

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт цифровых технологий

(наименование института полностью)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Разработка программного обеспечения

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка приложения для автоматизации учета складского участка
производства жидких моющих средств производственно-технологического
отдела предприятия»

Обучающийся

А.А. Ощепков

(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Казаченок

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

Тема бакалаврской работы «Разработка приложения для автоматизации учета складского участка производства жидких моющих средств производственно-технологического отдела предприятия».

Бакалаврская работа посвящена разработке автоматизированной системы складского учета производственно-технологического подразделения предприятия.

Целью данной выпускной бакалаврской работы является разработка программного приложения для автоматизации учета складского участка производства жидких моющих средств производственно-технологического отдела предприятия.

Объектом исследования бакалаврской работы является складской участок производства жидких моющих средств производственно-технологического отдела предприятия.

Предметом исследования выступает программное приложение для автоматизации складского учета производственного участка жидких моющих средств предприятия.

Практическая значимость бакалаврской работы заключается в разработке прикладного программного обеспечения складского учета производственно-технологического участка предприятия.

Результатом выпускной квалификационной работы является разработанное приложение для автоматизации деятельности склада и складских операций производственно-технологического подразделения предприятия.

Выпускная квалификационная работа представлена на 112 страницах и включает 81 рисунок и список используемых источников из 25 наименований.

Оглавление

Введение.....	5
Глава 1 Постановка задачи на разработку приложения для автоматизации складского учета участка производства жидких моющих средств производственно-технологического отдела предприятия ООО «АВЭЛЛА»....	8
1.1 Анализ и описание предметной области складского учета участка производства жидких моющих средств производственно-технологического отдела предприятия ООО «АВЭЛЛА»	8
1.2 Обзор и анализ аналогов программного обеспечения складского учета производственного участка.....	15
1.3 Разработка требований к программному обеспечению складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА».....	35
Глава 2 Проектирование программного обеспечения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»	51
2.1 Выбор методологии проектирования программного обеспечения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА».....	51
2.2 Логическое моделирование программного обеспечения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА».....	54
2.3 Моделирование данных системы складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА».....	57
Глава 3 Реализация и тестирование программного обеспечения складского учета участка производства жидких моющих средств производственно-технологического отдела предприятия ООО «АВЭЛЛА»	61
3.1 Реализация программного обеспечения складского учета участка производства жидких моющих средств ООО «АВЭЛЛА»	61

3.2 Тестирование программного обеспечения складского учета участка производства жидких моющих средств ООО «АВЭЛЛА»	90
Заключение.....	108
Список используемых источников	110
Приложение А Образец приходного ордера по сырью и материалам	113
Приложение Б Образец требования-накладной	115
Приложение В Образец приходного ордера готовой продукции	116
Приложение Г Код приложения, отвечающий за создание меню главной формы складского учета ООО «АВЭЛЛА».....	118
Приложение Д Код приложения, отвечающий за ввод информации о подразделениях складского учета ООО «АВЭЛЛА».....	121
Приложение Е Код приложения, отвечающий за ввод информации о поставщиках складского учета ООО «АВЭЛЛА»	126
Приложение Ж Код приложения, отвечающий за ввод информации о входящих материалах складского учета ООО «АВЭЛЛА»	133
Приложение И Код приложения, отвечающий за ввод информации о покупателях складского учета ООО «АВЭЛЛА».....	140
Приложение К Код приложения, отвечающий за ввод информации об отпускаемых продуктах складского учета ООО «АВЭЛЛА».....	147
Приложение Л Код приложения, отвечающий за вывод информации о поставщиках, сырье, покупателях и продукции складского учета ООО «АВЭЛЛА».....	154
Приложение М Код интегрированного отчета по подразделениям складского учета ООО «АВЭЛЛА».....	157

Введение

В современной деятельности коммерческих структур и хозяйствующих объектов существует объективная необходимость профессионального учета экономической активности предприятия. Наряду с множеством бизнес-процессов, составляющих деятельность фирмы, включающие сбыт, производство и продвижение продукции, организацию технологического обеспечения и бухгалтерского учета, центральное место занимает учет перемещения материальных ценностей в рамках производственно-коммерческой деятельности организации, что является прерогативой логистически-складских операций.

Для любой коммерческой организации, осуществляющей свою деятельность в условиях рыночной конкуренции, является жизненно необходимым организация собственных бизнес-процессов оптимальным образом. В плане же оптимизации рабочих процессов фирмы большое значение имеет их автоматизация. В этом смысле актуальность темы данной работы была обусловлена необходимостью автоматизации складского учета организации, занимающейся производством товаров народного потребления. Автоматизация складского учета фирмы позволяет рационально организовать приемку и учет сырья и материалов, их перемещение в производство, а также отгрузку готовой продукции покупателям, что позволяет оптимизировать рабочие процессы в данной сфере деятельности.

База исследования: исследование проводится на базе складской деятельности производственного участка жидких моющих средств Общества с ограниченной ответственностью «АВЭЛЛА» (далее ООО «АВЭЛЛА»).

Целью данной выпускной бакалаврской работы является разработка программного приложения для автоматизации учета складского участка производства жидких моющих средств производственно-технологического отдела предприятия.

Объектом исследования бакалаврской работы является складской участок производства жидких моющих средств производственно-технологического отдела предприятия.

Предметом исследования выступает программное приложение для автоматизации складского учета производственного участка жидких моющих средств предприятия.

Для достижения поставленной цели в рамках данной работы решаются следующие задачи:

- проанализировать и описать предметную деятельность предприятия, его организационную структуру, обеспеченность автоматизации складского учета;
- провести обзор и анализ аналогов программных средств складского учета предприятия;
- сформулировать требования к программному приложению складского учета производственного участка жидких моющих средств;
- выбрать методологию проектирования программного обеспечения складского учета производственного участка жидких моющих средств;
- произвести логическое моделирование программного обеспечения и моделирование данных системы складского учета производственного участка жидких моющих средств;
- реализовать разработку программного обеспечения складского учета участка производства жидких моющих средств;
- протестировать разработанное программное обеспечение складского учета участка производства жидких моющих средств.

Практическая значимость бакалаврской работы заключается в разработке прикладного программного обеспечения складского учета производственно-технологического участка предприятия.

Бакалаврская работа состоит из введения, трех глав, заключения, списка используемой литературы, приложений.

В первой главе проведен анализ и описание предметной области складского учета производственного участка жидких моющих средств. Также проведен обзор и анализ аналогов программного обеспечения, предлагаемых на рынке. На основании данной аналитической работы произведена разработка требований к программному обеспечению складского учета нашего предприятия.

Во второй главе осуществлен выбор методологии проектирования программного обеспечения складского участка производственно-технологического подразделения. С учетом этого, проведено логическое моделирование программного обеспечения участка складских операций. И, в итоге, произведено моделирование информационных потоков системы складского учета производственного участка жидких моющих средств предприятия.

Третья глава посвящена реализации программного обеспечения складского учета участка производства жидких моющих средств. И, с учетом установленных требований, проведено тестирование программного обеспечения складского учета участка производства жидких моющих средств.

Выпускная квалификационная работа представлена на 113 страницах и включает 81 рисунок и список литературы из 25 наименований.

Глава 1 Постановка задачи на разработку приложения для автоматизации складского учета участка производства жидких моющих средств производственно-технологического отдела предприятия ООО «АВЭЛЛА»

1.1 Анализ и описание предметной области складского учета участка производства жидких моющих средств производственно-технологического отдела предприятия ООО «АВЭЛЛА»

Общество с ограниченной ответственностью «АВЭЛЛА» является многопрофильной организацией. Данная организация занимается торговлей и производством товаров народного потребления, и была создана в 2018 году. Миссией ООО «АВЭЛЛА» является максимальное удовлетворение потребностей своих клиентов, что и обосновывает основные принципы работы Компании: открытость во взаимодействии с социумом, внимательное отношение к людям, как к клиентам, так и к собственному персоналу, увлеченный поиск инноваций в сфере научно-технологических разработок, оценка потенциала будущей технологии, ее разработка до готового продукта и выход с отвечающим потребностям клиентов продуктом на рынок. Данный подход определяет и стратегию развития Компании, которую можно обозначить как клиентоцентрированность и ориентация на инновации.

Организационная структура ООО «АВЭЛЛА» состоит из системы подразделений, выполняющих функциональные обязанности по направлениям производства. Процесс производства разделен по стадиям, которым и соответствуют подразделения.

- отдел снабжения, занимающийся материально-техническим обеспечением производства;
- производственно-технологический отдел, в задачи которого входит организация производства готовой продукции;
- отдел сбыта, который занимается реализацией готовой продукцией;

- отдел кадров, занимающийся подбором кадров Компании;
- бухгалтерия, занимающаяся учетом материальных ценностей и их потоками в рамках технологического процесса.

Общее руководство Компанией осуществляет директор.

На основании данных по подразделениям с помощью программы Microsoft Visio [7] была построена схема организационной структуры ООО «АВЭЛЛА» (рисунок 1).

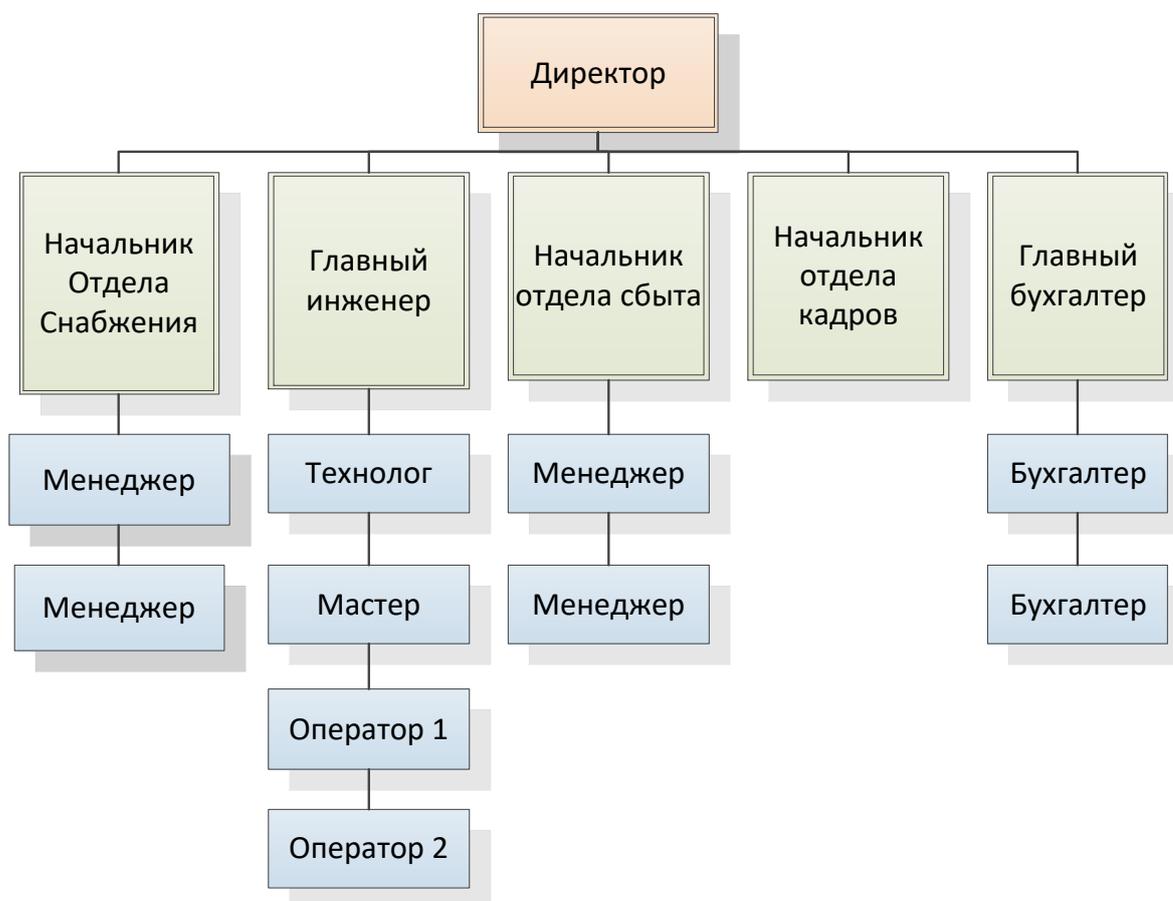


Рисунок 1 – Организационная схема ООО «АВЭЛЛА»

В ведении главного инженера находится производственно-технологический отдел, основной целью которого является обеспечение бесперебойного производства определенной номенклатуры продукции.

Данный отдел также включает производственное подразделение, занимающееся выпуском номенклатуры жидких моющих средств, в котором и проводились научно-исследовательские работы по объекту автоматизации.

Жидкие моющие средства производства ООО «АВЭЛЛА» изготавливаются согласно ГОСТ 31696 – 2012 «Продукция косметическая и гигиеническая моющая. Общие технические условия» [9] и ГОСТ 32478 – 2013 «Товары бытовой химии. Общие технические требования» [10], определяющие технические условия производства. Данные технические условия регулируют требования к поставляемому сырью и материалам, передачу их в производство, требованию к производимой продукции, выражающейся в соответствии ряду объективных показателей, упаковку и складирование готовой продукции.

Поэтому производственный процесс в рамках данного подразделения распределен между тремя основными модулями:

- склад сырья и материалов,
- производство и выпуск готовой продукции,
- склад готовой продукции.

В соответствии с данной принципиальной производственной схемой выстраивается и организационная структура управления данным подразделением. Складом материалов и сырья заведует кладовщик, в подчинении которого находятся 2 грузчика, Производством руководит мастер, в подчинении которого находятся 2 оператора, и Складом готовой продукции также заведует кладовщик, в подчинении которого находятся также 2 грузчика. С помощью программы Microsoft Visio также была построена схема организационной структуры подразделения производства жидких моющих средств ООО «АВЭЛЛА» (рисунок 2).

Для анализа предметной области склада производственного участка жидких моющих средств ООО «АВЭЛЛА», были проанализированы рабочие процессы выбранной области для того, чтобы в дальнейшем возможно было обоснование автоматизации данного участка работы.

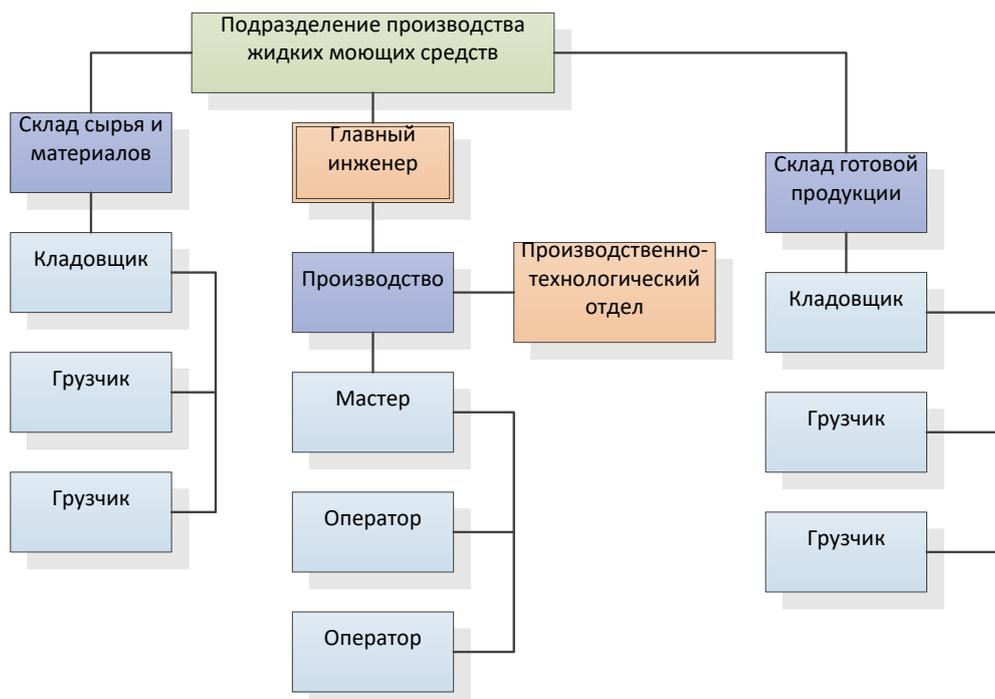


Рисунок 2 – Организационная схема подразделения производства жидких моющих средств ООО «АВЭЛЛА»

Согласно практическим рекомендациям Е.С. Рычковой, рабочие процессы были описаны с использованием трех параметров: вход, преобразование и выход [21].

Вход. На этом этапе происходит поступление всех необходимых ресурсов и материалов для производства:

- сырье: поступление химических компонентов, таких как ПАВ (поверхностно-активные вещества), консерванты, ароматизаторы, красители и другие добавки;
- упаковка: получение упаковочных материалов, таких как бутылки, крышки, этикетки и коробки;
- оборудование и инструменты: проверка наличия и состояния необходимого оборудования (смесители, насосы, линии розлива и т.д.), а также инструментов для контроля качества;
- документация: получение и проверка сопроводительных документов, сертификатов качества на сырье и упаковку.

Преобразование. Этот этап включает в себя все процессы, связанные с производством жидких моющих средств:

- приемка и хранение сырья: проверка качества поступившего сырья, его маркировка и хранение на складе;
- подготовка формул: разработка и корректировка рецептур в соответствии с требованиями и стандартами;
- смешивание: процесс смешивания компонентов в соответствии с формулой, который может включать нагревание, охлаждение, перемешивание и другие операции;
- контроль качества: проведение тестирования на каждой стадии производства для обеспечения соответствия стандартам (визуальный контроль, тесты на pH, вязкость и т.д.);
- розлив: перемещение готовой смеси в упаковку с использованием автоматизированных линий розлива;
- этикетирование и упаковка: нанесение этикеток на упаковку и укладка готовой продукции в коробки.

Выход. На этом этапе происходит завершение всех процессов и подготовка продукции к отгрузке:

- готовая продукция: формирование партий готовых жидких моющих средств, готовых к продаже;
- хранение: организация хранения готовой продукции на складе до момента отгрузки;
- отгрузка: подготовка и отправка продукции клиентам или в торговые сети, включающие оформление документов, транспортировку и контроль за выполнением условий поставки;
- отчетность: ведение учета произведенной продукции, анализ продаж и планирование дальнейших поставок.

Таким образом, рабочие процессы на складском участке производства жидких моющих средств можно представить, как последовательность этапов, начиная с поступления сырья и заканчивая отгрузкой готовой продукции.

На основании выделенных рабочих процессов, согласно рекомендациям Д.В. Кознова [17], были заложены принципиальные основы представления каждого рабочего процесса с описанием действующих лиц, действий, результатов и состояний.

Поступление сырья и материалов производится от поставщиков на основании товарной накладной по стандартной форме ТОРГ-12. По данной накладной кладовщик принимает по количеству поставляемое сырье и материалы, дает задание грузчикам, которые расставляют приходящую номенклатуру по местам на складе. Сырье и материалы приходятся кладовщиком на основании Приходного ордера по Типовой межотраслевой форме № М-4, утвержденной постановлением Госкомстата России от 30.10.97 № 71а, образец которого представлен в Приложении А. Сырье и материалы представлены основными типами – поверхностно-активные вещества, полиэтиленовые флаконы, крышки, этикетки, химикаты (комплексообразователи, консерванты, красители, отдушки, ведра и крышки, регуляторы кислотности/щелочности).

Далее, согласно производственного плана, сырье отгружается на производственный участок. Для этого кладовщик отпускает мастеру на производстве определенное количество сырья и материалов, требуемых для производства жидких моющих средств, а грузчики перемещают данное количество на производство. Отпуск производится на основании требования-накладной по типовой межотраслевой форме № М-11, утвержденной постановлением Госкомстата России от 30.10.97 № 71а, образец которого представлен в приложении Б. После выпуска продукции (жидких моющих средств) готовая продукция приходит на склад готовой продукции. При этом мастер сдает по количеству номенклатуру готовой продукции кладовщику. Оприходование продукции также производится на основании Приходного ордера по Типовой межотраслевой форме № М-4, утвержденной постановлением Госкомстата России от 30.10.97 № 71а, но только уже по

готовой продукции. Образец приходного ордера по готовой продукции представлен в Приложении В.

Для формального описания бизнес-процессов склада производственного участка жидких моющих средств была использована нотация Business Process Model and Notation (BPMN) с учетом практических рекомендаций [2]. Первоначально был описан бизнес-процесс КАК ЕСТЬ (AS-IS), а затем спроектирован как должен оптимально работать бизнес-процесс, то есть TO-BE.

В автоматизированном виде складской учет построен на основе программы Microsoft Excel, в котором отдельные операции складского учета разделены на отдельные листы [1]. Складской учет ведется в листах:

- поставщики и покупатели, где учитываются основные контрагенты и им присваиваются отдельные уникальные идентификационные номера;
- точки учета, где отдельно учитываются и идентифицируются различные стадии прохождения товара: склад сырья и материалов, производство, склад готовой продукции;
- приход и расход, где учитываются приходно-расходные операции по сырью и материалам и готовой продукции;
- оборотная ведомость, где представлены обороты и остатки по продукции.

Складской учет в автоматизированном виде организован правильно, так как учитывает все необходимые операции и стадии технологического процесса, связанного с перемещением продукции, однако имеет и ряд существенных недостатков.

Во-первых, Excel не позволяет хранить информацию отдельными блоками, поэтому информация представлена единым списком, что представляет определенные трудности при поиске информации.

Во-вторых, лист по размеру в Excel имеет ограничения, что в будущем создаст определенные трудности, когда количество операций увеличится и столкнется с ограничением, так как список не будет уместиться в листе.

В-третьих, сплошной список не позволяет сортировать и классифицировать информацию по задаваемым параметрам, по причине чего ежемесячно приходится файлы складского учета пересохранять с перенесением остатков, что также создает определенные неудобства.

Предлагается применить преимущества объектно-ориентированного программирования, которые позволят исправить недостатки используемого в текущее время программного продукта (Excel). Предлагается выстроить систему классов, управляемых методами, средствами программной среды C#.

Таким образом, объектом автоматизации был выбран складской учет с целью его усовершенствования для разрешения выявленных недостатков. Поскольку складской учет организован правильно, однако инструмент Excel имеет недостатки, поэтому необходимо разработать новое программное решение складского учета. Такое программное решение предлагается реализовать средствами объектно-ориентированного программирования. А для теоретического обоснования программного продукта предлагается построить диаграмму классов, на основе которой уже затем разработать блок-схему программы, которая позволит написать непосредственно программный код и протестировать его. Итак, на основании анализа рабочих процессов предприятия удалось уточнить предметную область автоматизации деятельности организации, направленную на разработку программного приложения складского учета производства жидких моющих средств.

1.2 Обзор и анализ аналогов программного обеспечения складского учета производственного участка

В современной ситуации на рынке программного обеспечения предлагаются следующие программные продукты, обеспечивающие

складской учет производственной деятельности. Обзор производился по ключевому поисковому запросу «Складской учет производственной деятельности» с использованием поисковика Yandex и было выбрано 10 наиболее приемлемых программных продукта.

Первый продукт, который был рассмотрен – это складская программа «Мой Склад» (www.moysklad.ru). Данный продукт является облачным и, в этом плане, более удобен для удаленной работы. «Мой склад» хорошо подходит для розничной и оптовой торговли, а также для сферы общепита, нежели для производства. Одной из характеристик является интеграция со всеми популярными сервисами и приложениями для бизнеса. Данный продукт больше подходит для малого и среднего бизнеса и предлагает следующие функциональные возможности.

Во-первых, интеграция с оборудованием и сервисами:

- удобный визуальный редактор этикеток и ценников, возможность массовой печати, поддержки сканеров, печати и генерации EAN-8, EAN-13, Code128 и др.;
- интеграция с фискальными регистраторами АТОЛ, ШТРИХ-М, Viki Print, принтером чеков;
- интеграция с МодульКассой и смарт-терминалами Эвотор, возможность подключения терминалов Verifone и PAX для оплаты картами;
- экспорт данных в 1С:Бухгалтерию для составления налоговой отчетности по принципу «документ-в-документ»;
- интеграция с сервисом для работы с терминалом сбора данных Клеверенс Склад 15: получение на склад нового товара и подбор товара для отгрузки поставщику, выбор распределения товара из ячеек, печать этикеток с названием, единицей измерения, штрих-кодом и номером ячейки.

Во-вторых, формирование и печать складских документов (М-11, ИНВ-3, ТОРГ-16, МХ-1 и других). Можно отправлять документы по email и

выгружать в ЭДО. Все данные надежно защищены, а доступ к системе открыт 24/7.

В-третьих, адресное хранение позволяет указывать складские ячейки для товаров, чтобы знать, где находится та или иная позиция. Система подскажет, куда лучше поместить вновь полученный товар, и сформирует лист подбора для ваших кладовщиков при подготовке отгрузки.

В-четвертых, учет по сериям и срокам годности – работает для конкретных товаров или их групп. Позволяет принимать, отгружать, перемещать и возвращать товар определенной серии – а значит продавать товар с более ранним сроком годности в первую очередь и списывать просроченную продукцию. В программе «Мой Склад» поддерживается любое количество складов и мест хранения, производственные рецепты и операции, работа со штрих-кодами, серийными номерами и модификациями товаров.

Помимо складского учета «Мой Склад» умеет:

- обрабатывать заказы покупателей (рисунок 3);

№	Время	Контрагент	Организация	Сумма	Выставлено сче...	Оплачено	Отгружено	Зарезервировано	Статус	Отправлено
00006	13.05.2022 14:10	ООО "Покупатель"		2 918,00	0,00	0,00	0,00	0,00	Отменен	
00005	13.05.2022 14:10	ООО "Покупатель"		11 095,00	0,00	0,00	0,00	0,00	Новый	
00004	13.05.2022 14:09	ООО "Покупатель"		3 238,00	0,00	0,00	0,00	0,00	Новый	
00003	13.05.2022 14:09	ООО "Покупатель"		3 764,00	0,00	0,00	0,00	0,00	Доставлен	
00002	13.05.2022 14:08	ООО "Покупатель"		5 145,00	0,00	0,00	0,00	0,00	Собран	
00001	13.05.2022 14:06	ООО "Покупатель"		1 797,00	0,00	0,00	0,00	0,00	Новый	
Итого				27 957,00	0,00	0,00	0,00	0,00		

Рисунок 3 – Обработка программой «Мой склад» заказов покупателей

- собирать финансовые показатели (рисунок 4);

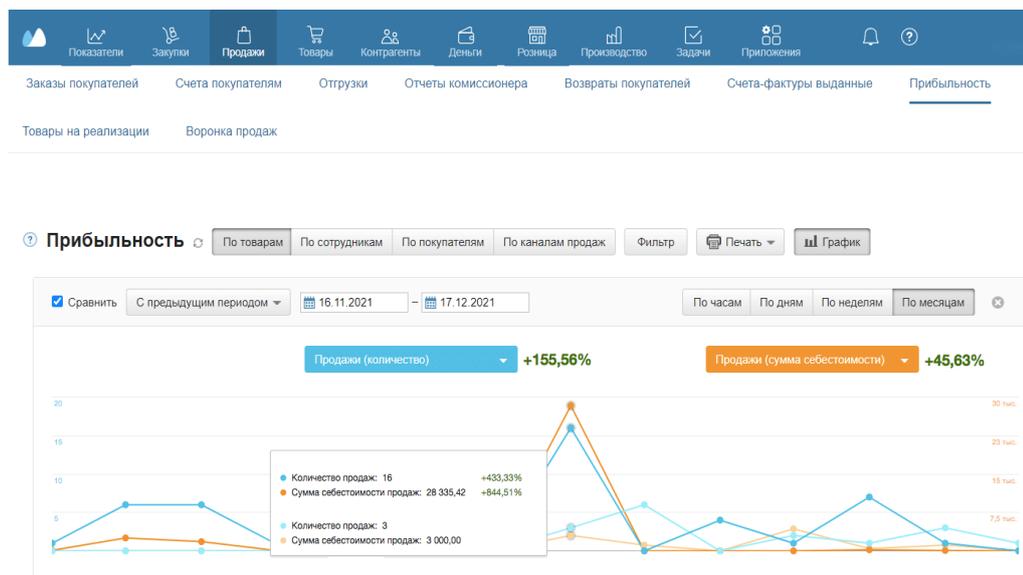


Рисунок 4 – Отчет «Прибыльность» программы «Мой склад»

- выводить отчет о прибылях и убытках (рисунок 5).

Прибыли и убытки Печать

Период: Вчера · Сегодня · Неделя · Месяц Организация

Статья	Прибыль (убыток)
Выручка	6 754 617,30
Себестоимость	363 737,84
Валовая прибыль	6 390 879,46
Операционные расходы	1 864 075,50
Списания	24 075,50
Аренда	100 000,00
Зарплата	240 000,00
Маркетинг и реклама	1 500 000,00
Операционная прибыль	4 526 803,96
Налоги и сборы	0,00
Чистая прибыль	4 526 803,96

Рисунок 5 – Отчет о прибылях и убытках программы «Мой склад»

Вторая программа, рассмотренная нами – это продукт Subtotal (www.subtotal.ru), которая представляет собой приложение для онлайн-кассы в розничном магазине или общепите. Позволяет контролировать сотрудников,

продажи, наценку, товарные остатки. Автоматизирует документооборот, печать ценников и некоторые другие типовые действия. Есть поддержка ЕГАИС, соответствие требованиям 54 ФЗ. Не получится управлять адресным хранением и доставкой.

Складские операции:

- прием, списание, перемещение товаров по складам и контроль всех операций;
- технологические карты для общепита и продажа составных товаров (комплектов);
- печать ценников с использованием готовых шаблонов или создание собственных;
- заказы поставщикам, оптимизация складских остатков, напоминания об окончании товара на складе;
- быстрая инвентаризация с помощью ТСД или сканера штрих-кода;
- установка цены с учетом себестоимости закупки;
- контроль сроков годности товаров, напоминания об окончании срока годности;
- мгновенная переоценка ассортимента или переоценка на дату;
- модификации товаров по цвету, размеру и любым произвольным параметрам;
- ведение базы поставщиков, отчеты и акты сверок с поставщиками;
- работа с УТМ (ЕГАИС) для приема ТТН и дальнейшей продажи алкоголя,
- оприходование маркированного товара из документов ЭДО.

Наибольшее преимущество данное приложение дает в кассовых операциях:

- рабочее место кассира на любом планшете или ноутбуке (Windows или Android);

- простой, интуитивный интерфейс в котором разберется даже ребенок;
- подключение сканера штрих-кода, банковского терминала, весов, экрана покупателя;
- работа с дисконтными картами, акционными товарами и рекомендации кассиру по продажам аксессуаров и пр.;
- удобный поиск товаров по категориям и характеристикам (размеры, цвета);
- применение различных цен на разных торговых точках в рамках одной розничной сети;
- резервирование товара, оформления возврата, печать товарных чеков;
- подключение онлайн-касс АТОЛ, ШТРИХ-М, ЭВОТОР, Дримкас;
- продажи алкоголя (ЕГАИС) и маркированной продукции (datamatrix).

Кассовое приложение SUBTOTAL значительно упрощает работу кассиров, помогает повышать средний чек и продажи. Работает в соответствии с требованиями действующего законодательства по онлайн-кассам (ФЗ 54) и позволяет продавать товары с маркировкой (ФЗ 487). Презентация программы представлена на рисунке 6.

Получите рекомендации по выбору онлайн-кассы

Подобрать онлайн-кассу

Интеграция

Подключите к SUBTOTAL все необходимое

<p>Онлайн-кассы</p> <p>Штрих-М, АТОЛ, Dreamkas, Эвотор</p>	<p>Торговое оборудование</p> <p>Сканеры штрих-кодов, принтеры ценников, весы и др.</p>	
<p>Эквайринг</p> <p>Оборудование для приема карточных платежей на кассе</p>	<p>Интернет-магазин</p> <p>Интеграция, синхронизация остатков, ассортимента, цен</p>	
<p>Бухгалтерия</p> <p>Интеграция с МоёДело, 1С Бухгалтерия</p>	<p>Банк</p> <p>Загрузка платежей из банка и автоматическое разнесение поступлений по ранее выставленным клиентам счетам (заказам)</p>	

Все партнёры

Попробовать

Я принимаю условия [Политики конфиденциальности](#)

Рисунок 6 – Презентация складского-кассового приложения Subtotal

Третьей программой, представленной в обзоре, является 1С: Розница (www.e-office24.ru), которая является программой для розничной торговли, включающая складской функционал, ориентированная на средний и крупный бизнес с налогообложением по ОСНО и УСН. Есть локальная версия и сайт, возможность вести учет в отдельных торговых точках и сетях. Как было сказано, 1С имеет широкий функционал, в том числе и складской, но при этом сложный интерфейс. Облачную версию нельзя доработать под свой бизнес, а для установки и обслуживания локальной понадобится помощь системного администратора.

Для процесса управления складом предусмотрены следующие возможности. Размещение и отбор товаров осуществляются в соответствии с заданной стратегией размещения и отбора для области хранения (рисунок 7).

Обувь (упаковки) (Область хранения) (1С:Предприятие)

Обувь (упаковки) (Область хранения)

Главное Правила размещения товаров Задачи Мои заметки Товары в ячейках

Записать и закрыть Записать Создать на основании Еще ?

Склад: Центральный склад Помещение: Коммерческие товары

Наименование: Обувь (упаковки)

Описание: Область хранения обуви

Ячейки используются для хранения обособленных товаров

Для хранения товаров, пришедших под обособленное обеспечение. В каждой ячейке хранятся товары только под одно назначение.

Использование ячеек определяется стратегиями отбора и размещения

Для хранения всех товаров согласно стратегии отбора и размещения.

При размещении **Монотоварность ячейки определяется:**

соблюдать строгую монотоварность по номенклатуре

по номенклатуре и характеристике

по номенклатуре, характеристике и серии

Для размещения только в монотоварные или в свободные ячейки.

учитывать приоритеты размещения

в монотоварные ячейки: 1

в смешанные ячейки: 2

в свободные ячейки: 3

в ячейки с другим товаром: 4

Приоритет отбора:

из монотоварных ячеек: 3

из смешанных ячеек: 1

"под остаток": 2

Наивысший приоритет равен 1.

Периодическая инвентаризация ячеек

Не проводить периодическую инвентаризацию в ячейках

Пересчитывать товары в ячейках каждые: 0 день (дней)

Рисунок 7 – Область хранения товара с 1С

При оформлении процесса пересчета товаров можно отразить фактическое и учетное количество товаров, отклонения фактического количества от учетного (рисунок 8).

🏠 ← → ☆ Пересчет товаров 00-00000012 от 29.04.2015 16:36:04 ×

Главное Документооборот Задачи Мои заметки

Провести и закрыть 📄 📄 📄 📄 📄 📄 📄 Печать Отчеты Еще ?

Статус: Выполнено

Информация Товары (75)

Отбор:

Только с отклонениями ↩ 🔍 📄 Изменить качество Заполнить по отбору Еще

N	Номенклатура	Характеристика	Назначение	Серия	Ед....	По учету	По факту	Отклонение
1	Ботинки женск...	36, Белый, 5, и...		<сер...		10,000	9,000	-1,000
2	Ботинки женск...	36, Белый, 9, и...		<сер...		10,000	11,000	1,000
3	Ботинки женск...	37, Бежевый, 5...		<сер...		10,000	10,000	
4	Ботинки женск...	37, Зеленый, 6...		<сер...		10,000	10,000	
5	Ботинки женск...	37, Красный, 5...		<сер...		10,000	10,000	
6	Ботинки женск...	37, Белый, 8, и...		<сер...		10,000	10,000	
7	Ботинки женск...	38, Бежевый, 6...		<сер...		11,000	11,000	
8	Ботинки женск...	43, Бежевый, 5...		<сер...		6,000	8,000	2,000
9	Ботинки женск...	44, Коричневы...		<сер...		7,000	5,000	-2,000
..	Ботинки женск...	45, Бежевый, 6...		<сер...		6,000	6,000	
..	Ботинки женск...	36, Зеленый, 5...		<сер...		4,000	4,000	
..	Ботинки женск...	37, Белый, 8, и...		<сер...		4,000	4,000	
..	Ботинки женск...	43, Бежевый, 5...		<сер...		6,000	6,000	

Рисунок 8 – Пересчет товаров в 1С

Если на складе выявлены излишки, недостачи и обнаружены «лишние» товары без проведения пересчета товаров (в процессе оперативной работы), то в таких случаях для регистрации расхождений применяются ордера на отражение излишков, недостач и порчи товаров, изменяющие фактические остатки товаров на складе (рисунок 9).

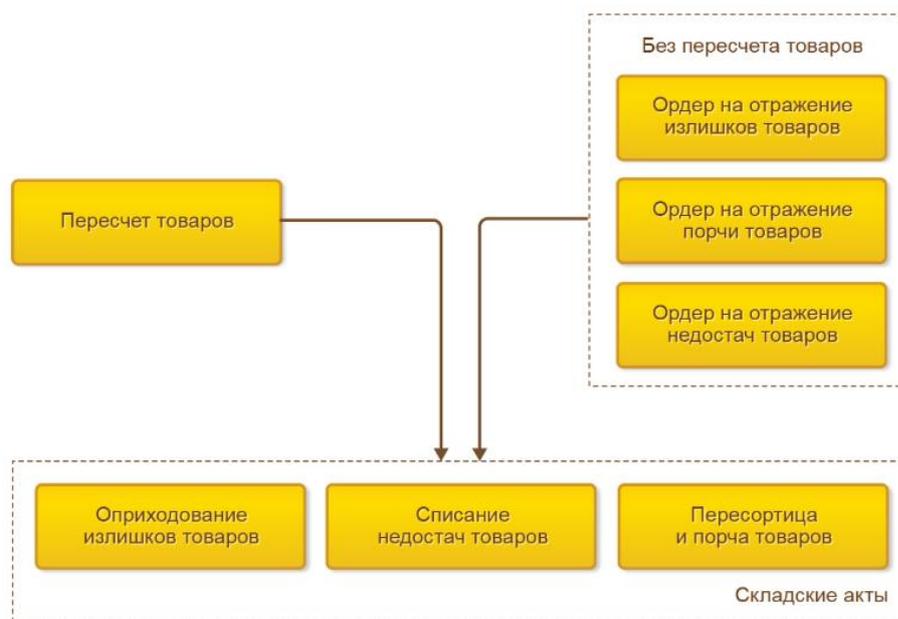


Рисунок 9 – Схема товародвижения в 1С

Серии и сроки годности являются разрезом (аналитикой) складского учета, могут указываться при поступлении, отгрузке товара и в документах внутреннего товарооборота для решения, например, следующих задач:

- управления рекламациями и организации постпродажного обслуживания – когда необходимо знать, от какого поставщика получен или какому покупателю передан конкретный экземпляр товара, комплектующие с какими сериями использованы в производстве конкретного экземпляра товара;
- учет и управление остатками товаров по срокам годности – необходимо знать, когда истекает срок годности запасов и какой на складе остаток товаров с определенным сроком годности;
- организация хранения товаров в отрезках, рулонах, бухтах и других единицах хранения переменного размера – в серии будет храниться фактический остаток материала в том или ином рулоне;
- учет товаров, потребительские свойства которых могут незначительно отличаться для разных производственных партий –

например, обои из разных партий могут незначительно отличаться оттенками (разнотон) и при отгрузке товара клиенту необходимо подбирать товар одного тона (одной серии) (рисунок 10).

Остатки товаров по срокам годности

В период: Неделя Следующая неделя Номенклатура:

Склад:

Сформировать Настройки... Варианты отчета Найти...

Остатки товаров по срокам годности

Параметры: Период окончания срока годности: 14.12.2015 - 20.12.2015
Количество товаров: В единицах хранения

Склад	Помещение	Остаток	Дней до истечения срока годности	Остаток срока годности, %
Артикул	Номенклатура, Характеристика	Назначение	Ед. изм.	Срок годности
Годен до				
Ларек "Розница"				
Арт-33344555	Ассорти (конфеты),		упак	60 сут
		10,000		
		5,000	Не годен	
		5,000	Не годен	
Арт-7777	Барбарис (конфеты),		кг	60 сут
		15,000		
		5,000	Не годен	
		5,000	Не годен	
		5,000	Не годен	
Арт-6666888	Белочка (конфеты),		кг	60 сут
		15,000		
		5,000	Не годен	
		7,000	Не годен	
		3,000	Не годен	
Арт-999999	Грильяж (конфеты),		кг	60 сут
		14,000		
		4,000	Не годен	
		8,000	Не годен	
		2,000	Не годен	
Арт-56789	Крупа "Геркулес",		кг	45 сут
		10,000		
		5,000	Не годен	
		5,000	Не годен	
Арт-89000	Крупа гречневая (весовая),		кг	45 сут
		15,000		
		10,000	Не годен	
		5,000	Не годен	
Арт-89000	Крупа гречневая (упак),		упак	45 сут
		10,000		
		10,000	Не годен	
Арт-7890000	Масло "Кремлевское",		шт	30 сут
		10,000		
		10,000	Не годен	
Арт-800000	Масло вологодское,		шт	30 сут
		10,000		
		10,000	Не годен	
Арт-789000	Масло деревенское,		пач	30 сут
		10,000		
		10,000	Не годен	
Арт-6666	Мишка (конфеты),		кг	60 сут
		15,000		
		5,000	Не годен	
		10,000	Не годен	
Арт-777788	Молоко "Домик в деревне" 1.5%,		шт	30 сут
		11,000		
		1,000	Не годен	
		10,000	Не годен	
Арт-980000	Молоко "Домик в деревне" 3.2%,		шт	60 сут
		9,000		

Рисунок 10 – Учет остатков товаров в 1С

Для поддержания запасов на складе при расширенном варианте обеспечивать потребности можно различными способами: покупка у поставщика; перемещение с одного склада на другой; сборка.

Четвертой программой, которая была включена в обзор, было приложение по Управлению ассортиментом от Эвотора (www.uchet-tovarov.evotor.ru). Это – облачное приложение для товарного учета от поставщика онлайн-касс Эвотор. Помогает работать с товарной номенклатурой на компьютере или смарт-терминале с автоматической синхронизацией данных. Контроль работы кассиров, печать ценников. Есть интеграция с Моим Складом. Соответствие требованиям законодательства, но за дополнительные модули – например, маркировку ЕГАИС – придется платить (рисунок 11).

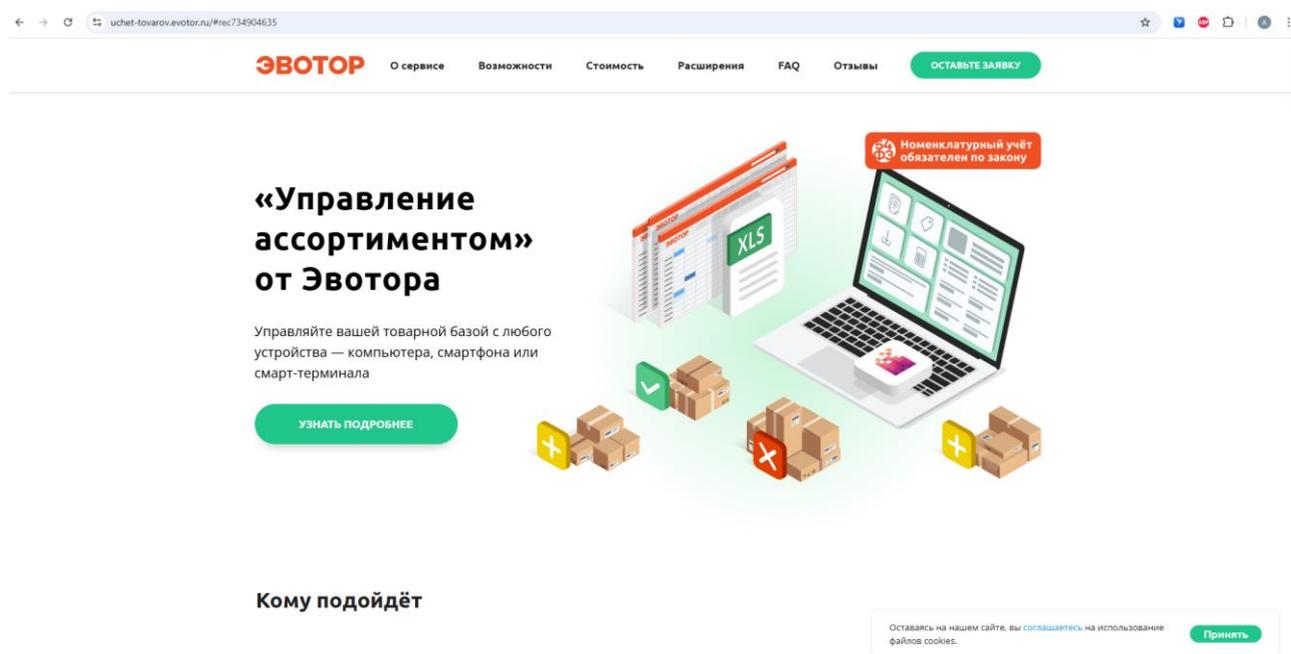


Рисунок 11 – Учет товаров от Эвотора

Пятой программой в нашем обзоре стал Антисклад (www.antisklad.ru), предлагающая все базовые функции для складского учета, необходимые для малого бизнеса. Из единого окна можно управлять несколькими складами и точками продаж. Есть детальная аналитика по прибыли, обороту, работе продавцов, поставщикам и клиентам. Простая программа для освоения для

новичков, поддерживающая работу с маркированным товаром, но платно. Нет мобильной версии и учета товаров на развес и розлив (рисунок 12).

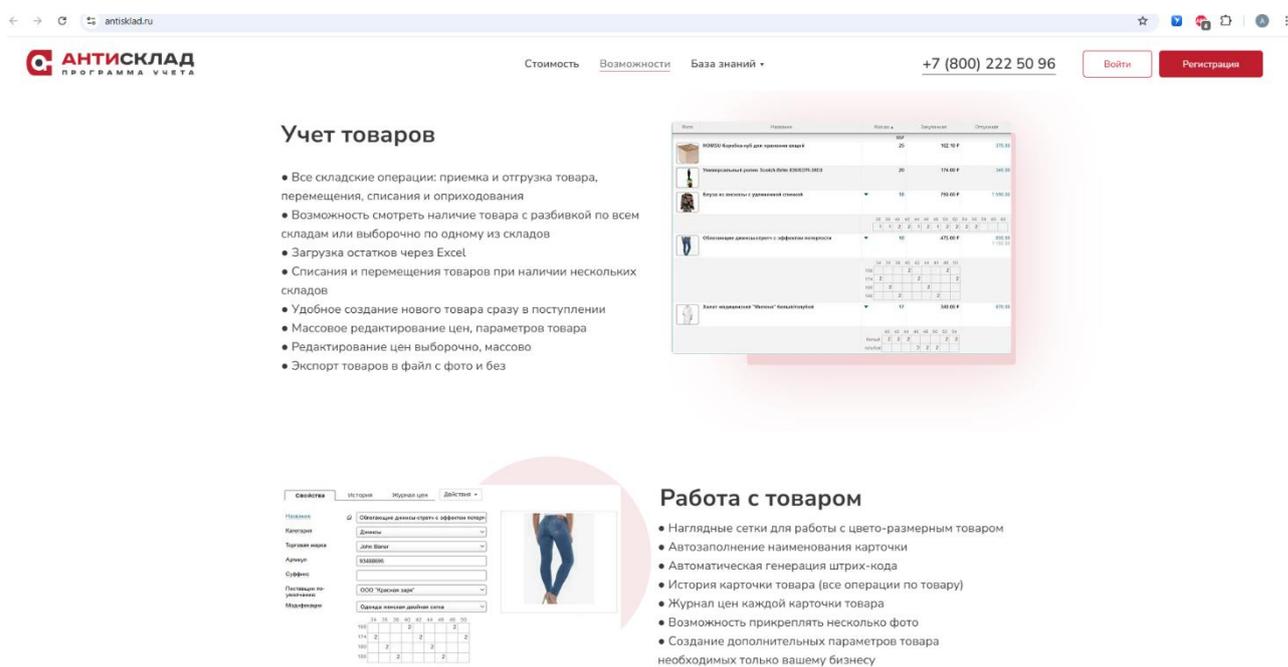


Рисунок 12 – Программа Антисклад

Шестая программа в обзоре – это Большая птица (www.bigbird.ru), складская программа для оптовой и розничной торговли, сферы услуг, интернет-магазинов. Позволяет делать списания и перемещения, контролировать остатки, проводить инвентаризацию. Подойдет для группы компаний, где применяются разные системы налогообложения. Есть интеграции с системами оплаты и кассовым оборудованием. Хотя обновления бывают редко.

Программа Большая птица позволяет вести журнал складских документов (рисунок 13) и производить инвентаризацию товаров (рисунок 14).

Учет	Дата	Номер	Вид документа	Фирма	Комментарий
<input checked="" type="checkbox"/>	09.12.2014 17:43	2	Списание	ИП Семенов Антон Иванович	Введено на основании инвентаризации № 1 от 09.12.2013
<input checked="" type="checkbox"/>	09.12.2014 17:42	2	Оприходование	ИП Семенов Антон Иванович	Введено на основании инвентаризации № 1 от 09.12.2013
<input checked="" type="checkbox"/>	09.12.2014 17:41	1	Списание	ИП Семенов Антон Иванович	
<input checked="" type="checkbox"/>	09.12.2014 17:41	1	Оприходование	ИП Семенов Антон Иванович	
<input type="checkbox"/>	09.12.2013 17:38	1	Инвентаризация	ИП Семенов Антон Иванович	
<input checked="" type="checkbox"/>	05.08.2013 08:00	3	Перемещение	ИП Семенов Антон Иванович	
<input checked="" type="checkbox"/>	02.08.2013 14:53	2	Перемещение	ИП Семенов Антон Иванович	
<input checked="" type="checkbox"/>	01.08.2013 12:57	1	Перемещение	ИП Семенов Антон Иванович	
<input checked="" type="checkbox"/>	01.02.2012 11:00	1	Перемещение	Техника-Комфорт ООО	

Рисунок 13 – Журнал складских документов

Инвентаризация № 1 от 09.12.2013 17:38:29 Автор: Семенов А.И.

Реквизиты документа | Печатная форма | Связанные документы | Файлы

Фирма: ИП Семенов Антон Иванович
 Склад: Склад (Савеловский) | Тип цен: Розничные

№	Номенклатура	Ед.	Коэф.	Учет. кол.	Факт. кол.	Отклонение	Цена	Учетная сумма	Сумма отклонения
85	Смартфон Apple iPhone 5 4" 32Gb White	шт.	1	1,000	1,000		35 988,00	35 988,00	
86	Смартфон Apple iPhone 5 4" 64Gb Black	шт.	1	1,000	1,000		39 588,00	39 588,00	
87	Смартфон Apple iPhone 5 4" 64Gb White	шт.	1	1,000	1,000		39 588,00	39 588,00	
88	Смартфон Explay Sky Plus 4" 4Gb	шт.	1	0,000	1,000	1	7 188,00	0,00	7 188,00
89	Смартфон HTC 8S 4" 4Gb Black	шт.	1	2,000	2,000		10 788,00	21 576,00	
90	Смартфон HTC 8S 4" 4Gb Blue	шт.	1	2,000	2,000		10 788,00	21 576,00	
91	Смартфон HTC 8X 4.3" 16Gb Black	шт.	1	2,000	2,000		15 588,00	31 176,00	
92	Смартфон HTC Desire C 3.5" 4Gb Black	шт.	1	2,000	2,000		6 588,00	13 176,00	
93	Смартфон HTC Desire C 3.5" 4Gb Red	шт.	1	2,000	2,000		6 588,00	13 176,00	
94	Смартфон HTC Desire C 3.5" 4Gb White	шт.	1	2,000	2,000		6 588,00	13 176,00	

Учетная сумма: 1 571 868,00 Сумма отклонения: 3 816,00

Рисунок 14 – Инвентаризация товаров

Позволяет вести остатки по складам (рисунок 15), делать отчеты по движениям по товару (рисунок 16), а также отчеты по партиям товаров (рисунок 17).

Остатки товаров: результаты

Печать Excel Обновить

Отчет по остаткам товаров

на 09.12.2014

Склад / Группа товаров / Товар	Ед.	Остаток
Склад: Магазин "Орехи и сухофрукты"		
Категория: ИП Семенов (орехи и сухофрукты)		
Категория: Орехи		
Грецкие орехи очищ. половинка	кг	1
Кешью жареный	кг	11
Кешью сырой	кг	1
Кунжут очищ.	кг	2
Миндаль золотой	кг	2
Фундук очищен.сырой	кг	7
Ядро абрикосовых косточек	кг	1
Категория: Прочее		
Кокосовая стружка Медиум 65%	кг	2
Категория: Сухофрукты		
Ананас палочки красные.нат,оран	кг	1
Ананас со вкусом дыни	кг	1
Арахис бланшированный	кг	12
Арахис жареный дробленый 2/4мм вакуум	кг	1

Рисунок 15 – Отчет по остаткам товаров в программе Большая птица

Движения товаров: результаты

Печать Excel Обновить

Отчет по движению товаров

за весь период

Склад / Товар	Ед.	Нач. остаток	Приход	Расход	Кон. остаток
Склад: Магазин "Орехи и сухофрукты"					
Ананас кубики 4-х цвет. 8/10мм					
Поступление товаров № 1 от 14.04.2014	кг		1	1	
Чек № 2 от 18.04.2014	кг	1			1
Ананас палочки 4-х цвет.					
Поступление товаров № 1 от 14.04.2014	кг		1		1
Чек № 4 от 18.04.2014	кг	1		1	
Ананас палочки красные.нат,оран					
Поступление товаров № 1 от 14.04.2014	кг		1		1
Ананас со вкусом дыни					
Поступление товаров № 1 от 14.04.2014	кг		1		1
Арахис бланшированный					
Поступление товаров № 1 от 14.04.2014	кг		16	4	12
Чек № 2 от 18.04.2014	кг	16		1	15
Чек № 4 от 18.04.2014	кг	15		3	12
Арахис жареный дробленый 2/4мм вакуум					
Поступление товаров № 1 от 14.04.2014	кг		1		1

Рисунок 16 – Отчет по движению товаров в программе Большая птица

Партии товаров: результаты

Печать Excel Обновить

Отчет по партиям товаров

за весь период

Фирма / Товар	Ед.	Нач. остаток		Приход		Расход		Кон. остаток	
		Кол-во	Себе-стоимость	Кол-во	Себе-стоимость	Кол-во	Себе-стоимость	Кол-во	Себе-стоимость
Манго натуральные листочки	кг			9	1 285,20			9	1 285,20
Поступление товаров № 1 от 14.04.2014	кг			9	1 285,20				
Миндаль жареный золотой	кг			1	251,60	1	251,60		
Поступление товаров № 1 от 14.04.2014	кг			1	251,60				
Чек № 3 от 18.04.2014	кг					1	251,60		
Миндаль золотой	кг			10	2 456,50	8	1 965,20	2	491,30
Поступление товаров № 1 от 14.04.2014	кг			10	2 456,50				
Чек № 3 от 18.04.2014	кг					1	245,65		
Чек № 5 от 18.04.2014	кг					7	1 719,55		
Миндаль резаный лепестки	кг			1	279,65	1	279,65		
Поступление товаров № 1 от 14.04.2014	кг			1	279,65				
Чек № 3 от 18.04.2014	кг					1	279,65		
Набор из СЗУ и АЗУ Depra Ultra для iPhone 5/iPod 30pin	шт.			1	1 030,00			1	1 030,00

Рисунок 17 – Складские операции в программе Большая птица

Седьмой программой в обзоре выступил Штрих-код Комбайн – приложение плюс программа, которая является мобильным приложением для Android и Windows, позволяющая смартфоном сканировать штрих-коды. Можно подключать несколько считывающих устройств. Программа для ПК при сканировании заносит коды в соответствующие списки. Есть интеграция с системой «Честный знак» и возможность кастомизации, но ввод некоторых данных возможен только вручную (рисунок 18).



Рисунок 18 – Мобильное приложение Штрих-код Комбайн

Восьмая программа обзора – LiteBox – облачное программное обеспечение для автоматизации мелкой розничной торговли: кассовый терминал + товароучетная система (www.litebox.ru) Разработано под малый бизнес, есть аналитические отчеты и управление закупками. Сервис обеспечивает безопасное хранение данных в облаке и возможность покупки, аренды и регистрации онлайн-кассы, однако требует немало доработок, чтобы подходить определенному виду бизнеса (рисунок 19).

ШК	Тип	Статус	Номер	Дата	Сумма розн. руб.	Сумма закуп. руб.	Дата последнего изменения	Комментарий
01625	Инвентаризация	Проверенный	1625	25.03.2021	15000.00	0.00	25.03.21 09:20:05	

№ п/п	↑ Наименование	Код	ШК	Ед. изм.	Остаток		Закупка, руб.		Продажа, руб.		Количество	
					Учетный	Факт	Цена	Сумма	Цена	Сумма		
1	Блокнот А6 (14.8 x 10.5 см)	19911	907321512 4039	шт	-8.000	50.000	58.000	0.00	0.00	100.00	5800.00	50
2	Калькулятор Assistant	19949		кг	26.000	30.000	4.000	0.00	0.00	300.00	1200.00	30.000
3	Тетрадь в клетку (48 листов)	19910	460300855 8328	шт	-1.000	50.000	51.000	0.00	0.00	20.00	1020.00	50

Рисунок 19 – Программное приложение Lite Box

Девятой программой в обзоре стала программа CloudShop (www.cloudshop.ru), которая также является и мобильное приложение для складского учета и торговли для небольших магазинов, которые работают по стандартной схеме покупки-продажи, и имеет необходимые функции для работы с отчетностью. В данной программе можно вести несколько магазинов, включая e-commerce и несколько складов. Данная программа имеет понятный интерфейс, но поддерживает не все кассовое оборудование и не имеет аналитики по кассирам и товарным группам (рисунок 20).

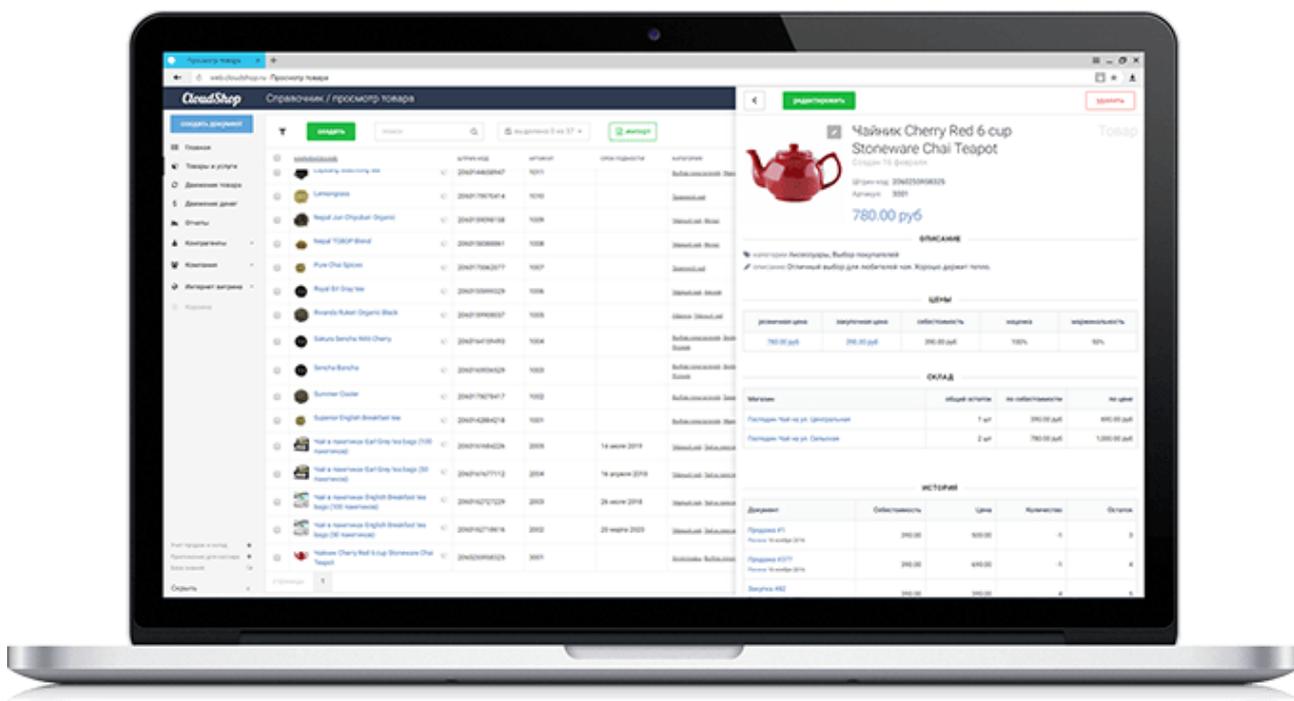


Рисунок 20 – Приложение CloudShop

И, наконец, десятой программой в обзоре стала программа Бухсофт: Торговля (www.buhsoft.ru), которая является программой для малого и среднего бизнеса с возможностью контролировать работу склада: перемещать продукцию от поставщика к покупателю, учитывать остатки на складе, оформлять первичные документы на перемещение. В программе заложены следующие функциональные возможности: ведение учета товарооборота; создание и печать бухгалтерских документов; поддержка кассового оборудования АТОЛ, ШТРИХ-М, ДРИМАКС; многопользовательский режим; автоматическая загрузка информации о товаре со сканера штрих-кода (рисунок 21).

Документ первичный

Покупка | Продажа

Дата документа: | Номер документа: | Номер телефона:

ИНН продавца: < - Ведите ИНН для добавления из интернета | Электронная почта:

Продавец:

ИНН покупателя: < - Ведите ИНН для добавления организации в интернете

Покупатель:

Введите наименование, количество и цену состава документа

	Наименование	Кольво	Ед.Изм	Цена	НДС	СуммаН	Сумма	Страна	ГТД
Справочник			шт		18%			Россия	
Справочник			шт		18%			Россия	
Справочник			шт		18%			Россия	
Справочник			шт		18%			Россия	
Справочник			шт		18%			Россия	
Справочник			шт		18%			Россия	
Справочник			шт		18%			Россия	
Справочник			шт		18%			Россия	
Справочник			шт		18%			Россия	
Справочник			шт		18%			Россия	

Печать (Торг-12, Счет-фактуру, Счет и пр.)

Сформировать через интернет | Задать вопрос

Рисунок 21 – Приложение Бухсофт: Торговля

Рассмотренные программные продукты были проанализированы по 12 ключевым параметрам, представленным в таблице 1.

Таблица 1 – Обзор 10 наиболее приемлемых программных продукта складского учета по 12 ключевым параметрам

Название	Подходит для			Поддерживает			Есть интеграции:			Простога освоения	Стабильность работы	Доступность технической поддержки
	Розницы	Общепита	производства	Маркировку	ЕГАИС	МРЦ	Маркет-плейса	Интернет-магазины	Кассовое оборудование			
Мой склад	*	*	*	*	*	*	*	*	*	*	*	*
Subtotal	*	*	-	*	*	-	-	*	*	*	*	*
1С: розница	*	*	-	-	-	-	*	*	*	-	*	*
Управление ассортиментом от Эвотора	*	*	-	*	*	-	*	*	*	*	*	*
Антисклад	*	-	-	*	-	-	*	*	*	*	*	*

продолжение таблицы 1

Большая птица	*	-	-	-	-	-	*	*	*	*	*	*
Штрих-код Комбайн	*	-	-	*	-	-	-	-	*	*	*	*
Lite box	*	-	-	*	*	-	-	*	*	-	*	*
Cloud shop	*	-	-	*	-	-	-	*	*	*	-	*
Бухсофт: торговля	*	-	-	*	-	-	-	-	*	*	*	-

Анализ аналогов программного обеспечения складского учета производственного участка был произведен с использованием ГОСТ Р ИСО/МЭК 9126–93 «Информационная технология. Оценка программной продукции. Характеристика качества и руководства по применению» [13] и ГОСТ 28195–99 «Оценка качества программных средств» [8]. По данным методикам ГОСТ ИСО/МЭК 9126-93 и ГОСТ 28195-99 был проведен сравнительный анализ (таблица 2). При этом, под практичностью подразумевалась применяемость приложения для производственного складского учета, а под эффективностью подразумевалась возможность применения производственного документооборота.

Таблица 2 – Сравнительный анализ показателей качества десяти избранных программных приложений с возможностью ведения складского учета

Показатели качества	Надежность	Практичность	Эффективность	Сопровождаемость	Мобильность	Наглядность	Удобство
Мой склад	5	4	1	5	5	3	3
Subtotal	3	1	1	4	4	5	3
1С: розница	5	1	1	3	3	5	2
Управление ассортиментом от Эвотора	4	1	1	4	3	3	3

продолжение таблицы 2

Антисклад	3	1	1	5	5	5	4
Большая птица	4	1	1	4	5	3	3
Штрих-код Комбайн	4	1	1	3	3	3	2
Lite box	3	1	1	3	5	5	4
Cloud shop	2	1	1	2	5	3	3
Бухсофт: торговля	3	1	1	3	4	4	4

Таким образом, сравнительный анализ предлагаемых на рынке программных приложений, имеющих возможности ведения складского учета, показал, что, видимо из-за конъюнктуры рынка, все предлагаемые продукты делают акцент на обслуживание розничной торговли и общепита. Рассмотренные продукты имеют удобные функции работы на мобильных устройствах и облачные технологии, имеют интеграции с кассовым оборудованием и маркировкой в ЕГАИС, однако не было найдено ни одного программного продукта, который бы предлагал функциональные возможности для ведения именно производственного складского учета с соответствующим документооборотом. Поэтому предлагаемый в данной работе программный продукт будет ориентирован именно под потребности производственного участка ООО «АВЭЛЛА» и, в этом плане, необходима разработка соответствующих функциональных требований к данному программному обеспечению.

1.3 Разработка требований к программному обеспечению складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

С целью разработки требований к программному обеспечению складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА», рассмотрим основные стандарты и методологии разработки

требований для того, чтобы выбрать оптимальный подход к решению данной задачи.

Прежде всего, стоит отметить классификацию требований к программному обеспечению, предложенную Карлом Вигерсом [6], которая включает три уровня:

- бизнес-требования (business requirements). Обычно формулируются топ-менеджерами или акционерами предприятия. Пример: система должна сократить срок оборачиваемости обрабатываемых на предприятии заказов в три раза;
- требования пользователей (user requirements). Например: система должна представлять диалоговые средства для ввода исчерпывающей информации о заказе, последующей фиксации информации в базе данных и маршрутизации информации о заказе к сотруднику, отвечающему за его планирование и исполнение;
- нефункциональные требования. Регламентируют внутренние и внешние условия или атрибуты функционирования системы. К ним относятся, например, внешние интерфейсы, атрибуты качества и ограничения.

Следующая классификация требований в SWEBOOK [25] включает следующие виды:

- требования к продукту и процессу. Параметры, относящиеся к продукту или процессу его создания;
- функциональные и нефункциональные требования. Функциональные описывают функции, которые выполняет программное обеспечение, а нефункциональные накладывают определённые ограничения;
- независимые свойства. Требования, которые не могут быть адресованы к одному из компонентов системы, а проявляются при взаимодействии;

- системные или программные требования. Относятся к системе в целом или к программной составляющей.

Международный стандарт IEEE Std 830-1998 “Recommended practice for software requirements specifications” [23] – это рекомендуемая методика составления спецификаций требований к программному обеспечению (Software Requirements Specification). Она описывает критерии проверки, качественно составленной SRS, её части и шаблоны.

Стандарт содержит два шаблона спецификации требований:

- System requirements specification (SyRS). Определяет технические требования для выбранной системы и удобства взаимодействия предполагаемой системы и человека. Определяет высокоуровневые требования к системе с точки зрения предметной области, а также информацию об общей цели системы, её целевой среде и ограничениях, допущениях и нефункциональных требованиях;
- Software requirements specification (SRS). Это спецификация требований для определённого программного изделия, программы или набора программ (продукт), которые выполняют определённые функции в конкретном окружении.

Межгосударственный стандарт ГОСТ 34.201-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем» [11] распространяется на автоматизированные системы (АС), используемые в различных сферах деятельности (управление, исследования, проектирование и т. п.), и устанавливает требования к видам, наименованию, комплектности и обозначению документов, разрабатываемых на стадиях создания АС [15].

Согласно рассмотренным стандартам, а именно К. Вигерса, SWEBOOK, IEEE Std 830-1998, ГОСТ 34.201-2020, были разработаны требования к программному приложению складского учета производственного участка жидких моющих средство ООО «АВЭЛЛА». Прежде всего, перечень

заинтересованных сторон в автоматизации складского учета предприятия, может включать следующие группы персонала:

- производственный персонал: отвечают за планирование и контроль производственных процессов и запасов;
- работники склада: непосредственно занимаются приемом, хранением и отгрузкой сырья и готовой продукции;
- отдел снабжения: отвечает за приобретение сырья и материалов;
- отдел по работе с клиентами: заинтересован в наличии готовой продукции для полного выполнения заказов клиентов;
- финансовый отдел: следит за затратами на складские операции и контролирует бюджет;
- отдел контроля качества: обеспечивает соответствие продукции стандартам качества и безопасности;
- отдел АСУ: отвечает за автоматизацию складского учета и поддержку программного обеспечения;
- отдел логистики: планирует и организует доставку сырья и готовой продукции;
- отдел маркетинга: анализирует спрос и помогает в планировании производства;
- руководство организацией: принимает стратегические решения, основываясь на данных о запасах и производительности.

Для каждой заинтересованной стороны были разработаны анкеты, с целью выявления потребностей.

Анкета для производственного персонала была представлена следующими вопросами:

- как вы оцениваете текущие процессы управления запасами на производственном участке;
- с какими основными проблемами вы сталкиваетесь при планировании производства;

- как вы взаимодействуете с другими отделами, такими как закупки и продажи;
- как вы оцениваете качество продукции, производимой на нашем участке;
- какие изменения вы хотели бы видеть в процессах управления производством.

На предложенные вопросы были даны следующие ответы:

- в целом, процессы достаточно эффективны, но есть необходимость в улучшении автоматизации для сокращения времени на обработку данных;
- основные проблемы связаны с недостаточной точностью прогнозов спроса и колебаниями в поставках сырья;
- регулярно проводятся совещания для обсуждения текущих запасов и потребностей, что помогает быть на одной волне и предотвращать дефицит;
- качество продукции на высоком уровне, но мы всегда ищем возможности для улучшения и снижения брака;
- хотелось бы внедрить более продвинутые системы мониторинга и отчетности, чтобы улучшить видимость производственных показателей в реальном времени.

Аналогичная анкета была предложена для ответов на сопутствующие вопросы складскими работниками:

- как вы оцениваете текущие процессы приема и хранения товаров на складе;
- с какими основными трудностями вы сталкиваетесь в процессе отгрузки продукции;
- как вы оцениваете уровень автоматизации складских операций;
- как вы взаимодействуете с другими отделами, такими как производство и логистика;

- какие изменения вы хотели бы видеть в условиях работы на складе.

На данные вопросы были получены следующие ответы:

- процессы в целом организованы, но иногда возникают задержки при приеме товаров из-за недостатка персонала;
- основные трудности связаны с нехваткой информации о заказах и неопределенностью в графиках доставки;
- автоматизация на среднем уровне и поэтому некоторые процессы можно улучшить с помощью новых технологий, чтобы сократить время на выполнение задач;
- поддерживается регулярное общение с производством, чтобы знать о текущих запасах, и с отделом логистики для координации отгрузок;
- хотелось бы улучшить условия труда, например, обеспечить лучшее освещение и механизмы для подъема тяжестей.

Также была разработана анкета для отдела снабжения, включающая соответствующие вопросы:

- как вы оцениваете текущие процессы выбора поставщиков;
- с какими основными трудностями вы сталкиваетесь при ведении переговоров с поставщиками;
- как вы оцениваете уровень автоматизации процессов закупок;
- как вы взаимодействуете с другими отделами, такими как производство и логистика;
- какие изменения вы хотели бы видеть в условиях работы отдела снабжения.

На рассмотренные вопросы были даны следующие типичные ответы:

- процессы выбора поставщиков эффективны, но иногда не хватает актуальной информации о рынке, что затрудняет принятие решений;
- хотелось бы улучшить доступ к обучающим материалам по новым технологиям ведения переговоров для повышения квалификации;

- основные трудности связаны с несоответствием ценовых предложений и сроков поставки нашим ожиданиям, что приводит к задержкам в производственном процессе;
- автоматизация в отделе на среднем уровне, поэтому есть возможности для внедрения электронных систем закупок для ускорения процесса;
- с производством происходит регулярное общение для понимания их потребностей и с отделом логистики для координации сроков поставок.

Следующая анкета была разработана для отдела по работе с клиентами и включала следующие типовые вопросы:

- как вы оцениваете текущие стратегии продаж нашей компании;
- с какими основными трудностями вы сталкиваетесь при работе с клиентами;
- как вы оцениваете уровень взаимодействия с другими отделами;
- как вы оцениваете уровень автоматизации процессов продаж;
- какие изменения вы хотели бы видеть в условиях работы вашего отдела.

На приведенные вопросы были даны следующие типичные ответы:

- автоматизация на среднем уровне, поскольку вместе с используемой CRM-системой можно было бы внедрить аналитические инструменты для отслеживания эффективности;
- стратегии продаж достаточно эффективны, хотя не хватает гибкости для адаптации к изменениям на рынке;
- основные трудности связаны с возражениями клиентов по поводу цены и недостатком информации о продукте;
- хотелось бы больше возможностей для обучения и повышения квалификации, а также улучшения мотивационных программ для сотрудников;

- взаимодействие с другими отделами хорошее, но иногда возникают недопонимания, особенно по поводу маркетинговых акций и новых продуктов.

Также было проведено анкетирование финансового отдела со следующим набором вопросов:

с какими основными трудностями вы сталкиваетесь при составлении финансовых отчетов;

- как вы оцениваете текущие финансовые процессы в вашей компании;
- как вы оцениваете уровень взаимодействия с другими отделами, такими как бухгалтерия и отдел по работе с клиентами;
- как вы оцениваете уровень автоматизации финансовых процессов;
- какие изменения вы хотели бы видеть в условиях работы финансового отдела.

На данные вопросы были даны следующие типовые ответы:

- основные трудности связаны с недостатком данных из других отделов и несоответствием сроков предоставления информации, затрудняющие своевременную подготовку финансовых отчетов;
- финансовые процессы организованы нормально, однако есть необходимость для более эффективной автоматизации с целью снижения ручного труда;
- взаимодействие с другими отделами в целом неплохое, но иногда возникают непонимания по поводу финансовых планов и бюджетов;
- автоматизация находится на среднем уровне, используется несколько программ для учета и отчетности, но есть возможности для улучшения, особенно в области прогнозирования;
- хотелось бы больше возможностей для повышения мотивации сотрудников, повышения их вовлеченности, чего можно было бы достичь повышением возможностей для повышения квалификации.

Аналогичные анкеты были обработаны для отдела контроля качества, включающие вопросы:

- с какими основными трудностями вы сталкиваетесь при проведении проверок качества;
- как вы оцениваете текущие процессы контроля качества в нашей компании;
- как вы оцениваете уровень автоматизации процессов контроля качества;
- как вы оцениваете уровень взаимодействия с другими отделами, с производством и маркетингом;
- какие изменения вы хотели бы видеть в условиях работы отдела контроля качества.

На данные вопросы были даны следующие ответы:

- хотелось бы больше возможностей для повышения квалификации и обучения, а также улучшение системы обратной связи, чтобы сотрудники могли делиться своими идеями и предложениями;
- процессы контроля качества организованы хорошо, однако не хватает четких стандартов оценки;
- основные трудности связаны с нехваткой времени на проверки и недостаточной информацией о новых продуктах, затрудняя оценку их качества;
- автоматизация на среднем уровне, то есть используется несколько инструментов для сбора данных, но есть возможности для улучшения, особенно в области отчетности;
- взаимодействие с другими отделами неплохое, несмотря на то что возникают недопонимания по поводу требований к качеству и сроков выполнения.

Для отдела автоматизированных систем управления были предложены следующие вопросы анкеты:

- как вы оцениваете уровень взаимодействия с другими отделами;

- как вы оцениваете текущие процессы разработки и внедрения программного обеспечения в вашей компании;
- с какими основными трудностями вы сталкиваетесь при решении технических проблем;
- как вы оцениваете уровень автоматизации в отделе АСУ;
- какие изменения вы хотели бы видеть в условиях работы отдела АСУ.

На представленные вопросы были получены следующие типичные ответы:

- основные трудности связаны с нехваткой времени на диагностику проблем и недостаточной поддержкой со стороны других отделов;
- процессы разработки организованы хорошо, но вместе с тем не хватает четкой документации, дающей четкое понимание требований;
- хотелось бы улучшить систему обмена знаниями внутри команды, и конечно больше возможностей для повышения квалификации и обучения;
- взаимодействие с другими отделами неплохое, но иногда возникают сложности из-за недостатка технических знаний у сотрудников других отделов;
- автоматизация процессов на среднем уровне, поскольку используются несколько инструментов для управления проектами, и в то же время есть возможности для улучшения, например, тестирования программного обеспечения.

Далее, была разработана анкета для отдела логистики, включающая следующие вопросы:

- как вы оцениваете текущие процессы управления цепочкой поставок в нашей компании;
- с какими основными трудностями вы сталкиваетесь при планировании поставок;

- как вы оцениваете уровень взаимодействия с другими подразделениями, такими как продажи и производство;
- как вы оцениваете уровень автоматизации процессов в отделе;
- какие изменения вы хотели бы видеть в условиях работы отдела логистики.

И на данные вопросы были получены следующие ответы:

- процессы управления цепочкой поставок в целом эффективны, но иногда возникают задержки из-за недостаточной координации с поставщиками;
- основные трудности связаны с изменчивостью спроса и недостаточной информацией о наличии товаров у поставщиков;
- взаимодействие с другими отделами хорошее, но иногда возникают недопонимания по поводу сроков поставок и запасов;
- автоматизация процессов на среднем уровне, а именно используется несколько систем для отслеживания поставок, но есть возможности для улучшения в области отчётности и анализа данных;
- хотелось бы больше возможностей для обучения и повышения квалификации, а также улучшения системы управления запасами, чтобы минимизировать издержки.

Следующая анкета была разработана для отдела маркетинга со следующими типовыми вопросами:

- какие изменения вы хотели бы видеть в условиях работы отдела маркетинга;
- как вы оцениваете использование аналитических инструментов для оценки результатов маркетинга;
- как вы оцениваете уровень сотрудничества с другими отделами, такими как продажи и производство;
- с какими основными трудностями вы сталкиваетесь при реализации маркетинговых кампаний;

- как вы оцениваете эффективность текущих маркетинговых стратегий нашей компании.

На представленные вопросы были даны следующие типичные ответы:

- основные трудности связаны с недостатком бюджета и ограниченными ресурсами, что затрудняет реализацию более масштабных проектов;
- сотрудничество с другими отделами в целом хорошее, хотя возникают периодически проблемы с коммуникацией, особенно когда речь идет о сроках запуска новых продуктов;
- в целом, маркетинговые стратегии работают хорошо, но могли бы улучшить онлайн-присутствие и адаптировать контент под разные целевые аудитории;
- используется несколько аналитических инструментов, но иногда не хватает времени на их полное освоение и интерпретацию данных для принятия решений;
- хотелось бы больше возможностей для креативного подхода в проектах и улучшение коммуникации внутри команды для более эффективного обмена идеями.

И, наконец, было проведено анкетирование руководства предприятия, которое включало рассмотрение следующих типовых вопросов:

- как вы оцениваете текущую стратегию развития предприятия;
- какие изменения вы считаете необходимыми для повышения эффективности работы компании;
- как вы видите будущее организации на ближайшие три – пять лет;
- с какими основными вызовами сталкивается компания в настоящее время;
- как вы оцениваете уровень взаимодействия между различными отделами.

На представленные вопросы руководством компании были даны следующие ответы:

- стратегия в целом эффективна, но необходимо уделить больше внимания инновациям и адаптации к изменениям на рынке;
- взаимодействие между отделами в целом хорошее, но иногда возникают проблемы с координацией;
- необходимо улучшить процессы принятия решений и внедрить более современные инструменты для анализа данных, чтобы лучше понимать потребности клиентов;
- в ближайшие 3–5 лет видится компания, которая активно развивается, внедряются новые технологии и расширяется присутствие на международных рынках;
- основные вызовы связаны с увеличением конкуренции и изменениями в потребительских предпочтениях.

На основании анализа данных анкет были выявлены бизнес-потребности ООО «АВЭЛЛА» в плане складского учета производственного участка жидких моющих средств, которые были преобразованы в бизнес-требования, а затем в функциональные требования, представленные в таблице 3.

Таблица 3 – Преобразование бизнес-потребностей в бизнес-требования и функциональные требования складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Бизнес-потребности	Бизнес-требования	Функциональные требования
Необходимо внедрить более продвинутые системы мониторинга и отчетности, чтобы улучшить видимость производственных показателей	Сделать производственные показатели складского учета производственного участка более видимыми для менеджеров по производству	Внедрить отчет об отгруженном сырье и материалах в производство жидких моющих средств

продолжение таблицы 3

Трудности связаны с нехваткой информации о заказах и неопределенностью в графиках доставки	Обеспечить информацию о заказах для склада производственного участка жидких моющих средств для отдела закупок	Разработать отчет о заказах сырья и материалах на склад производственного участка жидких моющих средств
Иногда возникают задержки при приеме товаров из-за недостатка персонала	Автоматизировать процесс приемки товара на склад производственного участка жидких моющих средств для складских работников	Сделать автоматизированный учет приемки сырья и материалов на склад производственного участка жидких моющих средств
Необходимо внедрение аналитических инструментов для отслеживания эффективности	Внедрить аналитический инструмент для отслеживания работы склада производственного участка для финансового отдела	Обеспечить обратную ведомость складского учета производственного участка жидких моющих средств
Трудности связаны с недостатком данных из других отделов и несоответствием сроков предоставления информации	Предоставлять отделу продаж информацию об отгружаемых товарах	Подготовить отчет об отгруженных товарах потребителям жидких моющих средств
Трудности связаны с нехваткой времени на проверки и недостаточной информацией о новых продуктах	Предоставить отделу контроля качества информацию о произведенных продуктах	Внедрить отчет о произведенной продукции и принятой на склад производственного участка жидких моющих средств
Трудности связаны с нехваткой времени на диагностику проблем	Разработать форму коммуникации с IT-отделом	Сделать форму отправки сообщений IT-отделу о возникающих трудностях эксплуатации системы

продолжение таблицы 3

Иногда возникают недопонимания по поводу сроков поставок и запасов	Предоставить отделу логистики информацию о запасах сырья и материалах на складе производственного участка жидких моющих средств	Внедрить отчет об остатках сырья и материалов склада производственного участка жидких моющих средств
Иногда возникают проблемы с коммуникацией, особенно когда речь идет о сроках запуска новых продуктов	Обеспечить отдел маркетинга информацией о новых продуктах	Разработать отчет о товарах с пометкой «новинки» для анализа отделом маркетинга
Необходимо уделить больше внимания инновациям и адаптации к изменениям на рынке	Обеспечить руководство компании обзорной информацией о потоках сырья, материалов, продуктов и товаров производственного участка жидких моющих средств	Внедрить общую оборотную ведомость складского учета производственного участка жидких моющих средств в разрезе поступления сырья и материалов, передачи их в производство, поступления готовой продукции на склад, и отгрузки товаров покупателям

Далее, определены системные требования к программному обеспечению для складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»:

- возможность создания задач, установки сроков и назначения ответственных;
- инструменты для отслеживания прогресса и управления ресурсами;
- управление контактами и клиентскими данными;
- анализ и отчетность по поставкам и отгрузкам;

- возможность интеграции с другими системами и инструментами, используемыми в компании ООО «АВЭЛЛА».

Таким образом, на основании описания и анализа предметной области – деятельности склада производственного участка жидких моющих средств ООО «АВЭЛЛА» - позволившая описать основные направления деятельности складского учета, а также на основании обзора и анализа аналогов программного обеспечения, были разработаны требования к предлагаемому программному обеспечению. Функциональные и системные требования к программному обеспечению складского учета производственного участка жидки моющих средств ООО «АВЭЛЛА» заключаются в возможности интеграции деятельности основных подразделений компании в работе склада, а именно создание ряда отчетов, которые позволяют отслеживать определенную информацию для каждого подразделения. Данный набор требований позволяет перейти к следующему концептуальному этапу разработки – к проектированию программного обеспечения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА».

Глава 2 Проектирование программного обеспечения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

2.1 Выбор методологии проектирования программного обеспечения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Описание и анализ характеристик предметной области автоматизации рабочих процессов ООО «АВЭЛЛА» и дальнейший анализ бизнес-потребностей, трансформированных в бизнес-требования, а затем в набор функциональных и системных требований к программному обеспечению складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА», позволили выбрать методологию логического проектирования UML [15], [20].

Системные требования к программному обеспечению были обобщены в диаграмме компонентов деятельности склада производственного участка жидких моющих средств ООО «АВЭЛЛА», представленной на рисунке 22.

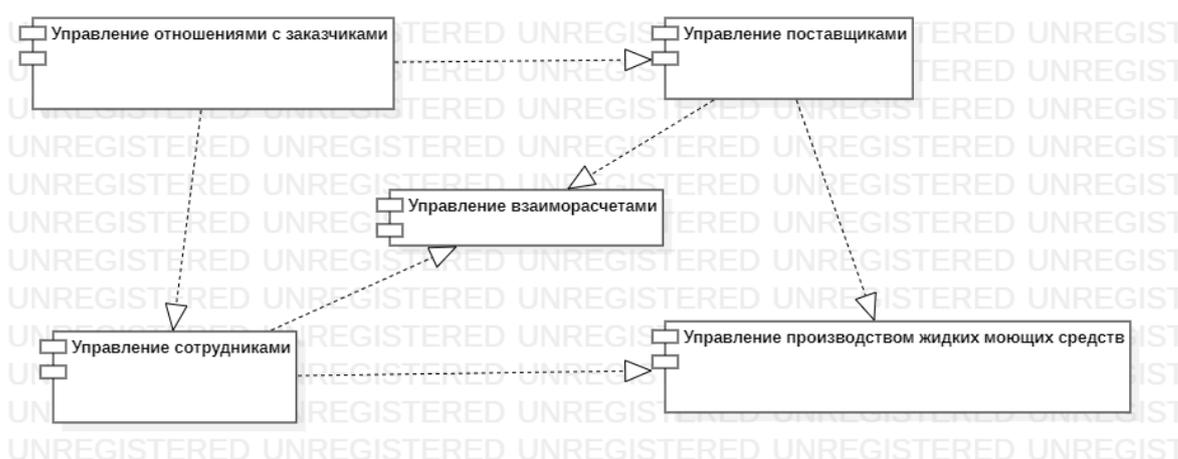


Рисунок 22 – Диаграмма компонентов деятельности склада производственного участка жидких моющих средств ООО «АВЭЛЛА»

На основании диаграммы компонентов деятельности склада производственного участка жидких моющих средств ООО «АВЭЛЛА», в рамках указанной методологии, была разработана диаграмма вариантов использования, представленная на рисунке 23.

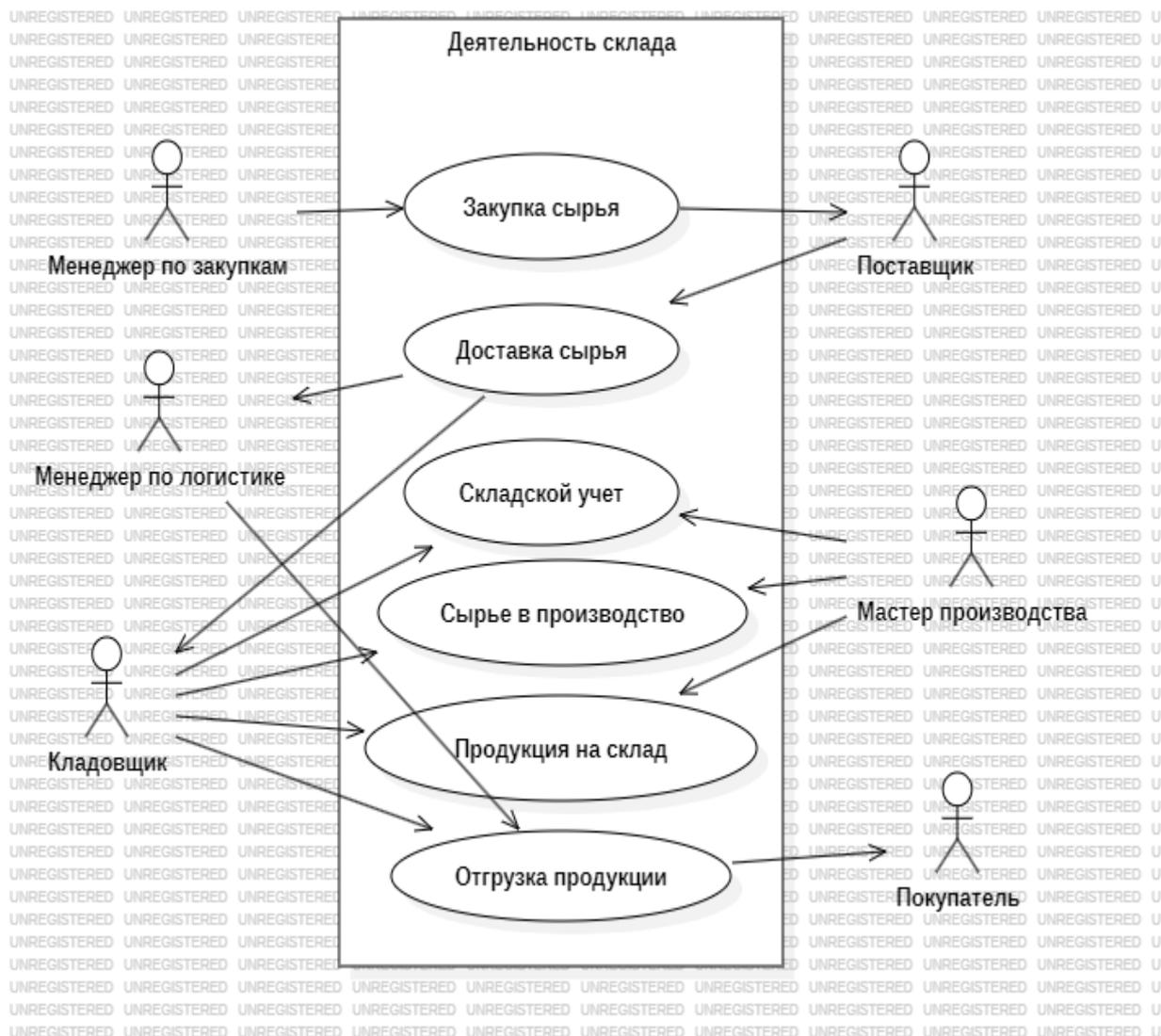


Рисунок 23 – Диаграмма вариантов использования деятельности склада производственного участка жидких моющих средств ООО «АВЭЛЛА»

Тем не менее, как было выяснено на этапе анализа бизнес-процессов и бизнес-требований Компании ООО «АВЭЛЛА», трансформированных в функциональные и системные требования, количество подразделений и нормативных требований было увеличено, что выражает более гибкую и

универсальную организационную структуру ООО «АВЭЛЛА». В основном, в качестве функциональных требований, выступило совершенствование информационных потоков для лучшего понимания бизнес-процессов и более развитых взаимодействий заинтересованных сторон бизнес-коммуникаций. Более конкретно, данные требования выразились в наличии соответствующих отчетов и форм обратной связи, которые и было решено разработать на основе существующего программного приложения в качестве его дальнейшего совершенствования и в качестве более продвинутой модели бизнес-процесса ТО-ВЕ.

Таким образом, объектом автоматизации был выбран складской учет производственного участка жидких моющих средств ООО «АВЭЛЛА» с целью его усовершенствования для удовлетворения функциональных и системных требований. Поскольку складской учет организован правильно, однако текущее программное решение не имеет достаточных средств для решения выявленных функциональных и системных требований, необходимо заметить следующее. Бизнес-процессы «AS-IS» фактически построены в виде функциональной модели «Складской учет», на основе этой же модели возможно усовершенствование ее до модели «ТО-ВЕ». И на основе такого методологического решения необходимо разработать новое программное решение складского учета. Такое программное решение предлагается реализовать средствами объектно-ориентированного программирования на платформе C#. А для теоретического обоснования программного продукта предлагается построить диаграмму классов, на основе которой уже затем разработать блок-схему программы, которая позволит написать непосредственно программный код и протестировать его. В этом плане, предлагается провести логическое моделирование программного обеспечения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА».

2.2 Логическое моделирование программного обеспечения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Необходимость совершенствования программного обеспечения складского учета производственного участка жидких моющих средств с учетом выявленных функциональных и системных требований, позволила логически смоделировать более продвинутую версию данного программного обеспечения, выраженную в разработанных диаграммах последовательности.

Логическая модель складского учета представляется двумя основными рабочими процессами – поступление сырья и материалов и отгрузка готовой продукции. Данные рабочие процессы, в свою очередь, аналогичны, но связями с разными бизнес-процессами. Поступление сырья и материалов связано с оприходованием на подразделение склада сырья и материалов производственного участка жидких моющих средств ООО «АВЭЛЛА» от поставщиков, а отпуск готовой продукции связан с отгрузкой покупателям. Поэтому общая логическая модель складского учета может быть представлена двумя аналогичными диаграммами последовательности.

Диаграмма последовательности поступления сырья и материалов на склад производственного участка ООО «АВЭЛЛА» представляется такими рабочими процессами, как добавление нового наименования, его изменение, удаление и выбор. Поэтому диаграмма последовательности представлена следующими процессами. Кладовщик осуществляет приемку сырья и материалов, которая может выразиться в выборе существующего наименования, либо введении нового наименования. Также выбирается поставщик, от которого приходится данное сырье или материалы. Кладовщик отправляет запрос в базу данных и, либо вводит новое наименование, либо выбирает существующее. При необходимости кладовщик может либо изменить данные по сырью или материалу, либо удалить его. Данный бизнес-процесс представлен на диаграмме последовательности ввода сырья и

материалов склада производственного участка жидких моющих средств ООО «АВЭЛЛА» на рисунке 24.

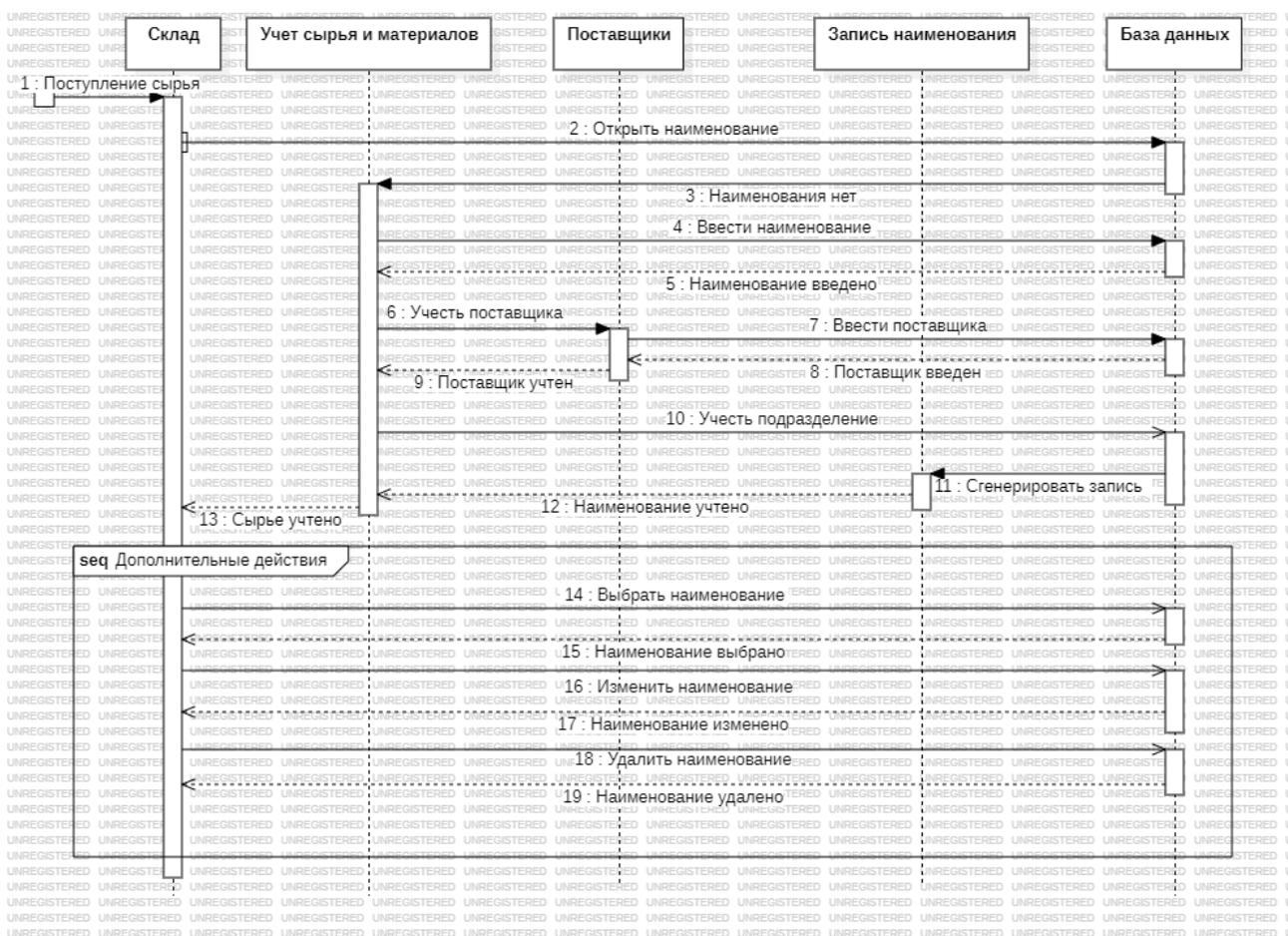


Рисунок 24 – Диаграмма последовательности оприходования сырья и материалов складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Диаграмма последовательности отгрузки готовой продукции со склада производственного участка ООО «АВЭЛЛА» представляется такими рабочими процессами, как добавление нового наименования, его изменение, удаление и выбор. Поэтому диаграмма последовательности отгрузки готовой продукции представлена следующими процессами. Кладовщик осуществляет отпуск готовой продукции, которая может выразиться в выборе существующего наименования, либо введении нового наименования. Также выбирается покупатель, на которого учитывается данная готовая продукция.

Кладовщик отправляет запрос в базу данных и, либо вводит новое наименование, либо выбирает существующее. При необходимости кладовщик может либо изменить данные по отпускаемой продукции, либо удалить ее. Данный бизнес-процесс представлен на диаграмме последовательности отгрузки готовой продукции склада производственного участка жидких моющих средств ООО «АВЭЛЛА» на рисунке 25.

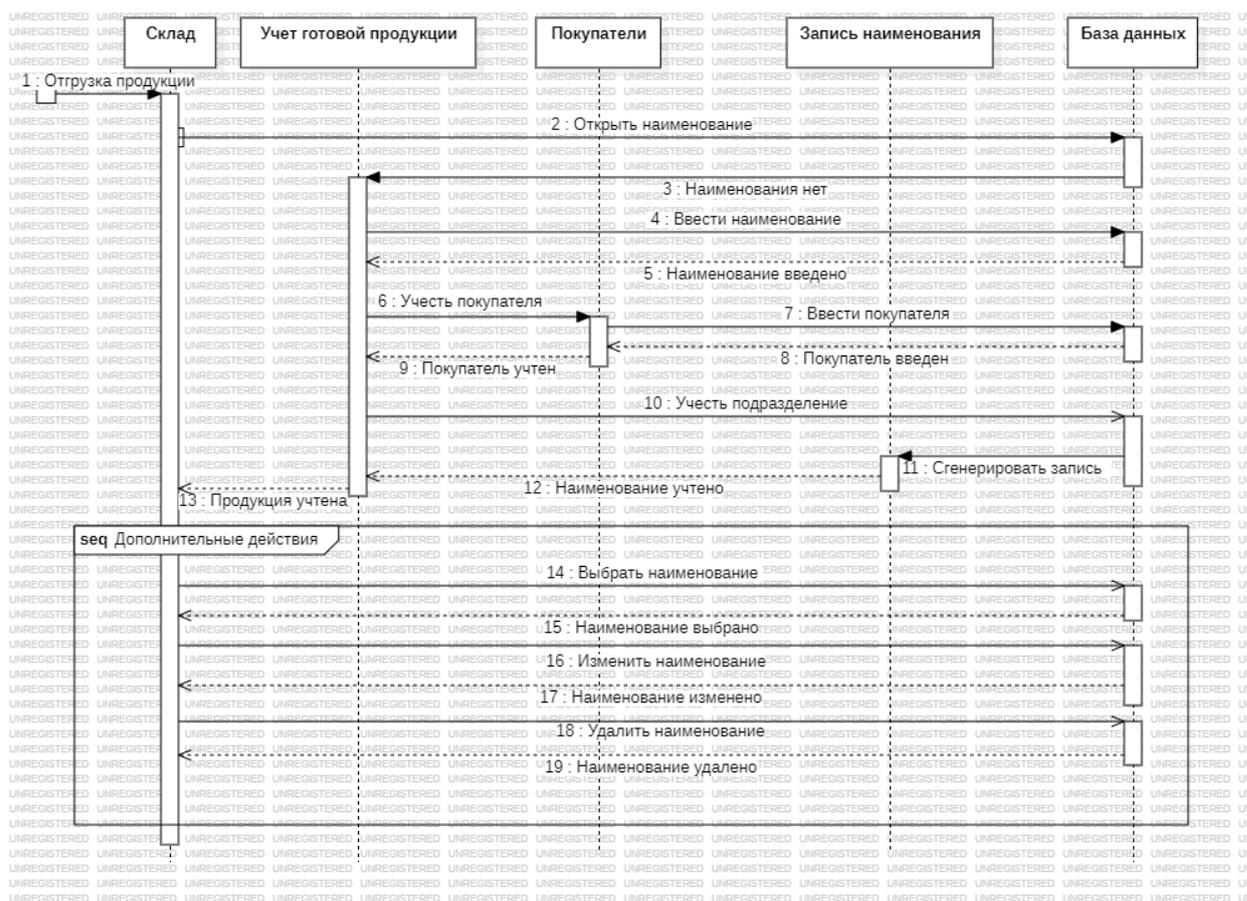


Рисунок 25 – Диаграмма последовательности отгрузки готовой продукции складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Предлагаемое решение бизнес-процесса модели Как должно быть (TO-BE) было выстроено на системе классов, управляемых методами, средствами программной среды C#. В этом плане было предложено выстроить информационно-управляющие потоки складского учета классами C#, которые

будут более универсальны и гибки в управлении. Были предложены следующие классы:

- обеспечение базовых компонентов системы: Suppliers() для поставщиков; Buyers() для покупателей; Departments() для точек учета; Income() для учета входящих сырья и материалов; Outcome() для учета отгружаемой готовой продукции; BalanceSheet() для предоставления общего отчета;
- обеспечение информационными потоками основных участников системы – IncomeProduce() для учета передачи сырья и материалов в производство; OutcomeProduct() для учета передаваемой из производство на склад готовой продукции; BalanceIncome() для учета остатков сырья; BalanceOutcome() для учета остатков готовой продукции; Feedback() для передачи обратной связи.

С учетом данных предложений была разработана диаграмма «ТО-ВЕ». Данная диаграмма позволит построить процесс моделирования данных системы складского учета.

2.3 Моделирование данных системы складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

С помощью программного продукта StarUML получаем следующий набор и диаграмму классов [4], [5]. Поскольку складской учет производственного участка разделен на получение сырья и отгрузку готовой продукции, то и организация программного продукта строилась на данных основных процессах [16]. Был составлена следующая система классов.

Класс Departments включает информацию о подразделениях, которые задействованы в перемещении материальных ценностей от поступления сырья на склад сырья и материалов, через передачу их в производство, передачу из производства на склад готовой продукции, и отгрузку готовой продукции со

склада готовой продукции покупателям. Поэтому в данном классе выделены следующие переменные: наименование отдела (DepartmentName:string), идентификационный номер отдела (DepartmentID:int) и ответственное лицо (Responsble:string). В данном классе использованы методы добавления информации (AddInfo), показа информации (ShowInfo), изменения информации (ChangeInfo) и учета информации о подразделениях (Department). Данный класс связан зависимостью с классом Оборотно-сальдовой ведомости, поскольку изменения в наборе подразделений отражаются в перемещениях материальных ценностей между подразделениями.

Класс Suppliers включает информацию о поставщиках сырья и материалов, включая переменные названия поставщика (SupplierName:string), идентификатора поставщика (SupplierID:int) и телефона поставщика (PhoneNumber:int). Данный класс также включает методы добавления, изменения и показа информации, а также метод учета информации о поставщиках (Supplier). Данный класс связан агрегированием с классом Departments, поскольку информация о поставщиках агрегируется в информации о подразделениях

Класс IncomeMaterial включает информацию о поставляемых материалах и сырье и состоит из таких переменных как Наименование материала (MaterialName:string), количество материала (MaterialQuantity:int), цены материала (materialPrice:float) и, соответственно, организован классами добавления, показа и изменения информации, а также методов учета информации о поставляемых материалах (IncomeMaterial). Данный класс связан ассоциацией с классом поставщиков один ко многим, поскольку один поставщик может поставлять множество материалов, а также связан агрегацией, так как поступающее сырье учитывается либо на складе сырья и материалов, либо расходуется в производство, а также связан зависимостью с классом оборотной ведомости, так как изменения в номенклатуре и количестве сырья отражается в складских оборотах.

Класс Buyers включает информацию о покупателях готовой продукции и включает переменные наименования покупателя (BuyerName:string), идентификационный номер покупателя (BuyerID:int), телефонный номер покупателя (PhoneNumber:int). Данный класс включает методы добавления, показа, изменения информации и метод учета покупателей (Buyers). Он связан агрегацией с классом подразделений и зависимостью с классом оборотно-сальдовой ведомости.

Класс OutcomeProduct включает информацию о готовой продукции и включает переменные наименования продукции (ProductName:string), количества (ProductQuantity:int), цены продукции (ProductPrice:float). Данный класс организован методами добавления, изменения, показа информации и метода учета готовой продукции (OutcomeProduct). Данный класс связан агрегацией с классом подразделений, зависимостью с классом оборотной ведомости и ассоциацией один ко многим с классом покупателей.

Класс BalanceSheet аккумулирует информацию о перемещениях материальных ценностей в складском учете и не имеет собственных переменных, но имеет метод, который показывает информацию об оборотах материалов и продукции между складами и производством, поставщиками и покупателями.

Начальный класс Menu является отправной точкой программного продукта, отвечающий за выбор той или иной операции, поэтому включает следующие основные переменные: 1. Ввести материал (EnterMaterial:int); 2. Ввести продукт (EnterProduct); 3. Показать обороты (ShowBalanceSheet). Данный класс включает единственный метод, показывающий меню (ShowMenu) и связан ассоциацией с классами материалов, продукции и оборотной ведомости.

Схематично диаграмма классов складского учета производства жидких моющих средств ООО «АВЭЛЛА» представлена на рисунке 26.

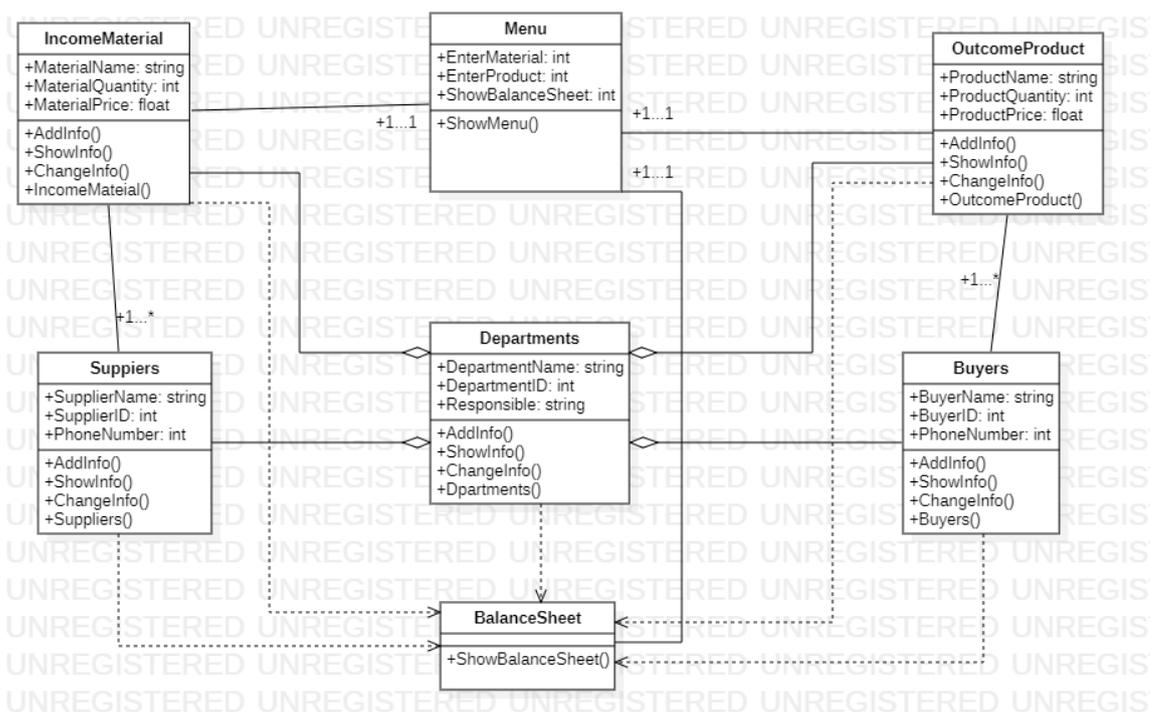


Рисунок 26 – Диаграмма классов складского учета производства жидких моющих средств ООО «АВЭЛЛА»

Далее, с помощью объектно-ориентированного языка программирования C# на платформе .NET, были подробно проработаны классы, интерфейсы и методы разрабатываемого программного продукта складского учета производства жидких моющих средств. Поскольку предполагается использовать WindowsForms, было решено в рамках теоретического этапа проработать сами классы, а затем в рамках практического этапа доработать программный интерфейс и пользовательский интерфейс.

Глава 3 Реализация и тестирование программного обеспечения складского учета участка производства жидких моющих средств производственно-технологического отдела предприятия ООО «АВЭЛЛА»

3.1 Реализация программного обеспечения складского учета участка производства жидких моющих средств ООО «АВЭЛЛА»

В рамках практического этапа было разработано приложение по организации складского учета ООО «АВЭЛЛА» на основе Windows Forms [22] с использованием языка программирования С# [3] в рамках платформы .NET [19]. Для этого были разработаны: графические формы приложения для создания удобного пользовательского интерфейса, реализованы классы программы, а также программный интерфейс для взаимодействия объектов программного приложения. С целью перехода на язык программирования С# была разработана блок-схема программного приложения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА», представленная на рисунке 27.

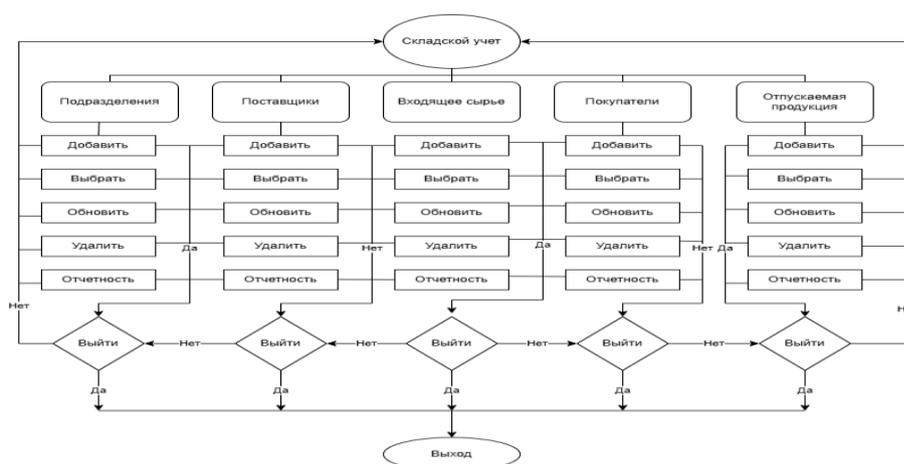


Рисунок 27 – Блок-схема складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Прежде всего, была разработана главная графическая форма приложения на основании WindowsForms. Для этого была применена Windows-форма, получившая наименование «Складской учет ООО «АВЭЛЛА»», в рамках которой было создано выпадающее Меню, включающее основные элементы складского учета ООО «АВЭЛЛА», а именно: Подразделения, Поставщики, Входящие Материалы, Покупатели, Отпускаемые Продукты, а также подменю Выход для возможности завершения программы. Кроме этого, включена справка о программном приложении. Выпадающее меню было разработано с помощью такого инструмента С# как ToolMenuStrip. Данная форма представлена в файлах Form1.cs и Form1.Designer.cs сборки проекта. Скрин главной формы представлен на рисунке 28, а соответствующий код представлен в Приложении Г.

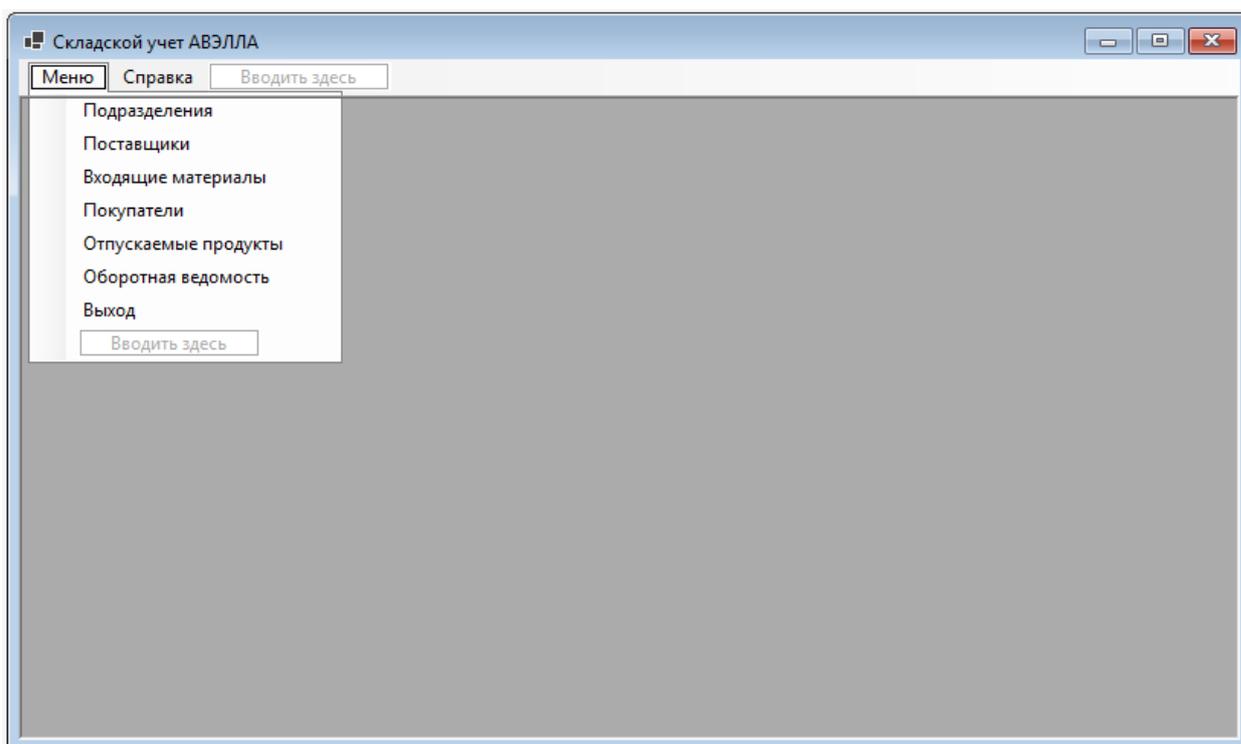


Рисунок 28 – Главная форма Приложения складского учета ООО «АВЭЛЛА»

Далее, по каждому пункту меню была разработана соответствующая отдельная форма. По подразделениям была создана отдельная форма, представляющая пользователю возможность ввода основных компонентов по разделу подразделения, а именно название подразделения, ID подразделения и ответственное лицо подразделения. Данные поля были реализованы с помощью таких инструментов WindowsForms как TextBox, MaskedTextBox и Button. Данная форма реализована в файлах Form2.cs и Form2.Designer.cs сборки проекта. Скрин данной формы представлен на рисунке 29, а код приведен в Приложении Д.

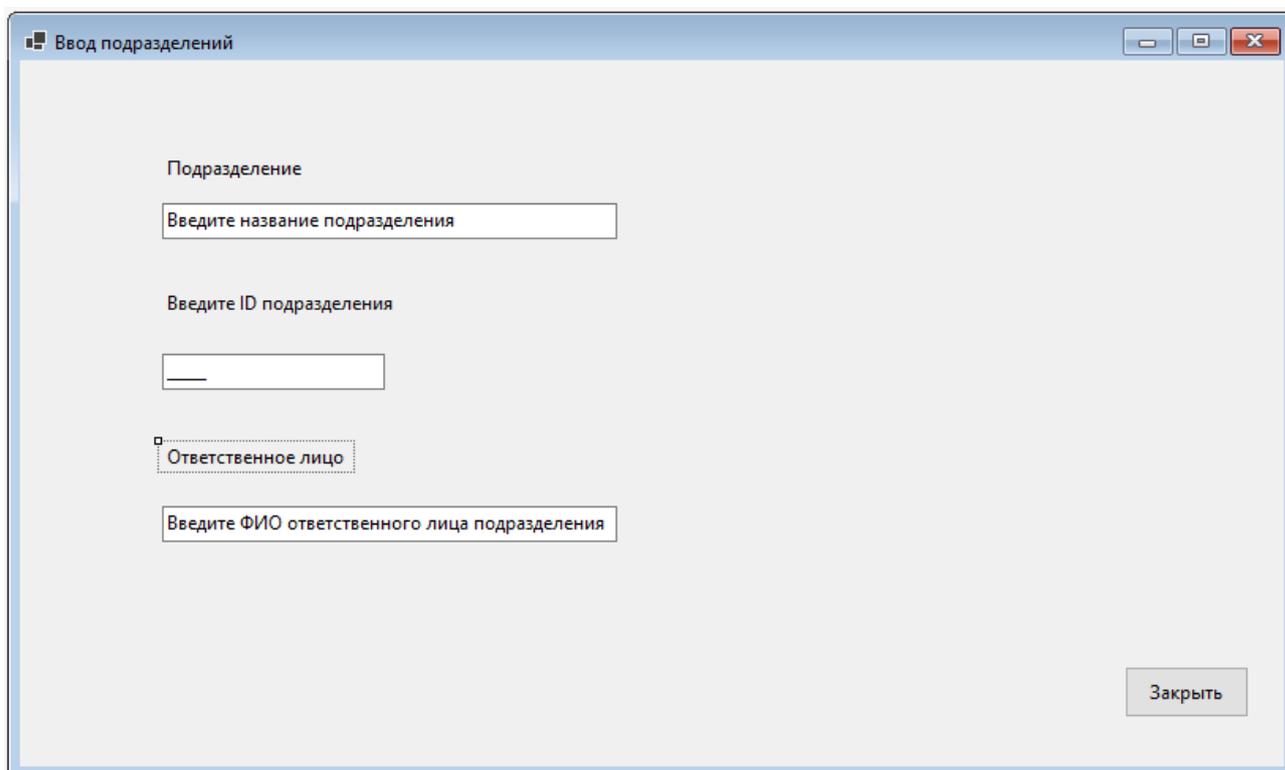
The image shows a screenshot of a Windows application window titled "Ввод подразделений". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area is light gray and contains three input fields. The first field is labeled "Подразделение" and has a text box with the placeholder text "Введите название подразделения". The second field is labeled "Введите ID подразделения" and has a text box with the placeholder text "___". The third field is labeled "Ответственное лицо" and has a text box with the placeholder text "Введите ФИО ответственного лица подразделения". In the bottom right corner of the window, there is a button labeled "Закреть".

Рисунок 29 – Форма Приложения складского учета ООО «АВЭЛЛА» для ввода информации о подразделениях

По поставщикам была также создана отдельная форма, которая предоставляет пользователю возможность ввода данных о поставщиках, а именно Название поставщика, ID поставщика и телефонный номер поставщика. Данные поля также реализованы с помощью WindowsForms-

инструментов – TextBox, MaskedTextBox и Button. Данная форма реализована в файлах Form3.cs и Form3.Designer.cs сборки проекта. Скрин данной формы представлен на рисунке 30, а код приведен в Приложении Е.

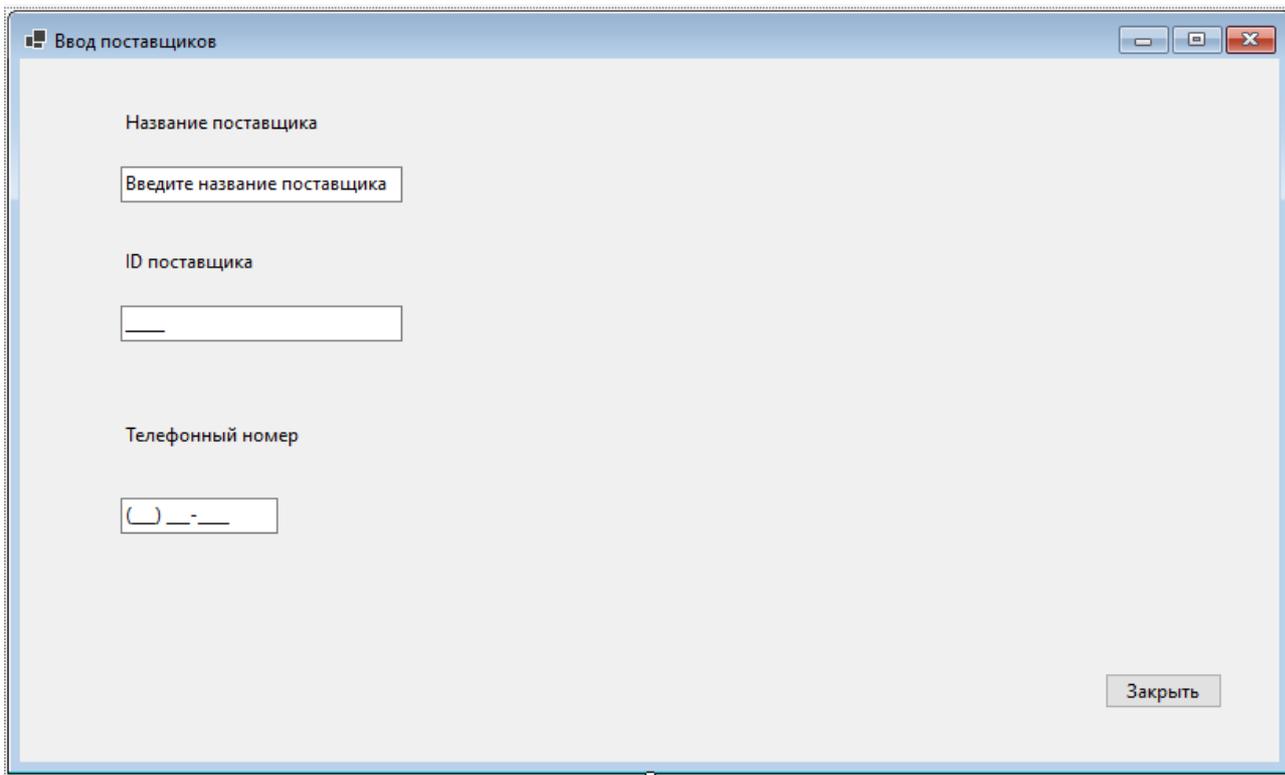
The image shows a screenshot of a Windows application window titled "Ввод поставщиков". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area is light gray and contains three input fields stacked vertically. The first field is labeled "Название поставщика" and has a placeholder text "Введите название поставщика". The second field is labeled "ID поставщика" and has a single underscore placeholder. The third field is labeled "Телефонный номер" and has a placeholder "() _-__". In the bottom right corner of the window, there is a button labeled "Закреть".

Рисунок 30 – Форма Приложения складского учета ООО «АВЭЛЛА» для ввода информации о поставщиках

Была создана отдельная форма по учету входящих материалов, позволяющая вводить информацию о входящем сырье и материалах в ракурсе Название материала, количество и цена. Данные характеристики реализованы в виде полей WindowsForms – TextBox, MaskedTextBox и Button. Данная форма реализована в файлах Form4.cs и Form4.Designer.cs сборки проекта. Скрин формы представлен на рисунке 31, а программный код приведен в Приложении Ж.

Ввод входящих материалов

Наименование материала

Введите входящие материалы

Количество

Цена

Заккрыть

Рисунок 31 – Форма Приложения складского учета ООО «АВЭЛЛА» для ввода информации о входящих материалах

По покупателям была также создана отдельная форма, которая предоставляет пользователю возможность ввода данных о покупателях, а именно Название покупателя, ID покупателя и телефонный номер покупателя. Данные поля также реализованы с помощью WindowsForms-инструментов – TextBox, MaskedTextBox и Button. Данная форма реализована в файлах Form5.cs и Form5.Designer.cs сборки проекта. Скрин данной формы представлен на рисунке 32, а код приведен в Приложении И.

Ввод Покупателя

Название покупателя

Введите название покупателя

ID покупателя

Телефонный номер

() _-__

Закрыть

Рисунок 32 – Форма Приложения складского учета ООО «АВЭЛЛА» для ввода информации о покупателях

Была также создана отдельная форма по учету отпускаемых продуктов, позволяющая вводить информацию об отпускаемых продуктах в разрезе Наименование продукта, количество и цена. Данные характеристики реализованы в виде полей WindowsForms – TextBox, MaskedTextBox и Button. Данная форма реализована в файлах Form6.cs и Form6.Designer.cs сборки проекта. Скрин формы представлен на рисунке 33, а программный код приведен в Приложении К.

Ввод отпускаемых продуктов

Наименование продукта

Введите наименование продукта

Количество

Цена

Закреть

Рисунок 33 – Форма Приложения складского учета ООО «АВЭЛЛА» для ввода информации об отпускаемых продуктах

Далее, для того чтобы реализовать Обратную ведомость, были реализованы классы в программном модуле Program.cs. А для реализации вывода информации в формы был реализованы основные интерфейсы, которые обеспечивают вывод информации о Названии (ShowName), идентификационном номере (ShowID), телефонном номере (ShowPhoneNumber), количестве (ShowQuantity) и цене (ShowPrice). Фрагмент кода, реализующего основные интерфейсы представлен ниже (рисунок 34).

```

interface ShowName
{
    string Name { get; set; }
    string ShowName(string name);
}

interface ShowID
{
    int ID { get; set; }
    int ShowID(int id);
}

interface ShowPhoneNumber
{
    int PhoneNumber { get; set; }
    int ShowPhoneNumber(int phoneNumber);
}

interface ShowQuantity
{
    int Quantity { get; set; }
    int ShowQuantity(int quantity);
}

interface ShowPrice
{
    float Price { get; set; }
    float ShowPrice(float price);
}

```

Рисунок 34 – Фрагмент кода, реализующего основные интерфейсы

Далее были реализованы основные классы приложения складского учета ООО «АВЭЛЛА». Первым был проработан класс Departments. Его переменные (DepartmentID:int; DepartmentName:string; Responsible:string) были прописаны в полях класса и в его конструкторе, а также были созданы соответствующие интерфейсам (ShowName(), ShowID(), ShowResponsible()), позволяющие добавлять и выводить информацию об отделах, связанных со складским учетом производственного участка жидких моющих средств. Получился следующий фрагмент кода (рисунок 35).

```

public class Departments // класс Отделы
{
    private string DepartmentName; // поля Отделы
    ▲ private int DepartmentID;
    ▲ private string Responsible;
    public Departments(string departmentName, int departmentID, string
responsible) // конструктор Отделы
    {
        DepartmentName = departmentName;
        DepartmentID = departmentID;
        Responsible = responsible;
    }

    public string ShowName(string namesD) // соответствующие методы-
интерфейсы
    {
        return DepartmentName;
    }

    public int ShowID(int idDp)
    {
        return DepartmentID;
    }

    public string ShowResponsible(string rsp)
    {
        return Responsible;
    }
}

```

Рисунок 35 – Фрагмент кода основных классов складского учета

Далее, был проработан класс Suppliers с переменными SupplierID:int, SupplierName:string, PhoneNumber:int, которые были прописаны в полях и конструкторе класса, к которым также были созданы соответствующие интерфейсам (ShowName(), ShowID(), ShowPhoneNumber()), которые

позволяют добавлять информацию о поставщиках, являющихся контрагентами складского учета производственного участка жидких моющих средств. Фрагмент код представлен ниже (рисунок 36).

```
public class Suppliers
{
    private string SupplierName; // поля Поставщики
    private int SupplierID;
    private int PhoneNumber;
    public Suppliers(string supplierName, int supplierID, int phoneNumber)
// конструктор Поставщики
    {
        SupplierName = supplierName;
        SupplierID = supplierID;
        PhoneNumber = phoneNumber;
    }

    public string ShowName(string namesP) // соответствующие методы-
интерфейсы
    {
        return SupplierName;
    }

    public int ShowID(int idSp)
    {
        return SupplierID;
    }

    public int ShowPhoneNumber(int pnSp)
    {
        return PhoneNumber;
    }
}
```

Рисунок 36 – Фрагмент кода класса Suppliers с переменными

Следующим был проработан класс входящего сырья и материалов IncomeMaterial с переменными MaterialName:int, MaterialQuantity:int,

MaterialPrice:float, также реализованные в полях, конструкторе и снабженные соответствующими интерфейсами (ShowName(), ShowQuantity(), ShowPrice()), которые позволяют добавлять информацию о наименовании поставляемых материалов, их количестве и цене, используемые в информации складского учета производственного участка жидких моющих средств. Данный фрагмент кода представлен далее (рисунок 37).

```
public class IncomeMaterial
{
    private string MaterialName;
    private int MaterialQuantity;
    private float MaterialPrice;
    public IncomeMaterial(string materialName, int materialQuantity, float
materialPrice)
    {
        MaterialName = materialName;
        MaterialQuantity = materialQuantity;
        MaterialPrice = materialPrice;
    }

    public string ShowName(string matNm)
    {
        return MaterialName;
    }
    public int ShowQuantity(int matQt)
    {
        return MaterialQuantity;
    }

    public float ShowPrice(float matPrice)
    {
        return MaterialPrice;
    }
}
```

Рисунок 37 – Фрагмент кода класса IncomeMaterial с переменными

После этого, был проработан класс Buyers с переменными BuyerName:string, BuyerID:int, PhoneNumber:int, поля и конструктор которых

также были проработаны, снабженные соответствующими интерфейсами (ShowName(), ShowID(), ShowPhoneNumber()), которые позволяют добавлять информацию о покупателях складского учета производственного участка жидких моющих средств. Фрагмент кода представлен ниже (рисунок 38).

```
public class Buyers
{
    private string BuyersName;
    private int BuyerID;
    private int PhoneNumberBr;

    public Buyers(string buyerName, int buyerID, int phoneNumberBr)
    {
        BuyersName = buyerName;
        BuyerID = buyerID;
        PhoneNumberBr = phoneNumberBr;
    }
    public string ShowName(string byNm)
    {
        return BuyersName;
    }

    public int ShowID(int byId)
    {
        return BuyerID;
    }

    public int ShowPhoneNumber(int byPN)
    {
        return PhoneNumberBr;
    }
}
```

Рисунок 38 – Фрагмент кода класса Buyers с переменными

И, наконец, был проработан класс OutcomeProduct, отвечающий за отпускаемую продукцию, с переменными ProductName:string,

ProductQuantity:int, ProductPrice:float, для которых были созданы соответствующие поля и конструктор, снабженные соответствующими интерфейсами (ShowName(), ShowQuantity(), ShowPrice()), которые позволяют добавлять информацию об отпускаемых товарах складского учета производственного участка жидких моющих средств. Фрагмент кода представлен ниже (рисунок 39).

```
public class OutcomeProduct
{
    private string ProductName; // поля Отпускаемые продукты
    private int ProductQuantity;
    private float ProductPrice;
    public OutcomeProduct(string productName, int productQuantity, float
productPrice) // конструктор Отпускаемые продукты
    {
        ProductName = productName;
        ProductQuantity = productQuantity;
        ProductPrice = productPrice;
    }

    public string ShowtName(string pdNm) // соответствующие
интерфейсы
    {
        return ProductName;
    }

    public int ShowQuantity(int pdQt)
    {
        return ProductQuantity;
    }

    public float ShowPrice(float pdPrice)
    {
        return ProductPrice;
    }
}
```

Рисунок 39 – Фрагмент кода класса OutcomeProduct с переменными

Далее была создана форма Обратной ведомости, которая предоставляет пользователю возможность вывода данных о поставщиках, сырье и

материалах, покупателях, отпускаемой продукции. Данные поля также реализованы с помощью WindowsForms-инструментов – TextBox, MaskedTextBox и Button. Данная форма реализована в файлах Form7.cs и Form7.Designer.cs сборки проекта. Скрин данной формы представлен на рисунке 40, а код приведен в Приложении Л.

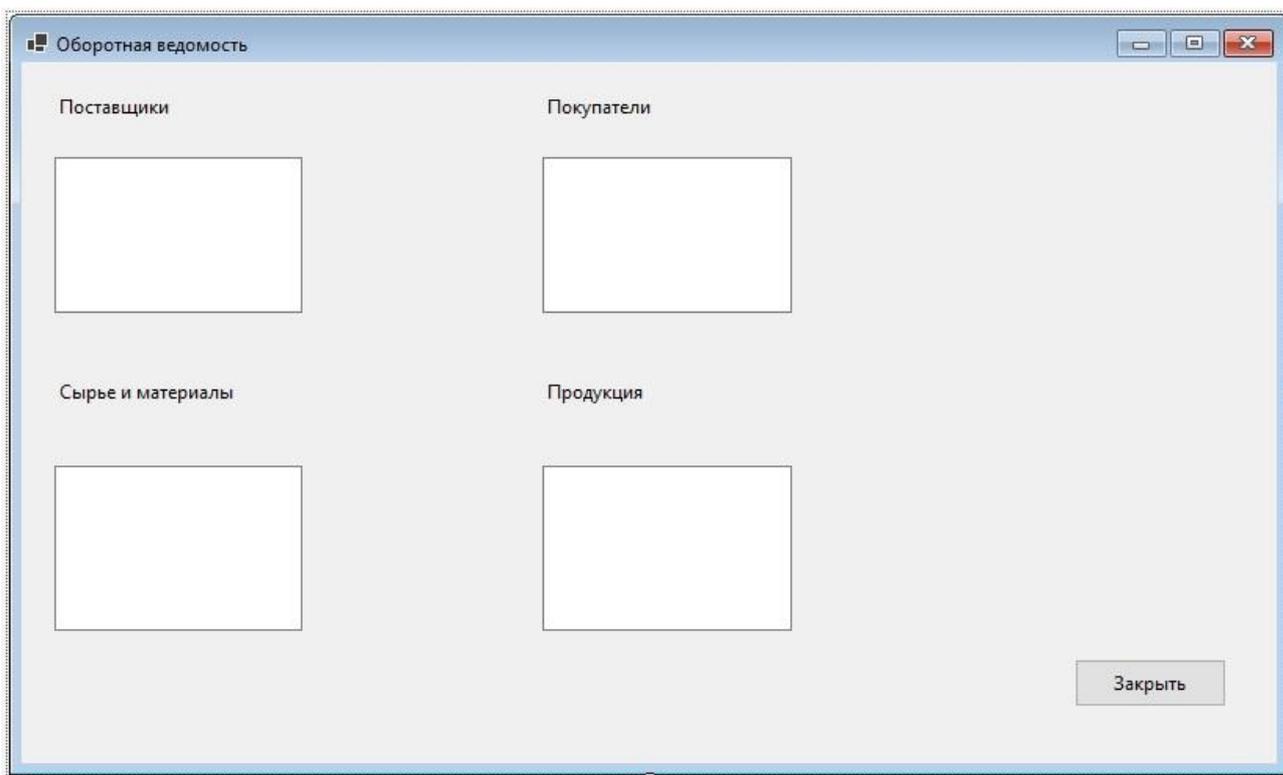


Рисунок 40 – Форма Приложения складского учета ООО «АВЭЛЛА» для вывода информации о поставщиках, сырье, покупателях и продукции

Подготавливаем базу данных AVELLAWarehouse.db с таблицами для товаров, поставщиков, приходов и расходов складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА», структура которой представлена ниже (рисунок 41).

```

CREATE TABLE Products (
    Id INTEGER PRIMARY KEY AUTOINCREMENT,
    Name TEXT NOT NULL,
    Description TEXT,
    Quantity REAL DEFAULT 0
);
CREATE TABLE Suppliers (
    Id INTEGER PRIMARY KEY AUTOINCREMENT,
    Name TEXT NOT NULL,
    ContactInfo TEXT
);
CREATE TABLE Incomes (
    Id INTEGER PRIMARY KEY AUTOINCREMENT,
    ProductId INTEGER REFERENCES Products(Id),
    SupplierId INTEGER REFERENCES Suppliers(Id),
    Date DATETIME DEFAULT CURRENT_TIMESTAMP,
    Quantity REAL CHECK (Quantity > 0)
);
CREATE TABLE Customers (
    Id INTEGER PRIMARY KEY AUTOINCREMENT,
    FullName TEXT NOT NULL,
    PhoneNumber TEXT,
    Email TEXT,
    Address TEXT
);
CREATE TABLE Expenses (
    Id INTEGER PRIMARY KEY AUTOINCREMENT,
    ProductId INTEGER REFERENCES Products(Id),
    Date DATETIME DEFAULT CURRENT_TIMESTAMP,
    Quantity REAL CHECK (Quantity > 0),
    CustomerId INTEGER REFERENCES Customers(Id)
);

```

Рисунок 41 – Фрагмент кода SQL-запроса на подготовку базы данных

Добавляем таблицу подразделений и связываем ее с существующими сущностями, такими как товары и поставщики. Соответствующие корректировки представлены ниже (рисунок 41).

```
-- Создаем таблицу подразделений
CREATE TABLE Departments (
    Id INTEGER PRIMARY KEY AUTOINCREMENT,
    Name TEXT NOT NULL UNIQUE
);
-- Добавляем колонку подразделения в таблицу продуктов
ALTER TABLE Products ADD COLUMN DepartmentId INTEGER
REFERENCES Departments(Id);
-- Добавляем колонку подразделения в таблицу поставок
ALTER TABLE Incomes ADD COLUMN DepartmentId INTEGER
REFERENCES Departments(Id);

-- Добавляем колонку подразделения в таблицу расходов
ALTER TABLE Expenses ADD COLUMN DepartmentId INTEGER
REFERENCES Departments(Id);
```

Рисунок 41 – Корректировки-связки сущностей базы данных

Реализуем интерфейс для просмотра списка подразделений и возможности добавлять новые подразделения, представленный в файле FormDepartments.cs (рисунок 42).

```

▶ using System.Windows.Forms;

public partial class DepartmentsForm : Form {
    public DepartmentsForm() {
        InitializeComponent();

        ListView departmentsList = new ListView();
        departmentsList.View = View.Details;
        departmentsList.Columns.Add("ID", 50);
        departmentsList.Columns.Add("Название подразделения", 200);
        Button btnAddDepartment = new Button { Text = "Добавить
подразделение" };
        btnAddDepartment.Click += BtnAddDepartment_Click;
        Controls.Add(departmentsList);
        Controls.Add(btnAddDepartment);
    }

    private void BtnAddDepartment_Click(object sender, EventArgs e) {
        var addDeptForm = new AddDepartmentForm();
        if(addDeptForm.ShowDialog() == DialogResult.OK) {
            LoadDepartmentsData(); // Обновляем список подразделений
        }
    }

    private void LoadDepartmentsData() {
        // Код загрузки данных из базы данных и отображения в списке
    }
}

```

Рисунок 42 – Фрагмент кода интерфейса для просмотра списка подразделений

А также метод загрузки данных по подразделениям LoadDepartmentsData() (рисунок 43).

```

using System;
using System.Collections.Generic;
using System.Data.SQLite;
using System.Windows.Forms;

partial class DepartmentsForm : Form
{
    private const string DB_PATH = "path_to_your_database.db"; // укажите реальный путь к
вашему файлу базы данных
    private void LoadDepartmentsData()
    {
        // Очистка текущего содержимого ListView
        listViewDepartments.Items.Clear();
        // Открываем подключение к базе данных
        using (SQLiteConnection conn = new SQLiteConnection($"Data Source={DB_PATH};"))
        {
            conn.Open();

            // SQL-запрос для извлечения данных подразделений
            string selectQuery = "SELECT Id, Name FROM Departments ORDER BY Name ASC";
            // Выполняем запрос
            using (SQLiteCommand command = new SQLiteCommand(selectQuery, conn))
            {
                using (SQLiteDataReader reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        // Извлекаем данные и создаем строку для ListView
                        ListViewItem item = new ListViewItem(reader.GetString(0)); // id
                        item.SubItems.Add(reader.GetString(1)); // Название подразделения

                        // Добавляем запись в ListView
                        listViewDepartments.Items.Add(item);
                    }
                }
            }
        }
    }
}

```

Рисунок 43 – Фрагмент кода метода загрузки данных по подразделениям

Однако платформа SQLite не смогла соединиться с созданной базой данных, выдавая логическую ошибку об отсутствии самой базы данных и ее компонентов. Поэтому было решено внедрить в проект базы данных при

помощи технологии ADO.NET, которая является более интегрированной в платформу .NET с использованием подхода набора данных DataSet.

Новая структура базы данных имеет следующие компоненты, представленные в таблице 4.

Таблица 4 - Структура базы данных приложения складского учета ООО «АВЭЛЛА»

Название	Атрибуты
Departments	Id INT PRIMARY KEY, Name TEXT NOT NULL
Suppliers	Id INT PRIMARY KEY, Name TEXT NOT NULL, ContactInfo TEXT
IncomeMaterial	Id INT PRIMARY KEY, SupplierId INT REFERENCES Suppliers(Id), DepartmentId INT REFERENCES Department(Id), Date DATETIME DEFAULT CURRENT_TIMESTAMP, Quantity REAL CHECK (Quantity>0), Price REAL CHECK (Quantity>0)
Buyers	Id INT PRIMARY KEY, Name TEXT NOT NULL, ContactInfo TEXT
OutcomeProduct	Id INT PRIMARY KEY, BuyersId INT REFERENCES Buyers(Id), DepartmentId INT REFERENCES Department(Id), Date DATETIME DEFAULT CURRENT_TIMESTAMP, Quantity REAL CHECK (Quantity>0), Price REAL CHECK (Price>0)

Данная структура была реализована в базе данных Warehouse, на основании которого был создан набор данных DataSetWarehouse, представленный на рисунке 44.

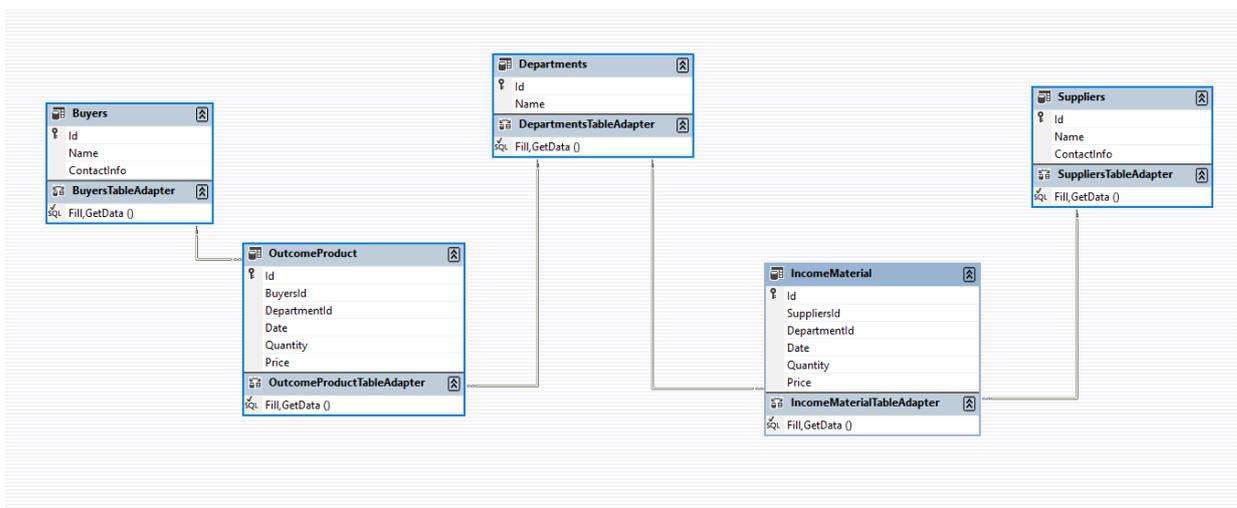


Рисунок 44 – Набор данных DataSetWarehouse базы данных складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Далее, была заново разработана форма Подразделений с учетом режимов доступности методов редактирования и сохранения данных по подразделениям. Дополнительный код формы представлен ниже (рисунок 45), а новая форма представлена на рисунке 46.

```

public partial class FormDepartments : Form
{
    public FormDepartments()
    ▲ {
        InitializeComponent();
        DisplayReadOnly(true);
    }
    public void DisplayReadOnly(bool ReadOnly)
    {
        this.textBoxNameDepartments.ReadOnly = ReadOnly;
        this.textBoxIdDepartment.ReadOnly = ReadOnly;
        this.textBoxResponsibleDepartment.ReadOnly = ReadOnly;
    }
    public void MenuItemEnabled(bool mode)
    {
        MenuItemEdit.Enabled = mode;
        MenuItemRemove.Enabled = mode;
        MenuItemNew.Enabled = mode;
        MenuItemSave.Enabled = !mode;
        MenuItemUndo.Enabled = !mode;
    }
    private void DisplayForm(bool mode)
    {
        DisplayReadOnly(mode);
        MenuItemEnabled(mode);
    }
}
}

```

Рисунок 45 – Фрагмент кода формы Подразделений

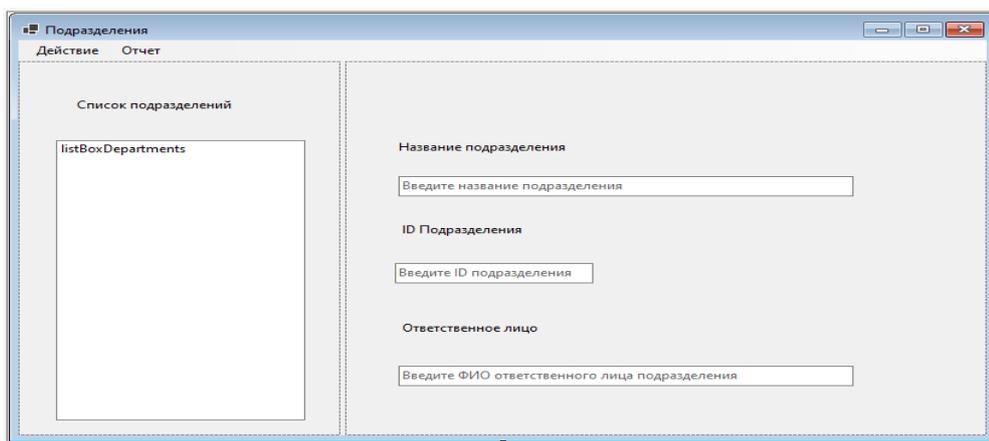


Рисунок 46 – Новая форма по подразделениям складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

В рамках данной формы были также реализованы дополнительные методы для создания нового подразделения, для сохранения данных, для отмены изменений, для удаления и редактирования данных. Данные методы связаны с набором данных DataSet и посредством него с базой данных dbo.Warehouse. Дополнительно, были добавлены защиты от ошибок пользователя, заключающиеся в блокировке ввода пользователем пустых полей, либо новых записей с существующим идентификационным номером. Таким образом, изменения в Windows Forms отражаются и в базе данных. Коды соответствующих методов представлены Приложении Д.

Развивая данную форму в нее, была интегрирована форма отчетности по подразделениям, которая позволяет конструировать определенные формы отчетности в Windows-приложениях. Таким образом, была создана форма отчета, представленная на рисунке 47, ее Windows-форма, представлена на рисунке 48, которые связывают базу данных складского учета, набор данных и позволяют выводить их на экран, а также распечатывать и сохранять отдельным документом в pdf-формате. Код создания sql-соединения и создаваемого на его основании отчета по подразделениям складского учета ООО «АВЭЛЛА» представлен в Приложении М.

Id	Name	Responsible	Comment
[Id]	[Name]	[Responsible]	

Рисунок 47 – Форма отчета по подразделениям складского участка жидких моющих средств ООО «АВЭЛЛА»

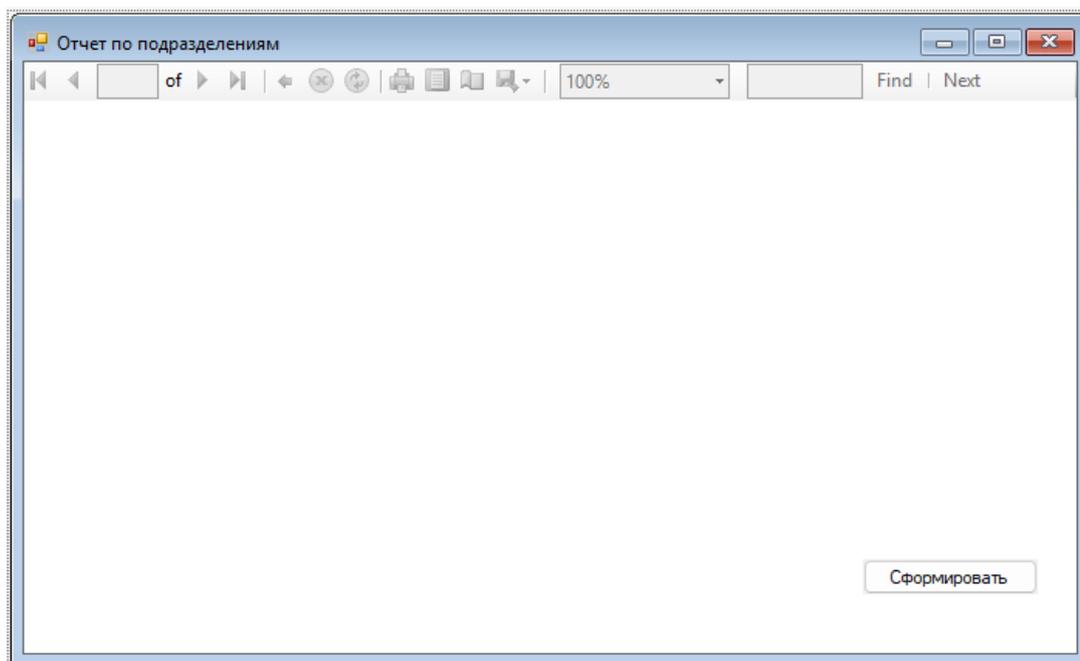


Рисунок 48 – Windows-форма отчета по подразделениям складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Также была разработана новая форма Поставщиков сырья и материалов, с защитой от пользовательских ошибок с интегрированной отчетностью и возможностью распечатывания и сохранения отчета по поставщикам, код которой представлен в Приложении Е, а скрин формы на рисунке 49 и рисунке 50 и 51.

Рисунок 49 – Новая форма по поставщикам сырья и материалов складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Id	Name	Contact Info	Comment
[Id]	[Name]	[ContactInfo]	

Рисунок 50 – Форма отчета по поставщикам складского участка жидких моющих средств ООО «АВЭЛЛА»

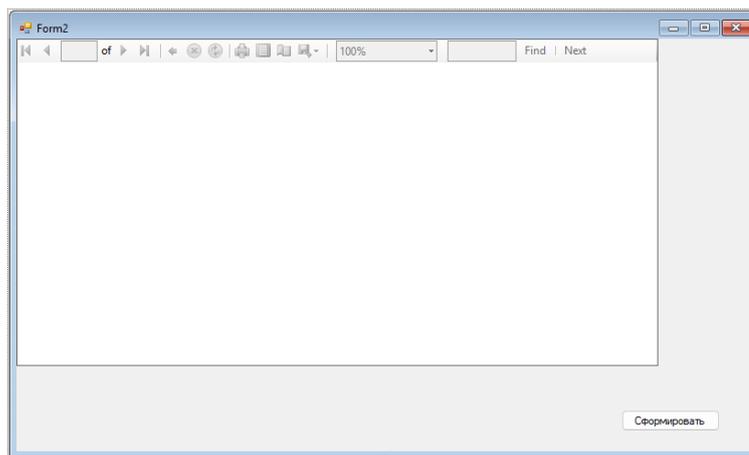


Рисунок 51 – Windows-форма отчета по поставщикам складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

В комплексе с остальными была разработана новая форма Входящего сырья и материалов, с защитой от пользовательских ошибок с интегрированной отчетностью и возможностью распечатывания и сохранения отчета по поставщикам, код которой представлен в Приложении Ж, а скрин формы на рисунке 52 и рисунках 53, 54.

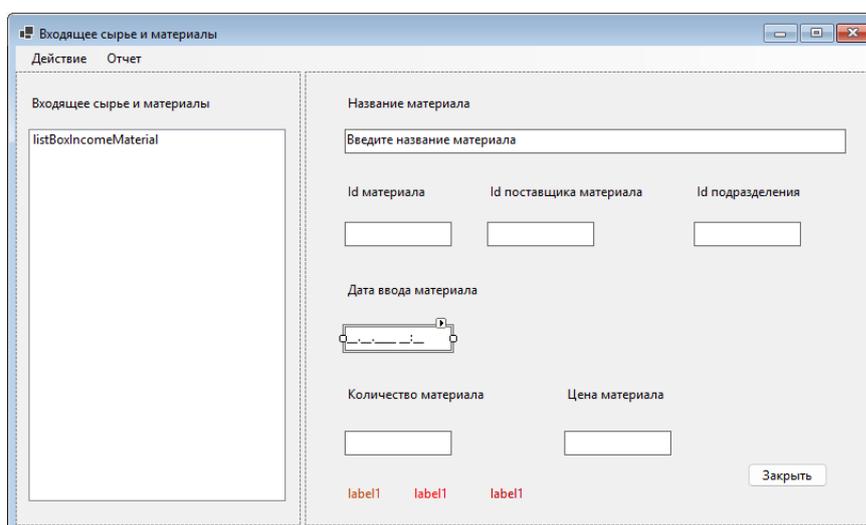


Рисунок 52 – Новая форма входящего сырья и материалов складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Id	Name	Supplier Id	Department	Date	Quantity	Price	Id
[Id]	[Name]	[SupplierId]	[DepartmentId]	[Date]	[Quantity]	[Price]	

Рисунок 53 – Форма отчета по сырью и материалам складского участка жидких моющих средств ООО «АВЭЛЛА»

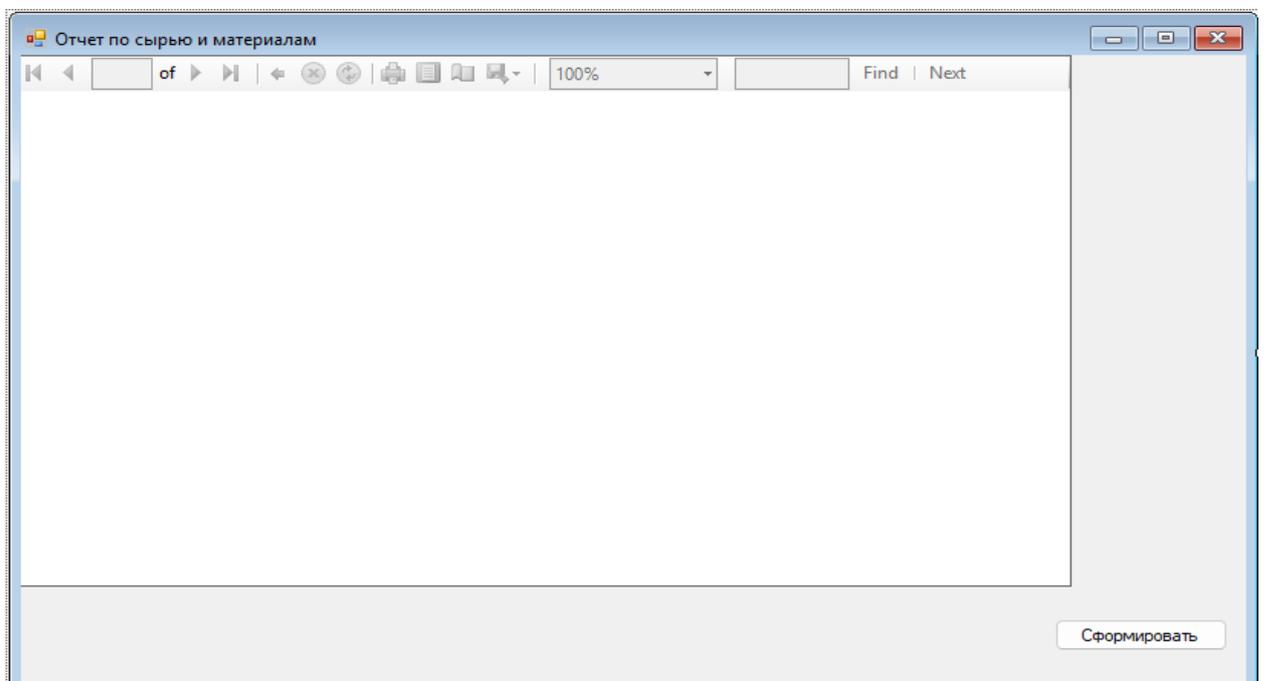


Рисунок 54 – Windows-форма отчета по сырью и материалам складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Также была разработана новая форма Покупателей, код которой представлен в Приложении И, а скрин формы на рисунке 55 и рисунках 56 и 57.

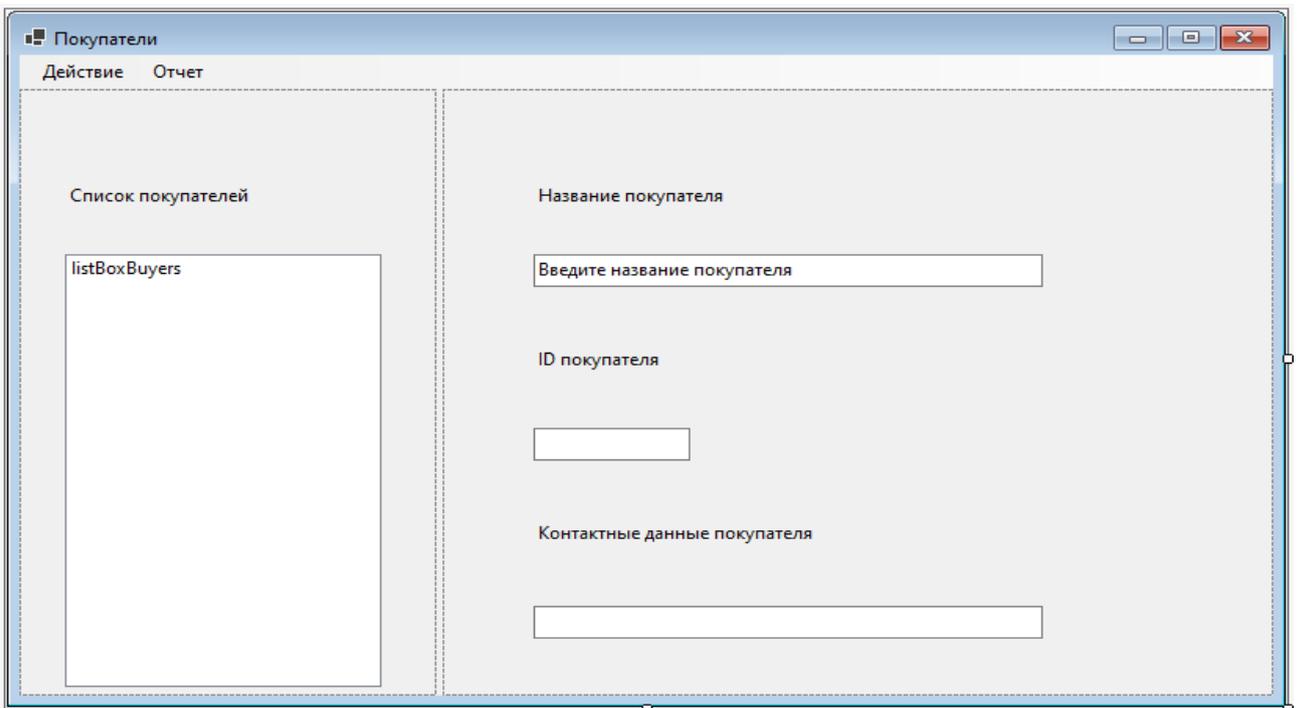


Рисунок 55 – Новая форма по покупателям складского учета
производственного участка жидких моющих средств ООО «АВЭЛЛА»

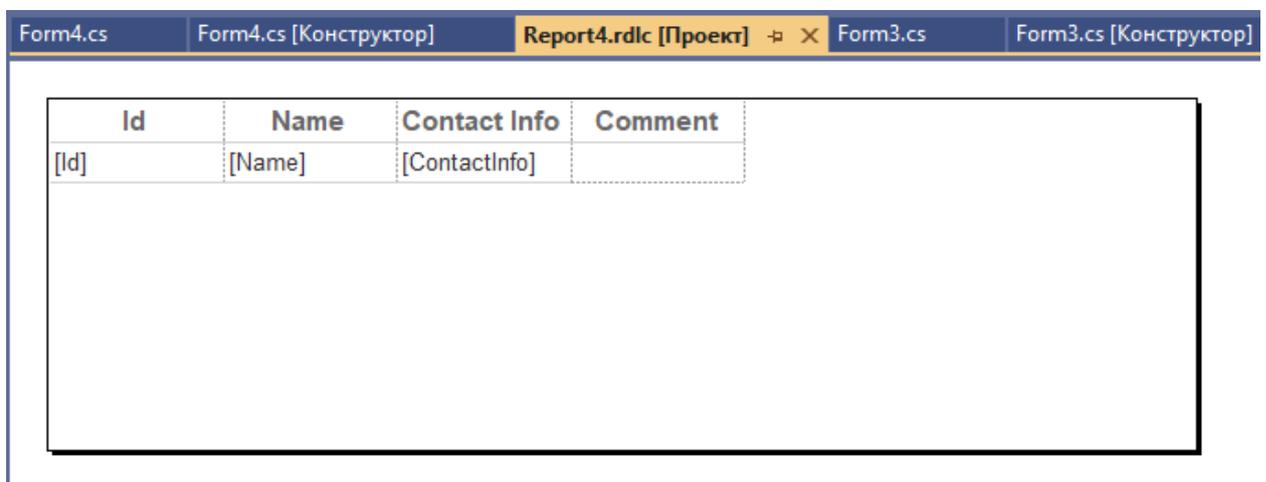


Рисунок 56 – Форма отчета по покупателям складского участка жидких
моющих средств ООО «АВЭЛЛА»

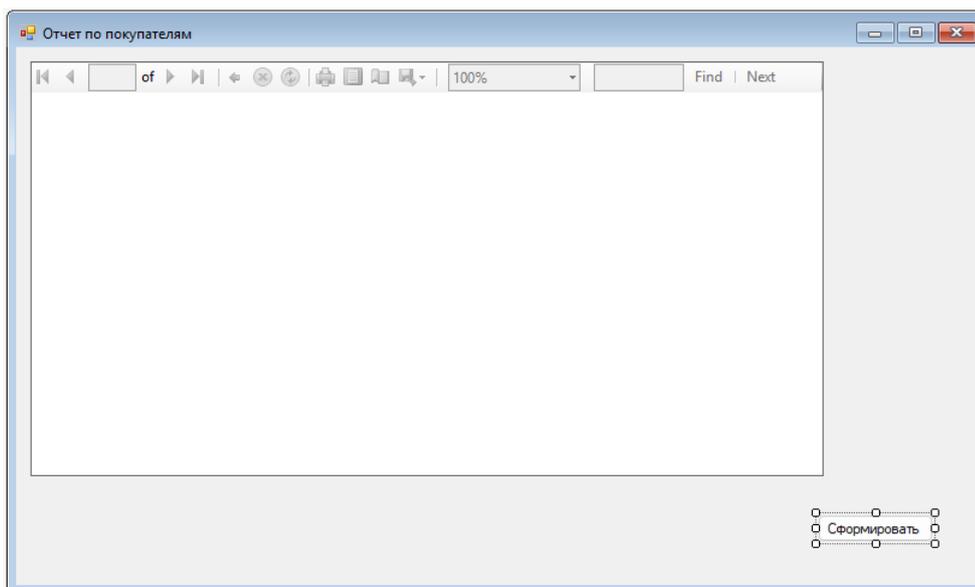


Рисунок 57 – Windows-форма отчета по покупателям складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

И наконец, была разработана новая форма Отпускаемой продукции, код которой представлен в Приложении К, а скрин формы на рисунке 58 и рисунках 59 и 60.

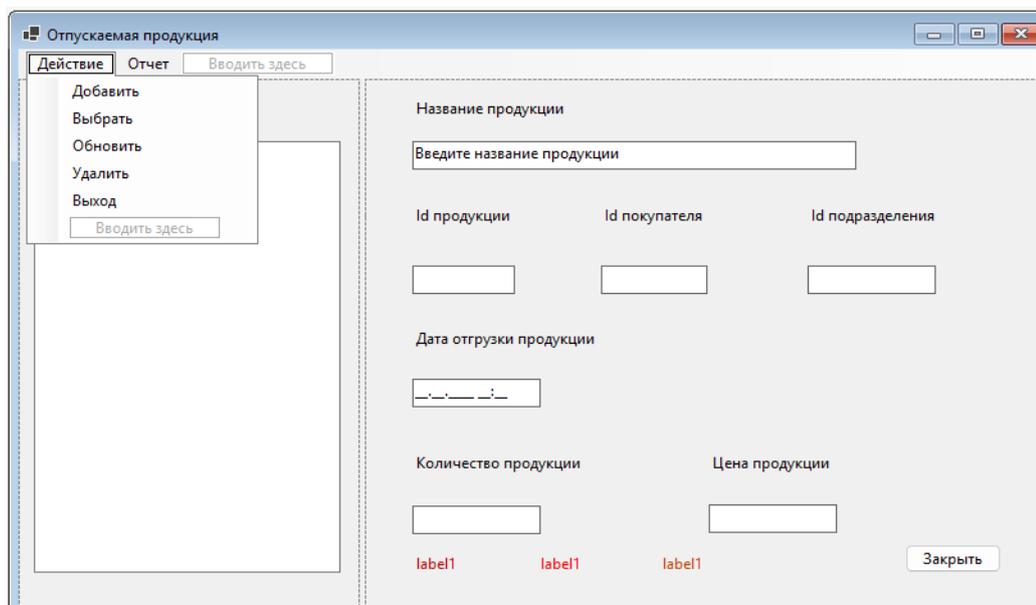


Рисунок 58 – Новая форма по отпускаемой продукции складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Id	Name	Buyers Id	Department	Date	Quantity	Price	Comment
[Id]	[Name]	[BuyersId]	[DepartmentId]	[Date]	[Quantity]	[Price]	

Рисунок 59 – Форма отчета по отпускаемой продукции складского участка жидких моющих средств ООО «АВЭЛЛА»

The screenshot shows a Windows application window titled "Отчет по продукции". At the top, there is a toolbar with various icons for navigation (back, forward, search) and printing. Below the toolbar is a large, empty rectangular area intended for the report content. At the bottom right of the window, there is a button labeled "Сформировать". The window title bar includes standard Windows window controls (minimize, maximize, close).

Рисунок 60 – Windows-форма отчета по отпускаемой продукции складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

В итоге, были соединены все интерфейсы, классы и формы и получилось приложение складского учета производственного участка ООО «АВЭЛЛА», которое позволяет вводить и сохранять информацию о поставщиках, материалах, покупателях, продукции.

3.2 Тестирование программного обеспечения складского учета участка производства жидких моющих средств ООО «АВЭЛЛА»

Тестирование Приложения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА» проводилось с учетом отечественных и международных стандартов. Согласно ГОСТ Р 56920-2016 «Системная и программная инженерия. Тестирование программного обеспечения» [12], тестирование программного обеспечения должно быть направлено на предоставление информации о программном продукте и нахождение максимально возможного числа дефектов на возможно ранних этапах процесса разработки при заданных ограничениях стоимости и графика разработки. А согласно международного стандарта ISO/IEC/IEEE 29119-1:2013(E) «Software and systems engineering Software testing Part 1: Concepts and definitions» [24], тестирование программного обеспечения определяется как процесс, содержащий в себе все активности жизненного цикла, как динамические, так и статические, касающиеся планирования, подготовки и оценки программного продукта и связанных с этим результатов работ с целью определить, что они соответствуют описанным требованиям, показать, что они подходят для заявленных целей и для определения дефектов.

Согласно общепринятой методологии, для тестирования программного обеспечения необходимо выбрать объект тестирования и обозначить основные функциональные требования к конкретному программному продукту [18]. Поэтому объектом тестирования выступило Приложение складского учета производственного участка жидких моющих средств, а более конкретно, набор возможностей, которые соответствуют функциональным требованиям продукта. Функциональные требования к приложению были сформированы на предыдущем этапе разработки программного продукта и были собраны и представлены в таблице 5.

Таблица 5 – Функциональные требования к программному обеспечению складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Объект – Приложение складского учета производства жидких моющих средств
Функциональные требования
Иметь возможность ввода подразделения
Учет поставщиков сырья и материалов
Учет ответственных лиц склада сырья и материалов
Учет входящего сырья и материалов
Иметь возможность ввода покупателей продукции
Учет отгружаемой готовой продукции
Учет ответственных лиц склада готовой продукции
Возможность формировать отчеты по поставщикам, сырью, продукции и покупателям
Возможность распечатывания отчета и сохранения его в форматах Word, PDF, Excel

На основании списка функциональных требований оформляется план тестирования [14]. Поэтому, с учетом сформированных и представленных функциональных требований программного обеспечения был сформирован план тестирования приложения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА», представленный в таблице 6.

Таблица 6 – План тестирования программного обеспечения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Сценарий (test scenario)	Действия (actions)	Ожидаемый результат (expected results)	Полученный результат (actual results)
Проверить возможность ввода названия подразделения	Ввести в Форму Название подразделения	В форме ввода будет введено полное название подразделения	Положительно
Проверить возможность ввода названия поставщика	Ввести в Форму идентификационный номер поставщика	В форме присутствует название и ID поставщика	Положительно

продолжение таблицы 6

Протестировать учет материалов	Ввести в форму название и количество материала	В форме ввода сырья и материалов введено название и количество материала	Положительно
Проверить ввод фамилии ответственного лица склада сырья и материалов	Ввести в форму ввода подразделения склада сырья и материалов ФИО ответственного лица	В форме подразделения склада сырья и материалов обозначено ФИО ответственного лица	Положительно
Протестировать ввод цены входящего сырья	Ввести в форму ввода сырья и материалов цену входящего сырья	В форме входящего сырья и материалов обозначена цена входящего сырья	Положительно
Проверить ввод покупателей продукции	Ввести в форму название покупателя и его идентификационный номер	В форме покупателей введено название и ID покупателя	Положительно
Протестировать учет отгружаемой продукции	Ввести в форму отгрузки готовой продукции название продукции	В форме готовой продукции обозначено название продукции	Положительно
Проверить учет количества и цены отгружаемой продукции	Ввести в форму готовой продукции количество и цену готовой продукции	В форме готовой продукции обозначены количество и цена продукта	Положительно
Протестировать учет подразделения склада готовой продукции	Ввести в форму подразделения склада готовой продукции	В форме введено подразделение склада готовой продукции	Положительно
Проверить учет ответственного лица склада готовой продукции	Ввести в форму склада готовой продукции ФИО ответственного лица	В форме склада готовой продукции введено ФИО ответственного лица	Положительно
Протестировать формирование отчета	Сформировать отчет	В сформированном отчете обозначена продукция, дата поставки, количество и цена	Положительно
Протестировать возможность распечатывания отчета и сохранения его в формате Word, PDF, Excel	Распечатать отчет и сохранить его в форматах Word, PDF, Excel	Распечатанный отчет и сохраненный файл отчета в формате WORD, PDF, Excel	Положительно

В соответствии с указанным планом тестирования программного приложения, были проверены все заявленные возможности приложения складского учета ООО «АВЭЛЛА». Прежде всего, проверена возможность ввода названия подразделения в соответствующую форму, которая дала положительный результат, и скрин которой представлен на рисунке б1.

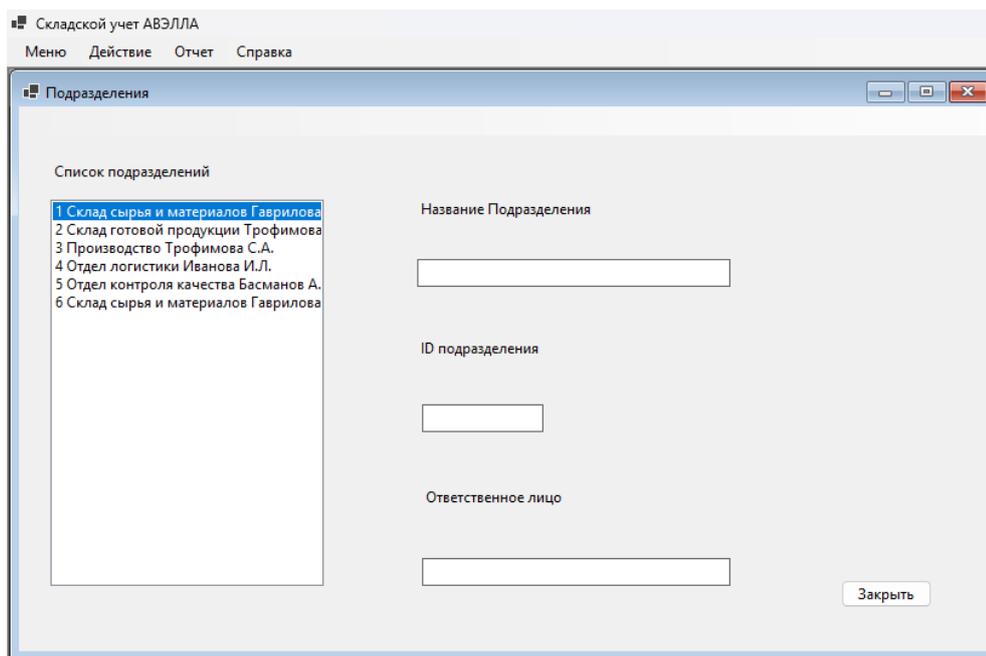


Рисунок б1 – Ввод склада сырья и материалов в форму ввода подразделений складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Далее была проверена возможность ввода названия и идентификационного номера поставщика в соответствующую форму складского учета ООО «АВЭЛЛА», что представлено на рисунке б2.

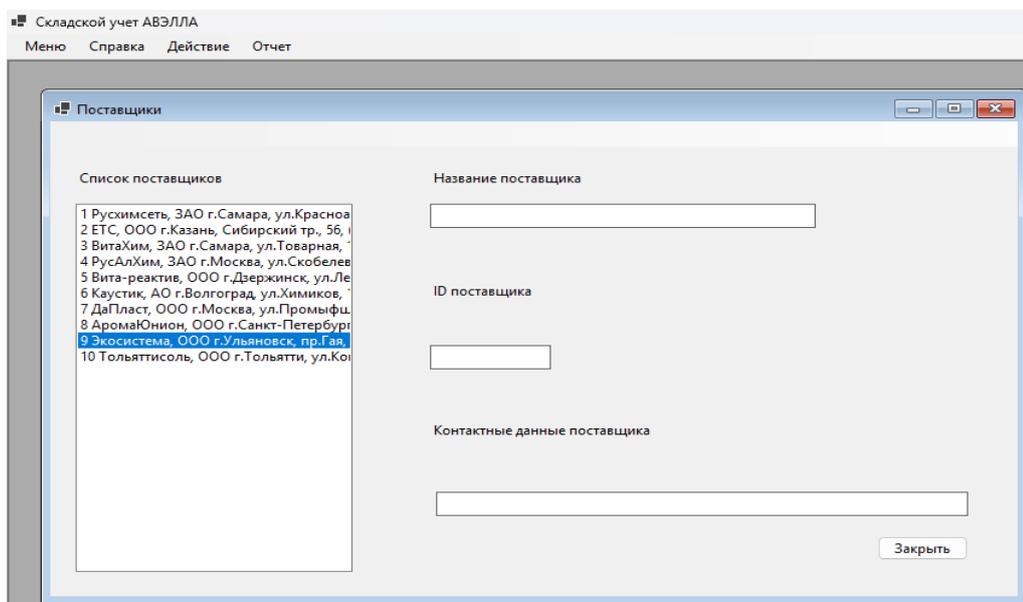


Рисунок 62 – Ввод названия и ID в форму поставщика складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Следующим пунктом плана тестирования программного продукта был ввод материалов, который также показал положительный результат и представлен на рисунке 63.

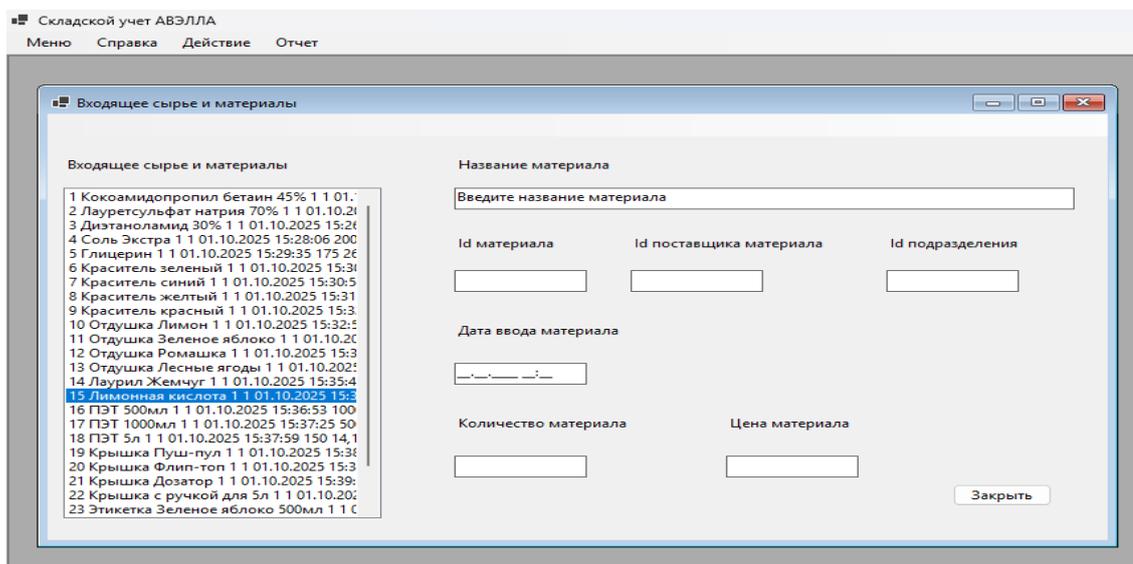


Рисунок 63 – Ввод названия материала в форму входящих сырья и материалов складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Следующим шагом плана тестирования программного обеспечения был ввод фамилии, имени, отчества ответственного лица склада сырья и материалов, показавший положительный результат и представленный на рисунке 64.

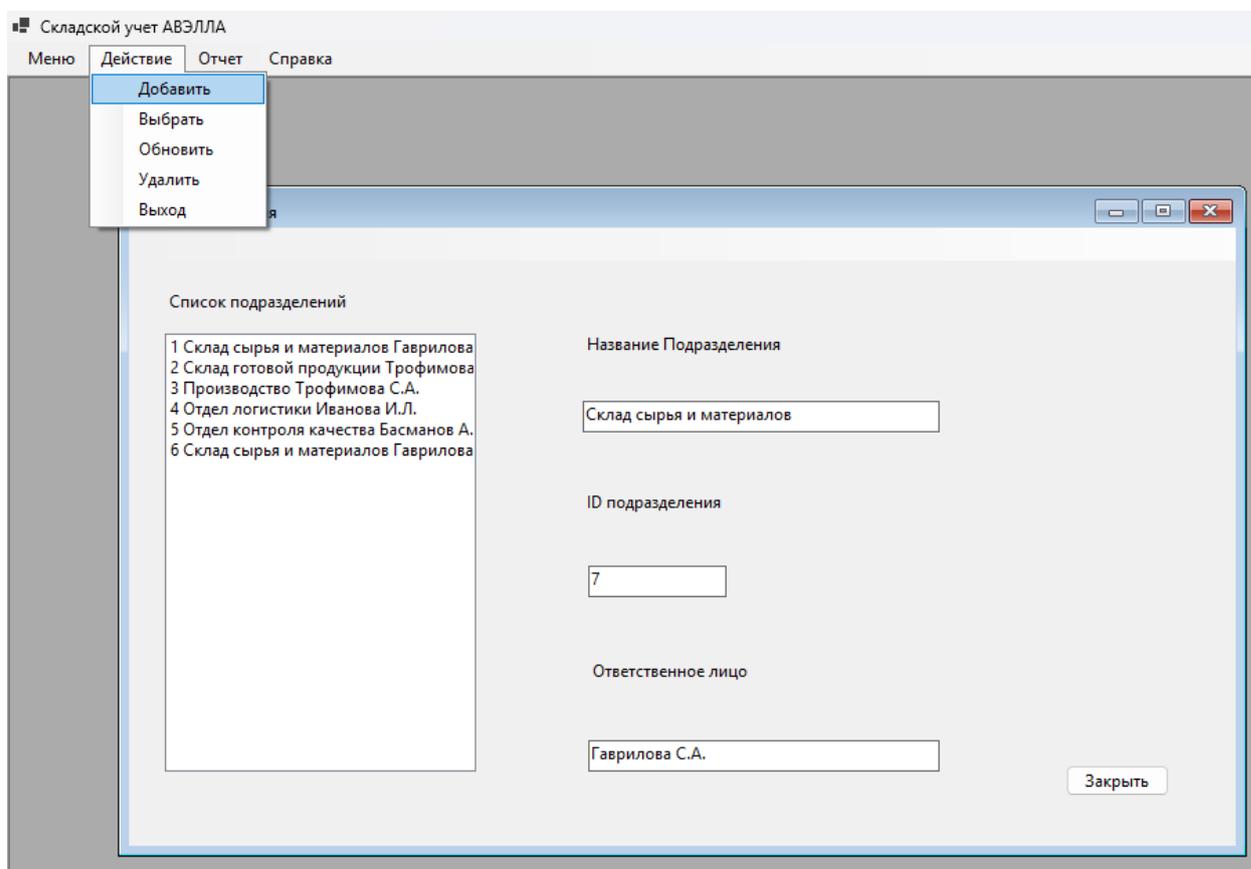


Рисунок 64 – Ввод фамилии ответственного лица склада материалов складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Далее, был протестирован ввод цены входящего сырья, показавший положительный результат и представленный на рисунке 65.

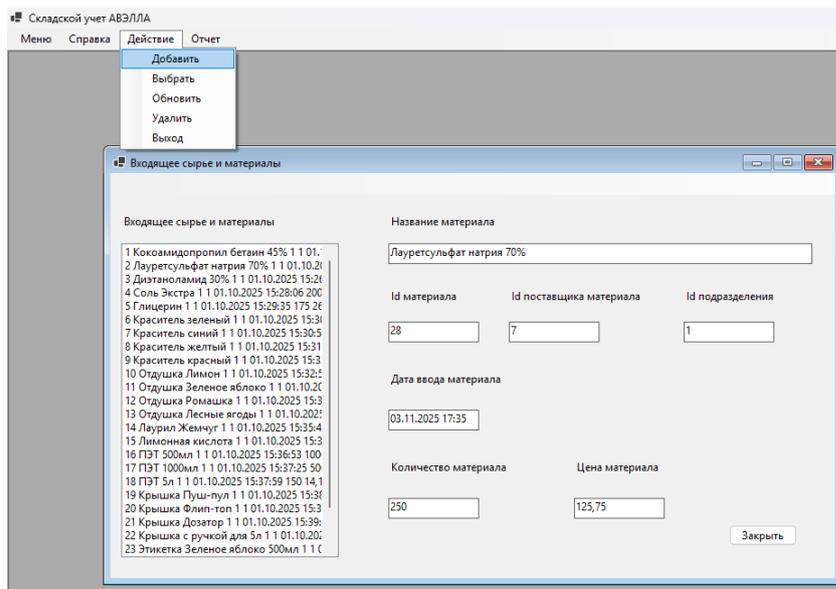


Рисунок 65 – Ввод цены материала в форму входящего сырья складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Выполнение пятого пункта плана тестирования заключалось в проверке ввода покупателей, который также был выполнен положительно и представлен на рисунке 66.

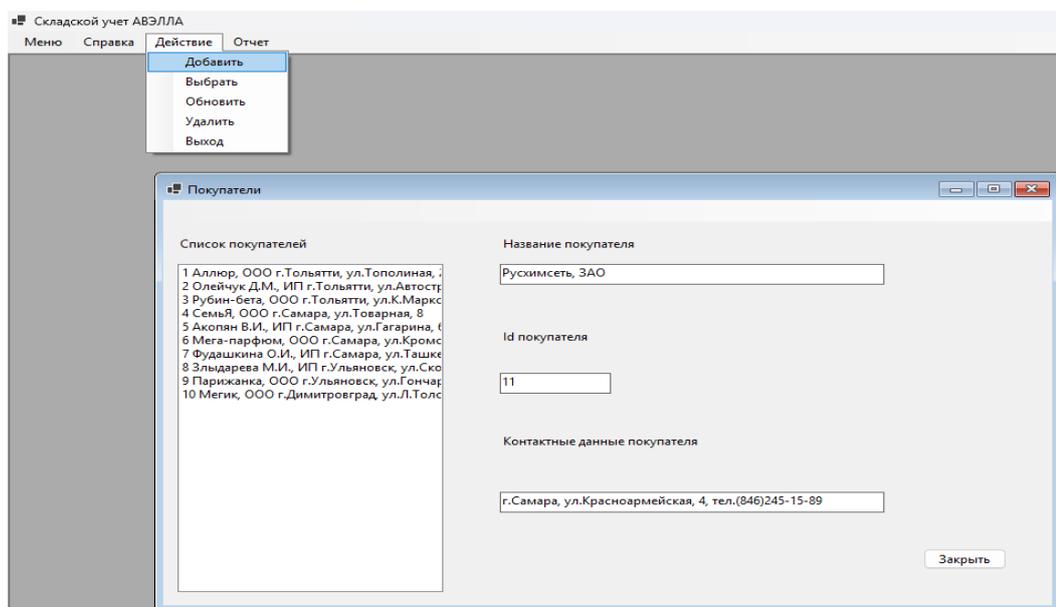


Рисунок 66 – Ввод данных покупателя в форму ввода покупателя складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

В следующем шаге плана тестирования было проведено тестирование учета отгружаемой продукции, в результате которого в форме было обозначено название новой продукции, представленное на рисунке 67.

Складской учет АВЭЛЛА
Меню Справка Действие Отчет

Отпускаемая продукция

Отпускаемая продукция

1 Жидкое мыло 500мл Зеленое Яблоко
2 Жидкое мыло 500мл Морской бриз
3 Жидкое мыло 500мл Полевая ромаш
4 Жидкое мыло 500мл Лесные ягоды 4 4
5 Жидкое мыло 500мл Березовое 5 5 04
6 Жидкое мыло 1000мл Зеленое яблок
7 Жидкое мыло 1000мл Морской бриз
8 Жидкое мыло 1000мл Полевая ромаш
9 Жидкое мыло 1000мл Лесные ягоды
10 Жидкое мыло 1000мл Березовое 10 1
11 Жидкое мыло 5л Зеленое яблоко 1 1
12 Жидкое мыло 5л Морской бриз 2 2 0
13 Жидкое мыло 5л Полевая ромашка
14 Жидкое мыло 5л Лесные ягоды 4 4 0
15 Жидкое мыло 5л Березовое 5 5 04.10
16 Средство для посуды 500мл Лимон 6
17 Средство для посуды 500мл Антибак
18 Средство для посуды 500мл Свежест
19 Средство для посуды 500мл Алое веж
20 Средство для посуды 500мл Грейпфр
21 Средство для посуды 1000мл Лимон
22 Средство для посуды 1000мл Антиба
23 Средство для посуды 1000мл Свежес

Название продукции
Введите название продукции

Id продукции Id покупателя Id подразделения

Дата отгрузки продукции

Количество продукции Цена продукции

Закреть

Рисунок 67 – Ввод названия готовой продукции в форму отпускаемой продукции складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Далее, также была проверена возможность ввода количества и цены готовой продукции, показавшая положительный результат, представленная на рисунке 68.

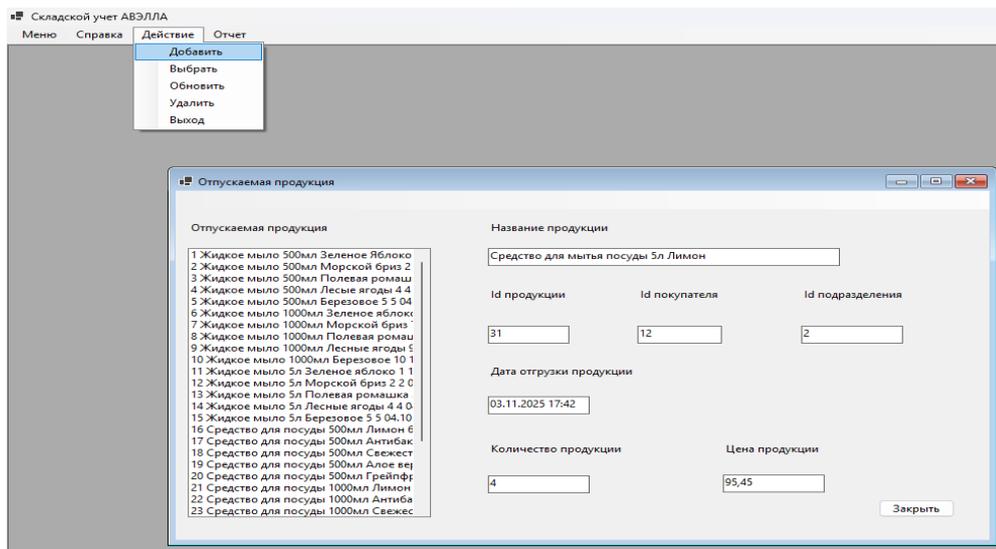


Рисунок 68 – Ввод цены и количества новой продукции в форме ввода отпускаемых продуктов складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

В рамках следующего шага было проведено тестирование ввода подразделения склада готовой продукции, показавшее положительный результат и представленное на рисунке 69.

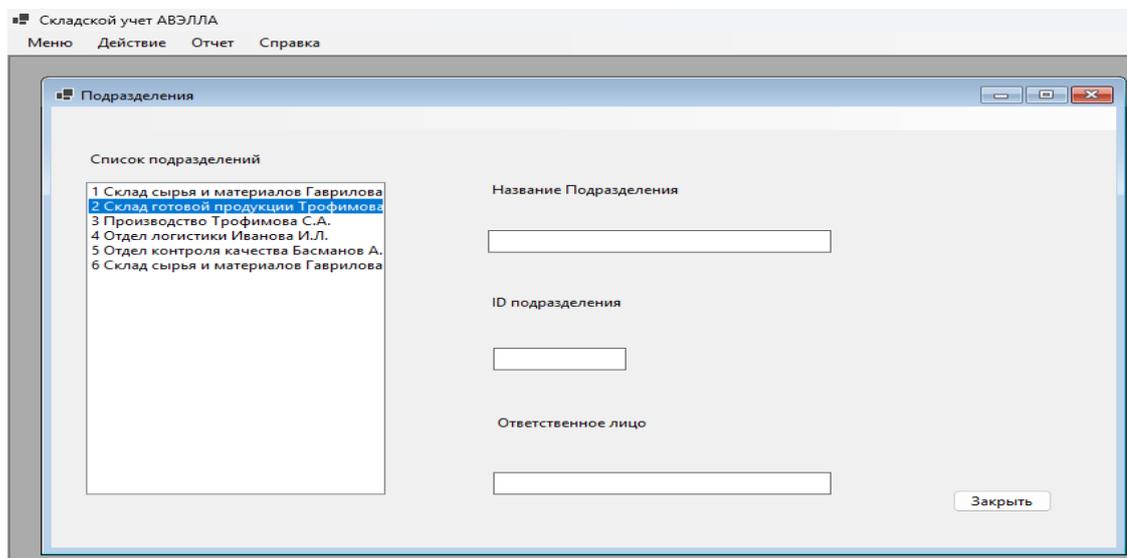


Рисунок 69 – Ввод склада готовой продукции в форму ввода подразделений складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Далее, была проверена возможность ввода фамилии ответственного лица в форме ввода нового подразделения склада готовой продукции, положительный результат которого представлен на рисунке 70.

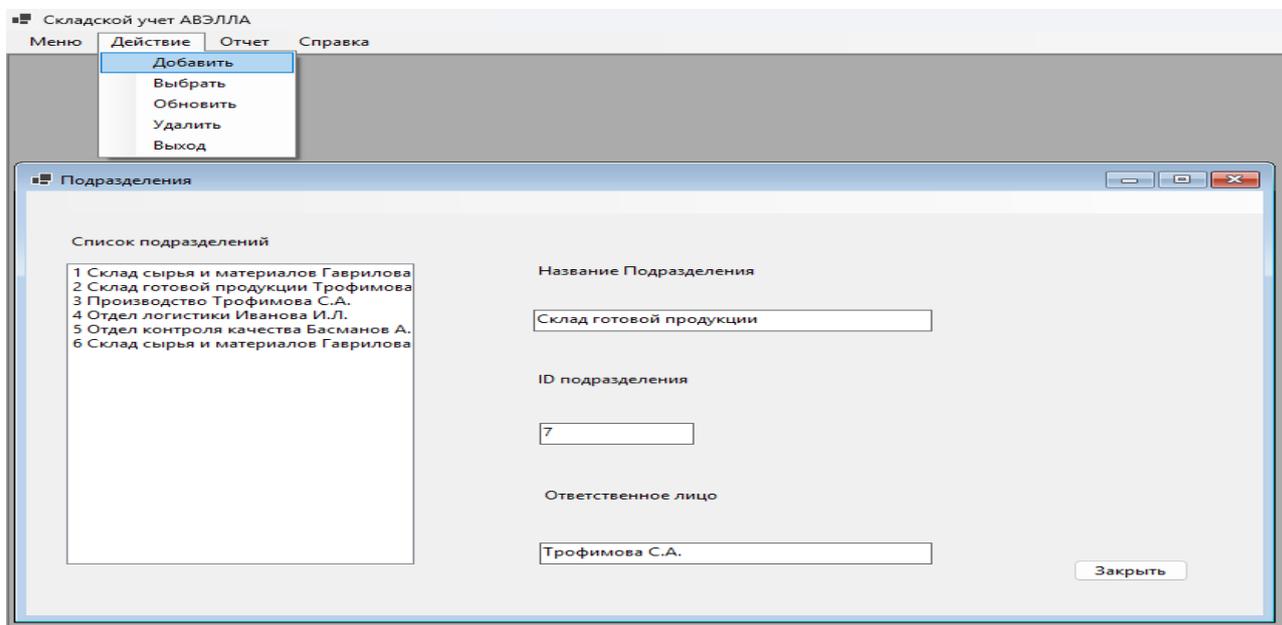


Рисунок 70 – Ввод фамилии ответственного лица в форму ввода склада готовой продукции складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Далее, была протестирована возможность формирования отчетности, в результате выполнения которой были сформированы отчеты о подразделениях, сырье, поставщиках, покупателях и отгружаемой продукции, что представлено на рисунках 71, 72, 73, 74 и 75.

Id	Name	Responsible	Comment
1	Склад сырья и материалов	Гаврилова В.Д.	
2	Склад готовой продукции	Трофимова С.А.	
3	Производство	Трофимова С.А.	
4	Отдел логистики	Иванова И.Л.	
5	Отдел контроля качества	Басманов А.Г.	

Рисунок 71 – Формирование отчетности по подразделениям складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Id	Name	Contact Info	Comment
1	Русхимсеть, ЗАО	г. Самара, ул. Красноармейская, 1, (846)300-01-55	
2	ETC, ООО	г. Казань, Сибирский тр., 56, (843)500-05-86	
3	ВитаХим, ЗАО	г. Самара, ул. Товарная, 15, (846)115-45-87	
4	РусАлХим, ЗАО	г. Москва, ул. Скобелевская, 19, (495)968-00-10	
5	Вита-реактив	г. Дзержинск, ул. Пенина, 15	

Рисунок 72 – Формирование отчетности по поставщикам складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Отчет по сырью и материалам

Id	Name	Supplier Id	Department Id	Date	Quantity	Price
1	Кокоамидо пропилен бетаин 45%	1	1	01.10.2025 15:15:42	215	201,15
2	Лауретсульфат натрия 70%	1	1	01.10.2025 15:18:59	125	175,24
3	Дизитанола мид 30%	1	1	01.10.2025 15:26:59	215	292,32
4	Соль Экстра	1	1	01.10.2025 15:28:06	200	15,75
5	Глицерин	1	1	01.10.2025 15:29:35	175	269,52
6	Краситель зеленый	1	1	01.10.2025 15:30:26	1	8550
7	Краситель синий	1	1	01.10.2025 15:30:54	1	8256

Сформировать

Рисунок 73 – Формирование отчетности по сырью складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Отчет по покупателям

Id	Name	Contact Info	Comment
1	Аплюр, ООО	г.Тольятти, ул.Тополиная, 24	
2	Олейчук Д.М., ИП	г.Тольятти, ул.Автостроителей, 68	
3	Рубин-бета, ООО	г.Тольятти, ул.К.Маркса, 66	
4	Семья, ООО	г.Самара, ул.Товарная, 8	
5	Акопян В.И., ИП	г.Самара, ул.Гагарина, 65	
6	Мега-парфюм, ООО	г.Самара, ул.Кромская, 4	

Сформировать

Рисунок 74 – Формирование отчетности по покупателям складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Id	Name	Buyers Id	Department Id	Date	Quantity	Price
1	Жидкое мыло 500мл Зеленое Яблоко	1	1	04.10.2025 10:50:46	12	25,16
2	Жидкое мыло 500мл Морской бриз	2	2	04.10.2025 10:55:01	12	25,16
3	Жидкое мыло 500мл Полевая ромашка	3	3	04.10.2025 10:59:07	12	25,16

Рисунок 75 – Формирование отчетности по продукции складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

И, наконец, была протестирована возможность распечатывания и сохранения в форматах Word, PDF, Excel отчетности по подразделениям, поставщикам, входящему сырью, покупателям и отпускаемой продукции. Отчетность была распечатана, однако, Visual Studio 2022 не предоставляет возможность предварительного просмотра печати, поэтому были сохранены скрины файлов в формате Word, PDF, Excel отчетности по подразделениям, поставщикам, сырью, покупателям и продукции, что представлено на рисунках 76, 77, 78, 79, 80 и 81.

Id	Name	Responsible	Comment
1	Склад сырья и материалов	Гаврилова В.Д.	
2	Склад готовой продукции	Трофимова С.А.	
3	Производство	Трофимова С.А.	
4	Отдел логистики	Иванова И.Л.	
5	Отдел контроля качества	Басманов А.Г.	

Рисунок 76 – Скрин сохраненных файлов отчетности по подразделениям в формате Excel складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

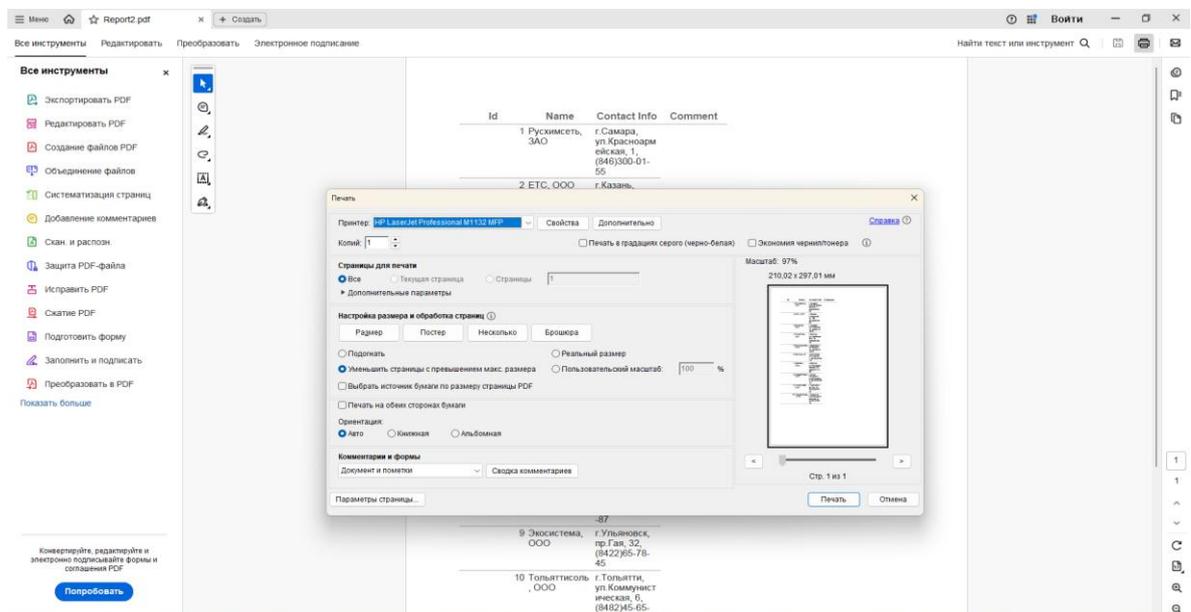


Рисунок 77 – Скрин сохраненных файлов отчетности по поставщикам в формате PDF и предварительный просмотр печати складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Отчет по сырью и материалам

Id	Name	Supplier Id	Department Id	Date	Quantity	Price
1	Кокоамидо пропил бетаин 45%	1	1	01.10.2025 15:15:42	215	201,15
2	Лауретсульфат натрия 70%	1	1	01.10.2025 15:18:59	125	175,24
3	Диэтанола мид 30%	1	1	01.10.2025 15:26:59	215	292,32
4	Соль Экстра	1	1	01.10.2025 15:28:06	200	15,75
5	Глицерин	1	1	01.10.2025 15:29:35	175	269,52
6	Краситель зеленый	1	1	01.10.2025 15:30:26	1	8550
7	Краситель синий	1	1	01.10.2025 15:30:54	1	8256

Сформировать

Рисунок 78 – Скрин сохраненных файлов отчетности по сырью в формате Word складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Автосохранение Report3 - режим совместности - Сохранено в этот компьютер

Id	Name	Supplier Id	Department Id	Date	Quantity	Price	Id
1:	Кокоамидо пропил бетаин 45%	1:	1:	01.10.2025 15:15:42:	215:	201,15:	:
2:	Лауретсульфат натрия 70%	1:	1:	01.10.2025 15:18:59:	125:	175,24:	:
3:	Диэтаноламид 30%	1:	1:	01.10.2025 15:26:59:	215:	292,32:	:
4:	Соль Экстра	1:	1:	01.10.2025 15:28:06:	200:	15,75:	:
5:	Глицерин	1:	1:	01.10.2025 15:29:35:	175:	269,52:	:
6:	Краситель зеленый	1:	1:	01.10.2025 15:30:26:	1:	8550:	:
7:	Краситель синий	1:	1:	01.10.2025 15:30:54:	1:	8256:	:
8:	Краситель желтый	1:	1:	01.10.2025 15:31:32:	1:	7985:	:
9:	Краситель красный	1:	1:	01.10.2025 15:32:04:	1:	10985:	:
10:	Соль Экстра	1:	1:	01.10.2025 15:32:04:	1:	1244:	:

Страница 1 из 2 Число слов: 271 русский Предложения текста: Все Специальные возможности: не поддерживаются

Рисунок 79 – Скрин сохраненных файлов отчетности по сырью в формате Word складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

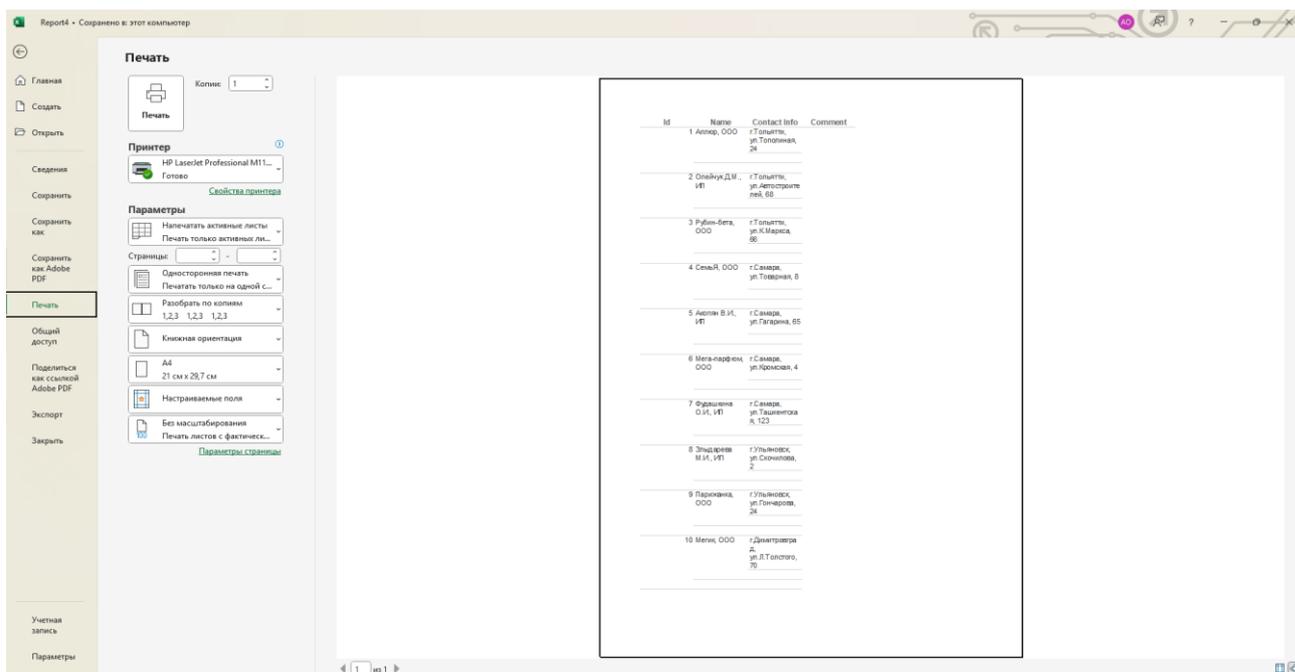


Рисунок 80 – Скрин предварительного просмотра печати сохраненного файла отчетности по покупателям в формате Excel складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

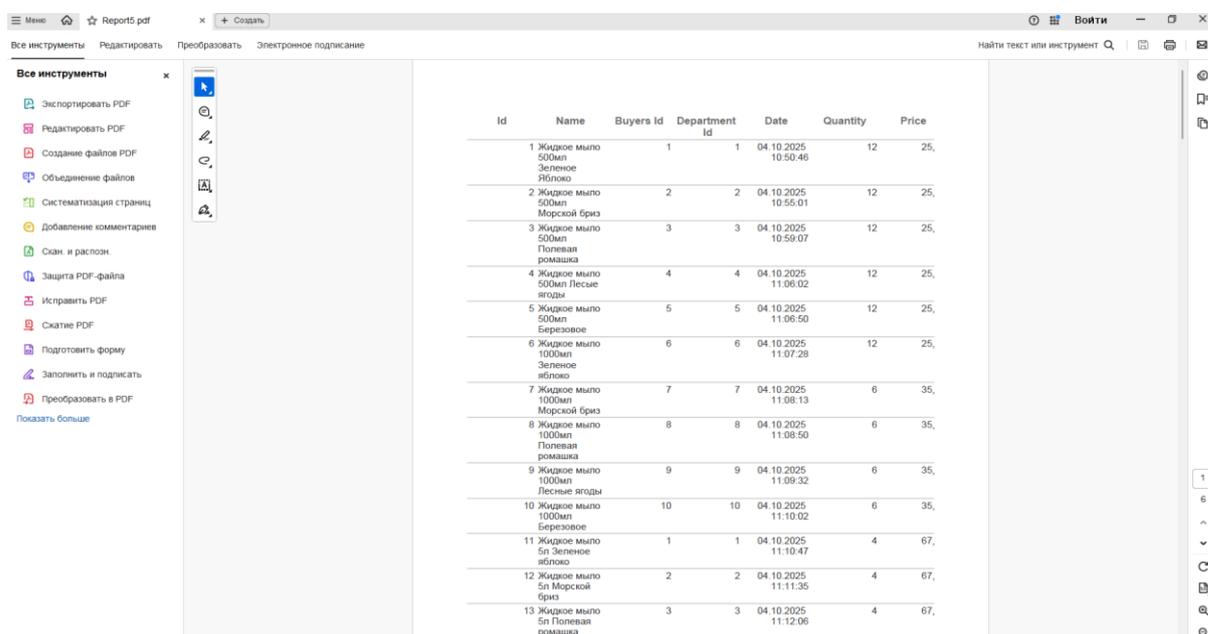


Рисунок 81 – Скрин сохраненного файла отчетности по продукции в формате PDF складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Результаты тестирования были обобщены в таблице 7 результатов тестирования программного обеспечения.

Таблица 7 – Результат тестирования программного обеспечения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА»

Сценарий (test scenario)	Действия (actions)	Ожидаемый результат (expected results)	Полученный результат (actual results)
Проверить возможность ввода названия подразделения	Ввести в Форму Название подразделения	В форме ввода будет введено полное название подразделения	Положительно
Проверить возможность ввода названия поставщика	Ввести в Форму идентификационный номер поставщика	В форме присутствует название и ID поставщика	Положительно
Протестировать учет материалов	Ввести в форму название и количество материала	В форме ввода сырья и материалов введено название и количество материала	Положительно
Проверить ввод фамилии ответственного лица склада сырья и материалов	Ввести в форму ввода подразделения склада сырья и материалов ФИО ответственного лица	В форме подразделения склада сырья и материалов обозначено ФИО ответственного лица	Положительно
Протестировать ввод цены входящего сырья	Ввести в форму ввода сырья и материалов цену входящего сырья	В форме входящего сырья и материалов обозначена цена входящего сырья	Положительно
Проверить ввод покупателей продукции	Ввести в форму название покупателя и его идентификационный номер	В форме покупателей введено название и ID покупателя	Положительно
Протестировать учет отгружаемой продукции	Ввести в форму отгрузки готовой продукции название продукции	В форме готовой продукции обозначено название продукции	Положительно
Проверить учет количества и цены отгружаемой продукции	Ввести в форму готовой продукции количество и цену готовой продукции	В форме готовой продукции обозначены количество и цена продукта	Положительно

продолжение таблицы 7

Протестировать учет подразделения склада готовой продукции	Ввести в форму подразделение склада готовой продукции	В форме введено подразделение склада готовой продукции	Положительно
Проверить учет ответственного лица склада готовой продукции	Ввести в форму склада готовой продукции ФИО ответственного лица	В форме склада готовой продукции введено ФИО ответственного лица	Положительно
Протестировать формирование отчета	Сформировать отчет	В сформированном отчете обозначена продукция, дата поставки, количество и цена	Положительно
Протестировать возможность распечатывания отчета и сохранения его в формате Word, PDF, Excel	Распечатать отчет и сохранить его в форматах Word, PDF, Excel	Распечатанный отчет и сохраненный файл отчета в формате WORD, PDF, Excel	Положительно

Таким образом, проведенное тестирование приложения складского учета производственного участка жидких моющих средств ООО «АВЭЛЛА» показало положительный результат, то есть приложение работоспособно и его возможно использовать для ведения складского учета в рамках производственного процесса по производству жидких моющих средств ООО «АВЭЛЛА».

Заключение

В результате выполнения выпускной квалификационной работы было разработано программное приложение, автоматизирующее деятельность складского учета производственно-технологического отдела производства жидких моющих средств.

Совокупность работ в данном направлении состоит из следующих основных результатов:

- проведен анализ деятельности производственно-технологического подразделения по направлению производства жидких моющих средств и сформулирован набор функциональных и нефункциональных требований к разрабатываемому программному приложению;
- проанализирован документооборот складского участка производства жидких моющих средств, на основании которого составлена схема информационных потоков предприятия;
- спроектированы модели предметной области базового и проектного вариантов, построена логическая модель базы данных для автоматизации деятельности склада производственного участка жидких моющих средств;
- проведен выбор оптимального решения задач автоматизации складского участка производства жидких моющих средств;
- разработано программное приложение с использованием языка программирования C# на базе платформы .NET с применением Windows-Forms. Приложение состоит из модулей учета подразделений, поставщиков, входящего сырья и материалов, покупателей и отгружаемой продукции;
- приложение разработано с учетом защиты от пользовательских ошибок и с возможностью вывода на печать и сохранения документов в форматах Word, PDF, Excel;

- проведено тестирование созданного программного приложения складского участка производства жидких моющих средств.

Таким образом, разработанное программное приложение обеспечивает сокращение времени на проведение складских операций, дает возможность исключить пользовательские ошибки и комфортно работать с документацией, и, наконец, позволяет руководителю предприятия эффективнее организовать производство жидких моющих средств. Данные результаты, в итоге, позволяют осуществлять деятельность предприятия более эффективным и оптимальным образом.

Список используемых источников

1. Автоматизация бизнес-процессов на предприятии : практикум для студентов направления специальности 1-27 01 01-01 «Экономика и организация производства (машиностроение)» : в 2 ч. / сост. : О. А. Лавренова, И. Р. Гребенников. Минск : БНТУ, 2020. – 96 с.
2. Автоматизация бизнес-систем: BPMS, CRM, ERP : учебное пособие / под общ. ред. В. М. Лаптева. Екатеринбург: Изд-во Урал. ун-та, 2024. – 152 с.
3. Александрова И.Л. Программирование на языке C#. Учебное пособие / И.Л. Александрова, Д.Н. Тумаков. Казань : Казанский государственный университет, 2017. – 112 с.
4. Бабич А. В. Введение в UML: учебное пособие. Москва : Ай Пи Ар Медиа, 2022. – 198 с.
5. Буч Г. Язык UML. Руководство пользователя. Москва : Академия АйТи, 2022. – 494 с.
6. Вигерс К., Битти Д. Разработка требований к программному обеспечению. М. : Русская редакция, 2019. – 736 с.
7. Гелмерс Скотт А. Microsoft Visio 2013. Шаг за шагом. М. : Эком, 2014. – 612 с.
8. ГОСТ 28195 – 99 «Оценка качества программных средств. Общие положения». – Москва : ИПК Издательство стандартов, 2001.
9. ГОСТ 31696 – 2012 «Продукция косметическая и гигиеническая моющая. Общие технические условия». – Москва : Стандартиформ, 2014.
10. ГОСТ 32478 – 2013 «Товары бытовой химии. Общие технические требования». – Москва : Стандартиформ, 2019.
11. ГОСТ 34.201 – 2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем». – Москва : Российский институт стандартизации, 2021.

12. ГОСТ Р 56920 – 2016 «Системная и программная инженерия. Тестирование программного обеспечения. Часть 1. Понятия и определения». – Москва : Стандартинформ, 2016.

13. ГОСТ Р ИСО/МЭК 9126 – 93 «Информационная технология. Оценка программной продукции. Характеристика качества и руководства по применению». – Москва : ИПК Издательство стандартов, 2004.

14. Гусев К. В. Системная и программная инженерия: методические указания / К. В. Гусев, Ю. А. Воронцов, Е. К. Михайлова. Москва: РТУ МИРЭА, 2021. – 30 с.

15. Зыков С.В. Проектирование и разработка корпоративных информационных систем: учебное пособие. Москва: Ай Пи Ар Медиа, 2023. – 394 с.

16. Каменнова М. С. Моделирование бизнес-процессов: учебник и практикум для вузов. Москва: Издательство Юрайт, 2025. – 534 с.

17. Кознов Д. В. Введение в программную инженерию : учебное пособие. Москва: Ай Пи Ар Медиа, 2020. – 305 с.

18. Коршикова Л. А. Информационные технологии и стандартизация: учебное пособие. Новосибирск: Изд-во НГТУ, 2018. – 76 с.

19. Платформа .NET Framework. Язык C#. : учеб.-метод. пособие / сост.: Якубович Д. А., Еропова Е. С. Владимир : Издательство «Шерлок-пресс», 2018, Ч. 1. – 48 с.

20. Проектирование информационных систем: учебно-методическое пособие / составители О. И. Евдошенко, Ю. С. Андрианова, А. А. Морозова. Астрахань: Астраханский государственный архитектурно-строительный университет, 2022. – 70 с.

21. Рычкова Е.С. Автоматизация бизнес-процессов торгового предприятия: сборник учебно-методических материалов. Благовещенск: Амурский государственный университет, 2017. – 25 с.

22. Сапаров А.Ю. Разработка Windows Forms приложений на языке программирования С++: учебно-методическое пособие / сост.: А.Ю. Сапаров, Ижевск, 2020. – 61 с.

23. IEEE Std 830-1998 “Recommended practice for software requirements specifications”.

24. ISO/IEC/IEEE 29119-1:2013(E) “Software and systems engineering Software testing Part 1:Concepts and definitions”.

25. SWEBOOK v 3.0 Guide to the software engineering body of knowledge / Edited by P. Porque, R.E. Fairly. IEEE Computer society, 2014. – 335 p.

Приложение А

Образец приходного ордера по сырью и материалам

Типовая межотраслевая форма № М-4
Утверждена постановлением Госкомстата
России
от 30.10.97 № 71а

ПРИХОДНЫЙ ОРДЕР № 74/ХМ

Организация **ООО "Авэлла"** по ОКПО **33744595**
 Структурное подразделение **-Склад ТМЦ** Форма по ОКУД **0315003**
 Коды

Дата составления	Код вида операции	Склад	Поставщик		Страховая компания	Корреспондирующий счет		Номер документа	
			наименование	код		счет, субсчет	код аналитического учета	сопроводительного	платежного
16.12.2020	-	№1	ЗАО "Русхимсель"	-	-	10	10.01	ЦБ-49231	-

Материальные ценности		Единица измерения		Количество		Цена, руб. коп.	Сумма без учета НДС, руб. коп.	Сумма НДС, руб. коп.	Всего с учетом НДС, руб. коп.	Номер паспорта	Порядковый номер по складской картотеке
наименование, сорт, размер, марка	номенклатурный номер	код	наименование	по документу	принято						
1	2	3	4	5	6	7	8	9	10	11	12
Лауретсульфат натрия 70%	71	796	кг	90	90	113,00	-	-	10170,00	-	71
Кокамидопропил бетаин 45%	76	796	кг	30	30	135,44			4063,20		76
Краситель желтый водорастворимый О-11	28	796	кг	25	25	80,00			2000,00		28

Рисунок А.1 – Образец приходного ордера по сырью и материалам

Продолжение Приложения А

Оборотная сторона формы № М-4

Материальные ценности		Единица измерения		Количество		Цена, руб.коп.	Сумма без учета НДС, руб.коп.	Сумма НДС, руб.коп.	Всего с учетом НДС, руб.коп.	Номер паспорта	Порядковый номер по складской картотеке
наименование, сорт, размер, марка	номенклатурный номер	код	наименование	по документу	принято						
1	2	3	4	5	6	7	8	9	10	11	12
Итого						X	-	2705,53	16233,20		

Принял Кладовщик  С.А. Трофимова
(должность) (подпись) (расшифровка подписи)

Сдал экспедитор  П.В. Алачихин
(должность) (подпись) (расшифровка подписи)

Рисунок А.2 – Образец приходного ордера по сырью и материалам (обратная сторона)

Приложение В

Образец приходного ордера готовой продукции

Типовая межотраслевая форма № М-4
Утверждена постановлением Госкомстата
России
от 30.10.97 № 71а

ПРИХОДНЫЙ ОРДЕР № 12/ВЛМ

Организация я <u>ООО "Авэлла"</u> Структурное подразделение <u>-Склад ГП</u>	Форма по ОКУД	Коды 0315003	по ОКПО 33744595
--	---------------	------------------------	----------------------------

Дата составления	Код вида операции	Склад	Поставщик		Страховая компания	Корреспондирующий счет		Номер документа	
			наименование	код		счет, субсчет	код аналитического учета	сопроводительного	платежного
11.12.2020	-	№2	ООО "Авэлла"	-	-	10	10.01	-	-

Материальные ценности		Единица измерения		Количество		Цена, руб. коп.	Сумма без учета НДС, руб. коп.	Сумма НДС, руб. коп.	Всего с учетом НДС, руб. коп.	Номер паспорта	Порядковый номер по складской картотеке
наименование, сорт, размер, марка	номенклатурный номер	код	наименование	по документу	принято						
1	2	3	4	5	6	7	8	9	10	11	12
Средство для мытья посуды Лимон 500мл	49	796	шт.	20	20	98,87	-	-	1977,40	-	49
Средство для мытья посуды Яблоко 1000мл	50	796	шт.	10	10	210,00	-	-	2100,00	-	50
Средство для мытья посуды Грейпфрут 5л	51	796	шт.	12	12	414,29	-	-	4971,48	-	51
Средство для мытья посуды Антибактериальное 10л	52	796	шт.	0	0	729,48	-	-	0	-	52
Средство для мытья посуды Алое вера 250мл	64	796	шт.	21	21	45,87	-	-	963,27	-	64

Рисунок В.1 – Образец приходного ордера готовой продукции

Продолжение Приложения В

Оборотная сторона формы № М-4

Материальные ценности		Единица измерения		Количество		Цена, руб.коп.	Сумма без учета НДС, руб.коп.	Сумма НДС, руб.коп.	Всего с учетом НДС, руб.коп.	Номер паспорта	Порядковый номер по складской картотеке
наименование, сорт, размер, марка	номенклатурный номер	код	наименование	по документу	принято						
1	2	3	4	5	6	7	8	9	10	11	12
Итого							X	-	1668,69	10012,15	

Принял Кладовщик  С.А. Трофимова
(должность) (подпись) (расшифровка подписи)

Сдал экспедитор  И.М. Забордин
(должность) (подпись) (расшифровка подписи)

Рисунок В.2 – Образец приходного ордера готовой продукции (обратная сторона)

Приложение Г

Код приложения, отвечающий за создание меню главной формы складского учета ООО «АВЭЛЛА»

```
namespace AVELLAapp
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
        otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
            System.ComponentModel.ComponentResourceManager(typeof(Form1));
            toolStrip1 = new ToolStrip();
            toolStripDropDownButton1 = new ToolStripDropDownButton();
            toolStripMenuItem1 = new ToolStripMenuItem();
            toolStripMenuItem2 = new ToolStripMenuItem();
            toolStripMenuItem3 = new ToolStripMenuItem();
            toolStripMenuItem4 = new ToolStripMenuItem();
            toolStripMenuItem5 = new ToolStripMenuItem();
            toolStripMenuItem6 = new ToolStripMenuItem();
            toolStripMenuItem7 = new ToolStripMenuItem();
            toolStrip1.SuspendLayout();
            SuspendLayout();
            //
            // toolStrip1
            //
            toolStrip1.Items.AddRange(new ToolStripItem[] {
            toolStripDropDownButton1 });
            toolStrip1.Location = new Point(0, 0);
            toolStrip1.Name = "toolStrip1";
            toolStrip1.Size = new Size(800, 25);
            toolStrip1.TabIndex = 0;
            toolStrip1.Text = "toolStrip1";
            //
            // toolStripDropDownButton1
        }
    }
}
```

Продолжение Приложения Г

```
//
    toolStripDropDownButton1.DisplayStyle =
ToolStripItemDisplayStyle.Text;
    toolStripDropDownButton1.DropDownItems.AddRange(new ToolStripItem[] {
toolStripMenuItem1, toolStripMenuItem2, toolStripMenuItem3, toolStripMenuItem4,
toolStripMenuItem5, toolStripMenuItem6, toolStripMenuItem7 });
    toolStripDropDownButton1.Image =
(Image)resources.GetObject("toolStripDropDownButton1.Image");
    toolStripDropDownButton1.ImageTransparentColor = Color.Magenta;
    toolStripDropDownButton1.Name = "toolStripDropDownButton1";
    toolStripDropDownButton1.Size = new Size(54, 22);
    toolStripDropDownButton1.Text = "Меню";
//
// toolStripMenuItem1
//
toolStripMenuItem1.Name = "toolStripMenuItem1";
toolStripMenuItem1.Size = new Size(205, 22);
toolStripMenuItem1.Text = "Подразделения";
//
// toolStripMenuItem2
toolStripMenuItem2.Name = "toolStripMenuItem2";
toolStripMenuItem2.Size = new Size(205, 22);
toolStripMenuItem2.Text = "Поставщики";
//
// toolStripMenuItem3
//
toolStripMenuItem3.Name = "toolStripMenuItem3";
toolStripMenuItem3.Size = new Size(205, 22);
toolStripMenuItem3.Text = "Входящие Материалы";
//
// toolStripMenuItem4
//
toolStripMenuItem4.Name = "toolStripMenuItem4";
toolStripMenuItem4.Size = new Size(205, 22);
toolStripMenuItem4.Text = "Покупатели";
//
// toolStripMenuItem5
//
toolStripMenuItem5.Name = "toolStripMenuItem5";
toolStripMenuItem5.Size = new Size(205, 22);
toolStripMenuItem5.Text = "Отпускаемые продукты";
//
// toolStripMenuItem6
//
toolStripMenuItem6.Name = "toolStripMenuItem6";
toolStripMenuItem6.Size = new Size(205, 22);
toolStripMenuItem6.Text = "Оборотная Ведомость";
//
// toolStripMenuItem7
//
toolStripMenuItem7.Name = "toolStripMenuItem7";
toolStripMenuItem7.Size = new Size(205, 22);
toolStripMenuItem7.Text = "Выход";
//
// Form1
AutoScaleDimensions = new.SizeF(7F, 15F);
AutoScaleMode = AutoScaleMode.Font;
ClientSize = new Size(800, 450);
Controls.Add(toolStrip1);
Name = "Form1";
```

Продолжение Приложения Г

```
Text = "Складской учет АВЭЛЛА";
toolStrip1.ResumeLayout(false);
toolStrip1.PerformLayout();
ResumeLayout(false);
PerformLayout();
}

#endregion

private ToolStrip toolStrip1;
private ToolStripDropDownButton toolStripDropDownButton1;
private ToolStripMenuItem toolStripMenuItem1;
private ToolStripMenuItem toolStripMenuItem2;
private ToolStripMenuItem toolStripMenuItem3;
private ToolStripMenuItem toolStripMenuItem4;
private ToolStripMenuItem toolStripMenuItem5;
private ToolStripMenuItem toolStripMenuItem6;
private ToolStripMenuItem toolStripMenuItem7;
}
}
```

Приложение Д

Код приложения, отвечающий за ввод информации о подразделениях складского учета ООО «АВЭЛЛА»

```
namespace AVELLAapp
{
    partial class Form2
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
        otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            button1 = new Button();
            textBox1 = new TextBox();
            label1 = new Label();
            label2 = new Label();
            maskedTextBox1 = new MaskedTextBox();
            label3 = new Label();
            textBox2 = new TextBox();
            SuspendLayout();
            //
            // button1
            //
            button1.Location = new Point(699, 386);
            button1.Name = "button1";
            button1.Size = new Size(79, 33);
            button1.TabIndex = 0;
            button1.Text = "Заккрыть";
            button1.UseVisualStyleBackColor = true;
            //
            // textBox1
            //
            textBox1.Location = new Point(90, 91);
            textBox1.Name = "textBox1";
            textBox1.Size = new Size(288, 23);
            textBox1.TabIndex = 1;
            textBox1.Text = "Введите название подразделения";
        }
    }
}
```

Продолжение Приложения Д

```
//
// label1
//
label1.AutoSize = true;
label1.Location = new Point(90, 61);
label1.Name = "label1";
label1.Size = new Size(92, 15);
label1.TabIndex = 2;
label1.Text = "Подразделение";
label1.Click += label1_Click;
//
// label2
//
label2.AutoSize = true;
label2.Location = new Point(90, 147);
label2.Name = "label2";
label2.Size = new Size(150, 15);
label2.TabIndex = 3;
label2.Text = "Введите ID подразделения";
//
// maskedTextBox1
//
maskedTextBox1.Location = new Point(90, 187);
maskedTextBox1.Mask = "00000";
maskedTextBox1.Name = "maskedTextBox1";
maskedTextBox1.Size = new Size(141, 23);
maskedTextBox1.TabIndex = 4;
maskedTextBox1.ValidatingType = typeof(int);
//
// label3
//
label3.AutoSize = true;
label3.Location = new Point(90, 245);
label3.Name = "label3";
label3.Size = new Size(119, 15);
label3.TabIndex = 5;
label3.Text = "Ответственное лицо";
//
// textBox2
//
textBox2.Location = new Point(90, 284);
textBox2.Name = "textBox2";
textBox2.Size = new Size(288, 23);
textBox2.TabIndex = 6;
textBox2.Text = "Введите ФИО ответственного лица подразделения";
//
// Form2
AutoSizeDimensions = new SizeF(7F, 15F);
AutoSizeMode = AutoScaleMode.Font;
ClientSize = new Size(800, 450);
Controls.Add(textBox2);
Controls.Add(label3);
Controls.Add(maskedTextBox1);
Controls.Add(label2);
Controls.Add(label1);
Controls.Add(textBox1);
Controls.Add(button1);
Name = "Form2";
Text = "Ввод подразделений";
ResumeLayout(false);
```

Продолжение Приложения Д

```
PerformLayout();
}

#endregion

private Button button1;
private TextBox textBox1;
private Label label1;
private Label label2;
private MaskedTextBox maskedTextBox1;
private Label label3;
private TextBox textBox2;
}
}
```

Код новой формы приложения, отвечающий за учет подразделений складского учета ООО «АВЭЛЛА»

```
public void WarehouseFill()
{
    daDepartments.Fill(dsWarehouse.Departments);
    MessageBox.Show("Method Fill worked out");
}

private void FormDepartments_Load()
{
    listBoxDepartments.Enabled = true;
    /*for (int i = 0; i < 5; i++)
    {
        this.listBoxDepartments.Items.Add(dsWarehouse.Departments[i].Name);
    }*/
    this.listBoxDepartments.DataSource = this.dsWarehouse.Tables["Departments"];
    this.listBoxDepartments.DisplayMember = "Name";

    textBoxNameDepartments.DataBindings.Add("Text", dsWarehouse, "Departments.Name");

    textBoxIdDepartment.DataBindings.Add("Text", dsWarehouse, "Departments.Id");
    //textBoxResponsibleDepartment.DataBindings.Add("Text", dsWarehouse,
    "Departments.Responsible");
}

private void MenuItemUndo_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Undo");
}

private void MenuItemNew_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Create");
    string nameDepartment = ShowPromptDialog("Enter Department name:", "Enter Form");
}
```

Продолжение Приложения Д

```
        DataRow rowDepartment =
this.dsWarehouse.Departments.NewDepartmentsRow();
        //rowDepartment["Id"] = 1;
        rowDepartment["Name"] = nameDepartment; //"Outcome Product Warehouse";

        dsWarehouse.Departments.Rows.Add(rowDepartment);

        int pos = this.dsWarehouse.Departments.Rows.Count + 1;
        this.BindingContext[dsWarehouse, "Departments"].Position = pos;

        DisplayForm(false);
        listBoxDepartments.Enabled = true;
        textBoxNameDepartments.Focus();

    }

    static string ShowPromptDialog(string message, string title)
    {
        Form prompt = new Form();
        prompt.Width = 300;
        prompt.Height = 150;
        prompt.Text = title;
        Label messageLabel = new Label() { Left = 50, Top = 20, Text = message
};
        TextBox textBox = new TextBox() { Left = 50, Top = 50, Width = 200 };
        Button okButton = new Button() { Text = "OK", Left = 50, Top = 80 };
        Button cancelButton = new Button() { Text = "Cancel", Left = 150, Top
= 80 };

        prompt.Controls.Add(messageLabel);
        prompt.Controls.Add(textBox);
        prompt.Controls.Add(okButton);
        prompt.Controls.Add(cancelButton);
        prompt.ShowDialog();
        okButton.Click += okButton_Click;
        return textBox.Text;
    }

    private static void okButton_Click(object? sender, EventArgs e)
    {
        throw new NotImplementedException();
    }

    private void MenuItemEdit_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Method Edit");
    }

    private void MenuItemSave_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Method Save");
        DataSetWarehouse.DepartmentsDataTable ds2 =
(DataSetWarehouse.DepartmentsDataTable)dsWarehouse.Departments.GetChanges(DataRowS
tate.Added);
        if (ds2 != null)
            try
            {
                daDepartments.Update(ds2);
                //ds2.Dispose();
                dsWarehouse.Departments.AcceptChanges();
            }
        }
    }
}
```

Продолжение Приложения Д

```
    }
    catch (Exception x)
    {
        string mes = x.Message;
        MessageBox.Show("Error of adding string in database Warehouse
" + mes, "Warning");
        this.dsWarehouse.Departments.RejectChanges();
    }
}

private void MenuItemRemove_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Remove");
    int pos = -1;
    pos = this.BindingContext[dsWarehouse, "Departments"].Position;
    string mes = textBoxNameDepartments.Text.ToString().Trim();
    MessageBox.Show("Delete data \n of department \n" + mes + "?",
"Warning",
    MessageBoxButtons.YesNo,
    MessageBoxIcon.Warning,
    MessageBoxDefaultButton.Button2);
    this.dsWarehouse.Departments.Rows[pos].Delete();
    if (this.dsWarehouse.Departments.GetChanges(DataRowState.Deleted) !=
null)
    {
        try
        {
            this.daDepartments.Update(dsWarehouse.Departments);
            this.dsWarehouse.Departments.AcceptChanges();
        }
        catch (Exception x)
        {
            string er = x.Message.ToString();
            MessageBox.Show("Error of deleting string in Warehouse database
" + er, "Warning");
            this.dsWarehouse.Departments.RejectChanges();
        }
    }
    else { this.dsWarehouse.Departments.RejectChanges(); }
    /*DisplayForm(true);
    this.dsWarehouse.Departments.Rows[4].Delete();
    DataSetWarehouse.DepartmentsDataTable ds3 =
(DataSetWarehouse.DepartmentsDataTable)dsWarehouse.Departments.GetChanges(DataRowS
tate.Modified);
    this.daDepartments.Update(ds3);
    dsWarehouse.Departments.AcceptChanges();*/
}

private void textBoxNameDepartments_TextChanged(object sender, EventArgs e)
{
    //nameDepartment = textBoxNameDepartments.Text;
}
}
```

Приложение Е

Код приложения, отвечающий за ввод информации о поставщиках складского учета ООО «АВЭЛЛА»

```
namespace AVELLAapp
{
    partial class Form3
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
        otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            button1 = new Button();
            label1 = new Label();
            textBox1 = new TextBox();
            label2 = new Label();
            maskedTextBox1 = new MaskedTextBox();
            label3 = new Label();
            maskedTextBox2 = new MaskedTextBox();
            SuspendLayout();
            //
            // button1
            //
            button1.Location = new Point(689, 393);
            button1.Name = "button1";
            button1.Size = new Size(75, 23);
            button1.TabIndex = 0;
            button1.Text = "Закрѳть";
            button1.UseVisualStyleBackColor = true;
            //
            // label1
            //
            label1.AutoSize = true;
            label1.Location = new Point(64, 33);
            label1.Name = "label1";
            label1.Size = new Size(129, 15);
            label1.TabIndex = 1;
        }
    }
}
```

Продолжение Приложения Е

```
label1.Text = "Название поставщика";
//
// textBox1
//
textBox1.Location = new Point(64, 69);
textBox1.Name = "textBox1";
textBox1.Size = new Size(179, 23);
textBox1.TabIndex = 2;
textBox1.Text = "Введите название поставщика";
//
// label2
//
label2.AutoSize = true;
label2.Location = new Point(64, 122);
label2.Name = "label2";
label2.Size = new Size(88, 15);
label2.TabIndex = 3;
label2.Text = "ID поставщика";
//
// maskedTextBox1
//
maskedTextBox1.Location = new Point(64, 158);
maskedTextBox1.Mask = "00000";
maskedTextBox1.Name = "maskedTextBox1";
maskedTextBox1.Size = new Size(179, 23);
maskedTextBox1.TabIndex = 4;
maskedTextBox1.ValidatingType = typeof(int);
//
// label3
//
label3.AutoSize = true;
label3.Location = new Point(64, 233);
label3.Name = "label3";
label3.Size = new Size(117, 15);
label3.TabIndex = 5;
label3.Text = "Телефонный номер";
//
// maskedTextBox2
//
maskedTextBox2.Location = new Point(64, 281);
maskedTextBox2.Mask = "(999) 000-0000";
maskedTextBox2.Name = "maskedTextBox2";
maskedTextBox2.Size = new Size(100, 23);
maskedTextBox2.TabIndex = 6;
//
// Form3
//
AutoSizeDimensions = new SizeF(7F, 15F);
AutoSizeMode = AutoScaleMode.Font;
ClientSize = new Size(800, 450);
Controls.Add(maskedTextBox2);
Controls.Add(label3);
Controls.Add(maskedTextBox1);
Controls.Add(label2);
Controls.Add(textBox1);
Controls.Add(label1);
Controls.Add(button1);
Name = "Form3";
Text = "Поставщики";
ResumeLayout(false);
PerformLayout();
```

Продолжение Приложения Е

```
    }  
  
#endregion  
  
private Button button1;  
private Label label1;  
private TextBox textBox1;  
private Label label2;  
private MaskedTextBox maskedTextBox1;  
private Label label3;  
private MaskedTextBox maskedTextBox2;  
    }  
}
```

Код новой формы приложения, отвечающий за учет поставщиков сырья и материалов складского учета ООО «АВЭЛЛА»

```
namespace AVELLAWarehouse  
{  
    public partial class FormSuppliers : Form  
    {  
        public FormSuppliers()  
        {  
            InitializeComponent();  
            DisplayReadOnly(true);  
            WarehouseFill();  
            FormDepartments_Load();  
        }  
  
        public void DisplayReadOnly(bool ReadOnly)  
        {  
            this.textBoxNameSupplier.ReadOnly = ReadOnly;  
            this.textBoxIdSupplier.ReadOnly = ReadOnly;  
            this.textBoxContactInfoSupplier.ReadOnly = ReadOnly;  
        }  
  
        public void MenuItemEnabled(bool mode)  
        {  
            undoToolStripMenuItem.Enabled = mode;  
            newToolStripMenuItem.Enabled = mode;  
            editToolStripMenuItem.Enabled = mode;  
            saveToolStripMenuItem.Enabled = !mode;  
            deleteToolStripMenuItem.Enabled = !mode;  
        }  
  
        private void DisplayForm(bool mode)  
        {  
            DisplayReadOnly(mode);  
            MenuItemEnabled(mode);  
        }  
  
        DataSetWarehouse dsWarehouse = new DataSetWarehouse();  
        DataSetWarehouseTableAdapters.SuppliersTableAdapter daSuppliers =  
new AVELLAWarehouse.DataSetWarehouseTableAdapters.SuppliersTableAdapter();  
        public void WarehouseFill()  
        {  
            daSuppliers.Fill(dsWarehouse.Suppliers);  
            MessageBox.Show("Method Fill worked out");  
        }  
    }  
}
```

Продолжение Приложения Е

```
private void FormDepartments_Load()
{
    listBoxSuppliers.Enabled = true;
    this.listBoxSuppliers.DataSource =
this.dsWarehouse.Tables["Suppliers"];
    this.listBoxSuppliers.DisplayMember = "Name";

    textBoxNameSupplier.DataBindings.Add("Text", dsWarehouse,
"Suppliers.Name");

    textBoxIdSupplier.DataBindings.Add("Text", dsWarehouse,
"Suppliers.Id");
    textBoxContactInfoSupplier.DataBindings.Add("Text", dsWarehouse,
"Suppliers.ContactInfo");
}

private void undoToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Undo");
}

private void newToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Create");
    string nameSupplier = ShowPromptDialog("Enter Supplier name:",
"Enter Form");
    DataRow rowSupplier =
this.dsWarehouse.Suppliers.NewSuppliersRow();
    rowSupplier["Name"] = nameSupplier;

    dsWarehouse.Suppliers.Rows.Add(rowSupplier);

    int pos = this.dsWarehouse.Suppliers.Rows.Count + 1;
    this.BindingContext[dsWarehouse, "Suppliers"].Position = pos;

    DisplayForm(false);
    listBoxSuppliers.Enabled = true;
    textBoxNameSupplier.Focus();
}

static string ShowPromptDialog(string message, string title)
{
    Form prompt = new Form();
    prompt.Width = 300;
    prompt.Height = 150;
    prompt.Text = title;
    Label messageLabel = new Label() { Left = 50, Top = 20, Text =
message };
    TextBox textBox = new TextBox() { Left = 50, Top = 50, Width =
200 };
    Button okButton = new Button() { Text = "OK", Left = 50, Top =
80 };
    Button cancelButton = new Button() { Text = "Cancel", Left = 150,
Top = 80 };

    prompt.Controls.Add(messageLabel);
    prompt.Controls.Add(textBox);
    prompt.Controls.Add(okButton);
    prompt.Controls.Add(cancelButton);
}
```

Продолжение Приложения Е

```
prompt.ShowDialog();
okButton.Click += okButton_Click;
return textBox.Text;
}

private static void okButton_Click(object? sender, EventArgs e)
{
    throw new NotImplementedException();
}

private void editToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Edit");
}

private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Save");
    DataSetWarehouse.SuppliersDataTable ds2 =
(DataSetWarehouse.SuppliersDataTable)dsWarehouse.Suppliers.GetChanges(DataRowState
.Added);
    if (ds2 != null)
        try
        {
            daSuppliers.Update(ds2);
            dsWarehouse.Suppliers.AcceptChanges();
        }
        catch (Exception x)
        {
            string mes = x.Message;
            MessageBox.Show("Error of adding string in database
Warehouse " + mes, "Warning");
            this.dsWarehouse.Suppliers.RejectChanges();
        }
}

private void deleteToolStripMenuItem_Click(object sender, EventArgs
e)
{
    MessageBox.Show("Method Remove");
    int pos = -1;
    pos = this.BindingContext[dsWarehouse, "Suppliers"].Position;
    string mes = textBoxNameSupplier.Text.ToString().Trim();
    MessageBox.Show("Delete data \n of supplier \n" + mes + "?",
"Warning",
MessageBoxButtons.YesNo,
MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2);
    this.dsWarehouse.Suppliers.Rows[pos].Delete();
    if (this.dsWarehouse.Suppliers.GetChanges(DataRowState.Deleted)
!= null)
    {
        try
        {
            this.daSuppliers.Update(dsWarehouse.Suppliers);
            this.dsWarehouse.Suppliers.AcceptChanges();
        }
        catch (Exception x)
        {
            string er = x.Message.ToString();

```

Продолжение Приложения Е

```
        MessageBox.Show("Error of deleting string in Warehouse  
database " + er, "Warning");  
  
        this.dsWarehouse.Suppliers.RejectChanges();  
    }  
    }  
else { this.dsWarehouse.Suppliers.RejectChanges(); }  
    }  
    }  
}
```

Код интегрированного отчета по поставщикам сырья и материалов складского учета ООО «АВЕЛЛА»

```
using Microsoft.Reporting.WinForms;  
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Data.SqlClient;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
  
namespace ReportsAVELLA  
{  
    public partial class Form2 : Form  
    {  
        public Form2()  
        {  
            InitializeComponent();  
        }  
  
        private void Form2_Load(object sender, EventArgs e)  
        {  
  
            this.reportViewer1.RefreshReport();  
        }  
        SqlConnection connection = new SqlConnection(@"Data  
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\VisualStudio\source\repos\AVELLA  
Warehouse\AVELLAWarehouse\DatabaseAVELLA.mdf;Integrated Security=True");  
  
        private void button1_Click(object sender, EventArgs e)  
        {  
            SqlCommand command = new SqlCommand("SELECT * FROM Suppliers",  
connection);  
            SqlDataAdapter d = new SqlDataAdapter(command);  
            DataTable dt = new DataTable();  
            d.Fill(dt);  
            reportViewer1.LocalReport.DataSources.Clear();  
        }  
    }  
}
```

Продолжение Приложения Е

```
ReportDataSource source = new ReportDataSource("DataSet1", dt);
reportViewer1.LocalReport.ReportPath =
@"D:\VisualStudio\source\repos\ReportsAVELLA\ReportsAVELLA\Report2.rdlc";
reportViewer1.LocalReport.DataSources.Add(source);
reportViewer1.RefreshReport();
}
}
}
```

Приложение Ж

Код приложения, отвечающий за ввод информации о входящих материалах складского учета ООО «АВЭЛЛА»

```
namespace AVELLAapp
{
    partial class Form4
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            button1 = new Button();
            textBox1 = new TextBox();
            label1 = new Label();
            label2 = new Label();
            maskedTextBox1 = new MaskedTextBox();
            maskedTextBox2 = new MaskedTextBox();
            label3 = new Label();
            SuspendLayout();
            //
            // button1
            //
            button1.Location = new Point(684, 395);
            button1.Name = "button1";
            button1.Size = new Size(75, 23);
            button1.TabIndex = 0;
            button1.Text = "Заккрыть";
            button1.UseVisualStyleBackColor = true;
            //
            // textBox1
            //
            textBox1.Location = new Point(65, 81);
            textBox1.Name = "textBox1";
            textBox1.Size = new Size(186, 23);
            textBox1.TabIndex = 1;
            textBox1.Text = "Введите входящие материалы";
        }
    }
}
```

Продолжение Приложения Ж

```
//  
// label1  
//  
label1.AutoSize = true;  
label1.Location = new Point(65, 45);  
label1.Name = "label1";  
label1.Size = new Size(152, 15);  
label1.TabIndex = 2;  
label1.Text = "Наименование материала";  
//  
// label2  
//  
label2.AutoSize = true;  
label2.Location = new Point(65, 129);  
label2.Name = "label2";  
label2.Size = new Size(72, 15);  
label2.TabIndex = 3;  
label2.Text = "Количество";  
//  
// maskedTextBox1  
//  
maskedTextBox1.Location = new Point(65, 165);  
maskedTextBox1.Mask = "00000";  
maskedTextBox1.Name = "maskedTextBox1";  
maskedTextBox1.Size = new Size(100, 23);  
maskedTextBox1.TabIndex = 4;  
maskedTextBox1.ValidatingType = typeof(int);  
//  
// maskedTextBox2  
//  
maskedTextBox2.Location = new Point(65, 243);  
maskedTextBox2.Mask = "00000";  
maskedTextBox2.Name = "maskedTextBox2";  
maskedTextBox2.Size = new Size(100, 23);  
maskedTextBox2.TabIndex = 5;  
maskedTextBox2.ValidatingType = typeof(int);  
//  
// label3  
//  
label3.AutoSize = true;  
label3.Location = new Point(65, 209);  
label3.Name = "label3";  
label3.Size = new Size(35, 15);  
label3.TabIndex = 6;  
label3.Text = "Цена";  
label3.Click += label3_Click;  
//  
// Form4  
//  
AutoScaleDimensions = new.SizeF(7F, 15F);  
AutoScaleMode = AutoScaleMode.Font;  
ClientSize = new Size(800, 450);  
Controls.Add(label3);  
Controls.Add(maskedTextBox2);  
Controls.Add(maskedTextBox1);  
Controls.Add(label2);  
Controls.Add(label1);  
Controls.Add(textBox1);  
Controls.Add(button1);  
Name = "Form4";  
Text = "Ввод входящих материалов";
```

Продолжение Приложения Ж

```
        ResumeLayout (false);
        PerformLayout ();
    }

    #endregion

    private Button button1;
    private TextBox textBox1;
    private Label label1;
    private Label label2;
    private MaskedTextBox maskedTextBox1;
    private MaskedTextBox maskedTextBox2;
    private Label label3;
}
}
```

Код новой формы приложения, отвечающий за учет входящего сырья и материалов складского учета ООО «АВЭЛЛА»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Security.Cryptography.X509Certificates;
using System.Data.Common;

namespace AVELLAWarehouse
{
    public partial class FormIncomeMaterial : Form
    {
        public FormIncomeMaterial()
        {
            InitializeComponent();
            DisplayReadOnly(true);
            WarehouseFill();
            FormIncomeMaterial_Load();
        }

        public void DisplayReadOnly(bool ReadOnly)
        {
            this.textBoxNameIncomeMaterial.ReadOnly = ReadOnly;
            this.textBoxIdIncomeMaterial.ReadOnly = ReadOnly;
            this.textBoxQuantityIncomeMaterial.ReadOnly = ReadOnly;
            this.textBoxPriceIncomeMaterial.ReadOnly = ReadOnly;
        }

        public void MenuItemEnabled(bool mode)
        {

```

Продолжение Приложения Ж

```
        editToolStripMenuItem.Enabled = mode;
        deleteToolStripMenuItem.Enabled = mode;
        createToolStripMenuItem.Enabled = mode;
        saveToolStripMenuItem.Enabled = !mode;
        undoToolStripMenuItem.Enabled = !mode;
    }

    private void DisplayForm(bool mode)
    {
        DisplayReadOnly(mode);
        MenuItemEnabled(mode);
    }

    DataSetWarehouse dsWarehouse = new DataSetWarehouse();
    DataSetWarehouseTableAdapters.IncomeMaterialTableAdapter daIncomeMaterial
= new AVELLAWarehouse.DataSetWarehouseTableAdapters.IncomeMaterialTableAdapter();

    public void WarehouseFill()
    {
        daIncomeMaterial.Fill(dsWarehouse.IncomeMaterial);
        MessageBox.Show("Method Fill worked out");
    }

    private void FormIncomeMaterial_Load()
    {
        listBoxIncomeMaterial.Enabled = true;
        this.listBoxIncomeMaterial.DataSource =
this.dsWarehouse.Tables["IncomeMaterial"];
        this.listBoxIncomeMaterial.DisplayMember = "Name";

        textBoxNameIncomeMaterial.DataBindings.Add("Text", dsWarehouse,
"IncomeMaterial.Name");
        textBoxIdIncomeMaterial.DataBindings.Add("Text", dsWarehouse,
"IncomeMaterial.Id");
        textBoxQuantityIncomeMaterial.DataBindings.Add("Text", dsWarehouse,
"IncomeMaterial.Quantity");
        textBoxPriceIncomeMaterial.DataBindings.Add("Text", dsWarehouse,
"IncomeMaterial.Price");
    }

    private void undoToolStripMenuItem_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Method Undo");
    }

    private void createToolStripMenuItem_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Method Create");
        string nameIncomeMaterial = ShowPromptDialog("Enter Income Material
name:", "Enter Form");
        DataRow rowIncomeMaterial =
this.dsWarehouse.IncomeMaterial.NewIncomeMaterialRow();
        rowIncomeMaterial["Name"] = nameIncomeMaterial;

        dsWarehouse.IncomeMaterial.Rows.Add(rowIncomeMaterial);

        int pos = this.dsWarehouse.IncomeMaterial.Rows.Count + 1;
        this.BindingContext[dsWarehouse, "IncomeMaterial"].Position = pos;

        DisplayForm(false);
    }
}
```

Продолжение Приложения Ж

```
listBoxIncomeMaterial.Enabled = true;
textBoxNameIncomeMaterial.Focus();

}

static string ShowPromptDialog(string message, string title)
{
    Form prompt = new Form();
    prompt.Width = 300;
    prompt.Height = 150;
    prompt.Text = title;
    Label messageLabel = new Label() { Left = 50, Top = 20, Text =
message };
    TextBox textBox = new TextBox() { Left = 50, Top = 50, Width =
200 };
    Button okButton = new Button() { Text = "OK", Left = 50, Top =
80 };
    Button cancelButton = new Button() { Text = "Cancel", Left = 150,
Top = 80 };

    prompt.Controls.Add(messageLabel);
    prompt.Controls.Add(textBox);
    prompt.Controls.Add(okButton);
    prompt.Controls.Add(cancelButton);
    prompt.ShowDialog();
    okButton.Click += okButton_Click;
    return textBox.Text;
}

private static void okButton_Click(object? sender, EventArgs e)
{
    throw new NotImplementedException();
}

private void editToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Edit");
}

private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Save");
    DataSetWarehouse.IncomeMaterialDataTable ds4 =
(DataSetWarehouse.IncomeMaterialDataTable)dsWarehouse.IncomeMaterial.GetChanges(Da
taRowState.Added);
    if (ds4 != null)
    {
        try
        {
            daIncomeMaterial.Update(ds4);
            dsWarehouse.IncomeMaterial.AcceptChanges();
        }
        catch (Exception x)
        {
            string mes = x.Message;
            MessageBox.Show("Error of adding string in database
Warehouse " + mes, "Warning");
            this.dsWarehouse.Departments.RejectChanges();
        }
    }
}
```

Продолжение Приложения Ж

```
e) private void deleteToolStripMenuItem_Click(object sender, EventArgs
    {
        MessageBox.Show("Method Remove");
        int pos = -1;
        pos = this.BindingContext[dsWarehouse,
"IncomeMaterial"].Position;
        string mes = textBoxNameIncomeMaterial.Text.ToString().Trim();
        MessageBox.Show("Delete data \n of department \n" + mes + "?",
"Warning",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2);
        this.dsWarehouse.IncomeMaterial.Rows[pos].Delete();
        if
(this.dsWarehouse.IncomeMaterial.GetChanges(DataRowState.Deleted) != null)
        {
            try
            {
                this.daIncomeMaterial.Update(dsWarehouse.IncomeMaterial);
                this.dsWarehouse.IncomeMaterial.AcceptChanges();
            }
            catch (Exception x)
            {
                string er = x.Message.ToString();
                MessageBox.Show("Error of deleting string in Warehouse
database " + er, "Warning");
                this.dsWarehouse.IncomeMaterial.RejectChanges();
            }
        }
        else { this.dsWarehouse.IncomeMaterial.RejectChanges(); }
    }
}
```

Код интегрированного отчета по входящему сырью и материалам складского учета ООО «АВЭЛЛА»

```
using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ReportsAVELLA
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }
    }
}
```

Продолжение Приложения Ж

```
private void Form3_Load(object sender, EventArgs e)
{
    this.reportViewer1.RefreshReport();
}

SqlConnection connection = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\VisualStudio\source\repos\AVELLA
Warehouse\AVELLAWarehouse\DatabaseAVELLA.mdf;Integrated Security=True");

private void button1_Click(object sender, EventArgs e)
{
    SqlCommand command = new SqlCommand("SELECT * FROM
IncomeMaterial", connection);
    SqlDataAdapter d = new SqlDataAdapter(command);
    DataTable dt = new DataTable();
    d.Fill(dt);
    reportViewer1.LocalReport.DataSources.Clear();
    ReportDataSource source = new ReportDataSource("DataSet1", dt);
    reportViewer1.LocalReport.ReportPath =
@"D:\VisualStudio\source\repos\ReportsAVELLA\ReportsAVELLA\Report3.rdlc";
    reportViewer1.LocalReport.DataSources.Add(source);
    reportViewer1.RefreshReport();
}
}
```

Приложение И

Код приложения, отвечающий за ввод информации о покупателях складского учета ООО «АВЭЛЛА»

```
namespace AVELLAapp
{
    partial class Form5
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
        otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            label1 = new Label();
            button1 = new Button();
            textBox1 = new TextBox();
            label2 = new Label();
            maskedTextBox1 = new MaskedTextBox();
            label3 = new Label();
            maskedTextBox2 = new MaskedTextBox();
            SuspendLayout();
            //
            // label1
            //
            label1.AutoSize = true;
            label1.Location = new Point(75, 51);
            label1.Name = "label1";
            label1.Size = new Size(125, 15);
            label1.TabIndex = 0;
            label1.Text = "Название покупателя";
            //
            // button1
            //
            button1.Location = new Point(686, 387);
            button1.Name = "button1";
            button1.Size = new Size(75, 23);
            button1.TabIndex = 1;
            button1.Text = "Закрыть";
        }
    }
}
```

Продолжение Приложения И

```
button1.UseVisualStyleBackColor = true;
//
// textBox1
//
textBox1.Location = new Point(75, 97);
textBox1.Name = "textBox1";
textBox1.Size = new Size(194, 23);
textBox1.TabIndex = 2;
textBox1.Text = "Введите название покупателя";
textBox1.TextChanged += textBox1_TextChanged;
//
// label2
//
label2.AutoSize = true;
label2.Location = new Point(75, 160);
label2.Name = "label2";
label2.Size = new Size(84, 15);
label2.TabIndex = 3;
label2.Text = "ID покупателя";
//
// maskedTextBox1
//
maskedTextBox1.Location = new Point(75, 199);
maskedTextBox1.Mask = "00000";
maskedTextBox1.Name = "maskedTextBox1";
maskedTextBox1.Size = new Size(100, 23);
maskedTextBox1.TabIndex = 4;
maskedTextBox1.ValidatingType = typeof(int);
//
// label3
//
label3.AutoSize = true;
label3.Location = new Point(75, 281);
label3.Name = "label3";
label3.Size = new Size(117, 15);
label3.TabIndex = 5;
label3.Text = "Телефонный номер";
//
// maskedTextBox2
//
maskedTextBox2.Location = new Point(75, 331);
maskedTextBox2.Mask = "(999) 000-0000";
maskedTextBox2.Name = "maskedTextBox2";
maskedTextBox2.Size = new Size(133, 23);
maskedTextBox2.TabIndex = 6;
//
// Form5
//
AutoScaleDimensions = new.SizeF(7F, 15F);
AutoScaleMode = AutoScaleMode.Font;
ClientSize = new Size(800, 450);
Controls.Add(maskedTextBox2);
Controls.Add(label3);
Controls.Add(maskedTextBox1);
Controls.Add(label2);
Controls.Add(textBox1);
Controls.Add(button1);
Controls.Add(label1);
Name = "Form5";
Text = "Ввод Покупателя";
```

Продолжение Приложения И

```
        ResumeLayout(false);
        PerformLayout();
    }

    #endregion

    private Label label1;
    private Button button1;
    private TextBox textBox1;
    private Label label2;
    private MaskedTextBox maskedTextBox1;
    private Label label3;
    private MaskedTextBox maskedTextBox2;
}
}
```

Код новой формы приложения, отвечающий за учет покупателей складского учета ООО «АВЭЛЛА»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Security.Cryptography.X509Certificates;
using System.Data.Common;

namespace AVELLAWarehouse
{
    public partial class FormBuyers : Form
    {
        public FormBuyers()
        {
            InitializeComponent();
            DisplayReadOnly(true);
            WarehouseFill();
            FormBuyers_Load();
        }

        public void DisplayReadOnly(bool ReadOnly)
        {
            this.textBoxNameBuyer.ReadOnly = ReadOnly;
            this.textBoxIdBuyer.ReadOnly = ReadOnly;
            this.textBoxContactInfoBuyer.ReadOnly = ReadOnly;
        }

        public void MenuItemEnabled(bool mode)
        {
            undoToolStripMenuItem.Enabled = mode;
            newToolStripMenuItemBuyer.Enabled = mode;
            editToolStripMenuItem.Enabled = mode;
        }
    }
}
```

Продолжение Приложения И

```
        saveToolStripMenuItem.Enabled = !mode;
        deleteToolStripMenuItem.Enabled = !mode;
    }

    private void DisplayForm(bool mode)
    {
        DisplayReadOnly(mode);
        MenuItemEnabled(mode);
    }

    DataSetWarehouse dsWarehouse = new DataSetWarehouse();
    DataSetWarehouseTableAdapters.BuyersTableAdapter daBuyers = new
    AVELLAWarehouse.DataSetWarehouseTableAdapters.BuyersTableAdapter();

    public void WarehouseFill()
    {
        daBuyers.Fill(dsWarehouse.Buyers);
        MessageBox.Show("Method Fill worked out");
    }

    private void FormBuyers_Load()
    {
        listBoxBuyers.Enabled = true;
        this.listBoxBuyers.DataSource = this.dsWarehouse.Tables["Buyers"];
        this.listBoxBuyers.DisplayMember = "Name";

        textBoxNameBuyer.DataBindings.Add("Text", dsWarehouse, "Buyers.Name");

        textBoxIdBuyer.DataBindings.Add("Text", dsWarehouse, "Buyers.Id");
        textBoxContactInfoBuyer.DataBindings.Add("Text", dsWarehouse,
"Buyers.ContactInfo");
    }

    private void undoToolStripMenuItem_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Method Undo");
    }

    private void newToolStripMenuItemBuyer_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Method Create");
        string nameBuyer = ShowPromptDialog("Enter Buyer name:", "Enter Form");
        DataRow rowBuyer = this.dsWarehouse.Buyers.NewBuyersRow();
        rowBuyer["Name"] = nameBuyer;

        dsWarehouse.Buyers.Rows.Add(rowBuyer);

        int pos = this.dsWarehouse.Buyers.Rows.Count + 1;
        this.BindingContext[dsWarehouse, "Buyers"].Position = pos;

        DisplayForm(false);
        listBoxBuyers.Enabled = true;
        textBoxNameBuyer.Focus();
    }

    static string ShowPromptDialog(string message, string title)
    {
        Form prompt = new Form();
        prompt.Width = 300;
        prompt.Height = 150;
        prompt.Text = title;
    }
}
```

Продолжение Приложения И

```
Label messageLabel = new Label() { Left = 50, Top = 20, Text = message
};
TextBox textBox = new TextBox() { Left = 50, Top = 50, Width = 200 };
Button okButton = new Button() { Text = "OK", Left = 50, Top = 80 };
Button cancelButton = new Button() { Text = "Cancel", Left = 150, Top
= 80 };
prompt.Controls.Add(messageLabel);
prompt.Controls.Add(textBox);
prompt.Controls.Add(okButton);
prompt.Controls.Add(cancelButton);
prompt.ShowDialog();
okButton.Click += okButton_Click;
return textBox.Text;
}

private static void okButton_Click(object? sender, EventArgs e)
{
    throw new NotImplementedException();
}

private void editToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Edit");
}

private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Save");
    DataSetWarehouse.BuyersDataTable ds5 =
(DataSetWarehouse.BuyersDataTable)dsWarehouse.Buyers.GetChanges(DataRowState.Added
);
    if (ds5 != null)
        try
        {
            daBuyers.Update(ds5);
            dsWarehouse.Buyers.AcceptChanges();
        }
        catch (Exception x)
        {
            string mes = x.Message;
            MessageBox.Show("Error of adding string in database Warehouse
" + mes, "Warning");
            this.dsWarehouse.Buyers.RejectChanges();
        }
}

private void deleteToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Remove");
    int pos = -1;
    pos = this.BindingContext[dsWarehouse, "Buyers"].Position;
    string mes = textBoxNameBuyer.Text.ToString().Trim();
    MessageBox.Show("Delete data \n of buyer \n" + mes + "?", "Warning",
MessageBoxButtons.YesNo, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);
    this.dsWarehouse.Buyers.Rows[pos].Delete();
    if (this.dsWarehouse.Buyers.GetChanges(DataRowState.Deleted) != null)
    {
        try
        {
```

Продолжение Приложения И

```
        this.daBuyers.Update(dsWarehouse.Buyers);
        this.dsWarehouse.Buyers.AcceptChanges();
    }
    catch (Exception x)
    {
        string er = x.Message.ToString();
        MessageBox.Show("Error of deleting string in Warehouse database
" + er, "Warning");
        this.dsWarehouse.Buyers.RejectChanges();
    }
}
else { this.dsWarehouse.Buyers.RejectChanges(); }
}
}
}
```

Код интегрированного отчета по покупателям складского учета ООО «АВЭЛЛА»

```
using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ReportsAVELLA
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();
        }

        private void Form4_Load(object sender, EventArgs e)
        {
            this.reportViewer1.RefreshReport();
        }

        SqlConnection connection = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\VisualStudio\source\repos\AVELLA
Warehouse\AVELLAWarehouse\DatabaseAVELLA.mdf;Integrated Security=True");

        private void button1_Click(object sender, EventArgs e)
        {
            SqlCommand command = new SqlCommand("SELECT * FROM Buyers", connection);
            SqlDataAdapter d = new SqlDataAdapter(command);
            DataTable dt = new DataTable();
            d.Fill(dt);
            reportViewer1.LocalReport.DataSources.Clear();
        }
    }
}
```

Продолжение Приложения И

```
ReportDataSource source = new ReportDataSource("DataSet1", dt);
reportViewer1.LocalReport.ReportPath
@"D:\VisualStudio\source\repos\ReportsAVELLA\ReportsAVELLA\Report4.rdlc";
reportViewer1.LocalReport.DataSources.Add(source);
reportViewer1.RefreshReport();
}
}
}
```

=

Приложение К

Код приложения, отвечающий за ввод информации об отпускаемых продуктах складского учета ООО «АВЭЛЛА»

```
namespace AVELLAapp
{
    partial class Form6
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            button1 = new Button();
            label1 = new Label();
            textBox1 = new TextBox();
            label2 = new Label();
            maskedTextBox1 = new MaskedTextBox();
            label3 = new Label();
            maskedTextBox2 = new MaskedTextBox();
            SuspendLayout();
            //
            // button1
            //
            button1.Location = new Point(673, 383);
            button1.Name = "button1";
            button1.Size = new Size(77, 27);
            button1.TabIndex = 0;
            button1.Text = "Заккрыть";
            button1.UseVisualStyleBackColor = true;
            //
            // label1
            //
            label1.AutoSize = true;
            label1.Location = new Point(55, 37);
            label1.Name = "label1";
            label1.Size = new Size(143, 15);
            label1.TabIndex = 1;
        }
    }
}
```

Продолжение Приложения К

```
label1.Text = "Наименование продукта";
//
// textBox1
//
textBox1.Location = new Point(55, 85);
textBox1.Name = "textBox1";
textBox1.Size = new Size(197, 23);
textBox1.TabIndex = 2;
textBox1.Text = "Введите наименование продукта";
//
// label2
//
label2.AutoSize = true;
label2.Location = new Point(55, 143);
label2.Name = "label2";
label2.Size = new Size(72, 15);
label2.TabIndex = 3;
label2.Text = "Количество";
//
// maskedTextBox1
//
maskedTextBox1.Location = new Point(55, 191);
maskedTextBox1.Mask = "00000";
maskedTextBox1.Name = "maskedTextBox1";
maskedTextBox1.Size = new Size(100, 23);
maskedTextBox1.TabIndex = 4;
maskedTextBox1.ValidatingType = typeof(int);
//
// label3
//
label3.AutoSize = true;
label3.Location = new Point(55, 262);
label3.Name = "label3";
label3.Size = new Size(35, 15);
label3.TabIndex = 5;
label3.Text = "Цена";
//
// maskedTextBox2
//
maskedTextBox2.Location = new Point(55, 322);
maskedTextBox2.Name = "maskedTextBox2";
maskedTextBox2.Size = new Size(100, 23);
maskedTextBox2.TabIndex = 6;
//
// Form6
AutoScaleDimensions = new.SizeF(7F, 15F);
AutoScaleMode = AutoScaleMode.Font;
ClientSize = new Size(800, 450);
Controls.Add(maskedTextBox2);
Controls.Add(label3);
Controls.Add(maskedTextBox1);
Controls.Add(label2);
Controls.Add(textBox1);
Controls.Add(label1);
Controls.Add(button1);
Name = "Form6";
Text = "Ввод отпускаемых продуктов";
ResumeLayout(false);
PerformLayout();
}
```

Продолжение Приложения К

```
#endregion

private Button button1;
private Label label1;
private TextBox textBox1;
private Label label2;
private MaskedTextBox maskedTextBox1;
private Label label3;
private MaskedTextBox maskedTextBox2;
}
}
```

Код новой формы приложения, отвечающий за учет отпускаемой продукции складского учета ООО «АВЭЛЛА»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Security.Cryptography.X509Certificates;
using System.Data.Common;

namespace AVELLAWarehouse
{
    public partial class FormOutcomeProduct : Form
    {
        public FormOutcomeProduct()
        {
            InitializeComponent();
            DisplayReadOnly(true);
            WarehouseFill();
            FormOutcomeProduct_Load();
        }

        public void DisplayReadOnly(bool ReadOnly)
        {
            this.textBoxNameOutcomeProduct.ReadOnly = ReadOnly;
            this.textBoxIdOutcomeProduct.ReadOnly = ReadOnly;
            this.textBoxQuantityOutcomeProduct.ReadOnly = ReadOnly;
            this.textBoxPriceOutcomeProduct.ReadOnly = ReadOnly;
        }

        public void MenuItemEnabled(bool mode)
        {
            editToolStripMenuItem.Enabled = mode;
            deleteToolStripMenuItem.Enabled = mode;
            createToolStripMenuItem.Enabled = mode;
        }
    }
}
```

Продолжение Приложения К

```
        saveToolStripMenuItem.Enabled = !mode;
        undoToolStripMenuItem.Enabled = !mode;
    }

    private void DisplayForm(bool mode)
    {
        DisplayReadOnly(mode);
        MenuItemEnabled(mode);
    }

    DataSetWarehouse dsWarehouse = new DataSetWarehouse();
    DataSetWarehouseTableAdapters.OutcomeProductTableAdapter
daOutcomeProduct = new
AVELLAWarehouse.DataSetWarehouseTableAdapters.OutcomeProductTableAdapter();

    public void WarehouseFill()
    {
        daOutcomeProduct.Fill(dsWarehouse.OutcomeProduct);
        MessageBox.Show("Method Fill worked out");
    }

    private void FormOutcomeProduct_Load()
    {
        listBoxOutcomeProduct.Enabled = true;
        this.listBoxOutcomeProduct.DataSource =
this.dsWarehouse.Tables["OutcomeProduct"];
        this.listBoxOutcomeProduct.DisplayMember = "Name";

        textBoxNameOutcomeProduct.DataBindings.Add("Text", dsWarehouse,
"OutcomeProduct.Name");
        textBoxIdOutcomeProduct.DataBindings.Add("Text", dsWarehouse,
"OutcomeProduct.Id");
        textBoxQuantityOutcomeProduct.DataBindings.Add("Text",
dsWarehouse, "OutcomeProduct.Quantity");
        textBoxPriceOutcomeProduct.DataBindings.Add("Text", dsWarehouse,
"OutcomeProduct.Price");
    }

    private void undoToolStripMenuItem_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Method Undo");
    }

    private void createToolStripMenuItem_Click(object sender, EventArgs
e)
    {
        MessageBox.Show("Method Create");
        string nameOutcomeProduct = ShowPromptDialog("Enter Outcome
Product name:", "Enter Form");
        DataRow rowOutcomeProduct =
this.dsWarehouse.OutcomeProduct.NewOutcomeProductRow();
        rowOutcomeProduct["Name"] = nameOutcomeProduct;

        dsWarehouse.OutcomeProduct.Rows.Add(rowOutcomeProduct);

        int pos = this.dsWarehouse.OutcomeProduct.Rows.Count + 1;
        this.BindingContext[dsWarehouse, "OutcomeProduct"].Position =
pos;

        DisplayForm(false);
    }
}
```

Продолжение Приложения К

```
listBoxOutcomeProduct.Enabled = true;
textBoxNameOutcomeProduct.Focus();
}
static string ShowPromptDialog(string message, string title)
{
    Form prompt = new Form();
    prompt.Width = 300;
    prompt.Height = 150;
    prompt.Text = title;
    Label messageLabel = new Label() { Left = 50, Top = 20, Text = message
};
    TextBox textBox = new TextBox() { Left = 50, Top = 50, Width = 200 };
    Button okButton = new Button() { Text = "OK", Left = 50, Top = 80 };
    Button cancelButton = new Button() { Text = "Cancel", Left = 150, Top
= 80 };
    prompt.Controls.Add(messageLabel);
    prompt.Controls.Add(textBox);
    prompt.Controls.Add(okButton);
    prompt.Controls.Add(cancelButton);
    prompt.ShowDialog();
    okButton.Click += okButton_Click;
    return textBox.Text;
}

private static void okButton_Click(object? sender, EventArgs e)
{
    throw new NotImplementedException();
}

private void editToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Edit");
}

private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Method Save");
    DataSetWarehouse.OutcomeProductDataTable ds5 =
(DataSetWarehouse.OutcomeProductDataTable)dsWarehouse.OutcomeProduct.GetChanges(DataRowState.Added);
    if (ds5 != null)
        try
        {
            daOutcomeProduct.Update(ds5);
            dsWarehouse.OutcomeProduct.AcceptChanges();
        }
        catch (Exception x)
        {
            string mes = x.Message;
            MessageBox.Show("Error of adding string in database Warehouse
" + mes, "Warning");
            this.dsWarehouse.OutcomeProduct.RejectChanges();
        }
}

private void deleteToolStripMenuItem_Click(object sender, EventArgs e)
{

```

Продолжение Приложения К

```
MessageBox.Show("Method Remove");
int pos = -1;
pos = this.BindingContext[dsWarehouse, "OutcomeProduct"].Position;
string mes = textBoxNameOutcomeProduct.Text.ToString().Trim();

        MessageBox.Show("Delete data \n of outcome product \n" + mes + "?",
"Warning",           MessageBoxButtons.YesNo,           MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2);
        this.dsWarehouse.OutcomeProduct.Rows[pos].Delete();
        if (this.dsWarehouse.OutcomeProduct.GetChanges(DataRowState.Deleted)
!= null)
        {
            try
            {
                this.daOutcomeProduct.Update(dsWarehouse.OutcomeProduct);
                this.dsWarehouse.OutcomeProduct.AcceptChanges();
            }
            catch (Exception x)
            {
                string er = x.Message.ToString();
                MessageBox.Show("Error of deleting string in Warehouse database
" + er, "Warning");
                this.dsWarehouse.OutcomeProduct.RejectChanges();
            }
        }
        else { this.dsWarehouse.OutcomeProduct.RejectChanges(); }
    }
}
}
```

Код интегрированного отчета по отпускаемой продукции складского учета ООО «АВЭЛЛА»

```
using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ReportsAVELLA
{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();
        }

        private void Form5_Load(object sender, EventArgs e)
        {

```

Продолжение Приложения К

```
        this.reportViewer1.RefreshReport();
    }

    SqlConnection connection = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\VisualStudio\source\repos\AVELLA
Warehouse\AVELLAWarehouse\DatabaseAVELLA.mdf;Integrated Security=True");

    private void button1_Click(object sender, EventArgs e)
    {
        SqlCommand command = new SqlCommand("SELECT * FROM
OutcomeProduct", connection);
        SqlDataAdapter d = new SqlDataAdapter(command);
        DataTable dt = new DataTable();
        d.Fill(dt);
        reportViewer1.LocalReport.DataSources.Clear();
        ReportDataSource source = new ReportDataSource("DataSet1", dt);
        reportViewer1.LocalReport.ReportPath =
@"D:\VisualStudio\source\repos\ReportsAVELLA\ReportsAVELLA\Report5.rdlc";
        reportViewer1.LocalReport.DataSources.Add(source);
        reportViewer1.RefreshReport();
    }
}
```

Приложение Л

Код приложения, отвечающий за вывод информации о поставщиках, сырье, покупателях и продукции складского учета ООО «АВЭЛЛА»

```
namespace AVELLAapp
{
    partial class Form7
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            button1 = new Button();
            label1 = new Label();
            textBox1 = new TextBox();
            label2 = new Label();
            textBox2 = new TextBox();
            label3 = new Label();
            textBox3 = new TextBox();
            label4 = new Label();
            textBox4 = new TextBox();
            SuspendLayout();
            //
            // button1
            //
            button1.Location = new Point(671, 383);
            button1.Name = "button1";
            button1.Size = new Size(97, 31);
            button1.TabIndex = 0;
            button1.Text = "Закреть";
            button1.UseVisualStyleBackColor = true;
            //
            // label1
            //
            label1.AutoSize = true;
            label1.Location = new Point(21, 21);
            label1.Name = "label1";
        }
    }
}
```

Продолжение Приложения Л

```
label1.Size = new Size(77, 15);
label1.TabIndex = 1;
label1.Text = "Поставщики";
//
// textBox1
//
textBox1.Location = new Point(21, 61);
textBox1.Multiline = true;
textBox1.Name = "textBox1";
textBox1.Size = new Size(158, 100);
textBox1.TabIndex = 2;
//
// label2
//
label2.AutoSize = true;
label2.Location = new Point(21, 204);
label2.Name = "label2";
label2.Size = new Size(118, 15);
label2.TabIndex = 3;
label2.Text = "Сырье и материалы";
//
// textBox2
//
textBox2.Location = new Point(21, 259);
textBox2.Multiline = true;
textBox2.Name = "textBox2";
textBox2.Size = new Size(158, 106);
textBox2.TabIndex = 4;
//
// label3
//
label3.AutoSize = true;
label3.Location = new Point(332, 21);
label3.Name = "label3";
label3.Size = new Size(73, 15);
label3.TabIndex = 5;
label3.Text = "Покупатели";
//
// textBox3
//
textBox3.Location = new Point(332, 61);
textBox3.Multiline = true;
textBox3.Name = "textBox3";
textBox3.Size = new Size(159, 100);
textBox3.TabIndex = 6;
//
// label4
//
label4.AutoSize = true;
label4.Location = new Point(332, 204);
label4.Name = "label4";
label4.Size = new Size(68, 15);
label4.TabIndex = 7;
label4.Text = "Продукция";
//
// textBox4
textBox4.Location = new Point(332, 259);
textBox4.Multiline = true;
textBox4.Name = "textBox4";
textBox4.Size = new Size(159, 106);
```

Продолжение Приложения Л

```
        textBox4.TabIndex = 8;
        //
        // Form7
        //
        AutoScaleDimensions = new SizeF(7F, 15F);
        AutoScaleMode = AutoScaleMode.Font;
        ClientSize = new Size(800, 450);
        Controls.Add(textBox4);
        Controls.Add(label4);
        Controls.Add(textBox3);
        Controls.Add(label3);
        Controls.Add(textBox2);
        Controls.Add(label2);
        Controls.Add(textBox1);
        Controls.Add(label1);
        Controls.Add(button1);
        Name = "Form7";
        Text = "Оборотная ведомость";
        ResumeLayout(false);
        PerformLayout();
    }

#endregion

private Button button1;
private Label label1;
private TextBox textBox1;
private Label label2;
private TextBox textBox2;
private Label label3;
private TextBox textBox3;
private Label label4;
private TextBox textBox4;
}
}
```

Приложение М

Код интегрированного отчета по подразделениям складского учета ООО «АВЭЛЛА»

```
using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ReportsAVELLA
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            this.reportViewer1.RefreshReport();

            SqlConnection connection = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\VisualStudio\source\repos\AVELLA
Warehouse\AVELLAWarehouse\DatabaseAVELLA.mdf;Integrated Security=True");

            private void button1_Click(object sender, EventArgs e)
            {
                SqlCommand command = new SqlCommand("SELECT * FROM Departments",
connection);

                SqlDataAdapter d = new SqlDataAdapter(command);
                DataTable dt = new DataTable();
                d.Fill(dt);
                reportViewer1.LocalReport.DataSources.Clear();
                ReportDataSource source = new ReportDataSource("DataSet1", dt);
                reportViewer1.LocalReport.ReportPath =
@"D:\VisualStudio\source\repos\ReportsAVELLA\ReportsAVELLA\Report1.rdlc";
                reportViewer1.LocalReport.DataSources.Add(source);
                reportViewer1.RefreshReport();
            }
        }
    }
}
```