Федеральное казенное профессиональное образовательное учреждение

«Новочеркасский технологический техникум-интернат»

Министерства труда и социальной защиты Российской Федерации

**КУРСОВОЙ ПРОЕКТ**

по профессиональному модулю

ПМ.02 Разработка и администрирования

на тему: Разработка базы данных и клиентского приложения для автоматизации учета продаж программного обеспечения

Выполнил студент (ка) \_\_\_\_\_3\_\_\_ курса \_\_\_\_\_П-2С\_\_\_ группы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Рашев К.Е\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель проекта О.В Полякова

К защите Защита принята с оценкой

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

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

Новочеркасск

**2017 – 2018 год**

ФКПОУ «НТТИ» Минтруда России

УТВЕРЖДАЮ

Зам. директора по УМР

\_\_\_\_\_\_\_\_\_Л.Ю. Саенко

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20 17 г.

**Задание**

на курсовое проектирование по МДК.02.2 Технология разработки и защиты баз данных профессионального модуля ПМ.02 Разработка и администрирование баз данных

Исполнитель, студент: \_\_\_\_\_\_\_\_\_\_\_Рашев К.Е\_\_\_\_\_\_\_\_\_\_\_\_\_ Группа:\_\_\_\_п-2с\_\_\_\_\_

Тема проекта: Разработка базы данных и прикладных программ для учёта продаж ПО

**График выполнения проекта**

|  |  |  |
| --- | --- | --- |
| Наименование раздела | Объем, % | Срок выполнения |
| Анализ предметной области и формирование требований пользователей | 20% |  |
| Проектирование и создание базы данных | 40% |  |
| Разработка прикладных программ | 10% |  |
| Защита данных | 10% |  |
| Тестирование | 10% |  |
| Оформление документации | 10% |  |
| Всего: | 100% |  |

Дата выдачи задания:…………..27.10.2017

Дата защиты: …………….15.11.20181

Преподаватель: .О.В. Полякова.............................

Рассмотрено на заседании ПЦК ОПД и СД специальности 09.02.03 ПКС

Председатель ПЦК: ........................... О.В. Полякова

**Реферат**

Проект 81 с., 30 рис., 9 табл. Базы данных, предназначены для автоматизации учета продаж программного обеспечения.

Объектом проектирования являются базы данных и прикладные программы для работы с ней.

Цель разработки - автоматизация учета продаж программного обеспечения.

В процессе работы проводились разработка концептуальной, логической и физической модели базы данных.

В результате проектирования созданы базы данных и пакеты прикладных программ для конечных пользователей, которые позволяют автоматизировать учета продаж программного обеспечения.

Разработанный программный продукт может быть внедрен после проведения дополнительного тестирования.

Содержание

[Введение 7](#_Toc535930640)

[1. Анализ предметной области и формирование требований пользователей 9](#_Toc535930641)

[1.2 Список пользователей базы данных 14](#_Toc535930642)

[1.3 Запросы пользователей. 16](#_Toc535930643)

[1.4 Перечень входных документов. 20](#_Toc535930644)

[2. Проектирование и создание баз данных 21](#_Toc535930645)

[2.1 Разработка концептуальной модели предметной области 21](#_Toc535930646)

[2.2 Логическое проектирование 26](#_Toc535930647)

[2.3 Создание базы данных в СУБД SQL Server 29](#_Toc535930648)

[2.4. Создание хранимых процедур 31](#_Toc535930649)

[2.6. Создание базы данных в СУБД MYSQL 34](#_Toc535930650)

[3 Разработка клиентского программного обеспечения 36](#_Toc535930651)

[3.1 Создание клиентской части в среде Delphi 36](#_Toc535930652)

[3.2 Создание модуля данных (Таблица 8) 37](#_Toc535930653)

[3.3 Разработка клиентского приложения на Delphi 39](#_Toc535930654)

[3.3.1 Форма авторизации и главного меню (*Favto, avto*). 39](#_Toc535930655)

[3.3.2 Форма ввода товара (*Fpost,post*). 39](#_Toc535930656)

[3.3.2 Форма ввода заказа (*Fzak,zak*). 43](#_Toc535930657)

[3.3.3 Форма изменения заказа (*Fzak2,zak2*). 46](#_Toc535930658)

[3.3.4 Форма списка отправки ПО (*Fspis, spid*). 48](#_Toc535930659)

[3.3.5 Форма книги продаж (*Fbook, book*). 50](#_Toc535930660)

[3.3.6 Форма рейтинга продаж (*Frait, rait*). 51](#_Toc535930661)

[3.3.7 Форма рейтинга продаж по категориям (*Fkrait, krait*). 51](#_Toc535930662)

[3.4 Разработка вэб-приложения 52](#_Toc535930663)

[3.4.1 Главное меню – Index.php 52](#_Toc535930664)

[3.4.2 Оформление договора – Dog.php 55](#_Toc535930665)

[3.4.3 Ввод товара – Post.php 57](#_Toc535930666)

[3.4.4 Оформление заказа – zak.php 61](#_Toc535930667)

[3.4.5 Изменение статуса заказа – upd\_zak.php 64](#_Toc535930668)

[3.4.6 Список для отправки ПО – spisok.php 69](#_Toc535930669)

[3.4.7 Книга продаж – book.php 70](#_Toc535930670)

[3.4.8 Рейтинг продаж – rait.php 72](#_Toc535930671)

[3.4.9 Рейтинг продаж – krait.php 73](#_Toc535930672)

[3.5Разработка клиентского программного обеспечения на C# 73](#_Toc535930673)

[3.5.1 Форма авторизации и главного меню (*Favto, form1.cs*). 73](#_Toc535930674)

[3.5.2 Форма ввода товара (*Fpost.Form2.cs*). 74](#_Toc535930675)

[3.5.3 Форма ввода заказа (*Fzak,form3.cs*). 78](#_Toc535930676)

[3.5.4 Форма изменения заказа (*Fstat,form4.cs*). 81](#_Toc535930677)

[3.5.5 Форма списка отправки ПО (*otptavka, form5.cs*). 83](#_Toc535930678)

[3.5.6 Форма отчётов (*otch, form6.cs*). 84](#_Toc535930679)

[4 Тестирование 86](#_Toc535930680)

[4.1 Модульное тестирование 86](#_Toc535930681)

[4.2 Тестирование интеграции 88](#_Toc535930682)

[4.3 Тестирование правильности 89](#_Toc535930683)

[5.Классификации информационной системы персональных данных 90](#_Toc535930684)

[Заключение 93](#_Toc535930685)

# Введение

Разработанная программа предназначена для хранения данных о программном обеспечении, которым располагает фирма и об объемах его реализации, а также информацию о клиентах. Для ведения отчетности, организации, поиска требуемого ПО и информации о покупателе в базе хранятся сведения, охватывающие все требуемые аспекты.

Коммерческое ПО. При создании программного продукта издатель, выполнив анализ рынка, заказывает у исполнителя разработку такого ПО, которое должно пользоваться на рынке спросом, и выделяет на его создание деньги. По окончании работ издатель получает все имущественные права на созданный продукт (право на тиражирование, продажу под собственной торговой маркой, право на получение дохода от программы любым способом). При этом может быть оговорено получение исполнителем некоторого процента (роялти) с каждой проданной копии (как правило, для программ, издающихся сотнями тысяч или миллионами копий, роялти составляет 1-3 %) — тогда он получает меньшую сумму на разработку или вообще создает программу за свой счет. Если же отчисления не предусмотрены, то все расходы по подготовке программы издатель берет на себя. Он также вкладывает средства в упаковку, рекламную кампанию, организации сетей сбыта и т. д. Издатель обеспечивает расходы, связанные с сопровождением продукта и технической поддержкой пользователей.

За исполнителем навечно остаются авторские права на программу — право указывать свое имя или логотип своей фирмы на начальной заставке, в документации, на упаковочной коробке.

Крупные компании имеют и подразделения разработки ПО, и отделы, занимающиеся его распространением, что помогает эффективно организовать весь процесс от производства программ до доставки их потребителю.

Условно-бесплатное ПО (shareware). В связи с активным развитием Интернета огромное число индивидуальных разработчиков получили возможность распространения своих программ по всему миру. Не имея средств на рекламные кампании, они предоставляют возможность получения ознакомительных версий их программ (демонстрационных или имеющих искусственные ограничения) через Интернет, Если человеку эта программа нравится, он оплачивает небольшую сумму и получает полную работоспособную версию.

В Интернете есть немало узлов, которые предлагают бесплатные услуги по размещению таких программ.

Бесплатное ПО (freeware, public domain). Такие программы не имеют никаких ограничений, однако автор может попросить заплатить ему некоторую сумму, не настаивая, впрочем, на этом (это метод freeware). Некоторые программы авторы называют «общественным достоянием» (public domain), ничего взамен не требуют и нередко распространяют такое ПО в исходных текстах.

Как правило, стимулом к созданию freeware/public domain-программ служит стремление повысить собственную квалификацию, установить контакты с коллегами, а в случае удачно созданной программы получить известность и, как правило, приглашение на хорошую работу.

# 1. Анализ предметной области и формирование требований пользователей

Разработанная программа предназначена для хранения данных о программном обеспечении, которым располагает фирма и об объемах его реализации, а также информацию о клиентах. Для ведения отчетности, организации, поиска требуемого ПО и информации о покупателе в базе хранятся сведения, охватывающие все требуемые аспекты.

Учитывать следующие атрибуты покупателей: ФИО, профессия, адрес, телефон.

Информация о программном обеспечении включает: название ПО, назначение, поставщик, стоимость.

По степени тиражируемости всё программное обеспечение делится на три категории:

* программное обеспечение, разрабатываемое на заказ;
* программное обеспечение для крупных корпораций и организаций;
* программное обеспечение для массового потребителя.

По степени переносимости программы делят на :

* платформозависимые;
* кроссплатформенные.

По способу распространения и использования программы делят на :

* несвободные (закрытые);
* открытые;
* свободные.

По назначению программы делят на:

* системные ;
* прикладные:
* Системы программирования.

Пользователь получает программное обеспечение вместе с лицензией, которая предоставляет ему право использовать программный продукт при условии выполнения положений о лицензировании. Как правило, эти условия ограничивают возможности пользователя передавать программный продукт другим пользователям, изменять код.

Часть программного обеспечения поставляется со свободной лицензией. Такие лицензии позволяют распространять программное обеспечение, а также модифицировать его.

Часть программного обеспечения распространяется как бесплатное. Существует также условно бесплатное программное обеспечение. В этом случае обычно пользователь бесплатно получает демонстрационную версию программного продукта с несколько ограниченными возможностями на определенный испытательный период, а после его окончания обязан или приобрести продукт, или деинсталлировать его.

Каждая покупка содержит только одно наименование ПО, при этом включает следующие поля: название ПО, ФИО покупателя, количество покупаемого ПО, сумма реализации, дата покупки.

* Виды лицензий:
  + Проприетарные лицензии:

Основной характеристикой проприетарных лицензий является то, что издатель ПО в лицензии даёт разрешение её получателю использовать одну или несколько копий программы, но при этом сам остаётся правообладателем всех этих копий. Одно из следствий такого подхода заключается в том, что практически все права на ПО остаются за издателем, а пользователю передаётся лишь очень ограниченный набор строго очерченных прав. Для проприетарных лицензий типично перечисление большого количества условий, запрещающих определённые варианты использования ПО, даже тех, которые без этого запрета были бы разрешены законом об авторском праве. Некоторые лицензии на ОС имеют ограничения на мультипроцессорность. Лицензии на профессиональное ПО могут иметь дополнительные ограничения. Так, лицензия SolidWorks имеет региональные ограничения, предусматривает сбор сведений об использовании ПО и возможность дистанционной блокировки. Большинство лицензий запрещают передачу ПО третьим лицам.

Наиболее значительным следствием применения проприетарной лицензии является то, что конечный пользователь обязан принять её, так как по закону владельцем ПО является не он, а издатель программы. В случае отказа принять лицензию пользователь вообще не может работать с программой.

* + Лицензии свободного и открытого ПО

В отличие от проприетарных, свободные и открытые лицензии не оставляют права на конкретную копию программы её издателю, а передают самые важные из них (составляющие 4 свободы по определению Фонда СПО, или попадающие под другое определение свободного или открытого ПО) конечному пользователю, который и становится владельцем. В результате пользователь по умолчанию получает важные права, которые закон об авторском праве по умолчанию даёт только владельцу копии, однако все авторские права на ПО по-прежнему остаются у издателя.

Главной отличительной чертой свободных лицензий является то, что они совершенно не ограничивают личное пользование — пользователь волен принимать или не принимать их: работать с программой он может и без лицензии. Однако если ему требуется какое-либо из дополнительных прав, которые даёт лицензия (например, на распространение ПО, или предоставление доступа к нему по сети (пример — Affero General Public License), он обязан принять лицензию и действовать в её рамках.

Все ПО делится по категориям:

Категория

* Антивирусы(147)
* Безопасность(195)
* Графика и дизайн(577)
* Офисные программы(711)
* Игры(1335)
* Обработка видео, звука(192)
* Программы для смартфонов(176)
* Операционные системы(43)
* Утилиты(809)
* Восстановление данных(359)
* Управление компанией(423)
* Навигация и карты(129)

Вендор — компания, которая выпускает, поставляет продукцию под собственной торговой маркой.

Вендоры

* Microsoft(125)
* Лаборатория Касперского(20)
* ESET(19)
* Adobe Systems Russia(28)
* InfoWatch(2)
* Autodesk, Inc.(37)
* ABBYY(20)
* RAR Lab(4)
* Kerio Technologies Inc.(3)
* Dr. WEB(25)
* Symantec Corporation(2)
* Veritas Software(6)
* Corel Corporation(21)
* 1С-Битрикс(2)
* ООО "АСКОН-Системы проектирования"(67)

Клиент может быть юридическим или физическим лицом

Оплата может производиться наличными или по безналичному расчету

После выбора нужного ПО клиент делает заказ, в котором указывает

\* Фамилия, имя, отчество E-mail Регион Страна Город: \* Телефон (с кодом города) Комментарий к заказу

После оплаты производится отправка ПО клиенту по почте или в электронном виде по e-mail или ссылке для скачивания

После отправки заказанного ПО менеджер по продажам делает соответствующую отметку об этом. Определим список пользователей базы данных.

## 1.2 Список пользователей базы данных

* Бухгалтер
* Специалист по поставкам ПО
* Продавец

Бухгалтер ведёт учёт по продажам ПО.

Продавец продаёт коробочные диски с лицензионным ПО и лицензионные ключи к программам. После оплаты он должен в течение суток отправить программный продукт клиенту

Специалист по поставкам программного обеспечения заключает договор с поставщиками.

Поиск осуществляется по трём основным критериям: ОС, прикладное ПО, системное ПО.

* ОС:
  + Windows
  + Mac OS
  + Linux
* Прикладное ПО:
  + Офисы
  + Графика
  + Браузеры
  + Общение
  + Почта
  + Антивирусы
  + Игры и развлечения
  + Музыка
  + Видео
  + Языки программирования
  + Базы данных.
* Системное ПО:
  + Драйверы
  + Утилиты

Покупатель преобретает коробочные диски с лицензионным ПО и лицензионные ключи к программам. А теперь определим запросы пользователей.

## 1.3 Запросы пользователей.

Специалист по продажам ПО и покупатель в соответствии с рисунком 1 подбирают ПО для определённых целей

В столбце «Вендор» определяется разработчик данного приложения.

В столбце «Тип ПО» определяется предназначение данного приложения.

В столбце «Тип Лицензий» определяется лицензия данного приложения.

Подбор ПО по критериям: Категория ТИП ПО, стоимость, тип лицензии.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Категория | Наименование ПО | Тип лицензии | Тип ПО | Вендор | Цена | Описание |
| Антивирусы | Касперский 6.0 | Проприетарная | Прикладное | Лаборатория Касперского | 600р. | Отечественный антивирус со своими уни-кальными функциями. |
|  |  |  |  |  |  |  |

Рисунок 1 –Подбор ПО по критериям

Бухгалтер в соответствии с рисунком 2 ведёт учёт книги продаж.

В столбце «Наименование покупателя» определяется имя покупателя или название корпорации, которая покупает приложения.

Книга продаж за период с\_\_\_ по\_\_\_

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номер счета-фактуры | Дата | Наименование ПО | Тип лицензии | Тип ПО | Вендор | Наименование покупателя | Стоимость продаж, освобождаемых от налога |
| 1 | 25.12.17 | Касперский 6.0 | Проприетарная | Прикладное | Лаборатория Касперского | Петров К.Е | 600р. |
|  |  |  |  |  |  |  |  |
| Всего |  |  |  |  |  |  |  |

Рисунок 2 – Книга продаж

Продавец формирует список ПО для отправки программного обеспечения на почту клиента в соответствии с

рисунком 3.

Список ПО для отправки клиенту на \_\_\_\_\_\_

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| номер счета-фактуры | Дата | Наименование ПО | Тип лицензии | Наименование покупателя | Адрес | Способ оплаты | Дата оплаты | Сумма |
| 1 | 25.12.17 | Касперский 6.0 | Проприетарная | Петров К.Е | [Zero777@inbox,ru](mailto:Zero777@inbox,ru) | Картой | 25.12.17 | 600р. |
|  |  |  |  |  |  |  |  |  |
| Всего |  |  |  |  |  |  |  |  |

Рисунок 3 –Список ПО для отправки клиенту

Бухгалтер формирует рейтинг продаж самых популярных программ в соответствии с рисунком 4.

Рейтинг продаж за период с\_\_\_ по\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категория ПО | Наименование ПО | Тип лицензии | Вендор | Объем продаж, руб |
| Антивирусы | Касперский 6.0 | Проприетарная | Лаборатория Касперского | 600р. |
|  |  |  |  |  |
|  |  |  |  |  |

Рисунок 4 – Рейтинг продаж

Бухгалтер делает рейтинг продаж самых популярных категорий программ в соответствии с рисунком 5.

Рейтинг продаж ПО по категориям за период с\_\_\_ по\_\_\_

|  |  |  |
| --- | --- | --- |
| Категория ПО | Вендор | Объем продаж, руб |
| Антивирусы | Лаборатория Касперского | 600р. |
|  |  |  |
|  |  |  |

Рисунок 5 – Рейтинг продаж ПО по критериям

## 1.4 Перечень входных документов.

**Перечень входных документов**

1. Договор с корпорацией, на поставку лицензионного ПО
2. Приказы о приеме и увольнении сотрудников.
3. Расценки на продажу ПО, выполняемые фирмой
4. Договор купли-продажи программного обеспечения
5. Документ об оплате.

# 2. Проектирование и создание баз данных

## 2.1 Разработка концептуальной модели предметной области

2.1.1 Выделил сущности из системного анализа:

* Договор
* Товар
* Заказ
* Категория
* ТипПО
* ТипЛицензии

2.1.2 Выбрал первичные ключи для каждой сущности

(первичный ключ должен быть уникальным):

* Договор - №Договора
* Товар-№Товара
* Заказ -№Заказа
* Категория-№КатегорииПО
* ТипПО-№ТипПО
* ТипЛицензии-№ТипЛицензии

2.1.3 Определил описательные атрибуты сущностей

Договор(№Договора,НаименованиеПоставщика,Вендор,БанковскиеРеквизитыПоставщика,ДатаПодписания);

Товар(№Товара,№Договора,НазваниеПО,№ТипПО,№Категории, Платформа, №ТипЛицензии, Язык,Разработчик,КоличествоКопий, Стоимость);

Категория(№Категории,КатегорияПО)

ТипПО(№ТипПО,ТипПО)

ТипЛицензии(№ТипЛицензии, ТипЛицензии,)

Заказ-(№Заказа,№Товара,НаименованиеПокупателя,профессия,телефон, Кол-во Копий, ОбщаяСтоимость, АдресДоставки(Е-mail,Дата Оплаты ,ДатаДоставки. Статус);

2.1.4 Выполнил спецификацию связей

* На Один договор выписывается много Товаров (1:М)
* Много Товаров имеют много Категорий(М:М)
* Много Товаров имеют много ТиповПО(М:М)
* Много Товаров имеют много ТиповЛицензий (М:М)
* На Один Товар поступает много Заказов (1:М)

2.1.5 Транзакции пользователя

* Т1 – Заключение договора с поставщиком

Входные данные: НаименованиеПоставщика, Вендор, БанковскиеРеквизитыПоставщика, ДатаПодписания

Выходные данные – Нет

Добавление в Договора -№Договора формируется автоматически

Ввод список товаров, которые будут по этому договору поставляться

В цикле

Входные данные: НазваниеПО, Разработчик, КоличествоКопий, Стоимость.

Выходных данных – №Товара, №ТипПО, №Категории, Платформа, №ТипЛицензииПО, Язык

Добавление в Товар №Товара, формируется автоматически

* Т2– Оформление Заказа

Входные данные: НаименованиеПокупателя, профессия, адрес, телефон, Кол-во Копий ~~,~~ АдресДоставки(Е-mail)

Выходные данные:

Подбор ПО по критериям: Категория ТИП ПО стоимость тип лицензии

Выбрать №Товара ,№Категории,№ТипПО,№ТипЛицензии.

Из таблиц ТОВАР,КАТЕГОРИИ,ТИППО,ТИПЛИЦЕНЗИИ

Входные данные: Категория ТИП ПО стоимость тип лицензии

Выходные данные :

Подбор ПО по критериям: Категория ТИП ПО стоимость тип лицензии

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Категория | Наименование ПО | Тип лицензии | Тип ПО | Вендор | Цена | Описание |

Выбор Категории,ТипПО,ТипЛицензии

Из таблиц Категории,ТипПО,ТипЛицензии

Входные данные: НаименованиеПокупателя, профессия, адрес, телефон, Кол-во Копий, АдресДоставки(Е-mail)

Выбрать,№Товара

Из таблицы ТОВАР

ОбщаяСтоимость =кол-во копий \*стоимость одного товара

Добавление в Заказы. №Заказа формируется автоматический

Интенсивность - случайным образом

* Т3 – Книга продаж за период с\_\_\_ по\_\_\_

Входные данные Дата1,Дата2(период с\_\_\_ по\_\_\_\_),

Выходные данные

Книга продаж за период с\_\_\_ по\_\_\_

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номер счета-фактуры | Дата | Наименование ПО | Тип лицензии | Тип ПО | Вендор | Наименование покупателя | Стоимость продаж, освобождаемых от налога |

Выбрать №заказа, ,№Категории,№ТипПО,№ТипЛицензии

Из таблиц Заказ,Категории,ТипПО,ТипЛицензии

По условию Если Дата находится в заданном периоде

Т4– Список ПО для отправки клиенту на \_\_\_\_\_\_

Выходные данные:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| номер счета-фактуры | Дата Оплаты | Наименование ПО | Тип лицензии | Наименование покупателя | АдресДоставки | Способ оплаты | Сумма |

Выбрать номер счета-фактуры Дата Наименование ПО Тип лицензии Наименование покупателя АдресДоставки Способ отправки Сумма Из таблиц товар,типлицензии,заказ где дата оплаты не равна нулю

Т5 – Рейтинг продаж за период с\_\_\_ по\_\_\_

Входные данные- , Дата1,Дата2(период с\_\_\_ по\_\_\_\_)

Выходных данных:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категория ПО | Наименование ПО | Тип лицензии | Вендор | Объем продаж, руб |

Выбрать Вендор, ,№Категории,№ТипЛицензии

Объём продаж = стоимость \* количество проданных копий

Из таблиц Товар,Категории, ТипЛицензии

По условию Если Дата формирования отчета находится в заданном периоде

Т6 – Рейтинг продаж ПО по категориям за период с\_\_\_ по\_\_\_

Входные данные- , Дата1,Дата2(период с\_\_\_ по\_\_\_\_)

Выходные данные:

|  |  |  |
| --- | --- | --- |
| Категория ПО | Вендор | Объем продаж, руб |

Выбрать Вендор, №Категории,

Из таблиц Договор,Категории.

Объём продаж = стоимость \* количество проданных копий

По условию Если находится в заданном периоде

2.1.6 Разработал концептуальной модели базы данных в виде модели "сущность-связь".

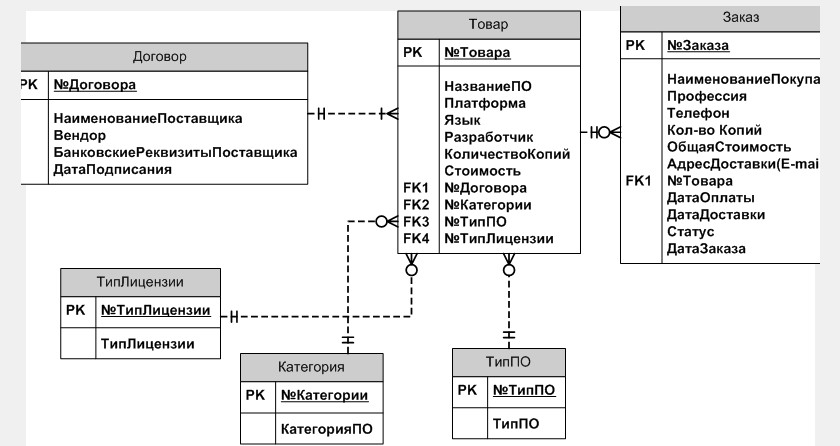


Рисунок 6 – Концептуальная модель

2.1.7 Нормализация отношений

Сущность Договор находится в 1НФ так как все входящие в отношение атрибуты атомарны (не составные)

и ни одно из ключевых полей не является пустым.

Сущность Договор находится во 2НФ так как схема отношения находится в 1НФ, и каждый ее не первичный атрибут функционально полно зависит от первичного ключа.

Сущность Договор находится в 3НФ так как схема отношения находится во 2НФ и каждый ее не первичный атрибут нетранзитивно зависит от первичного ключа.

Сущность Договор находится в 4НФ так как если схема отношения находится в 3НФ и в ней отсутствуют многозначные зависимости.

Сущность Договор находится в 5НФ так как схема отношения находится в 4НФ и в ней отсутствуют зависимости соединения.

## 2.2 Логическое проектирование

Имя базы данных ProdagaPO

Таблица 1 – логическая структура DOGOVOR-ДОГОВОР

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Ширина | Ограничения целостности | Примечание |
| Ndog | Числовой | 4 | Первичный ключ | №Договора |
| NamePos | Символьный | 100 | Не может быть пустым | НаименованиеПоставщика |
| Vendor | Строковый | 50 | Не может быть пустым | Вендор |
| Bankrekv | Символьный | 100 | Не может быть пустым | БанковскиеРеквизитыПоставщика |
| Dpodpis | Дата |  | Не может быть пустым | ДатаПодписания |

Таблица 2 – логическая структура TIPPO- ТИППО

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Ширина | Ограничения целостности | Примечание |
| NtipPO | Числовой | 4 | Первичный ключ | №ТипПО |
| TipPO | Символьный | 50 | Не может быть пустым | ТипПО |

Таблица 3 – логическая структура KATE- Категория

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Ширина | Ограничения целостности | Примечание |
| Nkat | Числовой | 4 | Первичный ключ | №Категории |
| Kat | Символьный | 50 | Не может быть пустым | Категория |

Таблица 4 – логическая структура TIPLISE-ТИПЛИЦЕНЗИЙ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Ширина | Ограничения целостности | Примечание |
| Ntiplis | Числовой | 4 | Первичный ключ | №ТипЛицензии |
| Tiplis | Символьный | 50 | Не может быть пустым | ТипЛицензии |

Таблица 5 – логическая структура Tovar-ТОВАР

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Ширина | Ограничения целостности | Примечание |
| Ntov | Числовой | 4 | Первичный ключ | №Товара |
| Ndog | Числовой | 4 | >0, Внешний ключ | №Договора |
| NamePO | Строковый | 100 | Не может быть пустым | НаименованиеПО |
| NtipPO | Числовой | 4 | >0, Внешний ключ | №ТипПО, |
| Nkat | Числовой | 4 | >0, Внешний ключ | №Категории |
| Plat | Строковый | 10 | Не может быть пустым | Платформа |
| Ntiplis | Числовой | 4 | >0, Внешний ключ | №ТипЛицензии |
| **Yaz** | Символьный | 100 | Не может быть пустым | Язык |
| Razrab | Строковый | 100 | Не может быть пустым | Разраюботчик |
| Kolvokop | Числовой | 10 | >0 | КоличествоКопий |
| Stomost | Числовой | 10,2 | >0 | Стоимость 1 товара |

Внешний ключ NDog ссылается на таблицу DOGOVOR

Внешний ключ NTiplis ссылается на таблицу KATEGORIA

Внешний ключ NTipPO ссылается на таблицу TIPPO

Внешний ключ Nkat ссылается на таблицу TIPLISENZY

Таблица 6– логическая структура ZAKAZ-Заказ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Ширина | Ограничения целостности | Примечание |
| Nzak | Числовой | 4 | Первичный ключ | №Заказа |
| Ntov | Числовой | 4 | >0,Внешний ключ | №Товара |
| Namepok | Строковый | 100 | Не может быть пустым | НаименованиеПокупателя |
| Dzak | Дата |  | Не может быть пустым | ДатаЗаказа |
| Prof | Символьный | 100 | Может быть пустым | Профессия |
| Tel | Символьный | 20 | Не может быть пустым и в начале должен быть (+7) | Телефон |
| Kookop | Числовой | 10,2 | >0 | Кол-воКопий |
| AdresDost | Символьный | 100 | Не может быть пустым | АдресДоставки(Е-mail) |
| Dopl | Дата |  |  | ДатаОплаты |
| Ddost | Дата |  |  | ДатаДоставки |
| Stat | Числовой | 2 | Доставлено(2,  Оплачено(1) Заказано(0) | Статус |
| Sposopl | Строковый | 50 | Не может быть пустым | Способ оплаты |

Внешний ключ NTov ссылается на таблицу TOVAR

## 2.3 Создание базы данных в СУБД SQL Server

Имя базы данных ProdagaPO

Создание БД

CREATE DATABASE prodajaPO

ON PRIMARY

(

NAME=prodajaPO\_Data,

FILENAME='C:\Users\Кирилл\Desktop\3 курс\пм-02\курсовой\Курсовой проект\ProdajaPO\_Data.mdf',

SIZE=7MB,

FILEGROWTH=3MB)

LOG ON

(

NAME=prodajaPO\_Log,

FILENAME=''C:\Users\Кирилл\Desktop\3 курс\пм-02\курсовой\Курсовой проект\ProdajaPO\_Log.ldf',

SIZE=3MB,

MAXSIZE=10MB,

FILEGROWTH=1MB

)

Создание таблицы DOGOVOR:

create table DOGOVOR(

Ndog int primary key identity(1, 1) not null,

NamePos varchar(100),

vendor varchar(50),

BankRekv varchar(100),

Dpodpis date

)

Создание таблицы TIPPO:

create table TIPPO(

NtipPO int primary key identity(1, 1) not null,

TipPO varchar(50)

)

Создание таблицы KATE:

create table KATE(

NKat int primary key identity(1, 1) not null,

Kat varchar(50)

)

Создание таблицы TIPLISE:

create table TIPLISE(

Ntiplis int primary key identity(1, 1) not null,

Tiplis varchar(50)

)

Создание таблицы TOVAR:

create table TOVAR(

Ntov int primary key identity(1, 1) not null,

Ndog int,

NamePO varchar(100),

NtipPO int,

Nkat int,

Plat varchar(10),

Ntiplis int,

Yaz varchar(100),

Razrab varchar(50),

Kolvokop int,

Stomost money,

Foreign key (NDog) references Dogovor,

Foreign key (NKAT) references KATE,

Foreign key (NTIPPO) references TIPPO,

Foreign key (NTIPLIS) references TIPLISE

)

Создание таблицы ZAKAZ:

Create table zakaz(

Nzak int primary key identity(1, 1) not null,

Ntov int,

Dzak date,

Namepok varchar(100),

Prof varchar(100),

Tel varchar(20),

Kookop int,

AdresDost varchar(10),

Dopl date,

Ddost date,

Stat int,

Sposopl varchar(10),

Foreign key (Ntov) references tovar

)

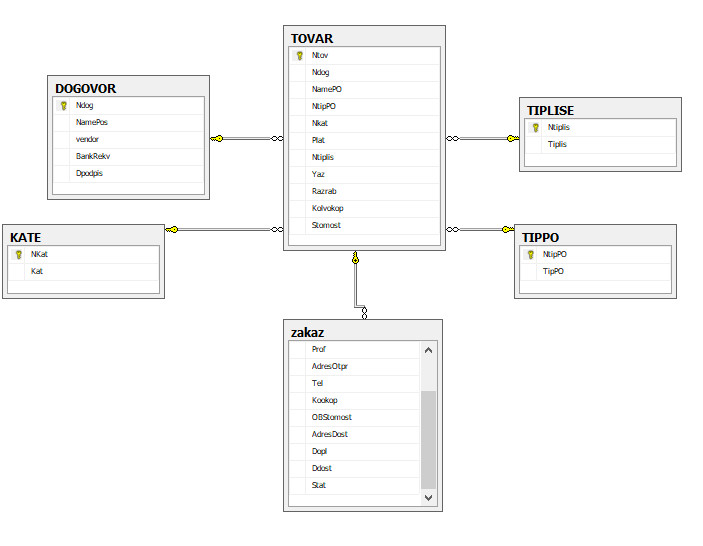
****

Рисунок 7- Диаграмма базы данных

## 2.4. Создание хранимых процедур

Создадим хранимую процедуру для заключения договора.

Т1-Заключение договора

create procedure OformDog

@Dpodpis date,@NamePos varchar(100),@vendor varchar(50),

@BankRekv varchar(100)

as

begin

insert into DOGOVOR(Dpodpis,NamePos,vendor,BankRekv)

values(@dpodpis,@NamePos,@vendor,@BankRekv)

end

Создадим хранимую процедуру для ввода товара.

Ввод товара

create procedure vvodtov

@Ndog int,

@NamePO varchar(100),

@NtipPO int,

@Nkat int,

@Plat varchar(10),

@Ntiplis int,

@Yaz varchar(100),

@Razrab varchar(50),

@Kolvokop int,

@Stomost money

as

begin

insert into TOVAR(Ndog,NamePO,NtipPO,Nkat,Ntiplis,Plat,Yaz,Razrab,Kolvokop,Stomost)

values(@Ndog,@NamePO,@NtipPO,@Nkat,@Ntiplis,@Plat,@Yaz,@Razrab,@Kolvokop,@Stomost)

End

Создадим хранимую процедуру для оформления заказа.

Т2-Оформление заказа

create procedure oformzak

@Ntov int,

@dzak date,

@Namepok varchar(100),

@Prof varchar(100),

@Tel varchar(10),

@Kookop int,

@AdresDost varchar(10),

@sposopl varchar(50),

as

begin

insert into zakaz(Ntov,dzak,Namepok,Prof,Tel,Kookop,AdresDost,sposopl,Dopl,Ddost)

values(@Ntov,@dzak,@Namepok,@Prof,@Tel,@Kookop,@AdresDost,@sposopl)

End

Создадим хранимую процедуру для отображения книги продаж.

Т3- Книга продаж

create procedure knigaprod

@data1 date,

@data2 date

as

begin

select zakaz.Nzak as [Номер счета-фактуры],zakaz.Dopl as [Дата оплаты],TOVAR.NamePO as [Наимеенование ПО],

TIPLISE.Tiplis as [Тип лицензии],TipPO.TipPO as [Тип ПО],DOGOVOR.vendor as [Вендор],zakaz.Namepok as [Наименование покупателя],

TOVAR.Stomost as [Стоимость продаж освобождённых от налогов ]

from zakaz,TIPLISE,TIPPO,TOVAR,DOGOVOR

where zakaz.Ntov=TOVAR.Ntov AND TOVAR.Ntiplis=TIPLISE.Ntiplis

AND TOVAR.NtipPO=TIPPO.NtipPO AND DOGOVOR.Ndog=TOVAR.Ndog

AND Dopl between @data1 and @data2

End

Создадим хранимую процедуру для отображения списка ПО для отправки клиентам.

Т4– Список ПО для отправки клиентам

create procedure otpravkaPO

as

begin

select zakaz.Nzak,zakaz.Dopl,TOVAR.NamePO,TIPLISE.Tiplis,

zakaz.Namepok,zakaz.AdresDost,zakaz.sposopl,

(zakaz.kookop\*tovar.Stomost) as summa from

zakaz,TIPLISE,TIPPO,TOVAR,DOGOVOR

where zakaz.Ntov=TOVAR.Ntov AND

TOVAR.Ntiplis=TIPLISE.Ntiplis

AND TOVAR.NtipPO=TIPPO.NtipPO AND DOGOVOR.Ndog=TOVAR.Ndog

AND zakaz.Dopl is not null and zakaz.Ddost is null AND Stat=1

End

Создадим хранимую процедуру для отображения рейтинга продаж ПО.

Т5– Рейтинг продаж

create procedure Raitprod

@data1 date,

@data2 date

as

begin

select Kat as [Категория ПО],NamePO as [Наименование ПО],

Tiplis as [Тип лицензии],vendor as [Вендор],sum(zakaz.kookop\*tovar.Stomost)

as [Объём продаж] from DOGOVOR, zakaz,TIPLISE,TIPPO,TOVAR,KATE

where TOVAR.Nkat=KATE.NKat AND DOGOVOR.Ndog=TOVAR.Ndog

and TOVAR.Ntov=zakaz.Ntov AND TIPLISE.Ntiplis=TOVAR.Ntiplis

AND Tippo.Ntippo=TOVAR.Ntippo

and Dopl between @data1 and @data2 GROUP BY Kat,NamePO,Tiplis,vendor

end

Создадим хранимую процедуру для отображения рейтинга продаж ПО по категориям.

Т6 – Рейтинг продаж ПО по категориям

create procedure RAITKATEPROD

@data1 date,

@data2 date

as

begin

select Kat as [Категория ПО], vendor as [Вендор],SUM(zakaz.kookop\*tovar.Stomost) as [Объём продаж] from DOGOVOR,TOVAR,KATE,zakaz where

zakaz.Ntov=TOVAR.Ntov AND

TOVAR.Nkat=KATE.NKat and DOGOVOR.Ndog=TOVAR.Ndog AND

Dopl between @data1 and @data2

group by vendor,Kat

end

Создадим хранимую процедуру для изменения статуса на оплачено.

Статус оплачено

create procedure opkach

@zak int,

@date1 date

as

begin

Update zakaz SET Stat=1, Dopl=@date1

where Nzak = @zak

end

Создадим хранимую процедуру для изменения статуса на доставлено.

Статус доставлено

create procedure dostav

@zak int,

@date date

as

begin

Update zakaz SET Stat=2, Ddost=@date

where Nzak = @zak

end

## 2.6. Создание базы данных в СУБД MYSQL

Создание таблицы DOGOVOR:

CREATE TABLE `Dogovor` (

`Ndog` int(4) NOT NULL,

`NamePos` varchar(100) NOT NULL,

`Vendor` varchar(50) NOT NULL,

`Bankrekv` varchar(100) NOT NULL,

`Dpodpis` date NOT NULL

)

Создание таблицы TIPPO:

CREATE TABLE `TIPPO` (

`NtipPO` int(11) NOT NULL,

`TipPO` varchar(50) NOT NULL

)

Создание таблицы KATE:

CREATE TABLE `KATE` (

`Nkat` int(11) NOT NULL,

`Kat` varchar(50) NOT NULL

)

Создание таблицы TIPLISE:

CREATE TABLE `TIPLISE` (

`Ntiplis` int(11) NOT NULL,

`Tiplis` varchar(50) NOT NULL

)

Создание таблицы TOVAR:

CREATE TABLE `Tovar` (

`Ntov` int(4) NOT NULL,

`Ndog` int(4) NOT NULL,

`NamePO` varchar(100) NOT NULL,

`NtipPO` int(4) NOT NULL,

`Nkat` int(4) NOT NULL,

`Plat` varchar(50) NOT NULL,

`Ntiplis` int(4) NOT NULL,

`Yaz` varchar(100) NOT NULL,

`Razrab` varchar(100) NOT NULL,

`Kolvokop` int(10) NOT NULL,

`Stomost` int(10) NOT NULL

)

Создание таблицы ZAKAZ:

CREATE TABLE `ZAKAZ` (

`Nzak` int(4) NOT NULL,

`Ntov` int(4) NOT NULL,

`Namepok` varchar(100) NOT NULL,

`Dzak` date NOT NULL,

`Prof` varchar(100) NOT NULL,

`Tel` varchar(20) NOT NULL,

`Kookop` int(10) NOT NULL,

`AdresDost` varchar(100) NOT NULL,

`Dopl` date DEFAULT NULL,

`Ddost` date DEFAULT NULL,

`Stat` int(2) NOT NULL,

`Sposopl` varchar(50) NOT NULL

)

# 3 Разработка клиентского программного обеспечения

## 3.1 Создание клиентской части в среде Delphi

3.1.1 Структура приложения

Создадим проект, форму для реализации функций меню *Fmain,main* клиентского приложения, а также все необходимые формы для реализации транзакций пользователей.

Перечень прикладного программного обеспечения с указанием назначения каждого программного модуля, приведен в таблице 7.

Таблица 7 - Перечень форм и модулей клиентского приложения

|  |  |  |  |
| --- | --- | --- | --- |
| Форма  (Name) | Модуль  (Save as..) | Назначение | Пользователи, имеющие право на выполнение. |
| *Favto.dfm* | *avto.pas* | *Форма авторизации* | *ВСЕ* |
| *Fpost.dfm* | *post.pas* | *Форма для заключения договоров и поставки товара(ПО)* | *Специалист по поставкам ПО* |
| *Fzak.dfm* | *zak.pas* | *Форма для магазина с ПО* | *Продавец* |
| *Fzak2.dfm* | *zak2.pas* | *Форма для изменения статуса с ПО* | *Продавец* |
| *Fbook.dfm* | *book.pas* | *Форма книга продажи ПО* | *Бухгалтер* |
| *Fspis.dfm* | *spis.pas* | *Форма Список ПО для отправки клиента* | *Продавец* |
| *Frait.dfm* | *rait.pas* | *Форма* *рейтинга продаж ПО* | *Бухгалтер* |
| *Frait2.dfm* | *rait2.pas* | *Форма Рейтинг продаж ПО по категориям* | *Бухгалтер* |
| *Fdm.dfm* | *dm.pas* | *Форма модуля данных* |  |

3.1.2 Модуль данных

На модуле данных расположены компоненты ADOConnection для подключения к базе данных, ADOQuery для вывода данных в таблицу, ADOStoredProc для выполнения хранимой процедуры, DataSourse для связи с ADOQuery.

## 3.2 Создание модуля данных (Таблица 8)

Таблица 8 - Компоненты для работы с БД

|  |  |  |
| --- | --- | --- |
| Имя компонента | Свойства | Назначение, имя объекта БД |
| *ADOConnection1* | *ConnectionString= Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=prodajaPO;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=BRAVO;Use Encryption for Data=False;Tag with column collation when possible=False;* | *Соединение с БД* |
| *ADOQueryD* | *Connection=ADOConnection1*  *SQL = ‘select \* from dogovor’* | *Выборка данных из таблицы*  *Dogovor* |
| *DataSourceD* | *DataSet= ADOQueryD* |
| *ADOQueryT* | *Connection=ADOConnection1*  *SQL = ‘select \* from Tovar’* | *Выборка данных из таблицы Tovar* |
| *DataSourceT* | *DataSet= ADOQueryT* |
| *ADOQueryTP* | *Connection=ADOConnection1*  *SQL = ‘select \* from TipPO’* | *Выборка данных из таблицы TIPPO* |
| *DataSourceTP* | *DataSet= ADOQueryTP* |
| *ADOQueryTL* | *Connection=ADOConnection1*  *SQL = ‘select \* from Tiplise’* | *Выборка данных из таблицы TipLise* |
| *DataSourceTL* | *DataSet= ADOQueryTL* |
| *ADOQueryZ* | *Connection=ADOConnection1*  *SQL = ‘select \* from Zakaz’* | *Выборка данных из таблицы Zakaz* |
| *DataSourceZ* | *DataSet= ADOQueryZ* |
| *ADOQueryK* | *Connection=ADOConnection1*  *SQL = ‘select \* from Kate’* | *Выборка данных из таблицы* |
| *DataSourceK* | *DataSet= ADOQueryK* |  |
| *ADOStoredProcInsD* | *Connection=ADOConnection1* | *Добавление договор* |
| *DataSourceproc1* | *DataSet= ADOStoredProcInsD* |  |
| *ADOStoredProcInsT* | *Connection=ADOConnection1* | *Добавление товара* |
| *DataSourceproc2* | *DataSet= ADOStoredProcInsT* |  |
| *ADOStoredProcInsZ* | *Connection=ADOConnection1* | *Добавление заказа* |
| *DataSourceproc3* | *DataSet= ADOStoredProcInsZ* |  |
| *ADOStoredProcKniga* | *Connection=ADOConnection1* | *Книга продаж* |
| *DataSourceproc4* | *DataSet= ADOStoredProcKniga* |  |
| *ADOStoredProcotprav* | *Connection=ADOConnection1* | *Список для отправки ПО* |
| *DataSourceproc5* | *DataSet= ADOStoredProcotprav* |  |
| *ADOStoredProcRaitProd* | *Connection=ADOConnection1* | *Рейтинг продаж* |
| *DataSourceproc6* | *DataSet= ADOStoredProcRaitProd* |  |
| *ADOStoredProcRaitKProd* | *Connection=ADOConnection1* | *Рейтинг продаж по категорий* |
| *DataSourceproc7* | *DataSet= ADOStoredProcRaitKProd* |  |
| *ADOStoredProcOPLACH* | *Connection=ADOConnection1* | *Изменение статуса заказа на оплачено* |
| *ADOStoredProcDostav* | *Connection=ADOConnection1* | *Изменение статуса заказа на доставлено* |

## 3.3 Разработка клиентского приложения на Delphi

### 3.3.1 Форма авторизации и главного меню (*Favto, avto*).

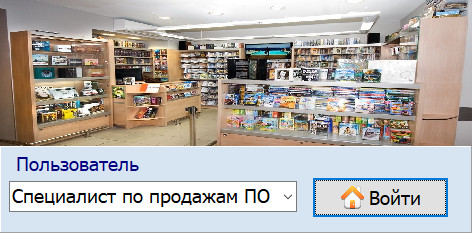


Рисунок 8–Внешний вид формы *Favto*

Для ввода данных в таблицу я использовал ComboBox

Выбор пользователя заменяет главное меню: Специалисту по продажам ПО принадлежит форма с заключением договора на поставку товара с поставщику. Продавец вводит данные для оформления заказа, меняет статус заказа, и также смотрит список для отправки ПО. Бухгалтер печатает отчёты.

Кнопка Войти заходит в базу данных и у каждого пользователя отобразится своя форма.

### 3.3.2 Форма ввода товара (*Fpost,post*).

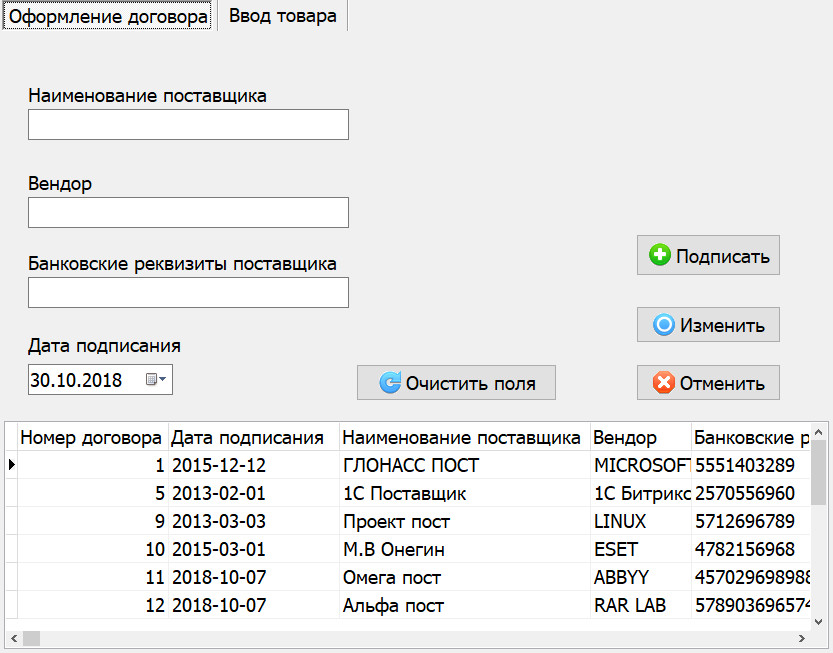


Рисунок 9–Внешний вид формы *Fpost*

Для ввода данных о поставщиках в таблицу я использовал LabeledEdit и DataTimePicker.

Кнопка Подписать добавляет нового поставщика в базу данных из введённых полей редактирования и переключает на ввод товара

Кнопка Изменить изменяет данные в таблице, включает живой запрос.

Кнопка Отменить отключает живой запрос.

Кнопка Очистить поля, обнуляет их для того чтобы пользователю было удобнее вводить данные.

После ввода данных о поставщиках в таблицу, включается ввод товара, для ввода товара в таблицу я использовал LabeledEdit и DBLookUpComboBox

Кнопка Добавить товар добавляет новый товар в базу данных из введённых полей редактирования и переключает на ввод товара

Кнопка Изменить изменяет данные в таблице, включает живой запрос.

Кнопка Отменить отключает живой запрос.

Кнопка Очистить поля, обнуляет их для того чтобы пользователю было удобнее вводить данные..

Листинг:

*procedure TFpost.bankKeyPress(Sender: TObject; var Key: Char);*

*begin*

*if not (Key in ['0'..'9', #8]) then begin*

*Key := #0;*

*messagebox(Application.Handle,'Ввод символа недопустим, вводите только Цифры','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);*

*end;1*

*end;*

*procedure TFpost.BCancClick(Sender: TObject);*

*begin*

*FDM.ADOQueryD.CancelBatch();*

*end;*

*procedure TFpost.Bclear2Click(Sender: TObject);*

*begin*

*namPO.Clear;*

*lang.Clear;*

*plat.Clear;*

*razr.clear;*

*kolkop.Clear;*

*stoim.Clear;*

*end;*

*procedure TFpost.BclearClick(Sender: TObject);*

*begin*

*nampost.Clear;*

*vend.Clear;*

*bank.Clear;*

*end;*

*procedure TFpost.BCnc2Click(Sender: TObject);*

*begin*

*FDM.ADOQueryT.CancelBatch();*

*end;*

*procedure TFpost.BSave2Click(Sender: TObject);*

*begin*

*FDM.ADOQueryT.UpdateBatch();*

*end;*

*procedure TFpost.BSaveClick(Sender: TObject);*

*begin*

*FDM.ADOQueryD.UpdateBatch();*

*end;*

*procedure TFpost.DBdogCellClick(Column: TColumn);*

*begin*

*DBLUpost.KeyValue := fdm.DataSourced.DataSet.FieldByName('ndog').AsInteger;*

*PageControl1.ActivePage:= TabSheet2;*

*end;*

*procedure TFpost.DtovClick(Sender: TObject);*

*begin*

*if (namPO.Text='') or (lang.Text='') or (razr.Text='') or (kolkop.Text='') or (stoim.Text='') then*

*begin*

*MessageBox(Application.Handle,'Проверьте ваши данные','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);*

*exit;*

*end;*

*FDM.ADOQueryT.Close;*

*FDM.ADOStoredProcInsT.Parameters.ParamByName('@ndog').Value:=FDM.ADOQueryD.FieldByName('ndog').AsString;*

*fdm.ADOStoredProcInsT.Parameters.ParamValues['@NamePo']:=namPO.Text;*

*FDM.ADOStoredProcInsT.Parameters.ParamByName('@ntipPO').Value:=FDM.ADOQueryTP.FieldByName('nTIPPO').AsString;*

*FDM.ADOStoredProcInsT.Parameters.ParamByName('@nkat').Value:=FDM.ADOQueryK.FieldByName('nkat').AsString;*

*FDM.ADOStoredProcInsT.Parameters.ParamByName('@ntiplis').Value:=FDM.ADOQueryTL.FieldByName('ntiplis').AsString;*

*fdm.ADOStoredProcInsT.Parameters.ParamValues['@plat']:=plat.Text;*

*fdm.ADOStoredProcInsT.Parameters.ParamValues['@yaz']:=lang.Text;*

*fdm.ADOStoredProcInsT.Parameters.ParamValues['@razrab']:=razr.Text;*

*fdm.ADOStoredProcInsT.Parameters.ParamValues['@kolvokop']:=StrToInt(kolkop.Text);*

*fdm.ADOStoredProcInsT.Parameters.ParamValues['@stomost']:=StrToCurr(stoim.Text);*

*FDM.ADOStoredProcInsT.ExecProc;*

*FDM.ADOQueryT.Close;*

*FDM.ADOQueryT.Open;*

*FDM.ADOQueryT.Last;*

*end;*

*procedure TFpost.FormActivate(Sender: TObject);*

*begin*

*FDM.ADOQueryD.Open; //включаем четыре списка для вывода информации*

*FDM.ADOQueryTP.Open;*

*FDM.ADOQueryK.Open;*

*FDM.ADOQueryTL.Open;*

*FDM.ADOQueryT.Open;*

*Fdm.ADOQueryD.OPEN;*

*dpost.Date:=Now;*

*end;*

*procedure TFpost.kolkopKeyPress(Sender: TObject; var Key: Char);*

*begin*

*if not (Key in ['0'..'9', #8]) then begin*

*Key := #0;*

*messagebox(Application.Handle,'Ввод символа недопустим, вводите только Цифры','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);*

*end;*

*end;*

*procedure TFpost.langKeyPress(Sender: TObject; var Key: Char);*

*begin*

*if (Key in ['0'..'9']) then begin*

*Key := #0;*

*messagebox(Application.Handle,'Ввод Цифры недопустим, вводите только Символы','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);*

*end;*

*end;*

*procedure TFpost.nampostKeyPress(Sender: TObject; var Key: Char);*

*begin*

*if (Key in ['0'..'9']) then begin*

*Key := #0;*

*messagebox(Application.Handle,'Ввод Цифры недопустим, вводите только Символы','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);*

*end;*

*end;*

*procedure TFpost.podpClick(Sender: TObject);*

*begin*

*if (nampost.Text='') or (vend.Text='') or (bank.Text='') then*

*begin*

*MessageBox(Application.Handle,'Проверьте ваши данные','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);*

*exit;*

*end;*

*FDM.ADOQueryD.Close;*

*fdm.ADOStoredProcInsD.Parameters.ParamValues['@Dpodpis']:=DateToStr(dpost.date);*

*fdm.ADOStoredProcInsD.Parameters.ParamValues['@NamePos']:=nampost.Text;*

*fdm.ADOStoredProcInsD.Parameters.ParamValues['@Vendor']:=vend.Text;*

*fdm.ADOStoredProcInsD.Parameters.ParamValues['@BankRekv']:=bank.Text;*

*FDM.ADOStoredProcInsD.ExecProc;*

*FDM.ADOQueryD.Close;*

*FDM.ADOQueryD.Open;*

*FDM.ADOQueryD.Last;*

*DBLUpost.KeyValue := fdm.DataSourced.DataSet.FieldByName('ndog').AsInteger;*

*PageControl1.ActivePage:= TabSheet2;*

*end;*

*procedure TFpost.stoimKeyPress(Sender: TObject; var Key: Char);*

*begin*

*if not (Key in ['0'..'9',',', #8]) then begin*

*Key := #0;*

*messagebox(Application.Handle,'Ввод символа недопустим, вводите только Цифры','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);*

*end;*

*end;*

*procedure TFpost.vendKeyPress(Sender: TObject; var Key: Char);*

*begin*

*if (Key in ['0'..'9']) then begin*

*Key := #0;*

*messagebox(Application.Handle,'Ввод Цифры недопустим, вводите только Символы','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);*

*end;*

*end;*

### 3.3.2 Форма ввода заказа (*Fzak,zak*).

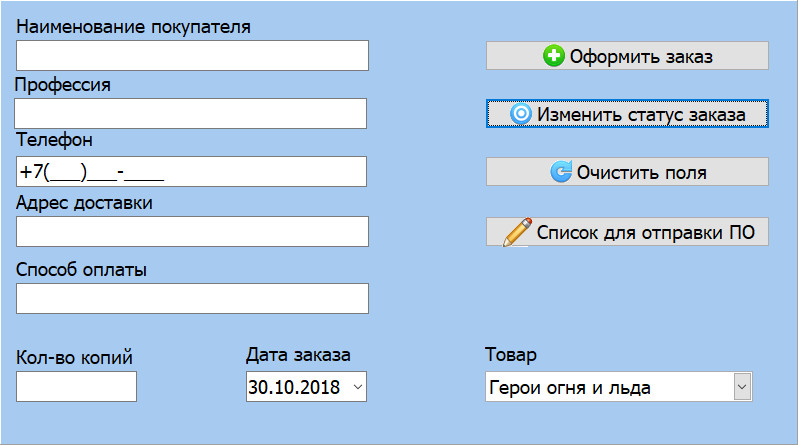


Рисунок 10–Внешний вид формы *Fzak*

Для ввода данных о заказах в таблицу я использовал LabeledEdit и DataTimePicker,DBLookUpComboBox.

Кнопка Оформить заказ добавляет новый заказ в базу данных из введённых полей редактирования и просит подтвердить оплату для отправки ПО.

Кнопка Изменить статус заказа включает общий список заказов и там можно изменить статус заказа.

Кнопка Список для отправки ПО открывает сводный отчёт для печати списка.

Кнопка Очистить поля, обнуляет их для того чтобы пользователю было удобнее вводить данные.

Листинг:

procedure TFzak.BCHANClick(Sender: TObject);

begin

FZAK2.Show();

end;

procedure TFzak.BClearClick(Sender: TObject);

begin

Namepok.Clear;

prof.Clear;

MaskEdit1.Clear;

kolkop.Clear;

sposopl.Clear;

adrd.Clear;

end;

procedure TFzak.BitBtn3Click(Sender: TObject);

begin

FDM.ADOQueryZ.CancelBatch;

end;

procedure TFzak.BspisClick(Sender: TObject);

begin

Fspis.Show;

end;

procedure TFzak.BTNADDClick(Sender: TObject);

begin

if (Namepok.Text='') or (prof.Text='') or (MaskEdit1.Text='') or (kolkop.Text='') or (adrd.Text='') or (sposopl.Text='') then

begin

MessageBox(Application.Handle,'Проверьте ваши данные','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);

exit;

end

ELSE BEGIN

FDM.ADOQueryZ.Close;

FDM.ADOStoredProcInsZ.Parameters.ParamByName('@ntov').Value:=FDM.ADOQueryT.FieldByName('Ntov').AsString;

fdm.ADOStoredProcInsZ.Parameters.ParamValues['@dzak']:=DateToStr(Dzak.Date);

fdm.ADOStoredProcInsZ.Parameters.ParamValues['@NamePok']:=Namepok.Text;

fdm.ADOStoredProcInsZ.Parameters.ParamValues['@prof']:=prof.Text;

fdm.ADOStoredProcInsZ.Parameters.ParamValues['@tel']:=MaskEdit1.text;

fdm.ADOStoredProcInsZ.Parameters.ParamValues['@Kookop']:=StrToInt(kolkop.text);

fdm.ADOStoredProcInsZ.Parameters.ParamValues['@AdresDost']:=adrd.text;

fdm.ADOStoredProcInsZ.Parameters.ParamValues['@sposopl']:=sposopl.text;

FDM.ADOStoredProcInsZ.ExecProc;

FDM.ADOQueryZ.Open;

FDM.ADOQueryZ.Last;

MessageBox(Application.Handle,'Ваш заказ оформлен, подтвердите оплату','ПродажаПО',MB\_OK+MB\_ICONINFORMATION);

Fzak2.Show;

END;

end;

procedure TFzak.FormActivate(Sender: TObject);

begin

fdm.ADOQueryT.Open;

fdm.ADOQueryZ.Open;

dzak.Date:=Now;

end;

procedure TFzak.kolkopKeyPress(Sender: TObject; var Key: Char);

begin

if not (Key in ['0'..'9', #8]) then begin

Key := #0;

messagebox(Application.Handle,'Ввод символа недопустим, вводите только Цифры','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);

end;

end;

procedure TFzak.MaskEdit1KeyPress(Sender: TObject; var Key: Char);

begin

if not (Key in ['0'..'9', #8]) then begin

Key := #0;

messagebox(Application.Handle,'Ввод символа недопустим, вводите только Цифры','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);

end;

*end;*

*procedure TFzak.NamepokKeyPress(Sender: TObject; var Key: Char);*

*begin*

*if (Key in ['0'..'9']) then begin*

*Key := #0;*

*messagebox(Application.Handle,'Ввод Цифры недопустим, вводите только Символы','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);*

*end;*

*end;*

*procedure TFzak.profKeyPress(Sender: TObject; var Key: Char);*

*begin*

*if (Key in ['0'..'9']) then begin*

*Key := #0;*

*messagebox(Application.Handle,'Ввод Цифры недопустим, вводите только Символы','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);*

*end;*

*end;*

*procedure TFzak.sposoplKeyPress(Sender: TObject; var Key: Char);*

*begin*

*if (Key in ['0'..'9']) then begin*

*Key := #0;*

*messagebox(Application.Handle,'Ввод Цифры недопустим, вводите только Символы','ПродажаПО ошибка',MB\_OK+MB\_ICONERROR);*

*end;*

*end;*

### 3.3.3 Форма изменения заказа (*Fzak2,zak2*).

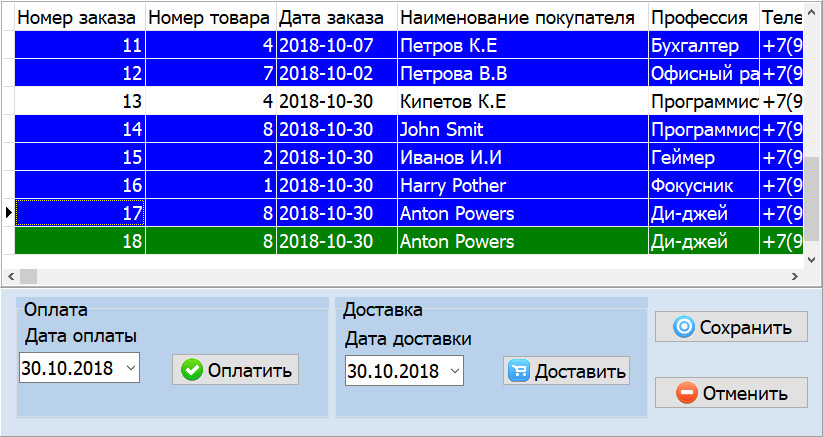


Рисунок 11–Внешний вид формы *Fzak2*

Для ввода данных о заказах в таблицу я использовал DataTimePicker

Кнопка Оплатить добавляет дату оплаты изменяет статус на 1, а также раскрашивает выбранную строку в синий цвет.

Кнопка Доставить добавляет дату доставки изменяет статус на 2, а также раскрашивает выбранную строку в зелёный цвет.

Кнопка Изменить изменяет данные в таблице, включает живой запрос.

Кнопка Отменить отключает живой запрос.

Листинг:

*procedure TFzak2.BcancClick(Sender: TObject);*

*begin*

*FDM.ADOQueryZ.CancelBatch();*

*end;*

*procedure TFzak2.BdostClick(Sender: TObject);*

*begin*

*fdm.ADOStoredProcDOST.Parameters.ParamValues['@zak']:=FDM.ADOQueryZ.FieldByName('Nzak').Value;*

*fdm.ADOStoredProcDOST.Parameters.ParamValues['@date']:=DateToStr(DDOST.Date);*

*FDM.ADOStoredProcDOST.ExecProc;*

*FDM.ADOQueryZ.Close;*

*FDM.ADOQueryZ.Open;*

*FDM.ADOQueryZ.Last;*

*end;*

*procedure TFzak2.BoplClick(Sender: TObject);*

*begin*

*fdm.ADOStoredProcOPKACH.Parameters.ParamValues['@zak']:=FDM.ADOQueryZ.FieldByName('Nzak').Value;*

*fdm.ADOStoredProcOPKACH.Parameters.ParamValues['@date1']:=DateToStr(DOPL.Date);*

*FDM.ADOStoredProcOPKACH.ExecProc;*

*FDM.ADOQueryZ.Close;*

*FDM.ADOQueryZ.Open;*

*FDM.ADOQueryZ.Last;*

*end;*

*procedure TFzak2.BsaveClick(Sender: TObject);*

*begin*

*FDM.ADOQueryZ.UpdateBatch();*

*end;*

*procedure TFzak2.DBGrid1DrawColumnCell(Sender: TObject; const Rect: TRect;*

*DataCol: Integer; Column: TColumn; State: TGridDrawState);*

*begin*

*if (FDM.ADOQueryZ.FieldByName('STAT').Value=2) then begin*

*DBGrid1.Canvas.Brush.Color:=Clgreen; //»зменить цвет фона*

*DBGrid1.Canvas.Font.Color:=ClWhite; //»зменить цвет шрифта*

*DBGrid1.Canvas.FillRect(Rect);*

*end;*

*if (FDM.ADOQueryZ.FieldByName('STAT').Value=1) then begin*

*DBGrid1.Canvas.Brush.Color:=Clblue; //»зменить цвет фона*

*DBGrid1.Canvas.Font.Color:=ClWhite; //»зменить цвет шрифта*

*DBGrid1.Canvas.FillRect(Rect);*

*end;*

*if (FDM.ADOQueryZ.FieldByName('STAT').Value=0) then begin*

*DBGrid1.Canvas.Brush.Color:=Clwhite; //»зменить цвет фона*

*DBGrid1.Canvas.Font.Color:=Clblack; //»зменить цвет шрифта*

*DBGrid1.Canvas.FillRect(Rect);*

*end;*

*DBGrid1.DefaultDrawColumnCell(Rect, DataCol, Column, State);*

*end;*

*procedure TFzak2.FormActivate(Sender: TObject);*

*begin*

*fdm.ADOQueryZ.Open;*

*dopl.Date:=Now;*

*DDOST.date:=Now;*

*end;*

### 3.3.4 Форма списка отправки ПО (*Fspis, spid*).

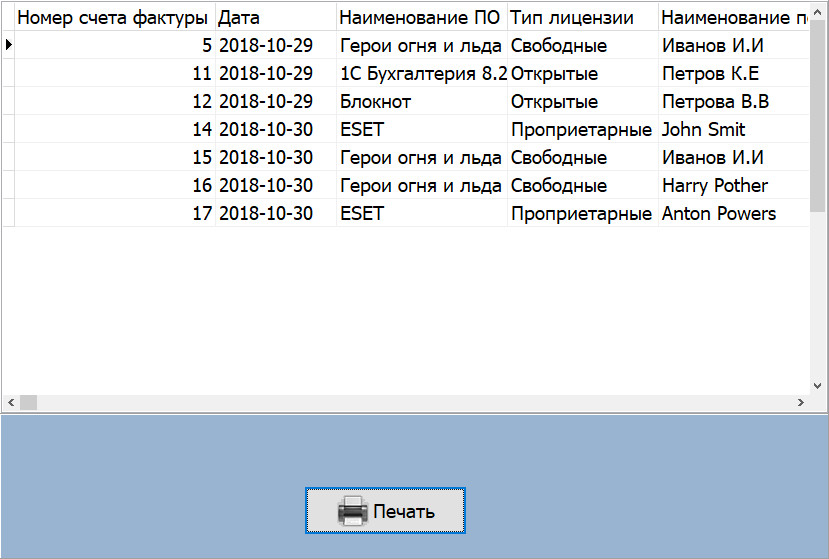


Рисунок 12–Внешний вид формы *Fspis.*

Для ввода данных в таблицу нет никаких данных, но сразу скажу, в данную таблицу попадают только те данные, в которых есть дата оплаты, но нет даты доставки.

Кнопка Печать выдаёт информацию в Excel.

Листинг:

*procedure TFspis.BTPrintClick(Sender: TObject);*

*begin*

*//Печать отчета в Excel*

*OE1:=CreateOleObject('Excel.Application'); //СоздатьOLE-объект Excel*

*OE1.WorkBooks.Add(); // Создать рабочую книгу Excel*

*OE1.Visible:=True; // Сделать Excel видимым*

*//Занести в ячейку значение- напечатать заголовок отчета*

*OE1.Cells[1,1].Value:='Список отпрааавки ПО';*

*OE1.Cells[2,1].Value:='№счета фактуры';*

*OE1.Cells[2,2].Value:='Дата ';*

*OE1.Cells[2,3].Value:='Наименование ПО';*

*OE1.Cells[2,4].Value:='Тип лицензии';*

*OE1.Cells[2,5].Value:='Тип ПО';*

*OE1.Cells[2,6].Value:='Вендор';*

*OE1.Cells[2,7].Value:='Наименование покупателя';*

*OE1.Cells[2,8].Value:='Адрес доставки';*

*OE1.Cells[2,9].Value:='Способ оплаты';*

*OE1.Cells[2,10].Value:='Стоимость продаж';*

*istr:=3; // Номер строки*

*FDM.ADOStoredProcotprav.First; //Перейти на первую запись*

*while not FDM.ADOStoredProcotprav.Eof do // Пока не будет достигнут конец файла*

*begin*

*//Заносить в ячейку Excel содержимое полей*

*OE1.Cells[istr,1].Value:= FDM.ADOStoredProcotprav.FieldByName('nzak').Value;*

*OE1.Cells[istr,2].Value:= FDM.ADOStoredProcotprav.FieldByName('dopl').AsString;*

*OE1.Cells[istr,3].Value:= FDM.ADOStoredProcotprav.FieldByName('namepo').AsString;*

*OE1.Cells[istr,4].Value:= FDM.ADOStoredProcotprav.FieldByName('tiplis').AsString;*

*OE1.Cells[istr,5].Value:= FDM.ADOStoredProcotprav.FieldByName('tippo').AsString;*

*OE1.Cells[istr,6].Value:= FDM.ADOStoredProcotprav.FieldByName('vendor').AsString;*

*OE1.Cells[istr,7].Value:= FDM.ADOStoredProcotprav.FieldByName('namepok').AsString;*

*OE1.Cells[istr,8].Value:= FDM.ADOStoredProcotprav.FieldByName('adresdost').AsString;*

*OE1.Cells[istr,9].Value:= FDM.ADOStoredProcotprav.FieldByName('sposopl').AsString;*

*OE1.Cells[istr,10].Value:= FDM.ADOStoredProcotprav.FieldByName('summa').Value;*

*//Перейти на следующую запись*

*FDM.ADOStoredProcotprav.Next;*

*//Увеличить номер строки в Excel*

*istr:=istr+1;*

*end;*

*if istr>4 then begin*

*OE1.CELLS[Istr,9].Value:='итого';*

*OE1.CELLS[Istr,10].formula:='=SUM(J3:J'+IntToStr(istr-1)+')';*

*end;*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Дополнительные команды для форматирования*

*OE1.Range['A3:j'+IntToStr(istr)].Select; // Выделить ячейки*

*//Обрамление сплошной линией всех выделенных ячеек*

*OE1.Selection.Borders[xlEdgeLeft].LineStyle:=xlContinuous;*

*OE1.Selection.Borders[xlEdgeRight].LineStyle:=xlContinuous;*

*OE1.Selection.Borders[xlEdgeTop].LineStyle:=xlContinuous;*

*OE1.Selection.Borders[xlEdgeBottom].LineStyle:=xlContinuous;*

*OE1.Selection.Borders[xlInsideVertical].LineStyle:=xlContinuous;*

*OE1.Selection.Borders[xlInsideHorizontal].LineStyle:=xlContinuous;*

*OE1.Range['D'+IntToStr(istr)+':D'+IntToStr(istr+1)].Select;*

*OE1.Selection.HorizontalAlignment := xlRight;*

*OE1.Selection.VerticalAlignment := xlCenter;*

*OE1.Range['A1:j'+IntToStr(istr)].Select;*

*OE1.Selection.ColumnWidth:=24;// Ширина столбца*

*OE1.Selection.RowHeight:=24;// Высота строки*

*// параметры печати - границы*

*OE1.ActiveSheet.PageSetup.LeftMargin:= 0.39;*

*OE1.ActiveSheet.PageSetup.RightMargin := 0.39;*

*OE1.ActiveSheet.PageSetup.TopMargin := 2.78;*

*OE1.ActiveSheet.PageSetup.BottomMargin := 0.78;*

*// параметры печати - ориентация*

*OE1.ActiveSheet.PageSetup.Orientation := xlLandscape;*

*OE1.ActiveSheet.PageSetup.Zoom := False ;*

*// Расположить на 1 странице по ширине OE1.ActiveSheet.PageSetup.FitToPagesWide := 1*

*// Расположить на 10 страницах по высоте OE1.ActiveSheet.PageSetup.FitToPagesTall := 10;*

*//Правый колонтитул внизу*

*OE1.ActiveSheet.PageSetup.RightFooter := DateToStr(Date) ;*

*//Предварительный просмотр*

*OE1.ActiveSheet.PrintPreview;*

*OE1.Range['j2:j'+IntToStr(istr)].Select;*

*OE1.Selection.NumberFormat:='0,0\_.'; //Формат числовой с 2 знаками после .*

*OE1.Selection.HorizontalAlignment := xlRight;//Выравнивание по правому краюпо горизонтали*

*OE1.Selection.VerticalAlignment := xlCenter; ;//Выравнивание по центру по вертикали*

*end;*

*procedure TFspis.FormActivate(Sender: TObject);*

*begin*

*fdm.ADOStoredProcotprav.ExecProc;*

*fdm.ADOStoredProcotprav.Close;*

*fdm.ADOStoredProcotprav.open;*

*end;*

### 3.3.5 Форма книги продаж (*Fbook, book*).

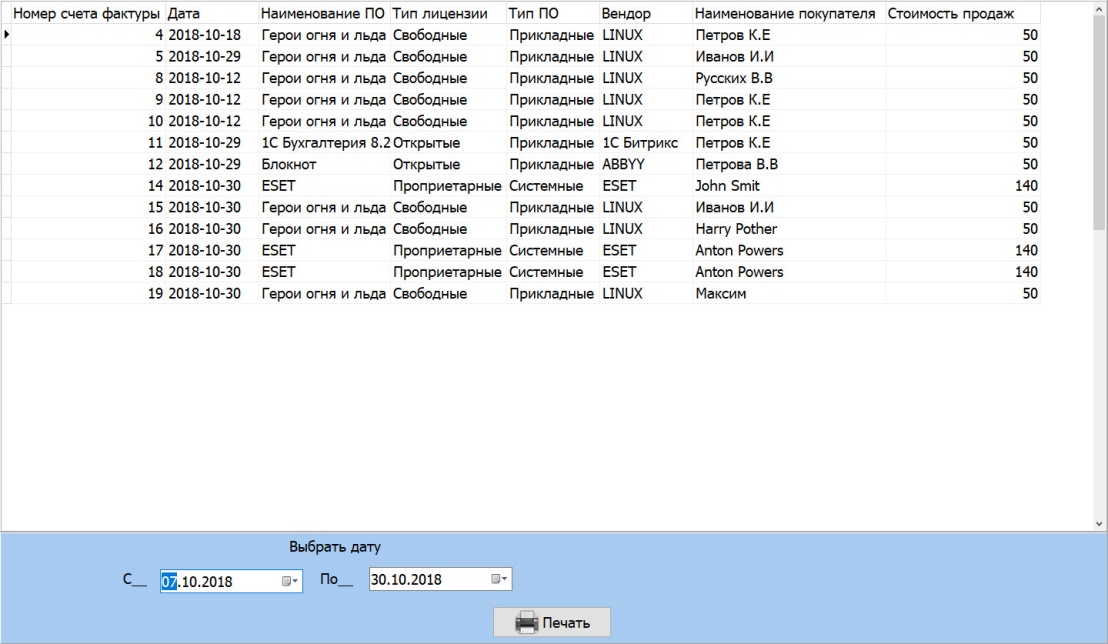


Рисунок 13 – Внешний вид формы *Fbook*

Для ввода данных в таблицу я использовал DateTimePicker,

Кнопка Печать выдаёт информацию в Excel.Код этой кнопки выглядит аналогично с Fspis

### 3.3.6 Форма рейтинга продаж (*Frait, rait*).

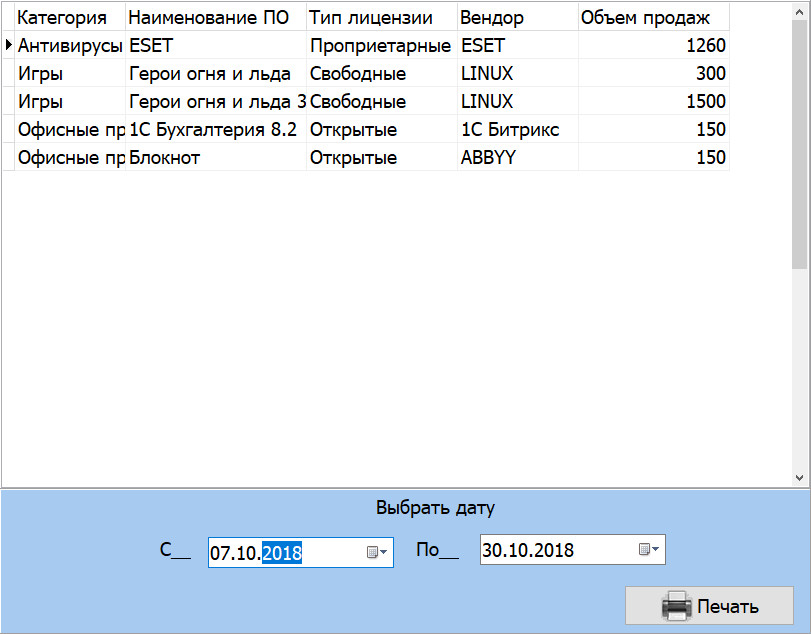


Рисунок 14– Внешний вид формы *Frait.*

Для ввода данных в таблицу я использовал DateTimePicker,

Кнопка Печать выдаёт информацию в Excel.Код этой кнопки выглядит аналогично с Fspia

### 3.3.7 Форма рейтинга продаж по категориям (*Fkrait, krait*).

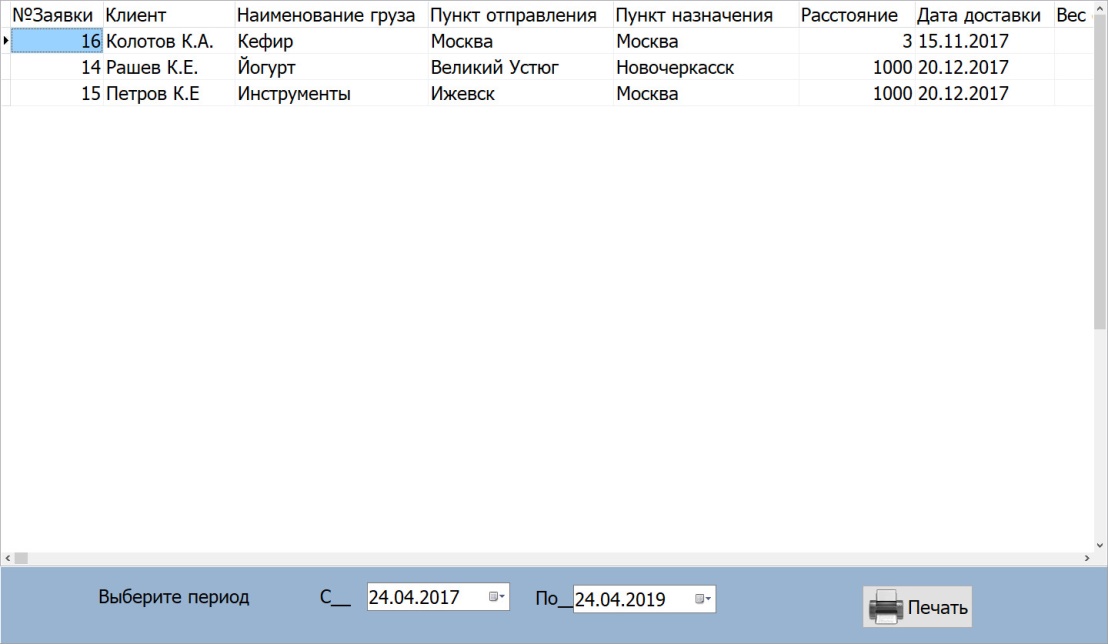


Рисунок 15 – Внешний вид формы *Fkrait.*

Для ввода данных в таблицу я использовал DateTimePicker,

Кнопка Печать выдаёт информацию в Excel.Код этой кнопки выглядит аналогично с Fspis

## 3.4 Разработка вэб-приложения

### 3.4.1 Главное меню – Index.php

На этой форме, я поставил просто таблицу с товарами. Данная страница представлена на рисунке 16.

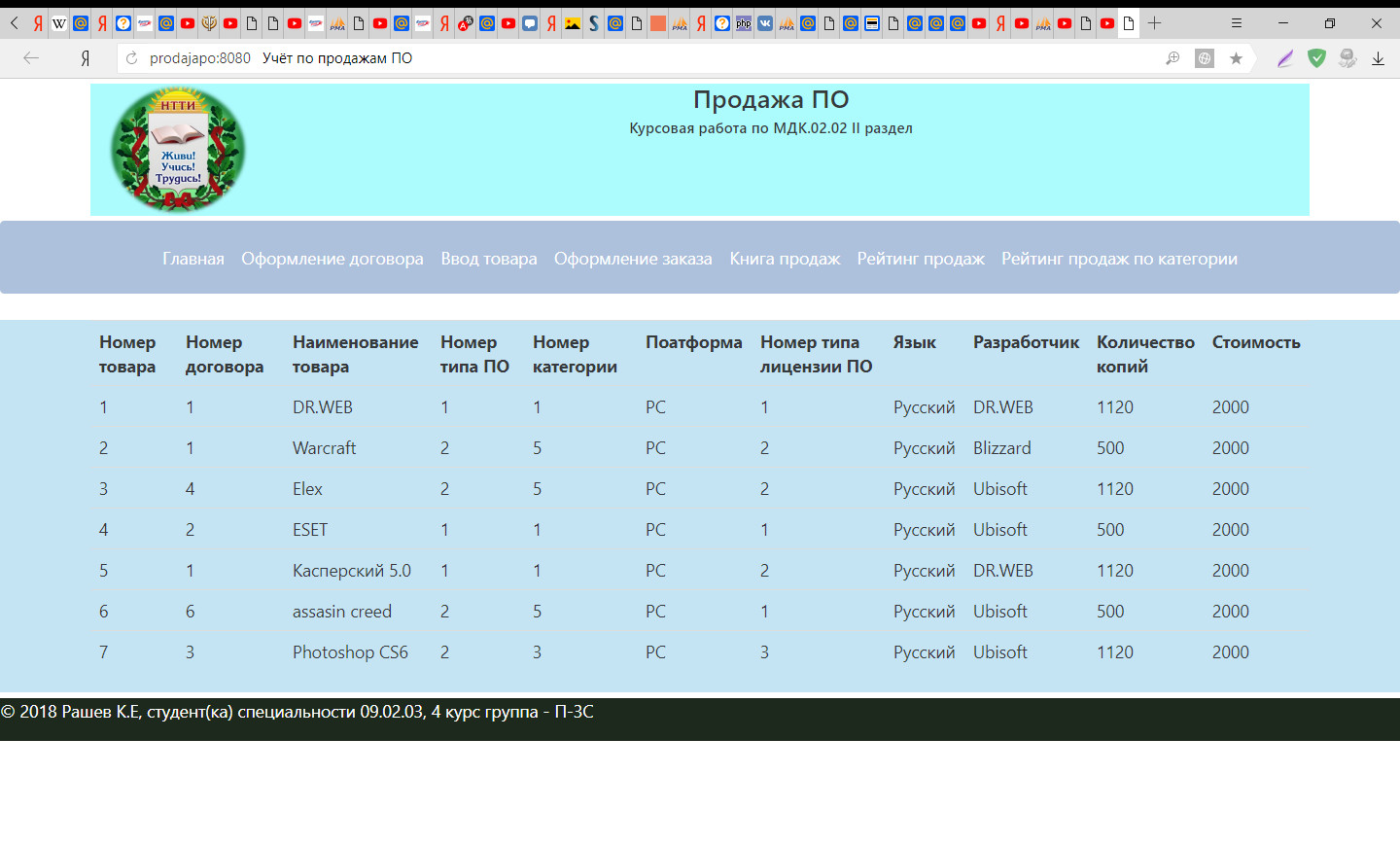


Рисунок 16 – Внешний вид формы *index.php.*

Ко всем страницам приложения подключим файлы с описанием области заголовка (header.php), меню страницы (navbar.php) и её нижней части (footer.php). В заголовке установим картинку и напишем текст заголовка. Эти элементы поместим в ячейки таблицы, чтобы выровнять их относительно друг друга. В блоке главного меню установим ссылки на все страница приложения. В подвале укажем краткую информацию об авторе сайта

Код подключаемого файла head.php

<!DOCTYPE html>

<head>

<title>Учёт по продажам ПО</title>

<link href= "/css/bootstrap.css" rel="stylesheet" type="text/css">

<link href= "../css/bootstrap-grid-3.3.1.css" rel="stylesheet" type="text/css">

<link href="../css/bootstrap-reboot.min.css" rel="stylesheet" type="text/css">

<link href= "/css/styl2e.css" rel="stylesheet" type="text/css">

<script src="../js/jquery-1.8.3.min.js"></script>

<script src="../js/jquery.maskedinput.min.js"></script>

<script src="../js/bootstrap.js"></script>

<script src="../js/bootstrap.bundle.min.js"></script>

<script src="../js/npm.js"></script>

<meta name="viewport" content="width=device-width, initial-scale=1" charset="utf-8">

</head>

<body>

Код подключаемого файла header.php:

<header>

<div class="container">

<div class="row">

<div class="col header" >

<div><img src="img/php\_mysql.png" width="12%" heigth="auto" align="left"></div>

<h3 align="center" > Продажа ПО </h3>

<h5 align="center">Курсовая работа по МДК.02.02 II раздел</h5>

</div>

</div>

</div>

</header>

Код подключаемого файла navbar.php

<nav class="navbar navbar-expand-lg " style="background: #10479856 !important">

<div class="container">

<div class="row">

<ul class="navbar-nav mr-auto">

<li><a class="nav-link text-white" href="../index.php">Главная</a></li>

<li><a class="nav-link text-white" href="../dog.php">Оформление договора</a></li>

<li><a class="nav-link text-white" href="../post.php">Ввод товара</a></li>

<li><a class="nav-link text-white" href="../zak.php">Оформление заказа</a></li>

<li><a class="nav-link text-white" href="../book.php">Книга продаж</a>

</li>

</ul>

<ul class="navbar-nav mr-auto">

<li><a class="nav-link text-white" href="../rait.php"> Рейтинг продаж </a></li>

<li><a class="nav-link text-white" href="../krait.php">Рейтинг продаж по категории</a></li>

</div>

</div>

</nav>

<div class="content">

Код подключаемого файла footer.php

<header>

<div class="container">

<div class="row">

<div class="col header" >

<div><img src="img/php\_mysql.png" width="12%" heigth="auto" align="left"></div>

<h3 align="center" > Продажа ПО </h3>

<h5 align="center">Курсовая работа по МДК.02.02 II раздел</h5>

</div>

</div>

</div>

</header>

Код:

<?php

$link = new mysqli('localhost','root','','prodpo');

if ($link->connect\_error)

{

echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';

die ('Cоединение не установлено');

}

$link->set\_charset('utf8');

include 'temp/head.php';

include 'temp/header.php';

include 'temp/navbar.php';

?>

<div class="container">

<div class="row">

<table class="table table-info table-responsive">

<tr>

<th>Номер товара</th>

<th>Номер договора</th>

<th>Наименование товара</th>

<th>Номер типа ПО</th>

<th>Номер категории</th>

<th>Платформа</th>

<th>Номер типа лицензии ПО</th>

<th>Язык</th>

<th>Разработчик</th>

<th>Количество копий</th>

<th>Стоимость</th>

</tr>

<?php

$results = $link->query("SELECT \* FROM tovar");

while($row = $results->fetch\_array())

{

echo '<tr>';

echo '<td>'.$row["Ntov"].'</td>';

echo '<td>'.$row["Ndog"].'</td>';

echo '<td>'.$row["NamePO"].'</td>';

echo '<td>'.$row["NtipPO"].'</td>';

echo '<td>'.$row["Nkat"].'</td>';

echo '<td>'.$row["Plat"].'</td>';

echo '<td>'.$row["Ntiplis"].'</td>';

echo '<td>'.$row["Yaz"].'</td>';

echo '<td>'.$row["Razrab"].'</td>';

echo '<td>'.$row["Kolvokop"].'</td>';

echo '<td>'.$row["Stomost"].'</td>';

echo '</tr>';

}

echo '</table>';

echo'</div>';

echo'</div>';

$results->free();

$link->close();

?>

<?php

include 'temp/footer.php';

?>

### 3.4.2 Оформление договора – Dog.php

Для ввода информации я использовал input type =text и data. После ввода данных, нажимаем на кнопку, введённые данные отобразятся в таблице. Реализация представлена на рисунке 17.

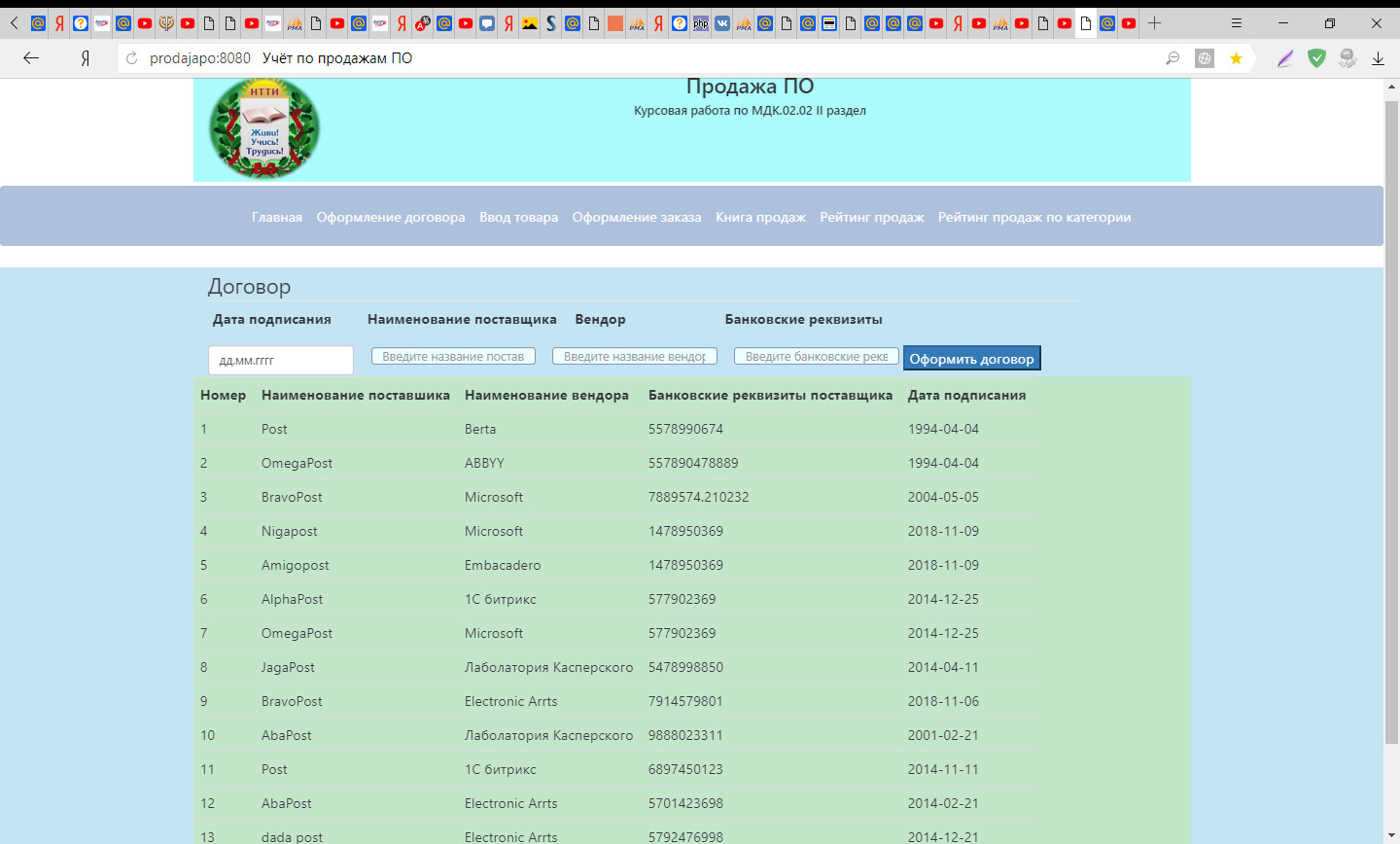


Рисунок 17 – Внешний вид формы *dog.php.*

Код:

<?php

$link = new mysqli('localhost','root','','prodPO');

if ($link->connect\_error)

{

echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';

die ('Cоединение не установлено');

}

$link->set\_charset('utf8');

if (!empty($\_POST)) {

$dat = $\_POST['dat'];

$post = $\_POST['post'];

$vend = $\_POST['vend'];

$bank = $\_POST['bank'];

$sql = $link->query("insert into DOGOVOR(NamePos,vendor,Bankrekv,Dpodpis) values( '$post', '$vend', '$bank','$dat')");

header("Location: http://".$\_SERVER['HTTP\_HOST'].dirname($\_SERVER['PHP\_SELF'])."/dog.php");

}

include 'temp/head.php';

include 'temp/header.php';

include 'temp/navbar.php';

?>

<div class="container">

<div class="row">

<div class="col">

<form method="post" role="form" class="form-inline">

<fieldset>

<legend>Договор</legend>

<div class="row">

<label for="" style="margin-left: 2%"> Дата подписания</label>

<label for="" STYLE="margin-left: 4%;"> Наименование поставщика</label>

<label for="" STYLE="margin-left: 2%;"> Вендор</label>

<label for="" STYLE="margin-left: 11%;"> Банковские реквизиты</label>

<div>

<div CLASS="col">

<input type="date" class="form-control" name="dat" placeholder="Введите дату подписания" requered>

<input type="text" class="form-control" name="post" placeholder="Введите название поставщика" requered>

<input type="text" class="form-control" name="vend" placeholder="Введите название вендора" requered>

<input type="text" class="form-control" id="b" name="bank" placeholder="Введите банковские реквизиты" requered>

<button type="submit" class="btn-primary" STYLE="margin-top: 2%">Оформить договор</button>

</div>

</fieldset>

</div>

<script>

$(document).ready(function()

{

$("#b").mask("9999999999");

}

);

</script>

</form>

</div>

</div>

<div class="container">

<div class="row">

<table class="table table-responsive table-success">

<tr>

<th>Номер</th>

<th>Наименование поставшика</th>

<th>Наименование вендора</th>

<th>Банковские реквизиты поставщика</th>

<th>Дата подписания</th>

</tr>

<?php

$results = $link->query("SELECT \* FROM dogovor");

while($row = $results->fetch\_array())

{

echo '<tr>';

echo '<td>'.$row["Ndog"].'</td>';

echo '<td>'.$row["NamePos"].'</td>';

echo '<td>'.$row["Vendor"].'</td>';

echo '<td>'.$row["Bankrekv"].'</td>';

echo '<td>'.$row["Dpodpis"].'</td>';

echo '</tr>';

}

echo '</table>';

echo'</div>';

echo'</div>';

$results->free();

$link->close();

?>

<?php

include 'temp/footer.php';

?>

### 3.4.3 Ввод товара – Post.php

Для ввода информации я использовал input type =text и select. После ввода данных, нажимаем на кнопку, введённые данные отобразятся в таблице. Реализация представлена на рисунке 18.

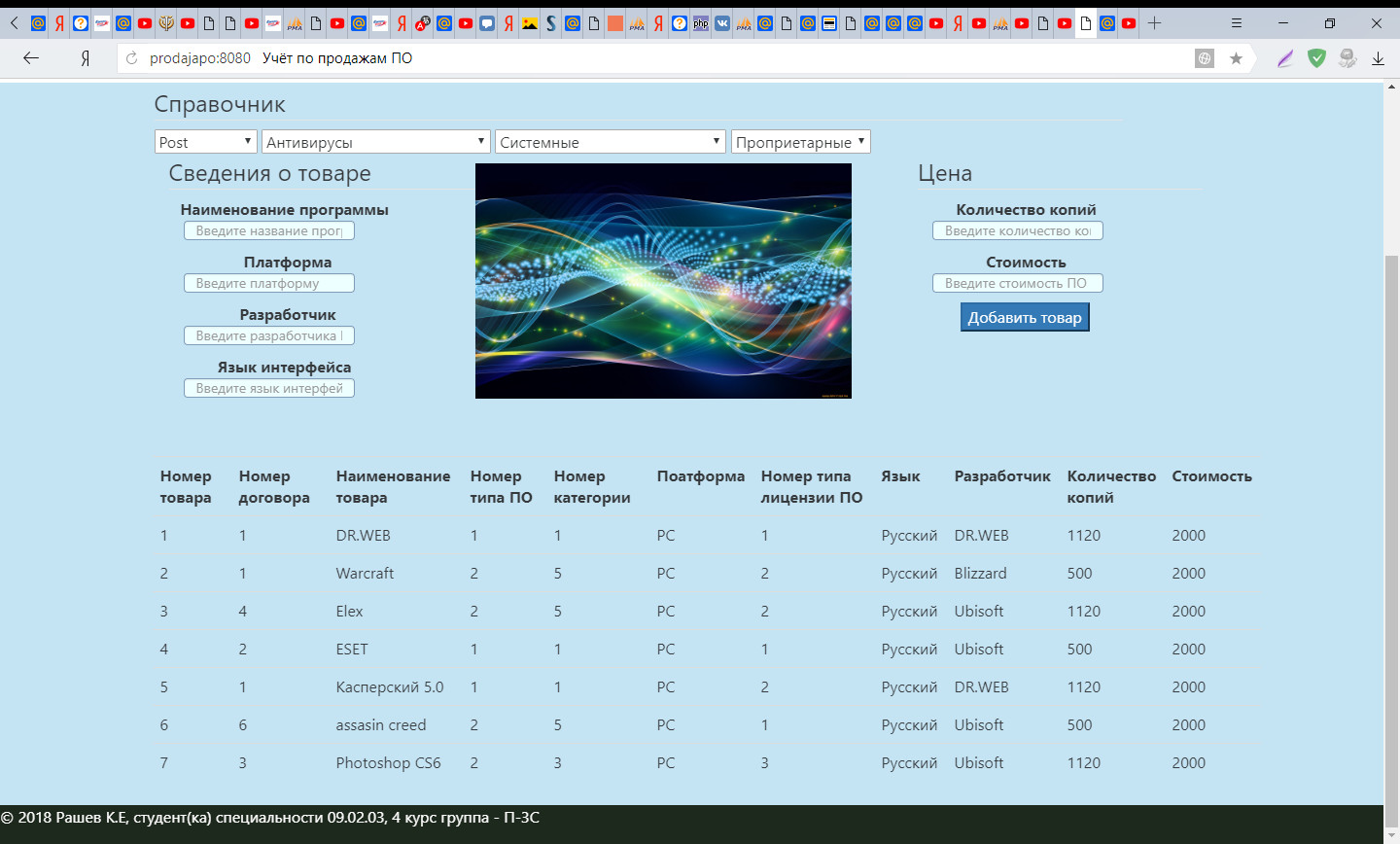


Рисунок 18 – Внешний вид формы *post.php.*

Код:

<?php  
$link = new mysqli('localhost','root','','prodpo');  
if ($link->connect\_error)  
{  
 echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';  
 die ('Cоединение не установлено');  
}  
$link->set\_charset('utf8');  
if (!empty($\_POST)) {  
 $dg = $\_POST['dogo'];  
 $kt = $\_POST['kat'];  
 $tp = $\_POST['tip'];  
 $tl = $\_POST['til'];  
 $np = $\_POST['nam'];  
 $pl = $\_POST['pla'];  
 $raz = $\_POST['razrab'];  
 $yz = $\_POST['yaz'];  
 $kol = $\_POST['kolkop'];  
 $st = $\_POST['stom'];  
 $sql = $link->query("INSERT INTO Tovar(Ndog,NamePO,NtipPO,Nkat,Plat,Ntiplis,Yaz,Razrab,Kolvokop,Stomost)   
values('$dg','$np','$tp','$kt','$pl','$tl','$yz','$raz','$kol','$st')");  
 *header*("Location: http://".$\_SERVER['HTTP\_HOST'].*dirname*($\_SERVER['PHP\_SELF'])."/post.php");  
}  
include 'temp/head.php';  
include 'temp/header.php';  
include 'temp/navbar.php';  
?>  
<div class="container">  
 <div class="row">  
 <div class="col">  
 <form method="post" role="form" class="form-inline">  
 <fieldset>  
 <legend>Справочник</legend>  
 <div class="form-group mb-lg-0 col-lg-offset-0">  
 <select name='dogo'>  
 <?  
 $que=$link->query("select *\** from Dogovor ");  
 while ($row = $que->fetch\_array())  
 {  
 echo '<option value="'.$row['Ndog'].'">'.$row['NamePos'].'</option>';  
 }  
 ?>  
 </select>  
 <select name='kat'>  
 <?  
 $query1=$link->query("SELECT *\** FROM Kate ");  
 while ($row = $query1->fetch\_array())  
 {  
 echo '<option value="'.$row['Nkat'].'">'.$row['Kat'].'</option>';  
 }  
 ?>  
 </select>  
 <select name='tip'>  
 <?  
 $query2=$link->query("SELECT *\** FROM tippo ");  
 while ($row = $query2->fetch\_array())  
 {  
 echo '<option value="'.$row['NtipPO'].'">'.$row['TipPO'].'</option>';  
 }  
 ?>  
 </select>  
 <select name='til'>  
 <?  
 $query3=$link->query("SELECT *\** FROM tiplise ");  
 while ($row = $query3->fetch\_array())  
 {  
 echo '<option value="'.$row['Ntiplis'].'">'.$row['Tiplis'].'</option>';  
 }  
 ?>  
 </select>  
 </div>  
 </fieldset>  
 <div class="container" style="float: left;">  
 <div class="form-group" >  
 <fieldset style="width: 150%;">  
 <legend>Сведения о товаре</legend>  
 <label for="" style="margin-left:-13%;">Наименование программы </label>  
 <input type="text" class="form-control" name="nam" placeholder="Введите название программы" requered>  
 <label for="" style="margin-left:-12%;">Платформа</label>  
 <input type="text" class="form-control" name="pla" placeholder="Введите платформу" requered>  
 <label for="" style="margin-left:-12%;">Разработчик</label>  
 <input type="text" class="form-control" name="razrab" placeholder="Введите разработчика ПО" requered>  
 <label for="" style="margin-left:-13%;">Язык интерфейса</label>  
 <input type="text" class="form-control" name="yaz" placeholder="Введите язык интерфейса ПО" requered>  
 </fieldset>  
 </div>  
 <img src="img/1920x1200\_1229770\_%5Bwww.ArtFile.ru%5D.jpg" style="width:50%; padding-left: 10% ">  
 <div class="form-group" style="float: right;">  
 <fieldset style="width: 150%;">  
 <legend>Цена</legend>  
 <label for="" style="margin-left:-12%;">Количество копий</label>  
 <input type="text" class="form-control" name="kolkop" placeholder="Введите количество копий" requered>  
 <label for="" style="margin-left:-12%;">Стоимость</label>  
 <input type="text" class="form-control" name="stom" placeholder="Введите стоимость ПО" requered>  
 <BR>  
 <button type="submit" class="btn-primary" style="margin-left: 15%">Добавить товар</button>  
 </fieldset>  
 </div>  
 <BR>  
 </div>  
 </form>  
 </div>  
</div>  
<br>  
<br>  
<?  
$link = new mysqli('localhost','root','','prodPO');  
if ($link->connect\_error)  
{  
 echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';  
 die ('Cоединение не установлено');  
}  
$link->set\_charset('utf8');  
?>  
<div class="container">  
 <div class="row">  
 <table class="table table-responsive">

<tr>  
 <th>Номер товара</th>  
 <th>Номер договора</th>  
 <th>Наименование товара</th>  
 <th>Номер типа ПО</th>  
 <th>Номер категории</th>  
 <th>Платформа</th>  
 <th>Номер типа лицензии ПО</th>  
 <th>Язык</th>  
 <th>Разработчик</th>  
 <th>Количество копий</th>  
 <th>Стоимость</th>  
 </tr>  
 <?php  
 $results = $link->query("SELECT *\** FROM tovar");  
 while($row = $results->fetch\_array())  
 {  
 echo '<tr>';  
 echo '<td>'.$row["Ntov"].'</td>';  
 echo '<td>'.$row["Ndog"].'</td>';  
 echo '<td>'.$row["NamePO"].'</td>';  
 echo '<td>'.$row["NtipPO"].'</td>';  
 echo '<td>'.$row["Nkat"].'</td>';  
 echo '<td>'.$row["Plat"].'</td>';  
 echo '<td>'.$row["Ntiplis"].'</td>';  
 echo '<td>'.$row["Yaz"].'</td>';  
 echo '<td>'.$row["Razrab"].'</td>';  
 echo '<td>'.$row["Kolvokop"].'</td>';  
 echo '<td>'.$row["Stomost"].'</td>';  
 echo '</tr>';  
 }  
 echo '</table>';  
 echo'</div>';  
 echo'</div>';  
 $results->free();  
 $link->close();  
 ?>  
 <?php  
 include 'temp/footer.php';  
 ?>

### 3.4.4 Оформление заказа – zak.php

Для ввода информации я использовал input type =text и data,select. После ввода данных, нажимаем на кнопку, введённые данные отобразятся в таблице, на следующей форме.Также можно перейти на изменить статус заказа по нажатию на кнопку, и посмотреть список ПО. Реализация представлена на рисунке 19.

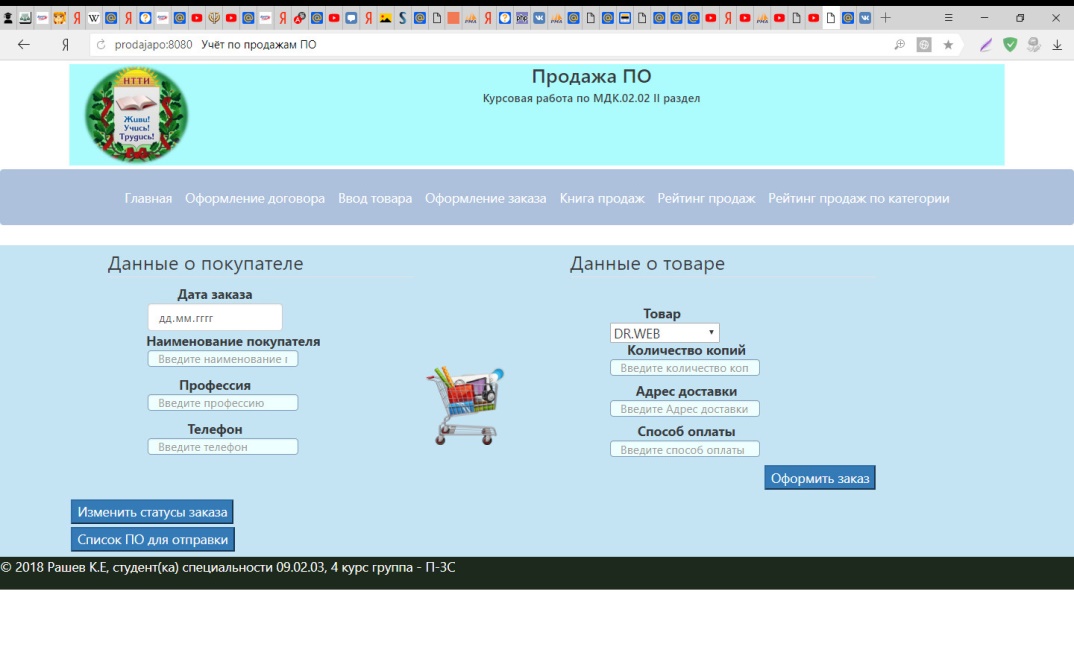


Рисунок 19 – Внешний вид формы *zak.php.*

Код:

<?php

$link = new mysqli('localhost','root','','prodpo');

if ($link->connect\_error)

{

echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';

die ('Cоединение не установлено');

}

$link->set\_charset('utf8');

if (!empty($\_POST)) {

$tr = $\_POST['tovo'];

$dz = $\_POST['dzaka'];

$np = $\_POST['nampok'];

$pr = $\_POST['pro'];

$te = $\_POST['tele'];

$kol = $\_POST['kolkop'];

$ad = $\_POST['adrd'];

$sp = $\_POST['spos'];

$sql = $link->query("insert into zakaz(Ntov,dzak,Namepok,Prof,Tel,Kookop,AdresDost,sposopl,stat)

values('$tr','$dz','$np','$pr','$te','$kol','$ad','$sp',0)");

header("Location: http://".$\_SERVER['HTTP\_HOST'].dirname($\_SERVER['PHP\_SELF'])."/upd\_zak.php");

}

include 'temp/head.php';

include 'temp/header.php';

include 'temp/navbar.php';

?>

<div class="container">

<div class="row">

<div class="col">

<form method="post" role="form" class="form-inline">

<div class="container">

<div class="form-group mb-2 col-5 " style="float: left;">

<fieldset>

<legend>Данные о покупателе</legend>

<label for="" style="margin-left: -15%;"> Дата заказа</label>

<input type="date" class="form-control" style="margin-left:13%;" name="dzaka" placeholder="Введите дату заказа" requered>

<br>

<label for="" style="margin-left: -9%;"> Наименование покупателя</label>

<input type="text" class="form-control" style="margin-left:13%;" name="nampok" placeholder="Введите наименование покупателя" requered>

<label for="" style="margin-left: -15%;"> Профессия</label>

<input type="text" class="form-control" style="margin-left:13%;" name="pro" placeholder="Введите профессию" requered>

<label for="" style="margin-left: -15%;"> Телефон</label>

<input type="text" class="form-control" style="margin-left:13%;" id="tel" name="tele" placeholder="Введите телефон" requered>

<script>

$(document).ready(function()

{

$("#tel").mask("+7-999-999-99-99");

}

);

</script>

</fieldset>

</div>

<img src="img/retail-finance-page\_512.png" style="width: 10%; margin-top: 15%">

<div class="form-group mb-2 col-5 " style="float: right;">

<fieldset>

<legend>Данные о товаре</legend>

<br>

<label for="" style="margin-left:-20%;"> Товар</label>

<select name='tovo' style="margin-left:13%;">

<?

$que2=$link->query("select \* from Tovar ");

while ($row = $que2->fetch\_array())

{

echo '<option value="'.$row['Ntov'].'">'.$row['NamePO'].'</option>';

}

?>

</select>

<label for="" style="margin-top: -1%; margin-left: -12%;">Количество копий</label>

<input type="text" class="form-control" style="margin-left:13%;" name="kolkop" placeholder="Введите количество копий" requered>

<label for="" style="margin-top: -1%; margin-left: -12%;"> Адрес доставки</label>

<input type="text" class="form-control" style="margin-left:13%;" name="adrd" placeholder="Введите Адрес доставки" requered>

<label for="" style="margin-top: -1%; margin-left: -12%;"> Способ оплаты </label>

<input type="text" class="form-control" style="margin-left:13%;" name="spos" placeholder="Введите способ оплаты" requered>

</fieldset>

<button type="submit" class="btn-primary" id="button" style="float:right" onclick="">Оформить заказ</button>

<script>

button.onclick = function() {

alert("Заказ оформлен,подтвердите оплату!");

};

</script>

</div>

</form>

</div>

</div>

<div class="form-inline">

<form method="post" style="float: right" role="form" class="form-inline" action="upd\_zak.php">

<button type="submit" class="btn-primary" style="float:right">Изменить статусы заказа</button>

</form>

<form method="post" style="float: right" role="form" class="form-inline" action="spisok.php">

<button type="submit" class="btn-primary" style="float:right">Список ПО для отправки</button>

</form>

</div>

</div>

<?

$link = new mysqli('localhost','root','','prodPO');

if ($link->connect\_error)

{

echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';

die ('Cоединение не установлено');

}

$link->set\_charset('utf8');

$link->close();

?>

<?php

include 'temp/footer.php';

?>

### 3.4.5 Изменение статуса заказа – upd\_zak.php

Для изменения статуса , я использовал data,select. После выбора товара и даты, нужно нажать оплатить или доставить. Если статус заказа равен 0- строка в таблице будет красной. . Если статус заказа равен 1- строка в таблице будет синей. . Если статус заказа равен 2- строка в таблице будет зелёной. Реализация представлена на рисунке 20.

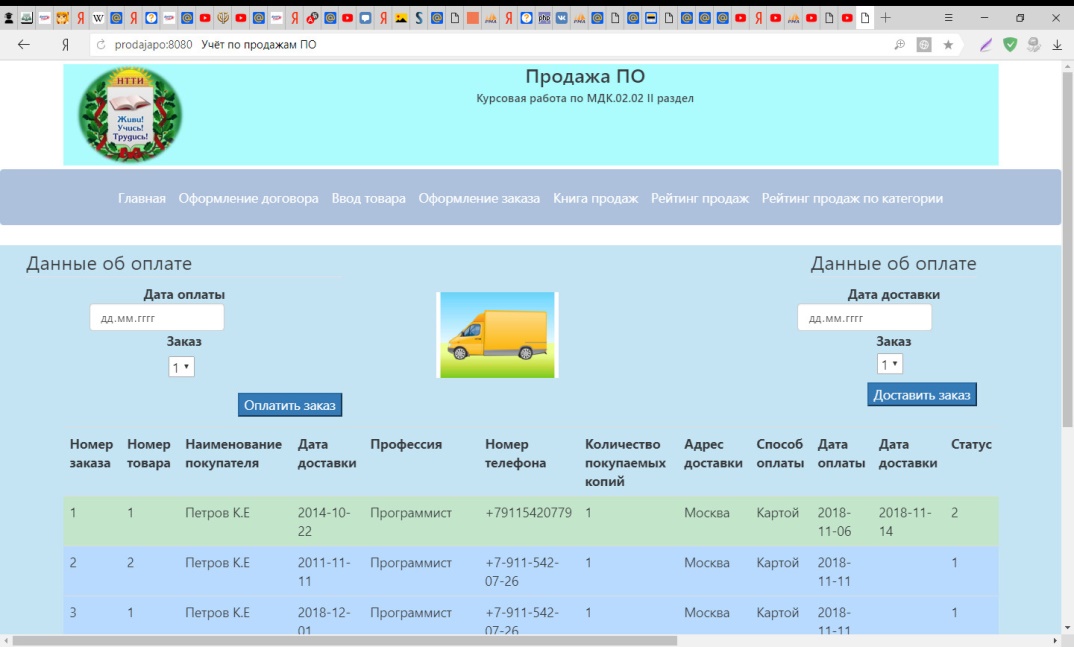


Рисунок 20 – Внешний вид формы upd\_*zak.php.*

Код:

<?php

$link = new mysqli('localhost','root','','prodpo');

if ($link->connect\_error)

{

echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';

die ('Cоединение не установлено');

}

$link->set\_charset('utf8');

include 'temp/head.php';

include 'temp/header.php';

include 'temp/navbar.php';

?>

<div class="container" style="float: left; margin-left:;: 90%;">

<form method="post" role="form" class="form-inline" style="float:left;" action="oplach.php">

<div class="form-group mb-2 col-5 " style="float: left; ">

<fieldset>

<legend>Данные об оплате</legend>

<label for="" > Дата оплаты</label>

<input type="date" class="form-control" style="margin-left: 20%;" name="opla" placeholder="Введите дату оплаты" requered>

<br>

<label for=""> Заказ</label>

<select name='zaka' style="margin-left:45%; margin-top: 2%; margin-bottom: 5%; ">

<?

$que=$link->query("select \* from zakaz ");

while ($row = $que->fetch\_array())

{

echo '<option value="'.$row['Nzak'].'">'.$row['Nzak'].'</option>';

}

?>

</select>

<br>

<button type="submit" class="btn-primary" style="float:right; margin-left: 10%">Оплатить заказ</button>

</fieldset>

</div>

<img src="img/delivery-van-vector.jpg" alt="" style="width:25%; padding-left: 10% ">

</form>

</div>

<div class="row">

<div class="container" style="float: right; margin-left: -100%;">

<form method="post" role="form" class="form-inline" action="dostav.php">

<div class="row " >

<fieldset>

<legend>Данные об оплате</legend>

<label for="" > Дата доставки</label>

<input type="date" class="form-control" style="margin-left: -8%;" name="dosta" placeholder="Введите дату оплаты" requered>

<br>

<label for=""> Заказ</label>

<select name='zaka2' style="margin-left: 40%; margin-top: 2%; margin-bottom: 5%; ">

<?

$que4=$link->query("select \* from zakaz ");

while ($row = $que4->fetch\_array())

{

echo '<option value="'.$row['Nzak'].'">'.$row['Nzak'].'</option>';

}

?>

</select>

<br>

<button type="submit" class="btn-primary" style="float:right;">Доставить заказ</button>

</fieldset>

</div>

</form>

</div>

</div>

<br>

<?

$link = new mysqli('localhost','root','','prodPO');

if ($link->connect\_error)

{

echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';

die ('Cоединение не установлено');

}

$link->set\_charset('utf8');

?>

<div class="container">

<div class="row">

<table class="table table-responsive">

<tr>

<th>Номер заказа</th>

<th>Номер товара</th>

<th>Наименование покупателя</th>

<th>Дата доставки</th>

<th>Профессия</th>

<th>Номер телефона</th>

<th>Количество покупаемых копий</th>

<th>Адрес доставки</th>

<th>Способ оплаты</th>

<th>Дата оплаты</th>

<th>Дата доставки</th>

<th>Статус</th>

</tr>

<?php

$results = $link->query("SELECT \* FROM zakaz");

while($row = $results->fetch\_array())

{

if($row["Stat"]==1) { echo "<tr class='table-primary'>";}

if ($row["Stat"]==0) {echo "<tr class='table-danger'>";}

if ($row["Stat"]==2) {echo "<tr class='table-success'>";}

echo '<td>'.$row["Nzak"].'</td>';

echo '<td>'.$row["Ntov"].'</td>';

echo '<td>'.$row["Namepok"].'</td>';

echo '<td>'.$row["Dzak"].'</td>';

echo '<td>'.$row["Prof"].'</td>';

echo '<td>'.$row["Tel"].'</td>';

echo '<td>'.$row["Kookop"].'</td>';

echo '<td>'.$row["AdresDost"].'</td>';

echo '<td>'.$row["Sposopl"].'</td>';

echo '<td>'.$row["Dopl"].'</td>';

echo '<td>'.$row["Ddost"].'</td>';

echo '<td>'.$row["Stat"].'</td>';

echo '</tr>';

}

echo '</table>';

echo' </div>';

echo '</div>';

$results->free();

$link->close();

?>

<?php

include 'temp/footer.php';

?>

Скрипт файл- oplach.php

Код

<?php  
$link = new mysqli('localhost','root','','prodPO');  
if ($link->connect\_error)  
{  
 echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';  
 die ('Cоединение не установлено');  
}  
$link->set\_charset('utf8');  
if (!empty($\_POST)) {  
 $dop = $\_POST['opla'];  
 $za = $\_POST['zaka'];  
 $sql = $link->query("Update Zakaz SET Stat=1, Dopl='$dop' where Nzak = '$za'");  
 $results = $link->query("SELECT *\** FROM Zakaz");  
 $results->free();  
 $link->close();  
 *header*("Location: http://".$\_SERVER['HTTP\_HOST'].*dirname*($\_SERVER['PHP\_SELF'])."/upd\_zak.php");  
}  
?>

Скрипт файл- dostav.php

Код

<?php

$link = new mysqli('localhost','root','','prodPO');

if ($link->connect\_error)

{

echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';

die ('Cоединение не установлено');

}

$link->set\_charset('utf8');

if (!empty($\_POST)) {

$dos = $\_POST['dosta'];

$za2 = $\_POST['zaka2'];

$sql = $link->query("Update Zakaz SET Stat=2, Ddost='$dos' where Nzak='$za2'");

$results = $link->query("SELECT \* FROM Zakaz");

$results->free();

$link->close();

header("Location: http://".$\_SERVER['HTTP\_HOST'].dirname($\_SERVER['PHP\_SELF'])."/upd\_zak.php");

}?>

### 3.4.6 Список для отправки ПО – spisok.php

Реализация списка для отправки ПО, предназначена для просмотра списка оплаченных заказов, которые необходимо доставить. Она представлена на рисунке 21.

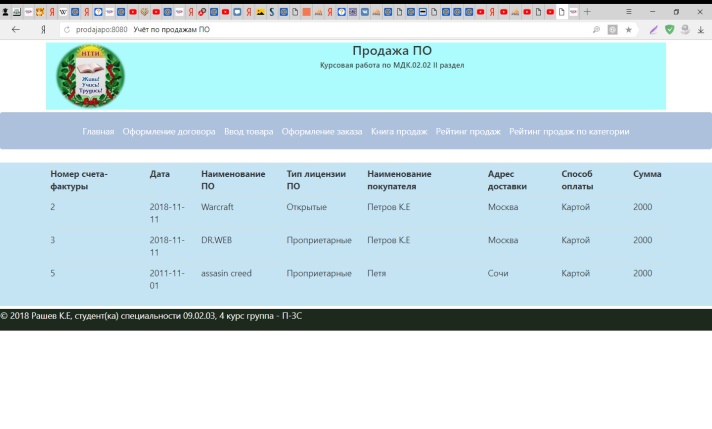


Рисунок 21 – Внешний вид формы *spisok.php.*

Код:

<?php

$link = new mysqli('localhost','root','','prodpo');

if ($link->connect\_error)

{

echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';

die ('Cоединение не установлено');

}

$link->set\_charset('utf8');

include 'temp/head.php';

include 'temp/header.php';

include 'temp/navbar.php';

?>

<div class="container">

<div class="row">

<table class="table table-responsive">

<tr>

<th>Номер счета-фактуры</th>

<th>Дата</th>

<th>Наименование ПО</th>

<th>Тип лицензии ПО</th>

<th>Наименование покупателя</th>

<th>Адрес доставки</th>

<th>Способ оплаты</th>

<th>Сумма</th>

</tr>

<?php

$results = $link->query("select zakaz.Nzak,zakaz.Dopl,TOVAR.NamePO,TIPLISE.Tiplis,

zakaz.Namepok,zakaz.AdresDost,zakaz.sposopl,

(zakaz.kookop\*tovar.Stomost) as summa from

zakaz,TIPLISE,TIPPO,TOVAR,DOGOVOR

where zakaz.Ntov=TOVAR.Ntov AND

TOVAR.Ntiplis=TIPLISE.Ntiplis

AND TOVAR.NtipPO=TIPPO.NtipPO AND DOGOVOR.Ndog=TOVAR.Ndog

AND zakaz.Dopl is not null and zakaz.Ddost is null AND Stat=1 ORDER BY nzak");

while($row = $results->fetch\_array())

{

echo '<tr>';

echo '<td>'.$row["Nzak"].'</td>';

echo '<td>'.$row["Dopl"].'</td>';

echo '<td>'.$row["NamePO"].'</td>';

echo '<td>'.$row["Tiplis"].'</td>';

echo '<td>'.$row["Namepok"].'</td>';

echo '<td>'.$row["AdresDost"].'</td>';

echo '<td>'.$row["sposopl"].'</td>';

echo '<td>'.$row["summa"].'</td>';

echo '</tr>';

}

echo '</table>';

echo'</div>';

echo'</div>';

$results->free();

$link->close();

?>

<?php

include 'temp/footer.php';

?>

### 3.4.7 Книга продаж – book.php

Для отображения данных необходимо указать период и затем нажать на кнопку. Результат представлен на рисунке 22.

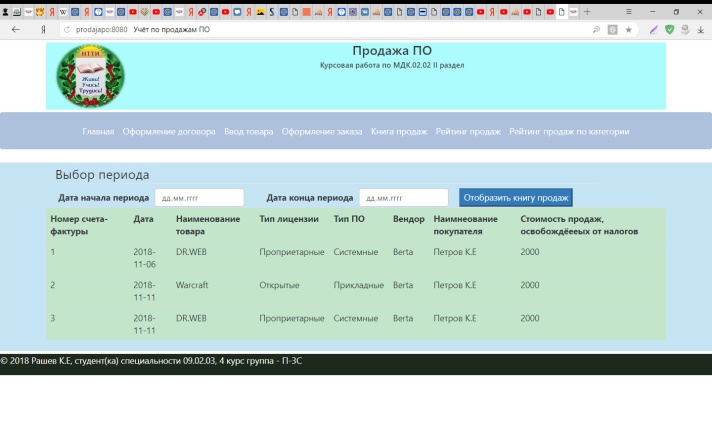


Рисунок 22 – Внешний вид формы *book.php.*

Код:

<?php

$link = new mysqli('localhost','root','','prodPO');

if ($link->connect\_error)

{

echo 'ошибка при попытке установки связи с сервером: '.$link->connect\_error.'<br>';

die ('Cоединение не установлено');

}

$link->set\_charset('utf8');

if (!empty($\_POST)) {

$d1 = $\_POST['d1'];

$d2 = $\_POST['d2'];

$results = $link->query("select zakaz.Nzak,zakaz.Dopl,TOVAR.NamePO,

TIPLISE.Tiplis,TipPO.TipPO,DOGOVOR.vendor,zakaz.

Namepok,TOVAR.Stomost from zakaz,TIPLISE,TIPPO,TOVAR,DOGOVOR

where zakaz.Ntov=TOVAR.Ntov AND

TOVAR.Ntiplis=TIPLISE.Ntiplis

AND TOVAR.NtipPO=TIPPO.NtipPO AND DOGOVOR.Ndog=TOVAR.Ndog

AND Dopl between '$d1' and '$d2' ORDER BY Nzak");

}

include 'temp/head.php';

include 'temp/header.php';

include 'temp/navbar.php';

?>

<div class="container">

<div class="row">

<div class="col">

<form method="post" role="form" class="form-inline">

<fieldset>

<legend>Выбор периода</legend>

<div class="row">

<label for="" style="margin-left: 2%; margin-right: 1% "> Дата начала периода</label>

<br>

<input type="date" class="form-control" name="d1" placeholder="Введите дату подписания" requered>

<br>

<label for="" STYLE="margin-left: 4%; margin-right: 1%; "> Дата конца периода</label>

<br>

<input type="date" class="form-control" name="d2" placeholder="Введите дату подписания" requered>

<button type="submit" class="btn-primary" STYLE="margin-left: 2%;">Отобразить книгу продаж</button>

</fieldset>

</div>

</form>

</div>

</div>

<div class="container">

<div class="row">

<table class="table table-responsive table-success">

<tr>

<th>Номер счета-фактуры</th>

<th>Дата</th>

<th>Наименование товара</th>

<th>Тип лицензии</th>

<th>Тип ПО</th>

<th>Вендор</th>

<th>Наимнеование покупателя</th>

<th>Стоимость продаж, освобождёееых от налогов</th>

</tr>

<?php

$d1 = $\_POST['d1'];

$d2 = $\_POST['d2'];

$results = $link->query("select zakaz.Nzak,zakaz.Dopl,TOVAR.NamePO,

TIPLISE.Tiplis,TipPO.TipPO,DOGOVOR.vendor,zakaz.

Namepok,TOVAR.Stomost from zakaz,TIPLISE,TIPPO,TOVAR,DOGOVOR

where zakaz.Ntov=TOVAR.Ntov AND

TOVAR.Ntiplis=TIPLISE.Ntiplis

AND TOVAR.NtipPO=TIPPO.NtipPO AND DOGOVOR.Ndog=TOVAR.Ndog

AND Dopl between '$d1' and '$d2' ORDER BY Nzak");

while($row = $results->fetch\_array())

{

echo '<tr>';

echo '<td>'.$row["Nzak"].'</td>';

echo '<td>'.$row["Dopl"].'</td>';

echo '<td>'.$row["NamePO"].'</td>';

echo '<td>'.$row["Tiplis"].'</td>';

echo '<td>'.$row["TipPO"].'</td>';

echo '<td>'.$row["vendor"].'</td>';

echo '<td>'.$row["Namepok"].'</td>';

echo '<td>'.$row["Stomost"].'</td>';

echo '</tr>';

}

echo '</table>';

echo'</div>';

echo'</div>';

$results->free();

$link->close();

?>

<?php

include 'temp/footer.php';

?>

### 3.4.8 Рейтинг продаж – rait.php

Для отображения данных необходимо указать период и затем нажать на кнопку. Результат представлен на рисунке 23.

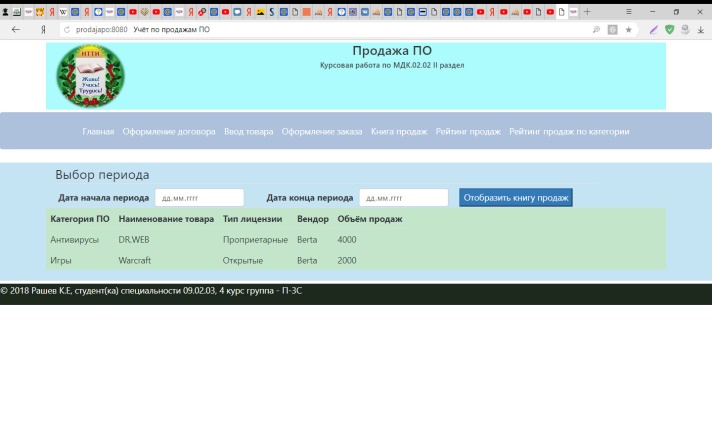


Рисунок 23– Внешний вид формы *rait.php.*

### 3.4.9 Рейтинг продаж – krait.php

Для отображения данных необходимо указать период и затем нажать на кнопку. Результат представлен на рисунке 24.

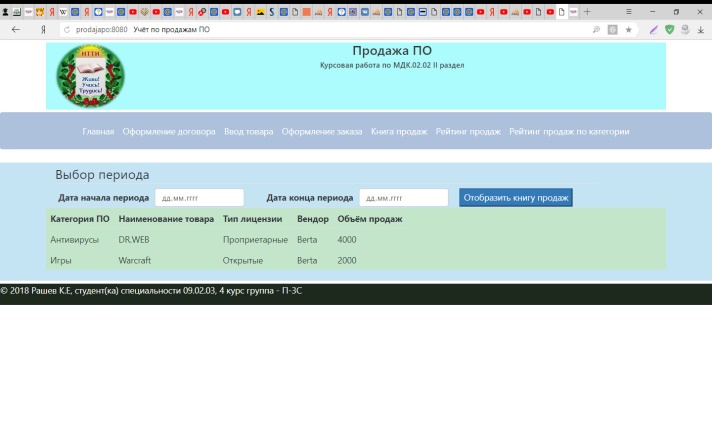


Рисунок 24– Внешний вид формы k*rait.php.*

## 3.5Разработка клиентского программного обеспечения на C#

### 3.5.1 Форма авторизации и главного меню (*Favto, form1.cs*).

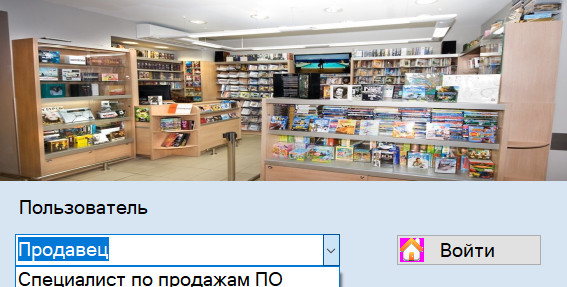


Рисунок 25–Внешний вид формы *Favto*

Для ввода данных в таблицу я использовал ComboBox

Выбор пользователя заменяет главное меню: Специалисту по продажам ПО принадлежит форма с заключением договора на поставку товара с поставщику. Продавец вводит данные для оформления заказа, меняет статус заказа, и также смотрит список для отправки ПО. Бухгалтер печатает отчёты.

Кнопка Войти заходит в базу данных и у каждого пользователя отобразится своя форма.

Листинг:

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

{

if (comboBox1.SelectedIndex == 0)

{

Fpost frm = new Fpost();

frm.ShowDialog();

}

if (comboBox1.SelectedIndex == 1)

{

Fzak frm = new Fzak();

frm.ShowDialog();

}

if (comboBox1.SelectedIndex == 2)

{

otch frm = new otch();

frm.ShowDialog();

}

}

### 3.5.2 Форма ввода товара (*Fpost.Form2.cs*).

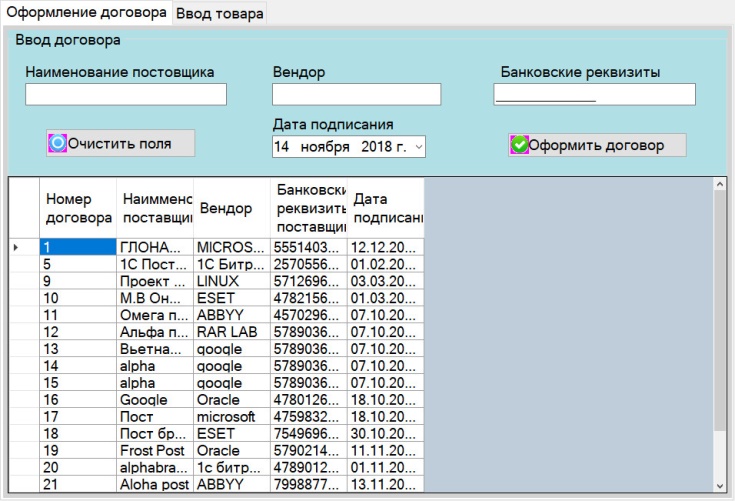


Рисунок 26–Внешний вид формы *Fpost*

Для ввода данных о поставщиках в таблицу я использовал Textbox и DataTimePicker.

Кнопка Подписать добавляет нового поставщика в базу данных из введённых полей редактирования и переключает на ввод товара

Кнопка Очистить поля, обнуляет их для того чтобы пользователю было удобнее вводить данные.

После ввода данных о поставщиках в таблицу, включается ввод товара, для ввода товара в таблицу я использовал LabeledEdit и DBLookUpComboBox

Кнопка Добавить товар добавляет новый товар в базу данных из введённых полей редактирования и переключает на ввод товара

Кнопка Очистить поля, обнуляет их для того чтобы пользователю было удобнее вводить данные..

Листинг:

public string ConnectionString = "";

public void conn(string CS, string cmdT, DataGridView dgv)

{

SqlDataAdapter Adapter = new SqlDataAdapter(cmdT, CS);

DataSet ds = new DataSet();

Adapter.Fill(ds, "Table");

dgv.DataSource = ds.Tables["Table"].DefaultView;

}

public void conn2(string CS, string cmdT, ComboBox CB, string field1, string field2)

{

//создание экземпляра адаптера

SqlDataAdapter Adapter = new SqlDataAdapter(cmdT, CS);

//создание объекта DataSet (набор данных)

DataSet ds = new DataSet();

Adapter.Fill(ds, "Table");

// привязка ComboBox к таблице БД

CB.DataSource = ds.Tables["Table"];

CB.DisplayMember = field1; //установка отображаемого в списке поля

CB.ValueMember = field2; //установка ключевого поля

}

public string select\_dog = "SELECT Ndog as [Номер договора], NamePos as [Наимменование поставщика], vendor as Вендор,BankRekv as [Банковские реквизиты поставщика], Dpodpis as [Дата подписания] FROM Dogovor";

public string select\_tov = "SELECT Ntov as [Номер товара],Ndog as [Номер договора],NamePO as [Наименование ПО],NtipPO as [Номер типа ПО], Nkat as [Номер категорий],"

+ "Plat as [Платформа],Ntiplis as [Номер типа лицензии],Yaz as [Язык],Razrab as [Разработчик],"

+ "Kolvokop as [Количество копий],Stomost [Стоимость 1-ой копии] FROM TOVAR";

const string select\_kat = "SELECT nkat as [Номер категорий], Kat as Название from Kate";

const string select\_tp = "SELECT ntippo as [Номер типа ПО], tippo as Название from Tippo";

const string select\_tl = "SELECT ntiplis as [Номер типа лицензии], tiplis as Название from Tiplise";

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

{ // Создадим новое подключение, в качестве параметра укажем строку подключения //ConnectionString.

SqlConnection conn1 = new SqlConnection();

conn1.ConnectionString = ConnectionString;

//Теперь можно устанавливать соединение, вызывая метод Open объекта

conn1.Open();

//создаем новый экземпляр SQLCommand

SqlCommand cmd = conn1.CreateCommand();

//определяем тип SQLCommand=StoredProcedure

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "[oformdog]";

//создаем параметр

cmd.Parameters.Add("@NAMEpos", SqlDbType.VarChar, 50);

cmd.Parameters["@NAMEpos"].Value = nampost.Text;

cmd.Parameters.Add("@Dpodpis", SqlDbType.Date, 70);

cmd.Parameters["@Dpodpis"].Value = dtp1.Value;

cmd.Parameters.Add("@vendor", SqlDbType.VarChar, 50);

cmd.Parameters["@vendor"].Value = vend.Text;

cmd.Parameters.Add("@bankrekv", SqlDbType.VarChar, 50);

cmd.Parameters["@bankrekv"].Value = bank.Text;

cmd.ExecuteScalar();

conn2(ConnectionString, select\_dog, comboBox1, "Наимменование поставщика", "Номер договора");

conn(ConnectionString, select\_dog, dgv1);

tabControl1.SelectedTab = tabPage2;

}

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

{

nampost.Text=string.Empty;

vend.Text = string.Empty;

bank.Text = string.Empty;

}

private void dgv1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

comboBox1.SelectedValue = dgv1[0, dgv1.CurrentRow.Index].Value.ToString();

tabControl1.SelectedTab = tabPage2;

}

private void bank\_KeyPress(object sender, KeyPressEventArgs e)

{

ponam(e);

}

private static void ponam(KeyPressEventArgs e)

{

if (Char.IsNumber(e.KeyChar) | (e.KeyChar == Convert.ToChar(",")) | e.KeyChar == '\b') return;

else

{

MessageBox.Show("Введите число",

"Сообщение");

e.Handled = true;

}

}

private static void pocha(KeyPressEventArgs e)

{

if (!Char.IsDigit(e.KeyChar)) return;

else

{

MessageBox.Show("Введите букву",

"Сообщение");

e.Handled = true;

}

}

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

{

// Создадим новое подключение, в качестве параметра укажем строку подключения //ConnectionString.

SqlConnection conn1 = new SqlConnection();

conn1.ConnectionString = ConnectionString;

//Теперь можно устанавливать соединение, вызывая метод Open объекта

conn1.Open();

//создаем новый экземпляр SQLCommand

SqlCommand cmd = conn1.CreateCommand();

//определяем тип SQLCommand=StoredProcedure

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "[vvodtov]";

//создаем параметр

cmd.Parameters.Add("@Ndog", SqlDbType.Int, 4);

cmd.Parameters["@Ndog"].Value = comboBox1.SelectedValue;

cmd.Parameters.Add("@nkat", SqlDbType.Int, 4);

cmd.Parameters["@nkat"].Value = comboBox2.SelectedValue;

cmd.Parameters.Add("@ntippo", SqlDbType.Int, 4);

cmd.Parameters["@ntippo"].Value = comboBox3.SelectedValue;

cmd.Parameters.Add("@ntiplis", SqlDbType.Int, 4);

cmd.Parameters["@ntiplis"].Value = comboBox4.SelectedValue;

cmd.Parameters.Add("@namepo", SqlDbType.VarChar, 70);

cmd.Parameters["@namepo"].Value = nampo.Text;

cmd.Parameters.Add("@plat", SqlDbType.VarChar, 10);

cmd.Parameters["@plat"].Value = plat.Text;

cmd.Parameters.Add("@yaz", SqlDbType.VarChar, 50);

cmd.Parameters["@yaz"].Value = lang.Text;

cmd.Parameters.Add("@razrab", SqlDbType.VarChar, 70);

cmd.Parameters["@razrab"].Value = razrab.Text;

cmd.Parameters.Add("@kolvokop", SqlDbType.Int,10);

cmd.Parameters["@kolvokop"].Value = kol.Text;

cmd.Parameters.Add("@stomost", SqlDbType.Money);

cmd.Parameters["@stomost"].Value = stom.Text;

cmd.ExecuteScalar();

conn(ConnectionString, select\_tov, dgv2);

}

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

{

nampo.Text = string.Empty;

lang.Text = string.Empty;

plat.Text = string.Empty;

razrab.Text = string.Empty;

stom.Text = string.Empty;

kol.Text = string.Empty;

}

private void kol\_KeyPress(object sender, KeyPressEventArgs e)

{

ponam(e);

}

private void stom\_KeyPress(object sender, KeyPressEventArgs e)

{

ponam(e);

}

private void lang\_KeyPress(object sender, KeyPressEventArgs e)

{ pocha(e); }

### 3.5.3 Форма ввода заказа (*Fzak,form3.cs*).

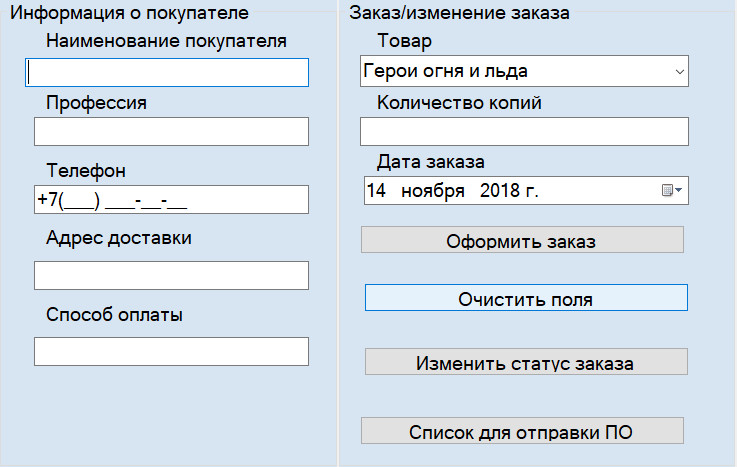


Рисунок 27–Внешний вид формы *Fzak*

Для ввода данных о заказах в таблицу я использовал TextBox и DataTimePicker,ComboBox.

Кнопка Оформить заказ добавляет новый заказ в базу данных из введённых полей редактирования и просит подтвердить оплату для отправки ПО.

Кнопка Изменить статус заказа включает общий список заказов и там можно изменить статус заказа.

Кнопка Список для отправки ПО открывает сводный отчёт для печати списка.

Кнопка Очистить поля, обнуляет их для того чтобы пользователю было удобнее вводить данные.

Листинг:

public string ConnectionString = "";

private void conn2(string CS, string cmdT, ComboBox CB, string field1, string field2)

{

//создание экземпляра адаптера

SqlDataAdapter Adapter = new SqlDataAdapter(cmdT, CS);

//создание объекта DataSet (набор данных)

DataSet ds = new DataSet();

Adapter.Fill(ds, "Table");

// привязка ComboBox к таблице БД

CB.DataSource = ds.Tables["Table"];

CB.DisplayMember = field1; //установка отображаемого в списке поля

CB.ValueMember = field2; //установка ключевого поля

}

const string select\_tov = "SELECT Ntov as Номер, NamePO as Наименование FROM TOVAR";

private static void ponam(KeyPressEventArgs e)

{

if (Char.IsNumber(e.KeyChar) | (e.KeyChar == Convert.ToChar(",")) | e.KeyChar == '\b') return;

else

{

MessageBox.Show("Введите число",

"Сообщение");

e.Handled = true;

}

}

private static void pocha(KeyPressEventArgs e)

{

if (!Char.IsDigit(e.KeyChar)) return;

else

{

MessageBox.Show("Введите букву",

"Сообщение");

e.Handled = true;

}

}

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

{

// Создадим новое подключение, в качестве параметра укажем строку подключения //ConnectionString.

SqlConnection conn1 = new SqlConnection();

conn1.ConnectionString = ConnectionString;

//Теперь можно устанавливать соединение, вызывая метод Open объекта

conn1.Open();

//создаем новый экземпляр SQLCommand

SqlCommand cmd = conn1.CreateCommand();

//определяем тип SQLCommand=StoredProcedure

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "[oformzak]";

//создаем параметр

cmd.Parameters.Add("@NAMEpok", SqlDbType.VarChar, 50);

cmd.Parameters["@NAMEpok"].Value = nampok.Text;

cmd.Parameters.Add("@Dzak", SqlDbType.Date, 70);

cmd.Parameters["@Dzak"].Value = dtp1.Value;

cmd.Parameters.Add("@prof", SqlDbType.VarChar, 50);

cmd.Parameters["@prof"].Value = prof.Text;

cmd.Parameters.Add("@tel", SqlDbType.VarChar, 50);

cmd.Parameters["@tel"].Value = tel.Text;

cmd.Parameters.Add("@adresdost", SqlDbType.VarChar, 50);

cmd.Parameters["@adresdost"].Value = adrd.Text;

cmd.Parameters.Add("@sposopl", SqlDbType.VarChar, 50);

cmd.Parameters["@sposopl"].Value = sposopl.Text;

cmd.Parameters.Add("@kookop", SqlDbType.Int, 10);

cmd.Parameters["@kookop"].Value = kolkop.Text;

cmd.Parameters.Add("@Ntov", SqlDbType.Int, 4);

cmd.Parameters["@Ntov"].Value = comboBox1.SelectedValue;

cmd.ExecuteScalar();

MessageBox.Show("Заказ оформлен, пожалуйста подтвердите оплату","ПродажаПО");

Fstat frm = new Fstat();

frm.ShowDialog();

}

private void nampok\_KeyPress(object sender, KeyPressEventArgs e)

{

pocha(e);

}

private void prof\_KeyPress(object sender, KeyPressEventArgs e)

{

pocha(e);

}

private void sposopl\_KeyPress(object sender, KeyPressEventArgs e)

{

pocha(e);

}

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

{

Fstat frm = new Fstat();

frm.ShowDialog();

}

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

{

nampok.Text = string.Empty;

prof.Text = string.Empty;

tel.Text = string.Empty;

adrd.Text = string.Empty;

sposopl.Text = string.Empty;

kolkop.Text = string.Empty;

}

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

{

otpravka frm = new otpravka();

frm.ShowDialog();

}

### 3.5.4 Форма изменения заказа (*Fstat,form4.cs*).

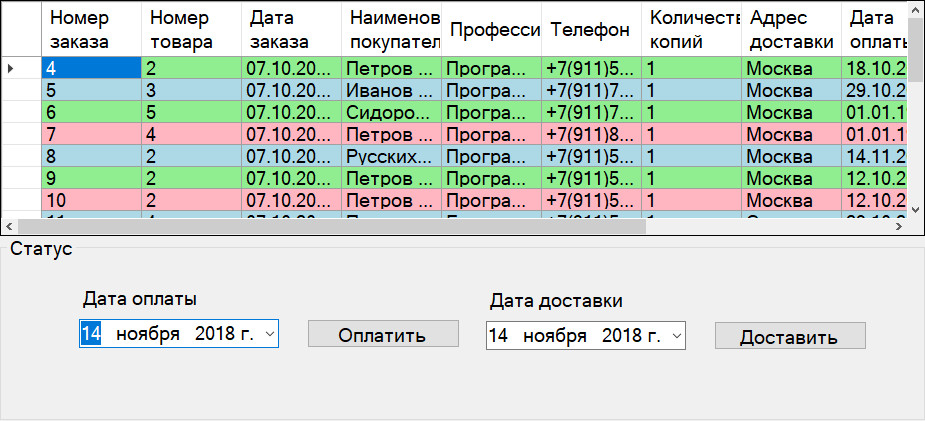


Рисунок 28–Внешний вид формы *Fstat*

Для ввода данных о заказах в таблицу я использовал DataTimePicker

Кнопка Оплатить добавляет дату оплаты изменяет статус на 1, а также раскрашивает выбранную строку в синий цвет.

Кнопка Доставить добавляет дату доставки изменяет статус на 2, а также раскрашивает выбранную строку в зелёный цвет.

Листинг:

*public string select\_zak = "SELECT Nzak as [Номер заказа],Ntov as [Номер товара],dzak as [Дата заказа],Namepok as [Наименование покупателя],Prof as [Профессия],Tel as [Телефон],Kookop as [Количество копий],AdresDost as [Адрес доставки],Dopl as [Дата оплаты],Ddost as [Дата доставки],Stat as [Статус],sposopl as [Способ оплаты] FROM zakaz";*

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

*{*

*// Создадим новое подключение, в качестве параметра укажем строку подключения //ConnectionString.*

*SqlConnection conn1 = new SqlConnection();*

*conn1.ConnectionString = ConnectionString;*

*//Теперь можно устанавливать соединение, вызывая метод Open объекта*

*conn1.Open();*

*//создаем новый экземпляр SQLCommand*

*SqlCommand cmd = conn1.CreateCommand();*

*//определяем тип SQLCommand=StoredProcedure*

*cmd.CommandType = CommandType.StoredProcedure;*

*//определяем имя вызываемой процедуры*

*cmd.CommandText = "[opkach]";*

*//создаем параметр*

*cmd.Parameters.Add("@Date1", SqlDbType.Date, 150);*

*//задаем значение параметра*

*cmd.Parameters["@Date1"].Value = dateTimePicker1.Value;*

*cmd.Parameters.Add("@zak", SqlDbType.Int, 4);*

*//определяем ID книговыдачи*

*string zak = dgv1[0, dgv1.CurrentRow.Index].Value.ToString();*

*//задаем значение параметра*

*cmd.Parameters["@zak"].Value = zak;*

*cmd.ExecuteNonQuery();*

*MessageBox.Show("Изменения внесены.", "Добавление записей");*

*conn(ConnectionString, select\_zak, dgv1);*

*}*

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

*{*

*// Создадим новое подключение, в качестве параметра укажем строку подключения //ConnectionString.*

*SqlConnection conn1 = new SqlConnection();*

*conn1.ConnectionString = ConnectionString;*

*//Теперь можно устанавливать соединение, вызывая метод Open объекта*

*conn1.Open();*

*//создаем новый экземпляр SQLCommand*

*SqlCommand cmd = conn1.CreateCommand();*

*//определяем тип SQLCommand=StoredProcedure*

*cmd.CommandType = CommandType.StoredProcedure;*

*//определяем имя вызываемой процедуры*

*cmd.CommandText = "[dostav]";*

*//создаем параметр*

*cmd.Parameters.Add("@Date", SqlDbType.Date, 150);*

*//задаем значение параметра*

*cmd.Parameters["@Date"].Value = dateTimePicker1.Value;*

*cmd.Parameters.Add("@zak", SqlDbType.Int, 4);*

*//определяем ID книговыдачи*

*string zak = dgv1[0, dgv1.CurrentRow.Index].Value.ToString();*

*//задаем значение параметра*

*cmd.Parameters["@zak"].Value = zak;*

*cmd.ExecuteNonQuery();*

*MessageBox.Show("Изменения внесены.", "Добавление записей");*

*conn(ConnectionString, select\_zak, dgv1);*

*SqlCommand cmd1 = conn1.CreateCommand();*

*cmd1.ExecuteNonQuery();*

*}*

*private void dgv1\_RowPrePaint(object sender, DataGridViewRowPrePaintEventArgs e)*

*{*

*DataGridView grid = (sender as DataGridView);*

*if ((e.RowIndex > -1) && (e.RowIndex < grid.RowCount - 1))*

*{*

*// раскрасить в зависимости от значения определенного поля*

*if (dgv1.Rows[e.RowIndex].Cells[10].Value.ToString() == "0")*

*grid.Rows[e.RowIndex].DefaultCellStyle.BackColor = Color.LightPink;*

*if (dgv1.Rows[e.RowIndex].Cells[10].Value.ToString() == "1")*

*grid.Rows[e.RowIndex].DefaultCellStyle.BackColor = Color.LightBlue;*

*if (dgv1.Rows[e.RowIndex].Cells[10].Value.ToString() == "2")*

*grid.Rows[e.RowIndex].DefaultCellStyle.BackColor = Color.LightGreen;*

*};*

### 3.5.5 Форма списка отправки ПО (*otptavka, form5.cs*).



Рисунок 29–Внешний вид формы *otpravka.*

Для ввода данных в таблицу нет никаких данных, но сразу скажу, в данную таблицу попадают только те данные, в которых есть дата оплаты, но нет даты доставки.

Листинг:

*const string otpravkapo = "select zakaz.Nzak as [номер счета-фактуры],zakaz.Dopl as [Дата],TOVAR.NamePO as [ Наименование ПО],TIPLISE.Tiplis as [Тип лицензии], zakaz.Namepok as [Наименование покупателя],zakaz.AdresDost as АдресДоставки ,zakaz.sposopl as [Способ оплаты],(zakaz.kookop\* tovar.Stomost) as сумма from zakaz,TIPLISE,TIPPO,TOVAR,DOGOVOR where zakaz.Ntov=TOVAR.Ntov AND TOVAR.Ntiplis=TIPLISE.Ntiplis AND TOVAR.NtipPO=TIPPO.NtipPO AND DOGOVOR.Ndog=TOVAR.Ndog AND zakaz.Dopl is not null and zakaz.Ddost is null AND Stat = 1";*

### 3.5.6 Форма отчётов (*otch, form6.cs*).

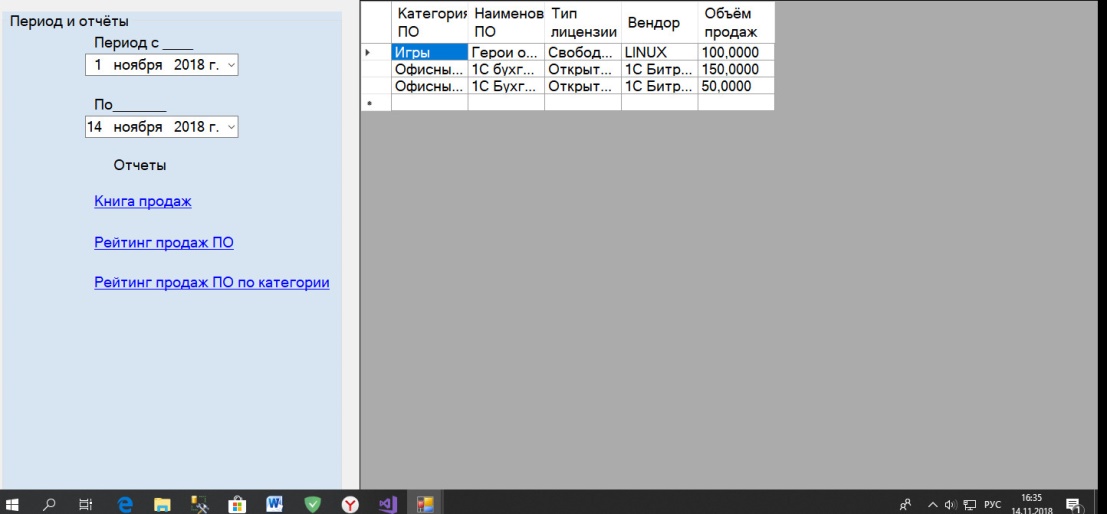


Рисунок 30 – Внешний вид формы *otch*

Для вывода данных из таблицы я использовал DateTimePicker,

Кнопка Печать выдаёт информацию в Excel.Код этой кнопки выглядит аналогично с Fspis

Листинг:

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

SqlConnection conn1 = new SqlConnection();

conn1.ConnectionString = ConnectionString;

//Теперь можно устанавливать соединение, вызывая метод Open объекта

conn1.Open();

//создаем новый экземпляр SQLCommand

SqlCommand cmd = conn1.CreateCommand();

//определяем тип SQLCommand=StoredProcedure

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "[Knigaprod]";

cmd.Parameters.Add("@Data1", SqlDbType.Date, 4);

cmd.Parameters["@Data1"].Value = dateTimePicker1.Value;

cmd.Parameters.Add("@Data2", SqlDbType.Date, 4);

cmd.Parameters["@Data2"].Value = dateTimePicker2.Value;

//создание экземпляра адаптера

SqlDataAdapter ReportAdapter = new SqlDataAdapter();

ReportAdapter.SelectCommand = cmd;

//создание объекта DataSet (набор данных)

DataSet dsReport = new DataSet();

//Заполнение таблицы Instance абора данных DataSet данными из БД

ReportAdapter.Fill(dsReport, "Knigaprod");

//Связываем источник данных компонента dataGridView на форме, с таблицей Report объекта dsInstance:

dgv1.DataSource = dsReport.Tables["Knigaprod"].DefaultView;

}

private void linkLabel2\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

SqlConnection conn1 = new SqlConnection();

conn1.ConnectionString = ConnectionString;

//Теперь можно устанавливать соединение, вызывая метод Open объекта

conn1.Open();

//создаем новый экземпляр SQLCommand

SqlCommand cmd = conn1.CreateCommand();

//определяем тип SQLCommand=StoredProcedure

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "[Raitprod]";

cmd.Parameters.Add("@Data1", SqlDbType.Date, 4);

cmd.Parameters["@Data1"].Value = dateTimePicker1.Value;

cmd.Parameters.Add("@Data2", SqlDbType.Date, 4);

cmd.Parameters["@Data2"].Value = dateTimePicker2.Value;

//создание экземпляра адаптера

SqlDataAdapter ReportAdapter = new SqlDataAdapter();

ReportAdapter.SelectCommand = cmd;

//создание объекта DataSet (набор данных)

DataSet dsReport = new DataSet();

//Заполнение таблицы Instance абора данных DataSet данными из БД

ReportAdapter.Fill(dsReport, "Raitprod");

//Связываем источник данных компонента dataGridView на форме, с таблицей Report объекта dsInstance:

dgv1.DataSource = dsReport.Tables["Raitprod"].DefaultView; }

private void linkLabel3\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

SqlConnection conn1 = new SqlConnection();

conn1.ConnectionString = ConnectionString;

//Теперь можно устанавливать соединение, вызывая метод Open объекта

conn1.Open();

//создаем новый экземпляр SQLCommand

SqlCommand cmd = conn1.CreateCommand();

//определяем тип SQLCommand=StoredProcedure

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "[RAITKATEPROD]";

cmd.Parameters.Add("@Data1", SqlDbType.Date, 4);

cmd.Parameters["@Data1"].Value = dateTimePicker1.Value;

cmd.Parameters.Add("@Data2", SqlDbType.Date, 4);

cmd.Parameters["@Data2"].Value = dateTimePicker2.Value;

//создание экземпляра адаптера

SqlDataAdapter ReportAdapter = new SqlDataAdapter();

ReportAdapter.SelectCommand = cmd;

//создание объекта DataSet (набор данных)

DataSet dsReport = new DataSet();

//Заполнение таблицы Instance абора данных DataSet данными из БД

ReportAdapter.Fill(dsReport, "RAITKATEPROD");

//Связываем источник данных компонента dataGridView на форме, с таблицей Report объекта dsInstance:

dgv1.DataSource = dsReport.Tables["RAITKATEPROD"].DefaultView;}

# 4 Тестирование

## 4.1 Модульное тестирование

Тестированию подвергаются: интерфейс каждого модуля для проверки правильности ввода-вывода; - внутренние структуры данных для проверки целостности сохраняемых данных; независимые пути – однократное выполнение всех операторов модуля; пути обработки ошибок; граничные условия.

4.1.1 Модуль «Ввод товара» post

Ввести данные договора с поставщиком, заполняя все поля ввода.

При нажатии на кнопку «Подписать» в БД должен добавиться новый договор и отобразиться в таблице на форме, а также активизировать новую страницу с вводом товара.

При нажатии на кнопку «Добавить товар» в БД должен добавиться новый товар и отобразиться в таблице на форме,.

Оставить не заполненным любое поле , щелкнуть по кнопке Добавить товар или Подписать . Должно быть выдано сообщение об ошибке. Если в некоторых полях, такие как например банковские реквизиты попытаться вместо цифры поставить символ, то он не пропишется и выдаст сообщение об ошибке.

При нажатии на кнопку «Изменить» включается живой запрос, который позволяет изменять данные в таблице

При нажатии на кнопку «Отменить» отключается живой запрос,

При нажатии на кнопку «Очистить поля» все введённые поля очищаются,

4.1.2 Модуль «Заказ» zak

Ввести данные заявки и договора, заполняя все поля ввода.

При нажатии на кнопку «Оформить заказ» в БД должен добавиться новый заказ и выдать сообщение о подтверждении оплаты заказа и должна появиться форма с выполнением заказа и отобразиться в таблице на другой форме.

Оставить не заполненным любое поле, щелкнуть по кнопке Сохранить договор. Должно быть выдано сообщение об ошибке. Если в ИНН, Банковские реквизиты попытаться вместо цифры поставить символ, то он не пропишется и выдаст сообщение об ошибке.

Оставить не заполненным любое поле, щелкнуть по кнопке Сохранить заявку. Должно быть выдано сообщение об ошибке. Если в вес, расстояние попытаться вместо цифры поставить символ, то он не пропишется и выдаст сообщение об ошибке.

4.1.3 Модуль zak2

При нажатии на кнопку «Оплатить» необходимая строка таблицы окрашивается в синий цвет, статус равен 1 и добавляется дата оплаты.

При нажатии на кнопку «Доставить» необходимая строка таблицы окрашивается в зелёный цвет, статус равен 2 и добавляется дата доставки.

При нажатии на кнопку «Изменить» включается живой запрос, который позволяет изменять данные в таблице

При нажатии на кнопку «Отменить» отключается живой запрос,

4.1.4 Модуль spis

При щелчке по кнопке «Печать» формируется отчет в формате Excel, форма которого соответствует требованиям пользователя.

4.1.5 Модуль book

Выбрать период, за который нужно сформировать отчет. При изменении даты начала или конца периода в таблице на форме должен отображаться книга продаж и необходимые сведения, которые входят в заданный период.

При щелчке по кнопке «Печать» формируется отчет в формате Excel, форма которого соответствует требованиям пользователя.

4.1.6 Модуль rait

Выбрать период, за который нужно сформировать отчет. При изменении даты начала или конца периода в таблице на форме должен отображаться рейтинг продаж и необходимые сведения, которые входят в заданный период.

При щелчке по кнопке «Печать» формируется отчет в формате Excel, форма которого соответствует требованиям пользователя.

4.1.7 Модуль krait

Выбрать период, за который нужно сформировать отчет. При изменении даты начала или конца периода в таблице на форме должен отображаться рейтинг продаж и необходимые сведения, которые входят в заданный период.

При щелчке по кнопке «Печать» формируется отчет в формате Excel, форма которого соответствует требованиям пользователя.

## 4.2 Тестирование интеграции

Для обнаружения ошибок интерфейса нужно проверить сначала все технологические цепочки обработки данных, а затем работу каждого модуля после каждого для проверки неблагоприятного влияния.

4.2.1 Технологические цепочки

*Модуль avto Модуль postМодуль zak Модуль main2*

*Модуль post*

*Модуль zak Модуль zak2Модуль spis*

*Модуль main2Модуль book Модуль rait Модуль krait*

4.2.2 Проверка неблагоприятного влияния модулей

Таблица 9 - Результаты тестирования интеграции

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Модули* | *Post* | *Zak* | *Spis* | *Book* | *Rait* | *Krait* |
| *Post* | *+* | *+* | *+* | *+* | *+* | *+* |
| *Zak* | *+* | *+* | *+* | *+* | *+* | *+* |
| *Spis* | *+* | *+* | *+* | *+* | *+* | *+* |
| *Book* | *+* | *+* | *+* | *+* | *+* | *+* |
| *Rait* | *+* | *+* | *+* | *+* | *+* | *+* |
| *Krait* | *+* | *+* | *+* | *+* | *+* | *+* |

## 4.3 Тестирование правильности

Введем данные контрольного примера.

После ввода распечатаем все отчеты за период *с 07.10.2018 по 30.10.2018*.

Проверим совпадение данных в отчете рейтинге продаж по категории.

Проверим совпадение итоговых сумм в отчете рейтинге продаж по категории с суммами, рассчитанными вручную.

Проверка показала, что данные отчетов формируются верно.

# 5 Классификации информационной системы персональных данных

«Разработка базы данных и клиентского приложения для автоматизации учета продажи ПО» Embacadero RAD STUDIO 2010

Рассмотрев исходные данные информационной системы персональных данных (ИСПДн) «Автоматизация учета продажи ПО», в соответствии с требованиями совместного приказа ФСТЭК России, ФСБ России, Мининформсвязи России от 13 февраля 2008 г. № 55/86/20 «Об утверждении порядка проведения классификации информационных систем персональных данных» определили:

1. Категория персональных данных, обрабатываемых в информационной системе (Хпд): 3
2. Объём обрабатываемых персональных данных (Хпдн.): К 3
3. Информационная система персональных данных, по заданным оператором характеристикам безопасности относится к : специальным
4. Структура информационной системы: локальная
5. Наличие подключений к сетям связи общего пользования и (или) сетям международного информационного обмена: нет
6. Режим обработки персональных данных в информационной системе: многопользовательский
7. Разграничение прав доступа пользователей: с разграничением
8. Местонахождения информационных систем персональных данных: в пределах РФ

По результатам анализа исходных данных информационной системы персональных данных «Автоматизация учета продажи ПО» присвоен класс К 3.

«Разработка базы данных и клиентского приложения для автоматизации учета продажи ПО» Visual studio 2017 c#

Рассмотрев исходные данные информационной системы персональных данных (ИСПДн) «Автоматизация учета продажи ПО», в соответствии с требованиями совместного приказа ФСТЭК России, ФСБ России, Мининформсвязи России от 13 февраля 2008 г. № 55/86/20 «Об утверждении порядка проведения классификации информационных систем персональных данных» определили:

1. Категория персональных данных, обрабатываемых в информационной системе (Хпд): 3
2. Объём обрабатываемых персональных данных (Хпдн.): К 3
3. Информационная система персональных данных, по заданным оператором характеристикам безопасности относится к : специальным
4. Структура информационной системы: локальная
5. Наличие подключений к сетям связи общего пользования и (или) сетям международного информационного обмена: нет
6. Режим обработки персональных данных в информационной системе: многопользовательский
7. Разграничение прав доступа пользователей: с разграничением
8. Местонахождения информационных систем персональных данных: в пределах РФ

По результатам анализа исходных данных информационной системы персональных данных «Автоматизация учета продажи ПО» присвоен класс К 3.

«Разработка базы данных и клиентского приложения для автоматизации учета продажи ПО» PHP

Рассмотрев исходные данные информационной системы персональных данных (ИСПДн) «Автоматизация учета продажи ПО», в соответствии с требованиями совместного приказа ФСТЭК России, ФСБ России, Мининформсвязи России от 13 февраля 2008 г. № 55/86/20 «Об утверждении порядка проведения классификации информационных систем персональных данных» определили:

1. Категория персональных данных, обрабатываемых в информационной системе (Хпд): 3
2. Объём обрабатываемых персональных данных (Хпдн.): К 3
3. Информационная система персональных данных, по заданным оператором характеристикам безопасности относится к : специальным
4. Структура информационной системы: распределённая
5. Наличие подключений к сетям связи общего пользования и (или) сетям международного информационного обмена: нет
6. Режим обработки персональных данных в информационной системе: многопользовательский
7. Разграничение прав доступа пользователей: с разграничением
8. Местонахождения информационных систем персональных данных: в пределах РФ

По результатам анализа исходных данных информационной системы персональных данных «Автоматизация учета продажи ПО» присвоен класс К 3.

# Заключение

В ходе выполнения курсового проекта была создана база данных и клиентское приложение для реализации требований автоматизации системы продаж ПО.

Все требования технического задания выполнены, реализованы функции:

* Оформление договора
* Ввод товара
* Оформление заказа
* Список отправки ПО.
* Книга продаж
* Рейтинг продаж
* Рейтинг продаж ПО

**Дополнительно:**

* Изменение статуса заказа
* Реализовал автоматизацию учёта продажи ПО на PHP
* Реализовал автоматизацию учёта продажи ПО на С#

Выполнено тестирование программного продукта, оформлена необходимая документация; продукт успешно сдан

В результате получены умения и опыт :

- сформировал умения:

* создавать объекты баз данных в современных системах управления базами данных и управлять доступом к этим объектам;
* работать с современными case-средствами проектирования баз данных;
* формировать и настраивать схему базы данных;
* разрабатывать прикладные программы с использованием языка SQL;
* использовать различные технологии доступа к данным;
* обеспечивать доступ к базам данных в пределах локальных сетей;
* создавать концептуальную, логическую и физическую модель базы данных;
* разрабатывать приложения баз данных

получил практический опыт:

* работы с объектами базы данных в файл-серверных и настольных СУБД;
* создания информационных систем на основе баз данных;