

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФГБОУ ВО «Брянский государственный технический университет»

**Кафедра «Компьютерные технологии и системы»**

Направление подготовки

09.03.02- «Информационные системы и технологии»

Профиль – Информационные технологии и программные комплексы

## **ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ** **(научно-исследовательская работа)**

Выполнил:

Студент группы О-222-ИСТ-ВДМ

«\_\_\_\_» \_\_\_\_\_ 2023 г.

Руководитель практики от БГТУ:

к.т.н., доц. каф. «КТС»

Кзаков Юрий Михайлович

«\_\_\_\_» \_\_\_\_\_ 2023 г.

Руководитель от базы практики:

«\_\_\_\_» \_\_\_\_\_ 2023 г.

МП

Брянск 2023

# ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

(научно-исследовательская работа)

Студент: Костромин Алексей Сергеевич.

Направление подготовки: 09.03.02 – «Информационные системы и технологии»

Профиль: «Информационные технологии и программные комплексы».

Сроки прохождения практики: с 28.06.2023 по 11.07.2023

Место прохождения практики: ФГБОУ ВО «БГТУ»

**Цель прохождения практики:** расширение и закрепление теоретических и практических знаний, умений, навыков и компетенций, полученных в процессе обучения, а также формирование новых знаний, умений, навыков и компетенций профессиональной деятельности **будущих бакалавров сфере разработки и использования информационных систем и технологий.**

## Задачи:

- приобретение новых специализированных знаний, умений, навыков и компетенций, ориентированных на конкретное рабочее место;
- исследование процесса планирования деятельности объекта производственной практики;
- анализ информационного обеспечения управления объектом производственной практики;

СОГЛАСОВАНО

УТВЕРЖДАЮ

\_\_\_\_\_  
*Руководитель практики от профильной  
организации*

«\_\_» \_\_\_\_\_ 20\_\_ г.

\_\_\_\_\_  
*Руководитель практики от университета*

«\_\_» \_\_\_\_\_ 20\_\_ г.

Задание принято к исполнению: \_\_\_\_\_  
(подпись обучающегося)

«\_\_» \_\_\_\_\_ 202\_ г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
ФГБОУ ВО «Брянский государственный технический университет»

**Кафедра «Компьютерные технологии и системы»**

## **Д Н Е В Н И К**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**(научно-исследовательская работа)**

**СТУДЕНТА**

Направление подготовки

09.03.02- «Информационные системы и технологии»

Профиль – Информационные технологии и программные комплексы

Брянск 2023г.

Студент

(Ф.И.О.)

Курс **3**

направление подготовки

**09.03.02** – «Информационные системы и технологии»

направляется  
на (в)

«

»

(предприятие, организация, фирма, компания)

### 1. Календарные сроки практики

По учебному плану начало 28.06.2023 конец 11.07.2023

Дата прибытия на практику 28.06 2023 г.

Дата выбытия с места практики 11.07 2023 г.

### 2. Руководитель практики от БГТУ

Кафедра «Компьютерные технологии и системы»

Ученое звание Кандидат технических наук, доцент

Фамилия Казаков

Имя Юрий

Отчество Михайлович

### 1. Руководитель практики от организации

Должность \_\_\_\_\_  
\_\_\_\_\_

Фамилия Казаков

Имя Юрий

Отчество Михайлович

Дата	Описание работы, выполненной студентом	Отметка руководителя от базы практики
1	2	3
	Знакомство с программой практики	Выполнено
	Изучение предметной области.	Выполнено
	Формулировка задания на прохождение производственной практики.	Выполнено
	Анализ проектных решений на предприятии.	Выполнено
	Выполнение задания производственной практики	Выполнено
	Подготовка материалов для отчетного семинара, оформление отчета по практике.	Выполнено
	Промежуточная аттестация (дифференцированный зачет)	Выполнено

Руководитель практики  
от кафедры (подпись) / (И.О.Фамилия)

Руководитель практики от  
организации (подпись) / (И.О.Фамилия)

### График прохождения практики (календарно-тематический план)

№	Этапы	Выполняемая работа	Продолжительность
1	Знакомство с программой практики	Знакомство с режимом работы, прохождение вводного инструктажа и получение информации и целях и содержании производственной практики	2
2	Изучение предметной области.	Изучение современных литературных источников и анализ деятельности организации	18
3	Формулировка задания на прохождение производственной практики..	Оформление и согласование задания на прохождение практики	6
4	Анализ проектных решений на предприятии..	Изучение основных проектных решений в сфере информатизации на предприятии.	10
5	Выполнение задания производственной практики	Обсуждение структуры и отдельных этапов информатизации в организации	55
6	Подготовка материалов для отчетного семинара, оформление отчета по практике.	Сбор материалов для отчета, подготовка к отчетному семинару, обсуждение отчета с руководителем.	15
7	Промежуточная аттестация (дифференцированный зачет)	Аудиторное представление отчета	2
	Итого		108

Руководитель практики от кафедры \_\_\_\_\_

(Подпись)

(ФИО)

Руководитель практики от предприятия \_\_\_\_\_

## Характеристика руководителя производственной практики о работе студента

Студент Брянского государственного технического университета Костромин Алексей Сергеевич проходил производственную практику с \_\_\_28.06.2023\_\_\_ года по \_\_\_11.07.2023\_\_\_ года в ФГБОУ ВО «Брянский государственный инженерно-технологический университет».

Костромин Алексей Сергеевич выполняла программу практики полно и качественно, добросовестно относилась к работе, проявляла инициативность и творческую активность.

Костромин Алексей Сергеевич продемонстрировал способность выполнять индивидуальные задания, предусмотренные программами практики;

■ соблюдение правила внутреннего трудового распорядка;

■ соблюдение требования охраны труда и пожар.

В процессе прохождения практики он проявил активность в овладении производственными навыками, трудовую дисциплину.

В целом теоретический уровень подготовки студента и качество выполняемой им работы можно оценить на «\_\_\_\_\_».

Руководитель практики  
предприятия

(подпись)

/ (И.О.Фамилия)

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	10
1. Аналитический раздел.....	12
1.1. Анализ предметной области и формирование требований к разрабатываемому ПО.....	12
1.2. Анализ среды разработки.....	13
2. Проектный раздел.....	15
2.1. Проектирование.....	15
2.3. Создание серверной части базы данных.....	18
2.4. Разработка клиентского приложения.....	19
2.5. Тестирование программного обеспечения.....	25
ЗАКЛЮЧЕНИЕ.....	27
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	28
ПРИЛОЖЕНИЯ.....	29

В наше время наблюдается активное развитие информационных технологий, а также значительный рост информационных потребностей, информационного обеспечения во все сферы человеческой деятельности. С каждым днём все труднее себе представить сколько-нибудь значимую информационную систему, которая не имела бы в качестве основы или важной составляющей базу данных. Из-за увеличения объёма информации и необходимостью её преобразования, растёт потребность организаций для осуществления существующих процессов обработки данных. К таким процессам можно отнести анализ, систематизацию, отображение и редактирование данных.

Актуальность выбора темы работы состоит в том, что информационная система для строительной компании поможет сэкономить много рабочего времени благодаря автоматизации некоторых процессов. Также, увеличится сохранность данных, так как вся информация будет храниться в едином месте.

Целью производственной практики является разработка информационной системы для автоматизации работы строительной компании по продаже стройматериалов.

Объектом исследования является процесс разработки приложений средствами Visual Studio 2022.

Предметом исследования производственной практики является возможности Visual Studio 2022 C#, для разработки интерфейса приложения и организации доступа к данным.

Для достижения цели, пост, необходимо решить следующие задачи:

1. собрать и проанализировать материалы предметной области и сформировать требования к создаваемому ПО;
2. обосновать выбор языка и инструментальных средств разработки;
3. произвести проектирование инфологической модели;
4. создать базу данных в Microsoft SQL Server 2019;

5. создать клиентское приложение в среде программирования Visual Studio 2022, то есть разработать интерфейс и осуществить доступ к данным.

Вид исследования-прикладной. Исследование будет осуществлено на теоретико-эмпирическом уровне, так как он предполагает применение на практике полученных теоретических знаний. В ходе исследования будут применяться следующие методы: наблюдение, анализ, синтез, моделирование, классификация.

## **1. АНАЛИТИЧЕСКИЙ РАЗДЕЛ**

### **1.1. Анализ предметной области и формирование требований к разрабатываемому ПО**

Работа с документами охватывает все процессы, которые относятся к фиксации на всевозможных носителях и оформлению по установленным правилам информации, важной для работы при оформлении заявок. Каждая организация, которая ведет функциональную работу, имеет потребность в автоматизации и упрощении протекающих информационных процессов.

Под автоматизированной системой понимается система способов и методов сбора, накопления, хранения, поиска, обработки и защиты управленческой информации на базе внедрения развитого ПО, средств вычислительной техники и связи, также методов, при помощи которых такие сведения предоставляются пользователям.

Главным достоинством автоматизации является уменьшение избыточности хранимых данных, экономии используемой памяти, сокращение расходов на неоднократные операции обновления лишних копий и уничтожение возможности появления противоречий по причине хранения в различных местах сведений об одном и том же объекте. Нарастивание степени достоверности информации и наращивание скорости ее обработки. Важным моментом является сокращение временных интервалов для осуществления поиска информации, который осуществляется с помощью экранных форм, в которых указываются нужные параметры поиска информации или какого-либо объекта.

ООО «СтройМирЛаб», в первую очередь, особая организация, занимающаяся основным видом деятельности – это торговлей строительными материалами и изделиями. Некоторые виды строительных материалов могут продаваться по другой цене, то есть по запросу клиента.

Информационная система «СтройМирЛаб» должна содержать информацию о продаваемых строительных материалах, клиентах

заказывающие строительные материалы, работающих сотрудников в ООО «СтройМирЛаб».

Программное обеспечение должно соответствовать следующим требованиям:

1. надежность и безопасность;
2. необходимо предусмотреть защиту от несанкционированного доступа к данным, ввода данных, их удаления;
3. предусмотреть обновление данных. Под операцией обновления понимают добавление, удаление и редактирование данных.

При разработке программного обеспечения необходимо обеспечить такие технологические процессы как:

1. появление нового товара;
2. появление нового сотрудника;
3. появление нового клиента;
4. появление нового заказа;
5. поиск по набору данных;
6. фильтрация по набору данных;
7. создание отчёта по оформленным заказам.

Для создания полноценного программного продукта, в процессе разработки необходимо выполнить все вышеперечисленные требования.

## **1.2. Анализ среды разработки**

Для создания диаграммы вариантов использования будет использована программа StarUML, а также проектирование инфологической модели в MySQL Workbench.

StarUML - это программный инструмент моделирования, который поддерживает UML (Интегрированный язык моделирования). StarUML поддерживает одиннадцать различных типов диаграмм, принятых в нотации UML 2. 0. Применение StarUML, одного из популярнейших программных инструментов моделирования, обеспечит свершение максимальной

производительности и качества программных проектов [10].

Для проектирования инфологической модели будет использована программа MySQL Workbench.

MySQL Workbench – мощный визуальный инструмент для администраторов баз данных, архитекторов баз данных и разработчиков MySQL. Этот инструмент предлагает моделирование данных, разработку SQL и инструменты администрирования для настройки и администрирования сервера.

MySQL Workbench позволяет создавать модели и управлять ими, преобразовывать динамическую базу данных в модель, а также создавать и редактировать таблицы и вставлять данные.

В ходе реализации поставленных задач разработка базы данных будет проводиться в Microsoft SQL Server 2019, а программное обеспечение в Visual Studio 2022 C#.

SQL Server – это программа, которая предназначена для хранения базы данных и обеспечения доступа к этим данным из других программ. В SQL Server включена поддержка языка XML и протокола HTTP, средства увеличения быстродействия и доступности, позволяющие распределить нагрузку и обеспечить непрерывную работу [1, с.271].

Microsoft SQL Server 2019 самая известная коммерческая СУБД. Она привязана к Windows, но это плюс, если пользоваться продуктами Microsoft. Влияет от платформы. И графический интерфейс, и программное обеспечение построены на командах. Поддерживает SQL, непроцедурные, нечувствительные к регистру и общие языки баз данных.

Для решения поставленной задачи необходимо использовать функциональную, эффективную и удобную платформу для разработки. В качестве такой платформы была выбрана среда .NET.

Среда разработки Visual Studio, поставляемая вместе с .NET, предоставляет необходимый инструментарий для эффективного и быстрого создания приложений с графическим интерфейсом [5, с.408].

Язык программирования C# призван реализовать компонентно-ориентированный подход к программированию, который способствует меньшей машинно-архитектурной зависимости результирующего программного кода, большей гибкости, переносимости и легкости повторного использования программ [2, с. 436].

Исходя из вышеперечисленного можно приступать к проектированию.

## **2. ПРОЕКТНЫЙ РАЗДЕЛ**

### **2.1. Проектирование**

Инфологическая модель представляет собой описание будущей базы данных, представленное с помощью естественного языка, формул, графиков, диаграмм, таблиц и других средств, понятных как разработчикам БД, так и обычным пользователям. Инфологическая модель создается по результатам проведения исследований предметной области. MySQL Workbench - средство разработки инфологической модели баз данных, который сочетает в себе графический интерфейс Windows, инструменты для построения ER-диаграмм, редакторы для создания логического и физического описания модели данных и прозрачную поддержку ведущих реляционных СУБД и настольных баз данных.

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных.

Существует множество подходов к построению моделей: Графовые модели, семантические сети, модель «сущность-связь» и т.д. Наиболее популярной из них оказалась модель «сущность-связь» или называемая ещё ER-моделью (от англ. Entity-Relationship, т.е. сущность-связь) [1, с.271].

Связи делятся на три типа по множественности: один-ко-одному (1:1), один-ко-многим (1: M), многие-ко-многим (M: M).

Связь один-ко-одному означает, что экземпляр одной сущности связан только с одним экземпляром другой сущности.

Связь один-ко-многим (1: M) означает, что один экземпляр сущности, расположенный слева по связи, может быть связан с несколькими экземплярами сущности, расположенными справа по связи.

Связь многие-ко-многим (M: M) означает, что несколько экземпляров первой сущности могут быть связаны с несколькими экземплярами второй сущности, и наоборот. Между двумя сущностями может быть задано сколько угодно связей с разными смысловыми нагрузками.

В ходе анализа предметной области были выделены сущности, которые нужны для создания инфологической модели: сотрудник, продукция, клиент, стоимость за расстояние, заказы. Просмотрев все сущности, можно приступать к проектированию ER-модели.

Для создания базы данных «СтройМирЛаб» была разработана ER-модель «СтройМирЛаб», данная модель представлена на рисунке Рисунок 1.

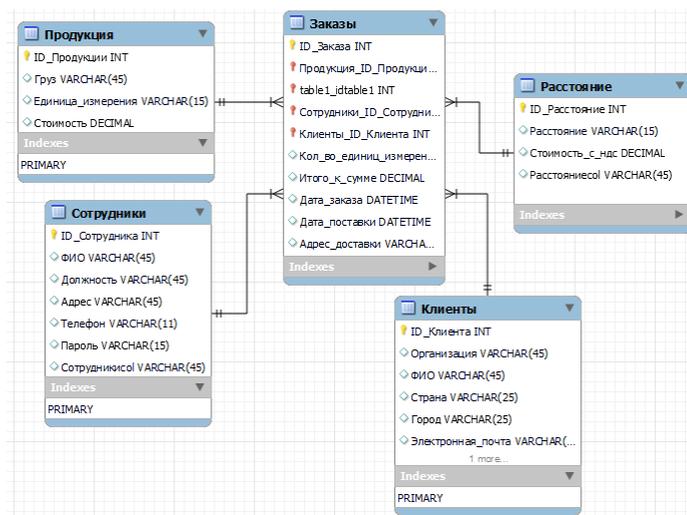


Рисунок 1 – ER-модель «СтройМирЛаб»

Для улучшения понимания подхода к разработке и выбору средств проектирования информационной системы создается описание вариантов использования, представляемое в виде UML-диаграммы вариантов использования и диаграммы деятельности.

Вариант использования — показывает поведение системы при взаимодействии ее с объектами или субъектами, находящимися за пределами данной системы, в нашем случае это будет пользователь приложения:

сотрудник [10]. На рисунке Рисунок 2 изображена диаграмма вариантов использования в нотации UML.



Рисунок 2 – Диаграмма вариантов использования

После выполненного анализа предметной области, в рамках которого были обозначены требования к создаваемой базе данных и программного обеспечения для организации работы с ней, а также после выбора инструментальных средств разработки и проектирования, можно приступить к разработке программного обеспечения.

## 2.2.

### 2.3. Создание серверной части базы данных

Для создания архитектуры базы данных, позволяющей реализовать отображение внешних баз данных различных форматов, используется серверная система управления базами данных. Microsoft SQL Server – это система управления реляционными базами данных, предоставляющая широкий диапазон средств для хранения информации и эффективного управления ею.

С помощью MySQL Workbench был создан SQL скрипт и экспортирован в СУБД Microsoft SQL Server Management Studio.

Вся информация в базе данных храниться в таблицах, которые являются отображением некоторых логических сущностей. База данных «СтройМирЛаб» содержит информацию о заказах, клиентах, сотрудников и товаре. В процессе проектирования были выделены следующие таблицы: «Заказы», «Клиенты», «Продукция», «Сотрудники», «Стоимость\_за\_расстояние». На рисунке Рисунок 3 представлена структура базы данных, реализованная с помощью Microsoft SQL Server Management Studio.

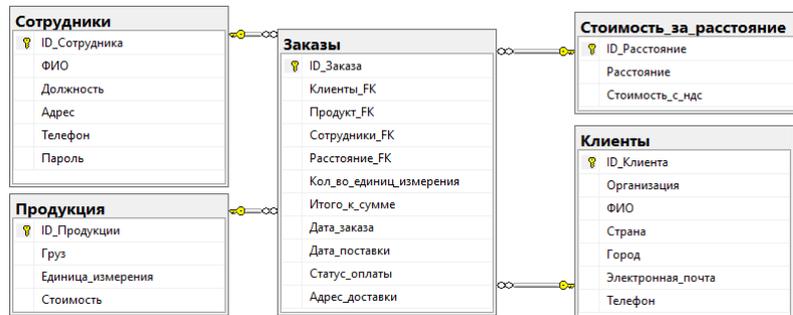


Рисунок 3 – Структура базы данных «СтройМирЛаб»

Связи между таблицами базы данных предоставляют возможность пользователю одновременно использовать данные, хранящиеся в разных таблицах. При создании связей между таблицами не обязательно, чтобы общие поля имели одинаковые имена, но общие поля должны иметь одинаковый тип данных.

После создания серверной части базы данных «СтройМирЛаб» можно

приступать к разработке клиентского приложения.

## **2.4. Разработка клиентского приложения**

Для разработки клиентского приложения используется среда визуального программирования Visual Studio 2022 на языке программирования C#.

База данных «СтройМирЛаб» была спроектирована в MySQL Workbench и экспортирована в СУБД Microsoft SQL Server Management Studio.

Среда визуального программирования Visual Studio 2022 C# для создания приложения предоставляет классы для работы с данным СУБД такие как: SqlConnection, DataTable, SqlDataAdapter, DataSet. В качестве соединения с Microsoft SQL Server Management Studio был выбран поставщик данных .NET Framework для SQL Server [8, с.896].

Проект программы содержит восемь форм:

1. «Вход в систему» – окно для входа на главную форму с помощью учётной записи.
2. «СтройМирЛаб» – главная форма, которая позволяет переходить на другие формы.
3. «Наша продукция» – форма, показывает информацию о продукции предприятия.
4. «Оформление заказа» – данная форма позволяет оформлять новые заказы и добавлять новых клиентов.
5. «Заказы» – форма, показывает информацию о составленных заказах;
6. «Сотрудники» – форма. показывает информацию о сотрудниках работающие в предприятии «СтройМирЛаб».
7. «Диаграммы» – окно предоставляющая пользователю просматривать диаграммы и сохранить их в нужное место.
8. «Отчёт» – данное окно нужно для просмотра сформированного отчёта с возможностью печати.

Для программного обеспечения была создана главная форма «СтройМирЛаб», предназначенная для навигации к другим формам.

Элементами главной формы являются картинки с надписями, открывающие прочие формы для решения иных задач, а также кнопка для закрытия главной формы. Например, при нажатии на картинку «Наша продукция» открывается форма «Наша продукция», с помощью которой в базу данных вносится и редактируется информация о продукции предприятия «СтройМирЛаб».

Главная форма представлена на рисунке Рисунок 4.



Рисунок 4 – Макет главной формы «СтройМирЛаб»

С главной формы «СтройМирЛаб» можно с помощью нажатия по картинке перейти на другие формы. Другие формы используются для просмотра и редактирования данных, так как они предоставляют более удобный способ работы над данными в таблицах. На рисунке Рисунок 5 представлена форма «Наша продукция», которая предназначена для просмотра, ввода, изменения, удаления, поиска и фильтрации данных.

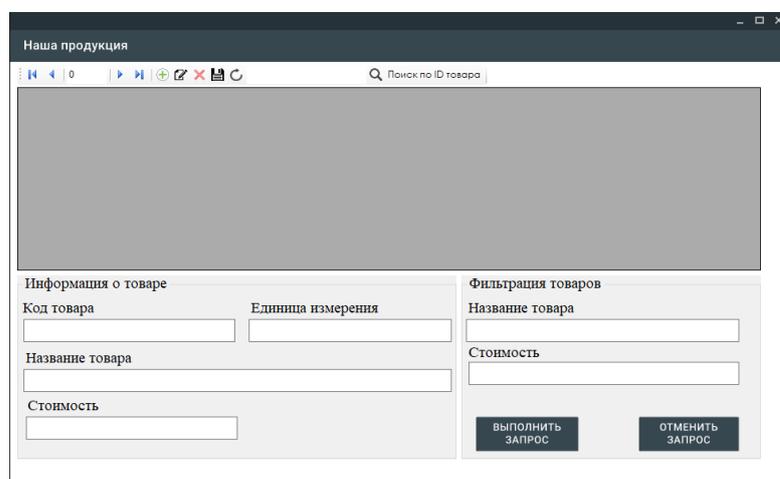


Рисунок 5 – Макет формы «Наша продукция»

Визуальные компоненты формы «Наша продукция»:

1. Label – графический элемент управления, предназначенный для отображения текста на форме.
2. TextBox – текстовое поле, которое предназначено для ввода и редактирования текста.
3. Button – это дочерний элемент управления, помещенный в Форму и используемый для обработки события click.
4. GroupBox – это контейнер, содержащий несколько элементов управления, которые связаны друг с другом.
5. BindingNavigator – элемент управления для перемещения по набору данных.

На форме «Наша продукция» содержатся элементы управления, при помощи которых осуществляется доступ к данным в таблице. Элементами управления являются текстовые поля для ввода и редактирования данных, переключатели в элементе управления, кнопки и т.д. В окне представлены кнопки для разных целей: для перехода по записям (предыдущая запись, следующая запись, первая запись, последняя запись), для работы с записями (удалить, добавить, редактировать, сохранить, поиск, фильтрация).

Для оформления нового заказа используется форма «Оформление заказа», представленная на рисунке Рисунок 6. Она содержит новые визуальные компоненты TabControl и ComboBox. TabControl – позволяет создать элемент управления с несколькими вкладками. ComboBox позволяет пользователю выбрать элемент из раскрывающегося списка или при необходимости ввести новый текст в текстовое поле элемента управления.

Данная форма позволяет сотруднику выбрать на первой странице TabControl выбрать клиента, а на второй уже оформлять сам заказ.

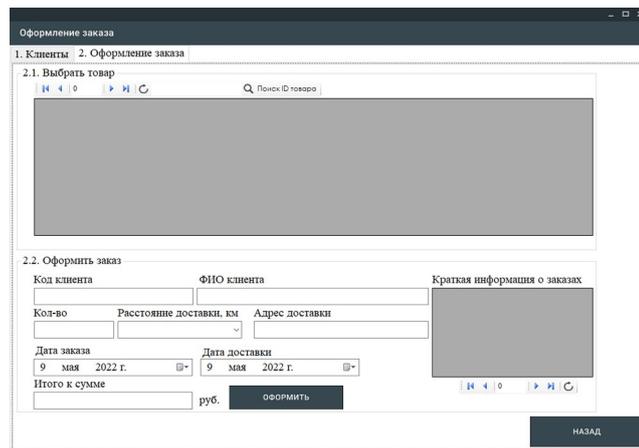


Рисунок 6 – Макет формы «Оформление заказа»

Для просмотра и внесения информации о клиенте используется форма «Оформление заказа», представленная на рисунке Рисунок 7. Помимо оформления заказа с помощью элемента TabControl можно перейти к выбору клиента и там присутствуют элементы предназначены для просмотра, ввода, изменения, удаления, поиска и фильтрации данных. Она содержит те же самые элементы управления, что и форма «Наша продукция».

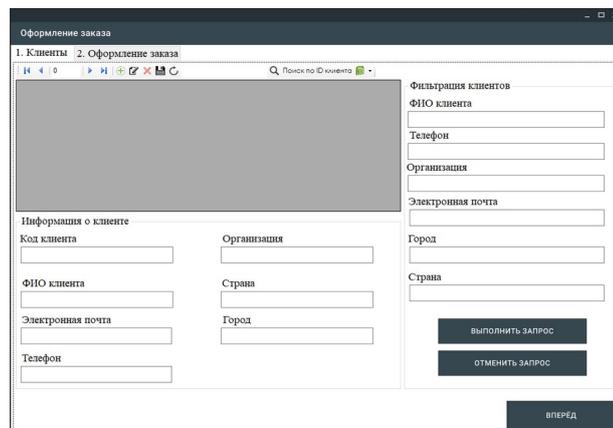


Рисунок 7 – Макет формы «Оформление заказа»

Форма «Сотрудники» предоставляет информацию о сотрудниках, работающих в предприятии «СтройМирЛаб». Также форма позволяет просматривать и вносить информацию о сотрудниках. Данная форма тоже содержит те же самые элементы управления, что и форма «Наша продукция», но при этом был добавлен способ редактирования пароля. Форма «Сотрудники» показана на рисунке Рисунок 8.

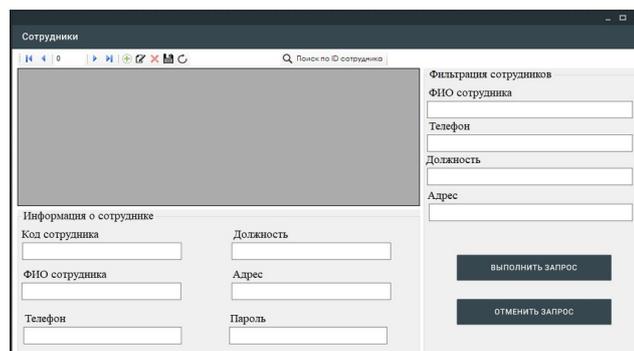


Рисунок 8 – Макет формы «Сотрудники»

Все оформленные заказы отображены на форме «Заказы», представленной на рисунке Рисунок 9. Она предназначена для просмотра, редактирования, поиска данных, а также организации фильтрации по параметрам: ФИО клиента, название груза, организация, итого к сумме, дата заказа, дата поставки. Форма «Заказы» содержит новый визуальный компонент DateTimePicker – представляет раскрывающийся по нажатию календарь, в котором можно выбрать дату.

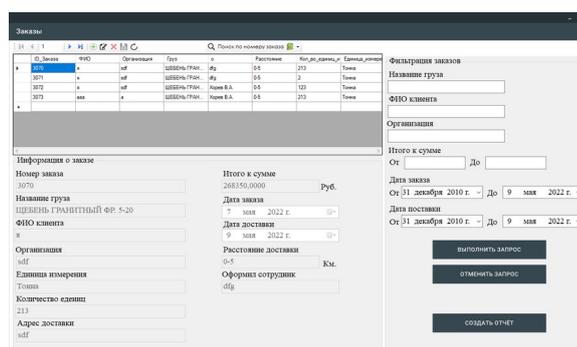


Рисунок 9 – Макет формы «Оформление заказа»

Для того, чтобы на запрос сотрудника был дан ответ, в запросе должен быть указан хотя бы один параметр. В программе реализована возможность формирования отчёта по оформленным заказам. Отчёт показан на рисунке Рисунок 10.

Окно «Диаграммы» предоставляет пользователю три вида диаграмм:

1. Движение выручки – данная диаграмма показывает движение выручки в период времени.
2. Статистика продаж сотрудников – показывает в процентном соотношении продажи сотрудников в период времени.

### 3. Статистика покупок клиентов – аналогична предыдущей диаграмме.

ID Заказа	Клиент	Организация	Груз	Кол-во	Единица измерения	Дата заказа	Дата поставки	Адрес доставки	Итого к сумме
3070 я	sdf	ЩЕБЕНЬ ГРАНИТНЫЙ ФР. 5-20		213	Тонна	07.05.2022 0:00:00	09.05.2022 0:00:00	sdf	268350,0000
3071 я	sdf	ЩЕБЕНЬ ГРАНИТНЫЙ ФР. 5-20		2	Тонна	01.05.2022 0:00:00	09.05.2022 0:00:00	sdf	4600,0000
3072 я	sdf	ЩЕБЕНЬ ГРАНИТНЫЙ ФР. 5-20		123	Тонна	09.05.2022 0:00:00	09.05.2022 0:00:00	sdsf	155850,0000
3073 aaa	a	ЩЕБЕНЬ ГРАНИТНЫЙ ФР. 5-20		213	Тонна	09.05.2022 0:00:00	09.05.2022 0:00:00	выпв	268563,0000

Рисунок 10 – Отчёт по заказам

Для реализации создания диаграмм был использован Chart – элемент является панелью, на которой можно создавать диаграммы и графики различных типов. Компонент Chart управляет координатными осями и формой отображения. Одна из диаграмм показана на рисунке Рисунок 11.

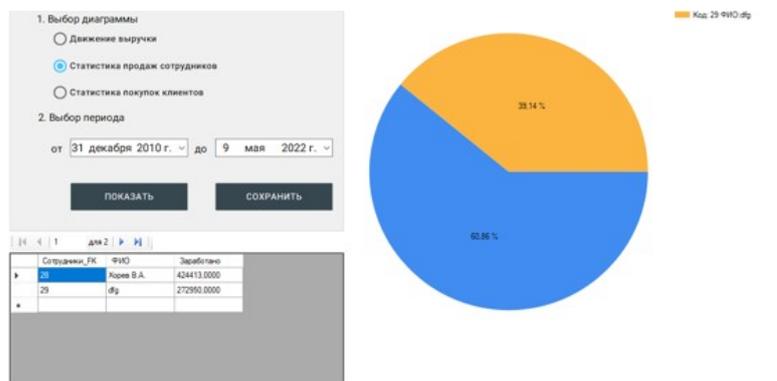


Рисунок 11 – Диаграмма «Статистика продаж сотрудников»

В результате проделанной работы был проведён анализ предметной области, составлено техническое задание, на основе чего были сформулированы требования к разрабатываемому программному обеспечению.

Спроектирована ER-модель при помощи MySQL Workbench. После инфологического проектирования была спроектирована диаграмма вариантов использования, при помощи программы StarUML. Далее была разработана структура базы данных в программной среде Microsoft SQL Server Management Studio. Были созданы и заполнены таблицы, отражающие информацию, необходимую для работы с базой данных.

При помощи Visual Studio 2022 были реализованы по сформулированным требованиям экранные формы и осуществлён доступ к данным с помощью

удобного клиентского приложения. Теперь нужно переходить к тестированию ПО.

## **2.5. Тестирование программного обеспечения**

Прежде чем приступать к тестированию клиентского приложения нужно понять, что понимается под тестированием программного продукта.

Тестирование программного продукта – это процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определенным образом.

Тестирование чёрного ящика или поведенческое тестирование-стратегия (метод) тестирования функционального поведения объекта (программы, системы) с точки зрения внешнего мира, при котором не используется знание о внутреннем устройстве тестируемого объекта. Под стратегией понимаются систематические методы отбора и создания тестов для тестового набора. Стратегия поведенческого теста исходит из технических требований и их спецификаций [12].

Тестирование по стратегии белого ящика-тестирование кода на предмет логики работы программы и корректности её работы с точки зрения компилятора того языка, на котором она писалась. Тестирование по стратегии белого ящика, также называемое техникой тестирования, управляемой логикой программы, позволяет проверить внутреннюю структуру программы. Исходя из этой стратегии тестировщик получает тестовые данные путем анализа логики работы программы.

В ходе тестирования клиентского приложения была выявлена ошибка, которая возникала при удалении пустой строки в таблице. Ошибка показана на рисунке Рисунок 12.



Рисунок 12 – Ошибка при удалении пустой строки

После изменения программного кода повторное тестирование не выявило каких-либо ошибок, которые бы приводили к неправильности работы клиентского приложения.

Итогом является готовый программный продукт – программное обеспечение для автоматизации процедуры оформления заказов клиентов на выполнение работ и оказание услуг для предприятия «СтройМирЛаб». База данных «СтройМирЛаб» позволит решить вопрос повышения производительности предприятия путем минимизации времени обработки информации о заказах за счет автоматизации данного процесса. Данное программное обеспечение позволит редактировать, просматривать, хранить и удалять данные о продукции, клиентах, сотрудниках и заказах.

## ЗАКЛЮЧЕНИЕ

В итоге проделанной работы был создан готовый программный продукт – база данных для предприятия «СтройМирЛаб» и клиентское приложение для доступа к данным, тем самым реализовав поставленную цель

В ходе разработки программного обеспечения были решены следующие задачи:

1. собрать и проанализировать материалы предметной области и сформировать требования к создаваемому ПО;
2. обосновать выбор языка и инструментальных средств разработки;
3. произвести проектирование инфологической модели;
4. создать базу данных в Microsoft SQL Server 2019;
6. создать клиентское приложение в среде программирования Visual Studio 2022, то есть разработать интерфейс и осуществить доступ к данным.

Проведено тестирование, в ходе которого была выявлена ошибка при удалении пустой строки в таблицах, которая в последствие была устранена. Программа успешно справляется со своей задачей, не требуя при этом большого количества ресурсов системы и не нарушая работы операционной системы.

Интерфейс разработанного клиентского приложения был подобран под стилистику Windows, влияющая на конструктивные решения, и придающая приложению ощущения целостности, а также не вызывает какого-либо раздражения глаз. Встроенные календари являются удобным механизмом для ввода дат, исключают ошибки ввода данных другого формата.

Разработанное программное обеспечение предназначено для решения такой задачи как сокращение времени, затрачиваемого на оформление заказа клиентов.

Данное программное обеспечение позволит вносить, редактировать, просматривать, хранить и удалять информацию, выполнять поиск и фильтрацию по выбранным критериям.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Агальцов В.П. Базы данных. В 2-х кн. Кн. 2. Распределенные и удаленные базы данных: Учебник / В.П. Агальцов. М.: ИД ФОРУМ: НИЦ Инфра-М, 2019. – 271 с.
2. Абрамян М. В. Visual C# на примерах. – М.: БХВ-Петербург, 2016. – 436 с.
3. Бьянкуцци, Ф. Пионеры программирования. Диалоги с создателями наиболее популярных языков программирования // Ф. Бьянкуцци, Ш. Уорден. – СПб.: Символ-плюс, 2018. – 608 с.
4. Вагнер, Б. C# Эффективное программирование // Билл Вагнер. – М.: ЛОРИ, 2017. – 320 с.
5. Гергель, В.П. Современные языки и технологии параллельного программирования: Учебник / В.П. Гергель. – М.: МГУ, 2019. – 408 с.
6. Ишкова, Э. А. Самоучитель C#. Начала программирования / Э.А. Ишкова. – М.: Наука и техника, 2018. – 496 с.
7. Колесников А. П. Методы численного анализа, изложенные на языке формул и алгоритмическом языке C#. – М.: Высшая школа, 2017. – 414 с.
8. Рихтер, Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#. – СПб: Питер, 2018. – 896 с.
9. Хейлсберг, А. Язык программирования C#. Классика Computers Science / А. Хейлсберг, М. Торгерсен, С. Вилтамут. – СПб.: Питер, 2018. – 784 с.
10. Диаграмма вариантов использования: [Электронный ресурс]. URL: <https://www.sites.google.com/site/metodsybd/blok-4-sredstvaproektirovania/4-1-diagramma-variantov-ispolzovania.html> (дата обращения: 03.04.2022).
11. Алгоритмы редактирования: [Электронный ресурс]. URL: <https://robotlandia.ru/abct2/0206add1.html> (дата обращения: 15.04.2022).
12. Методы и способы устранения ошибок: [Электронный ресурс]. URL: [https://studwood.ru/1586509/tovarovedenie/metody\\_sposoby\\_ustraneniya\\_oshibok.html](https://studwood.ru/1586509/tovarovedenie/metody_sposoby_ustraneniya_oshibok.html) (дата обращения: 6.05.2022).

# ПРИЛОЖЕНИЯ

## Приложение 1

### Листинг программы

#### Orders.cs

```
using MaterialSkin.Controls;
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 СтройМирЛаб
{
    public partial class Orders : MaterialForm
    {
        public string source =
            @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\
СтройМирЛаб.mdf;Integrated Security=True;Connect Timeout=30";

        public string query = "SELECT dbo.Заказы.ID_Заказа, dbo.Клиенты.ФИО,
dbo.Клиенты.Организация, dbo.Продукция.Груз, dbo.Сотрудники.ФИО AS о,
dbo.Стоимость_за_расстояние.Расстояние, dbo.Заказы.Кол_во_единиц_измерения,
dbo.Продукция.Единица_измерения, dbo.Заказы.Итого_к_сумме, dbo.Заказы.Дата_заказа,
dbo.Заказы.Дата_поставки, dbo.Заказы.Статус_оплаты, dbo.Заказы.Адрес_доставки FROM
dbo.Заказы INNER JOIN dbo.Клиенты ON dbo.Заказы.Клиенты_FK =
dbo.Клиенты.ID_Клиента INNER JOIN dbo.Продукция ON dbo.Заказы.Продукт_FK =
dbo.Продукция.ID_Продукции INNER JOIN dbo.Сотрудники ON dbo.Заказы.Сотрудники_FK
= dbo.Сотрудники.ID_Сотрудника INNER JOIN dbo.Стоимость_за_расстояние ON
dbo.Заказы.Расстояние_FK = dbo.Стоимость_за_расстояние.ID_Расстояние";

        bool add_client, edit_client, delete_client;
        public string Id_Client;
        public int rowIndexClient;
        public int deleterowindex;
        private SqlConnection sqlConnection = null;
        private SqlCommandBuilder sqlBuilder = null;
        private SqlDataAdapter sqlDataAdapter1 = null;
        private SqlDataAdapter sqlDataAdapter2 = null;

        private DataSet dataSet2 = null;

        private BindingSource bindingSource = null;
```

```

public Orders(int id_sotr)
{
    InitializeComponent();
    this.id_sotr = id_sotr;
}
public int id_sotr;

private void LoadData()
{
    try
    {

        sqlDataAdapter1 = new SqlDataAdapter("select * from Заказы", sqlConnection);
        sqlDataAdapter2 = new SqlDataAdapter(query, sqlConnection);

        sqlBuilder = new SqlCommandBuilder(sqlDataAdapter1);

        sqlBuilder.GetInsertCommand();
        sqlBuilder.GetUpdateCommand();
        sqlBuilder.GetDeleteCommand();

        dataSet2 = new DataSet();

        sqlDataAdapter2.Fill(dataSet2, "Заказы");

        bindingSource = new BindingSource();

        bindingSource.DataSource = dataSet2.Tables["Заказы"];

        clientBindingNavigator.BindingSource = bindingSource;
        dataGridView1.DataSource = bindingSource;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

private void ReloadData()
{
    try
    {

        int rowindex = rowIndexClient;

        dataSet2.Tables["Заказы"].Clear();
    }
}

```

```

sqlDataAdapter2.Fill(dataSet2, "Заказы");

bindingSource.DataSource = dataSet2.Tables["Заказы"];

clientBindingNavigator.BindingSource = bindingSource;
dataGridView1.DataSource = bindingSource;

//add , edit bool

if (delete_client)
{
    delete_client = false;
    if (dataGridView1.Rows.Count - 1 != rowindex) dataGridView1.CurrentCell =
dataGridView1.Rows[rowindex].Cells[0];
    else if (dataGridView1.Rows.Count > 1) dataGridView1.CurrentCell =
dataGridView1.Rows[dataGridView1.Rows.Count - 2].Cells[0];

}

UpdataTextBox();

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButtons.OK,
MessageBoxIcon.Error);

}

}
private void UpdataTextBox()
{

int rowindex = -1;
if (dataGridView1.CurrentRow != null) rowindex = dataGridView1.CurrentCell.RowIndex;

if (rowindex >= 0)
{
    DataGridViewRow row = dataGridView1.Rows[rowindex];
    textBox1.Text = row.Cells[0].Value.ToString();
    textBox4.Text = row.Cells[1].Value.ToString();
    textBox3.Text = row.Cells[2].Value.ToString();
    textBox2.Text = row.Cells[3].Value.ToString();
    textBox14.Text = row.Cells[4].Value.ToString();
    textBox13.Text = row.Cells[5].Value.ToString();
    textBox6.Text = row.Cells[6].Value.ToString();
    textBox5.Text = row.Cells[7].Value.ToString();
    textBox12.Text = row.Cells[8].Value.ToString();
    dateTimePicker1.Text = row.Cells[9].Value.ToString();
}
}

```

```

        dateTimePicker2.Text = row.Cells[10].Value.ToString();
        textBox11.Text = row.Cells[12].Value.ToString();

    }
    else
    {
        textBox1.Text = null;
        textBox4.Text = null;
        textBox3.Text = null;
        textBox2.Text = null;
        textBox14.Text = null;
        textBox13.Text = null;
        textBox6.Text = null;
        textBox5.Text = null;
        textBox12.Text = null;
        dateTimePicker1.Text = null;
        dateTimePicker2.Text = null;
        textBox11.Text = null;

    }

}

private void dataGridView1_CellStateChanged_1(object sender,
DataGridViewCellStateChangedEventArgs e)
{

    UpdateTextBox();

}

private void Orders_Shown(object sender, EventArgs e)
{
    this.MaximizeBox = false;

}

private void Orders_Load(object sender, EventArgs e)
{
    sqlConnection = new SqlConnection(source);
    sqlConnection.Open();
    LoadData();
    clientBindingNavigatorSaveItem.Enabled = false;
    DateTime date2000 = new DateTime(2010, 12, 31, 0, 0, 0);
    dateTimePicker3.Value = date2000;
    dateTimePicker6.Value = date2000;

}

private void materialRaisedButton1_Click(object sender, EventArgs e)
{

```

```

        string filtering = query + " and Дата_заказа between " + "\"" +
dateTimePicker3.Value.ToString("yyyy/MM/dd") + "\" +
        " and " + "\"" + dateTimePicker4.Value.ToString("yyyy/MM/dd") + "\" +
        " and Дата_поставки between " + "\"" +
dateTimePicker6.Value.ToString("yyyy/MM/dd") + "\" +
        " and " + "\"" + dateTimePicker5.Value.ToString("yyyy/MM/dd") + "\";

        if (textBox7.Text != "") filtering += " and Продукция.Груз LIKE " + "\"" + textBox7.Text +
"%\"";
        if (textBox8.Text != "") filtering += " and Клиенты.ФИО LIKE " + "\"" + textBox8.Text +
"%\"";
        if (textBox9.Text != "") filtering += " and Клиенты.Организация LIKE " + "\"" +
textBox9.Text + "%\"";
        if (textBox10.Text != "" && textBox15.Text != "") filtering += " and Итого_к_сумме
between " + textBox10.Text + " and " + textBox15.Text;

        sqlDataAdapter2 = new SqlDataAdapter(filtering, sqlConnection);

        sqlBuilder = new SqlCommandBuilder(sqlDataAdapter1);

        sqlBuilder.GetInsertCommand();
        sqlBuilder.GetUpdateCommand();
        sqlBuilder.GetDeleteCommand();
        ReloadData();
    }

    private void materialRaisedButton2_Click(object sender, EventArgs e)
    {
        sqlDataAdapter2 = new SqlDataAdapter(query, sqlConnection);
        ReloadData();

        textBox7.Text = null;
        textBox8.Text = null;
        textBox9.Text = null;
        textBox10.Text = null;
        textBox15.Text = null;
    }

    private void toolStripTextBox1_TextChanged(object sender, EventArgs e)
    {
        // Поиск
        bindingSource.Filter = "Convert ([ID_заказа], 'System.String') LIKE " +
toolStripTextBox1.Text + "%";

        sqlBuilder = new SqlCommandBuilder(sqlDataAdapter1);
        sqlBuilder.GetInsertCommand();
        sqlBuilder.GetUpdateCommand();
        sqlBuilder.GetDeleteCommand();
    }

```

```

private void toolStripButton2_Click(object sender, EventArgs e)
{
    //Добавить запись

    using (OrderRegistration orderRegistration = new OrderRegistration(id_sotr, string.Empty))
    {
        orderRegistration.ShowDialog();
    }
}

private void toolStripButton5_Click(object sender, EventArgs e)
{
    //Редактировать запись

    if (dataGridView1.CurrentRow != null)
    {
        edit_client = true;

        using (OrderRegistration orderRegistration = new OrderRegistration(id_sotr,
textBox1.Text))
        {
            orderRegistration.ShowDialog();
        }
    }
    else MessageBox.Show("Таблица пустая", "Информация о редактировании",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}

private void toolStripButton3_Click_1(object sender, EventArgs e)
{
    //Удалить запись
    if (dataGridView1.CurrentRow != null)
    {
        if (MessageBox.Show("Вы хотите удалить заказ?", "Удаление",
MessageBoxButtons.YesNo, MessageBoxIcon.Question)
== DialogResult.Yes)
        {
            deleterowindex = dataGridView1.CurrentRow.Index;
            var Id_FK = dataGridView1.Rows[deleterowindex].Cells[0].Value;

            dataGridView1.Rows.RemoveAt(deleterowindex);

            dataSet2.Tables["Заказы"].Rows[deleterowindex].Delete();

            try
            {
                using (var sc = new SqlConnection(source))
                using (var cmd = sc.CreateCommand())
                {

```

```

        sc.Open();
        cmd.CommandText = "DELETE FROM Заказы WHERE ID_Заказа = @word";
        cmd.Parameters.AddWithValue("@word", Id_FK);
        cmd.ExecuteNonQuery();
        cmd.Dispose();
    }
    delete_client = true;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}

    ReloadData();
}

}
else MessageBox.Show("Таблица пустая", "Информация об удалении",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}

private void clientBindingNavigatorSaveItem_Click(object sender, EventArgs e)
    { // Сохранить изменения
        if (dataGridView1.CurrentRow != null)
        {
            rowIndexClient = dataGridView1.CurrentRow.Index;
            Id_Client = dataGridView1.Rows[rowIndexClient].Cells[0].Value.ToString();
        }
    }

private void toolStripButton1_Click_1(object sender, EventArgs e)
    { //Обновить данные

        ReloadData();

    }

private void диаграммаToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Charts charts = new Charts();
        charts.source = source;

        charts.ShowDialog();
    }

private void materialRaisedButton3_Click(object sender, EventArgs e)
    {
        Report report = new Report();
    }

```

```

        report.source = source;
        report.query = query;
        report.sqlDataAdapter1 = sqlDataAdapter2;

        report.ShowDialog();
    }

    private void посчитатьКоличествоКлиентовToolStripMenuItem_Click(object sender,
EventArgs e)
    {

    }

    private void textBox10_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (!(Char.IsDigit(e.KeyChar)))
        {
            if (e.KeyChar != (char)Keys.Back)
            {
                e.Handled = true;
            }
        }
    }

    private void textBox15_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (!(Char.IsDigit(e.KeyChar)))
        {
            if (e.KeyChar != (char)Keys.Back)
            {
                e.Handled = true;
            }
        }
    }

    private void сортироватьНомерЗаказаПоУбываниюToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        bindingSource.Sort = "ID_Заказа DESC";
    }

    private void сортироватьНомерЗаказаПоВозрастаниюToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        bindingSource.Sort = "ID_Заказа ASC";
    }

    private void сортироватьФамилииКлиентовАЯToolStripMenuItem_Click(object sender,
EventArgs e)
    {

```

```

        bindingSource.Sort = "ФИО ASC";
    }

    private void сортироватьФамилииКлиентовЯАToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        bindingSource.Sort = "ФИО DESC";
    }

    private void Orders_FormClosing(object sender, FormClosingEventArgs e)
    {
        e.Cancel = !(MessageBox.Show("Выйти из формы?", "Выход",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question)
        == DialogResult.Yes);
    }

    private void Orders_FormClosed(object sender, FormClosedEventArgs e)
    {
        GC.Collect();
        GC.WaitForPendingFinalizers();
    }
}
}
}

```

## Диаграммы

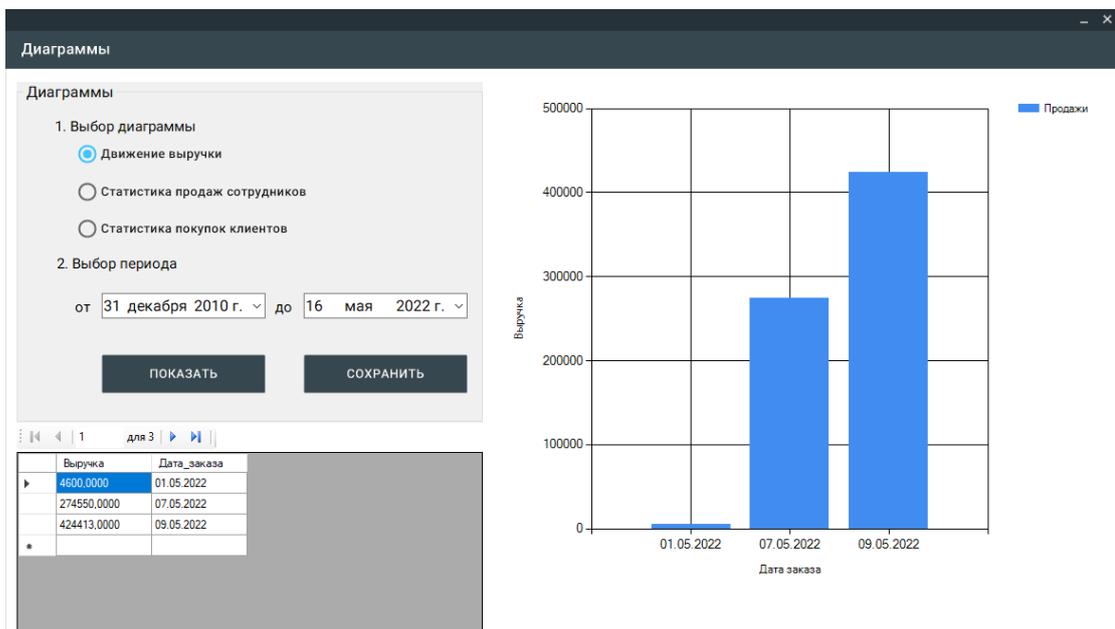


Рисунок 1 – Диаграмма «Движение выручки»

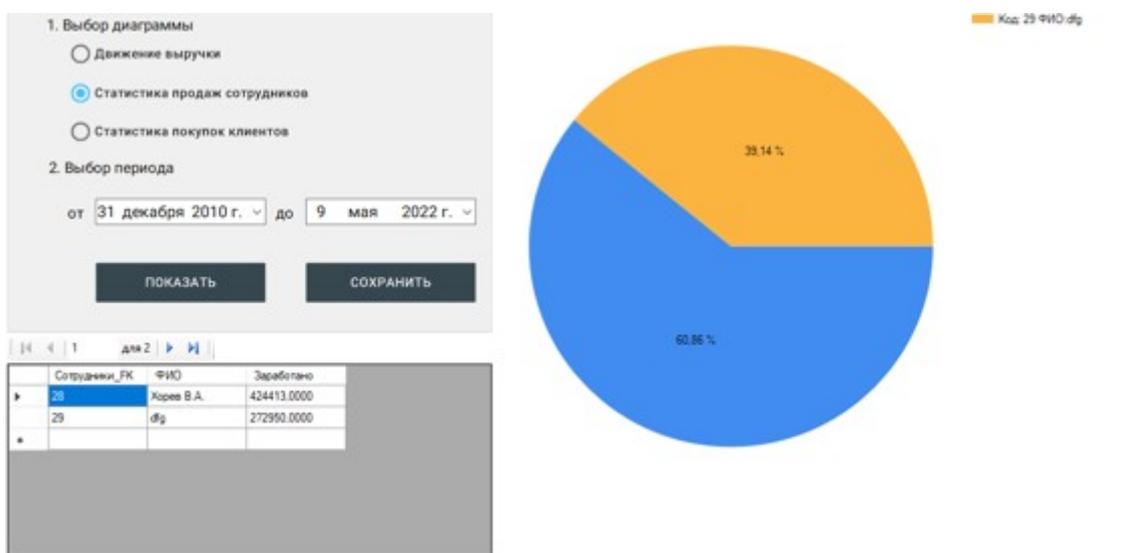


Рисунок 2 – Диаграмма «Статистика покупок клиентов»