# 

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_ **ИУК «Информатика и управление»\_\_\_\_\_\_\_\_\_**

**КАФЕДРА** \_\_ ***ИУК5 «Системы обработки информации»***

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе на тему:**

***Разработка прикладного программного обеспечения: ” Склад общего хранения ”.***

по дисциплине ***Архитектура автоматизированных систем обработки информации и управления***

Студент гр. ИУК5-41Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Захаренков Д.С.)

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Смирнов М.Е.)

(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Калуга, 2021

Калужский филиал   
федерального государственного бюджетного образовательного учреждения высшего образования

**«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»   
(КФ МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой **\_\_ИУК5\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_(Е.В. Вершинин)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине **Архитектура автоматизированных систем обработки**

**информации и управления**

Студент Захаренков Д.С. ИУК5-51Б\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы, индекс группы)

Руководитель\_\_ \_Смирнов М.Е.\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы)

График выполнения проекта: 25% к\_4\_нед., 50% к\_7\_нед., 75% к\_10\_нед., 100% к\_14\_нед.

**1. Тема курсового проекта**

**Разработка прикладного программного обеспечения: ” Склад общего хранения ”.**

**2. Техническое задание**

Разработать приложение для получения информации о товарах и последующее добавление его в корзину.. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. Оформление курсового проекта**

3.1. Расчетно-пояснительная записка на\_\_\_\_\_\_\_\_ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.)\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_ Смирнов М.Е \_/

(подпись) (Ф.И.О.)

Задание получил \_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_Захаренков Д.С. \_/ «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021\_г.

(подпись) (Ф.И.О.)

Оглавление

[**1.Техническое задание** 4](#_Toc89958773)

[**1.1.  Общие сведения** 4](#_Toc89958774)

[**1.1.1 Полное наименование системы и ее условное обозначение** 4](#_Toc89958775)

[**1.2.1 Организация утвердившая документ** 4](#_Toc89958776)

[**1.3.1 Перечень документов, на основании которых создается система, кем и когда утверждены эти документы** 4](#_Toc89958777)

[**1.4.1 Плановые сроки начала и окончания работы по созданию системы** 4](#_Toc89958778)

[**1.2. Назначение и цели создания (развития) системы** 4](#_Toc89958779)

[**1.2.1 Назначение системы** 4](#_Toc89958780)

[**1.2.2. Цели создания системы** 4](#_Toc89958781)

[**1.3. Характеристика объектов автоматизации** 4](#_Toc89958782)

[**1.4. требования к системе** 5](#_Toc89958783)

[**1.4.1. требования к системе в целом** 5](#_Toc89958784)

[**1.4.2. Требования к защите информации от несанкционированного доступа.** 5](#_Toc89958785)

[**1.5. состав и содержание работ по созданию системы** 5](#_Toc89958786)

[**1.6. порядок контроля и приемки системы** 7](#_Toc89958787)

[**1.6.1. Виды испытаний** 7](#_Toc89958788)

[**1.6.2.Общие требование к приемке работы** 7](#_Toc89958789)

[**1.7. требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие** 7](#_Toc89958790)

[**1.8. требования к документированию** 7](#_Toc89958791)

[**1.9. источники разработки** 7](#_Toc89958792)

[**2. Научно исследовательская часть** 9](#_Toc89958793)

[**2.1 Постановка задачи проектирования** 9](#_Toc89958794)

[**2.2 Описание предметной области** 9](#_Toc89958795)

[**2.3 Анализ аналоговых прототипов.** 14](#_Toc89958796)

[**2.4 Перечень задач, подлежащих решению в процессе разработ-ки.** 18](#_Toc89958797)

[**3.Проектно-конструкторская часть** 20](#_Toc89958798)

[**3.1. Разработка структуры системы** 20](#_Toc89958799)

[**3.2. Разработка алгоритмов обработки информации.** 20](#_Toc89958800)

[**3.3. Разработка архитектуры приложения.** 23](#_Toc89958801)

[**3.4. Разработка граф интерфейса взаимодействия пользователя с системой** 26](#_Toc89958802)

[**4.Проектно-технологическая часть** 29](#_Toc89958803)

[**4.1. Технологические решения, поддерживающие эксплуатационный цикл программы** 29](#_Toc89958804)

[**4.2. Разработка руководства пользователя** 29](#_Toc89958805)

[**Заключение** 31](#_Toc89958806)

[**Список источников:** 32](#_Toc89958807)

[**Приложение** 33](#_Toc89958808)

# **1.Техническое задание**

## **1.1.  Общие сведения**

### **1.1.1 Полное наименование системы и ее условное обозначение**

Полное обозначение системы:  
 Прикладное программное обеспечение: ”Склад общего хранения”.  
 Условное обозначение системы: ”Склад”

### **1.2.1 Организация утвердившая документ**

Организацией утвердившей документ является университет КФ МГТУ им. Баумана

### **1.3.1 Перечень документов, на основании которых создается система, кем и когда утверждены эти документы**

Методические указания к выполнению курсовой работы по предмету “Базы данных”.

### **1.4.1 Плановые сроки начала и окончания работы по созданию системы**

* Начало работы-1.09.2021
* Окончание работы-10.12.2021

## **1.2. Назначение и цели создания (развития) системы**

### **1.2.1 Назначение системы**

Назначением системы является автоматизация получения информации о товарах с помощью windows form и использованием базы данных.

**1.2.2. Цели создания системы** Основными целями создания системы являются:

* Формирование практических навыков по разработке и реализации баз данных.

**1.3. Характеристика объектов автоматизации  
 1.3.1 Краткие сведения об объекте автоматизации или ссылки на документы, содержащие такую информацию.**

Объектом автоматизации приложения является склад общего хранения. Автоматизации подлежит получение товара из базы данных, его обработка и возможность будущего добавление его в корзину.

## **1.4. требования к системе**

### **1.4.1. требования к системе в целом**

#### **1.4.1.1. требования к структуре и функционированию системы**

Функциональная структура системы должна выполнять следующие задачи: получение информации о товаре и возможность добавления товара в корзину.

#### **1.4.1.2. требования к эргономике и технической эстетике**

Реализация графического оконного режима посредством Windows form.

### **1.4.2. Требования к защите информации от несанкционированного доступа.**

Требования к защите информации от несанкционированного доступа, на текущем этапе работы отсутствуют.

#### **1.4.2.1 Требования к функциональным характеристикам**

Программа должна обеспечивать выполнение следующих функций:

* Получение товара из бд и запись его в форму;
* Добавление и последующее отображение товара в корзине;
* Поиск товара:

1. По названию;
2. По цене;
3. По артиклю;

#### **1.4.2.2 Требования к составу и параметрам технических средств**

Для использования программы необходимо иметь средство ввода информации (мышь, Touch Screen или клавиатура).

## **1.5. состав и содержание работ по созданию системы**

Стадии и этапы создания АС в общем случае приведены в (Таблица №1)

Таблица №1

|  |  |  |
| --- | --- | --- |
| Стадии | Этапы работ | Сроки исполнения |
| 1. Формирование требований к АС | 1.1. Обследование объекта и обоснование необходимости создания АС.  1.2. Формирование требований пользователя к АС.  1.3. Оформление отчёта о выполненной работе и заявки на разработку АС (тактико-технического задания) | 16.09.2021 |
| 2. Разработка концепции АС. | 2.1. Изучение объекта.  2.2. Проведение необходимых научно-исследовательских работ.  2.3. Разработка вариантов концепции АС, удовлетворяющего требованиям пользователя.  2.4. Оформление отчёта о выполненной работе. | 02.10.2021 |
| 3. Техническое задание. | Разработка и утверждение технического задания на создание АС. | 23.10.2021 |
| 4. Разработка системы. | 4.1. Разработка системы согласно техническому заданию. | 30.11.2021 |
| 5. Защита курсовой  работы. | 5.1 Подготовка презентации и речи для защиты курсовой работы  5.2 Защита курсовой работы | 10.12.2021 |

## **1.6. порядок контроля и приемки системы**

**1.6.1. Виды испытаний**  
Тесты на работоспособность будет выполнен через Smoke test(дымовой тест),которое обеспечивается минимальным количеством тестов на явные ошибки.

**1.6.2.Общие требование к приемке работы** Научный руководитель проверяет соответствие программного продукта техническому заданию, а также содержание пояснительной записка

## **1.7. требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие**

Работ по подготовке объекта автоматизации проводить не требуется. Для заказа пользователю требуется запустить файл с расширением .exe

## **1.8. требования к документированию**

Проектная документация должна быть разработана в соответствии с ГОСТ 34.201-89

По окончанию работы предъявлена расчетно-пояснительная записка в состав которой входят:

* техническое задание;
* научно-исследовательская часть;
* проектно-конструкторская часть;
* проектно-технологическая часть.

Также должна быть предоставлена графическая часть работы, выполненная формате А1 на 2 листах, в которую входят:

* демонстрационные чертежи;
* алгоритмические схемы.

## **1.9. источники разработки**

* Е.В. Вершинин, Р.Б. Бобров “Методические указания к выполнению курсовой работы” Кф МГТУ им. Баумана 2021
* ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированной системы.
* ГОСТ 34.601-90 Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы Автоматизированные системы. Требования к содержанию документов

# **2. Научно исследовательская часть**

## **2.1 Постановка задачи проектирования**

Задачей проектирования является формирование практических навыков по работе и реализации программного приложения с использованием баз данных.

Необходимо разработать программное обеспечение, позволяющее добавлять информацию о заказе в базу данных, а также возможность хранения товара в базе данных.

## **2.2 Описание предметной области**

Склады - это здания и сооружения, предназначенные для приемки, размещения и хранения поступивших на них товаров, подготовки их к потреблению и отпуску потребителю.

Специалисты используют несколько разных терминов для складов, чаще их называют распределительными и логистическими центрами.

Изготовителю продукции необходимы склады сырья и исходных материалов, с помощью которых обеспечивается непрерывность производственного процесса. Склады готовой продукции позволяют содержать запас, обеспечивающий непрерывность сбыта. На складах торговли накапливаются и ожидают своего потребителя готовые изделия.

современный крупный склад – это сложное техническое сооружение, которое состоит из многочисленных взаимосвязанных элементов, имеет определенную структуру и выполняет ряд функций по преобразованию материальных потоков, а также накоплению, переработке и распределению грузов между потребителями. При этом в силу многообразия параметров, технологических решений, конструкций оборудования и характеристик разнообразной номенклатуры, перерабатываемых грузов склады относят к сложным системам. В то же время склад сам является всего лишь элементом системы более высокого уровня – логистической цепи, которая и формирует основные и технические требования к складской системе, устанавливает цели и критерии ее оптимального функционирования, диктует условия переработки груза.

Склады классифицируют по критериям:

* По размерам: от небольших помещений, общей площадью в несколько сотен квадратных метров, до складов-гигантов, покрывающих площади в сотни тысяч квадратных метров.
* По высоте укладки грузов: в одних груз хранится не выше человеческого роста, в других необходимы специальные устройства, способные поднять и точно уложить груз в ячейку на высоте 24 м и более.
* По конструкции: размещаться в отдельных помещениях (закрытые), иметь только крышу или крышу и одну, две или три стены (полузакрытые). Некоторые грузы хранятся вообще вне помещений на специально оборудованных площадках, в так называемых открытых складах.
* По необходимости создавать и поддерживать специальный режим, например, температура, влажность.
* По количеству пользователей: склад может предназначаться для хранения товаров одного предприятия (индивидуальное пользование), а может, на условиях лизинга сдаваться в аренду физическим или юридическим лицам (коллективное пользование или склад-отель).
* По степени механизации складских операций: немеханизированные, механизированные, комплексно-механизированные, автоматизированные и автоматические.
* По возможностям доставки и вывоза груза с помощью железнодорожного или водного транспорта: пристанционные или портовые склады (расположенные на территории железнодорожной станции или порта), прирельсовые (имеющие подведенную железнодорожную ветку для подачи и уборки вагонов) и глубинные.
* По широте ассортимента хранимого груза: специализированные склады, склады со смешанным или с универсальным ассортиментом.
* По месту в процессе движения материальных потоков от первичного источника сырья до конечного потребителя готовой продукции:

Согласно этой классификации все складские помещения делятся на 6 классов: А+, А, В+, В, С, D. Соответствие склада тому или иному классу носит рекомендательный характер и зависит от многих факторов: месторасположение;

* этажность;
* высота;
* пролет;
* наличие инженерного оборудования (вентиляция, отопление и прочее оборудование, позволяющее создавать определенные климатические условия);
* наличие охранных систем и систем пожаротушения, видеонаблюдения; высота расположения полов и наличие антипылевого покрытия;
* наличие определенного количества ворот докового типа, погрузочно-разгрузочных площадок, регулируемых по высоте;
* наличие площадок для маневрирования и стоянки грузового и легкового автотранспорта; наличие офисных, вспомогательных помещений и строений;
* наличие ограждения территории с круглосуточной охраной и многое другое.

**Склад класса А+** Складские помещения класса “А+” – высшая категория складов согласно общей классификации. Такие помещения отвечают самым высоким требованиям складского хозяйства. Склад категории “А+” своего рода президентские апартаменты среди помещений этого предназначения.

**Складские помещения класса “А”** отвечают высоким требованиям логистики. От складов высшего класса они отличаются немногим. Различия в основном заключаются в площади самого помещения и высоте потолков. Играет роль также расположение склада относительно магистралей. Склады класса “А” – это профильные сооружения, призванные обеспечить наилучшие условия хранения любых типов груза.

**Склад класса B+ Складские помещения класса “B+”** – категория B+ объединяет складские помещения, построенные или переоборудованные под выполнение задач хранения различных типов грузов. Категория “В+” – своего рода люкс в классификации складских помещений. Сооружения данной категории имеют ряд преимуществ, таких как приемлемая стоимость в сочетании со всеми необходимыми условиями складского хозяйства.

**Склад класса B Складские помещения класса “B”** в полной мере можно считать оптимальным решением для компаний, умеющих считать свои деньги и в то же время пользоваться благами цивилизации без ненужных излишеств. Хранение грузов на складах этой категории выгодно и в то же время надежно. Склады отвечают современным требованиям.

**Складские помещения класса “C”** – это, чаще всего, утепленный ангар или капитальное производственное помещение, с высотой потолков не менее четырех метров. Здания могут быть любой этажности. Главное в случае многоэтажных строений – наличие грузовых лифтов в необходимом количестве. Полы – бетонные или из асфальта, покрытие отсутствует. Обязательным условием для складских помещений категории “С” является наличие ворот на нулевом уровне, для обеспечения заезда грузового транспорта внутрь помещения.

**Складские помещения класса “D”** – пожалуй, самые непритязательные в плане требований к их оснащению. Под склады данной категории используют подвальные помещения, объекты гражданской обороны, ангары, производственные помещения и прочие нежилые и технические площади. Требования здесь предъявляются минимальные.

**Основные функции склада**

**Преобразование производственного ассортимента в потребительский в соответствии со спросом**. Особое значение данная функция приобретает в распределительной логистике, где торговый ассортимент включает огромный перечень товаров различных производителей, отличающихся функционально, по конструктивности, размеру, форме, цвету и т.д. Создание нужного ассортимента на складе содействует эффективному выполнению заказов потребителей и осуществлению более частых поставок и в том объеме, который требуется клиенту.

**Складирование и хранение** позволяет выравнивать временную разницу между выпуском продукции и ее потреблением и дает возможность осуществлять непрерывное производство и снабжение на базе создаваемых товарных запасов.

**Преобразование материальных потоков**. Любой склад обрабатывает, по меньшей мере, три вида потоков: входной, выходной и внутренний. На складе происходит расформирование одних грузовых партий или грузовых единиц и формирование других, распаковка грузов, комплектование новых грузовых единиц, их упаковка, затаривание.

**Сокращение затрат на перевозку**. Многие потребители заказывают со складов партии «меньше чем вагон» или «меньше чем трейлер», что значительно увеличивает издержки, связанные с доставкой таких грузов. Для сокращения транспортных расходов склад может осуществлять унитизацию, т.е. объединение небольших партий грузов для нескольких клиентов, до полной загрузки транспортного средства. Еще одна форма объединения происходит в случае, когда заказчику необходимы материальные ресурсы от разных поставщиков. Противоположная операция – разбивка опта – также позволяет сократить расходы на перевозку грузов от одного поставщика ко множеству заказчиков на определенной территории.

**Предоставление услуг**. Очевидным аспектом этой функции является оказание клиентам различных услуг, обеспечивающих фирме высокий уровень обслуживания потребителей.

## **2.3 Анализ аналоговых прототипов.**

Для ведения учета товаров на складе было разработано огромное количество самых разнообразных программ. Каждая из них отличается способами хранения и добавления товаров, а также возможность автоматизации закупок и проверки товара на складе.

Openbravo



Рис.1 интерфейс программы Openbravo

Программа Openbravo, как аналог «1С: Торговля и склад», подходит для производства и торговой сферы. Приложение ориентировано на малый и средний бизнес. Openbravo является ERP-системой с открытым кодом, предназначенной для работы сети распределенных POS-систем, объединённых одной базой данных.

Плюсы:

* Финансовый учет
* Закупки, складской учет

Минусы:

* Для небольших торговых точек подходит слабо из-за сложности сопровождения и больших затрат

Программа «Ананас»

Приложение «Ананас» - это бесплатная, свободно распространяемая платформа для автоматизации учета. Она позволяет автоматизировать закупки, продажи и генерировать отчёты. Фактически «Ананас» является бесплатным аналогом 1С и способен работать на Linux и Windows.

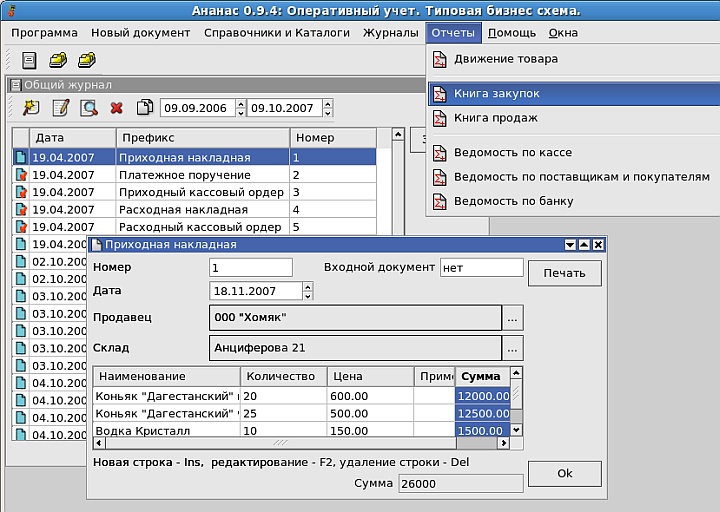


Рис.2 интерфейс приложения «Ананас»

Плюсы:

* Позволяет автоматизировать закупки
* Добавление и генерация отчёта

Минусы:

* Необходимость при первоначальной установке создавать бизнес-модель компании

GrossBee

GrossBee является системой, предназначенной для комплексной автоматизации предприятий и её можно отнести к ERP. В однопользовательском формате можно воспользоваться бесплатной версией приложения. Структура GrossBee является модульной и подстраивается индивидуально под клиента, поэтому коробочные решения у компании отсутствуют. Стоимость продукта также определяется отдельно в каждом случае.

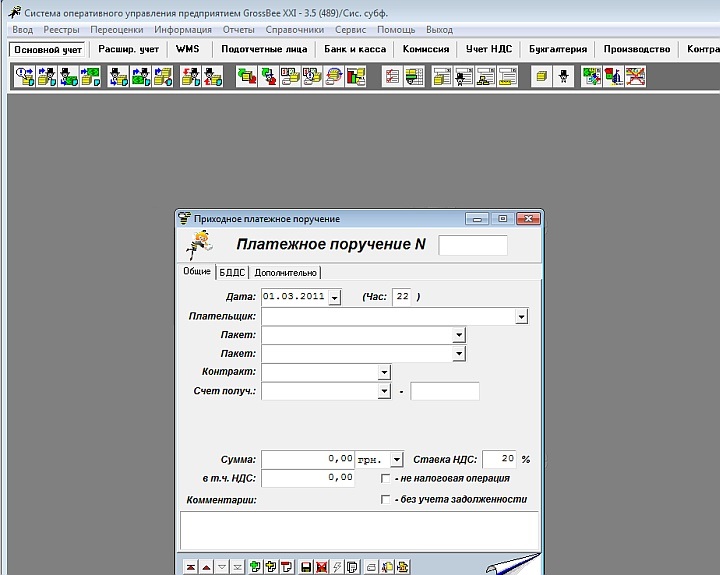


Рис.3 интерфейс системы GrossBee

Плюсы:

* Учет производства
* Учет договоров
* Учет денежных средств

Минусы:

* Не подходит для малого бизнеса из-за сложности внедрения и необходимости обучения.

Таким образом каждое приложение имеет ряд достоинств и недостатков, одни имеют удобный интерфейс вкупе с широким набором возможностей, например автоматизация закупок и генерация отчетов, другие позволяют вести учет. Поэтому целью моей работы является создание приложения для склада класса B позволяющее закупать товары со склада, а также удобный в использовании интерфейс.

## **2.4 Перечень задач, подлежащих решению в процессе разработ-ки.**

1. Изучение принципов системного программирования и работы Winows form функций.

2. Реализация актуальных получения и записи данных из бд в windows form.

3. Реализация пользовательского интерфейса программы, соответствующего заданным требованиям.

##### **2.5 Обоснование выбора инструментов и платформы для разра-ботки.**

Средой проектирования позволяющими реализовать подобное программное обеспечение, была выбрана Microsoft Visual Studio. Среда разработки Visual Studio представляет собой полный набор средств для разработки и создания веб приложений, настольных приложений и мобильных приложений.

Visual C# использует единую интегрированную среду разра-ботки (IDE), которая позволяет совместно использовать средства и упрощает создание решений на базе нескольких языков.

Преимущества выбранной среды программирования:

• Удобство интерфейса

• Возможность использования стандартных функций windows form

• Сравнительно не высокие требования к системе

Самыми популярными Операционными системами на данный момент являютя:

Windows — именно операционная система Microsoft считается самой популярной и наиболее знакомой простым пользователям из разных уголков мира. Программное обеспечение компании остается основой современных компьютеров. Его выбирают из-за простоты использования, быстрого запуска и возобновления работы. Последние версии операционной системы отличаются повышенной безопасностью и вниманием к конфиденциальности пользователей. С ней вы и ваши данные находятся под надежной защитой, и особенно переживать за свое личное пространство не приходится.

Ubuntu - операционная система на базе Linux, которая поставляется со всеми необходимыми инструментами. Она отлично подходит для разнообразных общественных организаций, школ и домашнего использова-ния. Операционная система поддерживается Canonical - глобальным разработчиком программного обеспечения, который сыскал славу на распространении Ubuntu. Она абсолютно бесплатная, и только это должно стать мотивирующим фактором для того, чтобы самостоятельно попробовать ее. Все-таки далеко не все готовы тратиться на программное обеспечение в любом возможном виде.

MacOS - эксклюзивная операционная система Apple, у которой свой путь развития. Ее нельзя полноценно установить ни на одно другое устройство, кроме Mac. Она заточена под конкретное железо, которое использует производитель, и работает на нем максимально гладко. Для macOS подходит только программное обеспечение, которое создано конкретно для этой платформы. За операционную систему не нужно платить - ее стоимость уже вложена в цену далеко не самого дешевого аппаратного обеспечения.

Платформой для разработки была выбрана Windows 10. На данный момент являющаяся самой распространенной ОС в мире (Рисунок 11)

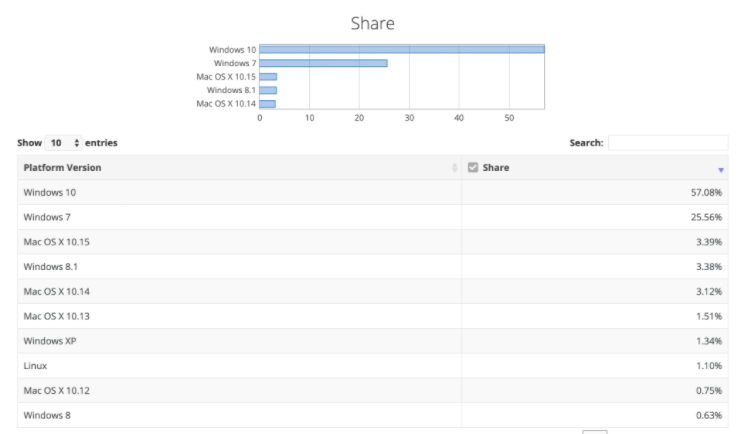


Рис.4 Статистика использования операционных систем в мире

В качестве языка программирования был выбран язык С#.

С#‎ разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов С#‎, которая отражает BCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR.

# **3.Проектно-конструкторская часть**

## **3.1. Разработка структуры системы**

Программа создана как Win32-приложение, работает на основе интерфейса Windows form. Графическое окружение создается с помощью конструктора формы.

Приложение должно давать пользователю возможность выбора товара и его количество, а также добавление его в корзину. Источником информации для приложения являются данные из базы данных. Потребителем является пользователь приложение. Приложение выполнено графическим интерфейсом, в котором пользователь выбирает нужные ему данные.

Все компоненты разрабатываются на Windows 10 в среде разработки Visual Studio 2019, используя С# в качестве языка программирования.

В Microsoft visual studio community выбирается создание приложения Windows с помощью «Приложение Windows Forms (.NET Framework)». Созданный проект сразу содержит код и основную форму приложения. Следующим требуется придумать структуру меню. На форме определить перечень элементов управления и отображения. Задать обработку событий для всех элементов управления и логику заполнения для всех элементов отображения.

## **3.2. Разработка алгоритмов обработки информации.**

При выборе сектора или поиск по параметрам приложение получает значение из базы данных и передает их в dataGridView. После чего пользователь выбирает товар, вводит нужное количество товара и нажимает кнопку. Затем приложение получает значение строки в которой находится товар, его цену и количество. Если товара достаточно происходит расчет цены за нужное количество товара и запись его в новый dataGridView в другой форме.



Рис.5 блок-схема программы

## **3.3. Разработка архитектуры приложения.**

Создаем на самой форме кнопки, текстбоксы и dataGridView.

После чего задаем значения для них

При загрузке формы добавляем в комбобокс1 и 2 значения  
comboBox1.Items.Add("Бытовая химия");

comboBox1.Items.Add("Пищевые продукты");

comboBox2.Items.Add("Поиск товара по названию");

comboBox2.Items.Add("Поиск товара по артиклю");

comboBox2.Items.Add("Поиск товара по цене");

При выборе одного из пунктов комбобокса 1 получаем нужные данные о товарах

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (comboBox1.SelectedIndex == 0)

{

Initil("product\_name, item\_number, remains, barcode, price\_per\_piece", " where number\_of\_sector = 1");

}

if (comboBox1.SelectedIndex == 1)

{

Initil("product\_name, item\_number, remains, barcode, price\_per\_piece" , " where number\_of\_sector = 2");

}

}

При нажатии на кнопку 1 мы получаем нужное нам значение из dataGridView1, а также количество товара из текстбокса 3.

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

{

if (comboBox1.SelectedIndex != 0 && comboBox1.SelectedIndex != 1 &&comboBox2.SelectedIndex != 0 && comboBox2.SelectedIndex != 1 && comboBox2.SelectedIndex != 2)

{

MessageBox.Show("Товар не выбран");

}

else {

int index = dataGridView1.CurrentCell.RowIndex;

int check = (int)dataGridView1[2, index].Value;

string check1 = textBox3.Text;

if (check1 != "")

{

int help = Convert.ToInt32(check1);

if (check >= help)

{

int price = help \* (int)dataGridView1[4, index].Value;

string item = dataGridView1[1, index].Value.ToString();

item\_number.Add(item);

Count.Add(help);

Price.Add(price);

}

else MessageBox.Show("Товара недостаточно");

} else MessageBox.Show("Количество товара не выбрано");

}

}

При нажатии на кнопку 3 мы производим поиск товара с помощью селекта

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

{

string search = textBox1.Text;

if (comboBox2.SelectedIndex != 0 && comboBox2.SelectedIndex != 1 && comboBox2.SelectedIndex != 2) MessageBox.Show("Категория поиска не выбрана");

else if (comboBox2.SelectedIndex == 0 && search != "") Initil("product\_name, item\_number, remains, barcode, price\_per\_piece", $@" where product\_name LIKE '%{search}%'");

else if (comboBox2.SelectedIndex == 1 && search != "") Initil("product\_name, item\_number, remains, barcode, price\_per\_piece", $@" where item\_number = {search}");

else if (comboBox2.SelectedIndex == 2 )

{

string check1 = textBox5.Text; string check2 = textBox7.Text;

Initil("\*", $@" where price\_per\_piece >= {check1} and price\_per\_piece <= {check2} ");

}

else if (search == "") MessageBox.Show("Поле поиска пусто");

}

При нажатии кнопки 2 мы передаем селект в функцию Initil2 которая передает информацию из бд в dataGridView другой формы, а также сумму всей покупки

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

{

string Str = "";

int price = 0;

int count = 1;

int count1 = item\_number.Count;

int count2 = 0;

if (count1 > 0)

{

foreach (string p in item\_number)

{

if (count < count1)

{

count++;

Str += $@"Select product\_name, price\_per\_piece , {Count[count2]} as Count from product where item\_number = {p} union ";

}

else Str += $@"Select product\_name, price\_per\_piece, {Count[count2]} as Count from product where item\_number = {p}";

count2++;

}

foreach (int p in Price)

{

price += p;

}

Initil2(Str);

newForm.textBox2.Text = price.ToString();

newForm.Show();

} else MessageBox.Show("Корзина пуста");

}

Функции Initil и Initil2 позволяют получать значения из базы данных в dataGridView

public void Initil(string a, string b)

{

NpgsqlConnection connect = new NpgsqlConnection("Server=localhost;Port=5432;Database=warehouse;User id=postgres;Password=1217112171dz;");

connect.Open();

NpgsqlCommand command = new NpgsqlCommand();

command.Connection = connect;

command.CommandType = CommandType.Text;

command.CommandText = $@"Select {a} from product {b}";

NpgsqlDataReader data\_reader = command.ExecuteReader();

if (data\_reader.HasRows)

{

DataTable data\_table = new DataTable();

data\_table.Load(data\_reader);

dataGridView1.DataSource = data\_table;

}

connect.Dispose();

connect.Close();

}

public void Initil2(string a)

{

NpgsqlConnection connect = new NpgsqlConnection("Server=localhost;Port=5432;Database=warehouse;User id=postgres;Password=1217112171dz;");

connect.Open();

NpgsqlCommand command = new NpgsqlCommand();

command.Connection = connect;

command.CommandType = CommandType.Text;

command.CommandText = a;

NpgsqlDataReader data\_reader = command.ExecuteReader();

if (data\_reader.HasRows)

{

DataTable data\_table = new DataTable();

data\_table.Load(data\_reader);

newForm.dataGridView2.DataSource = data\_table;

}

connect.Dispose();

connect.Close();

}

## **3.4. Разработка граф интерфейса взаимодействия пользователя с системой**

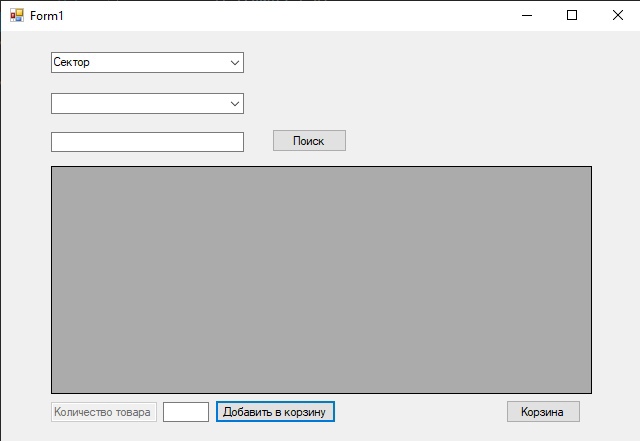


Рис.6 Главная форма со всеми элементами



Рис.7 Combobox2, Textbox1 и Button 3 используемые для поиска товара

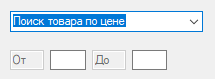


Рис.8 В зависимости от типа поиска добавляется Textbox5-8

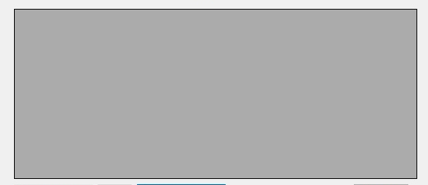
  
Рис.9 dateGridView для отображения информации из базы данных



Рис.10 С помощью Textbox 3 и Button 1 происходит добавления нужного количества товара в корзину



Рис.11 с помощью Button2 происходит открытие второй формы

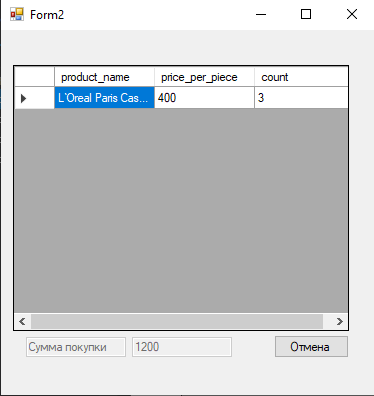


Рис.12 Вторая форма являющаяся корзиной

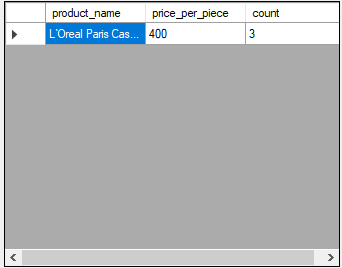


Рис.13 dateGridView для отображения информации о товарах в корзине



Рис.14 Сумма товара отображается в textbox 2

# **4.Проектно-технологическая часть**

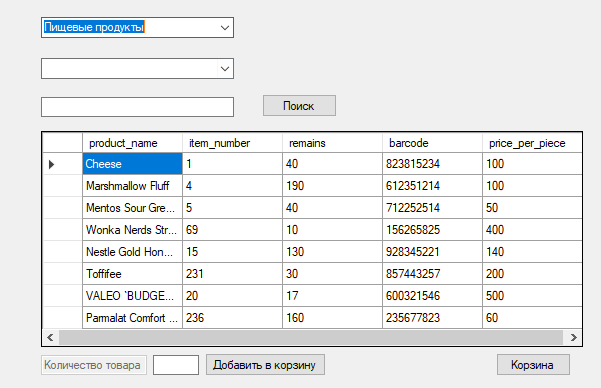
## **4.1. Технологические решения, поддерживающие эксплуатационный цикл программы**

В данное приложение можно добавить :

* функцию оплаты и добавление заказа и заказчика из формы в базу данных.
* Функция аутентификации пользователя и работника
* функцию проверки просрочен товар или нет.
* Функция проверки состояния заказа.
* Возможность получение работником заказа, информацию о котором он может посмотреть внутри приложения. После сбора возможность отметить статус выполнения и последующую отправку пользователю.

## **4.2. Разработка руководства пользователя**

После запуска приложения необходимо выбрать сектор который нас интересует или найти товар с помощью поиска

Рис.15 товар находящийся в нужном нам секторе

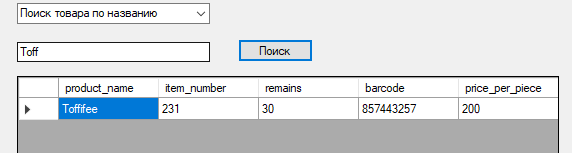


Рис.16 Найденный нами товар

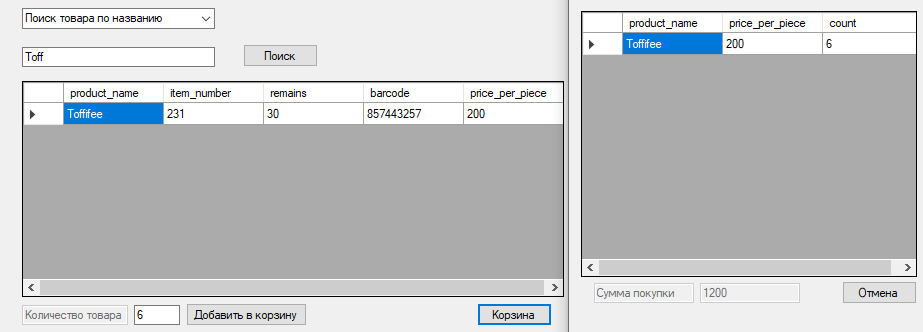


Рис.17 Добавляем интересующий нас товар в корзину

# **Заключение**

В ходе выполнения курсовой работы было разработано приложение с помощью windows form для получении информации о товаре и добавлении его в корзину.

Были сформированы навыки по разработке и реализации программного приложения с базой данных, а также реализованы все поставленные задачи.

# **Список источников:**

1. ГОСТ 34.201-90 Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы Автоматизированные системы. Требования к содержанию документов
2. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированной системы.
3. Е.В. Вершинин, Р.Б. Бобров “Методические указания к выполнению курсовой работы” Калуга КФ МГТУ им. Баумана 2021
4. В.Н.Шакин. Г.К.Сосновиков, И.Б.Юскова. Методические указания по дисциплине ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ БД. М., МТУСИ.2004 .
5. Г.К.Сосновиков, В.Н.Шакин, И.Б.Юскова. Методические указания и контрольные задания по дисциплине ОСНОВЫ ПОСТРОЕНИЯ БД. М., МТУСИ. 2004.
6. <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/controls/windows-forms-controls-by-function?view=netframeworkdesktop-4.8>
7. <https://ru.wikipedia.org/wiki/Windows_Forms>

# **Приложение**

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;

using Npgsql;

namespace WindowsForms

{

public partial class Form1 : Form

{

Form2 newForm = new Form2();

public void Initil(string a, string b)

{

NpgsqlConnection connect = new NpgsqlConnection("Server=localhost;Port=5432;Database=warehouse;User id=postgres;Password=1217112171dz;");

connect.Open();

NpgsqlCommand command = new NpgsqlCommand();

command.Connection = connect;

command.CommandType = CommandType.Text;

command.CommandText = $@"Select {a} from product {b}";

NpgsqlDataReader data\_reader = command.ExecuteReader();

if (data\_reader.HasRows)

{

DataTable data\_table = new DataTable();

data\_table.Load(data\_reader);

dataGridView1.DataSource = data\_table;

}

connect.Dispose();

connect.Close();

}

public void Initil2(string a)

{

NpgsqlConnection connect = new NpgsqlConnection("Server=localhost;Port=5432;Database=warehouse;User id=postgres;Password=1217112171dz;");

connect.Open();

NpgsqlCommand command = new NpgsqlCommand();

command.Connection = connect;

command.CommandType = CommandType.Text;

command.CommandText = a;

NpgsqlDataReader data\_reader = command.ExecuteReader();

if (data\_reader.HasRows)

{

DataTable data\_table = new DataTable();

data\_table.Load(data\_reader);

newForm.dataGridView2.DataSource = data\_table;

}

connect.Dispose();

connect.Close();

}

public Form1()

{

InitializeComponent();

}

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

{

comboBox1.Items.Add("Бытовая химия");

comboBox1.Items.Add("Пищевые продукты");

comboBox2.Items.Add("Поиск товара по названию");

comboBox2.Items.Add("Поиск товара по артиклю");

comboBox2.Items.Add("Поиск товара по цене");

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (comboBox1.SelectedIndex == 0)

{

Initil("product\_name, item\_number, remains, barcode, price\_per\_piece", " where number\_of\_sector = 1");

}

if (comboBox1.SelectedIndex == 1)

{

Initil("product\_name, item\_number, remains, barcode, price\_per\_piece" , " where number\_of\_sector = 2");

}

}

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

{}

List<string> item\_number = new List<string>();

List<int> Price = new List<int>();

List<int> Count = new List<int>();

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

{

if (comboBox1.SelectedIndex != 0 && comboBox1.SelectedIndex != 1 &&comboBox2.SelectedIndex != 0 && comboBox2.SelectedIndex != 1 && comboBox2.SelectedIndex != 2)

{

MessageBox.Show("Товар не выбран");

}

else {

int index = dataGridView1.CurrentCell.RowIndex;

int check = (int)dataGridView1[2, index].Value;

string check1 = textBox3.Text;

if (check1 != "")

{

int help = Convert.ToInt32(check1);

if (check >= help)

{

int price = help \* (int)dataGridView1[4, index].Value;

string item = dataGridView1[1, index].Value.ToString();

item\_number.Add(item);

Count.Add(help);

Price.Add(price);

}

else MessageBox.Show("Товара недостаточно");

} else MessageBox.Show("Количество товара не выбрано");

}

}

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

{

string Str = "";

int price = 0;

int count = 1;

int count1 = item\_number.Count;

int count2 = 0;

if (count1 > 0)

{

foreach (string p in item\_number)

{

if (count < count1)

{

count++;

Str += $@"Select product\_name, price\_per\_piece , {Count[count2]} as Count from product where item\_number = {p} union ";

}

else Str += $@"Select product\_name, price\_per\_piece, {Count[count2]} as Count from product where item\_number = {p}";

count2++;

}

foreach (int p in Price)

{

price += p;

}

Initil2(Str);

newForm.textBox2.Text = price.ToString();

newForm.Show();

} else MessageBox.Show("Корзина пуста");

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

textBox1.Clear();

if (comboBox2.SelectedIndex == 0)

{

textBox1.Visible = true;

textBox4.Visible = false;

textBox5.Visible = false;

textBox6.Visible = false;

textBox7.Visible = false;

}

if (comboBox2.SelectedIndex == 1)

{

textBox1.Visible = true;

textBox4.Visible = false;

textBox5.Visible = false;

textBox6.Visible = false;

textBox7.Visible = false;

}if (comboBox2.SelectedIndex == 2)

{

textBox1.Visible = false;

textBox4.Visible = true;

textBox5.Visible = true;

textBox6.Visible = true;

textBox7.Visible = true;

}

}

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

{

string search = textBox1.Text;

if (comboBox2.SelectedIndex != 0 && comboBox2.SelectedIndex != 1 && comboBox2.SelectedIndex != 2) MessageBox.Show("Категория поиска не выбрана");

else if (comboBox2.SelectedIndex == 0 && search != "") Initil("product\_name, item\_number, remains, barcode, price\_per\_piece", $@" where product\_name LIKE '%{search}%'");

else if (comboBox2.SelectedIndex == 1 && search != "") Initil("product\_name, item\_number, remains, barcode, price\_per\_piece", $@" where item\_number = {search}");

else if (comboBox2.SelectedIndex == 2 )

{

string check1 = textBox5.Text; string check2 = textBox7.Text;

Initil("\*", $@" where price\_per\_piece >= {check1} and price\_per\_piece <= {check2} ");

}

else if (search == "") MessageBox.Show("Поле поиска пусто");

}

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

{

}

}

}