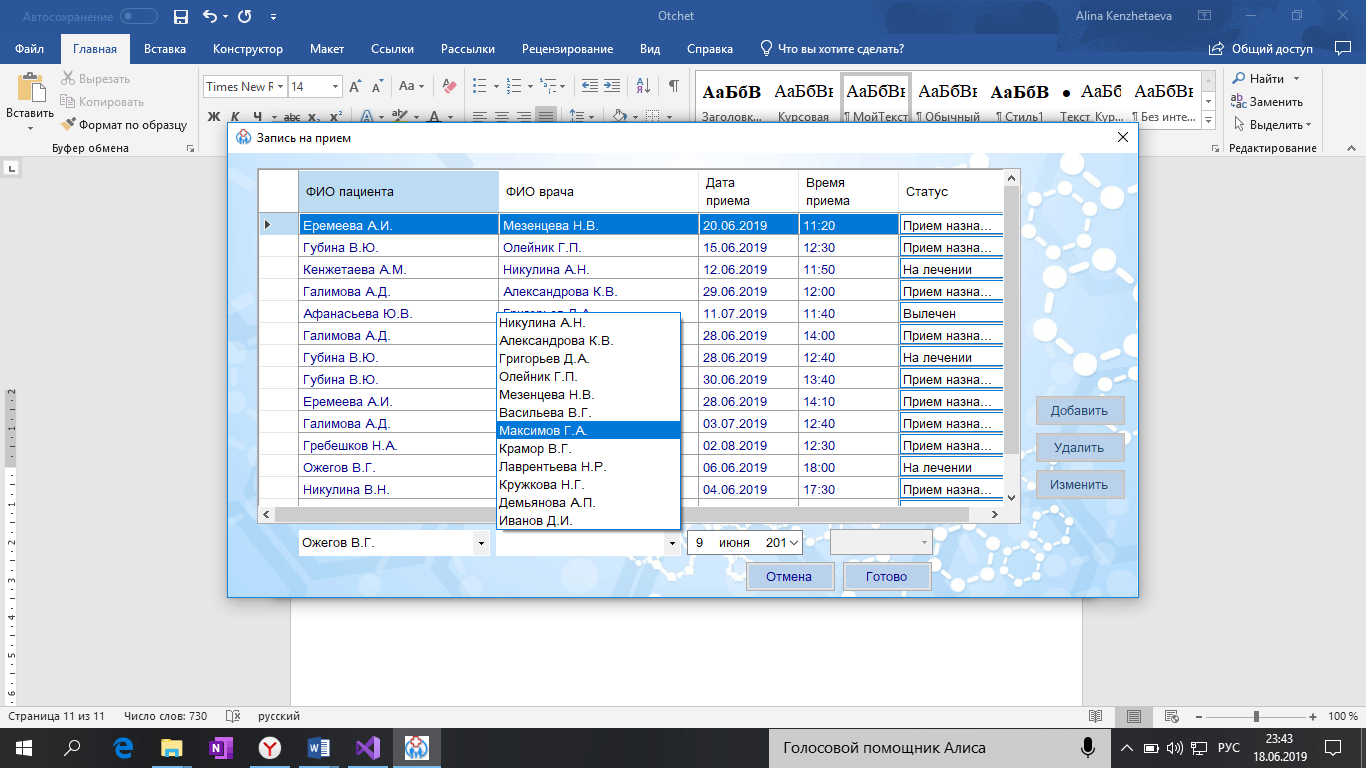
При удалении необходимо подтвердить действие (рис. 6).



*Рисунок 6 – диалоговое окно*

При выходе из форм «Пациенты», «Врачи» появляется сообщение о необходимости сохранения данных.

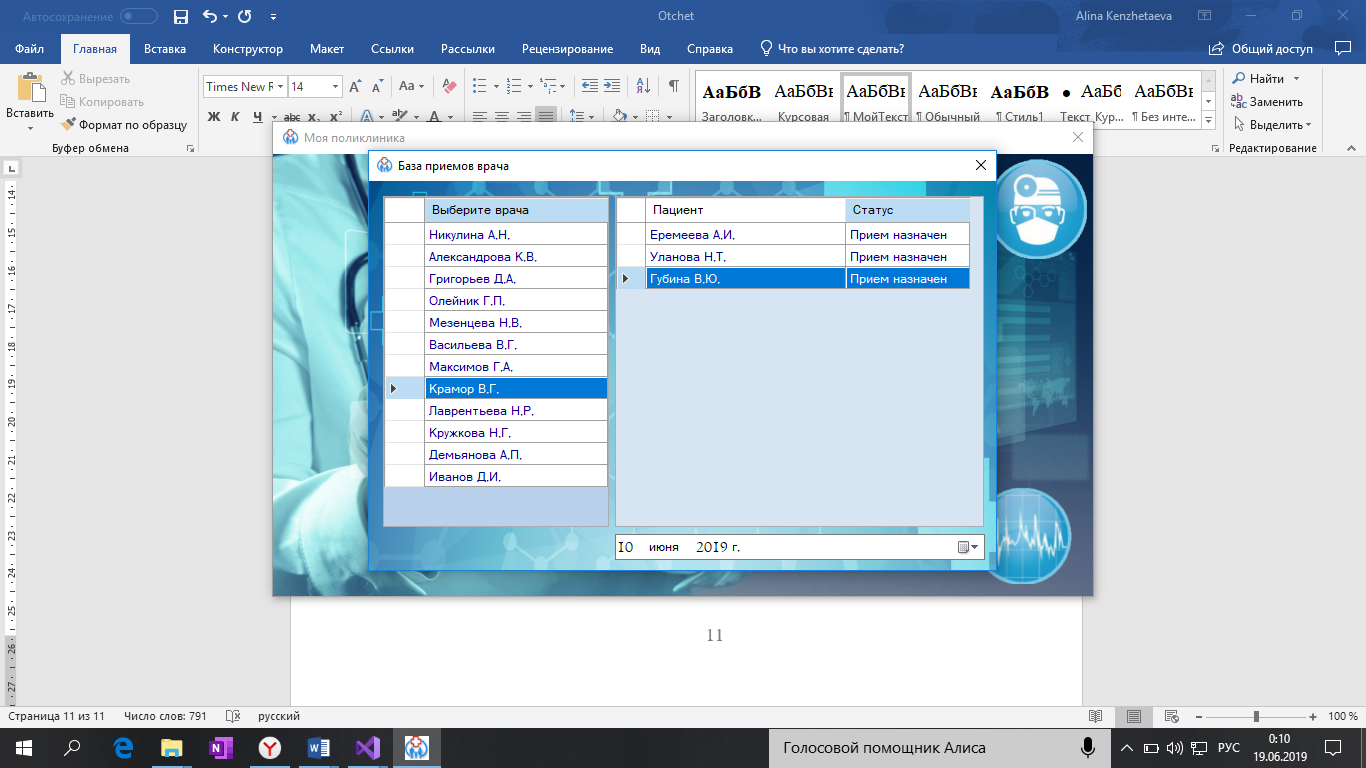
При изменении и добавлении данных на форме «Приемы» вместо текстовых строк появляются выпадающие списки, основанные на соответствующих базах, а также календарь для выбора даты и время приема (рис. 7).



*Рисунок 7 – добавление и изменение данных таблицы из формы «Приемы»*

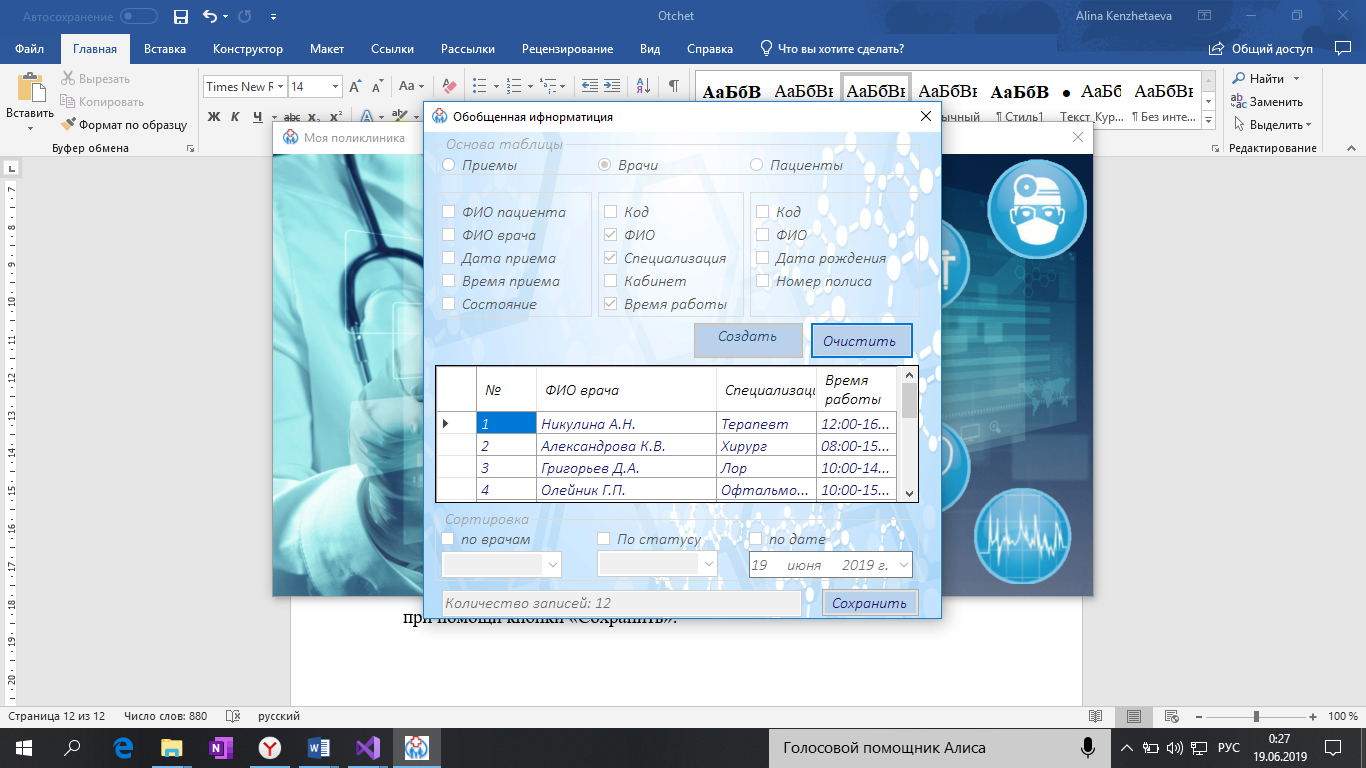
Одним из столбцов таблицы является «Статус», который изменяется прямо в таблице. Он принимает три значения: прием назначен, на лечении, вылечен. При добавлении записи, статус автоматически становится «Прием назначен».

В окне «База приемов врача» (рис. 8) представлены две связные таблицы. При выборе врача из списка в таблице справа отображаются все его пациенты и статус приема. При необходимости можно провести выборку по дате, выбрав необходимое число из календаря, расположенном в нижней части окна.



*Рисунок 8 – форма «База приемов врача»*

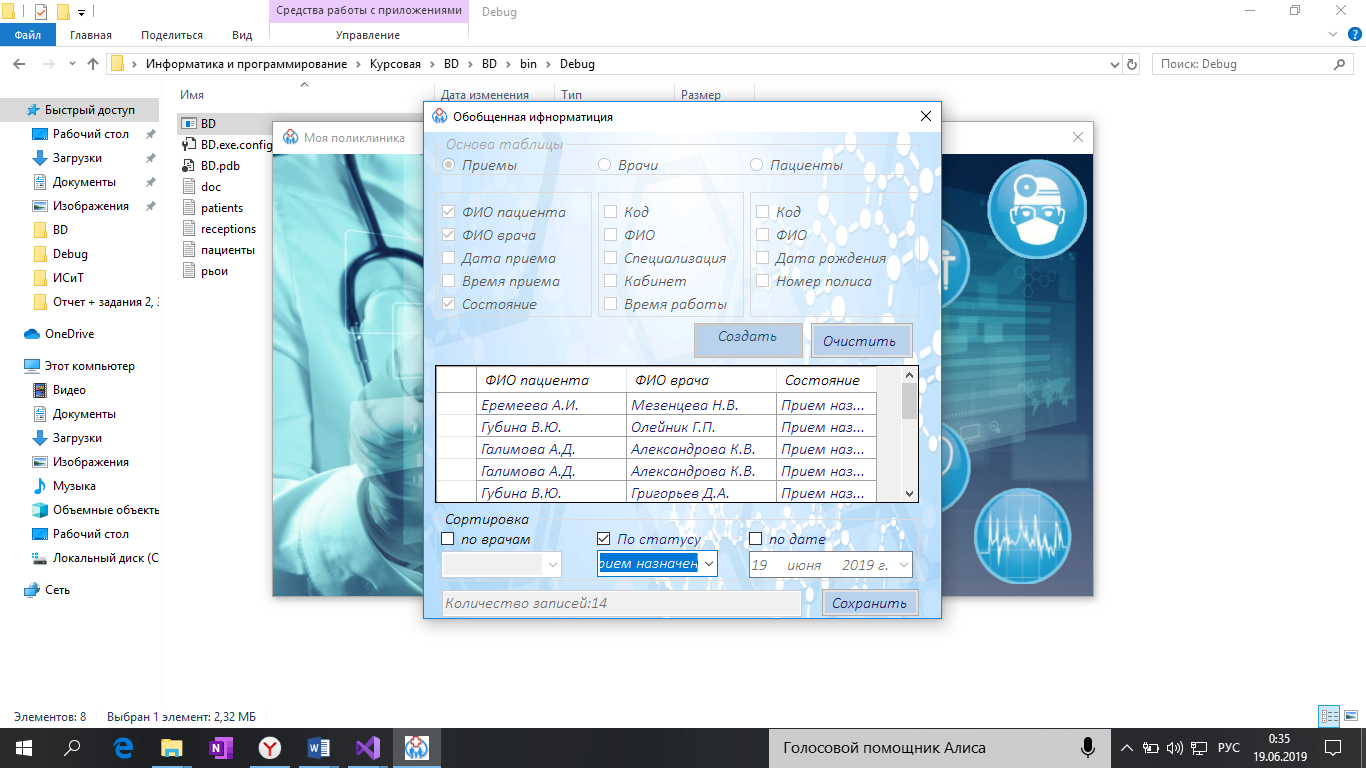
При переходе по кнопке «Для печати» появляется форма для создания текстовых документов. После выбора базы, на основе которой будет создаваться документ, становятся доступны соответствующие checkbox’ы, благодаря которым можно выбрать интересующую информацию. При нажатии кнопки «Создать», появляется таблица, в нижней части окна отображается количество записей в данной таблице (рис. 9).



*Рисунок 9 -форма «Обобщенная информация/для печати»*

Далее данные из этой таблице можно сохранить в виде текстового документа при помощи кнопки «Сохранить». Появляется окно, в котором необходимо вписать название будущего текстового документа. После ввода документ сохраняется в формате txt.

При создании таблицы на основе «Приемы» становится доступна сортировка по трем критериям: врачи, статус, дата. При их выборе таблица обновляется. (рис. 10)



*Рисунок 10 – сортировка в форме «Обобщенная информация/для печати»*

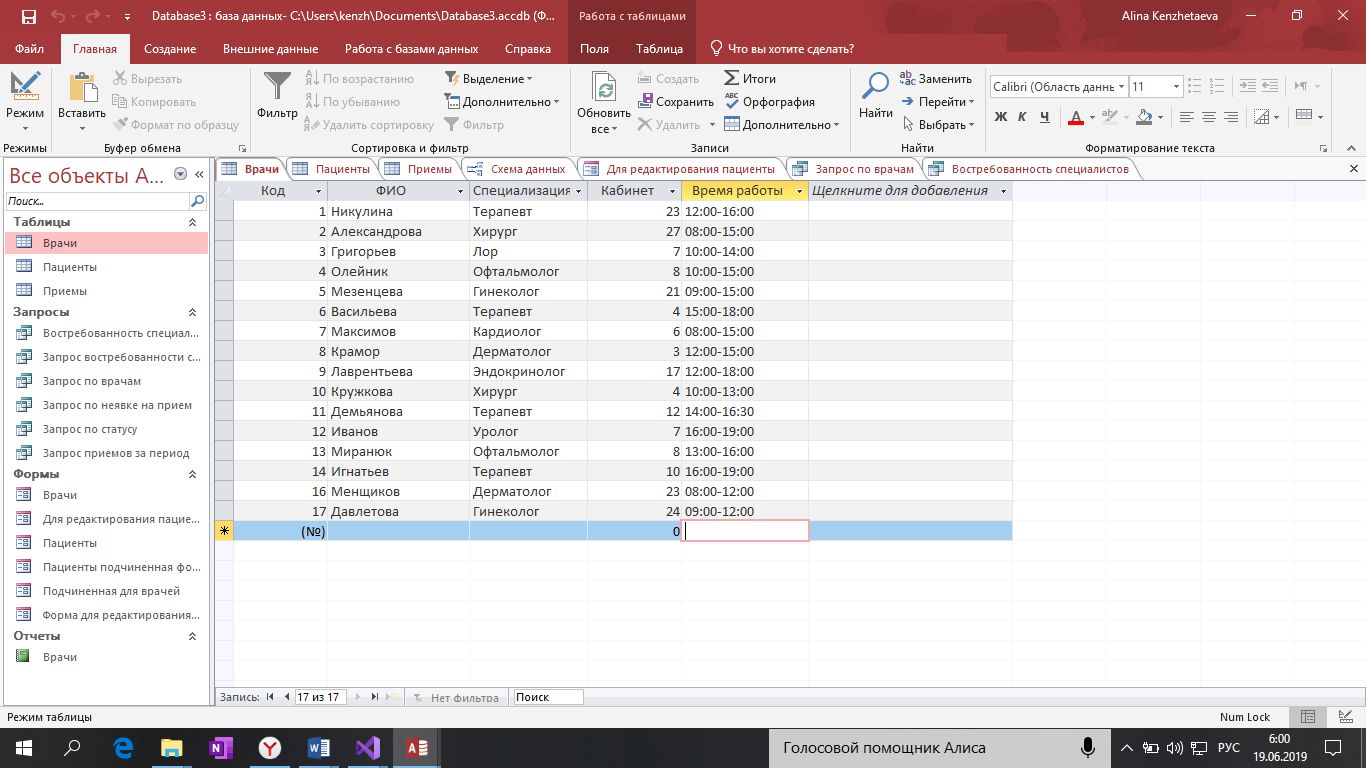
При нажатии на кнопку «Отменить», все данные исчезают.

В программе используется каскадное удаление записей. При удалении одного из элементов таблиц «Врачи», «Пациенты» происходит всех строк таблицы «Приемы» содержащие в себе этот элемент.

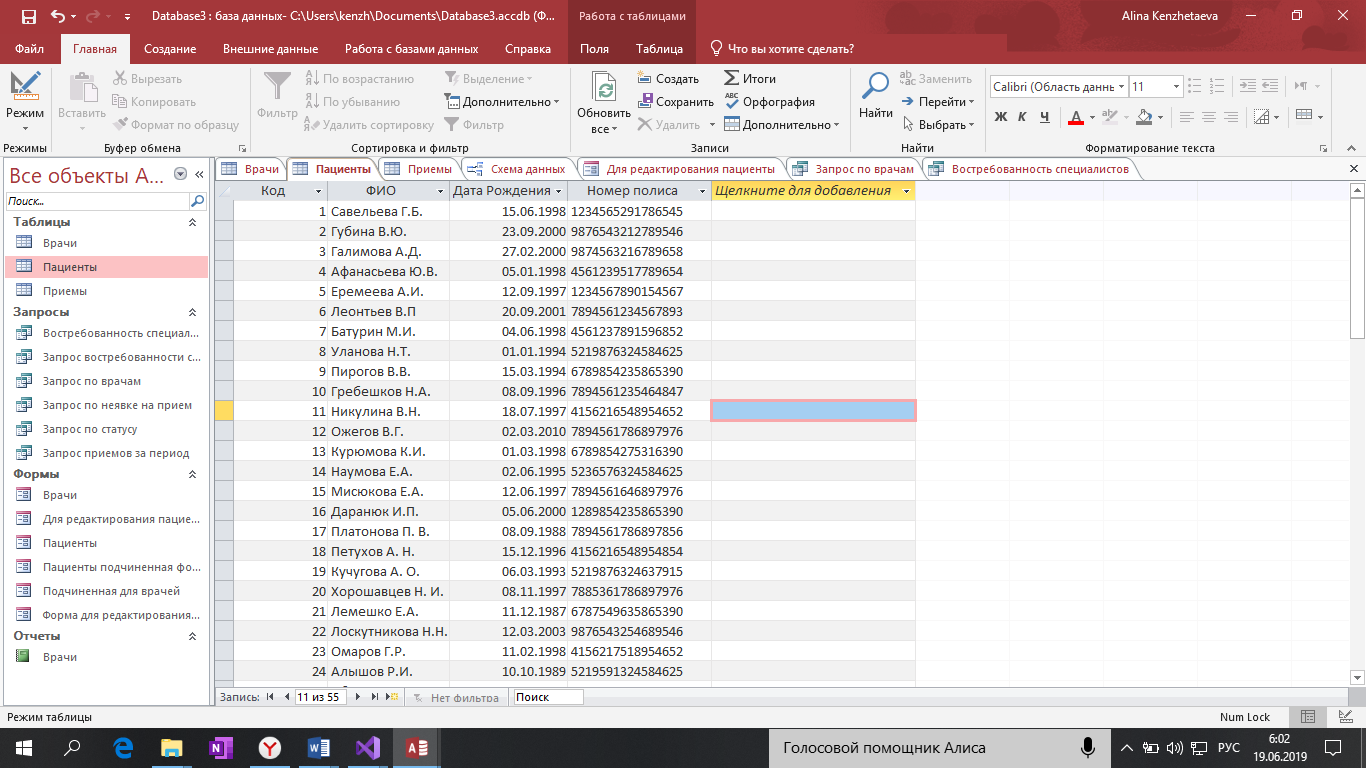
Реализована данная функция путем проверки главной таблицы на содержание удаленных элементов и их последующем удалении в цикле.

4. Описание базы данных Access

База данных представляет из себя 3 таблицы: Врачи, Пациенты, приемы. Для представления масштаба базы добавлен счетчик записей.

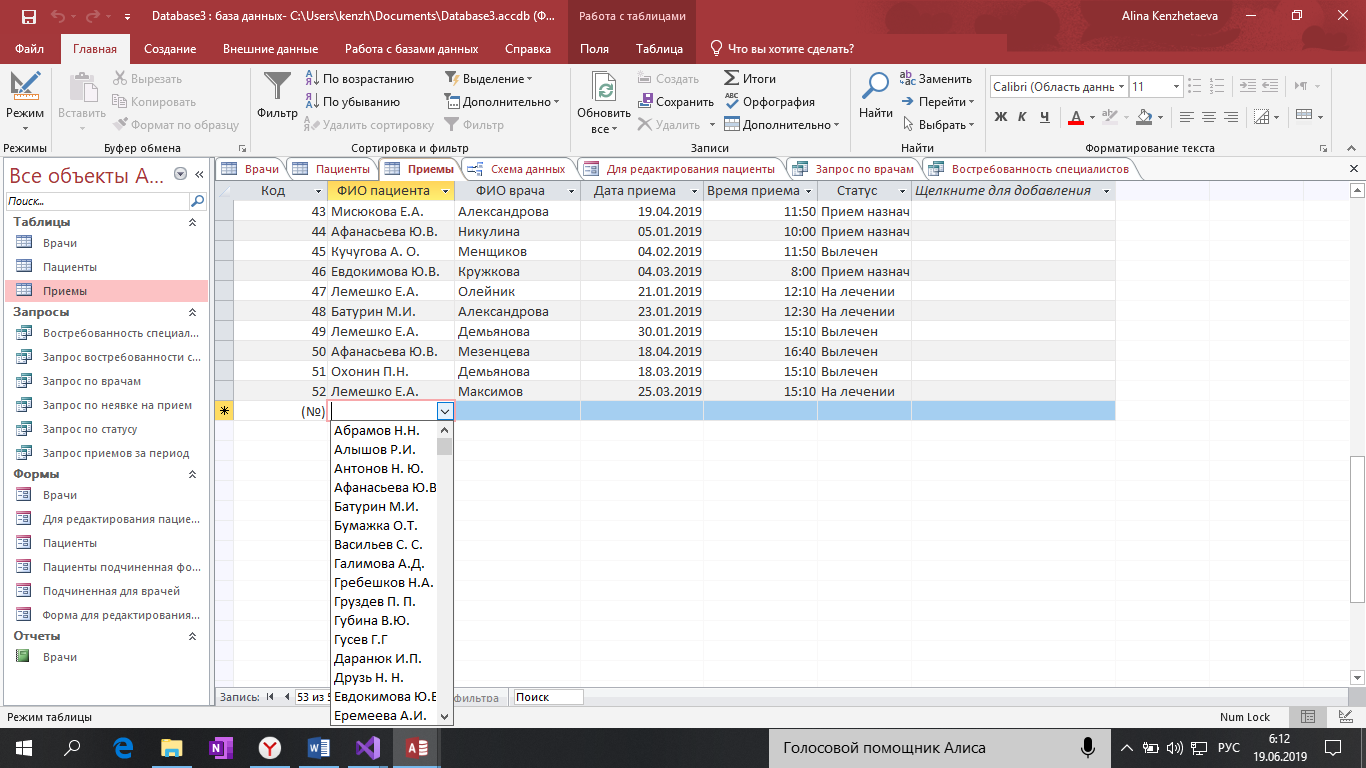


*Рисунок 11 – таблица «Врачи»*



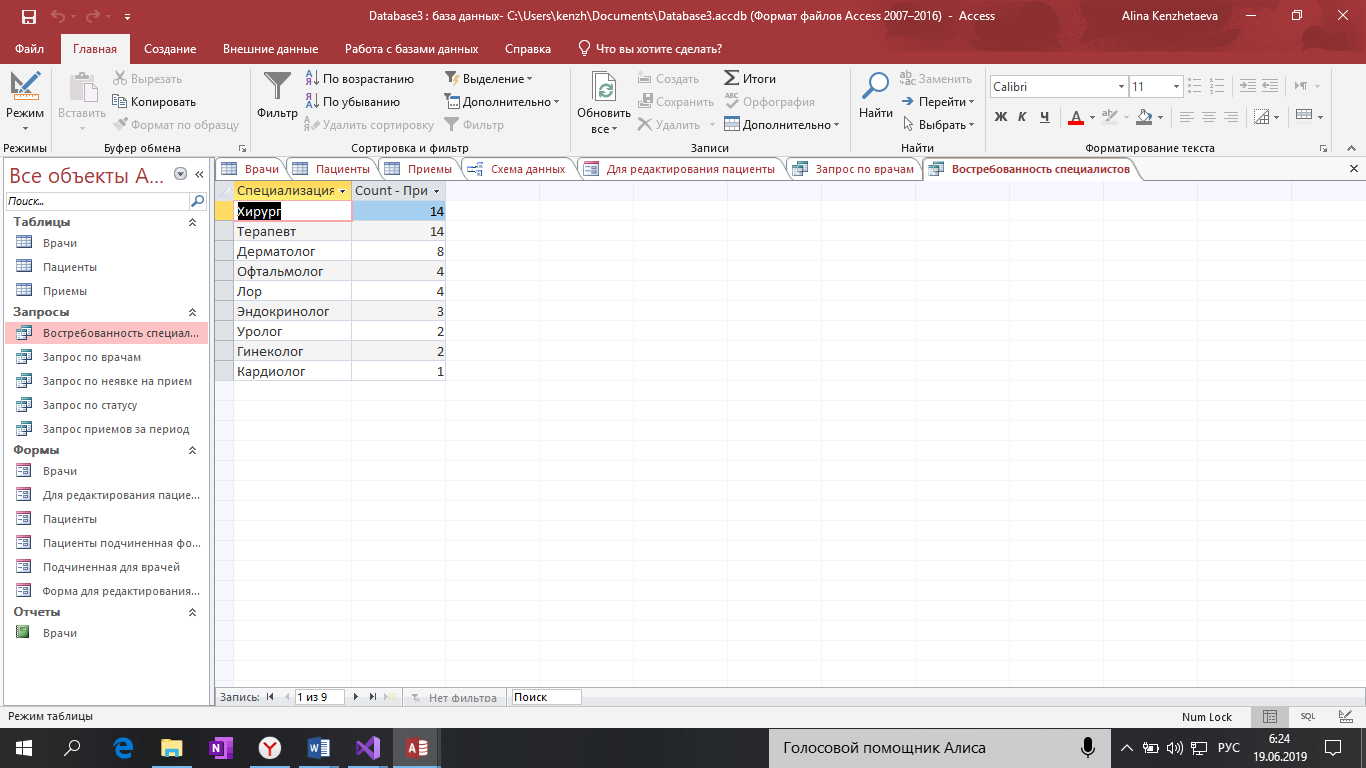
*Рисунок 12 – таблица «Пациенты»*

Таблица «Приемы» содержит информацию о пациенте, враче, дате и времени приема. В данной таблице реализована подстановка данных для первых двух столбцов. Данная функция работает на основе подстановок в Конструкторе таблиц (select Столбец from Таблица), источником у каждого столбца взят его соответствующий столбец в нужной таблице. А также один столбец с элементами заданного списка.



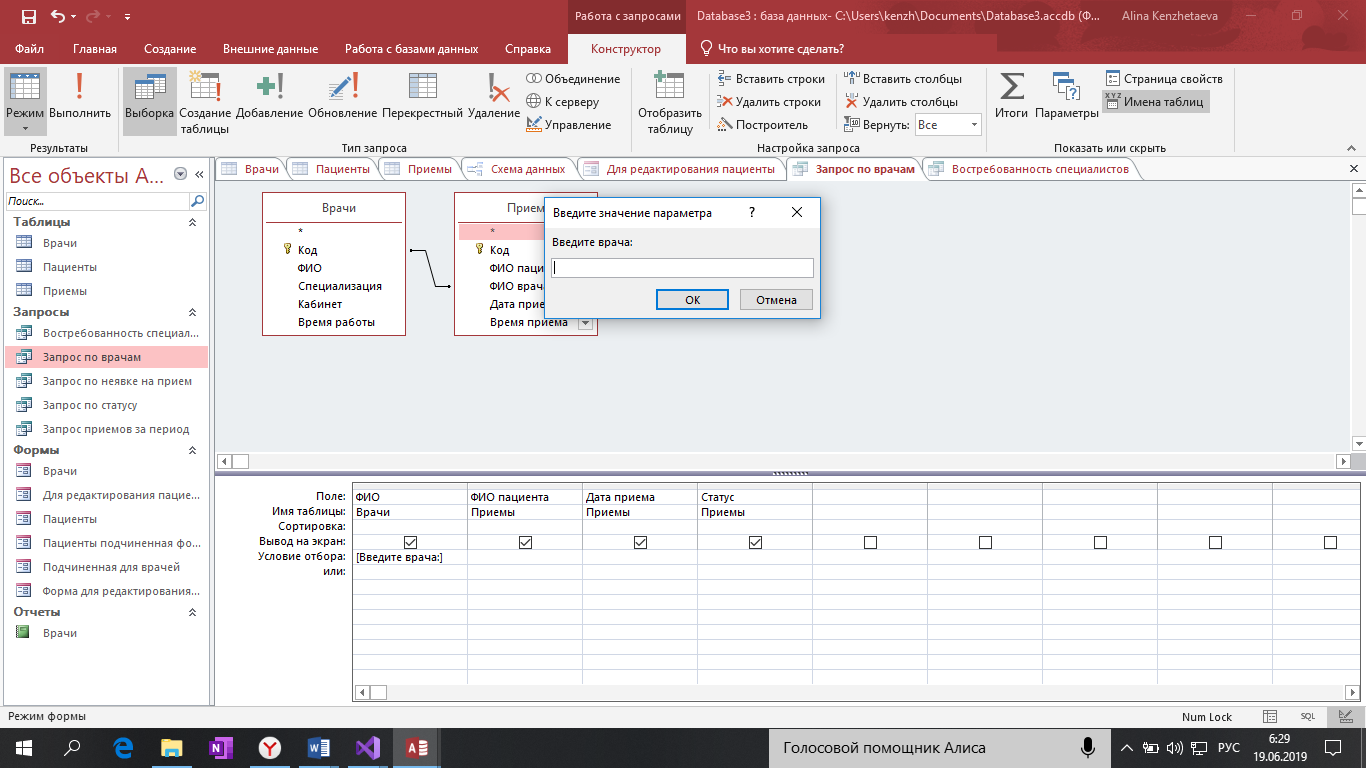
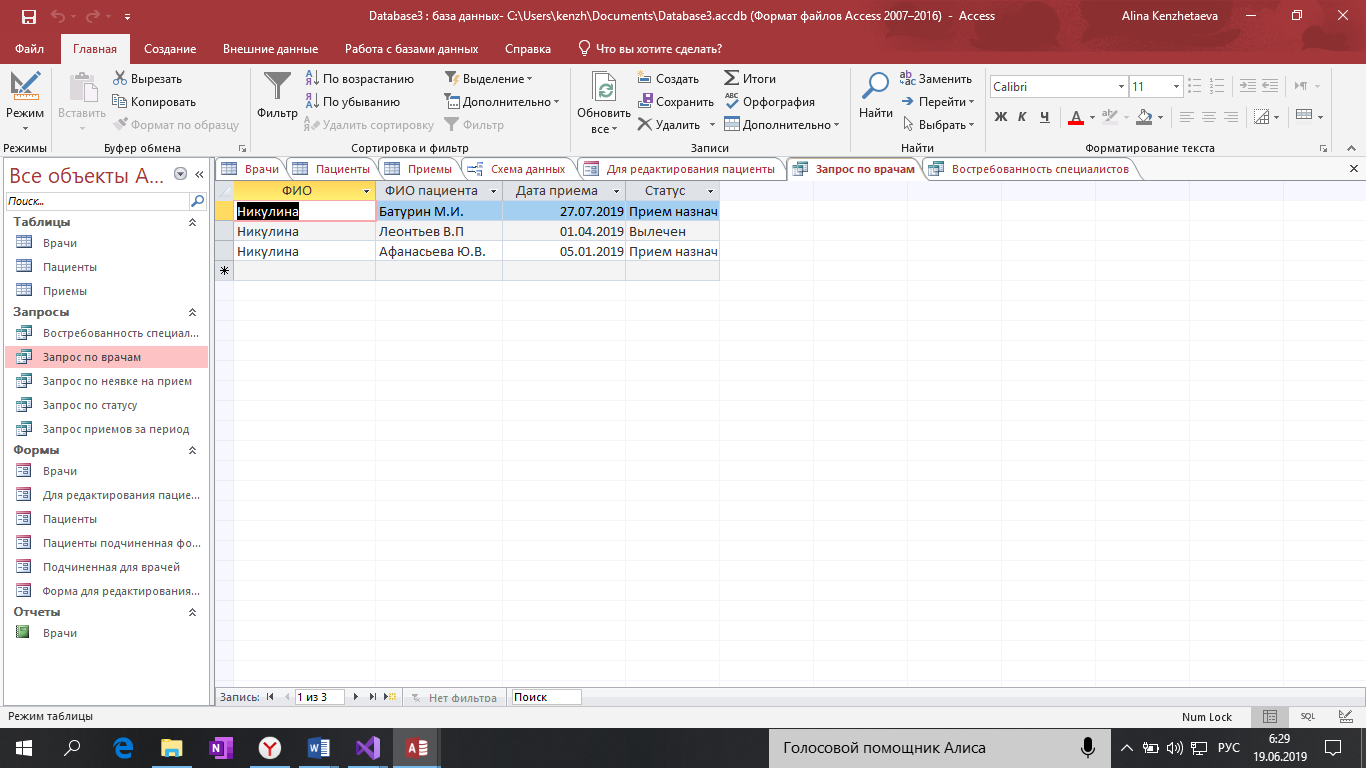
*Рисунок 13 – таблица «Приемы»*

База данных Access также имеет несколько запросов, основанных на всех трех таблицах. На основе таблиц «Врачи» и «Пациенты» был проведен анализ востребованности той или иной специализации. Таблица представлена по убыванию количества записей. (рис. 14).

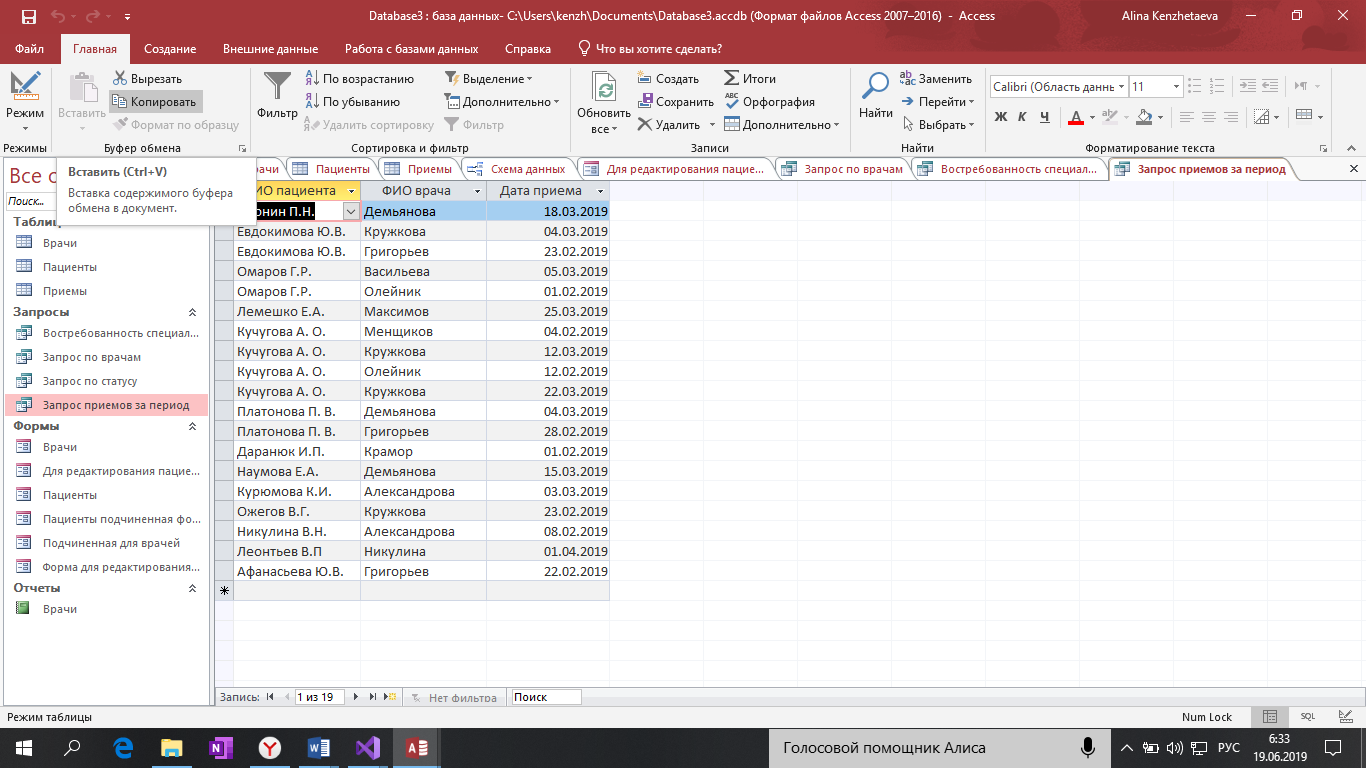


*Рисунок 14 – Востребованность специалистов различной направленности*

Созданы запросы по статусу пациента: записан на прием, находится на лечении, вылечен. При вводе статуса выдаются все записи с данным статусом. Аналогично реализован запрос по врачам. При вводе фамилии появляются все пациенты (рис. 15).

* Рисунок 15 – запрос по врачам*

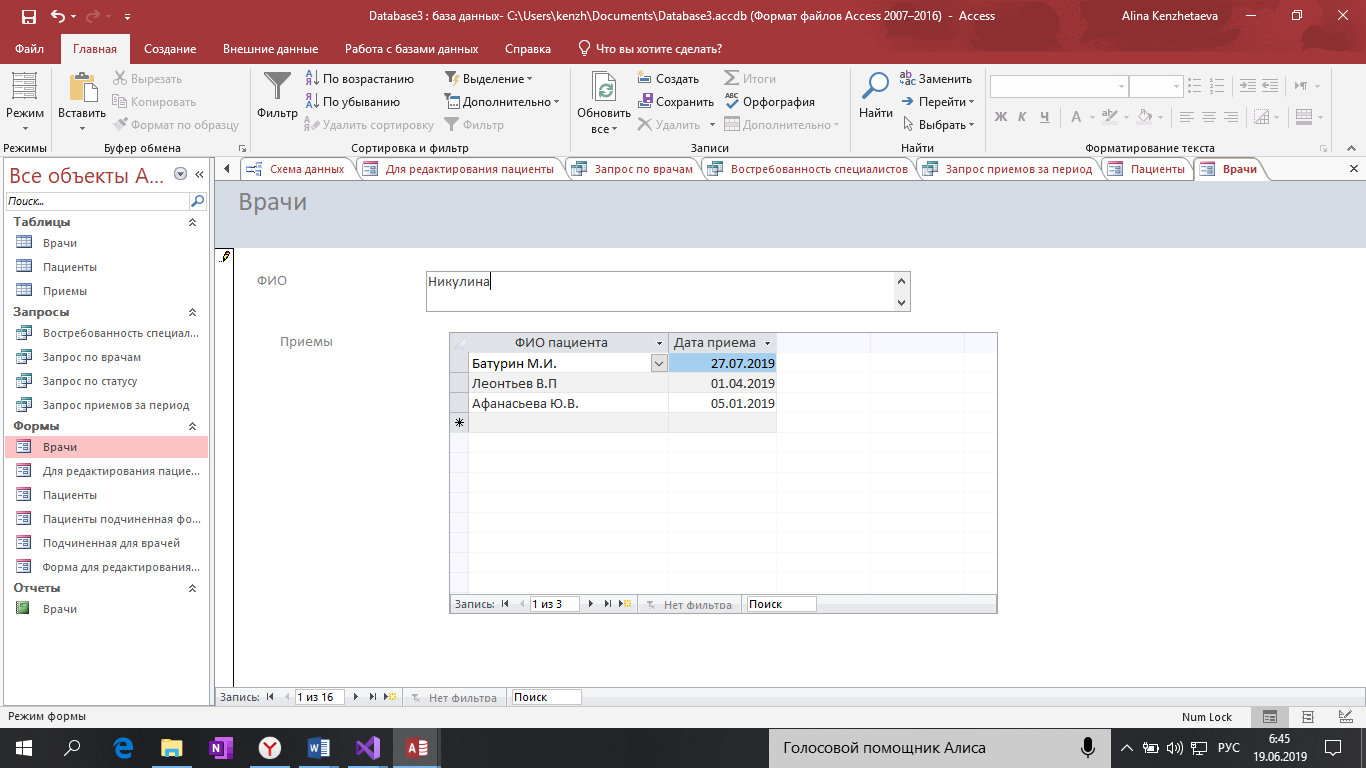
Запрос для отражения приемов за определенный период.



*Рисунок 16 – запрос приемов за период*

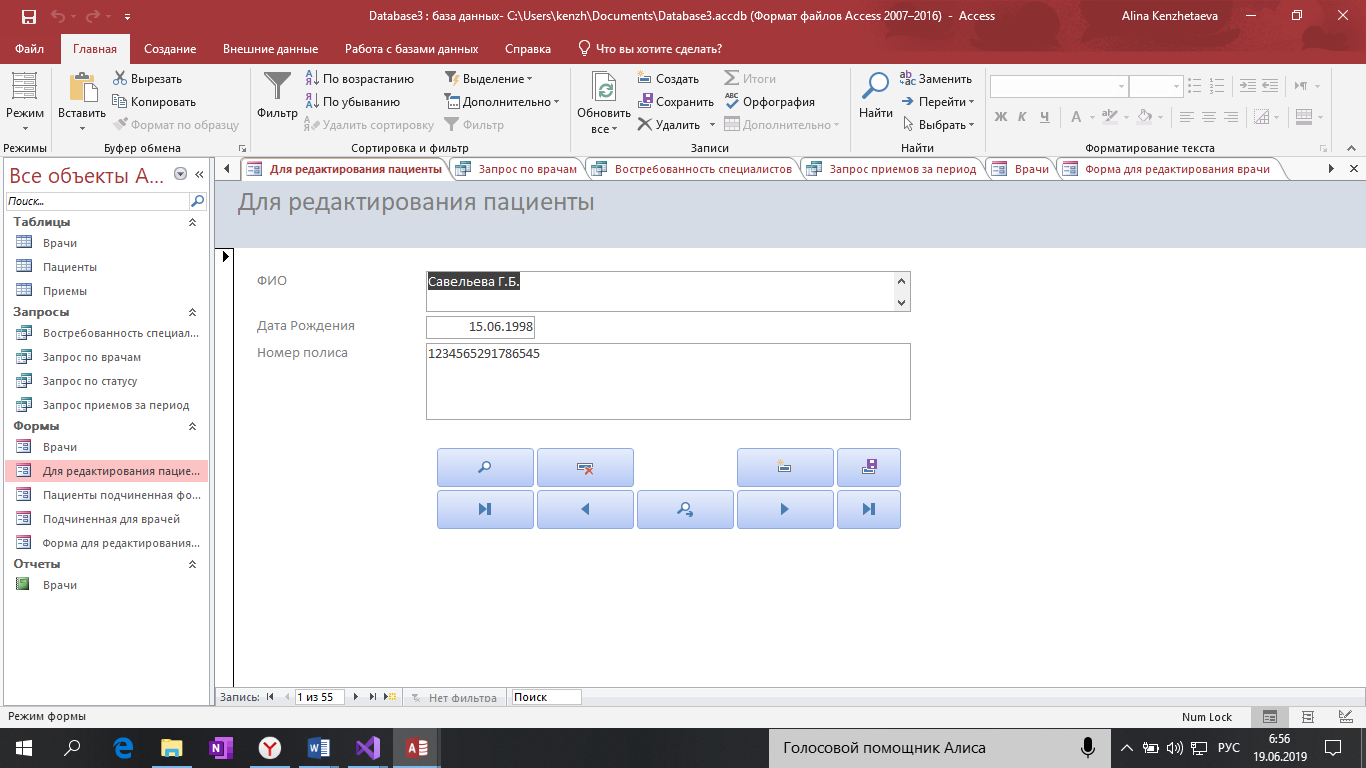
Реализован с помощью функции, введенной в условие отбора: Between [Введите начальную дату:] And [Введите конечную дату:].

В базе также реализованы несколько форм. Форма для просмотра информации содержит в себе подчиненную форму, выдающую значение из таблицы «Приемы» к фамилии врача (рис. 17).



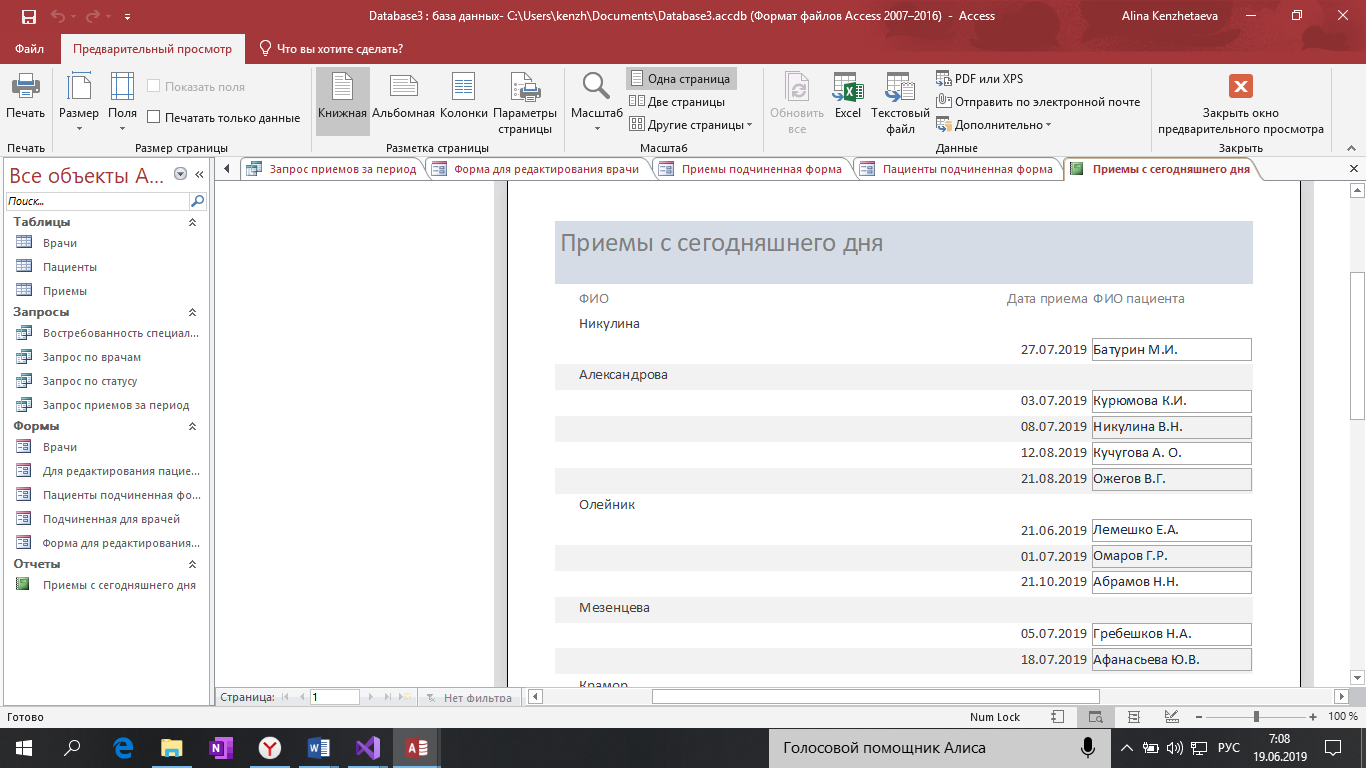
*Рисунок 17 – форма для просмотра информации по врачам.*

Форма для редактирования информации содержит в себе такие элементы   
как – след./пред. элемент, первый/последний элемент, удаление, добавление, сохранение изменений, а также поиск по существующим записям.



*Рисунок 18 – Редактирование информации*

Последний элемент базы данных – отчет. Содержит в себе информацию по приемам врачей, за определенный срок. На примере показаны все приемы у врачей на несколько месяцев вперед.



*Рисунок 19 - отчет*

Заключение

В ходе выполнения работы было написано приложение, который позволяет вести учет о приеме пациентов в поликлинике.

Разработка программного приложения для работы с базами данных позволяет выполнять операции по редактированию и добавлению информации, а также по ее изменению. Данное приложение дает возможность произвести выборку по нескольким критериям: по врачам, по статусу и по дате. Созданную на этой основе информацию можно сохранять в текстовый файл для дальнейшего использования.

При добавлении приема в базу, его статус принимает значение «Прием назначен». Благодаря этому существует возможность следить за динамикой лечения.

Одна из форм приложения разработана для наглядного представления приемов по выбранному врачу.

Также была реализована база данных в Access. Основой выступают БД по врачам и пациентам. База данных «Приемы» получает ключевую информацию из этих таблиц. Созданы несколько запросов: для получения всех пациентов врача, для печати данных из БД, для приемов за определенный период времени. Между таблицами установлена связь «один-ко многим».

В конечном итоге цель – разработка приложения для учета приемов в поликлинике достигнута. Все функции БД реализованы.

Список литературы

1. METANIT.COM - Сайт о программировании URL: <https://metanit.com>

2. Общие сведения о Windows Forms / Microsoft Docs URL: https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/windows-forms-overview.