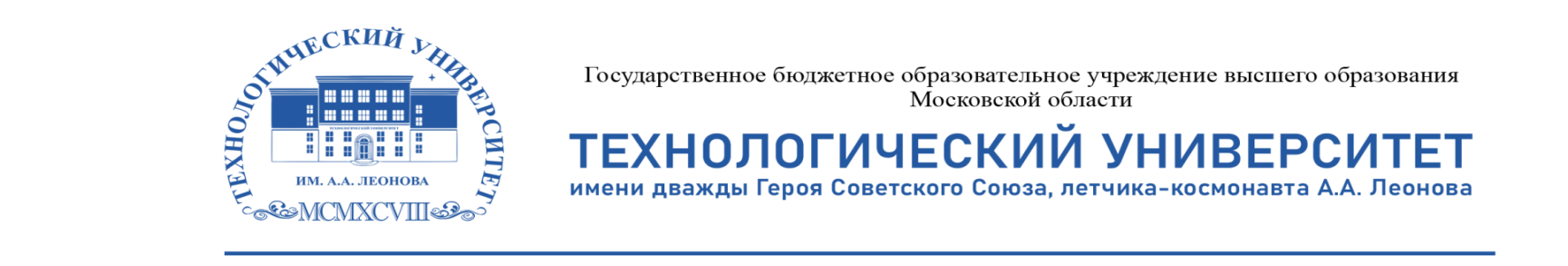
****

Колледж космического машиностроения и технологии

**Курсовой проект**

**Разработка системы классов для приложения «УПРАВЛЯЮЩАЯ КОМПАНИЯ ЖКХ»**

Пояснительная записка

КП.09.02.03.22.019ПЗ

Обучающийся группы П2-19 Серегин А. О.

Руководитель курсового проекта Гусятинер Л. Б.

Результат защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Королев, 2022 г.

**ОГЛАВЛЕНИЕ**

[**1. Введение 4**](#_Toc107782249)

[**2. Теоретическая часть 5**](#_Toc107782250)

[**2.1. Описание предметной области 5**](#_Toc107782251)

[**2.2. Описание существующих разработок 7**](#_Toc107782252)

[**3. Теоретическая часть 14**](#_Toc107782253)

[**3.1. Построение диаграммы прецедентов 14**](#_Toc107782254)

[**3.2. Выбор инструментов 15**](#_Toc107782255)

[**3.2.1. Язык программирования 15**](#_Toc107782256)

[**3.2.2. Среда разработки 15**](#_Toc107782257)

[**3.2.3. База данных 16**](#_Toc107782258)

[**3.2.4. Графический интерфейс 16**](#_Toc107782259)

[**3.3. Проектирование сценария 17**](#_Toc107782260)

[**3.4. Построение диаграммы классов 18**](#_Toc107782261)

[**3.5. Описание главного модуля 22**](#_Toc107782262)

[**3.6. Описание спецификаций к модулям 23**](#_Toc107782263)

[**3.7. Описание модулей 24**](#_Toc107782264)

[**3.8. Описание текстовых наборов модулей 25**](#_Toc107782265)

[**3.9. Описание применения средств откладки 26**](#_Toc107782266)

[**4. Эксплуатационная часть 27**](#_Toc107782267)

[**4.1. Руководство оператора 27**](#_Toc107782268)

[**4.1.1. Назначение программы 27**](#_Toc107782269)

[**4.1.2. Условия запуска программы 27**](#_Toc107782270)

[**4.1.3. Выполнение программы 27**](#_Toc107782271)

[**4.1.4. Сообщение оператору 28**](#_Toc107782272)

[**4.2. To-Do лист 29**](#_Toc107782273)

[**5. Заключение 30**](#_Toc107782274)

[**6. Список использованной литературы и интернет-ресурсов 31**](#_Toc107782275)

[**7. Приложения 32**](#_Toc107782276)

[**7.1. Приложение 1 32**](#_Toc107782277)

[**7.2. Приложение 2 32**](#_Toc107782278)

[**7.3. Приложение 3 32**](#_Toc107782279)

[**7.4. Приложение 4 33**](#_Toc107782280)

[**7.5. Приложение 5 33**](#_Toc107782281)

[**7.6. Приложение 6 34**](#_Toc107782282)

[**7.7. Приложение 7 34**](#_Toc107782283)

[**7.8. Приложение 8 35**](#_Toc107782284)

[**7.9. Приложение 9 35**](#_Toc107782285)

[**7.10. Приложение 10 36**](#_Toc107782286)

[**7.11. Приложение 11 36**](#_Toc107782287)

# **Введение**

Данный курсовой проект заключается в разработке системы классов для приложения «Управляющая компания «ЖКХ»». Разработка приложения будет происходить на языке программирования Python и фреймворка PyQt.

В первой части будут рассмотрены предметная область и существующие продукты по данной теме.

Во второй части будут описаны инструменты разработки программы и модули программы.

В третьей части описаны взаимодействие пользователя с программой и сообщения оператору.

В заключительной части будут приведены общие выводы по проекту.

# **Теоретическая часть**

## **Описание предметной области**

**Настольное приложение** – это программа, которая работает на компьютере пользователя. Она функционирует на стороне клиента и запускается в виде обыкновенного исполняемого файла.

Настольное приложение реализует следующие возможности:

1. Работа приложения как с подключением, так и без подключения к интернету;
2. Приложение запускается с загрузкой данных из памяти устройства, на котором установлено, не требует каждый раз загрузки параметров из сети;
3. Максимальная эффективность работы за счет возможности многопроцессорных систем;

**Управляющая компания (УК) –** это юридическое лицо, чья сфера деятельности заключается в обслуживании многоквартирных домов за плату от собственников.

УК отвечает за ремонт и обслуживание общего имущества собственников жилья, за уборку в подъезде и во дворе, вывоз мусора, оказывает жильцам услуги по ремонту коммуникаций в квартирах.

На основании договора с жильцами управляющая компания берет на себя ответственность за поддержание МКД и придомовой территории в должном санитарном и техническом состоянии, а также оказывает коммунальные услуги и управляет самим домом.

УК различаются по юридическим критерию: жилищные кооператив, акционерное общество и общество с ограниченной ответственностью.

Существует три типа управляющих компаний:

* Управляющие организации – это те компании, которые занимаются исключительно деятельностью по управлению многоквартирного дома. Все остальные вопросы насчет обеспечения надлежащих санитарных и технических условий решают подрядчики;
* Гибридные компании – организации, которые берут на себя ответственность как за управление жилым домом, так и за поддержание его в надлежащем состоянии, своевременный ремонт общественного имущества и обеспечение коммунальных услуг. Такой тип наиболее распространен на территории России;
* Эксплуатационными называются подрядные организации, нанятые управляющими компаниями для выполнения работ. Это менее распространенный вид организации.

Обязанности УК:

* Обслуживание;
* Содержание общедомового имущества;
* Контакты с поставщиками услуг.

## **Описание существующих разработок**

На сегодняшний день существует целый ряд систем для управляющих компаний «ЖКХ», но большинство из них существуют в виде сайтов.

1. Система «Инфокрафт ЖКХ 365» [1] - онлайн-сервис для управляющих компаний. С его помощью можно рассчитывать квартплату, заниматься бухгалтерскими учетом, создавать сайт и личные кабинеты жильцов и т. д.;

Сервис позволяет работать с программами, каждая из которых позволяет решать различные процессы. Например, «Формула ЖКХ» предназначена для расчета квартплаты, отдельный модуль бухгалтерии («Формула ЖКХ + Бухгалтерия») позволяет вести бухгалтерский учет на платформе 1С: Предприятие.

Данные пользователей хранятся в надежном ЦОДе с автоматической системой резервирования. При желании вы можете сохранить отдельную резервную копию на свой компьютер. Также вы в любой момент можете переходить из облачной версии к программам, установленным на компьютере, и обратно.

Возможности:

* Расчет квартплаты и печать квитанций (рисунок 1);
* «1С: Бухгалтерия 8» для бухгалтерского и налогового учета (УСН, ОСНО);
* Выгрузка данных в ГИС ЖКХ напрямую из программы (рисунок 2);
* Готовые сайт для вашей организации;
* Сдача отчетности через Интернет;
* Доступность из любой точки мира.

Обмен данными с ГИС ЖКХ – облачный сервис «Инфокрафт ЖКХ 365» интегрируется с ГИС ЖКХ через Excel-файлы и путем прямого обмена через защищенное соединение.

При обмене будут доступны следующие возможности:

* Прием и размещение показаний приборов учета;
* Выставления платежных документов за ЖКУ в электронном виде;
* Размещение договоров управления (уставов) и сведений о многоквартирных домах;
* Раскрытие информации о деятельности организации.

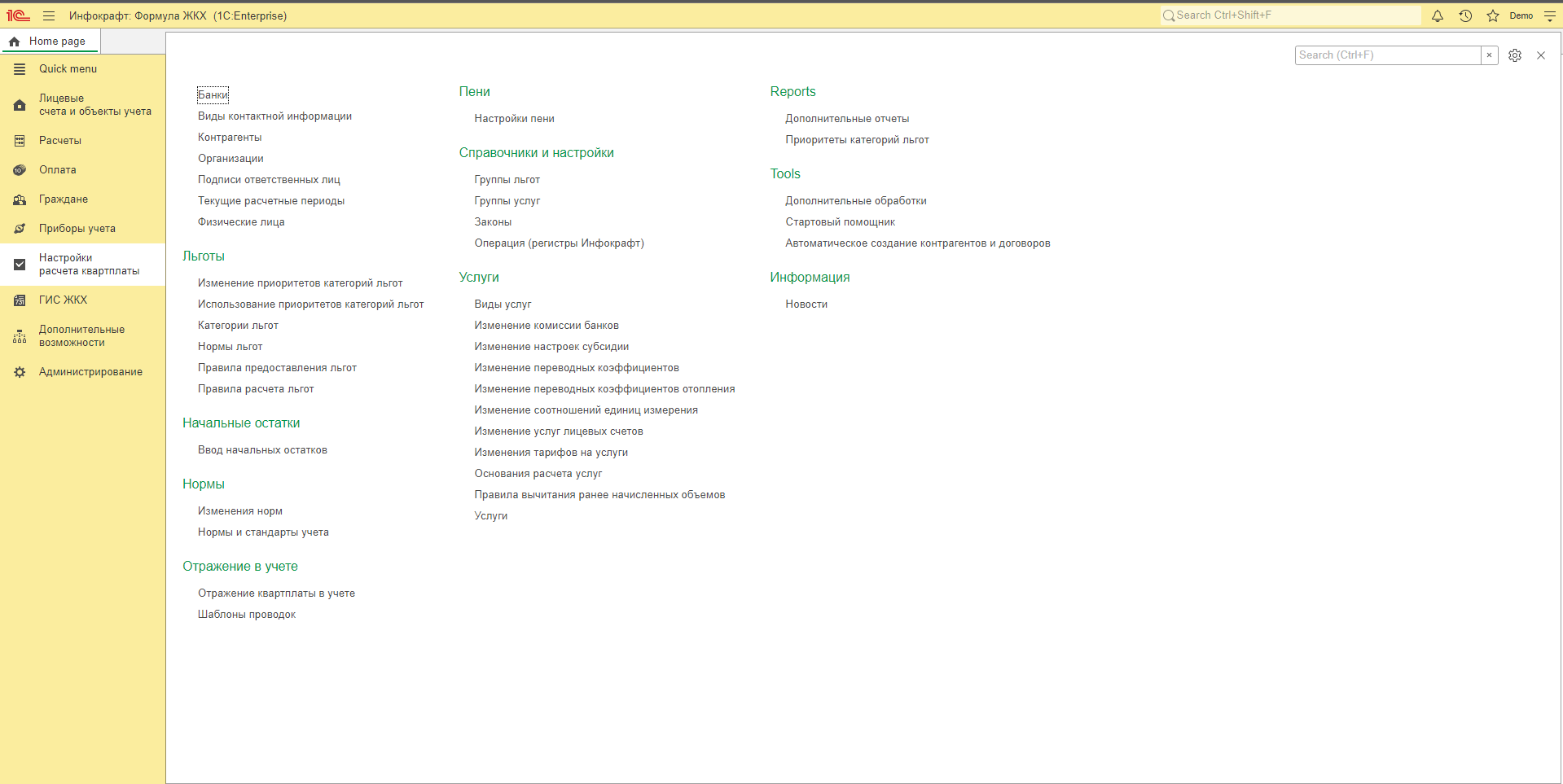


Рис. 1. Страница «Настройки расчета квартплаты»

Изображение выглядит как текст

Автоматически созданное описание

Рис. 2. Страница «Выгрузки данных в ГИС ЖКХ»

1. Система «Контур.ЖКХ» [2] - программа для ТСЖ и УК. Расчет квартплаты граждан, ведение паспортного учета, оформление квитанций. Полная интеграция с ГИС ЖКХ.

Расчетная система «Контур.ЖКХ» представляет собой специализированную билинговую программу и представляет предприятию следующие возможности:

* Хранить и обрабатывать информацию по лицевым счетам абонентов;
* Производить начисление за услуги, представленные на лицевом счете;
* Выполнять различные перерасчеты в автоматическом и ручном режимах;
* Использовать в работе коллективные и индивидуальные приборы учета (рисунок 4);
* Формировать квитанции на оплату в различных форматах, в том числе с использованием штрихкода;
* Начислять пени за просрочку оплаты услуг;
* Начислять льготы и субсидии на оплату услуг ЖКХ;
* Формировать статистическую и финансовую отчетность;
* Организовать паспортный учет граждан с автоматическим отражением данных о движении граждан в расчетной системе (рисунок 3);
* Организовать личный кабинет граждан в Интернет.

Изображение выглядит как стол

Автоматически созданное описание

Рис. 3. Отражение данных о населении

Изображение выглядит как стол

Автоматически созданное описание

Рис. 4. Отчет по показателям проборов учета

1. Система «Домиленд» [3] - сервис для управляющих организаций и единых расчетных центров, в котором можно вести базу собственников, дольщиков, арендаторов и прочих доверенных лиц, принимать показания водоснабжения, электроэнергии, отопления, газа и приборов учета, а также отслеживать должников по счетам.

Доступно приложение для сотрудников компании, в котором они могут просматривать историю активности жильца, отправлять показания индивидуальных приборов учета, работать с обращениями и отслеживать нарушения на объекте. Жильцы тоже могут пользоваться приложением, чтобы быстро решать вопросы по дому: размещать объявления, заказывать пропуск для гостей, вызывать сантехника, решать вопросы ремонта и разбираться со счетами и счетчиками.

Возможности системы для Управляющей организации:

* Онлайн прием обращений от жителей, работы с электронными заявками, маршрутизация на исполнителей, оценка по каждой заявке;
* Интеграция с биллингом компании или ЕРЦ;
* Проведение опросов и общих собраний собственников в очной, очно-заочной форме (Ст.47 ЖК РФ) и заочной форме с использованием системы непосредственно в мобильном приложении (Ст. 47.1 ЖК РФ);
* Ведение реестра собственников (Ст. 45 ЖК РФ);
* Коммуникация через sms/push из системы, в т.ч. по сегментированной базе клиентов;
* Организация приема платежей с пониженной комиссией 1% вне зависимости от оборота платежей организации;
* Мобильное приложение исполнителя (Март 2019);
* Брендирование мобильного приложения без расходов на разработку;
* Соблюдение ФЗ-№152 "О персональных данных" и ФЗ-№54 "О применении контроль-кассовой техники", Постановления Правительства №331 от 27 марта 2018 года;
* И другие возможности системы Домиленд.

Возможность мобильного приложения жителя:

* Просмотр начислений, суммы долга, тарифов, реквизитов и возможность оплаты;
* Подача показаний ИПУ;
* Отправка заявок в управляющую организацию;
* Получение Push уведомлений и просмотр новостей;
* Участие в опросах и общих собраниях;
* Просмотр нескольких помещений собственника;
* Добавление в сервис других пользователей: арендатора, члена семьи и т. д.

# **Теоретическая часть**

## **Построение диаграммы прецедентов**

Данный раздел содержит диаграмму прецедентов (рисунок 7), на которой отображается визуализация различных типов ролей в системе и то, как эти роли взаимодействуют с системой.

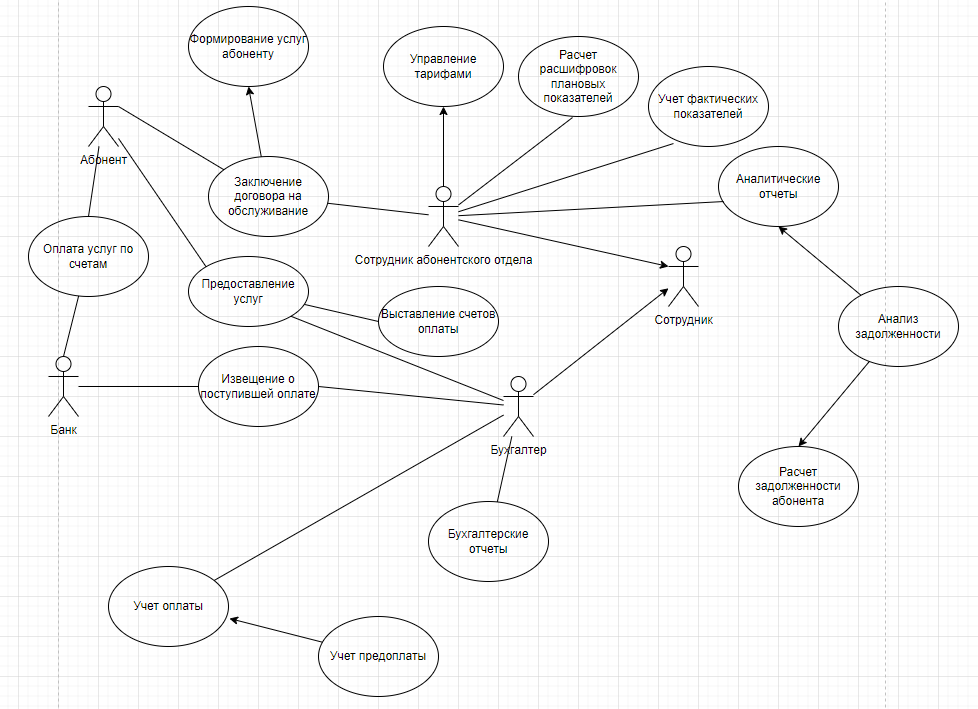


Рис. 5. Диаграмма прецедентов

## **Выбор инструментов**

### **Язык программирования**

Для написания проекта использовался Python [4] – высокоуровневый язык программирования, к преимуществам которого относят высокую производительность программных решений и структурированный, хорошо читаемый код. Синтаксис максимально облегчен, ядро имеет удобную структуру, а широкий перечень встроенных библиотек позволяет применять внушительный набор полезных функций и возможностей.

Python поддерживает несколько моделей программирования: структурное, объекта-ориентированное, функциональное. В языке присутствует динамическая типизация, автоматическое управление памятью, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных.

По данным индекса TIOBE [8] на 2022 год, Python является самым популярным языком программирования, оставив позади себя Java и семейство языков С (рисунок 6).

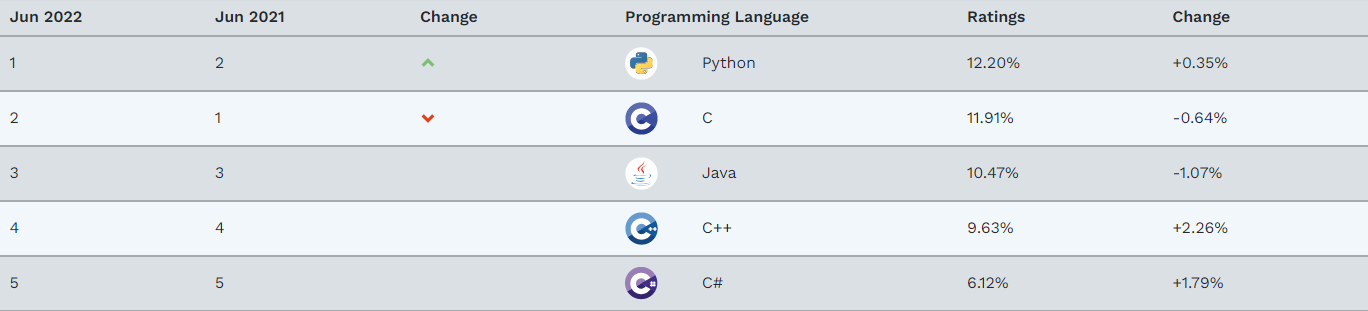


Рисунок 6. Таблица «Популярные языки программирования»

### **Среда разработки**

PyCharm [5] – интегрированная кроссплатформенная среда разработки для языка программирования Python предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. Разработка IDE ведется компанией JetBrains.

PyCharm предоставляется в двух вариантах: платный (PyCharm Professional Edition) и бесплатный (PyCharm Community Edition).

### **База данных**

Уже довольно много лет большинство приложений и сайтов хранят информацию или данные. Для этого они используют базу данных – это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. Обычно они управляются системой управления базами данных (СУБД).

В качестве базы данных в проекте была выбрана SQLite [6] – это быстрая и легкая встраиваемая одно файловая СУБД, написанная на языке С, которая не имеет сервера и позволяет хранить всю базу локально на одном компьютере. Для работы SQLite не нужны сторонние библиотеки или службы.

### **Графический интерфейс**

В качестве реализации графического интерфейса в проекте использовался PyQt [7] – это библиотека Python для создания приложения с графическим интерфейсом с помощью инструментария Qt. Библиотека предоставляет из себя набор модулей, содержащих большое количество классов и функций. Разработка ведётся компанией Riverbank Computing.

## **Проектирование сценария**

Данный раздел содержит сценарий проекта (рисунок 7) – определение сочетаний условий, которые приводят к определенным результатам.

При запуске приложения открывается главное окно, в котором предоставляется возможность выбора таблицы, добавить элемент, изменить элемент, удалить элемент, вывести в Excel или JSON.

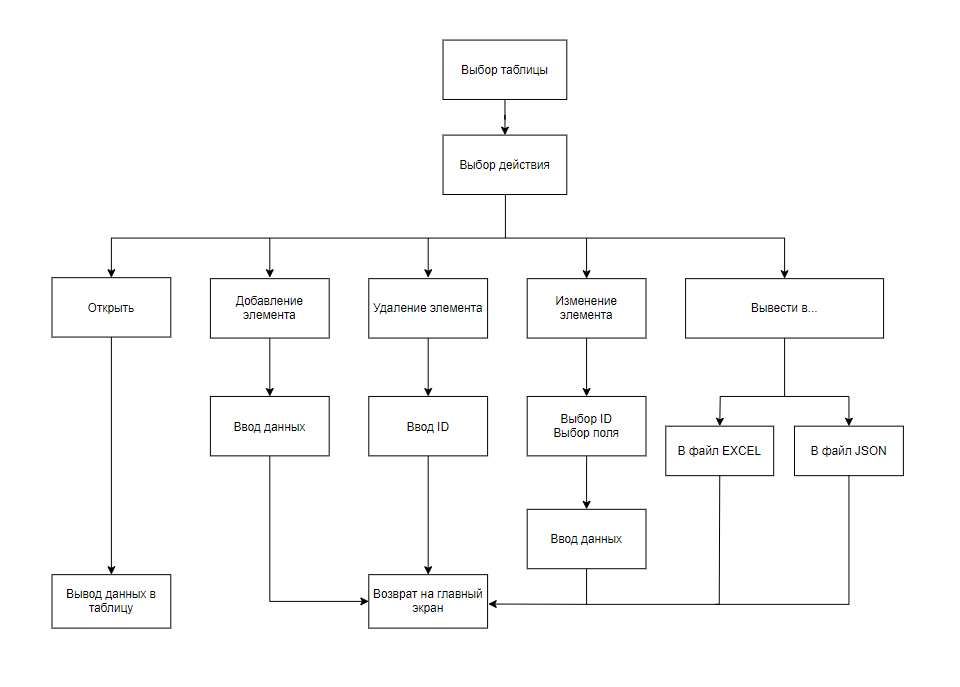


Рис. 7. Сценарий проекта

## **Построение диаграммы классов**

Данный раздел содержит диаграммы классов, их виды, а также связь между ними.

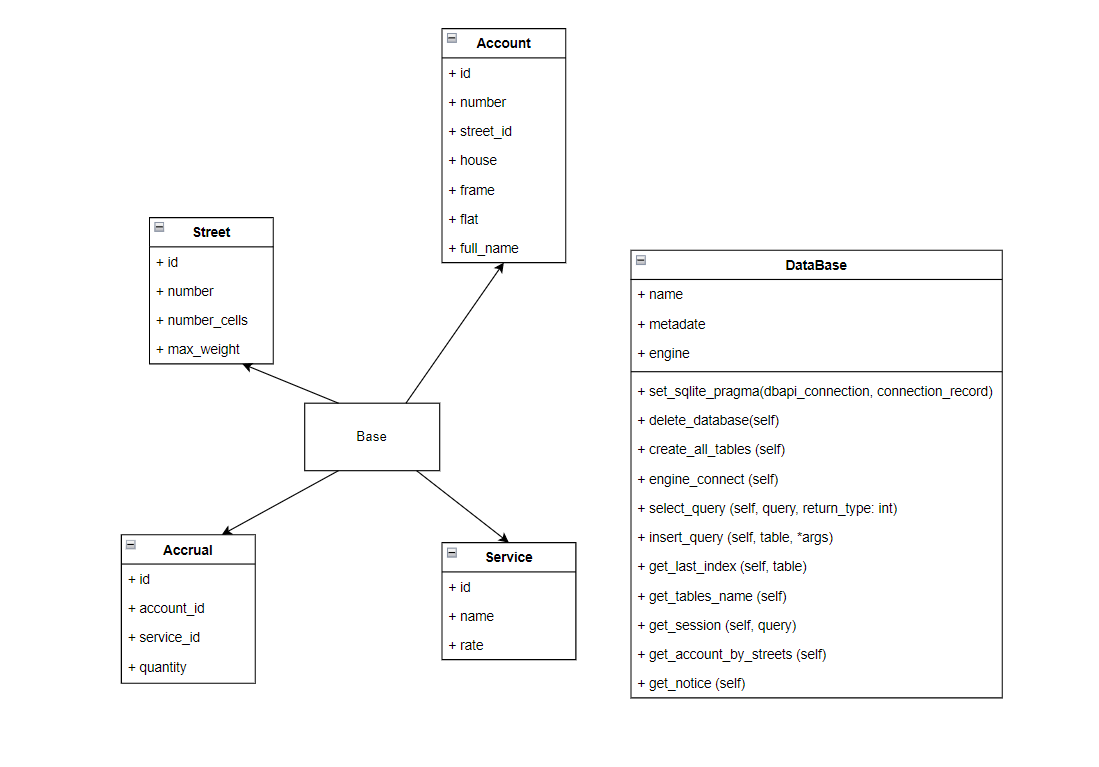
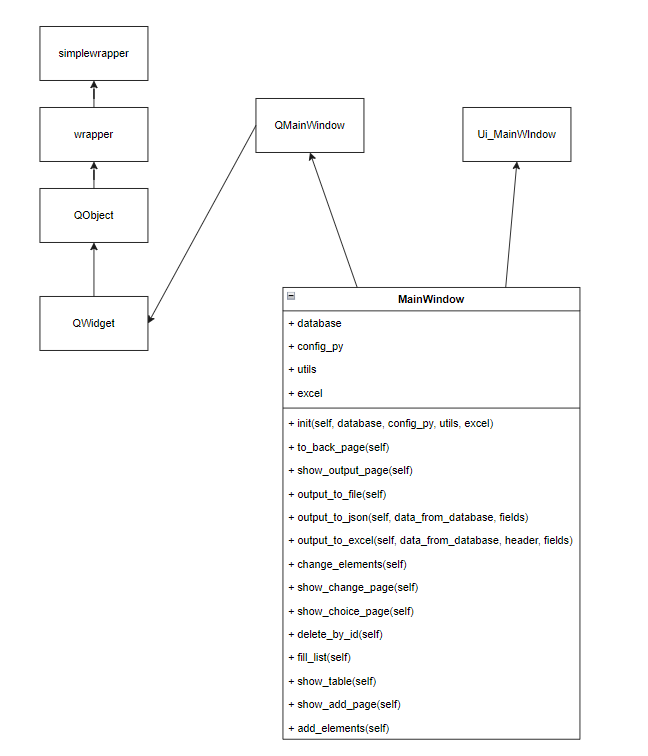


Рис. 8. Диаграмма классов раздел «База данных»

* Тип «База данных» (рисунок 8) полностью отвечает за создание и получение данных из базы, который включает в себя два модуля:
* Модуль описание моделей базы «models» содержит описание всех моделей таблиц. Под описанием понимается: наименование таблиц, полей и настройки полей;
* Модуль связь с базой данных «database» содержит реализацию работы с базой данных: подключение к базе, создание таблиц, запросы.

 Рис. 9. Диаграмма классов раздел «Графический интерфейс»

* Тип «Графический интерфейс» (рисунок 9) полностью отвечает за визуализацию приложения и включает в себя один модуль:
* Модуль работы главного окна «main\_connection». Он инициализирует окна и страницы приложения. Реализует логику взаимодействия пользователя с интерфейсом.

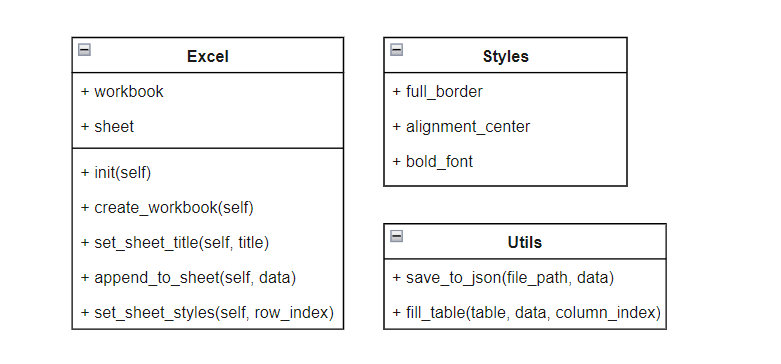


Рис.10. Диаграмма классов раздел «Утилиты»

* Тип «Вспомогательные» (рисунок 10) полностью отвечает за общие вспомогательные методы:
* Модуль «excel» отвечает за работу с одноименной программой;
* Модуль «styles\_excel» отвечает за стили для формирования текста в «MS Excel»;
* Модуль «utils» отвечает за методы, в равной степени используемые во всей программе в целом: вывод в файл c расширением json и т. д.

Изображение выглядит как текст

Автоматически созданное описание

Рис.11. Диаграмма классов раздел «Конфигурационные файлы»

* Тип «Константные файлы» (рисунок 13) включает в себя один модуль:
* Модуль «config» хранит списки и словари, необходимые в программе.

## **Описание главного модуля**

Главный модель (листинг 1) состоит из функции main. Данная функция включает в себя объекты необходимые для главного окна и запускает графический интерфейс.

Листинг 1. Главный модуль

import sys  
  
from PyQt5 import QtWidgets  
  
from config.config import Config  
from database.database import DataBase  
from gui.connection.main\_connection import MainWindow  
from utils.excel import Excel  
from utils.utils import Utils  
  
  
def main():  
 config = Config()  
 utils = Utils()  
 excel = Excel()  
  
 database = DataBase("composition")  
 database.create\_all\_tables()  
  
 app = QtWidgets.QApplication(sys.argv)  
 window = MainWindow(database, config, utils, excel)  
 window.show()  
 sys.exit(app.exec\_())  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

## **Описание спецификаций к модулям**

В проекте разработано множество модулей программы, оформленных в виде отдельных файлов.

Реализованные в проекте модули можно разделить на типы:

* База данных (рисунок 8):
* Описание моделей данных (models);
* Связь с базой данных (database);
* Графический интерфейс (рисунок 9):
* Работа главного окна (main\_connection);
* Вспомогательные (рисунок 10):
* Работа с выводом в Excel (excel);
* Работа со стилем в Excel (styles\_excel);
* Работа с сохранением в JSON (utils).
* Константные файлы (рисунок 11):
* Хранение списков и словарей (config);

Взаимодействие между модулями можно увидеть по диаграмме классов п.3.4 «Построение диаграммы классов».

## **Описание модулей**

* Тип «База данных» (рисунок 8) полностью отвечает за создание и получение данных из базы, который включает в себя два модуля:
* Модуль описание моделей базы «models» содержит описание всех моделей таблиц. Под описанием понимается: наименование таблиц, полей и настройки полей;
* Модуль связь с базой данных «database» содержит реализацию работы с базой данных: подключение к базе, создание таблиц, запросы.
* Тип «Графический интерфейс» (рисунок 9) полностью отвечает за визуализацию приложения и включает в себя один модуль:
* Модуль работы главного окна «main\_connection». Он инициализирует окна и страницы приложения. Реализует логику взаимодействия пользователя с интерфейсом.
* Тип «Вспомогательные» (рисунок 10) полностью отвечает за общие вспомогательные методы:
* Модуль «excel» отвечает за работу с одноименной программой;
* Модуль «styles\_excel» отвечает за стили для формирования текста в «MS Excel»;
* Модуль «utils» отвечает за методы, в равной степени используемые во всей программе в целом: вывод в файл c расширением json и т. д.
* Тип «Константные файлы» (рисунок 11) включает в себя один модуль:
* Модуль «config» хранит списки и словари, необходимые в программе.

## **Описание текстовых наборов модулей**

В проекте присутствует обработка ошибок, чтобы сообщить пользователю о проблеме и огородить его от сбоев программы и случайному завершению работы.

* Проверка на открытый файл (листинг 2).

В случае перезаписывания в открытый файл excel, отобразится уведомление о соответствующей ошибке в отдельном окне (приложение 11).

Листинг 2. Проверка на открытый файл.

try:  
 self.excel.workbook.save(filename=file\_path)  
 self.stackedWidget.setCurrentIndex(0)  
except PermissionError:  
 QMessageBox.warning(self, "ОШИБКА", "Закройте выбранный файл")

## **Описание применения средств откладки**

При написании проекта использовалось огромное количество раз встроенное средство откладки в IDE «PyCharm». В результате чего было исправлено большое количество ошибок различных видов.

При запуске программы без использования средства откладки в нештатных ситуациях не позволяет определить и диагностировать проблему: текст ошибки, стек вызовов с именами функций и номерами строк. Данную проблему решал запуск программы в режиме отладки, где повторное действие приводило к той же ошибке, но среда разработки при завершении программы показывала саму ошибку и ее суть.

# **Эксплуатационная часть**

## **Руководство оператора**

### **Назначение программы**

Разработка настольного приложения для сокращения времени просмотра базы данных содержащие сведение о грузах и размещении на стеллажах.

Дополнительные возможности программы:

* Добавление новых элементов;
* Изменение элементов;
* Удаление элементов;
* Вывод в Excel или JSON:
* Процент наполнения зала по сеансам;
* Самый популярный жанр.

### **Условия запуска программы**

Запуск программы с помощью файла с расширением .exe:

1. Стабильно работающее устройство на OC Windows 10.

Запуск программы с помощью исходного кода:

1. Стабильно работающее устройство на OC Windows 10;
2. Python версии 3.10 и выше;
3. Установленные библиотеки, указанные в файле «requirements.txt»

### **Выполнение программы**

При запуске программы появляется окно «ЖКХ» (приложение 1), с помощью которого можно: просмотреть любую таблицу в базе данных (приложение 2), добавить элемент в таблицу, изменить элемент в таблице, удалить элемент в таблице и вывести в Excel или JSON файл.

Страница «Добавить элемент» (приложение 3, 4, 5, 6) зависит от открытой таблицы, позволяет добавить новый элемент в базу данных.

Страница «Изменить элемент 1» (приложение 7) зависит от открытой таблицы, позволяет выбрать ID и поле, которые мы хотим изменить.

Страница «Изменить элемент 2» (приложение 8) показывает изменяемый ID и изменяемый поле, позволяет ввести новое значение.

Страница «Удалить элемент» (приложение 9) зависит от открытой таблицы, отображает ID, которые можно удалить.

Страница «Вывести в...» (приложение 10), предоставляется выбор вывода: процент наполнения зала по сеансам, самый популярный жанр и таблицу, в файл Excel или JSON.

### **Сообщение оператору**

В случае вывода в открытый файл Excel появится соответствующее сообщение об ошибке (приложение 11).

## **To-Do лист**

Данный проект находится на стадии разработки. Планируется исправление множества ошибок и добавления нового функционала:

1. Просмотр сразу нескольких таблиц;
2. Исправление ошибок с масштабированием;
3. Исправления мелких ошибок;
4. Добавление проверок при «Добавление элемента»;
5. Пересмотр дизайна основного окна.

# **Заключение**

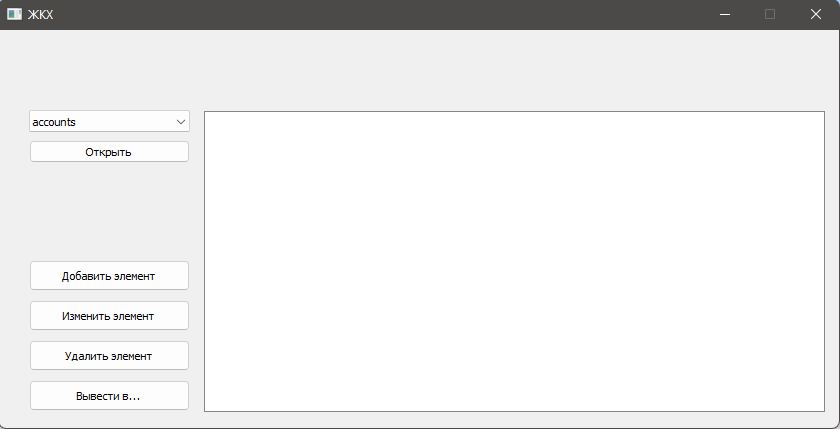
Проект после окончательной доработки может быть внедрен в управляющую компанию «ЖКХ», так как предоставляет минимальный, но основной функционал для работы в данной сфере.

# **Список использованной литературы и интернет-ресурсов**

1. Система «Инфокрафт ЖКХ 365»: <https://www.gkh365.ru/programms/1c_buhgalteria_zhkh/>
2. Система «Контур.ЖКХ»: <https://kontur.ru/kontur-jkh-kvartplata>
3. Система «Домиленд»: <https://domyland.ru/>
4. Язык программирования «Python»: <https://www.python.org/>
5. Среда разработки IDE «PyCharm»: <https://www.jetbrains.com/pycharm/>
6. База данных «SQLite»: <https://www.sqlite.org/index.html>
7. Графическая библиотека PyQt: <https://www.riverbankcomputing.com/software/pyqt/>
8. Таблица «Популярные языки программирования»: <https://www.tiobe.com/tiobe-index/>
9. Github: <https://github.com/lFaRenGeiTl>

# **Приложения**

## Приложение 1

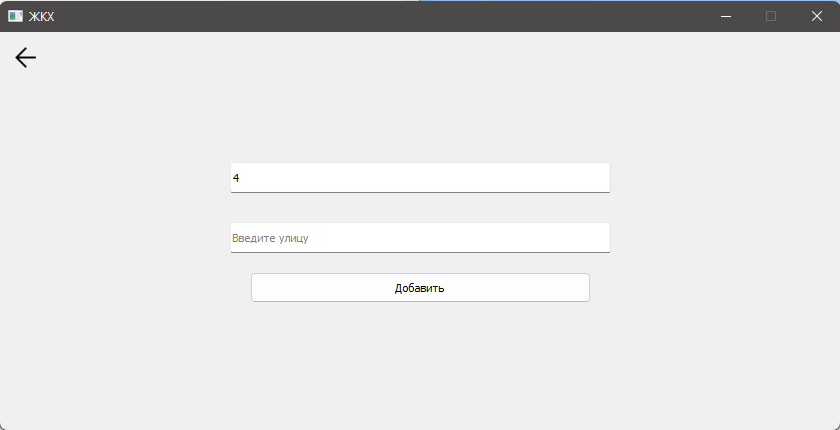


## Приложение 2

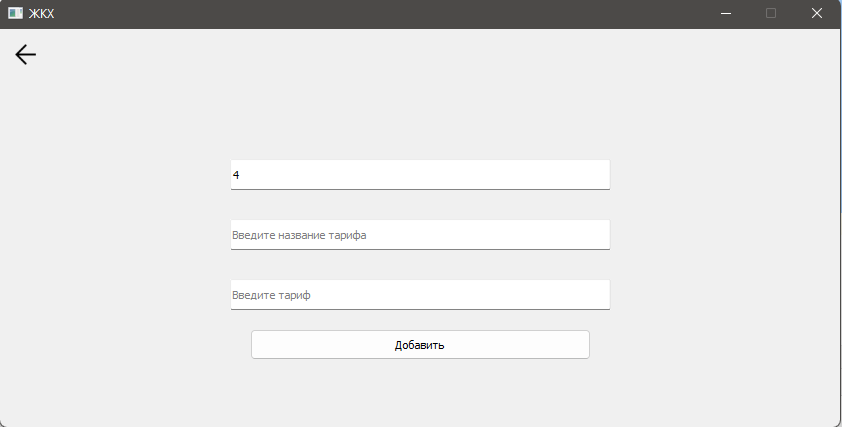
Изображение выглядит как текст

Автоматически созданное описание

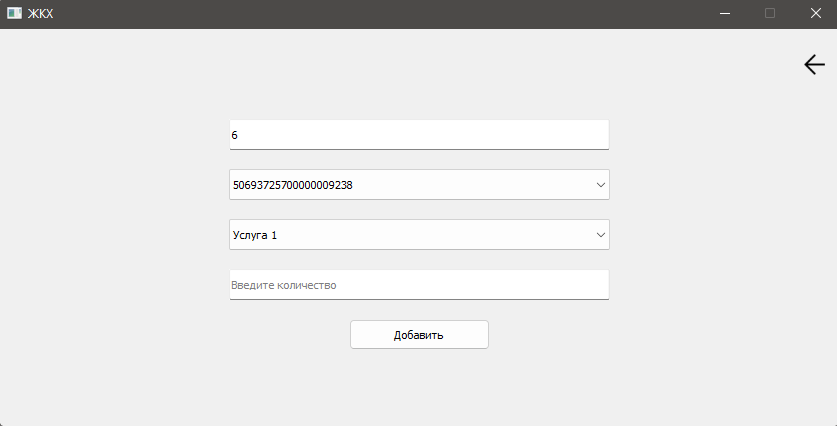
## Приложение 3



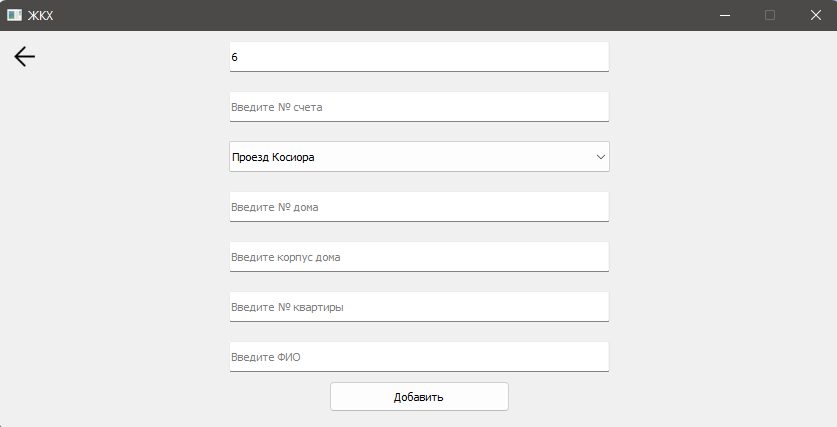
## Приложение 4



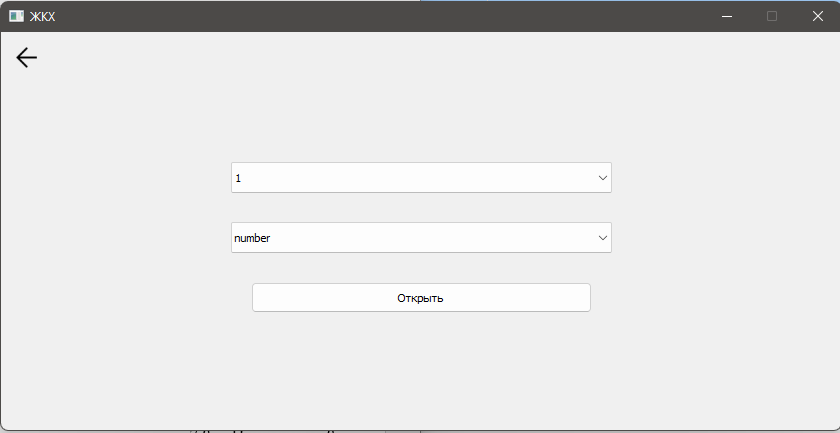
## Приложение 5



## Приложение 6



## Приложение 7

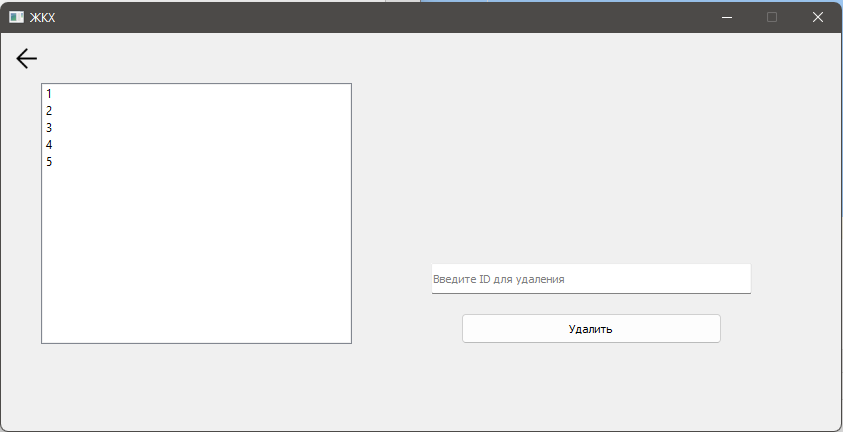


## Приложение 8

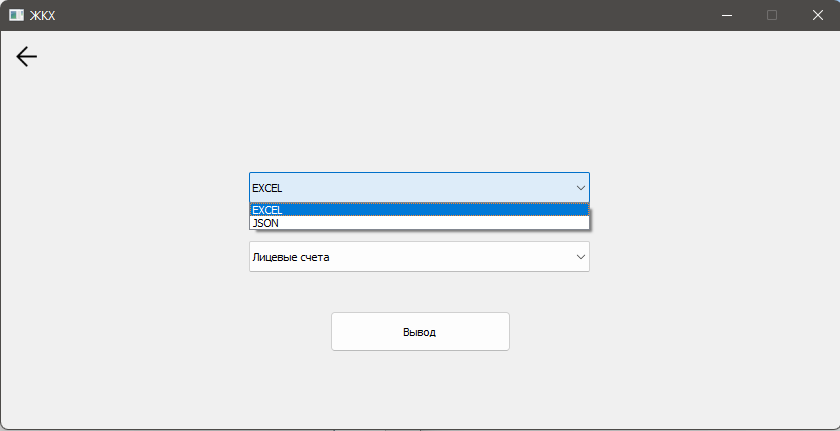
Изображение выглядит как текст

Автоматически созданное описание

## Приложение 9



## Приложение 10



## Приложение 11

