

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПРИЕМА, ОБРАБОТКИ,
ДОСТАВКИ И ВРУЧЕНИЯ МЕЖДУНАРОДНЫХ ПОСЫЛОК ДЛЯ
ОРАГАНИЗАЦИИ «ПОЧТА РОССИИ»**

Выпускная квалификационная работа
Обучающейся по направлению подготовки 02.03.02 Фундаментальная
информатика и информационные технологии
заочной формы обучения, группы 07001360
Ермакова Павла Геннадьевича

Научный руководитель
к.т.н., доцент
Чашин Ю.Г.

БЕЛГОРОД 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ НА РАЗРАБОТКУ.....	4
1.1 Особенности организации доставки	4
1.2 Обзор аналогов существующих систем.....	6
1.3 Постановка задачи	9
ГЛАВА 2. ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ	11
2.1 Информационное обеспечение.....	11
2.2 Программное обеспечение.....	13
ГЛАВА 3. РАЗРАБОТКА СТРУКТУРЫ ПРИЛОЖЕНИЯ.....	19
И БАЗЫ ДАННЫХ	19
3.1 Общие положения (дерево функций и сценарий диалога).....	19
3.2 Структурная схема пакета (дерево вызова программных модулей) .	20
3.3 Описание программных модулей.....	22
ГЛАВА 4. РЕАЛИЗАЦИЯ СИСТЕМЫ НА ПРЕДПРИЯТИИ	25
ЗАКЛЮЧЕНИЕ	48
СПИСОК ЛИТЕРАТУРЫ	50
ПРИЛОЖЕНИЕ	52

ВВЕДЕНИЕ

Целью дипломного проекта является разработка автоматизированной системы приема, обработки, доставки и вручения международных посылок для организации «Почта России».

Для достижения цели были сформулированы следующие задачи:

1. Выполнить анализ характеристик объекта автоматизации. Изучить организационную и функциональную структуру предприятия.
2. Обзор существующих автоматизированных систем.
3. Разработка базы данных приложения.
4. Систематизировать документооборот.
5. Изучить процесс функционирования предприятия. Разработать функциональную модель существующей системы.
6. Реализовать новый проект делового процесса и разработать модель деятельности автоматизированной системы.
7. Разработать алгоритмическое и программное обеспечение системы.
8. Разработать базу данных, обеспечивающая реализацию функций работы организации.

Структура дипломной работы. Дипломная работа состоит из введения, четырех глав, заключения, списка литературы и приложения.

В первой части рассмотрены аналоги существующих систем и постановка задачи на разработку.

Во второй части описан выбор инструментальных средств и обоснование выбора программного обеспечения.

В третьей части разработана структура приложения и базы данных.

В четвертой части описана реализация приложения и апробация на предприятии

В заключении приводится вывод о проделанной работе.

ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ НА РАЗРАБОТКУ

1.1 Особенности организации доставки

Производственная деятельность системы почтовой связи заключается в предоставлении населению, предприятиям, организациям и учреждениям основных и дополнительных услуг. К основным услугам относятся все производственные операции, необходимые для организации обслуживания клиентов для перемещения в пространстве почтовых отправлений. Дополнительными услугами для населения являются: упаковка почтовых отправлений; заполнение бланков сопроводительных адресов к посылкам, денежных переводов, описей вложения; написание адресатов и письменных сообщений; продажа или предоставление в аренду посылочных ящиков, доставка на дом почтовых отправлений, хранение посылок и бандеролей и др.

Для выполнения основных и дополнительных услуг в системе почтовой связи функционирует сеть структурных подразделений почтовой связи. Структурные подразделения почтовой связи организуют производственный процесс пересылки почтовых отправлений от отправителя к адресату, осуществляют техническую эксплуатацию и совершенствование средств автоматизации и механизации, обеспечивают все виды финансовой деятельности и экономическое взаимодействие со смежными предприятиями, занимаются подготовкой кадров и повышением их квалификации.

На крупных железнодорожных станциях, аэропортах, водных пристанях создаются отделения перевозки почты, осуществляющие обработку и обмен почты с магистральными транспортными средствами. В важнейших магистральных узлах в условиях концентрации больших объемов почтовых отправлений организуются зональные узлы, на которые возлагаются дополнительные функции по обработке большей части почтовых отправлений,

принимаемых на территории почтовой зоны. Такой зональный узел может также производить прием и выдачу посылок и бандеролей, укомплектованных в прямые партии и выполнять часть операций по обслуживанию клиентуры городского почтамта.

Наиболее близкими и доступными клиентуре являются отделения связи, предоставляющие практически все виды услуг почтовой связи. Отделение связи является самостоятельным производственным подразделением, производящим в ряде случаев обработку и обмен почтовых отправлений. Как правило, на территории региона размещается несколько отделений связи. Для организационно-хозяйственного управления отделениями связи создаются автоматизированные сортировочные центры (АСЦ) и отделения почтовой связи (ОПС). В областях с небольшой плотностью населения создаются межрайонные узлы связи (МРУС), осуществляющие административно-хозяйственное и функциональное руководство отделениями связи нескольких смежных районов. Образование МРУС в крупных городах способствует сглаживанию неравномерности нагрузки по часам суток и дням недели в городском почтамте или ЦОП (Центр обработки почты), что приводит к существенному повышению эффективности функционирования системы связи в целом.

В сельских местностях с небольшой плотностью населения обмен почты организуется посредством передвижных отделений связи. Для обслуживания сезонных работ и сезонных курортов временно создаются сезонные отделения связи.

В областных и республиканских городах для обслуживания крупных предприятий, организаций и учреждений, имеющих значительные объемы входящей корреспонденции, создаются пункты городской служебной почты (ГСП), осуществляющие доставку письменной корреспонденции и периодической печати, в большинстве случаев моторизованным способом.

Сумма оценки писем, посылок не ограничивается и назначается отправителем в рублях, но не ниже фактической стоимости вложения. Сумма

оценки почтовых отправлений указывается на лицевой стороне почтового отправления и на сопроводительном адресе к посылке цифрами и в скобках прописью в полных рублях. На страховых почтовых отправлениях, пересылаемых с описью вложения, сумма оценки, указанная на оболочке, должна соответствовать общей ценности, указанной в описи.

Почтовые отправления выдаются в учреждениях связи после предъявления документа, удостоверяющего личность адресата. При получении страховой почты адресат заносит в соответствующие почтовые документы требуемые реквизиты из удостоверения личности.

1.2 Обзор аналогов существующих систем

Программное обеспечение «WinPost» обеспечивает функционирование автоматизированного рабочего места оператора почтовой связи и решение всего комплекса задач операционного дня.

В информационной системе ПКТ «WinPost» реализованы следующие функции:

- приём, вручение, досыл и возврат как внутренних, так и международных почтовых отправлений;
- приём различных платежей и осуществление выплат клиентам;
- приём как внутренних, так и международных переводов, выдача переводов клиентам;
- приём телеграмм;
- оформление подписки на периодические издания;
- проведение электронных платежей;
- проведение возвратных операций;
- формирование «электронной очереди»;
- продажа товаров.

Оплата услуг в ИС ПКТ «WinPost» может быть осуществлена:

- по наличному расчету;
- по безналичному расчету;
- в счет авансового платежа;
- с расчетной пластиковой карты.

Во время выполнения различных операций и по окончании смены ИС ПКТ «WinPost» предусматривает формирование и вывод на устройство печати различных форм фискальных и не фискальных документов.

Пакет программ «ПАРТИОННАЯ ПОЧТА» является одним из элементов системы ОАСУ РПО. Пакет предназначен для приема, обработки и создания информации о регистрируемых почтовых отправлениях, обработки партионной почты, проверки корректности информации и тождественности информации и отправлений и другие операции.

Состав пакета программ:

1. РПО (регистрируемые почтовые отправления) – модуль позволяет производить операции с РПО: обмен информацией с базой ПРО РФ (экспорт/импорт); сбор информации; поиск почтовых отправлений и другие операции.

2. Ручной ввод информации об РПО – позволяет производить ввод данных об РПО за объекты почтовой связи, не имеющие возможность импортировать данные в электронном виде.

3. Предпочтовая подготовка партионных отправлений – позволяет формировать всю сопроводительную документацию к партионной почте. Функции программы: Формирование и/или редактирование информации о почтовых отправлениях в соответствии с требованиями почты (проверка на корректность почтового индекса и возможность доставки, проверка на соответствие виду и категории отправления, допустимому весу, наложенному платежу и т.д.). Формирование уникального почтового идентификатора для каждого регистрируемого почтового отправления в соответствии с требованиями почты. Автоматическая тарификация отправлений, включая

комбинированную доставку. Формирование списка формы 103 в электронном виде в соответствии с требованиями почты. Печать сопроводительной документации к партионной почте: ф.103, ф.16, ф.117, ф.113, ф.116, ф.10, ярлык ф.7, печать на конвертах и т.п. Ведение базы данных по отправлениям с возможностью установления отметок о прохождении почты и иных технологических отметок. Импорт замечаний и ошибок обработки информации об отправлениях на объекте почтовой связи, а также отображение результатов обработки в наглядной форме.

4. Контроль и обработка данных партионных отправок – используется на объектах почтовой связи и предназначен для контроля и обработки данных поступающих в виде форм 103 в электронном виде. Функции программы: Импорт и обработка данных списков ф.103, автоматический контроль тарификации отправок списка, включая автоматический перерасчет комбинированной и авиа доставки. Проверка соответствия адреса и индекса места назначения отправок, контроль уникальности и правильности составления почтового идентификатора. Хранение и использование информации о партионной почте для задач контроля и учета, розыска почтовых отправок, а также формирования отчетности, осуществления поиска отправок в базе данных по заданным параметрам, а также ведение реестра клиентов. Выдача результатов обработки информации пользователю услуг, сдающему партионную почту, как на бумаге, так и в электронном виде.

5. Так же в состав пакета программ входит ряд утилит:
- Расположение базы данных.
 - Резервные копии.
 - Управление доступом.
 - Перенос данных.
 - Калькулятор тарифов.

ИС «Почтамт – Сортировочный узел» - данный комплекс задач разработан в соответствии с приказом Генерального директора ФГУП «Почта России» №409

от 04.10.2004г. «О введении в действие Временного порядка обработки исходящих и транзитных почтовых отправлений, и почтовых емкостей». ИС «Почтамт – Сортировочный узел» должен заменить программно-технологические комплексы АРМ СУ и АРМ «Заказная почта», которые ранее эксплуатировались на почтовой сети.

Функции ИС «Почтамт – Сортировочный узел». В базовой версии 1.0.1.23 ИС «Почтамт – Сортировочный узел» реализованы следующие функции обработки почтовых отправлений и емкостей:

- Регистрация принятых почтовых емкостей.
- Формирование накладных ф.16 и ф.23 при заделке почтовых емкостей.
- Формирование накладных на отправку в адрес ОПС сдачи, а также на участок данного сортировочного узла.
- Формирование общей накладной ф.23-а и маршрутной накладной ф.24.
- Формирование накладной на передачу почты по смене.
- Нанесение на все формируемые накладные штрихкодовых идентификаторов в соответствии с РТМ 0008.01-03.
- Формирование отчетных документов: ведомостей учета поступления и отправки почтовых отправлений и емкостей, учета обработанных РПО и сформированных емкостей, справки о сменном движении почты и т.п.

1.3 Постановка задачи

Для повышения эффективности работы почтовой связи была разработана автоматизированная система.

Функциональные характеристики:

- Формировать заявки на перевозку груза;

- Автоматически включать услуги по перевозке груза, а также дополнительные услуги (грузчики, экспедитор и т.п.);
- Производить расчет стоимости заявки в режиме реального времени;
- Отражать согласование стоимости заявки с заказчиком;
- Отслеживать исполнение заявки;
- Производить перерасчет стоимости заявки;
- Подбирать автотранспорт исходя из параметров груза;
- По требуемому транспорту и описанию в заявке возможен подбор транспортного средства и водителя. Подбор транспорта осуществляется по различным критериям, основными являются: «вид транспорта», «длина», «ширина», «высота», «направление»;
- Формировать сопроводительные документы для перевозки груза;
- Формирование путевых листов, ТТН (товарно-транспортных накладных);
- Контролировать документооборот по заявке;
- Возможность контролировать получение (по факсу, по почте и т.д.), выписку документов по заявке и т.д.

Система должна быть совместима с техническими параметрами компьютера.

В базе данных программы должны вводиться и храниться следующие данные:

- ФИО клиента или наименование организации;
- Фактический адрес местонахождения;
- Контактные данные;
- Доверенность на получение посылок (для организации);
- Информация о доставляемых и врученных отправлениях.

ГЛАВА 2. ВЫБОР ИСТРУМЕНТАЛЬНЫХ СРЕДСТВ

2.1 Информационное обеспечение

Информационное обеспечение (ИО) предназначено для отражения информации, характеризующей состояние управляемого объекта и являющейся основой для принятия управленческих решений [8].

Однако, несмотря на существование унифицированной подсистемы документации, при обследовании большинства организаций постоянно выявляется целый комплекс типичных недостатков:

- чрезвычайно большой объем документов для ручной обработки;
- одни и те же показатели часто дублируются в разных документах;
- работа с большим количеством документов отвлекает специалистов от решения непосредственных задач;
- имеются показатели, которые создаются, но не используются.

Устранение указанных недостатков является одной из задач, стоящих при создании информационного обеспечения.

Разработка информационного обеспечения включает подготовку документов, содержащих информацию, необходимую для решения задачи, и анализ этой информации. Такой анализ позволит произвести формализацию данных, которая имеет целью их однозначное определение для хранения, поиска и обработки во внутри-машинной сфере.

1. Входные и выходные документы. За основу форм документов берутся бланки, используемые на предприятии. Их необходимо проанализировать, оценить полноту содержащейся в них информации, и использовать при проектировании системы.

2. Экранные формы - построения диалога с пользователем возможны следующие способы: диалог типа «Вопрос-ответ»; диалог типа «Меню», диалог на основе экранных форм, диалог на основе командного языка.

Для каждого из входных документов будут создаваться экранные формы ввода первичной информации. При этом необходимо обеспечить соответствие макета экранной формы форме входного документа. Не обязательно, чтобы внешний вид этих форм совпадал, но все вводимые элементы данных должны располагаться в том же относительном порядке и иметь такой же формат, что и в исходном документе.

Результатная информация должна отображаться на экранных формах в соответствии с формой выходного документа, т.е. на основе использования комбинированной формы с трехзонным расположением реквизитов и многострочной содержательной частью.

Все экранные формы должны быть удобными, понятными, эргономичными. Общий цвет информационной части должен быть спокойных тонов, не вызывающих усталости пользователя. Цвет полей, подлежащих вводу с клавиатуры, должен отличаться от цвета информационной части. Каждое поле должно быть снабжено подсказкой, которую следует выдавать на экран при неправильных действиях пользователя. Должна быть обеспечена возможность исправления ошибок во введенных данных.

3. Система классификации и кодирования. В системе необходимо учесть принятую на предприятии систему кодирования документов, а именно: документы одного типа нумеруются последовательно с начала года. Система кодирования артикулов также используется существующая на предприятии, т.к. она достаточна, привычна, и используется не только в автоматизируемом подразделении, но и в других отделах предприятия.

4. Информационная база. Центральным компонентом информационного обеспечения является информационная база (ИБА), представляющая собой организованную определенным способом совокупность

данных, хранимых в памяти вычислительной системы в виде файлов, с помощью которых удовлетворяются информационные потребности управленческих процессов и решаемых задач [1].

Существуют следующие способы организации информационной базы: совокупность локальных файлов, поддерживаемых функциональными пакетами прикладных программ, и интегрированная база данных, основанная на использовании универсальных программных средств загрузки, хранения, поиска и ведения данных, то есть системы управления базами данных (СУБД).

Локальные файлы обеспечивают более быстрое время обработки данных, однако при таком способе организации ИБ в информационной системе наблюдается высокая степень дублирования данных, несогласованность данных, отсутствие гибкости доступа к информации. Интегрированная база данных – это совокупность взаимосвязанных, хранящихся вместе данных при такой минимальной избыточности, которая допускает оптимальное их использование в различных приложениях.

Для решения поставленной задачи требуется интегрированная база данных, в которой взаимосвязаны данные.

Основными способами организации БД являются создание централизованных и распределенных БД [4]. В рассматриваемой задаче не предполагается распределенная структура ИС, т.к. задача небольшая по объему данных и по количеству пользователей.

Таким образом, для решения поставленной задачи необходима интегрированная централизованная база данных.

2.2 Программное обеспечение

В настоящее время лидерами на рынке сред разработки являются Microsoft Visual Studio, Delphi, C++ Builder, JBuilder, 1С:Предприятие. Любая из перечисленных сред позволит разработать современное клиент-серверное

приложение с современным пользовательским интерфейсом. Однако, в компании имеются наработки на 1С: Предприятие которые можно использовать при разработке ИС. Выбор среды разработки 1С: Предприятие позволит гораздо быстрее по сравнению с остальными из перечисленных сред разработки получить готовую систему, с интерфейсом, аналогичным ИС отделов формирования задания на выполнения работ участка сборки электрооборудования.

В работе был выбран язык программирования 1С, который был специально создан для ведения бухгалтерского учета и используется в программном продукте «1С предприятие». Данный язык является предметно-ориентированным языком высокого уровня. Встроенный язык имеет много общего с другими языками, такими как Java Script, Basic, что упрощает его освоение начинающим разработчикам. Приведу наиболее значимые особенности встроенного языка: предварительная компиляция, кэширование скомпилированных модулей в памяти, мягкая типизация также. Также язык 1С обладает очень интересной и для некоторой удобной особенностью: доступен двуязычный синтаксис команд: русский и английский. Русский язык программирования сильно удивляет новичков в сфере 1С программирования. При этом стоит отметить, что система «1С предприятие» является открытой для доработки и самостоятельной разработки, что позволяет достичь большой гибкости в организации системы.

Для разработки системы автоматизации мы разрабатываем модуль (конфигурацию) в системе 1С:Предприятие 8.3.

Система 1С:Предприятие 8.3 является очень гибкой и легко настраиваемой системой. Такая гибкость заложена в архитектуре системы, основные моменты которой описаны в данном разделе.

В основе системы лежит гибкая технологическая платформа, возможности которой позволяют создавать и модифицировать самые разнообразные бизнес приложения. Такая универсальность достигается благодаря тому, что

1С:Предприятие – это не просто программа, существующая в виде набора неизменяемых файлов, а совокупность программных инструментов с которыми работают разработчики и пользователи.

В обновлённой версии 1С:Предприятие 8.3 принципиально переработан механизм установки, запуска и обновления платформы. Предприняты существенные усилия для повышения удобства развертывания системы в крупных организациях.

Области применения:

- автоматизированная система подготовки консолидированной отчетности;
- автоматизация гостиничного бизнеса;
- автоматизация Бизнес процессов;
- решение задач планирования, бюджетирования и финансового анализа;
- расчет зарплаты и управление персоналом;
- автоматизация складской системы.

Клиентское приложение.

Возможность использования в 1С:Предприятие 8.3 четыре клиентских приложения: тонкий клиент, толстый клиент, веб-клиент, конфигуратор позволяет максимально адаптировать решение для конкретного пользователя в зависимости от объема задачи.

Толстый клиент позволяет использовать все возможности 1С:Предприятие 8.3 в плане исполнения прикладного кода, но он не поддерживает работу с информационными базами через интернет.

Тонкий клиент позволяет вести работу с базами через интернет, но обладает ограниченным набором функций встроенного языка. На тонком клиенте не доступны все прикладные типы данных, и он предназначен лишь для отображения и изменения данных в памяти.

Веб клиент не требует предварительной установки ко ПК и используется в среде интернет-браузера, что позволяет работать на любой платформе.

Конфигуратор позволяет выполнять разработку приложений и администрирование информационных баз.

Благодаря этим механизмам ориентированных на быструю разработку и удобство пользовательского интерфейса делают систему IC:Предприятие 8.3 очень удобной для разработки прикладных решений, что мы и будем делать в нашей работе.

Расширенный состав инструментов разработчиков для оптимизации прикладных решений. Групповая разработка (Team Development) приведена на рис. 2.1. Механизмы конфигуратора, обеспечивающие групповую разработку прикладного решения, позволяют группе разработчиков вносить изменения в конфигурацию одновременно, по мере выполнения каждым из них своего участка работы. Такой порядок внесения изменений обеспечивается возможностью определить права доступа каждого из разработчиков на модификацию объектов прикладного решения:



Рис. 2.1. Модель групповой разработки

Облачные технологии (Cloud Computing) - суть названия скрыть сложную инфраструктуру и технические детали этой технологии, а клиенту показать диаграмму в виде «облако» которая будет выполнять функцию интернета. В IC

Предприятию существует такая инфраструктура, которая может выполнять следующие задачи:

- Подключение к базе по протоколу HTTP (HTTPS) что дает возможность клиенту работать через интернет в любой точке мира;
- Благодаря веб-клиенту пользователь может работать с любого компьютера или мобильного так это не требует предварительной установки;
- Благодаря современным серверам и их защите 1С Предприятие может поддерживать большое количество пользователей;
- Технология разделения данных с помощью, которой прикладные решения могут работать в архитектуре multitenancy, это когда одно приложение, запущенное на сервере, обслуживает множество пользователей;
- 1С Предприятие позволяет развертывать приложения в модели SaaS. Когда программист разрабатывает и самостоятельно управляет прикладным решением, предоставляя клиенту доступ через интернет, такая бизнес-модель избавляет потребителя от всех затрат, связанных с установкой, обновлением и поддержкой оборудования и программного обеспечения; потребитель оплачивает лишь пользование услугой.

Трёхуровневая архитектура - данный вариант реализации программный комплекс состоит из трех уровней: Клиент, Сервер «1С предприятия», Сервер SQL (Structure Query Language - Структурированный Язык Запросов).

Главное назначение этого варианта построения – это повышение надежности и производительности. Самыми известными сторонними программами являются: MS SQL Server, PostgreSQL, причём последняя бесплатная и поставляется на дисках ИТС.

Упрощенное описание функций, выполняемых разными частями программного комплекса будет выглядеть так:

- Клиент – это то, что связано с выводом на экран монитора;
- Сервер 1С – запрос информации к серверу SQL на языке SQL, наложение ограничений прав пользователей на полученную информацию;

- Сервер SQL – хранение и изменение информации.

Система программ «1С:Предприятие» предназначена для автоматизации управления и учета на предприятиях различных отраслей, видов деятельности и типов финансирования, и включает в себя решения, разработанные самой фирмой «1С», ее партнерами и независимыми организациями.

Система «1С:Предприятие» состоит из передовой технологической платформы (ядра) и разработанных на ее основе прикладных решений («конфигураций»). Такая архитектура системы принесла ей высокую популярность, поскольку обеспечивает открытость прикладных решений, высокую функциональность и гибкость, масштабируемость от однопользовательских до клиент-серверных и территориально распределенных решений.

В работе с программами системы «1С:Предприятие 8.3» используется ряд объектов. К основным объектам ИС относятся:

Справочники – предназначены для хранения различных списков сведений, используемых в работе предприятия, например, перечень клиентов, грузов и т.п.

Документы – посредством документов вводится информация о хозяйственных операциях предприятия. Проведение документов – выполнение изменений в итогах ИС, например, увеличение количества груза на складах при выгрузке или уменьшение количества грузов при отгрузке. Документы используют сведения справочников и других документов.

Обработки – предназначены для автоматизации каких-либо действий, например, на основе анализа текущего состояния данных автоматически изменяются какие-либо итоги ИС.

Отчёты – используются для анализа хозяйственной деятельности предприятия. В отчёты выводятся данные, введённые посредством справочников, документов и обработок.

ГЛАВА 3. РАЗРАБОТКА СТРУКТУРЫ ПРИЛОЖЕНИЯ И БАЗЫ ДАННЫХ

3.1 Общие положения (дерево функций и сценарий диалога)

Для решаемой задачи не определена четкая последовательность действий при обработке данных. Все операции, связанные с вводом информации, должны выполняться в диалоговом режиме обработки в реальном масштабе времени. Для такого взаимодействия пользователей с программой удобнее всего использовать язык диалога. Программа реализует довольно большое количество функций, и для реализации такого диалога удобно использовать меню. Дерево функций системы показано на рис. 3.1. Сценарий диалога показан на рис. 3.2.



Рис. 3.1. Дерево функций системы

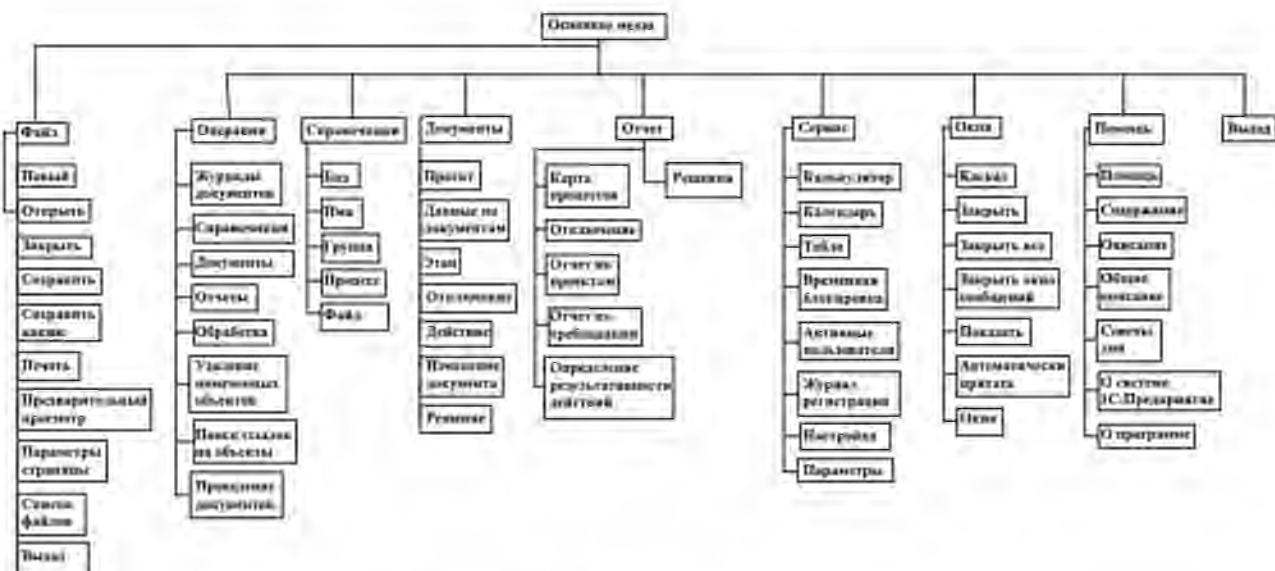


Рис. 3.2. Сценарий диалога

3.2 Структурная схема пакета (дерево вызова программных модулей)

Модель структурной схемы приведены на рис. 3.3. Модули структурной схемы пакета описаны в табл. 3.1.

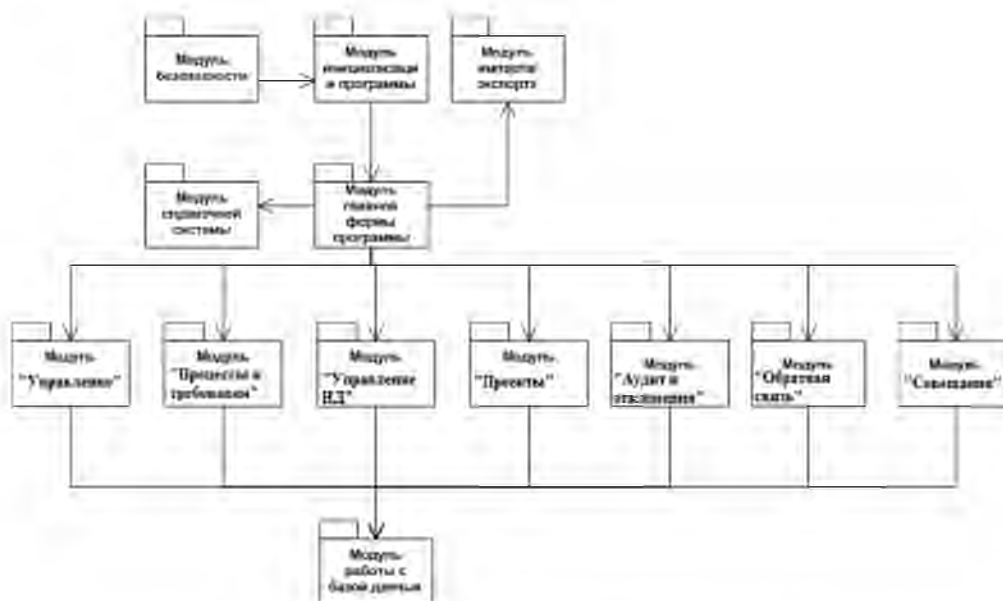


Рис. 3.3. Структурная схема пакета

Таблица 3.1

Описание функций структурной схемы пакета

п/п	Наименование модуля	Функции системы
1	Модуль безопасности	Содержит процедуры и функции, обеспечивающие ввод и проверку пароля, вход пользователей в систему, автоматический выход из системы через определенный промежуток времени
2	Модуль инициализации интерфейса программы	После успешного входа в систему, запускает программу, используя настройки прав доступа для учетной записи пользователя
3	Модуль импорта/экспорта	По назначенному расписанию или по запросу пользователя устанавливает соединение с серверами баз данных внешних систем и выполняет импорт или экспорт данных
4	Модуль взаимодействия с базой данных	Содержит процедуры и функции, используемые для установления соединения с базой данных ИС, а также управляющие операциями чтения, записи и редактирования данных
5	Модуль справочной системы	Содержит процедуры и функции, необходимые для отображения справочной информации
6	Модуль «Главная форма программы»	Содержит главное меню, из которого осуществляется вызов остальных экранных форм
7	Модуль «Управление»	Содержит процедуры и функции для ввода информации об организации, основных справочников
8	Модуль «Процессы и требования»	Содержит процедуры функции, позволяющие создавать и просматривать основные процессы и требования
9	Модуль ввода данных «Управление НД»	Содержит процедуры и функции, позволяющие вводить/редактировать данные, связанные с нормативными документами
10	Модуль ввода данных «Проекты»	Содержит процедуры и функции для ввода данных по проектам, этапам проектов

Таблица 3.1. Продолжение

п/п	Наименование модуля	Функции системы
11	Модуль ввода данных «Аудит и отклонения»	Содержит процедуры и функции для ввода данных по аудиту (проверки) продукции на участке электрооборудования
12	Модуль ввода данных «Обратная связь»	Содержит процедуры и функции для ввода данных по жалобам от других отделов по участку электрооборудования
13	Модуль ввода данных «Совещания»	Содержит процедуры и функции для ввода данных по решениям о качестве и проводимому контролю продукции на участке электрооборудования

3.3 Описание программных модулей

Работа с программой начинается с выбора пользователя и вывода информационного окна и активизации системы меню.

Работа программы осуществляется по диалоговому и событийному режиму, при этом под диалогом понимается предоставление пользователю нескольких альтернатив и обработка его выбора. В диалоговую систему входят главное меню с соответствующими всплывающими подменю, а также диалоговые окна. Под событиями понимаются процессы, активизируемые пользователем (например – нажатие функциональных клавиш), а также программные события – получение определенным полем фокуса редактирование или потеря фокуса ввода. На основании данных событий активизируются процедуры контроля допустимости данных.

Программа состоит из следующих основных модулей.

Глобальный модуль - конфигурация среды окружения, формирование основного экрана программы, создание системы главного меню и соответствующих подменю, активизация меню.

Процедуры формирования отчетов – обеспечение выдачи установленных форм документов на основании критериев, определяемых пользователем и информационной базы.

Модуль справочников и модуль документов - обеспечение ввода информации с первичных документов в базы данных, контроль за допустимостью значений, обеспечение ввода данных путем выбора из списка.

Все модули в программе связаны между собой по данным, которые анализируются на входе и вырабатываются на выходе. Данные в модули поступают через диалог с пользователем, параметры и документы информационной базы.

Для ведения информационной базы могут быть выполнены операции просмотра и печати документов, их редактирование, ведение нормативно-справочных документов, а также создание архивов и восстановление документов БД. Операции осуществляются путем выбора соответствующих пунктов в главном и подчиненных меню.

Данные через диалог могут быть получены прямым и косвенным способом. Прямой способ реализуется путем их ввода по шаблону или по запросу конкретных значений. Косвенный способ – путем меню.

Параметры (входные документы) – входные данные, полученные в виде конкретных значений, переданных в оперативной памяти смежным модулям (функциям).

Для каждого выделенного модуля разработана функционально-технологическая схема и детальный алгоритм ее кодирования. Блок-схема задачи работы со справочниками приведена на рис. 3.4.

Алгоритм функционирования разработанного приложения-модуля можно кратко описать следующей последовательностью действий. При загрузке модуля ожидается выбор одной из альтернатив главного меню. В зависимости от выбранной альтернативы главного меню происходит активация соответствующей процедуры, и синтез необходимых дочерних форм. После этого вычислительный процесс ожидает задействования элементов управления,

которые могут приводить к запуску различных программных процедур, входящих в состав проекта.

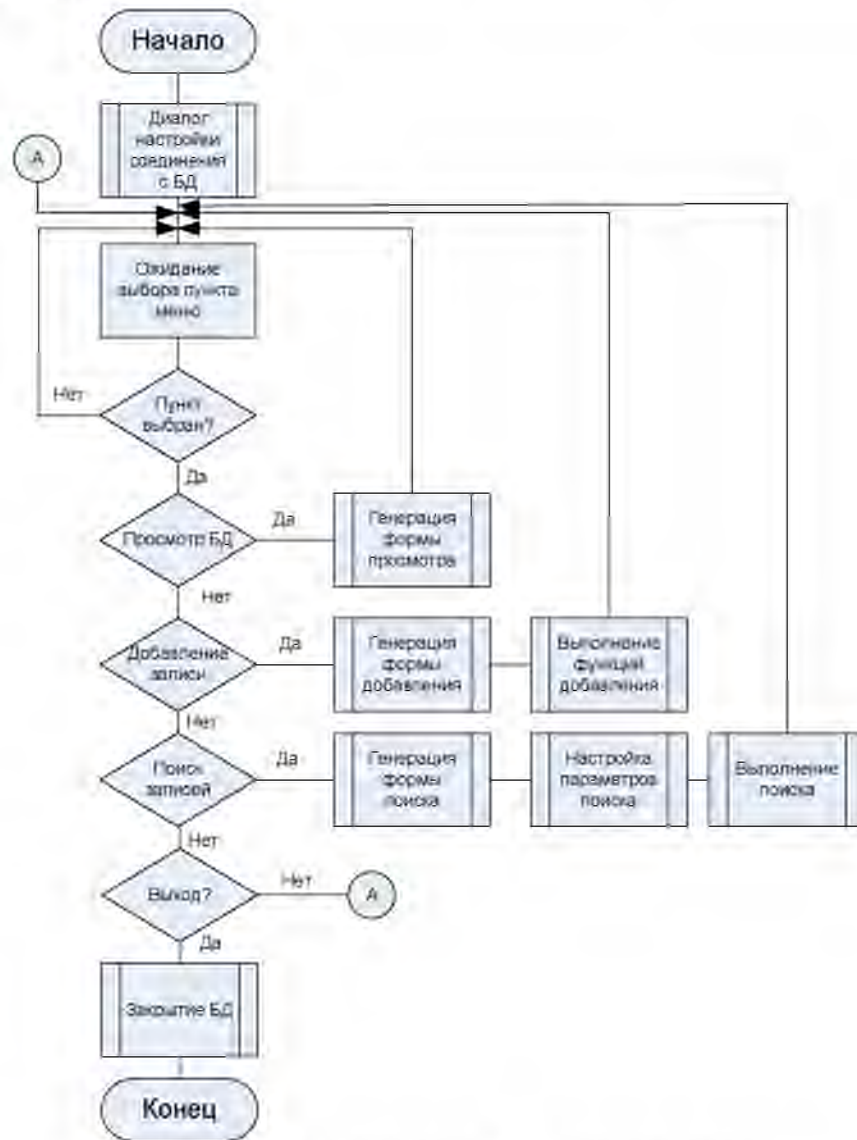


Рис. 3.4. Блок-схема работы приложения

ГЛАВА 4. РЕАЛИЗАЦИЯ СИСТЕМЫ НА ПРЕДПРИЯТИИ

В соответствии с требованиями система должна удовлетворять требованиям:

- Удобный интерфейс;
- Понятность конечному пользователю;
- Эффективность;
- Работоспособность.

После запуска, открывается главное окно программы «Обработка и доставки и приема» (рис. 4.1).

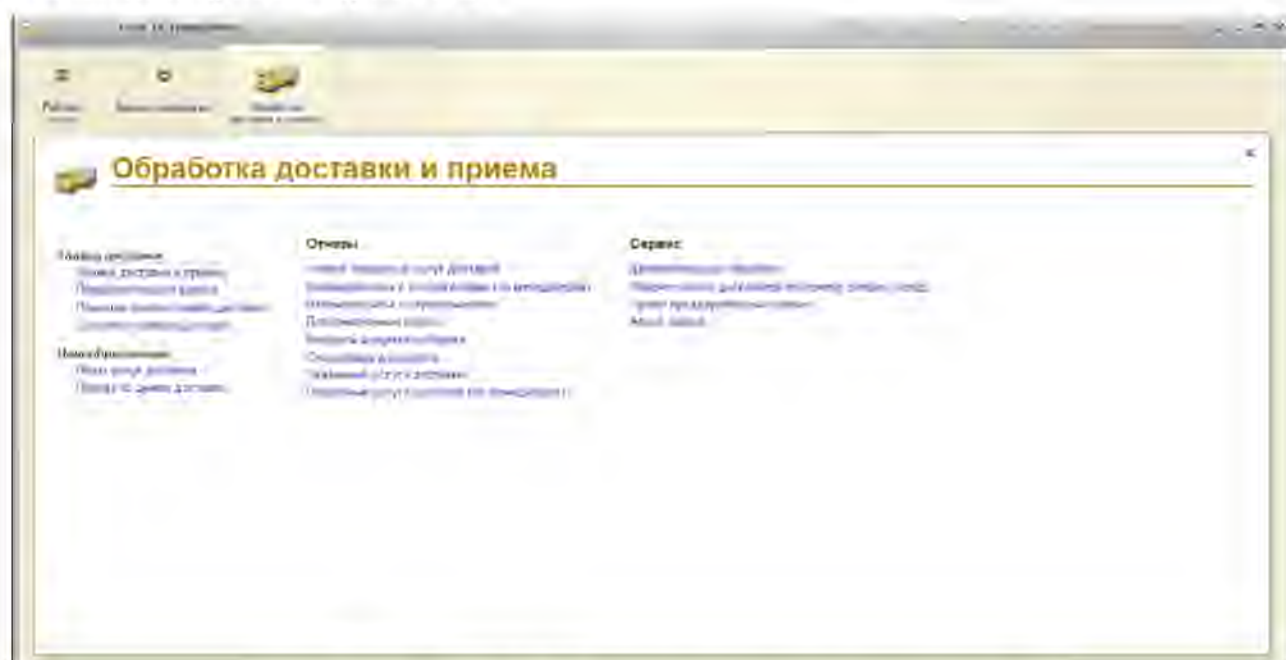


Рис. 4.1. Обработка и доставки и приема

«Журнал заявок: Заявка доставки и приема». Окно добавления заказа на доставку (рис. 4.2).



Рис. 4.2. Заявка доставки и приема

На вкладке «Заказ» (рис. 4.3) вводятся реквизиты, используемые для поиска подходящего транспортного средства. Поиск транспортного средства проводится в специальной форме подбора транспорта, которую можно запустить двумя способами: из текущей закладки по кнопке «Подбор транспорта» и меню «Доставка» - «Доступный транспорт». Подробнее см. описание «Поиск транспортного средства»:

- Транспорт – транспорт для перевозки. Вводится автоматически по результату выбора поиска транспорта. В случае если функция подбора не используется, выбирается из справочника транспортных средств.

- Водитель – водитель для выполнения текущего заказа. Вводится автоматически по результату выбора поиска транспорта. Выбирается из справочника «Водители».

- Вид – вид транспортного средства, требующегося заказчику. Выбирается из справочника «Виды транспорта». Служит одним из критериев в расчете цен закладки «Расчет стоимости». Вид транспорта используется в формировании прайс-листа, в расчете стоимости услуг и в подборе транспортного средства.

The screenshot displays a software window with a 'Заказ' (Order) tab. The interface is divided into several sections with various input fields and dropdown menus. On the left side, there are fields for 'Дата' (Date), 'Время' (Time), and 'Тип заказа' (Order type). The right side contains more detailed information, including 'Имя клиента' (Client name), 'Адрес' (Address), and 'Контактное лицо' (Contact person). At the bottom, there are fields for 'Организация' (Organization) and 'Телефон' (Phone). The overall layout is typical of a legacy business application.

Рис. 4.3. Вкладка Заказ

На вкладке «Маршрут» (рис. 4.4) - хранится информация о местах выгрузки и загрузки автомобиля или точек, в которые требуется следовать автомобилю:

- Тип - тип точки маршрута: загрузка/выгрузка/ таможня;
- Время – время, в которое нужно быть по указанному адресу;
- Адрес – адрес выгрузки/загрузки/таможни;
- Комментарий – комментарий;
- Организация – организация, находящаяся по указанному адресу;
- Контактное лицо – контактное лицо указанной организации;
- Телефон – телефон контактного лица.



Рис. 4.4. Вкладка Маршрут

Вкладка «Расчет стоимости» (рис. 4.5) - для расчета стоимости требуется установленное значение валюты управленческого учета (Сервис – Настройка параметров "Доставка" - Доходы и расходы).

По умолчанию, всегда устанавливается одна услуга. В каждой услуге можно изменить поле «вид услуги».

Вид услуги – вид услуги, оказываемой заказчику. Выбирается из справочника «Виды посреднических услуг». В качестве услуги в финансовых (регламентных) документах подставляются услуга, указанная в справочнике «Виды услуг доставки», как услуга для указанного вида услуги:

- Включать в стоимость – логическое значение. Включает/выключает услугу для расчета стоимости;
- Валюта – валюта расчета стоимости. Значение справочника «валюты»;
- Цена – цена за услугу;
- Количество – количество оказанных услуг;
- Сумма – сумма за услугу;
- Способ расчета – единица по услуге;
- Вид оплаты – метод расчета;
- Контрагент - юридическое или физическое лицо, оказывающее услугу. Выбирается из справочника «контрагенты»;

- Договор – договор с контрагентом на оказание услуг. Выбирается из справочника «договоры»;
- Не пересчитывать – флаг. Если флаг установлен, то цены в этой строке, по нажатию на кнопку «Перезаполнить цены», перезаполняться не будут. (Видимость данного столбца может быть включена/отключена в настройках пользователя: «Сервис» - «Настройка пользователя» - «Экседирование» - «Заявка» - «Скрывать флажки для ручного изменения цен»).

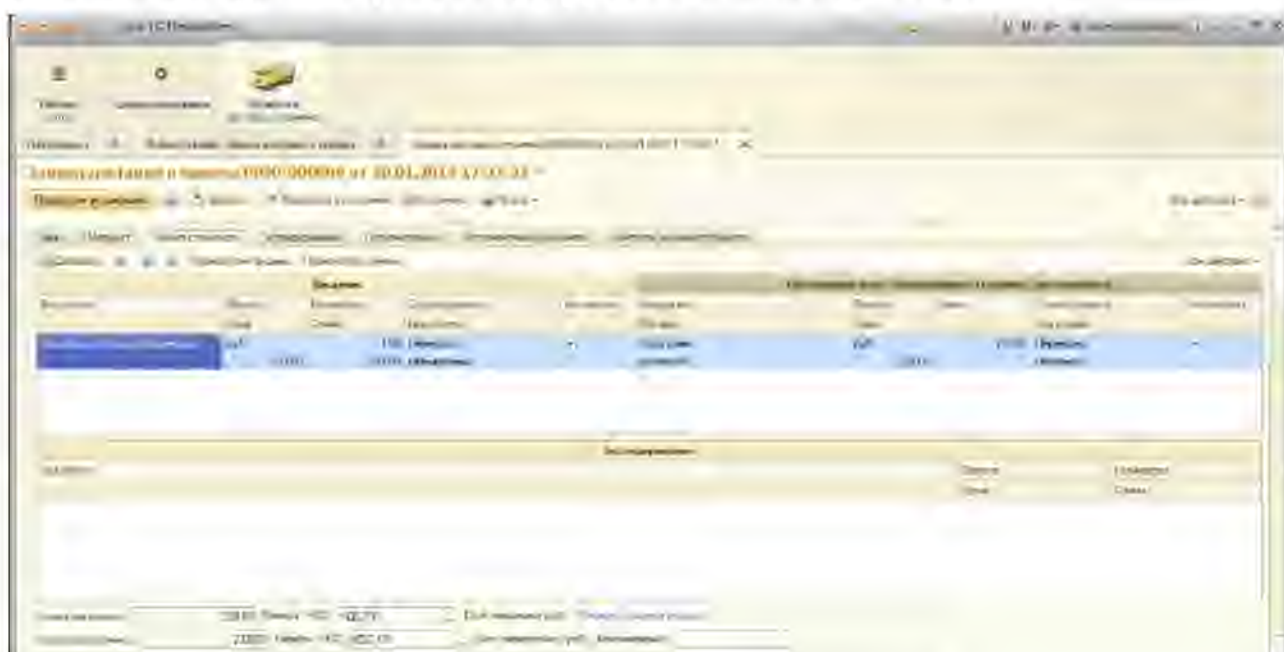


Рис. 4.5. Вкладка расчет стоимости

Вкладка «Дополнительно» (рис. 4.6) - объединяет информацию о заказчике услуги.

Параметры услуги:

- Организация – организация, осуществляющая посредническую услугу по текущей заявке;
- Автор – автор документа;
- Логист – логист, работавший с заказом;
- Телефон заказчика – телефон заказчика;
- Создать документ: «Путевой лист» - создание путевого листа на основании документа «Заявка доставки».

- Виды оплаты:
- Вид оплаты заказчика – вид оплаты заказчика. Значение берется, указанное в рабочем месте диспетчера;
- Вид оплаты перевозчику – вид оплаты перевозчику. Значение берется, указанное в рабочем месте диспетчера.

Дополнительно:

- Грузоотправитель – грузоотправитель;
- Грузополучатель – грузополучатель;
- Организация по адресу - организация, находящаяся по адресу загрузки;
- Организация по адресу разгрузки - организация, находящаяся по адресу выгрузки;
- Контактное лицо по адресу – контактное лицо по адресу загрузки;
- Контактное лицо по адресу разгрузки – контактное лицо по адресу разгрузки;
- Телефон по адресу – телефон контактного лица по адресу загрузки;
- Телефон по адресу разгрузки – телефон контактного лица по адресу разгрузки.

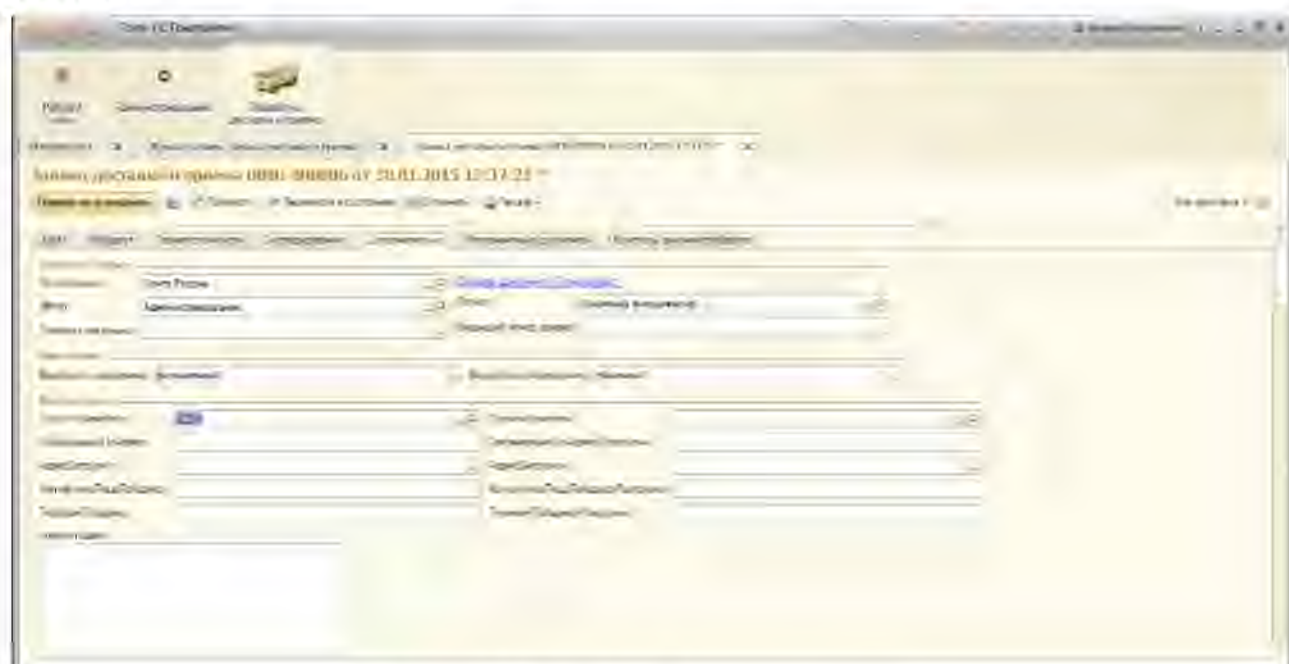


Рис. 4.6. Вкладка дополнительно

Вкладка «Регламентные документы» (рис. 4.7) – формирование документов реализации, поступления и счета.

Для ввода нового документа используются кнопки «Реализация», «Поступление», «Счет».

Созданные регламентные (финансовые) документы группируются по типу (реализация, поступление, счет), а общая сумма по документам рассчитывается в поле «Сумма документа».

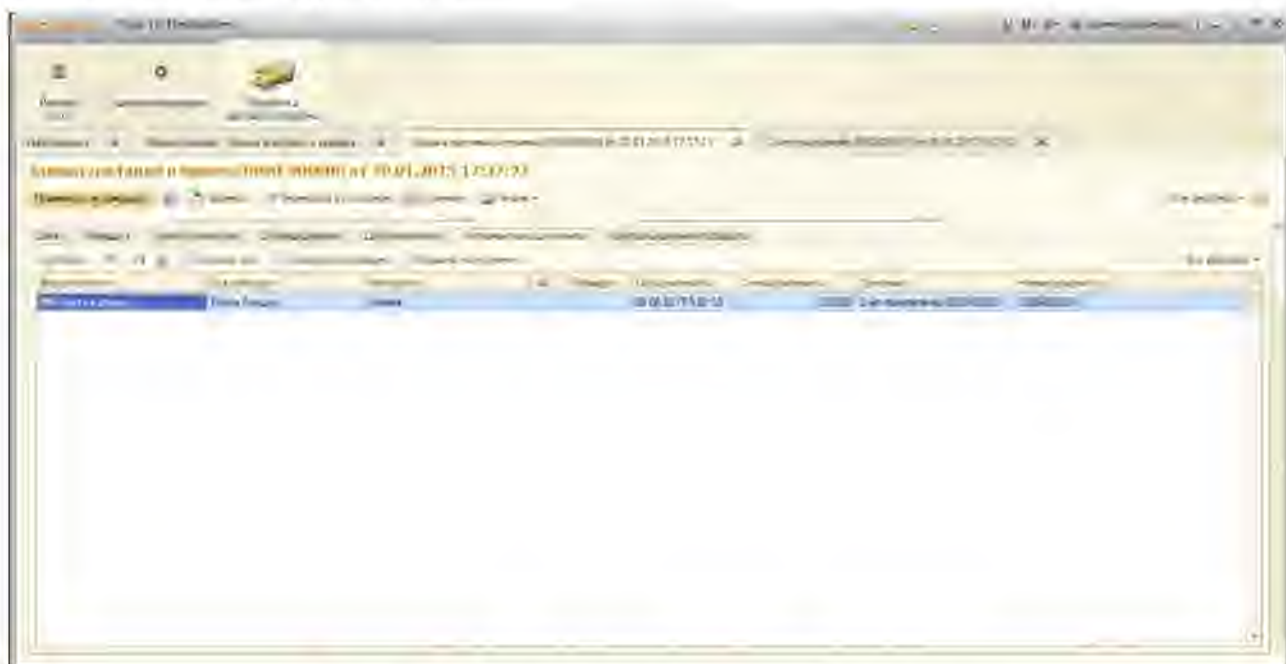


Рис. 4.7. Вкладка регламентные документы

Вкладка «Контроль документооборота» (рис. 4.8) - требуется для контроля получение (по факсу, по почте и т.д.), выписку документа можно использовать контроль документооборота. Контроль осуществляется на закладке «Контроль документооборота» в заявке «Доставка».

Список документов, которые требуется контролировать, заполняется по умолчанию или из шаблона. Для этого используются одноименные кнопки - «Заполнить по умолчанию» и «Заполнить из шаблона».

Документ "Заявка доставки" в состоянии "Выполнен" или "Отменен" попадает в журнал "Архив заявок".

Создание записи о клиенте, и внесении данных клиента (рис. 4.10).

Рис. 4.10. Предварительная заявка (создание)

Окно «Причины отмены заявок доставки» (рис. 4.11) - место хранения возможных причин отказа доставки.

Причина отмены	Имя
Отмена заявки	00000001
Отмена заявки	00000002
Отмена заявки	00000003

Рис. 4.11. Причины отмены заявок доставки

Окно «Состояния заявок доставки» (рис. 4.12) - содержит информацию о статусе документа «Заявка доставки». Статус отображает текущее состояние по заявке.

Стандартными статусами, используемыми в системе, могут быть:

- В планировании;
- На исполнении;
- Выполнена;
- Отменена;
- Подготовка документов.

Эти статусы являются predeterminedными и записываются в справочник при первом запуске программы.

Помимо predeterminedных состояний можно создавать собственные состояния заявок.

Для каждого состояния заявки обязательно необходимо заполнить поле «Журнал», выбрав из списка «В планировании» или «На исполнении». Если данное поле оставить пустым, то заявка не будет попадать на вкладки «В планировании» - «На исполнении» в рабочем месте диспетчера!

А также необходимо заполнить табличную часть, перечислив в ней все состояния, в которые может быть переведена заявка из текущего состояния, указать для каждого состояния в табличной части роль. Только пользователь с указанной ролью сможет перевести заявку из текущего состояния в указанное. Если таких ролей несколько, необходимо добавить несколько строчек в табличную часть.

Флаг «Точка старта» устанавливается, если данное состояние является начальным для заявки доставки.

Флаг «Архив» используется для перемещения заявки доставки, попадающей в данное состояние, в архив.



Рис. 4.12. Состояния заявок доставки

Справочник «Виды услуг доставки» (рис. 4.13) - содержит информацию о предоставляемых услугах на предприятии.

При создании вида услуги доставки необходимо указать наименование, выбрать тип услуги доставки из предложенного списка:

- Перевозка;
- Экспедирование;
- Предоставление грузчиков;
- Страхование;
- Прочее.

Для вида услуги «перепробег», рекомендуется указывать тип услуги доставки «Прочее».

Флаг «Не перевыставлять» позволяет отметить данный вид услуг, как не перевыставляемые. В этом случае, при использовании перевыставляемых транспортных услуг (задается в настройках параметров «Доставка», находящихся на закладке «Администрирование») при заполнении «Вида услуги доставки», в рабочем месте диспетчера, будут использоваться виды услуг с установленным данным флагом.

Доставка Доступный транспорт (рис. 4.14) - предназначен для отбора транспорта по определенным критериям.

Наиболее частые критерии поиска транспорта расположены на форме списка с доступным транспортом.

Доступность транспорта определяется следующим образом:

Доступность наемного транспорта берется из договоров с контрагентами;

Доступность собственного транспорта указывается в документе «Планирование работы транспорта»;

В подборе не будет участвовать транспорт (как свой, так и наемный), который указан в документе «Планирование работы транспорта» как недоступный;

Использование в подборе транспорта, уже выбранного для исполнения заявок доставки зависит от значения параметра «Учитывать доступность транспорта», указанного в настройках пользователя (Сервис - Настройки пользователя - Доставка - Подбор транспорта - Учитывать доступность транспорта):

Не учитывать доступность – в подборе доступного транспорта будет выводиться как транспорт, который еще не выбран для исполнения заявок, так и транспорт, который уже выбран.

Выделять недоступный транспорт – транспорт, который в указанный период уже выбран для исполнения заявок будет выделен красным цветом;

Учитывать доступность – транспорт, который в указанный период уже выбран для исполнения заявок не будет показываться при подборе доступного транспорта.

Для поиска необходимо указать период, на который требуется транспорт.

Результатом поиска является список транспортных средств по установленным критериям. Обязательным условием является договор на перевозку с контрагентом.



Рис. 4.14. Доступный транспорт

«Рабочее место диспетчера по приему заявок (город)» предназначено для быстрого ввода документов «Заявка Доставка». Открыть обработку можно из меню: «Посреднические услуги» - «Рабочее место диспетчера по приему заявок (город)». Если у пользователя установлена роль «Диспетчер», то «Рабочее место диспетчера по приему заявок» открывается в режиме рабочего стола (рис. 4.15).

Предлагаемая последовательность действий оператора.

Выбрать заказчика. При выборе заказчика табличная часть «История» заполняется последними введенными документами «Посредническая услуга» по данному контрагенту.

Найти в списке наиболее похожий документ и выполнить двойной клик левой клавишей мыши по строке. При выборе строки из истории все реквизиты заявки заполняются значениями выбранной.

Откорректировать заявку. Можно изменить любой реквизит, а также указать количество таких документов.

Нажать кнопку «Добавить». Заполненный документ будет добавлен в список «История» и будет выделен жирным шрифтом.

Когда все заявки будут добавлены - нажать кнопку «Записать». При нажатии кнопки «Записать», по еще незаписанным заявкам, будут созданы документы «Посредническая услуга».

Описание реквизитов «Рабочее место диспетчера по приему заявок».

Вкладка «Рабочее место» - предназначена для ввода данных, необходимых для формирования заявки доставки.



Рис. 4.15. Вкладка Рабочие место

Данная закладка условно разделена на логические группы: Заказчик, параметры, история заявок.

Группа «Заказчик»

- Заказчик – юридическое или физическое лицо заказывающие услуги перевозки. Выбирается из справочника «Контрагенты». В случае если справочник пуст, необходимо добавить контрагентов в справочник;

- Договор – договор с контрагентом, подробнее см. описание «договор с контрагентом»;

- Телефон – телефон контактного лица. Выбирается из списка телефонов выбранного контактного лица заказчика;

- Контактное лицо – контактное лицо предприятия заказчика. Выбирается из списка, формирующегося автоматически из контактных лиц заказчика.

Группа «Параметры».

- Вид услуги – услуга, по которой будут вестись расчеты. Выбирается из справочника «Виды услуг доставки». Подробно о настройке справочника смотрите описание настройки «Виды услуг доставки»;
- Вид транспорта – вид транспортного средства, требующегося заказчику. Выбирается из справочника «Виды транспорта»;
- Описание груза – текстовое представление в свободной форме;
- Вид транспорта используется в формировании прайс-листа, в расчете стоимости услуг и в подборе транспортного средства;
- Подробнее о подборе транспортного средства см. описание «Подбор транспортного средства»;
- Длина, ширина, высота объем, вес – запрашиваемые характеристики транспорта;
- Типы кузовов – тип требуемого кузова транспортного средства. В документе тип кузова представлен всеми значениями справочника «Типы кузовов»;
- Способ загрузки – требуемый вид загрузки в транспортное средство. В документе заказа способ загрузки представлен значениями справочника «Способы загрузки»;
- Стоимость груза – стоимость груза, указанная заказчиком. Значение требуется для заключения договора страхования.

В случае если требуются грузчики или страховка активируется соответствующий признак. При активации требования грузчиков, вводится количество человек и часов.

Информация об адресе, дате, времени подачи транспорта.

Значения наименования, контактное лицо и телефон можно выбрать из списка ранее введенных значений по данному контрагенту.

В случае если адресов для транспорта несколько вы можете активировать реквизит «Показать маршрут». После активации интерфейс изменится, и адресная часть будет отражена в таблице. В табличной части можно добавлять

требуемое количество точек с условным разделением на «тип операции»: выгрузку, загрузку, таможня. В каждой точке можно указать время, дату, организацию, контактное лицо и телефон.

Расчеты:

- Сумма (заказчик), Сумма (перевозчик) – реквизиты, используемые для ввода сумм по заявке вручную. В случае установленных в этих полях значений автоматический расчет не будет произведен;

- Сумма (перевозчик), Оплата (перевозчик) – реквизиты, служащие для выбора суммы и метода расчета по услуге.

Группа «История».

В случае заполнения поля «Заказчик», в форме заказа все реквизиты сбрасываются, а в истории выводятся выполненные заказы.

В системе можно оперативно ввести реквизиты из заявок, отраженных в истории. Выбор осуществляется двойным кликом левой клавиши мыши по строке заказа.

Для передачи заявки оператору по подбору водителя и транспорта выполняется команда «Записать документ».

Вкладка «В планировании» представляет собой журнал документов «Заявка доставки», находящихся в планировании (рис. 4.16).



Рис. 4.18. Диаграмма оказанных услуг

Таблица 4.1

Реализация по статусам и видам услуг

Состояние	Итого
Вид услуги	Сумма реализации
В планировании	455 507,20
Грузчики/перевозка	1 040,00
	1 040,00
Перевозка по городу / час	425 682,20
	425 682,20
Перевозка по городу/перевозка	3 000,00
	3 000,00
Перевозка/неперевыставляемая услуга	775,00
	775,00
Перепробег	25 000,00
	25 000,00
Экспедирование	10,00
	10,00
Выполнена	4 500,00
Перевозка по городу / час	4 500,00
	4 500,00
На исполнении	8 665 600,00
Перевозка по городу / час	8 629 100,00
	8 629 100,00
Перевозка по городу/перевозка	36 000,00
	36 000,00
Перевозка/неперевыставляемая услуга	500,00
	500,00
Отменена	692 600,00
Грузчики/перевозка	1 600,00

Таблица 4.1. Продолжение

	1 600,00
Перевозка по городу / час	690 000,00
	690 000,00
Экспедирование	1 000,00
	1 000,00
Подготовка документов	71 333,00
Перевозка без НДС	70 000,00
	70 000,00
Перевозка по городу/перевозка	333,00
	333,00
Перевозка собственным транспортом	1 000,00
	1 000,00
Итого	9 889 540,20

Окно «Взаиморасчеты с контрагентами» (рис. 4.19) - представляет отчет сумм заявок оплаченной суммы и суммы, оставшейся для оплаты.



Рис. 4.19. Окно взаиморасчеты с контрагентами

Окно «Контроль документооборота» (рис. 4.20) - вывод таблицы движения документов внутри предприятия, с номером документа и фактическим выполнением доставки клиенту.

Окно «Приём предварительных заявок» (рис. 4.22) - оператор предприятия вводит данные данные клиента, адрес загрузки, адрес выгрузки, вид транспорта, даты загрузки/выгрузки и т.д.

The image shows a screenshot of a software application window titled "Приём предварительных заявок" (Reception of preliminary orders). The window has a standard Windows-style title bar and a menu bar with options like "Файл", "Правка", and "Сервис". Below the menu bar, there are several tabs or sections, with the active one being "Приём предварительных заявок". The main area of the window is a form with various input fields and labels. The labels are in Russian and include: "Имя клиента:" (Client name), "Адрес загрузки:" (Loading address), "Адрес выгрузки:" (Unloading address), "Вид транспорта:" (Type of transport), "Дата загрузки:" (Loading date), "Дата выгрузки:" (Unloading date), "Имя организации:" (Organization name), "Адрес организации:" (Organization address), "Телефон:" (Phone), "Факс:" (Fax), "E-mail:" (E-mail), "Имя менеджера:" (Manager name), "Адрес менеджера:" (Manager address), "Телефон менеджера:" (Manager phone), "E-mail менеджера:" (Manager e-mail). At the bottom of the window, there is a status bar with some text and icons.

Рис. 4.22. Окно предварительных заявок

Окно «Архив» (рис. 4.23) - база хранения заявок доставки и приема посылок с возможностью поиска заявки по номеру документа, организации, состоянию заказа.

Дата	Имя	Содержание	Статус	Сумма	Вид заявки	Исполнитель
01.01.2024	Иванов И.И.	Заказ строительных материалов	В работе	10000	Заказ	Иванов И.И.
02.01.2024	Петров П.П.	Заказ строительных материалов	В работе	20000	Заказ	Петров П.П.
03.01.2024	Сидоров С.С.	Заказ строительных материалов	В работе	15000	Заказ	Сидоров С.С.
04.01.2024	Климов К.К.	Заказ строительных материалов	Завершено	8000	Заказ	Климов К.К.

Рис. 4.23. Окно архив заявок

ЗАКЛЮЧЕНИЕ

Темой дипломного проекта являлась автоматизированная система приема обработки доставки и вручения международных посылок для организации почта России. Актуальность данной темы обосновывается необходимостью соответствовать критерием эффективности использования рабочего труда современной экономики.

В результате был реализован программный продукт, призванный помочь в автоматизации приема обработки доставки и вручения международных посылок для организации почта России.

В процессе написания дипломного проекта были выполнены следующее стадии проектирования:

- анализ предметной области;
- анализ деятельности предприятия;
- анализ потоков информации на предприятии;
- проектирование базы данных;
- физическая реализация базы данных;
- разработка приложения пользователя;
- физическая реализация приложения пользователя.

В результате дипломного проектирования были закреплены навыки, полученные в результате обучения, а также получены практические навыки в области реализации информационных систем связанных с учетом основных средств.

Дальнейшая модернизация программного продукта позволит создать полноценный комплекс программных средств в области проведения и архивации тендеров, специализированный на работе с организациями, работающих в сфере прямых продаж.

Реализованный программный продукт позволяет осуществлять автоматизированное выполнение бизнес-процессов по приему обработки доставки и вручения международных посылок для организации почта России. Экономическая эффективность программного продукта обуславливается уменьшением затрачиваемого времени на выполнение бизнес-процессов, повышает качество работы, снижает вероятности возникновения ошибок.

При эксплуатации разработанного программного продукта от пользователя не требуется знаний выше уровня «пользователь», поскольку программный продукт обладает дружественным и интуитивно понятным интерфейсом. Тестирование программы доказало её работоспособность и готовность к внедрению.

СПИСОК ЛИТЕРАТУРЫ

1. Баженова И. Ю. , Основы проектирования приложений баз данных, Издательства: Бинوم. Лаборатория знаний, Интернет-университет информационных технологий, 2008 г., , 328 стр.
2. Вендров А.М., CASE-технологии. Современные методы и средства проектирования информационных систем - М.: Финансы и статистика, 2007 г, 456 стр.
3. Вигерс Карл, Разработка требований к программному обеспечению, Пер, с англ. - М.:Издательско-торговый дом "Русская Редакция", 2008. -576с.: ил
4. Владиславлев Д. Н.,Современные бизнес-технологии в торговле, Издательство: Ось-89, М., 2009 г., 208 стр.
5. Гвоздева Т. В., Б. А. Баллод, Проектирование информационных систем, М, Издательство: Феникс, 2009 г., 512 стр.
6. Голицына О. Л., И. И. Попов, Н. В. Максимов, Т. Л. Партыка, Информационные технологии, М, Издательство Инфра-М, 2009 г., 608 стр.
7. Емельянова Н. З., Партыка Т. Л., И. И. Попов, Проектирование информационных систем, М, Издательство: Форум, 2009 г., 432 стр.
8. ЕСПД ГОСТ 19.402-78- «Единая система программной документации»
9. ЕСПД ГОСТ 19.503-79 – «Руководство системного программиста. Требования к содержанию и оформлению»
10. ЕСПД ГОСТ 19.505-79 «Руководство оператора. Требования к содержанию и оформлению».
11. ЕСПД ГОСТ 24.207-80 «Требования к содержанию документов по программному обеспечению»
12. Илюшечкин В. М. , Основы использования и проектирования баз данных, М, Издательство Юрайт, 2010 г., 224 стр.

13. Керимов В. Э., Иванова Е. В., Сухов Р. А., Управленческий учет на предприятиях розничной торговли. Учебное пособие, Издательство: Экзамен, М., 160 стр.
14. Коробов Н. А., Комлев А. Ю. Информационные технологии в торговле, М., Академия, 2011 г., 176 стр.
15. Котляров В. П., Т. В. Коликова, Основы тестирования программного обеспечения, Издательства: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2009 г., 288 стр.
16. Крутик А.Б. Никольская Е.Г. Экономика предприятия — СПб.: Лань, 2005.
17. Крылов Э.И., Журавкова И.В. Анализ эффективности деятельности предприятия. – М.: Финансы и статистика, 2007.

ПРИЛОЖЕНИЕ

Перем мЭтоНовый;
 Перем мФормаОткрыта Экспорт;
 Перем мОбъектКопирования;

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

Если ОбменДанными.Загрузка Тогда

Возврат;

КонецЕсли;

Если Не ЭтоНовый()Тогда

новаВзаиморасчеты.ПроверитьДатуЗапретаРедактирования(ЭтотОбъект.Ссылка,
Отказ);

КонецЕсли;

Если Отказ Тогда

Возврат;

КонецЕсли;

Если Не Факт Тогда

мЭтоНовый = ЭтоНовый();

КонецЕсли;

Если Не ЗначениеЗаполнено(Автор) Тогда

Автор = ПараметрыСеанса.ТекущийПользователь;

КонецЕсли;

//Координатор, руководитель при изменении заявки

Если (РольДоступна("новаКоординатор"))

И Не ЗначениеЗаполнено(АвторЛогист)

```

ИЛИ РольДоступна("новаРуководительОтдела"))
И ( Перевозчик <> Ссылка.Перевозчик
ИЛИ ДоговорПеревозчика <> Ссылка.ДоговорПеревозчика
ИЛИ КонтактноеЛицоПеревозчика <> Ссылка.КонтактноеЛицоПеревозчика
ИЛИ Транспорт <> Ссылка.Транспорт
ИЛИ Водитель <> Ссылка.Водитель
ИЛИ РезультатВыполнения <> Ссылка.РезультатВыполнения) Тогда

```

```

АвторЛогист = ПараметрыСеанса.ТекущийПользователь;

```

```

КонецЕсли;

```

```

//Руководитель при изменении заявки

```

```

//Если РольДоступна("новаРуководительОтдела") Тогда

```

```

//

```

```

// Реквизиты = ЭтотОбъект.Метаданные().Реквизиты;

```

```

// Для Каждого МД Из Реквизиты Цикл

```

```

//

```

```

// Если МД.Имя = "Перевозчик" ИЛИ МД.Имя = "ДоговорПеревозчика"

```

```

// ИЛИ МД.Имя = "КонтактноеЛицоПеревозчика" ИЛИ МД.Имя =

```

```

"Транспорт"

```

```

// ИЛИ МД.Имя = "Водитель" ИЛИ МД.Имя =

```

```

"РезультатВыполнения" Тогда

```

```

//

```

```

// Если ЭтотОбъект[МД.Имя] <> Ссылка[МД.Имя]

```

```

// Тогда

```

```

// Автор = ПараметрыСеанса.ТекущийПользователь;

```

```

// Прервать;

```

```

// КонецЕсли;

```

```

//

```

```

// КонецЕсли;

```

```

//

```

```

// //Изменение других данных не должно приводить к изменению логиста.

```

```

// //++

```

```

//      //Если ЭтотОбъект[МД.Имя] <> Ссылка[МД.Имя] Тогда
//      //      АвторЛогист = ПараметрыСеанса.ТекущийПользователь;
//      //      Прервать;
//      //КонецЕсли;
//      //--
//
//      КонецЦикла;
//
//
//КонецЕсли;

новаСостояние = Неопределено;
Если ДополнительныеСвойства.Свойство("новаСостояние", новаСостояние) Тогда
    АвторЛогист = ПараметрыСеанса.ТекущийПользователь;
КонецЕсли;

ФормироватьВзаиморасчеты = Факт;

Если      ЭтоНовый()      И
Константы.почта_ПеревыставлятьТранспортныеУслуги.Получить() Тогда
    ПеревыставлятьУслуги = Истина;
КонецЕсли;

КонецПроцедуры
//Добавить учет в регистре
Процедура ПриЗаписи(Отказ)

    //НаборЗаписей      =
РегистрыСведений.новаСостоянияЗаявокЭкспедирования.СоздатьНаборЗаписей();
//СсылкаПлан = Ссылка;
//НаборЗаписей.Отбор.Услуга.Установить(СсылкаПлан);
Запись = Неопределено;
//Если мЭтоНовый Тогда
//      Запись = НаборЗаписей.Добавить();

```

```

//      Запись.Состояние = ЭтотОбъект.РезультатВыполнения;
//Иначе
//      НаборЗаписей.Прочитать();
//      Если НаборЗаписей.Количество() > 0 Тогда
//          Запись = НаборЗаписей[0];
//      КонецЕсли;
//КонецЕсли;

//Если Запись <> Неопределено Тогда
//      //ЗаполнитьЗначенияСвойств(Запись, ЭтотОбъект);
//      Запись.Состояние = Ссылка.РезультатВыполнения;
//      Запись.Услуга = СсылкаПлан;
//КонецЕсли;
//Если ((Не Ссылка.РезультатВыполнения.Архив)
//
И(Ссылка.РезультатВыполнения<>Справочники.новаСостоянияЗаявокЭкспедировани
я.Отменена)
//      И(Не Ссылка.ПометкаУдаления)) Тогда
//      НаборЗаписей.Записать();
//Иначе
//      НаборЗаписей.Очистить();
//      Запись
=
РегистрыСведений.новаСостоянияЗаявокЭкспедирования.СоздатьМенеджерЗаписи();
//      Запись.Услуга = СсылкаПлан;
//      Запись.Прочитать();
//      Запись.Удалить();
//КонецЕсли;
//НаборЗаписей = РегистрыСведений.почта_ЭтапыММП.СоздатьНаборЗаписей();
//СсылкаПлан = Ссылка;
//НаборЗаписей.Отбор.Регистратор.Установить(СсылкаПлан);
//НаборЗаписей.Записывать = Истина;
//
//Запись = Неопределено;
//      Если мЭтоНовый Тогда

```

```

//      Запись = НаборЗаписей.Добавить();
//      Иначе
//      НаборЗаписей.Прочитать();
//      Если НаборЗаписей.Количество() > 0 Тогда
//      Запись = НаборЗаписей[0];
//      Иначе
//      Запись = НаборЗаписей.Добавить();
//      КонецЕсли;
//
//      КонецЕсли;
//
//      Если Запись <> Неопределено Тогда
//      //ЗаполнитьЗначенияСвойств(Запись, ЭтотОбъект);
//      Запись.Состояние = Ссылка.РезультатВыполнения;
//      //Запись.Услуга = СсылкаПлан;
//      Запись.ДатаНачала      = Ссылка.ВремяС;
//      Запись.ДатаЗавершения = Ссылка.ВремяВыгрузкиС;
//      Запись.АдресНач       = Ссылка.АдресЗагрузки;
//      Запись.АдресКон       = Ссылка.АдресВыгрузки;
//      Запись.ИдМаршрута     = ИдМаршрута;
//      Запись.ЗаявкаММП     = Основание;
//      Запись.Заявка        = Ссылка;
//      КонецЕсли;
//      Если ((Не Ссылка.РезультатВыполнения.Архив)
//
И(Ссылка.РезультатВыполнения<>Справочники.новаСостоянияЗаявокЭкспедировани
я.Отменена)
//      И(Не Ссылка.ПометкаУдаления)) Тогда
//      НаборЗаписей.Записать(Истина);
//      Иначе
//      НаборЗаписей.Очистить();
//      Запись
=
РегистрыСведений.почта_ЭтапыММП.СоздатьМенеджерЗаписи();

```



```

//          //Запись.Услуга = СсылкаПлан;
//          Запись.Прочитать();
//          Запись.Удалить();
//      КонецЕсли;

```

КонецПроцедуры

Процедура УстановитьЦенуСтрокиГЧЗаказчик(СтрокаГЧ)

```

//Если Не СтрокаГЧ.ЗаказчикСуммаУказанаВручную Тогда
//      стЦена          =          новаЦенообразование.ПолучитьЦенуПрайса(ВремяС,
СтрокаГЧ.ВидУслуги,  Заказчик,  Заказчик.новаКатегорияКонтрагента,  ВидТранспорта,
Направление);
//      врЦена = 0;
//      врМинЦена = 0;
//      Если СтрокаГЧ.ЗаказчикВалюта = стЦена.Валюта Тогда
//          врЦена = стЦена.Цена;
//          врМинЦена = стЦена.МинЦена;
//      Иначе
//          врЦена          =
Ценообразование.ПересчитатьЦенуПриИзмененииВалюты(стЦена.Цена,      стЦена.Валюта,
СтрокаГЧ.ЗаказчикВалюта,,, Дата);
//          врМинЦена          =
Ценообразование.ПересчитатьЦенуПриИзмененииВалюты(стЦена.МинЦена,
стЦена.Валюта, СтрокаГЧ.ЗаказчикВалюта,,, Дата);
//      КонецЕсли;
//
//      Если врЦена*СтрокаГЧ.Количество < врМинЦена Тогда
//          СтрокаГЧ.ЗаказчикСумма = врМинЦена;
//          СтрокаГЧ.ЗаказчикЦена          =          врЦена;
//?(СтрокаГЧ.Количество=0,0,СтрокаГЧ.ЗаказчикСумма/СтрокаГЧ.Количество);
//      Иначе
//          СтрокаГЧ.ЗаказчикЦена = врЦена;

```

```

//          СтрокаГЧ.ЗаказчикСумма          =
СтрокаГЧ.ЗаказчикЦена*СтрокаГЧ.Количество;
//      КонецЕсли;
//КонецЕсли;
КонецПроцедуры

Процедура УстановитьЦенуСтрокиГЧПоставщик(СтрокаГЧ)
    Если Не СтрокаГЧ.ПоставщикСуммаУказанаВручную Тогда
        врВидТранспорта          =
?(ЗначениеЗаполнено(ВидТранспортаПеревозчика),ВидТранспортаПеревозчика,ВидТранспо
рта);
        стЦена          =          новаЦенообразование.ПолучитьЦенуПрайса(ВремяС,
СтрокаГЧ.ВидУслуги,          СтрокаГЧ.ПоставщикКонтрагент,
СтрокаГЧ.ПоставщикКонтрагент.новаКатегорияКонтрагента,          врВидТранспорта,
Направление);
        врЦена = 0;
        врМинЦена = 0;
        Если СтрокаГЧ.ПоставщикВалюта = стЦена.Валюта Тогда
            врЦена = стЦена.Цена;
            врМинЦена = стЦена.МинЦена;
        Иначе
            врЦена          =
новаЦенообразование.ПересчитатьЦенуПриИзмененииВалюты(стЦена.Цена,
стЦена.Валюта, СтрокаГЧ.ПоставщикВалюта,,, Дата);
            врМинЦена          =
новаЦенообразование.ПересчитатьЦенуПриИзмененииВалюты(стЦена.МинЦена,
стЦена.Валюта, СтрокаГЧ.ПоставщикВалюта,,, Дата);
        КонецЕсли;
        Если врЦена*СтрокаГЧ.Количество < врМинЦена Тогда
            СтрокаГЧ.ПоставщикСумма = врМинЦена;
            СтрокаГЧ.ПоставщикЦена          =          врЦена;
//?(СтрокаГЧ.Количество=0,0,СтрокаГЧ.ПоставщикСумма/СтрокаГЧ.Количество);
        Иначе
            СтрокаГЧ.ПоставщикЦена = врЦена;

```

```

        СтрокаТЧ.ПоставщикСумма = СтрокаТЧ.ПоставщикЦена*СтрокаТЧ.Количество;
        КонецЕсли;
    КонецЕсли;
КонецПроцедуры

// Возвращает строку табличной части "Расчет стоимости"
//
// Параметры:
// Тип <Строка>
//
// Возвращаемое значение:
// <Число>
//
Функция ПолучитьСтрокуТЧ(Тип) Экспорт
    Если Тип = "Перевозка" Тогда
        Возврат РасчетСтоимости.Найти(1, "ТипСтроки");
    ИначеЕсли Тип = "Перепробег" Тогда
        Возврат РасчетСтоимости.Найти(2, "ТипСтроки");
    ИначеЕсли Тип = "Грузчики" Тогда
        Возврат РасчетСтоимости.Найти(3, "ТипСтроки");
    ИначеЕсли Тип = "Страхование" Тогда
        Возврат РасчетСтоимости.Найти(4, "ТипСтроки");
    ИначеЕсли Тип = "Экспедирование" Тогда
        Возврат РасчетСтоимости.Найти(5, "ТипСтроки");
    КонецЕсли;
КонецФункции

// Устанавливает тип строки для табличной части "Расчет стоимости"
//
// Параметры:
// СтрокаТЧ <СтрокаТабличнойЧасти>;
// Тип <Строка>;
//
Процедура УстановитьТипСтрокиТЧ(СтрокаТЧ, Тип) Экспорт

```

```

Если Тип = "Перевозка" Тогда
    СтрокаГЧ.ТипСтроки = 1;
ИначеЕсли Тип = "Перепробег" Тогда
    СтрокаГЧ.ТипСтроки = 2;
ИначеЕсли Тип = "Грузчики" Тогда
    СтрокаГЧ.ТипСтроки = 3;
ИначеЕсли Тип = "Страхование" Тогда
    СтрокаГЧ.ТипСтроки = 4;
ИначеЕсли Тип = "Экспедирование" Тогда
    СтрокаГЧ.ТипСтроки = 5;
КонецЕсли;
КонецПроцедуры

// Возвращает тип строки табличной части "Расчет стоимости" в виде символьной строки
//
// Параметры:
// СтрокаГЧ <СтрокаТабличнойЧасти>
//
// Возвращаемое значение:
// <Строка>
//
Функция ПолучитьТипСтрокиГЧ(СтрокаГЧ) Экспорт
    Если СтрокаГЧ.ТипСтроки=1 Тогда
        Возврат "Перевозка";
    ИначеЕсли СтрокаГЧ.ТипСтроки=2 Тогда
        Возврат "Перепробег";
    ИначеЕсли СтрокаГЧ.ТипСтроки=3 Тогда
        Возврат "Грузчики";
    ИначеЕсли СтрокаГЧ.ТипСтроки=4 Тогда
        Возврат "Страхование";
    ИначеЕсли СтрокаГЧ.ТипСтроки=5 Тогда
        Возврат "Экспедирование";
    КонецЕсли;
КонецФункции

```

// Заполняет строку табличной части "Расчет стоимости"

//

// Параметры:

// СтрокаТЧ <СтрокаТабличнойЧасти>:

//

Процедура ЗаполнитьРеквизитыТЧ(СтрокаТЧ) Экспорт

СтрокаТЧ.ВключатьВСтоимость = Истина;

СтрокаТЧ.ЗаказчикВидОплаты =

Перечисления.новаВидОплаты.Безналичный;

СтрокаТЧ.ЗаказчикСпособРасчета = СтрокаТЧ.ВидУслуги.ТипЦены;

СтрокаТЧ.ЗаказчикВалюта = СтрокаТЧ.ВидУслуги.Валюта;

УстановитьЦенуСтрокиТЧЗаказчик(СтрокаТЧ);

СтрокаТЧ.ПоставщикВалюта = СтрокаТЧ.ВидУслуги.Валюта;

СтрокаТЧ.ПоставщикВидОплаты =

Перечисления.новаВидОплаты.Безналичный;

СтрокаТЧ.ПоставщикКонтрагент = Перевозчик;

СтрокаТЧ.ПоставщикДоговорКонтрагента = ДоговорПеревозчика;

СтрокаТЧ.ПоставщикСпособРасчета = СтрокаТЧ.ВидУслуги.ТипЦены;

УстановитьЦенуСтрокиТЧПоставщик(СтрокаТЧ);

//20092011 Колоколкин -----

Если Константы.почта_ПеревыставлятьТранспортныеУслуги.Получить() Тогда

СтрокаТЧ.ЗаказчикЦена = СтрокаТЧ.ПоставщикЦена;

СтрокаТЧ.ЗаказчикСумма = СтрокаТЧ.ПоставщикСумма;

КонецЕсли;

//

КонецПроцедуры

// Заполняет табличную часть "Расчет стоимости" данными соответствующими выбранным видам услуг

//

Процедура ЗаполнитьТабличнуюЧастьРасчетСтоимости() Экспорт

Если Не ЗначениеЗаполнено(ВидУслуги) Тогда Возврат, КонецЕсли;

Если ЗначениеЗаполнено(ВидУслуги) Тогда

СтрокаГЧ = ПолучитьСтрокуГЧ("Перевозка");

Если СтрокаГЧ <> Неопределено Тогда

НовСтрока = СтрокаГЧ;

Иначе

НовСтрока = РасчетСтоимости.Добавить();

КонецЕсли;

НовСтрока.ВидУслуги = ВидУслуги;

НовСтрока.ПоставщикКонтрагент= Перевозчик;

НовСтрока.ПоставщикДоговорКонтрагента = ДоговорПеревозчика;

Если ВидУслуги.ТипЦены =

Перечисления.новаТипыЦенУслугЭкспедирования.ЗаЕдиницуПеревозки Тогда

НовСтрока.Количество = 1;

ИначеЕсли ВидУслуги.ТипЦены =

Перечисления.новаТипыЦенУслугЭкспедирования.ЗаКилограмм Тогда

НовСтрока.Количество = Вес*1000;

ИначеЕсли ВидУслуги.ТипЦены =

Перечисления.новаТипыЦенУслугЭкспедирования.ЗаКилометр Тогда

НовСтрока.Количество = Пробег;

ИначеЕсли ВидУслуги.ТипЦены =

Перечисления.новаТипыЦенУслугЭкспедирования.ЗаПеревозку Тогда

НовСтрока.Количество = 1;

ИначеЕсли ВидУслуги.ТипЦены =

Перечисления.новаТипыЦенУслугЭкспедирования.ЗаЧас Тогда

Если ЗначениеЗаполнено(КоличествоЧасов) Тогда

НовСтрока.Количество =

КоличествоЧасов+(ПлюсЧасНаПодачу,1,0);

КонецЕсли;

КонецЕсли;

УстановитьТипСтрокиГЧ(НовСтрока,"Перевозка");

ЗаполнитьРеквизитыГЧ(НовСтрока);

КонецЕсли;

СтрокаГЧ = ПолучитьСтрокуГЧ("Перепробег");

Если СтрокаГЧ <> Неопределено Тогда

 СтрокаГЧ.ВидУслуги = ВидУслуги;

 ЗаполнитьРеквизитыГЧ(СтрокаГЧ);

КонецЕсли;

КонецПроцедуры

// Вызывается для нового объекта и заполняет данные по умолчанию

//

Процедура ИнициализацияНового() Экспорт

 Если Не ЗначениеЗаполнено(ВидУслуги) Тогда

 ОсновнаяУслуга

=

новаУправлениеПользователямиСервер.ПолучитьЗначениеНастройки("ОсновнаяУслугаПоП
еревозке");

 Если ЗначениеЗаполнено(ОсновнаяУслуга) Тогда

 ВидУслуги = ОсновнаяУслуга;

 КонецЕсли;

КонецЕсли;

врСумма = СуммаЗаказчика;

врСуммаП = СуммаПеревозчика;

ЗаполнитьТабличнуюЧастьРасчетСтоимости();

ДобавитьУдалитьСтрокуГрузчики();

ДобавитьУдалитьСтрокуЭкспедирование();

//++

Для Каждого СтрРС из РасчетСтоимости Цикл

 Если СтрРС.ТипСтроки <> 1 Тогда

 врСумма = врСумма - СтрРС.ЗаказчикСумма;

```

        врСуммаП = врСуммаП - СтрРс.ПоставщикСумма;
    КонецЕсли;
КонецЦикла;
//--

//+++
    СтрокаГЧ = ПолучитьСтрокуГЧ("Перевозка");
Если СтрокаГЧ <> Неопределено Тогда
    СтрокаГЧ.ЗаказчикВидОплаты = ВидОплатыЗаказчика;
    Если СуммаЗаказчикаУстановленаВручную Тогда
        СтрокаГЧ.ЗаказчикСумма = врСумма;
        СтрокаГЧ.ЗаказчикСуммаУказанаВручную = Истина;
        СтрокаГЧ.Количество = ?(СтрокаГЧ.Количество = 0, 1,
СтрокаГЧ.Количество);
        СтрокаГЧ.ЗаказчикЦена = ?(СтрокаГЧ.Количество = 0, 0,
СтрокаГЧ.ЗаказчикСумма/СтрокаГЧ.Количество);
    КонецЕсли;

    СтрокаГЧ.ПоставщикВидОплаты = ВидОплатыПеревозчику;
    Если СуммаПеревозчикаУстановленаВручную Тогда
        СтрокаГЧ.ПоставщикСумма = врСуммаП;
        СтрокаГЧ.ПоставщикСуммаУказанаВручную = Истина;
        СтрокаГЧ.Количество = ?(СтрокаГЧ.Количество = 0, 1,
СтрокаГЧ.Количество);
        СтрокаГЧ.ПоставщикЦена = ?(СтрокаГЧ.Количество = 0, 0,
СтрокаГЧ.ПоставщикСумма/СтрокаГЧ.Количество);
    КонецЕсли;

КонецЕсли;

РасчитатьИтоговыеСуммы();
ЗаполнитьТаблицуДокументов();

Автор = ПараметрыСеанса.ТекущийПользователь;

```



```

Если Не ЗначениеЗаполнено(Организация) Тогда
    Организация =
ОбщегоНазначенияБПВызовСервера.ПолучитьЗначениеПоУмолчанию("ОсновнаяОрганизац
ия");
КонецЕсли;

```

```

СтавкаНДСЗаказчик = Перечисления.новаСтавкиНДС.НДС18;
СтавкаНДСПеревозчик = Перечисления.новаСтавкиНДС.НДС18;

```

```

Если Маршрут.Количество() < 2 Тогда
    НовСтрока = Маршрут.Добавить();
    НовСтрока.Время = НачалоДня(ТекущаяДата()+3600*24);
    НовСтрока.Тип = Перечисления.новаТипыАдресов.Загрузка;
    НовСтрока = Маршрут.Добавить();
    НовСтрока.Время = НачалоДня(ТекущаяДата()+3600*24);
    НовСтрока.Тип = Перечисления.новаТипыАдресов.Выгрузка;
КонецЕсли;

```

```

РезультатВыполнения =
новаПосредническиеУслуги.ПолучитьСостояниеТочкаСтарта();

```

КонецПроцедуры

```

// Заполняет табличную часть "Регламентные документы" значениями из справочника
//

```

Процедура ЗаполнитьТаблицуДокументов() Экспорт

```

Запрос = Новый Запрос;

```

```

Запрос.Текст = "ВЫБРАТЬ

```

```

    | новаШаблоныДокументовТО.Ссылка,
    | новаШаблоныДокументовТО.ПометкаУдаления,
    | новаШаблоныДокументовТО.Предопределенный,
    | новаШаблоныДокументовТО.Код,
    | новаШаблоныДокументовТО.Наименование,
    | новаШаблоныДокументовТО.Использование

```

```

      ИЗ
      |   Справочник.новаШаблоныДокументовТО
новаШаблоныДокументовТО
      |   ГДЕ
      |   новаШаблоныДокументовТО.Использование = ИСТИНА";
      Результат = Запрос.Выполнить().Выгрузить();
      РегламентныеДокументы.Очистить();
      Для Каждого СтрокаТаб Из Результат Цикл
          НовСтрока = РегламентныеДокументы.Добавить();
          НовСтрока.Документ = СтрокаТаб.Ссылка;
      КонецЦикла;
КонецПроцедуры

// Добавляет или удаляет строку отражающую услугу по предоставлению грузчиков
// в табличную часть "Расчет стоимости".
//
Процедура ДобавитьУдалитьСтрокуГрузчики() Экспорт

    Если ТребуетсяГрузчики Тогда

        СтрокаТЧ = ПолучитьСтрокуТЧ("Грузчики");
        Если СтрокаТЧ <> Неопределено Тогда
            Возврат;
        КонецЕсли;

        УслугаГрузчики =
новаУправлениеПользователямиСервер.ПолучитьЗначениеНастройки("ОсновнаяУслугаГруз
чики");

        Если Не ЗначениеЗаполнено(УслугаГрузчики) Тогда
            Сообщить("Внимание. Не заполнено значение для услуги по
предоставлению грузчиков в настройках пользователя.");
            УслугаГрузчики = ВидУслуги;
        КонецЕсли;

```

```

НовСтрока = РасчетСтоимости.Добавить();
НовСтрока.ВидУслуги = УслугаГрузчики;
НовСтрока.Количество = КоличествоГрузчиков*КоличествоЧасовГрузчики,
//человекочасы
    УстановитьТипСтрокиГЧ(НовСтрока,"Грузчики");
    ЗаполнитьРеквизитыГЧ(НовСтрока);

Иначе
    СтрокаГЧ = ПолучитьСтрокуГЧ("Грузчики");
    Если СтрокаГЧ <> Неопределено Тогда
        Индекс = РасчетСтоимости.Индекс(СтрокаГЧ);
        РасчетСтоимости.Удалить(Индекс);
    КонецЕсли;

КонецЕсли;
КонецПроцедуры

// Добавляет или удаляет строку отражающую услугу по страхованию груза
// в табличную часть "Расчет стоимости".
//
Процедура ДобавитьУдалитьСтрокуСтрахование() Экспорт

    Если ТребуетсяСтраховка Тогда

        Запрос = Новый Запрос;

        Запрос = Новый Запрос;
        Запрос.Текст = "ВЫБРАТЬ
            |     новаСтраховка.Ссылка КАК Страховка
            |ИЗ
            |     Документ.новаСтраховка КАК новаСтраховка
            |ГДЕ
            |     новаСтраховка.Основание = &Основание";
        Запрос.УстановитьПараметр("Основание",ЭтотОбъект.Ссылка);

```

тзСтраховка = Запрос.Выполнить().Выгрузить();

Если тзСтраховка.Количество() = 0 Тогда

 Возврат;

КонецЕсли;

УслугаСтрахование

=

новаУправлениеПользователямиСервер.ПолучитьЗначениеНастройки("ОсновнаяУслугаПоСтрахованию");

Если Не ЗначениеЗаполнено(УслугаСтрахование) Тогда

 Сообщить("Внимание. Не заполнено значение для услуги по страхованию в настройках пользователя. Используется услуга по перевозке.");

 УслугаСтрахование = ВидУслуги;

КонецЕсли;

врСтраховка = Документы.новаСтраховка.ПустаяСсылка();

НовСтрока = ПолучитьСтрокуТЧ("Страхование");

Если НовСтрока = Неопределено Тогда

 НовСтрока = РасчетСтоимости.Добавить();

 НовСтрока.ВидУслуги = УслугаСтрахование;

 НовСтрока.Количество = 1;

 УстановитьТипСтрокиТЧ(НовСтрока, "Страхование");

 ЗаполнитьРеквизитыТЧ(НовСтрока);

КонецЕсли;

Если тзСтраховка.Количество() > 0 Тогда

 врСтраховка = тзСтраховка[0].Страховка;

 НовСтрока.ЗаказчикЦена = врСтраховка.СуммаЗаказчика;

 НовСтрока.ЗаказчикСумма = НовСтрока.ЗаказчикЦена *

НовСтрока.Количество;

 НовСтрока.ПоставщикКонтрагент = врСтраховка.СтраховаяКомпания;

 НовСтрока.ПоставщикДоговорКонтрагента = врСтраховка.Договор;

 НовСтрока.ПоставщикЦена = врСтраховка.СтраховаяПремия;

```

НовСтрока.ПоставщикСумма = НовСтрока.ПоставщикЦена *
НовСтрока.Количество;
    КонечЕсли;

    Иначе
        СтрокаТЧ = ПолучитьСтрокуТЧ("Страхование");
        Если СтрокаТЧ <> Неопределено Тогда
            Индекс = РасчетСтоимости.Индекс(СтрокаТЧ);
            РасчетСтоимости.Удалить(Индекс);
        КонечЕсли;

    КонечЕсли;
КонечПроцедуры

// Добавляет или удаляет строку отражающую услугу по экспедированию
// в табличную часть "Расчет стоимости".
//
Процедура ДобавитьУдалитьСтрокуЭкспедирование() Экспорт
    Если ТребуетсяЭкспедирование Тогда
        УслугаЭкспедирование =
новаУправлениеПользователямиСервер.ПолучитьЗначениеНастройки("ОсновнаяУслугаПоЭ
кспедированию");
        Если Не ЗначениеЗаполнено(УслугаЭкспедирование) Тогда
            Сообщить("Внимание. Не заполнено значение для услуги по
экспедированию в настройках пользователя.");
            УслугаЭкспедирование = ВидУслуги;
        КонечЕсли;

        СтрокаТЧ = ПолучитьСтрокуТЧ("Экспедирование");
        Если СтрокаТЧ <> Неопределено Тогда
            Возврат;
        КонечЕсли;

        НовСтрока = РасчетСтоимости.Добавить();

```

```

НовСтрока.ВидУслуги = УслугаЭкспедирование;

СтрокаПеревозка = ПолучитьСтрокуТЧ("Перевозка");
Если СтрокаПеревозка <> Неопределено И
ЗначениеЗаполнено(СтрокаПеревозка.Количество) Тогда
    НовСтрока.Количество = СтрокаПеревозка.Количество;
Иначе
    НовСтрока.Количество = 1;
КонецЕсли;
УстановитьТипСтрокиТЧ(НовСтрока,"Экспедирование");
ЗаполнитьРеквизитыТЧ(НовСтрока);

Иначе
    СтрокаТЧ = ПолучитьСтрокуТЧ("Экспедирование");
    Если СтрокаТЧ <> Неопределено Тогда
        Индекс = РасчетСтоимости.Индекс(СтрокаТЧ);
        РасчетСтоимости.Удалить(Индекс);
    КонецЕсли;

КонецЕсли;
КонецПроцедуры

// Заполняет табличную часть Расчет стоимости ценами по прайсу

Процедура ЗаполнитьЦеныТЧРасчетСтоимости() Экспорт
    Для Каждого СтрокаТЧ Из РасчетСтоимости Цикл
        ТипСтроки = ПолучитьТипСтрокиТЧ(СтрокаТЧ);
        Если ТипСтроки = "Страхование" Тогда
            Продолжить;
        КонецЕсли;
        //Если ЭтотОбъект.ТипПеревозки = Перечисления.почта_ТипПеревозки.авто Тогда
        //УстановитьЦенуСтрокиТЧЗаказчик(СтрокаТЧ);
        УстановитьЦенуСтрокиТЧПоставщик(СтрокаТЧ);
        //Иначе

```

// Предупреждение("Для данного типа перевозки цены устанавливаются вручную");

//КонецЕсли;

КонецЦикла;

КонецПроцедуры

Процедура РасчитатьИтоговыеСуммы() Экспорт

СуммаЗаказчика = 0;

СуммаПеревозчика = 0;

УпрВалюта = Константы.новаВалютаУправленческогоУчета.Получить();

Если Не ЗначениеЗаполнено(УпрВалюта) Тогда

Сообщить("Не заполнено значение валюты управленческого учета
"+Символы.ПС

+"(Сервис - Настройка ""Экспедирование"" - Доходы и расходы)");

Возврат;

КонецЕсли;

Для Каждого СтрокаТЧ Из РасчетСтоимости Цикл

//Если Не СтрокаТЧ.ВключатьВСтоимость Тогда Продолжить; КонецЕсли;

СуммаЗаказчика =

СуммаЗаказчика+Ценообразование.ПересчитатьЦенуПриИзмененииВалюты(СтрокаТЧ.ЗаказчикСумма,СтрокаТЧ.ЗаказчикВалюта, УпрВалюта,,,Дата);

СуммаПеревозчика =

СуммаПеревозчика+Ценообразование.ПересчитатьЦенуПриИзмененииВалюты(СтрокаТЧ.ПоставщикСумма,СтрокаТЧ.ПоставщикВалюта, УпрВалюта,,,Дата);

КонецЦикла;

КонецПроцедуры

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

```

// регистр почта_ЭтапыММП
Движение                = Движения.почта_ЭтапыММП.Добавить();
Движение.ЗаявкаММП      = Основание;
Движение.Заявка         = Ссылка;
Движение.ДатаНачала     = ВремяС;
Движение.ДатаЗавершения = ВремяВыгрузкиС;
Движение.АдресНач       = АдресЗагрузки;
Движение.АдресКон       = АдресВыгрузки;
Движение.Состояние      = РезультатВыполнения;
Движение.ИдМаршрута     = ИдМаршрута;
Движение.ТипПеревозки   = ТипПеревозки;
Движение.Стоимость      = СуммаПеревозчика;
Движение.ВидУслуги      = ВидУслуги;
Движение.Контрагент     = Перевозчик;
Движение.Валюта        = ВалютаПеревозчика;
Движения.почта_ЭтапыММП.Записать();

//Оповестить("ИзменениеЗаявки", , ЭтотОбъект);

Если                      РезультатВыполнения                =
Справочники.новаСостоянияЗаявокЭкспедирования.Выполнена Тогда

    ДвижениеГрузаПоЗаявкам(Ссылка);

КонецЕсли;

//Учет груза

//Для Каждого Стр Из Груз Цикл
//
//      НаборЗаписей                =
РегистрыСведений.почта_ГрузЗаявкаЭкспедирования.СоздатьНаборЗаписей();
//      //ЭлементОтбора = НаборЗаписей.Отбор.Добавить();

```



```

//      НаборЗаписей.Отбор.Партия.Установить(Ссылка);
//      НаборЗаписей.Отбор.Наименование.Установить(Стр.Наименование);
//      Запись = Неопределено;
//      НаборЗаписей.Прочитать();
//
//      Если НаборЗаписей.Количество(>) 0 Тогда
//          Запись = НаборЗаписей[0];
//      Иначе
//          Запись = НаборЗаписей.Добавить();
//      КонецЕсли;
//
//          Запись.Наименование = Стр.Наименование;
//          Запись.Вес          = Стр.Вес;
//          Запись.Объем        = Стр.Объем;
//          Запись.Места        = Стр.Места;
//          Запись.ТипКонтейнер = Стр.Контейнер;
//          Запись.НомерКонтейнера= Стр.НомерКонтейнера;
//          Запись.Партия      = Ссылка;
//          НаборЗаписей.Записать();
//
//      КонецЦикла;

```

КонецПроцедуры

Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)

СтандартнаяОбработка = Ложь;

//Если ТипЗнч(ДанныеЗаполнения) =

Тип("ДокументСсылка.новаЗаявкаЭкспедирования") Тогда

// ЗаполнитьЗначенияСвойств(ЭтотОбъект, ДанныеЗаполнения,, "Номер,Дата");

// ТабЧасти = ЭтотОбъект.Метаданные().ТабличныеЧасти;

// Для Каждого ТабЧасть Из ТабЧасти Цикл

```

//      ИмяТЧ = ТабЧасть.Имя,
//      Для Каждого СтрокаТЧ Из ДанныеЗаполнения[ИмяТЧ] Цикл
//          НовСтрокаТЧ = ЭтотОбъект[ИмяТЧ].Добавить();
//          ЗаполнитьЗначенияСвойств(НовСтрокаТЧ, СтрокаТЧ);
//      КонецЦикла;
//      КонецЦикла;
//      ЭтотОбъект.Факт = Истина;
//      ЭтотОбъект.ДокументПланирования = ДанныеЗаполнения;
//КонецЕсли;
КонецПроцедуры

// Возвращает доступные варианты печати документа.
//
// Возвращаемое значение:
// Структура, каждая строка которой соответствует одному из вариантов печати
//
Функция ПолучитьСтруктуруПечатныхФорм() Экспорт

    СтруктураМакетов = Новый Структура;

    СтруктураМакетов.Вставить("ЗаявкаНаПеревозку",    "Заявка на перевозку");
    СтруктураМакетов.Вставить("ЗаявкаЗаказчика",    "Заявка заказчика");

    Возврат СтруктураМакетов;

КонецФункции // ПолучитьСтруктуруПечатныхФорм()

//++
Процедура ДвижениеПоТранспортВЗаявке(Отказ)

    Движения.почтаТранспортНаИсполнении.Записывать = Истина;
    Движение = Движения.почтаТранспортНаИсполнении.Добавить();

    Движение.Транспорт = Транспорт;

```

```

Движение.Водитель                = Водитель;
Если Не КоличествоЧасов Тогда
    Движение.НачалоПериода = НачалоДня(ВремяС);
    Движение.КонецПериода  = КонецДня(ВремяС);
Иначе
    Движение.НачалоПериода = ВремяС;
    Движение.КонецПериода  = ВремяС+КоличествоЧасов*60*60;
Если ПлюсЧасНаПодачу Тогда
    Движение.КонецПериода = Движение.КонецПериода+60*60;
КонецЕсли;
КонецЕсли;
Движение.АдресЗагрузки            = АдресЗагрузки;
Движение.АдресВыгрузки           = АдресВыгрузки;
Движение.Перевозчик              = Перевозчик;
Движение.СуммаЗаказчика          = СуммаЗаказчика;
Движение.СуммаПеревозчика        = СуммаПеревозчика;

```

КонецПроцедуры

//--

```

мЭтоНовый                = Ложь;
мФормаОткрыта           = Ложь;
мОбъектКопирования     = Неопределено;

```

&НаКлиенте

Перем ПредЗначениеКоличествоСтрок;

&НаКлиенте

Перем ПредЗначениеВодитель;

&НаКлиенте

Перем НеПриОткрытии;

&НаКлиенте

Перем СписокТранспортаДляОкраскиСтроки;

&НаКлиенте

Перем СписокТранспортаДляУдаленияИзДС;

&НаКлиенте

Перем НедоступныйТранспорт;

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

```

//+++почта-Нова проверим использование модуля
ОтключитьЭкспедирование =
новаУправлениеПользователямиСервер.ПолучитьЗначениеНастройки("НеИспользоватьМод
улиКонфигурацииЭкспедирование");
Если ОтключитьЭкспедирование = Истина Тогда

    СтандартнаяОбработка = Ложь;
    Отказ = Истина;
    Сообщить("Отключено использование модулей конфигурации
""Экспедирование""");
    Возврат;
КонецЕсли;

Если ЭтаФорма.Параметры.Свойство("УслугаОбъект") Тогда
    УслугаОбъект = ЭтаФорма.Параметры.УслугаОбъект;
Иначе
    УслугаОбъект = Неопределено;
КонецЕсли;

Если ЭтаФорма.Параметры.Свойство("ЗаписыватьПриВыборе") Тогда
    ЗаписыватьПриВыборе = ЭтаФорма.Параметры.ЗаписыватьПриВыборе;
Иначе
    ЗаписыватьПриВыборе = Ложь;
КонецЕсли;

//++
ПериодС =
?(ЗначениеЗаполнено(УслугаОбъект.ВремяС),УслугаОбъект.ВремяС,НачалоДня(ТекущаяДа
та()));

```

```

ПериодПо = ?(ЗначениеЗаполнено(УслугаОбъект.ВремяПо), УслугаОбъект.ВремяПо,
КонецДня(ПериодС));

```

```
//--
```

```
НастроитьЖурналДоступногоТранспорта();
```

```
ЗаполнитьТипыКузоваСервер();
```

```
ЗаполнитьСпособыЗагрузкиСервер();
```

```
Если ЭтаФорма.Параметры.Свойство("ВидТранспорта") Тогда
```

```
    Вид = ЭтаФорма.Параметры.ВидТранспорта;
```

```
КонецЕсли;
```

```
Если ЭтаФорма.Параметры.Свойство("Перевозчик") Тогда
```

```
    Перевозчик = ЭтаФорма.Параметры.Перевозчик;
```

```
КонецЕсли;
```

```
Если ЭтаФорма.Параметры.Свойство("ОбъемКузова") Тогда
```

```
    ОбъемКузова = ЭтаФорма.Параметры.ОбъемКузова;
```

```
КонецЕсли;
```

```
Если ЭтаФорма.Параметры.Свойство("ВысотаКузова") Тогда
```

```
    ВысотаКузова = ЭтаФорма.Параметры.ВысотаКузова;
```

```
КонецЕсли;
```

```
Если ЭтаФорма.Параметры.Свойство("ШиринаКузова") Тогда
```

```
    ШиринаКузова = ЭтаФорма.Параметры.ШиринаКузова;
```

```
КонецЕсли;
```

```
Если ЭтаФорма.Параметры.Свойство("ГлубинаКузова") Тогда
```

```
    ГлубинаКузова = ЭтаФорма.Параметры.ГлубинаКузова;
```

```
КонецЕсли;
```

```
Если ЭтаФорма.Параметры.Свойство("Направление") Тогда
```

```
    Направление = ЭтаФорма.Параметры.Направление;
```

```
КонецЕсли;
```

```
Если ЭтаФорма.Параметры.Свойство("Грузоподъемность") Тогда
```

```
    Грузоподъемность = ЭтаФорма.Параметры.Грузоподъемность;
```

```
КонецЕсли;
```

```
Если ЭтаФорма.Параметры.Свойство("ТипыКузовов") Тогда
```

```
    Для Каждого элТипКузова Из ТипыКузовов Цикл
```

```

        элТипКузова.Пометка = Ложь;
    КонечЦикла;
    Для Каждого элТипКузова Из ЭтаФорма.Параметры.ТипыКузовов Цикл
        НайденыйЭлемент = ТипыКузовов.НайтиПоЗначению(элТипКузова);
        Если НайденыйЭлемент <> Неопределено Тогда
            НайденыйЭлемент.Пометка = Истина;
        КонечЕсли;
    КонечЦикла;
КонечЕсли;
Если ЭтаФорма.Параметры.Свойство("СпособыЗагрузки") Тогда
    Для Каждого элСпособЗагрузки Из СпособыЗагрузки Цикл
        элСпособЗагрузки.Пометка = Ложь;
    КонечЦикла;
    Для Каждого элСпособЗагрузки Из ЭтаФорма.Параметры.СпособыЗагрузки
Цикл
        НайденыйЭлемент
        =
СпособыЗагрузки.НайтиПоЗначению(элСпособЗагрузки);
        Если НайденыйЭлемент <> Неопределено Тогда
            НайденыйЭлемент.Пометка = Истина;
        КонечЕсли;
    КонечЦикла;
КонечЕсли;

    УстановитьПодменюНастроекСМС();

КонечПроцедуры

&НаСервере
Процедура ЗаполнитьТипыКузоваСервер()

    ТипыКузовов.Очистить();

    Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ

```

```

|   новаТипыКузовов.Ссылка,
|   новаТипыКузовов.Наименование
|ИЗ
|   Справочник.новаТипыКузовов КАК новаТипыКузовов
|ГДЕ
|   новаТипыКузовов.ПометкаУдаления = ЛОЖЬ";
ТЗТипыКузова = Запрос.Выполнить().Выгрузить();
Для Каждого Строка Из ТЗТипыКузова Цикл
    ТипыКузовов.Добавить(Строка.Ссылка, Строка.Наименование);
КонецЦикла;

```

КонецПроцедуры

&НаСервере

Процедура ЗаполнитьСпособыЗагрузкиСервер()

```

СпособыЗагрузки.Очистить();

Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
|   новаСпособыЗагрузки.Ссылка,
|   новаСпособыЗагрузки.Наименование
|ИЗ
|   Справочник.новаСпособыЗагрузки КАК новаСпособыЗагрузки
|ГДЕ
|   новаСпособыЗагрузки.ПометкаУдаления = ЛОЖЬ";
ТЗСпособыЗагрузки = Запрос.Выполнить().Выгрузить();
Для Каждого Строка Из ТЗСпособыЗагрузки Цикл
    СпособыЗагрузки.Добавить(Строка.Ссылка, Строка.Наименование);
КонецЦикла;

```

КонецПроцедуры

&НаСервере

Процедура НастроитьЖурналДоступногоТранспорта()

ЖурналДоступныйТранспорт.Параметры.УстановитьЗначениеПараметра("ПериодС",
ПериодС);

ЖурналДоступныйТранспорт.Параметры.УстановитьЗначениеПараметра("ПериодПо",
ПериодПо);

КонецПроцедуры

&НаКлиенте

Процедура ПриОткрытии(Отказ)

ПредЗначениеКоличествоСтрок = 0;

ПредЗначениеВодитель = Неопределено;

//Если Не ЗначениеЗаполнено(УслугаОбъект) Тогда

// Элементы.ФормаВыбрать.Видимость = Ложь;

//КонецЕсли;

ЭлементыОтбораПриИзменении(Элементы.Вид);

ЭлементыОтбораПриИзменении(Элементы.Перевозчик);

ЭлементыОтбораПриИзменении(Элементы.ОбъемКузова);

ЭлементыОтбораПриИзменении(Элементы.ВысотаКузова);

ЭлементыОтбораПриИзменении(Элементы.ШиринаКузова);

ЭлементыОтбораПриИзменении(Элементы.ГлубинаКузова);

ЭлементыОтбораПриИзменении(Элементы.Направление);

ЭлементыОтбораПриИзменении(Элементы.Грузоподъемность);

ЭлементыОтбораПриИзменении(Элементы.ТипыКузовов);

ЭлементыОтбораПриИзменении(Элементы.СпособыЗагрузки);

УчетТранспортаВЗаявках

=

новаУправлениеПользователямиСервер.ПолучитьЗначениеНастройки("УчетДоступностиТранспорта");

ЖурналДоступныйТранспорт.Параметры.УстановитьЗначениеПараметра("УчетДоступностиТранспорта", УчетТранспортаВЗаявках);

КонецПроцедуры

&НаКлиенте

Процедура ЭлементыОтбораПриИзменении(Элемент)

Если Элемент.Имя = "ТипыКузовов" Тогда

ИмяЭлемента = "ТипКузова";

ИначеЕсли Элемент.Имя = "СпособыЗагрузки" Тогда

ИмяЭлемента = "СпособЗагрузки";

Иначе

ИмяЭлемента = Элемент.Имя;

КонецЕсли;

Если ИмяЭлемента = "ТипКузова" Или ИмяЭлемента = "СпособЗагрузки" Тогда

СписокОтбора = Новый СписокЗначений;

Для Каждого ЭлСписка Из ЭтаФорма[Элемент.Имя] Цикл

Если ЭлСписка.Пометка Тогда

СписокОтбора.Добавить(ЭлСписка.Значение);

КонецЕсли;

КонецЦикла;

КонецЕсли;

ЛевоеЗначение = Неопределено;

Для

Каждого

Поле

Из

ЖурналДоступныйТранспорт.Отбор.ДоступныеПоляОтбора.Элементы Цикл

Если Строка(Поле.Поле) = ИмяЭлемента Тогда

ЛевоеЗначение = Поле.Поле;

Прервать;

КонецЕсли;

КонецЦикла;

Если ЛевоеЗначение = Неопределено Тогда

Возврат;

КонецЕсли;

Эл = Неопределено;

Для Каждого ЭлОтбора Из ЖурналДоступныйТранспорт.Отбор.Элементы Цикл

Если ЭлОтбора.ЛевоеЗначение = ЛевоеЗначение Тогда

Эл = ЭлОтбора;

Прервать;

КонецЕсли;

КонецЦикла;

Если ИмяЭлемента = "ТипКузова" Или ИмяЭлемента = "СпособЗагрузки" Тогда

ЗначениеЭлементаЗаполнено = (СписокОтбора.Количество() <> 0);

Иначе

ЗначениеЭлементаЗаполнено = ЗначениеЗаполнено(ЭтаФорма[Элемент.Имя]);

КонецЕсли;

Если Не ЗначениеЭлементаЗаполнено Тогда

Если Эл <> Неопределено Тогда

Эл.Использование = Ложь;

КонецЕсли;

Возврат;

КонецЕсли;

Если Эл = Неопределено Тогда

Эл

=

ЖурналДоступныйТранспорт.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));

КонецЕсли;

Эл.ЛевоеЗначение = ЛевоеЗначение;

Если ИмяЭлемента = "ТипКузова" Или ИмяЭлемента = "СпособЗагрузки" Тогда

Эл.ВидСравнения = ВидСравненияКомпоновкиДанных.ВСписке;

Эл.ПравоеЗначение = СписокОтбора;

Иначе

```

Эл.ВидСравнения = ?(ТипЗнч(ЭтаФорма[Элемент.Имя])=Тип("Число"),
ВидСравненияКомпоновкиДанных.БольшеИлиРавно,
ВидСравненияКомпоновкиДанных.Равно);

```

```

Эл.ПравоеЗначение = ЭтаФорма[Элемент.Имя];

```

```

КонецЕсли;

```

```

Эл.Использование = Истина;

```

```

////20110722 Абдрафикова С.А. Составления списка транспорта, используемого в
заявках

```

```

////++

```

```

//Если НеПриОткрытии Тогда

```

```

//    ПолучитьСписокТранспорта(),

```

```

//КонецЕсли;

```

```

////--

```

```

КонецПроцедуры

```

```

&НаКлиенте

```

```

Процедура ПериодПриИзменении(Элемент)

```

```

    ЖурналДоступныйТранспорт.Параметры.УстановитьЗначениеПараметра("ПериодС",
ПериодС);

```

```

КонецПроцедуры

```

```

&НаКлиенте

```

```

Процедура ПериодПоПриИзменении(Элемент)

```

```

    ЖурналДоступныйТранспорт.Параметры.УстановитьЗначениеПараметра("ПериодПо"
, ПериодПо);

```

```

КонецПроцедуры

```

```

// Осуществляет подстановку значения выбранного транспорта из формы подбора в объект
посреднической услуги

```

```

//

```

```

&НаКлиенте

```

Процедура ПодобратьТранспорт()

```
//Если Не ЗначениеЗаполнено(УслугаОбъект) Тогда
```

```
//    Возврат;
```

```
//КонецЕсли;
```

```
vrВыделенныеСтроки = Элементы.ЖурналДоступныйТранспорт.ВыделенныеСтроки;
```

```
ТекущиеДанные = Элементы.ЖурналДоступныйТранспорт.ТекущиеДанные;
```

```
Если ТекущиеДанные = Неопределено Тогда
```

```
    Возврат;
```

```
КонецЕсли;
```

```
Если vrВыделенныеСтроки.Количество() > 1 Тогда
```

```
    Ответ = Вопрос("Будет подобран транспорт: "+ТекущиеДанные.Транспорт  
                    +" и водитель: "+ТекущиеДанные.Водитель+").
```

```
Продолжить?",РежимДиалогаВопрос.ДаНетОтмена);
```

```
    Если Ответ <> КодВозвратаДиалога.Да Тогда
```

```
        Возврат;
```

```
    КонецЕсли;
```

```
КонецЕсли;
```

```
П = Новый Структура();
```

```
П.Вставить("ЗаявкаЭкспедирования", УслугаОбъект);
```

```
П.Вставить("Транспорт", ТекущиеДанные.Транспорт);
```

```
П.Вставить("Водитель", ТекущиеДанные.Водитель);
```

```
П.Вставить("Перевозчик", ТекущиеДанные.Перевозчик);
```

```
//П.Вставить("Договор", ТекущиеДанные.Договор);
```

```
П.Вставить("ВидТранспорта", ТекущиеДанные.Вид);
```

```
Оповестить("новаПодобранТранспорт", П);
```

```
ЭтаФорма.Закреть();
```

КонецПроцедуры

&НаКлиенте

Процедура ЖурналДоступныйТранспортВыбор(Элемент, ВыбраннаяСтрока, Поле,
СтандартнаяОбработка)

СтандартнаяОбработка = Ложь;

ПодобратьТранспорт();

КонецПроцедуры

&НаКлиенте

Процедура Выбрать(Команда)

ПодобратьТранспорт();

КонецПроцедуры

&НаКлиенте

Процедура НастроитьОповещениеПоСМС(Команда)

КолонкиФормы = Новый Массив();

Для Каждого Элемент Из
ЭтаФорма.ЖурналДоступныйТранспорт.Группировка.ДоступныеПоляПолейГруппировок.Эл
ементы Цикл

КолонкиФормы.Добавить(Строка(Элемент.Поле));

КонецЦикла;

КолонкиФормы.Добавить("НомерЗаявки");

КолонкиФормы.Добавить("ДатаЗаявки");

КолонкиФормы.Добавить("АдресЗагрузки");

КолонкиФормы.Добавить("АдресВыгрузки");

КолонкиФормы.Добавить("ВремяС");

КолонкиФормы.Добавить("Заказчик");

КолонкиФормы.Добавить("Направление");

ПараметрыНастройки = Новый Структура();

ПараметрыНастройки.Вставить("РеквизитыФормы", КолонкиФормы);

ПараметрыНастройки.Вставить("ФормаВладелец",

"Обработка.новаДоступныйТранспорт.Форма.ФормаПодбораТранспорта");

ОткрытьФорму("Справочник.новаНастройкиСМС.Форма.ФормаСписка",

ПараметрыНастройки);

КонецПроцедуры

&НаКлиенте

Процедура ОбработкаОповещения(ИмяСобытия, Параметр, Источник)

```
//Если ИмяСобытия = "новаИзмененШаблонСМСПодборТранспорта" Тогда
//    Если (Параметр.Свойство("ТЗСоответствиеПолей")) Тогда
//        ТЗСоответствиеПолей.Очистить();
//        Для Каждого СтрокаТЗ Из Параметр.ТЗСоответствиеПолей Цикл
//            НовСтрока = ТЗСоответствиеПолей.Добавить();
//            НовСтрока.РеквизитШаблона    = СтрокаТЗ.РеквизитШаблона;
//            НовСтрока.РеквизитФормы    = СтрокаТЗ.РеквизитФормы;
//        КонецЦикла;
//    КонецЕсли;
//    Если Параметр.Свойство("ШаблонСМС") Тогда
//        ШаблонСМС = Параметр.ШаблонСМС;
//    КонецЕсли;
//КонецЕсли;
```

```
Если ИмяСобытия = "новаЗакрытаФормаЗаявки" Тогда
```

```
    Если новаЗакрытаФормаЗаявкиНаСервере(УслугаОбъект) = Параметр Тогда
```

```
        ЭтаФорма.Закрыть();
```

```
    КонецЕсли;
```

```
КонецЕсли;
```

```
Если ИмяСобытия = "почтаИзмененТипКузова" Тогда
```

```
    ЗаполнитьТипыКузоваСервер();
```

```
КонецЕсли;
```

```
Если ИмяСобытия = "почтаИзмененСпособЗагрузки" Тогда
```

```
    ЗаполнитьСпособыЗагрузкиСервер();
```

```
КонецЕсли;
```

КонецПроцедуры

&НаКлиенте

Процедура ОповеститьПоСМС(Команда)

ЗаполнитьМассивВыделенныхСтрок();

СписокВодителей = ПолучитьСписокВодителей();

ТекстСМС = ПолучитьТекстСМС();

Если ЗначениеЗаполнено(СписокВодителей) Тогда

 ПараметрыРассылки = Новый Структура();

 ПараметрыРассылки.Вставить("СписокВодителей",СписокВодителей);

 ПараметрыРассылки.Вставить("Основание",УслугаОбъект);

 ПараметрыРассылки.Вставить("ТекстВТабл",ТекстСМС);

 ОткрытьФорму("ОбщаяФорма.новаФормаРассылкиСМС",

ПараметрыРассылки);

Иначе

 Сообщить("Не заполнен список водителей!");

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ЗаполнитьМассивВыделенныхСтрок()

 ВыделенныеСтроки.Очистить();

 Если Элементы.ЖурналДоступныйТранспорт.ВыделенныеСтроки.Количество() > 1

Тогда

 Для Каждого ЭлМассива Из

 Элементы.ЖурналДоступныйТранспорт.ВыделенныеСтроки Цикл

 Водитель =

 Элементы.ЖурналДоступныйТранспорт.ДанныеСтроки(ЭлМассива).Водитель;

 Если ЗначениеЗаполнено(Водитель) Тогда

 НовСтрока = ВыделенныеСтроки.Добавить();

 НовСтрока.Водитель = Водитель;

 КонецЕсли;

КонецЦикла;

Иначе

 ь=0;

Пока Истина Цикл

```

        ь=1+ь;
        Попытка
            Водитель
Элементы.ЖурналДоступныйТранспорт.ДанныеСтроки(ь).Водитель;
            Если ЗначениеЗаполнено(Водитель) Тогда
                НовСтрока = ВыделенныеСтроки.Добавить();
                НовСтрока.Водитель = Водитель;
            КонецЕсли;
        Исключение
            Прервать;
        КонецПопытки;
    КонецЦикла;
КонецЕсли;
КонецПроцедуры

&НаСервере
Функция новаЗакрытаФормаЗаявкиНаСервере(УслугаОбъект)
    Возврат УслугаОбъект.номер;
КонецФункции

&НаСервере
Функция ПолучитьСписокВодителей(НастройкаСМС = Неопределено)

    СписокВодителей = Новый Массив;

    Для Каждого СтрокаТЗ Из ВыделенныеСтроки Цикл
        СписокВодителей.Добавить(СтрокаТЗ.Водитель);
    КонецЦикла;

    Возврат СписокВодителей;

КонецФункции

```


&НаСервере

Функция ПолучитьТекстСМС(НастройкаСМС = Неопределено)

```

    Если НастройкаСМС = Неопределено Тогда
        НастройкаПоУмолчанию =
        новаСлужбаКороткихСообщений.ПолучитьНастройкуПоУмолчанию("Обработка.новаДосту
        пныйТранспорт.Форма.ФормаПодбораТранспорта");
    Иначе
        НастройкаПоУмолчанию = НастройкаСМС;
    КонецЕсли;

    Если Не ЗначениеЗаполнено(НастройкаПоУмолчанию) Тогда
        Возврат Неопределено;
    КонецЕсли;

    ТЗВыделенныеСтроки = ДанныеФормыВЗначение(ВыделенныеСтроки,
    Тип("ТаблицаЗначений"));
    ИменаКолонок = Новый Массив();
    Для Каждого КолонкаТЗ Из ТЗВыделенныеСтроки.Колонки Цикл
        ИменаКолонок.Добавить(КолонкаТЗ.Имя);
    КонецЦикла;
    Для Каждого СтрокаТЗ Из ВыделенныеСтроки Цикл
        Переформы = ПолучитьСтруктуруРеквизитов(СтрокаТЗ,
    ИменаКолонок);
        ТекстСМС =
        новаСлужбаКороткихСообщений.ПолучитьТекстСМС(НастройкаПоУмолчанию,
    Переформы);
        Возврат ТекстСМС;
    КонецЦикла;

    КонецФункции

```

&НаСервере

Функция ПолучитьСтруктуруРеквизитов(СтрокаТЗ, ИменаКолонок)

```

ПеремФормы = Новый Структура();
ПеремФормы.Вставить("НомерЗаявки",      УслугаОбъект.Номер);
ПеремФормы.Вставить("ДатаЗаявки",      УслугаОбъект.Дата);
ПеремФормы.Вставить("АдресЗагрузки",   УслугаОбъект.АдресЗагрузки);
ПеремФормы.Вставить("АдресВыгрузки",   УслугаОбъект.АдресВыгрузки);
ПеремФормы.Вставить("ВремяС",        УслугаОбъект.ВремяС);
ПеремФормы.Вставить("Заказчик",        УслугаОбъект.Заказчик);
ПеремФормы.Вставить("Направление",     УслугаОбъект.Направление);

```

Для Каждого ИмяКолонки Из ИменаКолонок Цикл

```

    ПеремФормы.Вставить(ИмяКолонки,      СтрокаТЗ[ИмяКолонки]);
КонецЦикла;

```

Возврат ПеремФормы;

КонецФункции

&НаСервере

Процедура УстановитьПодменюНастроекСМС()

```

//
//Запрос = Новый Запрос();
//Запрос.Текст = "ВЫБРАТЬ
//                |   новаНастройкиСМС.Ссылка,
//                |   новаНастройкиСМС.Наименование,
//                |   новаНастройкиСМС.ФормаВладелец,
//                |   новаНастройкиСМС.ШаблонСМС,
//                |   новаНастройкиСМС.ПоУмолчанию КАК ПоУмолчанию,
//                |   новаНастройкиСМС.Код
//                |ИЗ
//                |   Справочник.новаНастройкиСМС           КАК
новаНастройкиСМС
//                |ГДЕ

```

```

//          |      новаНастройкиСМС.ФормаВладелец = &ФормаВладелец
//          |
//          |УПОРЯДОЧИТЬ ПО
//          |      ПоУмолчанию";
//Запрос.УстановитьПараметр("ФормаВладелец",
"Обработка.новаДоступныйТранспорт.Форма.ФормаПодбораТранспорта");
//Результат = Запрос.Выполнить().Выгрузить();
//Для Каждого СтрокаТЗ Из Результат Цикл
//
//      Имя                                     =
?(ЗначениеЗаполнено(СтрокаТЗ.Наименование),СтрокаТЗ.Наименование,"Элемент") +
СтрокаТЗ.Код,
//      Имя = СтрЗаменить(Имя, " ", "");
//      Текст = СтрокаТЗ.Наименование;
//
//      Кнопка      =      Элементы.Добавить(Имя,      Тип("КнопкаФормы"),
Элементы.ЖурналДоступныйТранспортВсеНастройкиСМС);
//      Кнопка.Заголовок = Текст;
//
//      новаяКоманда = Команды.Добавить(Имя);
//      новаяКоманда.Действие = "ОповеститьПоСМСВсе";
//      Кнопка.ИмяКоманды = Имя;
//КонецЦикла;

```

КонецПроцедуры

&НаКлиенте

Процедура ОповеститьПоСМСВсе(Команда)

ЗаполнитьМассивВыделенныхСтрок();

НайденнаяНастройка = ПолучитьНастройкуДляКоманды(Команда Имя);

СписокВодителей = ПолучитьСписокВодителейДляНастройки(НайденнаяНастройка);

Если ЗначениеЗаполнено(СписокВодителей) Тогда

ПараметрыРассылки = Новый Структура();

ПараметрыРассылки.Вставить("СписокВодителей",СписокВодителей);

```

    ПараметрыРассылки.Вставить("Основание",УслугаОбъект);
    ПараметрыРассылки.Вставить("НастройкаСМС",НайденнаяНастройка);
    ОткрытьФорму("ОбщаяФорма.новаФормаРассылкиСМС",
ПараметрыРассылки);
    Иначе
        Сообщить("Не заполнено значение настройки СМС");
    КонецЕсли;
КонецПроцедуры

```

&НаСервере

Функция ПолучитьНастройкуДляКоманды(ИмяКоманды)

```

    Запрос = Новый Запрос();

```

```

    Запрос.Текст = "ВЫБРАТЬ

```

```

        |     новаНастройкиСМС.Ссылка,

```

```

        |     новаНастройкиСМС.Наименование,

```

```

        |     новаНастройкиСМС.ФормаВладелец,

```

```

        |     новаНастройкиСМС.ШаблонСМС,

```

```

        |     новаНастройкиСМС.ПоУмолчанию КАК ПоУмолчанию,

```

```

        |     новаНастройкиСМС.Код

```

```

    ИЗ

```

```

        |     Справочник.новаНастройкиСМС КАК новаНастройкиСМС

```

```

    ГДЕ

```

```

        |     новаНастройкиСМС.ФормаВладелец = &ФормаВладелец

```

```

        |

```

```

    УПОРЯДОЧИТЬ ПО

```

```

        |     ПоУмолчанию";

```

```

    Запрос.УстановитьПараметр("ФормаВладелец",

```

```

"Обработка.новаДоступныйТранспорт.Форма.ФормаПодбораТранспорта");

```

```

    Результат = Запрос.Выполнить().Выгрузить();

```

```

    НайденнаяНастройка = Неопределено;

```

```

    Для Каждого СтрокаТЗ Из Результат Цикл

```

```

        Имя = СтрокаТЗ.Наименование + СтрокаТЗ.Код;

```

```

        Имя = СтрЗаменить(Имя, " ", "");

```

```

Если ИмяКоманды = Имя Тогда
    НайденнаяНастройка = СтрокаТЗ.Ссылка;
    Прервать;
КонецЕсли;

```

```

КонецЦикла;

```

```

    Возврат НайденнаяНастройка;
КонецФункции

```

```

&НаСервере

```

```

Функция ПолучитьСписокВодителейДляНастройки(НастройкаСМС)

```

```

    Если Не ЗначениеЗаполнено(НастройкаСМС) Тогда

```

```

        Возврат Неопределено;

```

```

    КонецЕсли;

```

```

    спВодители = ПолучитьСписокВодителей(НастройкаСМС);

```

```

    Возврат спВодители;

```

```

КонецФункции

```

```

&НаКлиенте

```

```

Процедура ЖурналДоступныйТранспортТранспортНажатие(Элемент,
СтандартнаяОбработка)

```

```

    СтандартнаяОбработка = Ложь;

```

```

    // Вставить содержимое обработчика.

```

```

КонецПроцедуры

```

```

&НаКлиенте

```

```

Процедура ОчиститьОтбор(Команда)

```

```

    ОчиститьОтборСервер();

```

```

КонецПроцедуры

```

```

&НаСервере

```

Процедура ОчиститьОтборСервер()

```

Вид = Справочники.новаВидыТранспорта.ПустаяСсылка();
Перевозчик = Справочники.Контрагенты.ПустаяСсылка();
ОбъемКузова = 0;
ВысотаКузова = 0;
ШиринаКузова = 0;
ГлубинаКузова = 0;
Направление = Справочники.новаНаправления.ПустаяСсылка();
Грузоподъемность = 0;
Для Каждого Эл Из ТипыКузовов Цикл
    Эл.Пометка = Ложь;
КонецЦикла;
Для Каждого Эл Из СпособыЗагрузки Цикл
    Эл.Пометка = Ложь;
КонецЦикла;
Для Каждого ЭлОтбора Из ЖурналДоступныйТранспорт.Отбор.Элементы Цикл
    Если НРег(ЭлОтбора.ЛевоеЗначение) <> "удалить" И
    НРег(ЭлОтбора.ЛевоеЗначение) <> "недоступный" Тогда
        ЭлОтбора.Использование = Ложь;
    КонецЕсли;
КонецЦикла;

КонецПроцедуры

```