Краевое государственное бюджетное

профессиональное образовательное учреждение

«Барнаульский государственный педагогический колледж»

Курсовая работа

РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ «Автомастерская»

Отделение гуманитарного образования и информационных технологий

Информационные системы и программирование

Выполнил: Бахаев Д.Е.

студент группы 483

Руководитель: Пархатская А.М.

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

(подпись)

Барнаул 2023

# Введение

# В течение последнего десятилетия в нашей стране неуклонно растет спрос на специальности, связанные с сервисом.

# Сервис - это особый вид человеческой деятельности, который направлен на удовлетворение потребностей клиента путем оказания услуг, востребованных отдельными людьми, социальными группами или организациями.

# Потребности человека разнообразны и образуют сложно структурированную систему. Сервисная деятельность также разделяется на большое количество направлений (в частности станции технического обслуживания). Структура сервисной деятельности должна в общих чертах повторять структуру потребностей, соответствовать ей.

# С тех пор как появились первые механические устройства возникла необходимость в их обслуживании и ремонте. Если телегу крестьянин обслуживал сам, то за римской боевой колесницей следили до трех человек, каждый из которых был узким специалистом: ходовая, оружие, кузов. Изобретение механических двигателей, вначале паровых, а позже и внутреннего сгорания и стремительное развитие последних привело к появлению автомобиля в том виде, в котором мы привыкли видеть сейчас.

# Усложнение конструкции автомобиля повлияло на создание специализированных авторемонтных станций. Неоспоримым фактом является то, что появилась жесткая конкуренция между авторемонтными станциями. Необходимым стало совершенствовать ремонтные технологии. Начали появляться компании, которые предлагали специнструмент, материалы и оборудование.

# В данной курсовой рассмотрены услуги, предоставляемые на СТО, приведены примеры клиентской базы, разновидности услуг предоставляемые на СТО, отчёт по деталям которые хранятся на складе, прайс-листы с ценой и наименованием услуг, фамилиями мастеров - приемщиков принимающих автомобили на приём и автослесарей, производящих работу.

Проблема исследования:

Объект исследования: Процесс разработки программных модулей для компьютерных систем.

Предмет исследования: Разработка программного модуля «Автомастерская».

Цель исследования: Разработать программный модуль «Автомастерская».

Для достижения поставленной цели необходимо решить следующие задачи:

1. Проанализировать предметную область «Автомастерская» и рынок программных продуктов, используемых в Автомастерских.

2. Проанализировать и выбрать инструментарии разработки программного модуля «Автомастерская».

3. Спроектировать и реализовать программный модуль «Автомастерская».

## 1. Описание предметной области «Автомастерская»

Автозапчасти либо покупают у российских дилеров, либо завозят самостоятельно, сотрудничая с поставщиками из стран Азии. Что касается мелких автосервисов, то самостоятельно ввозить запчасти для них довольно рискованно, ведь детали приходится закупать на перспективу. Кроме того, когда речь идет о небольших партиях, рассчитывать на скидки от производителя не приходится. Важно точно учитывать сезонность и своевременность закупок: наличие товара на складе не только сократит время обслуживания каждого клиента, но и ускорит оборачиваемость средств.

Для специализированных автосервисов, где список необходимых деталей четко определен нормативами авто-производителей, проблема доставки запчастей решается проще. Существует две наиболее распространенные модели закупки запчастей:

- закупка деталей «с запасом». Требует значительных затрат, но зато снимает вопросы отсутствия необходимых запчастей в нужный момент. Наиболее эффективна такая схема для сетевых автосервисов, в работе которых рано или поздно оказывается востребованной практически любая деталь.

- работа со службами доставки. Все детали по мере необходимости заказывают в каком-нибудь Интернет-магазине. Но тогда их стоимость оказывается более высокой.

При любом из указанных выше способах поставок запчастей разрабатываемая нами база данных должна содержать список контрагентов с возможностью его редактирования и дополнения новыми поставщиками. Данный список включается в перечень справочников, содержащихся в нашей системе. При выборе нужной запчасти программный модуль в автоматическом режиме подключит необходимого поставщика по связующему ключу.

Станциям технического обслуживания (СТО) следует сконцентрироваться на продвижении наиболее рентабельных видов услуг. Как правило, это кузовной ремонт, рихтовка и покраска автомобилей, шиномонтаж, мойка. Стоит продумать и максимально эффективное их сочетание. Хорошую отдачу дает комбинирование мастерской по кузовному ремонту и лаборатории по подбору автоэмалей: клиент, купивший эмаль, зачастую заинтересован в качественном ремонте и покраске автомобиля, а автовладельцы, уже сделавшие ремонт, обычно заказывают покраску своей машины.

Повысить эффективность работы автосервиса можно за счет увеличения потока клиентов и снижения расходов. Для этого можно предпринять следующие шаги:

- расширить спектр оказываемых услуг;

- минимизировать затраты на приобретение оборудования за счет размещения заказов на осуществление тех или иных работ в специализированных компаниях);

- сократить арендные платежи (можно отказаться от бокса и ограничиться обычным вагончиком);

- использовать рекламу только в краткосрочном периоде, а затем переключаются на «ползучий маркетинг» -- новый клиент приходит по рекомендации старого и т.д.

Слесарный ремонт проводится на базе современно оснащенного предприятия, потенциал которого позволяет выполнить весь спектр работ. При ремонте подвески автосервис использует сертифицированные запасные части и расходные материалы. На все работы предоставляется гарантия.

При проведении кузовного ремонта автосервис использует современное оборудование, с помощью которого возможно устранение любых дефектов. Кроме восстановления геометрии кузова и покраски автомобиля в автосервисе возможно провести локальный кузовной ремонт поврежденной детали.

На профессиональном уровне проводят ремонт и обслуживание автоматической коробки переключения передач (АКПП). При проведении работ, связанных с устранением дефектов АКПП используются сертифицированные запчасти и расходные материалы.

В ряде случаев, когда повреждение носит незначительный характер, используется технология локального ремонта. Локальный ремонт осуществляется, когда имеют место царапины, сколы и неглубокие вмятины. В этом случае ремонту подвергается непосредственно место повреждения, а не полностью деталь.

При выполнении кузовных работ осуществляется контроль геометрии кузова. Геометрия кузова проверяется по контрольным точкам, которые заложены в программном обеспечении измерительной системы и заданы заводом производителем. При нарушении геометрии кузова его восстановление осуществляется с помощью современных стапелей.

Регулировка углов схождения колес проходит с помощью электронного стенда сход развала. Сход развала выполняется на замкнутой системе, что дополнительно позволяет определить состояние геометрии кузова. С целью обеспечения высокого качества работ сотрудники СТО регулярно проводит юстировку стенда сход развала.

СТО предоставляет услугу по замене ремней ГРМ на двигатели автомобилей. Как правило, замена ремня ГРМ происходит при проведении большого ТО и связана с поддержанием в работоспособном состоянии газораспределительного механизма. Для правильной замены ремня ГРМ используется специальный инструмент, наличие которого гарантирует соблюдение технологии и высокое качество работ.

Замену масла проводится по специальной программе "Экспресс-замена масла". Для этой цели существует специальный прайс-лист, при котором замена масла производит за 350 рублей. При проведении данной услуги используются сертифицированные запчасти и расходные материалы ведущих производителей. К которым относятся масла Mobil, Castrol, Liqui Moly. Вся процедура проводится в кратчайшие сроки.

## 2. Анализ программного приложения «Автомастерская»

В курсовом проекте необходимо разработать программное обеспечение для пользователя, позволяющее автоматизировать обработку данных базы данных для автосервиса с использованием языка программирования высокого уровня С# и Access.

В приложении должны быть реализованы следующие функции:

- вывод содержимого базы данных;

- добавление записей в базу данных;

- редактирование записей в базе данных;

- удаление записей из базы данных;

- поиск записей в базе данных.

Также должен быть предусмотрен вывод сообщений об ошибках в случае ввода неверных команд или некорректных данных пользователем. Необходимо разработать удобное меню управления функциями приложения, которое обеспечивало бы удобную работу пользователя. Меню управления, по сути, представляет собой консольное окно приложения и реализует функциональность перечисленных выше структурных элементов приложения.

## 3. Этапы и средства разработки программных модулей

Процесс разработки программного обеспечения (англ. software development process) — процесс, посредством которого потребности пользователей преобразуются в программный продукт. Существует несколько моделей такого процесса, каждая из которых описывает свой подход, в виде задач и/или деятельности, которые имеют место в ходе процесса. Однако во всех этих моделях основные этапы разработки схожи.

Основные этапы разработки (жизненный цикл) программного обеспечения:

1. Постановка задачи.
2. Анализ требований и разработка спецификаций.
3. Проектирование.
4. Реализация.
5. Внедрение и эксплуатация.

Работа над программным обеспечением начинается с составления документа, называемого «Техническое задание», который разрабатывается на первом и втором этапах жизненного цикла программного обеспечения.

На этапе проектирования разрабатывается проект архитектуры программных средств, определяются внутренние и внешние интерфейсы, устанавливается соответствие между требованиями и программным проектом.

На этапе реализации выбирается средство и инструменты с помощью которого разрабатываете продукт. Проводятся тестовые испытания, в случаях необходимости проводится отладка некорректно работающих процедур и функции программного модуля.

На этапе внедрения и эксплуатации продукт сдается заказчику. В случае необходимости организуется поддержка и сопровождение разработанного программного модуля.

Средства для создания приложений – локальные средства, обеспечивающие выполнение отдельных видов работ по созданию программ, делятся на:

* языки и системы программирования;
* инструментальная среда пользователя.

Средства разработки программного обеспечения:

* Lazarus — открытая среда разработки программного обеспечения на языке Object Pascal для компилятора Free Pascal. Основное преимущество Lazarus – это способность создавать исполняемые программы для различных операционных систем. В отличие от Java и т.п. Lazarus создаёт машинный код. Такой код не нуждается в интерпретаторе и работает в десять раз быстрее, чем промежуточный код.

Основное преимущество Lazarus является и его основным недостатком. Так как программы на Lazarus переводятся в машинный код, то необходимо каждый раз компилировать программу для различных операционных систем.

* PyCharm — интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django.

Плюсы: Прекрасное автодополнение кода, возможность работы с базами данных, возможность работы с кодом, находящимся на удалённой машине.

Минусы: Программа является платной, встречаются баги, в программе отсутствует русский язык

* Delphi — императивный, структурированный, объектно-ориентированный, высокоуровневый язык программирования со строгой статической типизацией переменных. Основная область использования — написание прикладного программного обеспечения.

Плюсы: Простота, скорость и эффективность. Delphi имеет один из самых быстрых компиляторов. Программы, написанные на Delphi, не требуется снабжать дополнительными библиотеками.

Минусы: Статическое присоединение (linking) библиотеки VCL и компонентов к исполняемому файлу. В Delphi вся информация о форме, включая свойства, настройки компонентов, значения по умолчанию, хранится в exe-файле, причем не оптимальным образом. Не хватает очень многих мощных средств C++. Малое число параметров оптимизации кода.

* Visual Studio — Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

Плюсы: Встроенный Web-сервер, поддержка множества языков при разработке, интуитивный стиль кодирования, высокая скорость разработки, возможности отладки.

Минусы: Отсутствие возможности отладчика отслеживать в коде режима ядра.

* SharpDevelop — свободная среда разработки для C#, Visual Basic .NET, Boo, IronPython, IronRuby, F#, C++. Обычно используется как альтернатива Visual Studio .NET.

Плюсы: Бесплатная программа, быстродействие, поддержка большинства языков, размер занимаемый на персональном компьютере, программа интуитивно понятна.

Минусы: Отсутствует мощный встроенный отладчик.

Для разработки модуля была выбрана среда разработки Visual Studio C#, так как это одна и лучших программ. Данный продукт позволяет разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, .NET Framework, Xbox и другие. Visual Studio включает в себя редактор исходного кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня.

## 4. Проектирование программного модуля «Автомастерская»

Прототип программного обеспечения — этап разработки программного обеспечения (ПО), процесс создания прототипа программы макета (черновой, пробной версии) программы, обычно с целью проверки пригодности предлагаемых для применения концепций, архитектурных и технологических решений, а также для представления программы заказчику на ранних стадиях процесса разработки.

Прототип позволяет также получить обратную связь от будущих пользователей, причем, именно тогда, когда это наиболее необходимо: в начале проекта еще есть возможность исправить ошибки проектирования практически без потерь.

Типы прототипирования.

Быстрое прототипирование. Основное преимущество такого подхода — в скорости: в ответ на свои требования заказчик почти сразу получает прототип интерфейса, и сразу может уточнить требования, до того как начато написание рабочего кода системы. Стоимость изменения требований на этом этапе очень низкая, поскольку нет кода, который нужно было бы переписывать. Быстрое прототипирование не обязательно выполняется в рамках той же платформы и тех же технологий, что и разрабатываемая система.

Эволюционное прототипирование. Такой подход имеет то преимущество, что на каждом шаге мы располагаем работающей системой, пусть и не обладающей всей нужной функциональностью, но улучшающейся с каждой итерацией. При этом, не тратятся ресурсы на код, который будет «выброшен». Эволюционный подход к прототипированию может быть выбран, исходя из предположения, что все необходимые требования к моменту начала разработки неизвестны, и будут определяться по мере создания программы; тогда на каждом этапе мы реализуем лишь те требования, которые известны и ясны. Иногда при этом разработчики сосредотачиваются на работе только над теми модулями системы, требования на которые уже определены. В некоторых случаях, когда речь идет о продукте под определенную незанятую нишу, пользователи начинают использовать систему еще до того, как она полностью дописана, в ожидании готовой системы, поскольку «недописанная система — это лучше, чем её полное отсутствие».

Основными преимуществами прототипирования являются сокращение времени и стоимости разработки за счёт того, что оценка прототипа позволяет на более ранних стадиях обнаружить недостаточность или несоответствие требований. Чем позднее проводятся изменения в спецификации, тем они дороже, поэтому уточнение «чего же пользователи/заказчики хотят на самом деле» на ранних стадиях разработки снижает общую стоимость.

Психологически важную роль играет также вовлечение заказчика в процесс разработки. Работа с прототипом позволяет будущим пользователям увидеть, как будет выглядеть будущая программа, и повлиять на её поведение, что уменьшает расхождения в представлении о программе между разработчиками и пользователями. Снижается и эффект почти неизбежного отторжения новой системы при внедрении, особенно когда она внедряется на место ранее использовавшейся.

Входе анализа предметной области было разработано техническое задание (приложение 1), в котором описан функционал системы.

Основой данного программного модуля должна быть реляционная база данных, которая должна хранить следующие сведения:

* информацию о товарах;
* информацию о поставщиках;
* информацию об покупателей;
* информацию о заказах;
* информацию о складе;
* информацию о кассе;
* информацию о категории поставщиков;
* информацию о странах;
* информацию о видах операций;

Данное приложение должно быть установлено на автоматизированное рабочее место пользователя и обеспечивать ему информационную поддержку в таких вопросах, как поиск автозапчастей для клиентов по необходимым параметрам, поиск телефона поставщиков.

Прототип интерфейса приложения представлен на рисунках

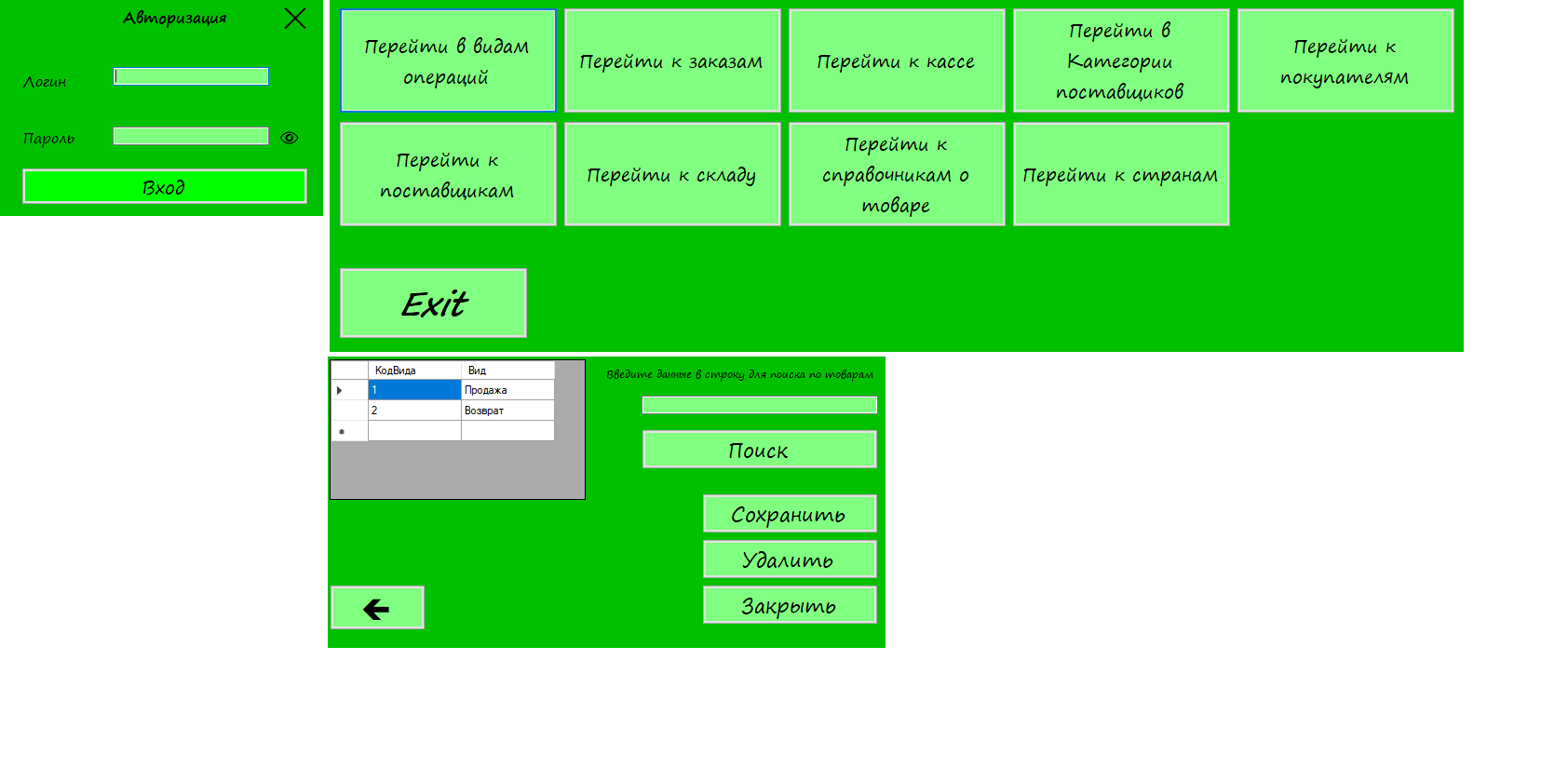


Рисунок 1 – Прототип интерфейса программного модуля «Автомастерская»

# 5. Реализация программного модуля «Автомастерская» с помощью объектно-ориентированный язык программирования С#

Разработка программного модуля, научилась с создания базы данных. База данных создана с помощью в Microsoft Access и подключена через «Источник данных» к нашему проекту Visual Studio C#. Структура базы данных представлена на рисунке 2.

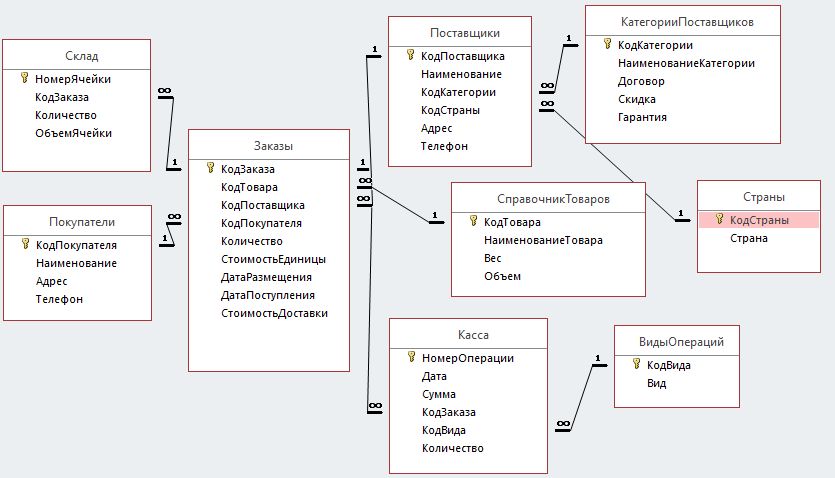


Рисунок 2 – Схема БД

Исходя из нашего приложения «Источником данных» выбран файл базы данных Microsoft Access (OLE DB) с обязательным расширением .mdb. При выборе объектов базы данных нами были использованы только «Таблицы».

В соответствии с прототипом нами были разработаны формы: окно авторизации, Логин , Пароль ;Glav где находятся все таблицы

При запуске приложения открывается окно авторизации (Form1 с названием «Авторизация»). На этой форме расположены элементы button, groupBox и 3 label в которых указана информация о фирме (рисунок 3).

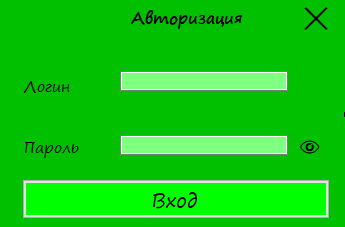


Рисунок 3 – Форма авторизации

При клике на элемент button с названием «Войти в систему» открывается новая форма в которой находится элементы label,textbox,button. В них добавлены 4 label, 2 textBox и 1 button. В textbox мы должны ввести логин и пароль чтоб войти в систему.

Код кнопки button «Вход»:

if(textBox1.Text == "1" & textBox2.Text == "1")

{

glav gl = new glav();

gl.Show();

this.Hide();

}

else

{

MessageBox.Show("Не правильно введен логин или пароль");

}

Форма ввода логина и пароля представлена на рисунке 4.

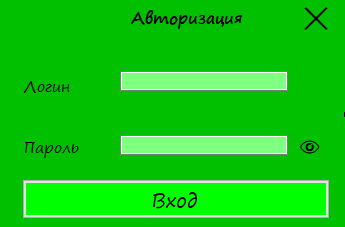


Рисунок 4 – Форма для ввода логина и пароля

На форме Password находится кнопка, с помощью которой можно как отобразить введенный нами пароль, так и скрыть (рисунки 5 и 6).



Рисунок 5 – Пароль скрыт.



Рисунок 6 – Пароль отображен.

Код отображения и скрытия пароля представлен ниже

{

if (vis)

{

textBox2.UseSystemPasswordChar = false;

vis = false;

}

else

{

textBox2.UseSystemPasswordChar = true;

vis = true;

}

}

После того как введен логин и пароль открывается новая форма Glav. На ней находятся 10 кнопок button для перехода по таблицам: «Перейти в видам операций», «Перейти к заказам», «Перейти к кассе», «Перейти к категории поставщиков», «Перейти к покупателям», «Перейти к поставщиком», «Перейти к складу», «Перейти к справочникам товаре», «Перейти к странам», (рисунок 7).

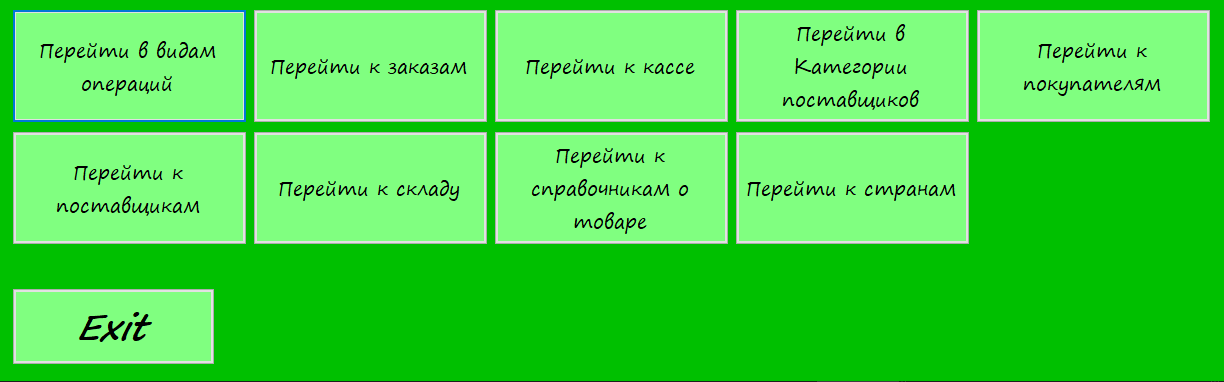


Рисунок 7 – Вид формы «Glav»

Код перехода по формам представлен ниже.

Form2 видыопераций = new Form2();

видыопераций.Show();

this.Hide();

Form3 заказы = new Form3();

заказы.Show();

this.Hide();

Form4 касса = new Form4();

касса.Show();

this.Hide();

Form6 категория\_поставщиков = new Form6();

категория\_поставщиков.Show();

this.Hide();

Form7 покупатели = new Form7();

покупатели.Show();

this.Hide();

Form8 поставщики = new Form8();

поставщики.Show();

this.Hide();

Form9 склад = new Form9();

склад.Show();

this.Hide();

Form10 Справочники\_о\_товаре = new Form10();

Справочники\_о\_товаре.Show();

this.Hide();

Form11 страны = new Form11();

страны.Show();

this.Hide();

Application.Exit();

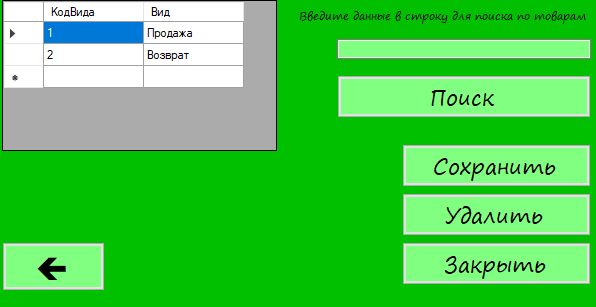
Пользователь может перейти на любую из девяти предложенных таблиц. Если нажать на кнопку «Перейти в видам операций», то откроется форма Form2, на ней находятся 1 label, 5 button, 1 textBox и dataGridView в которой представлена таблица с видами товаров (рисунок 8). 

Рисунок 8 – вид формы «Form2»

Кнопка «Поиск» позволяет находить значения в таблице

Код кнопки «Поиск» представлен ниже.

for (int i = 0; i < dataGridView1.RowCount; i++)

{

dataGridView1.Rows[i].Selected = false;

for (int j = 0; j < dataGridView1.ColumnCount; j++)

if (dataGridView1.Rows[i].Cells[j].Value != null)

if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))

{

dataGridView1.Rows[i].Selected = true;

break;

}

}

После того как ввели значения в поле textbox выведутся значения в dataGridView. Также имеется возможность закрыть форму Form2 по нажатию на кнопку «Закрыть». Есть кнопка сохранить сохраняет все изменения, которые мы внесли в dataGridView.

Код кнопки «Сохранить» представлен ниже.

видыОперацийTableAdapter.Update(бахаев\_АвтомастерскаяDataSet);

File.Delete(@"..\..\..\WindowsFormsApp1\Бахаев Автомастерская.mdb");

File.Copy("Бахаев Автомастерская.mdb", @"..\..\..\WindowsFormsApp1\Бахаев Автомастерская.mdb");

Кнопка удалить удаляет на форме Form2 и удаляет все изменения, которые мы внесли в dataGridView.

Код кнопки «Удалить» представлен ниже.

видыОперацийBindingSource.RemoveCurrent(); }

}

На каждой форме имеется кнопка возвращения в главное меню, в случае если случайно нажали не на ту таблицу можно спокойно вернуться в Glav.

Код кнопки «Назад».

glav gl = new glav();

gl.Show();

this.Hide()

У каждой формы изменены свойства BackColor - фоновый цвет компонента. На всех формах настроено свойство Font которое служит для того чтобы изменить шрифт, используемый для отображения текста на элементе управления. С помощью свойства StartPosition при выборе CenterScreen формы будут появляться по центру экрана. Все элементы на формах с помощью свойства Locked – определяет, можно ли перемещать элемент управления или изменять его размеры.

## Заключение

# Список использованных источников