**Оглавление**

[Введение 3](#_Toc512065591)

[1. Анализ выбранной предметной области 4](#_Toc512065592)

[1.1. Краткая характеристика предметной области 4](#_Toc512065593)

[1.2. Диаграмма классов 4](#_Toc512065594)

[1.3. Диаграммы использования 5](#_Toc512065595)

[2. Описание среды разработки 6](#_Toc512065596)

[3. Разработка программного приложения 8](#_Toc512065597)

[3.1. Разработка главного меню приложения 8](#_Toc512065598)

[3.2. Пример создания пользовательской формы 9](#_Toc512065599)

[3.3 Подключение базы данны 12](#_Toc512065600)

[3.4 Пример отчета 13](#_Toc512065601)

[4. Описание реализации приложения 14](#_Toc512065602)

[4.1. Дерево функций приложения 14](#_Toc512065603)

[4.2. Сценарий диалога 15](#_Toc512065604)

[5. Схема функционирования приложения 16](#_Toc512065605)

[Заключение 17](#_Toc512065606)

[Используемые источники 18](#_Toc512065607)

# **Введение**

Целью научно-исследовательской работы является разработка программного приложения для решения задач бизнес-процесса поддержки взаимоотношений с клиентами. Необходимо решить следующие задачи:

1. Рассмотреть предметную область исследования.
2. Построить диаграммы классов и использования.
3. Описать выбранную среду разработки.
4. Представить экранные формы программного приложения.
5. Представить листинг кода экранных форм приложения.
6. Описать реализацию приложения.
7. Привести схему функционирования программного приложения.

# **Анализ выбранной предметной области**

## **1.1. Краткая характеристика предметной области**

Объектом исследования данной выпускной квалификационной работы является ООО «Перекресток». Компания ООО «Перекресток» была создана в 1995 году. Основным видом экономической деятельности ООО «Перекресток» является розничная торговля в неспециализированных магазинах. Также ООО «Перекресток» работает еще по направлению розничной торговли напитками в специализированных магазинах.

Основная цель компании — обеспечить потребителям возможность покупать качественные товары самого широкого ассортимента по доступным ценам в комфортных условиях.

На данный момент размер уставного капитала 507 301,00 руб. ООО «Перекресток».

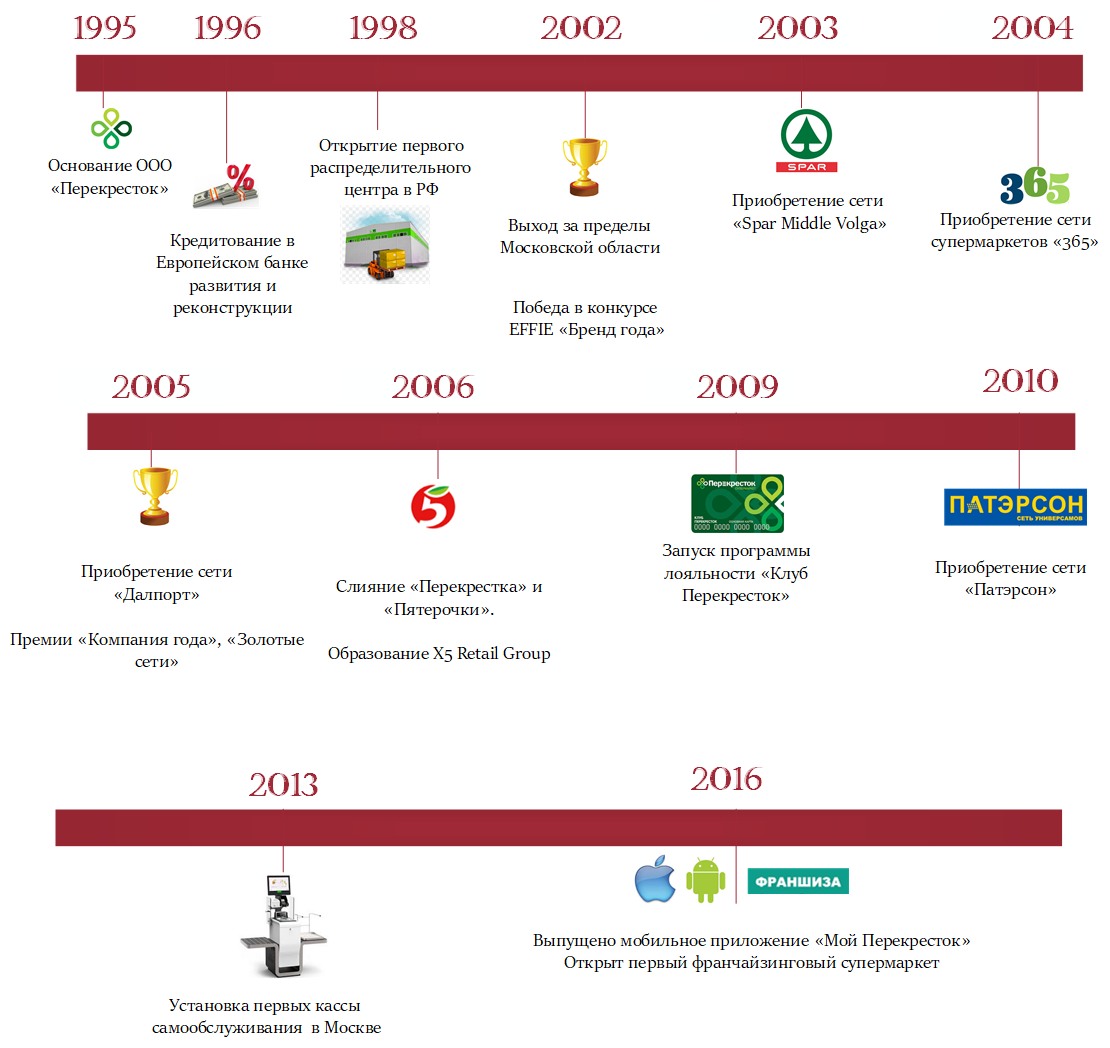


Рисунок 1 - История развития компании ООО «Перекресток»

Рисунок 2 – Организационная структура ООО «Перекресток»

Директор склада и логистики отслеживает за своевременной доставкой продуктов на склад магазина.

Маркетинговый отдел занимается маркетинговыми исследованиями, повышением рейтинга компании и поиском крупных корпоративных клиентов. Отдел планирования закупок занимается анализом деятельности магазина и планирование закупок для пополнения склада и запасов на складе филиала.

Управляющими механизмами являются: сотрудники отдела планирования закупок, отдела продаж, весь отдел склада и логистики

Предметом исследования данной дипломной работы является процесс управления запасами предприятия. Основной целью управления запасами является обеспечение бесперебойного процесса реализации продукции при минимизации совокупных затрат по обслуживанию запасов, определения оптимального количества запаса для каждой категории товаров с использованием методов поддержания оптимального запаса в цепочке поставок.

 Запасы создаются с целью:

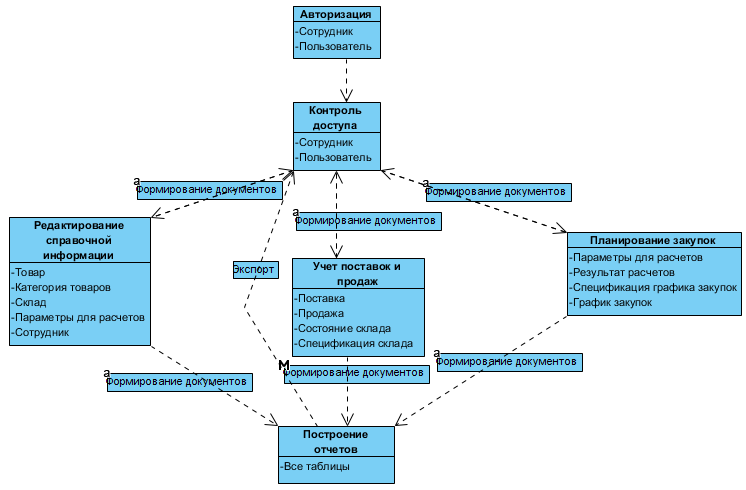
* создания возможности колебания покупательского спроса;
* нивелирования сезонных колебаний спроса на различные товары;
* получения скидки за покупку большой партии товаров;
* спекуляции на росте цен;
* снижения издержек при доставке товаров;
* реализации возможности моментального обслуживания клиентов.

Материальные запасы классифицируются как текущие и страховые. Текущими являются запасы, которые необходимы для обеспечения непрерывного торгового процесса. Величина текущих запасов постоянно меняется. Страховые запасы необходимы для поддержания непрерывного торгового процесса в случае возникновения непрерывных обстоятельств, таких как: изменение размера партии относительно указанного в договоре, наличие большого количества брака, задержка товара в пути и т.д. При правильной организации логистических процессов величина страховых запасов не меняется.

Рисунок 3 — Цикл процессов управления запасами

## **1.2. Диаграмма классов**

В результате анализа предметной области была сформирована диаграмма классов. На ней отражены все элементы базы данных, которые будут использоваться в прикладном приложении.

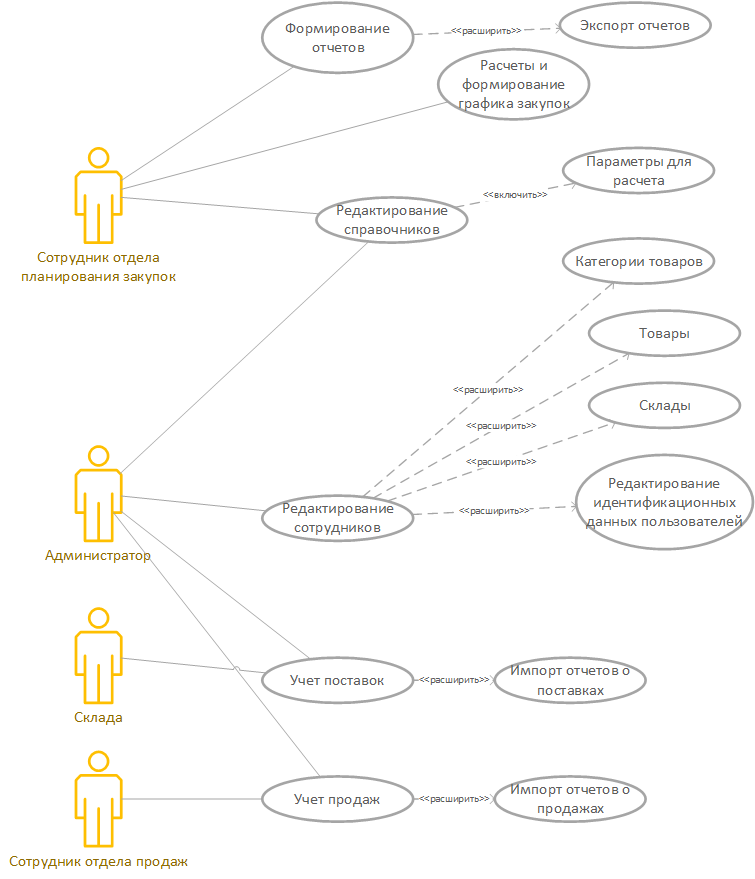


**Рисунок 4 ― Диаграмма классов**

Диаграмма классов (Рисунок 2) отражает различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. Диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

## **1.3. Диаграммы использования**

Для представления работы прикладного приложения была построена диаграмма прецедентов. Диаграмма прецедентов отражает отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. На рисунке 3 изображены три участника: клиент, администратор и врач. Пользователем данного модуля будет являться администратор. Вся собранная им информация будет использоваться клиентом и врачом.

****

**Рисунок 5 ― Диаграмма прецедентов**

# **2. Описание среды разработки**

Для разработки была выбрана такая среда разработки, как VisualStudio 2017 (C#). База данных Access 2016.

*MicrosoftVisualStudio 2017* — это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию. Благодаря простому интерфейсу программы, с ней не тяжело разобраться. С помощью VS мы можем создавать программы на таких известных языках программирования как: C, C++, C#, Visual Basic и т.д. На всех этих языках в VS мы можем создать любое приложение, любого формата — консольное приложение, обычную Windows форму, приложение для Windows Phone смартфона, библиотеку классов и тому подобные приложения. Студия очень удобна тем, что там не нужно прописывать абсолютно все коды элементов в ручном режиме (как в простом текстовом редакторе), в среде уже готовы все коды элементов управления (кнопки, чек-боксы, текстовые окна и тому подобные элементы).

*Язык программирования C#*.— объектно-ориентированный язык программирования. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к С++ и Java. Язык имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML. Переняв многое от своих предшественников — языков С++, Java, Delphi, Модула и Smalltalk — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++).

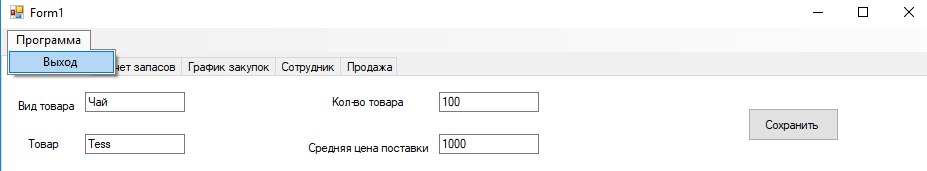
1. C# — является наследственным сыном двух мощных языков — C++ и Java,
2. C# — очень удобен в использовании (написании программ), у него достаточно простой синтаксис и мощные сигнатуры, благодаря которым мы можем создать базы данных не хуже чем SQL или LINQ,
3. C# — сейчас очень распространен и является одним из самых оптимальных языков программирования.

*Windows Forms* — интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде. Причём управляемый код — классы, реализующие API для Windows Forms, не зависят от языка разработки. То есть программист одинаково может использовать Windows Forms как при написании ПО на C#, С++, так и на VB.Net, J# и др.

*Microsoft Access 2016* — реляционная система управления базами данных (СУБД) корпорации Microsoft. Входит в состав пакета MicrosoftOffice. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA, в самом Access можно писать приложения, работающие с базами данных.

# **3. Разработка программного приложения**

## **3.1. Разработка главного меню приложения**

****

**Рисунок 6 Программа**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void выходToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_Курсовая\_С\_DataSet.Продажа". При необходимости она может быть перемещена или удалена.

this.продажаTableAdapter.Fill(this.\_Курсовая\_С\_DataSet.Продажа);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_Курсовая\_С\_DataSet.сотрудник". При необходимости она может быть перемещена или удалена.

this.сотрудникTableAdapter.Fill(this.\_Курсовая\_С\_DataSet.сотрудник);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_Курсовая\_С\_DataSet.Товар". При необходимости она может быть перемещена или удалена.

this.товарTableAdapter.Fill(this.\_Курсовая\_С\_DataSet.Товар);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_Курсовая\_С\_DataSet.График\_закупки". При необходимости она может быть перемещена или удалена.

this.график\_закупкиTableAdapter.Fill(this.\_Курсовая\_С\_DataSet.График\_закупки);

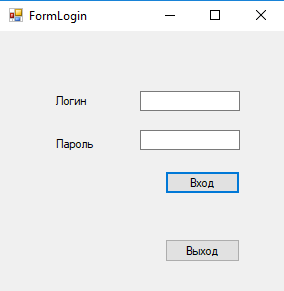
// TODO: данная строка кода позволяет загрузить данные в таблицу "\_Курсовая\_С\_DataSet.Расчет\_показателей\_управление\_запасами". При необходимости она может быть перемещена или удалена.

this.расчет\_показателей\_управление\_запасамиTableAdapter.Fill(this.\_Курсовая\_С\_DataSet.Расчет\_показателей\_управление\_запасами);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_Курсовая\_С\_DataSet.Запас\_товаров". При необходимости она может быть перемещена или удалена.

this.запас\_товаровTableAdapter.Fill(this.\_Курсовая\_С\_DataSet.Запас\_товаров);

## **3.2. Пример создания пользовательской формы**



**Рисунок 7 Форма входа**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class FormLogin : Form

{

public FormLogin()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "Архипов" && textBox2.Text == "1234")

{

Form1 form1 = new Form1();

form1.Show();

this.Hide();

}

else

{

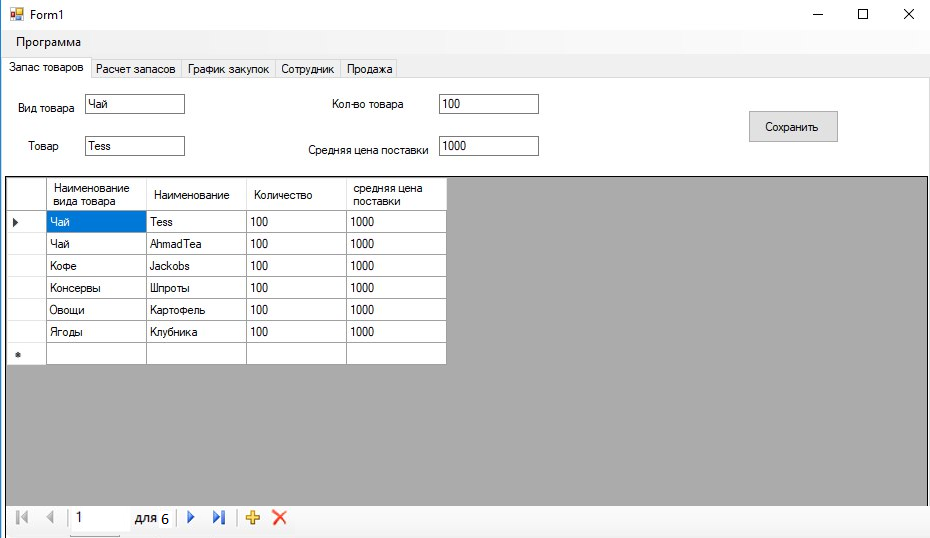
textBox1.Text = "";

textBox2.Text = "";

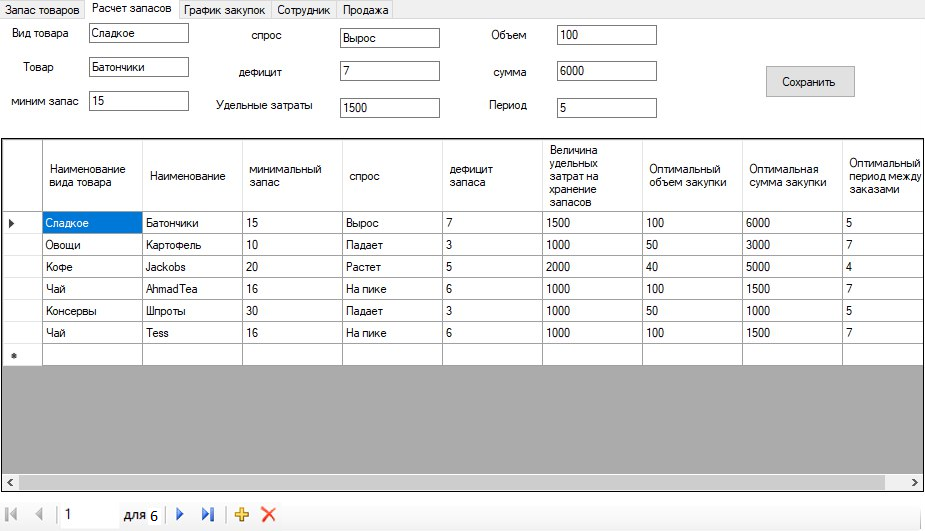
MessageBox.Show("Неправильный логин и пароль", "Ошибка");

}

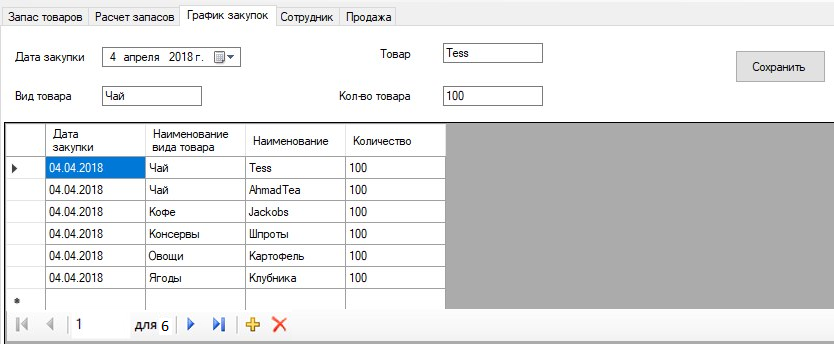
}

****

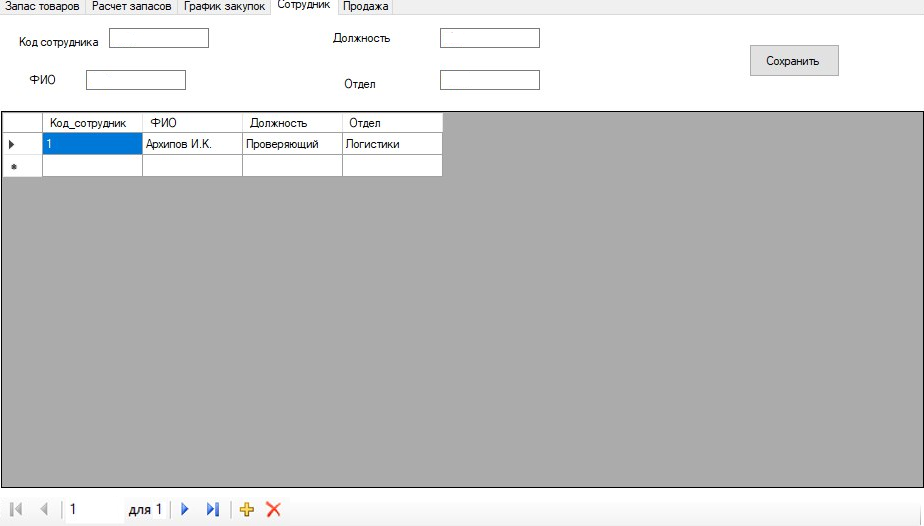
**Рисунок 8 Запас товаров**

****

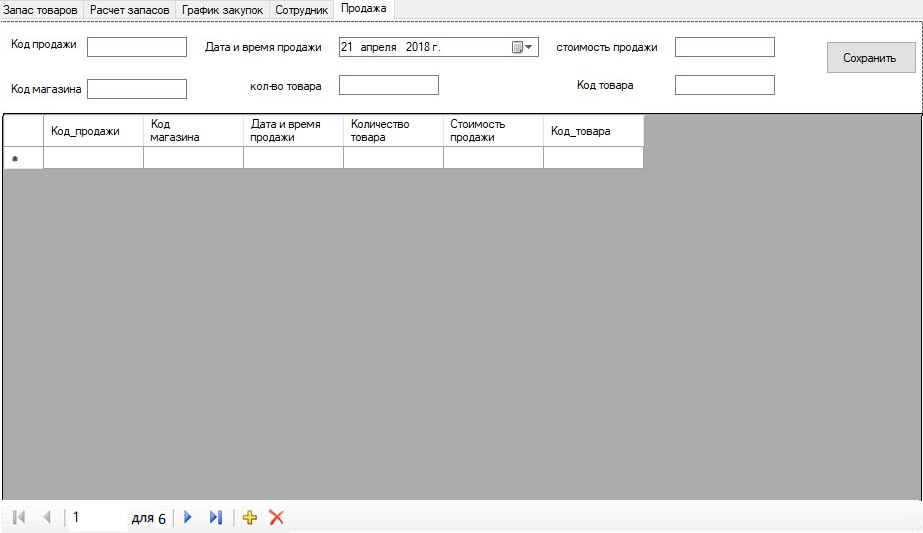
**Рисунок 9 Расчет запасов**

****

**Рисунок 10 График закупок**

****

**Рисунок 11 Сотрудник**

**Рисунок 12 Продажа**

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

this.запас\_товаровTableAdapter.Update(this.\_Курсовая\_С\_DataSet.Запас\_товаров);

}

private void label12\_Click(object sender, EventArgs e)

{

}

private void label8\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

}

private void menuStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

private void textBox24\_TextChanged(object sender, EventArgs e)

{

}

private void label23\_Click(object sender, EventArgs e)

{

}

private void textBox26\_TextChanged(object sender, EventArgs e)

{

}

private void программаToolStripMenuItem\_Click(object sender, EventArgs e)

private void button5\_Click(object sender, EventArgs e)

this.продажаTableAdapter.Update(this.\_Курсовая\_С\_DataSet.Продажа);

private void button4\_Click(object sender, EventArgs e)

this.сотрудникTableAdapter.Update(this.\_Курсовая\_С\_DataSet.сотрудник);

private void button3\_Click\_1(object sender, EventArgs e)

this.график\_закупкиTableAdapter.Update(this.\_Курсовая\_С\_DataSet.График\_закупки);

private void button2\_Click(object sender, EventArgs e)

this.расчет\_показателей\_управление\_запасамиTableAdapter.Update(this.\_Курсовая\_С\_DataSet.Расчет\_показателей\_управление\_запасами);

## **3.3 Подключение базы данны**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

****

## **3.4 Пример отчета**



**Рисунок 13 Отчет**



**Рисунок 14 Отчет**



**Рисунок 15 Отчет**

# **4. Описание реализации приложения**

## **4.1. Дерево функций приложения**

**Рисунок 16 ― Дерево функций**

Модель дерева функций относится к функциональному представлению и предназначена для описания иерархической структуры функций (включая статические связи между ними) бизнес-процессов предприятия. Функция — это задача, операция или действие, которые выполняются над объектом для достижения одной или нескольких целей.

## **4.2. Сценарий диалога**

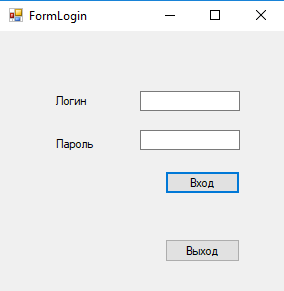
Разработка структуры сценария диалога представляет собой выявление состава функций, их иерархии и выбора средств организации взаимодействия пользователя с программой. Структура сценария диалога дает возможность определить состав его кадров, содержание каждого из кадров, а также их соподчиненность.

Развитие диалога во времени можно рассматривать как последовательность переходов системы из одного состояния в другое. Ни одно из этих состояний не должно быть тупиковым, т.е. пользователь должен иметь возможность перейти из любого текущего состояния диалога в требуемое (за один или несколько шагов). Для этого в ходе разработки интерфейса необходимо определить все возможные состояния диалога и пути перехода из одного состояния в другое ― разработать сценарий диалога.

**Рисунок 17 ― Сценарий диалога**

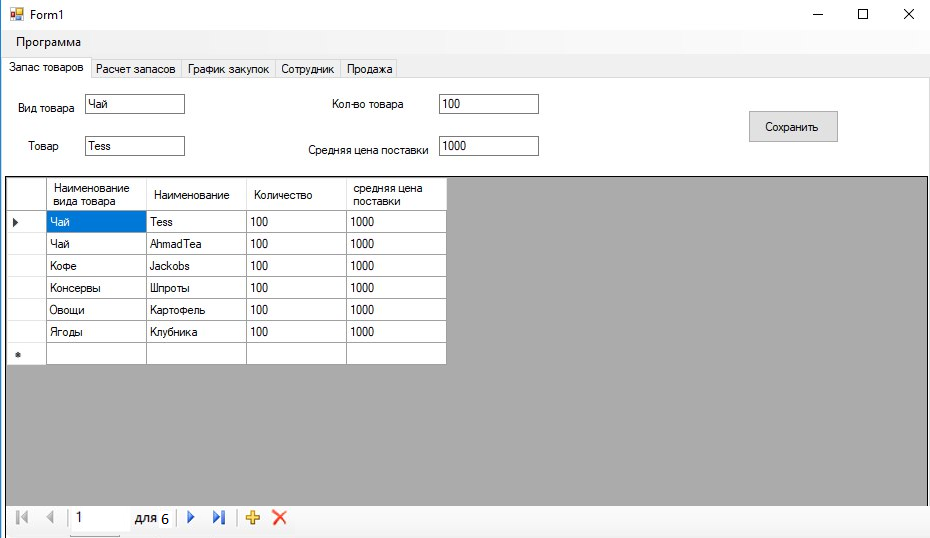
# **5.** **Схема функционирования приложения**

При запуске программного приложения открывается окно, где мы должны авторизоваться.



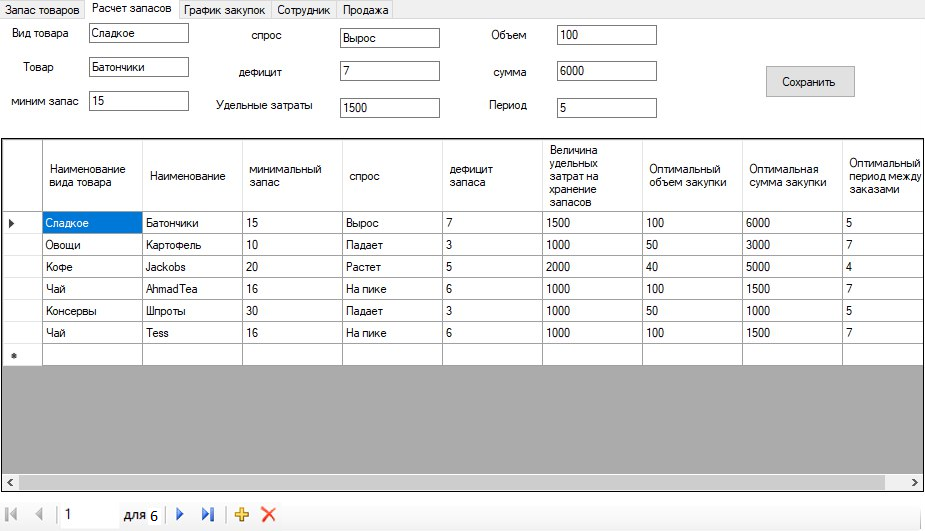
**Рисунок 18 ― Авторизация**

После чего нам откроется главная форма, на которой можно переадресовываться во все вкладки. С самого начала будет предоставлена работа с Запасом товаров.



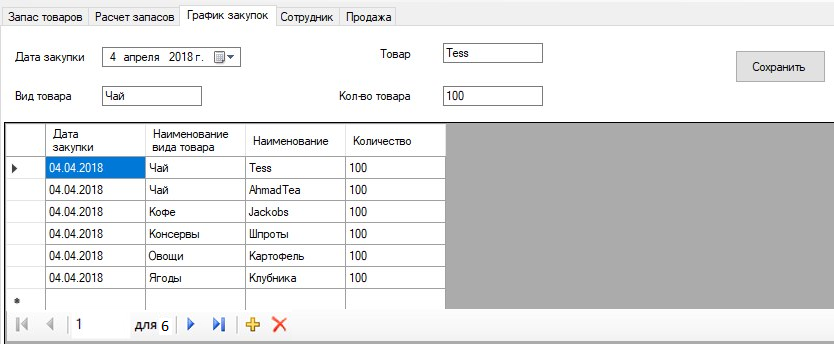
**Рисунок 19 - Запас товаров**

В следующей вкладке можно рассчитать запас этих товаров.



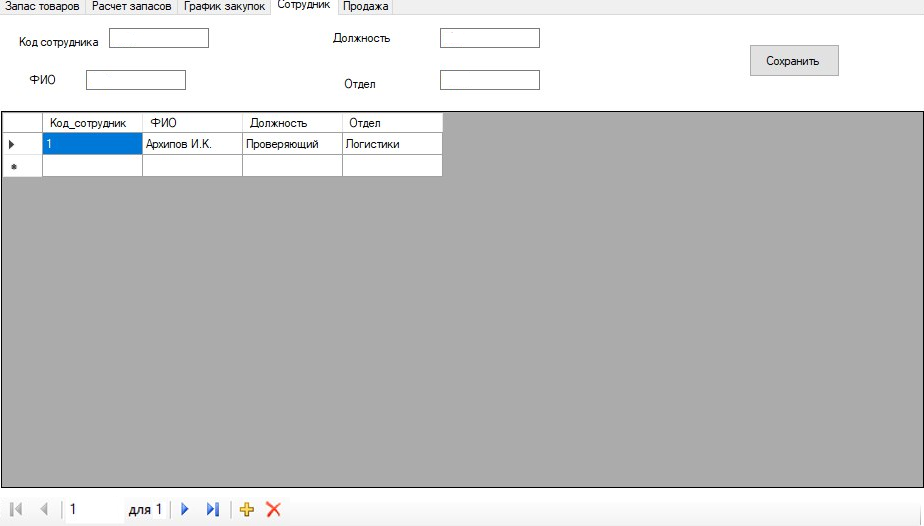
**Рисунок 20 – Расчет запасов**

На вкладке (Рис.21) можно произвести график закупок.



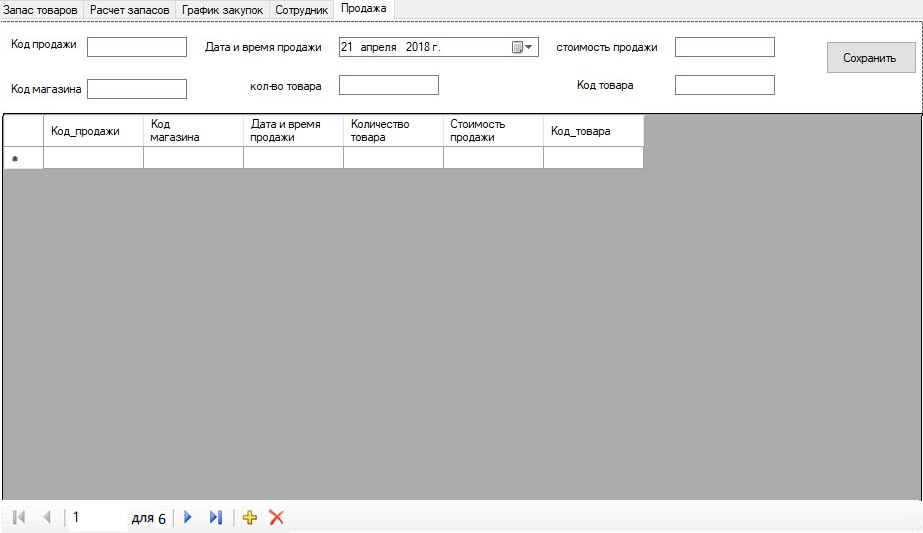
**Рисунок 21 – График закупок**

На рис.22 показана вкладка Сотрудник. Здесь можно добавлять и удалять сотрудников.



**Рисунок 22 – Сотрудник**

На рис.23 предоставлена вкладка Продажа, или как ее можно назвать в переносном смысле – чек.

****

**Рисунок 23 – Продажа**

# **Заключение**

В результате выполнения курсовой работы была разработана программа на языке Си#, позволяющая просматривать информацию из таблиц, редактировать их и вносить новые данные.

Данные вводятся пользователем с клавиатуры и выводятся в окне программы.

# **Используемые источники**

1. Г. Шилдт C#: Учебный курс. - Спб.: Питер, 2003. - 512с
2. Либерти, Д. Программирование на Си Шарп. - М.: Символ-плюс, 2005. - 684 с.
3. Богатырев, А. Язык программирования С [Электронный ресурс] / А. Богатырев.-электр. дан. - Режим доступа: http://www.refby.com. - Загл. с экрана.
4. Уэйт, М. Язык С. Руководство для начинающих. / М. Уэйт, С. Прага, Д. Мартин. - М.: Мир, 1995. - 521с.: ил.
5. Фаулер, М. UML в кратком изложении. / М. Фаулер. - М.: Мир, 2009 г. – 204 с.
6. https://www.visualstudio.com/