|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет экономики, менеджмента и бизнес-информатики* |
|  |
| Коковин Алексей Николаевич  **РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ПРИЁМА ЗАКАЗОВ НА ВЫПОЛНЕНИЕ РАБОТ ПО УСТАНОВКЕ ПРИБОРОВ УЧЁТА**  *Курсовая работа*  по направлению подготовки *09.03.04 Программная инженерия*  образовательная программа «Программная инженерия»   |  |  | | --- | --- | |  | Руководитель  к. ф.-м. н., доцент, доцент кафедры информационных технологий в бизнесе \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Л.Н. Лядова | |

Пермь, 2018 год

# Аннотация

Название работы: «Разработка информационной системы приёмов заказов на выполнение работ по установке приборов учёта».

Данная работа содержит в себе описание процесса анализа предметной области приёма заказов на выполнение работ по установке приборов учёта, проектирования и реализации соответствующей информационной системы, а также её тестирования и создания её документации.

Объём работы: 30 страниц.

Количество иллюстраций: 11.

Количество приложений: 13

Оглавление

[Введение 7](#_Toc509569443)

[Глава 1. Анализ задачи и разработка требований к системе 9](#_Toc509569444)

[1.1. Анализ аналогов 9](#_Toc509569445)

[1.2. Модель автоматизируемого бизнес-процесса 11](#_Toc509569446)

[1.3. Варианты использования 11](#_Toc509569447)

[1.4. Разработка требований к системе и техническое задание 12](#_Toc509569448)

[Глава 2. Создание модели данных 14](#_Toc509569449)

[2.1. ERD диаграмма 14](#_Toc509569450)

[2.2. ORD модель 15](#_Toc509569451)

[Глава 3. Проектирование 16](#_Toc509569452)

[3.1. Диаграмма последовательностей 16](#_Toc509569453)

[3.2. Диаграмма классов 16](#_Toc509569454)

[3.3. Пользовательский интерфейс 17](#_Toc509569455)

[Глава 4. Разработка запросов к данным 23](#_Toc509569456)

[4.1. Статические запросы к данным 23](#_Toc509569457)

[4.2. Динамические запросы данных 24](#_Toc509569458)

[Глава 5. Реализация информационной системы 25](#_Toc509569459)

[5.1. Диаграмма компонентов 25](#_Toc509569460)

[5.2. Класс Program 25](#_Toc509569461)

[5.3. Класс Operations 25](#_Toc509569462)

[5.4. Класс Checker 25](#_Toc509569463)

[5.5. Класс LessCont и классы-контейнеры для сущности 25](#_Toc509569464)

[5.6. Дополнение классов сущностей 26](#_Toc509569465)

[Глава 6. Тестирование и отладка приложения. Подготовка документации. 27](#_Toc509569466)

[6.1. Тестирование 27](#_Toc509569467)

[6.2. Руководство пользователя 27](#_Toc509569468)

[6.3. Руководство программиста 27](#_Toc509569469)

[Заключение 28](#_Toc509569470)

[Библиографический список 30](#_Toc509569471)

[Приложение А. IDEF0 диаграмма 31](#_Toc509569472)

[Приложение Б. Развёрнутое описание прецедентов. 35](#_Toc509569473)

[Приложение В. Техническое задание. 42](#_Toc509569474)

[1. Введение 42](#_Toc509569475)

[2. Основания для разработки 42](#_Toc509569476)

[3. Назначение разработки 42](#_Toc509569477)

[4. Требования к программе 42](#_Toc509569478)

[4.1. Требования к функциональным характеристикам 42](#_Toc509569479)

[4.2. Требования к надёжности 46](#_Toc509569480)

[4.2.1. Среднее время между отказами 46](#_Toc509569481)

[4.2.2. Время восстановления после отказа 46](#_Toc509569482)

[4.2.3. Потеря данных 46](#_Toc509569483)

[4.2.4. Недопустимая потеря данных 46](#_Toc509569484)

[4.3. Требования к условиям эксплуатации 46](#_Toc509569485)

[4.3.1. Климатические условия эксплуатации 46](#_Toc509569486)

[4.3.2. Требования к видам обслуживания 47](#_Toc509569487)

[4.3.3. Требования к численности и квалификации персонала 47](#_Toc509569488)

[4.4. Требования к составу и параметрам технических средств 47](#_Toc509569489)

[4.5. Требования к информационной и программной совместимости 48](#_Toc509569490)

[4.5.1. Требования к информационным структурам и методам решения 48](#_Toc509569491)

[4.5.2. Требования к исходным кодам и языкам программирования 48](#_Toc509569492)

[4.5.3. Требования к программным средствам, используемым программой 48](#_Toc509569493)

[5. Требования к программной документации 48](#_Toc509569494)

[6. Требования к технико-экономическим показателям 48](#_Toc509569495)

[6.1. Ориентировочная экономическая эффективность 48](#_Toc509569496)

[6.2. Предполагаемая годовая потребность 49](#_Toc509569497)

[7. Стадии и этапы разработки 49](#_Toc509569498)

[7.1. Стадии разработки 49](#_Toc509569499)

[7.2. Этапы разработки 49](#_Toc509569500)

[7.3. Содержание работ по этапам 49](#_Toc509569501)

[8. Порядок контроля и сдачи 50](#_Toc509569502)

[8.1. Виды испытаний 50](#_Toc509569503)

[8.2. Общие требования к приемке работы 50](#_Toc509569504)

[Приложение Г. ERD диаграмма. 51](#_Toc509569505)

[Приложение Д. ORD диаграмма. 52](#_Toc509569506)

[Приложение Е. Диаграмма последовательностей. 53](#_Toc509569507)

[Приложение Ж. Диаграмма классов. 57](#_Toc509569508)

[Приложение И. Статические запросы к сущности User 59](#_Toc509569509)

[Приложение К. Динамические запросы данных 62](#_Toc509569510)

[Приложение Л. Диаграмма компонентов. 63](#_Toc509569511)

[Приложение М. Тестирование приложения 64](#_Toc509569512)

[Приложение П. Руководство пользователя 78](#_Toc509569513)

[1. Введение 78](#_Toc509569514)

[1.1. Область применения 78](#_Toc509569515)

[1.2. Краткое описание возможностей 78](#_Toc509569516)

[1.3. Уровень подготовки пользователя 78](#_Toc509569517)

[1.4. Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю 78](#_Toc509569518)

[2. Назначение и условия применения 79](#_Toc509569519)

[3. Подготовка к работе 79](#_Toc509569520)

[3.1. Состав и содержание дистрибутивного носителя данных 79](#_Toc509569521)

[3.2. Порядок загрузки данных и программ 79](#_Toc509569522)

[3.3. Порядок проверки работоспособности 79](#_Toc509569523)

[4. Описание операций 79](#_Toc509569524)

[4.1. Выполняемые функции и задачи 79](#_Toc509569525)

[4.2. Описание операций технологического процесса обработки данных, необходимых для выполнения задач. 80](#_Toc509569526)

[5. Аварийный ситуации 88](#_Toc509569527)

[6. Рекомендации по освоению 89](#_Toc509569528)

[Приложение Р. Руководство программиста 90](#_Toc509569529)

[Назначение и условия применения систем 90](#_Toc509569530)

[Характеристика программы 90](#_Toc509569531)

[Обращение к программе 90](#_Toc509569532)

[Входные и выходные данные. 91](#_Toc509569533)

[Сообщения 91](#_Toc509569534)

# Введение

В целях эффективного и рационального использования энергетических ресурсов, согласно статье 13 Федерального закона «Об энергосбережении и о повышении энергетической эффективности и о внесении изменений в отдельные законодательные акты Российской федерации» № 261-ФЗ от 23 ноября 2009 года[1], все потребляемые энергоресурсы (вода, тепло, эл.энергия, газ) подлежат обязательному учёту с применением приборов учета. В связи с этим, сфера установки приборов учёта стремительно развивается, конкуренция на рынке повышается. Одним из способов получения конкурентного преимущества является автоматизация бизнес-процессов предприятия.

Отсутствие автоматизации на этапе принятия заказа негативно воздействует на эффективность работы предприятия: менеджеры по продажам, а иногда даже сами мастера по установке должны самостоятельно связываться с заказчиком, чтобы уточнить адрес, удобное время начала работ по установке приборов учёта, количество и наименования устанавливаемых счётчиков и прочие детали заказа. Автоматизация данной процедуры поможет не только сэкономить время на переговорах с заказчиком, но также позволит ввести учёт заказ и автоматизировать отчётность.

Проблема автоматизации бизнес-процессов является очень разработанной, для решения данной проблемы создано большое количество различных программных решений, однако решения конкретно для данной предметной области редки. Безусловно, многие предприятия уже реализовали определённые механизмы принятия заказов на приборы учёта. Однако большинство из них являются слишком примитивными и малоинформативными. В частности, находящиеся на первых местах в поисковой системе «Google» сайты [2-5] предлагают оставить только свою контактную информацию, а сам заказ формируется при разговоре по телефону. Таким образом, создание информационной системы, позволяющей автоматизировать приём заказа на установку счётчика, является актуальной проблемой.

*Объектом исследования* работы является учёт заказов на установку приборов учёта. *Предметом исследования* является информационная система, которая способна автоматизировать данный бизнес-процесс.

*Цель работы* – реализация информационной системы для приёма заказов на установку приборов учёта. Данная система должна производить взаимодействие с заказчиком посредством электронной почты, сайта или осуществлять поддержку оператору при формировании заказа по телефону.

Для достижения поставленной цели требуется выполнить ряд *задач:*

1. Произвести этап анализа.
   1. Выполнить анализ существующих решений задачи автоматизации учёта заказов на установку приборов учёта.
   2. Разработать модель автоматизируемого бизнес-процесса.
   3. Разработать варианты использования системы.
   4. Разработать требования к разрабатываемой информационной системе и составить техническое задание.
2. Создать модель базы данных.
   1. Выполнить проектирование сущностной модели базы данных.
   2. Выполнить перенос сущностной модели в объектную модель базы данных.
3. Выполнить проектирование разрабатываемой информационной системы.
   1. Разработать динамическую модель информационной системы.
   2. Разработать статическую модель информационной системы.
   3. Разработать пользовательский графический интерфейс.
4. Выполнить разработку запросов к данным.
5. Выполнить реализацию информационно системы.
   1. Реализовать информационную систему в соответствии с разработанными требованиями, описанными в техническом задании на данную систему.
   2. Произвести тестирование реализованной информационной системы.
   3. Разработать документацию по разработанной информационной системе.

В ходе решения поставленных задач предполагается использование следующих *методов исследования*: метод пошагового проектирования, методы объектно-ориентированного проектирования, методы объектно-ориентированного программирования.

*Практическая значимость* данного исследования заключается в реализации информационной системы, которая может быть внедрена на предприятии по установке приборов учёта.

# Анализ задачи и разработка требований к системе

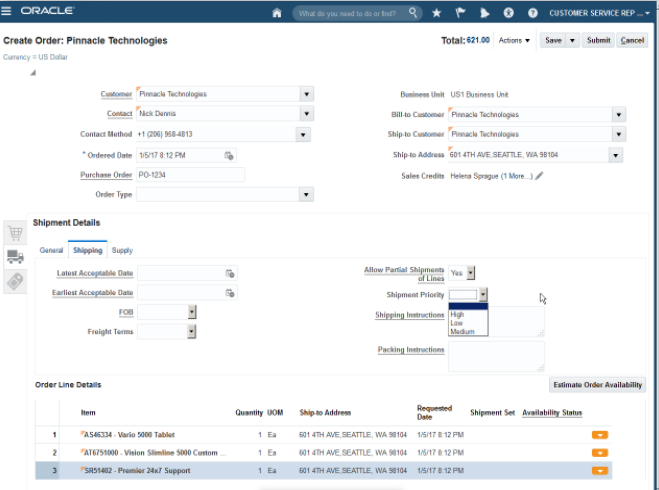
В данной главе производится анализ аналогов, строится модель автоматизируемого бизнес-процесса, и формулируются требования к разрабатываемой системе и составляется техническое задание.

## Анализ аналогов

Безусловно, задача обработки приёма заказов клиентов является далеко не новой. Существует множество различных настраиваемых систем, которые применимы практически в любой предметной области. В данной работе будет представлен сравнительный анализ трёх подобных систем.

Особенно заметна популярность подобных систем за рубежом. В частности, в данной работе будет рассмотрена система от компании Conexiom[6]. Данный программный продукт предоставляет возможность обработки заказов, поступающих из различных источников: электронная почта, веб-форма или сайт, которые после этого конвертируются, структурируются и передаются в уже существующую ERP систему. К сожалению, это означает, что данный продукт не может решить поставленную задачу сам по себе из-за требования наличия ERP системы.

Область управления заказами не была проигнорирована компанией Oracle. Программный продукт «Oracle Order Management»[7] (см. рис. 1.1) предоставляет возможность производить ввод информации о заказе с проверками в реальном времени: к примеру, если товара нет в наличии, то будет получено соответствующее оповещение. Кроме «ручного» заполнения заказа также доступно получение заказов из вэб или файлов в форматах edi или b2b. На этом возможности системы не ограничиваются, в любой момент система даёт возможность просмотреть различные экономические параметры, связанные с заказами, обновляемые в реальном времени. Данная система обладает двумя существенными недостатками, которые препятствуют его применению для решения данной задачи. Во-первых, данная система рассчитана на фирмы, размеры которых достаточно сложно достичь в выбранной предметной области. Результатом такого несоответствия является пониженная экономическая эффективность данного решения. Во-вторых, специфика данной предметной области (приборы учёта не являются просто товаром, они требуют услуг по их установке) увеличивает необходимость тщательной настройки данного программного обеспечения, что влечёт за собой необходимость найма программиста с сертификатом Oracle, спрос на которых достаточно высок на рынке труда и как следствие требует высокой заработной платы.



***Рисунок 1.1. Oracle Order Management***

Рассматривая возможные аналоги, невозможно не обратить внимание решения, используемые непосредственно в области установки приборов учёта. В частности, будет рассмотрена информация сайта Пермской компании ООО «Новогор-Прикамье»[8]. В первую очередь следует отметить, что данная компания занимает только приборами учёта воды, в то время как выбранная предметная область подразумевает в том числе и приборы учёта газа и электричества. К сожалению, информация об используемых данной компанией информационной системе отсутствует в свободном доступе, однако возможен анализ непосредственного бизнес-процесса приёма заказа. Данная компания предусмотрели 3 различных способа оформления заявки: в офисе компании, по телефону или отправить заполненный бланк по электронной почте. Полностью проанализировать бизнес-процесс данной компании не предоставляется возможным, однако достаточно длительное время обработки заказа (гарантируется выполнение заказа в течение 7 дней) может подразумевать за собой отсутствие автоматизации планирования работ.

## Модель автоматизируемого бизнес-процесса

Акторами автоматизируемого бизнес-процесса являются оператор – человек ответственный за учёт заказов и мастер – человек, ответственный за установку приборов учёта. Сам бизнес-процесс подразделяется на непосредственный приём заказа и его закрытие. Построенная IDEF0 диаграмма расположена в приложении А.

## Варианты использования



***Рисунок 1.2. Диаграмма прецедентов.***

Варианты использования системы представлены в виде диаграммы прецедентов (рис. 1.2). В ходе анализа были добавлены дополнительные прецеденты Выполнить анализ, и Заказать по email. Оба были введены, т.к. из-за автоматизации они будут целесообразны. Полное описание прецедентов находится в приложении Б.

## Разработка требований к системе и техническое задание

Основным функциональным требованием к системе является возможность составления и сохранения заказа в базе данных. Данный функционал является обязательным условием для автоматизации бизнес-процесса приёма заказа. Кроме того, должна быть реализована проверка входных данных: к примеру, длина номера паспорта всегда составляет 10 цифр, ровно как и длина ИНН организации. Тем не менее, ошибки ручного ввода являются скорее правилом, нежели исключением. Поэтому наличие функции изменения данных также является необходимостью. Тем не менее, делегировать возможность редактирования данным операторам является не самой безопасной практикой, поэтому требуется наличие различных категорий пользователей, а в частности Администратора и Оператора.

Данная система требует также и некоторых дополнительных функциональных возможностей. Выборка данных из базы данных по собираемым пользователем параметрам поможет группам пользователей, отличных от Оператора и Администратора. В частности, группу, имеющую данные функциональные возможности, имеет смысл назвать «Аналитики».

Из системных функциональных возможностей следует отметить возможность сохранения и загрузки базы данных в файл. Для выполнения данных операций было решено использовать сериализацию объектов Xml, поскольку данный формат данных зарекомендовал себя как проверенный временем, многое программное обеспечение поддерживает импорт данных именно данного формата, что открывает возможность взаимодействия с иными информационными системами. Кроме сериализации всей базы данных, имеет смысл производить сохранение результаты выборок данных в формате Xml, поскольку перенос данных о запросе в другое программное обеспечение позволит выполнить визуализацию данных и как следствие более качественный анализ данных и принять грамотное обоснованное управленческое решение.

Тем не менее, реализовать сохранение запросов только в формате Xml имеет свой недостаток: для обработки данных потребуется либо специализированное приложение, которое не является распространённым, либо создание оного. Чтобы данный недостаток больше не имел места быть, было решено реализовать сохранение результатов запросов при помощи программы Microsoft Office Excel. Данное программное обеспечение является широко распространённым среди потенциальных пользователей и позволяет выполнять операции над данными: различные расчёты и визуализацию. Единственным недостатком данного решения является требование наличия данного программного обеспечения на компьютере пользователя.

К нефункциональным требованиям к программе относится использование графического интерфейса пользователя Windows Forms. Программы, разработанные при помощи данной технологии, являются широко распространёнными, поэтому используемые элементы управления будут понятны конечному пользователю, что позволит уменьшить порог вхождения для использования данной программы.

По составленным требованиям было составлено техническое задание в соответствии с ГОСТ 19.201–78 [9], представленное в приложении В.

# Создание модели данных

В данной главе рассматривается построение диаграммы сущностей в нотации Crow’s foot и объектной модели при помощи средств Entity Framework.

## ERD диаграмма

Построенная ERD диаграмма представлена в приложении Г.

Главной сущностью данной системы является Заказ. Данная сущность определяется числом Номер заказа.

Заказ вводится в систему пользователями. Пользователь – это сущность, имеющая два строковых свойства Логин и Пароль, а также перечисление Тип пользователя (Администратор, Оператор или Аналитик). Связь Пользователь – Заказ имеет тип один ко многим.

Заказ имеет заказчика. Заказчик – это сущность, имеющая три строковых свойства ФИО, Номер телефона и Номер паспорта. Номера хранятся как строки для упрощения проверки корректности ввода и ввиду отсутствия необходимости выполнять над ними какие-либо операции кроме сравнения на равенство. Связь Заказчик – Заказ имеет тип один ко многим, т.к. возможна ситуация с заказчиками, имеющими несколько заказов. При этом существует подвид Заказчика – Компания. Он включает в себя 2 дополнительных строковых поля Название организации и ИНН организации.

Заказ имеет адрес. Адрес – это сущность, которая является последовательной комбинацией из сущностей Дом, Улица и Город. Улица и Город имеют строковые поля Названия, Дом и Адрес имеют поля Номер, однако Номер в сущности Адрес подразумевает номер квартиры, поэтому он имеет численный тип, а Номер в сущности Дом имеет строковый тип, т.к. существуют дома с номерами типа «1А».

Заказ имеет заказные позиции. Заказная позиция – это сущность, которая подразумевает за собой конкретные установленные счётчики. Данная сущность содержит 2 свойства типа Дата и Время под названиями Начало и Конец. Для заказанных счётчиков данные свойства являются рекомендуемыми временными промежутками для установки, в то время как для установленных счётчиков Начало означает время установки, а Конец – время окончания поддержки данного прибора. Кроме того, заказная позиция имеет регистрационный номер, который не имеет значения для ещё не установленного счётчика. Регистрационный номер имеет тип строка для повышения настраиваемости.

Каждая заказная позиция имеет статус. Статус заказной позиции – это сущность с единственным свойством Название типа строка. Было принято решение использовать отдельную сущность вместо перечисления для повышения настраиваемости системы.

Каждая заказная позиция содержит в себе счётчик. Счётчик – это сущность с единственным свойством Название типа строка. Каждый счётчик принадлежит к некоторому типу (например, Водопроводный, Электрический, Газовый). Тип счётчика – это отдельная сущность с строковым свойством название. Тип счётчика был выведен в отдельную сущность для повышения настраиваемости системы.

Заказная позиция может иметь закреплённого за ней человека, ответственного за его выполнение. Человек – это сущность с строковым свойством ФИО. Человек может устанавливать некоторые типы счётчиков. Чтобы избежать связи типа многие ко многим, введена промежуточная сущность Ставка со связями Человек-Ставка и Тип счётчиков – Ставка типа один ко многим.

Все сущности, кроме Заказа, определяются соответствующими свойствами Идентификационный номер. Это необходимо для устранение аномалии изменения.

## ORD модель

На основании построенной ERD модели была построена ORD модель при помощи Microsoft Visual Studio и ADO.NET Entity Framework. Разработанная модель расположена в приложении Д.

# Проектирование

В данной главе рассматривается построение модели поведения системы, статической структуры системы и пользовательский интерфейс.

## Диаграмма последовательностей

В ходе работы над курсовой работой была разработана диаграмма последовательностей. Данная диаграмма разбирает прецеденты «Приём заказа по телефону», «Отмена заказа по телефону», «Закрытие заказа» и «Приём заказа по электронной почте».

## Диаграмма классов

Диаграмма классов, разработанная при помощи средств Microsoft Visual Studio, находится в приложении Ж.

Operations – класс, в котором реализованы все основные функции работы с базой данных. Для каждой сущности реализованы 4 операции: добавление, редактирование, удаление, поиск по идентификационному номеру и выборка. Функции добавления, редактирования и удаления возвращают логическое значение, равное True при успешном применении операции, и False в противном случаи. Кроме того функции возвращают строку результата операции, в которой содержится сообщение об успешном добавлении или сообщение об ошибке. Помимо этого, в данном классе расположена функция входа в систему TryEntry, которая на вход принимает логин и пароль, и возвращает соответствующий экземпляр сущности User или null при ошибке и строку Message, являющейся сообщением о результате операции. Operations содержит событие вызова предупреждающего сообщения AttentionsMessage, которое происходит когда при удалении происходит каскадное удаление других записей.

Operations содержит поле cont – контекст базы данных, с которой работает программа. NextId – коллекция следующих индексов базы данных для каждого типа сущности, строящаяся при использовании InicializeNextId и используемая при считывании данных из XML файла.

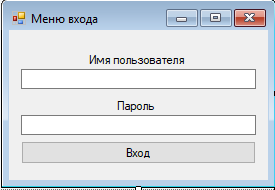
Класс XMLQuery реализует процессы сериализации всех данных базы данных в файл формата XML и десериализации файла и добавление соответствующих данных в базу данных.

Класс Checker реализует функции проверки строк, используемых для доабвления записей в базу данных. IsFio проверяет, находится ли строка в формате Фамилия Имя Отчество, IsHouseNumber проверяет, является ли строка номером дома в формате числа или «1А», IsINN проверяет, является ли строка номером ИНН организации, IsName проверяет, является ли данная строка названием, IsNumber проверяет, является ли строка натуральным числом и IsPhoneNumber проверят, является ли строка номером телефона.

Класс LessCont является промежуточным контейнером, который используется для сериализации, поскольку сериализировать сущности Entity Framework напрямую не является тривиальной задачей. LessCont содержит контейнеры классов типа \_Сущность, свойства которых копируют свойства сущностей объектной модели.

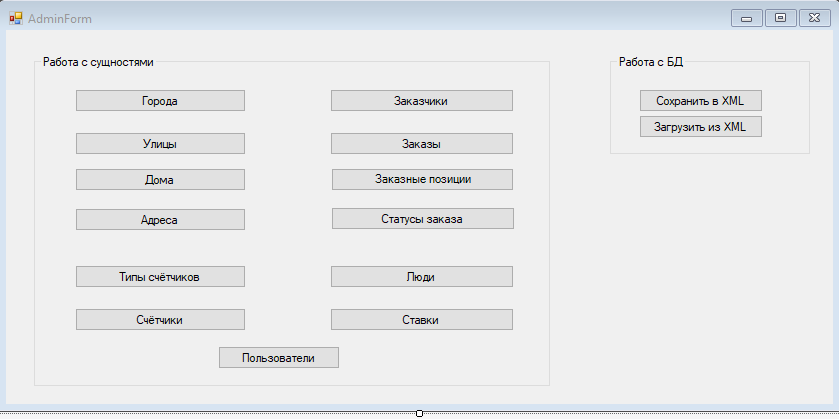
## Пользовательский интерфейс

При запуске программы запускается окно входа, в которое вводятся логин и пароль (рис. 3.1). Поскольку пользователей существует 3 типа: Администратор, Аналитик и Оператор, для каждого типа разработаны формы AdminForm, AnaliticForm и OperatorForm.



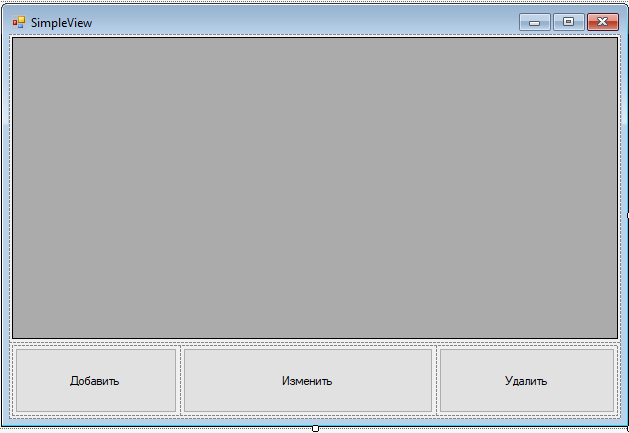
***Рисунок 3.1.Меню входа LoginForm***

Форма AdminForm (см. рис. 3.2) содержит кнопки лдя перехода для выполнения действий над записями каждой сущности, а также сохранение и загрузки базы данных в файл формата Xml.



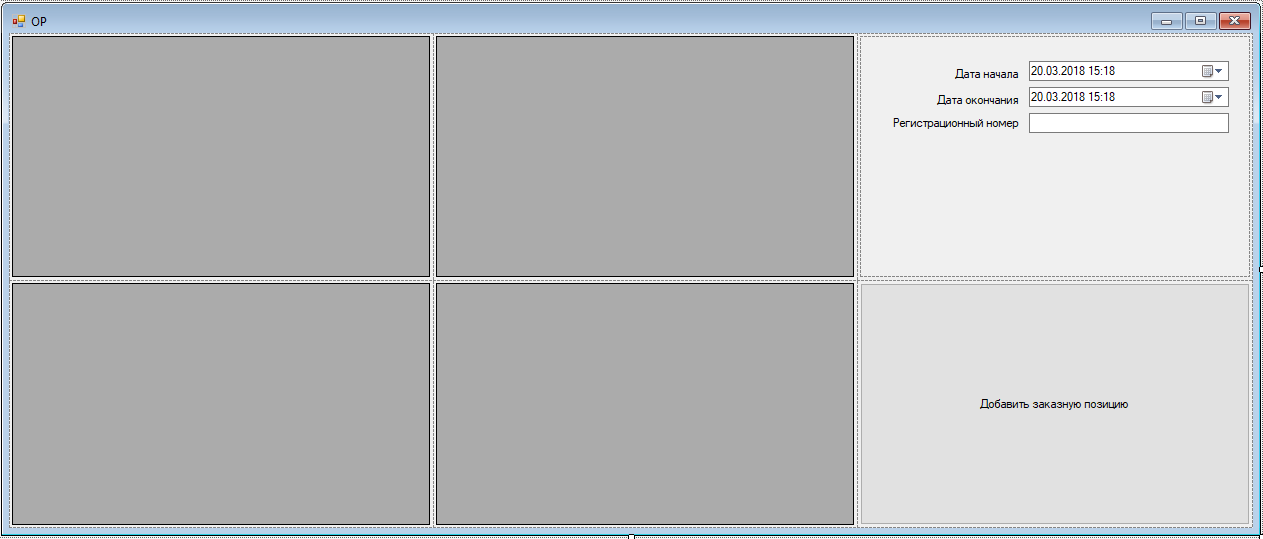
***Рисунок 3.2.Режим администратора AdminForm.***

При выборе сущности открывается форма SimpleView (рис. 3.3), содержащая элемент DataGridView, на который выводятся все записи данной сущности. Способ выделения элементов в таблице – выбор полной строки. После выделения нужной записи пользователь нажимает на кнопку с операцией, которую требуется выполнить над выделенной записью. Удаление происходит незамедлительно, а добавление и изменение открывают новую форму.



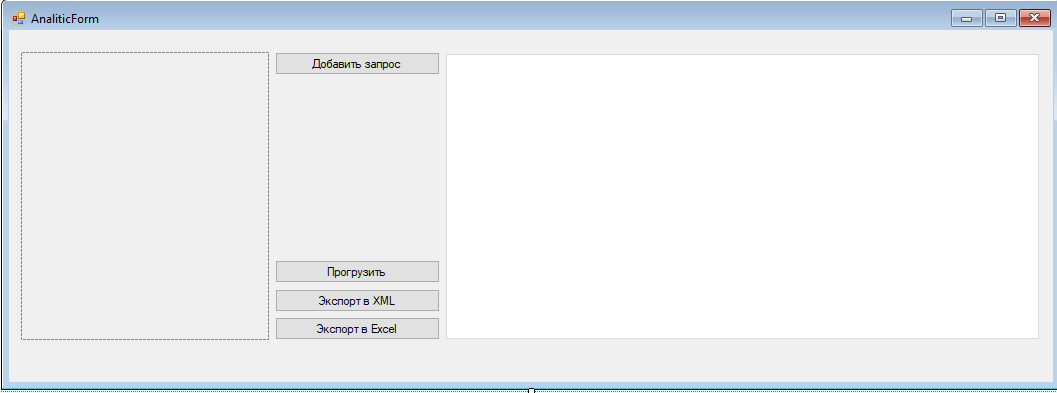
***Рисунок 3.3.Работа с сущностями SimpleView.***

Открываемая форма при добавлении и изменении различается в зависимости от типа выбранной сущность. По факту, каждый тип сущности Сущность имеет собственную форму OPСущность, каждая форма OPСущность наследует форму OP, содержащую свойства CurrentUser – объект сущности User, являющийся текущим пользователем, и Id – целое число, которое имеет значение при выборе изменения записи и является её идентификационным номером. К примеру, при работе с заказными позициями, форма содержит 4 элемента управления типа DataGridView, каждая из которых выводит все записи типа Status, Order, Meter и Person, а также 2 элемента DateTimePicker для свойств Начало и Конец, элемент Textbox для ввода регистрационного номера и для выполнения действия (рис. 3.4).



***Рисунок 3.4.Работа с сущностями типа OrderEntry в форме OPOrderEntry.***

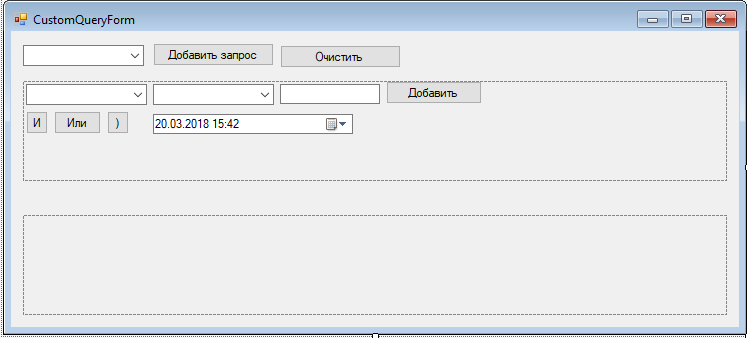
При входе пользователя типа Аналитик запускается форма AnaliticForm (см. рис. 3.5). Слева на форме расположен элемент FlowLayoutPanel, в котором отображаются записи о текущих запросах. Сами запросы задаются при нажатии соответствующей кнопки «Добавить запрос». Кнопка «Быстрая диаграмма заказы» запускает отображение соответсвующего запроса в отдельной форме. При нажатии кнопки «Прогрузить» начинается обработка введённых запросов, которые после загрузки отобразятся в элементах DataGridView, каждый из которых располагается на отдельной вкладке TabPage элемента TabControl, расположенного справа. При нажатии кнопки «Экспорт в Excel» открывается окно программы Microsoft Office Excel, в новой книге которого создаются вкладки с полученными результатами запросов, а при нажатии кнопки «Экспорт в XML» открывается диалоговое окно выбора директории, в которую сериализуются результаты запросов.



***Рисунок 3.5.Работа в режиме аналитика AnaliticForm.***

При нажатии кнопки «Добавить запрос» открывается форма CustomQueryForm (см. рис. 3.6). Данная форма реализует конструктор запросов, используемый для получения данных по заданным критериям. ComboBox, расположенный в левом правом углу, отвечает за выбираемый аналитиком тип сущности, записи которой будут результатом выборки. После выбора сущности активируются элементы, расположенные ниже. Левый ComboBox отвечает за выбор свойства сущности, следующий ComboBox отвечает зы выбор отношения к значению данного свойства (например «равно» и «неравно» для свойства Name сущности City, или «Вход», если свойство тоже является сущностью) который изменяет значение своих элементов в зависимости от выбранного свойства, TextBox, отвечает за значение в добавляемом условии, DateTimePicker, который виден только при рассмотрении сущности OrderEntry и свойств StartTime и EndTime. При нажатии кнопки «Добавить» полученное условие добавляется в выражение, которое отображается в элементе FlowLayoutPanel, расположенном ниже. Кнопки «И» и «ИЛИ» используются для задания логической связи налагаемых на данные нескольких условий между собой, а кнопка «Закрывающаяся скобка» отвечает за возврат к предыдущему рассматриваемому элементу.

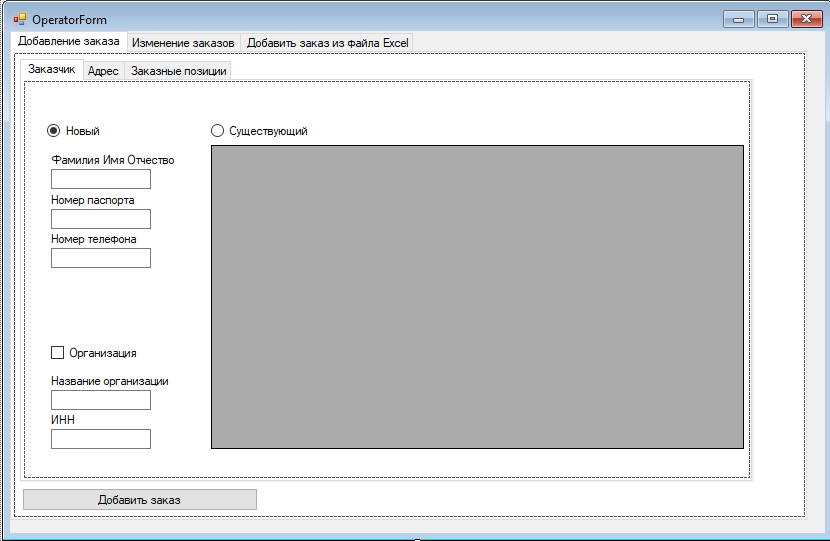
Для того, чтобы проиллюстрировать работу конструктора, рассмотрим как будет выглядеть следующая выборка: Вывести все заказные позиции, у которых заказчик имеет имя Коковин Алексей Николаевич, и имеет статус «Готово». Данная выборка должна выдать объекты «OrderEntry», поэтому в правом верхнем углу следует выбрать «OrderEntry». Затем следует указать первое условие: заказчик имеет имя Коковин Алексей Николаевич. Для этого следует выбрать свойство «Order» и отношение «Вход» и нажать добавить, затем выбрать свойство «CustomerOfCompany» (поскольку требуется включить и частных заказчиков, и компании), затем выбрать свойство «FIO» и отношение «равно», а в TextBox вписать строку «Коковин Алексей Николаевич» и нажать «Добавить». Для того, чтобы вернуться к сущности «OrderEntry» следует дважды нажать на закрывающую скобочку, нажать кнопку И, войти в свойство «Status», выбрать свойство «Name» с отношением «равно» и ввести в TextBox строку «Готово», нажать «Добавить» и нажать «Закрывающаяся скобка», чтобы вернуться к изначальной сущности. В результате запрос будет отображён ниже в следующем виде: «OrderEntry (Order(CustomerOrCompany Name == “Коковин Алексей Николаевич”)) И (Status Name == “Готово”). Для добавление сформированного запроса следует нажать на кнопку «Добавить запрос», а для очисти текущего запроса – «Очистить».



***Рисунок 3.6.Создание запроса CustomQueryForm.***

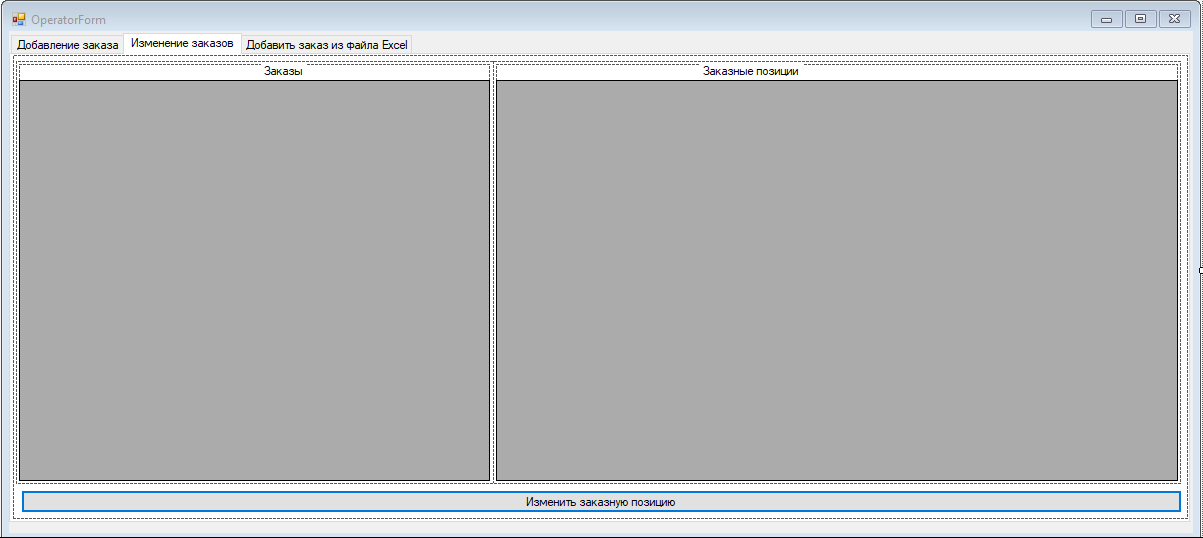
При заходе в режиме оператора открывается форма OperatorForm (см. рис. 3.7). Данная форма имеет 3 вкладки, отвечающие за функции ручного добавления заказа, изменения заказа и добавление заказа из файла Excel.

Вкладка «Добавление заказа» содержит 3 вкладки, отвечающие за ввод заказчика, адреса и заказных позиций соответственно. Ввод информации о заказчике подразумевает либо добавление нового заказчика, либо выбор из уже сущесвующих. В первом случае заполняются элементы TextBox, расположенные в левой части формы, в ином случае выбирается необходимая запись из DataGridView,который расположен левее. Вкладка Адрес содержит 4 элемента TextBox, отвечающие за ввод адреса, а вкладка «Заказные позиции» содержит 2 элемента DataGridView, используемые для выбора счётчиков и отображения уже добавленных заказных позиций.



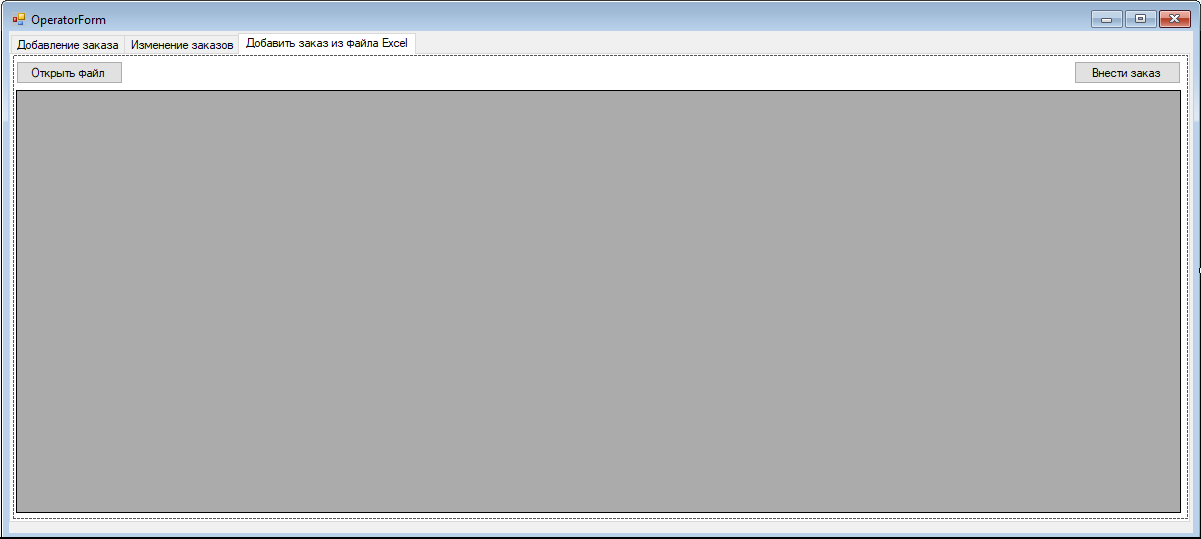
***Рисунок 3.7.Ручное добавление заказа в режиме оператора OperatorForm***

Вкладка «Изменение заказов» (рис. 3.8) содержит 2 элемента DataGridView, для заказов и их заказных позиций соответственно. При выборе заказа в левой таблице, в правой таблице отображаются его заказные позиции. Для редактирования заказной позиции производится выбор заказной позиции в правой таблице и нажатие на кнопку «Изменить заказ». В результате данных действий выполняется открытие формы OPOrderEntry » (см. рис. 3.4) в режиме изменения записи.



***Рисунок 3.8.Изменение заказных позиций в режиме оператора OperatorForm***

Для добавления заказа из файла формата Excel используется вкладка «Добавить заказ из файла Excel» (см. рис. 3.9). Вкладка представляет собой элемент DataGridView с двумя кнопками «Открыть файл» и «Внести заказ».



***Рисунок 3.9.Добавление заказа из файла Excel OperatorForm***

При нажатии кнопки «Открыть файл» открывается диалоговое окно выбора файла. Расширения поддерживаемых файлов – “.xls” и “.xlsx”. Затем выбранный файл открывается при помощи Microsoft Office Excel. Данные из файла переносятся в элемент DataGridView формы. После того, как пользователь удостоверится в корректности считывания данных, ему следует нажать на кнопку «Внести заказ», после чего заказ будет помещён в базу данных.

# Разработка запросов к данным

В данной главе рассмотрены реализованные запросы к данным базы данных.

## Статические запросы к данным

Статические запросы на добавление, изменение, поиск по ключу и удаление по ключу реализованы в классе Operations и имеют вид AddEntity, ChangeEntity, FindEntity и RemoveEntity соответственно. Рассмотрим примеры реализаций данных функций на единственной сущности User. Реализация данных функций находится в приложении И.

Функция AddUser принимает на вход следующие параметры: тип пользователя, логин, пароль, выходная строка сообщения и переменную save типа bool со значением по умолчанию равному true и отвечающие за вызов функции SaveChanges. В первую очередь функция запускает запрос, который проверяет, имеются ли пользователи, у которых логин идентичен входному. Подобная проверка по реальному ключу реализована практически во всех сущностях, кроме Order. Если проверка по реальному ключу показала, что подобная запись уже имеется, то соответствующее сообщение выводится в строку Res и функция возвращает false. После проверки происходит непосредственное создание экземпляра сущности и её добавление в базу данных. По окончании добавления строке результата присваивается соответствующее сообщение и функция принимает значение true. В ходе добавления записи в базу данных могу формироваться исключения, поэтому код функции находится в блоке try и при получении исключения строка результата принимает значение сообщения исключения и функция возвращает значение false.

Функция ChangeUser принимает на вход те же самые параметры, что и функиця AddUser с дополнительным параметром id – идентификационный номер изменяемой записи. В начале функции происходит проверка, имеется ли пользователь с таким же логином, но с другим идентификационным номером. Если такой пользователь есть, это означает, что мы пытаем изменить логин пользователя на логин другого уже существующего пользователя, что не является позволительным, поэтому в данном случае строка вывода результата получает соответствующее значение и функция возвращает значение false. Подобная проверка практически аналогична проверке, реализованной в функциях добавления, однако в выражение добавляется условие неравенства идентификационных номеров. После проверки производится поиск пользователя с введённым идентификационным номером, изменение его свойств в случае его наличия и сохранение изменений.

Функции типа RemoveEntity принимают на вход идентификационный номер id удаляемого объекта, строку Res для вывода сообщения, логическую переменную save обозначающую необходимость сохранения изменений и имеющую значение по умолчанию true, и логическую переменную check обозначающую необходимость вывода сообщения одобрения пользователя при удалении записи, имеющей другие зависимые записи, которые также будут удалены. Сначала функция удаления начинает поиск записи с соответствующим идентификационным номером, который требуется удалить. В случае отсутствия данной записи строка сообщения принимает соответствующее значение и функция возвращает значение false. Затем происходит удаление при выполнении одного из трёх условий: отсутствуют записи, зависимые от данной, переменная check принимает значение true или вызываемое событие предупреждающего сообщения принимает значение true. В противном случае удаление отменяется.

Функция поиска по идентификационному номеру является простейшей функцией, представляющей из себя запрос на выборку всех элементов с заданным идентификационным номером и применением функции FirstOrDefault к получаемой выборке.

## Динамические запросы данных

Помимо статических запросов были реализованы динамические запросы данных. Все динамические запросы данных являются запросами на выборку записей определённых сущностей, которые удовлетворяют некоторому условию. Условие выбора данных подаётся как входной параметр функции в виде делегата функции, принимающей на вход элемент сущности а на выход выдающей логическое значение, которое равно true, если данный элемент сущности следует добавить в выборку. Как результат, созданные функции

Необходимость использования динамических запросов данных обусловлена необходимостью реализации конструктора запросов по критериям, определяемым пользователем. Код разработанных запросов находится в приложении К.

# Реализация информационной системы

В данной главе расположено описание реализованной информационной системы по разработанным классам.

## Диаграмма компонентов

Диаграмма компонентов реализованной системы находится в приложении Л

## Класс Program

В класс Program были внесены дополнительные методы, используемые при работе с элементами DataGridView. В частности были реализованы методы скрытия неинформативных столбцов, появляющихся при указании коллекции сущностей как источник данных, а также метод переименования заголовков столбцов на русский язык.

## Класс Operations

В классе Operations были реализованы все запросы к данным, описанные в главе 4. Помимо запросов к данным, была реализована функция InitializeNextId, которая инициализирует коллекцию NextId, используемую при загрузке данных из файла xml.

## Класс Checker

В классе Checker реализованы функции проверки корректности строк, используемых для создания экземпляров сущностей. Функции проверки IsName, IsFIO, IsPhoneNumber, IsINN, IsHouseNumber,IsNumber,IsNumberAr реализованы при помощи использования соответствующих регулярных выражений.

## Класс LessCont и классы-контейнеры для сущности

Сериализовать объекты типа сущностей напрямую не является тривиальной задачей. Каждая связь типа один ко многим создаёт объект интерфейса IEnumerable, сериализация которых не поддерживается. Поэтому попытка сериализовать объект сущности приведёт к генерации исключения. Для решения данной проблемы были созданы классы-контейнеры типа \_Сущность, в которые копируются все значимые характеристики объекта: все его свойства, не являющиеся сущностями, и идентификационные номера связанных объектов сущностей. Для каждого контейнера была реализована функция Tran, которая производит преобразование объектов сущности в объект соответствующего контейнера.

Для сохранения и загрузки всей базы данных в файл формата xml был реализован соответствующий класс-контейнер LessCont, который содержит в себе списки объектов контейнеров сущностей.

Алгоритм метода заполнения контейнера LessCont достаточно тривиален: для каждой сущности из базы данных выбираются все записи и каждая из них трансформируется в соответствующий контейнер, который добавляется в список объектов контейнеров сущностей. После заполнения объекта LessCont он сериализуется в указанный файл xml.

Алгоритм метода записи данных из контейнера LessCont в базу данных менее очевиден. Вначале происходит инициализация коллекции следующих индексов. Затем для каждой сущности создаётся словарь индексов, в котором приводятся в соответствие исходные индексы и индексы записей базы данных. Порядок обработки сущностей таков, что ни для какой сущности, сущность, зависимая от данной, не была добавлена в базу данных. Созданные словари индексов используются для сохранения соответствия между индексами, хранимыми для сохранения связи в искомом xml файле и индексами добавляемых элементов.

## Дополнение классов сущностей

Для более эффективного использования объектов типа DataGridView, для каждой сущности была переопределена функция ToString. Данное решение позволяет отображать в таблице информацию о сущности в сущности, зависимой от данной.

# Тестирование и отладка приложения. Подготовка документации

## Тестирование

Разработанное программное обеспечение было протестировано с применением метода «Чёрного ящика»[10]. Документация по проведённому тестированию находится в приложении М.

## Руководство пользователя

Для разработанного программного обеспечения было разработано руководство пользователя в соответствии с РД 50‑34.698‑90 [11]. Данное руководство находится в приложении П.

## Руководство программиста

Для разработанного программного обеспечения было разработано руководство программиста в соответствии с ГОСТ 19.504–79 [12].

# Заключение

В результате курсовой работы были выполнены следующие задачи:

1. Произведён этап анализа.
   1. Выполнен анализ существующих решений задачи автоматизации учёта заказов на установку приборов учёта, что позволило выделить конкурентные преимущества разрабатываемого программного обеспечения.
   2. Разработана модель автоматизируемого бизнес-процесса при помощи нотации IDEF0.
   3. Разработаны варианты использования системы при помощи нотации диаграмм прецедентов и приведены полные описания прецедентов.
   4. Разработаны требования к разрабатываемой информационной системе и составить техническое задание.
2. Создана модель базы данных.
   1. Выполнено проектирование сущностной модели базы данных при помощи нотации ERD Crow’s foot.
   2. Выполнить перенос сущностной модели в объектную модель базы данных в формате ERMX, реализованый в ADO.Net Entity Framework.
3. Выполнено проектирование разрабатываемой информационной системы.
   1. Разработана динамическая модель информационной системы в форме диаграммы последовательностей.
   2. Разработана статическая модель информационной системы в форме диаграммы классов, сгенерированной при помощи Visual Studio.
   3. Разработан пользовательский графический интерфейс при помощи Windows Forms.
4. Выполнена разработку запросов к данным при использовании языка С# и LINQ выражений.
5. Выполнена реализацию информационно системы.
   1. Реализована информационная система в соответствии с разработанными требованиями, описанными в техническом задании на данную систему. Для реализации был использован язык C# и среда разработки Visual Studio.
   2. Произведено тестирование реализованной информационной системы при помощи технологии «Чёрного ящика».
   3. Разработана документацию по разработанной информационной системе в составе Руководство пользователя и Руководство программиста.

В ходе реализации информационной системы было выполнено закрепление и углубление знаний по дисциплинам «Программирование», «Введение в программную инженерию», «Конструирование программного обеспечения», был приобретены навыки и опыт проектирования, конструирования информационных систем.

Реализованная информационная система удовлетворяет разработанному техническому заданию и пригодна для использования в компании, занимающейся установкой приборов учёта.

В ходе разработки были выявлены некоторые улучшения, которые могли бы значительно повысить качество разработанной системы. В первую очередь, не является оптимальной работа с адресами клиентов. Использование сторонних приложений, реализующих работу с адресами и картами, могло бы решить проблему хранения данной информации. К сожалению, данное решение не являлось возможным ввиду недостаточных знаний в области веб-программирования.

Следующим важным улучшением, которое могло иметь место, было автоматизация распределения работников по заказам. В текущей версии системы данная задача должна выполняться либо при помощи стороннего приложения, либо вручную. Однако какие-либо тривиальные решения не позволят обеспечить необходимый уровень оптимизации, поскольку в распределении заказов участвует много факторов: расположение точек заказов относительно друг друга и офиса компании, пробки, время, необходимое на выполнение операции установки, время, занятое выполнением других заказов и прочих. Реализация данной системы потребует использование стороннего приложения для работы с адресами, картами и пробками, а также сложных алгоритмов расчёта и перерасчёта расписания.

Последним улучшением, которое могло иметь место является использование сети интернет и браузерных форм для принятия заказа. Данный функционал будет являться целью выполнения производственной практики.

# Библиографический список

1. Федеральный закон от 23.11.2009 N 261-ФЗ (ред. от 29.07.2017) "Об энергосбережении и о повышении энергетической эффективности и о внесении изменений в отдельные законодательные акты Российской Федерации"
2. Сайт компании «Сантех59»[Электронный ресурс] URL: https://santeh59.jimdo.com/ Дата обращения: 15.12.2017
3. Сайт компании «ООО АКВАГАРАНТ»[Электронный ресурс] URL https://аквагарант59.рф/ Дата обращения: 15.12.2017
4. Сайт компании «Городская служба муж на час» [Электронный ресурс] URL http://gorsluzba.ru/ustanovka\_schetchikov\_vody\_perm/ Дата обращения: 15.12.2017
5. Сайт компании «Санторемо» [Электронный ресурс] URL https://santoremo.ru/san\_schet.php Дата обращения: 15.12.2017
6. Sales Order Automation [Электронный ресурс]/Conexiom URL: https://conexiom.com/sales-order-automation/ Дата обращения: 15.12.2017
7. Управление заказами [Электронный ресурс]/Oracle Россия и СНГ URL: https://www.oracle.com/ru/applications/supply-chain-management/solutions/order-management.html Дата обращения: 15.12.2017
8. Сайт компании «ООО Новогор-Пермь» URL: https://www.novogor.perm.ru/ Дата обращения: 15.12.2017
9. ГОСТ 19.201–78. Техническое задание. Требования к содержанию и оформлению. М., 1978. С 78-80. (Единая система программной документации).
10. [Электронный ресурс] Плаксин М.А. Тестирование и отладка программ для профессионалов будущих и настоящих / 2-e изд. (эл.) / М: БИНОМ. Лаборатория знаний, 2013 – 167 с.
11. РД 50 34.698 90. Автоматизированные системы требования к содержанию документов. М., 1990. С 9-10.
12. ГОСТ 19.504–79. Руководство программиста. Требования к содержанию и оформлению. М., 1978. (Единая система программной документации).

# Приложение А. IDEF0 диаграмма



***Рисунок А.1. IDEF0. Бизнес-процесс принятия заказа. Нулевой уровень декомпозиции.***

 ***Рисунок А.2. IDEF0. Бизнес-процесс принятия заказа. Первый уровень декомпозиции А0.***

 ***Рисунок А.3. IDEF0. Бизнес-процесс принятия заказа. Второй уровень декомпозиции А1.***

 ***Рисунок А.4. IDEF0. Бизнес-процесс принятия заказа. Третий уровень декомпозиции А1.3.***

 ***Рисунок А.5. IDEF0. Бизнес-процесс принятия заказа. Второй уровень декомпозиции А2.*** 

***Рисунок А.6. IDEF0. Бизнес-процесс закрытия заказа. Нулевой уровень декомпозиции.***

 ***Рисунок А.7. IDEF0. Бизнес-процесс закрытия заказа. Первый уровень декомпозиции B0.***



***Рисунок А.8. IDEF0. Бизнес-процесс закрытия заказа. Второй уровень декомпозиции B3.***

# Приложение Б. Развёрнутое описание прецедентов

Название: Принятие заказа по email

Акторы: заказчик, оператор.

Описание: Заказчик отправляет по email документ в формате excel, заполненный по шаблону, в который помещается вся информация о заказе и заказчике. В ответ заказчик получает email с результатом добавления заказа.

Триггер: email заказчика

Основной поток:

|  |  |
| --- | --- |
| Действия акторов | Действия системы |
| 1. Заказчик отправляет email с заполненным документом excel (E2). | 1. Система производит предварительную проверку содержания excel документа(E1). Создаёт заказ. |
|  | 1. Система проверяет правильность списка заказанных счётчиков (E1). Помещает в заказ. |
|  | 1. Система проверяет правильность введённых данных о заказчике (E1). Помещает в заказ. |
| 1. Оператор определяет подходящий временной промежуток и помещает в него заказ(E4) | 1. Заказу присваивается временной промежуток. |
| 1. Отправляет предварительную версию заказа на email заказчика. |  |
| 1. Заказчик отправляет email c подтверждением заказа (E2). |  |
| 1. Оператор получает подтверждение от заказчика (E3). Устанавливает заказ как подтверждённый. Инцидент завершается. |  |

Альтернативные потоки:

E1: Были введены некорректные данные. Система отправляет заказчику email с описанием ошибки. Заказ удаляется. Инцидент завершается.

E2: Заказчик отправляет email с отменой заказа. Система удаляет заказ и отправляет заказчику «извинительный» email.

E3: Подтверждение от заказчика не пришло в течении определённого промежутка времени. В таком случаи заказ помечается как отменённый. Инцидент заканчивается.

Название: Принятие заказа по телефону

Акторы: заказчик, оператор

Описание: Заказчик звонит на телефон компании, ему отвечает оператор. Заказчик изъявляет желание установить счётчики и выбирает какие именно. Оператор запрашивает у заказчика его контактную информацию и паспортные данные, если он заказывает как частное лицо или контактную информацию с информацией о предприятии, если он заказывает как юридическое лицо. Затем оператор выбирает свободное для выполнения заказа и удобные для заказчика дату и время прихода мастера. В конце информация сверяется и звонок заканчивается.

Триггер: Звонок заказчика

Основной поток:

|  |  |
| --- | --- |
| Действия акторов | Отклик системы |
| 1. Заказчик выбирает счётчики | 1. Формируется заказ на счётчики. |
| 1. После завершения перечисления счётчиков, оператор информирует систему. | 1. Сообщает оператору состав заказа и его общую стоимость. |
| 1. Оператор сообщает сумму заказчику (Е1). Оператор запрашивает тип лица заказчика. Если заказчик частное лицо, то выполняется подпоток S1 иначе подпоток S2. Начинается прецедент «Проверка данных о заказчике». | 1. Данные о заказчике добавляются в заказ. |
| 1. Оператор запрашивает у заказчика удобное время для работ мастера и открывает в системе расписание работ. | 1. Предоставляет оператору список доступных временных интервалов для выполнения заказа. |
| 1. Оператор предлагает временной интервал (Е1, E2). Заказчик соглашается с выбранным временным интервалом. Оператор добавляет данный временной интервал в заказ | 1. Временной интервал добавляется в заказ(Е3). |
| 1. Оператор подтверждает полученный заказ у заказчика (E1,E2,Е4,E5). Оператор подтверждает заказ в системе. | 1. Заказ принимает статус подтверждённого. |
| 1. Звонок завершается. |  |

Подпотоки:

S1: Заказчик – частное лицо. Оператор запрашивает контактную информацию и паспортные данные заказчика. Данные заносятся в систему. Инцидент продолжается.

S2: Заказчик – юридическое лицо. Оператор запрашивает информацию о предприятии и контактную информацию заказчика. Данные заносятся в систему. Инцидент продолжается.

Альтернативные потоки:

Е1: Заказчик отменяет заказ. Звонок завершается. Оператор отменяет заказ в системе. Прецедент завершается.

Е2: Заказчик не соглашается с выбранным временным интервалам. Происходит переход к шагу 7 основного потока.

Е3: Временной промежуток оказывается уже занят. Система выдаёт соответствующее сообщение. Переход к шагу 7 основного потока.

E4: Были выбраны неверные счётчики. Оператор запрашивает список требуемых счётчиков у заказчика и вручную редактирует заказ в системе. Переход к шагу 11 основного потока.

E5: Были введены неверные данные о заказчике. Оператор запрашивает в соответствии с S1 и S2 и вручную изменяет заказ. Начинается прецедент «Проверка данных о заказчике». Если результат положительный, переход к шагу 11 основного потока, иначе данный поток начинается сначала.

Название: Проверить данные о заказчике.

Акторы: Оператор

Описание: Оператор инициализирует проверку и вводит в систему информацию о заказчике. Система либо выдаёт положительный результат, либо ошибку и тип ошибки.

Триггер: инициализация проверки оператором.

Основной поток:

|  |  |
| --- | --- |
| Действия акторов | Отклик системы |
| 1. Оператор вводит информацию о заказчике. | 1. Система проверяет информацию о заказчике (E1-E7). Система выводит результат проверки. Прецедент завершается. |

Альтернативные потоки:

E1. Было введено некорректное ФИО, соответствующая ошибка записывается в результат. Прецедент продолжается.

E2. Было введены некорректные паспортные данные, соответствующая ошибка записывается в результат. Прецедент продолжается.

E3. Было введено неверное название организации, соответствующая ошибка записывается в результат. Прецедент продолжается.

Е4. Было введено неверный ИНН организации, соответствующая ошибка записывается в результат. Прецедент продолжается.

Е5. Название организации и ИНН были введены верно, но не соответствуют друг другу, соответствующая ошибка записывается в результат. Прецедент продолжается.

Е6. Был введён некорректный адрес, соответствующая ошибка выводится в результат. Прецедент продолжается.

Е7. Был введён некорректный номер телефона, соответствующая ошибка выводится в результат. Прецедент продолжается.

Название: Закрыть заказ

Акторы: Мастер, Оператор

Описание: После выезда на заказ, мастер при помощи оператора регистрирует результаты заказов.

Триггер: Обращение мастера к оператору

Основной поток:

|  |  |
| --- | --- |
| Действия акторов | Отклик системы |
| 1. Мастер обращается к оператору. Оператор регистрирует время возвращения мастера в систему | 1. Система возвращает обратную связь после регистрации времени возвращения мастера. |
| 1. Оператор вводит номер заказа, с которого приехал мастер. | 1. Система возвращает информацию о данном заказе. |
| 1. Если заказ был отменён, выполняется дополнительный поток S1. Выполняется прецедент «Ввести данные о счётчиках». Оператор закрывает заказ. | 1. Система возвращает сообщение о закрытом заказе. Прецедент завершается. |

Дополнительные потоки:

S1: Заказ был отменён. Оператор удаляет заказ. Система выдаёт сообщение о удалении заказа. Прецедент завершается.

Название:

Ввести данные о счётчиках

Акторы: Оператор, Мастер

Описание: оператор вносит в заказ информацию об установленных счётчиках.

Основной поток:

|  |  |
| --- | --- |
| Действия акторов | Отклик системы |
| 1. Оператор спрашивает у мастера количество установленных счётчиков и отправляет данное количество системе |  |
| 1. Оператор вводит тип счётчика, модель, серийный номер и инвентарный номер. (E1) | 1. Система принимает информацию о счётчике (E2) и добавляет данную информацию в заказ. Если не все счётчики были введены, переход на шаг 2. |
| 1. Оператор вместе с мастером проверяют информацию о счётчиках в заказе и подтверждают её (E3). | 1. Система возвращает сообщение, что информация о счётчиках была подтверждена. Прецедент завершается. |

Альтернативные потоки:

E1: Мастер и оператор хотят досрочно завершить ввод информации о счётчиках. Система выдаёт диалоговое окно с вариантами «Сохранить изменения» и «Не сохранять». Если выбран пункт «Сохранить изменения», то происходит переход на шаг 4, иначе система удаляет введённую информацию о счётчиках и прецедент завершается.

Е2: Была обнаружена ошибка в введённых оператором данных. Система выводит сообщение с описанием ошибки, происходит переход на шаг 2.

Название: отмена заказа по телефону

Акторы: заказчик, оператор

Описание: Заказчик совершает звонок, сообщает о намерении отменить заказ, сообщает оператору номер заказа и личные данные. Оператор осуществляет отмену заказа.

Основной поток:

| Действия акторов | Отклик системы |
| --- | --- |
| 1. Заказчик совершает звонок, сообщает о намерении отменить заказ и сообщает его номер. Оператор выполняет поиск в системе по номеру заказа. | 1. Система находит заказ(E1) и выдаёт информацию о заказе |
| 1. Оператор запрашивает личные данные заказчика для его идентификации (Е2). Выполняет отмену заказа. | 1. Система отменяет заказ. |
| 1. Звонок завершается, прецедент завершён. |  |

Альтернативные потоки:

Е1: Система не находит заказ по номеру. Оператор переспрашивает номер заказа и повторяет поиск. Если система находит заказ, переход к шагу 3. Иначе оператор запрашивает личные данные и выполняет поиск по ним. Если система находит заказ, оператор выполняет отмену заказа, переход на шаг 4, иначе оператор сообщает заказчику, что заказ не найден, предлагает иные пути решения проблемы, заканчивает звонок и прецедент завершается.

Е2: Информация найденного заказа не соответствует информации, переданной заказчиком. Оператор просит повторить номер заказа. Если предшествующий номер заказа был введён неверно, то оператор выполняет поиск в системе по новому номеру заказа и переход на шаг 2. Иначе оператор производит поиск в системе по личным данным. Если заказ найден, оператор отменяет заказ и переход к шагу 4, иначе оператор сообщает заказчику, что заказ не найден, предлагает иные пути решения проблемы, заканчивает звонок и прецедент завершается.

Название: выполнить анализ.

Акторы: аналитик.

Описание: Аналитик выбирает представляемые данные и выводит их в подходящем для него формате, на основании которых он проводит анализ и делает обоснованное управленческое решение.

Основной поток:

|  |  |
| --- | --- |
| Действия акторов | Отклик системы |
| 1. Аналитик выбирает запрос при помощи готовых запросов или строит запрос при помощи конструктора запросов. | 1. Предоставляет данные из базы данных согласно отправленному запросу. |
| 1. Если аналитик выводит данные в формате Excel или XML, то начинается дополнительный поток S1. Иначе аналитик представляет данные в таблице и графиках напрямую в программе, после чего выполняет анализ данных. Инцидент завершается |  |

Дополнительные потоки:

S1: Аналитик выбирает соответствующую опцию. Если выбран вариант XML – создаётся соответствующий файл в формате XML и инцидент завершается. Иначе открывается окно программы Microsoft Office Excel, на листе открытого документа находятся данные из базы данных согласно запросу. Аналитик использует возможности программы Microsoft Office Excel для представления данных, анализа и сохранения результата. Инцидент завершается.

# Приложение В. Техническое задание.

# Введение

Наименование программы – «Автоматизированная система заказов на приборы учёта». Программа предназначена для применения в отделе продаж заказчика.

# Основания для разработки

Основанием для разработки является рабочий учебный план для второго курса направления «09.03.04. Программная инженерия» образовательная программа «Программная инженерия» факультета экономики, менеджмента и бизнес-информатики НИУ ВШЭ-Пермь на 2018/2019 учебный год, утверждённый 27 декабря 2017г. первым проректором В.В. Радаевым.

# Назначение разработки

Программа предназначена для помощи сотрудникам фирмы, получающим заказ по телефону: упрощение ввода контактной информации, поиск подходящего времени, помощь с предварительным выбором счётчиков, для автоматического оформления заказа при помощи заполненного по шаблону файла, а также для регистрации результата заказа: успешен ли заказ, если успешен, то какие именно счётчики были установлены.

# Требования к программе

# Требования к функциональным характеристикам

| Функция | Ввод | Обработка | Вывод |
| --- | --- | --- | --- |
| 1. Оформление заказа. | PhoneNumber, FIO,  Passport,  CompanyName  INN,  Address,  House,  Street,  City – строки  OrderEntries – список заказных позиций | PhoneNumber – телефонный номер заказчика,  FIO – ФИО частного лица или название организации, Passport – номер паспорта при частном заказчике, INN – ИНН организации заказчика, CompanyName – название организации  City, Street, House, Address – адрес для установки счётчика,  ClientData – совокупность представленных выше данных, которая отправляется и хранится в БД | ClientData - данные о заказе |
| 1. Оформление заказа по файлу. | File – файл формата .xlsx | ClientData – совокупность данных о заказе, отправляемая и хранимая в БД, аналогичная выводу функции (1), но составляемая на основе файла File, оформленного в соответствии с шаблоном | ClientData – данные о заказе |
| 1. Изменение данных о заказе. |  |  |  |
| * 1. Изменение имени | ID – номер  Name – строка | ClientData – изменённые данные о клиенте, взятые из БД и имеющие идентификационный номер ID, имя изменённое на Name, а остальные данные оставленные без изменения, и отправленные обратно в БД на место ID | ClientData – данные о клиенте |
| * 1. Изменение телефона | ID – номер  PhoneNumber – строка | ClientData – изменённые данные о клиенте, взятые из БД и имеющие идентификационный номер ID, номер телефона изменённое на PhoneNumber, а остальные данные оставленные без изменения, и отправленные обратно в БД на место ID | ClientData – данные о клиенте |
| * 1. Изменение адреса | ID – номер  Address – строка | ClientData – изменённые данные о клиенте, взятые из БД и имеющие идентификационный номер ID, адрес изменённый на Address, а остальные данные оставленные без изменения, и отправленные обратно в БД на место ID | ClientData – данные о клиенте |
| * 1. Изменение номера паспорта | ID – номер  Passport – строка | ClientData – изменённые данные о клиенте, взятые из БД и имеющие идентификационный номер ID, номер паспорта изменённый на Passport, остальные данные оставленные без изменения, и отправленные обратно в БД на место ID | ClientData – данные о клиенте |
| * 1. Изменение ИНН организации | ID – номер  INN – строка | ClientData – изменённые данные о клиенте, взятые из БД и имеющие идентификационный номер ID, ИНН организации изменённое на INN, а остальные данные оставленные без изменения, и отправленные обратно в БД на место ID | ClientData – данные о клиенте |
| * 1. Изменение удобного времени | ID – номер  TimeStart, TimeEnd - время | ClientData – изменённые данные о клиенте, взятые из БД и имеющие идентификационный номер ID,время начала и конца удобного времени для заказа установленные на TimeStart и TimeEnd соответственно, а остальные данные оставленные без изменения, и отправленные обратно в БД на место ID | ClientData – данные о клиенте |
| 1. Составление отчёта по запросу, составленному при помощи конструктора в Excel | Нажатия на элементы управления | File.xlsx – файл, содержащий информацию в соответствии с построенным запросом. | File.xlsx |
| 1. Составление отчёта по запросу, составленному при помощи конструктора в Xml | Нажатия на элементы управления | File.xml – файл, содержащий информацию в соответствии с построенным запросом. | File.xml |

# Требования к надёжности

# Среднее время между отказами

Требования не предъявляются.

# Время восстановления после отказа

Время восстановления после отказа со стороны клиентского приложения не должно превышать время, необходимое для повторного запуска приложения. Время восстановления после отказа со стороны базы данных также не должно превышать время, необходимое для перезапуска базы данных.

# Потеря данных

В результате отказа со стороны клиентского приложения возможна потеря данных о заказе, который ещё не был отправлен на базу данных. В результате отказа со стороны базы данных потеря данных возможна, таким образам требуется производить ежедневное сохранение копии данных базы данных.

# Недопустимая потеря данных

В результате отказа недопустима потеря исполняемых данных программы как со стороны клиентского приложения, так со стороны базы данных.

# Требования к условиям эксплуатации

# Климатические условия эксплуатации

Программа должна выполняться на компьютере, находящемся в климатических условиях, удовлетворяющих требованиям к его функционированию.

# Требования к видам обслуживания

Программа требует ежедневного сохранения копии состояния базы данных, выполняемое системным администратором.

# Требования к численности и квалификации персонала

Минимальное количество персонала, необходимое для работы программы должно состовлять не менее 2 штатных единиц – системный администратор и пользователь программы – оператор.

Системный администратор должен иметь высшее профильное образование, сертификаты компании-производителя операционной системы. В образовательной программе полученного высшего образования должна быть дисциплина, рассматривающая работу с базами данных SQL, либо требуется сертификат о прохождении курсов на данную тему. В перечень задач, выполняемых системным администратором, должны входить:

* задача поддержания работоспособности технических средств;
* задачи установки (инсталляции) и поддержания работоспособности системных программных средств – операционной системы;
* задача установки (инсталляции) программы;
* задачи установки и сопровождения базы данных.

Пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

Персонал должен быть аттестован на II квалификационную группу по электробезопасности (для работы с конторским оборудованием).

# Требования к составу и параметрам технических средств

В состав технических средств должен входить персональный компьютер, включающий в себя:

* Процессор x64: AMD Opteron, AMD Athlon 64, Intel Xeon с поддержкой Intel EM64T, Intel Pentium IV с поддержкой EM64T. с тактовой частотой не ниже 1,6 ГГЦ (рекомендуется 2,0ГГЦ);
* 1 ГБ оперативной памяти (рекомендуется 4 ГБ и более);
* Жёсткий диск – 30 ГБ свободного места на диске с файловой системой NTFS;
* Монитор с разрешением 800x600 или более высоким

# Требования к информационной и программной совместимости

# Требования к информационным структурам и методам решения

При оформлении заказа по файлу, он должен представлять из себя оформленный по шаблону .xls или .xlsx документ.

# Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть реализованы на языке C#. В качестве интегрированной среды разработки программы должна быть использована среда Microsoft Visual Studio 2017.

# Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 7, 8, 8.1 или 10. Допускается использование любого стабильного пакета обновлений перечисленных выше операционных систем.

# Требования к программной документации

В состав программной документации должны входить:

* Техническое задание;
* Программа и методика испытаний;
* Руководство системного программиста;
* Руководство пользователя;
* Ведомость эксплуатационных документов.

# Требования к технико-экономическим показателям

# Ориентировочная экономическая эффективность

Стоимость российского аналога «1С:Управление торговлей» составит 6700 рублей. Стоимость разработки составит 0 рублей. Таким образом, экономические преимущества составят 6700 рублей.

# **Предполагаемая годовая потребность**

365 запусков программы в год.

# Стадии и этапы разработки

# Стадии разработки

Разработка должна быть проведена в три стадии:

* техническое задание;
* технический (и рабочий) проекты;
* внедрение.

# Этапы разработки

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

* разработка программы;
* разработка программной документации;
* испытания программы.

На стадии «Внедрение» должен быть выполнен этап разработки «Подготовка и передача программы».

# Содержание работ по этапам

На этапе разработки тех. задания должны быть выполнены перечисленные ниже работы:

* постановка задачи;
* определение и уточнение требований к техническим средствам;
* определение требований к программе;
* определение стадий, этапов и сроков разработки программы и документации на нее;
* выбор языков программирования;
* согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

* разработка, согласование и утверждение программы и методики испытаний;
* проведение приемо-сдаточных испытаний;
* корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах заказчика.

# Порядок контроля и сдачи

# Виды испытаний

Приемосдаточные испытания программы должны проводиться на объекте заказчика согласно разработанной (не позднее сдачи курсовой) исполнителем и согласованной заказчиком «Программы и методики испытаний» (не позднее сдачи курсовой). Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний. За тем следует защита курсовой работы перед комиссией, которая определяет итоговую оценку за данную работу.

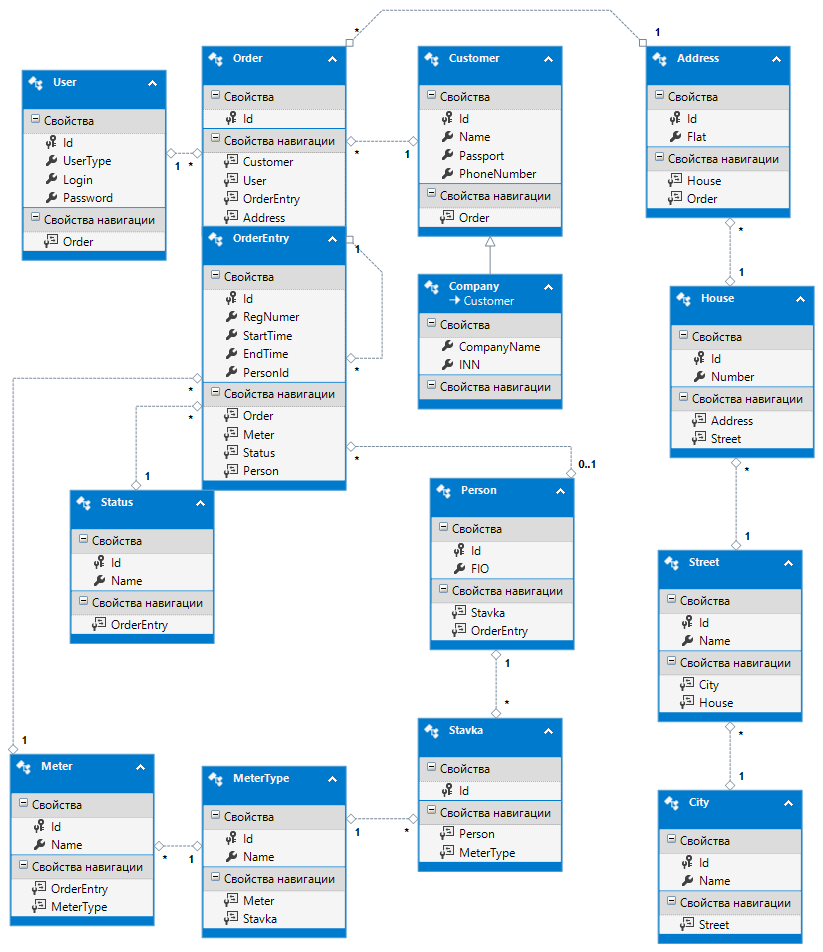
# Общие требования к приемке работы

На основании протокола испытаний исполнитель совместно с заказчиком подписывают акт приема -сдачи программы в эксплуатацию.

# Приложение Г. ERD диаграмма

***Рисунок Г.1. ERD диаграмма в нотации Crow’s foot***

# Приложение Д. ORD диаграмма.



***Рисунок Д.1. ORD диаграмма ADO.Net Entity Framework.***

# Приложение Е. Диаграмма последовательностей



***Рисунок Е.1.Диаграмма прецедента Заказ по телефону***



***Рисунок Е.2.Диаграмма прецедента Отмена заказа***

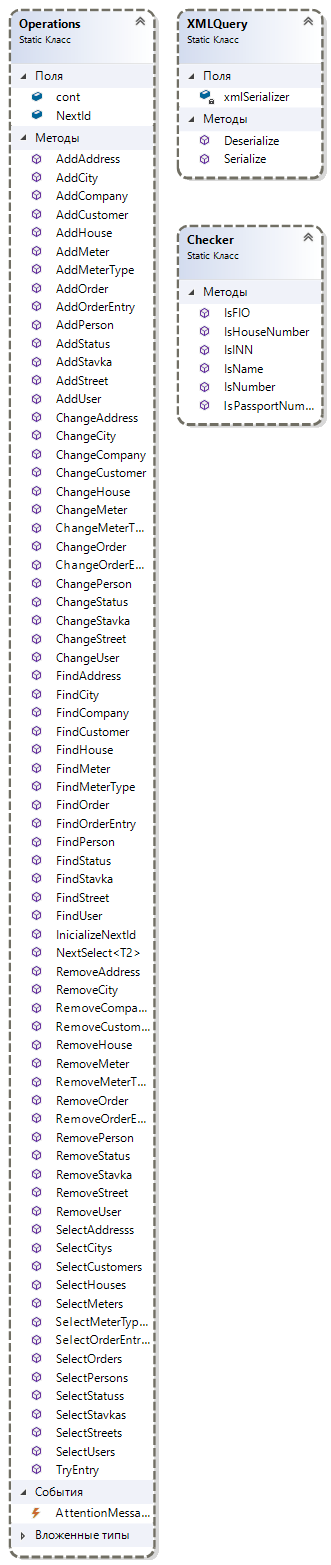


***Рисунок Е.3.Диаграмма прецедента Закрытие заказа.***

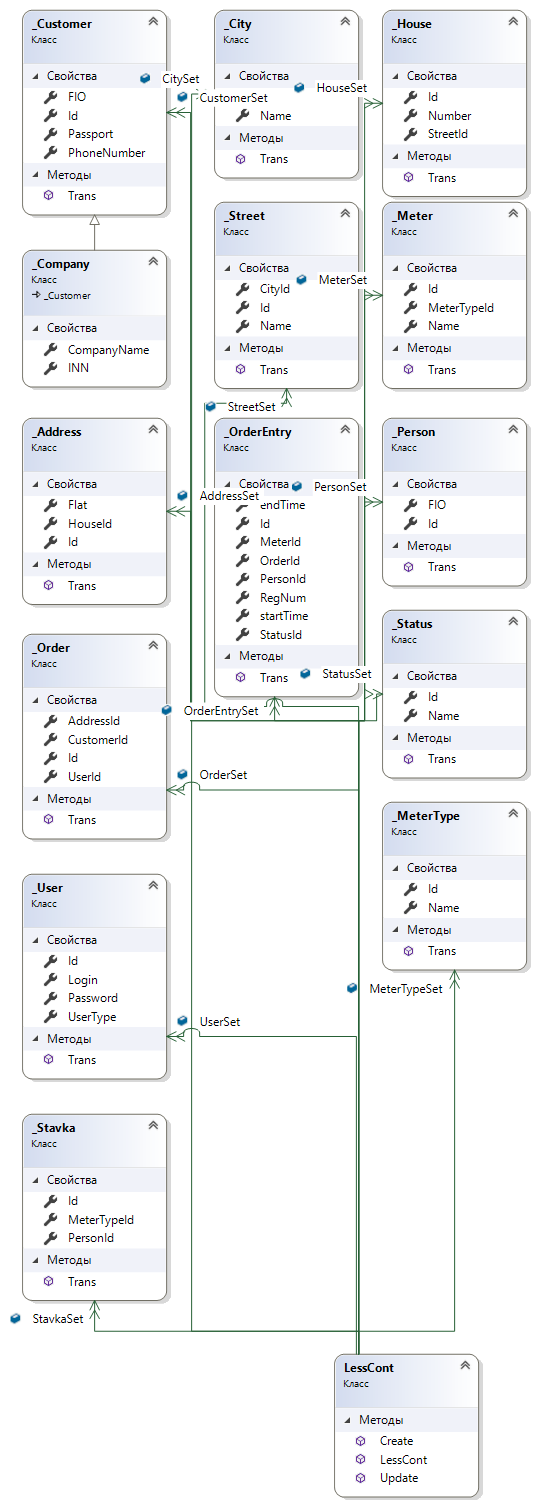


***Рисунок Е.4.Диаграмма прецедента Заказ по email.***

# Приложение Ж. Диаграмма классов



***Рисунок Ж.1.Диаграммаклассов. Классы основных операций***



***Рисунок Ж.2.Диаграммаклассов. Классы оболочек для XML сериализации.***

# Приложение И. Статические запросы к сущности User

#region Users

/// <summary>

/// Функция добавления нового пользователя

/// </summary>

/// <param name="userType">Тип нового пользователя</param>

/// <param name="Login">Логин пользователя</param>

/// <param name="Password">Пароль пользователя</param>

/// <returns>Результат добавления</returns>

public static bool AddUser(UserType userType, string Login, string Password,out string Res,bool save=true)

{

try

{

if ((from u in cont.UserSet where u.Login==Login select u).Count() > 0)

{

Res = "Пользователь с данным именем уже существует";

return false;

}

User user = new User

{

Login = Login,

Password = Password,

UserType = userType

};

cont.UserSet.Add(user);

if (save) cont.SaveChanges();

Res = "Пользователь " + Login + " успешно добавлен";

return true;

}

catch (Exception e)

{

Res = e.Message;

return false;

}

}

/// <summary>

/// Функция изменения пользователя

/// </summary>

/// <param name="Id">Идентификационный номер пользователя</param>

/// <param name="userType">Тип пользователя</param>

/// <param name="Login">Имя пользователя</param>

/// <param name="Password">Пароль</param>

/// <param name="Res">Сообщение результата изменения</param>

/// <returns>Результат изменения</returns>

public static bool ChangeUser(int Id, UserType userType, string Login, string Password, out string Res, bool save=true)

{

if ((from u in cont.UserSet where u.Login == Login&&u.Id!=Id select u).Any())

{

Res = "Пользователь с данным именем уже существует";

return false;

}

var a = (from u in cont.UserSet where u.Id==Id select u).FirstOrDefault();

if (a == null)

{

Res = "Нет пользователя с таким идентификационным номером";

return false;

}

a.Login = Login;

a.Password = Password;

a.UserType = userType;

if (save) cont.SaveChanges();

Res = "Изменение успешно";

return true;

}

/// <summary>

/// Функция удаления пользователя

/// </summary>

/// <param name="id">Идентификационный номер пользователя</param>

/// <param name="Res">Сообщение результата удаления</param>

/// <returns>Результат удаления</returns>

public static bool RemoveUser(int id, out string Res, bool save=true, bool check = true)

{

try

{

var u = cont.UserSet.Find(id);

if (u == null)

{

Res = "Нет пользователя с данным идентификационным номером";

return false;

}

if (u.Order.Count == 0 ||!check||AttentionMessage("Вы действительно хотите удалить пользователя" +

u+ "?\n"+u.Order.Count+" заказов а также связанные с ними объекты будут удалены."))

{

cont.UserSet.Remove(u);

if (save) cont.SaveChanges();

Res = "Пользователь " + u.Login + " успешно удалён";

return true;

}else

{

Res = "Удаление отменено";

return false;

}

}

catch(Exception e)

{

Res = e.Message;

return false;

}

}

/// <summary>

/// Поиск пользователя по идентификационному номеру

/// </summary>

/// <param name="Id">Идентификационный номер</param>

/// <returns>Пользователь</returns>

public static User FindUser(int Id) => (from u in cont.UserSet where u.Id == Id select u).FirstOrDefault();

/// <summary>

/// Функция входа в систему

/// </summary>

/// <param name="Login">Логин</param>

/// <param name="Password">Пароль</param>

/// <param name="Message">Получаемое сообщение</param>

/// <returns>Пользователь, под которым происходит вход, null при ошибке входа</returns>

public static User TryEntry(string Login, string Password,out string Message)

{

try

{

Message = string.Empty;

var user = (from u in cont.UserSet

where u.Login == Login

select u).ToList();

if (user.Count == 0)

{

Message += "Такого пользователя не существует";

return null;

}

else

{

if (user.Count > 1)

Message += "Несколько пользователей имеют одинаковый логин!";

foreach (User u in user)

if (u.Password == Password)

{

//Login complete

Message += "Успешный вход :" + u;

return u;

}

Message += "Неверный пароль";

return null;

}

} catch(Exception e)

{

Message = e.Message;

return null;

}

}

#endregion Users

# Приложение К. Динамические запросы данных

public static IEnumerable<User> SelectUsers(Func<User, bool> predicate ) => (from p in cont.UserSet.Local where predicate(p) select p);

public static IEnumerable<Order> SelectOrders(Func<Order, bool> predicate ) => (from p in cont.OrderSet.Local where predicate(p) select p);

public static IEnumerable<OrderEntry> SelectOrderEntrys(Func<OrderEntry, bool> predicate ) => (from p in cont.OrderEntrySet.Local where predicate(p) select p);

public static IEnumerable<Customer> SelectCustomers(Func<Customer, bool> predicate ) => (from p in cont.CustomerSet.Local where predicate(p) select p);

public static IEnumerable<Meter> SelectMeters(Func<Meter, bool> predicate) => (from p in cont.MeterSet.Local where predicate(p) select p);

public static IEnumerable<MeterType> SelectMeterTypes(Func<MeterType, bool> predicate ) => (from p in cont.MeterTypeSet.Local where predicate(p) select p);

public static IEnumerable<Stavka> SelectStavkas(Func<Stavka, bool> predicate ) => (from p in cont.StavkaSet.Local where predicate(p) select p);

public static IEnumerable<Person> SelectPersons(Func<Person, bool> predicate ) => (from p in cont.PersonSet.Local where predicate(p) select p);

public static IEnumerable<City> SelectCitys(Func<City, bool> predicate ) => (from p in cont.CitySet.Local where predicate(p) select p);

public static IEnumerable<Street> SelectStreets(Func<Street, bool> predicate ) => (from p in cont.StreetSet.Local where predicate(p) select p);

public static IEnumerable<House> SelectHouses(Func<House, bool> predicate ) => (from p in cont.HouseSet.Local where predicate(p) select p);

public static IEnumerable<Address> SelectAddresss(Func<Address, bool> predicate ) => (from p in cont.AddressSet.Local where predicate(p) select p);

public static IEnumerable<Status> SelectStatuss(Func<Status, bool> predicate ) => (from p in cont.StatusSet.Local where predicate(p) select p);

public static IEnumerable<T2> NextSelect<T2>(Func<T2, bool> predicate , IEnumerable<T2> Prev) => (from p in Prev where predicate(p) select p);

# Приложение Л. Диаграмма компонентов



***Рисунок Л.1.Диаграмма компонентов***

# Приложение М. Тестирование приложения

***Таблица М.1. Критерии чёрного ящика компонентного тестирования.***

| Критерии | T1 | T2 | T3 | T4 | T5 | T6 | T7 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Вход в систему | | | | | |  |  |
| Входные данные | | | | | |  |  |
| Корректный логин и пароль | + |  |  |  |  |  |  |
| Неверный логин и пароль |  | + |  |  |  |  |  |
| Неверный пароль с корректным логином |  |  | + |  |  |  |  |
| Логин и пароль - пустые строки |  |  |  | + |  |  |  |
| Пароль - пустая строка |  |  |  |  | + |  |  |
| Выходные данные | | | | | |  |  |
| Успешный вход | + |  |  |  |  |  |  |
| Введите логин и пароль |  |  |  | + |  |  |  |
| Неверный пароль |  |  | + |  | + |  |  |
| Такого пользователя не существует |  | + |  |  |  |  |  |
| Добавление пользователя | | | | | | | |
| Входные данные | | | | | | | |
| Логин - пустая строка | + |  |  |  |  |  |  |
| Пароль - пустая строка |  | + |  |  |  |  |  |
| Логин и пароль - пустые строки |  |  | + |  |  |  |  |
| Логин, уже имеющийся в базе данных |  |  |  | + |  |  |  |
| Логин не имеющийся в базе данных |  |  |  |  | + | + | + |
| Тип пользователя: Администратор |  |  |  |  | + |  |  |
| Тип пользователя: Оператор |  |  |  |  |  | + |  |
| Тип пользователя: Аналитик |  |  |  |  |  |  | + |
| Выходные данные | | | | | | | |
| Логин не может быть пустой строкой | + |  | + |  |  |  |  |
| Пароль не можеть быть пустой строкой |  | + |  |  |  |  |  |
| Пользователь с данным именем уже существует |  |  |  | + |  |  |  |
| Пользователь успешно добавлен |  |  |  |  | + | + | + |
| Изменение пользователя | | | | | |  |  |
| Входные данные | | | | | |  |  |
| Логин - пустая строка |  |  | + |  |  |  |  |
| Пароль - пустая строка |  | + |  |  |  |  |  |
| Логин и пароль - пустые строки | + |  |  |  |  |  |  |
| Существует пользователь, отличный от данного, с данным логином |  |  |  | + |  |  |  |
| Корректные логин и пароль |  |  |  |  | + |  |  |
| Выходные данные | | | | | |  |  |
| Логин не может быть пустой строкой | + |  | + |  |  |  |  |
| Пароль не может быть пустой строкой |  | + |  |  |  |  |  |
| Пользователь с данным именем уже существует |  |  |  | + |  |  |  |
| Успешное изменение пользователя |  |  |  |  | + |  |  |
| Добавление типа счётчика | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Название - пустая строка | + |  |  |  |  |  |  |
| Название содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректное название |  |  | + |  |  |  |  |
| Название совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное добавление |  |  | + |  |  |  |  |
| Уже есть тип приборов учёта с данным названием |  |  |  | + |  |  |  |
| Изменение типа счётчика | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Название - пустая строка | + |  |  |  |  |  |  |
| Название содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректное название |  |  |  | + |  |  |  |
| Название совпадает с другим, уже имеющимся в базе данных, элементом |  |  | + |  |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное изменение |  |  |  | + |  |  |  |
| Уже есть тип приборов учёта с данным названием |  |  | + |  |  |  |  |
| Добавление счётчика | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Название - пустая строка | + |  |  |  |  |  |  |
| Название содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректное название |  |  |  | + |  |  |  |
| Название совпадает с другим, уже имеющимся в базе данных, элементом |  |  | + |  |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное добавление |  |  |  | + |  |  |  |
| Уже есть прибор учёта с данным названием |  |  | + |  |  |  |  |
| Изменение типа счётчика | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Название - пустая строка | + |  |  |  |  |  |  |
| Название содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректное название |  |  |  | + |  |  |  |
| Название совпадает с другим, уже имеющимся в базе данных, элементом |  |  | + |  |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное изменение |  |  |  | + |  |  |  |
| Уже есть прибор учёта с данным названием |  |  | + |  |  |  |  |
| Добавление работника | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| ФИО - пустая строка | + |  |  |  |  |  |  |
| ФИО имеет неверный формат |  | + |  |  |  |  |  |
| Корректное ФИО |  |  | + |  |  |  |  |
| ФИО совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное добавление |  |  | + |  |  |  |  |
| Уже есть работник с данным ФИО |  |  |  | + |  |  |  |
| Изменение работника | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| ФИО - пустая строка | + |  |  |  |  |  |  |
| ФИО имеет неверный формат |  | + |  |  |  |  |  |
| Корректное ФИО |  |  | + |  |  |  |  |
| ФИО совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное изменение |  |  | + |  |  |  |  |
| Уже есть работник с данным ФИО |  |  |  | + |  |  |  |
| Добавление города | | | | |  |  |  |
| Критерии | Т1 | Т2 | Т3 | T4 |  |  |  |
| Входные данные | | | | |  |  |  |
| Название - пустая строка | + |  |  |  |  |  |  |
| Название содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректное название |  |  | + |  |  |  |  |
| Название совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное добавление |  |  | + |  |  |  |  |
| Уже есть город с данным названием |  |  |  | + |  |  |  |
| Изменение города | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Название - пустая строка | + |  |  |  |  |  |  |
| Название содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректное название |  |  | + |  |  |  |  |
| Название совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное изменение |  |  | + |  |  |  |  |
| Уже есть город с данным названием |  |  |  | + |  |  |  |
| Добавление улицы | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Название - пустая строка | + |  |  |  |  |  |  |
| Название содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректное название |  |  | + |  |  |  |  |
| Название и город совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное добавление |  |  | + |  |  |  |  |
| В городе City уже есть улица Name |  |  |  | + |  |  |  |
| Изменение улицы | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Название - пустая строка | + |  |  |  |  |  |  |
| Название содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректное название |  |  | + |  |  |  |  |
| Название и город совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное изменение |  |  | + |  |  |  |  |
| В городе City уже есть улица Name |  |  |  | + |  |  |  |
| Добавление дома | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Номер - пустая строка | + |  |  |  |  |  |  |
| Номер содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректный Номер |  |  | + |  |  |  |  |
| Номер, улица и город совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное добавление |  |  | + |  |  |  |  |
| В городе City на улице Street уже есть дом номер Number |  |  |  | + |  |  |  |
| Изменение дома | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Номер - пустая строка | + |  |  |  |  |  |  |
| Номер содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректный Номер |  |  | + |  |  |  |  |
| Номер, улица и город совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное изменение |  |  | + |  |  |  |  |
| В городе City на улице Street уже есть дом номер Number |  |  |  | + |  |  |  |
| Добавление адреса | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Номер - пустая строка | + |  |  |  |  |  |  |
| Номер содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректный Номер |  |  | + |  |  |  |  |
| Номер, дом, улица и город совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное добавление |  |  | + |  |  |  |  |
| В городе City на улице Street в доме House уже есть дом номер Number |  |  |  | + |  |  |  |
| Изменение адреса | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Номер - пустая строка | + |  |  |  |  |  |  |
| Номер содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректный Номер |  |  | + |  |  |  |  |
| Номер, улица и город совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное изменение |  |  | + |  |  |  |  |
| В городе City на улице Street в доме House уже есть дом номер Number |  |  |  | + |  |  |  |
| Добавление ставки | | |  |  |  |  |  |
| Входные данные | | |  |  |  |  |  |
| Новая пара типа Тип счётчика – Работник | + |  |  |  |  |  |  |
| Данная пара типа Тип счётчика - Работник уже существует |  | + |  |  |  |  |  |
| Выходные данные | | |  |  |  |  |  |
| Успешное добавление | + |  |  |  |  |  |  |
| У данного работника уже есть данная ставка |  | + |  |  |  |  |  |
| Изменение ставки | | |  |  |  |  |  |
| Входные данные | | |  |  |  |  |  |
| Новая пара типа Тип счётчика – Работник | + |  |  |  |  |  |  |
| Данная пара типа Тип счётчика - Работник уже существует |  | + |  |  |  |  |  |
| Выходные данные | | |  |  |  |  |  |
| Успешное изменение | + |  |  |  |  |  |  |
| У данного работника уже есть данная ставка |  | + |  |  |  |  |  |
| Добавление заказчика | | | | | |  |  |
| Входные данные | | | | | |  |  |
| Неверная строка ФИО | + |  |  |  |  |  |  |
| Неверная строка телефонного номера |  | + |  |  |  |  |  |
| Неверная строка номера паспорта |  |  | + |  |  |  |  |
| Уже есть заказчик с данным номером паспорта |  |  |  |  | + |  |  |
| Корретная новая запись |  |  |  | + |  |  |  |
| Выходные данные | | | | | |  |  |
| Неверная строка ФИО | + |  |  |  |  |  |  |
| Неверная строка телефонного номера |  | + |  |  |  |  |  |
| Неверная строка номера паспорта |  |  | + |  |  |  |  |
| Уже есть заказчик с данным номером паспорта |  |  |  |  | + |  |  |
| Успешное добавление |  |  |  | + |  |  |  |
| Изменение заказчика | | | | | |  |  |
| Входные данные | | | | | |  |  |
| Неверная строка ФИО | + |  |  |  |  |  |  |
| Неверная строка телефонного номера |  | + |  |  |  |  |  |
| Неверная строка номера паспорта |  |  | + |  |  |  |  |
| Уже есть заказчик с данным номером паспорта |  |  |  |  | + |  |  |
| Корретная новая запись |  |  |  | + |  |  |  |
| Выходные данные | | | | | |  |  |
| Неверная строка ФИО | + |  |  |  |  |  |  |
| Неверная строка телефонного номера |  | + |  |  |  |  |  |
| Неверная строка номера паспорта |  |  | + |  |  |  |  |
| Уже есть заказчик с данным номером паспорта |  |  |  |  | + |  |  |
| Успешное изменение |  |  |  | + |  |  |  |
| Добавление компании | | | | | | | |
| Входные данные | | | | | | | |
| Неверная строка ФИО | + |  |  |  |  |  |  |
| Неверная строка телефонного номера |  | + |  |  |  |  |  |
| Неверная строка номера паспорта |  |  | + |  |  |  |  |
| Неверная строка ИНН |  |  |  | + |  |  |  |
| Неверная строка названия компании |  |  |  |  | + |  |  |
| Корректная новая запись |  |  |  |  |  | + |  |
| Уже есть компания с данным номером ИНН |  |  |  |  |  |  | + |
| Выходные данные | | | | | | | |
| Неверная строка ФИО | + |  |  |  |  |  |  |
| Неверная строка телефонного номера |  | + |  |  |  |  |  |
| Неверная строка номера паспорта |  |  | + |  |  |  |  |
| Неверная строка ИНН |  |  |  | + |  |  |  |
| Неверная строка названия компании |  |  |  |  | + |  |  |
| Успешное добавление |  |  |  |  |  | + |  |
| Уже есть компания с данным номером ИНН |  |  |  |  |  |  | + |
| Изменение компании | | | | | | | |
| Входные данные | | | | | | | |
| Неверная строка ФИО | + |  |  |  |  |  |  |
| Неверная строка телефонного номера |  | + |  |  |  |  |  |
| Неверная строка номера паспорта |  |  | + |  |  |  |  |
| Неверная строка ИНН |  |  |  | + |  |  |  |
| Неверная строка названия компании |  |  |  |  | + |  |  |
| Корректная новая запись |  |  |  |  |  | + |  |
| Уже есть компания с данным номером ИНН |  |  |  |  |  |  | + |
| Выходные данные | | | | | | | |
| Неверная строка ФИО | + |  |  |  |  |  |  |
| Неверная строка телефонного номера |  | + |  |  |  |  |  |
| Неверная строка номера паспорта |  |  | + |  |  |  |  |
| Неверная строка ИНН |  |  |  | + |  |  |  |
| Неверная строка названия компании |  |  |  |  | + |  |  |
| Успешное изменение |  |  |  |  |  | + |  |
| Уже есть компания с данным номером ИНН |  |  |  |  |  |  | + |
| Добавление статуса | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Название - пустая строка | + |  |  |  |  |  |  |
| Название содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректное название |  |  | + |  |  |  |  |
| Название совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное добавление |  |  | + |  |  |  |  |
| Уже есть статус заказа с данным названием |  |  |  | + |  |  |  |
| Изменение статуса | | | | |  |  |  |
| Входные данные | | | | |  |  |  |
| Название - пустая строка | + |  |  |  |  |  |  |
| Название содержит нетекстовые символы |  | + |  |  |  |  |  |
| Корректное название |  |  | + |  |  |  |  |
| Название совпадает с другим, уже имеющимся в базе данных, элементом |  |  |  | + |  |  |  |
| Выходные данные | | | | |  |  |  |
| Неверная строка названия | + | + |  |  |  |  |  |
| Успешное изменение |  |  | + |  |  |  |  |
| Уже есть статус заказа с данным названием |  |  |  | + |  |  |  |

***Таблица M.2. Протокол компонентного тестирования.***

| № теста | Входные данные | Ожидаемый результат | Реальный результат |
| --- | --- | --- | --- |
| Вход в систему | | | |
| 1 | root 123qwe | Успешный вход: root | + |
| 2 | lol lol | Такого пользователя не существует | + |
| 3 | root root | Неверный пароль | + |
| 4 |  | Введите логин и пароль | + |
| 5 | root | Неверный пароль | + |
| Добавление пользователя | | | |
| 1 | " " "123qwe" UserType.Admin | Логин не может быть пустой строкой | + |
| 2 | "login" " " UserType.Admin | Пароль не может быть пустой строкой | + |
| 3 | " " " " UserType.Admin | Логин не может быть пустой строкой | + |
| 4 | root 123 UserType.Admin | Пользователь с данным логином уже существует | + |
| 5 | admin admin UserType.Admin | Пользователь Admin успешно добавлен | + |
| 6 | op op UserType.Operator | Пользователь op успешно добавлен | + |
| 7 | an an UserType.Analitic | Пользователь an успешно добавлен | + |
| Удаление пользователя | | | |
| 1 | Выбран пользователь и выбрано удаление | Пользователь успешно удалён | + |
| Изменение пользователя | | | |
| 1 | Выбран пользователь admin и логин с паролем изменяются на " " " " | Логин не может быть пустой строкой | + |
| 2 | Выбран пользователь admin и логин с паролем изменяются на "admin" " " | Логин не может быть пустой строкой | + |
| 3 | Выбран пользователь admin и логин с паролем изменяются на " " "admin" | Пароль не может быть пустой строкой | + |
| 4 | Выбран пользователь admin и логин с паролем изменяются на "root" "admin" | Пользователь с данным именем уже существует | + |
| 5 | Выбран пользователь admin и логин с паролем изменяются на "ad1" "123" | Успешное изменение пользователя | + |
| Добавление типа счётчиков | | | |
| 1 | " " | Неверная строка названия | + |
| 2 | "/////" | Неверная строка названия | + |
| 3 | "Электрический" | Успешное добавление | + |
| 4 | "Электрический" | Уже есть тип приборов учёта с данным названием | + |
| Изменение типа счётчиков | | | |
| 1 | " " | Неверная строка названия | + |
| 2 | "/////" | Неверная строка названия | + |
| 3 | "Электрический" | Уже есть тип приборов учёта с данным названием | + |
| 4 | "Газовый" | Успешное изменение | + |
| Добавление счётчика | | | |
| 1 | " " | Неверная строка названия | + |
| 2 | "/////" | Неверная строка названия | + |
| 3 | "М1" | Успешное добавление | + |
| 4 | "M1" | Уже есть прибор учёта с даннм названием | + |
| Изменение счётчика | | | |
| 1 | " " | Неверная строка названия | + |
| 2 | "/////" | Неверная строка названия | + |
| 3 | "М2" | Успешное изменение | + |
| 4 | "M1" | Уже есть прибор учёта с даннм названием | + |
| Добавление работника | | | |
| 1 | " " | Неверная строка ФИО | + |
| 2 | "abc" | Неверная строка ФИО | + |
| 3 | Петров Пётр Петрович | Успешное добавление | + |
| 4 | Петров Пётр Петрович | Уже есть работник с данным ФИО | + |
| Изменение работника | | | |
| 1 | " " | Неверная строка ФИО | + |
| 2 | "abc" | Неверная строка ФИО | + |
| 3 | Иванов Иван Иванович | Успешное добавление | + |
| 4 | Петров Пётр Петрович | Уже есть работник с данным ФИО | + |
| Добавление города | | | |
| 1 | " " | Неверная строка названия | + |
| 2 | "/////" | Неверная строка названия | + |
| 3 | "Кунгур" | Успешное добавление | + |
| 4 | "Кунгур" | Уже есть город с даннм названием | + |
| Изменение города | | | |
| 1 | " " | Неверная строка названия | + |
| 2 | "/////" | Неверная строка названия | + |
| 3 | "Пермь" | Успешное добавление | + |
| 4 | "Кунгур" | Уже есть город с даннм названием | + |
| Добавление улицы | | | |
| 1 | " " Пермь | Неверная строка названия | + |
| 2 | "////" Пермь | Неверная строка названия | + |
| 3 | "Уинская" Пермь | Успешное добавление | + |
| 4 | "Уинская" Пермь | В городе Пермь уже есть улица Уинская | + |
| Изменение улицы | | | |
| 1 | " " Пермь | Неверная строка названия | + |
| 2 | "////" Пермь | Неверная строка названия | + |
| 3 | "Уинская" Кунгур | Успешное изменение | + |
| 4 | "Уинская" Пермь | В городе Пермь уже есть улица Уинская | + |
| Добавление дома | | | |
| 1 | " " Уинская Пермь | Неверная строка названия | + |
| 2 | "////" Уинская Пермь | Неверная строка названия | + |
| 3 | "34" Уинская Пермь | Успешное добавление | + |
| 4 | "34" Уинская Пермь | В городе Пермь на улице Уинская уже есть дом номер 34 | + |
| Изменение дома | | | |
| 1 | " " Уинская Пермь | Неверная строка номера | + |
| 2 | "////" Уинская Пермь | Неверная строка номера | + |
| 3 | "34" Уинская Кунгур | Успешное изменение | + |
| 4 | "34" Уинская Пермь | В городе Пермь на улице Уинская уже есть дом номер 34 | + |
| Добавление адреса | | | |
| 1 | " " 34 Уинская Пермь | Неверная строка номера | + |
| 2 | "////" 34 Уинская Пермь | Неверная строка номера | + |
| 3 | "413" 34 Уинская Пермь | Успешное добавление | + |
| 4 | "413" 34 Уинская Пермь | В городе Пермь на улице Уинская в доме 34 уже есть квартира 413 | + |
| Изменение адреса | | | |
| 1 | " " 34 Уинская Пермь | Неверная строка номера | + |
| 2 | "////" 34 Уинская Пермь | Неверная строка номера | + |
| 3 | "413" 34 Уинская Кунгур | Успешное изменение | + |
| 4 | "413" 34 Уинская Пермь | В городе Пермь на улице Уинская в доме 34 уже есть квартира 413 | + |
| Добавление ставки | | | |
| 1 | Иванов Иван Иванович, Газовый | Успешное добавление | + |
| 2 | Иванов Иван Иванович, Газовый | У работника Иванов Иван Иванович уже имеется данная ставка | + |
| Изменение ставки | | | |
| 1 | Иванов Иван Иванович, Электрический | Успешное изменение | + |
| 2 | Иванов Иван Иванович, Газовый | У работника Иванов Иван Иванович уже имеется данная ставка | + |
| Добавление заказчика | | | |
| 1 | ФИО - "ekfelfsef" | Неверная строка ФИО | + |
| 2 | Телефонный номер - "abc" | Неверная строка телефонного номера | + |
| 3 | Номер паспорта - "123" | Неверная строка номера паспорта | + |
| 4 | ФИО - Коковин Алексей Николаевич Номер - 26162 Номер паспорта - 5923123223 | Успешное добавление | + |
| 5 | ФИО - Коковин Алексей Николаевич Номер - 26162 Номер паспорта - 5923123223 | Уже есть заказчик с данным номером паспорта | + |
| Изменение заказчика | | | |
| 1 | ФИО - "ekfelfsef" | Неверная строка ФИО | + |
| 2 | Телефонный номер - "abc" | Неверная строка телефонного номера | + |
| 3 | Номер паспорта - "123" | Неверная строка номера паспорта | + |
| 4 | ФИО - Сидиров Сидор Сидоровичч Номер - 263212 Номер паспорта - 5985595544 | Успешное изменение | + |
| 5 | ФИО - Коковин Алексей Николаевич Номер - 26162 Номер паспорта - 5923123223 | Уже есть заказчик с данным номером паспорта | + |
| Добавление компании | | | |
| 1 | ФИО - "ekfelfsef" | Неверная строка ФИО | + |
| 2 | Телефонный номер - "abc" | Неверная строка телефонного номера | + |
| 3 | Номер паспорта - "123" | Неверная строка номера паспорта | + |
| 4 | ИНН - "123" | Неверная строка ИНН | + |
| 5 | Название компании -"abc//" | Неверное название компании | + |
| 6 | ФИО - Сидиров Сидор Сидорович Номер - 263212 Номер паспорта - 5985595544 ИНН - 1234567891 Название компании - ООО Компания | Успешное добавление | + |
| 7 | ФИО - Сидиров Сидор Сидорович Номер - 263212 Номер паспорта - 5985595544 ИНН - 1234567891 Название компании - ООО Компания | Уже есть компания с данным ИНН | + |
| Изменение компании | | | |
| 1 | ФИО - "ekfelfsef" | Неверная строка ФИО | + |
| 2 | Телефонный номер - "abc" | Неверная строка телефонного номера | + |
| 3 | Номер паспорта - "123" | Неверная строка номера паспорта | + |
| 4 | ИНН - "123" | Неверная строка ИНН | + |
| 5 | Название компании -"abc//" | Неверное название компании | + |
| 6 | ФИО - Сидиров Сидор Сидорович Номер - 263212 Номер паспорта - 5985595544 ИНН - 9876543210 Название компании - ООО Компания | Успешное изменение | + |
| 7 | ФИО - Сидиров Сидор Сидорович Номер - 263212 Номер паспорта - 5985595544 ИНН - 1234567891 Название компании - ООО Компания | Уже есть компания с данным ИНН | + |
| Добавление статуса заказной позиции | | | |
| 1 | " " | Неверная строка названия | + |
| 2 | "////" | Неверная строка названия | + |
| 3 | "Отменён" | Успешное добавление | + |
| 4 | "Отменён" | Уже есть статус с данным названием | + |
| Изменение статуса заказной позиции | | | |
| 1 | " " | Неверная строка названия | + |
| 2 | "////" | Неверная строка названия | + |
| 3 | "Принят" | Успешное изменение | + |
| 4 | "Отменён" | Уже есть статус с данным названием | + |

# Приложение П. Руководство пользователя

## Введение

## Область применения

Требования настоящего документа применяются при:

* предварительных комплексных испытаниях;
* опытной эксплуатации;
* приемочных испытаниях;
* промышленной эксплуатации.

## Краткое описание возможностей

Информационная система «Автоматизированная система заказов на приборы учёта» предназначена для оптимизации технологии получения заказов и принятия тактических и стратегических управленческих решения на основе данных о данных заказах.

Данная система предоставляет возможность работы с базой данных заказов. Для добавления заказов и получения настраиваемых запросов возможно использование Microsoft Office Excel. Для работы требуется подключение к серверу типа SQL.

## Уровень подготовки пользователя

Пользователь данной системы должен иметь опыт работы с ОС MS Windows (7,8,10), навык работы с ПО Microsoft Office Excel, а также обладать следующими знаниями:

* знать предметную область принятия заказов на установку приборов учёта

## Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю

Информационная система «Автоматизированная система заказов на приборы учёта». Руководство пользователя.

## Назначение и условия применения

Данная система предназначена для автоматизации бизнес-процесса приёма заказов, а также для созданий выборок данных, соответствующих нуждам пользователя.

Работа с данной системой возможна при наличии подключенного сервера SQL.

Работа с данной системой доступна всем пользователям, добавленным администратором с возможностями, указанными при создании.

## Подготовка к работе

## Состав и содержание дистрибутивного носителя данных

Для работы с данной системой необходимо следующее программное обеспечение:

* Microsoft Office Excel
* Microsoft SQL или подобный для установки сервера.

## Порядок загрузки данных и программ

Перед началом работы с данной системой на рабочем месте пользователя требуется выполнить следующие действия:

1. Необходимо загрузить искомые исполняемые файлы с внешнего носителя.
2. Произвести подключение к серверу SQL путём изменения строки подключения

## Порядок проверки работоспособности

Для проверки работоспособности данной системы с рабочего места требуется выполнить следующие действия:

* 1. Запустить приложение.
  2. Ввести логин и пароль.
  3. Убедиться, что вход был произведён успешно.

В случае, если приложение не запускается или вход не производится следует обратиться к системному администратору.

## Описание операций

## 4.1. Выполняемые функции и задачи

Данная система выполняет функции и задачи, представленные в таблице ниже:

|  |  |  |
| --- | --- | --- |
| Функции | Задачи | Описание |
| Обеспечивает возможность ведения отчётности по заказам | Добавление заказа | В ходе выполнения данной задачи пользователю системы предоставляется возможность работы с формами для добавления заказа и непосредственно произвести запись заказа в базу данных. |
| Добавление заказа из файла Excel | В ходе выполнения данной задачи пользователю системы предоставляется возможность внести файл Excel в программу в целях извлечения из него информации о заказе и добавления заказа в базу данных. |
| Изменение заказа | В ходе выполнения данной задачи пользователю системы предоставляется возможность работы с формами для изменения заказа и непосредственно произвести изменение записи заказа в базу данных. |
| Формирование табличных форм отчётности | В ходе выполнения данной задачи пользователю системы предоставляется возможность формирования табличного отчёта по выбранным пользователем критериям. |

## 4.2. Описание операций технологического процесса обработки данных, необходимых для выполнения задач.

Задача: «Добавление заказа».

Операция 1: Вход в систему

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к корпоративной сети.
2. База данных SQL доступна.
3. Произведена верная настройка подключения приложения к базе данных SQL.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведённые в п.3.2. настоящего документа.

Основные действия в требуемой последовательности:

1. На иконке «Система заказов» рабочего стола произвести двойной щелчок левой кнопкой мыши.
2. В открывшемся окне в поле «Логин» ввести имя пользователя, в поле «Пароль» ввести пароль пользователя. Нажать кнопку «Далее».

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 15-30 секунд.

Операция 2: Создание заказа

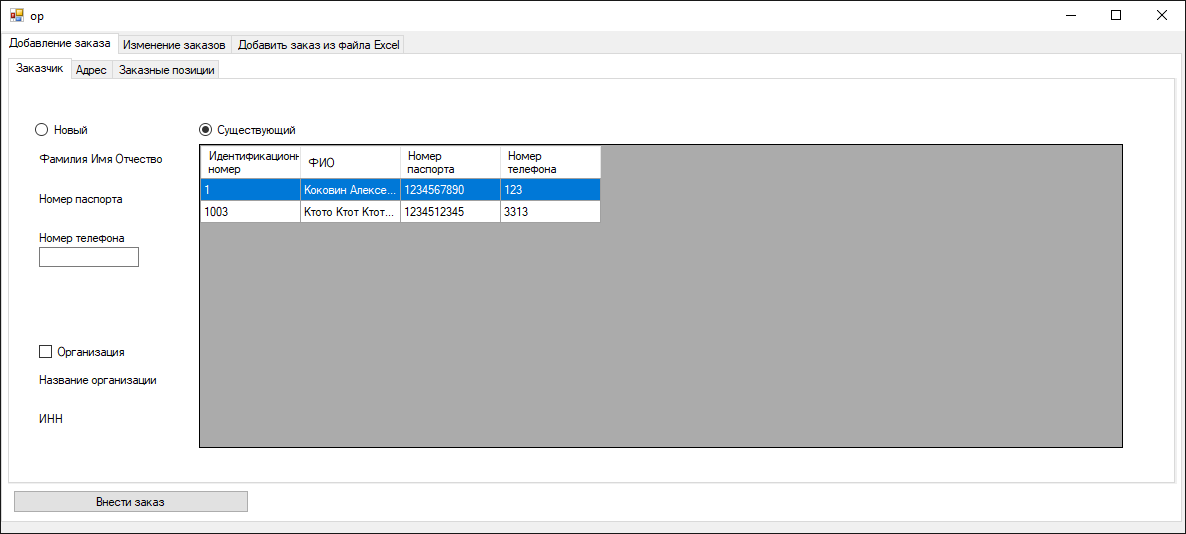
Условия, при соблюдении которых возможно выполнение операции:

Успешный вход с пользователем типа «Оператор».

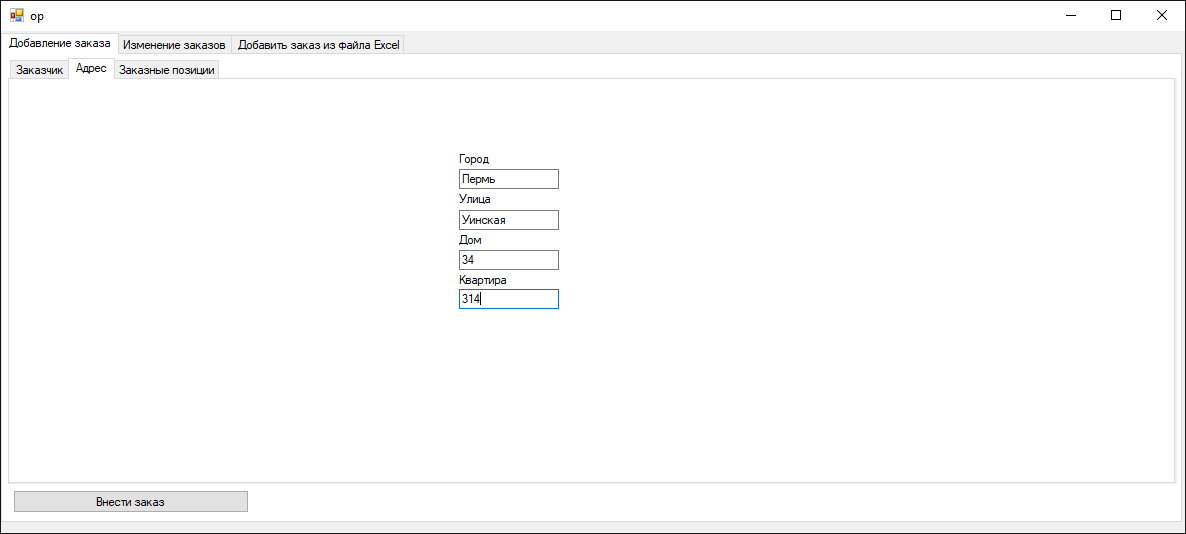
Подготовительные действия: не требуются.

Основные действия в требуемой последовательности:

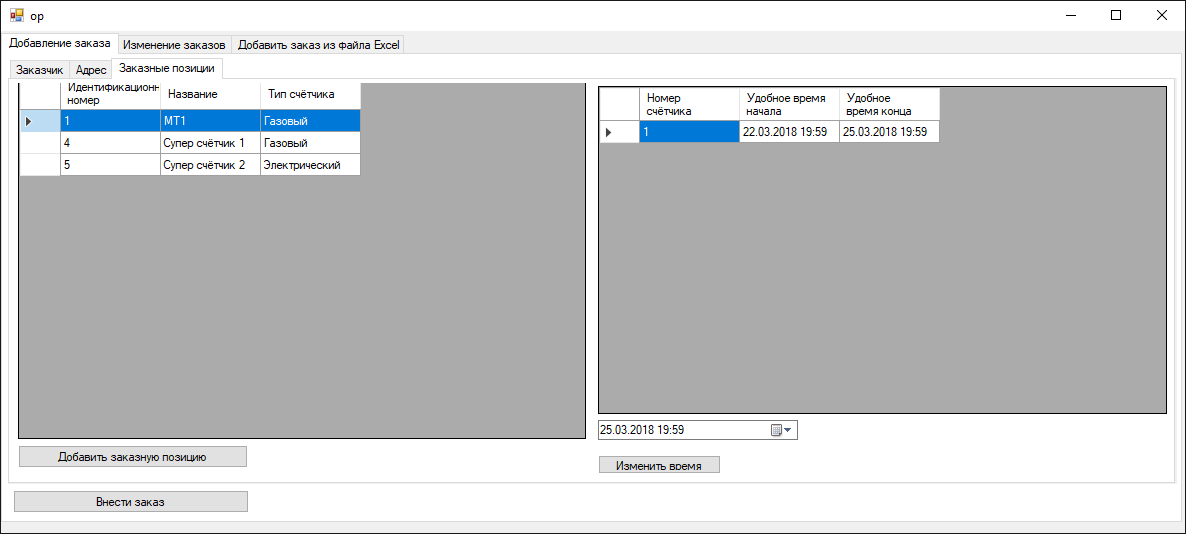
1. В появившемся окне выбрать закладку «Добавление заказа».
2. Ввести данные о заказчике или выбрать уже существующего



1. Перейти во вкладку «Адрес» и ввести адрес заказчика.



1. Перейти во вкладку «Заказные позиции» и добавить требуемые счётчики и удобное время для установки каждого из них



1. Нажать на кнопку «Внести заказ».

Заключительные действия:

После завершения работы с программой следует нажать на крестик в правом верхнем углу.

Ресурсы, расходуемые на операцию: 30-60 секунд.

Задача: «Добавление заказа из файла Excel».

Операция 1: Вход в систему

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к корпоративной сети.
2. База данных SQL доступна.
3. Произведена верная настройка подключения приложения к базе данных SQL.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведённые в п.3.2. настоящего документа.

Основные действия в требуемой последовательности:

1. На иконке «Система заказов» рабочего стола произвести двойной щелчок левой кнопкой мыши.
2. В открывшемся окне в поле «Логин» ввести имя пользователя, в поле «Пароль» ввести пароль пользователя. Нажать кнопку «Далее».

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 15-30 секунд.

Операция 2: Ввод заказа из файла Excel.

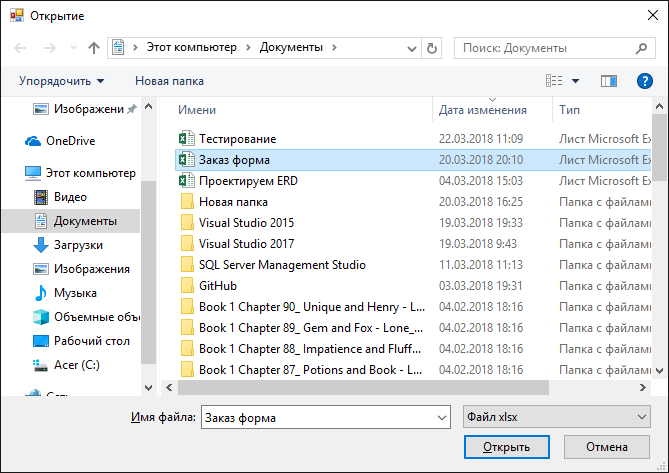
Условия, при которых возможно выполнение операции:

Успешный вход в систему с пользователем типа «Оператор».

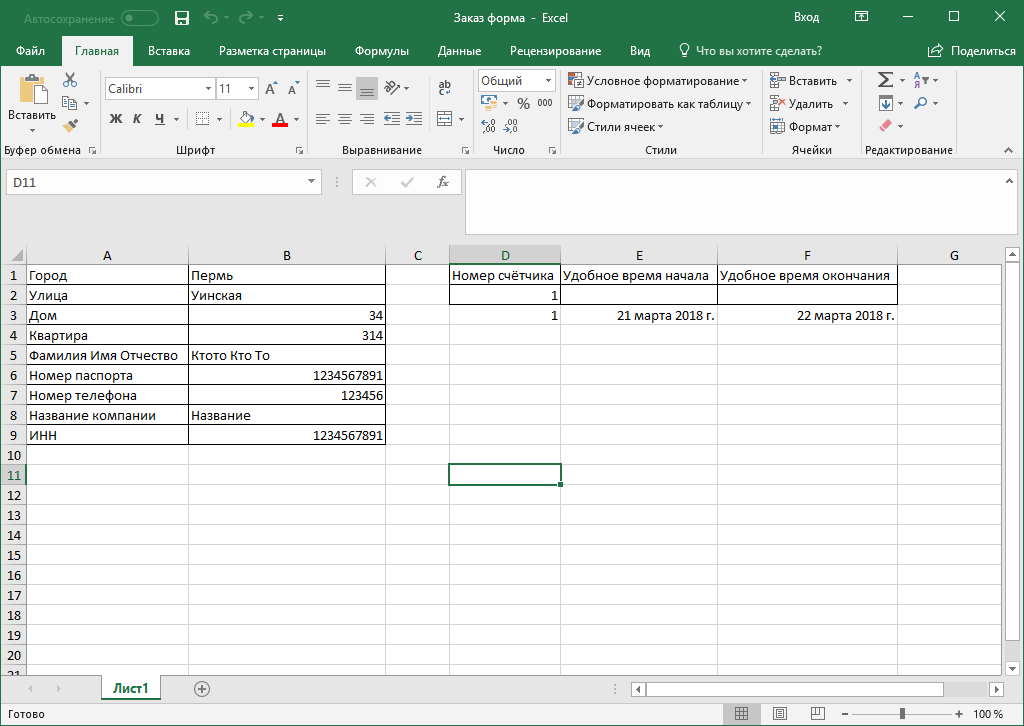
Подготовительные действия: расположить файл Excel с заказом на жёсткий диск компьютера.

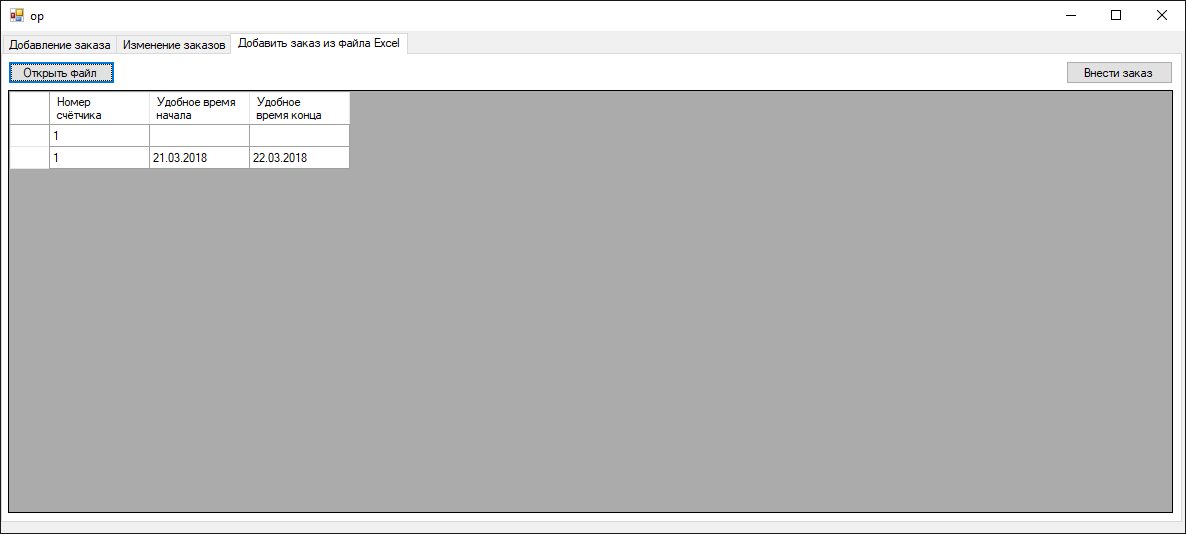
Основные действия в требуемой последовательности:

1. Выбрать вкладку «Добавить заказ из файла Excel».
2. Нажать на кнопку «Открыть файл».
3. Выбрать файл Excel с добавляемым заказом и нажать ОК.



1. Данный файл откроется в Excel, данные из файла будут переданы в программу.





1. Убедиться в правильности переноса данных и нажать кнопку «Внести заказ».

Заключительные действия:

После завершения работы с программой следует нажать на крестик в правом верхнем углу.

Ресурсы, расходуемые на операцию: 30-60 секунд.

Задача «Изменение заказа».

Операция 1: Вход в систему

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к корпоративной сети.
2. База данных SQL доступна.
3. Произведена верная настройка подключения приложения к базе данных SQL.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведённые в п.3.2. настоящего документа.

Основные действия в требуемой последовательности:

1. На иконке «Система заказов» рабочего стола произвести двойной щелчок левой кнопкой мыши.
2. В открывшемся окне в поле «Логин» ввести имя пользователя, в поле «Пароль» ввести пароль пользователя. Нажать кнопку «Далее».

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 15-30 секунд.

Операция 2: Изменение заказа

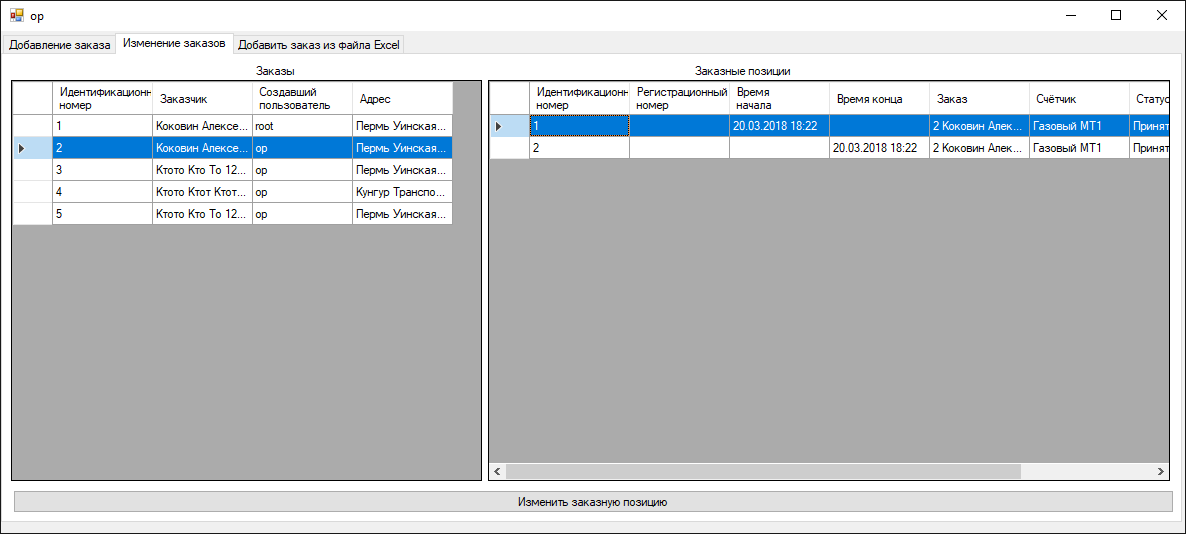
Условия, при соблюдении которых возможно выполнение операции:

Успешный вход в систему с пользователем типа «Оператор».

Подготовительные действия: не требуются.

Основные действия в требуемой последовательности.

1. Выбрать вкладку «Изменение заказов».
2. В левой таблице выбрать изменяемый заказ.
3. В правой таблице выбрать изменяемую заказную позицию.



1. Нажать на кнопку «Изменить заказную позицию».
2. Выполнить изменения.

Заключительные действия:

После завершения работы с программой следует нажать на крестик в правом верхнем углу.

Ресурсы, расходуемые на операцию:

30-60 секунд.

Задача: «Формирование табличных форм отчётности»

Операция 1: Вход в систему

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя подключен к корпоративной сети.
2. База данных SQL доступна.
3. Произведена верная настройка подключения приложения к базе данных SQL.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведённые в п.3.2. настоящего документа.

Основные действия в требуемой последовательности:

1. На иконке «Система заказов» рабочего стола произвести двойной щелчок левой кнопкой мыши.
2. В открывшемся окне в поле «Логин» ввести имя пользователя, в поле «Пароль» ввести пароль пользователя. Нажать кнопку «Далее».

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 15-30 секунд.

Операция 2: Добавление запросов

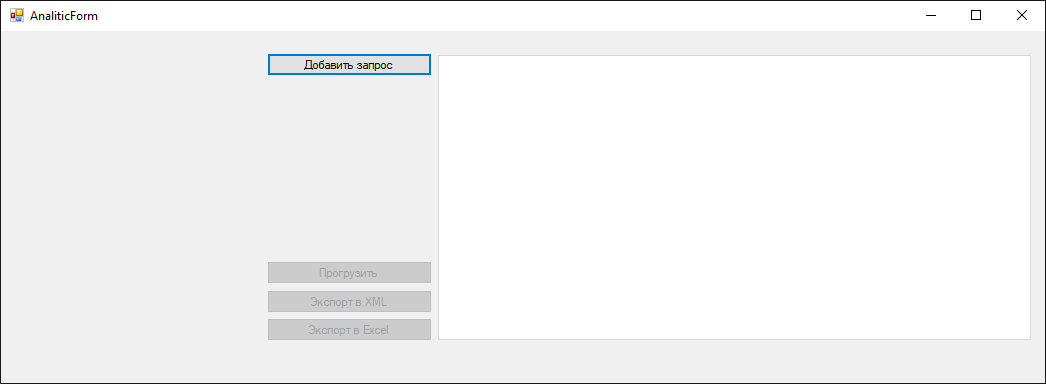
Условия, при которых возможно выполнение операции:

Успешный вход с пользователем типа «Аналитик».

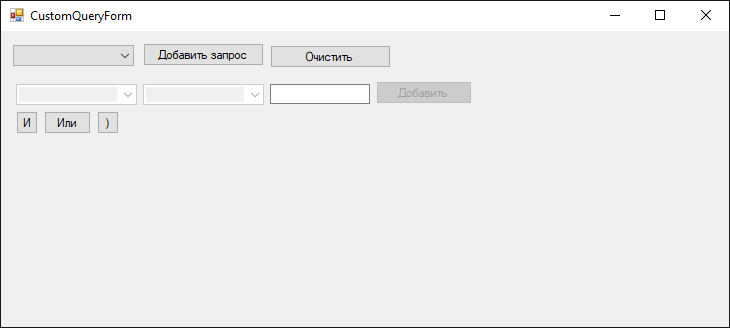
Подготовительные действия: не требуются.

Основные действия в требуемой последовательности:

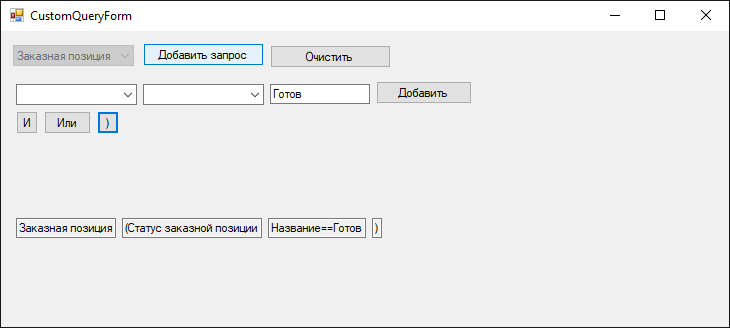
1. Нажать на кнопку «Добавить запрос».



1. Выполнить построение запроса в открывшемся окне.



1. Нажать на кнопку «Добавить запрос».



Заключительные действия: не требуются.

Затрачиваемые ресурсы: 1-2 минуты.

Операция 3. Прогрузка запросов и загрузка в файл.

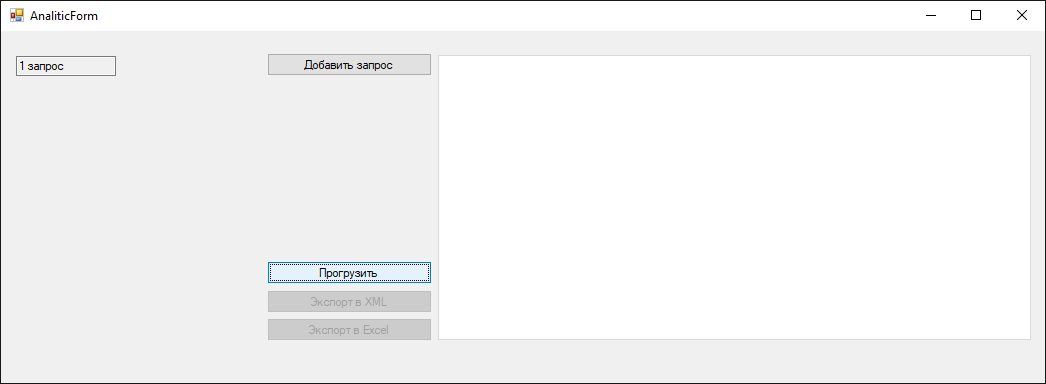
Условия, при соблюдении которых возможно выполнение операции:

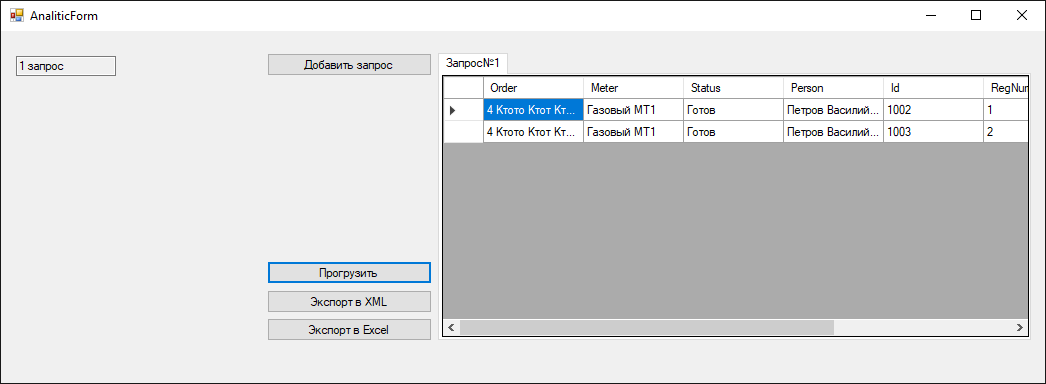
Был добавлен хотя бы один запрос.

Подготовительные действия: не требуются.

Основные действия в требуемой последовательности:

1. Нажать на кнопку «Прогрузить».





1. Для экспорта в файл формата Xml нажать на кнопку «Экспорт в XML», для экспорта в файл формата Excel нажать на кнопку «Экспорт в Excel» и выполнить сохранение в открывающемся диалоговом окне.

Заключительные действия: После завершения работы с программой следует нажать на крестик в правом верхнем углу.

Затрачиваемые ресурсы: 5-10 секунд.

## Аварийный ситуации

В случае возникновения ошибок при работе системы, не описанных ниже в данном разделе, необходимо обращаться к сотруднику подразделения технической поддержки либо к ответственному системному администратору.

| Класс ошибки | Ошибка | Описание ошибки | Требуемые действия пользователя при возникновении ошибки |
| --- | --- | --- | --- |
| Программа | База данных не найдена | Проблема с подключением к базе данных | Для устранения проблем с подключением к базе данных требуется обратиться к системному администратору. |
| Программа | Ошибка: неверный логин/пароль | Система не распознаёт вводимую пару логин/пароль как верную | Повторить попытку входа, при повторении ошибки обратиться к системному администратору. |

## Рекомендации по освоению

В качестве контрольного примера рекомендуется выполнить все операции задач, описанные в п. 4.2. настоящего документа.

# Приложение Р. Руководство программиста

## Назначение и условия применения систем

Для работы системы необходимы следующие условия:

* Наличие программы Microsoft Excel 2007 и позднее.
* Объём оперативной памяти должен быть не ниже 1 ГБ.
* Наличие сервера SQL.

## Характеристика программы

Временные характеристики:

* Время первого запроса к базе данных может превышать 20 секунд, последующие запросы не превышают 10 секунд.
* Время запуска компонентов Microsoft Excel может превышать 10 секунд и зависит от загруженности системы.

Режимы работы. Существует три режима работы в соответствующие типам пользователей. В режиме администратора пользователь имеет возможность вносить изменения в любых сущностях базы данных и выполнять сохранение/загрузку базы данных с использованием файлов формата xml. В режиме оператора пользователь имеет возможность вносить новые записи заказов и заказных позиций, а также изменять записи заказных позиций. В режиме аналитика пользователь не имеет доступа к изменению записей.

## Обращение к программе

Внешнее управление содержимым системы осуществляется через Windows Forms интерфейс, описанный в документе «руководство пользователя».

Чтобы выполнить подключение к серверу следует изменить строку подключения к базе данных, чтобы она соответствовала имеющейся базе данных.

Внутреннее управление базой данных происходит автоматически посредством Microsoft SQL или подобным.

Процесс запуска базы данных.

Чтобы запустить базу данных в первый раз программист должен выполнить ряд шагов.

1. Создать новый пустой экземпляр базы данных SQL
2. Запустить созданную базу данных
3. Выполнить код SQL, расположенный в файле model1.edmx.sql Данная операция инициализирует таблицы сущностей и добавляет в систему супер-пользователя с логином root и паролем 123qwe. Данная учётная запись используется для подключения к базе данных через программный интерфейс. После настройки пользователей, у данного пользователя рекомендуется поменять пароль либо выполнить его удаление.

## Входные и выходные данные.

Данные базы данных возможно загрузить из файла типа Xml. Для этого в интерфейсе программы при входе под учётной записью администратора нажмите на кнопку «Загрузить из XML» и в открывающейся форме выберите искомый файл.

Данный базы данных возможно сохранить в файл типа Xml. Для этого в интерфейсе программы при входе под учётной записью администратора нажмите на кнопку «Сохранить в XML» и в открывающейся форме выберите каталог для сохраняемого файла.

## Сообщения

При загрузке базы данных возможна ошибка: «В Xml файле присутствует ошибка». Это означает, что файл, который был выбран, содержит неверную структуру. Для ознакомления с требуемой структурой данных рекомендуется заполнить базу данных небольшим количеством тестовых записей и выполнить сохранение базы данных в файл.

# Приложение C. Листинги и дополнительные материалы

Листинги программы и дополнительные материалы находятся по ссылке: http://github.com/ankokovin/CourseWork