

**Колледж космического машиностроения и технологий**

**КУРСОВОЙ ПРОЕКТ**

по дисциплине МДК.02.02. Технология разработки и защиты баз данных

**РАЗРАБОТКА СИСТЕМЫ контроля и управления крупным сельскохозяйственным предприятием**

Пояснительная записка

КП.09.02.03.23.15ПЗ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Обучающийся группы | | П1-19 | Погуляев Ф.В. | |
| Руководитель курсового проекта | | | Стрельников С.Д. | |
| Результат защиты |  | | |  |

г. Королёв, 2023 г.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc131703192)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc131703193)

[1.1. Описание предметной области 5](#_Toc131703194)

[1.2. Описание существующей разработки 5](#_Toc131703195)

[2. ПРОЕКТНАЯ ЧАСТЬ 7](#_Toc131703196)

[2.1. Обоснование инструментов разработки 7](#_Toc131703197)

[2.2. Проектирование сценария работы проекта 11](#_Toc131703198)

[2.3. Разработка диаграммы классов 13](#_Toc131703199)

[2.4. Разработка ER-диаграммы баз данных 16](#_Toc131703200)

[2.5. Описание модулей проекта 19](#_Toc131703201)

[2.6. Тестирование проекта 19](#_Toc131703202)

[2.7. Руководство оператора 22](#_Toc131703203)

[ЗАКЛЮЧЕНИЕ 30](#_Toc131703204)

[Список используемых источников 31](#_Toc131703205)

# **ВВЕДЕНИЕ**

Тема курсового проекта – «Разработка системы контроля и управления крупным сельскохозяйственным предприятием

Актуальность данного курсового проекта заключается в повышении качества контроля и удобства доступа к данным о животных, что повышает общую скорость реагирования различных систем предприятия.

Целью данного курсового проекта является разработка клиентской части десктопного приложения, которое будет взаимодействовать с базой данных.

В первой части будет рассмотрена предметная область и существующие продукты по данной теме.

Во второй части будут описаны инструменты и модули, которые были разработаны и описано руководство оператора.

В заключительной части будет и заключение по курсовому проекту.

# **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## **Описание предметной области**

Целью данной работы является разработка десктопного приложения для электронного учета сельскохозяйственного скота, находящегося на балансе предприятия, а также ведения учета прививок. В данной ситуации предлагается следующее решение поставленной задачи:

Десктопное приложение в функционал которого входит:

* Возможность редактирования данных каждого животного;
* Управление базой данных через визуальный интерфейс;
* Фильтр, определяющий статус вакцинации животного;
* Составление отчета в .xls таблице.

## **Описание существующей разработки**

В этом разделе рассмотрены некоторые уже имеющиеся программы по данной тематике.

Практически все из них предоставляют похожий функционал:

* Управление базой данных;
* Запись заказов клиентов.

Примеры программ:

Один из примеров системы контроля и управления крупным сельскохозяйственным предприятием - это 1С:Предприятие 8. Управление сельскохозяйственным предприятием. Эта система разработана на базе 1С:Предприятие 8. Управление производственным предприятием и включает в себя помимо отраслевого функционала все возможности базовой конфигурации (Рисунок 1).

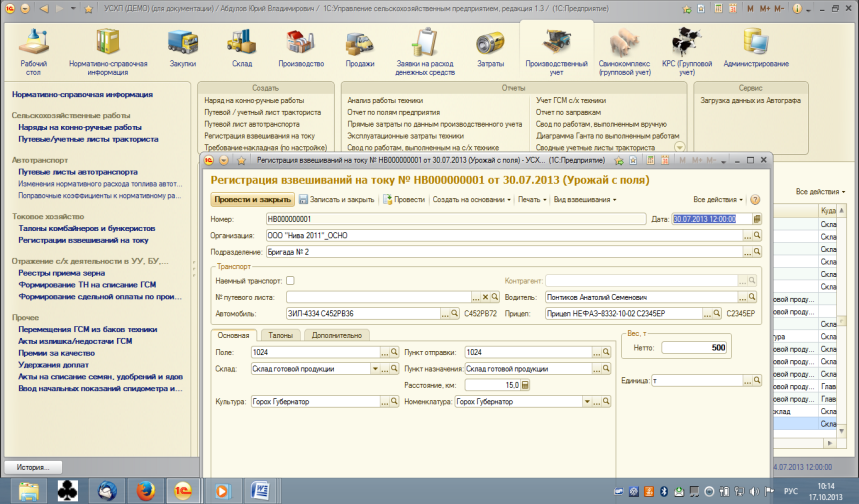


Рисунок 1. Интерфейс 1С: Предприятие 8. Управление сельскохозяйственным предприятием

Другой пример - это система «Soft.Farm» - комплексное IT-решение для агропроизводителей, которое объединяет данные из других систем в единый формат и создает прозрачную аналитическую систему сельскохозяйственной деятельности для принятия взвешенных управленческих решений.

Оба этих приложения имеют достаточно высокую стоимость подписки, перегруженный интерфейс и проблемы с настройкой и установкой.

# **ПРОЕКТНАЯ ЧАСТЬ**

## **Обоснование инструментов разработки**

Для создания проекта, основным инструментом был выбран PyCharm: the Python IDE, потому что он является одним из лучших инструментов для разработки на Python. Для разработки визуальной части интерфейса использовался популярный модуль для Python – PyQt 5. В качестве СУБД был выбран SQLite.

Язык Python был выбран по следующим факторам:

* Понятность кода – простое зрительное восприятие программ, написанных на этом языке.
* Наличие мощного фреймворка для разработки клиентских приложений;
* Наличие подробной документации.

Сравнение Python с другими языками программирования представлено в таблице 1.

Таблица 1. сравнения языков программирования

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **C++** | **С#** | **Python** |
| Читабельность | 5 | 7 | 8 |
| Простота | 4 | 6 | 8 |
| GUI | 9 | 8 | 7 |
| Скорость разработки | 5 | 6 | 8 |
| Знание языка | 5 | 3 | 8 |
| Итого баллов | 28 | 30 | 39 |

Причины, по которым был выбран данный инструмент и СУБД:

**PyCharm**

PyCharm – это среда разработки, разрабатываемая компанией JetBrains для разработки приложений на языках, работающих с JVM. Представляет собой комплекс интегрированных между собой инструментов программирования. Включает в себя:

* Редактор кода;
* Инструменты рефакторинга кода;
* Инструменты разработки Android-Приложений;
* Интеграцию с системой контроля версий Git.

Intellij IDEA делает разработку максимально продуктивной благодаря функциям автодополнения и анализа кода, мгновенной подсветке ошибок и быстрым исправлениям. Автоматические рефакторинги помогают эффективно редактировать код, а удобная навигация позволяет мгновенно перемещаться по проекту.

Возможности форматирования и настройки стиля кода помогают писать понятный код, который легко поддерживать.

**PyQt 5**

PyQt 5 - это набор инструментов для создания графических пользовательских интерфейсов (GUI) на языке программирования Python. PyQt 5 предоставляет разработчикам гибкую и мощную платформу для создания приложений с интуитивно понятным и привлекательным интерфейсом.

PyQt 5 был выпущен в 2016 году и представляет собой обновленную версию PyQt 4, предназначенную для работы с Python 3. Он предоставляет множество новых функций и улучшений, включая поддержку многопоточности, возможность создания анимированных интерфейсов и многое другое.

Одной из особенностей PyQt 5 является его интеграция с Qt Designer - инструментом для создания графических интерфейсов без необходимости писать код. Qt Designer позволяет разработчикам создавать пользовательские интерфейсы, используя визуальный редактор, и экспортировать их в код Python.

PyQt 5 также предоставляет множество виджетов для создания графических элементов интерфейса, таких как кнопки, поля ввода, таблицы, списки и многое другое. Разработчики могут легко настраивать их свойства и методы, чтобы создавать уникальные интерфейсы, которые соответствуют требованиям их приложений.

Для удобства работы с PyQt 5, в него встроена документация, которая доступна как в онлайн-режиме, так и в виде локальной справки. Это позволяет разработчикам быстро находить нужную информацию и решать проблемы, возникающие при работе с библиотекой.

Кроме того, PyQt 5 имеет мощную систему сигналов и слотов, которая позволяет связывать различные элементы интерфейса и реагировать на события. Это делает процесс разработки более удобным и эффективным.

В целом, PyQt 5 является мощным инструментом для создания графических пользовательских интерфейсов на Python. Он предоставляет разработчикам гибкую и интуитивно понятную платформу для создания приложений с привлекательным интерфейсом, что делает его популярным выбором для разработчиков по всему миру.

**Qt Designer**

Qt Designer - это графический инструмент для создания пользовательских интерфейсов в PyQt 5. Он позволяет разработчикам создавать интерфейсы без необходимости писать код, используя визуальный редактор (Рисунок 2).

Преимущества Qt Designer:

1. Быстрота и простота создания интерфейсов. С помощью Qt Designer можно создавать интерфейсы за несколько минут, что позволяет сократить время разработки приложений.
2. Возможность предварительного просмотра интерфейсов. Разработчики могут видеть, как будет выглядеть интерфейс на практике, что позволяет быстро вносить изменения и улучшать его.
3. Интеграция с PyQt 5. Qt Designer интегрирован непосредственно в PyQt 5, что позволяет разработчикам экспортировать созданные интерфейсы в код Python и использовать их в своих приложениях.
4. Наличие множества готовых элементов интерфейса. Qt Designer предоставляет множество готовых элементов интерфейса, таких как кнопки, поля ввода, таблицы и многое другое, что упрощает процесс создания интерфейса.
5. Возможность создания красивых и современных интерфейсов. Qt Designer позволяет создавать интерфейсы с помощью различных стилей и тем, что позволяет создавать красивые и современные интерфейсы.

В целом, Qt Designer является мощным инструментом для создания пользовательских интерфейсов в PyQt 5. Он позволяет разработчикам быстро и легко создавать интерфейсы, что ускоряет процесс разработки приложений.

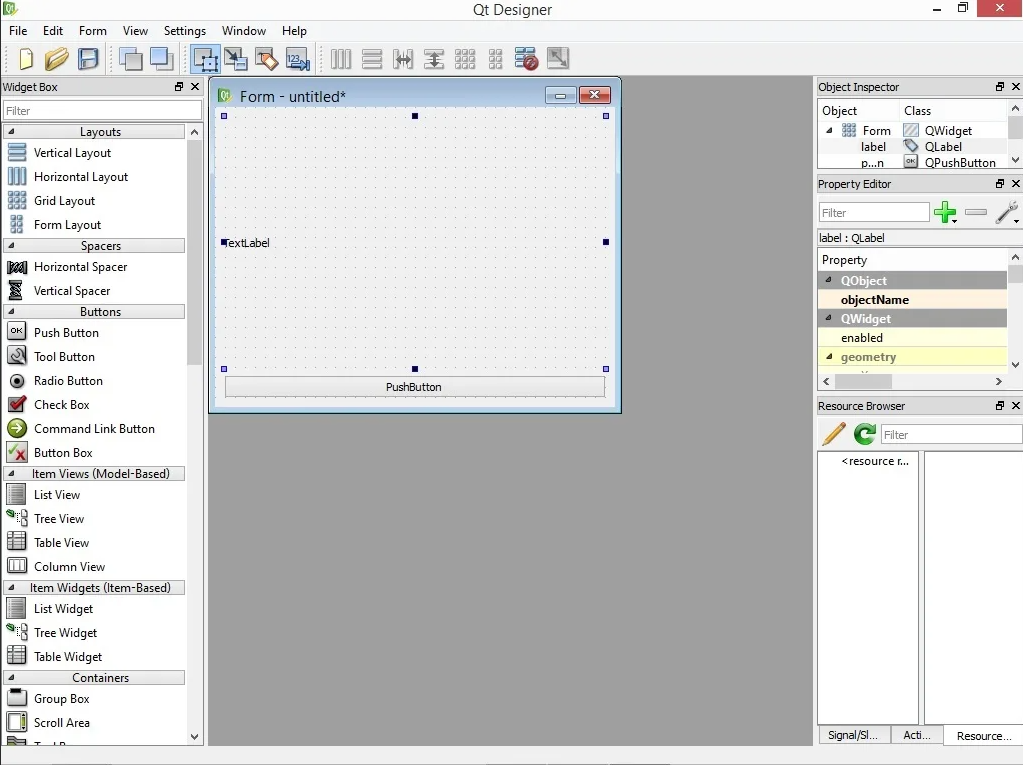


Рисунок 2. Главное окно Qt Designer

**SQLite**

SQLite - это библиотека, которая реализует маленький, быстрый, самодостаточный, высоконадежный, полнофункциональный движок базы данных SQL. SQLite является самым используемым движком базы данных в мире. Он встроен во все мобильные телефоны и большинство компьютеров и поставляется в комплекте со многими другими приложениями, которые люди используют каждый день.

Один из основных преимуществ SQLite — это его компактность и самодостаточность. Он не требует отдельного сервера или конфигурации и может быть легко встроен в другие приложения. Это делает его идеальным для использования в мобильных приложениях или в других сценариях, где требуется легковесная база данных.

Однако у SQLite есть и некоторые ограничения. Например, он не поддерживает многопользовательский доступ к данным на уровне сервера. Это означает, что, если несколько пользователей пытаются одновременно изменить данные в базе данных SQLite, может возникнуть конфликт.

## **Проектирование сценария работы проекта**

Сценарий работы проекта - это последовательность действий, которые должны быть выполнены в процессе разработки приложения. Этот сценарий включает в себя такие этапы, как планирование, проектирование, написание кода, тестирование и внесение изменений.

Каждый этап включает в себя определенные задачи и действия, которые должны быть выполнены для достижения конечной цели - создания качественного и функционального приложения. Сценарий работы проекта также может включать в себя работу с клиентом или заказчиком, чтобы убедиться, что приложение отвечает всем их требованиям и ожиданиям.

Важно отметить, что сценарий работы проекта может быть изменен в процессе разработки приложения, если появятся новые требования или проблемы, которые необходимо решить. В целом, правильное планирование и выполнение сценария работы проекта помогает обеспечить успешную разработку приложения и его высокое качество.

Проектирование сценария работы проекта имеет множество плюсов, вот некоторые из них:

1. Четкое понимание целей проекта: сценарий работы помогает определить цели и ожидания проекта, что помогает управлять его ходом и достигать желаемых результатов.
2. Улучшение коммуникации: создание сценария работы позволяет улучшить коммуникацию между участниками проекта, так как все члены команды понимают, что ожидается от них и какие действия необходимы для достижения целей.
3. Улучшение планирования: сценарий работы помогает разработать более точный и реалистичный план работы, учитывая ресурсы, временные рамки и другие ограничения.
4. Снижение рисков: при проектировании сценария работы можно выявить потенциальные проблемы и риски, а также разработать планы действий для их предотвращения или минимизации.
5. Улучшение качества продукта: благодаря более точному планированию и управлению ходом проекта, качество продукта может быть улучшено, что положительно скажется на его конечном результате.
6. Экономия времени и ресурсов: сценарий работы помогает оптимизировать использование времени и ресурсов, что может привести к экономии бюджета проекта.
7. Улучшение управления проектом: сценарий работы является важным инструментом для управления проектом, он помогает организовать работу команды и достичь целей проекта в срок.

В данном разделе приведен сценарий использования программы пользователем приложения (Рисунок 3).

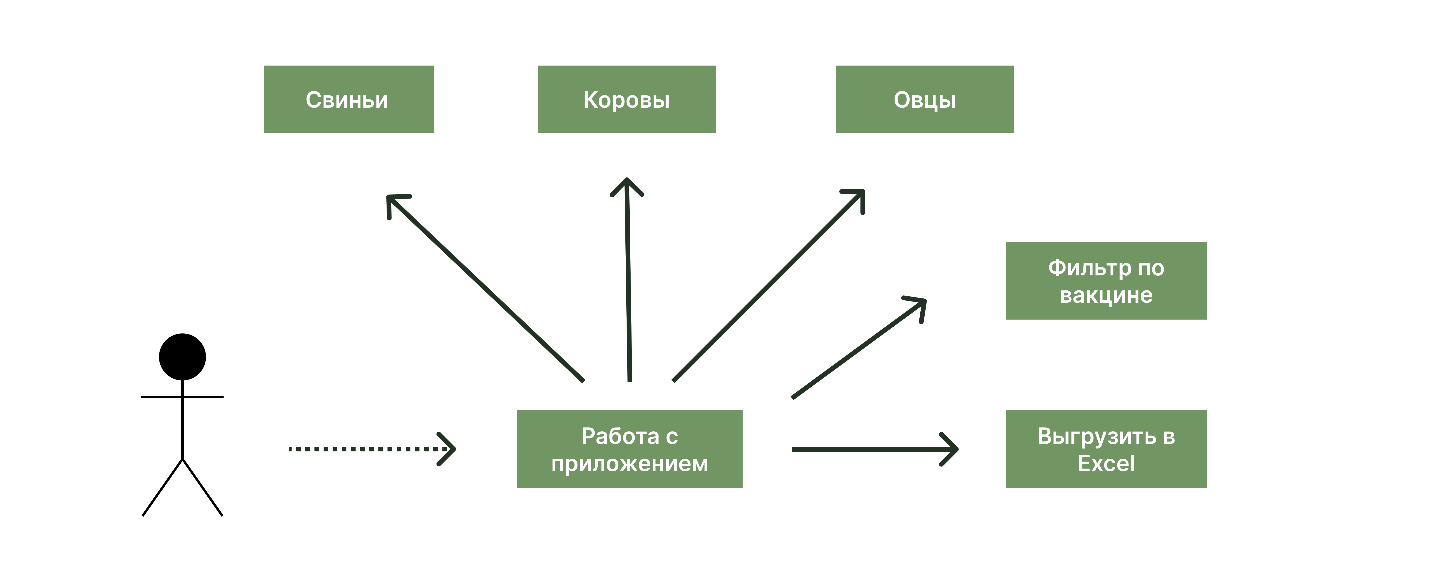


Рисунок 3. Сценарий использования

Пользователь после запуска приложения может выполнить следующие действия:

1. Управление имеющимся рогатым скотом;
2. Формирование отчета в Excel;
3. Добавление/удаление/редактирование данных в таблицах;
4. Отслеживание вакцинированного и не вакцинированного скота.

При выходе из программы, она завершает свою работу.

## **Разработка диаграммы классов**

Диаграмма классов - это графическое представление структуры классов в программе или системе. Она показывает отношения между классами, их атрибуты и методы. Диаграмма классов используется для проектирования объектно-ориентированных систем и приложений, позволяет легче понять и визуализировать структуру программы, определить связи между классами, а также упростить процесс разработки и поддержки кода. Кроме того, диаграмма классов может использоваться для документирования системы и обучения новых разработчиков.

Диаграмма классов состоит из прямоугольников, которые представляют классы, и стрелок, которые показывают связи между классами. Существуют различные типы связей, такие как наследование, агрегация, композиция и др.

На диаграмме классов можно указать атрибуты класса (переменные) и методы (функции), которые могут быть вызваны для работы с объектами этого класса. Также можно указать модификаторы доступа для атрибутов и методов, что позволяет контролировать доступ к ним из других классов.

Диаграмма классов является одной из основных диаграмм UML (Unified Modeling Language), которая используется в проектировании программных систем и приложений. Она помогает разработчикам лучше понимать структуру системы и ее компонентов, а также упрощает коммуникацию между членами команды разработки.

Кроме того, диаграмма классов может использоваться для создания кода. Например, с помощью инструментов CASE (Computer Aided Software Engineering) можно автоматически генерировать классы и методы на основе диаграммы классов.

В целом, диаграмма классов является важным инструментом в проектировании и разработке программных систем и приложений. Она помогает лучше понимать структуру системы и ее компонентов, а также упрощает процесс разработки, поддержки и документирования кода.

В данном разделе представлены все классы, использующиеся в проекте, а также их отношения между собой (Рисунок 4).

1. SystemCRM: класс содержит набор функций, импортируемых модулей, реализующих логику приложения, взаимодействие с БД;
   1. Dict\_animals - словарь с наименованиями животных;
   2. Dict\_genres – словарь с гендерами животных;
   3. Async\_init- точка вхождения в асинхронность;
   4. Combo\_tbl – производит загрузку таблицы, опираясь на значение ComboBox.
   5. Load\_date - загрузка данных с определенной таблицы из БД sql Database.db. Срабатывает при выборе таблицы из списка.
   6. Add\_new\_row – добавление новой строки. Срабатывает при нажатии на кнопку «Добавить»;
   7. Delete\_row – удаление выбранной строки. Срабатывает при нажатии на кнопку «Удалить»;
   8. Save\_data – сохранение данных из выбранной таблицы в таблицу sql. Срабатывает при нажатии на кнопку «Сохранить».
2. UiInterface: класс инициализирует виджеты и GUI приложения, производит их первоначальную настройку.
   1. main\_win – переменная содержит объект визуального интерфейса;
   2. setup\_ui – функция, создающая необходимые виджеты и стили CSS к ним.
3. Database: класс инициализирует базу данных, если она была не создана, наполняет ее шаблонными данными, содержит набор функцию для взаимодействия с БД.
   1. Db – путь к файлу БД;
   2. Tbl\_data – составляет и выполняет запрос на получение данных из таблицы, получаемой в аргументах;
   3. Data\_save – составляет и выполняет запрос на сохранение данных таблицы, получаемой в аргументах;
   4. Get\_animals – возвращает наименование всех видом животных.

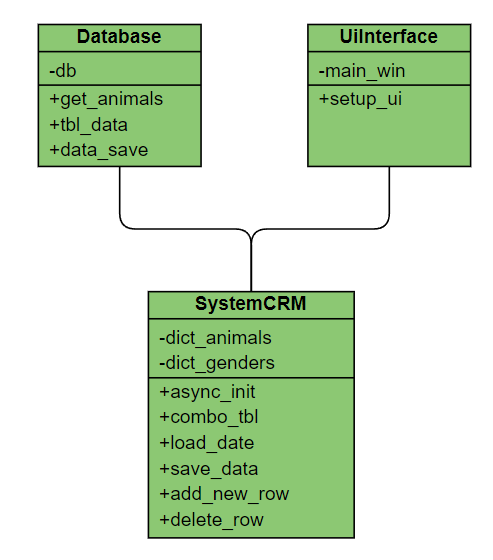


Рисунок 4. Диаграмма классов

## **Разработка ER-диаграммы баз данных**

ER-диаграммы (Entity-Relationship diagrams) являются графическими моделями, используемыми для описания сущностей и отношений между ними в базах данных. Эти диаграммы помогают программистам и аналитикам понять структуру и связи между данными в системе.

В ER-диаграммах используются три основных элемента: сущности, атрибуты и связи. Сущности представляют объекты, которые нужно хранить в базе данных, например, клиенты, заказы или продукты. Атрибуты определяют свойства каждой сущности, такие как имя, адрес или цена. Связи показывают отношения между сущностями, например, заказы могут быть связаны с клиентами, а продукты - с категориями.

ER-диаграммы могут быть использованы для проектирования баз данных, а также для анализа существующих баз данных. Они помогают увидеть структуру данных и понять, как они связаны друг с другом. Это может помочь в оптимизации запросов и улучшении производительности баз данных.

Программисты могут использовать ER-диаграммы для создания схем баз данных и для документирования баз данных. Для создания ER-диаграмм могут использоваться специальные инструменты, такие как ERD-диаграммы в Microsoft Visio или онлайн-инструменты для создания ER-диаграмм.

В целом, ER-диаграммы являются важным инструментом для проектирования и анализа баз данных. Они помогают программистам лучше понимать структуру данных и связи между ними, что может привести к более эффективной работе с базами данных и улучшению производительности системы в целом.

В приложении для хранения всей нужной информации была создана база данных. На рисунке 5 представлена ER-модель Базы данных. Так же ниже описана структура таблиц и назначение полей.

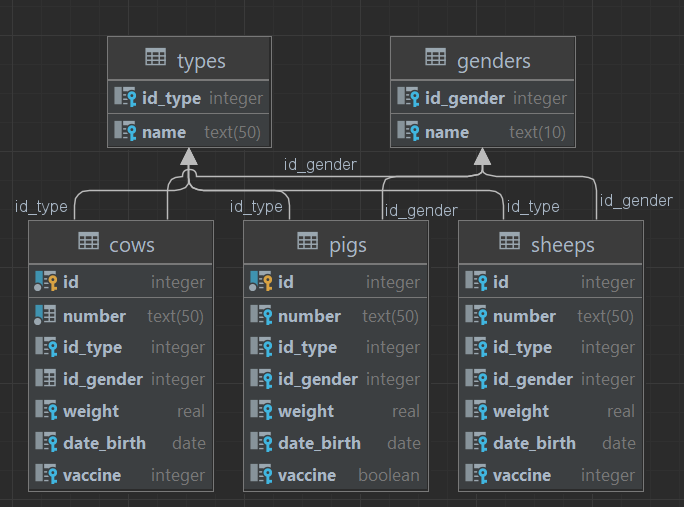


Рисунок 5. ER-модель

База данных состоит из 5 таблиц:

1. Types – таблица хранит виды животных;
   1. id\_type – первичный ключ;
   2. name – наименование вида животного.
2. Genres – таблица хранит наименование гендера животных;
   1. id\_gender – первичный ключ;
   2. name – наименование вида животного.
3. Cows – таблица для хранения данных о коровах;
   1. id – первичный ключ;
   2. number– уникальный номер животного;
   3. id\_type – внешний ключ таблицы types;
   4. id\_gender – внешний ключ таблицы genders;
   5. weight – вес животного;
   6. date\_birth – дата рождения животного;
   7. vaccine – наличие вакцины у животного.
4. Pigs – таблица для хранения данных о свиньях;
   1. id – первичный ключ;
   2. number– уникальный номер животного;
   3. id\_type – внешний ключ таблицы types;
   4. id\_gender – внешний ключ таблицы genders;
   5. weight – вес животного;
   6. date\_birth – дата рождения животного;
   7. vaccine – наличие вакцины у животного.
5. Sheeps - таблица для хранения данных об овцах;
   1. id – первичный ключ;
   2. number– уникальный номер животного;
   3. id\_type – внешний ключ таблицы types;
   4. id\_gender – внешний ключ таблицы genders;
   5. weight – вес животного;
   6. date\_birth – дата рождения животного;
   7. vaccine – наличие вакцины у животного.

## **Описание модулей проекта**

Помимо стартового модуля, в проект входят и другие модули с фрагментами и интерфейсные модули.

Также в стартовый модуль входит код, который запускает саму программу при ее запуске. Полный код всех модулей находится в «Приложение 1».

async\_init() – данный метод запустится при старте программы, как входная точка в асинхронность;

load\_data() – этот метод вызывается для загрузки данных из БД в определенную таблицу;

add\_new\_row() – данный метод вызывается для добавления новой строки в определенную таблицу;

delete\_row() – данный метод вызывается для удаление строки в определенной таблице;

save\_data() – данный метод берёт сохраняет данные редактируемой таблицы.

## **Тестирование проекта**

В этом разделе будут продемонстрированы результаты тестирования программы.

**Тест. Отладка**

В этом разделе показано умение применять средства отладки. Так как я работаю в PyCharm, отладчик уже встроен в программу.

Для проверки была искусственно создана ошибка в коде. Вот что показал отладчик (Рисунок 6, 7).

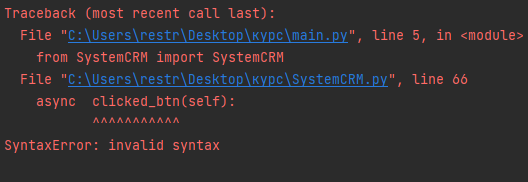


Рисунок 6. Ошибка в коде

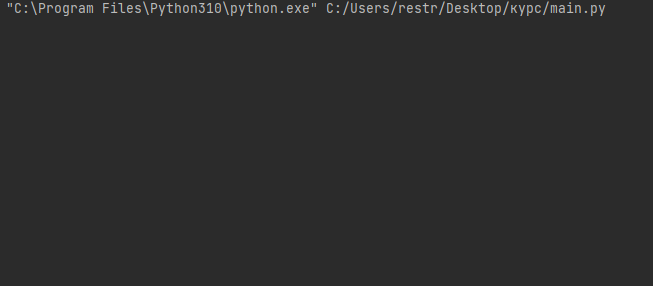


Рисунок 7. Исправление ошибки

**Тест. Срабатывание кнопки «Добавить»**

Действия: добавление новой строки в таблице, с помощью кнопки «Добавить».

Ожидаемый результат: отображение новой строки в таблице.

Результат теста:

Сперва была выбрана таблица, после этого была нажата кнопка «Добавить» (Рисунок 8).

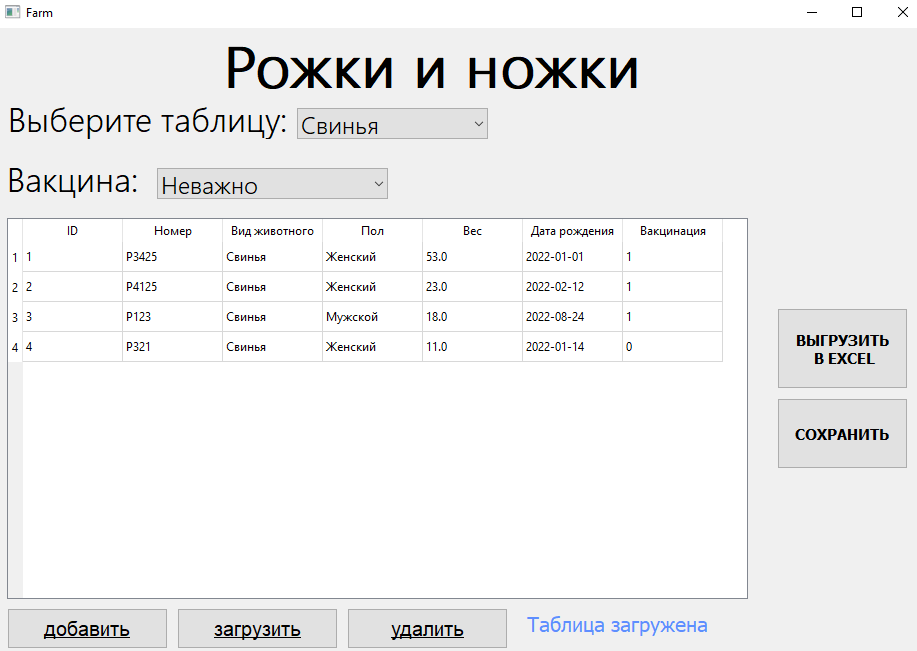


Рисунок 8. Таблица свиней

После этого была добавлена новая строка в таблицу (Рисунок 9).

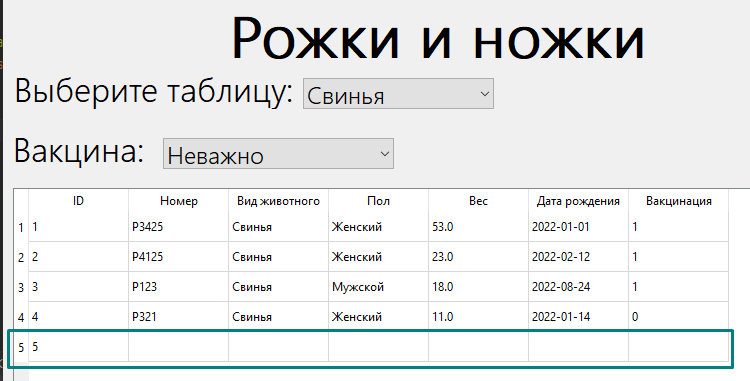


Рисунок 9. Добавленная строка

В консоли отладчика не было выявлено никаких ошибок.

**Тест. Загрузка данных в Excel файл**

Действия: загрузить таблицу коров и нажать на кнопку «Выгрузить в Excel»

Ожидаемый результат: файл формата .xlsx в папке «ExcelData», в котором находятся данные из таблицы программы.

Результат теста:

Сперва была выбрана таблица коров, после этого была нажата кнопка «Выгрузить в Excel» (Рисунок 10).

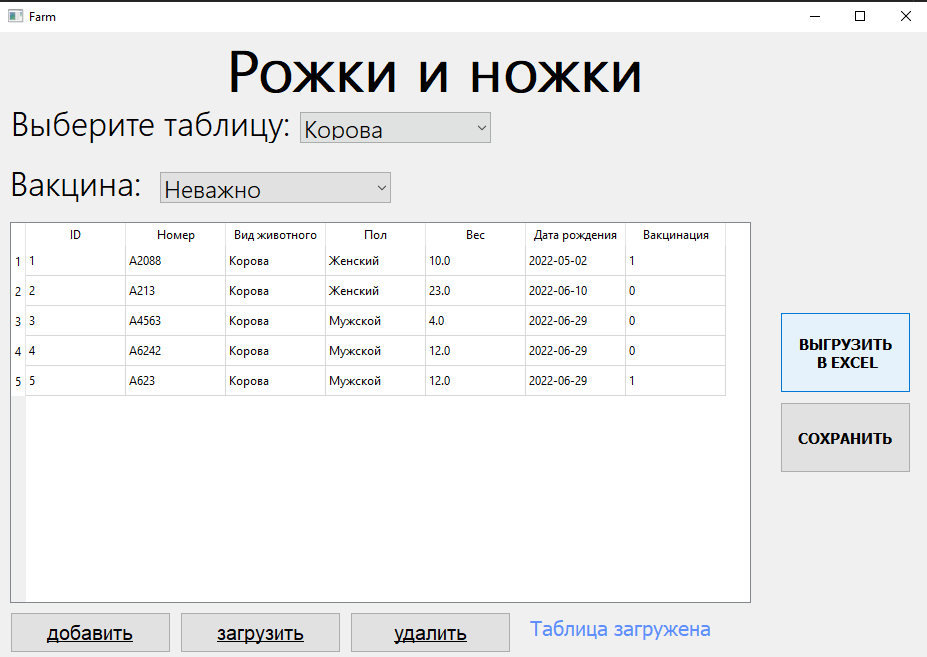


Рисунок 10. Таблица коров

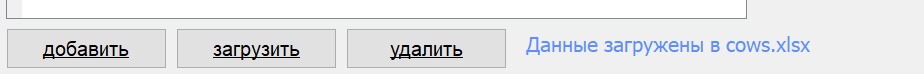
Строка статуса отобразила информацию об успешной загрузке данных в Excel файл «cows.xlsx» (Рисунок 11). 

Рисунок 11. Загрузка таблицы букета «Салют»

Далее был открыт файл cows.xlsx. Данные из таблицы были загружены в полном соответствии (Рисунок 12).

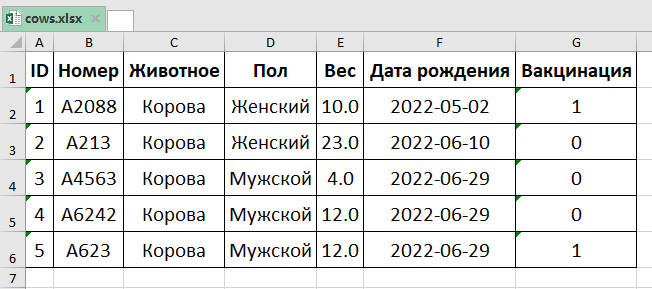


Рисунок 12. Загрузка таблицы букета «Салют»

В консоли отладчика не было выявлено никаких ошибок.

## **Руководство оператора**

В данном программном документе приведено руководство оператора по применению и эксплуатации программы «Farm».

В данном программном документе, в разделе «Назначение программы» указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе «Условия выполнения программы» указаны условия, необходимые для выполнения программы (минимальный состав аппаратных и программных средств и т.п.).

В данном программном документе, в разделе «Выполнение программы» указана последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузку и управляет выполнением программы, а также ответы программы на эти команды.

Оформление программного документа «Руководство оператора» произведено по требованиям ЕСПД (ГОСТ 19.101-77 [[1]](#footnote-1)1), ГОСТ 19.103-77 [[2]](#footnote-2)2), ГОСТ 19.104-78\* [[3]](#footnote-3)3), ГОСТ 19.105-78\* [[4]](#footnote-4)4), ГОСТ 19.106-78\* [[5]](#footnote-5)5), ГОСТ 19.505-79\* [[6]](#footnote-6)6), ГОСТ 19.604-78\* [[7]](#footnote-7)7)).

Далее на страницах 18-25 приведено содержание разработанного документа «Руководство оператора» на спроектированное приложение.

1. **Назначение программы**
   1. **Функциональное назначение программы**

Специальное программное обеспечение «Farm» используется для изучения специализированного технического IT английского.

* 1. **Эксплуатационное назначение программы**

Специальное программное обеспечение «Farm» может эксплуатироваться в любых сельскохозяйственных предприятиях, обладающих любым поголовьем скота.

* 1. **Состав функций**
     1. **Функция регистрации.**

Эта функция позволяет зарегистрироваться новому пользователю.

* + 1. **Функция авторизации.**

Эта функция позволяет пользователю авторизоваться.

* + 1. **Функция просмотра статей.**

Эта функция позволяет просмотреть определённые статьи в приложении.

* + 1. **Функция тестирования.**

Эта функция позволяет проходить тестирование по словам из определённых статей в приложении.

* + 1. **Функция правописания слов.**

Эта функция позволяет упражняться путем правописания определённых слов из статей в приложении.

1. **Условия выполнения программы**
   1. **Минимальный состав аппаратных средств**

* ОС: Windows 10 и выше;
* Процессор: 32-битные и 64-битные: ARM, MIPS, x86. Тактовая частота не ниже 1,8 ГГц;
* ОЗУ: 2 ГБ;
* Место на жестком диске: 10 МБ;
* Видеоадаптер: DirectX версии не ниже 9 с драйвером WDDM 1.0.;
* Дисплей: 720 x 1600.
  1. **Минимальный состав программных средств**

Дополнительные программные средства не требуются.

* 1. **Требование к пользователю**

Конечный пользователь программы должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

1. **Выполнение программы**
   1. **Загрузка и запуск программы**

Для запуска программы требуется запустить ярлык исполняемого файла.

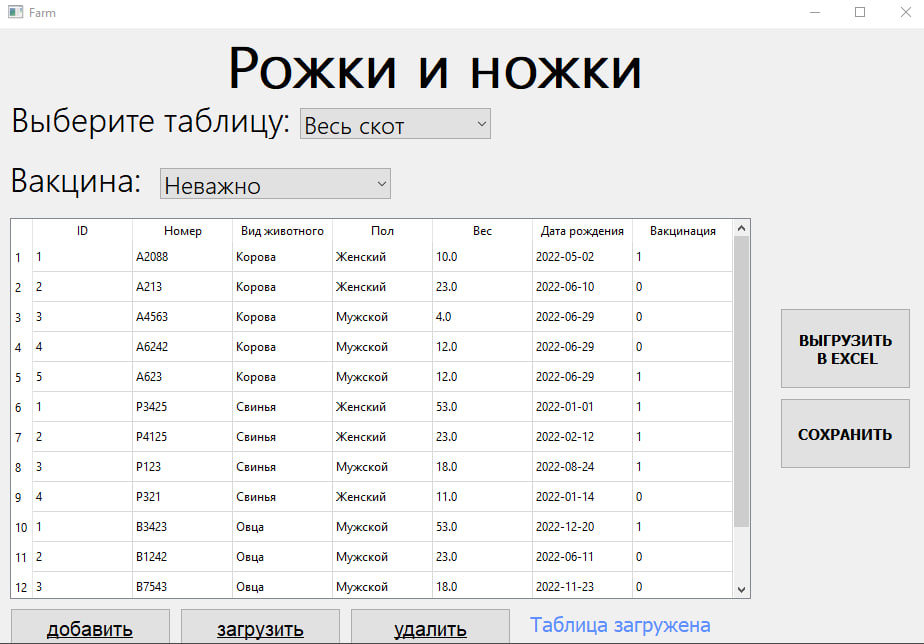


Рисунок 13. Основной интерфейс приложения

На рисунке 13 отображен основной интерфейс приложения.

Представлены следующие элементы:

1. Фильтр вида скота;
2. Фильтр вакцинации;
3. Кнопка добавления строк;
4. Кнопка загрузки редактированных данных в таблицу.
5. Кнопка удаления выбранных строк.
6. Кнопка выгрузки данных в .xls таблицу.
7. Кнопка сохранения изменений.

Редактирование данных происходит через двойной клик по необходимому полю (Рисунок 14).

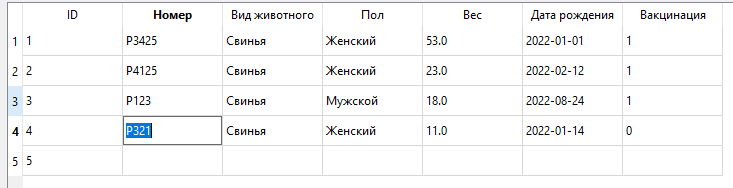


Рисунок 14. Редактирование данных

Настройка фильтра животных совершается в выпадающем окне, вызываемом через одиночный клик по указанному полю (Рисунок 15).

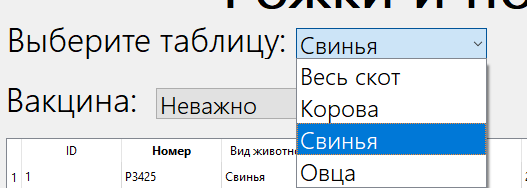


Рисунок 15. Фильтр вида животных

Фильтр вакцинации настраивается в выпадающем окне, вызываемом через одиночный клик по указанному полю (Рисунок 16).

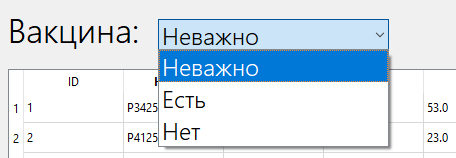


Рисунок 16. Фильтр вакцинации

Для добавления строки необходимо выбрать необходимый вид скота и нажать на кнопку «добавить» (Рисунок 17).

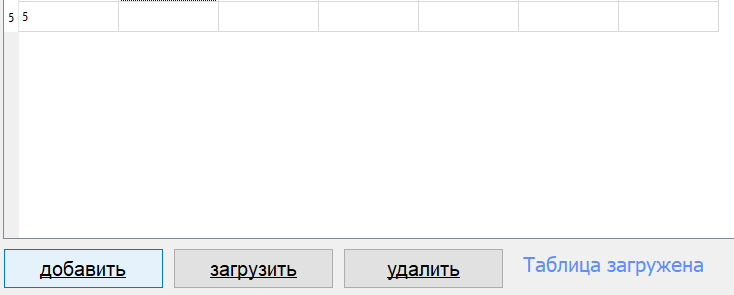


Рисунок 17. Добавление строки

Для загрузки новой строки в базу данных необходимо нажать кнопку «загрузить» (Рисунок 18).

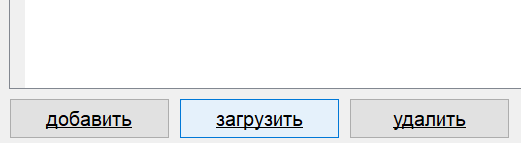


Рисунок 18. Загрузка новой строки в таблицу

Для удаления строки необходимо выбрать строку и нажать кнопку «удалить» (Рисунок 19).

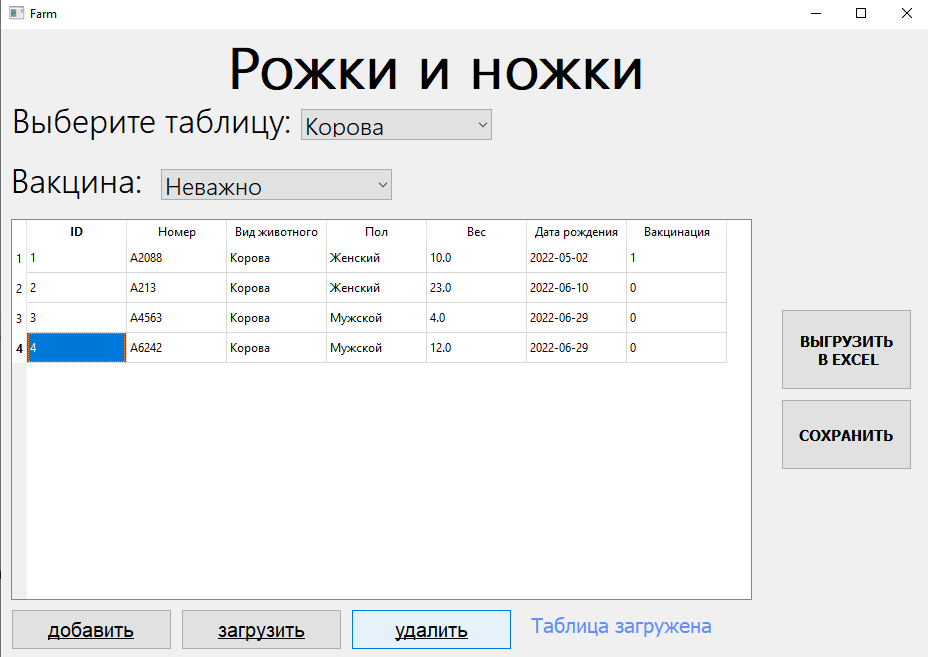


Рисунок 19. Удаление строки

Для выгрузки данных в .xls таблицу необходимо нажать кнопку «Выгрузить в Excel» (Рисунок 20).

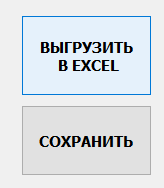


Рисунок 20. Выгрузка данных

Для сохранения измененных данных необходимо нажать кнопку «Сохранить» (Рисунок 21).

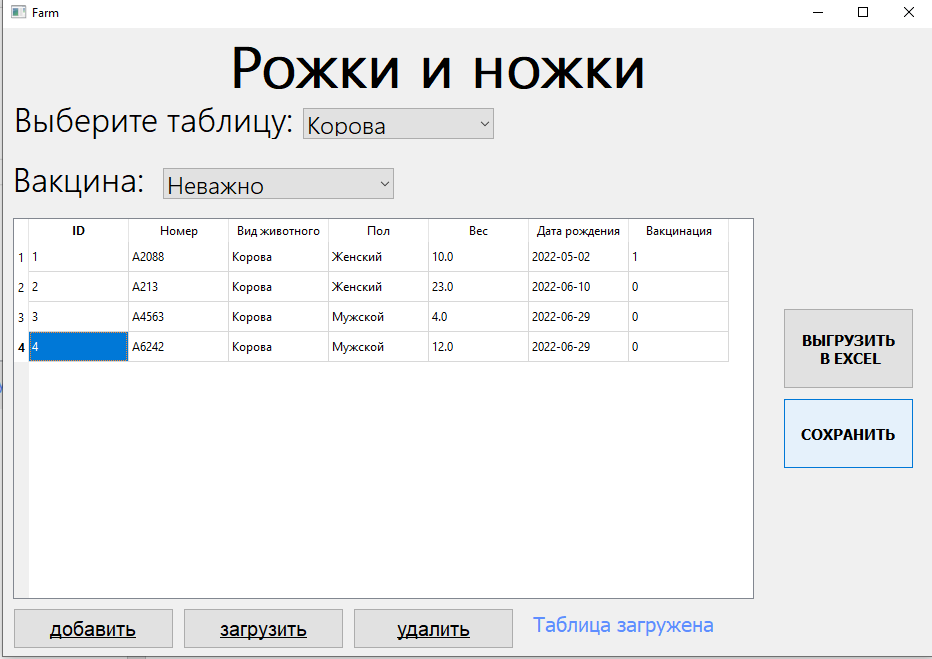


Рисунок 21. Сохранение данных

Статус операции и возможные ошибки отображаются в левом нижнем углу интерфейса (Рисунок 22).

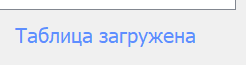


Рисунок 22. Статус операции

Выгруженные .xls таблицы сохраняются в директории программы в папке «ExcelData» (Рисунок 23).

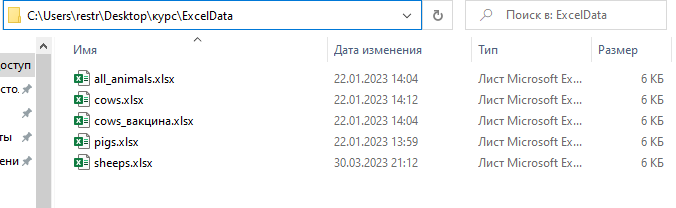


Рисунок 23. Директория выгруженных таблиц

* 1. **Завершение работы программы**

Чтобы завершить работу программы нужно принудительно завершить процесс через стандартные элементы интерфейса.

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсового проекта была написана программа «Farm» для упрощения и удобства контроля и редактирования данных сельскохозяйственного предприятия.

В ходе работы были проанализированы: предметная область, существующие разработки, получены навыки по созданию десктопных приложений и работе с БД.

Также планируется продолжение работы над данным проектом с целью расширения возможностей и удобства приложения для пользователей.

# **Список используемых источников**

1. “Beginning PyQt: A Hands-on Approach to GUI Programming with PyQt6” Автор: Joshua Willman. Издательство: Apress. Дата публикации: 2-е издание (17 марта 2022 года);
2. “Python and Qt: The Best Parts” Автор: Michael Herrmann. Обновлено: март 2022 года;
3. “Create GUI Applications with Python & Qt6” Автор: Martin Fitzpatrick. Дата публикации: Март 2021 года;
4. “Beginning PyQt: A Hands-on Approach to GUI Programming” Автор: Joshua Willman. Издательство: Apress. Дата публикации: 1-е издание (29 мая 2020 года);
5. “Modern PyQt: Create GUI Applications for Project Management, Computer Vision, and Data Analysis” Автор: Joshua Willman. Издательство: Apress. Дата публикации: 1-е издание (9 декабря 2020 года);
6. Интернет-ресурс «The Python Wiki» [Электронный ресурс] URL: <https://wiki.python.org/moin/PyQt> (дата обращения: 25.01.2023).
7. Интернет-ресурс «WebforMyself» [Электронный ресурс] URL: <https://webformyself.com/rukovodstvo-po-pyqt-python-gui-designer/> (дата обращения: 17.02.2023);
8. Интернет-ресурс «GeeksforGeeks» [Электронный ресурс] URL: <https://www.geeksforgeeks.org/> (дата обращения: 11.02.2023);

**Приложение 1.**

Находится в файле «Приложение 1.docx».

1. 1) ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов [↑](#footnote-ref-1)
2. 2) ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов [↑](#footnote-ref-2)
3. 3) ГОСТ 19.104-78\* ЕСПД. Основные надписи [↑](#footnote-ref-3)
4. 4) ГОСТ 19.105-78\* ЕСПД. Общие требования к программным документам [↑](#footnote-ref-4)
5. 5) ГОСТ 19.106-78\* ЕСПД. Общие требования к программным документам, выполненным печатным способом [↑](#footnote-ref-5)
6. 6) ГОСТ 19.505-79\* ЕСПД. Руководство оператора. Требования к содержанию и оформлению [↑](#footnote-ref-6)
7. 7) ГОСТ 19.604-78\* ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом [↑](#footnote-ref-7)