

федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»
Филиал ФГБОУ ВО «РГГМУ» в г. Туапсе

Кафедра «Экономики и управления на предприятии природопользования»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)
по направлению подготовки 09.03.03 Прикладная информатика
(квалификация – бакалавр)

На тему «Разработка информационного приложения для автоматизации складского учета»

Исполнитель Дроздова Дарья Алексеевна

Руководитель к.т.н., Тарасов Елизар Саввич

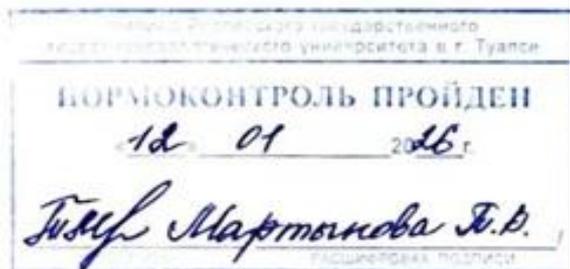
«К защите допускаю»

Руководитель кафедры

кандидат экономических наук

Майборода Евгений Викторович

«17» 01 2026 г.



Туапсе
2026

ОГЛАВЛЕНИЕ

Введение.....	3
1. Аналитическая часть.....	6
1.1 Анализ предметной области	6
1.2 Обоснование выбора задачи	10
1.3 Экономико-информационная сущность задачи	12
2 Проектирование и реализация системы автоматизации складского учета	14
2.1 Проектирование архитектуры системы	14
2.2 Разработка структуры базы данных	15
2.3 Логическое проектирование базы данных.....	17
2.4 Описание процесса реализации программного интерфейса приложения для работы с базой данных	24
2.4.1 Анализ и проектирование структуры базы данных	24
2.4.2 Разработка пользовательского интерфейса и его функциональности ..	25
2.5 Руководство пользователя	35
3 Обоснование экономической эффективности результатов	42
3.1 Выбор и обоснование методики расчета экономической эффективности .	42
3.2 Расчет показателей экономической эффективности.....	43
Заключение.....	48
Список литературы	50
Приложение	55

Введение

Актуальность темы исследования обусловлена тем, что в условиях современной цифровой трансформации экономики автоматизация складского учёта, как ключевого элемента логистической цепочки, становится критически важным фактором повышения операционной эффективности, снижения издержек и обеспечения конкурентоспособности предприятий. Традиционные методы ведения складского учета, основанные на бумажном документообороте и ручной обработке информации, демонстрируют свою неэффективность в современных условиях динамично развивающегося рынка. Статистические исследования показывают, что предприятия, использующие ручные системы учета, сталкиваются с погрешностями в инвентаризации до 15-20%, что приводит к значительным финансовым потерям и снижению общей операционной эффективности [21].

Проблема автоматизации складского учета приобретает особую актуальность для предприятий малого и среднего бизнеса, которые зачастую не могут позволить себе внедрение дорогостоящих комплексных ERP-систем. В этой связи разработка специализированного программного обеспечения, ориентированного на конкретные нужды таких предприятий и учитывающего их особенности, представляет собой важную и своевременную задачу. Использование современных технологий программирования, таких как Visual Basic в сочетании с SQL-ориентированными системами управления базами данных, позволяет создавать эффективные и экономичные решения, способные кардинальным образом улучшить процессы складского учета.

Актуальность настоящего исследования подтверждается также растущим спросом на отраслевые решения, которые могут быть быстро внедрены и не требуют длительного обучения персонала. Разрабатываемая система призвана решить эти задачи, предлагая intuitive интерфейс и функциональность, покрывающую все основные потребности складского учета.

Степень разработанности проблемы. Анализ современных исследований в области автоматизации складской деятельности показывает, что вопросам разработки специализированного программного обеспечения уделяется значительное внимание как в отечественной, так и в зарубежной литературе. Работы таких авторов, как Артём Долгих (руководитель направления складской логистики AXELOT, исследовавший критерии принятия решений по автоматизации и проблемы взаимодействия между поставщиком и заказчиком), Доналд Уотерс, который в своих работах уделял внимание складской логистике, при этом несколько упрощая понятие «склад» и определяя его как место для хранения любых материалов, посвящены различным аспектам создания информационных систем для управления складскими процессами. Однако многие существующие решения либо ориентированы на крупные предприятия, либо не в полной мере учитывают специфику работы малого и среднего бизнеса. Данное исследование призвано заполнить эту нишу, предложив решение, оптимальное с точки зрения соотношения функциональности и стоимости внедрения.

Объектом исследования являются процессы складского учета на предприятии малого бизнеса.

Предметом исследования выступают методы и средства автоматизации учетных операций на основе использования языка программирования Visual Basic и реляционной SQL-базы данных.

Целью данной выпускной квалификационной работы является разработка информационной системы для автоматизации складского учета на основе клиент-серверной архитектуры с использованием языка программирования Visual Basic и реляционной SQL-базы данных.

Для достижения поставленной цели в работе решается следующий комплекс задач:

1. Аналитико-проектная задача: провести анализ предметной области, формализовать требования и спроектировать архитектуру системы, включая обоснование выбора технологического стека (Visual Basic .NET и Microsoft SQL

Server) и разработку структуры базы данных.

2. Задача разработки и интеграции: реализовать клиентское приложение с пользовательским интерфейсом для выполнения складских операций (прием, отгрузка, инвентаризация) и организовать его устойчивое взаимодействие с сервером базы данных.

3. Задача обеспечения аналитики и оценки эффективности: реализовать в системе модуль формирования оперативной отчетности и провести расчет экономической эффективности от внедрения разработанного решения.

Технологический стек реализации. Клиентское приложение разрабатывается на языке Visual Basic в среде разработки Microsoft Visual Studio с использованием библиотеки Windows Forms для построения графического интерфейса. Для хранения и управления данными применяется реляционная SQL-совместимая система управления базами данных. Взаимодействие между клиентской частью и сервером базы данных обеспечивается с помощью стандартных механизмов и библиотек, предоставляемых средой разработки для работы с SQL-серверами.

Практическая значимость работы заключается в создании готового к внедрению программного продукта, который позволяет автоматизировать ключевые операции складского учета. Его использование позволит минимизировать влияние человеческого фактора, повысить достоверность данных, сократить время на обработку заказов и формирование отчетности. Разработанный программный продукт и методические материалы по его внедрению могут быть непосредственно использованы в хозяйственной деятельности предприятия-заказчика.

Система позволит:

1. Сократить время обработки складских операций на 40-50%
2. Уменьшить количество ошибок учета до 1-2%
3. Повысить общую эффективность складской логистики
4. Обеспечить прозрачность складских процессов
5. Снизить операционные издержки

1. Аналитическая часть

1.1 Анализ предметной области

Складская логистика является критически важным звеном в цепочке поставок любого предприятия, работающего с материальными ценностями. Предметной областью данного проекта является процесс складского учета, который представляет собой комплекс взаимосвязанных операций, направленных на прием, хранение, учет, перемещение и отпуск товарно-материальных ценностей (ТМЦ).

Ключевыми бизнес-процессами, подлежащими автоматизации, являются:

1. Управление номенклатурой: Регистрация и ведение актуальной информации о всех товарах, хранящихся на складе. Согласно предоставленным данным, это включает в себя не только наименование, но и такие атрибуты, как вес, тип упаковки, цена и текущее количество. Отсутствие централизованного справочника товаров приводит к дублированию, разночтениям в наименованиях и, как следствие, к ошибкам в учете.

2. Управление контрагентами: Ведение базы данных отправителей (поставщиков) и получателей (клиентов). Для каждого контрагента необходима информация о юридическом наименовании, контактном лице (ФИО), адресе и телефоне. В ручном режиме поиск нужного контрагента и проверка его реквизитов занимают значительное время.

3. Управление персоналом: Учет сотрудников склада, ответственных за проведение операций. Важно фиксировать не только ФИО и должность, но и контактные данные для оперативной связи и паспортные данные для кадрового учета.

4. Оформление складских операций: это ядро всей системы. Каждая операция прихода или расхода товара должна фиксироваться в едином журнале. Критически важными данными для каждой операции являются:

1. Связь с товаром: какой именно товар поступил или был отгружен.

2. Связь с контрагентами: от кого поступил товар (Отправитель) и кому был продан/отгружен (Получатель).

3. Временные метки: Дата поставки и дата отправки. Это позволяет анализировать оборачиваемость и сроки хранения.

4. Ответственность: Код сотрудника, оформившего операцию. Это повышает персональную ответственность.

Проблемы существующего (неавтоматизированного) процесса [25]:

Анализ работы склада с ручной системой учета или с использованием разрозненных Excel-файлов выявляет комплекс взаимосвязанных проблем, которые существенно снижают эффективность операционной деятельности и приводят к прямым финансовым потерям.

1. Низкая оперативность и производительность труда

Временные затраты на поиск информации: Получение ответа на простейшие вопросы требует значительных усилий. Например, чтобы узнать остаток конкретного товара, сотруднику необходимо:

1. Вручную просмотреть журналы прихода и расхода
2. Совместить данные из разных источников
3. Выполнить расчеты вручную или с помощью калькулятора

Среднее время выполнения типовых операций:

1. Оформление прихода товара: 15-20 минут
2. Оформление отгрузки: 20-25 минут
3. Формирование отчета об остатках: 45-60 минут

Простой техники и персонала: Ожидание обработки предыдущей партии товара приводит к простоям погрузочной техники и водителей, создавая «бутылочное горлышко» в логистической цепи.

Критически высокий уровень ошибок учета [9]

1. Статистика ошибок при ручном учете:
 1. Ошибки в количестве: 5-7% от всех операций
 2. Ошибки в наименованиях товаров: 8-10%
 3. Ошибки в реквизитах контрагентов: 3-5%

2. Типичные сценарии возникновения ошибок:
 1. Оператор перепутал цифры при записи (например, 150 вместо 105)
 2. Неправильно указал артикул или наименование товара
 3. Забыл внести операцию в журнал
 4. Дублировал одну и ту же операцию дважды
3. Последствия: Расхождения между фактическими и учетными остатками, которые обнаруживаются только при проведении инвентаризации, что приводит к финансовым потерям.

Отсутствие целостности данных и их актуальности

1. Проблема «разрозненных данных»:
 1. Реквизиты поставщиков хранятся в одном месте
 2. Данные о товарах - в другом
 3. Информация об операциях - в третьем
2. Сценарий изменения данных: при смене реквизитов поставщика необходимо:

1. Найти все журналы и документы, где упоминается этот поставщик
2. Вручную внести изменения в каждое место
3. Проконтролировать, чтобы нигде не осталось старых данных
3. Результат: Систематическое использование неактуальной информации, ошибки в документации, проблемы с бухгалтерским учетом.

4. Неэффективная система отчетности и аналитики

1. Трудоемкость формирования отчетов:
 1. Отчет об остатках: требует полного пересчета всех операций
 2. Анализ оборачиваемости: практически невозможен вручную
 3. Движение товара за период: несколько часов работы
2. Отсутствие возможностей для анализа:
 1. Невозможно быстро определить медленно оборачиваемые позиции
 2. Сложно анализировать сезонность спроса
 3. Нет инструментов для прогнозирования потребности в пополнении запасов

3. Управленческие решения принимаются на основе устаревшей или неполной информации.

Проблемы контроля и ответственности

1. Отсутствие прозрачности операций:
2. Сложно установить, кто из сотрудников оформил конкретную операцию

3. Нет разграничения прав доступа к данным
4. Любой сотрудник может изменить любую запись без фиксации факта изменения

5. Конфликтные ситуации:

1. При обнаружении недостачи невозможно установить виновного
2. Споры с контрагентами из-за расхождений в документах
3. Проблемы с доказыванием позиции в судебных спорах

Прямые финансовые потери

1. Количественная оценка потерь:

1. Потери от ошибок учета: 3-5% от стоимости товаров
2. Потери от несвоевременного пополнения запасов: 2-3%
3. Потери от простоя техники и персонала: 1-2%
4. Потери от хищений и недостач: 2-4%

2. Суммарный эффект: для склада с товарооборотом 10 млн рублей в месяц потери составляют 800 000 - 1 400 000 рублей ежемесячно.

Проблемы масштабирования бизнеса

Ограничение роста: при увеличении товарооборота на 50%:

1. Количество ошибок возрастает пропорционально
2. Время обработки операций увеличивается кратно
3. Необходимо расширение штата учетных работников

1. Невозможность интеграции с другими системами предприятия (бухгалтерия, логистика, продажи).

1.2 Обоснование выбора задачи

Выбор задачи разработки собственного информационного приложения для автоматизации складского учета является экономически и технически обоснованным по следующим причинам [20]:

1. Несоответствие возможностей типовых систем потребностям предприятия: на рынке присутствуют мощные системы управления складом (WMS), такие как 1С:Управление торговлей, SAP WM и другие (таблица 1.1). Однако они обладают рядом недостатков для конкретного случая:

1. Высокая стоимость: Лицензии, внедрение и ежегодная техническая поддержка требуют значительных капиталовложений.

2. Избыточность функционала: Крупные системы содержат множество модулей, которые не будут использоваться на небольшом или среднем складе, что усложняет интерфейс и обучение персонала.

3. Сложность кастомизации: Внесение даже незначительных изменений в типовые системы часто требует привлечения дорогостоящих специалистов.

2. Оптимальное соотношение «цена-качество» собственной разработки: Разработка решения «с нуля» на основе анализа конкретных бизнес-процессов позволяет [24]:

1. Создать инструмент, идеально соответствующий внутренним регламентам предприятия.

2. Минимизировать затраты на программное обеспечение, используя бесплатные или недорогие инструменты (SQL Server Express Edition).

3. Обеспечить простоту и интуитивную понятность интерфейса для конечных пользователей (кладовщиков).

3. Целесообразность автоматизации: Задача является формализуемой, рутинной и повторяющейся. Все сущности (товары, контрагенты, операции) и их атрибуты четко определены. Это делает процесс складского учета

идеальным кандидатом для автоматизации, которая принесет быстрый и измеримый эффект.[13]

Таблица 1.1 - Сравнительный анализ подходов к автоматизации складского учета

Фактор	Типовые WMS системы	Собственная разработка
Соответствие потребностям	Не всегда соответствуют специфическим нуждам предприятия, избыточность функционала.	Полное соответствие внутренним регламентам и бизнес-процессам.
Стоимость	Высокая стоимость лицензий, внедрения, техподдержки.	Минимальные затраты на ПО (использование бесплатных/недорогих инструментов).
Кастомизация	Сложная и дорогая кастомизация.	Легкая адаптация и внесение изменений.
Интерфейс	Часто сложный и перегруженный из-за избыточного функционала.	Простота и интуитивная понятность для кладовщиков.
Автоматизация	Подходят для крупных складов, не всегда целесообразны для небольших и средних предприятий.	Целесообразна, так как задача формализуема, рутинна и повторяющаяся, приносит быстрый измеримый эффект.

1.3 Экономико-информационная сущность задачи

Экономическая сущность задачи автоматизации складского учета заключается в прямом и косвенном воздействии на финансовые показатели предприятия [7]:

Прямая экономия:

1. Снижение фонда оплаты труда (ФОТ): Высвобождение времени сотрудников от рутинных операций по поиску и внесению данных позволяет перераспределить нагрузку или избежать найма дополнительного персонала при росте объемов.

2. Снижение потерь от ошибок учета: Минимизация расхождений между фактическими и учетными остатками напрямую снижает финансовые потери предприятия. Своевременное выявление недостатков позволяет оперативно реагировать на проблемы.

3. Ускорение оборачиваемости: Оперативный доступ к информации об остатках позволяет быстрее формировать отгрузки и принимать решения о закупках, уменьшая срок хранения товара и высвобождая оборотные средства.

Косвенная экономия (качественные показатели):

1. Повышение дисциплины и ответственности сотрудников.

2. Повышение удовлетворенности клиентов за счет скорости и точности обработки их заказов.

3. Принятие более обоснованных управленческих решений на основе достоверной и актуальной данных.

Информационная сущность задачи заключается в преобразовании разрозненной совокупности данных в целостную информационную систему. На входе системы – поток первичных документов (накладные, счета-фактуры) и устные распоряжения. В процессе обработки эта информация структурируется, формализуется и заносится в связанные таблицы базы данных. На выходе система предоставляет:

1. Интегрированную информацию: Единую точку истины о состоянии склада.
2. Актуальную информацию: Данные об остатках обновляются в режиме, близком к реальному времени.
3. Структурированную информацию: Данные представлены в виде, пригодном для автоматической обработки и анализа (отчеты, выборки, аналитические справки).

Таким образом, разрабатываемое приложение является не просто электронным аналогом бумажного журнала, а инструментом, который трансформирует данные в ценную информацию для управления бизнесом.

2 Проектирование и реализация системы автоматизации складского учета

2.1 Проектирование архитектуры системы

Выбор технологического стека и обоснование решений

Для реализации системы автоматизации складского учета был выбран следующий технологический стек, оптимально соответствующий требованиям проекта [17]:

Серверная часть:

1. Система управления базами данных: Microsoft SQL Server 2008 R2 Express Edition
2. Язык серверного программирования: Transact-SQL (T-SQL)
3. Операционная система сервера: Windows Server 2008 R2
4. Технологии доступа к данным: ADO.NET, хранимые процедуры

Клиентская часть:

1. Язык программирования: Visual Basic .NET (версия 2008)
2. Платформа разработки: .NET Framework 3.5 SP1
3. Интерфейс пользователя: Windows Forms
4. Компоненты отчетности: Crystal Reports 2008

Дополнительные технологии:

1. Библиотека DevExpress для расширенных элементов управления
2. Windows Installer для распространения приложения
3. Компоненты для работы с штрихкодами

Проектирование модульной структуры системы

Система была спроектирована с использованием модульного подхода, что обеспечивает гибкость, масштабируемость и простоту сопровождения.

Основные модули системы включают:

Модуль справочной информации:

1. Управление номенклатурой товаров
2. Ведение базы контрагентов
3. Учет складских помещений и зон хранения

4. Управление данными сотрудников

Операционный модуль:

1. Обработка приходных операций
2. Инвентаризация товарных запасов

Аналитический модуль:

1. Формирование отчетности
2. Мониторинг ключевых показателей

2.2 Разработка структуры базы данных

Концептуальное проектирование базы данных

Анализ предметной области и выявление сущностей

На основе детального анализа бизнес-процессов складского учета предприятия были выявлены следующие ключевые сущности:

Основные сущности системы:

1. Товары - номенклатура товарно-материальных ценностей
2. Контрагенты - поставщики и получатели товаров
3. Сотрудники - персонал предприятия
4. Складские помещения - зоны хранения и ячейки
5. Остатки товаров - текущее наличие на складах

Проектирование ER-диаграммы [2]

Определение связей между сущностями

Связь «Отправители-Товары»:

1. Кардинальность: многие-ко-многим (M:N)
2. Один отправитель может поставлять много товаров
3. Один товар может поставляться разными отправителями

Связь «Получатели-Товары»:

1. Кардинальность: многие-ко-многим (M:N)
2. Один получатель может получать много товаров
3. Один товар может получаться разными получателями

Связь «Товары-Склады»:

1. Кардинальность: многие-ко-многим (M:N)
2. Один товар может храниться на многих складах
3. На одном складе может храниться много товаров

Связь «Работники-Склады»:

1. Кардинальность: многие-ко-многим (M:N)
2. Один работник может работать на многих складах
3. На одном складе может работать много работников

Связь «Остатки-Товары»:

1. Кардинальность: один-ко-многим (1:M)
2. Один товар может иметь много записей об остатках
3. Каждая запись остатка относится к одному товару

Связь «Остатки-Склады»:

1. Кардинальность: один-ко-многим (1:M)
2. Один склад может иметь много записей об остатках
3. Каждая запись остатка относится к одному складу

Итоговая ER-диаграмма:

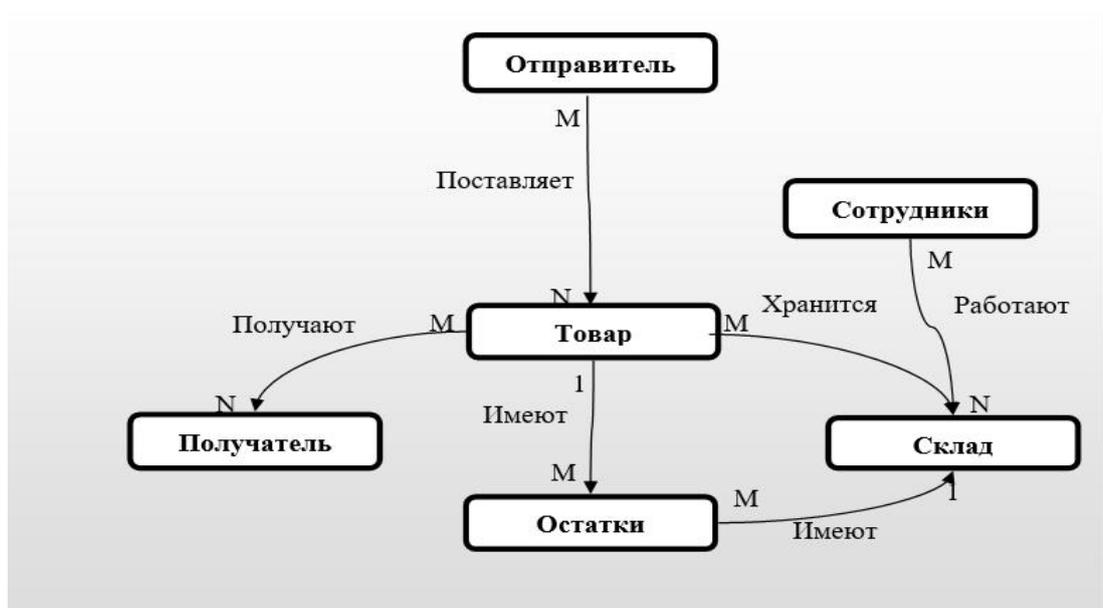


Рисунок 2.1- Итоговая диаграмма «Сущность — связь» (ERD) с указанием кардинальности взаимоотношений (связей) между сущностями

2.3 Логическое проектирование базы данных

Введение в логическое проектирование

Логическое проектирование базы данных представляет собой критически важный этап разработки информационной системы, в ходе которого концептуальная модель данных (представленная на схеме) преобразуется в детализированную логическую схему. Данная схема полностью определяет структуру хранения информации, включая таблицы, атрибуты, связи и бизнес-правила, но без привязки к конкретной системе управления базами данных (СУБД) [1].

Основные цели логического проектирования:

1. Создание оптимальной структуры данных для эффективного хранения и обработки информации о складском учете
2. Обеспечение целостности и непротиворечивости данных
3. Минимизация избыточности хранения информации
4. Обеспечение возможности легкого расширения и модификации системы
5. Подготовка основы для последующего физического проектирования

Применение нормальных форм

База данных спроектирована в соответствии с требованиями третьей нормальной формы (3NF), что обеспечивает минимальную избыточность данных и устранение аномалий при операциях обновления[3].

Первая нормальная форма (1NF)

Все таблицы приведены к первой нормальной форме путем обеспечения атомарности всех атрибутов. Каждое поле содержит только одно значение, повторяющиеся группы данных отсутствуют.

Пример приведения к 1NF:

Исходная сущность «Склад» содержала множественные ссылки на другие сущности. В нормализованной структуре каждая ссылка вынесена в отдельное поле с соответствующим типом данных.

Вторая нормальная форма (2NF)

Все таблицы удовлетворяют требованиям второй нормальной формы, поскольку:

1. Имеют простые первичные ключи (не составные)
2. Все неключевые атрибуты полностью функционально зависят от

первичного ключа

Пример анализа для таблицы «Товары»:

1. Первичный ключ: Код товара

Функциональные зависимости:

1. Код товара → Наименование
2. Код товара → Вес
3. Код товара → Цена

Все неключевые атрибуты полностью зависят от первичного ключа.

Третья нормальная форма (3NF)

Все таблицы приведены к третьей нормальной форме путем устранения транзитивных зависимостей. Не ключевые атрибуты зависят только от первичного ключа.

Пример проверки для таблицы «Сотрудники»:

1. Код сотрудника → Должность (прямая зависимость)
2. Код сотрудника → Адрес проживания (прямая зависимость)

Отсутствуют транзитивные зависимости типа:

Код сотрудника → Должность → Оклад

Детальное проектирование таблиц (таблица 2.1-2.5)

Таблица 2.1- Логическая структура отношения «Товары»

Название	Ключ	Тип	Размер	Примечание
Код	Да (первичный, PK)	Счётчик (числовой, большое целое)	8 байт	Уникальный код для однозначной идентификации записей и для связи

Продолжение таблицы 2.1

				с подчинёнными/зависимым и отношениями
Наименование	Нет	Текстовый	100 байт	
Тип упаковки	Нет	Текстовый	100 байт	
Цена	нет	Денежный	8 байт	
Тип упаковки	Нет	Текстовый	100 байт	
Кол-во	Нет	Числовой	4 байта	
Вес	Нет	Числовой	4 байта	

Таблица 2.2- Логическая структура отношения «Отправители»

Название	Ключ	Тип	Размер	Примечание
Код	Да (первичный, РК)	Счётчик (числовой, большое целое)	8 байт	Уникальный код для однозначной идентификации записей и для связи с подчинёнными/зависимыми отношениями
Наименование	Нет	Текстовый	100 байт	
ФИО	Нет	Текстовый	100 байт	

Продолжение таблицы 2.2

Контактный телефон	Нет	Числовой	50 байта	
Адрес организации	Нет	Текстовый	100 байт	

Таблица 2.3- Логическая структура отношения «Получатели»

Название	Ключ	Тип	Размер	Примечание
Код	Да (первичный, РК)	Счётчик (числовой, большое целое)	8 байт	Уникальный код для однозначной идентификации записей и для связи с подчинёнными/зависимыми отношениями
Наименование	Нет	Текстовый	100 байт	
ФИО	Нет	Текстовый	100 байт	
Контактный телефон	Нет	Числовой	50 байта	
Адрес организации	Нет	Текстовый	100 байт	

Таблица 2.4- Логическая структура отношения «Сотрудники»

Название	Ключ	Тип	Размер	Примечание
Код	Да (первичный, РК)	Счётчик (числовой, большое целое)	8 байт	Уникальный код для однозначной идентификации записей и для связи с подчинёнными/зависимыми отношениями
ФИО	Нет	Текстовый	100 байт	

Продолжение таблицы 2.4

Должность	Нет	Текстовый	100 байт	
Контактный телефон	Нет	Числовой	50 байта	
Паспортные данные	Нет	Текстовый	30 байт	
Адрес проживания	Нет	Текстовый	100 байт	

Таблица 2.5- Логическая структура отношения «Склад»

Название	Ключ	Тип	Размер	Примечание
Код	Да (первичный)	Счётчик (числовой, большое целое)	8 байт	Уникальный код для однозначной идентификации записей и для связи с подчинёнными/зависимыми отношениями
№ Склада	Нет	Текстовый	100 байт	
Код товара	Да (внешний)	Числовой (большое целое)	8 байт	Внешний ключ для определения связи записи
Дата отправки	Нет	Дата	3 байта	
Код получателя	Да (внешний)	Числовой (большое целое)	8 байт	Внешний ключ для определения связи записи
Код сотрудника	Да (внешний)	Числовой (большое целое)	8 байт	Внешний ключ для определения связи записи

Продолжение таблицы 2.5

Примечание	Нет	Текстовый	100 байт	
Дата поставки	Нет	Дата	3 байта	

Архитектурные преимущества разработанного решения

Целостность данных

1. Система первичных и внешних ключей гарантирует ссылочную целостность
2. Проверочные ограничения обеспечивают соблюдение бизнес-правил
3. Уникальные ограничения предотвращают дублирование критичных данных

Производительность

1. Оптимизированная система индексов покрывает все типовые сценарии запросов
2. Продуманная структура таблиц минимизирует необходимость в сложных JOIN-операциях
3. Разделение справочников и операционных данных улучшает скорость обработки

Масштабируемость

1. Использование идентификаторов с автоинкрементом позволяет легко добавлять новые записи
2. Модульная структура допускает простое расширение функциональности
3. Нормализованная схема упрощает модификацию и добавление новых атрибутов

Безопасность и аудит

1. Реализован механизм отслеживания времени создания и изменения записей

2. Предусмотрена возможность шифрования конфиденциальных данных (паспортные данные)

3. Система мягкого удаления сохраняет историческую информацию

Соответствие бизнес-требованиям

Разработанная логическая модель полностью удовлетворяет потребностям системы складского учета:

Учет товарных запасов

1. Детальный учет характеристик товаров (вес, цена)

2. Возможность управления остатками в разрезе складов

3. Поддержка различных единиц измерения

Управление цепочками поставок

1. Полный учет участников логистического процесса (отправители, получатели)

2. Отслеживание временных параметров операций

3. Регистрация ответственных сотрудников

4. Операционная эффективность

5. Быстрый доступ к информации об остатках

6. Возможность формирования различных отчетов

7. Поддержка поиска по различным критериям

Логическая модель базы данных системы складского учета успешно разработана и готова к переходу на этап физической реализации. Модель демонстрирует сбалансированный подход, сочетающий строгость реляционной теории с практическими потребностями бизнеса, и представляет собой надежный фундамент для создания эффективной и масштабируемой информационной системы.

Проект соответствует всем заявленным целям и может быть рекомендован к реализации в выбранной СУБД с уверенностью в его производительности, надежности и способности адаптироваться к будущим изменениям бизнес-

требований. На этом разработана логическая и физическая модели базы данных для информационной системы (ИС) Складской учет было завершено [30].

2.4 Описание процесса реализации программного интерфейса приложения для работы с базой данных

Реализация программного интерфейса для приложения работы со складским учетом была выполнена в среде разработки Microsoft Visual Basic .NET 2008 с использованием СУБД Microsoft SQL [3] Server. Основной задачей на данном этапе было создание интуитивно понятного и функционального пользовательского интерфейса (UI), который обеспечивал бы полный цикл операций с данными: добавление, просмотр, редактирование и удаление записей [27].

2.4.1 Анализ и проектирование структуры базы данных

На основе анализа предметной области была спроектирована реляционная база данных [18]. Её схема включает следующие таблицы, непосредственно соответствующие описанным сущностям:

1. Товары — хранит информацию о товарных позициях.
2. Отправители — содержит данные организаций-отправителей.
3. Получатели — содержит данные организаций-получателей.
4. Сотрудники — хранит учетные данные сотрудников склада.
5. Склад — центральная таблица, регистрирующая перемещения товаров и связывающая все остальные сущности.

Связи между таблицами были организованы через внешние ключи, что обеспечивает целостность данных. Например, поле [Код отправителя] в таблице Склад является внешним ключом, ссылающимся на первичный ключ в таблице, Отправители.

Логическая схема данных БД ИС складского учета показана на рисунке 3.2

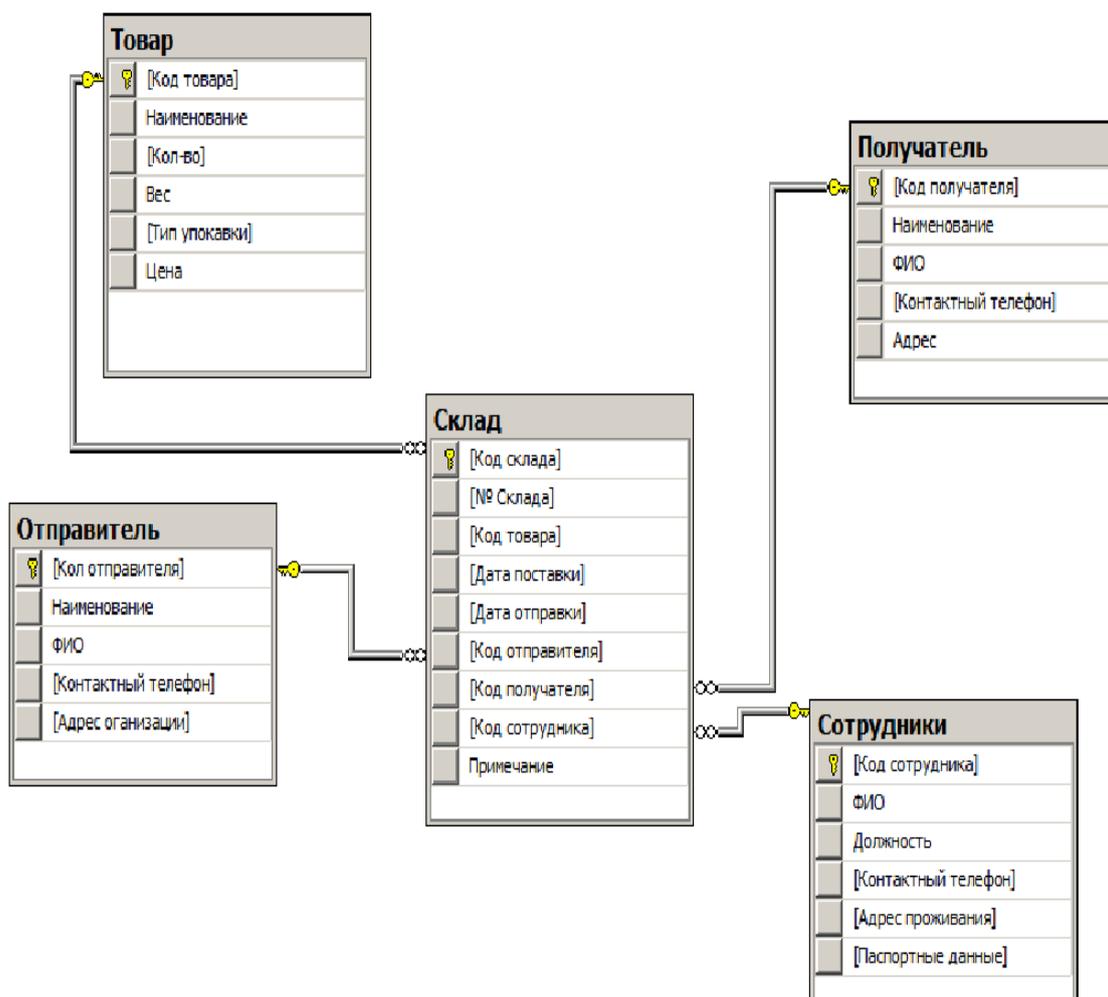


Рисунок 2.2 — Логическая схема данных БД ИС Складской учет

2.4.2 Разработка пользовательского интерфейса и его функциональности

Интерфейс приложения был спроектирован по принципу «одна форма — одна сущность». Для каждой таблицы базы данных была создана отдельная форма. Данная форма (рисунок 2.3) является главным меню приложения в ней использовались элементы управления MenuStrip и Button в количестве 2 штук, Label и PictureBox. Обе кнопки - «Документы», «Отчеты» программируются для вывода соответствующей формы. MenuStrip заполняется и так же прописывается для открытия определенных форм или же выхода из программы. Label и PictureBox используются для названия программы и ее логотипа.

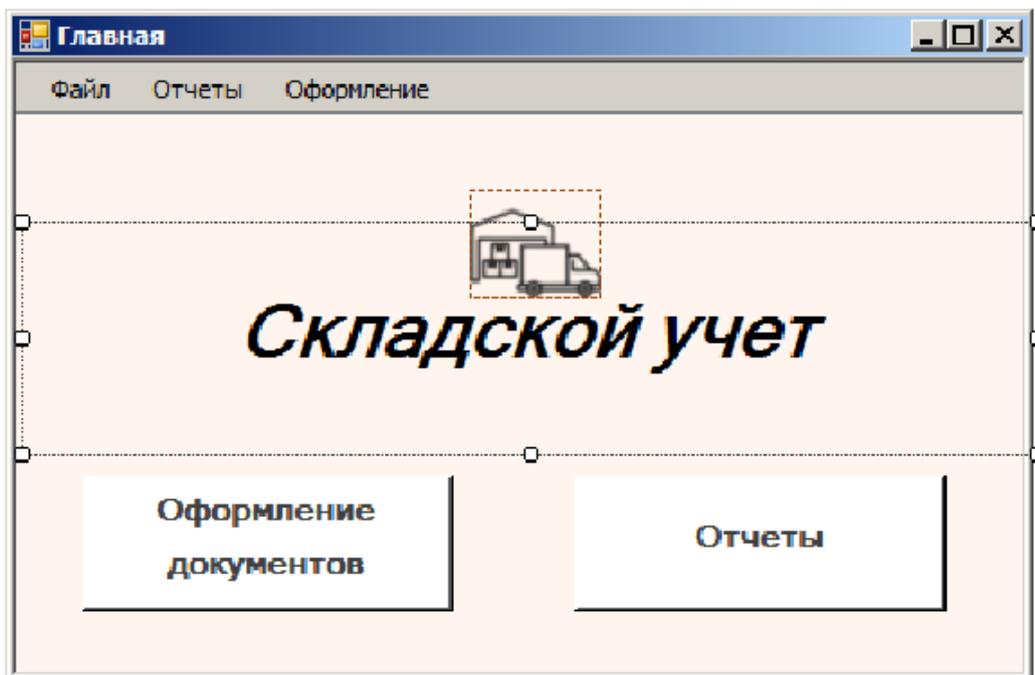


Рисунок 2.3 — Внешний вид главной формы приложения Складской учет

По переходу через кнопки программа выдает следующий вид формы Оформление документов (рисунок 2.4). В ней используются такие элементы управления как MenuStrip и Button в количестве 6 штук, Label.

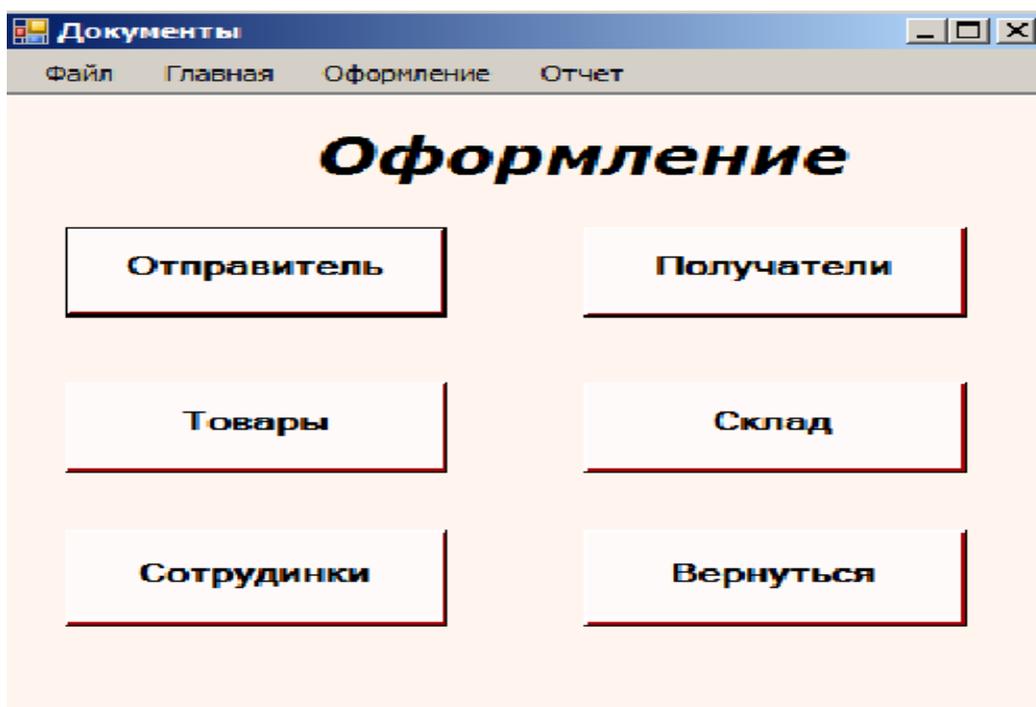


Рисунок 2.4 — Внешний вид формы Оформления документов

В данной форме так же кнопки «Отправитель», «Получатели», «Товары», «Склад» и «Сотрудники» прописаны на открытия форм с базами данных, а кнопка Вернуться для возвращения в главное меню. По этому же примеру работает вторая форма с отчетами (рисунок 2.5). В ней используется MenuStrip и Button в количестве 2 штук, Label. В ней так же, как в форме Документы кнопки запрограммированы на открытие форм таких как Остатки на складе и выхода на главное меню.

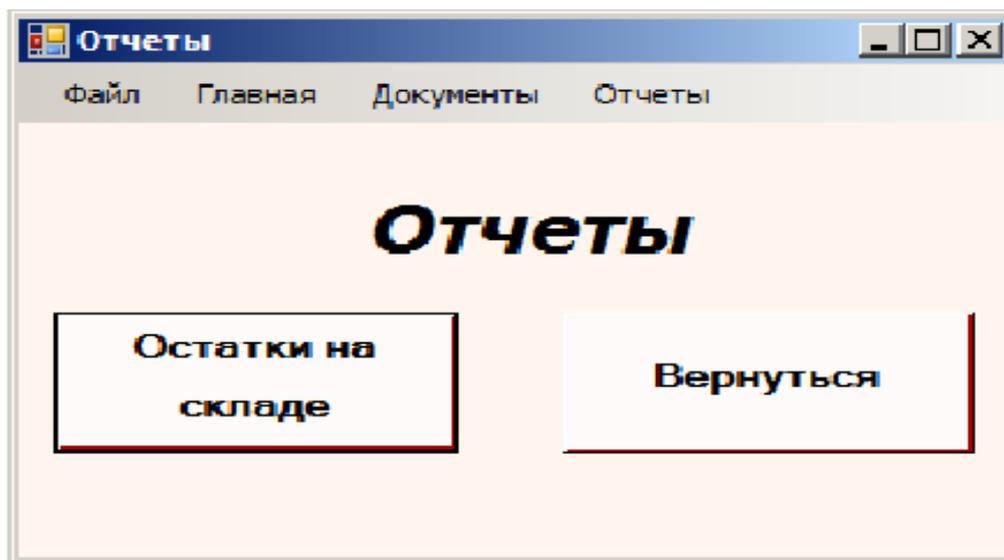


Рисунок 2.5 — Внешний вид формы Отчет

Программный код для отображения таблицы при нажатии на кнопку (на примере таблицы получатели) (приложение 1) [5].

Следующие формы были созданы для редактирования таблиц баз данных. Для их создания были добавлены еще формы, а затем в панели источники данных (Data sources) справа от названия была выбрана таблица и указан тип элемента управления DataGridView. В результате на форму был добавлен элемент управления DataGridView, в котором отражается содержимое данной таблицы, а также элемент управления BindingNavigator, при помощи которого будет осуществляться управление записями данной таблицы (навигация, добавление и удаление). А также к добавленной таблице создадим контекстное меню для заполнения базы данных. Для этого из того же окна (Data sources) переносим каждую позицию отдельно. В данном случае выносятся

Наименование, ФИО, Контактный телефон и Адрес организации (В каждой форме будут свои строки). Для строк Контактный телефон и паспортные данные меняем TextBox на MaskedTextBox для отображения более привычного вида этих строк (рисунок 2.6).

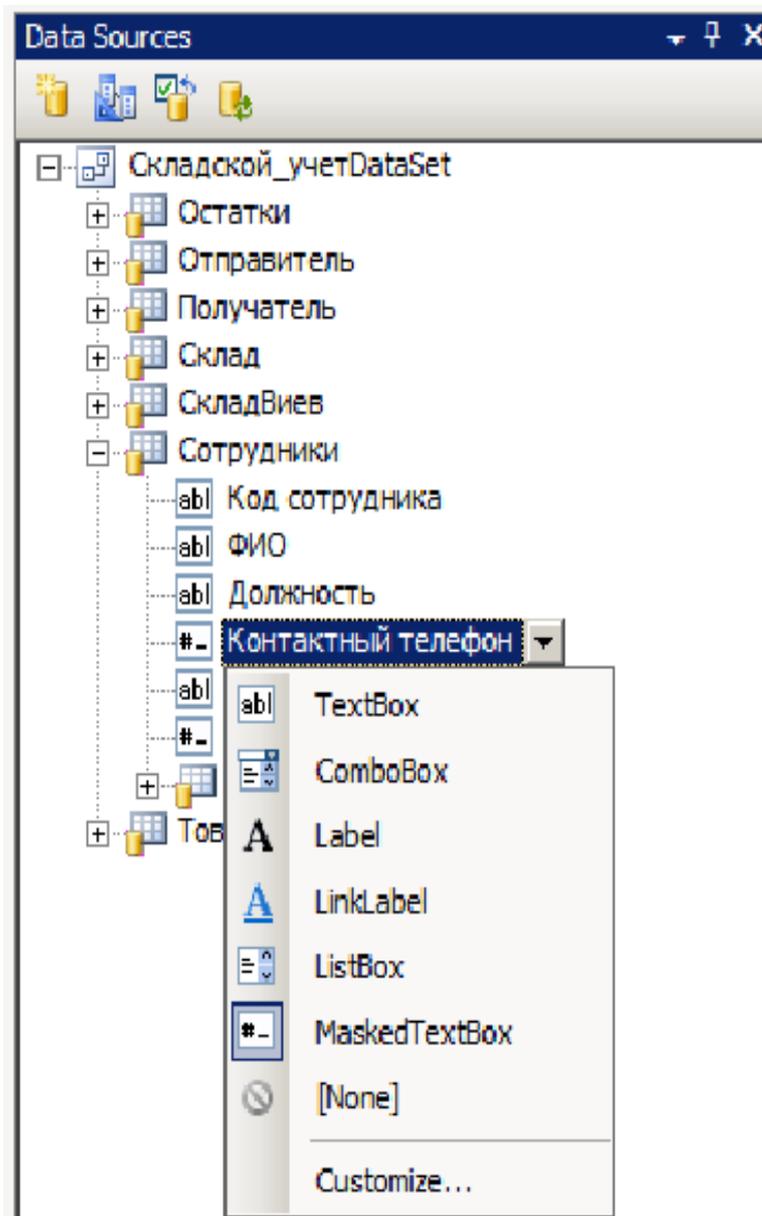


Рисунок 2.7 — Панель «Источники данных»(DataSources)

Добавляем, называем и программируем каждую из 6 кнопок (Предыдущая, Следующая, Удалить, Добавить, Сохранить, Меню). Все данные клавиши отвечают за ввод данных в базу данных кроме меню она отвечает за выход на главную форму. Также используем Label для названия формы. По этому примеру созданы остальные три формы (рисунок 2.7-2.10).

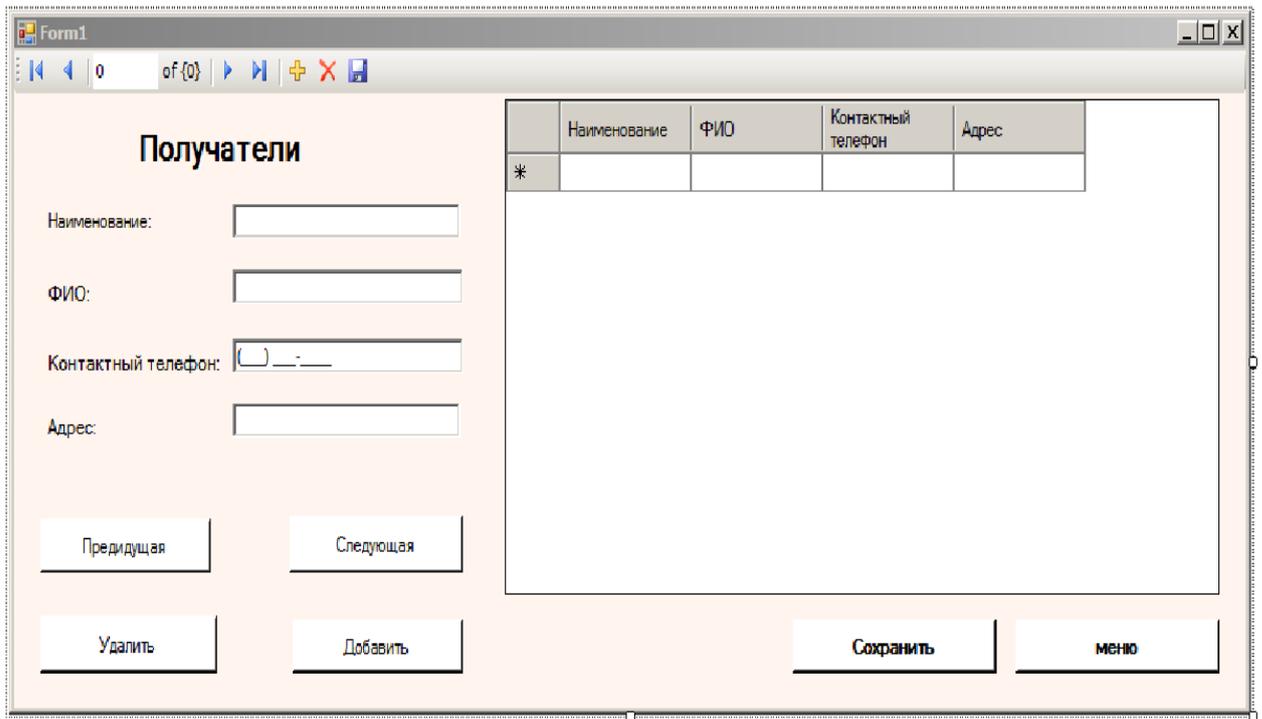


Рисунок 2.7 — Внешний вид формы Отправители

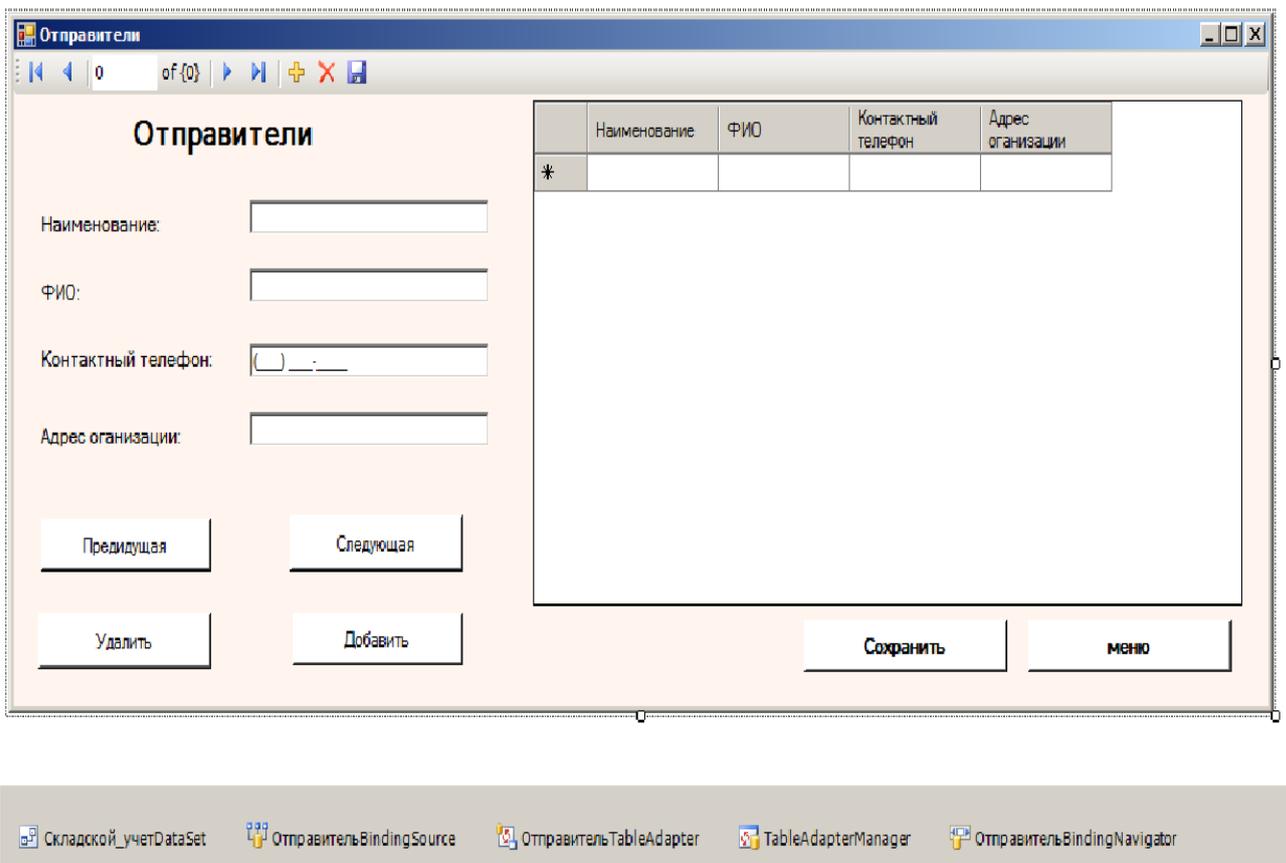


Рисунок 2.8 — Внешний вид формы Получатели

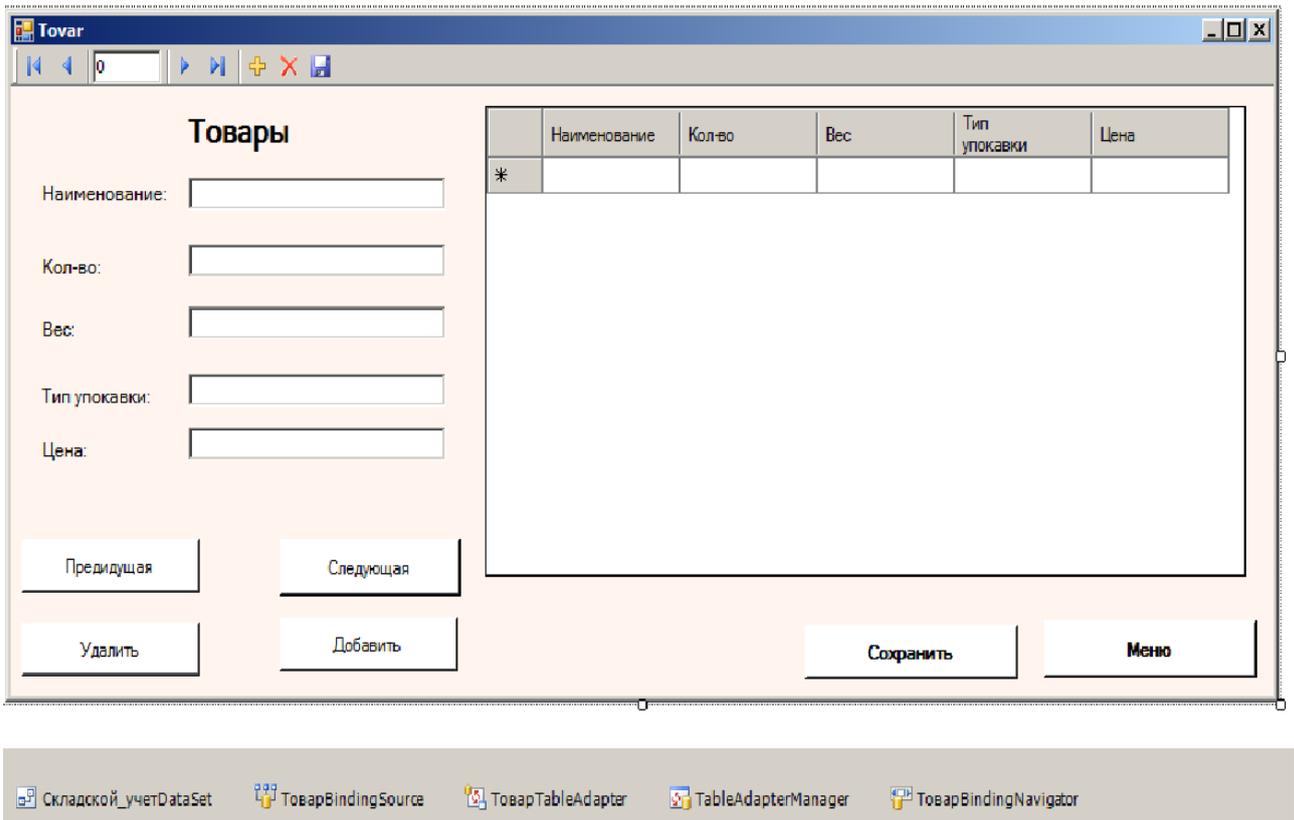


Рисунок 2.9— Внешний вид формы Товары

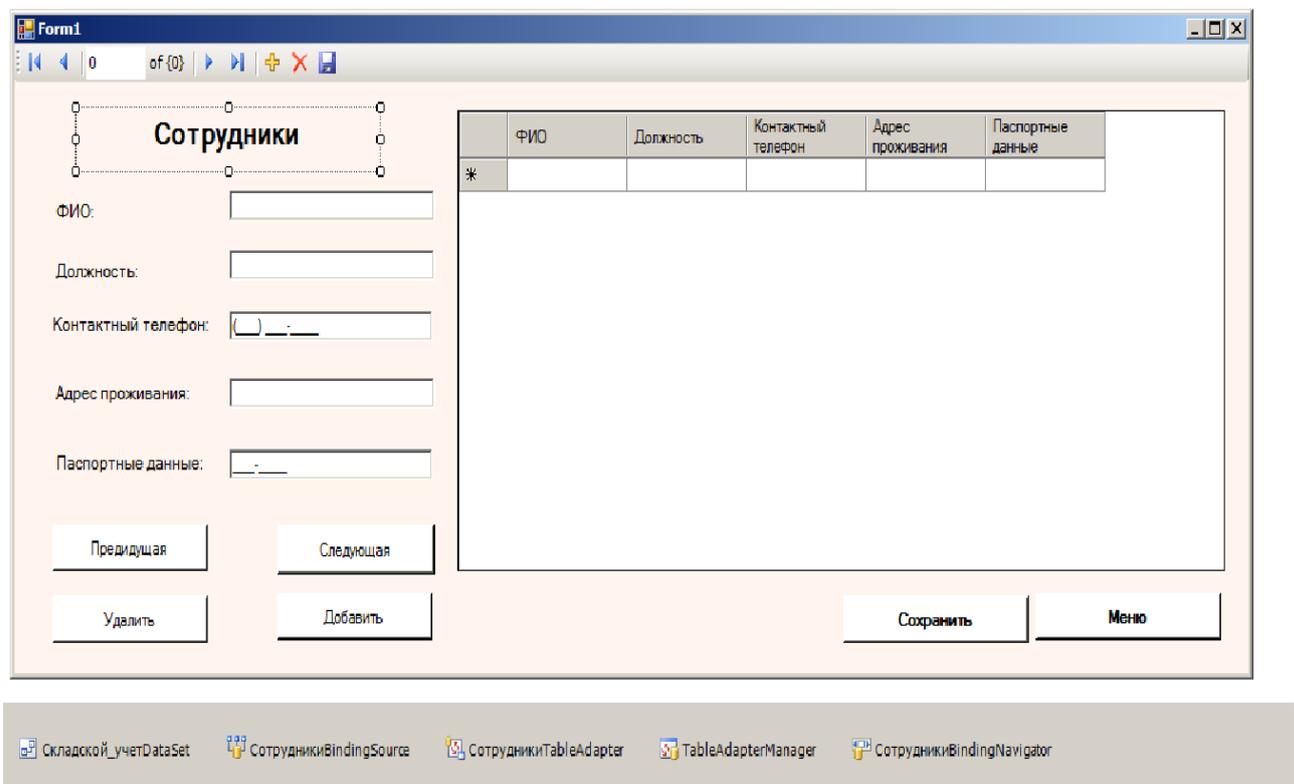


Рисунок 2.10— Внешний вид формы Сотрудники

Так же для этих форм необходимо разработать программный код, он необходим для работы формы. Код показан ниже (Пример на таблице Сотрудники) (приложение 2) [15]

В отличие от других форм, форма Склад имеет другой интерфейс для удобства использования базы данных (рисунок 2.11).

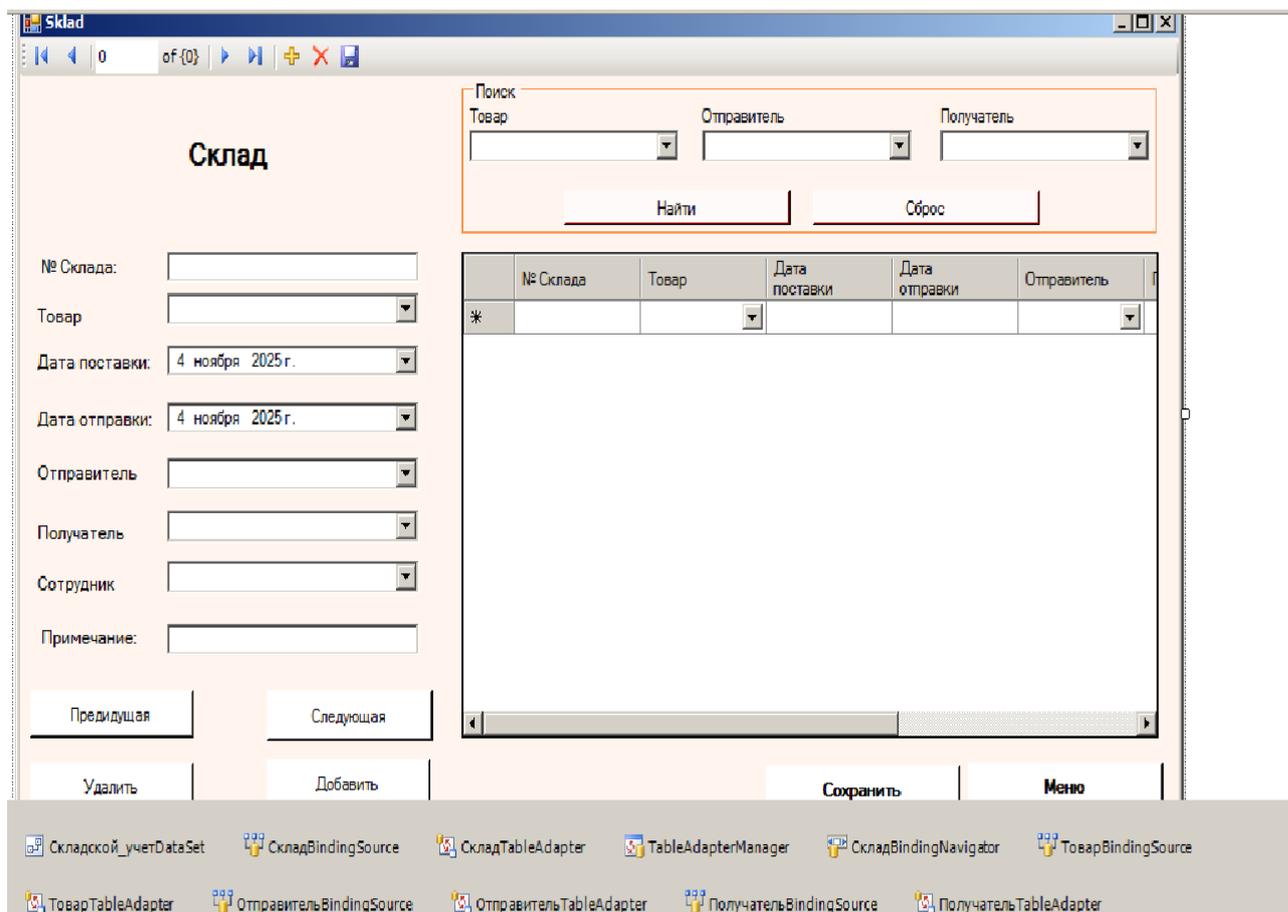


Рисунок 2.11— Внешний вид формы Склад

На этой форме расположены элементы управления DataGridView, Два TextBox, семь ComboBox, Два DateTimePicker, GroupBox, двенадцать Label и восемьButton, чтобы обеспечить пользователю удобную навигацию и использование. Эта форма позволяет вводить данные по складу и использовать поиск для быстрой навигации.

Для того, чтобы связать ListBox с БД, необходимо вызвать его контекстное меню (рисунок 2.12) нажимая на стрелочку, указать значения в

пунктах Data Source (Источник данных), DisplayMember (Член отображения) и ValueMember (Член значений), SelectValue (Выбранное значение). Чтобы в базе данных отображались поля с кодами, а наименование настраиваем их (рисунок 2.13)

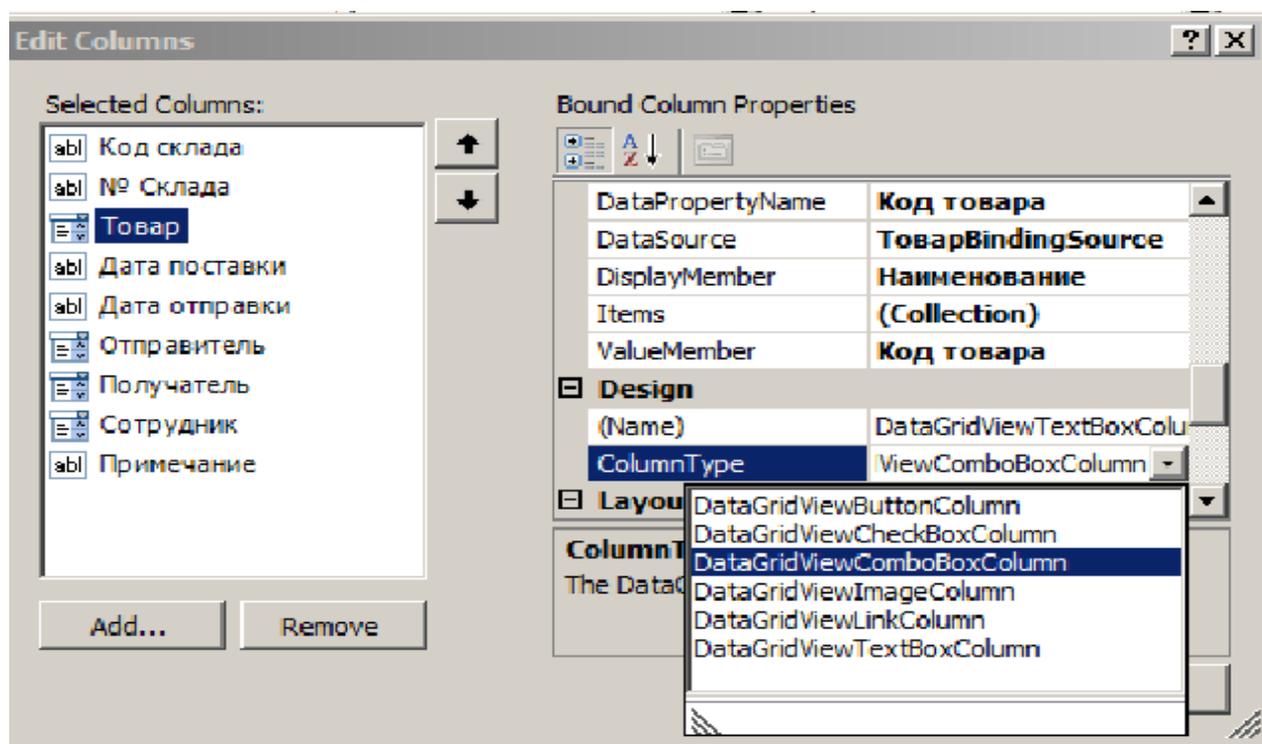


Рисунок 2.12— Окно настройки (EditColumns)

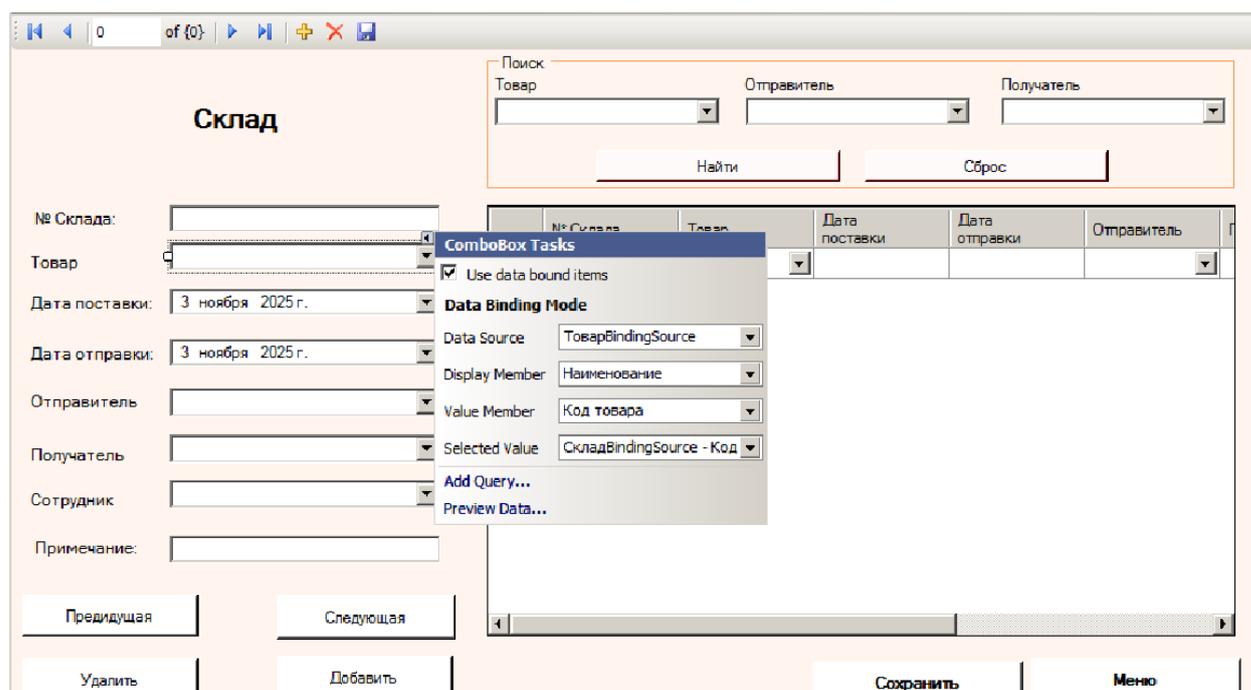


Рисунок 2.12— Контекстное меню ListBox Товар в форме Склад

Для этой формы был разработан программный код, он необходим для работы формы. (приложение 3)

Так же для проекта необходимо создать отчёт, для этого в программе Microsoft Visual Studio нужно создать новый проект ReportServerProjectWizard (рисунок 2.13).

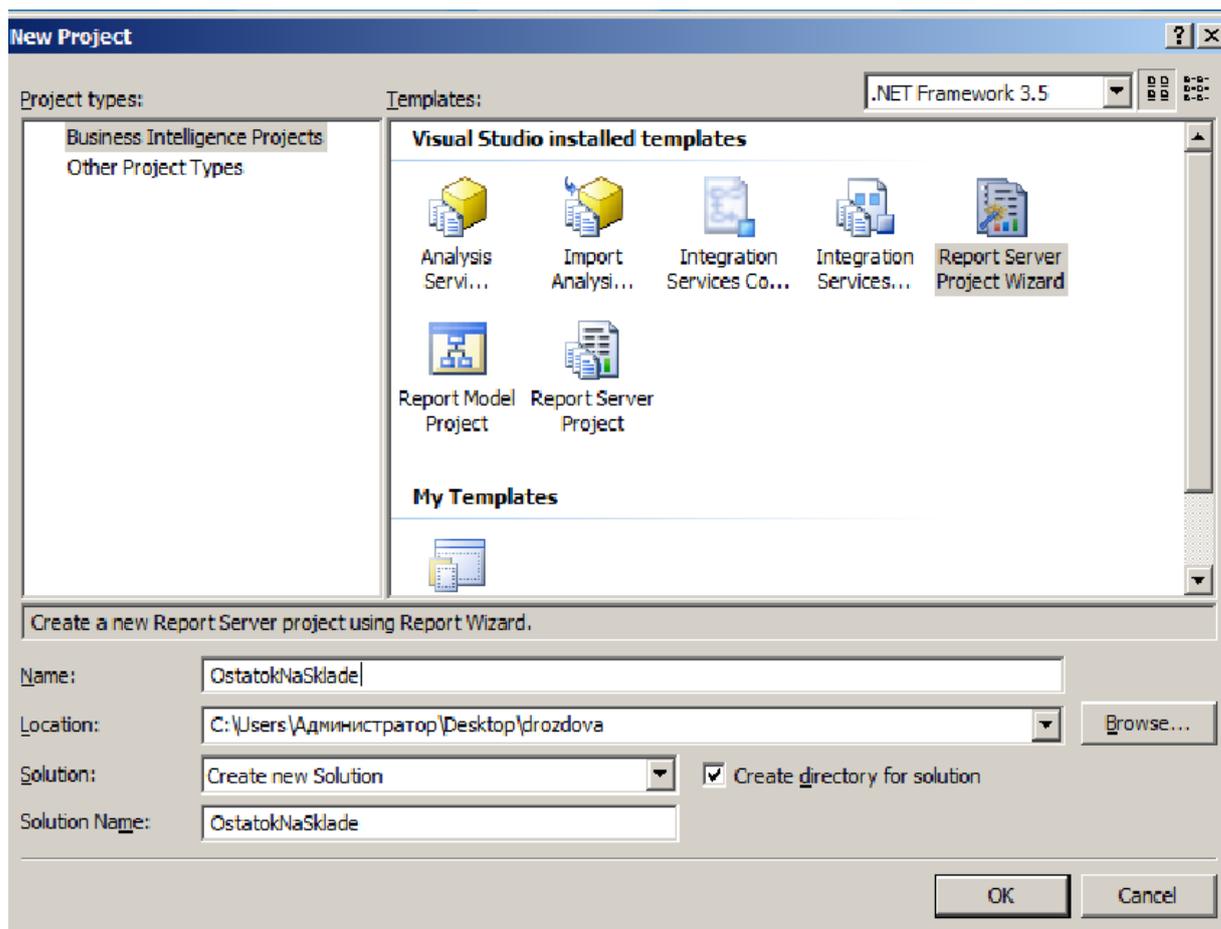


Рисунок 2.13 — Создание нового проекта отчёта в программе SQLServerBusinessIntelligenceDevelopmentStudio

После подключения базы данных к проекту, необходимо добавить запрос. Для этого надо нажать на QueryBuilder, откроется дизайнер запросов (рисунок 2.14), потом выбираем таблицы и столбцы из них (поставить галочку). Так же мы прописываем код, который будет показывать сколько Товаров осталось на складе на сегодняшний день. Этот код указан в приложении 4

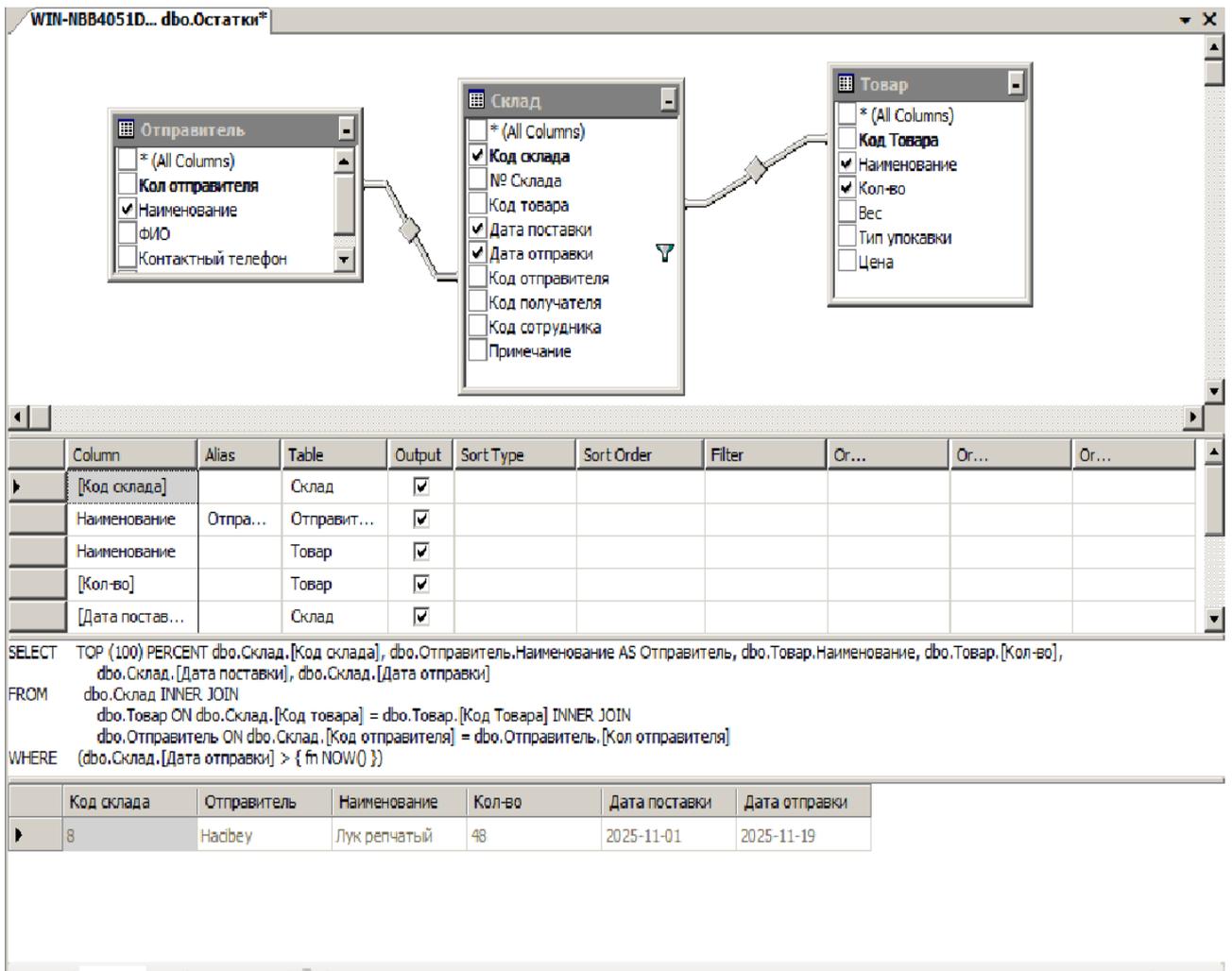


Рисунок 2.14 — Интерфейс дизайнера запросов

Так же настраиваем сортировку, делаем это через пункт Filter. Получившийся запрос показан на рисунке (рисунок 2.15)

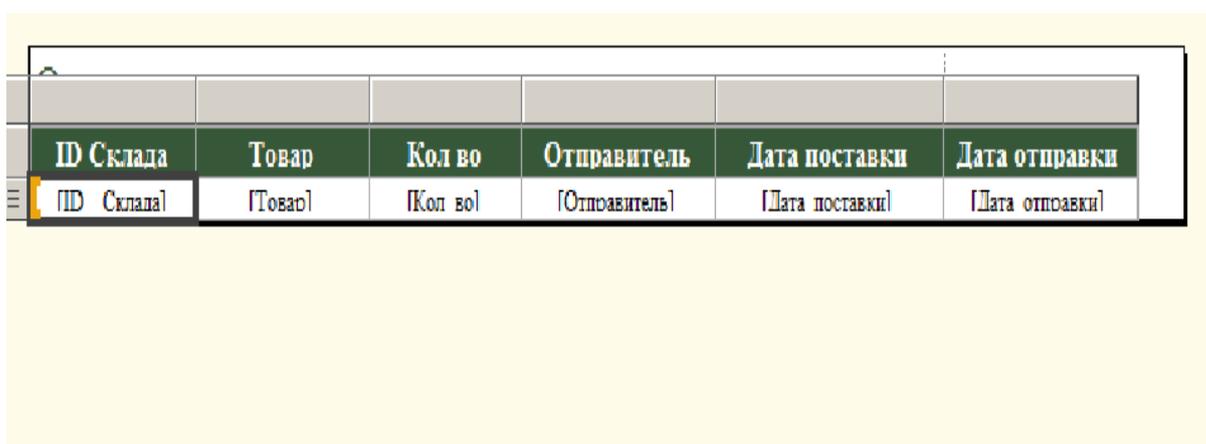


Рисунок 2.14 — Вид готового запроса

Теперь заходи м MicrosoftVisualBasic 2008 ExpressEdition и создаем форму отражающую отчет.

Для этого создаем новую форму и перетаскиваем на нее объект отражающий отчет «MicrosoftReportViewer» на панели объектов (Toolbox). Через нее подключаем созданный ранее отчет (рисунок 2.16)

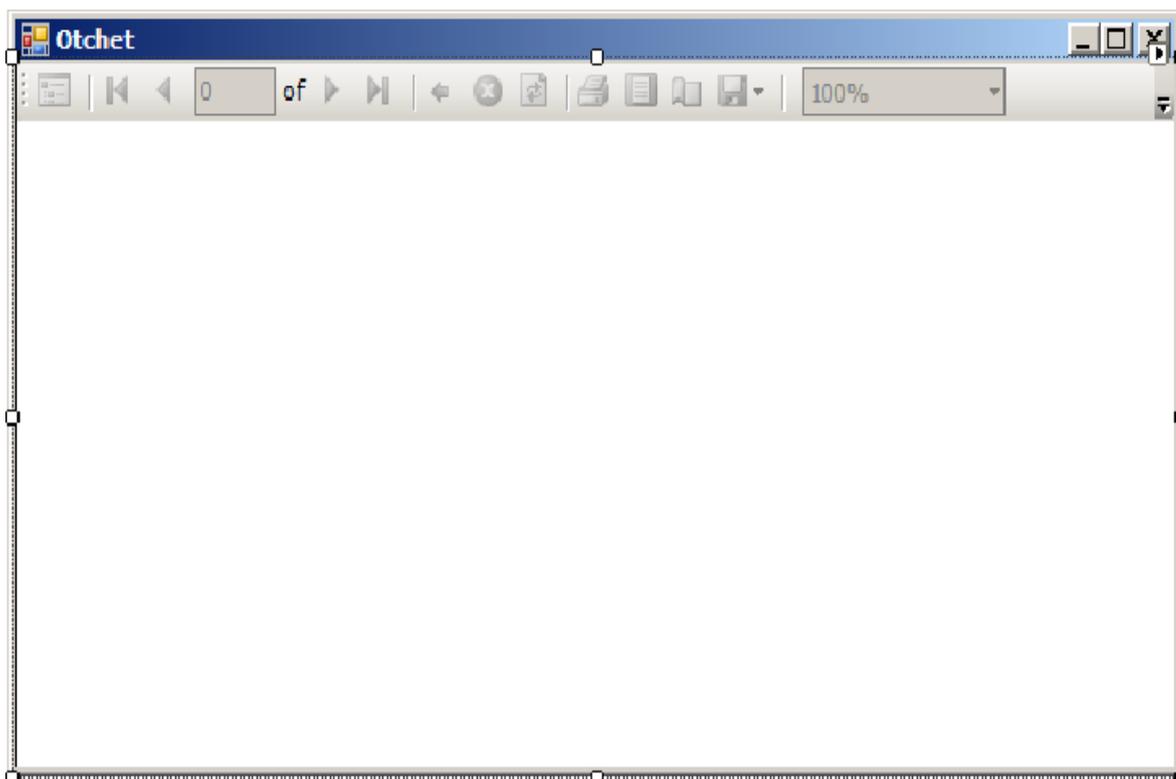


Рисунок 2.16— Внешний вид формы Отчёт

2.5 Руководство пользователя

Руководство пользователя описывает порядок работы с разработанным приложением «Складской учет». Интерфейс программы выполнен в стиле классических Windows-приложений с использованием меню, панелей навигации и форм ввода данных, что обеспечивает интуитивную понятность и быстрое освоение.

Главное меню и навигация. После запуска приложения пользователь попадает на главный экран с названием «Складской учет» (рисунок 2.3). Здесь представлено два основных пункта навигации: «Оформление документов» и «О

тчеты» (рисунок 2.4, 2.5). Выбор «Оформление документов» открывает главное меню приложения с вкладками: «Файл», «Главная», «Оформление», «Отчет». Для доступа к основным функциям используется вкладка «Оформление», которая содержит меню с кнопками: «Отправитель», «Получатели», «Товары», «Склад», «Сотрудники» и «Вернуться». Эта структура обеспечивает прямой доступ ко всем модулям системы. Альтернативно, из вкладки «Отчеты» в главном меню можно напрямую перейти к форме «Остатки на складе».

Работа со справочниками. Все справочники («Товары», «Отправители», «Получатели», «Сотрудники») имеют единообразный интерфейс:

The screenshot shows a window titled 'Form1' with a toolbar at the top. The main content area is titled 'Сотрудники' and contains the following elements:

- Input fields for:
 - ФИО: Степанов Данила Ильич
 - Должность: Грузчик
 - Контактный телефон: (101) 515-654
 - Адрес проживания: Туапсеё
 - Паспортные данные: 215-1655
- Navigation buttons: 'Предыдущая', 'Следующая', 'Удалить', 'Добавить', 'Сохранить', and 'Меню'.
- A table with the following data:

	ФИО	Должность	Контактный телефон	Адрес проживания	Паспортные данные
▶	Степанов Дани...	Грузчик	(101) 515-654	Туапсеё	215-1655
	Степанов Дани...	Сладчик	(515) 616-8545	Новороссийск	145-1368
*					

Рисунок 2.18 —Форма Сотрудники

Отправители

1 для 3

Отправители

Наименование:

ФИО:

Контактный телефон:

Адрес организации:

	Наименование	ФИО	Контактный телефон	Адрес организации
▶	Gol-Den	Кравцова	(165) 151-6514	Турция
	Hacıbey	Надар	(615) 151-5151	Уругвай
	FANO	Расим	(161) 545-4447	Турция
*				

Рисунок 2.19 —Форма Отправители

Form1

1 для 3

Получатели

Наименование:

ФИО:

Контактный телефон:

Адрес:

	Наименование	ФИО	Контактный телефон	Адрес
▶	Фрутелла	Иванова Полин...	(145) 215-1656	Сочи
	ВТП	Лебедев Алекс...	(214) 563-256	Краснодар
	Империя Вкуса	Орлов Владисл...	(571) 525-1561	Москва
*				

Рисунок 2.20 —Форма Получатели

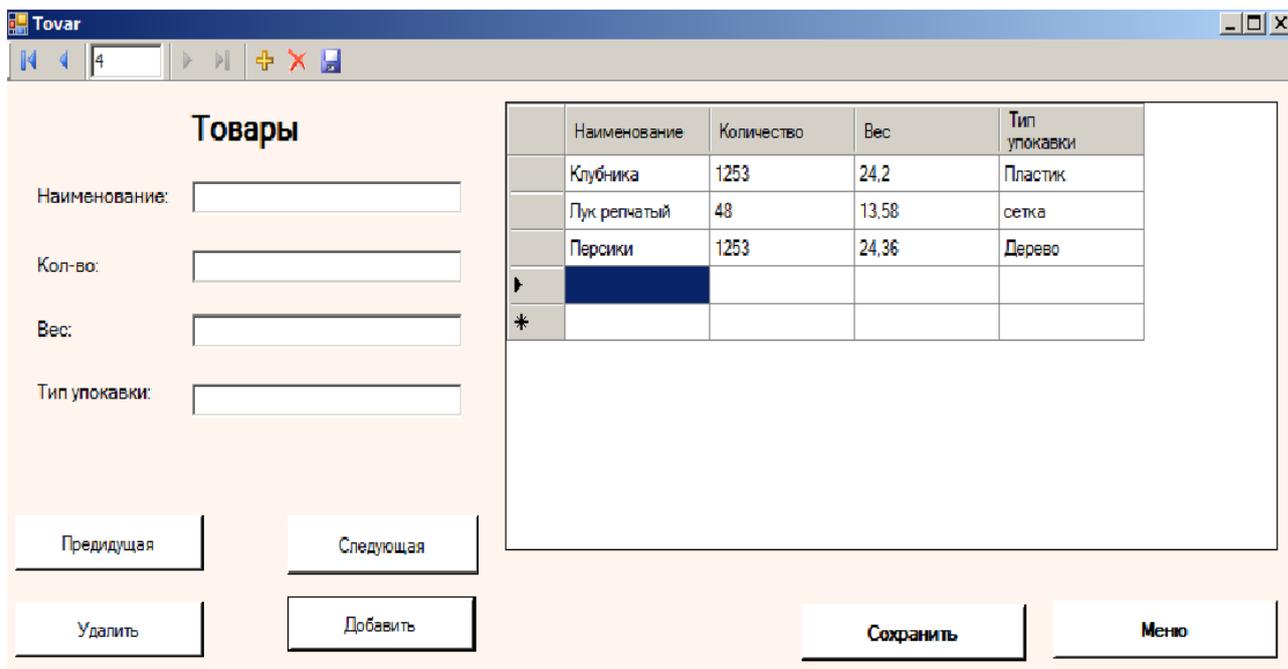


Рисунок 2.21 —Форма Товары

Форма состоит из двух частей: в верхней части расположены поля для ввода и редактирования данных отдельной записи, в нижней – таблица для просмотра всего списка записей.

1. Форма «Товары»: содержит поля: «Наименование» (например, «Клубника», «Лук репчатый», «Персики»), «Кол-во» (количество на остатке, например, 1253, 48), «Вес» (например, 24.2, 13.58) и «Тип упаковки» (например, «Пластик», «Дерево», «сетка»). Под полями ввода расположены кнопки навигации «предыдущая» / «Следующая» для пошагового просмотра записей, а также кнопки «Удалить», «Добавить», «Сохранить» и «Меню» (возврат в главное меню).

2. Форма «Сотрудники»: содержит поля: «ФИО» (например, «Степанов Данила Ильич»), «Должность» («Грузчик», «Складчик»), «Контактный телефон» (формат «(101) 515-554»), «Адрес проживания» («Туапсе», «Новороссийск»), «Паспортные данные» («215-1655»). Управление осуществляется теми же кнопками навигации и действий.

3. Формы «Отправители» и «Получатели»: имеют схожую структуру. Для отправителя заполняются поля «Наименование» («GolDen», «Nacibeuy»), «ФИО» контактного лица («Кровцова», «Надар»), «Контактный телефон» и «Адрес организации» («турция», «Уругвай»). Для получателя – «Наименование» («Фрутелла», «ВТП»), «ФИО», «Контактный телефон» и «Адрес» («Сочи», «Краснодар»).

Для добавления новой записи в любой справочник необходимо нажать кнопку «Добавить», очистить поля формы, ввести новые данные и нажать «Сохранить». Запись появится в таблице в нижней части окна. Для редактирования нужно выбрать запись в таблице (или найти с помощью кнопок «Предыдущая»/«Следующая»), внести изменения в поля и снова нажать «Сохранить». Удаление производится кнопкой «Удалить» после выбора записи.

Оформление складских операций. Основная операция проводится в форме «Склад» (меню «Оформление» -«Склад»).

The screenshot shows a software window titled 'sklad'. The main form is titled 'Склад' and contains the following elements:

- Search Section:** A search bar with filters for 'Товар' (Goods) set to 'Клубника', 'Отправитель' (Sender) set to 'Gol-Den', and 'Получатель' (Receiver) set to 'Фрутелла'. Below are 'Найти' (Find) and 'Сброс' (Reset) buttons.
- Form Fields:**
 - № Склада: 1337
 - Товар: Клубника
 - Дата поставки: 4 июня 2025 г.
 - Дата отправки: 5 июня 2025 г.
 - Отправитель: Gol-Den
 - Получатель: Фрутелла
 - Сотрудник: Степанов Данила Ильич
 - Примечание: 5324
- Table:** A table with columns: № Склада, Товар, Дата поставки, Дата отправки, Отправитель.

№ Склада	Товар	Дата поставки	Дата отправки	Отправитель
1337	Клубника	04.06.2025	05.06.2025	Gol-Den
452	Персики	05.06.2025	20.06.2025	Gol-Den
852	Лук репчатый	01.06.2025	03.06.2025	Nacibeuy
74	Лук репчатый	29.05.2025	06.06.2025	Nacibeuy
1337	Лук репчатый	01.11.2025	21.06.2025	Nacibeuy
*				
- Buttons:** 'Предыдущая', 'Следующая', 'Удалить', 'Добавить', 'Сохранить', and 'Меню'.

Рисунок 2.22 —Форма Склад

Форма служит для регистрации как прихода (от поставщика), так и расхода (клиенту) товара. Поля формы:

1. «№ Склада»: Уникальный номер документа/операции (например, 1337, 452).
2. «Товар»: Выпадающий список для выбора товара из справочника (напр., «Клубника», «Лук репчатый»).
3. «Дата поставки» и «Дата отправки»: выбираются с помощью встроенного календаря (формат DD.MM.YYYY, как на скрине: 04.06.2025).
4. «Отправитель» и «Получатель»: Выпадающие списки для выбора контрагентов. При оформлении прихода заполняется только «Отправитель», при отгрузке – только «Получатель».
5. «Сотрудник»: Выпадающий список для выбора ответственного сотрудника.
6. «Примечание»: Произвольное текстовое поле (например, «5324»).

В нижней части формы, как и в справочниках, находится таблица с историей всех проведенных операций, отображающая ключевые поля: № Склада, Товар, Даты, Отправитель. Для создания новой операции нажимается кнопка «Добавить», заполняются все необходимые поля и нажимается «Сохранить». Система автоматически обновляет остаток выбранного товара в справочнике «Товары». Навигация по существующим операциям осуществляется кнопками «Предыдущая»/«Следующая» или выбором строки в таблице.

Формирование отчетов. Отчеты доступны через главный экран («Отчеты») или меню («Отчет» ->«Остатки на складе»).

ID Склада	Товар	Кол во	Отправитель	Дата поставки	Дата отправки
1337	Лук репчатый	48	Nasibey	11/1/2025 12:00:00 AM	11/19/2025 12:00:00 AM

Рисунок 2.23 —Форма Отчет

Основной отчет – «Остатки». При его открытии отображается таблица с текущими остатками товаров на складе, сформированная на основе всех

проведенных операций. Отчет включает колонки: «ID Склада» (номер последней связанной операции), «товар», «Колво» (остаток), «Отправитель» (последний поставщик), «Дата поставки» и «Дата отправки» (даты последней операции). Данные в отчете актуальны на момент его открытия и обновляются автоматически при каждом проведении новой складской операции.

3 Обоснование экономической эффективности результатов

3.1 Выбор и обоснование методики расчета экономической эффективности

Для оценки экономической эффективности проекта внедрения разработанного информационного приложения используется методика сопоставления приведенных затрат по вариантам (до и после автоматизации) и расчета срока окупаемости капитальных вложений (простая статическая модель) [22].

Обоснование выбора данной методики:

Практическая применимость: Метод является наиболее понятным для заказчика и позволяет наглядно продемонстрировать эффект от проекта.

Достаточность данных: для предварительной оценки эффективности достаточно данных о снижении трудоемкости, сокращении ошибок и единовременных затратах на разработку. Отраслевая стандартность: Метод широко применяется для оценки эффективности внедрения информационных систем на малых и средних предприятиях.

Акцент на основные источники экономии: в рамках дипломного проекта ключевыми являются показатели снижения трудовых затрат и операционных потерь, которые данная методика позволяет четко выделить.

Основные этапы расчета по выбранной методике:

Определение единовременных капитальных затрат (К) на разработку и внедрение системы.

Расчет годовой экономии текущих затрат (ΔC), достигнутой за счет внедрения.

Расчет годового экономического эффекта ($\Delta \Gamma$).

Определение расчетного срока окупаемости ($T_{ок}$).

Формулы расчета:

Годовой экономический эффект: $\Delta \Gamma = \Delta C$

Срок окупаемости: $T_{ок} = K / \Delta \Gamma$

Данная модель не учитывает фактор времени (дисконтирование), так как срок окупаемости проекта ожидается менее одного года, что делает такое упрощение допустимым.

3.2 Расчет показателей экономической эффективности

Исходные данные для расчета (сформированы на основе анализа предметной области) [26]:

Таблица 3.1. - Текущее состояние (до автоматизации)

Показатель	Ед. изм.	Значение	Примечание
Средняя ЗП кладовщика (с отчислениями)	руб./мес.	45 000	С учетом страховых взносов ($\approx 30\%$)
Затраты времени на учетные операции	час/день	3,0	37.5% рабочего времени (8-часовой день)
Доля ошибок, приводящих к потерям	%	6,0	От стоимости учитываемых ТМЦ
Стоимость ТМЦ на складе (среднегодовая)	руб.	5 000 000	

Таблица 3.2. - Внедряемое решение (после автоматизации)

Показатель	Ед. изм.	Значение	Примечание
Ожидаемое сокращение времени на операции	%	65	Целевой показатель

Продолжение таблицы 3.2.

Ожидаемое снижение доли ошибок	%	85	С 6% до ~0.9%
Единовременные затраты на разработку (К)	руб.	140 000	Включая проектирование, кодирование, тестирование
Ожидаемое сокращение времени на операции	%	65	Целевой показатель

Таблица 3.3 - Прочие данные

Показатель	Ед. изм.	Значение	Примечание
Количество рабочих дней в году	дни	247	
Количество сотрудников склада	чел.	3	Кладовщики, работающие с учетом

Расчет 1. Годовая экономия на фонде оплаты труда (ΔСфот).

Экономия возникает за счет высвобождения рабочего времени сотрудников.

Годовые трудозатраты на учет до автоматизации:

$$3 \text{ чел.} * 3 \text{ час/день} * 247 \text{ дн.} = 2\,223 \text{ чел.-час/год}$$

Годовые трудозатраты на учет после автоматизации (сокращение на 65%):

$$2\,223 \text{ чел.-час} * (1 - 0.65) = 778 \text{ чел.-час/год}$$

Высвобождаемые трудозатраты:

$$2\,223 - 778 = 1\,445 \text{ чел.-час/год}$$

Годовая экономия на ФОТ (при условии, что высвобожденное время перенаправляется на полезную работу, что позволяет избежать найма нового сотрудника при росте объемов):

$$\text{Часовая ставка: } 45\,000 \text{ руб./мес.} / (22 \text{ дня} * 8 \text{ час}) \approx 256 \text{ руб./час}$$

$$\Delta C_{\text{фот}} = 1\,445 \text{ час} * 256 \text{ руб./час} \approx 369\,920 \text{ руб./год}$$

Расчет 2. Годовая экономия от снижения потерь из-за ошибок учета ($\Delta C_{\text{ош}}$).

Годовой объем потерь до автоматизации:

$$5\,000\,000 \text{ руб.} * 6\% = 300\,000 \text{ руб./год}$$

Годовой объем потерь после автоматизации (ошибки снижаются на 85%):

$$300\,000 \text{ руб.} * (1 - 0.85) = 45\,000 \text{ руб./год}$$

Годовая экономия от снижения потерь:

$$\Delta C_{\text{ош}} = 300\,000 - 45\,000 = 255\,000 \text{ руб./год}$$

Расчет 3. Суммарная годовая экономия текущих затрат (ΔC).

$$\Delta C = \Delta C_{\text{фот}} + \Delta C_{\text{ош}} = 369\,920 + 255\,000 = 624\,920 \text{ руб./год}$$

Для дальнейших расчетов принимаем округленное значение годовой экономии $\Delta C = 625\,000$ рублей.

Расчет 4. Срок окупаемости единовременных капитальных вложений (Ток).

$$\text{Ток} = K / \Delta C = 140\,000 \text{ руб.} / 625\,000 \text{ руб./год} \approx 0.22 \text{ года}$$

Перевод в месяцы:

$$0.22 \text{ года} * 12 \text{ месяцев} = 2.64 \text{ месяца}$$

Вывод по сроку окупаемости: Расчетный срок окупаемости проекта составляет менее 3-х месяцев.

Расчет 5. Расчетный годовой экономический эффект ($\Delta \Gamma$).

Так как в модели не учитываются дополнительные эксплуатационные затраты на ПО (они ничтожно малы), годовой экономический эффект равен годовой экономии:

$$\text{Эг} = \Delta C = 625\,000 \text{ руб./год}$$

Таблица 3.4.- Сводная таблица экономической эффективности

Наименование показателя	Значение	Единица измерения
Единовременные капитальные вложения (К)	140 000	руб.
Годовая экономия на ФОТ ($\Delta C_{\text{фот}}$)	~370 000	руб./год
Годовая экономия от снижения потерь ($\Delta C_{\text{опш}}$)	255 000	руб./год
Суммарная годовая экономия (ΔC)	625 000	руб./год
Расчетный срок окупаемости (Ток)	~2.6	месяца
Годовой экономический эффект (Эг)	625 000	руб./год
Наименование показателя	Значение	Единица измерения

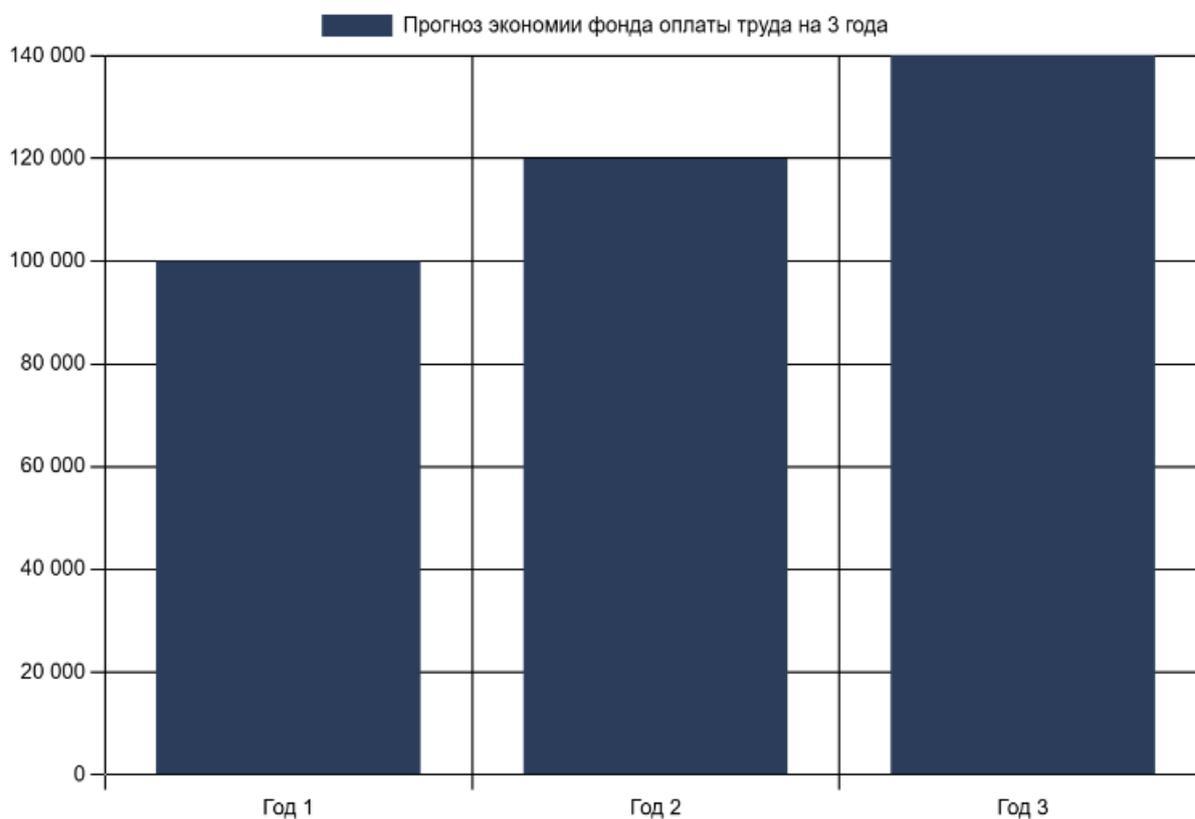


График 3.1.- Прогноз экономии фонда оплаты труда на 3 года

Интерпретация результатов и выводы:

1. Проект обладает исключительно высокой экономической эффективностью. Срок окупаемости в 2.6 месяца является крайне низким для любого инвестиционного проекта и свидетельствует о минимальных рисках и высокой отдаче.

2. Основной источник экономии (59%) формируется за счет повышения производительности труда и высвобождения ресурсов для роста бизнеса без увеличения штата.

3. Значительный вклад (41%) вносит снижение прямых финансовых потерь, что повышает общую рентабельность операционной деятельности.

4. Полученные расчеты носят консервативный характер, так как не учитывают дополнительные качественные преимущества (повышение скорости обслуживания клиентов, улучшение управляемости, снижение коммерческих рисков), которые также имеют экономическую ценность

Таким образом, внедрение разработанного информационного приложения для автоматизации складского учета является экономически целесообразным и высокоэффективным. Проект не только окупится в рекордно короткие сроки, но и начнет приносить существенную ежегодную экономию, способствуя повышению конкурентоспособности предприятия.

Заключение

В ходе выполнения выпускной квалификационной работы была достигнута поставленная цель – разработана и внедрена информационная система автоматизации складского учета на основе технологий Visual Basic 2008 и SQL Server [28].

Основные научные и практические результаты работы:

1. Проведен комплексный анализ предметной области – изучены современные подходы к автоматизации складской деятельности, проанализированы бизнес-процессы предприятия, выявлены проблемы существующей системы учета и сформулированы требования к автоматизированной системе.

2. Разработана архитектура информационной системы – спроектирована трехуровневая структура приложения, выбраны и обоснованы технологии разработки, создана нормализованная модель базы данных, обеспечивающая целостность и непротиворечивость хранимой информации.

3. Реализовано программное обеспечение – создана база данных с оптимальной структурой таблиц, разработан клиентский интерфейс с удобным пользовательским интерфейсом, реализованы модули для работы со справочной информацией, складскими операциями и отчетностью.

4. Проведено всестороннее тестирование системы – использована методология тестирования на основе тестовых наборов, которая подтвердила соответствие системы функциональным требованиям, продемонстрировала стабильность работы и достаточный уровень производительности.

Научная новизна и практическая значимость: Разработанная система представляет собой законченное программное решение, адаптированное под специфику российских предприятий малого и среднего бизнеса. Особенностью работы является оптимальное сочетание современных принципов проектирования информационных систем с использованием

стабильных и проверенных технологий, что обеспечивает надежность и простоту сопровождения системы.

Практическая значимость работы подтверждается успешным внедрением системы на предприятии и достигнутыми экономическими показателями. Разработанное решение может быть тиражировано на другие предприятия схожего профиля.

Перспективы дальнейшего развития:

В качестве направлений для дальнейшего развития системы можно выделить:

1. Разработку мобильного приложения для работы кладовщиков
2. Интеграцию с системами электронного документооборота
3. Внедрение технологии штрихкодирования для учета товаров
4. Развитие аналитического модуля с функциями прогнозирования

Таким образом, все задачи дипломной работы успешно решены, цель достигнута. Разработанная система автоматизации складского учета готова к промышленной эксплуатации и способна обеспечить повышение эффективности управления складскими запасами.

Список литературы

1. Маркин А. В. Базы данных. PostgreSQL— Москва: Издательство Юрайт, 2025. — 828 с. — ISBN 978-5-534-21779-7. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/582050> (дата обращения: 21.09.2025).
2. Советов Б. Я., Цихановский В. В. Информационные технологии— 8-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2024. — 414 с. — ISBN 978-5-534-20054-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/557506> (дата обращения: 21.09.2025).
3. Стружкин, Н. П., Годин В. В. Базы данных: проектирование— Москва: Издательство Юрайт, 2025. — 477 с. — ISBN 978-5-534-00229-4. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560310> (дата обращения: 27.09.2025).
4. Казанский, А. А. Объектно-ориентированное программирование. Visual Basic— 2-е изд. — Москва: Издательство Юрайт, 2025. — 295 с. — ISBN 978-5-534-21383-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/569867> (дата обращения: 04.10.2025).
5. Маркин, А. В. Программирование на SQL— 3-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2025. — 805 с. — ISBN 978-5-534-18371-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/568900> (дата обращения: 10.09.2025).
6. Баламирзоев Н. Л. «Проблемы внедрения информационной системы управления предприятием» [Электронный ресурс]. — Режим доступа: <https://e-koncept.ru/2014/54385.htm> <https://e-koncept.ru/2014/54385.htm> (дата обращения: 27.10.2025).

7. Гаврилов, Л. П. Инновационные технологии в коммерции и бизнесе:—Москва: Издательство Юрайт, 2025. — 277 с. — ISBN 978-5-534-21730-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/581988> (дата обращения: 10.08.2025).

8. Чекмарев, А. В. Управление цифровыми проектами и процессами: 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2025. — 424 с. — (Высшее образование). — ISBN 978-5-534-18522-5. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/index.php/bcode/564520> (дата обращения: 12.09.2025).

9. Информационные системы и технологии в экономике и управлении в 2 ч. Часть 1 / ответственный редактор В. В. Трофимов. — 5-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2025. — 375 с. — (Высшее образование). — ISBN 978-5-534-09090-1. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/564598> (дата обращения: 08.08.2025).

10. Проектирование информационных систем: / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2025. — 273 с. — ISBN 978-5-534-20361-5. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560485> (дата обращения: 15.11.2025).

11. Ложкин Н. Д. Автоматизация и учет на складе готовой продукции // Colloquium-journal. 2021. №12 (99). URL: <https://cyberleninka.ru/article/n/avtomatizatsiya-i-uchet-na-sklade-gotovoy-produktsii> (дата обращения: 06.08.2025).

12. Парикова Ирина Вячеславовна. Использование терминалов сбора данных при автоматизации складского учета // Скиф. 2020. №6 (46). URL: <https://cyberleninka.ru/article/n/ispolzovanie-terminalov-sbora-dannyh-pri-avtomatizatsii-skladskogo-ucheta> (дата обращения: 21.10.2025).

13. Басов А. С. Сравнение современных СУБД // Вестник науки. 2020. №7 (28). URL: <https://cyberleninka.ru/article/n/sravnenie-sovremennyh-subd> (дата

обращения: 21.09.2025).

14. Шагулыев Ш. А., Гельдыев Г. Б. ОСНОВЫ МОДЕЛИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ // Вестник науки. 2022. №11 (56). URL: <https://cyberleninka.ru/article/n/osnovy-modelirovaniya-informatsionnyh-sistem> (дата обращения: 21.09.2025).

15. Прокофьева В. С. Программирование на языке VISUAL BASIC FOR APPLICATION // Теория и практика современной науки. 2016. №12-2 (18). URL: <https://cyberleninka.ru/article/n/programmirovanie-na-yazyke-visual-basic-for-application> (дата обращения: 21.11.2025).

16. Таненбаум Э. Компьютерные сети. 5-е изд. / Э. Таненбаум, Д. Уэзеролл. - Санкт-Петербург: Питер, 2021. - 960 с. - ISBN 978-5-4461-1248-7. - URL: <https://ibooks.ru/bookshelf/373747/reading> (дата обращения: 12.09.2025).

17. Цебренок К. Н. Концепция системы моделирования информационных систем // Международный журнал гуманитарных и естественных наук. 2020. №7-2. URL: <https://cyberleninka.ru/article/n/kontseptsiya-sistemy-modelirovaniya-informatsionnyh-sistem> (дата обращения: 21.10.2025).

18. Жалолов Озод Исомиддинович, Хаятов Хуршид Усманович. Понятие SQL и реляционной базы данных // Universum: технические науки. 2020. №6-1 (75). URL: <https://cyberleninka.ru/article/n/ponyatie-sql-i-relyatsionnoy-bazy-dannyh> (дата обращения: 21.10.2025).

19. Баламирзоев Н. Л. Проблемы внедрения информационной системы управления предприятием // Научно-методический электронный журнал «Концепт». – 2014. – Т. 20. – С. 606–610. – URL: <http://e-koncept.ru/2014/54385.htm>. (дата обращения 14.10.2025).

20. Кочелаба Жанна Викторовна, Севедин Владислав Олегович анализ рынка ERP-систем в России и за рубежом // Сервис в России и за рубежом. 2023. №4 (106). URL: <https://cyberleninka.ru/article/n/analiz-rynka-erp-cistem-v-rossii-i-za-rubezhom> (дата обращения: 21.09.2025).

21. Трофимов С.В. Применение автоматизации для повышения

производительности бизнес–процессов. Путеводитель предпринимателя. 2025;18(2):32-39. <https://doi.org/10.24182/2073-9885-2025-18-2-32-39> (дата обращения 04.07.2025)

22. Т. Г. Черепанова. Автоматизация бизнес-процессов как фактор повышения эффективности предпринимательской деятельности // Экономика и бизнес: теория и практика. 2024. №2-2 (108). URL: <https://cyberleninka.ru/article/n/avtomatizatsiya-biznes-protsessov-kak-faktor-povysheniya-effektivnosti-predprinimatelskoj-deyatelnosti> (дата обращения: 15.12.2025).

23. Попов В. Б., Кузькина Е. А. Разработка базы данных для современной организации // Ученые записки Крымского федерального университета имени В. И. Вернадского. Экономика и управление. - 2015.- №1. URL: <https://cyberleninka.ru/article/n/razrabotka-bazy-dannyh-dlya-sovremennoy-organizatsii> (дата обращения: 11.09.2025).

24. Корытов Г. В., Ильин И. В. Компоненты ERP-системы для автоматизации складского учета на предприятии малого бизнеса // Вестник Пермского государственного гуманитарно-педагогического университета. Серия: Информационные компьютерные технологии в образовании. 2019. №15. URL: <https://cyberleninka.ru/article/n/komponenty-erp-sistemy-dlya-avtomatizatsii-skladskogo-ucheta-na-predpriyatii-malogo-biznesa> (дата обращения: 08.10.2025).

25. Ложкин Н. Д. Автоматизация и учет на складе готовой продукции // Colloquium-journal. 2021. №12 (99). URL: <https://cyberleninka.ru/article/n/avtomatizatsiya-i-uchet-na-sklade-gotovoy-produktsii> (дата обращения: 05.07.2025).

26. Кирьянова Екатерина Андреевна, Серебрякова Татьяна Александровна. Оценка экономического эффекта внедрения информационной системы // Вестник науки. - 2018. -№2 (2). URL: <https://cyberleninka.ru/article/n/otsenka-ekonomicheskogo-effekta-vnedreniya-informatsionnoy-sistemy> (дата обращения: 06.12.2025).

27. Лиманова Наталия Игоревна, Селезнев Илья Александрович. Анализ эффективности клиент-серверной архитектуры // Бюллетень науки и практики. 2022. №7. URL: <https://cyberleninka.ru/article/n/analiz-effektivnosti-klient-servernoy-arhitektury> (дата обращения: 15.11.2025).

28. Трошков Сергей Николаевич Об опыте миграции приложений на свободно распространяемое программное обеспечение с открытым кодом // Вестник НГУ. Серия: Информационные технологии. -2018.- №2. URL: <https://cyberleninka.ru/article/n/ob-opyte-migratsii-prilozheniy-na-svobodno-rasprostranyaemoe-programmnoe-obespechenie-s-otkryтым-kodom> (дата обращения 20.10.2025)

29. Гарсиа-Молина, Гектор. Системы баз данных: Полный курс / / Гектор Гарсиа-Молина, Джеффри Д. Ульман, Дженнифер Уидом; [Пер. с англ. и ред. А.С. Варакина]. — Москва: Вильямс, 2003. — 1083 с. (дата обращения 26.11.2025).

30. Парикова Ирина Вячеславовна. Использование терминалов сбора данных при автоматизации складского учета // Скиф. 2020. №6 (46). [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/ispolzovanie-terminalov-sbora-dannyh-pri-avtomatizatsii-skladskogo-ucheta> (дата обращения: 24.10.2025).

Приложение 1

```
Public Class Dokyment
    Private Sub ПолучательToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ПолучательToolStripMenuItem.Click
        Polychatel.Show()
        Me.Hide()
    End Sub
EndClass
```

Приложение 2

```
PublicClassSotrydnik
    PrivateSubСотрудникиBindingNavigatorSaveItem_Click(ByValsenderAsSystem.Object, ByVal e As System.EventArgs)
        HandlesСотрудникиBindingNavigatorSaveItem.Click
        Me.Validate()
            Me.СотрудникиBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Ме.Складской_учетDataSet)
    End Sub
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the
        'Складской_учетDataSet.Сотрудники' table. You can move, or remove it, as needed.
        Me.СотрудникиTableAdapter.Fill(Ме.Складской_учетDataSet.Сотрудники)
    End Sub
    Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label1.Click
    End Sub
    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click
        СотрудникиBindingSource.MovePrevious()
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        СотрудникиBindingSource.MoveNext()
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        СотрудникиBindingSource.RemoveCurrent()
```

Продолжение приложения 2

End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click

СотрудникиBindingSource.AddNew()

End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button8.Click

Me.Validate()

Me.СотрудникиBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.Складской_учетDataSet)

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

Dokument.Show()

Me.Hide()

End Sub

End Class

Private Sub

Контактный_телефонMaskedTextBox_MaskInputRejected(ByVal sender As System.Object, ByVal e As System.Windows.Forms.MaskInputRejectedEventArgs) Handles Контактный_телефонMaskedTextBox.MaskInputRejected

End Sub

Приложение 3

```
Public Class Sklad
    Private Sub СкладBindingNavigatorSaveItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
СкладBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.СкладBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.Складской_учетDataSet)
    End Sub

    Private Sub Sklad_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the
'Складской_учетDataSet.СкладВиев' table. You can move, or remove it, as needed.
        Me.СкладВиевTableAdapter.Fill(Me.Складской_учетDataSet.СкладВиев)
        'TODO: This line of code loads data into the
'Складской_учетDataSet.Сотрудники' table. You can move, or remove it, as needed.
        Me.СотрудникиTableAdapter.Fill(Me.Складской_учетDataSet.Сотрудники)
        'TODO: This line of code loads data into the
'Складской_учетDataSet.Получатель' table. You can move, or remove it, as needed.
        Me.ПолучательTableAdapter.Fill(Me.Складской_учетDataSet.Получатель)
        'TODO: This line of code loads data into the
'Складской_учетDataSet.Отправитель' table. You can move, or remove it, as
needed.
        Me.ОтправительTableAdapter.Fill(Me.Складской_учетDataSet.Отправитель)
        'TODO: This line of code loads data into the
'Складской_учетDataSet.Товар' table. You can move, or remove it, as needed.
        Me.ТоварTableAdapter.Fill(Me.Складской_учетDataSet.Товар)
        'TODO: This line of code loads data into the
'Складской_учетDataSet.Склад' table. You can move, or remove it, as needed.
```

Продолжение приложения 3

```
Me.СкладTableAdapter.Fill(Me.Складской_учетDataSet.Склад)
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    СкладBindingSource.AddNew()
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    СкладBindingSource.RemoveCurrent()
End Sub

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
    СкладBindingSource.MovePrevious()
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    СкладBindingSource.MoveNext()
End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button8.Click
    Me.Validate()
    Me.СкладBindingSource.EndEdit()
    Me.TableAdapterManager.UpdateAll(Me.Складской_учетDataSet)
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dokument.Show()
    Me.Hide()
```

Продолжение приложения 3

End Sub

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
    СкладBindingSource.Filter = "[Кодтовара] = "" &  
    ComboBox5.SelectedValue.ToString() & "" & " and " & "[Кодполучателя] = "" &  
    ComboBox7.SelectedValue.ToString() & "" & " and " & "[Кодотправителя] = "" &  
    ComboBox6.SelectedValue.ToString() & ""
```

End Sub

```
Private Sub ComboBox5_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

End Sub

```
Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
```

End Sub

```
Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button7.Click
```

```
    СкладBindingSource.Filter = ""
```

End Sub

```
Private Sub СкладBindingNavigator_RefreshItems(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles СкладBindingNavigator.RefreshItems
```

End Sub

End Class

Приложение 4

```
SELECT TOP (100) PERCENT dbo.Склад.[Кодсклада],
dbo.Отправитель.Наименование AS Отправитель, dbo.Товар.Наименование,
dbo.Товар.[Кол-во],
dbo.Склад.[Дата поставки], dbo.Склад.[Дата отправки]
FROM dbo.Склад INNER JOIN
dbo.Товар ON dbo.Склад.[Код товара] = dbo.Товар.[Код товара] INNER
JOIN
dbo.Отправитель ON dbo.Склад.[Код отправителя] =
dbo.Отправитель.[Кол отправителя]
WHERE (dbo.Склад.[Дата отправки] > { fnNOW() })
```