**Государственное бюджетное профессиональное образовательное учреждение**

**Астраханской области «Астраханский колледж вычислительной техники»**

|  |
| --- |
| УТВЕРЖДЕНО |
| Цикловой комиссией |
| специальности 09.02.07 |
| ПЦК \_\_\_\_\_\_\_\_\_ Ю.С. Андрианова |
| «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |
| Специальность | *09.02.07 Информационные системы и программирование* | | |
| МДК 02.01 | *Технология разработки программного обеспечения* | | |
| Группа | *РП– 31* | | |
| **ЗАДАНИЕ** | | | |
| **на курсовое проектирование**  **Муханов Азамат Юрьевич** | | | |
|  | | | |
| (фамилия, имя, отчество студента) | | | |
| Тема курсового проекта | | ***Автоматизированная информационная система*** | |
| ***«Похоронное бюро»*** | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Дата выдачи задания | « | 26 | » | июня | |  | 20 | 20 | г. |
| Руководитель |  | | | | Ю.С. Андрианова | | | | |
| Студент |  | | | | А.Ю.Муханов | | | | |

**Содержание**

# **ВВЕДЕНИЕ**

# **1 ТЕХНИЧЕСКИЙ ПРОЕКТ**

## **1.1 Описание предметной области**

## **1.2 Анализ требований и уточнение спецификаций, выбор технологии, языка и среды разработки**

## **1.3 Описание технологии обработки информации**

### **1.3.1 Разработка диаграммы вариантов использования**

### **1.3.2 Разработка концептуальной модели предметной области**

## **1.4 Выбор методов решения задачи и разработка основных алгоритмов предметной области**

## **1.5 Требования к интерфейсу**

## **1.6 Минимальный состав технических средств**

# **2 РАБОЧИЙ ПРОЕКТ**

## **2.1 Общие сведения о работе программы.**

### **2.1.1 Общий алгоритм программного продукта**

## **2.2 Функциональное назначение**

## **2.3 Общий алгоритм программного продукта**

## **2.4 Руководство пользователя**

## **2.5 Заключение**

## **2.6 Список использованных источников**

# **ВВЕДЕНИЕ**

Любая финансовая деятельность нуждается в упорядочивании и постоянном контроле. **Ритуальная отрасль** – не исключение. Динамично развивающийся рынок этой сферы включает в себя большой ассортимент предложений. Современный подход к автоматизации позволяет достигнуть высоких результатов, эффективности и структурированности. Отсутствие упорядоченности создает хаос, поэтому тем, кто уже использовал в своей работе эти новшества, сложно представить рабочий процесс без них. Ознакомиться с многообразием полезных возможностей **ритуальной программы** можно только на личном примере. Консервативный подход с использованием таблиц Excel давно признан неэффективным. Структура и удобство этой программы устарела, взамен появляются новые программные продукты. На базе новых систем возможно создание удобного мобильного предложения для ритуальных агентов. Оторванность от рабочего стационарного места компенсируется техническими инновациями. Агент на выезде может получать сообщения напрямую из офиса, а его работа может координироваться другим сотрудником. Владелец агентства также получает преимущество, всегда зная, где находятся штатные сотрудники и чем они заняты в настоящий момент. К примеру, менеджер ритуального бюро по продажам может отправить клиенту, прибегавшему ранее к услугам захоронения, предложения об установке могильных плит или облагораживанию захоронения. Скорее всего, многие клиенты посчитают целесообразным воспользоваться услугами проверенного на личном примере добросовестного исполнителя. Одним словом, **автоматизация ритуальных услуг** удобна как для владельцев этой сферы бизнеса, так и для клиентов.

# **1 ТЕХНИЧЕСКИЙ ПРОЕКТ**

## **1.1 Описание предметной области**

В настоящее время сфера ритуальных услуг по уровню использования современных технологий в настоящее время существенно отстает от других видов деятельности. Так ещё совсем недавно в большинстве организаций ритуальной отрасли не было даже компьютеров. В то же время динамичное движение времени, общества, технологий не позволяет сохранить успех и эффективность практически любых предприятий при сохранении консервативных методов работы. При разработке Автоматизированной системы деятельности Похоронного бюро было проведено обcледование предметной облаcти. В результате в АС «Похоронное бюро» иcпользуютcя cледующие входные данные:

* информация о сотрудниках;
* информация о товаре на складе;
* информация о совершенных покупках

Входными данными являются данные полученные из запросов. Данные программы хранятся в базе данных MySql. База данных состоит из трех таблиц: пользователи, где пользователями являются сотрудники похоронного бюро, чек о покупке и таблица с информацией о товаре, хранящемся на складе. В курсовом проекте иcпользуются проверки на входные данные, к примеру, проверка на правильность пароля или проверка на пустоту в полях.

## **1.2 Анализ требований и уточнение спецификаций, выбор технологии, языка и среды разработки**

C# — объектно-ориентированный язык программирования. Язык имеет [статическую типизацию](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), поддерживает [полиморфизм](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BC%D0%BE%D1%80%D1%84%D0%B8%D0%B7%D0%BC_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), [перегрузку операторов](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BE%D0%B2) [делегаты](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BB%D0%B5%D0%B3%D0%B0%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), атрибуты, [события](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B9%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [свойства](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [обобщённые](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) типы и методы, [итераторы](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80), [анонимные функции](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F).

Среда разработки Visual Studio 2019 была специально создана для нужд .NET, и располагает всеми необходимыми инструментами.

В частности, в Visual Studio 2019 имеются различные шаблоны графического интерфейса пользователя, среди которых наиболее интересными являются Windows Forms. Приложение Windows Forms представляет собой событийно-ориентированное приложение, поддерживаемое Microsoft .NET Framework.

Для выполнения поставленной задачи, в связи с выше сказанным, было принято решение использовать язык программирования С# и среду разработки Visual Studio 2019

## **1.3 Описание технологии обработки информации**

### **1.3.1 Разработка диаграммы вариантов использования**

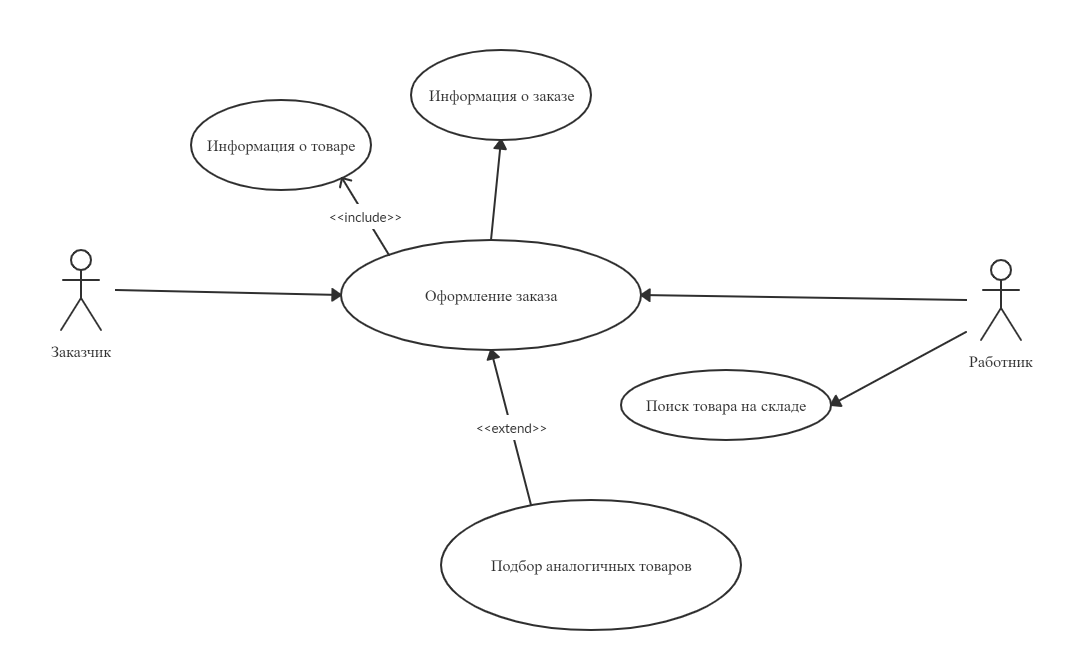


Рисунок 1 – Диаграмма вариантов использования

### **1.3.2 Разработка концептуальной модели предметной области**

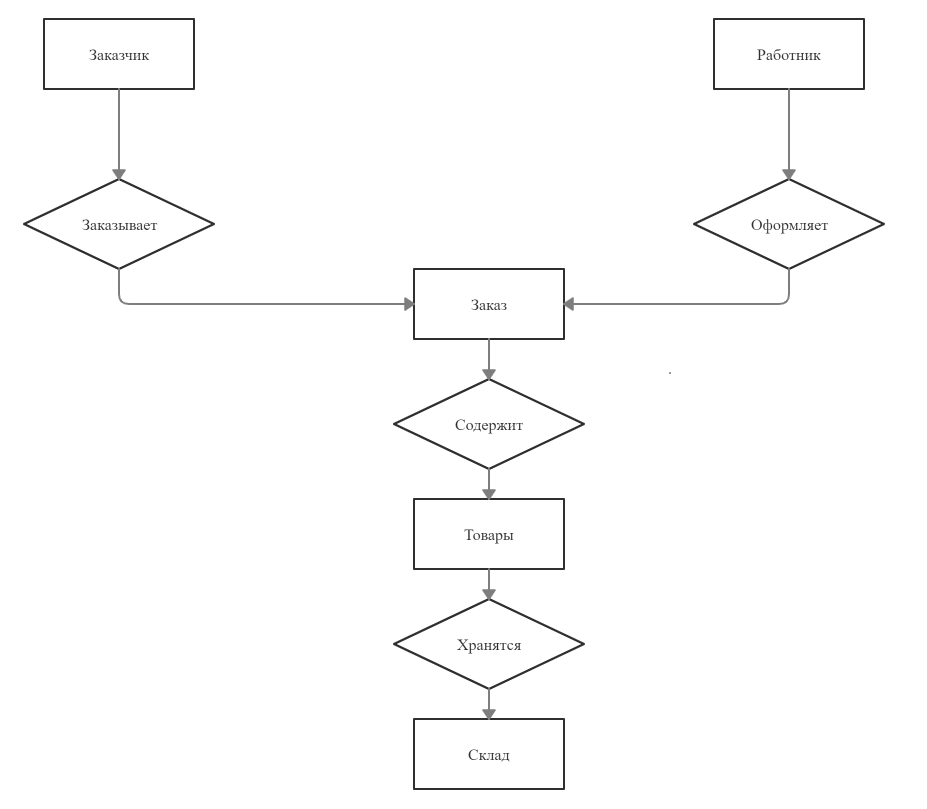


Рисунок 2 – Концептуальная модель предметной области

## **1.4 Выбор методов решения задачи и разработка основных алгоритмов предметной области**

Для реализации основной задачи курсового проектирования, написать автоматизированную систему похоронного бюро, были разработаны следующие алгоритмы:

1 Просмотр предыдущих покупок

2 Удаление данных о предыдущих покупках

3 Добавление товаров в БД

4 Изменение данных в БД

5 Удаление данных в БД

6 Добавление пользователя в БД

7 Подсчет стоимости товаров

8 Оформление заказа

9 Авторизация пользователя в программе

Авторизовавшись в программе пользователь попадет в форму с информацией об уже совершенных покупках. Нажав на кнопку «удалить», пользователь удалит выбранную запись из БД. Аналогично работают и другие функции программы.

В программе предусмотрены проверки при неправильных действиях пользователя. Сообщения будет выводиться в соответствии с действием

## **1.5 Требования к интерфейсу**

Интерфейс выполнен в одним стиле. Ниже представлено стартовое окно программы, не учитывая авторизацию и регистрацию

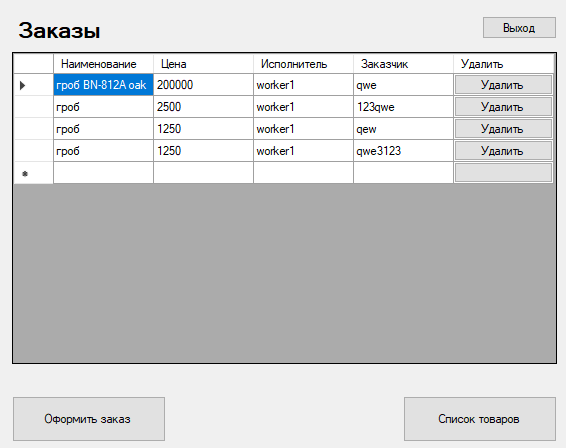


Рисунок 3- Форма вывода заказов

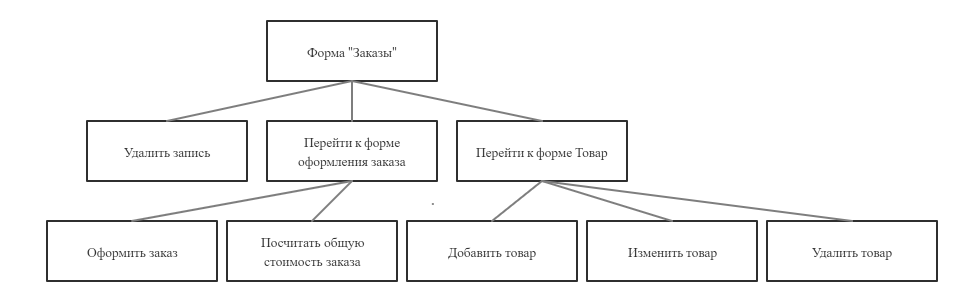


Диаграмма 1 – Иерархия меню

## **1.6 Минимальный состав технических средств**

Минимальные системные требования:

Для успешной работы в программе требуется персональный компьютер с операционной системой от Microsoft Windows 7 и выше. Требования к аппаратно-программному обеспечению:

1 Процессор — Pentium III 500 МГц (рекомендуется Pentium IV 1 ГГц);

2 Оперативная память — 512Мб;

3 Место на жестком диске — 30Мб;

6 Процессор — Pentium IV 1,8 ГГц; оперативная память — 4гб

Минимальный состав программных средств

Системные программные средства, используемые АС «Похоронное бюро» должны быть представлены локализованной версией операционной системы Windows 7 или выше

Для функционирования АС «Похоронной бюро»предустановленное программное обеспечение стороннего разработчика не требуется.

# **2 РАБОЧИЙ ПРОЕКТ**

## **2.1 Общие сведения о работе программы.**

### **2.1.1 Общий алгоритм программного продукта**

Форма «Авторизация»

Откроем программу и появляется форма «Авторизация». Весь алгоритм авторизации выполняется в кнопке – button2\_Click. Вводим в поля «логин» и «пароль» данные, если пароль и логин совпадают пользователю откроется форма «Заказы», иначе пользователю выведется сообщение об ошибке. Если пользователь не зарегистрирован в системе он может перейти по кнопке «Зарегистрироваться»

Форма «Регистрация»

Открывается форма «Регистрация». Весь алгоритм происходит в кнопке –SignUpButton.

Вводим в поля «логин» и «пароль» данные, и пользователя регистрирует в системе, ограничений на количество символов нет. Если пользователь не введет данные в одно из полей ему выведется сообщение об ошибке.

Форма «Заказы»

Загрузка данных происходит в событии Form4\_Load, данные загружаются в «dataGridView1». Для удаления записи из таблицы предусмотрена кнопка «DeleteBtn» расположенная напротив каждой записи. Кнопки button2\_Click и button3\_Click открывает другие формы.

Форма «Товары»

Загрузка данных происходит в событии Main\_Load, данные загружаются в «dataGridView1». Для добавление данных в таблице имеется кнопка «AddBtn», для изменения данных «UpdateBtn», для удаления «DeleteBtn». Кнопка button2\_Click вернет пользователя на форму заказов.

Форма «Оформление заказа»

Заполнение comboBox1 происходит в событии Forn5\_Load. Кнопка button2\_Click рассчитывает стоимость покупки. Кнопка button1\_Click вносит данные о заказе в базу данных

## **2.2 Функциональное назначение**

Функциональное назначение программы заключается в том, чтобы упростить и автоматизировать работу похоронного бюро.

## **2.3 Общий алгоритм программного продукта**

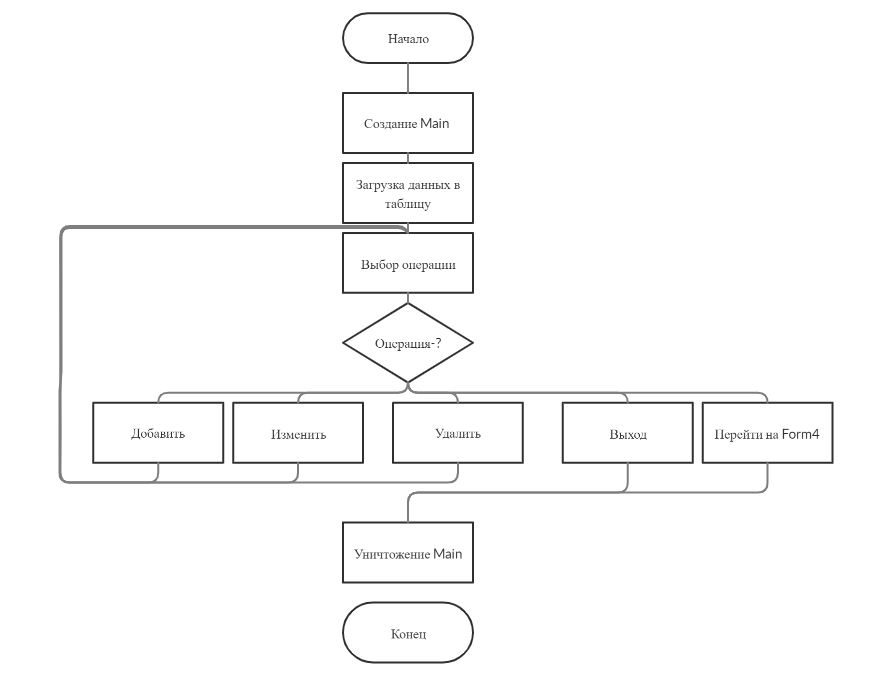


Рисунок 5 – Алгоритм формы Main

Алгоритм предназначен для вывода, добавления, редактирования и удаления информации.

Входными данными являются данные полученные из базы данных и операция выбранная пользователем, выходными же являются результат выполненной пользователем операции.

Сначала происходит генерация формы «Main» и последующая загрузка данных из базы данных в dataGridView1, если пользователь выберет кнопку «Добавить», предварительно напечатав данные в соответствующие поля, тогда кнопка сработает и добавит запись в базу данных. Если пользователь выберет кнопку «Изменить», предварительно напечатав данные в соответствующие поля, тогда кнопка сработает и изменит запись в базу данных. Если пользователь выберет кнопку «Удалить тогда кнопка сработает и удалит запись в базу данных. Если пользователь нажмет кнопку «Выход» тогда программа закроется. Если пользователь нажмет кнопку «Назад» его вернет на стартовую форму.

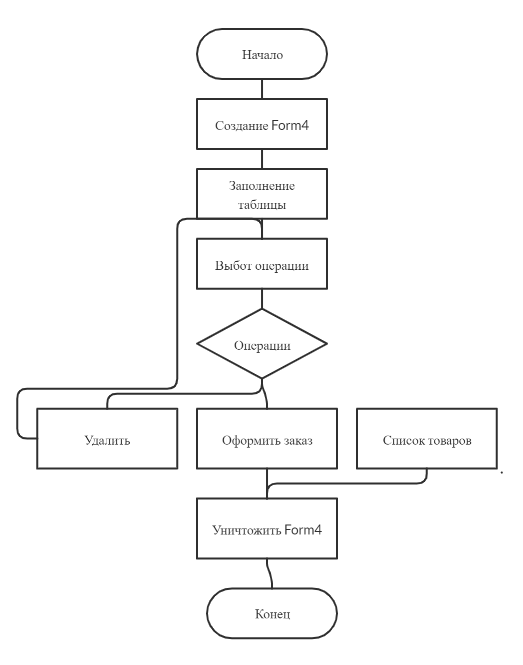


Рисунок 6 - Алгоритм Формы Form4

Алгоритм предназначен для вывода и удаления информации по заказам, а также переходом между другими формами.

Входными данными являются данные полученные из базы данных и операция выбранная пользователем, выходными же являются результат выполненной пользователем операции.

Сначала происходит генерация формы «Form4» и последующая загрузка данных из базы данных в dataGridView1, если пользователь выберет кнопку «Удалить», тогда кнопка сработает и удалит запись в базу данных. Если пользователь выберет кнопку «Оформить заказ» его перенаправит на форму «Form5». Если пользователь выберет кнопку «Список товаров» его перенаправит на форму «Form3».

## **2.4 Руководство пользователя**

Запустив файл «course\_work.exe» откроется окно авторизации (Рисунок 7)

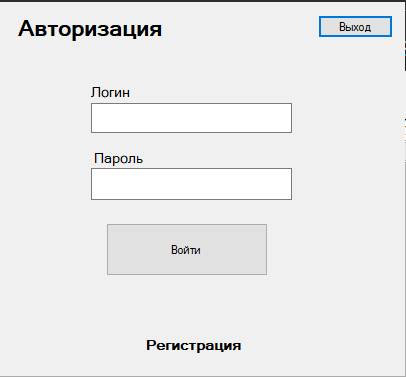


Рисунок 7 – Окно авторизации

Если пользователь зарегистрирован, то он вводит логин и пароль и может продолжить работу, если же пользователь не зарегистрирован, тогда он нажимает кнопку «регистрация» и переходит на форму регистрации (Рисунок 8).

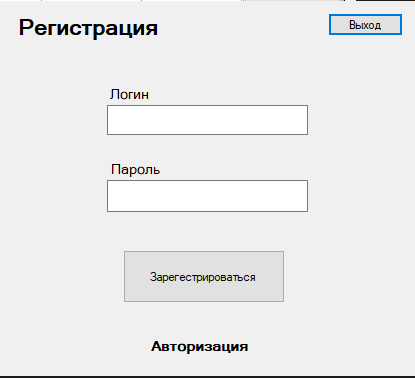


Рисунок 8 – Форма регистрации

Регистрация происходит аналогичным образом, что и авторизация. Зарегистрировавшись и войдя в программу пользователь перейдет на форму «Заказы»(Рисунок 9).

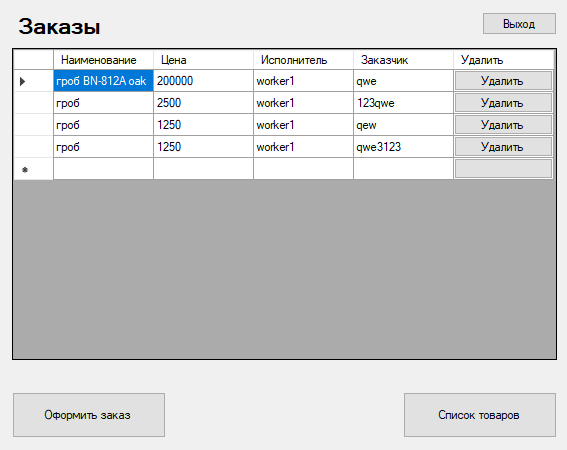


Рисунок 9 – Форма «Заказы»

Нажав на кнопку «Удалить», пользователь удалит строчку на которой находилась кнопка

Нажав на кнопку «Оформить заказ» пользователь перейдет на форму «Оформление заказа» (Рисунок 10). В поле «получатель» вводится имя заказчика, в раскрывающемся меню «Товар» пользователь выбирает один из нескольких товаров и выбирает количество воспользовавшись кнопками или просто введя число. Далее пользователь нажимает кнопку «Рассчитать стоимость» и получает общую стоимость всего заказа. Затем пользователь нажимает кнопку «Оформить», получает сообщение о успехе или об ошибке если пропустил один из пунктов приведенных выше.

Нажав на кнопку «Назад» пользователь попадет обратно на форму «Заказы»(Рисунок 9) перейдя по кнопке «Список товаров» открывается форма «Товары»(Рисунок 11)

В форме «Товары» выводятся все товары хранящиеся на складе. Для того чтобы добавить новую запись требуется кликнуть по последнему пустому полю и написать нужный вам товар, и так заполнив все поля и нажав кнопку «Добавить» пользователь добавит новую запись в таблицу и соответственно в базу данных. Чтобы изменить запись пользователю требуется выбрать поле которое он хочет изменить, написать текст и нажать кнопку «Изменить», данные в таблице изменятся. Если пользователь хочет удалить запись, то ему требуется выбрать кнопку удалить на требуемой строчке и нажать кнопку, данные пропадут из таблицы.

Для завершения работы программы пользователю нужно нажать кнопку «Выход» на любой форму.

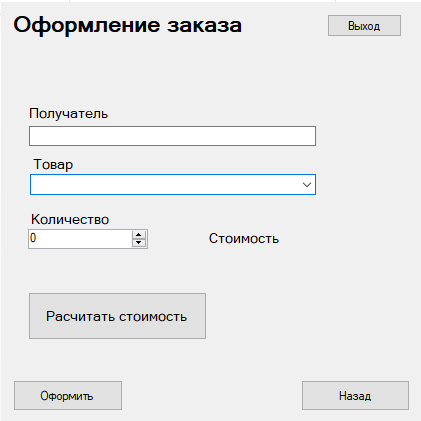


Рисунок 10 – форма «Оформление заказов»

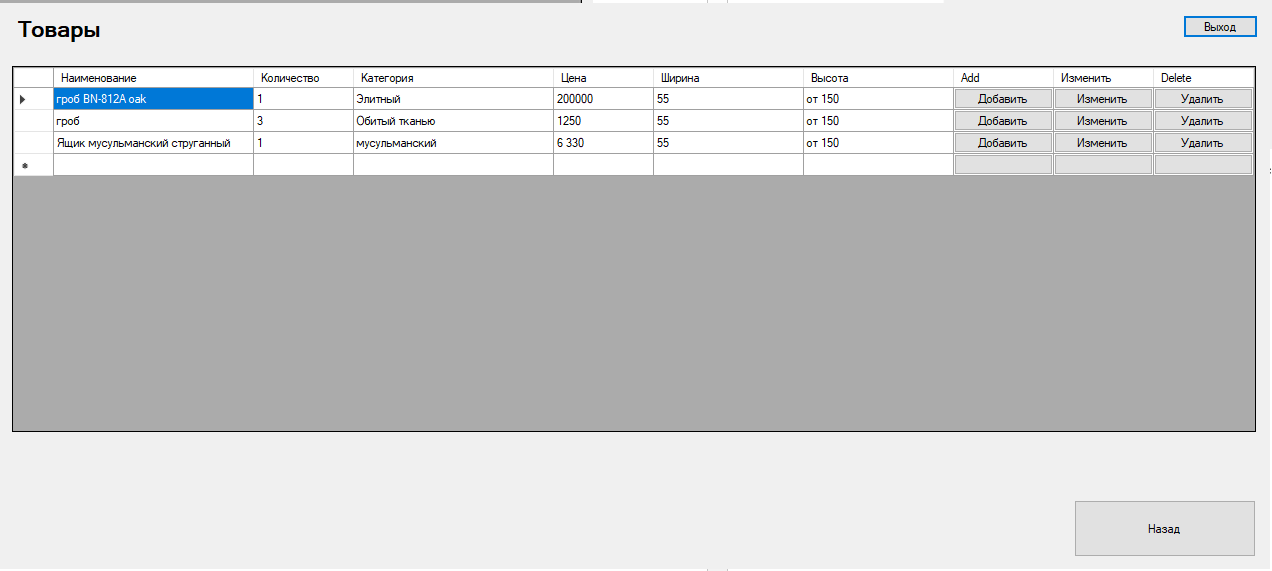


Рисунок 11 – Форма «Товары»

## **2.5 Заключение**

В заключении стоит сказать, что основные функции программы были реализованы: регистрация и авторизация пользователей, добавление, изменение и удаление товаров, создание заказа и его сохранение.

## **2.6 Список использованных источников**

1 <https://metanit.com/sharp/tutorial/>

2 <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/classes-and-structs/static-constructors>

3 Методические указание Адриановой Ю.С.