МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

| *Факультет* | *Кафедра* |
|---|---|
| вычислительной техники | МОиПЭВМ |

Направление подготовки 09.03.02 «Информационные системы и технологии»

**БАКАЛАВРСКАЯ РАБОТА**

на тему

# Информационная система управления ветеринарной клиникой

**Студент** _____ Рыбалкина Анастасия Владимировна
(подпись, дата) (ФИО полностью)

**Руководитель** _____ Попова Н.А.
(подпись, дата) (фамилия, инициалы)

**Нормоконтролёр** _____ Такташкин Д.В.
(подпись, дата) (фамилия, инициалы)

*Работа допущена к защите* (протокол заседания кафедры от _____№_____)

**Заведующий кафедрой** _____ Макарычев П.П.
(подпись) (фамилия, инициалы)

*Работа защищена с отметкой* _____ (протокол заседания ГЭК от _____№____)

**Секретарь ГЭК** _____ Попова Н.А.
(подпись) (фамилия, инициалы)

Пенза, 2022

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

| *Факультет* | *Кафедра* |
|---|---|
| Вычислительной техники | МО и ПЭВМ |

«Утверждаю»
Заведующий кафедрой
_____П.П. Макарычев
«__»_____2021 г

# ЗАДАНИЕ
# НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
# БАКАЛАВРА

1. Студент Рыбалкина Анастасия Владимировна гр. 18ВИ1 факультета __ВТ_____
<Фамилия, имя, отчество полностью>
2.Тема работы Информационная система управления ветеринарной клиникой
_____

Тема утверждена приказом ПГУ №__321/0__от "_18_"___апреля___2022 г.

3. Руководитель работы__Попова Н.А._____

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Назначение: автоматизация управления ветеринарной клиники
_____
Основные функции системы:
1. Учет серий медикаментов_____
2. Учет амбулаторных карт питомцев_____
3. Учет записей на прием к врачу_____
4. Формирование отчетов по сериям медикаментов_____
_____
Технология разработки: Предметно-ориентированное программирование_____
Язык программирования: Встроенный язык программирования 1С: Предприятие_____
Среда программирования: Технологическая платформа 1С: Предприятие 8.3_____
_____
_____
_____

5. Перечень подлежащих разработке вопросов

1. Анализ предметной области и постановка задачи_____

2. Анализ требований на разработку
3.Разработка программных средств
4. Тестирование программных средств
5. Оформление ВКР

5. Календарный график выполнения работы

| № п/п | Наименование этапов работы | Объем работы | Срок выполнения | Подпись руководителя |
|---|---|---|---|---|
| 1. | Анализ предметной области и постановка задачи | 20% | 12.04 – 18.04 | |
| 2. | Анализ требований на разработку | 10% | 19.04 – 25.04 | |
| 3. | Разработка программных средств | 30% | 26.04 – 26.05 | |
| 4. | Тестирование программных средств | 20% | 27.05 – 1.06 | |
| 5. | Оформление ВКР | 20% | 2.06 – 6.06 | |

Дата выдачи задания "___"_____2022 г.

Руководитель бакалаврской работы _____

(подпись, дата)

Задание к исполнению принял студент _____
(подпись, дата)

Работу к защите допустить
Декан факультета _____
(подпись, дата)

# Реферат

Пояснительная записка содержит 80 листов, 42 рисунка, 30 использованных источников, 15 таблиц и 2 приложения.

ВЕТЕРИНАРНАЯ КЛИНИКА, ВЕТЕРИНАРИЯ, ЖИВОТНЫЕ, МЕДИКАМЕНТЫ, АНАЛИЗЫ, 1С: ПРЕДПРИЯТИЕ, UML, IDEF0, ERWIN

Предмет работы: разработка информационной системы управления ветеринарной клиникой.

Объектом работы является: автоматизация деятельности ветеринарной клиники.

Цель данной работы: разработка информационной систем для автоматизации деятельности ветеринарной клиники.

Технология разработки – платформа «1С: Предприятие 8.3».

Результаты работы: спроектирована и разработана конфигурация для автоматизации деятельности ветеринарной клиники в системе «1С: Предприятие 8.3».

Область применения: ветеринарные клиники, занимающиеся приемом пациентов и проведения анализов, при которых находится аптека.

# Содержание

Введение ..... 6
1 Постановка задачи и анализ требований ..... 8
1.1 Постановка задачи ..... 8
1.2 Анализ информационных систем аналогичного назначения ..... 10
1.3 Выбор технологий проектирования и разработки ..... 14
2 Проектирование информационной системы ..... 17
2.1 Анализ бизнес процессов предприятия ..... 17
2.2 Проектирование логической модели базы данных ..... 20
2.3 Проектирование ПО ..... 23
2.3.1 Анализ прецедентов ..... 24
2.3.2 Диаграмма состояний ..... 26
2.3.3 Диаграмма деятельности ..... 28
2.3.4 Диаграмма последовательности ..... 31
2.3.5 Диаграмма кооперации ..... 32
2.4 Проектирование интерфейса ..... 34
2.5 Планирование разработки и оценки бюджета ..... 36
3 Реализация системы ..... 40
3.1 Реализация структур программного средства ..... 40
3.2 Реализация кода программного средства ..... 54
3.3 Тестирование программного средства ..... 55
3.4 Расчет метрик кода ..... 58
Заключение ..... 61
Список использованных источников ..... 62
Приложение А. Листинг программы ..... 65
Приложение Б. Результаты тестирования ..... 75

# Введение

Ветеринария (ветеринарная медицина) – область научных знаний и практической деятельности, направленных на борьбу с болезнями животных, охрану людей от зооантропонозов (инфекций общих для животных и человека), выпуск доброкачественной в санитарном отношении продукции и решение ветеринарно-санитарных проблем защиты окружающей среды [1].

Всероссийский центр изучения общественного мнения предоставил данные исследования от Grow from Knowledge о том, сколько россиян имеют домашних животных. Россия является одной из стран с наибольшим количеством семей, имеющих домашних питомцев. Владельцами домашних животных среди опрошенных оказались 73 процента [2].

В последнее время всё больше людей осознают необходимость похода к врачу для обеспечения здоровой и комфортной жизни своих питомцев. В связи с этим возникла необходимость автоматизации работы ветеринарных клиник, для упрощения ведения карточек питомцев, записи пациентов, регулирования движений медикаментов, оказания различных услуг.

Важную роль в обеспечении эффективного функционирования ветеринарной клиники играет регистратура, являясь одним из самых напряженных участков. Именно с регистратуры начинается вся работа и именно от неё складываются первые впечатления владельцев животных. Поэтому необходима оперативная организация работы, для быстрой, беспрепятственной регистрации новых клиентов. Также необходимо регулировать интенсивность потока пациентов для равномерной нагрузки врачей. Важно обеспечение систематизированного хранения медицинской документации пациентов, а также корректного отображения для врача.

В преобладающей части современных предприятий объем информационных потоков внутри его бизнес-процессов рано или поздно достигает точки, когда начинают требоваться средства для хранения, обработки, упорядочивания этой информации, поддержки ее в актуальном состоянии.

Таким средством выступает информационные системы, центром которых является программное обеспечение. При этом далеко не во всех сферах деятельности предприятия может использоваться готовые ИС. Это может быть создание ИС с нуля или детальное конфигурирование в рамках определенной программной платформы.

Предмет работы: разработка информационной системы управления ветеринарной клиникой.

Объектом работы является: информационная система управления ветеринарной клиникой.

Цель данной работы: разработка информационной системы для автоматизации управления ветеринарной клиникой.

Для поставленной цели потребуются следующие задачи:

– провести анализ предметной области;
– провести сравнительный анализ продуктов-аналогов;
– провести анализ бизнес-процессов организации;
– провести анализ системы с помощью диаграмм UML;
– разработать ПО согласно составленным требованиям;
– протестировать функционал ПО;
– составить документацию.

На основе выполненного анализа была разработана конфигурация, которая позволяет вести учет медикаментов, врачей, питомцев, записей к врачу и анализы.

# 1 Постановка задачи и анализ требований

## 1.1 Постановка задачи

Назначение ветеринарной клиники – это комплексное или раздельное лечение питомцев, которая осуществляется в виде профилактических ветеринарно-санитарных мер по предупреждению и устранению болезней животных, а также проведение диагностических исследований.

Ранее посещение ветеринарной клиники представляло из себя следующую последовательность действий. Приходит клиент и обращается в регистратуру. Если это его первое посещение, то заводится амбулаторная карта для его питомца, с которой он отправляется на приём к врачу. Приём осуществляется в порядке живой очереди. После посещения врача владелец животного сдаёт амбулаторную карту обратно в регистратуру. Повторное посещение контролируется персоналом регистратуры. При этом возникает ряд проблем:

– человеческий фактор (ошибки регистраторов в заполнении карт);

– трата большого количества времени для работы с амбулаторными картами (поиск нужной карты, утеря карты);

– неупорядоченность записи животных на приём к врачу;

– формирование реестра зарегистрированных животных в «ручную»;

– отсутствие оптимизации в работе регистратуры.

От успешной работы регистратуры в значительной степени зависит эффективность деятельности всей ветеринарной клиники. С целью уменьшения издержек и возникает необходимость в автоматизации работы клиники.

Назначение информационной системы заключается в реализации функций эффективного управления данными. Разработанная информационная система должна повысить эффективность работы ветеринарной клиники, а также аптеки находящейся при ней, с помощью

автоматизированных процессов обработки данных. Это должно привести к уменьшению издержек, связанных с затратами времени на обслуживание клиентов. Так же информационная система должна представлять объективные данные для анализа работы учреждения.

Система должна обеспечивать взаимодействие между регистратурой и врачом, а также между самой ветеринарной поликлиникой и аптекой, находящейся при клинике.

Система должна обеспечивать необходимые потребности ветеринарной клиники:

– доступ к базе данных животных;
– доступ к базе данных врачей;
– формирование амбулаторной карты для новых животных;
– запись на прием к врачу;
– запись на обследования, вакцинации, сдачу анализов;
– ведение карточки питомца;
– работа с расписанием приема специалистов, работы кабинетов, направление питомцев к специалистам;
– ведение отчетной деятельности;
– ведение учета медикаментов;
– обеспечение взаимосвязи отдела регистратуры, врачей, а также прикреплённой к ветеринарной клинике аптеки.

Система должна вести учет владельцев животных. По каждому клиенту можно посмотреть полную информацию о его питомцах, что являются пациентами ветеринарной клиники.

Система должна иметь учет основных складски операций – поступление и продажа товаров. Учет остатков на складе.

Система должна иметь эргономичный, а также интуитивно понятный интерфейс. Это нужно для обеспечения низкого уровня входа в пользование данной системой сотрудников поликлиники. Кнопки вызова функций системы должны быть доступны только в тех состояниях системы, когда

вызов функций имеет смысл. Таким образом мы уменьшим вероятность ошибки со стороны пользователя. Цветовая гамма системы не должна перенапрягать глаза пользователей, в связи с постоянным пользованием системы, что может сказаться на физическом состоянии работников.

Пользователями системы будут являться сотрудники регистратуры, врачи, а также фармацевты аптеки.

## 1.2 Анализ информационных систем аналогичного назначения

Перед разработкой информационной системы необходимо провести анализ аналогов. Это необходимо, во-первых, чтобы оценить объем работ, а во-вторых – для учета достоинств и недостатков существующих подобных систем.

Для сравнения были выбраны следующие информационные системы для автоматизации управления ветеринарной клиникой:

– БИТ.Айболит;
– Ветменеджер;
– VetDesk.

В качестве критериев оценивания информационных систем были выбраны следующие критерии:

– функциональная полнота;
– качество пользовательского интерфейса;
– быстродействие;
– уровень требований к комплексу технических средств;
– качество защиты данных от несанкционированного доступа;
– трудоемкость освоения и внедрения;
– стоимость.

Первой рассмотрим информационную систему БИТ.Айболит. Это программа для управления ветеринарной клиникой на базе 1С. Оснащена календарём планирования, который отображает записи пациентов к врачам. Программа оперативно определяет клиента, администратор может просматривать историю посещений. Это помогает быстро подбирать нужные

услуги и врачей, записывать пациента на приём за секунды.

Программа разработана на основе глубокого изучения потребностей ветеринарных клиник и является комплексным решением, охватывающим основные контуры управления и оперативного учета в ветеринарной клинике. Программное решение даёт возможность оперативно принимать заявки, планировать загрузку специалистов, предоставляет мощный инструмент работы с клиентской базой, повышая уровень сервиса и качества оказываемых услуг. Продукт дает общее информационное пространство для ведения отчетности, а также автоматизирует такие процессы как:

– регистратура – журнал записи питомцев;
– call-центр – организация IP-телефонии;
– медицинский учет – электронные медицинские карты;
– управление – доступ ко всем отчетам и показателям.

Продукт направлен на снижение издержек и повышение эффективности работы благодаря точности учёта материалов, товарных остатков, денежных средств и взаиморасчётов

Программа показывает, какую прибыль приносит клиника и насколько стабильно. Аналитика в программе поможет сделать фокус на услугах-локомотивах, чтобы бизнес приносил деньги. Аналитика по продажам, среднему чеку помогает составить прогнозный план роста на год в разрезе каждой специализации, врачам, услугам [3].

К недостаткам данной системы можно отнести:

– торможения системы во время работы;
– отсутствие возможности организовать аптеку при поликлинике.

Вторая рассматриваемая система это Ветменеджер. Это программа, подходящая для управления ветеринарной клиникой, аптекой и зоомагазином. Она имеет мобильное приложение, благодаря которому у врача под рукой всегда необходимая для работы информация: запланированные записи на прием, медицинские карты пациентов, счета. Это значительно экономит время врача на приеме. Имеется онлайн-запись на

прием. Клиенты могут записываться через интернет в любое время, не отвлекаясь от своих занятий.

Модули счета. Учет акций, скидки, надбавки и различные особенности (ночной или экстренный прием), карты лояльности. Возможность быстро и эффективно выставлять счета клиентам за выполненные услуги. Электронные медкарты. Быстрый поиск и добавление информации в базы данных, шаблоны осмотров и анализов. Имеются готовые лечебные схемы.

Имеется планировщик для ведения приема по записи. Включены онлайн-кассы и фискальные регистраторы АТОЛ, Эвотор, Checkbox. Учет товаров. Получение информации о состоянии склада, отслеживание сроков годности.

Система интегрирована с лабораторными системами Idexx и Zoetis (Abaxis). Ветменеджер позволяет настроить связь с лабораторными анализаторами Idexx и Zoetis и получать медицинские карты с результатами анализов [4].

Данная программа является облачной. Из этого следует что минусом будет являться зависимость от подключения к интернету. Ещё одним недостатком является сложность в освоении. Для полноценного освоения понадобится время.

Третьей рассматриваемой системой является VetDesk. VetDesk представляет из себя настольное приложение, полностью автоматизирующее документооборот ветеринарной клиники.

Программа позволяет полностью отказаться от бумажной картотеки клиентов и их питомцев, так же, как и от бумажных бланков результатов анализов, рекомендаций и назначений. Благодаря быстрому поиску открытие карточки питомца или нахождение нужного приёма, анализа или счета занимает считанные секунды даже в том случае, если база данных накопила существенный объем. Ведение счетов и статистика по ним поможет руководителю предприятия быть всегда в курсе финансовых показателей. Справочники, которые можно заполнять «на лету» помогут быстро и

правильно заполнить нужные формы и не забыть ничего существенного.

Приложение работает с различными документами, которые объединены в определенную структуру. Понимание этой структуры обеспечивает понимание принципов работы и особенностей интерфейса приложения.

Различные типы документов описывают различные сущности, такие как карта питомца, прием или операция и другие. Данные документов отображаются и редактируются с помощью соответствующих форм приложения. Документы логически связаны между собой по принципу «один ко многим». Другими словами, каждый тип документа может быть связан с одним или несколькими документами другого типа. Структура является строго типизированной, то есть определенный тип документа может быть связан только с другими определенными типом или типами и ни с какими другими.

Во избежание случайной потери данных и в соответствии с концепцией целостности данных приложение отслеживает связанные данные в соответствии со структурой и не дает удалить связанные данные, если документ имеет дочерние связи.

Например, если питомец имеет зарегистрированные приемы, а приемы в свою очередь содержат анализы, то невозможно сразу же удалить карточку питомца или прием. В случае такой попытки возникнет ошибка целостности данных [5].

Минусами данной системы являются отсутствие авторизации для различных пользователей. Интерфейс приложения интуитивно не понятен и сложен в изучении.

Согласно рассмотренным аналогам информационных систем ветеринарной клиники мне удалось составить сравнение этих систем.

Сравнение систем приведено в таблице 1.

Таблица 1 – Сравнение аналогов

| Критерий | БИТ.Айболит | Ветменеджер | VetDesk | Разрабатываемая система |
|---|---|---|---|---|
| Функциональная полнота | высокая | средняя | средняя | средняя |
| Качество пользовательского интерфейса | высокое | высокое | среднее | высокое |
| Быстродействие | среднее | низкое | высокое | высокое |
| Трудоемкость освоения и внедрения | средняя | средняя | высокая | низкая |
| Уровень требований к комплексу технических средств | высокий | средний | высокий | средний |
| Качество защиты данных от несанкционированного доступа | высокое | среднее | высокое | низкое |
| Стоимость | высокая | средняя | средняя | низкая |

В результате сравнительного анализа были выявлены достоинства и недостатки рассмотренных аналогов, а также составлена таблица сравнения.

### 1.3 Выбор технологий проектирования и разработки

Для проектирования автоматизированной системы ветеринарной

клиники были выбраны диаграммы нотации Visual Paradigm, а также CASE средства ERwin Data Modeler и Ramus Educational.

Visual Paradigm (VP-UML) – это свободно распространяемый инструмент UML CASE, поддерживающий UML 2, SysML и нотацию моделирования бизнес-процессов (BPMN) из группы управления объектами (OMG). В дополнение к поддержке моделирования, он предоставляет возможности генерации отчетов и разработки кода [6].

ERwin – средство разработки структуры базы данных (БД). ERwin сочетает графический интерфейс Windows, инструменты для построения ER-диаграмм, редакторы для создания логического и физического описания модели данных и прозрачную поддержку ведущих реляционных СУБД и настольных баз данных.

Возможны две точки зрения на информационную модель и, соответственно, два уровня модели. Первый – логический (точка зрения пользователя), который описывает данные, задействованные в бизнесе предприятия. Второй – физический, который определяет представление информации в БД. ERwin объединяет их в единую диаграмму, имеющую несколько уровней представления [7].

Ramus Educational – это программа, при помощи которой можно создавать визуальные диаграммы, используемы для наглядного отображения различных бизнес-процессов. Программа поддерживает методологии: DFD и IDEF0 [8].

Для разработки конфигурации выбрана платформа «1С: Предприятие».

Система программ «1С: Предприятие» состоит из технологической платформы (ядра) и разработанных на ее основе прикладных решений («конфигураций»). Такая архитектура системы принесла ей высокую популярность, поскольку обеспечивает открытость прикладных решений, их функциональность и гибкость, короткие сроки внедрения, высокую производительность, масштабируемость от одного до десятков тысяч рабочих мест, работу в режиме «облачного» сервиса и на мобильных

устройствах.

В основе системы программ «1С: Предприятие» лежит единая технологическая платформа. Она является фундаментом для построения всех прикладных решений. Наличие единой технологической платформы не просто облегчает создание отдельных прикладных решений и обеспечивает их невысокую стоимость [9].

Наличие единой технологической платформы и общей методологии позволяет создавать специализированные и индивидуальные решения на базе стандартных, добавляя в них только необходимые отличия, учитывающие специфику отрасли или конкретного предприятия.

Преимущества использования платформы «1С: Предприятие»:

– широкий выбор как универсальных, так и отраслевых продуктов для ведения деятельности в конкретной области, например, логистической, фармацевтической, производственной, строительной, торговой, финансовой и др.;

– каждый продукт учитывает российские реалии ведения бизнеса, нормы и требования действующего законодательства;

– присутствует поддержка одновременной работы большого количества пользователей без критичного снижения времени реакции системы;

– масштабируемость «1С: Предприятие» обеспечивает хороший «запас» по количеству пользователей и функциональности системы. Прикладное решение можно использовать в многопользовательском варианте без изменения прикладного решения;

– для «1С: Предприятие» не требуется приобретения дополнительных лицензий для работы с распределенными базами данных, так как функциональность механизмов обмена данными включена в платформу.

Вышеперечисленные пункты показывают причину выбора платформы «1С: Предприятие» для написания собственной конфигурации с необходимым функционалом.

## 2 Проектирование информационной системы

### 2.1 Анализ бизнес процессов предприятия

SADT (аббревиатура выражения Structured Analysis and Design Technique – методология структурного анализа и проектирования) – это методология, разработанная специально для того, чтобы облегчить описание и понимание искусственных систем, попадающих в разряд средней сложности.

Графический язык SADT организует естественный язык вполне определенным и однозначным образом, за счет чего SADT и позволяет описывать системы, которые до недавнего времени не поддавались адекватному представлению [10].

Набор структурных компонентов языка, их характеристики и правила, определяющие связи между компонентами, представляют собой синтаксис языка. Компоненты синтаксиса IDEF0 – блоки, стрелки, диаграммы и правила. Блоки представляют функции, определяемые как деятельность. Стрелки представляют данные или материальные объекты, связанные с функциями. Правила определяют, как следует применять компоненты [11].

На рисунке 1 представлена контекстная диаграмма функционала ветеринарной клиники в методологии IDEF0.

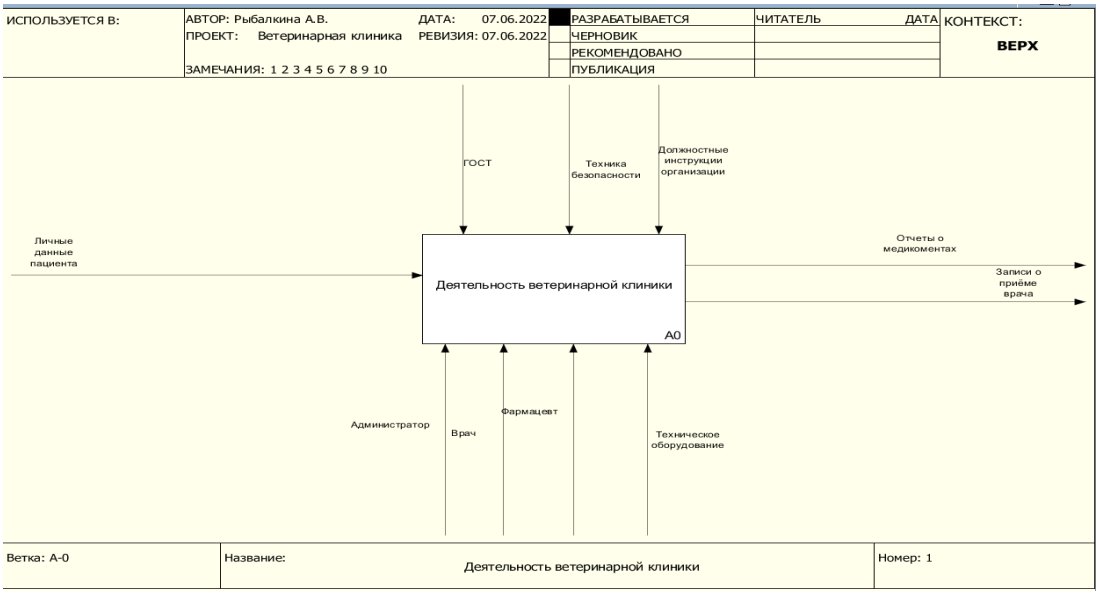



Рисунок 1 – Контекстная диаграмма IDEF0

Входными данными в данной контекстной диаграмме являются личные данные пациента. Выходными данными являются записи о приеме врача и отчеты о медикаментах. Ресурсами являются администратор, врач, фармацевт, техническое оборудование и пользователь. Механизмы управления - должностные инструкции и ГОСТ.

На основе созданной контекстной диаграммы в результате детализации обобщенного процесса «Деятельность ветеринарной клиники» была разработана детализирующая диаграмма, представленная на рисунке 2.

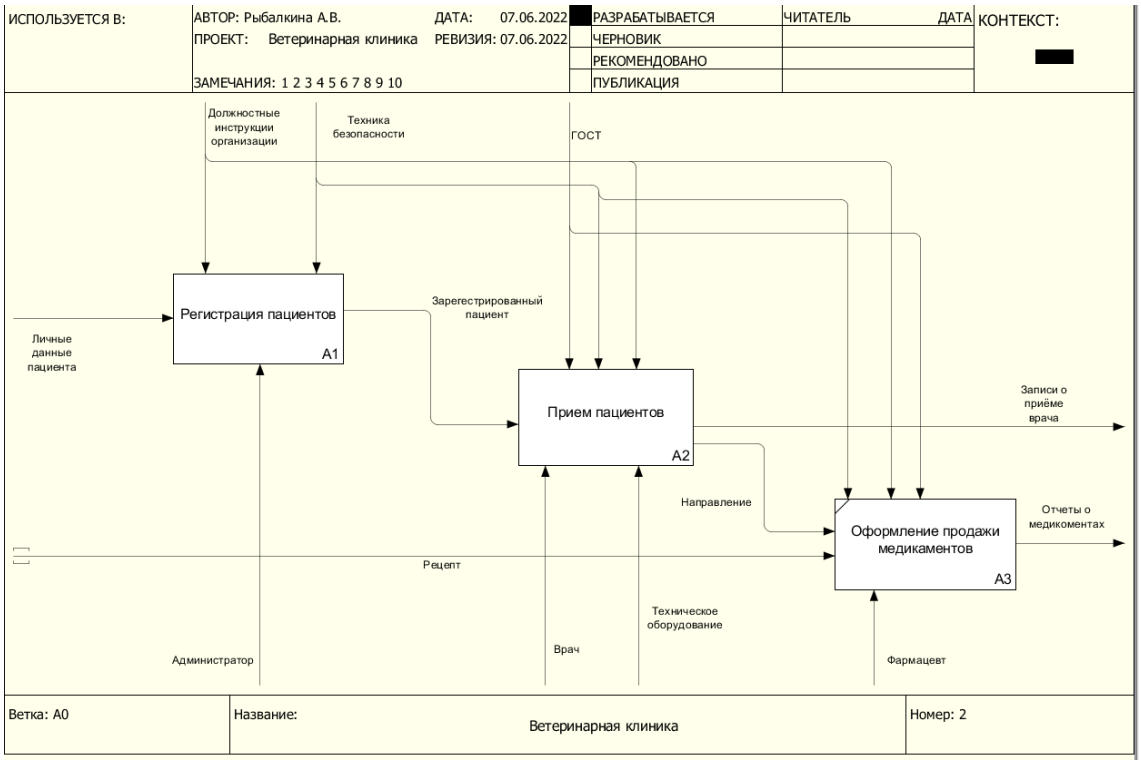



Рисунок 2 – Детализирующая диаграмма процесса «Деятельность ветеринарной клиники» в нотации IDEF0

В процессе декомпозиции системы «Деятельность ветеринарной клиники» выделены 3 процесса: «Регистрация питомцев», «Прием питомцев», «Оформление продажи медикаментов».

Процесс «Регистрация питомцев» предназначен для занесения данных о питомце и владельце в базу, а также оформлении карточки питомца и записи на прием к врачу или сдачу анализов.

Процесс «Прием питомцев» предназначен для внесения данных о приеме питомца, о его состоянии, о прогнозируемом лечении, а также заполнения формы заказа для аптеки при ветеринарной клинике.

Процесс «Оформление продажи» предназначен для оформления заказов медикаментов на прямую от покупателя, а также заказов, поступивших через врача.

Декомпозиция процесса «Регистрация питомцев» представлена на рисунке 3.

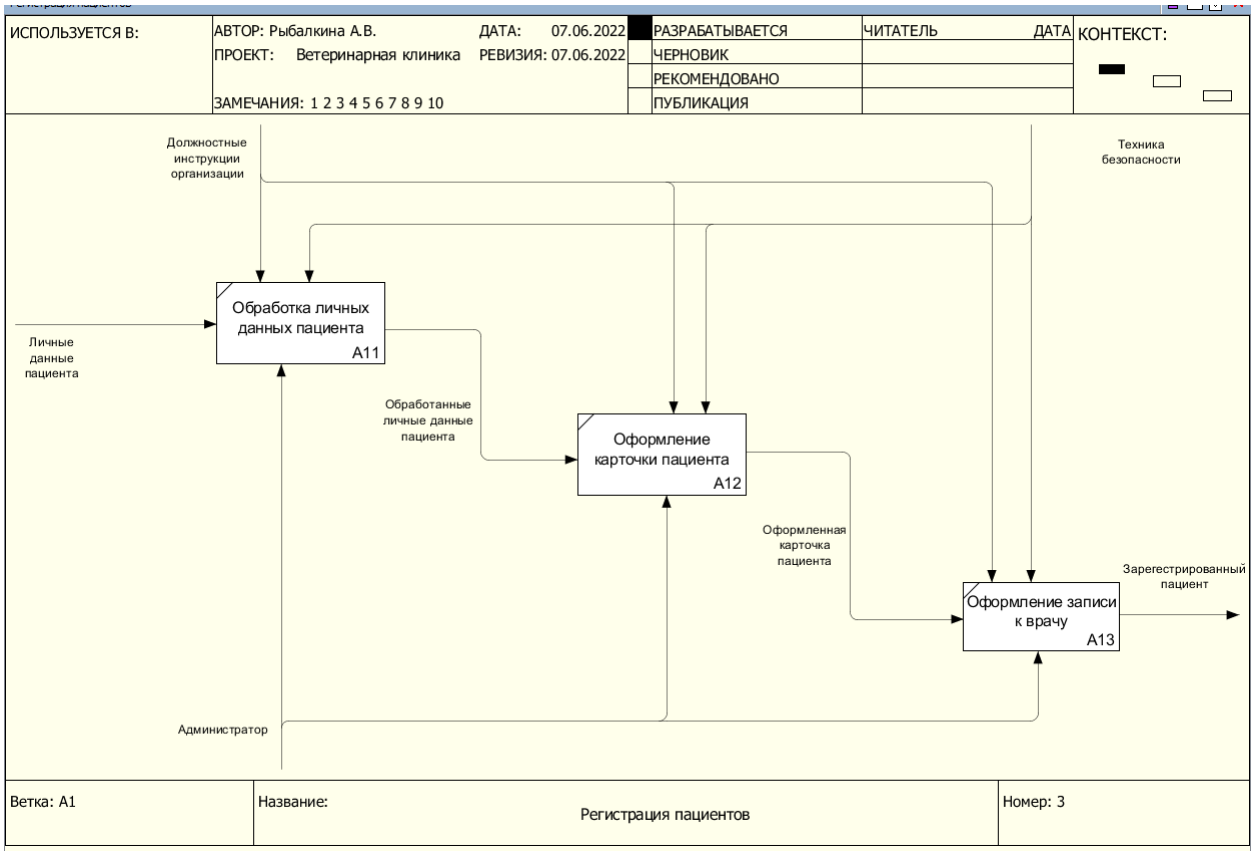



Рисунок 3 – Детализирующая диаграмма процесса «Регистрация питомцев»

При декомпозиции были выделены три процесса:

– процесс «Обработка личных данных пациента» предназначен для занесения данных о клиенте в систему;

– процесс «Оформление карточки пациента» предназначен для оформления амбулаторной карты пациента;

– процесс «Оформление записи к врачу» предназначен для записи пациента на прием.

Декомпозиция процесса «Оформление продажи медикаментов» представлена на рисунке 4.

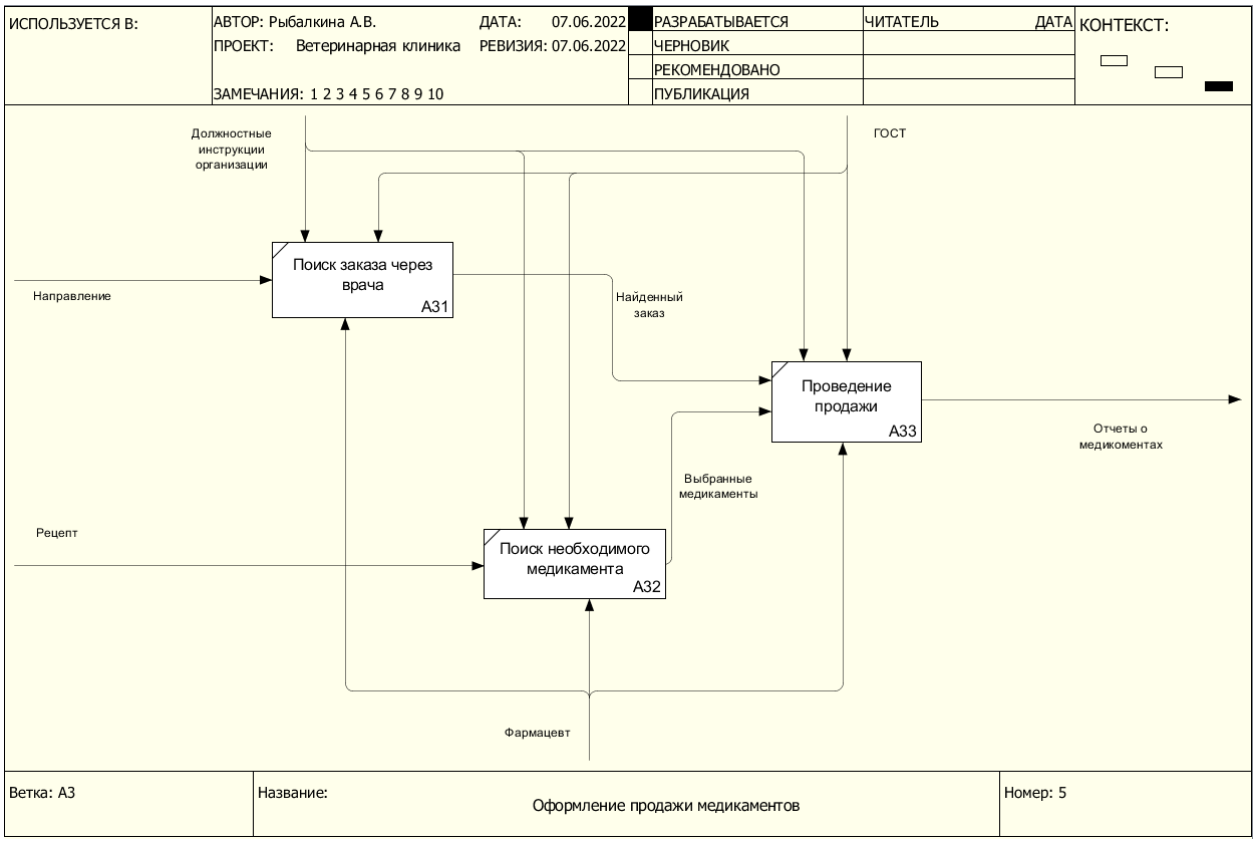



Рисунок 4 – Детализирующая диаграмма процесса «Оформление продажи медикаментов»

При декомпозиции были выделены следующие процессы:

– процесс «Поиск заказа через врача» предназначен для поиска заказа, поступившего через врача;

– процесс «Поиск медикамента» предназначен для выбора медикамента необходимого клиенту;

– процесс «Проведение продажи» предназначен для проведения документа продажи медикамента.

## 2.2 Проектирование логической модели базы данных

Логический уровень представления данных – уровень представления данных в виде структуры данных, к которым относятся иерархические,

сетевые структуры и другие виды структур, используемые при организации данных в вычислительных системах [12].

Проектирование логической модели было проведено с помощью CASE-средства ERwin Data Modeler. Средством разработки является 1С: Предприятие, поэтому физическая модель не требуется.

Таким образом спроектировали логическую модель базы данных для информационной систему управления ветеринарной клиникой. Данная модель содержит 10 таблиц, связанных между собой логическими связями.

На рисунке 5 представлена логическая модель базы данных.

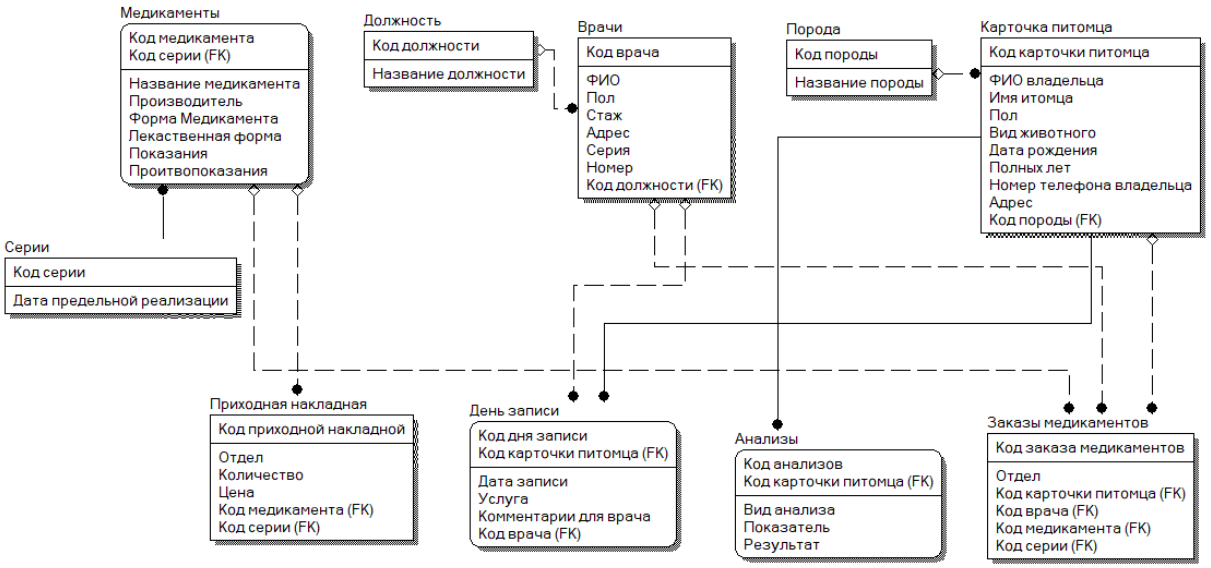



Рисунок 5 – Логическая модель данных

Описание сущностей и их атрибутов логической модели приведены в таблице 2.

Таблица 2 – Описание сущностей и их атрибутов логической модели

| Сущность | Атрибуты | Роль атрибута | Описание атрибута |
|---|---|---|---|
| Медикаменты | Код медикамента | PK | Уникальный код медикамента в системе |
| | Код серии | FK | Уникальный код серии в системе |
| | Название медикамента | | Название медикамента |
| | Производитель | | Производитель медикамента |
| | Форма медикамента | | Форма пользования медикамента |

Продолжение таблицы 2

| Сущность | Атрибуты | Роль атрибута | Описание атрибута |
|---|---|---|---|
| | Лекарственная форма | | Лекарственная форма медикамента |
| | Показания | | Показания к применению |
| | Противопоказания | | Противопоказания |
| Серии | Код серии | PK | Уникальный код серии в системе |
| | Дата предельной реализации | | Дата предельной реализации медикамента |
| Приходная накладная | Код приходной накладной | PK | Уникальный код приходной накладной в системе |
| | Код медикамента | FK | Уникальный код медикамента в системе |
| | Код серии | FK | Уникальный код серии в системе |
| | Отдел | | Отдел |
| | Количество | | Количество медикаментов |
| | Цена | | Цена за медикамент |
| Врачи | Код врача | PK | Уникальный код врача в системе |
| | ФИО | | ФИО |
| | Пол | | Пол |
| | Стаж | | Стаж |
| | Адрес | | Адрес |
| | Серия | | Серия паспорта |
| | Номер | | Номер паспорта |
| | Код должности | FK | Уникальный код должности в системе |
| Должность | Код должности | PK | Уникальный код должности в системе |
| | Название должности | | Название должности |
| День записи | Код дня записи | PK | Уникальный код дня записи в системе |
| | Код карточки питомца | FK | Уникальный код карточки питомца |
| | Дата записи | | Дата записи |
| | Услуга | | Название услуги |
| | Комментарий для врача | | Комментарий для врача |
| | Код врача | FK | Уникальный код врача в системе |
| Анализы | Код анализов | PK | Уникальный код анализов в системе |
| | Код карточки питомца | FK | Уникальный код карточки питомца |
| | Вид анализа | | Вид анализа |
| | Показатель | | Показатель анализа |
| | Результат | | Результат анализа |
| Карточка питомца | Код карточки питомца | PK | Уникальный код карточки питомца |
| | ФИО владельца | | ФИО владельца |

Окончание таблицы 2

| Сущность | Атрибуты | Роль атрибута | Описание атрибута |
|---|---|---|---|
| | Имя питомца | | Имя питомца |
| | Пол | | Пол |
| | Вид животного | | Вид животного |
| | Дата рождения | | Дата рождения |
| | Полных лет | | Полных лет |
| | Номер телефона владельца | | Номер телефона владельца |
| | Код карточки питомца | PK | Уникальный код карточки питомца |
| | Адрес | | Адрес проживания |
| | Код породы | FK | Уникальный код породы в системе |
| Заказы медикаментов | Код заказа медикаментов | PK | Уникальный код заказа медикаментов в системе |
| | Отдел | | Отдел |
| | Код карточки питомца | FK | Уникальный код карточки питомца |
| | Код врача | FK | Уникальный код врача в системе |
| | Код медикамента | FK | Уникальный код медикамента в системе |
| | Код серии | FK | Уникальный код серии в системе |
| Порода | Код породы | PK | Уникальный код породы в системе |
| | Название породы | | Название породы |

Данная модель является нормализованный, так как содержит простые атрибуты и каждый не ключевой атрибут зависит от идентификатора, а также находится в третьей нормальной форме.

## 2.3 Проектирование ПО

Назначение проектирования состоит в определении внутренних свойств системы и детализации её внешних (видимых) свойств на основе выданного заказчиком технического задания к программному обеспечению (ПО). Это техническое задание подвергается анализу.

Проектирование ПО включает следующие основные виды деятельности:

- выбор метода и стратегии решения;
- выбор представления внутренних данных;
- разработка основного алгоритма;

– документирование ПО;

– тестирование и подбор тестов;

– выбор представления входных данных.

Проектирование системы осуществляется с использованием UML-диаграмм с помощью CASE-средства Visual Paradigm.

### 2.3.1 Анализ прецедентов

Диаграмма вариантов использования (диаграмма прецедентов) – это исходное концептуальное представление или концептуальная модель системы в процессе ее проектирования и разработки.

Назначение данной диаграммы состоит в следующем: проектируемая программная система представляется в форме так называемых вариантов использования (прецедентов), с которыми взаимодействуют внешние сущности или актеры. При этом актером или действующим лицом называется любой объект, субъект или система, взаимодействующая с моделируемой бизнес-системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая служит источником воздействия на моделируемую систему так, как определит разработчик. Каждый прецедент определяет набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой и собственно выполнение прецедентов.

В общем случае диаграмма вариантов использования представляет собой граф специального вида, который является графической нотацией для представления конкретных вариантов использования, актеров и отношений между этими элементами. При этом отдельные элементы диаграммы заключают в прямоугольник, который обозначает границы проектируемой системы [13].

Диаграмма прецедентов представлена на рисунке 6.

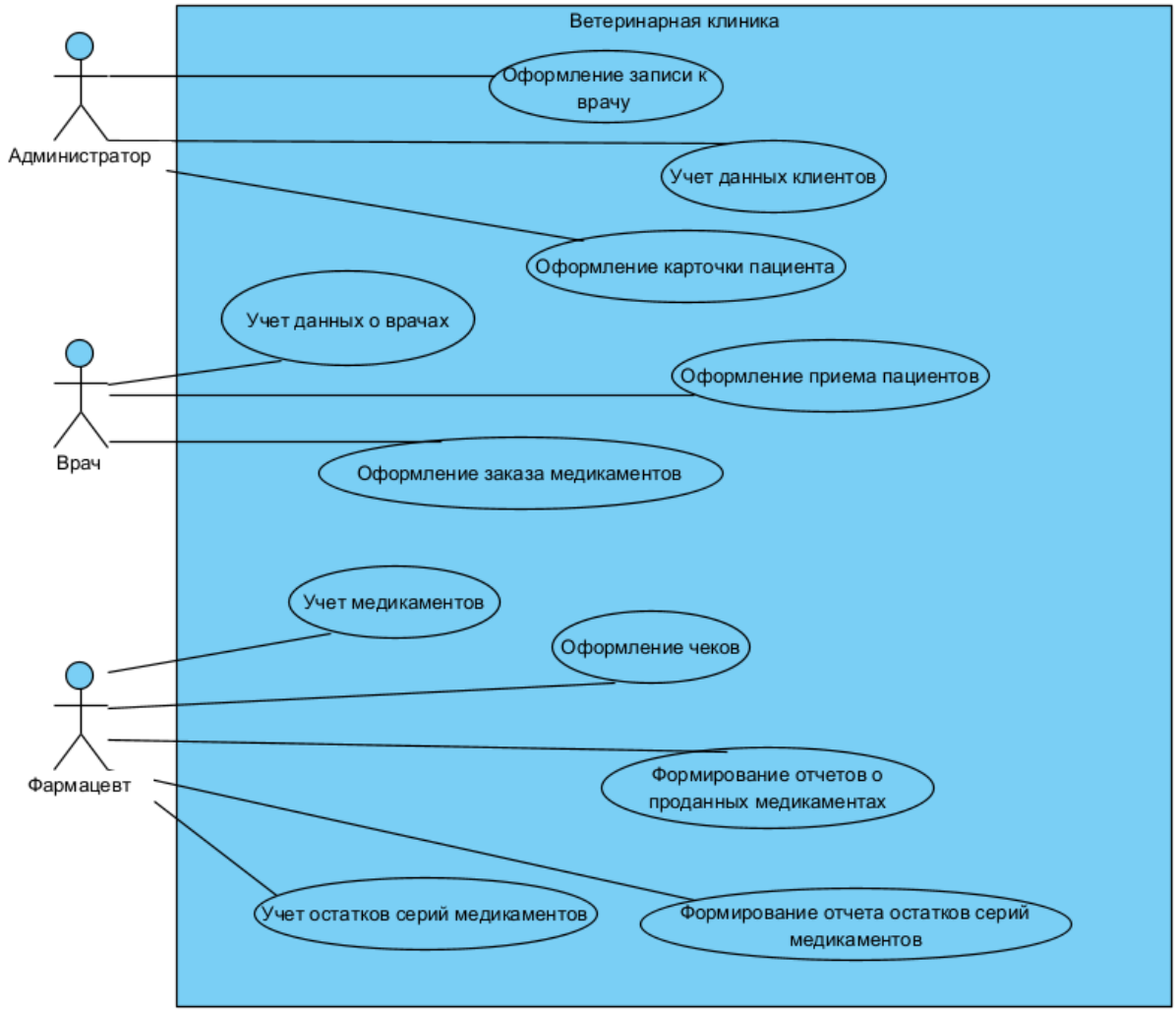



Рисунок 6 – Диаграмма прецедентов

Описание действующих лиц представлено в таблице 3, а краткое описание вариантов использования в таблице 4.

Таблица 3 – Список действующих лиц

| Наименование лица | Профиль, подготовка и навыки |
|---|---|
| Администратор | Регулярный пользователь информационной системы, большой опыт работы с подобным ПО. Может модифицировать интерфейс. |
| Врач | Регулярный пользователь информационной системы, малый опыт работы с подобным ПО. Вносить изменения и корректировать работу приложения не может. |
| Фармацевт | Регулярный пользователь информационной системы, средний опыт работы с подобным ПО. Вносить изменения и корректировать работу приложения не может. |

Таблица 4 – Описание вариантов использования

| Действующее лицо | Цель | Краткое описание |
|---|---|---|
| Администратор | Оформление записи к врачу | Администратор может добавлять, изменять и удалять записи к врачу |
| | Учет данных клиентов | Администратор может добавлять данные о клиентах |
| | Оформление карточки пациента | Администратор может изменять и удалять данные о клиентах |
| Врач | Учет данных о врачах | Врач может добавлять и изменять данные о врачах |
| | Оформление заказа медикаментов | Врач может добавлять заказ на медикаменты |
| | Оформление приема пациентов | Врач может добавлять и изменять данные о приеме пациента |
| Фармацевт | Учет медикаментов | Фармацевт может добавлять или изменять данные о медикаментах |
| | Оформление чеков | Фармацевт может добавлять изменять и удалять данные в чеке |
| | Формирование отчетов о проданных медикаментах | Фармацевт формирует отчеты проданных медикаментов |
| | Формирование отчета остатков серий медикаментов | Фармацевт формирует отчет остатков серий медикаментов |
| | Учет остатков серий медикаментов | Фармацевт может добавлять изменять или удалять серии медикаментов |

В результате анализа, были выявлены и описаны прецеденты системы, а также построена диаграмма вариантов использования.

### 2.3.2 Диаграмма состояний

Диаграмма состояний показывает, как объект переходит из одного состояния в другое. Очевидно, что диаграммы состояний служат для

моделирования динамических аспектов системы (как и диаграммы последовательностей, кооперации, прецедентов и, как мы увидим далее, диаграммы деятельности). Часто можно услышать, что диаграмма состояний показывает автомат, но об этом мы поговорим подробнее чуть позже. Диаграмма состояний полезна при моделировании жизненного цикла объекта.

От других диаграмм диаграмма состояний отличается тем, что описывает процесс изменения состояний только одного экземпляра определенного класса - одного объекта, причем объекта реактивного, то есть объекта, поведение которого характеризуется его реакцией на внешние события [14].

Диаграмма состояний представлена на рисунке 7.

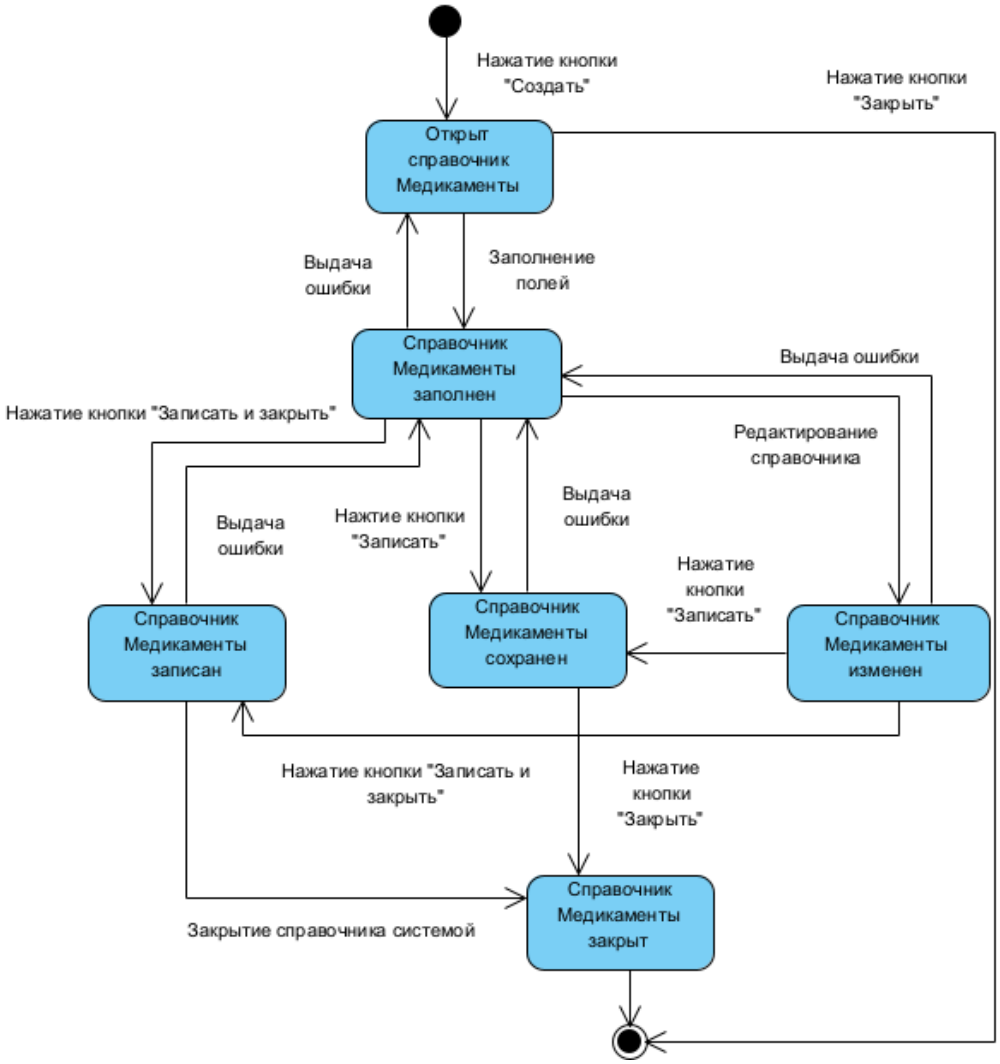



Рисунок 7 – Диаграмма состояний

Описание диаграммы состояний представлено в таблице 5.

Таблица 5 – Описание диаграммы состояний

| Состояние | Краткое описание | Вызывающие события |
|---|---|---|
| Открыт справочник Медикаменты | Открыт справочник Медикаменты для дальнейших действий | Нажатие кнопки «Создать» |
| Справочник Медикаменты заполнен | Поля справочника медикаменты заполнены | Заполнение полей |
| Справочник Медикаменты изменен | Поля справочника Медикаменты изменены | Редактирование справочника |
| Справочник Медикаменты сохранен | Сохранение справочника Медикаменты не проведенным | Нажатие кнопки «Записать» |
| Справочник Медикаменты записан | Проведение справочника Медикаменты | Нажатие кнопки «Записать и закрыть» |
| Справочник Медикаменты закрыт | Справочник Медикаменты закрыт | Закрытие справочника системой |

Диаграмма состояний помогает описать поведение отдельно взятого объекта.

### 2.3.3 Диаграмма деятельности

Диаграммы деятельности – это технология, позволяющая описывать логику процедур, бизнес-процессы и потоки работ. Во многих случаях они напоминают блок схемы, но принципиальная разница между диаграммами деятельности и нотацией блок схем заключается в том, что первые поддерживают параллельное процессы.

Диаграмма деятельности позволяет любому, кто выполняет данный процесс, выбирать порядок действий. Другими словами, диаграмма только

устанавливает правила обязательной последовательности действий, которым я должен следовать. Это важно для моделирования бизнес-процессов, поскольку эти процессы часто выполняются параллельно [15].

Диаграмма деятельности представлена на рисунке 8.

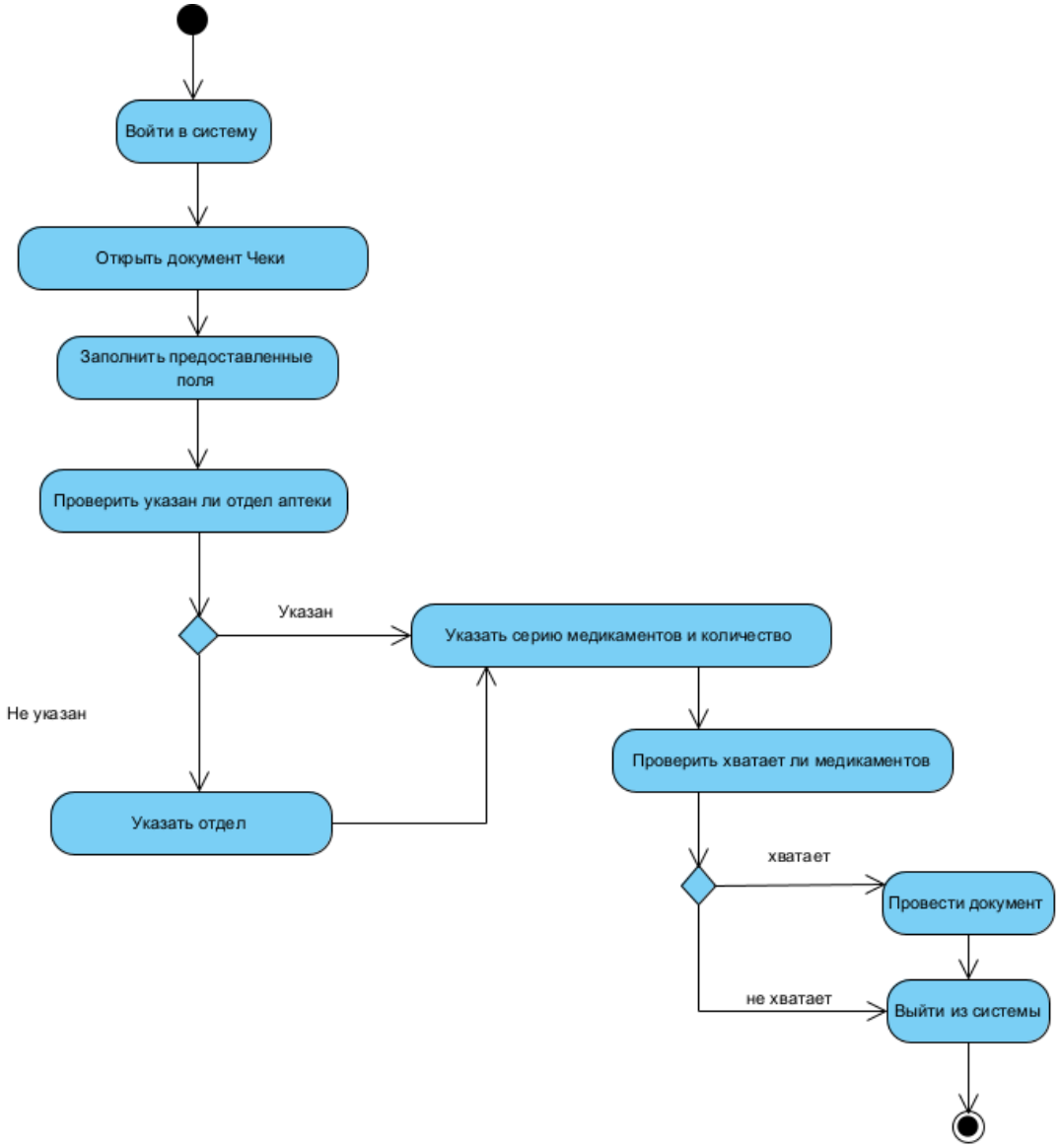



Рисунок 8 – Диаграмма деятельности

Описание диаграммы состояний представлено в таблице 6.

Таблица 6 – Описание диаграммы деятельности

| Название узла | Тип узла | Описание | Входные потоки | Выходные потоки |
|---|---|---|---|---|
| Войти в систему | Узел действия | Вход в систему | - | Открыть документ Чеки |

Продолжение таблицы 6

| Название узла | Тип узла | Описание | Входные потоки | Выходные потоки |
|---|---|---|---|---|
| Открыть документ Чеки | Узел действия | Открытие документа Чеки | Войти в систему | Заполнить предоставленные поля |
| Заполнить предоставленные поля | Узел действия | Заполнение полей в документе | Открыть документ Чеки | Проверить указан ли отдел аптеки |
| Проверить указан ли отдел аптеки | Узел действия | Проверка указан ли отдел | Заполнить предоставленные поля | Указать отдел/ Указать серию медикаментов и количество |
| Указать отдел | Узел действия | Выбрать отдел | Проверить указан ли отдел аптеки | Указать серию медикаментов и количество |
| Указать серию медикаментов и количество | Узел действия | Выбрать медикамент его серию и указать количество | Проверить указан ли отдел аптеки | Проверить хватает ли медикаментов |
| Проверить хватает ли медикаментов | Узел действия | Проверка достаточно ли медикаментов в наличии | Указать серию медикаментов и количество | Провести документ/Выйти из системы |
| Провести документ | Узел действия | Проведение документа | Проверить хватает ли медикаментов | - |
| Выйти из системы | Узел действия | Выход из системы | Проверить хватает ли медикаментов | - |

Диаграмма деятельности показывает логику процедур, бизнес-процессы и потоки работ.

### 2.3.4 Диаграмма последовательности

Диаграмма последовательности – UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) информационной системы в рамках прецедента [16].

Диаграмма последовательности представлена на рисунке 9.

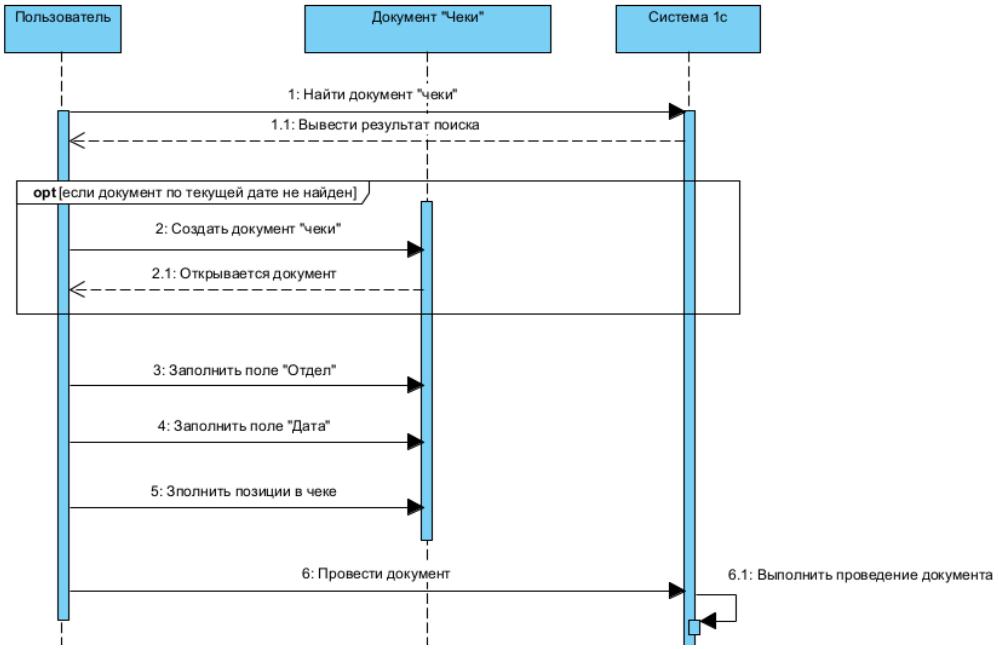



Рисунок 9 – Диаграмма последовательности

Описание линий жизни и сообщений диаграммы последовательности представлено в таблицах 7 и 8.

Таблица 7 – Линии жизни диаграммы последовательности

| Линия жизни | Описание |
|---|---|
| Пользователь | Человек который пользуется системой |
| Документ Чеки | Документ в котором производятся действия |
| Система 1С | Система в которой работает пользователь |

Таблица 8 – Сообщения диаграммы последовательности

| Сообщение | Описание | Инициатор | Получатель | Тип/Условие |
|---|---|---|---|---|
| Найти документ Чеки | Найти нужный документ | Пользователь | Система 1С | Вызов |
| Вывести результат поиска | Вывод результата | Система 1С | Пользователь | Ответ |
| Создать документ Чеки | Создать новый документ чеки | Пользователь | Документ Чеки | Вызов |
| Открывается документ | Открытие документа | Документ Чеки | Пользователь | Ответ |
| Заполнить поле Отдел | Заполнение поля Отдел | Пользователь | Документ Чеки | Вызов |
| Заполнить поле Дата | Заполнение поля Дата | Пользователь | Документ Чеки | Вызов |
| Заполнить позиции в чеке | Заполнение позиций в чеке | Пользователь | Документ Чеки | Вызов |
| Провести документ | Проведение документа | Пользователь | Документ Чеки | Вызов |

На диаграмме последовательности объекты в основном представляют экземпляры класса или сущности, обладающие поведением.

### 2.3.5 Диаграмма кооперации

Кооперация (collaboration) – спецификация множества объектов отдельных классов, совместно взаимодействующих с целью реализации отдельных вариантов использования в общем контексте моделируемой системы [17].

Диаграмма последовательности представлена на рисунке 10.

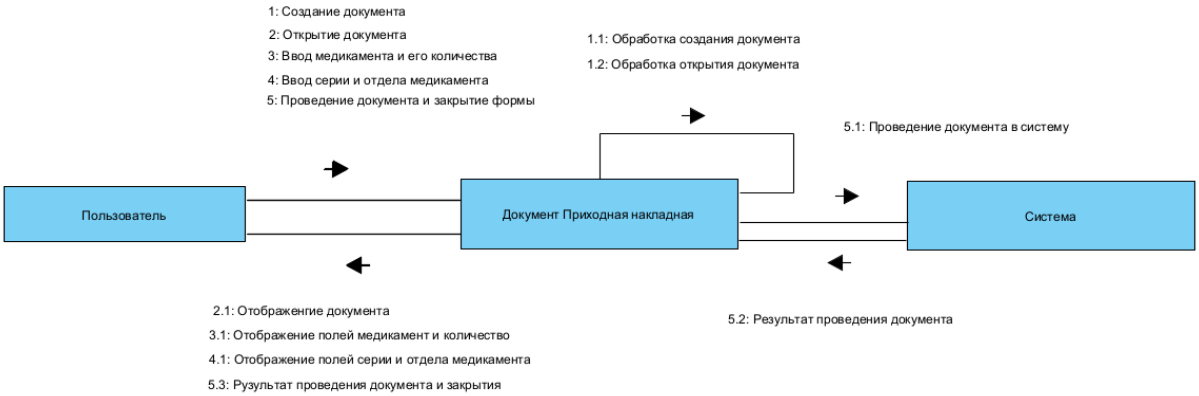



Рисунок 10 – Диаграмма кооперации

Описание объектов и связей диаграммы кооперацій представлено в таблице 9.

Таблица 9 – Объекты и связи диаграммы кооперацій

| Название компонента | Описание | Входные потоки | Выходные потоки |
|---|---|---|---|
| Пользователь | Исполнитель действий над системой | 2.1: Отображение документа<br>3.1: Отображение полей медикамент и количество<br>4.1: Отображение полей серии и отдела медикаментов<br>5.3: Результат проведения документа и закрытые | 1: Создание документа<br>2: Открытие документа<br>3: Ввод медикамента и его количества<br>4: Ввод серии и отдела медикамента<br>5: Проведение документа и закрытие формы |
| Документ Приходная накладная | Интерфейс взаимодействия пользователя и системы | 1.1: Обработка создания документа<br>1.2: Обработка открытия документа<br>5.2: Результат проведения документа<br>1: Создание документа | 1.1: Обработка создания документа<br>1.2: Обработка открытия документа<br>5.1: Проведение документа в систему |

Продолжение таблицы 9

| Название компонента | Описание | Входные потоки | Выходные потоки |
|---|---|---|---|
| | | 2: Открытие документа<br>3: Ввод медикамента и его количества<br>4: Ввод серии и отдела медикамента<br>5: Проведение документа и закрытие формы | 2.1: Отображение документа<br>3.1: Отображение полей медикамент и количество<br>4.1: Отображение полей серии и отдела медикаментов<br>5.3: Результат проведения документа и закрытые |
| Система | Исполнитель команд пользователя | 5.1:Провдеение документа в систему | 5.2: Результат проведения документа |

Кооперация определяет структуру поведения системы в терминах взаимодействия участников этой кооперации.

### 2.4 Проектирование интерфейса

Пользовательский интерфейс (ПИ) представляет собой совокупность программных и аппаратных средств, осуществляющих взаимодействие пользователя с информационной системой. ПИ включает в себя систему меню, диалоговые формы, сообщения об ошибках, справочную систему [18].

Концепция пользовательского интерфейса системы 1С: Предприятие 8 ориентирована на комфортную эффективную работу и соответствует современным тенденциям.

Панель разделов – это наиболее крупное разделение функциональности прикладного решения. Она расположена в верхней части основного окна и соответствует верхнему уровню подсистем, добавленных в конфигурацию. С ее помощью осуществляется переход к другим разделам программы.

Панель функций текущего раздела содержит самые востребованные и часто используемые команды, позволяющие просматривать информацию в

списках, быстро создавать новые объекты, выполнять типовые обработки или строить популярные отчеты.

Панель открытых предназначена для частого переключения между открытыми формами. Каждой открытой форме соответствует отдельная закладка.

Главная панель предназначена для быстрого доступа к основным функциям прикладного решения: меню функций, глобальному поиску, центру оповещений, истории, избранному, к текущему пользователю и главному меню – подробнее [19].

Пользовательский интерфейс разрабатываемой программы показан на рисунке 11.

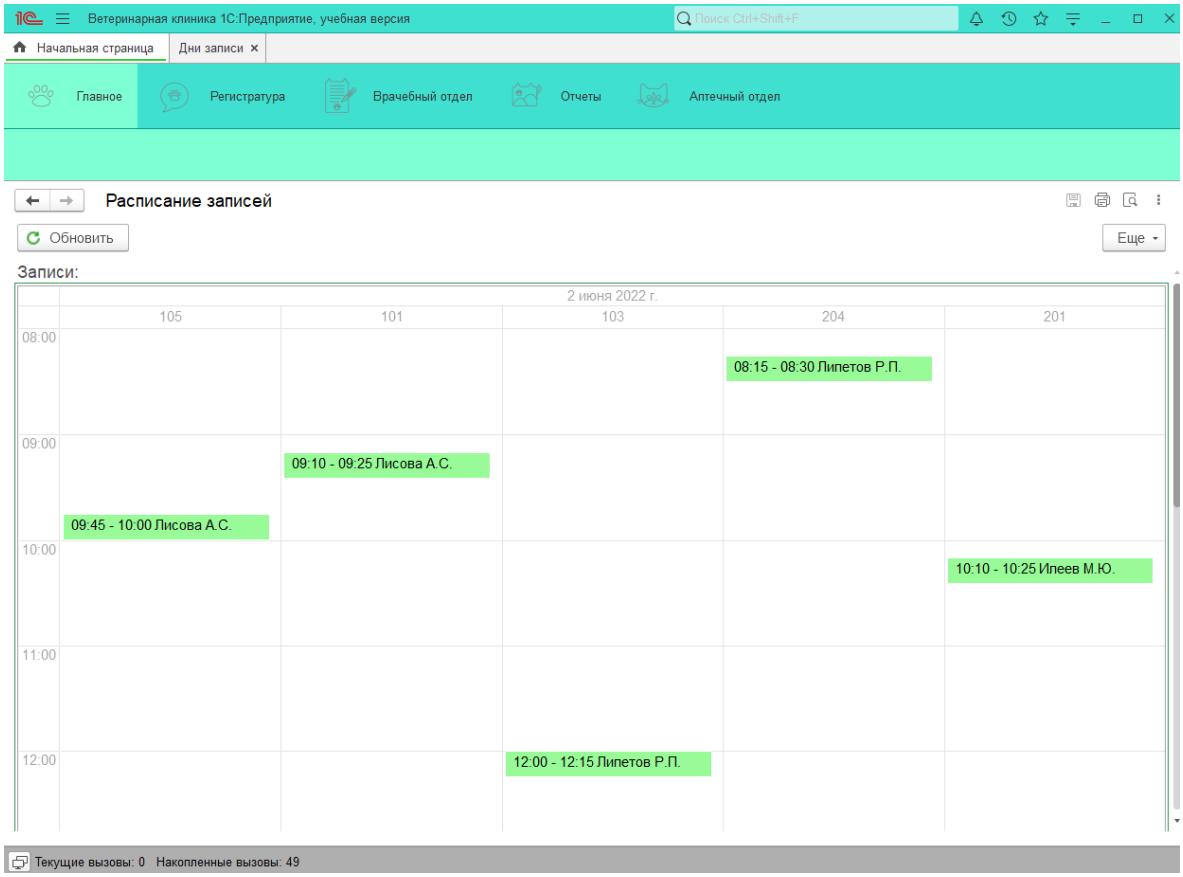



Рисунок 11 – Интерфейс программы

Раздел «Регистратура» содержит справочники и документы, необходимые для регистрации питомцев и их владельцев в базе данных, а затем непосредственно записи на прием или сдачу анализов: «Дни записи», «Кабинет», «Услуги», «Карточка питомца».

Раздел «Врачебный отдел» включает справочники и документы необходимые для приема питомцев и оформления результатов анализов: «Анализы», «Врач», «Карточка питомца».

Раздел «Аптечный отдел» содержит справочники и документы, необходимые для реализации движения медикаментов, их приход и продажу: «Заказы медикаментов», «Приходная накладная», «Чеки».

Раздел «Отчеты и справочники» хранит все отчеты организации, а также основную информацию о ее деятельности: «Вид животного», «Должность», «Единицы измерения», «Кабинет», «Лекарственная форма», «Отделы», «Пол», «Порода», «Серии», «Услуги», «Форма медикамента», «Ведомость остатков», «Список врачей».

### 2.5 Планирование разработки и оценки бюджета

Термин «управление проектом» применяется для обозначения успешного управления изменениями некоторой системы. Под успешным управлением подразумевается такое управление, когда оно обеспечивает достижение всех намеченных целей системы (согласно сформулированным требованиям) [20].

Диаграмма Ганта (a Gantt Chart) – это визуальный способ отображения запланированных задач. Это удобный способ показать, какая работа планируется к выполнению в определенный день и время. Gantt Charts также помогают командам контролировать даты начала и окончания любого проекта [21].

В процессе планирования было выделено 12 задач. Задачи проекта представлены в таблице 10.

Таблица 10 – Задачи проекта

| Номер | Название задачи | Длительность (дни) |
|---|---|---|
| 1 | Изучение предметной области | 3 |
| 2 | Поиск готовых решений | 1 |

Продолжение таблицы 10

| Номер | Название задачи | Длительность (дни) |
|---|---|---|
| 3 | Сравнительный анализ готовых решений | 1 |
| 4 | Анализ функциональных требований | 2 |
| 5 | Анализ нефункциональных требований | 2 |
| 6 | Выбор средств разработки | 1 |
| 7 | Проектирование информационной системы управления ветеринарной клиники | 3 |
| 8 | Реализация системы ветеринарной клиники | 20 |
| 8 | Тестирование методом причинно-следственных связей | 1 |
| 10 | Тестирование интерфейса | 1 |
| 11 | Вычисление метрик кода | 2 |
| 12 | Составление отчета | 3 |

В ProjectLibre создадим проект с начальной датой 12.04.2022. Занесем выделенные нами задачи в проект.

Ресурсы необходимые для выполнения каждой задачи проекта представлены в таблице 11.

Таблица 11 – Ресурсы проекта

| Номер | Название ресурса | Ставка |
|---|---|---|
| 1 | Программист | 240 руб./час |
| 2 | Тестировщик | 120 руб./час |

Распределение ресурсов по задачам представлено на рисунке 12.

| Название | Продолжит... | Начало | Окончание | Предшествующие | Название ресурса |
|---|---|---|---|---|---|
| **Создание информационной системы уп** | 40 дней? | **12.04.22 8:00** | **06.06.22 17:00** | | |
| Изучение предметной области | 3 дней? | 12.04.22 8:00 | 14.04.22 17:00 | | Программист |
| Поиск готовых решений | 1 день? | 15.04.22 8:00 | 15.04.22 17:00 | 2 | Программист |
| Сравнительный анализ готовыхрешений | 1 день? | 18.04.22 8:00 | 18.04.22 17:00 | 3 | Программист |
| Анализ функциональных требований | 2 дней? | 19.04.22 8:00 | 20.04.22 17:00 | 4 | Программист |
| Анализ нефункциональных требований | 2 дней? | 21.04.22 8:00 | 22.04.22 17:00 | 5 | Программист |
| Выбор средств разработки | 1 день? | 25.04.22 8:00 | 25.04.22 17:00 | 6 | Программист |
| Проектирование информационной системы | 3 дней? | 26.04.22 8:00 | 28.04.22 17:00 | 7 | Программист |
| Реализация системы управления ветерина | 20 дней? | 29.04.22 8:00 | 26.05.22 17:00 | 8 | Программист |
| Тестирование методом причинно-следстве | 1 день? | 27.05.22 8:00 | 27.05.22 17:00 | 9 | Тестировщик |
| Тестирование интерфейса | 1 день? | 30.05.22 8:00 | 30.05.22 17:00 | 10 | Тестировщик |
| Вычисление метрик кода | 2 дней? | 31.05.22 8:00 | 01.06.22 17:00 | 11 | Тестировщик |
| Составление отчета | 3 дней? | 02.06.22 8:00 | 06.06.22 17:00 | 12 | Тестировщик |

Рисунок 12 – Список задач проекта

На рисунке 13 представлены ресурсы, занесенные в проект.

| Название | Тип | Ини... | Максим... | Стандартная ставка | Ставка сверхур... | Начислять | Основной к... |
|---|---|---|---|---|---|---|---|
| Программист | Работа | П | 100% | 240 руб./час | 0 руб./час | Пропорционально | Пятидневка |
| Тестировщик | Работа | Т | 100% | 120 руб./час | 0 руб./час | Пропорционально | Пятидневка |

Рисунок 13 – Ресурсы проекта

Диаграмма Ганта представлена на рисунке 14.

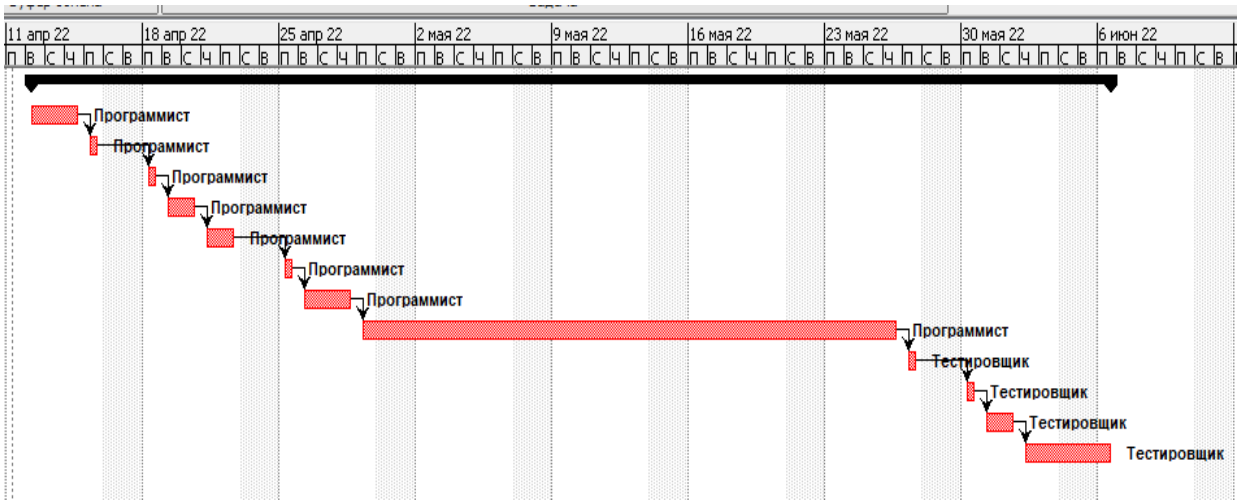



Рисунок 14 – Диаграмма Ганта

Загруженность ресурсов представлена на рисунке 15.

| 1 | Программист | |
|---|---|---|
| Код задачи | Задача | Работа |
| 2 | Изучение предметной области | 24 часов |
| 3 | Поиск готовых решений | 8 часов |
| 4 | Сравнительный анализ | 8 часов |
| 5 | Анализ функциональных | 16 часов |
| 6 | Анализ нефункциональных | 16 часов |
| 7 | Выбор средств разработки | 8 часов |
| 8 | Проектирование | 24 часов |
| 9 | Реализация системы | 160 часов |
| | | 264 часов |

| 2 | Тестировщик | |
|---|---|---|
| Код задачи | Задача | Работа |
| 10 | Тестирование методом | 8 часов |
| 11 | Тестирование интерфейса | 8 часов |
| 12 | Вычисление метрик кода | 16 часов |
| 13 | Составление отчета | 24 часов |
| | | 56 часов |

Рисунок 15 – Загруженность ресурсов

Автоматически в ProjectLibre была посчитана стоимость реализации проекта. Затраты на реализацию проекта представлены на рисунке 16.

| Costs | | | |
|---|---|---|---|
| Scheduled | 70080 руб. | Remaining | 70080 руб. |
| Baseline | 0 руб. | Actual | 0 руб. |
| | | Variance | 0 руб. |

Рисунок 16 – Затраты на реализацию задачи

Стоимость реализации проекта составляет 70 080 руб. Разработка проходила с 12 апреля 2022 года по 6 июня 2022 года. Разработка составляет 40 рабочих дней. Трудозатраты на реализацию составляют 320 часов.

# 3 Реализация системы

## 3.1 Реализация структур программного средства

Создавая проект в «1С: Предприятие 8.3» мы начинаем работу с конфигуратор. Мы создаем объекты конфигурации, которые отражают реальные объекты предметной области.

В ходе разработки информационной системы управления ветеринарной клиники были использованы следующие объекты конфигурации:

– справочники;

– документы;

– отчеты;

– регистры накопления;

– регистры сведений;

– перечисления.

Для наглядного представления структуры компонентов нашей системы построим диаграмму компонентов.

Диаграммы компонентов UML используются при моделировании физических аспектов объектно-ориентированных систем, которые используются для визуализации, определения и документирования систем на основе компонентов, а также для построения исполняемых систем с помощью прямого и обратного проектирования [22].

Диаграмма компонентов разрабатываемой системы представлена на рисунке 17.

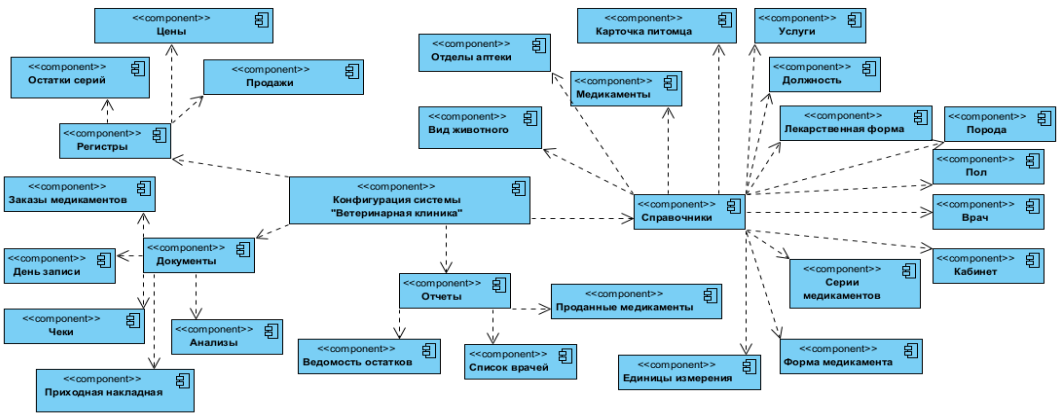



Рисунок 17 – Диаграмма компонентов

Подробнее рассмотрим структуру выделенных объектов.

Справочники – это прикладные объекты конфигурации. Они позволяют хранить в информационной базе данные, имеющие одинаковую структуру и списочный характер. Это может быть, например, список сотрудников, перечень товаров, список поставщиков или покупателей [23].

Для системы ветеринарной клиники были созданы справочники, приведенные ниже.

Справочник «Медикаменты» содержит информацию о медикаменте, его форме, производителе, показаниях и противопоказаниях. Форма элемента справочника «Медикаменты» представлена на рисунке 18.

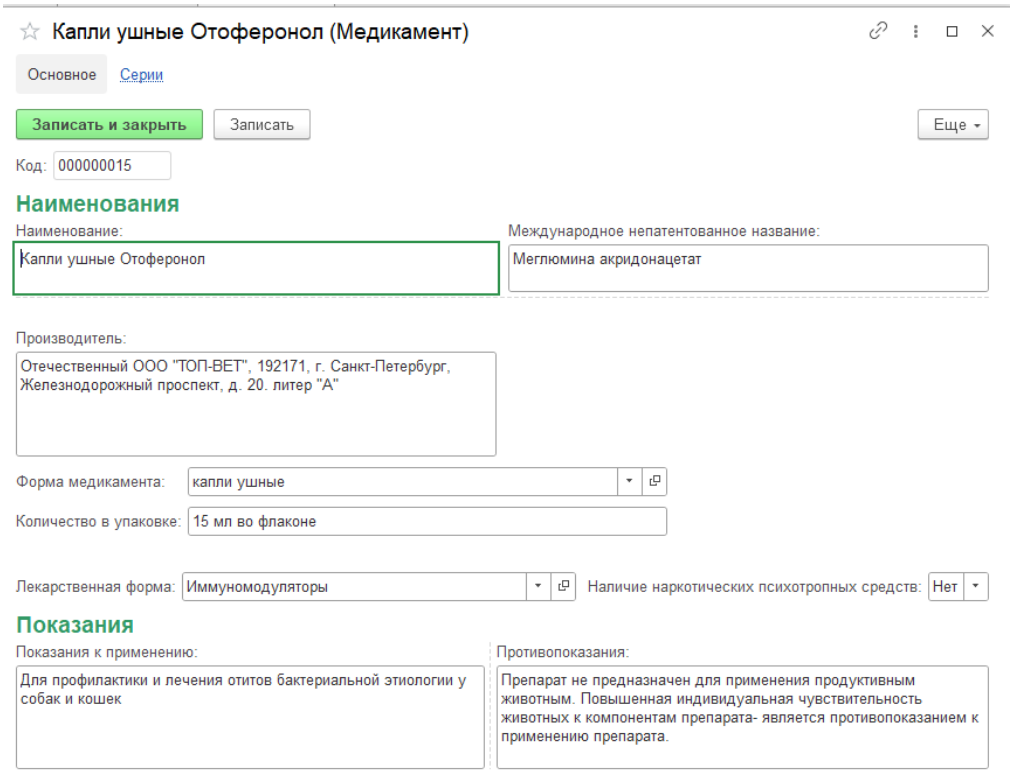



Рисунок 18 – Форма элемента справочника «Медикаменты»

Справочник «Серии» содержит информацию о сериях медикаментов и дату их предельной реализации. Форма элемента справочника «Серии» представлена на рисунке 19.

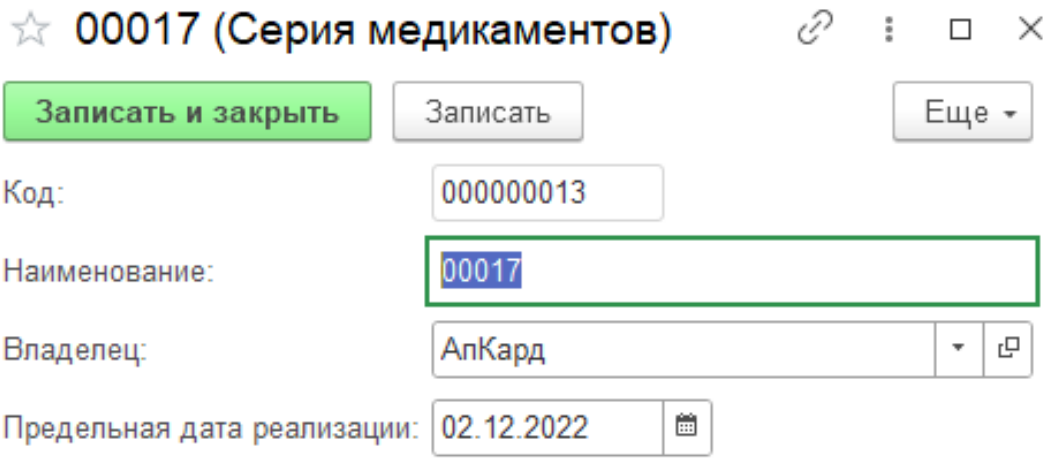



Рисунок 19 – Форма элемента справочника «Серии»

Справочник «Отделы» содержит информацию о существующих отделах хранения медикаментов. Форма списка справочника «Отделы» приведена на рисунке 20.

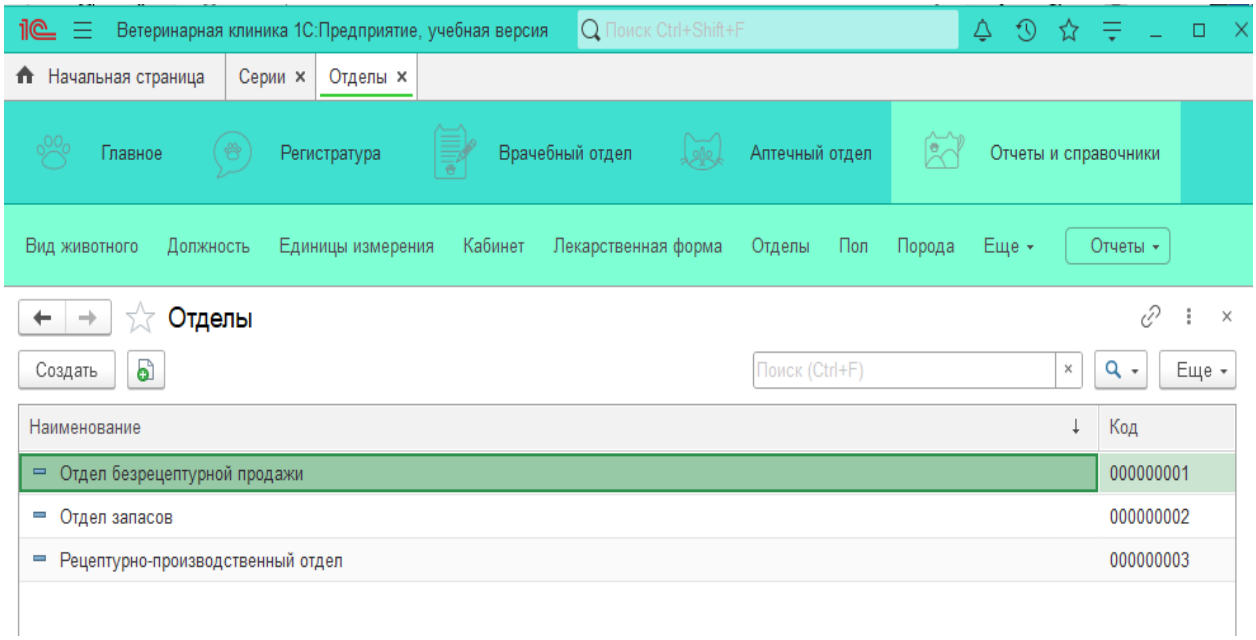



Рисунок 20 – Форма списка справочника «Отделы»

Справочник «Кабинет» содержит информацию о номерах кабинетов и какую услуг можно в них получить. Форма списка справочника «Кабинеты» представлена на рисунке 21.

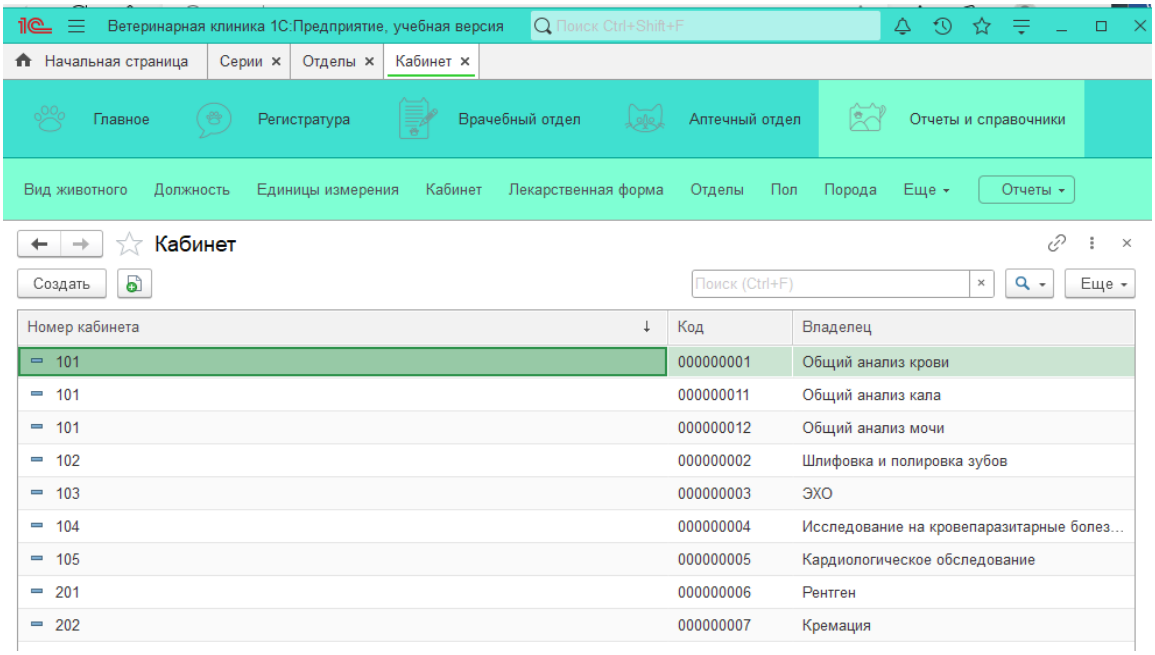



Рисунок 21 – Форма списка справочника «Кабинет»

Справочник «Врач» содержит информацию о врачах, их должность, стаж работы, а также место проживания, прописки и данные паспорта. Форма элемента справочника «Врач» представлена на рисунке 22.

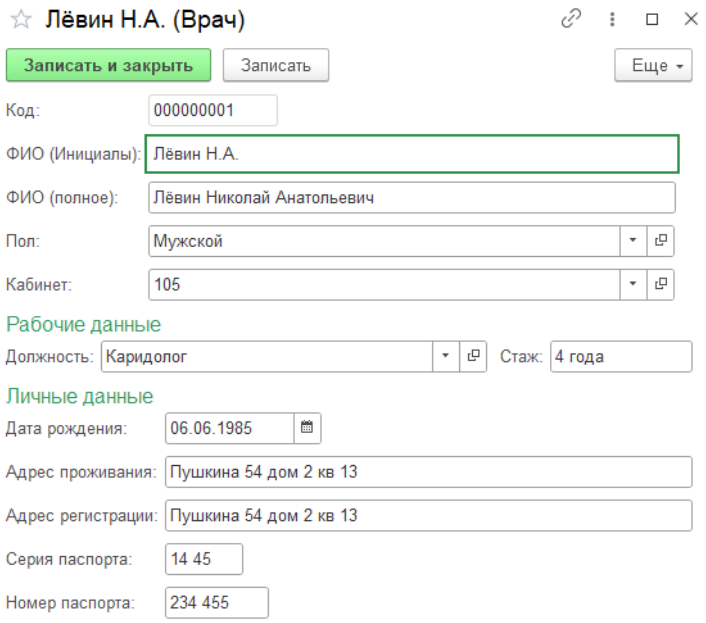



Рисунок 22 – Форма элемента справочника «Врач»

Справочник «Карточка питомца» содержит информацию о владельце и питомце. Также содержит историю болезни, историю вакцинации. Форма элемента справочника «Карточка питомца» представлена на рисунке 23.

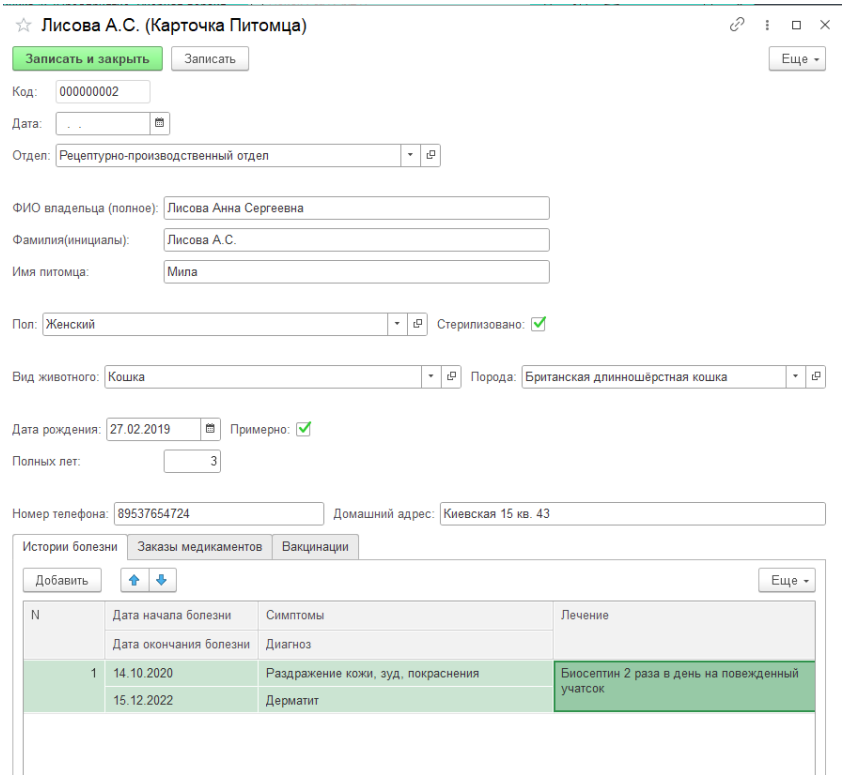



Рисунок 23 – Форма элемента справочника «Карточка питомца»

Справочник «Услуги» содержит информацию о услугах предоставляемых ветеринарной клиникой. Форма списка справочника «Услуги» представлена на рисунке 24.

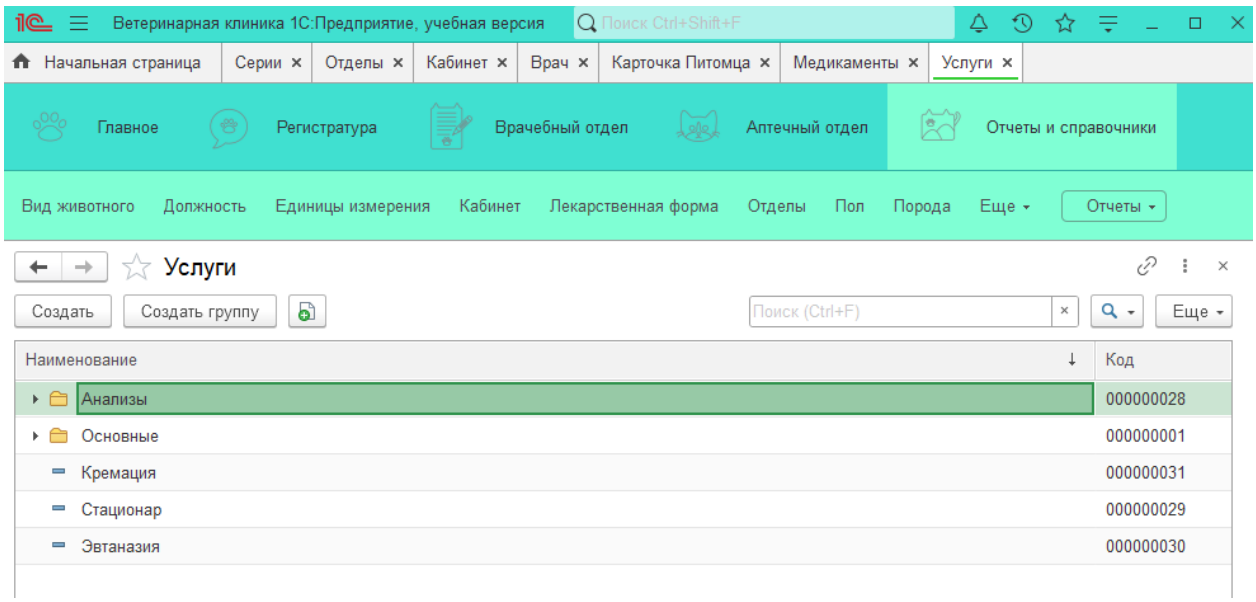



Рисунок 24 – Форма списка справочника «Услуги»

Справочник «Вид животного» содержит виды домашних животных. Форма списка справочника «Вид животного» представлена на рисунке 25.

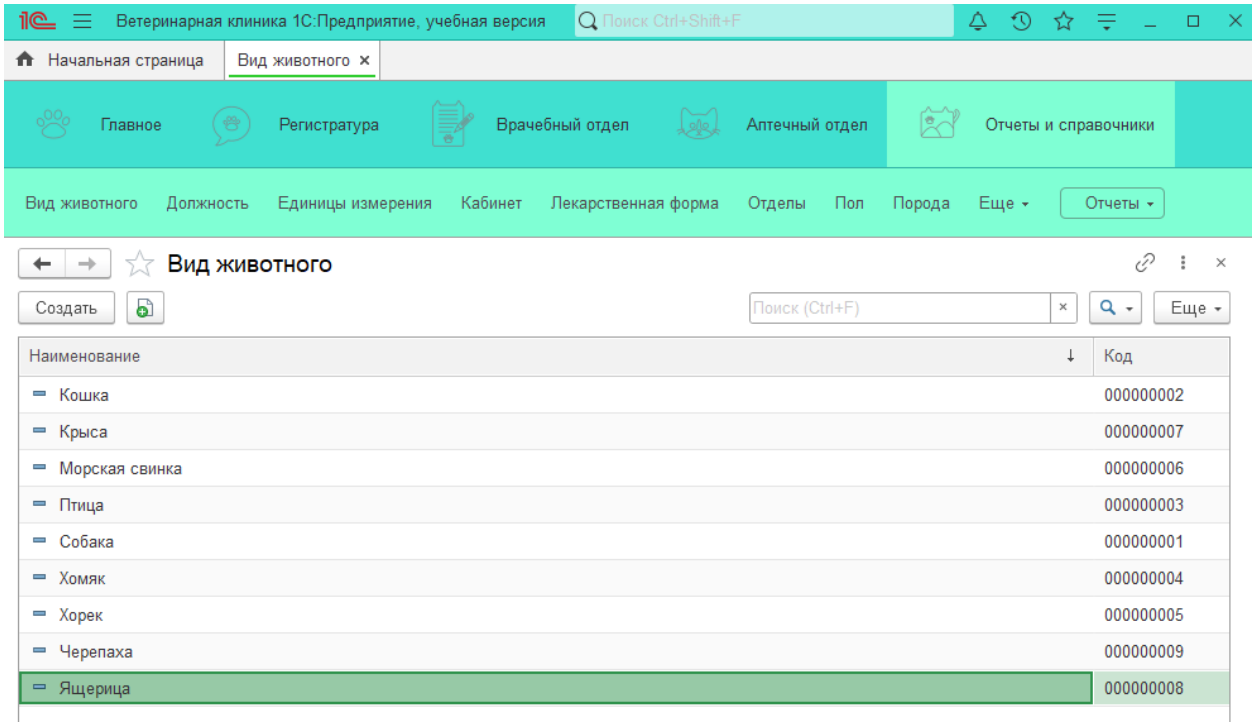



Рисунок 25 – Форма списка справочника «Вид животного»

Справочник «Должность» содержит имеющиеся должности врачей ветеринаров. Форма списка справочника «Должность» представлена на рисунке 26.

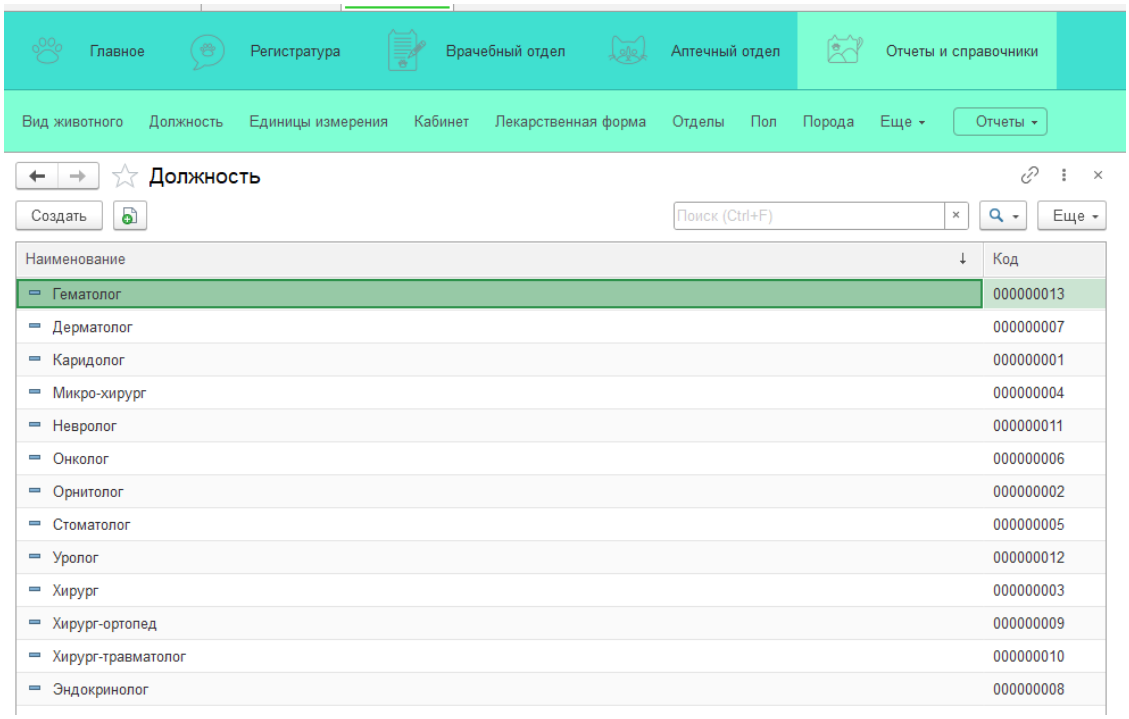



Рисунок 26 – Форма списка справочника «Должность»

Справочник «Форма медикамента» содержит виды форм медикаментов. Форма списка справочника «Форма медикаментов» представлена на рисунке 27.

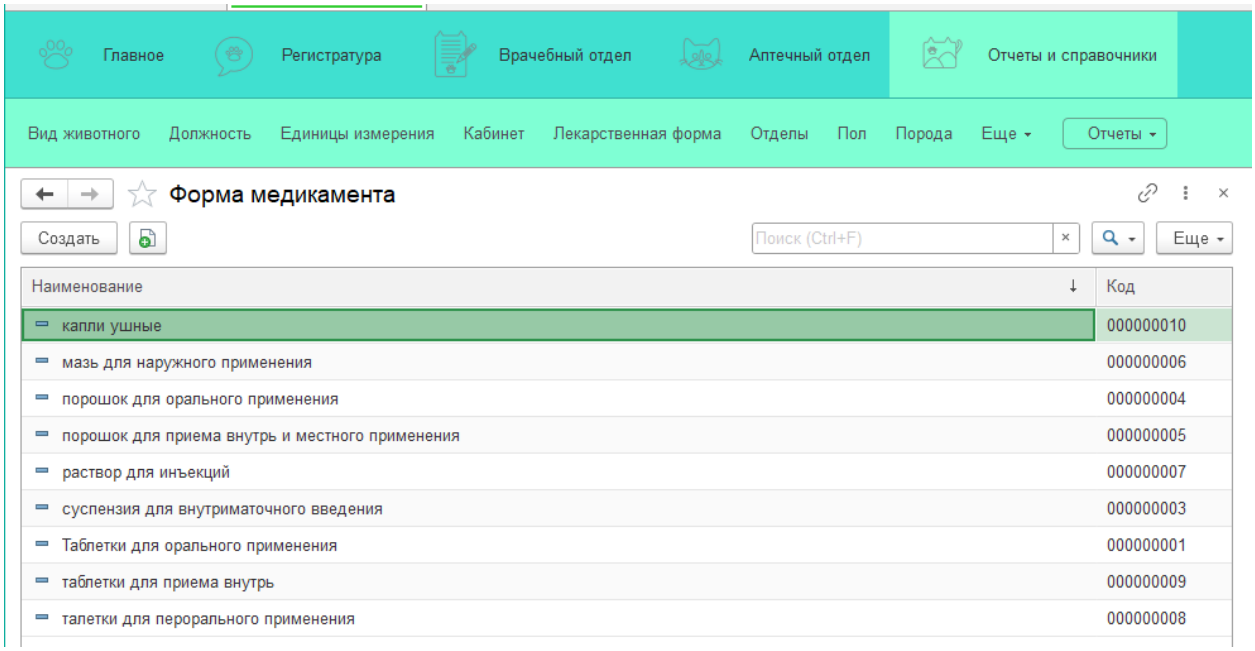



Рисунок 27 – Форма списка справочника «Форма медикамента»

Справочник «Лекарственная форма» содержит виды лекарств по симптоматике. Форма списка справочника «Лекарственная форма» представлена на рисунке 28.

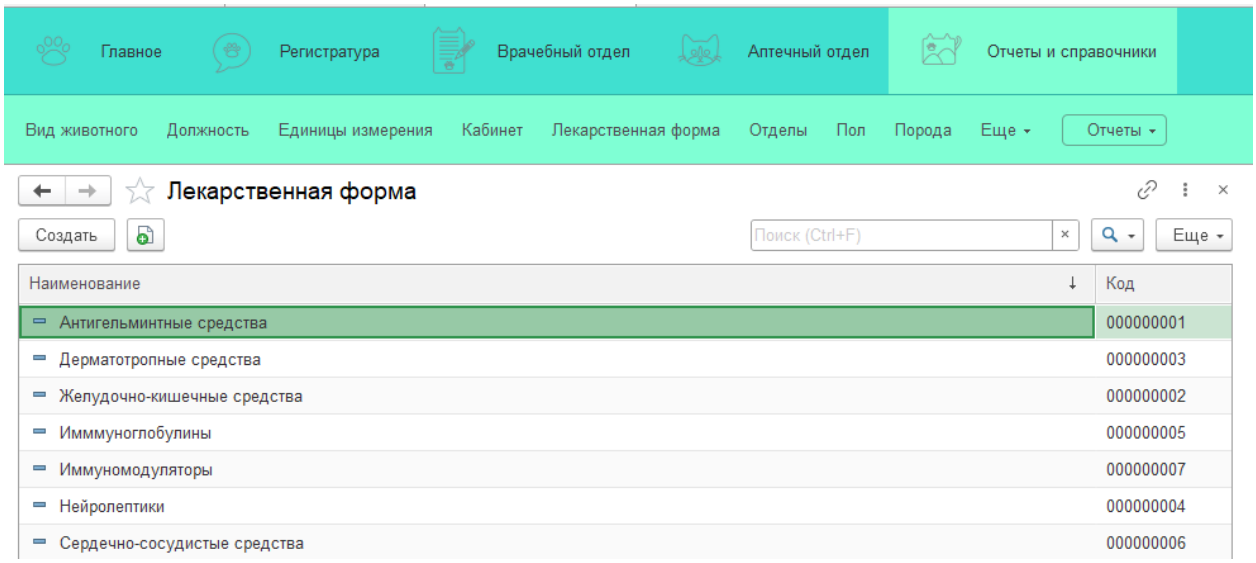



Рисунок 28 – Форма списка справочника «Лекарственная форма»

Справочник «Порода» содержит наименования пород животных. Форма списка справочника «Порода» представлена на рисунке 29.

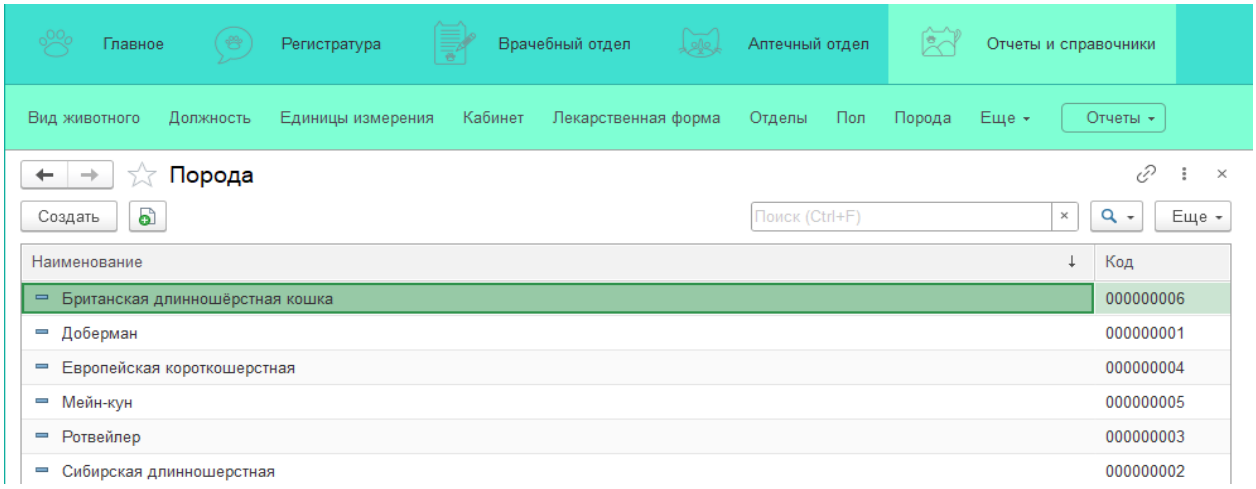



Рисунок 29 – Форма списка справочника «Порода»

Справочник «Пол» содержит обозначение половой принадлежности. Форма списка справочника «Пол» представлена на рисунке 30.

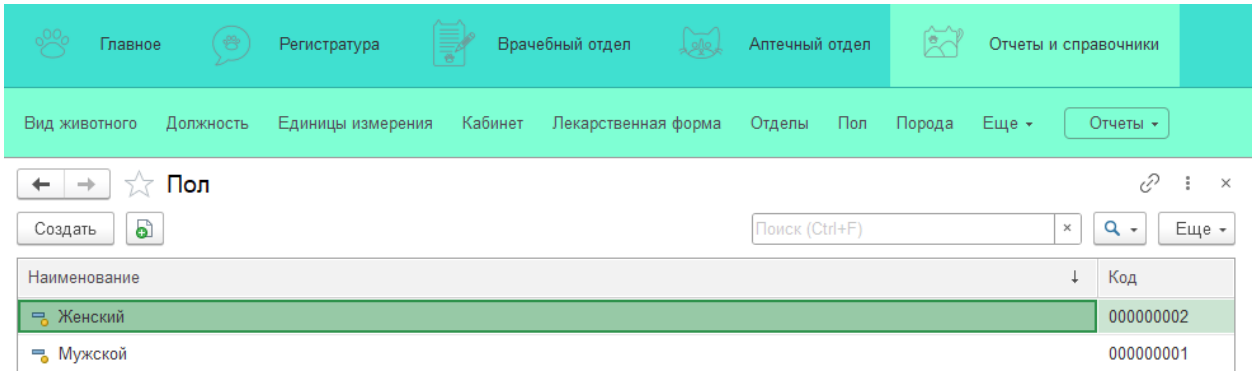



Рисунок 30 – Форма списка справочника «Пол»

Справочник «Единицы измерения» содержит различные единицы измерения. Форма списка справочника «Единицы измерения» представлена на рисунке 31.

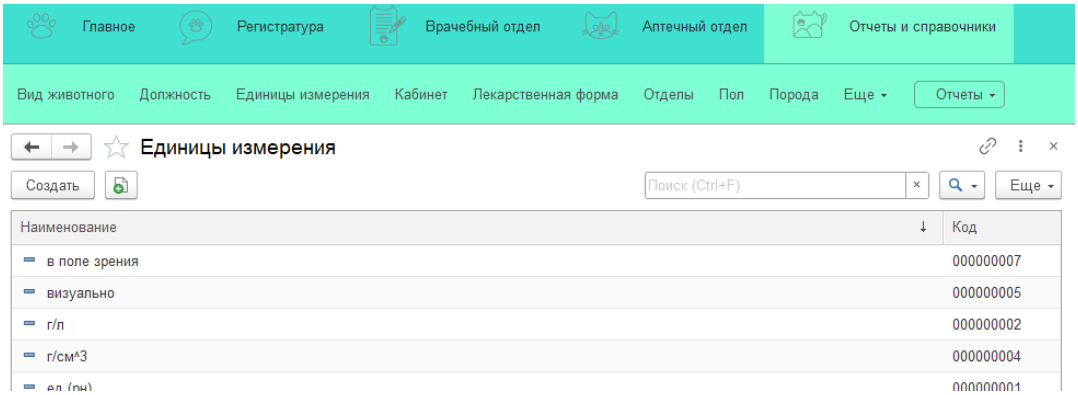



Рисунок 31 – Форма списка справочника «Единицы измерения»

Документы предназначены для фиксирования информации о происходящих событиях в подразделении или в организации в целом. Функциональность документа отличается от функциональности всех остальных объектов информационной базы. Это отличие связано с тем, что документ обладает способностью проведения. Факт проведения документа означает, что событие, которое он отражает, привело к изменению состояния учета [24].

Для системы ветеринарной клиники созданы документы, приведенные ниже.

Документ «Приходная накладная» фиксирует закупку серии медикаментов. Состоит в подсистеме «Аптечный отдел». Форма документа «Приходная накладная» представлена на рисунке 32.

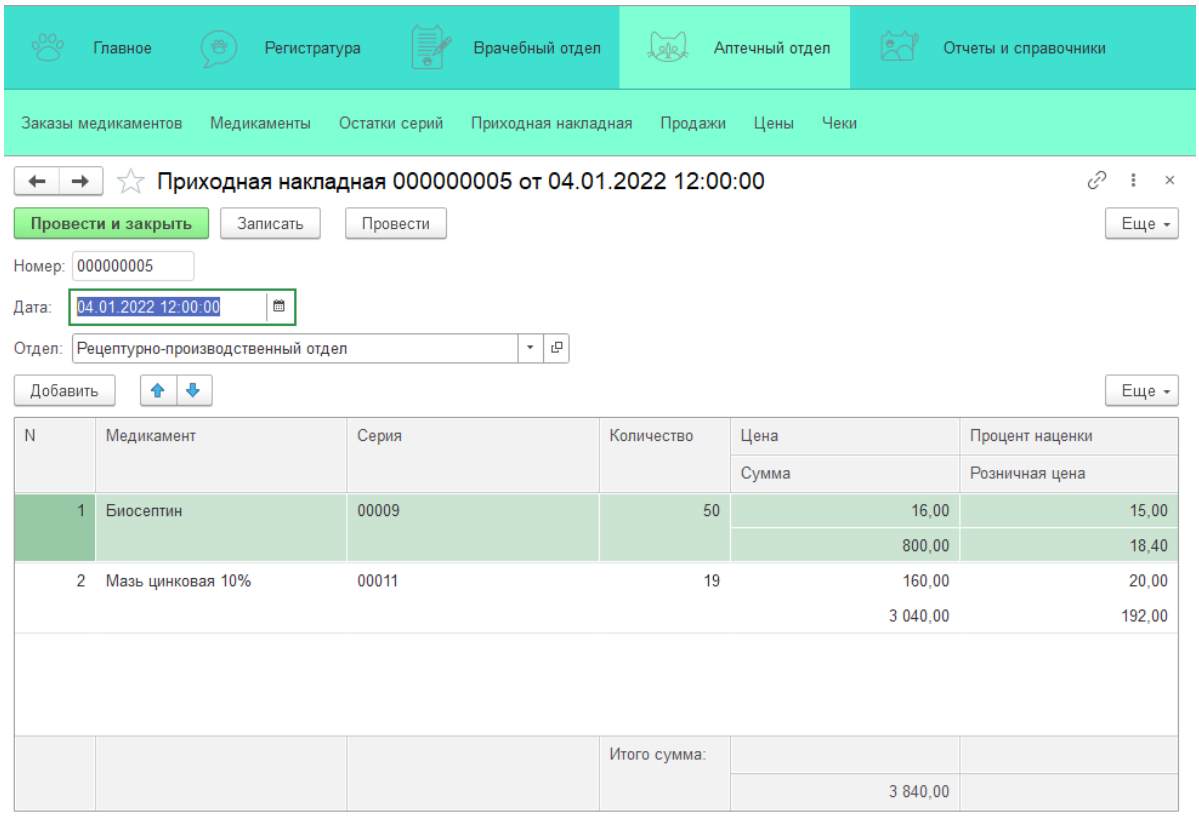



Рисунок 32 – Форма документа «Приходная накладная»

Документ «Чеки» фиксирует продажу медикаментов. Форма документа «Чеки» представлена на рисунке 33.

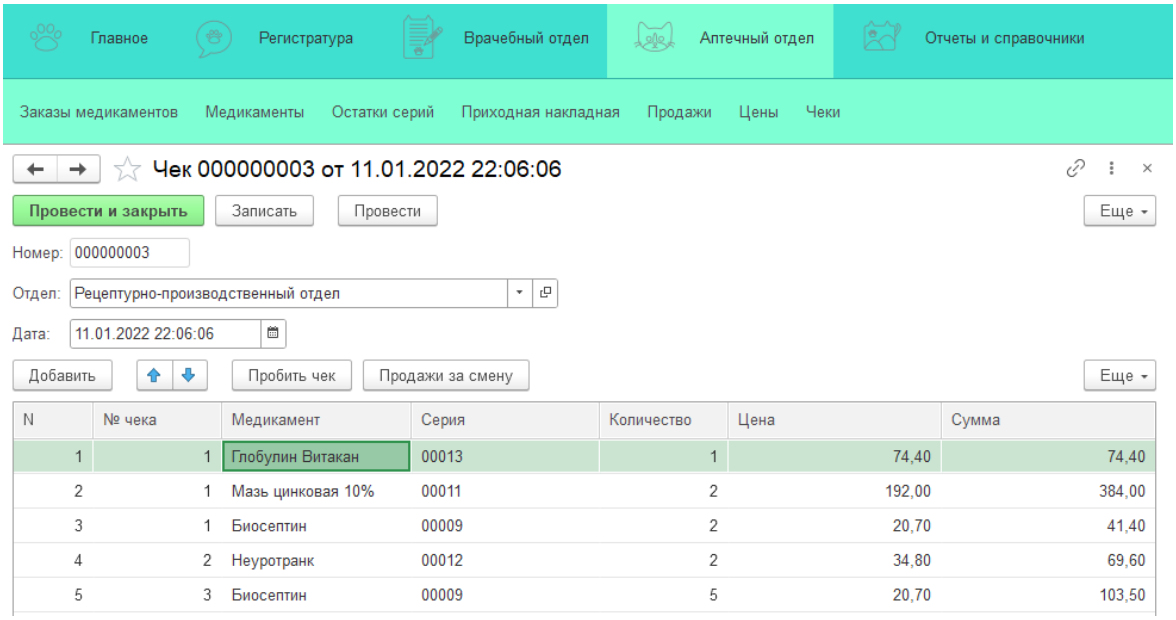



Рисунок 33 – Форма документа «Чеки»

Документ «День записи» фиксирует запись пациента к врачу или на анализ. Форма элемента документа «День записи» представлена на рисунке 34.

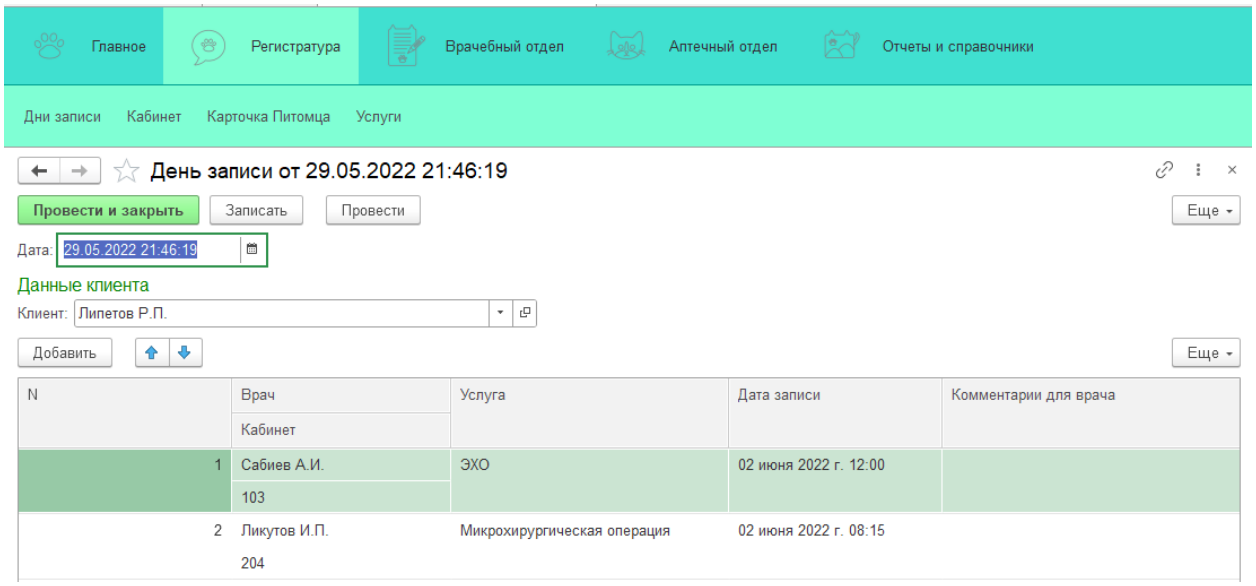



Рисунок 34 – Форма документа «День записи»

Документ «Заказы медикаментов» фиксирует заказ на медикаменты, сделанный через врача в аптеку при ветеринарной клинике. Форма документа «Заказы медикаментов» представлены на рисунке 35.

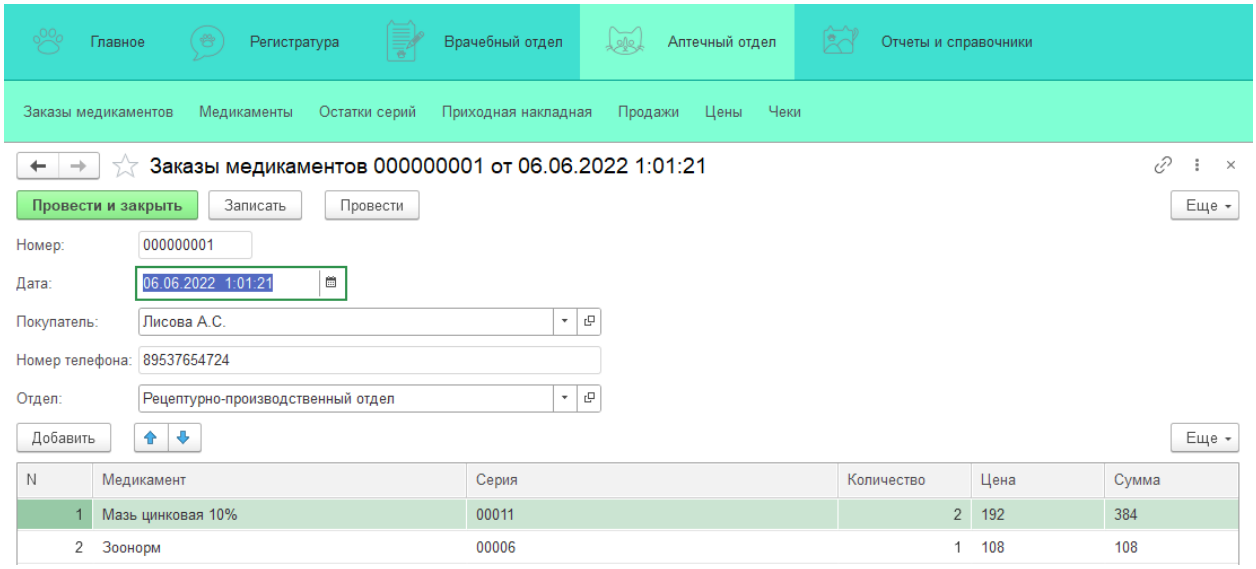



Рисунок 35 – Форма документа «Заказы медикаментов»

Документ «Анализы» фиксирует результаты проведенных анализов. Форма документа «Анализы» представлена на рисунке 36.

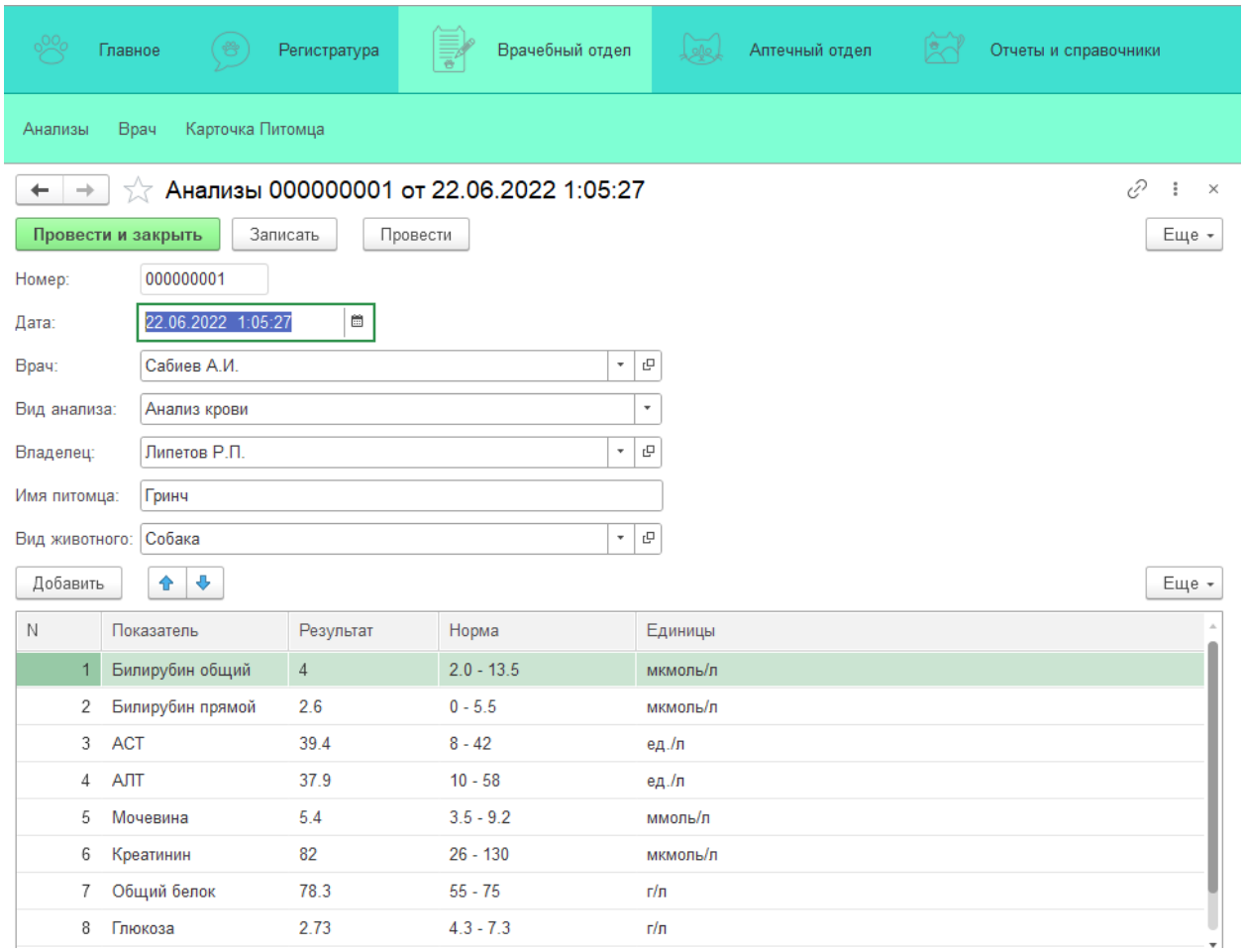


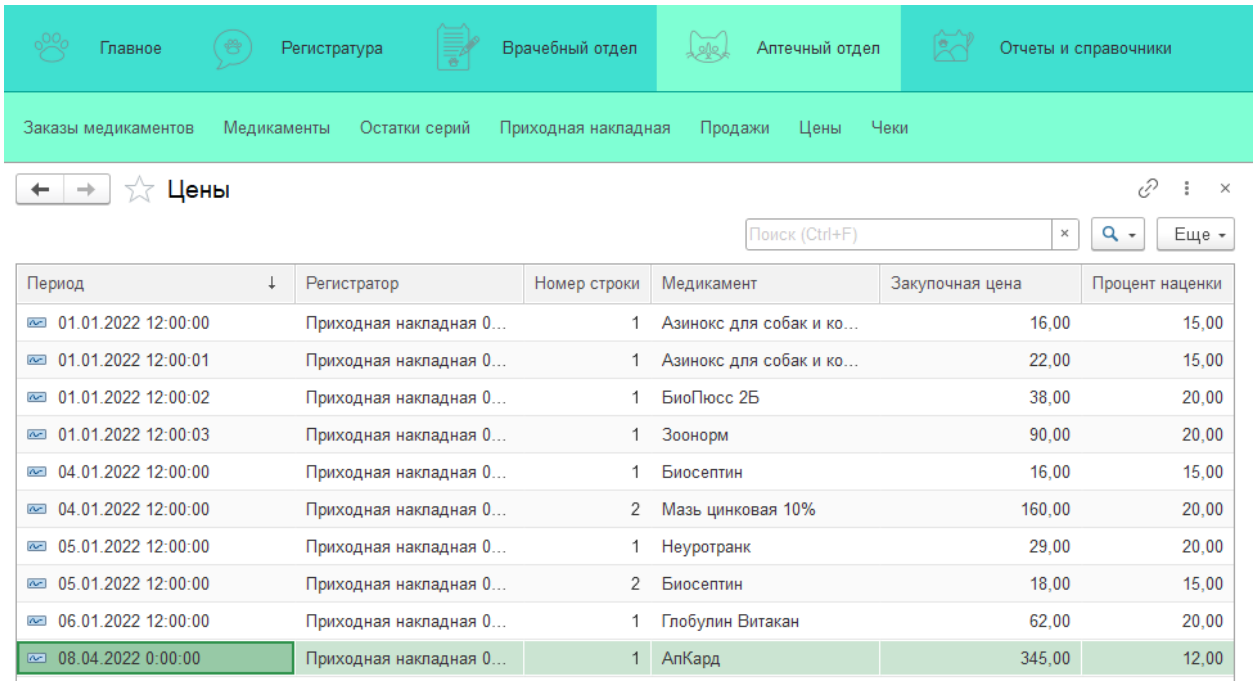

Рисунок 36 – Форма документа «Анализы»

Регистры сведений – это прикладные объекты конфигурации. Они позволяют хранить в прикладном решении произвольные данные в разрезе нескольких измерений. Например, в регистре сведений можно хранить курсы валют в разрезе валют, или цены предприятия в разрезе номенклатуры и типа цен [25].

Регистры накопления – это прикладные объекты конфигурации. Они составляют основу механизма учета движения средств (финансов, товаров, материалов и т.д.), который позволяет автоматизировать такие направления, как складской учет, взаиморасчеты, планирование [26].

Для системы ветеринарной клиники были созданы регистры, приведенные ниже.

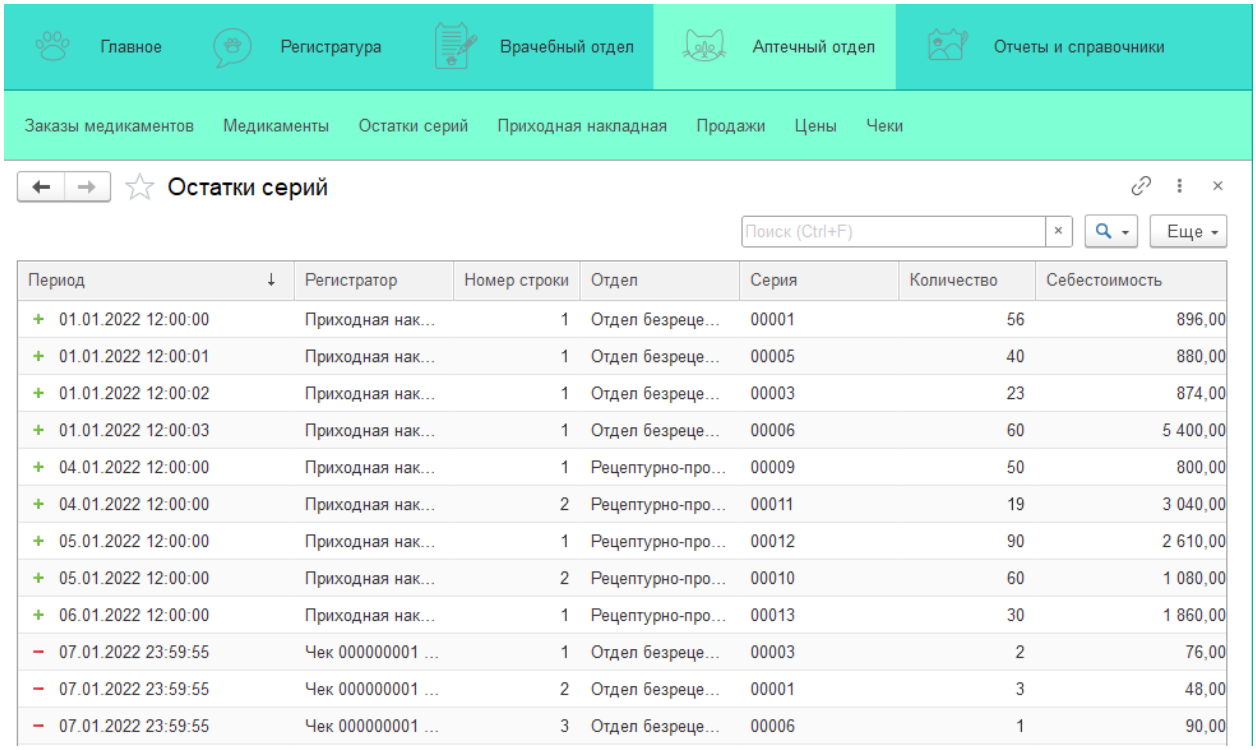
Регистр сведений «Цены» хранит закупочные цен серий медикаментов. Форма списка регистра «Цены» представлена на рисунке 37.




| Период | Регистратор | Номер строки | Медикамент | Закупочная цена | Процент наценки |
|---|---|---|---|---|---|
| 01.01.2022 12:00:00 | Приходная накладная 0... | 1 | Азинокс для собак и ко... | 16,00 | 15,00 |
| 01.01.2022 12:00:01 | Приходная накладная 0... | 1 | Азинокс для собак и ко... | 22,00 | 15,00 |
| 01.01.2022 12:00:02 | Приходная накладная 0... | 1 | БиоПюсс 2Б | 38,00 | 20,00 |
| 01.01.2022 12:00:03 | Приходная накладная 0... | 1 | Зоонорм | 90,00 | 20,00 |
| 04.01.2022 12:00:00 | Приходная накладная 0... | 1 | Биосептин | 16,00 | 15,00 |
| 04.01.2022 12:00:00 | Приходная накладная 0... | 2 | Мазь цинковая 10% | 160,00 | 20,00 |
| 05.01.2022 12:00:00 | Приходная накладная 0... | 1 | Неуротранк | 29,00 | 20,00 |
| 05.01.2022 12:00:00 | Приходная накладная 0... | 2 | Биосептин | 18,00 | 15,00 |
| 06.01.2022 12:00:00 | Приходная накладная 0... | 1 | Глобулин Витакан | 62,00 | 20,00 |
| 08.04.2022 0:00:00 | Приходная накладная 0... | 1 | АпКард | 345,00 | 12,00 |

Рисунок 37 – Форма списка регистра «Цены»

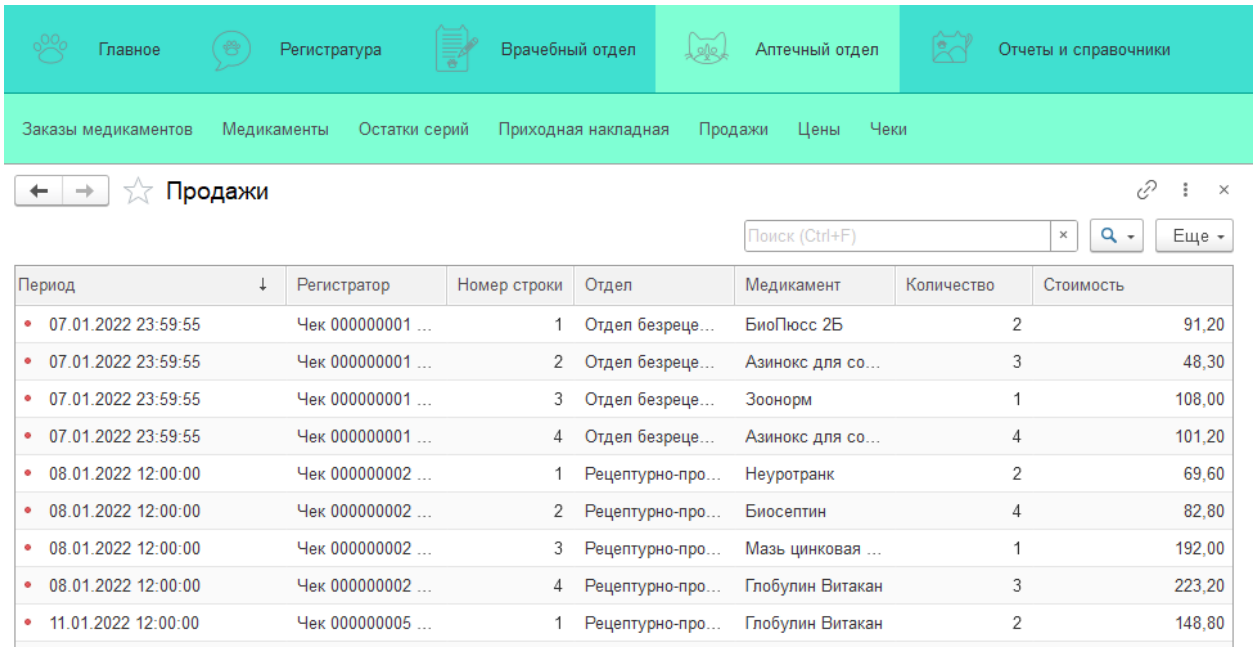
Регистр накопления «Остатки серий» предназначен для накопления информации о движения серий медикаментов. Форма списка регистра «Остатки серий» представлена на рисунке 38.




| Период | Регистратор | Номер строки | Отдел | Серия | Количество | Себестоимость |
|---|---|---|---|---|---|---|
| + 01.01.2022 12:00:00 | Приходная нак... | 1 | Отдел безреце... | 00001 | 56 | 896,00 |
| + 01.01.2022 12:00:01 | Приходная нак... | 1 | Отдел безреце... | 00005 | 40 | 880,00 |
| + 01.01.2022 12:00:02 | Приходная нак... | 1 | Отдел безреце... | 00003 | 23 | 874,00 |
| + 01.01.2022 12:00:03 | Приходная нак... | 1 | Отдел безреце... | 00006 | 60 | 5 400,00 |
| + 04.01.2022 12:00:00 | Приходная нак... | 1 | Рецептурно-про... | 00009 | 50 | 800,00 |
| + 04.01.2022 12:00:00 | Приходная нак... | 2 | Рецептурно-про... | 00011 | 19 | 3 040,00 |
| + 05.01.2022 12:00:00 | Приходная нак... | 1 | Рецептурно-про... | 00012 | 90 | 2 610,00 |
| + 05.01.2022 12:00:00 | Приходная нак... | 2 | Рецептурно-про... | 00010 | 60 | 1 080,00 |
| + 06.01.2022 12:00:00 | Приходная нак... | 1 | Рецептурно-про... | 00013 | 30 | 1 860,00 |
| − 07.01.2022 23:59:55 | Чек 000000001 ... | 1 | Отдел безреце... | 00003 | 2 | 76,00 |
| − 07.01.2022 23:59:55 | Чек 000000001 ... | 2 | Отдел безреце... | 00001 | 3 | 48,00 |
| − 07.01.2022 23:59:55 | Чек 000000001 ... | 3 | Отдел безреце... | 00006 | 1 | 90,00 |

Рисунок 38 – Форма списка регистра «Остатки серий»

Регистр накопления «Продажи» предназначен для накопления информации о продажах серий медикаментов. Форма списка регистра «Продажи» представлена на рисунке 39.




| Период | Регистратор | Номер строки | Отдел | Медикамент | Количество | Стоимость |
|---|---|---|---|---|---|---|
| 07.01.2022 23:59:55 | Чек 000000001 ... | 1 | Отдел безреце... | БиоПюсс 2Б | 2 | 91,20 |
| 07.01.2022 23:59:55 | Чек 000000001 ... | 2 | Отдел безреце... | Азинокс для со... | 3 | 48,30 |
| 07.01.2022 23:59:55 | Чек 000000001 ... | 3 | Отдел безреце... | Зоонорм | 1 | 108,00 |
| 07.01.2022 23:59:55 | Чек 000000001 ... | 4 | Отдел безреце... | Азинокс для со... | 4 | 101,20 |
| 08.01.2022 12:00:00 | Чек 000000002 ... | 1 | Рецептурно-про... | Неуротранк | 2 | 69,60 |
| 08.01.2022 12:00:00 | Чек 000000002 ... | 2 | Рецептурно-про... | Биосептин | 4 | 82,80 |
| 08.01.2022 12:00:00 | Чек 000000002 ... | 3 | Рецептурно-про... | Мазь цинковая ... | 1 | 192,00 |
| 08.01.2022 12:00:00 | Чек 000000002 ... | 4 | Рецептурно-про... | Глобулин Витакан | 3 | 223,20 |
| 11.01.2022 12:00:00 | Чек 000000005 ... | 1 | Рецептурно-про... | Глобулин Витакан | 2 | 148,80 |

Рисунок 39 – Форма списка регистра» «Продажи»

Отчеты – это прикладные объекты конфигурации. Они предназначены для обработки накопленной информации и получения сводных данных в

удобном для просмотра и анализа виде. Конфигуратор позволяет формировать набор различных отчетов, достаточных для удовлетворения потребности пользователей системы в достоверной и подробной выходной информации [27].

Для системы ветеринарной клиники были созданы следующие ниже отчеты.

Отчет «Ведомость остатков» выводит список остатков медикаментов на складах. Форма отчета представлена на рисунке 40.

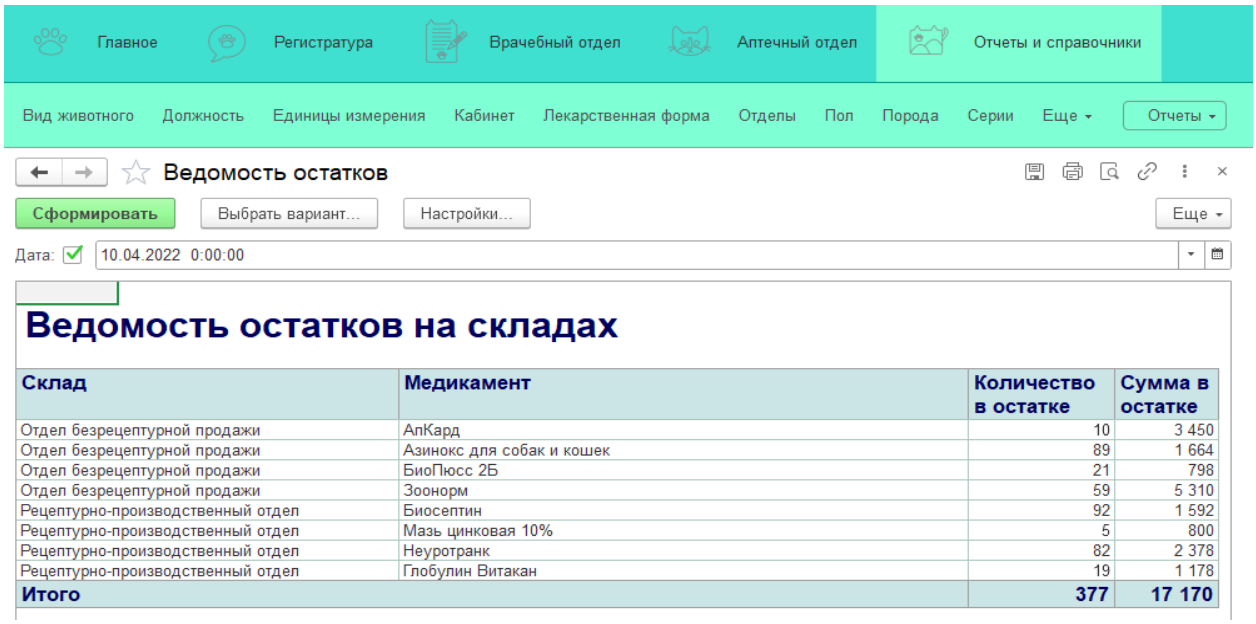



Рисунок 40 – Отчет «Ведомость остатков»

Отчет «Список врачей» выводит список врачей их должности и стаж работы. Печатная форма отчета представлена на рисунке 41.

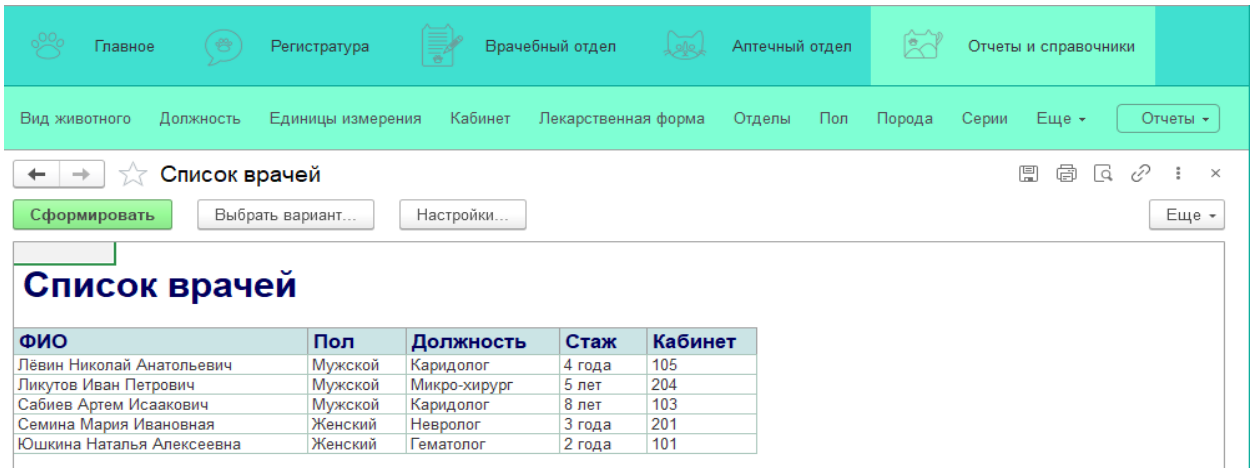
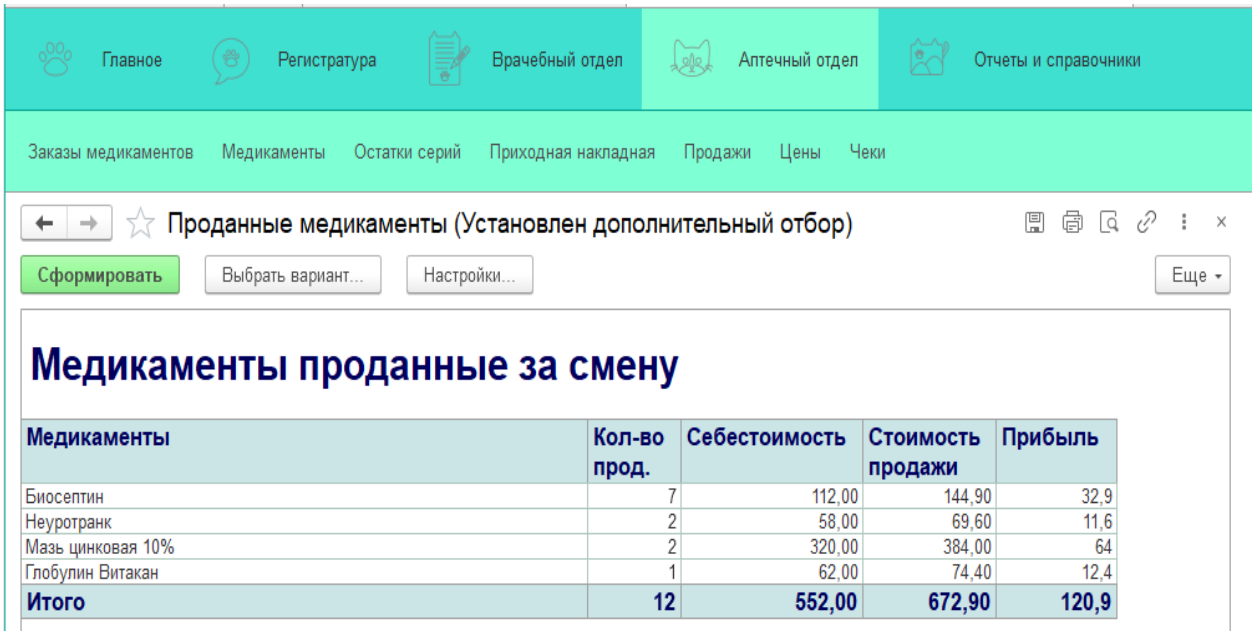



Рисунок 41 – Отчет «Список врачей»

Отчет «Проданные медикаменты» выводит список проданных медикаментов за смену, стоимость продажи, а также полученную прибыль. Печатная форма отчета представлена на рисунке 42.




**Медикаменты проданные за смену**

| Медикаменты | Кол-во прод. | Себестоимость | Стоимость продажи | Прибыль |
|---|---|---|---|---|
| Биосептин | 7 | 112,00 | 144,90 | 32,9 |
| Неуротранк | 2 | 58,00 | 69,60 | 11,6 |
| Мазь цинковая 10% | 2 | 320,00 | 384,00 | 64 |
| Глобулин Витакан | 1 | 62,00 | 74,40 | 12,4 |
| **Итого** | **12** | **552,00** | **672,90** | **120,9** |

Рисунок 42 – Отчет «Проданные медикаменты»

Разрабатываемая система управления ветеринарной клиникой состоит из 14 справочников, 5 документов, 3 регистров, 3 отчетов.

### 3.2 Реализация кода программного средства

При разработке информационной системы управления ветеринарной клиники используется технологическая платформа 1С: Предприятие 8.3, имеющая встроенный язык. Программный код пишется в различных модулях объектов конфигураций.

Приведем пример программного кода модуля объекта документа «Приходная накладная». Для регистрации движения медикамента в регистре была разработана процедура, приведенная в листинге 1.

Листинг 1 – Программный код модуля объекта документа «Приходная накладная»

```
Процедура ОбработкаПроведения(Отказ, Режим)
```

```
	// регистр ОстаткиСерий Приход
	Движения.ОстаткиСерий.Записывать = Истина;
	Для Каждого ТекСтрокаПриходнаяНакладная Из ПриходнаяНакладная Цикл
		Движение = Движения.ОстаткиСерий.Добавить();
		Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
		Движение.Период = Дата;
		Движение.Отдел = Отдел;
		Движение.Серия = ТекСтрокаПриходнаяНакладная.Серия;
		Движение.Количество = ТекСтрокаПриходнаяНакладная.Количество;
		Движение.Себестоимость = ТекСтрокаПриходнаяНакладная.Сумма;

		// регистр Цены
		Движения.Цены.Записывать = Истина;
		ЗакупочнаяЦенаИПроцентНаценки =
РаботаСДанными.ПолучитьЗакупочнуюЦену(Дата,
ТекСтрокаПриходнаяНакладная.Медикамент);
		Если ТекСтрокаПриходнаяНакладная.Цена <>
ЗакупочнаяЦенаИПроцентНаценки.ЗакупочнаяЦена ИЛИ
			ТекСтрокаПриходнаяНакладная.ПроцентНаценки <>
ЗакупочнаяЦенаИПроцентНаценки.ПроцентНаценки Тогда
			Движение = Движения.Цены.Добавить();
			Движение.Период = Дата;
			Движение.Медикамент =
ТекСтрокаПриходнаяНакладная.Медикамент;
			Движение.ЗакупочнаяЦена = ТекСтрокаПриходнаяНакладная.Цена;
			Движение.ПроцентНаценки =
ТекСтрокаПриходнаяНакладная.ПроцентНаценки;
		КонецЕсли;

	КонецЦикла;

КонецПроцедуры
```

Программный код остальных объектов конфигурации приведен в приложении А.

### 3.3 Тестирование программного средства

Цель функционального тестирования – обнаружение несоответствий между реальным поведением реализованных функций и ожидаемым поведением в соответствии со спецификацией и исходными требованиями. Функциональные тесты должны охватывать все реализованные функции с

учетом наиболее вероятных типов ошибок. Тестовые сценарии, объединяющие отдельные тесты, ориентированы на проверку качества решения функциональных задач [28].

Проведем функциональное тестирование работы создания и проведения документов, создадим печатную форму документов. Результаты представлены в таблице 13.

Таблица 13 – Функциональное тестирование работы системы

| № п/п | Название теста | Предполагаемый результат | Полученный результат | Результат тестирования |
|---|---|---|---|---|
| 1 | Создание документа «Приходная накладная» | Открыть документ «Приходная накладная», заполнить все реквизиты и табличную часть, нажать кнопку «Провести и закрыть». Документ проведен и выведет список документов | Соответствует предполагаемому результату | Тест пройден (рисунок Б.1 – Б.2) |
| 2 | Создание документа «Чеки» | Открыть документ «Чеки», заполнить все реквизиты и табличную часть, нажать кнопку «Провести и закрыть». Документ проведен и выведет список документов | Соответствует предполагаемому результату | Тест пройден (рисунок Б.3 – Б.4) |
| 4 | Создание документа «День записи» | Открыть документ «День записи», заполнить все реквизиты и табличную часть, нажать кнопку «Провести и закрыть». Документ проведен и выведет список документов | Соответствует предполагаемому результату | Тест пройден (рисунок Б.5 – Б.6) |
| 5 | Создание печатной формы документа «День записи» | Открыть документ «День записи», заполнить документ, нажать кнопку «Печать». После появится печатный вариант документа «День записи» | Соответствует предполагаемому результату | Тест пройден (рисунок Б.7) |

Продолжение таблицы 13

| № п/п | Название теста | Предполагаемый результат | Полученный результат | Результат тестирования |
|---|---|---|---|---|
| 6 | Создание документа «Анализы» | Открыть документ «Анализы», заполнить все реквизиты и табличную часть, нажать кнопку «Провести и закрыть». Документ проведен и выведет список документов | Соответствует предполагаемому результату | Тест пройден (рисунок Б.8 – Б.9) |
| 7 | Создание печатной формы документа «Анализы» | Открыть документ «Анализы», заполнить все реквизиты и табличную часть, нажать кнопку «Печать». После появится печатный вариант документа «Анализы» | Соответствует предполагаемому результату | Тест пройден (рисунок Б.10) |
| 8 | Создание документа «Заказы медикаментов» через «Карточка питомца» | Открыть справочник «Карточка питомца», заполнить все реквизиты и табличную часть, нажать кнопку «Сделать заказ». Найти созданный документ «Заказы медикаментов» открыть его и нажать кнопку «Провести и закрыть». Документ проведен и выведет список документов | Соответствует предполагаемому результату | Тест пройден (рисунок Б.11 – Б.13) |
| 9 | Создание печатной формы документа Заказы медикаментов | Открыть документ «Заказы медикаментов», заполнить все реквизиты и табличную часть, нажать кнопку «Печать». После появится печатный вариант документа «Заказы медикаментов» | Соответствует предполагаемому результату | Тест пройден (рисунок Б.14) |

Проведя функционально тестирование работы систем, убедились в ее корректной работе, выполнение задач без видимых ошибок и сбоев.

### 3.4 Расчет метрик кода

Количество строк кода (англ. Source Lines of Code — SLOC) – это метрика программного обеспечения, используемая для измерения его объёма с помощью подсчёта количества строк в тексте исходного кода [29].

Проведем подсчет данной метрики в процедуре ОбработкаПроведения, служащее для корректного движения данных документа Приходная накладная. Процедура приведена в листинге 2.

Листинг 2 – Процедура ОбработкаПроведения()

```
Процедура ОбработкаПроведения(Отказ, Режим)
	// регистр ОстаткиСерий Приход
	Движения.ОстаткиСерий.Записывать = Истина;
	Для Каждого ТекСтрокаПриходнаяНакладная Из ПриходнаяНакладная Цикл
		Движение = Движения.ОстаткиСерий.Добавить();
		Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
		Движение.Период = Дата;
		Движение.Отдел = Отдел;
		Движение.Серия = ТекСтрокаПриходнаяНакладная.Серия;
		Движение.Количество = ТекСтрокаПриходнаяНакладная.Количество;
		Движение.Себестоимость = ТекСтрокаПриходнаяНакладная.Сумма;

		// регистр Цены
		Движения.Цены.Записывать = Истина;
		ЗакупочнаяЦенаИПроцентНаценки =
РаботаСДанными.ПолучитьЗакупочнуюЦену(Дата,
ТекСтрокаПриходнаяНакладная.Медикамент);
		Если ТекСтрокаПриходнаяНакладная.Цена <>
ЗакупочнаяЦенаИПроцентНаценки.ЗакупочнаяЦена ИЛИ
			ТекСтрокаПриходнаяНакладная.ПроцентНаценки <>
ЗакупочнаяЦенаИПроцентНаценки.ПроцентНаценки Тогда
			Движение = Движения.Цены.Добавить();
			Движение.Период = Дата;
			Движение.Медикамент =
ТекСтрокаПриходнаяНакладная.Медикамент;
			Движение.ЗакупочнаяЦена = ТекСтрокаПриходнаяНакладная.Цена;
```

```
                Движение.ПроцентНаценки =
ТекСтрокаПриходнаяНакладная.ПроцентНаценки;
            КонецЕсли;

      КонецЦикла;

КонецПроцедуры
```

В данной программе имеем:

- 24 физических строк кода;
- 18 логических строк кода;
- 2 строк комментариев.

Рассчитаем количественные показатели по метрике Холстеда для процедуры ОбработкаПроведения().

Основные метрические характеристики программы:

- $n_1$ – число простых (уникальных) операторов в программе;
- $n_2$ – число простых (уникальных) операндов в программе;
- $N_1$ – общее число всех операторов в программе;
- $N_2$ – общее число всех операндов в программе [30].

На основе перечисленных характеристик могут быть рассчитаны значения основных показателей, формулы которых представлены ниже.

Словарь программы:

$$n = n_1 + n_2$$

Длина программы:

$$N = N_1 + N_2$$

Объем программы:

$$V = N \times \log_2 n$$

Сложность программы:

$$HD = \left( \frac{n_1}{2} \times \frac{N_2}{n_2} \right)$$

В таблице 14 приведены операторы процедуры ОбработкаПроведения().

Таблица 14 – Операторы процедуры

| Операторы | Число вхождений |
|---|---|
| = | 15 |
| . | 31 |
| Если | 1 |
| ; | 17 |
| <> | 2 |
| Для каждого | 1 |
| Или | 1 |
| n1 = 7 | N1 = 68 |

В таблице 15 приведены операнды процедуры ОбработкаПроведения().

Таблица 15 – Операнды процедуры

| Операнды | Число вхождений |
|---|---|
| Движения.ОстаткиСерий.Записывать | 1 |
| Движения.Цены.Записывать | 1 |
| ЗакупочнаяЦенаИПроцентНаценки | 1 |
| Движение | 2 |
| n2 = 4 | N2 = 5 |

Используя основные характеристики, которые рассчитали в таблицах 9 и 10, определим значения показателей по вышеприведенным формулам:

– словарь программы (n): 11;
– длина программы (N): 73;
– объем программы (V): 252,5;
– сложность программы (HD): 4,375.

Объем программы характеризует максимально компактный текст программы. Измеряется объем не количеством слов, а числом двоичных разрядов. От сложности программы зависит производительность труда разработчиков, а следовательно, сроки и стоимость разработки.

# Заключение

В процессе выполнения выпускной квалификационной работы был проведен анализ предметной области, в рамках которого для сравнительной характеристики были выбраны информационные системы для управления ветеринарной клиникой. В ходе сравнения систем с аналогичным функционалом была определена структура и основные функции разрабатываемой конфигурации. Проведен анализ функций и бизнес-процессов организации. В том числе был проведен анализ системы с помощью различных диаграмм UML: диаграмма прецедентов, диаграмма состояний, диаграмма деятельности, диаграмма последовательности, диаграмма кооперации. Представлено описание выбранных средств проектирования и разработки.

В результате была разработана конфигурация, позволяющая управлять деятельностью ветеринарной клиники, а также деятельностью аптеки при этой клинике. Тестирование программных средств заключалось в определении метода тестирования и выполнения тестов. Результаты функционального тестирования показали исправность работы системы.

## Список использованных источников

1. Минеева Т.И «История ветеринарии»: учебное пособие для студентов вузов, обучающихся по специальности 310800 – Ветеринария. / Т.И. Минеева. – Изд-во Лань, 2005. – 382 с.

2. Сайт Grow from Knowledge. Исследование GfK: домашние животные. [Электронный ресурс]. URL: https://www.gfk.com/ru/press/issledovanie-gfk-bolshe-vsego-domashnikh-zhivotnykh-v-rossii-ssha-i-latinskoi-amerike (дата обращения: 22.04.2022).

3. Первый БИТ [Электронный ресурс]. URL: https://penza.1cbit.ru (дата обращения: 22.04.2022).

4. Ветменеджер [Электронный ресурс]. URL: https://vetmanager.ru (дата обращения: 22.04.2022)

5. VetDesk [Электронный ресурс]. URL: https://vetdesk.ru (дата обращения: 22.04.2022)

6. Построение UML-модели в среде Visual Paradigm. [Электронный ресурс]. URL:http://sp.cmc.msu.ru/courses/ooap/exerb2020.html (дата обращения: 23.04.2022)

7. Пилецкий И.И, «Проектирование, разработка и сопровождение баз данных с использованием CASE-средств» / И.И. Пилецкий – Минск: БГУИР, 2009 – 116 с.

8. Кара-Ушанов В.Ю. «Функционально-структурное моделирование в системе Ramus Educational» – Екатеринбург, 2019 – 67 с.

9. Почему система программ «1С: Предприятие». [Электронный ресурс]. URL: https://1c.ru/rus/products/1c/predpr/why-predpr.htm (дата обращения: 23.04.2022)

10. Дэвид А. Марка «Методология структурного анализа и проектирования» – 243 с.

11. Методология функционального моделирования IDEF0/Научно-исследовательский центр CALS – технологий «Прикладная Логистика» – 75с.

12. Чигарина Е.И. «Базы данных учебное пособие» / Е.И. Чигарина – Самара: Издательство СГАУ, 2015 – 208 с.

13. Элементы графической нотации диаграммы вариантов использования [Электронный ресурс]. URL: https://intuit.ru/studies/courses/32/32/lecture/1004 (дата обращения: 6.06.2022)

14. Виды диаграмм UML [Электронный ресурс]. URL: https://intuit.ru/studies/courses/1007/229/lecture/5954?page=4 (дата обращения: 6.06.2022)

15. Фаулер М. UML. Основы, 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с.

16. Диаграмма последовательности [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Диаграмма_последовательности (дата обращения: 6.06.2022)

17. Элементы графической нотации диаграммы кооперации [Электронный ресурс]. URL: https://intuit.ru/studies/courses/32/32/lecture/1012 (дата обращения: 6.06.2022)

18. Спицина И.А. Применение системного анализа при разработке пользовательского интерфейса информационных систем: учеб. пособие / И.А. Спицина, К.А. Аксёнов. — Екатеринбург: Изд-во Урал. ун-та, 2018. – 100 с.

19. Концепция пользовательского интерфейса [Электронный ресурс]. URL: https://v8.1c.ru/platforma/kontseptsiya-polzovatelskogo-interfeysa/ (дата обращения: 6.06.2022)

20. Сапаров А.Ю. Проектный практикум. Управление проектом в ProjectLibre: учебно-методическое пособие / Сост.: А.Ю. Сапаров, Ижевск, 2020. 48 с.

21. Как диаграммы Ганта упрощают работу с проектами [Электронный ресурс]. URL: https://habr.com/ru/company/hygger/blog/415271/ (дата обращения: 6.06.2022)

22. What is Component Diagram? [Электронный ресурс]. URL: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/ (дата обращения: 6.06.2022)

23. Справочники [Электронный ресурс]. URL: https://v8.1c.ru/platforma/spravochniki/ (дата обращения: 6.06.2022)

24. Кашаев С.М. Программирование в 1С: Предприятие 8.3. — СПб.: Питер, 2014. — 304 с

25. Регистр сведений [Электронный ресурс]. URL: https://v8.1c.ru/platforma/registr-svedeniy/ (дата обращения: 6.06.2022)

26. Регистр накопления [Электронный ресурс]. URL: https://v8.1c.ru/platforma/registr-nakopleniya/ (дата обращения: 6.06.2022)

27. Отчет [Электронный ресурс]. URL: https://v8.1c.ru/platforma/otchet/ (дата обращения: 6.06.2022)

28. Методы проверки и тестирования программ и систем [Электронный ресурс]. URL: https://intuit.ru/studies/courses/2190/237/lecture/6130?page=2 (дата обращения: 6.06.2022)

29. Количество строк кода [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Количество_строк_кода (дата обращения: 6.06.2022)

30. Семахин А. М. Методы верификации и оценки качества программного обеспечения: учебное пособие. – Курган: Изд-во КГУ, 2018. – 150 с.

Приложение А

ЛИСТИНГ ПРОГРАММЫ

(обязательное)

Листинг А.1 – Код модуля формы документа «Приходная накладная»

```
&НаКлиенте
Процедура ПриходнаяНакладнаяКоличествоПриИзменении(Элемент)
     РаботаСДокументами.РассчитатьСумму(Элементы.ПриходнаяНакладн
ая.ТекущиеДанные);
     РассчитатьРозничнуюЦену();
КонецПроцедуры

&НаКлиенте
Процедура ПриходнаяНакладнаяЦенаПриИзменении(Элемент)
     РаботаСДокументами.РассчитатьСумму(Элементы.ПриходнаяНакладн
ая.ТекущиеДанные);
     РассчитатьРозничнуюЦену();
КонецПроцедуры

&НаКлиенте
Процедура ПриходнаяНакладнаяМедикаментПриИзменении(Элемент)
     Строка = Элементы.ПриходнаяНакладная.ТекущиеДанные;
     ЗакупочнаяЦенаИПроцентНаценки =
РаботаСДанными.ПолучитьЗакупочнуюЦену(Объект.Дата,
Строка.Медикамент);
     Строка.Цена = ЗакупочнаяЦенаИПроцентНаценки.ЗакупочнаяЦена;
     Строка.ПроцентНаценки =
ЗакупочнаяЦенаИПроцентНаценки.ПроцентНаценки;
КонецПроцедуры

&НаКлиенте
Процедура ПриходнаяНакладнаяПроцентНаценкиПриИзменении(Элемент)
     РассчитатьРозничнуюЦену();
КонецПроцедуры

&НаКлиенте
Процедура РассчитатьРозничнуюЦену()
     Строка = Элементы.ПриходнаяНакладная.ТекущиеДанные;
     Строка.РозничнаяЦена = Строка.Цена + (Строка.Цена *
Строка.ПроцентНаценки / 100);
КонецПроцедуры
```

Листинг А.2 – Код модуля объекта документа «Приходная накладная»

```
Процедура ОбработкаПроведения(Отказ, Режим)

     // регистр ОстаткиСерий Приход
     Движения.ОстаткиСерий.Записывать = Истина;
     Для Каждого ТекСтрокаПриходнаяНакладная Из
ПриходнаяНакладная Цикл
          Движение = Движения.ОстаткиСерий.Добавить();
          Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
          Движение.Период = Дата;
          Движение.Отдел = Отдел;
          Движение.Серия = ТекСтрокаПриходнаяНакладная.Серия;
          Движение.Количество =
ТекСтрокаПриходнаяНакладная.Количество;
```

```
          Движение.Себестоимость =
ТекСтрокаПриходнаяНакладная.Сумма;

          // регистр Цены
          Движения.Цены.Записывать = Истина;
          ЗакупочнаяЦенаИПроцентНаценки =
РаботаСДанными.ПолучитьЗакупочнуюЦену(Дата,
ТекСтрокаПриходнаяНакладная.Медикамент);
          Если ТекСтрокаПриходнаяНакладная.Цена <>
ЗакупочнаяЦенаИПроцентНаценки.ЗакупочнаяЦена ИЛИ
               ТекСтрокаПриходнаяНакладная.ПроцентНаценки <>
ЗакупочнаяЦенаИПроцентНаценки.ПроцентНаценки Тогда

               Движение = Движения.Цены.Добавить();
               Движение.Период = Дата;
               Движение.Медикамент =
ТекСтрокаПриходнаяНакладная.Медикамент;
               Движение.ЗакупочнаяЦена =
ТекСтрокаПриходнаяНакладная.Цена;
               Движение.ПроцентНаценки =
ТекСтрокаПриходнаяНакладная.ПроцентНаценки;
          КонецЕсли;

     КонецЦикла;

КонецПроцедуры
```

Листинг А.3 – Код модуля формы документа «Чеки»

```
&НаКлиенте
Процедура ЧекиМедикаментПриИзменении(Элемент)
     Строка = Элементы.Чеки.ТекущиеДанные;
     Строка.Цена =
РаботаСДанными.ПолучитьРозничнуюЦену(Объект.Дата,
Строка.Медикамент);
     СерияМедикамента =
НайтиОстаткиСерийМедикамента(Строка.Медикамент);
     //Если осталась только одна серия, то она устанавливется в
графу "Серия", иначе в графу ничего ничего не выводится
     Строка.Серия = ?(СерияМедикамента = Неопределено,
Неопределено, СерияМедикамента);
КонецПроцедуры

&НаКлиенте
Процедура ЧекиКоличествоПриИзменении(Элемент)
     РаботаСДокументами.РассчитатьСумму(Элементы.Чеки.ТекущиеДанн
ые);
КонецПроцедуры

&НаКлиенте
Процедура ЧекиЦенаПриИзменении(Элемент)
     РаботаСДокументами.РассчитатьСумму(Элементы.Чеки.ТекущиеДанн
ые);
```

```
КонецПроцедуры

&НаСервере
Функция НайтиОстаткиСерийМедикамента (Медикамент)

     Элементы.ЧекиСерия.СписокВыбора.Очистить();

     Запрос = Новый Запрос;
     Запрос.Текст =
     "ВЫБРАТЬ
     |    ОстаткиСерийОстатки.Серия КАК Серия,
     |    Серии.ПредельнаяДатаРеализации КАК
ПредельнаяДатаРеализации,
     |    ЕСТЬNULL(ОстаткиСерийОстатки.КоличествоОстаток, 0) КАК
КоличествоОстаток
     |ИЗ
     |    РегистрНакопления.ОстаткиСерий.Остатки КАК
ОстаткиСерийОстатки
     |         ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Серии КАК Серии
     |         ПО ОстаткиСерийОстатки.Серия = Серии.Ссылка
     |ГДЕ
     |    Серии.Владелец = &Медикамент
     |    И ОстаткиСерийОстатки.Отдел = &Отдел";

     Запрос.УстановитьПараметр("Медикамент", Медикамент);
     Запрос.УстановитьПараметр("Отдел", Объект.Отдел);
     РезультатЗапроса = Запрос.Выполнить();
     ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

     // Если осталась только одна серия у медикамента ее
наименование появляется в графе «Серия"
     Если ВыборкаДетальныеЗаписи.Количество() = 1 И
ВыборкаДетальныеЗаписи.Следующий()Тогда
          Возврат ВыборкаДетальныеЗаписи.Серия;
     КонецЕсли;
     //Если осталось несколько серий, то формируется список
значений для выбора, в котором высвечивается  номер серии,
предельная дата реализации и количество (остаток) в отделе
     Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

     Элементы.ЧекиСерия.СписокВыбора.Добавить(ВыборкаДетальныеЗап
иси.Серия, "Серия - " + Строка(ВыборкаДетальныеЗаписи.Серия)+",
предельная дата реализации - " +
Строка(ВыборкаДетальныеЗаписи.ПредельнаяДатаРеализации)+",
количество(остаток) - " +
Строка(ВыборкаДетальныеЗаписи.КоличествоОстаток));
     КонецЦикла;

КонецФункции

&НаКлиенте
Процедура ЧекиПередНачаломДобавления(Элемент, Отказ, Копирование,
Родитель, Группа, Параметр)
     Если Объект.Отдел.Пустая() Тогда
```

```
          Отказ = Истина;
           Сообщение = Новый СообщениеПользователю;
           Сообщение.Текст = "Перед началом добавления чеков
необходимо заполнить реквизит ""Отдел"".";
           Сообщение.Сообщить();
     КонецЕсли;
КонецПроцедуры

&НаКлиенте
Процедура ЧекиПриИзменении(Элемент)
     Если Объект.Чеки.Количество() > 1 Тогда
          Элементы.Отдел.Доступность = Ложь;
      Иначе
           Элементы.Отдел.Доступность = Истина;
     КонецЕсли;
КонецПроцедуры

&НаКлиенте
Процедура ПробитьЧек(Команда)
     ЭтаФорма.Записать(Новый Структура("РежимЗаписи",
РежимЗаписиДокумента.Проведение));
КонецПроцедуры

&НаКлиенте
Процедура ПродажиЗаСмену(Команда)
     //формирование отчета "Проданные медикаменты" с передачей
параментра - ссылки чека (текущий отрытый документ)
     УсловиеОтбора = Новый
Структура("Регистратор",Объект.Ссылка);
     ПараметрыОтчета =  Новый
Структура("Отбор,СформироватьПриОткрытии", УсловиеОтбора,
Истина);
     ОткрытьФорму("Отчет.ПроданныеМедикаменты.Форма",ПараметрыОтч
ета);
КонецПроцедуры
```

Листинг А.4 – Код модуля объекта документа «Чеки»

```
Процедура ОбработкаПроведения(Отказ, РежимПроведения)

     Запрос = Новый Запрос;
     Запрос.Текст =
     "ВЫБРАТЬ
     |    ЧекиЧеки.НомерСтроки КАК НомерСтроки,
     |    ЧекиЧеки.Медикамент КАК Медикамент,
     |    ЧекиЧеки.Серия КАК Серия,
     |    ЕСТЬNULL(ЧекиЧеки.Количество, 0) КАК Количество,
     |    ЕСТЬNULL(ЧекиЧеки.Сумма, 0) КАК Сумма,
     |    ЕСТЬNULL(ОстаткиСерийОстатки.КоличествоОстаток, 0) КАК
КоличествоОстаток,
     |    ЕСТЬNULL(ОстаткиСерийОстатки.СебестоимостьОстаток, 0)
КАК СебестоимостьОстаток
     |ИЗ
```

```
     |    Документ.Чеки.Чеки КАК ЧекиЧеки
     |         ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ОстаткиСерий.Остатки КАК ОстаткиСерийОстатки
     |         ПО ЧекиЧеки.Ссылка.Отдел =
ОстаткиСерийОстатки.Отдел
     |              И ЧекиЧеки.Серия = ОстаткиСерийОстатки.Серия
     |ГДЕ
     |    ЧекиЧеки.Ссылка = &Ссылка
     |    И ЧекиЧеки.Проведено = ЛОЖЬ";

     Запрос.УстановитьПараметр("Ссылка", Ссылка);
     РезультатЗапроса = Запрос.Выполнить();
     Выборка = РезультатЗапроса.Выбрать();

     НачалоОбработкиСтрокВыборки = Ложь;

     Пока Выборка.Следующий() Цикл

          Если Выборка.Количество > Выборка.КоличествоОстаток
Тогда
               Отказ = Истина;
               Сообщение = Новый СообщениеПользователю;
               Сообщение.Текст = "Строка " + Выборка.НомерСтроки
+ ". В отделе """ + Отдел + """ не хватает медикамента """ +
Выборка.Медикамент + """ серии """ + Выборка.Серия + """ в
количестве " + (Выборка.Количество - Выборка.КоличествоОстаток)
+ " ед.";
               Сообщение.Сообщить();
          КонецЕсли;

          Если Не Отказ Тогда

               СтрокаТЧ = Чеки.Получить(Выборка.НомерСтроки-1);
               СтрокаТЧ.Проведено = Истина;

               //поиск максимального значения из чеков (самого
последнего пробитого чека) и установка новых номеров чеков
               НомерПредыдущегоЧека =
РаботаСДанными.НайтиНомерПредыдущегоЧека(Чеки.ВыгрузитьКолонку("
НомерЧека"));
               Если Не НачалоОбработкиСтрокВыборки Тогда
                    НовыйНомерЧека = НомерПредыдущегоЧека + 1;
                    НачалоОбработкиСтрокВыборки = Истина;
               КонецЕсли;
               СтрокаТЧ.НомерЧека = НовыйНомерЧека;

               // регистр ОстаткиСерий Расход
               Движение = Движения.ОстаткиСерий.Добавить();
               Движение.ВидДвижения =
ВидДвиженияНакопления.Расход;
               Движение.Регистратор = Ссылка;
               Движение.Период = Дата;
               Движение.Отдел = Отдел;
               Движение.Серия = Выборка.Серия;
```

```
                Движение.Количество = Выборка.Количество;
                Движение.Себестоимость =
Выборка.СебестоимостьОстаток / Выборка.КоличествоОстаток *
Выборка.Количество;
                Движения.ОстаткиСерий.Записать(Ложь);

                // регистр Продажи
                Движение = Движения.Продажи.Добавить();
                Движение.Период = Дата;
                Движение.Регистратор = Ссылка;
                Движение.Отдел = Отдел;
                Движение.Медикамент = Выборка.Медикамент;
                Движение.Количество = Выборка.Количество;
                Движение.Стоимость = Выборка.Сумма;
                Движения.Продажи.Записать(Ложь);

          КонецЕсли;

     КонецЦикла;
     //запись измененных реквизитов формы
     Записать(РежимЗаписиДокумента.Запись);
КонецПроцедуры

Процедура ОбработкаУдаленияПроведения(Отказ)
     Для каждого СтрокаТЧ Из Чеки Цикл
          СтрокаТЧ.Проведено = Ложь;
          СтрокаТЧ.НомерЧека = 0;
     КонецЦикла;
КонецПроцедуры
```

Листинг А.5 – Код модуля формы документа «День записи»

```
&НаКлиенте
Процедура ЗаписиВрачПриИзменении(Элемент)
     НоваяСтрока = Элементы.Записи.ТекущиеДанные;
     НоваяСтрока.Кабинет =
ЗаписиВрачПриИзмененииНаСервере(НоваяСтрока.Врач);
КонецПроцедуры

&НаСервере
Функция ЗаписиВрачПриИзмененииНаСервере(Врач)

     Запрос = Новый Запрос;
     Запрос.Текст =
          "ВЫБРАТЬ
          |    Врач.Кабинет КАК Кабинет
          |ИЗ
          |    Справочник.Врач КАК Врач
          |ГДЕ
          |    Врач.Ссылка = &Ссылка";

     Запрос.УстановитьПараметр("Ссылка", Врач);
```

```
     РезультатЗапроса = Запрос.Выполнить();

     ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

     если ВыборкаДетальныеЗаписи.Следующий() тогда
          Возврат ВыборкаДетальныеЗаписи.Кабинет;
     Конецесли;

КонецФункции
```

Листинг А.6 – Код модуля формы справочника «Карточка клиента»

```
&НаСервере
Процедура ЗаказМедикаментовНаСервере()


     НовыйДокумент =
Документы.ЗаказыМедикаментов.СоздатьДокумент();
     НовыйДокумент.Дата = ТекущаяДата();
     НовыйДокумент.Покупатель = Объект.Наименование;
     НовыйДокумент.Отдел = Объект.Отдел;

          //{{КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА
     // Данный фрагмент построен конструктором.
     // При повторном использовании конструктора, внесенные
вручную изменения будут утеряны!!!

     Запрос = Новый Запрос;
     Запрос.Текст =
          "ВЫБРАТЬ
          |    КарточкаяКлиентаЗаказМедикаментов.Медикамент КАК
Медикамент,
          |    КарточкаяКлиентаЗаказМедикаментов.Серия КАК Серия,
          |    КарточкаяКлиентаЗаказМедикаментов.Количество КАК
Количество,
          |    КарточкаяКлиентаЗаказМедикаментов.Цена КАК Цена,
          |    КарточкаяКлиентаЗаказМедикаментов.Сумма КАК Сумма
          |ИЗ
          |    Справочник.КарточкаяКлиента.ЗаказМедикаментов КАК
КарточкаяКлиентаЗаказМедикаментов";

     РезультатЗапроса = Запрос.Выполнить();

     ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

     Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
          СтрокаТ = НовыйДокумент.Медикаменты.Добавить();
          СтрокаТ.Медикамент = ВыборкаДетальныеЗаписи.Медикамент;
          СтрокаТ.Серия = ВыборкаДетальныеЗаписи.Серия;
          СтрокаТ.Количество = ВыборкаДетальныеЗаписи.Количество;
          СтрокаТ.Цена = ВыборкаДетальныеЗаписи.Цена;
          СтрокаТ.Сумма = ВыборкаДетальныеЗаписи.Сумма;
// Вставить обработку выборки ВыборкаДетальныеЗаписи
```

```
    КонецЦикла;


    НовыйДокумент.Записать();

КонецПроцедуры

&НаКлиенте
Процедура ЗаказМедикаментов(Команда)
    ЗаказМедикаментовНаСервере();
КонецПроцедуры

&НаКлиенте
Процедура ЗаказМедикаментовМедикаментПриИзменении(Элемент)
    Строка = Элементы.ЗаказМедикаментов.ТекущиеДанные;
    Строка.Цена = РаботаСДанными.ПолучитьРозничнуюЦену(Дата,
Строка.Медикамент);
    СерияМедикамента =
НайтиОстаткиСерийМедикамента(Строка.Медикамент);
    //Если осталась только одна серия, то она устанавливется в
графу "Серия", иначе в графу ничего ничего не выводится
    Строка.Серия = ?(СерияМедикамента = Неопределено,
Неопределено, СерияМедикамента);
КонецПроцедуры

&НаСервере
Функция НайтиОстаткиСерийМедикамента (Медикамент)

    Элементы.ЗаказМедикаментовСерия.СписокВыбора.Очистить();

    Запрос = Новый Запрос;
    Запрос.Текст =
    "ВЫБРАТЬ
    |    ОстаткиСерийОстатки.Серия КАК Серия,
    |    Серии.ПредельнаяДатаРеализации КАК
ПредельнаяДатаРеализации,
    |    ЕСТЬNULL(ОстаткиСерийОстатки.КоличествоОстаток, 0) КАК
КоличествоОстаток
    |ИЗ
    |    РегистрНакопления.ОстаткиСерий.Остатки КАК
ОстаткиСерийОстатки
    |        ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Серии КАК Серии
    |        ПО ОстаткиСерийОстатки.Серия = Серии.Ссылка
    |ГДЕ
    |    Серии.Владелец = &Медикамент
    |    И ОстаткиСерийОстатки.Отдел = &Отдел";

    Запрос.УстановитьПараметр("Медикамент", Медикамент);
    Запрос.УстановитьПараметр("Отдел", Объект.Отдел);
    РезультатЗапроса = Запрос.Выполнить();
    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

    // Если осталась только одна серия у медикамента ее
наименование появляется в графе «Серия"
```

```
     Если ВыборкаДетальныеЗаписи.Количество() = 1 И
ВыборкаДетальныеЗаписи.Следующий()Тогда
          Возврат ВыборкаДетальныеЗаписи.Серия;
     КонецЕсли;
     //Если осталось несколько серий, то формируется список
значений для выбора, в котором высвечивается  номер серии,
предельная дата реализации и количество (остаток) в отделе
     Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

     Элементы.ЗаказМедикаментовСерия.СписокВыбора.Добавить(Выборк
аДетальныеЗаписи.Серия, "Серия - " +
Строка(ВыборкаДетальныеЗаписи.Серия)+", предельная дата
реализации - " +
Строка(ВыборкаДетальныеЗаписи.ПредельнаяДатаРеализации)+",
количество(остаток) - " +
Строка(ВыборкаДетальныеЗаписи.КоличествоОстаток));
     КонецЦикла;

КонецФункции

&НаКлиенте
Процедура ЗаказМедикаментовКоличествоПриИзменении(Элемент)
     РаботаСДокументами.РассчитатьСумму(Элементы.ЗаказМедикаменто
в.ТекущиеДанные);
КонецПроцедуры

&НаКлиенте
Процедура ЗаказМедикаментовЦенаПриИзменении(Элемент)
     РаботаСДокументами.РассчитатьСумму(Элементы.ЗаказМедикаменто
в.ТекущиеДанные);
КонецПроцедуры
```

Листинг А.7 – Код общего модуля «Работа с данными»

```
Функция ПолучитьЗакупочнуюЦену(АктуальнаяДата, Медикамент)
Экспорт
     Отбор = Новый Структура("Медикамент", Медикамент);
     ЗакупочнаяЦенаИПроцентНаценки =
РегистрыСведений.Цены.ПолучитьПоследнее(АктуальнаяДата, Отбор);
     Возврат ЗакупочнаяЦенаИПроцентНаценки;
КонецФункции

Функция ПолучитьРозничнуюЦену(АктуальнаяДата, Медикамент)
Экспорт
     Отбор = Новый Структура("Медикамент", Медикамент);
     ЗакупочнаяЦенаИПроцентНаценки =
РегистрыСведений.Цены.ПолучитьПоследнее(АктуальнаяДата, Отбор);
     РозничнаяЦена = ЗакупочнаяЦенаИПроцентНаценки.ЗакупочнаяЦена
+ (ЗакупочнаяЦенаИПроцентНаценки.ЗакупочнаяЦена *
ЗакупочнаяЦенаИПроцентНаценки.ПроцентНаценки /100);
     Возврат РозничнаяЦена;
КонецФункции
```

Приложение Б

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

(обязательное)

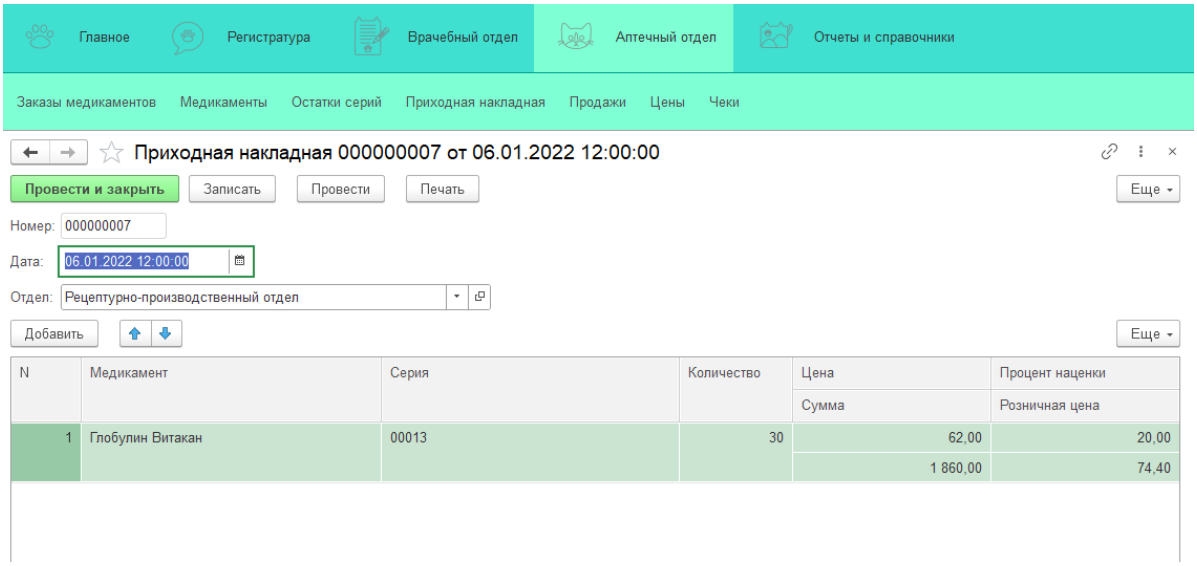



Рисунок Б.1 – Создание документа «Приходная накладная»

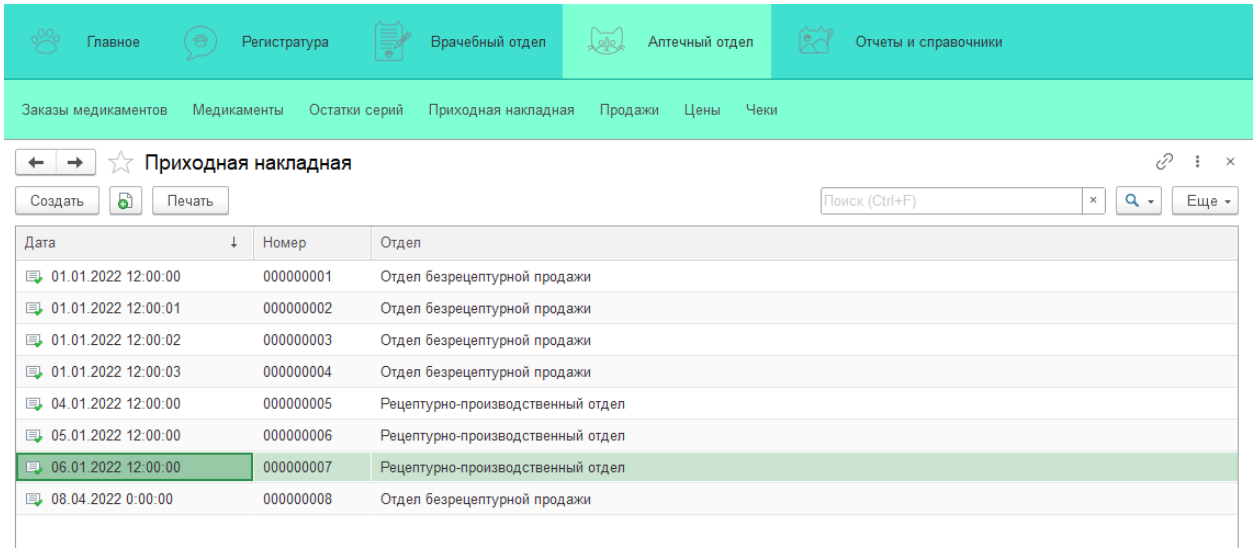



Рисунок Б.2 – Форма списка документа «Приходная накладная»

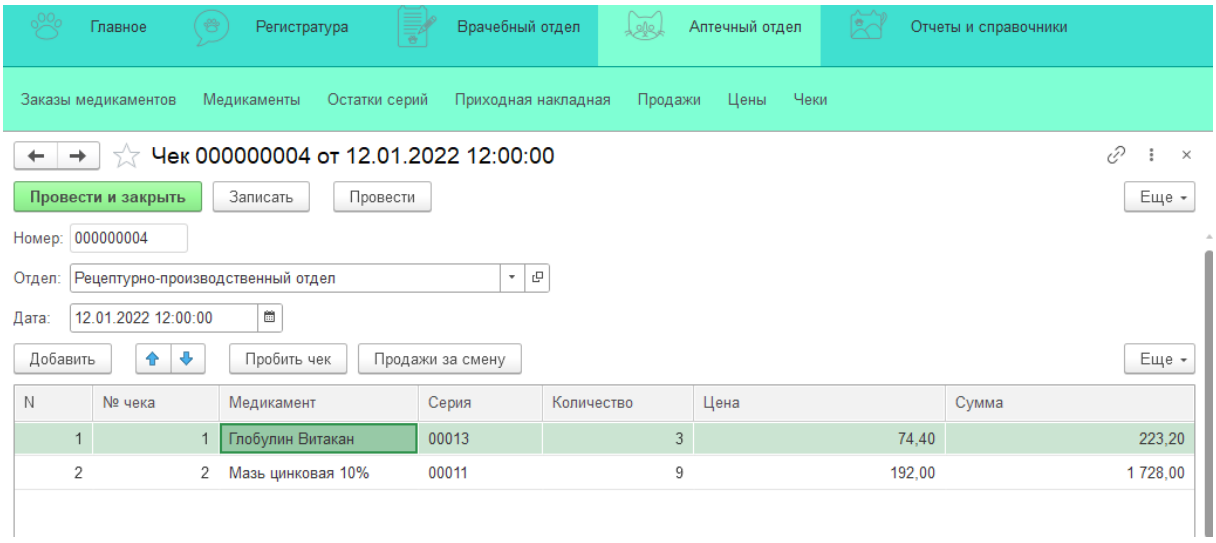



Рисунок Б.3 – Создание документа «Чеки»

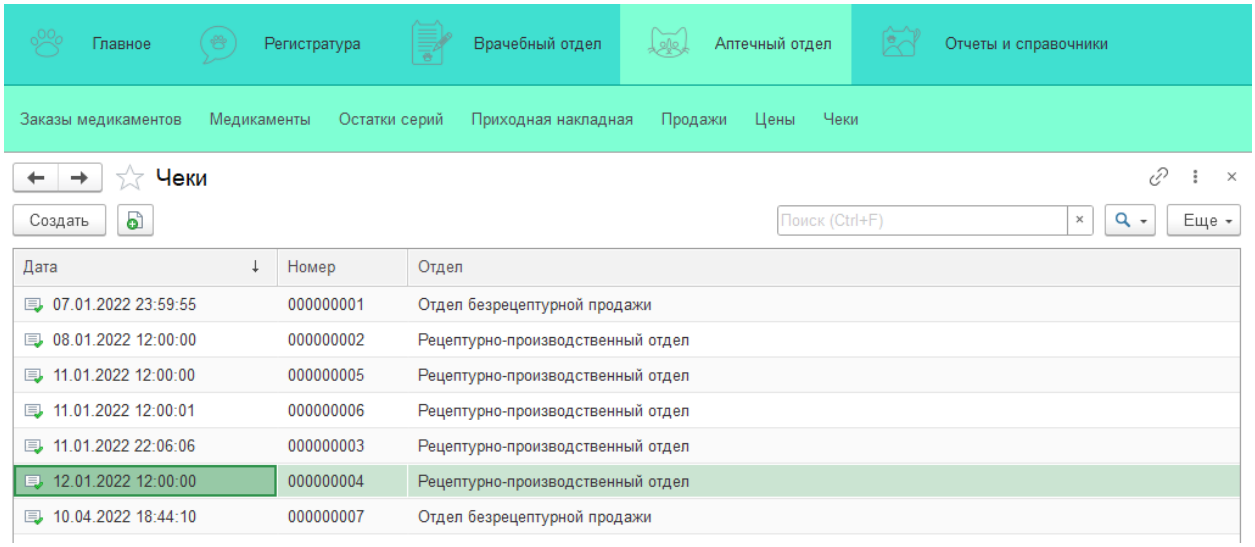



Рисунок Б.4 – Форма списка документа «Чеки»

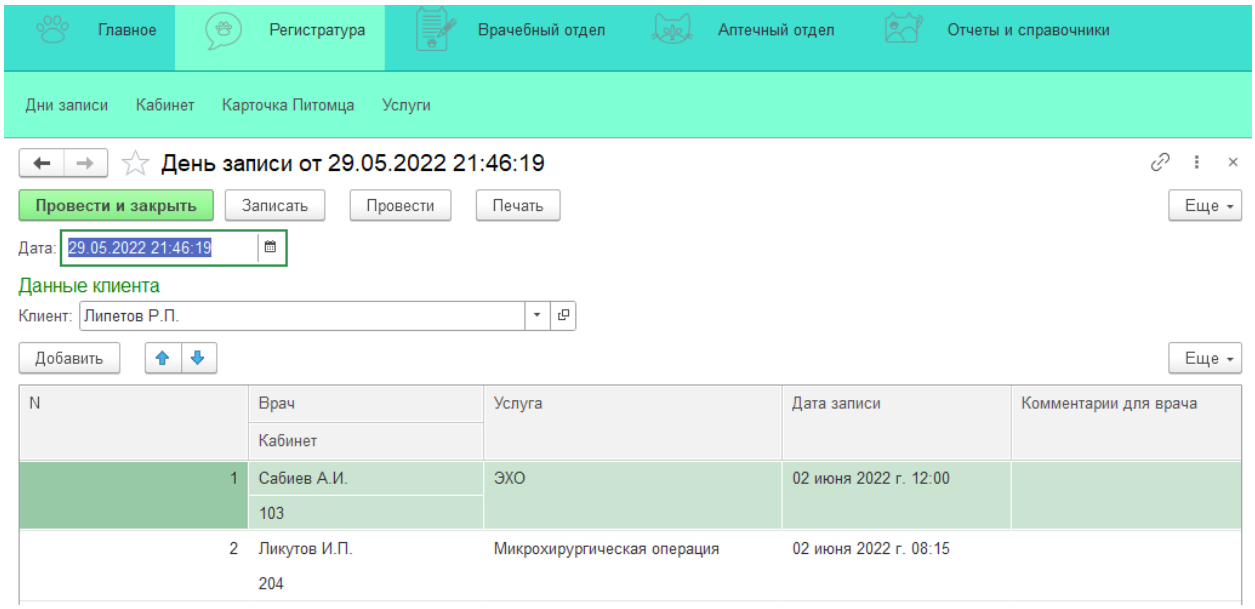



Рисунок Б.5 – Создание документа «День записи»

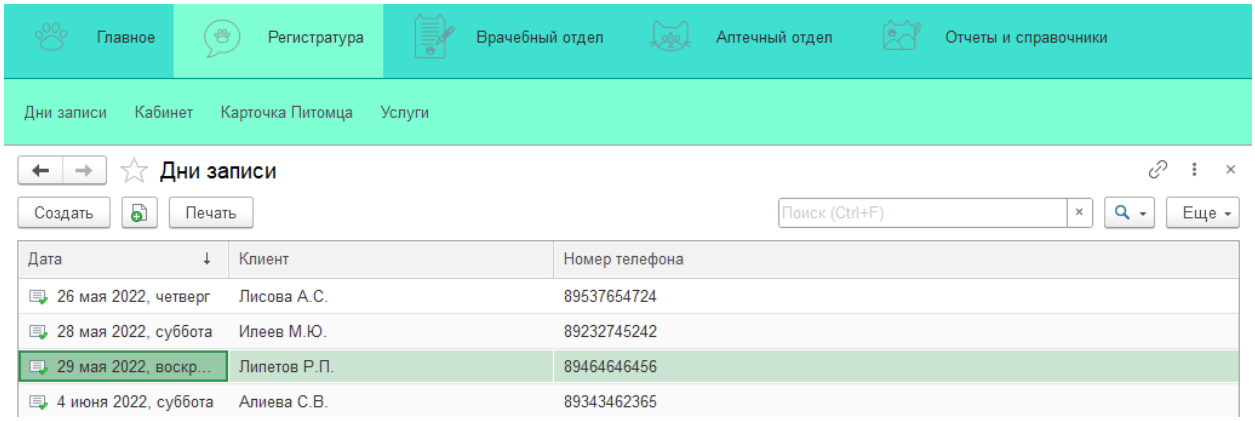



Рисунок Б.6 – Форма списка документа «День записи»

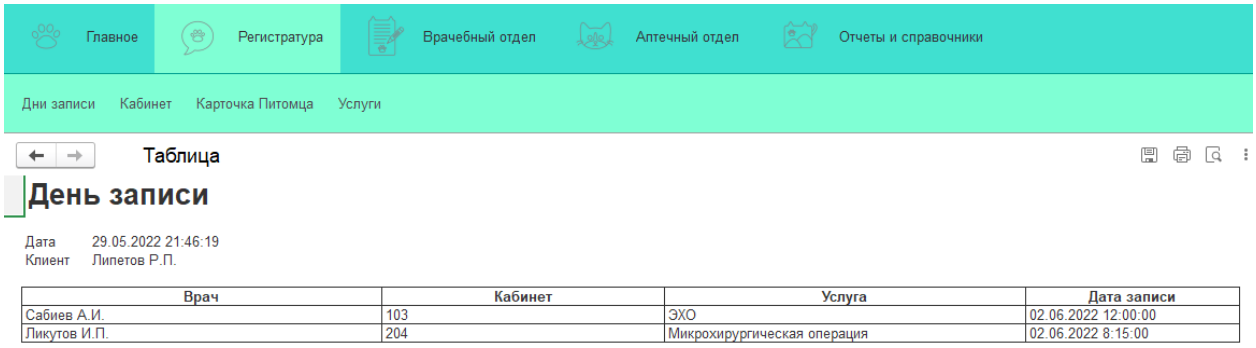



Рисунок Б.7 – Печатная форма документа «День записи»

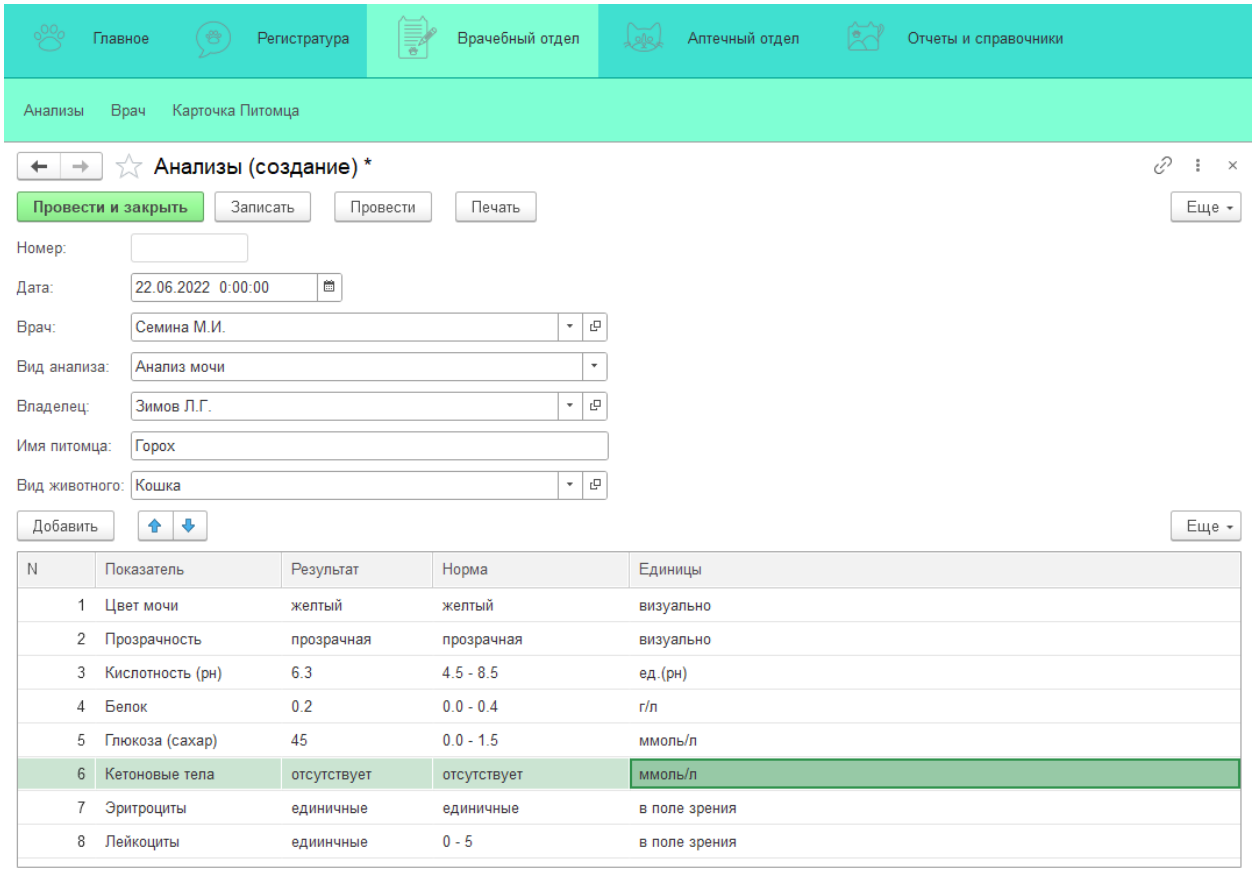



Рисунок Б.8 – Создание документа «Анализы»

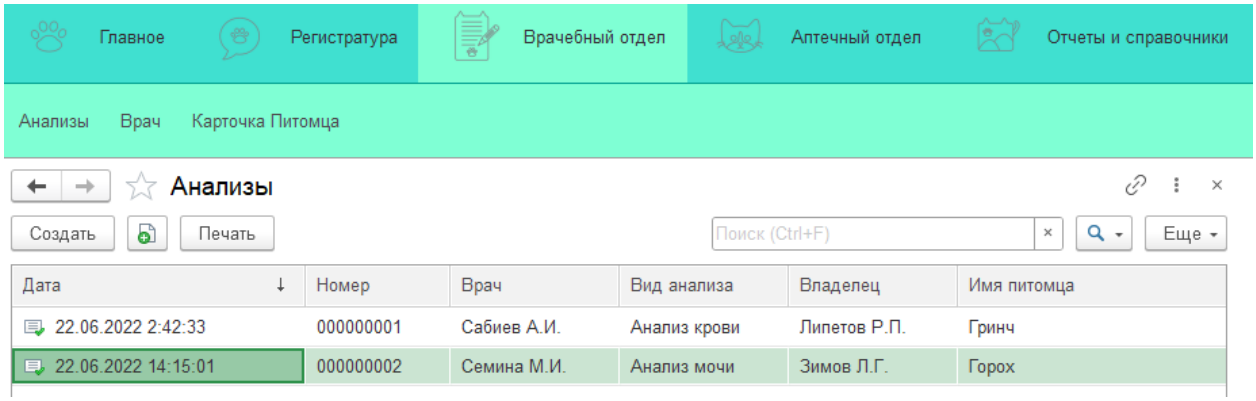



Рисунок Б.9 – Форма списка документа «Анализы»

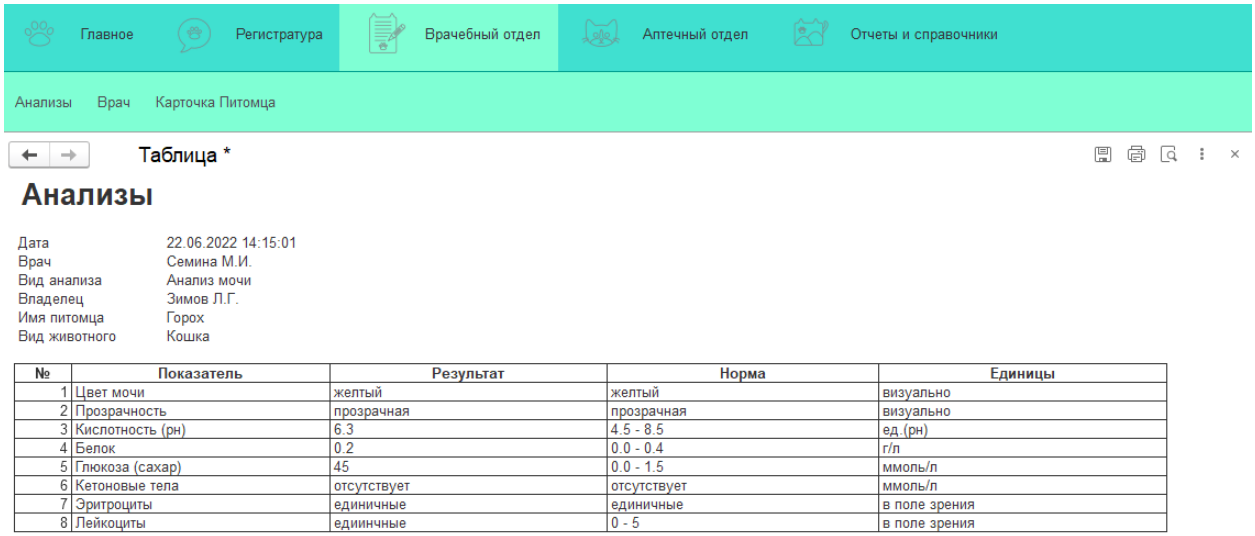



Рисунок Б.10 – Печатная форма документа «Анализы»

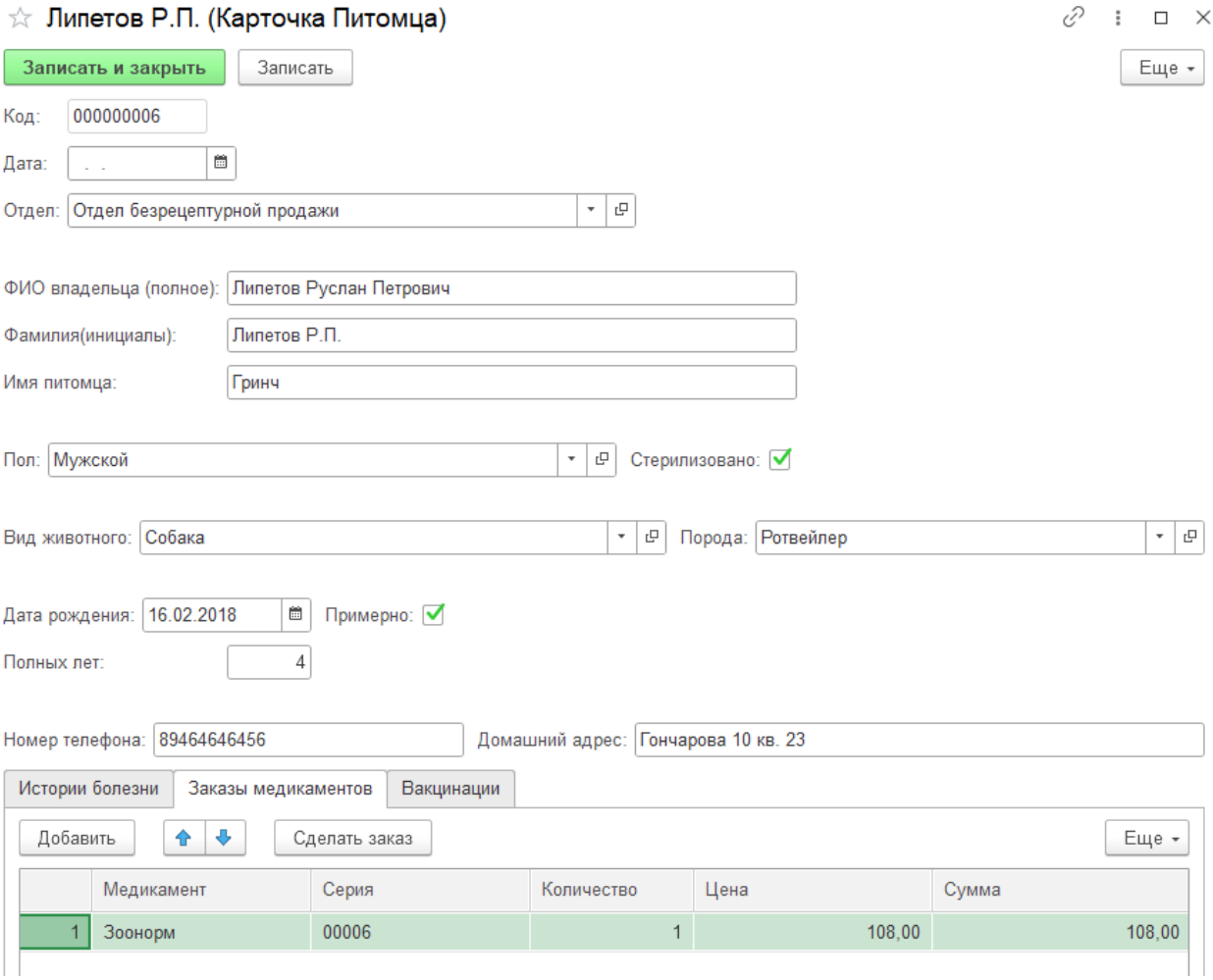



Рисунок Б.11 – Создание документа «Заказы медикаментов» через «Карточка питомца»

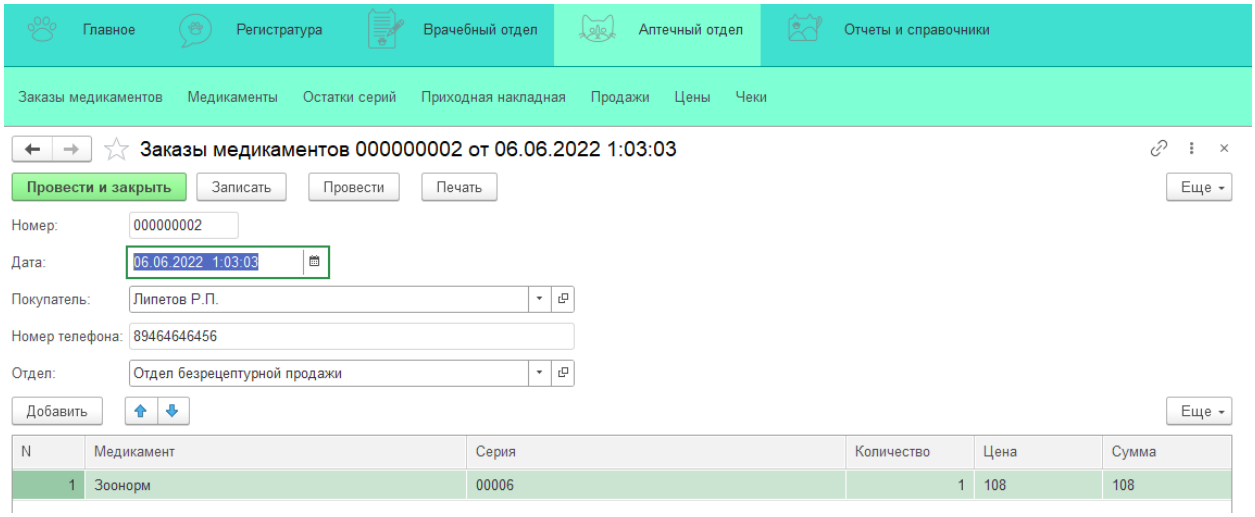



Рисунок Б.12 – Форма созданного документа «Заказы медикаментов»

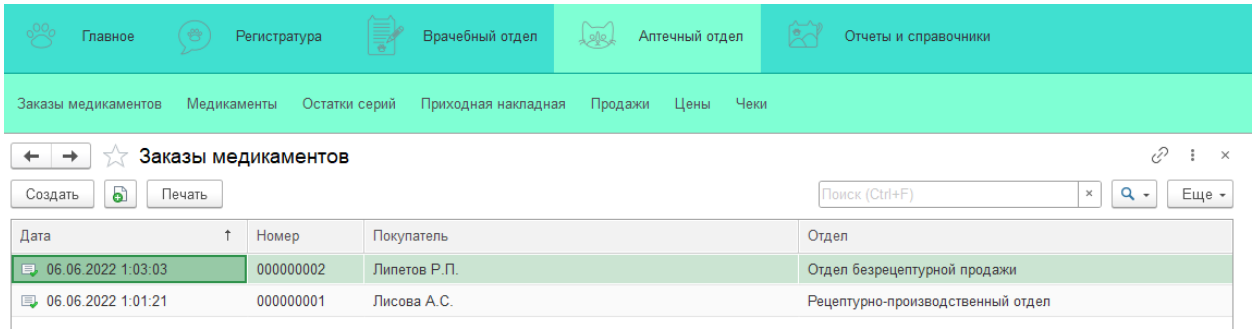



Рисунок Б.13 – Форма списка документа «Заказы медикаментов»

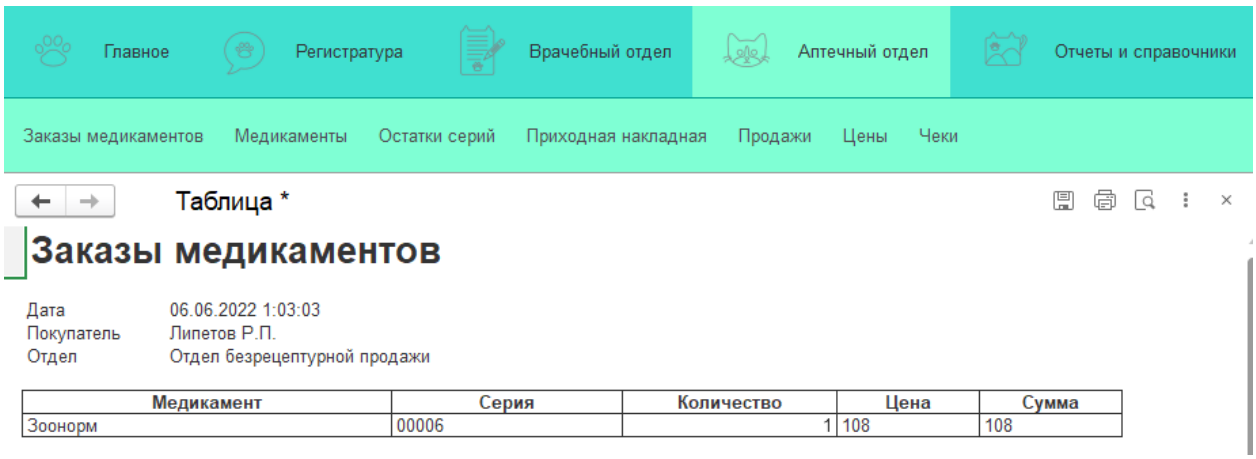



Рисунок Б.14 – Печатная форма документа «Заказы медикаментов»