Оглавление

[Введение 4](#_Toc27496738)

[1 Постановка задачи на проектирование интерфейса 5](#_Toc27496739)

[2 Предметная область 6](#_Toc27496740)

[3 Обоснование проектных решений по видам обеспечения 8](#_Toc27496741)

[3.1 По техническому обеспечению 8](#_Toc27496742)

[3.2 По информационному обеспечению 8](#_Toc27496743)

[3.2.1 Технология организации доступа к данным 12](#_Toc27496744)

[3.2.2 Дерево функций и сценарий диалога 12](#_Toc27496745)

[3.2.3 Диаграмма IDEF 0 14](#_Toc27496746)

[4 Интерфейс пользователя 17](#_Toc27496747)

[Заключение 26](#_Toc27496748)

[Библиографический список 27](#_Toc27496749)

[Приложения 28](#_Toc27496750)

[Приложение А Программный код главной формы 28](#_Toc27496751)

[Приложение Б Программный код формы «Пациенты» 31](#_Toc27496752)

[Приложение В Программный код формы «Отделения клиники» 33](#_Toc27496753)

[Приложение Г Программный код формы «Врачи» 35](#_Toc27496754)

[Приложение Д Программный код формы «Лекарства» 38](#_Toc27496755)

[Приложение Е Программный код формы «Посещения» 40](#_Toc27496756)

[Приложение Ж Программный код формы «Назначения» 43](#_Toc27496757)

[Приложение З Программный код формы «Назначения лечащего врача пациенту» 45](#_Toc27496758)

[Приложение И Программный код формы «Справка клиники» 47](#_Toc27496759)

# Введение

В современном информационном обществе большую роль играют информационные системы - взаимосвязанные совокупности средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели. Информационные системы применяется практически везде, как в промышленной сфере, так и в офисах, и выполняют они особо значимую роль.

Многие системы в той или иной степени связаны с функциями долговременного хранения и обработки информации. Базы данных совмещают в себе все аспекты деятельности какого-либо предприятия в форматизированном виде, пригодном для прикладного использования.

.

# 1 Постановка задачи на проектирование интерфейса

Разрабатываемая информационная система предназначена для содержания информации о ведении документации коммерческой медицинской клиники, для формирования отчета «Назначения лечащего врача пациенту», в зависимости от указанных и выявленных симптомах заболевания, для формирования отчета «Справка клиники», подтверждающий прохождение данным пациентом лечения данного заболевания в данной клинике. По необходимости документы могут быть напечатаны.

Подсистема вывода справочников:

* данные о пациентах;
* данные об отделениях клиники;
* данные о врачах;
* данные о лекарствах;
* данные о посещениях;
* данные о назначениях.

Система должна обеспечить формирование и печать следующих отчетов:

* отчет «Назначения лечащего врача пациенту»;
* отчет «Справка клиники».

# 2 Предметная область

Прежде чем проектировать информационную систему, необходимо проанализировать предметную область, с которой связана дальнейшая работа.

Медицина – система научных знаний и практической деятельности, целями которой являются укрепление и сохранение здоровья, продление жизни людей, предупреждение и лечение болезней человека, а также облегчения страданий от физических и психических недугов.

Медицина делится на теоретическую, доказательную и практическую:

* Теоретическая, или академическая, медицина изучает норму и патологию тела человека теоретически, оперируя уже известными фактами, несёт учебную, просветительскую и научную функции.
* Доказательная, или экспериментальная, медицина выдвигает гипотезы относительно тех или иных состояний, препаратов, подходов, методов лечения, и проверяет их на практике путём двойных слепых плацебо-контролируемых и других клинических исследований.
* Практическая, или клиническая, медицина использует знания, полученные доказательной медициной и оформленные теоретической медициной, в клинической практике для диагностики, лечения, профилактики и других медицинских процедур.

Здоровье – состояние физического, психического и социального благополучия человека, при котором отсутствуют заболевания, а также расстройства функций органов и систем организма.

Медицинская помощь - комплекс мероприятий, направленных на поддержание и (или) восстановление здоровья и включающих в себя предоставление медицинских услуг.

Медицинская услуга - медицинское вмешательство или комплекс медицинских вмешательств, направленных на профилактику, диагностику и лечение заболеваний, медицинскую реабилитацию и имеющих самостоятельное законченное значение;

Лечение - комплекс медицинских вмешательств, выполняемых по назначению медицинского работника, целью которых является устранение или облегчение проявлений заболевания или заболеваний либо состояний пациента, восстановление или улучшение его здоровья, трудоспособности и качества жизни.

Пациент - физическое лицо, которому оказывается медицинская помощь или которое обратилось за оказанием медицинской помощи независимо от наличия у него заболевания и от его состояния.

Медицинская организация - юридическое лицо независимо от организационно-правовой формы, осуществляющее в качестве основного (уставного) вида деятельности медицинскую деятельность на основании лицензии, выданной в порядке, установленном законодательством Российской Федерации о лицензировании отдельных видов деятельности. Положения настоящего Федерального закона, регулирующие деятельность медицинских организаций, распространяются на иные юридические лица независимо от организационно-правовой формы, осуществляющие наряду с основной (уставной) деятельностью медицинскую деятельность, и применяются к таким организациям в части, касающейся медицинской деятельности. В целях настоящего Федерального закона к медицинским организациям приравниваются индивидуальные предприниматели, осуществляющие медицинскую деятельность.

Медицинский работник - физическое лицо, которое имеет медицинское или иное образование, работает в медицинской организации и в трудовые обязанности которого входит осуществление медицинской деятельности; физическое лицо, которое является индивидуальным предпринимателем, непосредственно осуществляющим медицинскую деятельность.

Лечащий врач - врач, на которого возложены функции по организации и непосредственному оказанию пациенту медицинской помощи в период наблюдения за ним и его лечения.

Заболевание - возникающее в связи с воздействием патогенных факторов нарушение деятельности организма, работоспособности, способности адаптироваться к изменяющимся условиям внешней и внутренней среды при одновременном изменении защитно-компенсаторных и защитно-приспособительных реакций и механизмов организма.

Клинические рекомендации - документы, содержащие основанную на научных доказательствах структурированную информацию по вопросам профилактики, диагностики, лечения и реабилитации, в том числе протоколы ведения (протоколы лечения) пациента, варианты медицинского вмешательства и описание последовательности действий медицинского работника с учетом течения заболевания, наличия иных факторов, влияющих на результаты оказания медицинской помощи.

# 3 Обоснование проектных решений по видам обеспечения

## 3.1 По техническому обеспечению

Техническое обеспечение – это комплекс технических средств, предназначенных для обеспечения работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

Комплекс технических средств составляют:

* вычислительные машины или компьютеры (рабочие станции, персональные компьютеры, серверы), являющиеся центральным звеном системы обработки данных;
* периферийные технические средства, обеспечивающие ввод и вывод информации;
* сетевые коммуникации (компьютерные сети и телекоммуникационное оборудование) для передачи данных;
* средства оргтехники и связи.

Компьютеры предназначены для реализации комплексных технологий обработки и хранения информации и являются базой интеграции всех современных технических средств обеспечения управления информационными ресурсами.

Периферийные устройства позволяют вводить информацию в компьютер или выводить её из него.

Сетевые коммуникации предназначены, в основном, для реализации технологий передачи информации и предполагают как автономное функционирование, так и функционирование в комплексе со средствами компьютерной техники.

Организационная техника предназначена для реализации технологий хранения, представления и использования информации, а также для выполнения различных вспомогательных операций в рамках тех или иных технологий информационной поддержки управленческой деятельности.

Исходя из имеющегося оборудования (Ноутбук на базе AMD A4-9120 RADEON R3, 4 COMPUTE CORES 2C+2G 2.20 CHz, жесткий диск 500 Gb, оперативная память 4 Gb, встроенная видеокарта AMD Radeon™ 520, принтер лазерный HP LaserJet Pro 200 color MFP m276n) делаем вывод о достаточности. Поскольку дополнительного приобретения какой-либо техники не требуется, это позволяет экономить финансовые ресурсы.

## 3.2 По информационному обеспечению

Информационное обеспечение – важнейший элемент информационной системы, который сочетает в себе единую систему классификации и кодирования информации, унифицированные системы документации, схемы информационных потоков в организации, а также методику построения БД.

В состав информационного обеспечения включаются два комплекса: компоненты внемашинного информационного обеспечения (документы, справочники) и внутримашинного информационного обеспечения (макеты/экранные формы для ввода первичных данных в ЭВМ или вывода результатной информации, структура информационной базы: входных, выходных файлов, базы данных).

В качестве входных данных используются:

* данные о пациентах;
* данные об отделениях клиники;
* данные о врачах;
* данные о лекарствах;
* данные о посещениях;
* данные о назначениях.

Выходной информацией системы являются отчеты:

* «Назначения лечащего врача пациенту», в зависимости от указанных и выявленных симптомах заболевания;
* «Справка клиники», подтверждающая прохождение данным пациентом лечения данного заболевания в данной клинике.

В результате решения задачи рассчитываются результатные показатели, которые требуется выдать на материальный носитель в виде, удобном для пользователя. Так как результатный документ используется для осуществления процессов управления, он должен отвечать следующим требованиям:

* Полнота информации;
* Своевременность предоставления информации управленческому персоналу;
* Достоверность предоставляемой информации;
* Хорошая читаемость (логичность построения форм и наличие хорошо отредактированного текста шапок документов);
* Отсутствие показателей, рассчитываемых вручную.

Для моделирования данных чаще всего используется диаграмма «сущность-связь» (ERD), при помощи которой выделяются важные сущности, их свойства и отношения друг с другом. С помощью ERD определяются важные для конкретного случая объекты или сущности, их отношения, свойства и атрибуты.

Диаграмма «сущность-связь» для разрабатываемой информационной системы ведения документации коммерческой медицинской клиники представлена на рисунке 1.

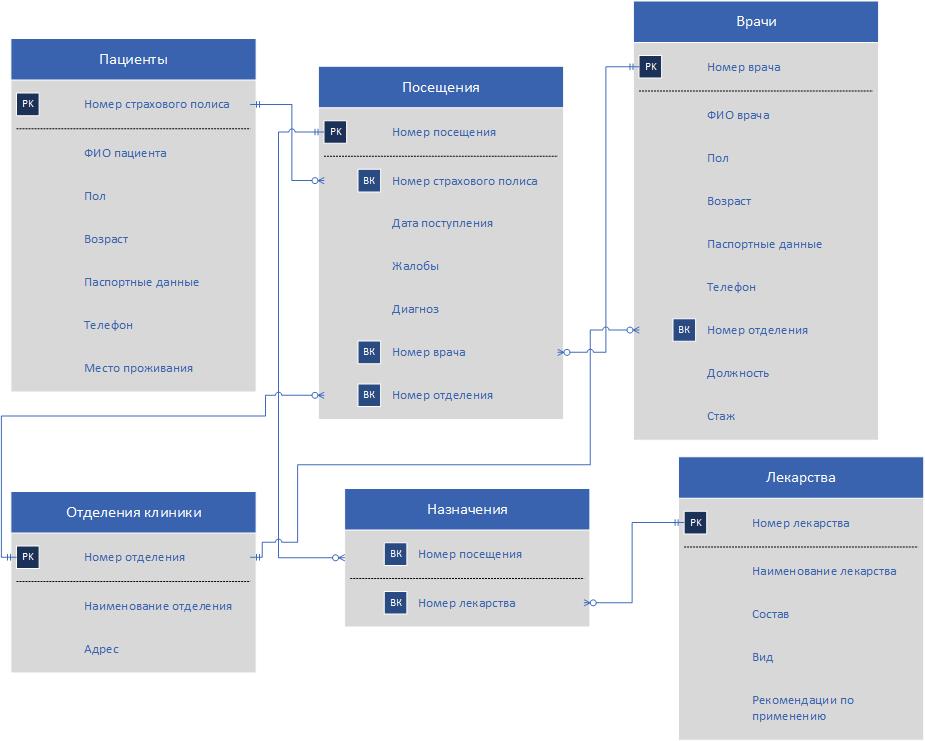


Рисунок 1 – Диаграмма «сущность-связь» для разрабатываемой системы

Рассмотрим подробнее, какие реальные объекты предметной области отражают выделенные сущности и как отношения между сущностями на диаграмме соответствуют взаимосвязям объектов на практике.

Структура связанных сущностей приведена в таблице 1.

Таблица 1 – Описание атрибутов базы данных

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Сущность | Тип сущности | Атрибут | Тип данных | Ключевое поле |
| 1 | Пациенты | Главная | Номер страхового полиса | Числовой | Первичный ключ |
| ФИО пациента | Текстовый |  |
| Пол | Текстовый |  |
| Возраст | Текстовый |  |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | Паспортные данные | Текстовый |  |
| Телефон | Текстовый |  |
| Место проживания | Текстовый |  |
| 2 | Отделения клиники | Главная | Номер отделения | Числовой | Первичный ключ |
| Наименование отделения | Текстовый |  |
| Адрес | Текстовый |  |
| 3 | Врачи | Подчиненная | Номер врача | Числовой | Первичный ключ |
| ФИО врача | Текстовый |  |
| Пол | Текстовый |  |
| Возраст | Текстовый |  |
| Паспортные данные | Текстовый |  |
| Телефон | Текстовый |  |
| Номер отделения | Числовой | Внешний ключ |
| Должность | Текстовый |  |
| Стаж | Текстовый |  |
| 4 | Лекарства | Главная | Номер лекарства | Числовой | Первичный ключ |
| Наименование лекарства | Текстовый |  |
| Состав | Текстовый |  |
| Вид | Текстовый |  |
| Рекомендации по применению | Текстовый |  |
| 5 | Посещения | Подчиненная | Номер посещения | Числовой | Первичный ключ |
| Номер страхового полиса | Числовой | Внешний ключ1 |
| Дата поступления | Текстовый |  |
| Жалобы | Текстовый |  |
| Диагноз | Текстовый |  |
| Номер врача | Числовой | Внешний ключ2 |
| Номер отделения | Числовой | Внешний ключ3 |
| 6 | Назначения | Подчиненная | Номер посещения | Числовой | Внешний ключ1 |
| Номер лекарства | Числовой | Внешний ключ2 |

### 3.2.1 Технология организации доступа к данным

Технологии доступа к данным являются прослойкой между специальным интерфейсом программы/приложения, с помощью которого одна программа/приложение может взаимодействовать с другой (application programming interface – API), конкретного сервера и приложением пользователя, предоставляя программисту простой унифицированный механизм работы с данными.

В данной курсовой работе использовалась система управления базами данным Microsoft SQL Server 2019.

На рисунке 2 отражена физическая схема базы данных для данной информационной системы.

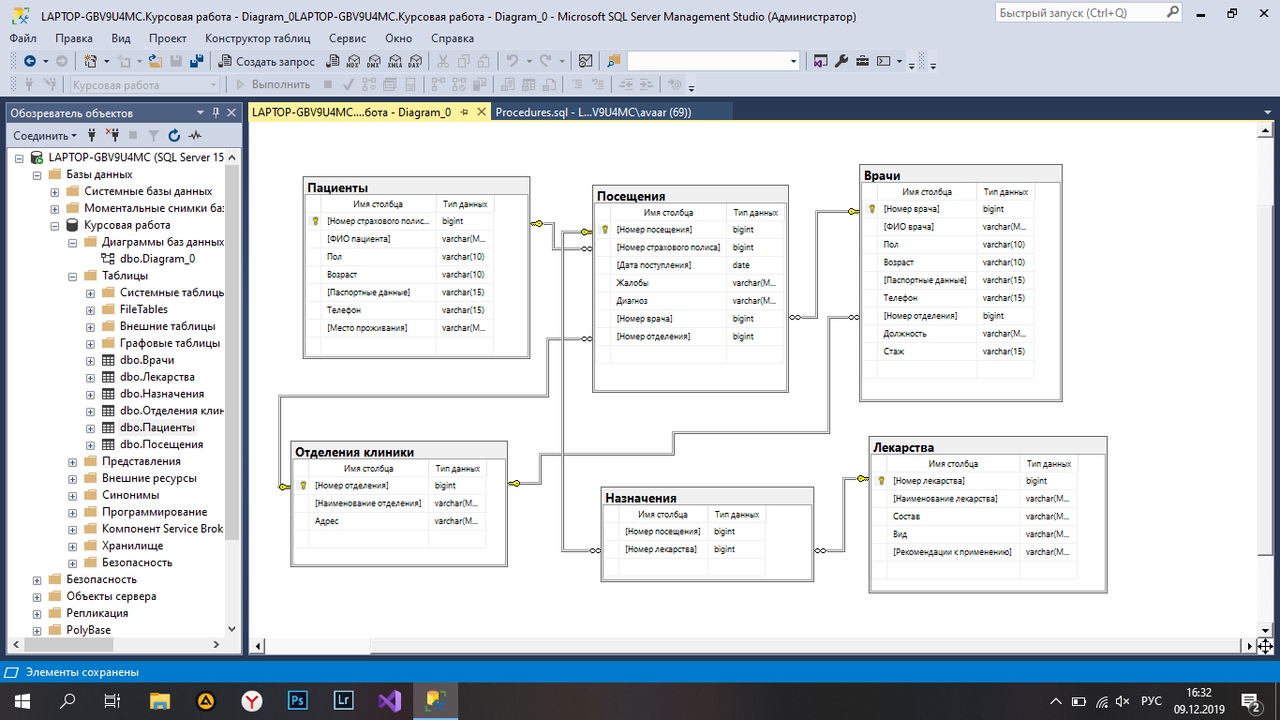


Рисунок 2 – Физическая схема БД

### 3.2.2 Дерево функций и сценарий диалога

Дерево функций представляет собой иерархическую структуру.

Все действия (функции) программного продукта можно разделить на два подмножества функций: основные функции и служебные функции.

Основные функции управления и обработки данных свойственны как информационным системам любой специализации, так и именно в данном модуле. Они отражают особенности процесса обработки информации, получения результатов, ведения информационной базы проекта. Эти функции организованы в том порядке и в том составе, которые продиктованы условиями обработки и управления данными в конкретной предметной области. Вместе основные и служебные функции призваны автоматизировать решение задач, сформулированных в постановке задачи на разработку информационной системы ведения документации коммерческой медицинской клиники. Дерево функций для данной системы представлено на рисунке 3.



Рисунок 3 – Дерево функций

Развитие диалога во времени можно рассматривать как последовательность переходов системы из одного состояния в другое. Ни одно из этих состояний не должно быть тупиковым, т.е. пользователь должен иметь возможность перейти из любого текущего состояния диалога в требуемое (за один или несколько шагов). Для этого в ходе разработки интерфейса необходимо определить все возможные состояния диалога и пути перехода из одного состояния в другое - разработать сценарий диалога.

Цели разработки сценария:

* выявление и устранение возможных тупиковых ситуаций в ходе развития диалога;
* выбор рациональных путей перехода из одного состояния диалога в другое (из текущего в требуемое);
* выявление неоднозначных ситуаций, требующих оказания дополнительной помощи пользователю.

Сценарий диалога для данной системы представлен на рисунке 4.

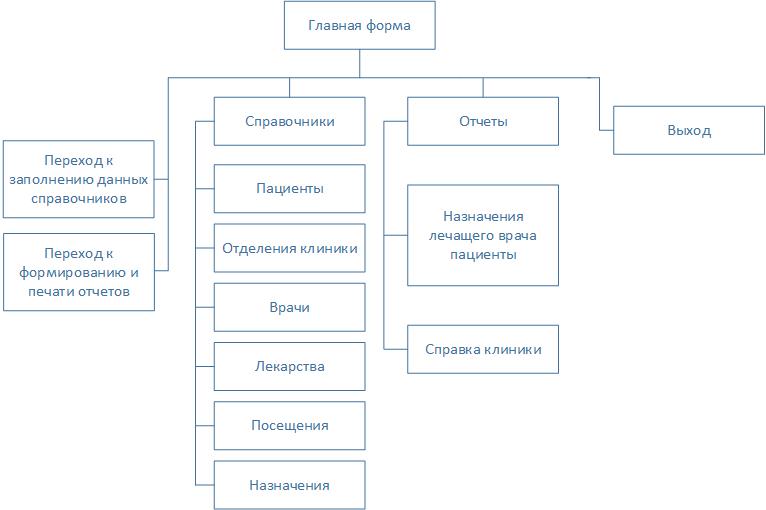


Рисунок 4 – Сценарий диалога

Главная форма включает в себя переход к заполнению данных справочников «Пациенты», «Отделения клиники», «Врачи», «Лекарства», «Посещения» и «Назначения». Также из главной формы можно перейти к формированию и печати отчетов и выйти из приложения.

### 3.2.3 Диаграмма IDEF 0

IDEF0 — методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является ее акцент на соподчиненность объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ).

Функциональная модель IDEF0 представляет собой набор блоков, каждый из которых представляет собой «черный ящик» с входами и выходами, управлением и механизмами, которые детализируются (декомпозируются) до необходимого уровня. Наиболее важная функция расположена в верхнем левом углу. А соединяются функции между собой при помощи стрелок и описаний функциональных блоков. При этом каждый вид стрелки или активности имеет собственное значение. Данная модель позволяет описать все основные виды процессов, как административные, так и организационные.

Стрелки могут быть:

* Входящие (слева) – вводные, которые ставят определенную задачу;
* Исходящие (справа) – выводящие результат деятельности;
* Управляющие (сверху вниз) – механизмы управления (положения, инструкции и пр.);
* Механизмы (снизу вверх) – что используется для того, чтобы произвести необходимую работу.



Рисунок 5 – Контекстная диаграмма IDEF 0

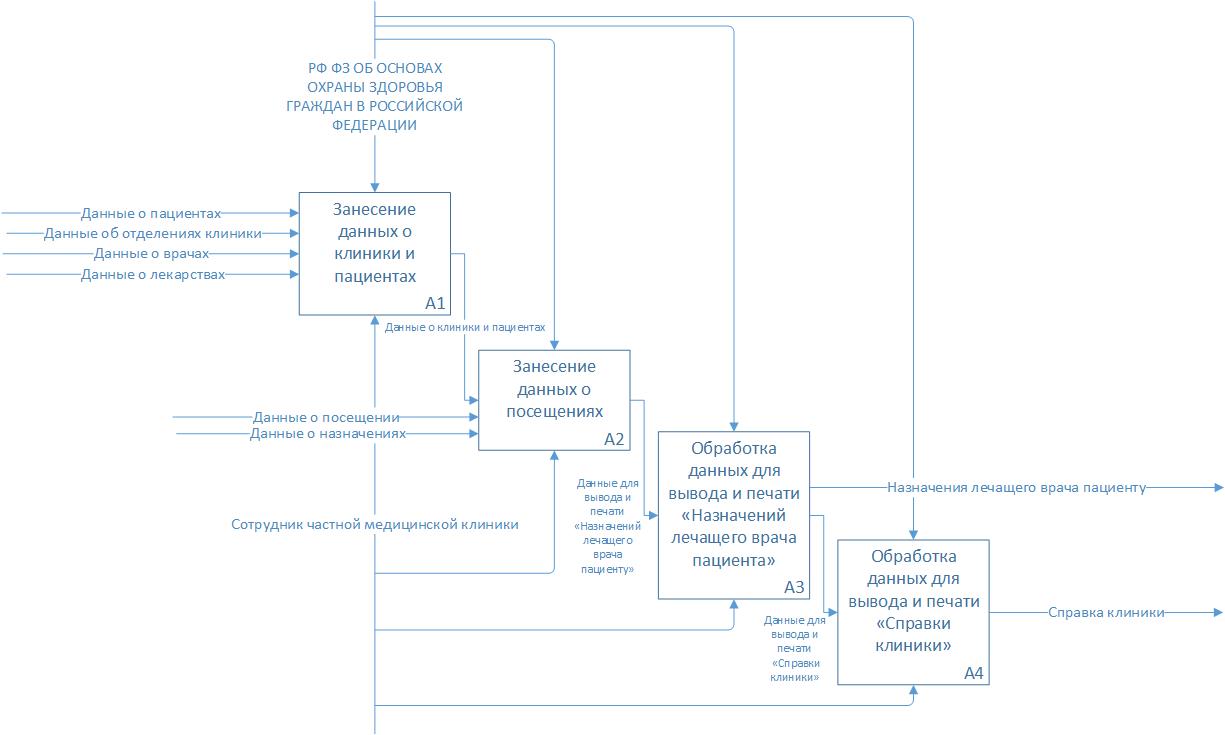


Рисунок 6 – Детализированная диаграмма IDEF 0

# 4 Интерфейс пользователя

Интерфейс для диалога с пользователем разрабатывался с использованием интерфейса программирования приложений Windows Forms в Visual Studio 2019 с помощью языка программирования высокого уровня Visual C#. Его главная форма представлена на рисунке 7.

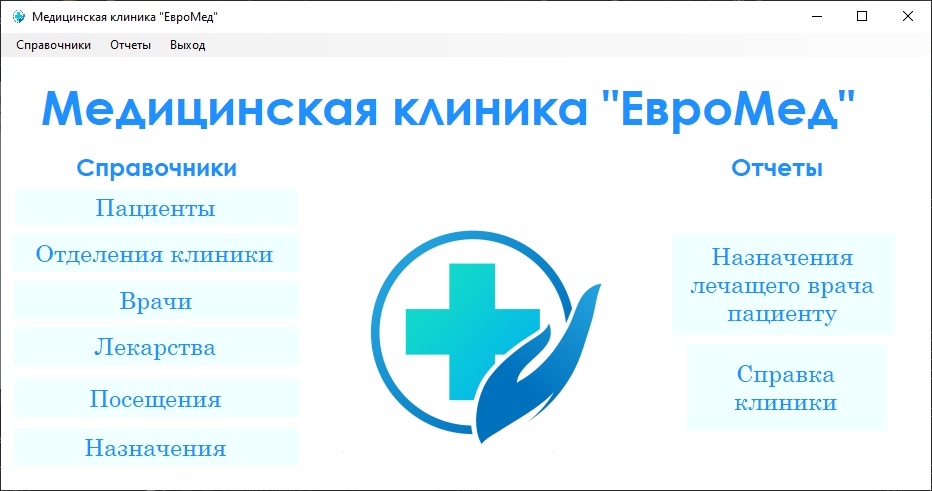


Рисунок 7 – Главная форма

При запуске программы перед пользователем встает главная форма. На ней он видит условное название медицинской клиники, ее логотип, контекстное меню и главное меню, позволяющее перейти в интерфейс по работе со справочниками и отчетами. В системе представлено 6 справочников: «Пациенты», «Отделения клиники», «Врачи», «Лекарства», «Посещения», «Назначения» и 2 отчета: «Назначения лечащего врача пациенту» и «Справка клиники».

Чтобы перейти к нужному справочнику или отчету, пользователю нужно нажать на соответствующую кнопку в главном или контекстном меню. Чтобы выйти из приложения, нужно нажать на кнопку «Выход».

При нажатии кнопки «Пациенты» пользователь попадет на форму «Пациенты», представленную на рисунке 8.

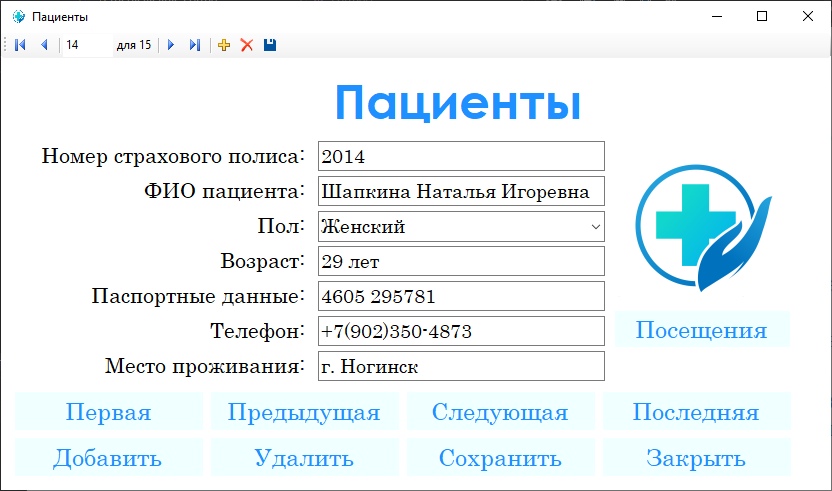


Рисунок 8 – Форма «Пациенты»

На данной форме присутствуют необходимые для лечения данные пациентов, посещающих медицинскую клинику, навигационные панель и кнопки для удобного отображения данных, кнопка закрытия формы, а также кнопка для перехода со справочника «Пациенты» к справочнику «Посещения», который сразу же выдает данные о визите нужного пользователю пациента (рисунок 9).

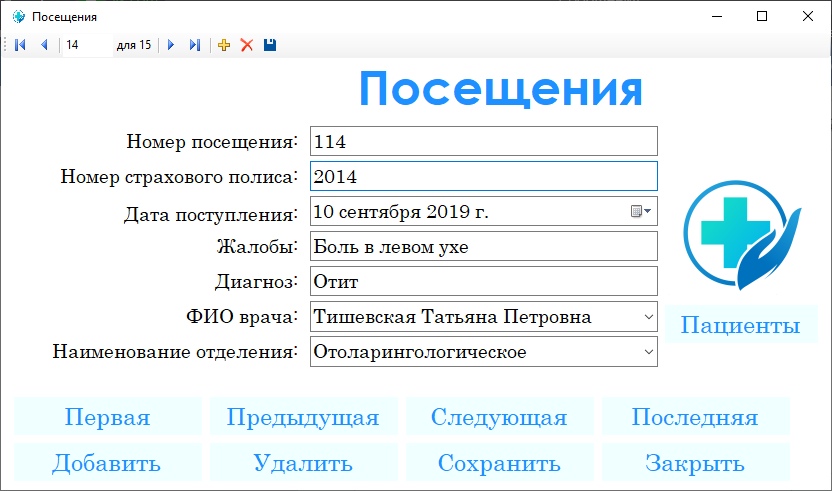


Рисунок 9 – Форма «Посещения»

На форме «Посещения» присутствует кнопка к возвращению на форму «Пациенты», на который показывается вся необходимая информация о том или ином пациенте, который посещал клинику. На всех формах справочников присутствуют навигационная панель и кнопки, кнопка закрытия форм.

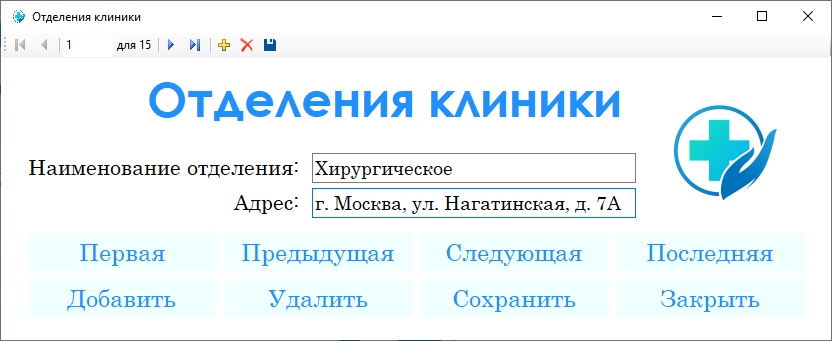


Рисунок 10 – Форма «Отделения клиники»

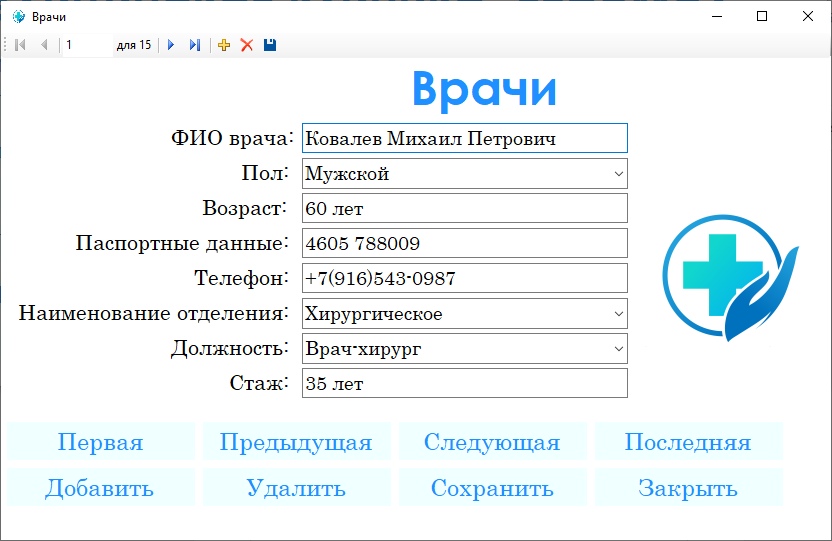


Рисунок 11 – Форма «Врачи»

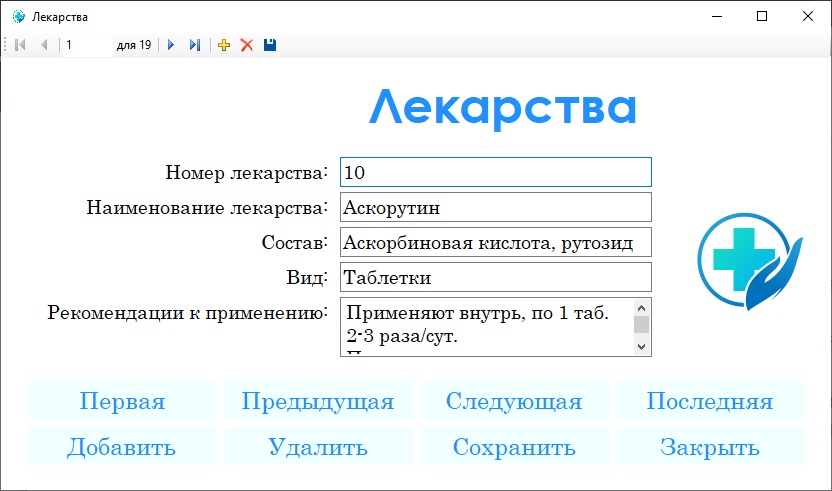


Рисунок 12 – Форма «Лекарства»

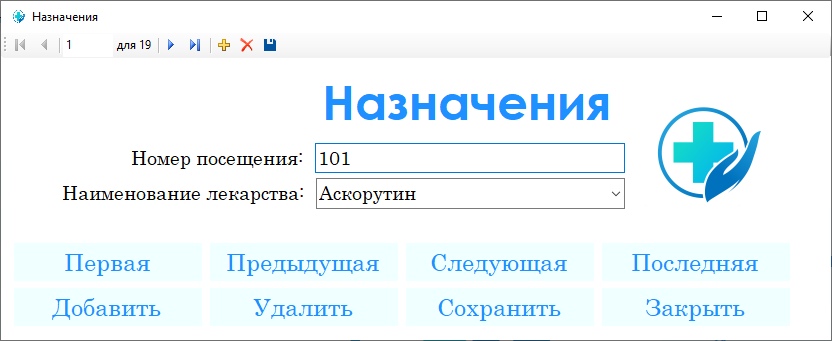


Рисунок 13 – Форма «Назначения»

При нажатии кнопки «Назначения лечащего врача пациенту» на главной форме пользователь попадет на форму отчета «Назначения лечащего врача пациенту», представленную на рисунке 15. Отчет реализован через хранимую процедуру и написан на языке запросов SQL в Microsoft SQL Server Management Studio 2019. Запрос представлен на рисунке 14.

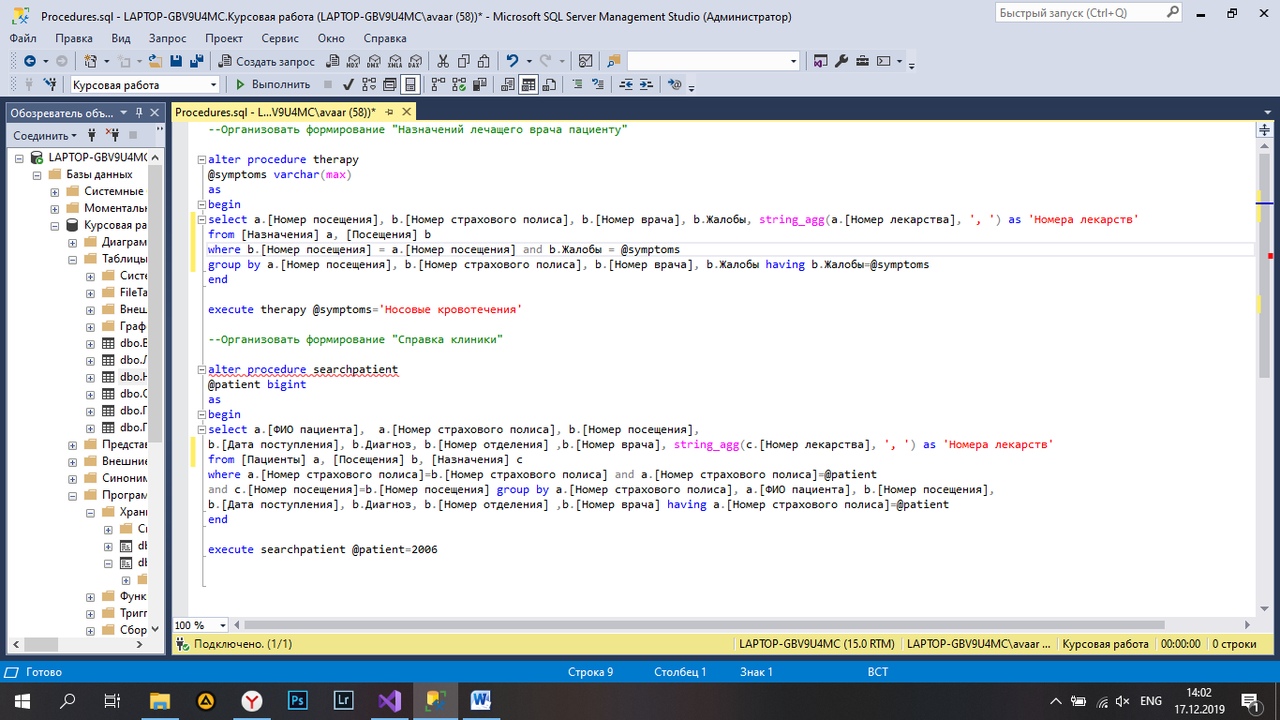


Рисунок 14 – Запрос для отчета «Назначения лечащего врача пациенту»

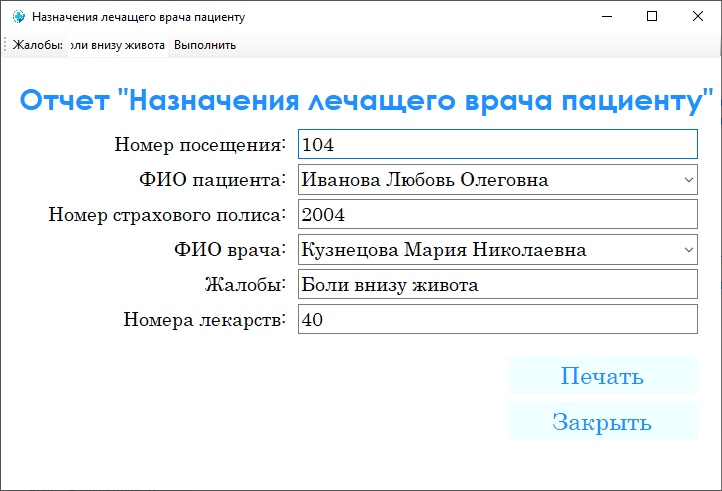


Рисунок 15 – Форма «Назначения лечащего врача пациенту»

На форме пользователь формирует отчет при вводе своих жалоб в окно на верхней панели и нажатии кнопки «Выполнить». По желанию пользователя отчет может быть распечатан. Кнопка «Закрыть» приводит к закрытию данной формы.

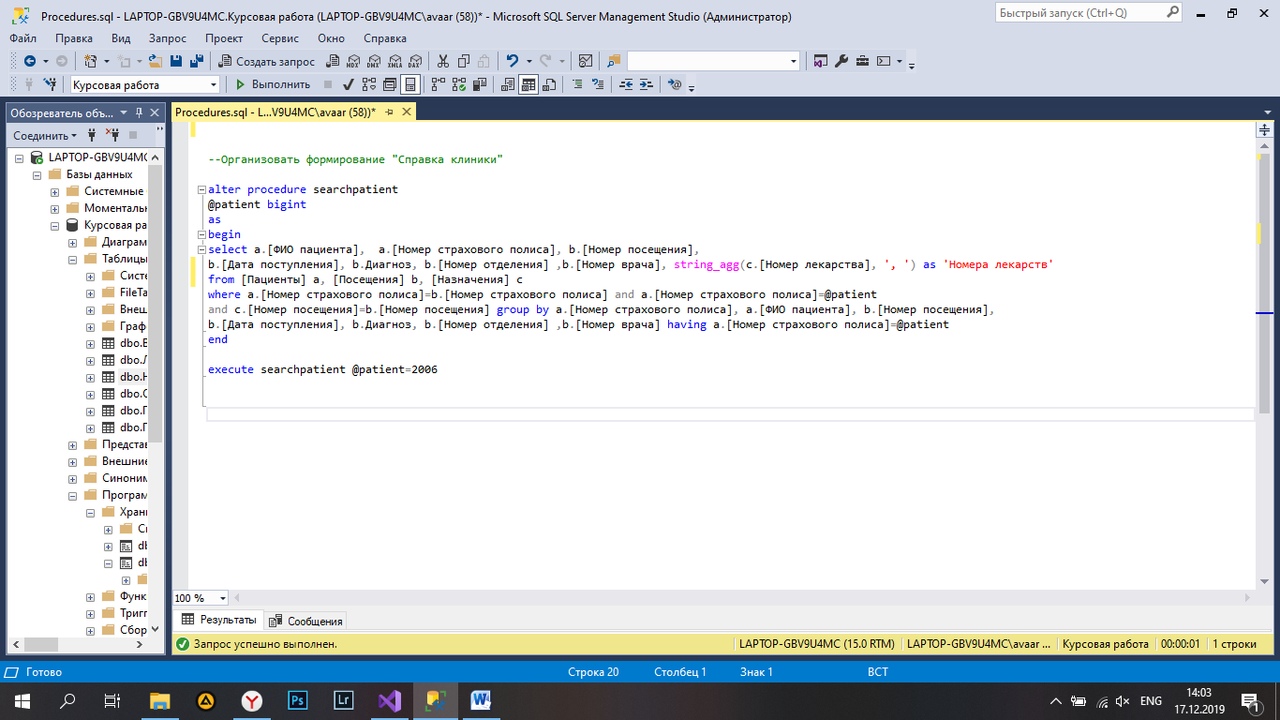


Рисунок 16 – Запрос для отчета «Справки клиники»

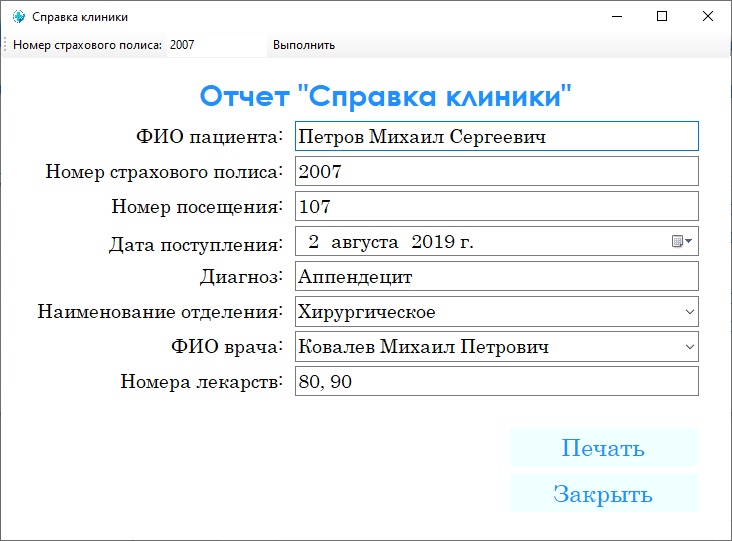


Рисунок 17 – Форма «Справка клиники»

При нажатии кнопки «Печать» на формах для формирования отчетов пользователь увидит диалоговые окна выбора принтера и процесса печати (Рисунок 18 и Рисунок 19).

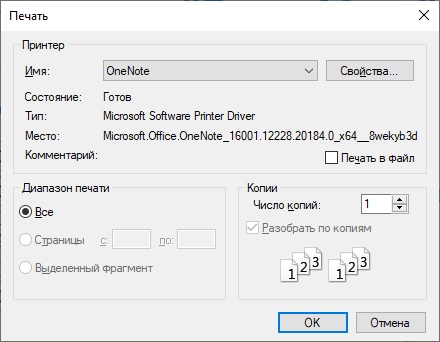


Рисунок 18 – Диалоговое окно выбора принтера

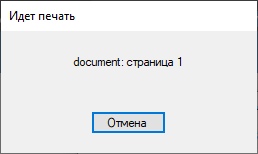


Рисунок 19 – Диалоговое окно процесса печати

Сформированные отчеты после печати будут выглядеть следующим образом (Рисунок 20 и Рисунок 21).

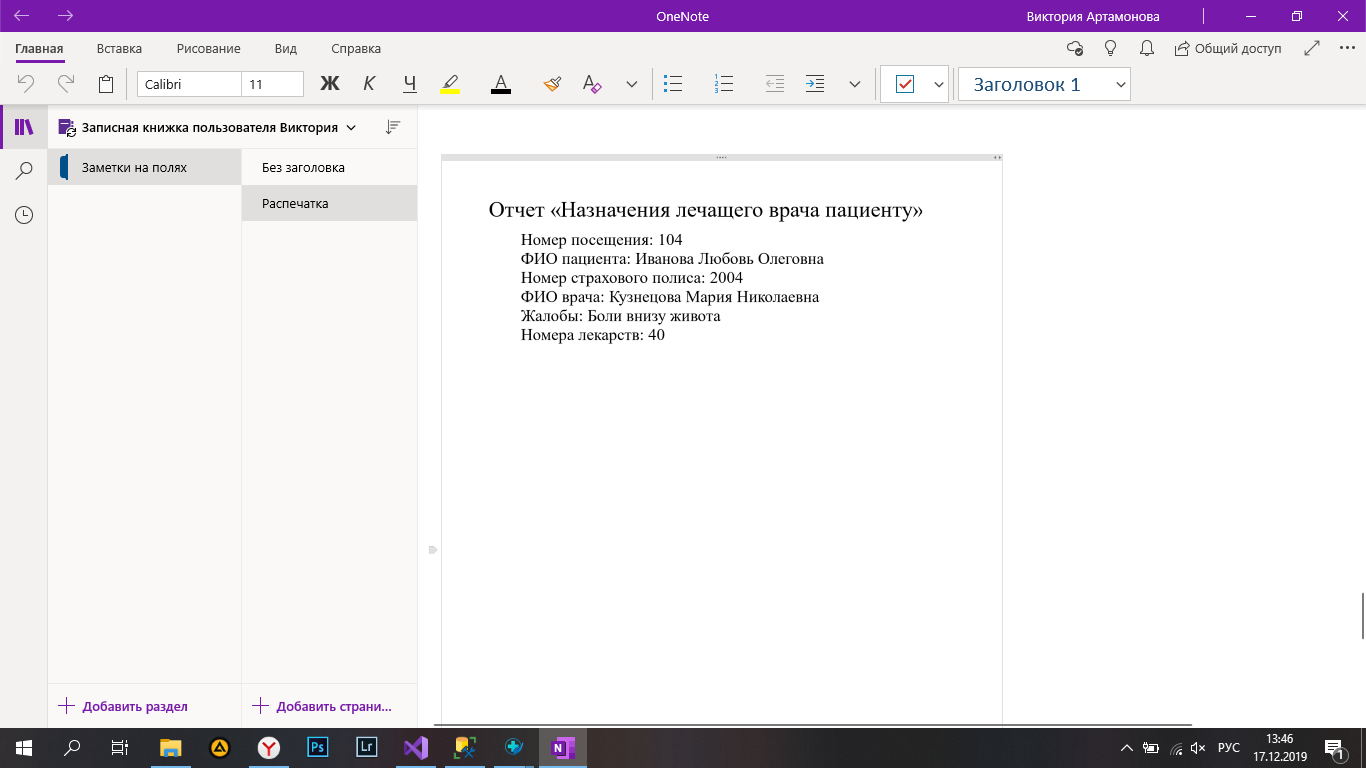


Рисунок 20 – Напечатанный отчет «Назначения лечащего врача пациенту»

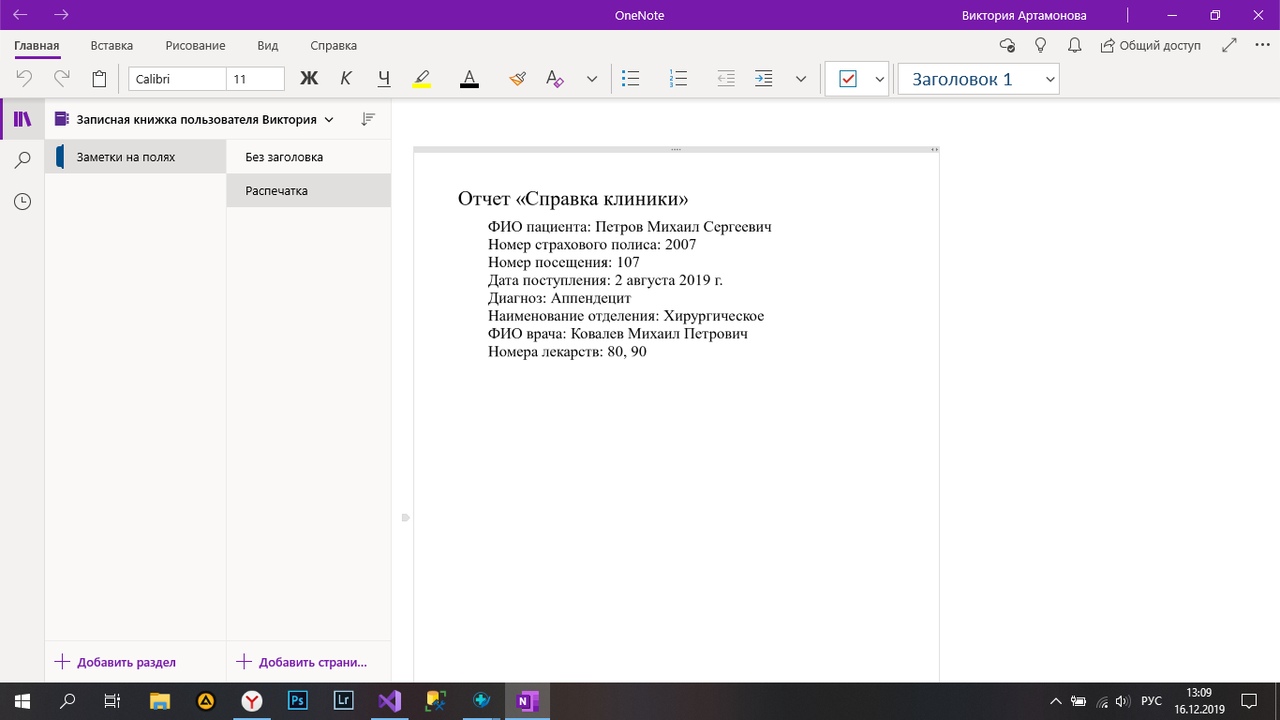


Рисунок 21 – Напечатанный отчет «Справка клиники»

# Заключение

В ходе данной работы были выполнены все поставленные задачи. Была разработана база данных, благодаря которой был реализован пользовательский интерфейс информационной системы – «Ведение документации коммерческой медицинской клиники». Разработка проводилась на языках программирования C# и SQL c использованием таких программных средств, как инструментальные средства моделирования предметной области в среде Microsoft Visio.

# Библиографический список

1. А.К. Гультява и В.А. Машина Проектирование и дизайн пользовательского интерфейса - 239с.
2. Дж. Раскин. Интерфейс: новые направления в проектировании компьютерных систем. - [Текст], М: 2005. - 272с.
3. С.Ф. Сергеев, П.И. Падерно, Н.А. Назаренко. Введение в проектирование интеллектуальных интерфейсов. - [Текст], М: 2011. - 108с.
4. С.Ф. Сергеев. Методы тестирования и оптимизации интерфейсов информационных систем. - [Текст], М: 2013. - 115с.
5. Федеральный закон «Об основах охраны здоровья граждан в Российской Федерации».

# Приложения

## Приложение А Программный код главной формы

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace medicalclinic

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form2 f = new Form2(null);

f.Show();

}

private void button2\_Click\_1(object sender, EventArgs e)

{

Form3 f = new Form3();

f.Show();

}

private void button3\_Click\_1(object sender, EventArgs e)

{

Form4 f = new Form4();

f.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

Form5 f = new Form5();

f.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

Form6 f = new Form6(null);

f.Show();

}

private void button6\_Click(object sender, EventArgs e)

{

Form7 f = new Form7();

f.Show();

}

private void пациентыToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form2 f = new Form2(null);

f.Show();

}

private void отделенияКлиникиToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form3 f = new Form3();

f.Show();

}

private void врачиToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form4 f = new Form4();

f.Show();

}

private void лекарстваToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form5 f = new Form5();

f.Show();

}

private void посещенияToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form6 f = new Form6(null);

f.Show();

}

private void назначенияToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form7 f = new Form7();

f.Show();

}

private void назначенияЛечащегоВрачаПациентыToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form8 f = new Form8();

f.Show();

}

private void справкаКлиникиToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form9 f = new Form9();

f.Show();

}

private void button7\_Click(object sender, EventArgs e)

{

Form8 f = new Form8();

f.Show();

}

private void button8\_Click(object sender, EventArgs e)

{

Form9 f = new Form9();

f.Show();

}

private void выходToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

## Приложение Б Программный код формы «Пациенты»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace medicalclinic

{

public partial class Form2 : Form

{

public Form2(string Nomer)

{

InitializeComponent();

if (Nomer != null)

nomer = Nomer;

}

static string nomer;

private void пациентыBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.пациентыBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void Form2\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Пациенты". При необходимости она может быть перемещена или удалена.

this.пациентыTableAdapter.Fill(this.medicalclinicDataSet.Пациенты);

if (nomer != null)

for (int i = 0; i < 16; i++)

{

if (nomer != номер\_страхового\_полисаTextBox.Text)

пациентыBindingSource.MoveNext();

}

}

private void button1\_Click\_1(object sender, EventArgs e)

{

пациентыBindingSource.MoveFirst();

}

private void button2\_Click\_1(object sender, EventArgs e)

{

пациентыBindingSource.MovePrevious();

}

private void button3\_Click\_1(object sender, EventArgs e)

{

пациентыBindingSource.MoveNext();

}

private void button4\_Click\_1(object sender, EventArgs e)

{

пациентыBindingSource.MoveLast();

}

private void button5\_Click\_1(object sender, EventArgs e)

{

пациентыBindingSource.AddNew();

}

private void button6\_Click\_1(object sender, EventArgs e)

{

пациентыBindingSource.RemoveCurrent();

}

private void button7\_Click\_1(object sender, EventArgs e)

{

this.Validate();

this.пациентыBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void button8\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

private void button9\_Click(object sender, EventArgs e)

{

Form6 f = new Form6(номер\_страхового\_полисаTextBox.Text);

f.Show();

}

}

}

## Приложение В Программный код формы «Отделения клиники»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace medicalclinic

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void отделения\_клиникиBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.отделения\_клиникиBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void Form3\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Отделения\_клиники". При необходимости она может быть перемещена или удалена.

this.отделения\_клиникиTableAdapter.Fill(this.medicalclinicDataSet.Отделения\_клиники);

}

private void button1\_Click\_1(object sender, EventArgs e)

{

отделения\_клиникиBindingSource.MoveFirst();

}

private void button2\_Click\_1(object sender, EventArgs e)

{

отделения\_клиникиBindingSource.MovePrevious();

}

private void button3\_Click\_1(object sender, EventArgs e)

{

отделения\_клиникиBindingSource.MoveNext();

}

private void button4\_Click\_1(object sender, EventArgs e)

{

отделения\_клиникиBindingSource.MoveLast();

}

private void button5\_Click\_1(object sender, EventArgs e)

{

отделения\_клиникиBindingSource.AddNew();

}

private void button6\_Click\_1(object sender, EventArgs e)

{

отделения\_клиникиBindingSource.RemoveCurrent();

}

private void button7\_Click\_1(object sender, EventArgs e)

{

this.Validate();

this.отделения\_клиникиBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void button8\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

}

}

## Приложение Г Программный код формы «Врачи»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace medicalclinic

{

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

}

private void врачиBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.врачиBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void Form4\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Отделения\_клиники". При необходимости она может быть перемещена или удалена.

this.отделения\_клиникиTableAdapter.Fill(this.medicalclinicDataSet.Отделения\_клиники);

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Врачи". При необходимости она может быть перемещена или удалена.

this.врачиTableAdapter.Fill(this.medicalclinicDataSet.Врачи);

}

private void номер\_отделенияLabel\_Click(object sender, EventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

врачиBindingSource.MoveFirst();

}

private void button2\_Click\_1(object sender, EventArgs e)

{

врачиBindingSource.MovePrevious();

}

private void button3\_Click\_1(object sender, EventArgs e)

{

врачиBindingSource.MoveNext();

}

private void button4\_Click\_1(object sender, EventArgs e)

{

врачиBindingSource.MoveLast();

}

private void button5\_Click\_1(object sender, EventArgs e)

{

врачиBindingSource.AddNew();

}

private void button6\_Click\_1(object sender, EventArgs e)

{

врачиBindingSource.RemoveCurrent();

}

private void button7\_Click\_1(object sender, EventArgs e)

{

this.Validate();

this.врачиBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void button8\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

}

}

## Приложение Д Программный код формы «Лекарства»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace medicalclinic

{

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

}

private void лекарстваBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.лекарстваBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void Form5\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Лекарства". При необходимости она может быть перемещена или удалена.

this.лекарстваTableAdapter.Fill(this.medicalclinicDataSet.Лекарства);

}

private void button1\_Click(object sender, EventArgs e)

{

лекарстваBindingSource.MoveFirst();

}

private void button2\_Click(object sender, EventArgs e)

{

лекарстваBindingSource.MovePrevious();

}

private void button3\_Click(object sender, EventArgs e)

{

лекарстваBindingSource.MoveNext();

}

private void button4\_Click(object sender, EventArgs e)

{

лекарстваBindingSource.MoveLast();

}

private void button5\_Click(object sender, EventArgs e)

{

лекарстваBindingSource.AddNew();

}

private void button6\_Click(object sender, EventArgs e)

{

лекарстваBindingSource.RemoveCurrent();

}

private void button7\_Click(object sender, EventArgs e)

{

this.Validate();

this.лекарстваBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void button8\_Click(object sender, EventArgs e)

{

this.Close();

}

private void наименование\_лекарстваTextBox\_TextChanged(object sender, EventArgs e)

{

}

private void номер\_лекарстваTextBox\_TextChanged(object sender, EventArgs e)

{

}

private void видTextBox\_TextChanged(object sender, EventArgs e)

{

}

private void составTextBox\_TextChanged(object sender, EventArgs e)

{

}

private void рекомендации\_к\_применениюTextBox\_TextChanged(object sender, EventArgs e)

{

}

}

}

## Приложение Е Программный код формы «Посещения»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace medicalclinic

{

public partial class Form6 : Form

{

public Form6(string Nomer)

{

InitializeComponent();

if (Nomer != null)

nomer = Nomer;

}

static string nomer;

private void посещенияBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.посещенияBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void Form6\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Отделения\_клиники". При необходимости она может быть перемещена или удалена.

this.отделения\_клиникиTableAdapter.Fill(this.medicalclinicDataSet.Отделения\_клиники);

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Врачи". При необходимости она может быть перемещена или удалена.

this.врачиTableAdapter.Fill(this.medicalclinicDataSet.Врачи);

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Посещения". При необходимости она может быть перемещена или удалена.

this.посещенияTableAdapter.Fill(this.medicalclinicDataSet.Посещения);

if (nomer != null)

for (int i = 0; i < 16; i++)

{

if (nomer != номер\_страхового\_полисаTextBox.Text)

посещенияBindingSource.MoveNext();

}

}

private void button1\_Click(object sender, EventArgs e)

{

посещенияBindingSource.MoveFirst();

}

private void button2\_Click(object sender, EventArgs e)

{

посещенияBindingSource.MovePrevious();

}

private void button3\_Click(object sender, EventArgs e)

{

посещенияBindingSource.MoveNext();

}

private void button4\_Click(object sender, EventArgs e)

{

посещенияBindingSource.MoveLast();

}

private void button5\_Click(object sender, EventArgs e)

{

посещенияBindingSource.AddNew();

}

private void button6\_Click(object sender, EventArgs e)

{

посещенияBindingSource.RemoveCurrent();

}

private void button7\_Click(object sender, EventArgs e)

{

this.Validate();

this.посещенияBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void button8\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button9\_Click(object sender, EventArgs e)

{

Form2 f = new Form2(номер\_страхового\_полисаTextBox.Text);

f.Show();

}

}

}

## Приложение Ж Программный код формы «Назначения»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace medicalclinic

{

public partial class Form7 : Form

{

public Form7()

{

InitializeComponent();

}

private void назначенияBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.назначенияBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void Form7\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Лекарства". При необходимости она может быть перемещена или удалена.

this.лекарстваTableAdapter.Fill(this.medicalclinicDataSet.Лекарства);

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Назначения". При необходимости она может быть перемещена или удалена.

this.назначенияTableAdapter.Fill(this.medicalclinicDataSet.Назначения);

}

private void button1\_Click(object sender, EventArgs e)

{

назначенияBindingSource.MoveFirst();

}

private void button2\_Click(object sender, EventArgs e)

{

назначенияBindingSource.MovePrevious();

}

private void button3\_Click(object sender, EventArgs e)

{

назначенияBindingSource.MoveNext();

}

private void button4\_Click(object sender, EventArgs e)

{

назначенияBindingSource.MoveLast();

}

private void button5\_Click(object sender, EventArgs e)

{

назначенияBindingSource.AddNew();

}

private void button6\_Click(object sender, EventArgs e)

{

назначенияBindingSource.RemoveCurrent();

}

private void button7\_Click(object sender, EventArgs e)

{

this.Validate();

this.назначенияBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.medicalclinicDataSet);

}

private void button8\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

## Приложение З Программный код формы «Назначения лечащего врача пациенту»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Drawing.Printing;

namespace medicalclinic

{

public partial class Form8 : Form

{

private string text = "";

public Form8()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

text = $"Номер посещения: {номер\_посещенияTextBox.Text}\nФИО пациента: {comboBox2.Text}\nНомер страхового полиса: {номер\_страхового\_полисаTextBox.Text}\nФИО врача: {comboBox1.Text}\nЖалобы: {жалобыTextBox.Text}\nНомера лекарств: {номера\_лекарствTextBox.Text}\n";

PrintDocument printDocument = new PrintDocument();

printDocument.PrintPage += PrintPageHandler;

PrintDialog printDialog = new PrintDialog();

printDialog.Document = printDocument;

if (printDialog.ShowDialog() == DialogResult.OK)

printDialog.Document.Print();

}

void PrintPageHandler(object sender, PrintPageEventArgs e)

{

e.Graphics.DrawString("Отчет «Назначения лечащего врача пациенту»\n\n", new Font("Times New Roman", 24), Brushes.Black, 0, 0);

e.Graphics.DrawString(text, new Font("Times New Roman", 18), Brushes.Black, 50, 50);

}

private void fillToolStripButton\_Click(object sender, EventArgs e)

{

try

{

this.therapyTableAdapter.Fill(this.medicalclinicDataSet.therapy, symptomsToolStripTextBox.Text);

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

private void Form8\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Лекарства". При необходимости она может быть перемещена или удалена.

this.лекарстваTableAdapter.Fill(this.medicalclinicDataSet.Лекарства);

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Врачи". При необходимости она может быть перемещена или удалена.

this.врачиTableAdapter.Fill(this.medicalclinicDataSet.Врачи);

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

## Приложение И Программный код формы «Справка клиники»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Drawing.Printing;

namespace medicalclinic

{

public partial class Form9 : Form

{

private string text = "";

public Form9()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

text = $"ФИО пациента: {фИО\_пациентаTextBox.Text}\nНомер страхового полиса: {номер\_страхового\_полисаTextBox.Text}\nНомер посещения: {номер\_посещенияTextBox.Text}\nДата поступления: {дата\_поступленияDateTimePicker.Text}\nДиагноз: {диагнозTextBox.Text}\nНаименование отделения: {comboBox1.Text}\nФИО врача: {comboBox2.Text}\nНомера лекарств: {номера\_лекарствTextBox.Text}\n";

PrintDocument printDocument = new PrintDocument();

printDocument.PrintPage += PrintPageHandler;

PrintDialog printDialog = new PrintDialog();

printDialog.Document = printDocument;

if (printDialog.ShowDialog() == DialogResult.OK)

printDialog.Document.Print();

}

void PrintPageHandler(object sender, PrintPageEventArgs e)

{

e.Graphics.DrawString("Отчет «Справка клиники»\n\n", new Font("Times New Roman", 24), Brushes.Black, 0, 0);

e.Graphics.DrawString(text, new Font("Times New Roman", 18), Brushes.Black, 50, 50);

}

private void fillToolStripButton\_Click(object sender, EventArgs e)

{

try

{

this.searchpatientTableAdapter.Fill(this.medicalclinicDataSet.searchpatient, new System.Nullable<long>(((long)(System.Convert.ChangeType(patientToolStripTextBox.Text, typeof(long))))));

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

private void Form9\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Лекарства". При необходимости она может быть перемещена или удалена.

this.лекарстваTableAdapter.Fill(this.medicalclinicDataSet.Лекарства);

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Врачи". При необходимости она может быть перемещена или удалена.

this.врачиTableAdapter.Fill(this.medicalclinicDataSet.Врачи);

// TODO: данная строка кода позволяет загрузить данные в таблицу "medicalclinicDataSet.Отделения\_клиники". При необходимости она может быть перемещена или удалена.

this.отделения\_клиникиTableAdapter.Fill(this.medicalclinicDataSet.Отделения\_клиники);

}

private void button2\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

}

}