**МИОНОБРНАУКИ РОССИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ»  
(СПбГЭУ)**

Факультет Информатики и Прикладной Математики  
Кафедра

**КУРСОВАЯ РАБОТА**  
по дисциплине:  
**«Информатика и программирование»**

Тема: Создание программного комплекса для компрессии и декомпрессии информации в электронном виде

Направление (специальность) *09.03.03 Прикладная информатика*

Направленность (специализация) *Управление бизнес-процессами и проектами*

Обучающийся *Стерлин Павел Михайлович*

Группа *ПИ-2201* Подпись:\_\_\_\_\_\_\_\_\_\_\_\_

Проверил: Сотавов А. К.

Должность: \_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_ Дата: \_\_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург  
2023

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc103717750)

[ВВЕДЕНИЕ 3](#_Toc103717751)

[1. ИСТОРИЯ РАЗВИТИЯ ТЕХНОЛОГИЙ ПРЕДСТАВЛЕНИЯ ДАННЫХ 5](#_Toc103717752)

[2. МОДЕЛЬ РАБОТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПО УПРАВЛЕНИЮ ЗАКУПКАМИ 15](#_Toc103717753)

[2.1 МОДУЛЬ «РАБОТА С ПОСТАВЩИКАМИ» 18](#_Toc103717754)

[2.2 МОДУЛЬ «ДЕТАЛИ ЗАКАЗА» 20](#_Toc103717755)

[2.3 МОДУЛЬ «СКЛАД» 21](#_Toc103717756)

[2.4 МОДУЛЬ «СОХРАНЕНИЕ» 21](#_Toc103717757)

[3. РЕАЛИЗАЦИЯ ПРОГРАММЫ НА ЯЗЫКЕ C# 22](#_Toc103717758)

[ЗАКЛЮЧЕНИЕ 34](#_Toc103717759)

[СПИСОК ЛИТЕРАТУРЫ 35](#_Toc103717760)

[ПРИЛОЖЕНИЕ А 36](#_Toc103717761)

# ВВЕДЕНИЕ

**Актуальность темы.** Количество данных создаваемых человечеством поистине огромно, увеличиваясь по экспоненте каждый год. Часть этих данных попадет в долгосрочное хранилище — архив. Будь то обязанность компаний хранить данные о пользователях или необходимость такой услуги для функционирования организации, например соц сети, где пользователи ожидают сохранения своих данных на длительный срок, для всего этого необходимы мощность для хранения данных. Но нередко к таким данным не нужен быстрый доступ, как например к исполняемым файлам, поэтому для увеличения вместительности можно использовать компрессию данных. В таком случае мы обмениваем быстродействие на вместительность. Степень такого обмена может варьироваться, например мы можем использовать сложный алгоритм компрессии для хранения фотографий на сервисе, но при их отправке клиенту мы столкнемся с необходимостью их декомпрессировать, что вызовет повышенную нагрузку на сервер, что замедлит работу нашего клиента. Поэтому создание приложения с гибкой настройкой возможности компрессии является актуальной для сферы задачей.

**Цель исследования:** создание программного обеспечения, на основе дерева Хаффмана, позволяющего компрессировать и декомпрессировать данные.

**Для достижения цели исследования поставлены следующие задачи**:

* Разобраться в теории работы алгоритмов компрессии, математически доказать их оптимальность;
* Составить модель работы программы для работы с архивами;
* Реализовать на основе составленных в модели алгоритмов приложение.

**Объект исследования**: серверы и персональные компьютеры.

**Предмет исследования**: архивация данных, представленных в электронном виде.

# ИСТОРИЯ РАЗВИТИЯ АВТОМОБИЛЕЙ И ИХ ОБСЛУЖИВАНИЯ

На данный момент автомобили с двигателями внутреннего сгорания до сих пор остаются актуальными, хотя и их эпоха подходит к концу. Все-таки услуги ремонта таких типов автомобилей наиболее распространены, поэтому в данной работе будем рассматривать именно их.

Первые автомобили, вышедшие в массовое производство, появились в 1888 году в Германии, основателем компании, производившей их был Карл Бенц. По его лицензии автомобили производились также во Франции и США. «Панар и Левассор» стала первой компанией, начавшей производить свои автомобили с 4-цилиндровыми двигателями. С 1888 года ближайшие 2 десятилетия появлялось множество различных компаний по производству автомобилей, производили бензиновые, дизельные и электрические двигатели, конкуренты использовали различные технологии. Примерно к 1910-м годам установилось преобладание бензиновых ДВС на рынке. В те далекие годы были разработаны, но, в силу недостатка технологий производства, не доведены до рабочего состояния многие современные разработки, например, полный привод, несколько клапанов в камере сгорания или гибридный двигатель, работающий и на газу, и на бензине. В принципе из-за отсутствия каких-либо стандартов производства автомобилей в то время изобретения появлялись неравномерно, и каждый производил все как считал нужным, так, например, некоторыми авто можно было управлять ручкой вместо рулевого колеса, также на большинстве моделей можно было ездить только с определенной скоростью (отсутствие коробки передач). Почти все машины имели кузов открытого типа, и для передачи крутящего момента чаще использовалась цепь, а не карданный вал.

Примерно к концу первой мировой войны начал формироваться некий стандарт вида автомобиля – задний привод, двигатель внутреннего сгорания расположен в передней части кузова, имелась шестеренчатая трансмиссия, а кузов стал более напоминать автомобильное купе, чем просто повозку без лошадей.

Существование множества компаний по производству авто, и пока отсутствие жесткой монополии на рынке позволило автомобильной промышленности быстро развивать новые технологии. Наиболее прорывными из них являлись система зажигания, работающая на электричестве, независимая система подвески, барабанные тормоза на все колеса, ступенчатые трансмиссии, позволявшие ехать с разной скоростью.

В так называемую Винтажную эру автомобилей, начавшейся с конца первой мировой до приблизительно 1930 года большую популярность приобрели автомобили с закрытым металлическим кузовом, стандартным управлением (рулевое колесо), передним расположением двигателя. Развитие последнего проходило в ускоренном темпе: флагманами выступали многоклапанные ДВС с верхним распредвалом, а также роскошные V-образные 8-, 12-, 16-цилиндровые мощные двигатели. В 1924 году Германном Рейслером была изобретена первая автоматическая двуступенчатая планетарная коробка передач с преобразователем крутящего момента. К сожалению, она так и не вышла в производство, зато ее более усовершенствованные аналоги приведут в будущем к развитию автоматических трансмиссий.

С 1930-х годов и до завершения второй мировой войны автомобили стали приобретать более элегантный и аэродинамичный вид кузова, лидирующие компании патентовали ранее используемые технологии механики, например передний привод был запатентован Андре Ситроеном, а независимая подвеска – компанией «Mercedes-benz».

В послевоенную эру автомобили стали больше походить на наши современные авто, которые мы знаем. В дизайне автомобилей тех годов стал преобладать полноценный цельнообъемный кузов, без выделяющихся крыльев, подножек, также полноценные багажники и капоты. Мощность двигателей с развитием технологий росла, так же как и скорость езды. Но с нефтяным кризисов 1973 года популярными стали более экономные и миниатюрные автомобили. К концу 20-го века становились все более распростаненными внедорожники и универсалы, а прорыными технологиями стали роторный двигатель и турбокомпрессоры. В СССР звание народного автомобиля занял Фиат-124 (ВАЗ-2101), став символом массовой автомобилизации.

С конца девяностых и по сей день продолжается так называемая современная эпоха авто, с 90-х годов в двигателях стали использоваться инжекторная система подачи топлива, а также компьютерное управление зажиганием. Это существенно повысило мощность и эффективность ДВС. Более того, появились системы помощи в управлении автомобилем, такие как противопробуксовочная система, система стабилизации заноса и ABS.

Большинство автомобилей в наше время имеют цельный несущий кузов, переднее расположение двигателя и передний привод. В плане модели преобладают хетчбеки, кроссоверы с небольшим объемом двигателя (до 3,0 литров). В последние годы набирают популярность гибриды или полностью электрические автомобили, и, если 10 лет назад это было новаторством, то сегодня многие автомобильные компании заявляют о полном переходе на электричество к 2030-м годам.

С развитием технологий автомобилестроения развивались и технологии ремонта автомобилей, так до 20-го века, ввиду большой роскоши и редкости машин, ремонтом занимался личный шофер или бригада механиков. Только в1910-х, после того как рынок авто добрался и до обычного потребителя, стали появляться и специальные организации по ремонту – автосервисы. Первые автомастерские появлялись на основе мастерских по ремонту велосипедов или тракторов. Здесь же машину и заправляли. Со временем начали появляться и специальные образовательные учреждения, обучавшие обслуживанию и ремонту авто.

Со временем происходило повышение квалификации рабочих, параллельно как развивались технологии оборудования, использовавшееся при ремонте. На заре автопромышленности механики использовали обычные разводные гаечные глючи и рычаги, затем гидравлические домкраты и насосы. Много позже появились электрические инструменты, такие как резчики по металлу, шуруповерты и гайковерты, которые сильно облегчали ручной труд. Повысилась и надежность ремонта с увеличением точности инструментов, таких как станция для настройки развала-схождения, электронной сварки и измерительных инструментов. К концу 20-го века начали появляться станки для шиномонтажа и компьютерные системы диагностики автомобиля.

Последние значительно сократили время, требуемое для нахождения неисправности автомобиля. При наличии на более-менее современных автомобилях компьютерного блока управления двигателем (ECU) и других систем электронной статистики и настройки, стало возможным определение неисправности ключевых систем авто в считанные часы. Также данные системы открыли широчайший потенциал для тюнинга двигателя. Так как ECU отвечает за управление такими процессами как зажигание, подача топлива и подача воздуха, в определенных пропорциях, можно расширить заводские настройки ДВС и таким образом увеличить мощность двигателя. Конечно, чаще всего это сильно влияет на ресурс самого двигателя, а также на другие главные системы автомобиля.

В наиболее современных моделях реализовано динамичное автоматическое управление подвеской и масштабирование мощности и крутящего момента, что позволяет владельцу получать максимум от производительности на треке или при движении по пересеченной местности, а также низкий расход топлива и комфортное передвижение на дорогах общего пользования.

В современном дилерском сервисе реализовано множество автоматических систем диагностики, заказа запчастей, а также бухгалтерского учета и учета человеко-часов. Автоматизация диагностики, например, позволяет менее квалифицированным рабочим также обеспечивать хорошее обслуживание и сервис. Специально разработанные глобальные базы данных производителей, таких как «Вольво», «Мерседес-Бенц», «Ауди» и др., предоставляют информацию об автомобиле через сервисное ПО и ПО самого автомобиля. На примере производителя «Вольво», где используется программное обеспечение «VIDA», при подключении автомобиля к сервисному компьютеру отображаются все основные данные автомобиля, такие как пробег, часы работы двигателя, VIN-номер и др., а также неисправности двигателя, проводки, подвески, различных датчиков температуры и в принципе всей электронной части автомобиля. Проанализировав множество данных с датчиков автомобиля, программа предоставляет коды ошибок и часто рекомендации по их устранению, либо замене запчастей. В программе присутствуют инструкции по сборке и разборке всех частей любой модели «Вольво» от 1990-х годов, воспользовавшись которыми более или менее квалифицированный механик может выполнить замену запчасти. Единственный недостаток данной программы – это отсутствие какой-либо информации по ремонту запчасти или целой системы автомобиля. Ведь программа выдает только коды ошибок, а уже их совокупность должен оценить специалист и принять решение по устранению неисправности, так что полной замены человеческого труда в диагностике быть пока что не может.

В этом программном обеспечении существует возможность обновления ПО самого автомобиля. Так как база данных глобальная, с серверов, расположенных в Швеции, загружается актуальная версия ПО для компьютеризированных систем автомобиля. Таким же образом и происходит загрузка настроек для чип-тюнинга двигателя, за дополнительную плату клиента. Установка чип-тюнинга занимает более длительное время, чем обновление, так как происходит автоматическая полная перенастройка всех систем автомобиля. После установки тюнинга также проводятся дополнительные тесты на работоспособность авто, и перенастройка датчиков подачи воздуха, топлива и др.. только после проверки безопасности и границ погрешностей происходит передача автомобиля клиенту.

Помимо этого, в программе присутствует функция автоматического формирования отчета по ошибкам для консолидации неисправностей в едином документе для клиента. В этот документ вносятся коды ошибок, рекомендации от специалиста и артикулы запчастей, подлежащих замене. Также предоставляется информация и VIN-номере, пробеге и модели автомобиля. Вся эта отчетность автоматически загружается в базу данных клиента, который пользуется услугами данного сервисного дилерского центра, где уже специалистами по работе с клиентами формируется полный отчет вместе со стоимостью запчастей и стоимостью услуг, осуществляемых механиком.

Однако существуют не только дилерские (официальные) ПО и базы данных. Мультимарочные базы данных включают информацию сразу по многим маркам автомобилей. Наиболее известными продуктами являются базы по диагностике и ремонту, а именно, Mitchell-on-Demand, BOSCH ESI, VIVID WorkShop, Alldata, Autodata, Atris WM-KAT-Technik, Open@Car, Tolerance data, CAPS, ATSG и др. Отдельно следует выделить каталоги запасных частей (EPC – Electronic Parts Catalog). В них содержится информация о запасных частях, их применимости, цене, взаимозаменяемости, внешнее изображение. Каталоги запчастей делятся на каталоги оригинальных (произведенных или рекомендованных автопроизводителем) и неоригинальных (произведенных сторонними производителями) запчастей.

На современном этапе развития ИС и технологий выделяют следующие компьютерные информационные системы, используемые на станциях технического обслуживания:

* ERP-система (Enterprise Resource Planning System – система планирования ресурсов предприятия) – корпоративная информационная система, предназначенная для автоматизации учета и управления.
* WMS (Warehouse Management System – система управления складом) – обеспечивает комплексную автоматизацию управления складскими процессами.
* MRP (Material Requirement Planning – планирование материальных потребностей) – методология, используемая для планирования производственных циклов и запасов: потребности в компонентах и материалах для планирования производства и доставки потребителям; поддержка низких уровней запасов; планирование закупочных операций, производственных операций, расписаний доставки.
* SCM (Supply Chain Management – управление цепочками поставок) – система управления всеми этапами снабжения предприятия и контроля товародвижения на предприятии, охватывающая весь цикл закупки сырья, производства и распространения товара: производство, поставки, месторасположение, запасы, транспортировка и информация

На сегодняшний день на рынке информационных продуктов представлено достаточно много готовых решений для реализации функций управления автодилерским предприятием.

Каждый программный продукт уникален в своем роде. В каждом можно найти достоинства и недостатки. Однако, все инструментальные средства достаточно долго находятся на рынке, получая обновления от разработчиков и совершенствуясь от версии к версии.

Суммируя вышесказанное, можно прийти к выводу о том, что в современном сервисе для эффективной работы механиков и в целом всего ремонтного предприятия невозможно обойтись без специализированных баз данных, ПО для диагностики, а так же ПО для учета склада и БД для работы с клиентами.

Рассмотрим устройство и функциональные возможности ERP-системы на примере 1С:ERP Управление предприятием.

У этого многофункционального ПО существует множество модулей:

* Мониторинг и анализ показателей деятельности
* Управление финансами и бюджетирование
* Казначейство
* Международный финансовый учет
* Управление продажами
* Управление взаимоотношениями с клиентами
* Управление закупками
* Управление складом и запасами
* Регламентированный учет
* Управление затратами и расчет себестоимости
* Управление персоналом и расчет заработной платы
* Управление производством
* Организация ремонтов
* Планирование запасов

Далее более подробно мы будем рассматривать модуль управления закупками, в нем реализуются следующие функции:

* подбор поставщиков товаров;
* различные условия закупок;
* поддержка различных схем приема товаров от поставщика;
* различные варианты формирования заказов поставщикам и контроль их исполнения;
* корректировка и закрытие заказов поставщикам;
* мониторинг цен поставщиков;
* оформление поставки товаров;
* составление графиков поставок и графиков платежей;
* корректировка поступлений и возвраты поставщикам.

Общую работу данного модуля можно представить в виде следующей блок-схемы:



Рисунок 1 – Блок-схема работы ПО управления закупками в 1С:ERP.

Изначально подразумевается, что существует некая база поставщиков, которую также можно ввести в программе, по каждому поставщику можно зарегистрировать действующие цены (прайс-лист поставщика):

* прайс-лист поставщика регистрируется в рамках соглашения с поставщиком;
* предусмотрена возможность регистрировать цены в соответствии с различными условиями.

Для регистрации условий закупок существует элемент соглашения об условиях закупок, в котором можно:

* указать поставщика;
* определить условия оплаты (порядок оплаты, этапы оплаты, использование договоров);
* указать срок поставки товаров — используется для расчета предполагаемой даты поставки;
* указать склад поставки товаров от поставщика и др.

Присутствует возможность оформления договора с поставщиком, который формируется на основе данных по соглашению.

Далее заказы поставщикам могут формироваться автоматически на основании расчета потребностей в обеспечении товарами в соответствии с потребностями в обеспечении, также заказ поставщику может быть введен на основании ранее оформленных заказов покупателей, сделки с клиентом.

Контроль оплаты товаров по заказам поставщиков производится в соответствии с датой оплаты, указанной в заказе поставщику.

Контроль выполнения обязательств поставщиком осуществляется при помощи статусов и состояний документа. Статусы для документа присваиваются вручную. Состояние — информация о заказе, которая рассчитывается автоматически, исходя из имеющихся в информационной базе данных (статус, оплата, наличие на складе и т. д.).

После всего этого поставленный товар может быть зарегистрирован на складе и отправлен, например, на хранение.

В данном модуле также предусмотрена и необходимость корректировки товаров, а также их возврат, так как может быть такое, что товары могут быть получены фактически после поступления документов и в неполном количестве, или поставщик пришлет исправленные документы с другой ценой за единицу.

Это одни из самых важных основных функций данного модуля ПО по управлению закупками,

Проведя анализ данного модуля ПО 1С:ERP Управление предприятием, я понимаю, что в своем решении буду реализовывать следующие функции:

* Указание поставщиков и доступных складов
* Реализация возможности заказа запчасти у поставщиков
* Регистрация доставленных запчастей на складе

# МОДЕЛЬ РАБОТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПО УПРАВЛЕНИЮ ЗАКУПКАМИ

Данная программа позволяет пользователю добавлять поставщиков, запчасти, принадлежащие им, с указанием количества запчасти. Также есть функция заказа, в которой реализуется поиск по названию запчасти, введенным пользователем, в списке поставщиков, затем пользователю предлагается меню из списка поставщиков, у которых имеется данная деталь, затем данные о выбранной запчасти добавляются на склад, и далее хранятся там. Пользователь в любой момент времени имеет доступ к добавлению поставщиков и деталей, а также к просмотру состояния склада. Также пользователь может сохранить данные, введенные во время работы программы в файлы. При следующем запуске данные будут загружены в программу, и пользователь сможет продолжить работу.

Программа помогает упростить ведение учета склада, информации о поставщиках на станциях технического обслуживания.

Для упрощения разработки приложения целесообразно составить модель его работы, и основываясь на ней реализовать полноценное решение. Принцип работы будущей программы для больших наглядности и понимания можно представить в виде блок-схем, где каждый модуль будет рассматриваться отдельно.

Целиком схема работы программы такова:



Рисунок 2 – Общая схема работы программы

В программе присутствуют 4 основных блока, в соответствии с рисунком 2:

* Блок, где указываются доступные поставщики, а также детали в наличии
* Блок составления заказа на доставку запчастей
* Блок, в котором отображается текущее итоговое состояние склада запчастей
* Блок, сохраняющий данные пользователя в файл

Эти функции действительно являются основными при разработке приложения для управления закупками и складом

Далее подробно будем рассмотрим каждый из 4 основных блоков по порядку, отдельно рассмотрев модель блока работы с поставщиками а также блока составления заказа.

## 2.1 МОДУЛЬ «РАБОТА С ПОСТАВЩИКАМИ»



Рисунок 3 – Модуль работы с поставщиками – общее представление

В данном модуле происходит основной ввод информации для работы программы, пользователь указывает сначала имя каждого поставщика, затем запчасти, имеющиеся у него в наличии, а также их количество.

Пользователь выбирает нужный пункт меню, либо «добавить поставщиков», либо «добавить запчасти».

При выборе «добавить поставщиков» предлагается ввести нужное количество поставщиков, затем пользователь вводит поставщиков по одному.

При выборе «добавить запчасть» пользователю предлагается ввести название запчасти, затем ее количество, и предлагается выбор поставщика, к которому добавить эту запчасть.

Затем предлагается ввести еще одну запчасть, либо переход к основному меню.

## 2.2 МОДУЛЬ «ДЕТАЛИ ЗАКАЗА»



Рисунок 4 – Детали заказа – общее представление

В следующем модуле пользователю предлагается ввести название необходимой ему запчасти, затем, если запчасть есть в наличии, пользователю предлагается список поставщиков, у которых есть данная запчасть, вместе с ее названием и количеством.

Пользователь может выбрать одного из поставщиков, и затем данная запчасть будет добавлена на склад, а запись о наличии данной детали у поставщика будет стерта.

Если же запчасти нет в наличии, то будет выведено сообщение о том, что ее нет в наличии ни у одного из поставщиков. И произойдет переход к основному меню.

## 2.3 МОДУЛЬ «СКЛАД»

В данном модуле пользователю предоставляется отчет о деталях, находящихся на складе. В консоль выводится название запчасти и ее количество на складе.

## 2.4 МОДУЛЬ «СОХРАНЕНИЕ»



Рисунок 4.1

При выборе в главном меню пункта «сохранить» происходит сохранение данных о записанных поставщиках (имя, название запчасти, ее количество) в XML-файл.

Также сохраняются данные о деталях на складе (название, количество) тоже в отдельный XML-файл. Если данные уже были записаны, то они перезаписываются заново, отражая все пользовательские изменения.

При запуске программы, если данные 2 файла существуют, то из них будет загружена информация о поставщиках и складе, если нет, то они будут созданы при сохранении. В случае, если пользователь во время работы программы не выберет пункт главного меню «сохранить», при завершении работы программы данные не будут сохранены.

# РЕАЛИЗАЦИЯ ПРОГРАММЫ НА ЯЗЫКЕ C#

На основе проведенного исследования и разработки модели будущей информационной системы мной было принято решение разработать консольное приложение на языке c#. Применяя рациональный подход в разработке, я поделил программу на основные блоки – методы: «Menu», где происходит вывод основного меню программы, а также ввод пользователем выбранного пункта, «Post», в котором пользователь может добавлять поставщиков и запчасти, «order», в котором пользователь может заказать запчасть, «warehouse», в котором пользователь может посмотреть состояние склада, «saveUser», который позволяет пользователю сохранить данные о поставщиках и складе.

Программа осуществлена в одном программном файле, в одном классе «Program» с использованием методов. В основном методе «Main», с которого начинается работа программы, после загрузки пользовательских файлов происходит вызов основного метода программы – «Menu». Этот метод отвечает за вывод меню программы, а также переход к другим методам. Каждому пункту меню в программе соответствует свой метод. При завершении работы одного из методов, вызываемых внутри «Menu», происходит возврат к основному методу «Menu», откуда пользователь далее выбирает пункты меню и так повторяется, пока не будет выбран выход из программы при помощи return. Также внутри основных методов, реализующих работу с поставщиками, складом, и отвечающих за сохранением в файлы, вызываются вспомогательные методы для записи данных, их проверки. В каждом методе или блоке, где присутствует пользовательский ввод с клавиатуры, осуществляется обработка ошибок, а также проверка правильности ввода данных во избежание критической ошибки в работе программы.

Далее рассмотрим ключевые фрагменты кода, которые помогут более подробно понять устройство программы. Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 –метод «Main»

В методе «Main», где есть начало выполнения самой программы, сначала проходит проверка на существование пользовательских файлов, как видно на рисунке 5 – «ware.xml» и «user.xml». Если такие файлы существуют, то есть были созданы при использовании программы ранее, то информация из них будет загружена в специальные списки, хранящие информацию о поставщиках и о складе. Об устройстве этих списков будет сказано ниже. Рассмотрим подробнее методы и поля, используемые здесь.

Так как загрузка информации в списки из этих файлов происходит практически идентичным образом, рассмотрим, как происходит загрузка файла «user.xml».

Итак, если данный файл существует в корневой папке приложения, создается переменная «userLoad», в которую прочитывается документ целиком. Далее создается переменная «root», которой присваивается название корневого элемента файла – «data». Далее при помощи оператора «foreach» происходит считывание информации с каждого элемента под названием «dealer» в корневом элементе «data» следующим способом. Локальной переменной «n» присваивается значение атрибута «name» элемента, переменной «р» присваивается значение элемента «part», и затем переменной «q» присваивается значение элемента «quantity». Далее значения этих переменных добавляются в список main как одна запись, соответственно, название, имя запчасти, ее количество. Таким же образом происходит загрузка и файла «ware.xml», с отличием только в том, что запись происходит в отдельный список, в котором в записи указывается только имя запчасти и ее количество.

Далее происходит вызов метода «Menu», который и осуществляет основную работу программы.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 – метод «Menu»

В данном методе происходит вывод пользовательского текстового меню на экран консоли с помощью «Console.WriteLine()». Затем идет конструкция обработки возможных ошибок ввода «try» – «catch» с оператором «if». Далее введенное пользователем значение сохраняется в переменную «input\_1», и, в зависимости от введенного значения, вызывается метод, соответствующий каждому пункту меню. Метод «Post» отвечает за работу с поставщиками, метод «order» отвечает за оформление заказа, метод «warehouse» отвечает за вывод содержимого склада на экран консоли, метод «saveUser» отвечает за сохранение пользовательских данных в файлы, а в случае, если пользователь захочет завершить программу и введет 5, произойдет выход из основного метода «Menu» с помощью оператора return, и программа завершится.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7.1 – метод «Post» - часть 1

В методе «Post» осуществляется вывод меню для работы с поставщиками (добавление поставщиков, добавление запчастей). Введенное пользователем значение записывается в переменную input\_2, присутствует обработка неверно введенных значений. Далее в зависимости от введенной пользователем команды происходит следующее: если пользователь ввел 1 (добавить поставщиков), то далее пользователю будет предложено ввести желаемое количество добавляемых поставщиков, и введенное значение присваивается переменной «dealerNum» (присутствует обработка ввода ошибочного значения). Далее с помощью метода «dealerAdd» с аргументом «dealerNum» происходит добавление поставщиков в список «main». После выполнения метода «dealerAdd» происходит переход к основному меню («Menu»).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7.2 – метод «dealerAdd»

Метод, представленный на рисунке 7.2, принимая целое значение на вход, выводит на экран консоли сообщение с просьбой ввести имя поставщика, затем, с помощью цикла «for», пользователю поочередно предлагается ввести имя каждого дилера. Введенное значение присваивается переменной «s», затем в список «main» добавляется запись. После завершения цикла будет выведено сообщение о том, что поставщики были добавлены.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7.3 – метод «Post» - часть 2

Рассмотрим 2 часть метода «Post», представленную на рисунке 7.3. Если пользователем ранее (рисунок 7.1) было введено значение 2 (добавить запчасти), будет вызван метод «partsAdd», который осуществляет создание записи определенного дилера уже с добавлением запчастей (название, количество), а затем пользователю будет предложено выбрать, хочет ли он добавить еще одну запчасть, если да, то цикл повторится, в обратном случае метод «Post» завершается, и происходит переход к основному меню.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7.4 – метод «partsAdd»

Метод «partsAdd», представленный на рисунке 7.4, осуществляет создание пользовательских записей в списке «dealers». Пользователю предлагается вести название запчасти, введенное значение присваивается переменной «s», затем пользователю предлагается ввести количество запчасти (в наличии), введенное значение присваивается переменной «q», после этого с помощью метода «dealerDisplay» происходит вывод доступных дилеров в консоль, и пользователь выбирает, к какому из введенных ранее дилеров добавить запчасть. После этого происходит добавление записи в список «main» с переменными «SS», «s», «q», соответственно название дилера, название запчасти, ее количество.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7.5 – метод «dealerDisplay»

Для того, чтобы из списка «main» пользователю не выводились на экран повторяющиеся названия дилеров, а только уникальные, используется метод, представленный на рисунке 7.5. внутри метода создается список «names», в который записываются имена дилеров из всех записей списка «main». Затем создается строковый массив «names\_1», и с помощью полей «.Distinct().ToArray()» в этот массив добавляются только уникальные записи списка «names», затем они выводятся на экран, вместе с индексом, а пользователю предлагается выбрать из списка дилеров. Выбранное имя записывается в глобальную переменную «SS» для использования в другом методе («partsAdd»).

Перейдем к рассмотрению следующего основного метода, вызываемого в «Menu»

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8 – метод «order»

В методе, представленном на рисунке 8, как было сказано ранее, осуществляется «оформление заказа». Пользователю предлагается ввести название необходимой запчасти, и, если она имеется хотя бы у одного из поставщиков, запись будет выведена на экран, и пользователь сможет «заказать» эту запчасть.

Если имя запчасти соответствует имени запчасти в одной или нескольких записях в списке main, в консоль выводится индекс записи в списке main, далее имя дилера, запчасти, количество запчасти. Для выбора нужного дилера пользователю предлагается ввести индекс одной из выведенных записей, затем в список ware добавляется запись о «заказанной» запчасти, ее имени и количестве. Запись, ранее содержащая эту информацию в списке main, удаляется с помощью метода «RemoveAt».

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 – метод «warehouse»

Метод, представленный на рисунке 9 отвечает за вывод содержимого склада, то есть заказанных запчастей, на экран консоли. В методе с помощью оператора цикла «for» осуществляется вывод каждой записи списка «ware»: название запчасти и ее количество.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 – метод «saveUser»

Данный метод, представленный на рисунке 10, вызываемый также из главного метода «Menu», осуществляет сохранение данных пользователя в XML-файлы.

Создается переменная типа XDocument – «user», создается корневой элемент документа – переменная «root» с именем «data» далее для каждой записи списка «main» создается элемент dealer, у него создается атрибут «name», ему присваивается значение имени дилера из списка. Далее внутри элемента dealer создаются элементы «part» и «quantity», в которые записывается значения имени детали и ее количества соответственно. Наконец элемент dealer добавляется к корневому элементу data, и так происходит для каждой записи main. После этого элемент «data» добавляется в переменную user, и документ сохраняется под названием «user.xml». аналогично происходит сохранение файла «ware.xml».

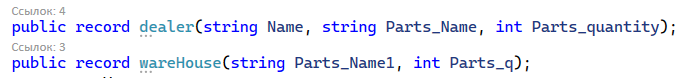


Рисунок 11 – record «dealer» и record «wareHouse»

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11.1 – используемые глобальные переменные класса Program

Конструкции, представленные на рисунке 11, использовались для определения шаблонов записи списков main и ware. Запись в списке main поделена на 3 части: имя, имеющее строковый тип, название запчасти, тоже строковый тип, количество запчасти, имеющее тип «int». Запись в списке ware поделена на 2 части: аналогично название запчасти, имеющее строковый тип, количество запчасти, имеющее тип «int».

Приведя все основные функциональные блоки программы, целесообразно перед ее применением провести тестирование. Его результаты можно просмотреть в приложении А. После проведенного тестирования можно сделать вывод о том, что приложение работает исправно, обработка ошибок работает как и ожидалось, а запись в файл происходит в соответствии с введенными пользовательскими значениями. В целом программа функционирует исправно.

Так как все основные требования к программному коду были выполнены, что означает то, что в программном коде присутствуют конструкции:

* операторов условия;
* операторов множественного выбора;
* циклов со счётчиком;
* циклов с предусловием;
* циклов с постусловием;
* методов с возвращаемым результатом;
* методов, использующих параметры, передаваемые по ссылке;
* методов использующих переменное количество параметров;
* предусмотреть в программном коде работу с файлами;
* чтение исходных данных из файла и с консоли;
* запись результирующих данных исполнения программы в файл;
* документирующие комментарии (///).

Так как программа в состоянии корректно обрабатывать исходные данные из файла, ввода пользователя; все ее основные функции прошли тестирование; и в итоге она была успешно скомпилирована в MS Visual Studio 2022, можно сделать итоговый вывод о том, что разработанное решение работает корректно и справляется с задачами, поставленными в первой главе при помощи функциональных блоков – методов «Menu», «Post», «order», «warehouse». Таким образом поставленные во введении данной курсовой работы задачи были выполнены.

# ЗАКЛЮЧЕНИЕ

На сегодняшний день практически ни одно успешное предприятие не обходится без цифровых решений в ведении бизнеса. Это же и относится к сервисам, станциям технического обслуживания. Ведь использования программного обеспечения по управлению закупками, складами, предприятием, сильно облегчает работу бухгалтеров, а также позволяет управляющим лицам компании получать более точные отчеты и, следовательно, планировать лучший бизнес-план для развития организации. Вдобавок такое решение может облегчить жизнь и самим механикам, ведь работая с цифровизированным складом, когда можно в секунду понять сколько какой запчасти имеется в наличии, легче составлять ремонтный план для клиента.

В рамках данной курсовой работы была подробно рассмотрена и изучена история развития сервиса, пример устройства конкурентного решения. Была создана модель работы программного обеспечения и на ее основе реализовано консольное приложение, выполненное на языке c#.

Данное приложение быстро и стабильно работает, выполняет все соответствующие ему функции, интерфейс интуитивно понятен, а само приложение тратит минимум ресурсов ПК. Из недостатков можно выделить привязанность к пользовательским данным, у программы нет функции обращения к серверу, следовательно глобально применить это решение на нескольких ПК не удастся. Но тем не менее приложение все же помогает вести учет закупок и склада, а при некоторой доработке может полноценно использоваться на предприятиях. Автор не видит поводов сомневаться в перспективе применения данного решения на любых станциях технического обслуживания или дилерских центрах.

# СПИСОК ЛИТЕРАТУРЫ

Шумаков Д. С., Зимина Л. В. – Особенности автоматизированных информационных систем автодилерских предприятий // Экономическая среда. - 2016. - №2. - С. 53-56.

История автомобиля [Электронный ресурс] // Википедия - Режим доступа: https://ru.wikipedia.org/wiki/История\_автомобиля, свободный.

1С:ERP Управление предприятием [Электронный ресурс] // 1С:Предприятие 8 - Режим доступа: https://v8.1c.ru/erp/purchasing/, свободный .

Полное руководство по языку программирования С# 10 и платформе .NET 6 [Электронный ресурс] // METANIT.COM - Режим доступа: https://metanit.com/sharp/tutorial/, свободный.

Документация по .NET [Электронный ресурс] // режим доступа: https://docs.microsoft.com/ru-ru/dotnet, свободный.

Уроки программирования [Электронный ресурс] // Режим доступа: https://vscode.ru/prog-lessons, свободный.

Документация C# [Электронный ресурс] // github - Режим доступа: https://github.com/dotnet/docs/tree/main/docs/csharp, свободный.

Авторемонтное предприятие [Электронный ресурс] // Википедия - Режим доступа: https://ru.wikipedia.org/wiki/Авторемонтное\_предприятие, свободный.

ОСНОВЫ АЛГОРИТМИЗАЦИИ [Электронный ресурс] // Режим доступа: https://de.unecon.ru/pluginfile.php/592618/mod\_resource/content/1/методичка.pdf, закрытый.

СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ [Электронный ресурс] // ГОСТ - Режим доступа: https://docs.cntd.ru/document/9041994, свободный.

# ПРИЛОЖЕНИЕ А

Изображение выглядит как текст

Автоматически созданное описание





Рисунок 12 – цикл работы программы



Рис 13 – «ware.xml»