

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерная техника

«Допущен к защите»

Заведующий кафедрой _____

Куралбаев З. К. профессор
(Ф.И.О., ученая степень, звание)

_____ « _____ » _____ 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка функциональной информационной системы

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Турсун Александр Димуриевич ВТУ-11-1
(Фамилия и инициалы) группа

Научный руководитель Мусаева Т.Т. старший преподаватель
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Ермешина З. Д.
(Фамилия и инициалы, ученая степень, звание)
Ермешина « 24 » мая 20 14 г.
(подпись)

по безопасности жизнедеятельности:

Беликеева А. С.
(Фамилия и инициалы, ученая степень, звание)
Беликеева « 2 » июня 20 14 г.
(подпись)

по применению вычислительной техники:

_____ (Фамилия и инициалы, ученая степень, звание)
_____ « _____ » _____ 20__ г.
(подпись)

Нормоконтролер: Турсун Д. М.
(Фамилия и инициалы, ученая степень, звание)

Турсун « _____ » _____ 20__ г.
(подпись)

Рецензент: Мурзабеков М. Ж. д.т.н., доцент факультета
(Фамилия и инициалы, ученая степень, звание)

М. Мурзабеков « _____ » _____ 20__ г.
(подпись)

Алматы 2014 г.

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Высшейшей школы и программное обеспечение
Специальность _____
Кафедра Компьютерные технологии

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Мернов Александр Дмитриевич
(фамилия, имя, отчество)

Тема проекта Разработка французской информационной системы

утверждена приказом ректора № 115 от «24» сентября 20__ г.

Срок сдачи законченной работы «__» _____ 20__ г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Разработка французской информационной системы на территории г. Премонте в департаменте Шерурр-и-К.
Цель проекта создание системы для организации рабочего места

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

Дипломный проект "Разработка французской информационной системы" преследует своей целью разработать систему для

Перечень графического материала (с точным указанием обязательных чертежей)

- Детальная структура колонны
- График себестоимости изготовления
- Опись строительства вагона
- Опись деталей вагона
- Опись строения элементов
- Опись деталей по вагону
- Опись строения элементов
- Опись деталей деталей об отделке в вагоне
- Опись отделки цеха
- Опись отделки по вагону
- Опись отделки по вагону
- Опись отделки деталей.

Рекомендуемая основная литература

- Базоцев З.Т. Организация работы бригады в организации
- Беломестова А.С. Методические указания к выполнению работы
- Беломестова А.С. Базоцев З.Т. Организация работы бригады в организации
- Беломестова А.С. Организация работы бригады в организации

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Беломестова	Беломестова З.Д.	15.04-27.08.14	Беломестова
Б.И.И.	Беломестова А.С.	20.03-2.06.14	Беломестова

Аннотация

Данный дипломный проект «Фармацевтическая информационная система» разработан для компании «Интерфарма-К», цель проекта разработать универсальную систему для быстрого получения своевременной информации о закупках и продажах товара, для распечатки документов на отгрузку, перевозку товара, для хранения электронных данных на необходимые поставки. Главной проблемой для компании стояла задача сократить время работы сотрудников и повысить доход компании путем консолидации лекарственных средств, а так же упорядочивания информации.

Андатпа

Айтылмыш дипломдык жоба фармацевтикалык акпараттык жуйе компания «Интерфарма-К» үшін эзірле жобанын максатынын эзерле эмбебаб жуйе уактылы акпараттын аспа жалап ал ушин туралы сату және тауардын тасыламына. Басты мәселемен компания ушин кыскарту максат кызметкердин жумысынын уакытын және жогарылат компаниянын тобасынын жолымен дерилик акы пулдин ныгаюынын турды, ал олай гой акпаратын ретиуленин.

Содержание

Введение.....	8
1. Принципы разработки	10
1.1 Платформа 1С Предприятие	10
1.2 Рекомендации для разработки конфигурации	13
1.2.1 Базовый принцип программирования управляемой формы в 1С .	15
1.2.2 Групповая разработка прикладных решений.....	21
1.2.3 Система прав доступа.....	26
1.2.4 Система типов	29
1.2.5 Управление торговлей.....	34
1.2.6 Механизм запросов	40
2 Инструменты разработки	45
2.1 Установка и настройка 1С Предприятия 8.....	45
2.2 Пользователь USR1CV81 или USR1CV82 и его права	45
2.3 Регистрация сервера 1С Предприятия как сервиса Windows.....	47
3 Этапы разработки конфигурации.....	52
3.1 Описание автоматизируемых процессов.....	52
3.2 Создание логической модели проектируемой системы.....	53
3.3 Разработка архитектуры	54
3.4 Проектирование интерактивных операций	54
3.5 Подготовка справки	55
3.6 Управление проектом и изменениями	56
3.7 Работа с ошибками.....	57
3.8 Модуль «Общий»	57
3.9 Модуль «Логистика»	106
3.10 Модуль «Бизнес процессы»	108
4 Технично–экономическое обоснование.....	111
4.1 Цель проекта.....	111
4.2 Трудовые ресурсы, используемые в работе	111
4.3 Оборудование, используемое в работе	111
4.4 Программное обеспечение, используемое в работе	112
4.5 Сроки реализации проекта	112
4.6 Расчет затрат и стоимости работ по реализации проекта.....	113
4.6.1 Расчет фонда оплаты труда.....	114
4.6.2 Расчет затрат по социальному налогу.....	116
4.6.3 Расчет амортизационных отчислений.....	116
4.6.4 Расчет затрат на электроэнергию	117
4.6.5 Расчет накладных расходов	118
4.6.6 Суммарные затраты на реализацию проекта	118
4.6.7 Цена реализации проекта	119
5 Безопасность жизнедеятельности.....	120
5.1 Анализ условий труда при разработке мобильного приложения ...	120
5.2 Расчет естественного освещения.....	123

Заключение	127
Список используемой литературы	128
Приложение А	129
Приложение Б.....	135

Введение

Актуальность темы. Компания Интерфарма–К создана в целях повышения устойчивости и конкурентоспособности фармацевтической отрасли Республики Казахстан, развития фармацевтической промышленности путем консолидации государственных закупок лекарственных средств.

Предметом деятельности Компании являются:

- Организация открытых торгов по закупке лекарственных средств в рамках гарантированного объема бесплатной медицинской помощи;
- Организация хранения лекарственных средств в соответствии с требованиями надлежащих дистрибьюторских практик и Законодательства Республики Казахстан;
- Организация процессов логистики лекарственных средств в государственные медицинские организации;
- Создание информационной системы для интеграции процессов логистики Единого дистрибьютора, заказчика и поставщиков, а также для получения актуальной информации об обороте, товарных остатков.

В феврале 2009 года Правительство Республики Казахстан приняло решение о создании компании «Интерфарма–К» в структуре АО «Фонд национального благосостояния «Самрук–Казына» (постановление Правительства РК №134 от 11 февраля 2009 года).

Постановлением Правительства Республики Казахстан №516 от 25 мая 2013 года права владения и пользования стопроцентной долей участия в товариществе с ограниченной ответственностью «Интерфарма–К» были переданы Министерству здравоохранения Республики Казахстан.

Система Единого дистрибьютора создана с целью обеспечения лекарственными средствами населения в рамках гарантированного объема бесплатной медицинской помощи, повышения устойчивости и конкурентоспособности фармацевтической отрасли Республики Казахстан, развития фармацевтической промышленности путем консолидации государственных закупок лекарственных средств.

Основные направления деятельности Единого дистрибьютора:

- Организация открытых торгов по закупке лекарственных средств в рамках гарантированного объема бесплатной медицинской помощи.
- Организация хранения лекарственных средств в соответствии с требованиями надлежащих дистрибьюторских практик и Законодательства Республики Казахстан.
- Организация процессов логистики лекарственных государственных медицинские организации.
- Создание информационной системы для интеграции процессов логистики Единого дистрибьютора, заказчика и поставщиков, а также для получения актуальной информации об обороте, товарных остатков.

Практика показывает, что с учетом вышеизложенных и многих других особенностей функционирования компаний 1С Предприятие как нельзя лучше подходит для решения широкого спектра задач. Благодаря приложениям 1С значительно упростилась процедура ведения учета для предоставления информации налоговым органам и использования ее внутренними службами организации. Также повысилась оперативность подготовки управленческих отчетов и расчета данных, на основании которых вычисляются финансовые показатели.

Система 1С Предприятие снимает проблему конфликта целей учета. Увидеть это наглядно можно на примере сведений, относящихся к ключевым финансовым показателям платежеспособности и рентабельности организации. При некоторой схожести они требуют абсолютно разных подходов к анализу, первый устанавливается оценкой активов компании с их ликвидностью, второй проработкой объемов и структур доходов и расходов, и соотношения данных показателей. В рамках 1С Предприятия организация единого информационного поля бухгалтерских данных и их использование в соответствии с потребностями пользователей наиболее эффективны.

В сложное кризисное и посткризисное время актуальная информация, необходимая для принятия обдуманных решений, способна помочь сохранению и укреплению положения организации на рынке. Система 1С Предприятие 8 за счет применения современных программных разработок и интернет-технологий позволяет консолидировать все нужные данные (в том числе получаемые от сотрудников, находящихся в командировках, и присланные из обособленных и/или отдаленных подразделений). Гибкость приложений дает возможность значительному количеству пользователей вести одновременную работу без снижения производительности программ и ограничения доступа к информации.

1 Принципы разработки

1.1 Платформа 1С Предприятие

Система программ «1С Предприятие 8» включает в себя платформу и прикладные решения, для автоматизации деятельности организаций и частных лиц. Сама платформа не является программным продуктом для использования конечными пользователями, которые обычно работают с одним из многих прикладных решений (конфигураций), разработанных на данной платформе. Такой подход позволяет автоматизировать различные виды деятельности, используя единую технологическую платформу «1С Предприятие 8».

Гибкость платформы позволяет применять 1С Предприятие 8 в самых разнообразных областях:

- автоматизация производственных и торговых предприятий, бюджетных и финансовых организаций, предприятий сферы обслуживания и т.д.;
- поддержка оперативного управления предприятием;
- автоматизация организационной и хозяйственной деятельности;
- ведение бухгалтерского учета с несколькими планами счетов и произвольными измерениями учета, регламентированная отчетность;
- широкие возможности для управленческого учета и построения аналитической отчетности, поддержка много валютного учета;
- решение задач планирования, бюджетирования и финансового анализа;
- расчет зарплаты и управление персоналом;
- другие области применения.

Система 1С Предприятие 8 имеет в своей основе ряд механизмов, определяющих концепцию создания прикладных решений. Наличие этих механизмов позволяет максимально соотнести технологические возможности с бизнес-схемой разработки и внедрения прикладных решений.

В качестве ключевых моментов можно выделить изоляцию разработчика от технологических подробностей, алгоритмическое программирование только бизнес-логики приложения, использование собственной модели базы данных и масштабируемость прикладных решений без их доработки. Состав прикладных механизмов 1С Предприятия ориентирован на решение задач автоматизации учета и управления предприятием. Использование проблемно-ориентированных объектов позволяет разработчику решать самый широкий круг задач складского, бухгалтерского, управленческого учета, расчета зарплаты, анализа данных и управления на уровне бизнес-процессов. В 1С Предприятии 8 реализован современный дизайн интерфейса и повышена комфортность работы пользователей при работе с системой в течение длительного времени.

Технологическая платформа обеспечивает различные варианты работы прикладного решения: от персонального однопользовательского, до работы в масштабах больших рабочих групп и предприятий. Ключевым моментом

масштабируемости является то, что повышение производительности достигается средствами платформы, и прикладные решения не требуют доработки при увеличении количества одновременно работающих пользователей.

Система 1С Предприятие 8 является открытой системой. Предоставляется возможность для интеграции практически с любыми внешними программами и оборудованием на основе общепризнанных открытых стандартов.

В системе 1С Предприятие 8 имеется целый набор средств, с помощью которых можно создавать, обрабатывать и обмениваться данными различных форматов, осуществлять доступ ко всем объектам системы 1С Предприятие 8, реализующим ее функциональные возможности, поддерживать различные протоколы обмена, поддерживать стандарты взаимодействия с другими подсистемами, создавать собственные интернет-решения.

Механизм Web-сервисов позволяет создавать Web-сервисы в конфигурации 1С Предприятия 8, а также взаимодействовать в конфигурации 1С Предприятия 8 с веб-сервисами, опубликованными сторонними поставщиками. Система прав доступа позволяет разрешать доступ пользователей только к тем данным, которые необходимы им для выполнения определенных функций в прикладном решении. Разработчик может создавать наборы прав, соответствующие должностям пользователей или виду деятельности. Например, могут быть введены такие наборы прав, как «Главный бухгалтер», «Кладовщик», «Менеджер», «Начальник отдела» и т.д.

Механизмы обмена данными, реализованные в технологической платформе 1С Предприятие 8, позволяют создавать территориально распределенные информационные системы как на основе информационных баз 1С Предприятия 8, так и с участием других информационных систем. Например, можно организовать работу главного офиса, филиалов и складов предприятия в единой информационной базе, или обеспечить взаимодействие информационной базы 1С Предприятия 8 с существующей базой данных MySQL.

Web-расширение, хотя и входит в состав технологической платформы, поставляется в виде отдельного программного продукта и позволяет встраивать доступ к данным 1С Предприятия в существующие Web-сайты и Web-приложения, а так же создавать готовые Web-приложения, использующие информационную базу 1С Предприятия 8.

Все тексты конфигурации и базы данных хранятся в формате UNICODE. Это значит, что любая текстовая информация может включать одновременно символы различных языков. Для основных европейских языков поддерживаются национальные представления дат, чисел, а также порядок сортировки текстов. Для конкретной информационной базы предоставляется возможность дополнительной настройки представления чисел, дат, времени.

В 1С Предприятии 8 предусмотрена возможность создания многоязычных прикладных решений – разработчик может создавать

конфигурацию сразу на нескольких языках, а каждый пользователь – выбирать свой язык.

Мощные средства формирования отчетов и печатных форм обеспечивают широкие возможности оформления и интерактивной работы. Интеллектуальное построение иерархических, многомерных и кросс-отчетов, получение любых аналитических данных с произвольной настройкой пользователем без изменения прикладного решения. Группировки и расшифровки в отчетах, детализация и агрегирование информации, сводные таблицы для анализа многомерных данных, динамическое изменение структуры отчета, различные типы диаграмм для графического представления экономической информации.

Современный дизайн интерфейса обеспечивает легкость освоения для начинающих и высокую скорость работы для опытных пользователей. Значительное ускорение массового ввода информации благодаря функции «ввод по строке» и эффективному использованию клавиатуры. Облегчение работы неподготовленных пользователей, быстрое освоение системы, удобные средства работы с большими динамическими списками. Механизм полнотекстового поиска поддерживает указание поисковых операторов (И, ИЛИ, НЕ, РЯДОМ и др.). Возможно выборочное включение прикладных объектов и реквизитов в полнотекстовый поиск.

Администратору прикладного решения предоставляются широкие возможности для управления работой пользователей и контроля действий, которые они выполняют. Также система предоставляет развитые механизмы обновления прикладного решения с использованием различных протоколов обмена данными, в том числе и через Интернет.

Технологическая платформа 1С Предприятие содержит средство разработки, с помощью которого создаются новые или изменяются существующие прикладные решения. Это средство разработки называется «конфигуратор». Так как он включен в стандартную поставку 1С Предприятия 8, то пользователь может самостоятельно разработать или модифицировать прикладное решение (адаптировать его под себя), возможно, с привлечением сторонних специалистов.

Опыт внедрения прикладных решений на платформе 1С Предприятие 8 показывает, что система позволяет решать задачи различной степени сложности. В то же время, внедрение большой информационной системы предъявляет повышенные требования по сравнению с небольшим или средним внедрением. Информационная система масштаба предприятия должна обеспечивать приемлемую производительность в условиях одновременной и интенсивной работы большого количества пользователей, которые используют одни и те же информационные и аппаратные ресурсы в конкурентном режиме.

«1С Корпоративный инструментальный пакет 8» (1С КИП) предназначен для повышения производительности, масштабируемости и надежности информационных систем на платформе 1С Предприятие 8 за счет обнаружения и автоматического анализа возможных технических проблем на любых стадиях внедрения (в том числе на стадии проектирования), получения объективной

информации о производительности системы, получения полной технической информации о проблемах производительности, имеющихся в системе, с целью дальнейшей оптимизации ее кода. Корпоративный инструментальный пакет может использоваться как самостоятельно (например, для оценки применимости какой-либо типовой конфигурации для решения задач клиента) так и в качестве дополнительного инструмента, позволяющего провести анализ «узких мест» и повысить производительность и масштабируемость внедряемой или уже внедренной системы. Корпоративный инструментальный пакет будет наиболее полезен при внедрении относительно крупных корпоративных информационных систем, характеризующихся большой нагрузкой на систему и большим количеством рабочих мест.

1.2 Рекомендации для разработки конфигурации

Адаптация тиражной конфигурации под требования конкретного пользователя должна быть продумана с точки зрения последующих обновлений. Существующий в платформе 1С Предприятие 8 механизм поддержки значительно упрощает данный процесс, но в случае внесения в конфигурацию поставщика достаточно серьезных изменений, интеграция обновлений, содержащихся в новой версии поставщика, в модифицированную конфигурацию требует ручной работы. Приведенный ниже список рекомендаций получен на основе анализа использования данного механизма и призван упростить решение этой задачи. Здесь дается сводный перечень рекомендаций:

- Не рекомендуется снимать объекты с поддержки. Нормальным правилом для объекта является «Редактируется с сохранением поддержки». Полное снятие с поддержки имеет смысл, только если дальнейшее развитие объекта предполагается производить собственными силами. Также снятие объекта с поддержки требуется, если возникает необходимость удаления объекта.

- Следует с осторожностью относиться к удалению объектов поставщика, даже в том случае, если не предполагается их непосредственное использование. Алгоритмы конфигурации поставщика могут использовать эти объекты для «внутренних нужд», и их удаление может привести к нарушению логической целостности конфигурации.

- Сортировка объектов метаданных. Особенность реализации объединения конфигураций заключается в том, что в случае серьезных различий в составе поддерева объектов выбор правила «Порядок из конфигурации поставщика» не гарантирует полного сохранения порядка после объединения. Если сохранение данного порядка представляется критичным, можно после обновления конфигурации выполнить объединение с конфигурацией поставщика, вызываемое из диалога настройки поддержки. Повторное выполнение объединения (после обновления) приведет к полному восстановлению порядка.

- Сопоставление объектов. При выполнении обновления новым объектам поставщика можно поставить в соответствие пользовательские объекты. Однако следует очень аккуратно относиться к этой процедуре, поскольку впоследствии изменить данное соответствие будет невозможно.

- Добавление подчиненных объектов. Необходимость добавить реквизит или новый макет (форму) в объект, например справочник, не является поводом для снятия самого справочника с поддержки. Механизм поддержки обеспечит сохранение этого реквизита после обновления справочника. Но следует помнить, что не любое добавление подчиненного объекта одинаково просто. Например, добавление измерения в регистр серьезно меняет его функционирование.

- Обновление конфигурации в среде групповой разработки. Самая лучшая рекомендация заключается в полном захвате всех объектов конфигурации перед обновлением. Если объект не захвачен в хранилище, то его обновление выполнено не будет. Наконец захват корневого объекта конфигурации является обязательным требованием выполнения обновления.

- Редактирование конфигурации в процессе настройки обновления. Мы не рекомендуем пользоваться этой возможностью. Во-первых, последующее обновление сравнения потребует времени. Во-вторых, если в процессе редактирования появился новый объект, для которого нужно выполнить обновление, то расстановка правил по умолчанию для него выполнена не будет. Если подобная необходимость все-таки возникает, следует после завершения редактирования закрыть окно сравнения конфигураций и заново выполнить команду обновления конфигураций. В отличие от нажатия кнопки «Обновить», будет произведена полная установка правил обновления объектов.

- Не рекомендуется без серьезной необходимости переименовывать объекты метаданных, процедуры и функции. Следует помнить, что имя, используемое в модуле для доступа к какому-либо объекту, может формироваться динамически, и обнаружение и исправление таких мест может представлять собой сложную задачу. Кроме того, изменение большого количества модулей создаст дополнительные проблемы при последующих обновлениях.

- Локализация текстов модулей. Редактирование параметров функции НСтр() рекомендуется производить с использованием средства «Редактирование текстов интерфейса», а не прямым исправлением модулей. В случае использования в строках некоторых знаков препинания, корректная форма шаблона для функции НСтр() может выглядеть довольно сложной. Лучше поручить это перекодирование программе.

- Объединение сложных свойств. Следует помнить, что действительное объединение таких свойств, как формы, макеты или интерфейсы, то есть установка для них правил «Объединять с приоритетом ...», является достаточно сложным процессом, и требует обязательного ручного контроля результатов. Рекомендуется пользоваться средствами формирования визуального отчета о сравнении таких свойств в целях анализа различий. Иногда может оказаться

целесообразным отказаться от объединения и заново внести доработки в новую версию формы поставщика.

- Модификация общих модулей. При разработке собственных универсальных процедур, рекомендуется располагать их в новых модулях, а не помещать в модули поставщика. Если возникает необходимость модификации модулей поставщика, при обновлении следует помнить о возможности настройки объединения на уровне отдельных процедур.

- По аналогии с модификацией модулей, рекомендуется по возможности добавлять новые процедуры и функции, а не изменять существующие. Если подобное невозможно (например, требуется модификация обработчика события), следует оформлять добавляемый код в отдельную процедуру, а в процедуре поставщика вставлять только ее вызов.

- Анализ изменений внесенных поставщиком. Хотя механизм формирования отчета о сравнении предоставляет достаточно полную информацию о внесенных изменениях, его анализ может занять много времени. Рекомендуется внимательно ознакомиться с прилагаемым к новой версии конфигурации поставщика файлом описания внесенных изменений. Это может помочь выбрать правильную стратегию для обновления различных объектов.

- Не следует обновлять объекты, используя копирование через буфер обмена. Это не только нарушит нормальное функционирование механизма поддержки, но может привести к нарушению логической целостности конфигурации и потере данных.

- Ознакомьтесь с разделами ИТС, посвященными механизму поставки и поддержки конфигураций. Понимание принципов его функционирования повысит эффективность его использования.

1.2.1 Базовый принцип программирования управляемой формы в 1С

Начнем с небольшого описания понятия «управляемая форма» и связанных концепций платформы 1С. В 2008 году стала доступна новая версия платформы 1С Предприятие 8.2 (далее Управляемое приложение), которая полностью меняет весь слой работы с интерфейсом. Сюда относится и командный интерфейс, и формы, и оконная система. При этом не только меняется модель разработки пользовательского интерфейса в конфигурации, но и предлагается новая архитектура разделения функциональности. Управляемое приложение поддерживает следующие типы клиентов:

- Толстый клиент (обычный и управляемый режим запуска).
- Тонкий клиент.
- Веб-клиент.

В управляемом приложении используются формы, построенные на новой технологии. Они называются «Управляемые формы». Для облегчения перехода прежние формы (т.н. Обычные формы) также поддерживаются, но их функциональность не развивается, и они доступны только в режиме запуска. Основные отличия управляемых форм для разработчика:

– Декларативное, а не «по пикселям» описание структуры. Конкретное размещение элементов выполняется системой автоматически при отображении формы.

– Вся функциональность формы описывается в виде реквизитов и команд. Реквизиты – это данные, с которыми работает форма, а команды – выполняемые действия.

– Форма выполняется и на сервере, и на клиенте.

– В контексте клиента, недоступны практически все прикладные типы, и соответственно невозможно изменить данные в информационной базе.

– Для каждого метода или переменной формы обязательно должна быть указана директива компиляции, определяющая, место выполнения (клиент или сервер) и доступ к контексту формы.

Перечислим директивы компиляции методов формы 1С Предприятие:

– &НаКлиенте.

– &НаСервере.

– &НаСервереБезКонтекста.

– &НаКлиентеНаСервереБезКонтекста.

Проиллюстрируем перечисленное. На рисунке 1.1 приведен пример управляемой формы и ее модуля в режиме разработки.

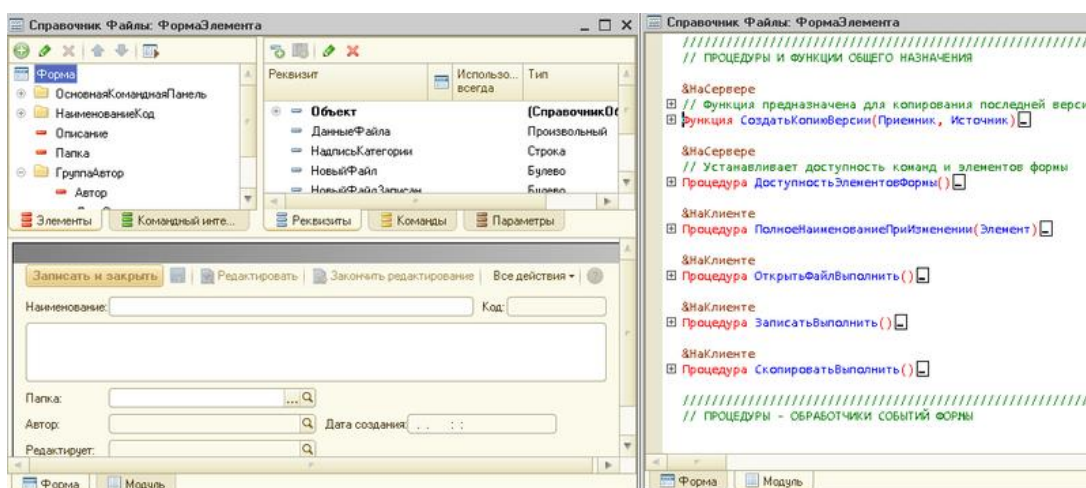


Рисунок 1.1 – Управляемая форма

Прошло уже несколько лет как новая версия платформы 1С активно используется и выпущено множество решений (конфигураций) как фирмой 1С, так и ее многочисленными партнерами. Сложилось ли за это время у разработчиков единое понимание принципов клиент–серверного взаимодействия при создании форм, и изменился ли подход к реализации программных модулей в новых архитектурных реалиях? Рассмотрим структуру кода (модуль формы) в нескольких формах одной типовой конфигурации и попробуем найти закономерности. Под структурой будем понимать секции кода (чаще всего это блоки комментариев) выделенные разработчиком для

группировки методов и директивы компиляции этих методов.

Секция обработчиков событий:

- Метод– наклиенте.
- Метод– на сервере.
- Метод– на клиенте.
- Секция служебных процедур и функций.
- Вспомогательные функции управления вводом.
- Служебные процедуры и функции.
- Документы оплаты.
- Обработчики событий.
- Служебные процедуры на сервере.
- Служебные процедуры на сервере без контекста.
- Обработчики событий шапки.
- Процедуры общего назначения.
- Обработчики событий формы.
- Процедуры подсистемы «контактная информация».

По сути, структура кода отсутствует, аналогична тому, что было в 1С
Предприятие 8.1:

- Неинформативные слова «Общие, Служебные, Вспомогательные».
- Робкие попытки разделить клиентские и серверные методы.
- Часто методы группируются по интерфейсным элементам «Работа с табличной частью Товары, Контактной информацией».

И не будем забывать, что это все в рамках одной конфигурации. Почему существующий стандарт разработки от фирмы 1С не помогает?

Посмотрим опубликованные на дисках ИТС и в различных «Пособиях разработчика...» принципы, рекомендуемые при написании управляемой формы.

- Минимизируйте число серверных вызовов.
- Максимум вычислений на сервере.
- Не контекстные вызовы сервера быстрее контекстных.
- Програмируйте с учетом клиент–серверного взаимодействия.

Это лозунги, абсолютно верные, но как их реализовать? Клиент–серверное взаимодействие используется в различных программных технологиях не один десяток лет. Ответ на обозначенные вопросы давно известны:

- Remote Facade (далее Интерфейс удаленного доступа).
- Data Transfer Object (далее Объект переноса данных).

Мартин Фаулер нашел решение этих проблем, его описание данных принципов:

- Объект, потенциально предназначенный для удаленного доступа, должен иметь интерфейс с низкой степенью детализации, что позволит максимально уменьшить количество вызовов, необходимых для выполнения определенной процедуры. Вместо того, чтобы запрашивать счёт и все его пункты отдельно, надо считать и обновить все пункты счёта за одно обращение.

Это влияет на всю структуру объекта. Запомните: интерфейс удаленного доступа не содержит логики домена.

– Если бы я был заботливой мамой, то обязательно сказал бы своему ребенку: «Никогда не пиши объекты переноса данных!» В большинстве случаев объекты переноса данных представляют собой не более чем раздутый набор полей. Ценность этого омерзительного монстра состоит исключительно в возможности передавать по сети несколько элементов информации за один вызов – прием, который имеет большое значение для распределенных систем.

Прикладной программный интерфейс доступный разработчику при разработке управляемой формы, содержит много примеров данных принципов. Например метод ОткрытьФорму(), типичный интерфейс для 1С Предприятие 8.

```
ПараметрыОткрытия = Новый Структура("Параметр1,
Параметр2, Параметр3", Значение1, Значение2, Значение3);
Форма = ОткрытьФорму(ИмяФормы, ПараметрыОткрытия);
Сравните с принятым в 1С Предприятие v8.1 стилем и заметите разницу.
```

```
Форма = ПолучитьФорму(ИмяФормы);
Форма.Параметр1 = Значение1;
Форма.Параметр2 = Значение2;
Форма.Открыть();
```

В контексте управляемой формы множество «Объектов переноса данных». Можно выделить системные и определяемые разработчиком.

Системные моделируют на клиенте прикладной объект, в виде одного или несколько элементов данных формы. Создать их вне привязки к реквизитам формы нельзя.

- ДанныеФормыСтруктура;
- ДанныеФормыКоллекция;
- ДанныеФормыСтруктураСКоллекцией;
- ДанныеФормыДерево;

Преобразование системных объектов переноса данных в прикладные типы и обратно в 1С Предприятие выполняется методами:

- ЗначениеВДанныеФормы();
- ДанныеФормыВЗначение();
- КопироватьДанныеФормы();
- ЗначениеВРеквизитФормы();
- РеквизитФормыВЗначение();

Часто явное преобразование используется при адаптации существующего решения. Методы могут ожидать (использовать особенности) входные параметры, например ТаблицаЗначений, а не ДанныеФормыКоллекция, или метод был определен в контексте прикладного объекта и стал недоступен.

Пример 1С v8.1:

```
// на клиенте в контексте формы
ЗаполнитьКэшПользователей(ПодразделениеСсылка)
```

Пример 1С v8.2:

```
// на сервере в контексте формы
```

```
ОбработкаОбъект = РеквизитФормыВЗначение ("Объект");
ОбработкаОбъект.ЗаполнитьКэшПользователей (Подразделен
иеСсылка);
```

```
ЗначениеВРеквизитФормы (ОбработкаОбъект, "Объект");
```

Объекты переноса данных, структура которых определяется разработчиком это небольшое подмножество типов доступных и на клиенте и на сервере. Наиболее часто в качестве параметров и результатов методов «огрубленного» интерфейса используются:

- Примитивные типы (строка, число, булево)
- Структура
- Соответствие
- Массив
- Ссылки на прикладные объекты (уникальный идентификатор и текстовое представление)

Пример: метод принимает список заказов для изменения статуса и возвращает клиенту описание ошибок.

```
&НаСервереБезКонтекста
Функция                СерверИзменитьСтатусЗаказов (Заказы,
НовыйСтатус)
    Ошибки                =        Новый        Соответствие ();        //
[заказ] [описание ошибки]
    Для Каждого Заказ Из Заказы Цикл
        НачатьТранзакцию ();
        ДокОб = Заказ.ПолучитьОбъект ();
        ... другие действия, возможно не только с
заказом...

        ОтменитьТранзакцию ();
        Ошибки.Вставить (Заказ, ОписаниеОшибки ());
    КонечПопытки;
    КонечЦикла;
    Возврат Ошибки;
КонечФункции // СерверИзменитьСтатусЗаказов ()
```

Структурируем код. Главные цели, которые должен отражать модуль управляемой формы и подходы к решению.

- Четкое разделение клиентского и серверного кода. Не будем забывать, в момент выполнения это два взаимодействующих процесса, в каждом из которых существенно отличается доступный функционал.

- Четкое выделение интерфейса удаленного доступа, какие методы сервера можно вызывать с клиента, а какие нельзя? Названия методов удаленного интерфейса начинаются с префикса «Сервер». Это позволяет, читая код сразу видеть переход управления на сервер, и упрощает использование контекстной подсказки. Отметим, что официальная рекомендация (ИТС) предлагает именовать методы с постфиксами, например, так ИзменитьСтатусЗаказовНаСервере(). Однако повторим не все серверные

методы можно вызывать с клиента, и поэтому более важна логическая доступность, а не место компиляции. Поэтому префиксом «Сервер» отмечаем только методы доступные для клиента, метод–пример назовем СерверИзменитьСтатусЗаказов().

- Удобочитаемость. Дело вкуса, принимаем порядок, когда модуль начинается с процедур создания формы на сервере и методов удаленного доступа.

- Сопровождаемость. Должно быть однозначно определено место для добавления нового кода. Важный момент, автоматически создаваемые конфигуратором заготовки методов добавляются в конец модуля. Т.к чаще всего автоматически создаются обработчики событий элементов формы, то соответствующий блок расположен последним, чтобы не перетаскивать каждый обработчик в другое место модуля.

Ниже приведена базовая структура модуля смотрите рисунок 1.2, реализующая перечисленные цели.

- Графический вариант наглядно показывает основной поток выполнения.

- Текстовый вариант это пример оформления шаблона для быстрой вставки структуры в новый модуль формы.

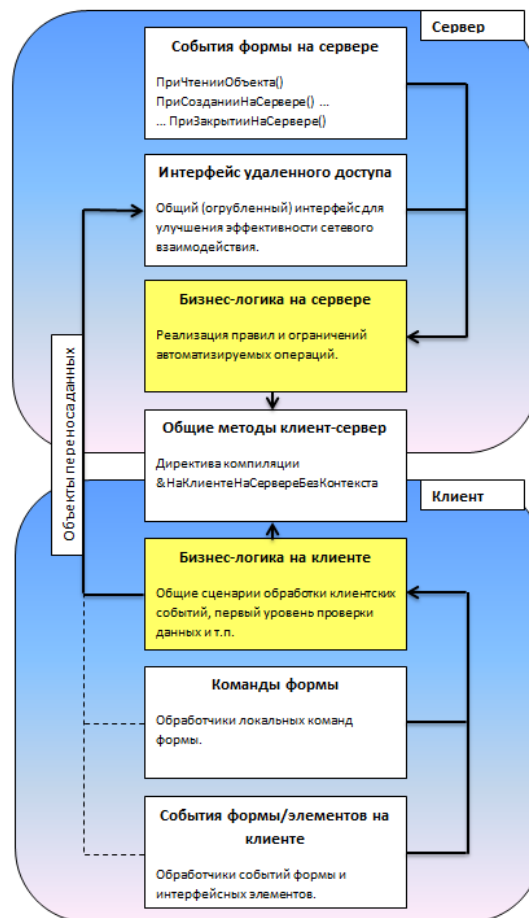


Рисунок 1.2 – Базовая структура модуля

1.2.2 Групповая разработка прикладных решений

Механизмы configurator, 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:

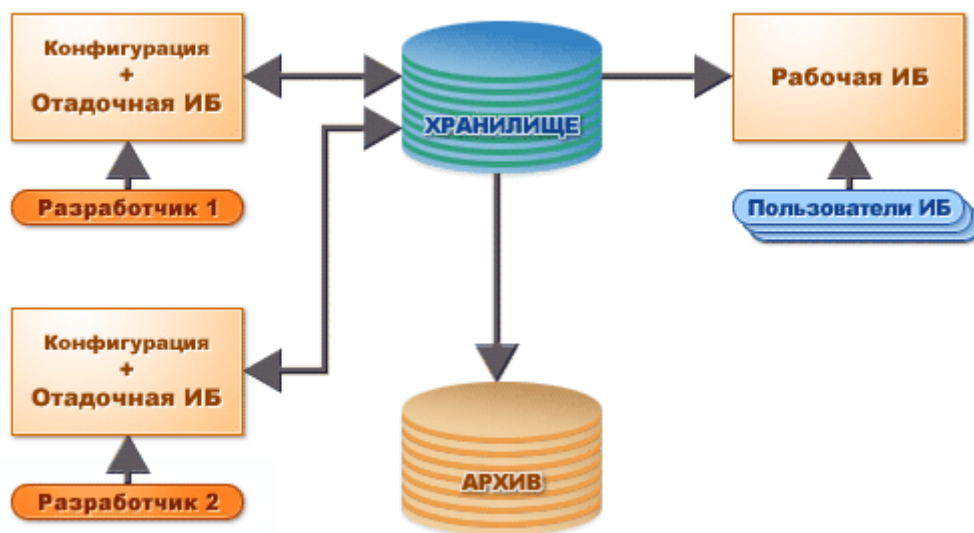


Рисунок 1.3 – Права доступа разработчиков

Хранилище конфигурации. Хранилище конфигурации является средством, позволяющим осуществлять групповую разработку прикладных решений. Также хранилище конфигурации обеспечивает версионирование изменений, выполняемых в разрабатываемой конфигурации. В силу этого использование хранилища может быть очень полезным и для одного разработчика, т.к. позволяет документировать изменения, выполняемые в прикладном решении и работать с версиями.

Для осуществления групповой разработки прикладного решения на общедоступном сетевом ресурсе создается хранилище конфигурации и назначается его администратор:

Администратор осуществляет формирование списка пользователей, имеющих доступ к хранилищу, может просматривать список пользователей, подключенных к хранилищу и освобождать объекты конфигурации от захвата:

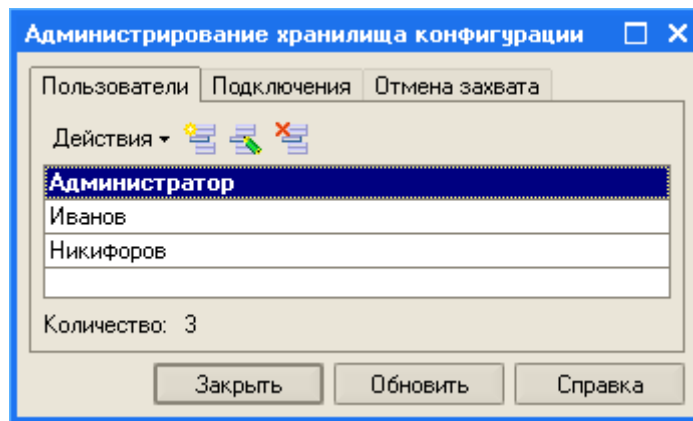


Рисунок 1.4 – Администрирование хранилища конфигураций

Для того чтобы иметь возможность модифицировать прикладное решение, расположенное в хранилище, разработчику необходимо подключиться к хранилищу, указав имя пользователя и пароль:

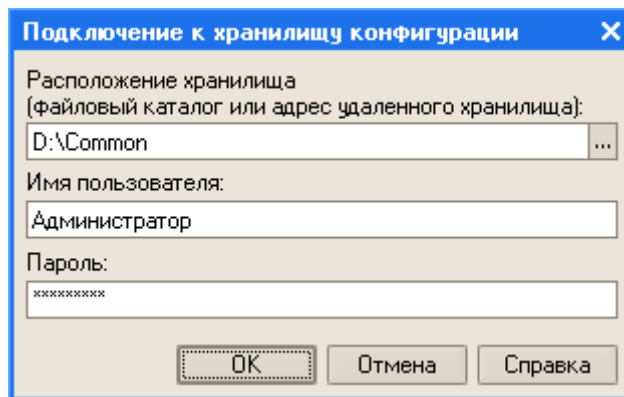


Рисунок 1.5 – Подключение к хранилищу конфигураций

Окно хранилища конфигурации. При групповой разработке прикладное решение рассматривается как набор объектов, закрытых для изменения. Каждый из пользователей, допущенных к работе с хранилищем, может «захватить» для изменения произвольное число объектов, не захваченных другими пользователями. Каждый объект может быть захвачен только одним пользователем.

Каждый из разработчиков, подключенных к хранилищу, редактирует захваченные в хранилище объекты и отлаживает прикладное решение на своей текущей информационной базе так же, как и в обычном режиме. После внесения изменений в объект прикладного решения, разработчик может поместить измененный объект в хранилище с тем, чтобы другие пользователи могли обновить этот объект в своих конфигурациях. При этом разработчик может снабдить выполненные изменения текстовым комментарием смотри рисунок 1.6.

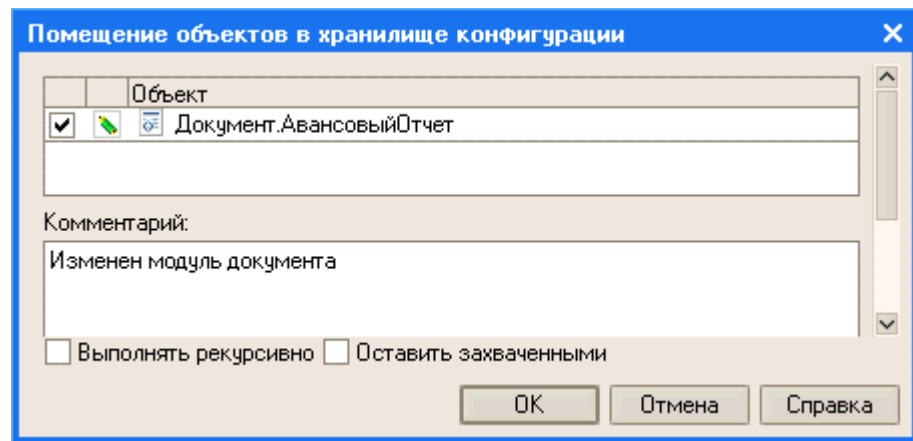


Рисунок 1.6 – Помещение объектов в хранилище конфигурации

В любой момент времени можно выполнить сравнение текущей конфигурации с хранилищем или выполнить сохранение хранилища как конфигурации.

История хранилища. Конфигуратор 1С Предприятия поддерживает ведение истории хранилища, смотрите рисунок 1.7.

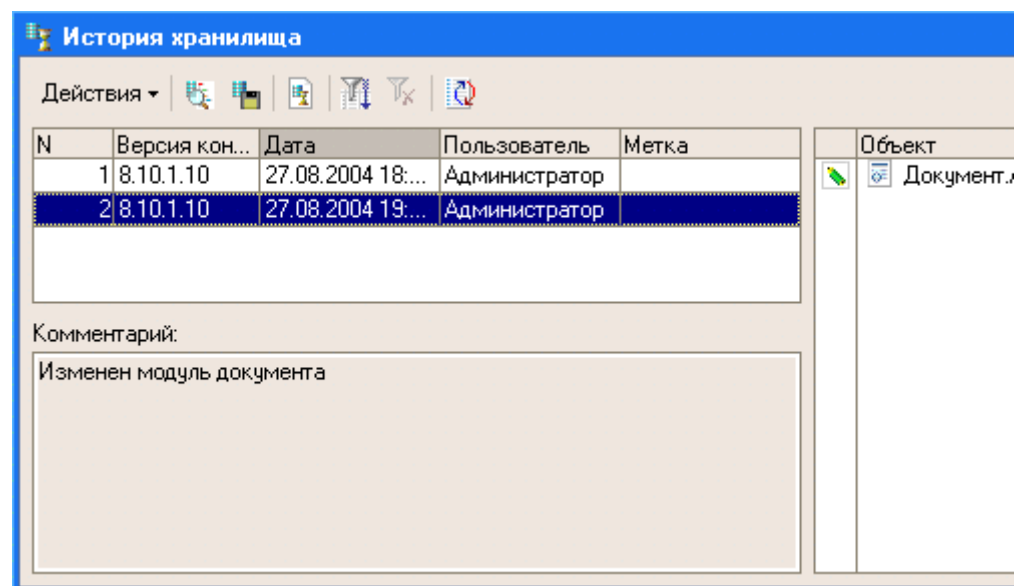


Рисунок 1.7 – История хранилища

Каждая строка списка отображает очередную версию прикладного решения в хранилище. Каждую версию можно открыть для просмотра, загрузить вместо текущей, сравнить с текущей или сохранить в файл на диске.

Поддерживается возможность отката назад и удаления ненужных версий, опубликованных в хранилище, а также возможность удаления самых ранних ненужных версий путем сокращения до нужной версии.

Существует возможность вывода отчетов по истории хранилища, содержащих информацию об изменении отдельных элементов прикладного решения и всего прикладного решения в целом.

Отчет по версиям хранилища отражает состав добавленных или измененных объектов.

Отчет по объектам разработки содержит информацию об изменениях, которые были внесены в конкретные объекты прикладного решения.

Таким образом, использование хранилища полезно и для одного разработчика, т.к. история хранилища позволяет документировать изменения, выполняемые в прикладном решении и работать с версиями.

Работа с хранилищем в окне конфигурации. Функции работы с хранилищем доступны не только из окна хранилища, но из окна конфигурации. В нем, так же, как и в окне хранилища, отображается состояние объектов конфигурации, смотрите рисунок 1.8.

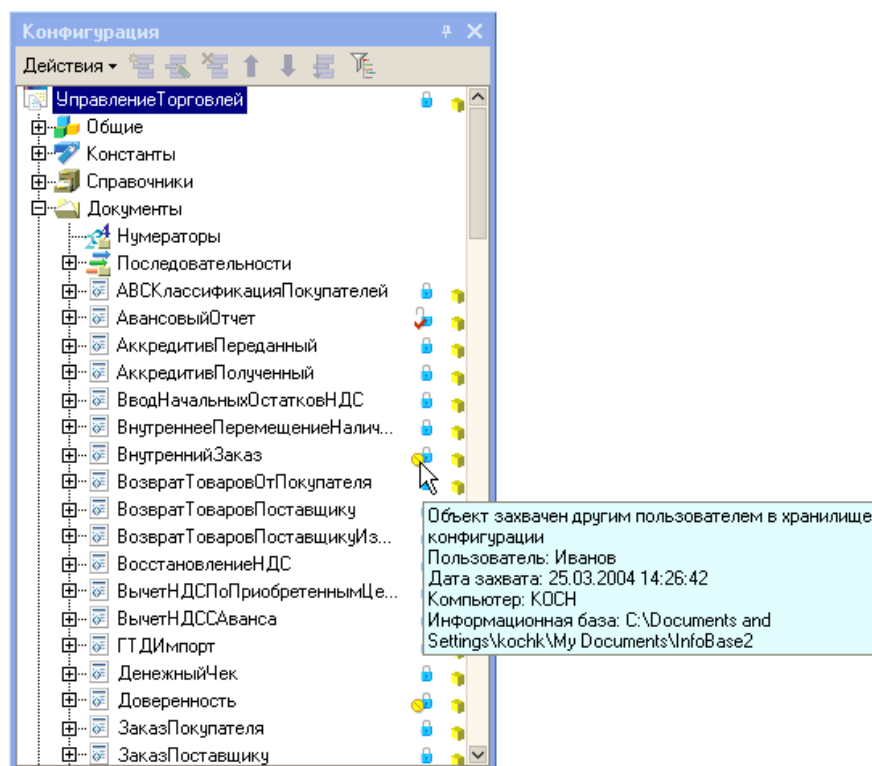


Рисунок 1.8 – Окно конфигурации

Находясь в окне конфигурации, разработчик может захватывать объекты в хранилище, отменять захват, помещать объекты в хранилище, сравнивать объект с объектом, находящимся в хранилище и получать историю объекта хранилища.

Работа с хранилищем без подключения. Некоторые действия с хранилищем можно выполнять без подключения. Если текущая конфигурация не подключена к хранилищу, разработчик может установить соединение с хранилищем.

Удаленная работа с хранилищем конфигурации. Начиная с версии 8.1.11, с хранилищем конфигурации можно работать, используя не только общий сетевой ресурс, но и соединение по локальной сети (используя протокол tcp) и

интернет–соединение (используя протокол http). В общем случае взаимодействие разработчиков с хранилищем конфигурации может выглядеть следующим образом, смотрите рисунок 1.9.

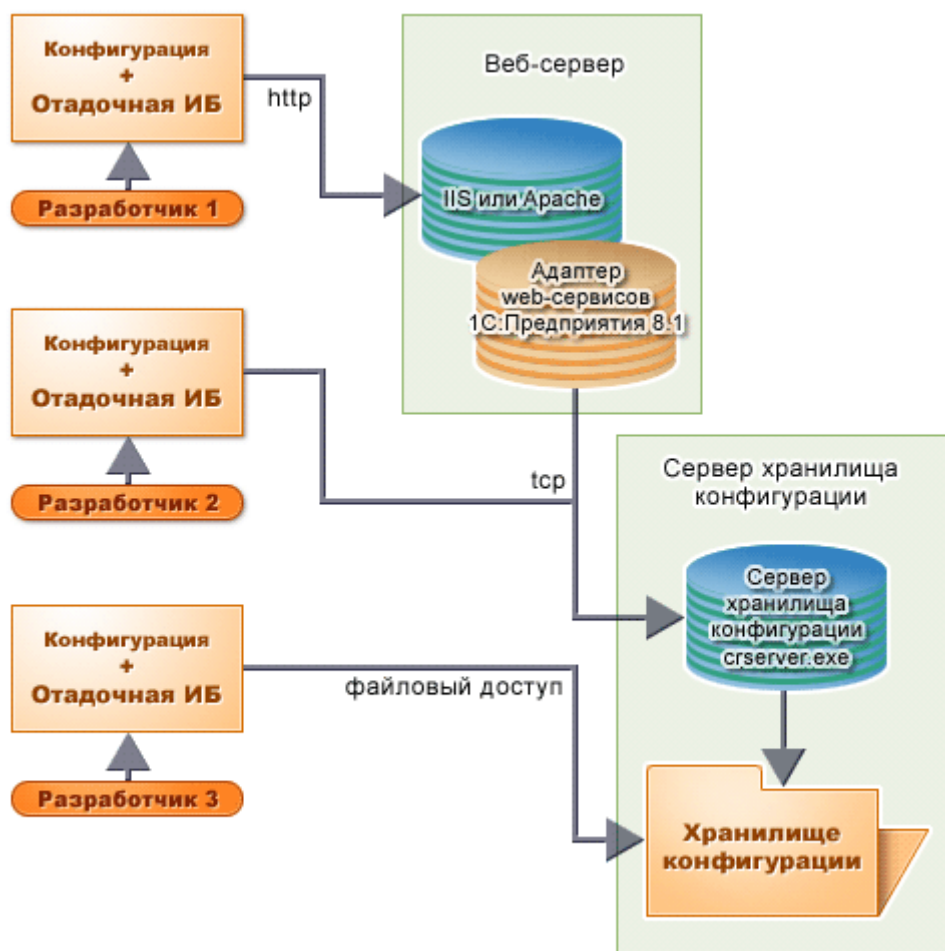


Рисунок 1.9 – Взаимодействие разработчика с хранилищем конфигурации

Хранилище конфигурации может располагаться на компьютере под управлением операционных систем как Windows, так и Linux. В операционной системе Windows сервер хранилища конфигурации может быть запущен как приложение или установлен как сервис. В операционной системе Linux сервер хранилища конфигурации может быть запущен как процесс или как демон. При этом:

- При подключении по tcp протоколу «клиентское приложение» взаимодействует с сервером хранилища конфигурации. Тот в свою очередь взаимодействует непосредственно с хранилищем.

- При подключении по http протоколу "клиентское приложение" подключается к веб–серверу. Веб–сервер обращается к серверу хранилища, который взаимодействует с хранилищем конфигурации.

1.2.3 Система прав доступа

Система прав доступа позволяет описывать наборы прав, соответствующие должностям пользователей или виду деятельности. Структура прав определяется конкретным прикладным решением. Кроме этого, для объектов, хранящихся в базе данных (справочники, документы, регистры и т.д.) могут быть определены права доступа к отдельным полям и записям. Например, пользователь может оперировать документами (накладными, счетами и т.д.) определенных контрагентов и не иметь доступа к аналогичным документам других контрагентов.

Роли. Для реализации ограничения прав доступа в прикладных решениях предназначены специальные объекты конфигурации Роли.

Интерактивные и основные права. Все права, поддерживаемые системой 1С Предприятие, можно разделить на две большие группы: основные и интерактивные. Основные права описывают действия, выполняемые над элементами данных системы или над всей системой в целом, и проверяются всегда, независимо от способа обращения к данным. Интерактивные права описывают действия, которые могут быть выполнены пользователем интерактивно. Соответственно проверяются они только при выполнении интерактивных операций стандартными способами, причем в клиент–серверном варианте все проверки прав (кроме интерактивных) выполняются на сервере.

Основные и интерактивные права взаимосвязаны. Например, существует основное право Удаление, которому соответствуют два интерактивных права: Интерактивное удаление и Интерактивное удаление помеченных. Если пользователю запрещено Удаление, то и все интерактивные "удаления" также будут запрещены для него. В то же время, если пользователю разрешено Интерактивное удаление помеченных, это значит, что Удаление ему также разрешается.

Кроме того, основные права могут зависеть друг от друга. В результате образуются довольно сложные цепочки взаимосвязей, которые отслеживаются системой автоматически: как только разработчик снимает разрешение на какое–либо право, система сама снимает разрешения на все права, которые зависят от этого права. И наоборот, при установке какого–либо права разработчиком, система сама устанавливает все права, от которых это право зависит.

Например, для того, чтобы пользователь имел право «Интерактивное удаление помеченных», ему необходимо обладать интерактивными правами Редактирование. Это право, в свою очередь, требует наличия интерактивного права «Просмотр», смотрите рисунок 1.10.

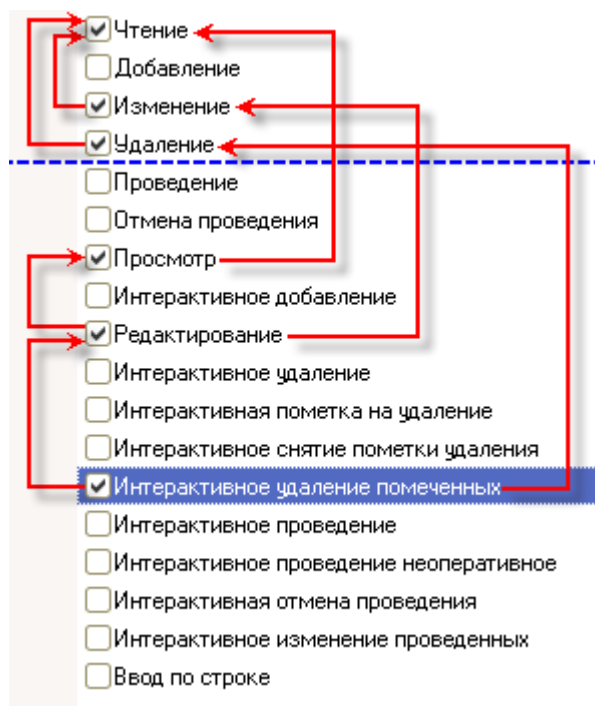


Рисунок 1.10 – Схема зависимости основных прав

Право «Интерактивное удаление помеченных» требует наличия основного права «Удаление». Интерактивное право «Редактирование» требует наличия основного права «Изменение». Интерактивное право «Просмотр» требует наличия основного права «Чтение». Кроме этого основные права «Изменение» и «Удаление» требуют наличия основного права «Чтение».

Ограничение доступа к данным на уровне записей и полей. Среди действий над объектами, хранящимися в базе данных (справочниками, документами и т.д.), есть действия, отвечающие за чтение или изменение информации, хранящейся в базе данных. К таким действиям относятся:

- чтение получение записей или их фрагментов из таблицы базы данных;
- добавление новых записей без изменения существующих;
- изменение существующих записей;
- удаление некоторых записей без внесения изменений в оставшиеся.

Для этих действий в процессе настройки ролей могут быть заданы дополнительные условия на данные (ограничение доступа к данным). В этом случае над конкретным объектом, хранимым в базе данных, может быть выполнено запрошенное действие только в том случае, если ограничение доступа к данным для данных этого объекта принимает значение «истина». Аналогичные условия могут быть заданы и для таблиц базы данных, не имеющих объектной природы (регистров), смотрите рисунок 1.11.

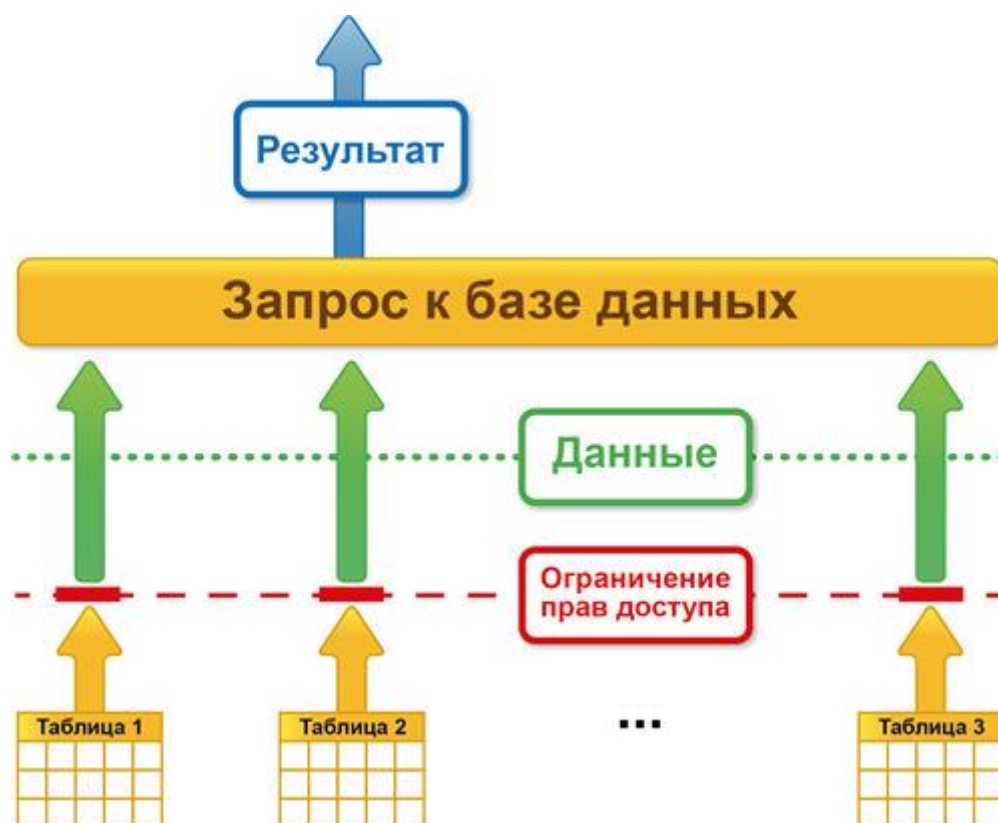


Рисунок 1.11 – Ограничение прав доступа для таблиц

Для объектных таблиц и регистров сведений могут быть заданы разные ограничения для различных полей таблицы, что позволяет определять ограничения не только на уровне записей базы данных, но и на уровне отдельных ее полей.

Ограничение доступа к данным представляет собой условие, описанное на языке, который является подмножеством языка запросов. Это условие применяется для каждой записи таблицы базы данных, над которой выполняется операция. Если условие принимает значение "истина", то операция выполняется, а если нет, то не выполняется. Условие ограничения доступа может быть уточнено с помощью инструкций препроцессора (#ЕСЛИ <условие>, #ТОГДА.. и др.), что сделает его более эффективным. При просмотре списков и формировании отчетов существует возможность обеспечить отображение только тех данных, доступ к которым пользователю разрешен.

Для регистров накопления, бухгалтерского учета и расчета условия позволяют разграничить доступ по значениям измерений (для регистров бухгалтерского учета по балансовым измерениям), а для объектных данных и регистров сведений условия позволяют разграничивать доступ к данным по любым полям.

Условия ограничения можно ввести вручную или создать с помощью конструктора ограничений доступа к данным.

Параметры сеанса. Параметры сеанса представляют собой объекты прикладного решения, которые предназначены для использования в ограничениях доступа к данным для текущего сеанса (но могут применяться и для других целей). Их значения сохраняются в течение данного сеанса 1С Предприятия. Использование параметров сеанса позволяет снизить время доступа к данным при ограничении доступа на уровне записей и полей.

Привилегированные модули. Существует возможность назначения привилегированных модулей. В такие модули могут быть перенесены операции, использующие данные, на которые у текущего пользователя нет прав.

Например, пользователю могут быть назначены права, позволяющие создавать новый документ. Однако никаких прав на регистр, в котором этот документ создает движения при проведении, пользователю не дано. В такой ситуации процедура проведения документа может быть вынесена в привилегированный модуль, который выполняется на сервере без проверки прав. В результате, несмотря на то, что соответствующий регистр для пользователя недоступен, пользователь все же сможет проводить созданные им документы.

Привилегированный режим исполнения программного кода. Привилегированный режим исполнения кода, аналогичный режиму работы кода привилегированных модулей, можно включить/выключить средствами встроенного языка. Для этого в глобальном контексте предусмотрена процедура `УстановитьПривилегированныйРежим()`, а также функция `ПривилегированныйРежим()`, которая позволяет определить, включен привилегированный режим, или нет.

Использование привилегированного режима позволяет, во-первых, ускорить работу, так как не будут накладываться ограничения на доступ к данным, а во-вторых, позволяет выполнять операции с данными от лица пользователей, которым эти данные недоступны.

Привилегированный режим рекомендуется использовать тогда, когда с логической точки зрения нужно отключить проверку прав, или когда можно отключить проверку прав, чтобы ускорить работу. Допустимо использовать привилегированный режим тогда, когда работа с данными от лица некоторого пользователя не нарушает установленные для этого пользователя права доступа.

1.2.4 Система типов

Система типов – это особая система, по которой организуются данные, используемые прикладными решениями. Система типов позволяет представить информацию реального мира в терминах, «понятных» для 1С Предприятия 8.

Система типов предоставляет широкие возможности как для описания непосредственно бизнес-логики прикладных решений, так и для выполнения задач промежуточной обработки данных.

Описание системы типов содержится в синтакс-помощнике, во встроенной справке и в документации.

Основной особенностью системы типов является то, что есть типы, существующие в любом прикладном решении. Сами эти типы определены на уровне платформы и присутствуют всегда, независимо от действий разработчика. Наряду с ними в конкретном прикладном решении могут существовать различные типы данных, присущие именно этому конкретному прикладному решению. Для таких типов данных на уровне платформы определены лишь общие правила их создания, шаблоны. А конкретные типы данных создаются платформой уже исходя из того, каким образом разработчик формирует и модифицирует структуру прикладного решения.

Типы данных, определенные на уровне платформы. Набор типов, которыми могут оперировать прикладные решения, довольно разнообразен. Он позволяет решать как задачи обработки данных, так и задачи представления этих данных пользователю и интерактивной работы с ними. Можно выделить несколько основных категорий типов данных.

Примитивные типы. Примитивные типы данных – это такие типы как Строка, Число, Дата, Булево и другие. Эти типы не являются чем-то особенным для 1С Предприятия 8. Как правило, такие типы данных существуют и в других программных системах.

Значения примитивных типов являются простыми неделимыми значениями, в которых нельзя выделить отдельные составляющие. Например, значениями типа Число могут быть 1, 8, 15 и др. Чтобы создать значение примитивного типа, в тексте программы нужно указать его литерал-символьный идентификатор значения.

Универсальные коллекции значений. Также, существуют более сложные типы данных. Например, платформа поддерживает целый ряд типов, которые представляют собой универсальные коллекции значений: Массив, Структура, СписокЗначений и другие.

Общие типы. Кроме этого в платформе реализованы специфические типы данных, реализующие ту или иную функциональность прикладных решений.

Общие типы называют также общими объектами. Значения этих типов, в отличие от значений примитивных типов, представляют собой совокупность значений отдельных свойств объекта. Поэтому их называют экземплярами объектов.

Экземпляры объектов создаются с помощью специального оператора встроенного языка – Новый.

Интерфейсные типы. Интерфейсные типы позволяют организовывать визуальное взаимодействие прикладного решения с пользователем. В основном это типы, связанные с работой формы их элементов, смотрите рисунок 1.12.

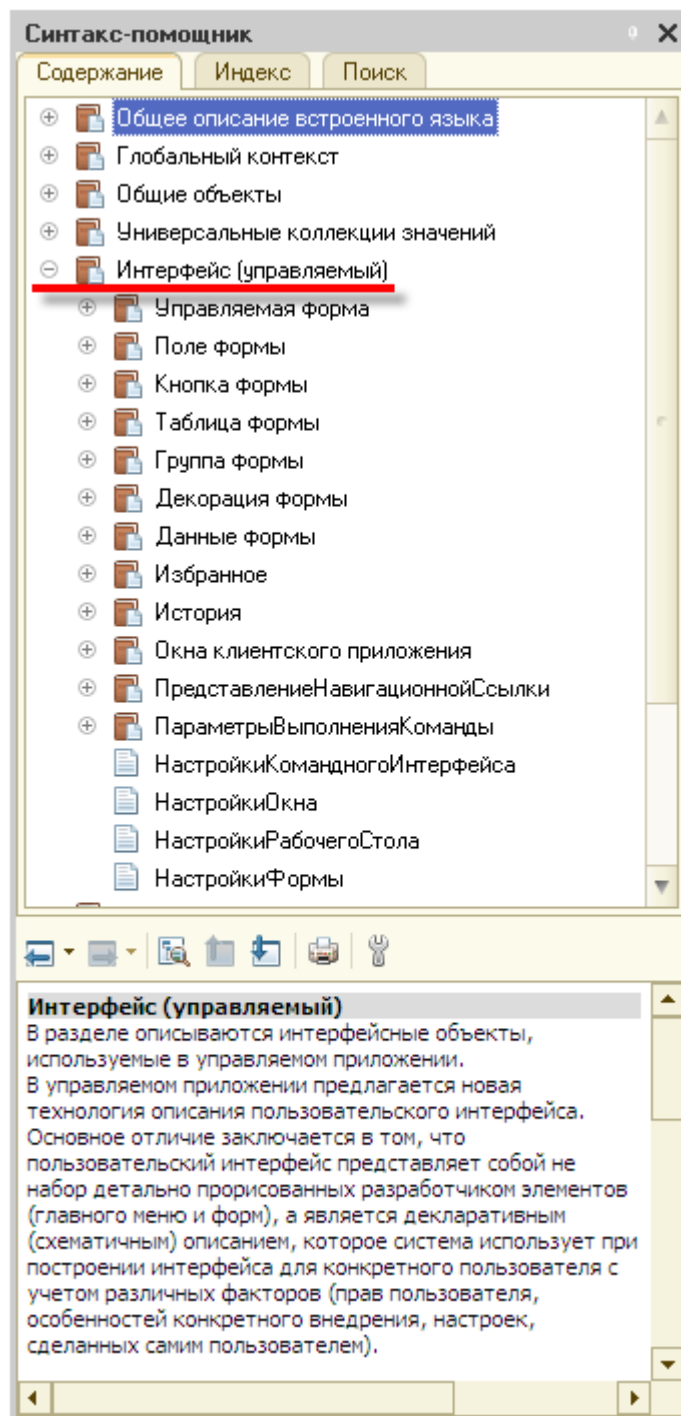


Рисунок 1.12 – Синтакс–помощник

Типы данных, образуемые в прикладном решении. Однако, наряду с типами данных, которые определены на уровне платформы, конкретное прикладное решение может использовать уникальные типы данных, существующие только в этом конкретном прикладном решении. Причем платформа будет полностью поддерживать работу с этими типами данных точно так же, как и с типами, которые определены на уровне самой платформы.

Как правило, появление новых типов данных в прикладном решении связано с использованием прикладных объектов конфигурации. Поэтому такие типы называют еще прикладными типами или прикладными объектами.

На уровне платформы поддерживается несколько классов (шаблонов) прикладных объектов, которые сами по себе не могут быть использованы в конкретном прикладном решении. Например, можно перечислить такие классы прикладных объектов как Справочники, Документы, Регистры сведений, Планы видов характеристик и пр.

Для каждого класса прикладных объектов определена соответствующая ему базовая функциональность: типы таблиц базы данных, которые должны быть созданы для хранения данных, типовые формы, типовые объекты языка, наборы прав и пр.

Разработчик, создавая прикладное решение, не имеет возможности использовать эти классы напрямую, однако может добавить в свое прикладное решение новый объект конфигурации, наследующий всю функциональность того или иного класса, смотрите рисунок 1.13.

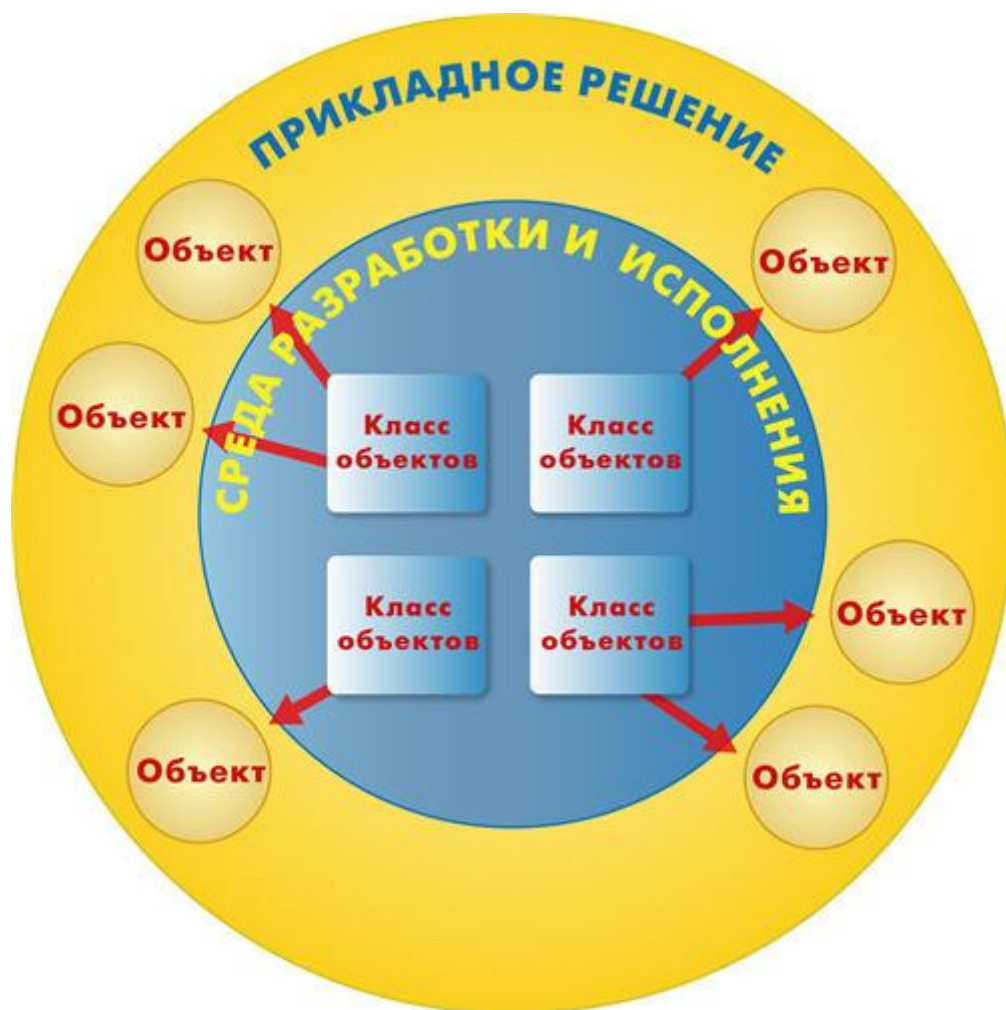


Рисунок 1.13 – Создание прикладного решения

Например, разработчик может добавить в свое прикладное решение новый справочник Номенклатура, который будет наследовать функциональность класса Справочники, или новый документ КассовыйОтчет, который будет наследовать функциональность класса Документы.

Сразу же после такого добавления разработчику становятся доступны новые типы данных, состав которых определяется принадлежностью объекта конфигурации к тому или иному классу прикладных объектов.

Например, после создания нового справочника Номенклатура, становятся доступны следующие типы данных:

- СправочникМенеджер.Номенклатура,
- СправочникСсылка.Номенклатура,
- СправочникОбъект.Номенклатура,
- СправочникВыборка.Номенклатура,
- СправочникСписок.Номенклатура.

Система типов описывает лишь общую "структуру" такого типа, правила, по которым будут формироваться объекты этого типа. Конкретное имя типа, состав свойств и методов объекта будут зависеть от того, как разработчик назовет объект конфигурации и какие, например, реквизиты, табличные части он в него добавит.

В то же время, после создания нового регистра накопления ПродажиКомпании, состав новых типов данных будет уже другим:

- РегистрНакопленияМенеджер.ПродажиКомпании,
- РегистрНакопленияВыборка.ПродажиКомпании,
- РегистрНакопленияСписок.ПродажиКомпании,
- РегистрНакопленияНаборЗаписей.ПродажиКомпании,
- РегистрНакопленияЗапись.ПродажиКомпании,
- РегистрНакопленияКлючЗаписи.ПродажиКомпании.

Следует еще раз отметить, что эти типы данных не поддерживаются платформой изначально, и существуют только в конкретном прикладном решении.

Еще один момент, на котором следует акцентировать внимание, проще всего продемонстрировать на примере.

Допустим, в прикладном решении созданы два новых справочника: Номенклатура и Цены. Несмотря на то, что оба эти объекта унаследовали функциональность соответствующего класса Справочники, и для них в прикладном решении был создан один и тот же состав типов данных, «одноименные» типы данных будут являться различными типами данных. Например, СправочникОбъект.Номенклатура и СправочникОбъект.Цены– это различные типы данных.

Так происходит потому, что разработчик может дополнительно к базовой функциональности, унаследованной от соответствующего класса, добавить свою, особенную для каждого объекта конфигурации. Например, оба упомянутых выше справочника могут содержать табличные части (это унаследовано от класса Справочники). Однако для справочника Цены

разработчик не создаст ни одной табличной части, в то время как для справочника Номенклатура он создаст, например, три табличные части. Очевидно, что структура хранения данных типа СправочникОбъект.Номенклатура будет значительно отличаться от структуры хранения данных типа СправочникОбъект.Цены.

1.2.5 Управление торговлей

«1С Управление торговлей 8» это современный инструмент для повышения эффективности бизнеса торгового предприятия.

«1С Управление торговлей 8» позволяет в комплексе автоматизировать задачи оперативного и управленческого учета, анализа и планирования торговых операций, обеспечивая тем самым эффективное управление современным торговым предприятием.

Предметная область, автоматизируемая с помощью "1С Управление торговлей 8", смотрите рисунок 1.14

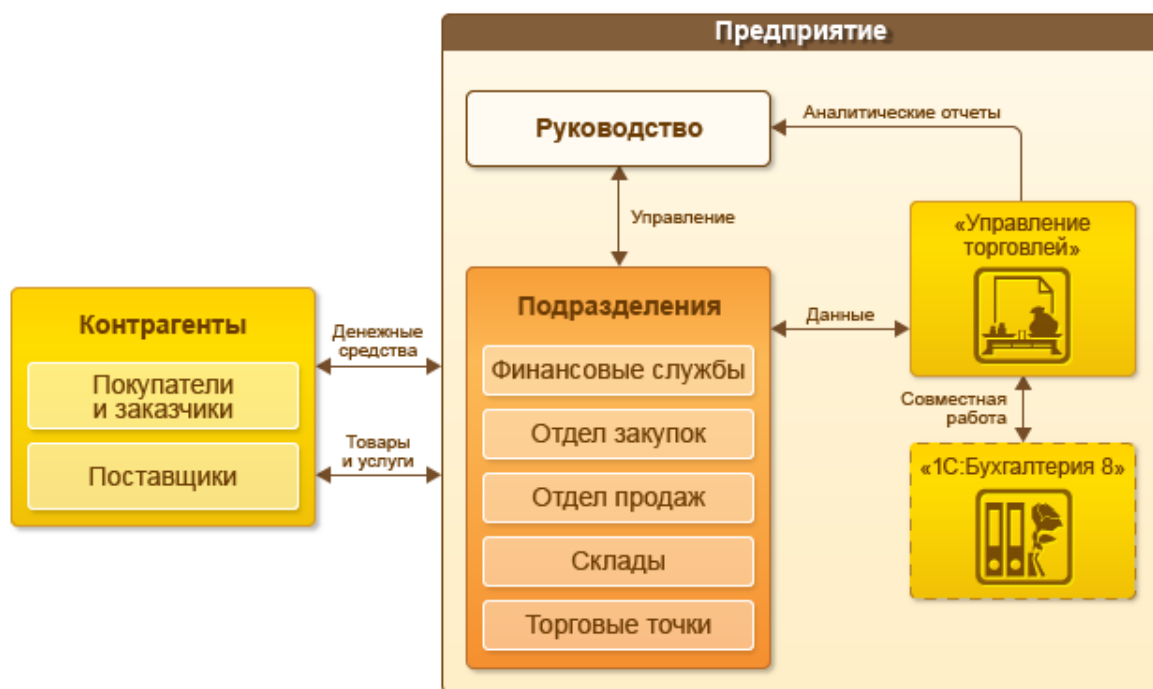


Рисунок 1.14 – 1С Управление торговлей 8

"1С Управление торговлей 8" автоматизирует следующие направления хозяйственной деятельности:

- управление отношениями с клиентами
- управление правилами продаж
- управление процессами продаж
- управление торговыми представителями
- управление запасами
- управление закупками

- управление складом
- управление доставкой товаров
- управление финансами
- контроль и анализ целевых показателей деятельности предприятия.

Рассмотрим подробнее каждый пункт.

Управление отношениями.

Управление отношениями с клиентами, известное также как CRM или Customer Relationship Management, является неотъемлемой функциональной областью современной комплексной информационной системы предприятия.

CRM – это концепция управления отношениями с клиентами в условиях активной конкуренции, нацеленная на максимальное освоение потенциала каждого клиента и партнера в интересах предприятия.

Концепция CRM предполагает регулярный сбор и анализ информации о каждом клиенте, реальном и потенциальном: как клиент отреагировал на деловое предложение, доволен ли он качеством обслуживания, меняются ли его предпочтения со временем, насколько аккуратно он выполняет взятые на себя обязательства, и в конечном итоге сколько дохода клиент приносит (или мог бы принести) предприятию.

Отслеживаются все стадии отношений с клиентом. Тщательно улавливаются признаки опасного ухудшения отношений, поскольку, как известно, на конкурентном рынке затраты на привлечение нового клиента на порядок превышают затраты на удержание имеющегося клиента.

Концепция CRM предусматривает гармоничное сочетание формального подхода и индивидуального отношения к каждому клиенту. Но если количество активных клиентов предприятия измеряется в десятках или сотнях, а количество потенциальных клиентов – соответственно в сотнях или тысячах, то полная реализация концепции CRM приведет к накоплению огромных массивов информации, работать с которыми без привлечения специальных средств автоматизации будет просто невозможно.

Конфигурация содержит средства автоматизации концепции CRM. Функциональные возможности конфигурации позволяют предприятию успешно управлять отношениями с покупателями, поставщиками, смежниками и любыми другими контрагентами.

Предусматривается регистрация всех действий по заключению и исполнению сделок, регистрация всех контактов с клиентами как реальными, так и потенциальными.

Конфигурация позволяет делать следующее:

- использовать персонифицированный подход к нуждам и требованиям каждого клиента;
- хранить полную контактную информацию по контрагентам и их сотрудникам, историю взаимодействия с ними;
- управлять процессом продаж с использованием механизма бизнес-процессов (сделки с клиентом);

- анализировать незавершенные и планировать предстоящие сделки с покупателями и потенциальными клиентами;
- регистрировать каждое обращение потенциального клиента и в дальнейшем анализировать процент привлечения клиентов, а также производить анализ удовлетворения первичного спроса;
- оперативно контролировать состояние запланированных контактов и сделок;
- проводить интегрированный BCG-анализ отношений с клиентами;
- регистрировать и оперативно обрабатывать претензии клиентов;
- анализировать и оценивать эффективность работы менеджеров с клиентами.

Автоматизированные механизмы управления отношениями с клиентами можно использовать не только как эффективное орудие в конкурентной борьбе за выгодных клиентов. Предоставляется хранилище информации о клиентах предприятия. Сотрудник, принявший звонок незнакомого ему клиента, может прямо в процессе телефонного разговора войти в курс дела, оперативно пролистав сведения о клиенте и о последних контактах с ним на экране своего компьютера (досье клиента).

Возможности конфигурации по управлению отношениями с клиентами могут быть востребованы коммерческим директором, директором по маркетингу, сотрудниками маркетинговых, сбытовых и снабженческих подразделений предприятия.

Правила продаж.

Программа позволяет задавать правила продаж, индивидуальные для клиентов или типовые для сегментов клиентов.

К ним относятся различные условия по оплате, ценовые условия, условия по скидкам.

Совместное использование типовых соглашений и сегментов позволяет быстро менять правила продаж и те условия, при которых они предоставляются:

- Для изменения правил продаж конкретному партнеру достаточно переместить его в другой сегмент, причем это может быть сделано автоматически, согласно заданным правилам.
- Для изменения правил продаж для всех клиентов из определенного сегмента – достаточно один раз изменить типовое соглашение, используемое для клиентов данного сегмента.

Программа контролирует соблюдение правил продаж: продажа с отклонением от установленных правил может быть совершена только после дополнительного согласования с руководителем. Для важных клиентов могут быть оформлены индивидуальные соглашения о правилах продаж, смотрите рисунок 1.15

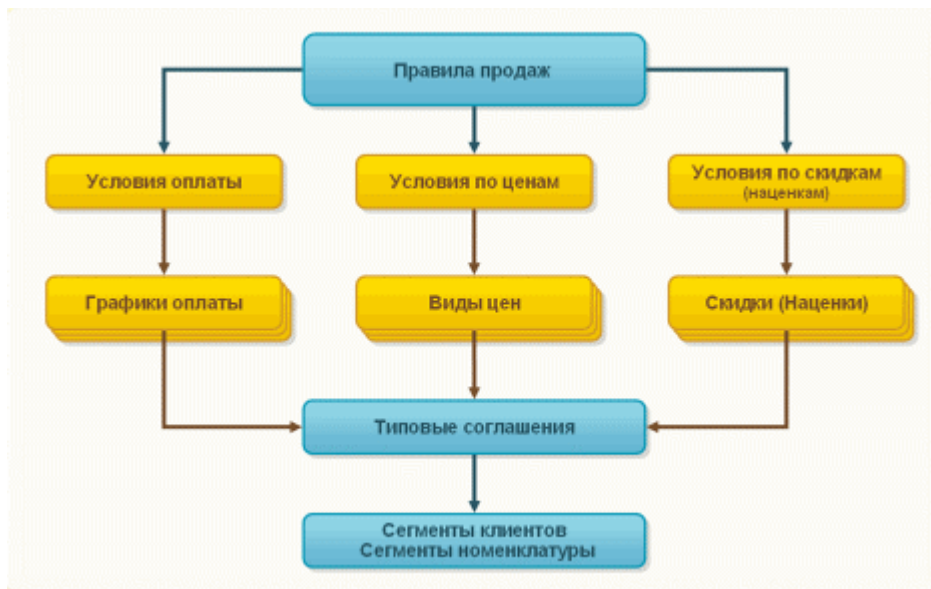


Рисунок 1.15 – Правила продаж

Автоматизация продаж.

В программе полностью отслеживаются процессы продаж. При этом продажи можно оформлять как по полному циклу оформления продажи (начиная с момента согласования коммерческого предложения на основании данных о первичном интересе клиента в оформленной с ним сделке), так и без предварительно оформленных коммерческих предложений и заказов клиентов, по так называемой «упрощенной схеме». Возможные схемы документооборота продаж смотрите на рисунке 1.16



Рисунок 1.16 – Схемы документооборота

Управление торговыми представителями.

В программе реализована функциональность по управлению торговыми представителями (торговыми агентами): закрепление клиентов за агентами, назначение графика посещения, планирование посещений клиентов, формирование детальных заданий по сбору заказов и решению иных произвольных вопросов. По результатам отработанных заданий в информационной базе могут быть автоматически созданы заказы клиентов. Также фиксируется информация о расходах торгового представителя (авансовые отчеты) и собранная информация о клиентах. Смотрите рисунок 1.17.

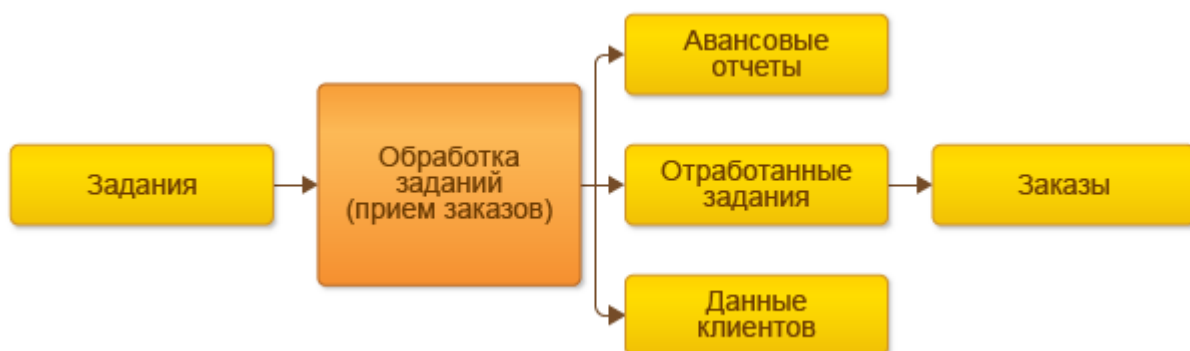


Рисунок 1.17 – Обработка заданий

Предусмотрена также возможность самостоятельной работы торговых представителей без оформления предварительных заданий. Торговым представителям передается информация о товарах и клиентах торгового предприятия. После работы с клиентами торговый представитель фиксирует оформленные заказы, уточняет сведения о клиентах или вводит данные о новых клиентах, фиксирует данные о расходах. Смотрите рисунок 1.18.

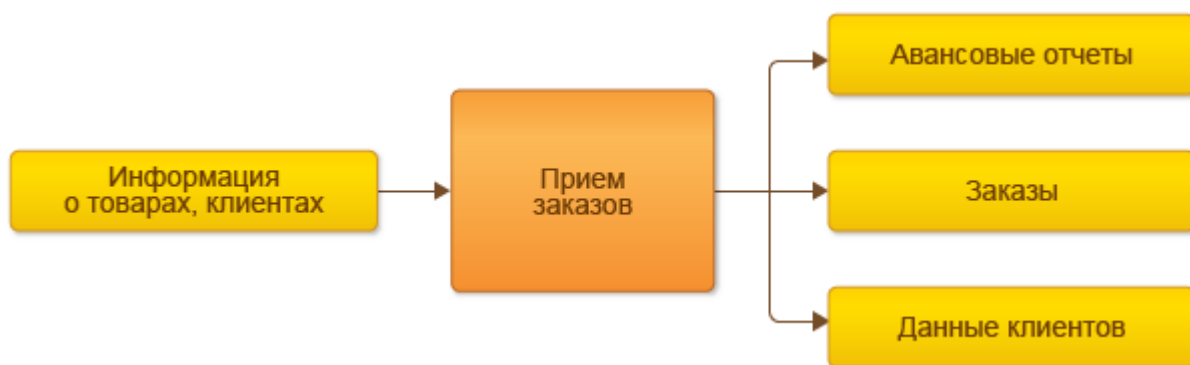


Рисунок 1.18 – Прием заказов

Анализ деятельности торговых представителей производится с помощью различных отчетов: можно сравнить деятельность торговых представителей с

точки зрения эффективности их работы, а также провести план – фактный анализ деятельности торговых представителей.

Склад.

В прикладном решении реализован детальный оперативный учет товаров на складах. Обеспечивается полный контроль запасов товаров на предприятии.

Прикладное решение позволяет:

- управлять остатками товаров в различных единицах измерения на множестве складов
- вести отдельный учет собственных товаров, товаров, принятых и переданных на реализацию
- детализировать расположение товара на складе по местам хранения, что позволяет оптимизировать сборку товаров на складе
- учитывать серии товаров (серийные номера, сроки годности и т.д.)
- задавать произвольные характеристики товаров (цвет, размер и т.д.)
- учитывать ГТД и страну происхождения
- оформлять операции сборки/разборки товаров
- резервировать товары

Организация складского хозяйства может быть различной, структура может быть, как простой, так и достаточно иерархически сложной. Склады или места хранения могут быть как на территории предприятия, так и расположены удаленно. Смотрите рисунок 1.19.

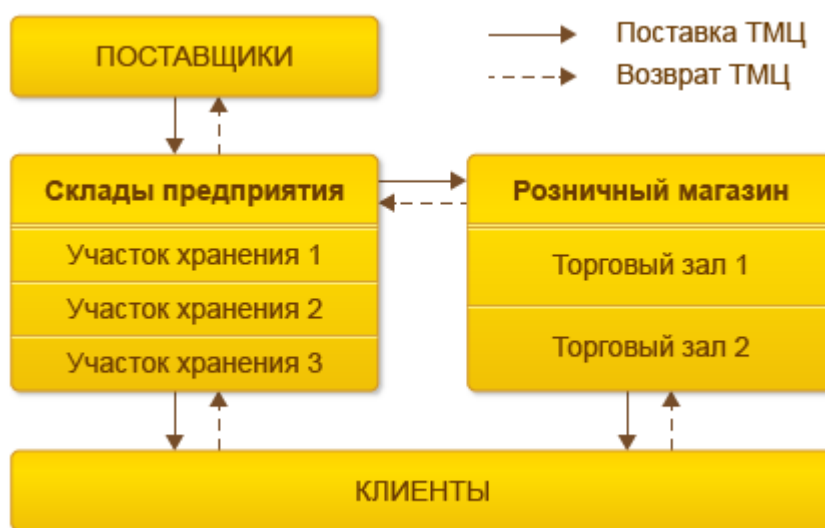


Рисунок 1.19 – Организация складского хозяйства

Сведения о складских запасах могут вводиться в информационную систему с высокой степенью детализации: до уровня характеристик товаров (цвет, размер, габариты и т.д.), до уровня серийных номеров и сроков годности товаров.

Для ускорения процессов поступления, отгрузки и инвентаризации товаров на складах можно использовать различные виды торгового оборудования: сканеры штрих-кодов, терминалы сбора данных.

В программе реализована схема ордерного складского учета. Она может независимо включаться для операций отгрузки и поступления. Складские ордера выписываются на основании документов-распоряжений, в качестве которых могут выступать заказы или накладные. Ведется учет полученных, но невыполненных распоряжений; технология работы склада может строиться целиком «от электронных распоряжений».

В программе реализовано адресное хранение товаров, т.е. ведение остатков товаров в разрезе мест хранения (ячеек, полок, стеллажей), и упаковок товара.

При этом возможно, как справочное размещение по ячейкам, когда указывается только в каких ячейках товар в принципе может находиться, так и контроль остатков по ячейкам, когда ведется точный учет по количеству товара в каждой ячейке.

При использовании контроля остатков в складских ячейках: система позволяет управлять раскладкой товара по местам хранения при поступлении, сборкой с мест хранения при отгрузке, перемещением и разупаковкой товара. Алгоритмы, заложенные в систему, автоматически подбирают оптимальные места хранения при размещении товаров. Для товаров, которые хранятся и отгружаются в упаковках разного размера – возможно автоматическое формирование заданий на разупаковку, при дефиците упаковок меньшего размера.

В любой момент можно проконтролировать процесс сборки товаров на складе в соответствии с выданными "электронными" распоряжениями или процесс размещения поступившего товара.

Реализован многошаговый процесс инвентаризации товара, включающий формирование распоряжений на инвентаризацию, выдачу распоряжений на пересчет остатков в местах хранения, раздельное отражение излишков и недостач в оперативном и финансовом учете.

1.2.6 Механизм запросов

Это один из способов доступа к данным, которые поддерживает платформа. Используя этот механизм, разработчик может читать и обрабатывать данные, хранящиеся в информационной базе; изменение данных с помощью запросов невозможно. Это объясняется тем, что запросы специально предназначены для быстрого получения и обработки некоторой выборки из больших массивов данных, которые могут храниться в базе данных.

Табличный способ доступа к данным.

Запросы реализуют табличный способ доступа к данным, которые хранятся в базе данных. Это означает, что все данные представляются в виде совокупности, связанных между собой таблиц, к которым можно обращаться

как по–отдельности, так и к нескольким таблицам во взаимосвязи. Смотрите рисунок 1.20.

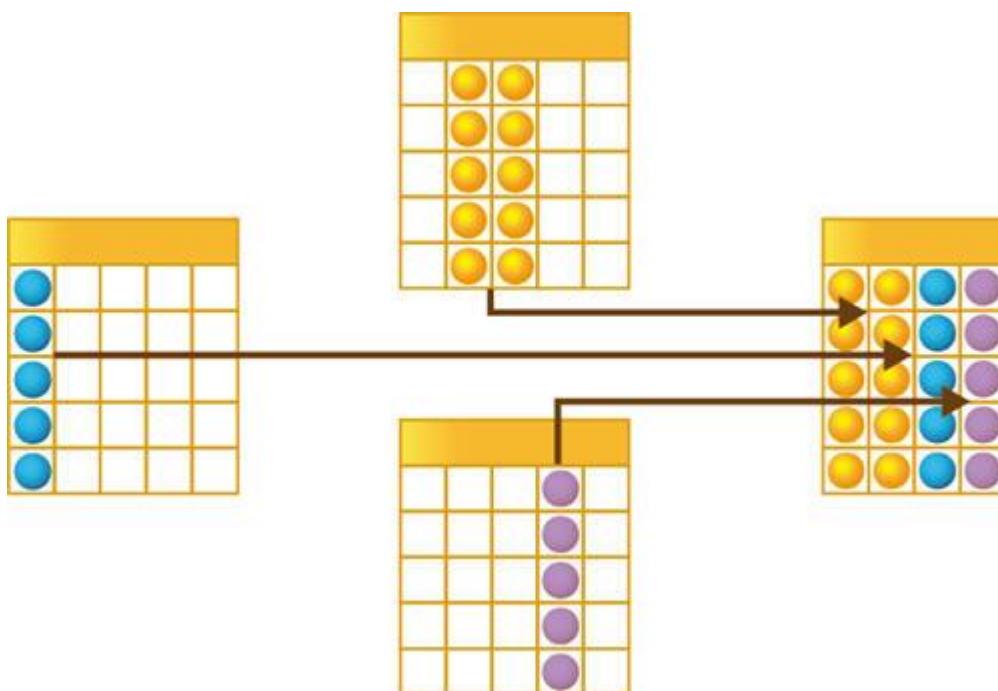


Рисунок 1.20 – Табличный способ доступа к данным

Такой способ работы с данными позволяет получать сложные выборки данных, сгруппированные и отсортированные определенным образом. Для этих выборок могут быть рассчитаны общие и промежуточные итоги, наложены ограничения на количество или состав записей и пр.

Язык запросов

Для того чтобы разработчик имел возможность использовать запросы для реализации собственных алгоритмов, в платформе реализован язык запросов. Этот язык основан на SQL, но при этом содержит значительное количество расширений, ориентированных на отражение специфики задач и на максимальное сокращение усилий по разработке прикладных решений. Можно перечислить наиболее существенные возможности, реализуемые языком запросов:

Обращение к полям через точку («.»).

Если поля какой–либо таблицы имеют ссылочный тип (хранят ссылки на объекты другой таблицы), разработчик может в тексте запроса сослаться на них через «.» , при этом количество уровней вложенности таких ссылок система не ограничивает. Смотрите рисунок 1.21

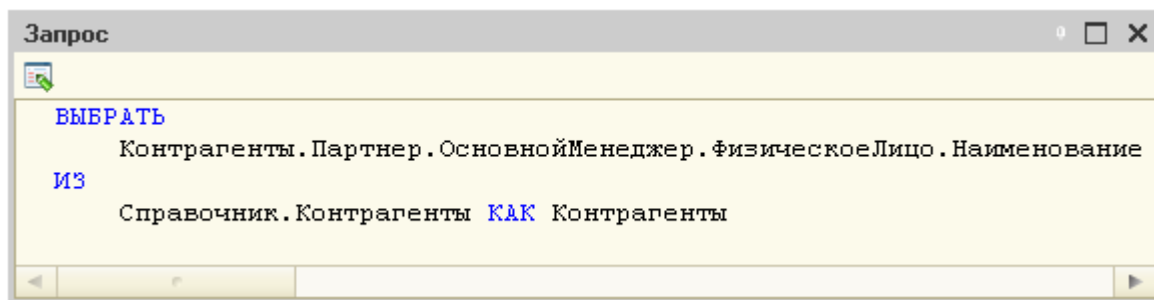


Рисунок 1.21 – Обращения к полям через точку

Обращение к вложенным таблицам (табличным частям документов и элементов справочников).

Система поддерживает обращения к вложенным табличным частям и как к отдельным таблицам, и как к целым полям одной таблицы. Например, при обращении к документу Реализация товаров (содержащему табличную часть Товары с составом отгружаемых товаров), мы можем считать табличную часть как отдельную таблицу, смотрите рисунок 1.22.

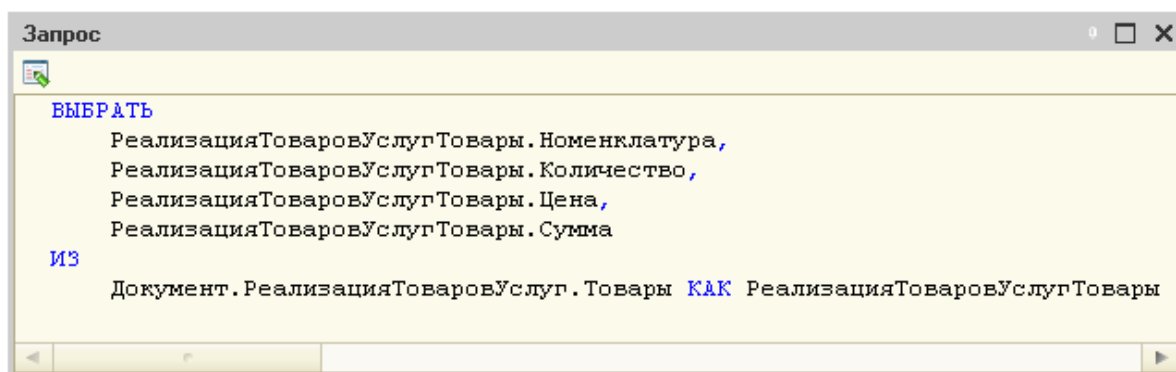


Рисунок 1.22 – Обращение к вложенным таблицам

Но также мы можем считать заголовочную запись документа, в которой значением поля Товары будут все записи вложенной таблицы, подчиненные этому объекту (документу):

Автоматическое упорядочивание.

Для выбора наиболее правильного («естественного») порядка вывода информации на экран или в отчет разработчику в большинстве случаев достаточно задать режим автоматического упорядочивания. Смотрите рисунок 1.23.

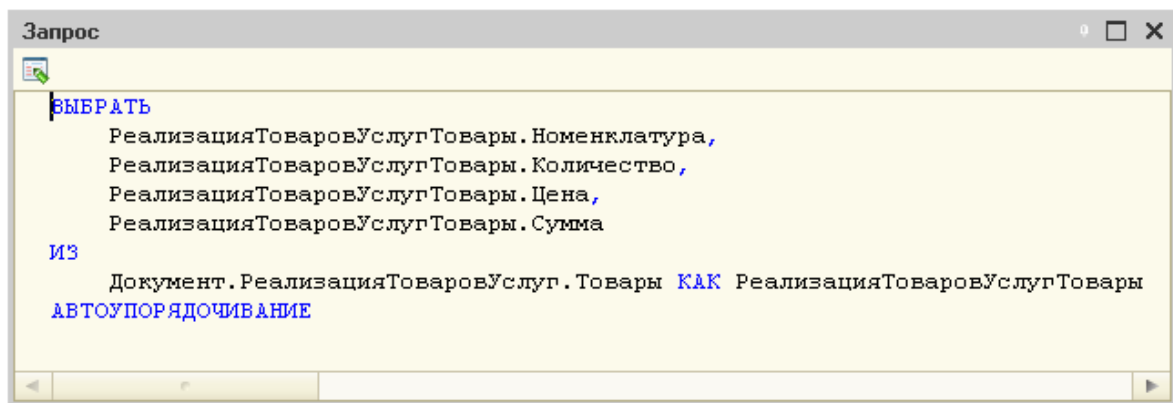


Рисунок 1.23 – Автоматическое упорядочивание

Многомерное и многоуровневое формирование итогов.

Итоги и подитоги формируются с учетом группировки и иерархии, обход уровней может выполняться в произвольном порядке с подведением подитогов, обеспечивается корректное построение итогов по временным измерениям. Смотрите рисунок 1.24

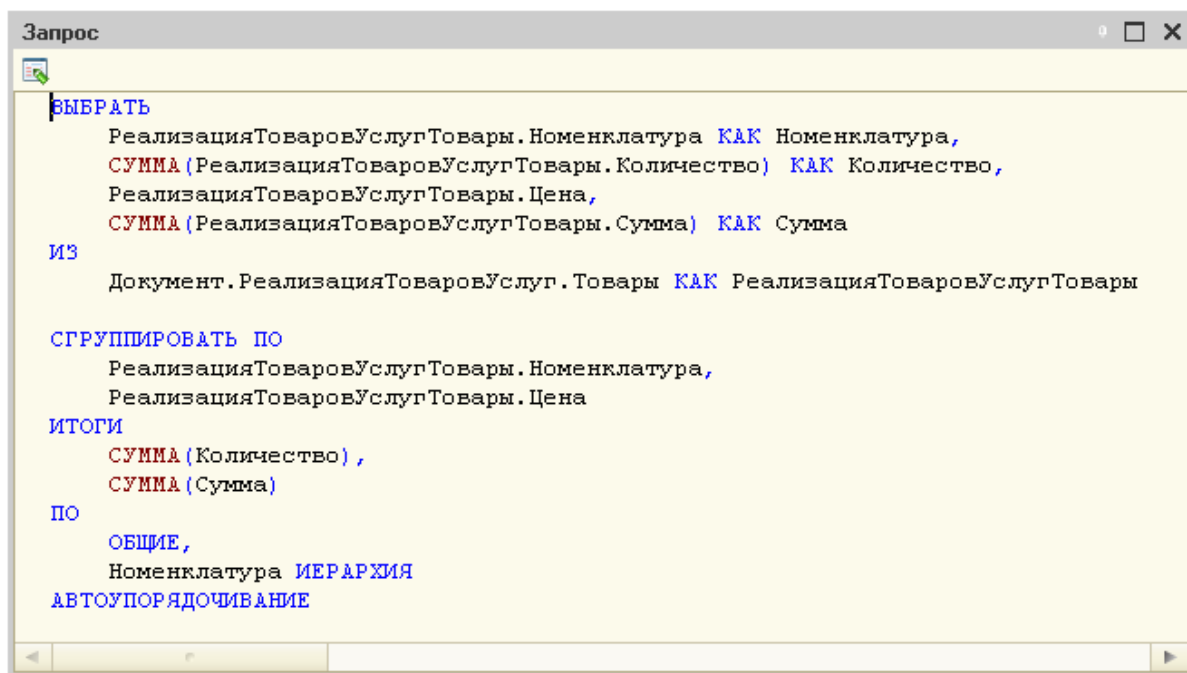


Рисунок 1.24 – Многомерное и многоуровневое формирование итогов

Поддержка виртуальных таблиц.

Виртуальные таблицы, предоставляемые системой, позволяют получить практически готовые данные для большинства прикладных решений без необходимости составления сложных запросов. Например, такая виртуальная таблица может предоставить данные по остаткам товаров в разрезе периодов на какой-то момент времени. Смотрите рисунок 1.25.

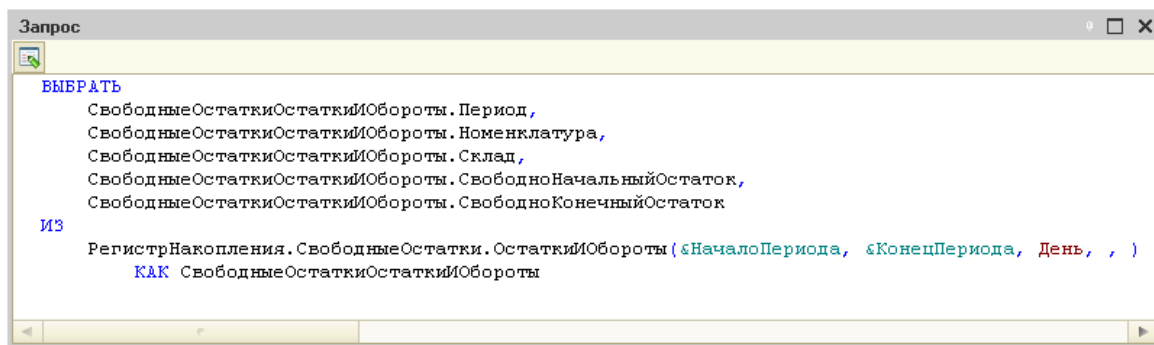


Рисунок 1.25 – Поддержка виртуальных таблиц

Пакетные запросы.

Для более удобной работы с временными таблицами в языке запросов поддерживается работа с пакетными запросами – таким образом, создание временной таблицы и ее использование помещаются в один запрос. Пакетный запрос представляет собой последовательность запросов, разделенных символом ";". Запросы исполняются один за другим. Результатом выполнения пакетного запроса в зависимости от используемого метода будет являться либо результат, возвращаемый последним запросом пакета, либо массив результатов всех запросов пакета в той последовательности, в которой следуют запросы в пакете. Смотрите рисунок 1.26.

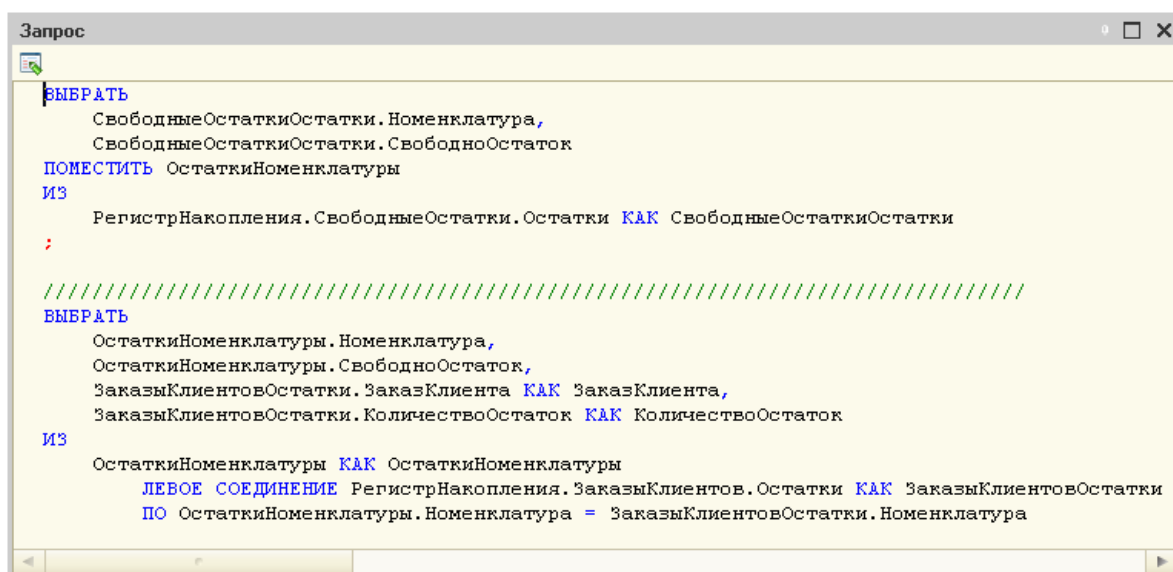


Рисунок 1.26 – Пакетные запросы

2 Инструменты разработки

2.1 Установка и настройка 1С Предприятия 8

В большинстве случаев для установки 1С Предприятия 8.x в варианте «клиент–сервер» достаточно запуска программы установки 1С Предприятия 8.x. При этом сервер 1С Предприятия получает стандартные значения параметров, необходимые для его нормального функционирования.

Рассмотрим установку сервера 1С Предприятия более детально. В процессе установки сервера 1С Предприятия 8.x программа установки 1С Предприятия 8.x выполняет следующие действия.

Копирует загрузочные модули сервера 1С Предприятия в каталог, указанный программе установки 1С Предприятия в качестве конечной папки.

Если в процессе установки выбрано «Создать пользователя USR1CV81», то создает пользователя USR1CV81. От имени этого пользователя работает сервер 1С Предприятия 8.1, если он запускается как сервис. Ему доступны только те ресурсы, которые необходимы серверу 1С Предприятия. Важно, что серверу 1С Предприятия для работы необходимы два каталога: общий каталог с данными сервера (обычно "C:\Program Files\1cv81\server") и каталог временных файлов (обычно "C:\Documents and Settings\usr1cv81\Local Settings\Temp" или "C:\WINNT\Temp"). Пользователь USR1CV81 получает права на общий каталог с данными сервера. Каталог временных файлов обычно доступен всем пользователям.

Если в процессе установки включено «Установить сервер 1С Предприятия 8.1 как сервис Windows», то регистрирует в Windows сервис агента сервера 1С Предприятия и запускает его. При первом запуске создается кластер серверов 1С Предприятия с настройками по умолчанию. В нем один рабочий сервер и один рабочий процесс. Адрес рабочего сервера совпадает с именем компьютера, на котором выполнена установка.

2.2 Пользователь USR1CV81 или USR1CV82 и его права

Сервер 1С Предприятия является серверным приложением работа которого не должна зависеть от того, какой пользователь вошел в серверный компьютер в интерактивном режиме, если вообще кто–нибудь вошел. Поэтому при установке сервера 1С Предприятия желательно создать специального пользователя USR1CV81, наделенного правами, минимально необходимыми для сервера 1С Предприятия, и не предназначенного для интерактивного входа. Сервер 1С Предприятия представляется системе Windows пользователем USR1CV81.

Рассмотрим подробнее права, устанавливаемые пользователю USR1CV81. Сервер 1С Предприятия использует следующие каталоги:

– Каталог загрузочных модулей находится в каталоге, заданном программе установки 1С Предприятия в качестве конечной папки. В нем расположены загрузочные модули сервера 1С Предприятия. Пользователю USR1CV81 необходимы права на чтение данных и запуск программ из этого каталога и его подкаталогов. Он получает эти права неявно, благодаря включению в группу Users.

– Каталог данных сервера обычно имеет имя "C:\Program Files\1cv81\server". Пользователю USR1CV81 необходимы полные права на этот каталог. Программа установки 1С Предприятия при создании пользователя USR1CV81 наделяет его правами на этот каталог.

– Каталог временных файлов обычно имеет имя "C:\Documents and Settings\usr1cv81\Local Settings\Temp" или "C:\WINNT\Temp", которое определяется значением переменной TEMP окружения пользователя или переменной TEMP системного окружения.

Посмотреть значение этой переменной можно в диалоге System Properties (Start-> Settings-> Control Panel-> System-> Advanced-> Environment Variables). Программа установки 1С Предприятия задает пользователю USR1CV81 полные права на этот каталог. Обычно при установке Windows каталог временных файлов доступен всем пользователям посредством включения в его список доступа группы CREATOR OWNER. Однако, это доступ не полный. В частности, всем пользователям не доступна операция поиска файлов в этом каталоге. Установка пользователю USR1CV81 полных прав на каталог временных файлов позволяет серверу 1С Предприятия выполнять все необходимые ему операции. Посмотреть список доступа можно в диалоге свойств каталога на закладке Security. Наличие группы CREATOR OWNER позволяет обращаться к каталогу любому пользователю, создающему какие-нибудь файлы в этом каталоге или владеющему какими-нибудь файлами в этом каталоге. При этом в списке доступа созданного файла вместо группы CREATOR OWNER будет записан пользователь, создавший файл. Среди пользователей, которым разрешен доступ в этот каталог, должен быть и пользователь USR1CV81, наделенный полными правами на этот каталог. Важно иметь в виду, что каталог временных файлов определенного пользователя (в том числе и пользователя USR1CV81) определяется комбинацией переменных окружения этого пользователя и системных переменных окружения. Чтобы узнать этот каталог, программа установки 1С Предприятия запрашивает контекст пользователя USR1CV81. В Windows 2000 пользователю, от имени которого запускается программа установки 1С Предприятия, могут потребоваться привилегии: Act as part of the operating system и Bypass traverse checking. Проверить привилегии пользователя можно утилитой Local Security Settings в ветке Local Policies-> User Rights Assignment. В процессе установки нового программного обеспечения программа установки обычно получает эти привилегии автоматически.

2.3 Регистрация сервера 1С Предприятия как сервиса Windows

Сервер 1С Предприятия является простым консольным Windows приложением и может быть запущен интерактивно. Однако для постоянного использования это неудобно, поскольку ставит запуск сервера 1С Предприятия от входа итерактивного пользователя в серверный компьютер. Чтобы исключить эту зависимость, сервер 1С Предприятия может запускаться как сервис Windows. Для этого он должен быть зарегистрирован в менеджере сервисов Windows.

Для просмотра списка сервисов Windows и их параметров предназначена утилита Component Services (Start-> Settings-> Control Panel-> Administrative Tools-> Services). Сервер 1С Предприятия представлен в списке сервисов сервисом "Агент сервера 1С Предприятия 8.1". Параметры сервиса определяют запуск процесса "Агент сервера 1С Предприятия" (ragent), пользователя, от имени которого он запускается, а также способ перезапуска в аварийных ситуациях.

В диалоге свойств сервиса "Агент сервера 1С Предприятия 8.1" на закладке General показана строка запуска процесса ragent, который является Агентом сервера 1С Предприятия. Обычно эта строка имеет вид:

```
"C:\Program Files\1cv81\bin\ragent.exe"-srvc-agent-regport 1541-port 1540-range 1560:1591-d "C:\Program Files\1cv81\server"
```

В ней указано, процессом Агента сервера является загрузочный модуль "C:\Program Files\1cv81\bin\ragent.exe", процесс ragent запускается как сервис Windows и должен управляться менеджером сервисов (-srvc), используется как Агент сервера 1С Предприятия (-agent), при первом запуске сервиса должен быть создан кластер с параметрами по умолчанию и главным IP-портом номер 1541 (-regport 1541). По этому порту клиентские приложения должны соединяться с информационными базами, зарегистрированными в кластере, IP-порт агента сервера должен иметь номер 1540 (-port 1540). По этому порту Консоль кластера должна соединяться с центральным сервером для выполнения административных функций при запуске процессов кластера на данном сервере им будут динамически назначаться IP-порты из диапазона 1560-1591 (-range 1560:1591).

Общие данные кластера будут размещены в каталоге "C:\Program Files\1cv81\server" (-d "C:\Program Files\1cv81\server"). Сервис "Агент сервера 1С Предприятия 8.1" может быть добавлен или удален не только при установке или удалении 1С Предприятия программой установки 1С Предприятия 8.1, но и вручную. Для этого можно исполнить из командной строки утилиту ragent, указав ей соответствующие параметры. Для создания сервиса нужно указать параметр -instsrvc и параметры: -usr- имя пользователя, от имени которого должен быть запущен сервис, -pwd- пароль этого пользователя. При этом остальные параметры станут параметрами строки запуска Агента сервера 1С Предприятия как сервиса. Например, для стандартной регистрации сервиса

Агента сервера 1С Предприятия в отладочном режиме набор параметров должен быть таким:

```
"C:\Program Files\1cv81\bin\ragent.exe"-instsrvc-usr .\USR1CV81-pwd  
Password-regport 1541-port 1540-range 1560:1591-d "C:\Program  
Files\1cv81\server"-debug
```

Для удаления сервиса нужно указать параметр-rmsrvc. Например

```
"C:\Program Files\1cv81\bin\ragent.exe"-rmsrvc
```

Иногда бывает полезно изменить строку запуска Агента сервера или другие параметры сервиса Агента, например, включить режим отладки, или создать несколько сервисов разных версий. Диалог свойств сервиса не позволяет редактировать строку запуска сервисного приложения и некоторые другие параметры, например, идентификатор сервиса. Для редактирования потребуется утилита regedit, предназначенная для просмотра и редактирования системного реестра Windows.

Запустите утилиту regedit (откройте Start-> Run и наберите regedit) и выберите ветку.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\1C:Enterpris  
e 8.1 Server Agent
```

Среди ее параметров есть параметр ImagePath, значением которой является строка запуска Агента сервера 1С Предприятия. Здесь можно добавить новые параметры строки запуска или поменять значения существующих. Полный список возможных параметров приведен в книге "1С Предприятие 8.1 Клиент-сервер" документации.

При необходимости регистрации нескольких независимых сервисов Агента сервера 1С Предприятия нужно указать им разные загрузочные модули, разные порты и разные каталоги данных кластера. Еще требуется зарегистрировать их с разными идентификаторами сервисов.

Создать первый сервис

```
"C:\Program Files\1cv81\bin\ragent.exe"-srvc-agent-regport 1541-port  
1540-range 1560:1591-d "C:\Program Files\1cv81\server"
```

При помощи утилиты regedit изменить идентификатор зарегистрированного сервиса, для этого выбрать ветку

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\1C:Enterpris  
e 8.1 Server Agent
```

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\1C:Enterpris  
e 8.1 Server Agent First
```

Создать второй сервис

```
"C:\Program Files\1cv81_10\bin\ragent.exe"-srvc-agent-regport 1641-port  
1640-range 1660:1691-d "C:\Program Files\1cv81_10\server"
```

Быть может, его идентификатор тоже изменить. Для этого: выбрать ветку

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\1C:Enterpris  
e 8.1 Server Agent
```

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\1C:Enterpris  
e 8.1 Server Agent Second
```


Как уже говорилось, программа установки 1С Предприятия копирует загрузочные модули 1С Предприятия и выполняет необходимую регистрацию в СОМ и в менеджере сервисов Windows. Выше приведена информация, необходимая для понимания внутренних механизмов этой регистрации. Если на серверном компьютере установлен не только сервер, но и клиентская часть 1С Предприятия, то она готова к работе сразу после установки (и подключения ключей защиты).

Чтобы сервер 1С Предприятия был доступен с других компьютеров в локальной сети, необходимо проверить сетевые настройки на серверном и клиентском компьютере, а также для сети в целом. Для передачи данных между клиентскими приложениями и сервером 1С Предприятия, а также между процессами кластера серверов используется TCP/IP. От правильности его настройки зависит работа 1С Предприятия в варианте клиент–сервер.

Процессы кластера серверов 1С Предприятия соединяются друг с другом по адресам, определенным в качестве значений свойства "Компьютер" диалога свойств рабочих серверов. Для кластера необходимо, чтобы значением свойства "Компьютер" был либо IP–адрес в точечной нотации, либо такой символический адрес, по которому может быть определен IP адрес при помощи функции `gethostbyname`, определенной в программном интерфейсе протокола TCP. Определение IP–адреса выполняется либо на основании локальной таблицы символических адресов (`C:\WINNT\system32\drivers\etc\hosts`), либо по таблицам адресов в доступных DNS серверах. Если по символическому адресу рабочего сервера его IP–адрес не определяется или определяется неправильно (например, IP–адрес не совпадает с фактическим IP–адресом данного компьютера), то кластер работать не будет. Важно, чтобы имена компьютеров и их адреса, определенные в Windows на каждом из рабочих серверов кластера, не противоречили их именам в DNS.

На каждом рабочем сервере процессы кластера используют следующие порты: IP порт рабочего сервера (обычно 1540); IP порты из диапазонов IP портов рабочего процесса (обычно 1560–1591). Кроме того, на центральном сервере кластера используется порт кластера (обычно 1541). Если в системе используются сетевые экраны, то передача данных по этим портам должна быть разрешена. Вместо разрешения портов из приведенного списка можно разрешить передачу данных процессам кластера (`ragent`, `rmngr`, `rphost`).

Соединение клиентского приложения 1С Предприятия с сервером выполняется в 2 этапа. Сначала оно устанавливает соединение с менеджером кластера. При этом используется адрес центрального сервера (символический или числовой) и порт кластера (обычно 1541). Далее клиентское приложение устанавливает соединение с одним из рабочих процессов. В качестве его адреса используется значение свойства «Компьютер» соответствующего рабочего сервера и порт рабочего процесса, который выбирается из диапазона IP портов рабочего сервера. Передача данных на эти порты должна быть разрешена во всех сетевых экранах на маршруте от компьютера клиентского приложения до компьютеров кластера серверов 1С Предприятия. Определение IP адреса

серверных процессов выполняется при помощи функции `gethostbyname` на компьютере клиента. Важно, чтобы имена центрального и рабочих серверов и их адреса, определенные в Windows на каждом из серверов кластера, не противоречили их именам в DNS, доступном компьютеру клиента.

И последнее. Очевидно, что для успешного доступа к серверу 1С Предприятия с других компьютеров он должен быть в сети и должны быть выполнены необходимые для этого настройки. Подключение к сети и методика настройки относятся к администрированию сетей на базе Microsoft Windows и описаны в соответствующих инструкциях.

1С Предприятие в варианте «клиент–сервер» использует для хранения данных SQL–сервер. При этом к SQL–серверу обращается только Сервер 1С Предприятия. Клиенты 1С Предприятия непосредственного доступа к SQL–серверу не имеют. Установка и настройка SQL–сервера подробно описана в документации по Microsoft SQL Server. Для успешной работы Сервера 1С Предприятия с SQL–сервером необходимо обратить особое внимание на следующие настройки.

Необходимые компоненты SQL–сервера. Для доступа к SQL–серверу со стороны Сервера 1С Предприятия на компьютере Сервера 1С Предприятия должны быть установлены компоненты Microsoft Data Access 2.6 или более поздний.

Аутентификация пользователя SQL–сервером. Права доступа к базам данных SQL–сервера определяются пользователем, от имени которого происходит обращение к базам данных. С компьютера, на котором установлен SQL–сервер, запустим утилиту SQL Server Enterprise Manager, найдем узел Local (Console Root→ Microsoft SQL Servers→ SQL Server Group→ (Local)) и откроем его свойства. На закладке Security можно видеть, что SQL–сервер поддерживает два способа аутентификации пользователей: SQL Server and Windows и Windows only. Аутентификация Windows позволит Серверу 1С Предприятия обращаться к SQL–серверу только от имени пользователя `USR1CV81`, что не позволяет различать права доступа до различных информационных баз, обслуживаемых одним сервером 1С Предприятия. Рекомендуется выбирать режим SQL Server and Windows. В этом случае обращение к конкретной информационной базе будет выполняться от имени пользователя, который задан в качестве пользователя SQL–сервера при создании данной информационной базы. Важно, что этот пользователь должен иметь не только полные права на базу данных информационной базы, но и права на создание баз данных в SQL–сервере и на чтение таблиц базы данных Master.

Сетевые протоколы для доступа к SQL–серверу. Если Сервер 1С Предприятия и SQL–сервер размещены на разных компьютерах, то необходимо выполнить настройки сетевых протоколов доступа к SQL–серверу. Это можно сделать при помощи утилиты SQL Server Client Network Utility. На закладке General можно выбрать список сетевых протоколов, используемых для доступа к SQL–серверу. Наиболее быстрым и универсальным является использование

протокола TCP/IP. При использовании других протоколов необходимо иметь в виду, что некоторые из них, например Named Pipes, выполняют дополнительную аутентификацию средствами Windows при обмене данными с SQL-сервером. В этом случае для успешной работы с SQL-сервером на компьютере с SQL-сервером должен быть зарегистрирован пользователь USR1CV81, наделенный соответствующими правами. Протокол доступа к данному SQL-серверу может быть изменен на закладке Alias.

3. Этапы разработки конфигурации

3.1 Описание автоматизируемых процессов

При проектировании конфигурации важно, чтобы ее функционал отвечал реальным потребностям предприятий. Поэтому важно очертить тот круг процессов, которые позволяет автоматизировать информационная система.

СППР позволяет зафиксировать перечень автоматизируемых процессов, процессы при этом могут быть сгруппированы по усмотрению пользователя. Смотрите рисунок 3.1.

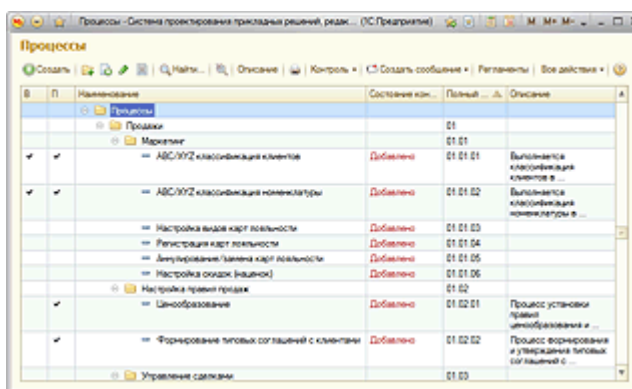


Рисунок 3.1 – Группировка процессов

При описании процесса фиксируется его описание, отражающее суть процесса, события начала и окончания процесса.

Процесс детализируется до отдельных шагов, исполняемых конкретным исполнителем. Смотрите рисунок 3.2.

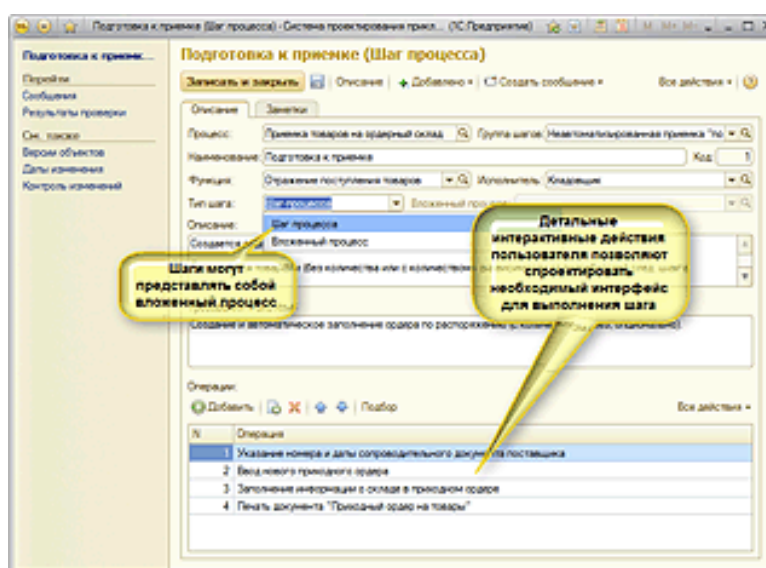


Рисунок 3.2 – Детализация процесса

3.2 Создание логической модели проектируемой системы

Логическая модель системы позволяет описать функциональность конфигурации, увязав ее с составом обрабатываемой информации и исполнителями.

Логическая модель в СППР строится с использованием методологии IDEF0. В рамках создания логической модели описываются функции системы и производится их декомпозиция. Смотрите рисунок 3.3.

Полный код	Наименование	Описание	Подсистема
1	Управление торговлей		
1.1	Торговля		
1.1.1	Продажи		Продажи
1.1.1.1	Маркетинг	Функция предназначена для: - сбора и обработки ...	Маркетинг
1.1.1.2	Оптовые продажи	Функция предназначена для автоматизации процесса оптовых прода...	Продажи
1.1.1.3	Управление сделками	Функция предназначена для: - настройки ведения сдел...	ПроведениеСделок
1.1.1.3.1	Настройка ведения сделок	Функция предназначена для определения следующих настроек ...	ПроведениеСделок
1.1.1.3.2	Проведение сделок	Функция предназначена для: - регистрации и просмот...	ПроведениеСделок
1.1.1.3.3	Анализ эффективности сделок	Функция предназначена	ПроведениеСделок

Рисунок 3.3 – Функции системы

Основой описания функции является ее IDEF схема. Схема позволяет в наглядной форме отразить взаимосвязь отдельных (дочерних) функций, потоков данных и исполнителей. Смотрите рисунок 3.4.

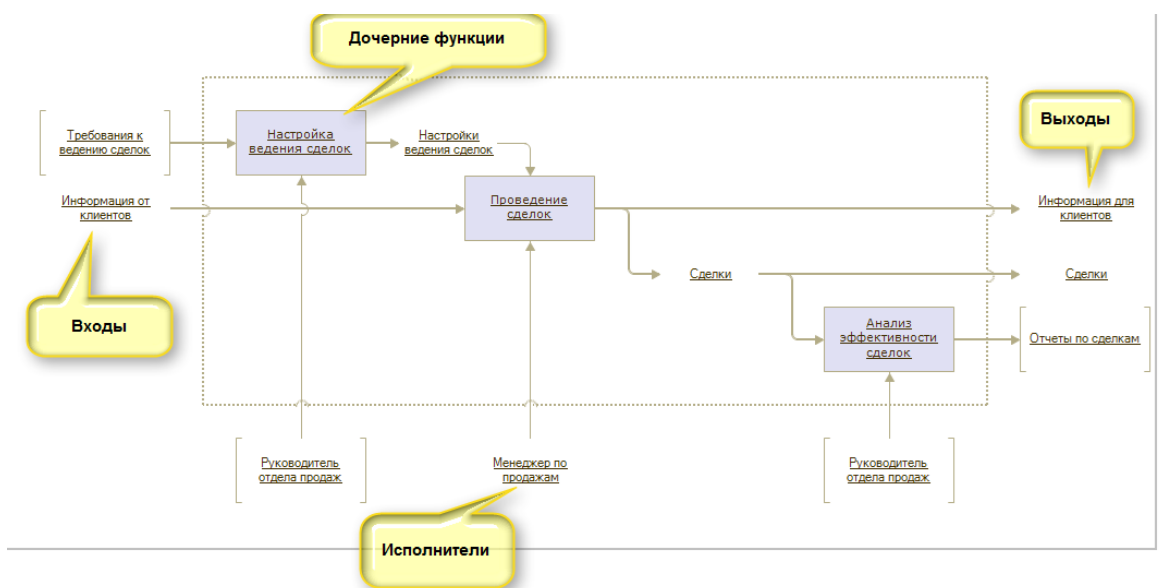


Рисунок 3.4– IDEF схема

3.3 Разработка архитектуры

Разработка архитектуры конфигурации выполняется на основе логической модели. При этом метаданные соотносятся с объектами данных, перечень которых определяется при разработке функций. Смотрите рисунок 3.5

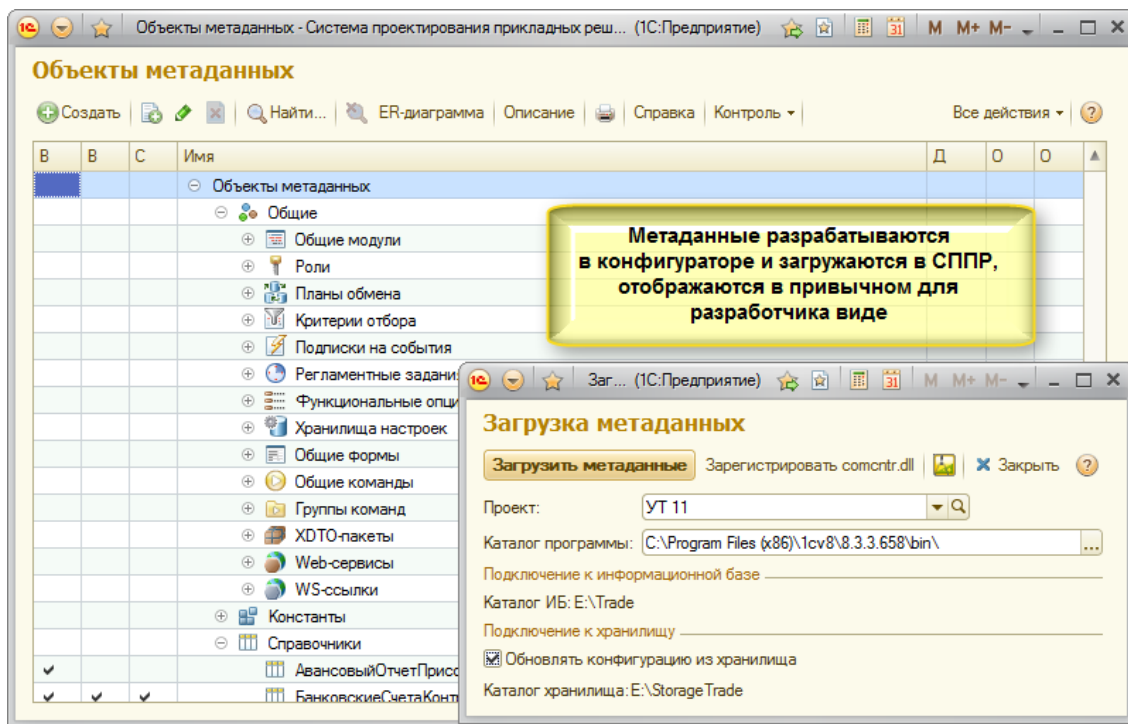


Рисунок 3.5 – Объекты метаданных

3.4 Проектирование интерактивных операций

При работе с системой в рамках того или иного процесса пользователь выполняет определенные действия, реализуя таким образом один из возможных сценариев работы.

Описание последовательностей интерактивных операций, выполняемых пользователем в системе, позволяет проанализировать, реализуем ли закладываемый в систему функционал в рамках конкретного автоматизируемого процесса. Смотрите рисунок 3.6.

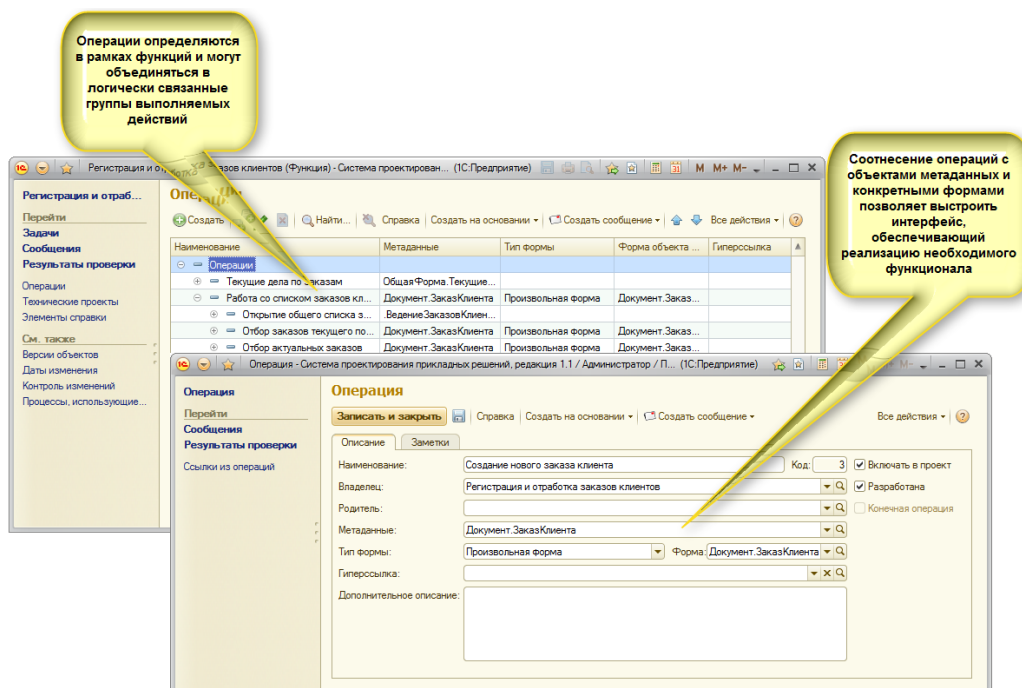


Рисунок 3.6 – Последовательность интерактивных операций

3.5 Подготовка справки

СППР позволяет автоматически формировать тексты справки для разрабатываемой конфигурации. Подготовленные тексты справки в формате html могут быть выгружены из СППР и загружены в конфигурацию штатными средствами конфигуратора.

Справка формируется в едином стиле, с использованием единой структуры описания, исходя из взаимосвязей подсистем, объектов метаданных и операций функций. Стили оформления справки (шрифты, отступы, выделения) могут настраиваться непосредственно в СППР. Смотрите рисунки 3.7, 3.8.

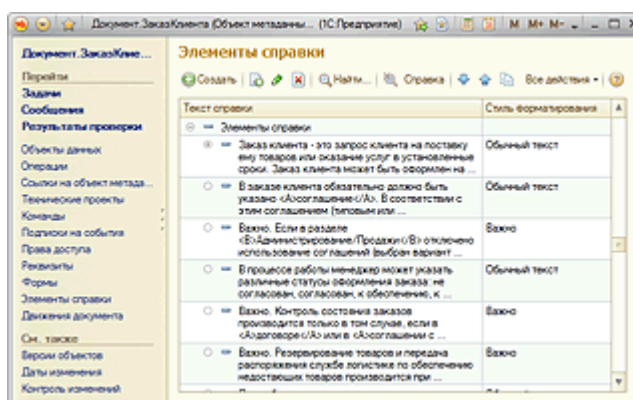


Рисунок 3.7 – Элемент справки

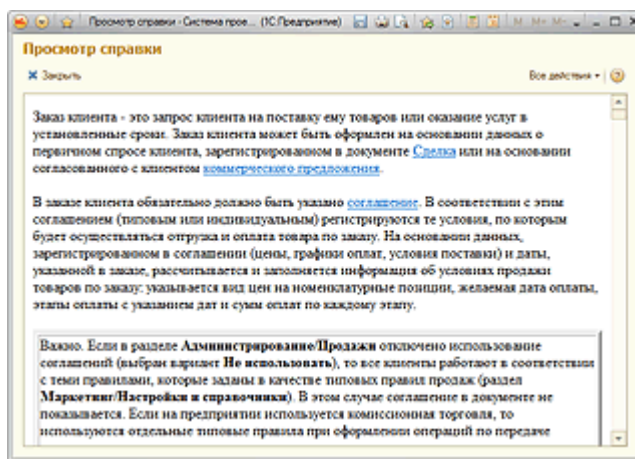


Рисунок 3.8 – Просмотр справки

3.6 Управление проектом и изменениями

Для управления проектом и изменениями в СППР используется функциональность ведения технических проектов. Данная функциональность позволяет организовать коллективную работу над проектом, с отслеживанием прохождения различных этапов проекта. При этом возможна гибкая настройка этапов, согласование этих этапов, уведомление участников команды разработки об изменениях.

Использование технических проектов обеспечивает внесение изменений в имеющийся проект таким образом, чтобы эти изменения были увязаны с логической моделью, были прозрачны и информативны для других участников проекта. Смотрите рисунок 3.9.

Технические проекты

Степень участия: Ответственный/участ... | Участник: | Статус: Активные... | Версия: Актуаль... | Все действия

Создать | Найти... | Описание | Создать на основании | Контроль

№	Наименование	Код	С	Статус	Предстоящая контрольная...	С...	Начало
0	Диагностика формирования графика	1 127	1	Активен	6, Помещение метаданны...	26.06....	13.12.2012
0	Перепланирование графика	1 128	1	Активен	6, Помещение метаданны...	26.06....	25.12.2012
0	Брак при производстве	819	1	Активен	6, Помещение метаданны...	26.06....	23.04.2013
0	Доработки бюджетирования и контроля лимитов.	1 477	1	Активен	4, Согласование разработ...	27.06....	18.06.2013
0	Доработки отражения документов в регламентированном уче...	1 699	1	Активен	3, Согласование проектн...	27.06....	18.06.2013
0	Отработка прочих входящих требований по обменам и перено...	1 700	1	Активен	5, Тестирование и аудит	27.06....	10.06.2013
0	Переработка давальческого сырья (учет у переработчика) (ча...	1 701	1	Активен	4, Согласование разработ...	27.06....	05.06.2013
0	Заполнение налоговой отчетности (имущество и т.д.)	1 702	1	Активен	4, Согласование разработ...	27.06....	19.06.2013
0	Отражение зарплаты в регламентированном учете	1 707	1	Активен	4, Согласование разработанного проекта	27.06....	27.05.2013

Рисунок 3.9 – Технические проекты

3.7 Работа с ошибками

СППР позволяет вести регистрацию ошибок по разрабатываемым проектам, в разрезе версий, сроков исправления, разделов проекта, статусов и т.д. Функционал системы предлагает готовую методику работы с ошибками, с возможностью формирования различных отчетов, публикации информации об ошибках. Система позволяет настроить связи между проектами, указать, какие проекты–библиотеки включаются в проект, с учетом конкретных версий проектов. Это позволяет получать информацию о наличии в проекте ошибок, источниками которых являются используемые библиотеки. Смотрите рисунок 3.10.

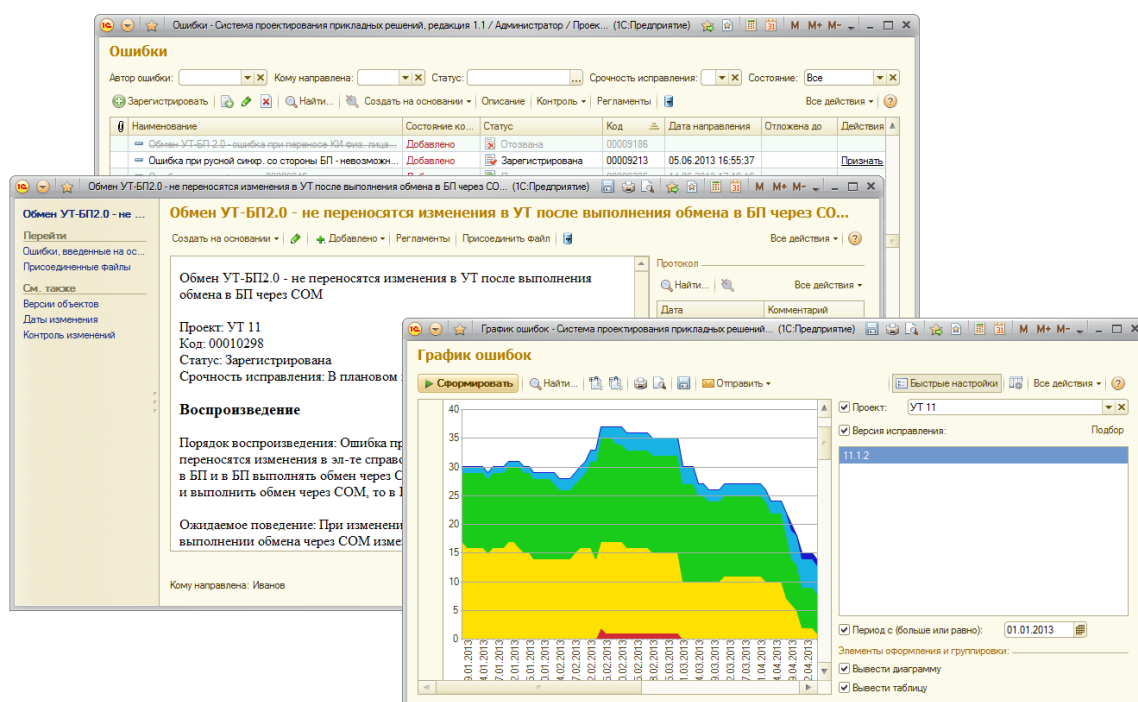


Рисунок 3.10 – Регистрация ошибок

3.8 Модуль «Общий»

В модуле «Общем» используется следующий код:

```
Функция ПолучитьЗапросПоТоварамНакладная (Док) экспорт  
Запрос = Новый Запрос;  
Запрос.Текст =  
"ВЫБРАТЬ  
| Товары.НомерСтроки,  
| Товары.Номенклатура.СтранаПроисхождения,  
| Товары.ЕдиницаИзмерения.Представление  
ЕдиницаИзмерения,  
| Товары.Цена КАК Цена,
```

КАК

```

| Товары.СуммаНДС КАК СуммаНДС,
| Товары.Сумма КАК Сумма,
| Товары.Количество КАК Количество,
| Товары.Номенклатура КАК Номенклатура,
| Товары.ПроцентСкидкиНаценки КАК ПроцентСкидки,
| Товары.СуммаСкидкиНаценки КАК СуммаСкидки
| ИЗ
| Документ.РеализацияТоваровУслуг.Товары КАК Товары
| ГДЕ
| Товары.Ссылка = &Док
|
| УПОРЯДОЧИТЬ ПО
| Номенклатура
| ИТОГИ
| СУММА (СуммаНДС) ,
| СУММА (Сумма) ,
| СУММА (Количество) ,
| КОЛИЧЕСТВО (РАЗЛИЧНЫЕ Номенклатура)
| ПО
| ОБЩИЕ
| АВТОУПОРЯДОЧИВАНИЕ";

```

```

Запрос.УстановитьПараметр ("Док", Док.Ссылка) ;
РезультатЗапроса = Запрос.Выполнить () ;
возврат РезультатЗапроса;
КонецФункции

```

```

Функция ПолучитьЗапросПоТоварамВозвратПоставщику (Док)
экспорт
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| Товары.НомерСтроки,
| Товары.Номенклатура.СтранаПроисхождения,
| Товары.ЕдиницаИзмерения.Представление          КАК
ЕдиницаИзмерения,
| Товары.Цена КАК Цена,
| Товары.СтавкаНДС,
| Товары.СуммаНДС КАК СуммаНДС,
| Товары.Сумма КАК Сумма,
| Товары.Количество КАК Количество,
| Товары.Номенклатура КАК Номенклатура
| ИЗ
| Документ.ВозвратТоваровПоставщику.Товары КАК Товары
| ГДЕ

```

```

| Товары.Ссылка = &Док
|
| УПОРЯДОЧИТЬ ПО
| Номенклатура
| ИТОГИ
| СУММА (СумманДС) ,
| СУММА (Сумма) ,
| СУММА (Количество) ,
| КОЛИЧЕСТВО (РАЗЛИЧНЫЕ Номенклатура)
| ПО
| ОБЩИЕ
| АВТОУПОРЯДОЧИВАНИЕ";

```

```

Запрос.УстановитьПараметр ("Док", Док.Ссылка) ;
РезультатЗапроса = Запрос.Выполнить () ;
возврат РезультатЗапроса ;
КонецФункции

```

Функция

```

ПолучитьЗапросПоТоварамЗаявка (Док, ИмяДокумента) экспорт
    // в запросе сразу приводим количество товара к
    единицам хранения остатков (т.е. с учетом коэффициента)
    Запрос = Новый Запрос ("ВЫБРАТЬ
    | ДокументТовары.Номенклатура КАК Номенклатура,
    | ДокументТовары.Номенклатура.СтранаПроисхождения КАК
    СтранаПроисхождения,
    | ДокументТовары.Номенклатура.Производитель КАК
    Производитель,
    // | ДокументТовары.ЕдиницаИзмерения.Представление
    КАК ЕдиницаИзмерения,
    |
    | ДокументТовары.Номенклатура.ЕдиницаХраненияОстатков.П
    редставление КАК ЕдиницаИзмерения,
    | ДокументТовары.Коэффициент КАК Коэффициент,
    |
    | ДокументТовары.СерияНоменклатурыАналитика КАК
    СерияНоменклатурыАналитика,
    |
    | ДокументТовары.Количество * ВЫБОР
    | КОГДА ДокументТовары.Коэффициент <> 0
    | И (НЕ ДокументТовары.Коэффициент ЕСТЬ
    NULL )
    | ТОГДА ДокументТовары.Коэффициент
    | ИНАЧЕ 1
    | КОНЕЦ КАК Количество,

```

```

| ДокументТовары.Цена КАК Цена,
| ДокументТовары.Сумма КАК Сумма,
|//| ДокументТовары.СуммаБезСкидок КАК
СуммаБезСкидок, // поле больше не используется- где
нужно, вычисляется
| ДокументТовары.СуммаНДС КАК СуммаНДС,
| ДокументТовары.ПроцентСкидкиНаценки КАК
ПроцентСкидки,
| ДокументТовары.ХарактеристикаНоменклатуры КАК
ХарактеристикаНоменклатуры,
| ДокументТовары.СуммаСкидкиНаценки КАК СуммаСкидки,
| КатегорииОбъектов.Категория
|ИЗ
| Документ."+ИмяДокумента+".Товары КАК ДокументТовары
| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.КатегорииОбъектов
КАК КатегорииОбъектов
| ПО ДокументТовары.Номенклатура =
КатегорииОбъектов.Объект
| ГДЕ
| ДокументТовары.Ссылка = &Док
|
|УПОРЯДОЧИТЬ ПО
| Номенклатура
|ИТОГИ
| КОЛИЧЕСТВО (РАЗЛИЧНЫЕ Номенклатура) ,
| СУММА (Сумма) ,
| СУММА (СуммаНДС)
| ПО
| ОБЩИЕ
| АВТОУПОРЯДОЧИВАНИЕ" );

```

```

Запрос.УстановитьПараметр ("Док", Док.Ссылка) ;
РезультатЗапроса = Запрос.Выполнить ();

```

```

возврат РезультатЗапроса;

```

```

КонецФункции

```

```

Функция

```

```

ПолучитьЗапросПоТоварамВнутреннийЗаказ (Док, ИмяДокумента)
экспорт

```

```

Запрос = Новый Запрос ("ВЫБРАТЬ

```

```

                                | ДокументТовары.Номенклатура
КАК Номенклатура,
                                |
                                | ДокументТовары.Номенклатура.СтранаПроисхождения   КАК
СтранаПроисхождения,
                                |
                                | ДокументТовары.Номенклатура.Производитель   КАК
Производитель,
                                |
                                | ДокументТовары.ЕдиницаИзмерения.Представление   КАК
ЕдиницаИзмерения,
                                |
                                | ДокументТовары.Количество
КАК Количество,
                                |
ДокументТовары.ХарактеристикаНоменклатуры   КАК
ХарактеристикаНоменклатуры
                                | ИЗ
                                |
Документ."+ИмяДокумента+".Товары КАК ДокументТовары
                                | ГДЕ
                                | ДокументТовары.Ссылка =
&Док
                                |
                                | УПОРЯДОЧИТЬ ПО
                                | Номенклатура
                                | ИТОГИ
                                | КОЛИЧЕСТВО (РАЗЛИЧНЫЕ
Номенклатура)
                                | ПО
                                | ОБЩИЕ
                                | АВТОУПОРЯДОЧИВАНИЕ" );

Запрос.УстановитьПараметр ("Док", Док.Ссылка) ;
                                РезультатЗапроса =
Запрос.Выполнить ( ) ;

    возврат РезультатЗапроса;

КонецФункции

Функция
ПолучитьСтрокуЯчеек (Товар, Характеристика, Серия) Экспорт

Если ЗначениеНеЗаполнено (Характеристика) Тогда

```

```

        ХарактеристикаТовара
Справочники.ХарактеристикиНоменклатуры.ПустаяСсылка ();
    Иначе
        ХарактеристикаТовара = Характеристика;
    КонецЕсли;

Запрос = Новый Запрос;

Запрос.Текст =
"ВЫБРАТЬ
| РазмещениеПоячейкамОстатки.Номенклатура,
| РазмещениеПоячейкамОстатки.СерияНоменклатуры,
| РазмещениеПоячейкамОстатки.Ячейка,
| РазмещениеПоячейкамОстатки.КоличествоОстаток
| ИЗ
| РегистрНакопления.РазмещениеПоячейкам.Остатки (
|
|     Номенклатура = &Товар
|     И     СерияНоменклатуры     =     &Серия)     КАК
РазмещениеПоячейкамОстатки
| ГДЕ
| РазмещениеПоячейкамОстатки.КоличествоОстаток > 0";

Запрос.УстановитьПараметр ("Товар", Товар);
Запрос.УстановитьПараметр ("Серия", Серия);

Результат = Запрос.Выполнить ();
Выборка = Результат.Выбрать ();
СтрокаЯчеек = "";
Пока Выборка.Следующий () Цикл
    СтрокаЯчеек = СтрокаЯчеек +
Строка (Выборка.Ячейка) + " ";
КонецЦикла;

возврат СтрокаЯчеек;

КонецФункции

Функция
ПолучитьЗапросПоСериямЗаявка (Док, ИмяДокумента) экспорт

    Запрос = Новый Запрос ("ВЫБРАТЬ
    | ДокументСерииНоменклатуры.Номенклатура     КАК
Номенклатура,

```

```

| ДокументСерииНоменклатуры.СерияНоменклатуры          КАК
Серия,

| ДокументСерииНоменклатуры.Количество * ВЫБОР
| КОГДА ДокументТовары.Коэффициент <> 0
| И (НЕ ДокументТовары.Коэффициент ЕСТЬ
NULL )
| ТОГДА ДокументТовары.Коэффициент
| ИНАЧЕ 1
| КОНЕЦ КАК Количество,

|
ДокументСерииНоменклатуры.СерияНоменклатуры.Сертифика
т КАК Сертификат,
|
ДокументСерииНоменклатуры.СерияНоменклатуры.СрокГодно
сти КАК ГоденДо,
| ДокументТовары.Коэффициент КАК Коэффициент
| ИЗ
| Документ."+ИмяДокумента+".СерииНоменклатуры          КАК
ДокументСерииНоменклатуры

| ВНУТРЕННЕЕ СОЕДИНЕНИЕ
Документ."+ИмяДокумента+".Товары КАК ДокументТовары
| ПО ДокументСерииНоменклатуры.Номенклатура =
ДокументТовары.Номенклатура

| ГДЕ
| ДокументСерииНоменклатуры.Ссылка = &Док
| И ДокументТовары.Ссылка = &Док");

Запрос.УстановитьПараметр ("Док", Док.Ссылка) ;
РезультатЗапроса = Запрос.Выполнить ();

возврат РезультатЗапроса;

КонецФункции

Функция ПолучитьЗапросПоСериямНакладная (Док) экспорт
Запрос = Новый Запрос ("ВЫБРАТЬ
| Серии.Номенклатура          КАК
Номенклатура,

```

```

        | Серии.СерияНоменклатуры КАК
Серия,
        | Серии.Количество КАК
Количество,
        |
        | Серии.СерияНоменклатуры.Сертификат КАК Сертификат,
        |
        | Серии.СерияНоменклатуры.СрокГодности КАК ГоденДо
        | ИЗ
        |
        | Документ.РеализацияТоваровУслуг.СерииНоменклатуры КАК
Серии
        | ГДЕ
        | Серии.Ссылка = &Док");

```

```

Запрос.УстановитьПараметр ("Док", Док.Ссылка) ;
        РезультатЗапроса =
Запрос.Выполнить ();

```

возврат РезультатЗапроса;

КонецФункции

Функция

ПолучитьЗапросПоСериямВозвраты (Док, ИмяДокумента) экспорт

```

        Запрос = Новый Запрос ("ВЫБРАТЬ
        | Серии.Номенклатура КАК
Номенклатура,
        | Серии.СерияНоменклатуры КАК
Серия,
        | Серии.Количество КАК
Количество,
        |
        | Серии.СерияНоменклатуры.Сертификат КАК Сертификат,
        |
        | Серии.СерияНоменклатуры.СрокГодности КАК ГоденДо
        | ИЗ
        |
        | Документ." + ИмяДокумента + ".СерииНоменклатуры КАК Серии
        | ГДЕ
        | Серии.Ссылка = &Док");

```



```
Запрос.УстановитьПараметр ("Док", Док.Ссылка) ;  
                                РезультатЗапроса =  
Запрос.Выполнить () ;
```

```
    возврат РезультатЗапроса ;
```

```
КонецФункции
```

```
Функция
```

```
ПолучитьЗапросПоСериямПриходныйОрдер (Док, ИмяДокумента)  
экспорт
```

```
    Запрос = Новый Запрос ("ВЫБРАТЬ  
                                |   Серии.Номенклатура           КАК  
Номенклатура,  
                                |   Серии.СерияНоменклатуры   КАК  
Серия,  
                                |   Серии.Количество         КАК  
Количество,  
                                |  
Серии.СерияНоменклатуры.Сертификат КАК Сертификат,  
                                |   Серии.Ячейка КАК Ячейка,  
                                |  
Серии.СерияНоменклатуры.СрокГодности КАК ГоденДо  
                                |   ИЗ  
                                |  
Документ." + ИмяДокумента + ".СерииНоменклатуры КАК Серии  
                                |   ГДЕ  
                                |   Серии.Ссылка = &Док");
```

```
Запрос.УстановитьПараметр ("Док", Док.Ссылка) ;  
                                РезультатЗапроса =  
Запрос.Выполнить () ;
```

```
    возврат РезультатЗапроса ;
```

```
КонецФункции
```

```
Функция
```

```
ПолучитьЗапросПоТоварамПеремещение (Док, ИмяДокумента)  
экспорт
```

```
    Запрос = Новый Запрос ("ВЫБРАТЬ
```

```

КАК Номенклатура,
| ДокументТовары.Номенклатура
|
ДокументТовары.Номенклатура.СтранаПроисхождения КАК
СтранаПроисхождения,
|
ДокументТовары.Номенклатура.Производитель КАК
Производитель,
|
ДокументТовары.ЕдиницаИзмерения.Представление КАК
ЕдиницаИзмерения,
|
ДокументТовары.Количество
КАК Количество,
|
ДокументТовары.Цена КАК
Цена,
|
ДокументТовары.Сумма КАК
Сумма,
|
ДокументТовары.ХарактеристикаНоменклатуры
| ИЗ
|
Документ."+ИмяДокумента+".Товары КАК ДокументТовары
| ГДЕ
| ДокументТовары.Ссылка =
&Док
|
| УПОРЯДОЧИТЬ ПО
| Номенклатура
| ИТОГИ
| КОЛИЧЕСТВО (РАЗЛИЧНЫЕ
Номенклатура),
| СУММА (Сумма)
| ПО
| ОБЩИЕ
| АВТОУПОРЯДОЧИВАНИЕ");

Запрос.УстановитьПараметр ("Док", Док.Ссылка);
|
| РезультатЗапроса =
Запрос.Выполнить ();

возврат РезультатЗапроса;
КонецФункции

```

```

Функция
ПолучитьЗапросПоТоварамСписание (Док,ИмяДокумента) экспорт
    Запрос = Новый Запрос ("ВЫБРАТЬ
        | СписаниеТовары.Номенклатура
КАК Номенклатура,
        |
        | СписаниеТовары.Номенклатура.СтранаПроисхождения КАК
СтранаПроисхождения,
        |
        | СписаниеТовары.Номенклатура.Производитель КАК
Производитель,
        |
        | СписаниеТовары.ЕдиницаИзмерения.Представление КАК
ЕдиницаИзмерения,
        | СписаниеТовары.Количество
КАК Количество,
        | СписаниеТовары.Цена КАК
Цена,
        | СписаниеТовары.Сумма КАК
Сумма,
        |
        | СписаниеТовары.ХарактеристикаНоменклатуры
        | ИЗ
        |
        | Документ.СписаниеТоваров.Товары КАК СписаниеТовары
        | ГДЕ
        | СписаниеТовары.Ссылка =
&Док
        |
        | УПОРЯДОЧИТЬ ПО
        | Номенклатура
        | ИТОГИ
        | КОЛИЧЕСТВО (РАЗЛИЧНЫЕ
Номенклатура),
        | СУММА (Сумма)
        | ПО
        | ОБЩИЕ
        | АВТОУПОРЯДОЧИВАНИЕ");

Запрос.УстановитьПараметр ("Док", Док.Ссылка);
        РезультатЗапроса =
Запрос.Выполнить ();

возврат РезультатЗапроса;

```

КонецФункции

Функция ПолучитьЗапросПоСертификатуНакладная (Серия)
экспорт

```
Запрос = Новый Запрос ("ВЫБРАТЬ ПЕРВЫЕ 1
                        | Сертификаты.Ссылка,
                        | Сертификаты.Заявитель,
                        | Сертификаты.ДействителенДо,
                        |
Сертификаты.ВыданНаОсновании,
                        | Сертификаты.Партия,
                        |
Сертификаты.СтранаПроизводитель,
                        |
Сертификаты.НомерСертификата,
                        | Сертификаты.ЕстьОригинал
                        |ИЗ
                        |
Справочник.СерииНоменклатуры.Сертификаты           КАК
Сертификаты
                        | ГДЕ
                        | Сертификаты.Ссылка           =
&Серия");
```

```
Запрос.УстановитьПараметр ("Серия", Серия);
                        РезультатЗапроса           =
Запрос.Выполнить ();
```

возврат РезультатЗапроса;

КонецФункции

Функция ПолучитьУдаленныеСтрокиТоваров (Товары, Ссылка)
Экспорт

```
// получает удаленные строки таблицы товаров в
документе, до их факт. удаления
// может вызываться из процедуры модуля док-та
"ПередЗаписью()"
Запрос=Новый Запрос;
Запрос.Текст="ВЫБРАТЬ
| Документ.Ссылка,
| Документ.Номенклатура,
| Документ.ХарактеристикаНоменклатуры
```

```

| ИЗ
| Документ."+Ссылка.Метаданные().Имя+".Товары КАК
Документ
| ГДЕ
| Документ.Ссылка = &Ссылка
| И (НЕ Документ.Номенклатура В (&Товары)");
Запрос.УстановитьПараметр("Ссылка", Ссылка);
Запрос.УстановитьПараметр("Товары", Товары.Выгрузить(,
"Номенклатура"));
Возврат Запрос.Выполнить().Выгрузить();
КонецФункции

```

```

Процедура
ПроверитьПустыеСтрокиТаблицыТоваровПередЗаписью(Объект, От
каз) Экспорт
Для Каждого Строка из Объект.Товары Цикл
Если ЗначениеНеЗаполнено(Строка.Номенклатура) Тогда
Сообщить("В таблице товаров не заполнено поле
Номенклатура в строке "+Строка.НомерСтроки+"!");
Отказ=Истина;
КонецЕсли;
КонецЦикла;
Для Каждого Строка из Объект.СерииНоменклатуры Цикл
Если ЗначениеНеЗаполнено(Строка.Номенклатура) Тогда
Сообщить("В таблице серий не заполнено поле
Номенклатура в строке "+Строка.НомерСтроки+"!");
Отказ=Истина;
КонецЕсли;
КонецЦикла;
КонецПроцедуры

```

```

Процедура
ПроверитьПовторяющиесяСтрокиВТаблТоваровСерий(Объект, Отка
з, мРежимИСГ=Ложь) Экспорт
Если Не мРежимИСГ Тогда
мТЗТов=Объект.Товары.Выгрузить();
мТЗТов.Колонки.Добавить("Счт");
мТЗТов.ЗаполнитьЗначения(1, "Счт");
мТЗТов.Свернуть("Номенклатура", "Счт");
Для Каждого СтрТЗТов из мТЗТов Цикл
Если СтрТЗТов.Счт>1 Тогда
Сообщить("В таблице товаров повторяется
товар "+СтрТЗТов.Номенклатура);
Отказ=Истина;
КонецЕсли;

```

```

        КонечЦикла;
    КонечЕсли;
    мТЗСер=Объект.СерииНоменклатуры.Выгрузить ();
    мТЗСер.Колонки.Добавить ("Счт");
    мТЗСер.ЗаполнитьЗначения (1, "Счт");
    мТЗСер.Свернуть ("Номенклатура, СерияНоменклатуры", "Счт
");
    Для Каждого СтрТЗСер из мТЗСер Цикл
        Если СтрТЗСер.Счт>1 Тогда
            Сообщить ("В таблице серий повторяются серии
товара "+стрТЗСер.Номенклатура);
            Отказ=Истина;
        КонечЕсли;
    КонечЦикла;
    КонечПроцедуры

```

```

#Если Сервер Тогда

```

```

    Процедура ОтменитьЗаписьОКорректировкеЗаказа (мОбъект)
    Экспорт
        // отмена записи о корректировке заказа покупателя в
целом
        мЗапись=РегистрыСведений.СкорректированныеЗаказы;
        мОтбор=Новый Структура ("ЗаказПокупателя");
        мОтбор.ЗаказПокупателя=мОбъект;
        мВыборка=мЗапись.Выбрать (мОтбор);
        Пока мВыборка.Следующий() Цикл
            мВыборка.ПолучитьМенеджерЗаписи().Удалить();
        КонечЦикла;
    КонечПроцедуры

```

```

    Процедура
    СоздатьЗаписьОКорректировкеЗаказа (мОбъект, мТовар) Экспорт
        // добавление отметки о корректировке товаров в
заказе покупателя
        мЗапись=
РегистрыСведений.СкорректированныеЗаказы.СоздатьМенеджерЗ
аписи();
        мЗапись.ЗаказПокупателя=мОбъект;
        мЗапись.Номенклатура=мТовар;
        мЗапись.Прочитать();
        Если не мЗапись.Выбран() Тогда
            //добавим новую
            мЗапись.ЗаказПокупателя=мОбъект;
            мЗапись.Номенклатура=мТовар;

```

```
    мЗапись .ФлагИзм=Истина ;
    мЗапись .Записать ( ) ;
КонецЕсли ;
КонецПроцедуры
```

Функция

```
ПроверитьНаОтметкуОКорректировке (мОбъект, мТовар) Экспорт
    мЗапрос=Новый Запрос ;
    мЗапрос .Текст="ВЫБРАТЬ
    | СкорректированныеЗаказы.ФлагИзм
    | ИЗ
    | РегистрСведений.СкорректированныеЗаказы           КАК
СкорректированныеЗаказы
    | ГДЕ
    | СкорректированныеЗаказы.ЗаказПокупателя = &мЗаказ
    | И СкорректированныеЗаказы.Номенклатура = &мТовар" ;
    мЗапрос .УстановитьПараметр ("мЗаказ", мОбъект) ;
    мЗапрос .УстановитьПараметр ("мТовар", мТовар) ;
    мРезультат=мЗапрос .Выполнить ( ) ;
    Если мРезультат .Пустой ( ) Тогда
        Возврат Ложь ;
    Иначе
        Возврат Истина ;
    КонецЕсли ;
КонецФункции
```

Процедура

```
ПровестиСнятьПодчиненныйСФВыданный (ДокИсточник, Провести)
Экспорт
    Пере мСФ ;
    Если Провести Тогда
        ПодчиненныйДок =
ПолучитьПодчиненныйДокумент (ДокИсточник, "СчетФактураВыдан
ный", Ложь) ;
    Иначе
        ПодчиненныйДок =
ПолучитьПодчиненныйДокумент (ДокИсточник, "СчетФактураВыдан
ный") ;
    КонецЕсли ;
    Если ПодчиненныйДок<>Неопределено Тогда
        мСФ=ПодчиненныйДок .ПолучитьОбъект ( ) ;
        Если Провести Тогда
            Если не мСФ .Проведен Тогда
                мСФ .Записать (РежимЗаписиДокумента .Проведение) ;
```

```

        КонечЕсли;
    Иначе
        Если СФ.Проведен Тогда

СФ.Записать (РежимЗаписиДокумента.ОтменаПроведения) ;
        КонечЕсли;
    КонечЕсли;
КонечЕсли;
КонечПроцедуры
#КонечЕсли

Процедура
грОбновитьПрайс (Товары, Склад, Знак, НеУчитыватьКоличествоВД
окументе=Ложь) экспорт

    //МассивНовыхЭлементов = Новый Массив;
    //МассивЭлементовДляУдаления = Новый Массив;

    //ТипЦены = СтруктураИсходныхПараметров.ТипЦены;

    // здесь отработаем вариант отсутствия колонки
"Количество", напр. документ
    // "УстЦенНоменклатуры"
    Попытка
        ЕстьКолонкаКоличества=Товары[0].Количество;
    //фиктивное присвоение значения- для проверки
    // существования колонки
        ЕстьКолонкаКоличества=Истина;
    Исключение
        ЕстьКолонкаКоличества=Ложь;
    КонечПопытки;
    Если НеУчитыватьКоличествоВДокументе Тогда // сказано
не учитывать- так и сделаем
        ЕстьКолонкаКоличества=Ложь;
    КонечЕсли;

    Для Каждого СтрокаТовары из Товары Цикл
        Номенклатура = СтрокаТовары.Номенклатура;
        Если
ЗначениеНеЗаполнено (СтрокаТовары.ХарактеристикаНоменклату
ры) Тогда
            Характеристика
Справочники.ХарактеристикиНоменклатуры.ПустаяСсылка ( ) ;
        Иначе

```



```

Характеристика
СтрокаТовары.ХарактеристикаНоменклатуры;
    КонечЕсли;
    Если Знак Тогда

        ДобавитьЭлементВПрайс (Номенклатура, Характеристика, Склад);
    Иначе
        Запрос = Новый Запрос;

        Запрос.УстановитьПараметр ("Номенклатура"
, Номенклатура);
        Запрос.УстановитьПараметр ("Характеристика"
Характеристика);
        Запрос.УстановитьПараметр ("Склад"
Склад);
        //Запрос.УстановитьПараметр ("ПустаяВалюта"
, Справочники.Валюты.ПустаяСсылка ());
        //Запрос.УстановитьПараметр ("ТипЦен"
, ТипЦены);
        Запрос.УстановитьПараметр ("ПустаяЕдиница"
, Справочники.ЕдиницыИзмерения.ПустаяСсылка ());

        Запрос.Текст =
        "ВЫБРАТЬ
        |     ВложенныйЗапрос.КоличествоОстаток           КАК
КоличествоОстаток,
        |     ВложенныйЗапрос.КоличествоВРезерве           КАК
КоличествоВРезерве,
        |     ВложенныйЗапрос.Номенклатура,
        |     ВложенныйЗапрос.КоличествоОстаток-
ВложенныйЗапрос.КоличествоВРезерве КАК СвободныйОстаток
        | ИЗ
        |     (ВЫБРАТЬ
        |         ТоварыНаСкладахОстатки.Номенклатура     КАК
Номенклатура,
        |         СУММА (ВЫБОР
        |             КОГДА
ТоварыНаСкладахОстатки.КоличествоОстаток ЕСТЬ NULL
        |                 ТОГДА 0
        |             ИНАЧЕ
ТоварыНаСкладахОстатки.КоличествоОстаток
        |                 КОНЕЦ) КАК КоличествоОстаток,
        |         СУММА (ВЫБОР

```

```

|
|          КОГДА
ТоварыВРезервеСкладахОстатки.КоличествоОстаток ЕСТЬ NULL
|
|          ТОГДА 0
|
|          ИНАЧЕ
ТоварыВРезервеСкладахОстатки.КоличествоОстаток
|
|          КОНЕЦ) КАК КоличествоВРезерве
|
|          ИЗ
|
|          РегистрНакопления.ТоварыНаСкладах.Остатки (
|
|          ,
|          Номенклатура = &Номенклатура
|          И Склад = &Склад
|          И ХарактеристикаНоменклатуры =
&Характеристика) КАК ТоварыНаСкладахОстатки
|
|          ЛЕВОЕ
|          СОЕДИНЕНИЕ
РегистрНакопления.ТоварыВРезервеНаСкладах.Остатки (
|
|          ,
|          Номенклатура = &Номенклатура
|          И Склад = &Склад
|          И ХарактеристикаНоменклатуры =
&Характеристика) КАК ТоварыВРезервеСкладахОстатки
|
|          ПО ТоварыНаСкладахОстатки.Склад =
ТоварыВРезервеСкладахОстатки.Склад
|
|          И
ТоварыНаСкладахОстатки.Номенклатура
ТоварыВРезервеСкладахОстатки.Номенклатура
|
|          СГРУППИРОВАТЬ ПО
|          ТоварыНаСкладахОстатки.Номенклатура) КАК
ВложенныйЗапрос";

```

```

РезультатЗапроса = Запрос.Выполнить ();

```

```

ВыборкаЗапроса = РезультатЗапроса.Выбрать ();

```

```

Если РезультатЗапроса.Пустой() Тогда

```

```

    УдалитьЭлементИзПрайса (Номенклатура, Характеристика);

```

```

    Иначе

```

```

        ВыборкаЗапроса.Следующий ();

```

```

        Если ЕстьКолонкаКоличества Тогда

```

```

            ДокКоличество=СтрокаТовары.Количество;

```

```

        Иначе

```

```

            ДокКоличество=0;

```

```

        КонецЕсли;

```

Если (ВыборкаЗапроса.СвободныйОстаток-
ДокКоличество) > 0 Тогда

ДобавитьЭлементВПрайс (Номенклатура, Характеристика, Склад);

Иначе

УдалитьЭлементИзПрайса (Номенклатура, Характеристика);

КонецЕсли;

КонецЕсли;

КонецЕсли;

КонецЦикла;

КонецПроцедуры

Процедура грОбновитьПрайсПолностью (Склад) экспорт

ОчиститьПрайс ();

Запрос = Новый Запрос;

Запрос.УстановитьПараметр ("Склад" ,
Склад);

Запрос.УстановитьПараметр ("ПустаяЕдиница" ,
Справочники.ЕдиницыИзмерения.ПустаяСсылка ());

```
Запрос.Текст =
"ВЫБРАТЬ
| ВложенныйЗапрос.КоличествоОстаток           КАК
КоличествоОстаток,
| ВложенныйЗапрос.КоличествоВРезерве           КАК
КоличествоВРезерве,
| ВложенныйЗапрос.Номенклатура,
| ВложенныйЗапрос.ХарактеристикаНоменклатуры
| ИЗ
| (ВЫБРАТЬ
|     СУММА (ВЫБОР
|             КОГДА
ТоварыНаСкладахОстатки.КоличествоОстаток ЕСТЬ NULL
|                 ТОГДА 0
|             ИНАЧЕ
ТоварыНаСкладахОстатки.КоличествоОстаток
|                 КОНЕЦ) КАК КоличествоОстаток,
|     СУММА (ВЫБОР
```

```

|          КОГДА
ТоварыВРезервеСкладахОстатки.КоличествоОстаток ЕСТЬ NULL
|          ТОГДА 0
|          ИНАЧЕ
ТоварыВРезервеСкладахОстатки.КоличествоОстаток
|          КОНЕЦ) КАК КоличествоВРезерве,
|          ТоварыНаСкладахОстатки.Номенклатура          КАК
Номенклатура,
|
|          ТоварыНаСкладахОстатки.ХарактеристикаНоменклатуры КАК
ХарактеристикаНоменклатуры
| ИЗ
|          РегистрНакопления.ТоварыНаСкладах.Остатки(,
Склад = &Склад) КАК ТоварыНаСкладахОстатки,
|
|          РегистрНакопления.ТоварыВРезервеНаСкладах.Остатки(,
Склад = &Склад) КАК ТоварыВРезервеСкладахОстатки
|
| СГРУППИРОВАТЬ ПО
|          ТоварыНаСкладахОстатки.Номенклатура,
|
|          ТоварыНаСкладахОстатки.ХарактеристикаНоменклатуры)
КАК ВложенныйЗапрос";

    РезультатЗапроса = Запрос.Выполнить ();

    ВыборкаЗапроса = РезультатЗапроса.Выбрать ();
    Пока ВыборкаЗапроса.Следующий () Цикл
        НовыйЭлемент = Справочники.Прайс.СоздатьЭлемент ();
        НовыйЭлемент.Наименование =
? (ЗначениеНеЗаполнено (ВыборкаЗапроса.Номенклатура.НаименованиеПолное),
ВыборкаЗапроса.Номенклатура.Наименование,
ВыборкаЗапроса.Номенклатура.НаименованиеПолное);
        НовыйЭлемент.Номенклатура =
ВыборкаЗапроса.Номенклатура;
        НовыйЭлемент.ХарактеристикаНоменклатуры =
ВыборкаЗапроса.ХарактеристикаНоменклатуры;

    Если
    ЕстьПросроченныйСертификатИлиИСТ (ВыборкаЗапроса.Номенклатура,
Склад) Тогда
        НовыйЭлемент.Статус
=Перечисления.СтатусыТовараВПрайсе.ИСТ;
    Иначе

```

```

        НовыйЭлемент.Статус
=Перечисления.СтатусыТовараВПрайсе.ГотовКПродаже;
        КонецЕсли;
        Попытка
            НовыйЭлемент.Записать ();
        Исключение
            #Если Клиент Тогда
                Сообщить ("Не удалось внести элемент
"+ВыборкаЗапроса.Номенклатура+" в подбор!");
            #КонецЕсли

```

```

        ЗаписьЖурналаРегистрации ("Данные.Ошибка", УровеньЖурна
лаРегистрации.Ошибка, ВыборкаЗапроса.Номенклатура.Метаданн
ые (), ВыборкаЗапроса.Номенклатура, "Ошибка добавления в
подбор");
        КонецПопытки

```

```

        КонецЦикла;
        КонецПроцедуры

```

```

        Функция
        ЕстьПросроченныйСертификатИлиИСТ (Номенклатура, Склад)
        Экспорт
            // есть ли у номенклатуры просроченный сертификат или
серия ИСТ
            Если
                ЗначениеНеЗаполнено (Константы.ЧислоМесИСТ.Получить ()) или
                Константы.ЧислоМесИСТ.Получить () = 0 Тогда
                    Возврат Ложь;
                КонецЕсли;
                Запрос = Новый Запрос;
                Запрос.Текст = "ВЫБРАТЬ
                | гр_СерииНоменклатурыОстатки.СерияНоменклатуры
                | ИЗ
                | РегистрНакопления.гр_СерииНоменклатуры.Остатки (
                |     &ДатаКон,
                |     Номенклатура = &Номенклатура
                |     И     Склад     =     &Склад)     КАК
                гр_СерииНоменклатурыОстатки
                | ГДЕ
                | гр_СерииНоменклатурыОстатки.КоличествоОстаток > 0";
                Запрос.УстановитьПараметр ("Склад", Склад);
                Запрос.УстановитьПараметр ("ДатаКон", ТекущаяДата ());

```

```

Запрос.УстановитьПараметр ("Номенклатура", Номенклатура
);
ТЗСерий=Запрос.Выполнить().Выгрузить();

Запрос = Новый Запрос;
Запрос.Текст="ВЫБРАТЬ
| МИНИМУМ(сертификаты.ДействителенДоМакс)           КАК
ДействителенДо,
| МИНИМУМ(серии.СрокГодности) КАК СрокГодности
| ИЗ
| (ВЫБРАТЬ
|   МИНИМУМ(СерииНоменклатуры.СрокГодности)           КАК
СрокГодности,
|   СерииНоменклатуры.Владелец КАК Владелец
| ИЗ
|   Справочник.СерииНоменклатуры                       КАК
СерииНоменклатуры
| ГДЕ
|   СерииНоменклатуры.Ссылка В (&ТЗСерий)
|
| СГРУППИРОВАТЬ ПО
|   СерииНоменклатуры.Владелец) КАК серии
| ЛЕВОЕ СОЕДИНЕНИЕ (ВЫБРАТЬ
|   МАКСИМУМ(Сертификаты.ДействителенДо)           КАК
ДействителенДоМакс,
|   Сертификаты.Ссылка КАК Серия
| ИЗ
|   Справочник.СерииНоменклатуры.Сертификаты КАК
Сертификаты
| ГДЕ
|   Сертификаты.Ссылка В (&ТЗСерий)
|
| СГРУППИРОВАТЬ ПО
|   Сертификаты.Ссылка) КАК сертификаты
| ПО серии.Владелец = сертификаты.Серия.Владелец";
Запрос.УстановитьПараметр ("ТЗСерий", ТЗСерий);
Выборка=Запрос.Выполнить().Выбрать();
Если Выборка.Следующий() Тогда
    ДатаИСТГ=
ДобавитьМесяц(ТекущаяДата(), Константы.ЧислоМесИСТГ.Получит
ь());
    Если не ЗначениеНеЗаполнено(Выборка.СрокГодности)
Тогда
        Если (Выборка.СрокГодности<ДатаИСТГ) Тогда
            Возврат Истина;

```

```

        КонечЕсли;
    КонечЕсли;
    Если не ЗначениеНеЗаполнено (Выборка.ДействителенДо)
Тогда
        Если (Выборка.ДействителенДо<ДатаИСГ) Тогда
            Возврат Истина;
        КонечЕсли;
    КонечЕсли;
    КонечЕсли;
    Возврат Ложь;
    КонечФункции

Функция        ПроверитьСоотвСписанияВнутрЗаказу (Объект)
Экспорт
    // Объект- проверяемый внутренний заказ
    // возвращает: Истина- все нормально, Ложь- ошибка
    // проверим соответствие внутреннего заказа и
    списаний товаров
    Ошибка=Ложь;
    //выбираем подчиненные документы, включая
    непроведенные (но не помеченные на удаление)
    ПодчиненныеДоки =
    ПолучитьПодчиненныйДокумент (Объект.Ссылка, "СписаниеТоваро
    в", Ложь, Истина);
    Если не ПодчиненныеДоки=неопределено Тогда

        СписПодДок=ПодчиненныеДоки.ВыгрузитьКолонку ("Подчинен
        ныйДокумент");
        Запрос=Новый Запрос;
        Запрос.Текст= "ВЫБРАТЬ
        | СУММА (СписаниеТоваровТовары.Количество)           КАК
Количество,
        | СписаниеТоваровТовары.Номенклатура
        | ИЗ
        | Документ.СписаниеТоваров.Товары                   КАК
СписаниеТоваровТовары
        | ГДЕ
        | СписаниеТоваровТовары.Ссылка В (&ДокиСпис)
        | И СписаниеТоваровТовары.ДокументРезерва           =
&ВнЗаказ
        |
        | СГРУППИРОВАТЬ ПО
        | СписаниеТоваровТовары.Номенклатура";
        Запрос.УстановитьПараметр ("ДокиСпис", СписПодДок);
        Запрос.УстановитьПараметр ("ВнЗаказ", Объект.Ссылка);

```

```

    Результат=Запрос.Выполнить ();
    Если Не Результат.Пустой() Тогда
        // проверим, не списывается ли товара больше,
        чем резервировалось.
        ТЗТов=Объект.Товары.Выгрузить ();
        Выборка=Результат.Выбрать ();
        Пока Выборка.Следующий() Цикл

        СтрокаТов=ТЗТов.Найти(Выборка.Номенклатура,"Номенклатура");

        Если СтрокаТов=неопределено Тогда
            Ошибка=Истина;
            #Если Клиент Тогда
                Сообщить ("Списываемый товар
                "+Выборка.Номенклатура+" во внутр.заказе не
                найден!", СтатусСообщения.Внимание);
            #КонецЕсли
            ИначеЕсли
                СтрокаТов.Количество<Выборка.Количество Тогда
                    Ошибка=Истина;
                    #Если Клиент Тогда
                        Сообщить ("Количество списываемого
                        товара "+СокрЛП(Выборка.Номенклатура)+" больше количества
                        во внутр.заказе!", СтатусСообщения.Внимание);
                    #КонецЕсли
                КонецЕсли;
            КонецЦикла;
        КонецЕсли;
    Возврат не Ошибка;
КонецФункции

```

Процедура
 ДобавитьЭлементВПрайс (Номенклатура, Характеристика, Склад)
 Экспорт

```

    перем мОрганизация;
    мОрганизация=неопределено;
    Если Не ЗначениеНеЗаполнено(Склад) Тогда
        Если Не ЗначениеНеЗаполнено(Склад.Организация)
            Тогда
                мОрганизация=Склад.Организация;
            Иначе
                мОрганизация=ПараметрыСеанса.ТекущаяОрганизация;
        КонецЕсли
    КонецЕсли

```



```

        КонецЕсли;
Иначе
    мОрганизация=ПараметрыСеанса.ТекущаяОрганизация;
КонецЕсли;

Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| Прайс.Номенклатура КАК Номенклатура
| ИЗ
| Справочник.Прайс КАК Прайс
| ГДЕ
| Прайс.ПометкаУдаления = ЛОЖЬ
| И Прайс.Номенклатура = &Номенклатура
| И          Прайс.ХарактеристикаНоменклатуры          =
&Характеристика";

Запрос.УстановитьПараметр ("Номенклатура", Номенклатура
);
Запрос.УстановитьПараметр ("Характеристика", Характерис
тика);
    Попытка
        Выборка = Запрос.Выполнить().Выбрать();
    Исключение
        #Если Клиент Тогда
            Сообщить("Не удалось заблокировать подбор!
Возможно идет обновление прайса.");
        #КонецЕсли
    Возврат;
КонецПопытки;

Выборка.Следующий();
Если ЗначениеНеЗаполнено(Выборка.Номенклатура) Тогда
    НовыйЭлемент = Справочники.Прайс.СоздатьЭлемент();
    НовыйЭлемент.Наименование          =
? (ЗначениеНеЗаполнено(Номенклатура.НаименованиеПолное), Но
менклатура.Наименование, Номенклатура.НаименованиеПолное);
    НовыйЭлемент.Номенклатура = Номенклатура;
    НовыйЭлемент.ХарактеристикаНоменклатуры          =
Характеристика;
    НовыйЭлемент.Цена          =
ПолучитьЦенуНоменклатурыДляПрайса(Номенклатура, Характерис
тика, Справочники.ТипыЦенНоменклатуры.Оптовая, мОрганизация
);

```

```

        Если НовыйЭлемент.Цена="" Тогда
            НовыйЭлемент.Статус
=Перечисления.СтатусыТовараВПрайсе.НетЦены;
            ИначеЕсли
ЕстьПросроченныйСертификатИлиИСГ (Номенклатура, Склад)
Тогда
            НовыйЭлемент.Статус
=Перечисления.СтатусыТовараВПрайсе.ИСГ;
            Иначе
            НовыйЭлемент.Статус
=Перечисления.СтатусыТовараВПрайсе.ГотовКПродаже;
            КонецЕсли;
            Попытка
                НовыйЭлемент.Записать ();
            Исключение
                #Если Клиент Тогда
                    Сообщить ("Не удалось внести элемент
"+Номенклатура+" в подбор!");
                #КонецЕсли
                Если ПланыОбмена.ГлавныйУзел () =Неопределено
Тогда // запись работает только для ГП

            ЗаписьЖурналаРегистрации ("Данные.Ошибка", УровеньЖурна
лаРегистрации.Ошибка, Номенклатура.Метаданные (), Номенклату
ра, "Ошибка добавления в подбор");
            КонецЕсли;
            КонецПопытки

        КонецЕсли;

        КонецПроцедуры

        Процедура
УдалитьЭлементИзПрайса (Номенклатура, Характеристика)
        Если Константы.ФлагОбновленияПодбора.Получить () Тогда
            Возврат; // не удаляем- идет обновление прайса
        КонецЕсли;

        Запрос = Новый Запрос;
        Запрос.Текст =
"ВЫБРАТЬ
| Прайс.Номенклатура КАК Номенклатура,
| Прайс.Ссылка КАК Прайс
| ИЗ
| Справочник.Прайс КАК Прайс

```

```

| ГДЕ
| Прайс.ПометкаУдаления = ЛОЖЬ
| И Прайс.Номенклатура = &Номенклатура
| И Прайс.ХарактеристикаНоменклатуры =
&Характеристика";

Запрос.УстановитьПараметр ("Номенклатура", Номенклатура
);
Запрос.УстановитьПараметр ("Характеристика", Характерис
тика);

Попытка
Выборка = Запрос.Выполнить().Выбрать();
Исключение
#Если Клиент Тогда
Сообщить ("Не удалось заблокировать подбор!
Возможно идет обновление прайса.");
#КонецЕсли
Возврат;
КонецПопытки;
Пока Выборка.Следующий() Цикл
Если НЕ ЗначениеНеЗаполнено (Выборка.Номенклатура)
Тогда
ЭлементДляУдаления =
Выборка.Прайс.ПолучитьОбъект();
Попытка
ЭлементДляУдаления.Удалить();
Исключение
#Если Клиент Тогда
Сообщить ("Не удалось удалить элемент
"+Выборка.Номенклатура+" в подборе!");
#КонецЕсли
КонецПопытки;
КонецЕсли;
КонецЦикла;

КонецПроцедуры

Функция
ПолучитьЦенуНоменклатурыДляПрайса (Номенклатура, Характерис
тика, ТипЦены, Организация= неопределено) Экспорт

Запрос = Новый Запрос ("ВЫБРАТЬ ПЕРВЫЕ 1
| ВЫБОР

```

```

НЕОПРЕДЕЛЕННО | КОГДА Цены.Цена =
NULL | ИЛИ Цены.Цена =
| ТОГДА """"
| ИНАЧЕ Цены.Цена
| КОНЕЦ КАК Цена
| ИЗ
|
РегистрСведений.ЦеныНоменклатуры.СрезПоследних (
| &ТекДата,
| Номенклатура =
&Номенклатура
| И
ХарактеристикаНоменклатуры = &Характеристика
| И ТипЦен = &ТипЦены
| И
(Организация.Ссылка ЕСТЬ NULL
| ИЛИ Организация
= &Организация) ) КАК Цены
| ГДЕ
| Цены.Цена <> 0
|
| УПОРЯДОЧИТЬ ПО
| Цены.Период УБЫВ" );

ХарактеристикаНоменклатуры =
? (ЗначениеНеЗаполнено (Характеристика) , Справочники.ХарактеристикиНоменклатуры.ПустаяСсылка ( ) , Характеристика) ;
Запрос.УстановитьПараметр ("Номенклатура" , Номенклатура
) ;

Запрос.УстановитьПараметр ("Организация" , Организация) ;
Запрос.УстановитьПараметр ("ТекДата" , КонецДня (ТекущаяДата ( ) ) ) ;

Запрос.УстановитьПараметр ("Характеристика" , ХарактеристикаНоменклатуры) ;
Запрос.УстановитьПараметр ("ТипЦены" , ТипЦены) ;

Выборка = Запрос.Выполнить ( ) .Выбрать ( ) ;
Выборка.Следующий ( ) ;
Цена = Выборка.Цена;

возврат Строка (Цена) ;

```

КонецФункции

```
Процедура ОчиститьПрайс () экспорт
Выборка = Справочники.Прайс.Выбрать ();
Пока Выборка.Следующий () Цикл
    ОбъектСпр = Выборка.ПолучитьОбъект ();
    ОбъектСпр.Удалить ();
    Сообщить (ОбъектСпр.Ссылка);
КонецЦикла;
КонецПроцедуры
```

#Если Сервер Тогда

Функция

ПолучитьПодчиненныйДокумент (ДокИсточник, ИмяПодчиненногоДокумента, ТолькоПроведенные=Истина, СписокПодчиненных=Ложь) экспорт

```
ПодчиненныйДок = Неопределено;
Запрос = Новый Запрос;
Запрос.УстановитьПараметр ("ДокИсточник", ДокИсточник);
Запрос.Текст =
```

```
"ВЫБРАТЬ
| Док.Ссылка КАК ПодчиненныйДокумент,
| Док.Проведен КАК Проведен
| ИЗ
| Документ." + ИмяПодчиненногоДокумента + " КАК Док
| ГДЕ
| Док.";
```

Если

ИмяПодчиненногоДокумента="ПоступлениеТоваровУслуг" Тогда

```
    Запрос.Текст = Запрос.Текст+"СкладОрдер";
```

ИначеЕсли

(ИмяПодчиненногоДокумента="РеализацияТоваровУслуг")

или

(ИмяПодчиненногоДокумента="ВозвратТоваровОтПокупателя")

Тогда

```
    Запрос.Текст = Запрос.Текст+"Сделка";
```

ИначеЕсли

(ИмяПодчиненногоДокумента="ПриходныйКассовыйОрдер") Тогда

```
    Запрос.Текст =
```

```
Запрос.Текст+"РасшифровкаПлатежа.Сделка";
```

ИначеЕсли

ИмяПодчиненногоДокумента="ПеремещениеТоваров" Тогда

```

        Если
ТипЗнч (ДокИсточник) =Тип ("ДокументСсылка.ВнутреннийЗаказ")
Тогда
        Запрос.Текст = Запрос.Текст+"ВнутреннийЗаказ";
        Иначе
        Запрос.Текст = Запрос.Текст+"Основание";
        КонечЕсли;
        ИначеЕсли ИмяПодчиненногоДокумента="СписаниеТоваров"
Тогда
        Запрос.Текст =
Запрос.Текст+"ИнвентаризацияТоваровНаСкладе";
        ИначеЕсли
(ИмяПодчиненногоДокумента="СчетФактураВыданный")
        или
        (ИмяПодчиненногоДокумента="СчетФактураПолученный")
Тогда
        Запрос.Текст = Запрос.Текст+"ДокументОснование";
        КонечЕсли;
        Запрос.Текст = Запрос.Текст+" = &ДокИсточник
| И Док.ПометкаУдаления = Ложь";
        Если ТолькоПроведенные Тогда
        Запрос.Текст = Запрос.Текст+"
| И Док.Проведен = Истина";
        КонечЕсли;
        Результат = Запрос.Выполнить ();

        Если Не Результат.Пустой() Тогда
        Если СписокПодчиненных Тогда
        ПодчиненныйДок=Результат.Выгрузить ();
        Иначе //выбираем первый документ
        Выборка = Результат.Выбрать ();
        Пока Выборка.Следующий() Цикл
        ПодчиненныйДок =
Выборка.ПодчиненныйДокумент;
        Прервать;
        КонечЦикла;
        КонечЕсли;
        КонечЕсли;

        возврат ПодчиненныйДок;

        КонечФункции
#КонечЕсли

//Серик 06.01.2007

```

```

//функция проверяет можно ли пользователю изменять
статусы с документах отгрузки
Функция РазрешитьИзменениеСтатусовОтгрузки() Экспорт
    СписокФильтров =
ПолучитьЗначениеПраваПользователя (ПланыВидовХарактеристик
.ПраваПользователей.ИзменениеСтатусовОтгрузки, Истина);

    Если (СписокФильтров.Количество() = 0)
        ИЛИ (СписокФильтров.Количество() > 1) Тогда
        Возврат Истина;
    Иначе
        Возврат СписокФильтров[0].Значение;
    КонецЕсли;

КонецФункции //

РазрешитьРедактированиеЦенВДокументах()

Функция ПроверитьТолькоЦифры (Пар, Режим="Очистка")
Экспорт
    Ответ="";
    Для Н=1 По СтрДлина (Пар) Цикл
        Сим=Сред (Пар, н, 1);
        Если НЕ (Найти ("0123456789", Сим)) Тогда
            Если Режим="Проверка" Тогда
                Возврат Ложь;
            Иначе
                Продолжить;
            КонецЕсли;
        Иначе
            Ответ=Ответ+Сим;
        КонецЕсли;
    КонецЦикла;
    Если Режим="Проверка" Тогда
        Возврат Истина;
    Иначе
        Возврат Ответ;
    КонецЕсли;
КонецФункции

Функция ДатаИзСтроки (СтрокаДата) Экспорт
СтрокаДатаДо =СтрокаДата;
СтрокаДата =ПроверитьТолькоЦифры (СтрокаДата);

```

```

Если   СтрДлина (СтрокаДата) = 6 Тогда
    ДатаИмпорт = СокрЛП (СтрокаДата) ;
    Год = Прав (ДатаИмпорт, 2) ;
    Если Число (Лев (Год, 1)) > 5 Тогда
        Год = "19" + Год ;
    Иначе
        Год = "20" + Год ;
    КонецЕсли ;
    Месяц = Сред (ДатаИмпорт, 3, 2) ;
    День = Лев (ДатаИмпорт, 2) ;
    Попытка
        Дата80 = Дата (Год + Месяц + День) ;
    Исключение
        Сообщить ("Ошибка конвертации даты:
'" + ДатаИмпорт + "' -> '" + Год + "." + Месяц + "." + " '") ;
        Возврат ПустоеЗначениеТипа ("Дата") ;

    Конецпопытки ;
    Возврат Дата80 ;

ИначеЕсли СтрДлина (СтрокаДата) = 8 Тогда
    ДатаИмпорт = СокрЛП (СтрокаДата) ;
    Год = Прав (ДатаИмпорт, 4) ;
    Месяц = Сред (ДатаИмпорт, 3, 2) ;
    День = Лев (ДатаИмпорт, 2) ;
    Попытка
        Дата80 = Дата (Год + Месяц + День) ;
    Исключение
        Сообщить ("Ошибка конвертации даты:
'" + ДатаИмпорт + "' -> '" + Год + "." + Месяц + "." + " '") ;
        Возврат ПустоеЗначениеТипа ("Дата") ;
    Конецпопытки ;
    Возврат Дата80 ;
Иначе
    Сообщить ("Неверный формат переданной строки.
Полученный параметр: '" + СтрокаДатаДо + "'. Обработанный
параметр: '" + СтрокаДата) ;
    Возврат ПустоеЗначениеТипа ("Дата") ;
КонецЕсли ;

КонецФункции

// <Описание функции>
//

```



```

// Параметры
// <Параметр1> - <Тип.Вид>- <описание параметра>
//               <продолжение описания параметра>
// <Параметр2> - <Тип.Вид>- <описание параметра>
//               <продолжение описания параметра>
//
// Возвращаемое значение:
//   <Тип.Вид> - <описание возвращаемого значения>
// Рауль 15.01.2007
Функция
НайтиПодчиненныйДокументПоВиду (ДокументСсылка,
ВидДокумента, РеквизитОснование) Экспорт
    НайденныйДокумент = Неопределено;
    Если НЕ ЗначениеНеЗаполнено (ДокументСсылка) Тогда

        Запрос = Новый Запрос;

        // Установим параметры запроса
        Запрос.УстановитьПараметр ("ДокументСсылка",
ДокументСсылка);

        Запрос.Текст =
        "ВЫБРАТЬ
        | Ссылка
        | ИЗ
        | Документ." + ВидДокумента + "
        |
        | ГДЕ
        | " + РеквизитОснование + " = &ДокументСсылка";

        ВыборкаИзЗапроса = Запрос.Выполнить().Выбрать();

        Если ВыборкаИзЗапроса.Следующий() Тогда
            НайденныйДокумент = ВыборкаИзЗапроса.Ссылка;
        КонецЕсли;

    КонецЕсли;

    Возврат НайденныйДокумент;
КонецФункции // НайтиПодчиненныйДокументПоВиду ()

Функция
ПолучитьПодчиненныйДокументКорректировкаДолгаДляНакладной
(Заказ) Экспорт

```

```

НайденныйДокумент = Неопределено;

Если НЕ ЗначениеНеЗаполнено (Заказ) Тогда

    Запрос = Новый Запрос;

    // Установим параметры запроса
    Запрос.УстановитьПараметр ("ДокументСсылка", Заказ);

    Запрос.Текст =
        "ВЫБРАТЬ
        |   КорректировкаДолга.Ссылка
        | ИЗ
        |   Документ.КорректировкаДолга                                КАК
КорректировкаДолга
        | ГДЕ
        |   КорректировкаДолга.Основание = &ДокументСсылка";

    ВыборкаИзЗапроса = Запрос.Выполнить ().Выбрать ();

    Если ВыборкаИзЗапроса.Следующий () Тогда
        Если   ВыборкаИзЗапроса.Ссылка.Проведен = Истина
Тогда
            НайденныйДокумент = ВыборкаИзЗапроса.Ссылка;
        КонецЕсли;
    КонецЕсли;

КонецЕсли;

Возврат НайденныйДокумент;

КонецФункции

Функция ПолучитьСкладПоУзлуПланаОбмена () экспорт
    Организация =
ПланыОбмена.ПланОбменаВыборочный.ЭтотУзел ().Организация;
    Если ЗначениеНеЗаполнено (Организация) тогда
        Организация=ПараметрыСеанса.ТекущаяОрганизация;
    КонецЕсли;
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |   Склады.Ссылка КАК Склад
        | ИЗ
        |   Справочник.Склады КАК Склады

```

```

| ГДЕ
| Склады.Организация = &Организация";
Запрос.УстановитьПараметр ("Организация", Организация) ;
ВыборкаПоСкладу = Запрос.Выполнить ().Выбрать ();
Пока ВыборкаПоСкладу.Следующий () Цикл
    Если ВыборкаПоСкладу.Склад =
Справочники.Склады.Транзитный Тогда
    Продолжить ;
    КонецЕсли ;
    Склад = ВыборкаПоСкладу.Склад ;
КонецЦикла ;
возврат Склад ;
КонецФункции

```

```

Функция ПолучитьДень (ИсходнаяДата,
КоличествоПрибавляемыхДней) Экспорт

```

```

    Запрос = Новый Запрос ("ВЫБРАТЬ
    |
    ДОБАВИТЬКДАТЕ (&ИсходнаяДата, ДЕНЬ, &Количество) КАК
Дата");

```

```

    Запрос.УстановитьПараметр ("ИсходнаяДата",
ИсходнаяДата) ;

```

```

    Запрос.УстановитьПараметр ("Количество",
КоличествоПрибавляемыхДней) ;

```

```

    Возврат Запрос.Выполнить ().Выгрузить () [0] .Дата ;

```

```

КонецФункции

```

```

Функция
ПолучитьЦенуТовараПоХарактеристикам (Склад, Товар, Характери
стика = Неопределено, Серия = Неопределено, Организация =
неопределено) Экспорт

```

```

    Характеристика = ? (Характеристика =
Неопределено, Справочники.ХарактеристикиНоменклатуры.Пуста
яСсылка (), Характеристика) ;

```

```

    Серия = ? (Серия =
Неопределено, Справочники.СерииНоменклатуры.ПустаяСсылка ()
, Серия) ;

```

```

    Запрос = Новый Запрос ;
    Запрос.Текст =

```

```

"ВЫБРАТЬ ПЕРВЫЕ 1
| Цены.Номенклатура,
| Цены.ХарактеристикаНоменклатуры КАК Характеристика,
| Цены.СерияНоменклатуры КАК Серия,
| Цены.Цена
| ИЗ
| РегистрСведений.ЦеныНоменклатуры.СрезПоследних (
|
|     Номенклатура = &Товар
|     И     ХарактеристикаНоменклатуры     =
&Характеристика
|     И СерияНоменклатуры = &Серия
|     И (Организация.Ссылка ЕСТЬ NULL
|     ИЛИ Организация = &Организация) ) КАК
Цены
|     Где Цены.Цена<>0
|
| УПОРЯДОЧИТЬ ПО
| Цены.Период УБЫВ";

```

```

Запрос.УстановитьПараметр ("Товар", Товар) ;
Запрос.УстановитьПараметр ("Характеристика", Характерис
тика) ;

```

```

Запрос.УстановитьПараметр ("Серия", Серия) ;

```

```

Запрос.УстановитьПараметр ("Организация", Организация) ;

```

```

Результат = Запрос.Выполнить () ;
Если Не Результат.Пустой () Тогда
    Выборка = Результат.Выбрать () ;
    Выборка.Следующий () ;
    Цена = Выборка.Цена ;

```

```

Иначе

```

```

    Цена = 0 ;

```

```

КонецЕсли ;

```

```

возврат Цена ;

```

```

КонецФункции

```

```

Функция    ПолучитьСебестоимостьТовара (Товар, Склад     =
Неопределено) Экспорт
// Здесь нужно вычислить себестоимость товара

```

```
//СкладДляЗапроса = ? (Склад =  
Неопределено, Справочники.Склады.ПустаяСсылка ( ), Склад) ;
```

```
Запрос= новый Запрос;  
//Запрос.УстановитьПараметр ("Склад", Склад) ;  
Запрос.УстановитьПараметр ("Номенклатура", Товар) ;  
//Запрос.УстановитьПараметр ("Склад",  
СкладДляЗапроса) ;  
Запрос.Текст=  
"ВЫБРАТЬ  
|  
СУММА (ПартииТоваровНаСкладахОстатки.КоличествоОстаток  
) КАК КоличествоОстаток,  
|  
СУММА (ПартииТоваровНаСкладахОстатки.СтоимостьОстаток)  
КАК СтоимостьОстаток,  
| ПартииТоваровНаСкладахОстатки.Номенклатура  
| ИЗ  
| РегистрНакопления.ПартииТоваровНаСкладах.Остатки (,  
Номенклатура = &Номенклатура) КАК  
ПартииТоваровНаСкладахОстатки  
|  
| СГРУППИРОВАТЬ ПО  
| ПартииТоваровНаСкладахОстатки.Номенклатура";  
РезультатЗапроса=Запрос.Выполнить () ;  
Выборка = РезультатЗапроса.Выбрать () ;  
Выборка.Следующий () ;  
Если (Выборка.КоличествоОстаток<>0) И  
 (Выборка.КоличествоОстаток<>Неопределено) И  
 (Выборка.СтоимостьОстаток<>Неопределено) Тогда  
 Возврат  
Выборка.СтоимостьОстаток/Выборка.КоличествоОстаток;  
КонецЕсли;  
  
Возврат 0;  
  
КонецФункции
```

```
Функция  
ПолучитьСтруктуруОстатковТовараПоХарактеристикам (Склад, То  
вар, Характеристика = Неопределено) Экспорт
```

Характеристика = ? (Характеристика =
Неопределено, Справочники. ХарактеристикиНоменклатуры. Пуста
яСсылка () , Характеристика) ;

```
Запрос = Новый Запрос;  
Запрос.Текст =  
"ВЫБРАТЬ  
| ТоварыНаСкладах.Номенклатура КАК Номенклатура,  
| ТоварыНаСкладах.ХарактеристикаНоменклатуры КАК  
ХарактеристикаНоменклатуры,  
| ТоварыНаСкладах.КоличествоОстаток,  
| СУММА (ВЫБОР  
| КОГДА ТоварыНаСкладах.КоличествоОстаток ЕСТЬ  
NULL  
| ТОГДА 0  
| ИНАЧЕ ТоварыНаСкладах.КоличествоОстаток  
| КОНЕЦ- ВЫБОР  
| КОГДА  
ТоварыВРезервеНаСкладах.КоличествоОстаток ЕСТЬ NULL  
| ТОГДА 0  
| ИНАЧЕ  
ТоварыВРезервеНаСкладах.КоличествоОстаток  
| КОНЕЦ) КАК КоличествоСвободныйОстаток  
| ИЗ  
| РегистрНакопления.ТоварыНаСкладах.Остатки (  
| &Дата,  
| Склад = &Склад  
| И Номенклатура = &Товар) КАК ТоварыНаСкладах  
| ЛЕВОЕ СОЕДИНЕНИЕ  
РегистрНакопления.ТоварыВРезервеНаСкладах.Остатки (  
| &Дата,  
| Склад = &Склад  
| И Номенклатура = &Товар) КАК  
ТоварыВРезервеНаСкладах  
| ПО ТоварыВРезервеНаСкладах.Номенклатура =  
ТоварыНаСкладах.Номенклатура  
| И  
ТоварыВРезервеНаСкладах.ХарактеристикаНоменклатуры =  
ТоварыНаСкладах.ХарактеристикаНоменклатуры  
|  
| СГРУППИРОВАТЬ ПО  
| ТоварыНаСкладах.Номенклатура,  
| ТоварыНаСкладах.ХарактеристикаНоменклатуры,  
| ТоварыНаСкладах.КоличествоОстаток";
```

```

        Запрос.УстановитьПараметр ("Дата", КонецДня (ТекущаяДата
    ())) ;
        Запрос.УстановитьПараметр ("Склад", Склад) ;
        Запрос.УстановитьПараметр ("Товар", Товар) ;
        Запрос.УстановитьПараметр ("Характеристика", Характерис
тика) ;

        Результат = Запрос.Выполнить () ;
        Если Не Результат.Пустой () Тогда
            Выборка = Результат.Выбрать () ;
            Выборка.Следующий () ;
            КоличествоОстаток = Выборка.КоличествоОстаток ;
            КоличествоСвободныйОстаток =
Выборка.КоличествоСвободныйОстаток ;
        Иначе
            КоличествоОстаток = 0 ;
            КоличествоСвободныйОстаток = 0 ;
        КонецЕсли ;

        Структура = Новый Структура ;
        Структура.Вставить ("КоличествоОстаток", КоличествоОста
ток) ;
        Структура.Вставить ("КоличествоСвободныйОстаток", Колич
ествоСвободныйОстаток) ;

        возврат Структура ;

        КонецФункции

// параметры: Объект- сам объект
// заполняет уникальным идентификатором реквизит
объекта ИД80
// и реквизит GUID (если он есть) и проводит
сравнение на случай изменения идентификатора
Процедура ЗаполнитьИД80иGUIDОбъекта (Объект) Экспорт
Если
Объект.Метаданные ().Реквизиты.найти ("ИД80") =Неопределено
Тогда
    // нет ведущего реквизита- выходим
    Возврат ;
КонецЕсли ;
ЕстьGUID= (не
Объект.Метаданные ().Реквизиты.найти ("GUID") =Неопределено)
; //проверяем наличие GUID

```

```

мСсылкаОбъекта=Объект.Ссылка;
Если мСсылкаОбъекта.Пустая() Тогда // новый объект
    Если
ЗначениеНеЗаполнено (Объект.ПолучитьСсылкуНового()) Тогда
// это новый документ, созданный здесь
    Если Лев (ТипЗнч (Объект) , Найти (ТипЗнч (Объект) , "
") -1) ="Документ" Тогда

        СсылкаНового=Документы [Объект.Метаданные () .Имя] .Получ
итьСсылку ();
        ИначеЕсли
Лев (ТипЗнч (Объект) , Найти (ТипЗнч (Объект) , "
") -
1) ="Справочник" Тогда

        СсылкаНового=Справочники [Объект.Метаданные () .Имя] .Пол
учитьСсылку ();
        КонецЕсли;
        Объект.УстановитьСсылкуНового (СсылкаНового) ;
        КонецЕсли;
        мСсылкаОбъекта=Объект.ПолучитьСсылкуНового ();
        КонецЕсли;
        ИД=Строка (мСсылкаОбъекта.УникальныйИдентификатор ());
//а теперь проверяем на расхождения и заполняем
        Если СокрЛП (Объект.ИД80) <> СокрЛП (ИД) Тогда
            Если ЗначениеЗаполнено (Объект.ИД80) Тогда
//проверка на сдвиг ИД
                Если ЕстьGUID Тогда //сообщать нужно, когда есть
GUID и он тоже изменен, чтобы каждый раз не выдавалось
                    Если СокрЛП (Объект.GUID) <> СокрЛП (ИД) Тогда
//GUID существует и изменился или не существует- сообщаем
                        Если
ПланыОбмена.ГлавныйУзел () =Неопределено Тогда // запись в
журнал только на ГП

                            ЗаписьЖурналаРегистрации ("Данные.ИзменениеГУИД", Урове
ньЖурналаРегистрации.Информация, Объект.Метаданные (), мСсыл
каОбъекта, "ИД80. "+СокрЛП (Объект.ИД80)+" Текущий ГУИД
"+СокрЛП (ИД) );

                            ЗаписьЖурналаРегистрации ("Данные.ИзменениеГУИД", Урове
ньЖурналаРегистрации.Информация, Объект.Метаданные (), Обьек
т.Ссылка, "Изменен GUID. был "+СокрЛП (Объект.GUID)+" стал
"+СокрЛП (ИД) );
                                КонецЕсли;

```



```
        Объект.GUID = ИД; // заполняем новым значением
```

```
        КонечЕсли;  
        КонечЕсли;  
        Иначе // просто заполняем ИД80 и GUID  
        Объект.ИД80 = ИД;  
        Если ЕстьGUID Тогда  
            Объект.GUID = ИД;  
        КонечЕсли;  
        КонечЕсли;  
        КонечЕсли;  
        КонечПроцедуры
```

```
Функция ПроверитьРольПользователя(Роль) Экспорт  
// проверим правильность выбора вида договора  
Если Не РольДоступна("Администратор") Тогда  
    Если Роль="Закупки" Тогда  
        Если ((РольДоступна("МенеджерПоЗакупкам")  
или  
РольДоступна("МенеджерПоЗакупкамСОграничениемПравДоступа"  
))  
и (не (РольДоступна("МенеджерПоПродажам")  
или  
РольДоступна("МенеджерПоПродажамСОграничениемПравДоступа"  
)))) Тогда  
            Возврат Истина;  
        КонечЕсли;  
    ИначеЕсли Роль="Продажи" Тогда  
        Если ((РольДоступна("МенеджерПоПродажам")  
или  
РольДоступна("МенеджерПоПродажамСОграничениемПравДоступа"  
))  
и (не (РольДоступна("МенеджерПоЗакупкам")  
или  
РольДоступна("МенеджерПоЗакупкамСОграничениемПравДоступа"  
)))) Тогда  
            Возврат Истина;  
        КонечЕсли;  
    КонечЕсли;  
    Возврат Ложь;  
КонечФункции
```

```
//Возвращает таблицу себестоимости товаров с учетом  
средней рентабельности бартерных сделок
```

```

// срок расчета для бартера берется из параметра.
Если не заполнено, то 3 месяца
// все это нормально будет работать только для ГП- на
филиалах не хватит данных по бартеру
// описание работы механизма см. в справке к отчету
"ОтчетПоСебиЦенамДляМаркетинга"
// Параметры Дата, фильтр по товарам
(необязательный) - массив значений;
Функция
РасчетСебестоимостиСУчетомБартера (мДата, ФильтрТов=Неопред
елено, Склад=Неопределено, КолМес) Экспорт
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
| ПартииТоваровНаСкладахОстатки.Номенклатура КАК
Номенклатура,
|
| ПартииТоваровНаСкладахОстатки.Производит
ель КАК Производитель,
| ПартииТоваровНаСкладахОстатки.СтатусПартии КАК
СтатусПартии,
| ПартииТоваровНаСкладахОстатки.ДокументОприходования
КАК ДокументОприходования,
| ПартииТоваровНаСкладахОстатки.КоличествоОстаток КАК
КоличествоОстаток,
| ПартииТоваровНаСкладахОстатки.СтоимостьОстаток КАК
СтоимостьОстаток,
|
| ПартииТоваровНаСкладахОстатки.ДокументОприходования.Д
оговорКонтрагента КАК ДоговорКонтрагента,
| 0 КАК ПроцентСкидки
| ИЗ
| РегистрНакопления.ПартииТоваровНаСкладах.Остатки (
| &ДатаОтч, ";
Если ФильтрТов<>Неопределено Тогда
Запрос.Текст=Запрос.Текст+"
| (Номенклатура В (&СписТов)) И";
КонецЕсли;
Если Склад<>Неопределено Тогда
Запрос.Текст=Запрос.Текст+"
| (Склад=&Склад) И";
КонецЕсли;
Запрос.Текст=Запрос.Текст+"
|
| (ДокументОприходования.ДоговорКонтрагента.ВидДоговора
<> &ВидДоговора

```

```

        |
        |             ИЛИ
ДокументОприходования.ДоговорКонтрагента.ВидДоговора ЕСТЬ
NULL) ) КАК ПартииТоваровНаСкладахОстатки
    |ГДЕ
    | ПартииТоваровНаСкладахОстатки.КоличествоОстаток > 0
    | И ПартииТоваровНаСкладахОстатки.СтоимостьОстаток >
0
    |
    |УПОРЯДОЧИТЬ ПО
    | Номенклатура
    |АВТОУПОРЯДОЧИВАНИЕ";
Запрос.УстановитьПараметр("ДатаОтч", мДата);
Если ФильтрТов<>Неопределено Тогда
    Запрос.УстановитьПараметр("СписТов", ФильтрТов);
КонецЕсли;
Если Склад<>Неопределено Тогда
    Запрос.УстановитьПараметр("Склад", Склад);
КонецЕсли;
Запрос.УстановитьПараметр("ВидДоговора", Перечисления.
ВидыДоговоровКонтрагентов.Бартерный);
#Если Клиент Тогда
    Состояние("Получение данных по не бартерным
товарам...");
#КонецЕсли
РезультатНеБартер = Запрос.Выполнить().Выгрузить();
Запрос3 = Новый Запрос;
Запрос3.Текст="ВЫБРАТЬ
| ПланирСкидкиПоставщиковСрезПоследних.ПроцСкидки
|ИЗ
|
РегистрСведений.ПланирСкидкиПоставщиков.СрезПоследних
(
    |     &ДатаДок,
    |     ДоговорПоставки = &Договор
    |         И СтатусТовара = &Статус
    |         И Товар = &Товар) КАК
ПланирСкидкиПоставщиковСрезПоследних";
// здесь допишем планируемые скидки
Для каждого детали из РезультатНеБартер Цикл
    #Если Клиент Тогда
        Состояние("Получение данных по скидкам
поставщиков для не бартера...");
    #КонецЕсли
    Если Не
ЗначениеНеЗаполнено(Детали.ДоговорКонтрагента) тогда

```

Запрос3.УстановитьПараметр ("Товар", Детали.Номенклатура);

Запрос3.УстановитьПараметр ("Договор", Детали.ДоговорКонтрагента);

Запрос3.УстановитьПараметр ("Статус", Детали.СтатусПартии);

Запрос3.УстановитьПараметр ("ДатаДок", Детали.ДокументОприходования.Дата);

Скидки=Запрос3.Выполнить ();

Если Не Скидки.Пустой() тогда

Скидка = Скидки.Выбрать ();

Скидка.Следующий ();

Детали.ПроцентСкидки=Скидка.ПроцСкидки;

Детали.СтоимостьОстаток=Детали.СтоимостьОстаток/ (1+Детали.ПроцентСкидки/100);

КонецЕсли;

КонецЕсли;

КонецЦикла;

// теперь нужно посчитать бартерные товары

Запрос = Новый Запрос;

Запрос.Текст ="ВЫБРАТЬ

| ПартииТоваровНаСкладахОстатки.Номенклатура КАК
Номенклатура,

|
ПартииТоваровНаСкладахОстатки.Номенклатура.Производитель КАК
Производитель,

| ПартииТоваровНаСкладахОстатки.СтатусПартии КАК
СтатусПартии,

| ПартииТоваровНаСкладахОстатки.ДокументОприходования
КАК ДокументОприходования,

| ПартииТоваровНаСкладахОстатки.КоличествоОстаток КАК
КоличествоОстаток,

| ПартииТоваровНаСкладахОстатки.СтоимостьОстаток КАК
СтоимостьОстаток,

|
ПартииТоваровНаСкладахОстатки.ДокументОприходования.ДоговорКонтрагента
КАК ДоговорКонтрагента,

| 0 КАК ПроцентСкидки

| ИЗ

| РегистрНакопления.ПартииТоваровНаСкладах.Остатки (

```

|           &ДатаОтч, ";
Если ФильтрТов<>Неопределено Тогда
    Запрос.Текст=Запрос.Текст+"
    | (Номенклатура В (&СписТов)) И";
КонецЕсли;
Если Склад<>Неопределено Тогда
    Запрос.Текст=Запрос.Текст+"
    | (Склад=&Склад) И";
КонецЕсли;
Запрос.Текст=Запрос.Текст+"
|
(ДокументОприходования.ДоговорКонтрагента.ВидДоговора =
&ВидДоговора) ) КАК ПартииТоваровНаСкладахОстатки
| ГДЕ
| ПартииТоваровНаСкладахОстатки.КоличествоОстаток > 0
| И ПартииТоваровНаСкладахОстатки.СтоимостьОстаток >
0
|
| СГРУППИРОВАТЬ ПО
| ПартииТоваровНаСкладахОстатки.Номенклатура,
|
ПартииТоваровНаСкладахОстатки.Номенклатура.Производит
ель,
| ПартииТоваровНаСкладахОстатки.СтатусПартии,
|
ПартииТоваровНаСкладахОстатки.ДокументОприходования,
| ПартииТоваровНаСкладахОстатки.КоличествоОстаток,
| ПартииТоваровНаСкладахОстатки.СтоимостьОстаток,
|
ПартииТоваровНаСкладахОстатки.ДокументОприходования.Д
оговорКонтрагента
|
| УПОРЯДОЧИТЬ ПО
|
ПартииТоваровНаСкладахОстатки.ДокументОприходования.Д
ата,
| Номенклатура
| АВТОУПОРЯДОЧИВАНИЕ";
Запрос.УстановитьПараметр("ДатаОтч", мДата);
Если ФильтрТов<>Неопределено Тогда
    Запрос.УстановитьПараметр("СписТов", ФильтрТов);
КонецЕсли;
Если Склад<>Неопределено Тогда
    Запрос.УстановитьПараметр("Склад", Склад);
КонецЕсли;

```

```

Запрос.УстановитьПараметр ("ВидДоговора", Перечисления.
ВидыДоговоровКонтрагентов.Бартерный);
#Если Клиент Тогда
    Состояние ("Получение данных по бартерным
товарам...");
#КонецЕсли
РезультатБартер = Запрос.Выполнить().Выгрузить();
// теперь рассчитаем себестоимость бартера
// Только для ГП, для филиала бартерную наценку не
рассчитываем
Если ПланыОбмена.ГлавныйУзел()=Неопределено Тогда
    Запрос3 = Новый Запрос;
    Запрос3.Текст="ВЫБРАТЬ
| ПланирСкидкиПоставщиковСрезПоследних.ПроцСкидки
| ИЗ
|
РегистрСведений.ПланирСкидкиПоставщиков.СрезПоследних
(
|     &ДатаДок,
|     ДоговорПоставки = &Договор
|     И Товар = &Товар) КАК
ПланирСкидкиПоставщиковСрезПоследних";
    Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ
| ПродажиПродСумма.Клиент КАК Клиент,
| ПродажиПродСумма.СтоимостьОборот,
| ПродажиСебСумма.СебестоимостьПродажи,
| ПродажиПродСумма.Регистратор,
| ПродажиПродСумма.Номенклатура,
|
ПродажиСебСумма.ДокументОприходованияДоговорКонтрагент
та КАК ДоговорПоставки
| ИЗ
|     (ВЫБРАТЬ
|         ПродажиОбороты.ДоговорКонтрагента.Владелец
КАК Клиент,
|         СУММА (ПродажиОбороты.СтоимостьОборот) КАК
СтоимостьОборот,
|         ПродажиОбороты.Регистратор КАК Регистратор,
|         ПродажиОбороты.Номенклатура КАК Номенклатура
|     ИЗ
|         РегистрНакопления.Продажи.Обороты (
|             &ДатаНач,
|             &ДатаКон,
|             Регистратор,

```

```

| (ДоговорКонтрагента.ВидДоговора =
&ВидДоговора)
| И (ДоговорКонтрагента.Владелец В
(&Клиенты) ) ) КАК ПродажиОбороты
|
| СГРУППИРОВАТЬ ПО
| ПродажиОбороты.ДоговорКонтрагента.Владелец,
| ПродажиОбороты.Регистратор,
| ПродажиОбороты.Номенклатура) КАК
ПродажиПродСумма
| ВНУТРЕННЕЕ СОЕДИНЕНИЕ (ВЫБРАТЬ
|
ПродажиСебестоимостьОбороты.ЗаказПокупателя.Контрагент
т КАК Клиент,
|
СУММА (ПродажиСебестоимостьОбороты.СтоимостьОборот)
КАК СебестоимостьПродажи,
| ПродажиСебестоимостьОбороты.Регистратор
КАК Регистратор,
| ПродажиСебестоимостьОбороты.Номенклатура
КАК Номенклатура,
|
ПродажиСебестоимостьОбороты.ДокументОприходования.Дог
оворКонтрагента КАК
ДокументОприходованияДоговорКонтрагента
| ИЗ
|
РегистрНакопления.ПродажиСебестоимость.Обороты (
| &ДатаНач,
| &ДатаКон,
| Регистратор,
|
(ЗаказПокупателя.ДоговорКонтрагента.ВидДоговора =
&ВидДоговора)
| И
(ЗаказПокупателя.ДоговорКонтрагента.Владелец В
(&Клиенты) ) ) КАК ПродажиСебестоимостьОбороты
|
| СГРУППИРОВАТЬ ПО
|
ПродажиСебестоимостьОбороты.ЗаказПокупателя.Контрагент
т,
| ПродажиСебестоимостьОбороты.Регистратор,
|
ПродажиСебестоимостьОбороты.Номенклатура,

```

```

|
ПродажиСебестоимостьОбороты.ДокументОприходования.Дог
оворКонтрагента) КАК ПродажиСебСумма
|      ПО      ПродажиПродСумма.Регистратор      =
ПродажиСебСумма.Регистратор
|      И      ПродажиПродСумма.Номенклатура      =
ПродажиСебСумма.Номенклатура";

```

```

Запрос.УстановитьПараметр ("ДатаНач", КонецМесяца (Добав
итьМесяц (мДата, -КолМес) ) ) ;

```

```

Запрос.УстановитьПараметр ("ДатаКон", мДата) ;

```

```

Запрос.УстановитьПараметр ("ВидДоговора", Перечисления.
ВидыДоговоровКонтрагентов.Бартерный) ;

```

```

МассивКлиенты = Новый Массив;

```

```

Для каждого детали из РезультатБартер Цикл

```

```

МассивКлиенты.Добавить (Детали.ДоговорКонтрагента.Влад
елец) ;

```

```

КонецЦикла;

```

```

Запрос.УстановитьПараметр ("Клиенты", МассивКлиенты) ;

```

```

#Если Клиент Тогда

```

```

Состояние ("Получение данных по себ. бартерных
товаров...") ;

```

```

#КонецЕсли

```

```

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

```

```

Для Каждого деталиСеб из РезультатСебБартер Цикл

```

```

#Если Клиент Тогда

```

```

Состояние ("Получение данных по скидкам
поставщиков для бартера...") ;

```

```

#КонецЕсли

```

```

Если ЗначениеНеЗаполнено (ДеталиСеб.Номенклатура)

```

```

или

```

```

ЗначениеНеЗаполнено (ДеталиСеб.ДоговорПоставки) Тогда

```

```

Продолжить;

```

```

КонецЕсли;

```

```

Запрос3.УстановитьПараметр ("Товар", ДеталиСеб.Номенкла
тура) ;

```

```

Запрос3.УстановитьПараметр ("Договор", ДеталиСеб.Догово
рПоставки) ;

```

```

Запрос3.УстановитьПараметр ("ДатаДок", ДеталиСеб.Регист
ратор.Дата) ;

```



```
Скидки=Запрос3.Выполнить ();
Если Не Скидки.Пустой() тогда
    Скидка = Скидки.Выбрать ();
    Скидка.Следующий ();
```

```
ДеталиСеб.СебестоимостьПродажи=ДеталиСеб.Себестоимост
ьПродажи/ (1+Скидка.ПроцСкидки/100);
    КонецЕсли;
КонецЦикла;
```

```
РезультатСебБартер.Свернуть ("Клиент", "СтоимостьОборот
, СебестоимостьПродажи");
```

```
Для каждого детали из РезультатБартер Цикл
    ПроцентНаценки=0;
```

```
Для Каждого ДеталиСеб из РезультатСебБартер
Цикл;
```

```
    Если
```

```
ДеталиСеб.Клиент<>Детали.ДоговорКонтрагента.Владелец
Тогда
```

```
        Продолжить;
```

```
    КонецЕсли;
```

```
    ПроцентНаценки=(ДеталиСеб.СтоимостьОборот-
ДеталиСеб.СебестоимостьПродажи) / ДеталиСеб.СебестоимостьПр
одажи;
```

```
        // теперь уменьшим себестоимость проданного
на бартер товара, на процент бартерной наценки
```

```
Детали.СтоимостьОстаток=Детали.СтоимостьОстаток/ (1+Пр
оцентНаценки);
```

```
    КонецЦикла;
```

```
КонецЦикла;
```

```
КонецЕсли; // по ГП
```

```
Результат=РезультатНеБартер.Скопировать ();
```

```
// соберем результаты по бартеру и не бартеру вместе
```

```
Для Каждого Детали из РезультатБартер Цикл
```

```
    Строка=Результат.Добавить ();
```

```
    Строка.Номенклатура=Детали.Номенклатура;
```

```
    Строка.Производитель=Детали.Производитель;
```

```
    Строка.СтатусПартии=Детали.СтатусПартии;
```

```
Строка.ДокументОприходования=Детали.ДокументОприходов
ания;
```

```
Строка.КоличествоОстаток=Детали.КоличествоОстаток;
```

```
Строка.СтоимостьОстаток=Детали.СтоимостьОстаток;
```

```
Строка.ДоговорКонтрагента=Детали.ДоговорКонтрагента;  
КонецЦикла;  
Возврат Результат;  
КонецФункции
```

3.9 Модуль «Логистика»

В данном модуле используется следующий код:

```
// <Описание функции>  
//  
// Параметры  
// <Параметр1> - <Тип.Вид>- <описание параметра>  
//               <продолжение описания параметра>  
// <Параметр2> - <Тип.Вид>- <описание параметра>  
//               <продолжение описания параметра>  
//  
// Возвращаемое значение:  
// <Тип.Вид> - <описание возвращаемого значения>  
//  
Функция      ИнформацияПоячейкам (      СписокНоменклатуры,  
СписокЯчеек) Экспорт  
  
Запрос = Новый Запрос;  
  
Запрос.Текст = "ВЫБРАТЬ  
                |  
РазмещениеПоячейкамСрезПоследних.Номенклатура,  
                |  
РазмещениеПоячейкамСрезПоследних.Ячейка,  
                |  
РазмещениеПоячейкамСрезПоследних.Период  
                |ИЗ  
                |  
РегистрСведений.РазмещениеПоячейкам.СрезПоследних (  
                |           &МоментВремени,  
                |           Номенклатура                               В  
(&СписокНоменклатуры)  
                |           И Ячейка В (&СписокЯчеек)  
КАК РазмещениеПоячейкамСрезПоследних";  
  
Запрос.УстановитьПараметр (  
"СписокНоменклатуры", СписокНоменклатуры);
```

```
Запрос.УстановитьПараметр( "СписокЯчеек",  
СписокЯчеек);
```

```
РезультатЗапроса = Запрос.Выполнить();  
Возврат РезультатЗапроса;
```

```
КонецФункции // ИнформацияПоЯчейкам()
```

```
Функция
```

```
ПроверкаНаличияАдреса (Товар, Серия, Адрес, МоментВремени)
```

```
Экспорт
```

```
// надо составить запрос по существующим данным,  
чтобы избежать двойного удаления или
```

```
// добавления (т.е. ситуации, когда остаток-1 или 2)
```

```
Запрос = новый запрос;
```

```
Запрос.Текст="ВЫБРАТЬ
```

```
| РазмещениеПоЯчейкамОстатки.Номенклатура,
```

```
| РазмещениеПоЯчейкамОстатки.СерияНоменклатуры,
```

```
| РазмещениеПоЯчейкамОстатки.Ячейка,
```

```
| РазмещениеПоЯчейкамОстатки.КоличествоОстаток
```

```
| ИЗ
```

```
| РегистрНакопления.РазмещениеПоЯчейкам.Остатки (
```

```
| &ДатаКон,
```

```
| Номенклатура=&Товар
```

```
| и СерияНоменклатуры = &Серия
```

```
| и Ячейка = &Ячейка) КАК
```

```
РазмещениеПоЯчейкамОстатки";
```

```
Запрос.УстановитьПараметр ("Товар", Товар);
```

```
Запрос.УстановитьПараметр ("Серия", Серия);
```

```
Запрос.УстановитьПараметр ("Ячейка", Адрес);
```

```
Запрос.УстановитьПараметр ("ДатаКон", МоментВремени);
```

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

```
Если Результат.Пустой() тогда
```

```
Возврат Ложь;
```

```
КонецЕсли;
```

```
Выборка = Результат.Выбрать();
```

```
Выборка.Следующий();
```

```
Если Выборка.КоличествоОстаток<>0 Тогда
```

```
Возврат Истина;
```

```
КонецЕсли;
```

```
КонецФункции
```

3.10 Модуль «Бизнес процессы»

```
// <Описание процедуры>
//
// Параметры
// <Параметр1> - <Тип.Вид>- <описание параметра>
//               <продолжение описания параметра>
// <Параметр2> - <Тип.Вид>- <описание параметра>
//               <продолжение описания параметра>
//
Процедура СоздатьБизнесПроцесс (Объект, ТочкаМаршрута
= 0) Экспорт

    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        | Задача2.БизнесПроцесс,
        | Задача2.ТочкаМаршрута,
        | Задача2.Выполнена
        | ИЗ
        | Задача.Задача2 КАК Задача2
        | ГДЕ
        | Задача2.БизнесПроцесс.ПриходныйОрдерНаТовары
&ПриходныйОрдерНаТовары";

    Запрос.УстановитьПараметр ("ПриходныйОрдерНаТовары",
Объект.Ссылка);
    РезультатЗапроса = Запрос.Выполнить();
    Если РезультатЗапроса.Пустой() = Ложь Тогда

        Отказ = Истина;
        //Сообщить ("БП существует!");

    Иначе

        БизнесПроцесс
        БизнесПроцессы.ПоступлениеТоваровИУслуг.СоздатьБизнесПроц
        есс();
        БизнесПроцесс.Дата = ТекущаяДата();
        БизнесПроцесс.ПриходныйОрдерНаТовары
        Объект.Ссылка;
        БизнесПроцесс.Контрагент = Объект.Контрагент;
        //БизнесПроцесс.Договор
        Объект.ДоговорКонтрагента;
```

```

        БизнесПроцесс.Записать ();

        БизнесПроцесс.Старт (БизнесПроцессы.ПоступлениеТоваров
ИУслуг.ТочкиМаршрута.Старт2);
        БизнесПроцесс.АктивироватьИнтерактивно ();

        КонечЕсли;

        КонечПроцедуры // СоздатьБизнесПроцесс ()
        // <Описание процедуры>
        //
        // Параметры
        // <Параметр1> - <Тип.Вид>- <описание параметра>
        // <продолжение описания параметра>
        // <Параметр2> - <Тип.Вид>- <описание параметра>
        // <продолжение описания параметра>
        //
        Процедура
        СоздатьБизнесПроцессПоступлениеТоваровПоПриходномуОрдеру (
        Объект, ТочкаМаршрута = 0) Экспорт
            БизнесПроцесс
        БизнесПроцессы.ПриходныйОрдерНаТовары.СоздатьБизнесПроцес
        с ();
            БизнесПроцесс.Дата = ТекущаяДата ();
            БизнесПроцесс.ПоступлениеТоваровИУслуг
        Объект.Ссылка;
            БизнесПроцесс.Контрагент = Объект.Контрагент;
            БизнесПроцесс.Договор = Объект.ДоговорКонтрагента;

        БизнесПроцесс.Записать ();
        БизнесПроцесс.Старт (БизнесПроцессы.ПоступлениеТоваров
ИУслуг.ТочкиМаршрута.Старт2);
        БизнесПроцесс.АктивироватьИнтерактивно ();
        КонечПроцедуры // СоздатьБизнесПроцесс ()
        // <Описание процедуры>
        //
        // Параметры
        // <Параметр1> - <Тип.Вид>- <описание параметра>
        // <продолжение описания параметра>
        // <Параметр2> - <Тип.Вид>- <описание параметра>
        // <продолжение описания параметра>
        //

```

```

Процедура
ОповеститьОКонтролеСкладаПоступлениеТоваровУслуг ( )
Экспорт

    Запрос = Новый Запрос;

    Запрос.Текст = "ВЫБРАТЬ
                    |
ПоступлениеТоваровИУслугТочки.Ссылка
                    |ИЗ
                    |
БизнесПроцесс.ПоступлениеТоваровИУслуг.Точки           КАК
ПоступлениеТоваровИУслугТочки"

КонецПроцедуры
// <Описание процедуры>
//
// Параметры
// <Параметр1> - <Тип.Вид>- <описание параметра>
//                <продолжение описания параметра>
// <Параметр2> - <Тип.Вид>- <описание параметра>
//                <продолжение описания параметра>
//
Процедура
УправлениеБизнесПроцессомПоступлениеТоваровСертификация (
Объект, ТаблицаСерий) Экспорт
    //
    ДокументПриходныйОрдер =
Объект.ПоступлениеТоваровИУслуг;
    //Если существует поступление товаров...

    ДокументОприходывания =
НайтиПодчиненныйДокументПоВиду (ДокументПриходныйОрдер.Ссы
лка, "ПоступлениеТоваровУслуг", "Сделка");
    //Определим все ли товары сертифицированы?

    Для каждого Строка Из ТаблицаСерий Цикл
        КонецЦикла;
    КонецПроцедуры //
УправлениеБизнесПроцессомПоступлениеТоваровСертификация (
)

```

4 Техничко–экономическое обоснование

4.1 Цель проекта

Целью данного дипломного проекта является создание конфигурации для 1С Предприятия 8, предназначенным для департамента логистики фармацевтической компании. Разработка приложения производится в конфигураторе 1С Предприятие 8, на встроенном языке.

4.2 Трудовые ресурсы, используемые в работе

В работе над данным проектом задействованы:

- руководитель проекта– постановка задачи и разработка схемы;
- программист– разработка алгоритмов и программирование.

Количество сотрудников, задействованных в проекте, и их месячная заработная плата представлено в таблице 4.1.

Таблица 4.1 – Данные о сотрудниках

Должность	Количество	Зарботная плата в месяц
Руководитель проекта	1	140 000
Программист	2	110 000
Итого	3	360 000

4.3 Оборудование, используемое в работе

Характеристики оборудования, используемого в работе, а также его стоимость приведены в таблице 4.2.

Таблица 4.2 – Перечень оборудования

Название оборудования	Характеристики	Количество	Стоимость за единицу с учетом НДС, тенге
НОУТБУК LENOVO G580	Intel® Core™ i5–3210M, 6 Gb DDR3, 1000 Gb HDD, NVIDIA® GeForce® GT635 (2 ГБ)	1	136 900
НОУТБУК LENOVO G580	Intel® Core™ i5–3210M, 6 Gb DDR3, 1000 Gb HDD, NVIDIA® GeForce® GT635 (2 ГБ)	1	136 900

НОУТБУК LENOVO G580	Intel® Core™ i5–3210M, 6 Gb DDR3, 1000 Gb HDD, NVIDIA® GeForce® GT635 (2 ГБ)	1	136 900
Итого		4	410700

4.4 Программное обеспечение, используемое в работе

При разработке приложения используется данное следующее программного обеспечения:

- Windows 8.1– операционная система
- 1С Предприятие 8– управление торговлей для Казахстана (USB)
- 1С Предприятие 8– клиентская лицензия на 50 рабочих мест

Программное обеспечение, использованное в работе и соответствующая ему стоимость приведены в таблице 4.3.

Таблица 4.3 – Перечень программного обеспечения

Наименование	Количество копий	Стоимость с учетом НДС, тенге
Windows 8.1	3	Предустановлено, цена ПО входит в стоимость ноутбука
1С Предприятие 8 управление торговлей для Казахстана (USB)	1	85 200
1С Предприятие 8 клиентская лицензия на 50 рабочих мест	1	828 000
Итого		913200

4.5 Сроки реализации проекта

Процесс проектирования и разработки конфигурации включает в себя 5 этапов:

- Постановка задачи и сбор данных;
- Разработка модулей;
- Разработка интерфейса;
- Тестирование на работоспособность;
- Оформление и сдача отчета.

В таблице 4.4 приведен график реализации проекта.

Таблица 4.4 – Этапы и сроки реализации проекта

Перечень работ		Неделя от начала работ							
1 этап	Постановка задачи	■	■						
	Анализ предметной области	■	■						
	Изучение литературы	■	■						
	Планирование		■						
2 этап	Разработка модулей			■	■	■			
	Внедрение в общую конфигурацию				■	■			
3 этап	Разработка интерфейса				■	■			
4 этап	Тестирование модулей					■	■	■	■
	Отладка модулей						■	■	■
5 этап	Оформление и сдача отчета								■

4.6 Расчет затрат и стоимости работ по реализации проекта

Разработка конфигурации под 1С Предприятие 8 требует большого количества интеллектуальных затрат сотрудников, выполняющих работу, а также необходимых технических средств для ее реализации. Все это требует финансовых вложений, на основе которых высчитывается конечная стоимость проекта.

Затраты на разработку данного приложения вычисляются по формуле:

$$C = \text{ФОТ} + C_{\text{н}} + A + \text{Э} + N \quad (4.1)$$

где ФОТ – фонд оплаты труда;

$C_{\text{н}}$ – социальный налог;

A – амортизационные отчисления;

Э – затраты на электроэнергию;

$C_{\text{пр}}$ – прочие расходы;

N – накладные расходы.

4.6.1 Расчет фонда оплаты труда

ФОТ складывается из основной и дополнительной заработной платы сотрудников и рассчитывается по формуле:

$$\text{ФОТ} = \text{З}_{\text{осн}} + \text{З}_{\text{доп}} \quad (4.2)$$

Для расчета затрат на основную заработную плату используются данные о средневзвешенном заработке и фактическом времени работы каждого сотрудника.

Среднедневной заработок:

$$D = \frac{\text{ЗПм}}{D_p} \quad (4.3)$$

где ЗПм – ежемесячный размер заработной платы;
D_p – количество рабочих дней в месяце (21 день).

1) Руководитель проекта:

$$D = \frac{140000}{21} = 6667 \text{ тенге/день};$$

2) Программист:

$$D = \frac{110000}{21} = 5238 \text{ тенге/день}.$$

Заработная плата за один час работы сотрудника рассчитывается по формуле:

$$H = \frac{D}{\text{Ч}_p} \quad (4.4)$$

где D – средний дневной заработок работника;
Ч_p – количество часов рабочего дня (8 часов).

Руководитель проекта:

$$H = \frac{6667}{8} = 833 \text{ тенге/час};$$

Программист:

$$H = \frac{5238}{8} = 655 \text{ тенге/час}.$$

Длительность цикла в днях по каждому виду работ определяется по формуле:

$$t_n = \frac{T}{q_n * z * K} \quad (4.5)$$

где T – трудоемкость этапа, норма-час;

q_n – количество исполнителей по этапу;
 Z – продолжительность рабочего дня, $z = 8$ часов;
 K – коэффициент выполнения норм времени, $K = 1,1$.

Полученную величину t_n округляем в большую сторону до целых дней.

$t_1 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2$ дня; Руководитель проекта, постановка задачи;

$t_2 = \frac{24}{2 \cdot 8 \cdot 1,1} \approx 3$ дня; Руководитель проекта, программист, анализ предметной области;

$t_3 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2$ дня; Программист, изучение литературы;

$t_4 = \frac{8}{1 \cdot 8 \cdot 1,1} \approx 1$ день; Программист, планирование;

$t_5 = \frac{48}{1 \cdot 8 \cdot 1,1} \approx 6$ дней; Программист, разработка модулей;

$t_6 = \frac{24}{1 \cdot 8 \cdot 1,1} \approx 3$ дня; Программист, внедрение в общую конфигурацию;

$t_7 = \frac{24}{1 \cdot 8 \cdot 1,1} \approx 3$ дня; Программист, разработка интерфейса;

$t_8 = \frac{96}{1 \cdot 8 \cdot 1,1} \approx 12$ дней; Программист, тестирование модулей;

$t_9 = \frac{88}{1 \cdot 8 \cdot 1,1} \approx 11$ дней; Программист, отладка модулей;

$t_{11} = \frac{8}{1 \cdot 8 \cdot 1,1} \approx 1$ день; Руководитель проекта, оформление и сдача отчета.

В таблице 4.5 приведены сводные результаты расчета затрат на основную заработную плату сотрудников.

Таблица 4.5 – Сводные результаты расчета затрат на основную заработную плату

Наименование этапов работ	Исполнитель	Трудоемкость		Длительность цикла, дни	Зар. плата за час работы, тенге	Сумма зар. платы, тенге
		Нормы – часы	% от общей трудоемкости			
Постановка задачи	Руководитель проекта	16	5	2	833	13328
Анализ предметной области	Программист	24	5	3	655	15720
Анализ предметной области	Анализ предметной области	24	5	3	833	19992

Изучение литературы	Программист	16	2,5	2	655	10480
Планирование	Программист	8	2,5	1	655	5240
Разработка модулей	Программист	48	40	6	655	31440
Внедрение в общую конфигурацию	Программист	24	2,5	3	655	15720
Разработка интерфейса	Программист	24	5	3	655	15720
Тестирование модулей	Программист	96	20	12	655	62880
Отладка модулей	Программист	88	15	6	655	57640
Оформление и сдача отчета	Руководитель проекта	8	2,5	1	833	6664
Итого		376	100	42		254824

Дополнительная заработная плата составляет 10% от основной заработной платы и рассчитывается по формуле:

$$Z_{\text{доп}} = Z_{\text{осн}} * 0,1 \quad (4.6)$$

$$Z_{\text{доп}} = 254824 * 0,1 = 25482,4 \text{ тенге}$$

В результате расчетов, согласно формуле 4.2, суммарный фонд оплаты труда составит:

$$\text{ФОТ} = 205488 + 20548,8 = 280306,4 \text{ тенге}$$

4.6.2 Расчет затрат по социальному налогу

Социальный налог составляет 11% от дохода сотрудника и рассчитывается по формуле:

$$C_{\text{н}} = (\text{ФОТ} - \text{ПО}) * 11\% \quad (4.7)$$

где ПО – пенсионные отчисления, который составляют 10% от ФОТ и не облагаются социальным налогом, рассчитываются по формуле:

$$\text{ПО} = \text{ФОТ на } 10\%$$

$$\text{ПО} = 280306,4 * 0,1 = 28030,64 \text{ тенге.}$$

Таким образом социальный налог составит:

$$C_{\text{н}} = (280306,4 - 28030,64) * 0,11 = 277223,02 \text{ тенге.}$$

4.6.3 Расчет амортизационных отчислений

Амортизационные отчисления рассчитываются по формуле:

$$A_i = \frac{H_A * C_{пер} * N}{100 * 12 * n} \quad (4.9)$$

где H_A – норма амортизации;
 $C_{пер}$ – первоначальная стоимость оборудования;
 N – количество дней на выполнение работ;
 n – количество рабочих дней в месяце.

Следовательно, амортизационные отчисления по используемому оборудованию и ПО, в соответствии с формулой 4.9 составят:

На оборудование:

$$A_1 = \frac{40 * 410700 * 40}{100 * 12 * 21} = 26076 \text{ тенге.}$$

На программное обеспечение:

$$A_2 = \frac{40 * 913200 * 40}{100 * 12 * 21} = 57980 \text{ тенге.}$$

Суммарные затраты на амортизацию рассчитываются по формуле 3.10:

$$A = A_1 + A_2 \quad (4.10)$$

$$A = 26076 + 57980 = 84056 \text{ тенге.}$$

4.6.4 Расчет затрат на электроэнергию

Так как в процессе реализации проекта используется техническое оборудование, необходимо рассчитать затраты на электроэнергию, потребляемую данным оборудованием.

Для расчета затрат на электроэнергию используется формула 4.11.

$$\mathcal{E} = \mathcal{E}_{эл.эн.об.} + \mathcal{E}_{доп.} \quad (4.11)$$

где $\mathcal{E}_{эл.эн.об.}$ – затраты на электроэнергию для оборудования;

$\mathcal{E}_{доп.}$ – затраты электроэнергии на дополнительные нужды;

Расходы электроэнергии на оборудование рассчитываются по формуле 4.12.

$$\mathcal{E}_{эл.эн.об.} = W * T * S * K_{исп} \quad (4.12)$$

где W – потребляемая оборудованием мощность, кВт;

T – время работы, часы;

S – тариф (1кВт/час = 16,9 тенге);

$K_{исп}$ – коэффициент использования ($K_{исп} = 0,9$).

Потребляемая мощность LENOVO G580 составляет 90 Вт.

Время высчитывается на основе количества рабочих дней и рабочих часов в день.

Таким образом общая сумма затрат на электроэнергию для оборудования:

$$\mathcal{E}_{эл.эн.об.} = (0,09 + 0,065) * (40 * 8) * 16,9 * 0,9 = 754,4 \text{ тенге.}$$

Затраты на дополнительные нужды берутся в размере 5% от затрат на оборудование и рассчитываются по формуле:

$$Z_{\text{доп.}} = Z_{\text{эл.эн.об.}} * 5\% \quad (4.13)$$

И составляют:

$$Z_{\text{доп.}} = 754,4 * 0,05 = 37,7 \text{ тенге.}$$

Суммарные затраты на электроэнергию составляют:

$$\Theta = 754,4 + 37,7 = 792,1 \text{ тенге.}$$

4.6.5 Расчет накладных расходов

Накладные расходы рассчитываются в размере 50% от всех затрат.

$$H = (\text{ФОТ} + O_c + A + \Theta) * 50\% \quad (4.14)$$

$$H = (226036,8 + 22603,68 + 23711 + 792,1) * 0,5 = 272747,53 \text{ тенге.}$$

4.6.6 Суммарные затраты на реализацию проекта

Таким образом себестоимость разработки данного приложения, согласно формуле 4.1 составляет:

$$C = 226036,8 + 22377,64 + 23711 + 792,1 + 272747,53 = 545665,07 \text{ тенге}$$

Сводные результаты расчета стоимости разработки приложения и их структура представлены на рисунке 4.1 и в таблице 4.6:



Рисунок 4.1 – Структура себестоимости конфигурации

Таблица 4.6– Затраты на разработку приложения

Наименование затрат	Сумма, тенге
ФОТ	280306,4
Социальный налог	277223,02
Амортизационные отчисления	84056
Затраты на электроэнергию	792,1
Накладные расходы	272747,53
Итого	915125,05

4.6.7 Цена реализации проекта

Цена проекта складывается из себестоимости и желаемого чистого дохода.

$$C_{\text{п}} = C + \Pi \quad (4.15)$$

Для определения начальной цены используется желаемый уровень рентабельности. Для данной отрасли он составляет 25%.

$$C_{\text{п}} = C * \left(1 + \frac{P}{100}\right) \quad (4.16)$$

$$C_{\text{п}} = 915125,05 * \left(1 + \frac{25}{100}\right) = 1143906,31$$

Цена реализации проекта складывается из цены проект и НДС, рассчитывается по формуле:

$$C_{\text{р}} = C_{\text{п}} + \text{НДС} \quad (4.17)$$

где НДС – налог на добавленную стоимость.

Согласно Налоговому Кодексу Республики Казахстан НДС составляет 12%, то есть в данном случае равен:

$$\text{НДС} = 137268,75 \text{ тенге.}$$

В итоге получаем цену реализации проекта равной:

$$C_{\text{р}} = 1143906,31 + 137268,75 = 1281174,75 \text{ тенге.}$$

Выводы

Итоговая стоимость разработки конфигурации 1С Предприятие для компании «Интерфарма–К» составила 1281174,75 тенге, в которую заложены все возможные затраты при разработки программного продукта.

Наибольшую стоимость проекта составляют затраты на программное обеспечение, более 50%, задействованных в процессе данной разработки.

Разработка конфигурации 1С Предприятие является дорогим проектом, требующих больших интеллектуальных и финансовых затрат, цена реализации 545665,07 тенге. Однако все вложенные средства легко окупаются при условии высокого качества продукта и активного использования, позволяя получить прибыль по истечению определенного времени.

5 Безопасность жизнедеятельности

5.1 Анализ условий труда при разработке конфигурации 1С Предприятие

Разработка конфигурации «Фармацевтическая информационная система» для компании «Интерфарма–К» осуществляется с использованием компьютерной техники и электронного оборудования. В рассматриваемом помещении работает 3 сотрудника, двое из которых имеют свое рабочее место.

При разработке прикладного приложения большую роль играет правильная организация условий труда в рабочем помещении. Так как данная работа связана с длительным нахождением за компьютером, необходимо учесть нормы естественного и искусственного освещения, для обеспечения благоприятных условий работы.

Характеристики рабочего помещения:

Рассматриваемое рабочее помещение, расположено в здании, находящемся вдали от железнодорожных путей или нагруженных автомагистралей, аэропорта и так далее, поэтому внешних источников шума, влияющих на процесс работы нет.

План помещения представлен на рисунке 5.1.

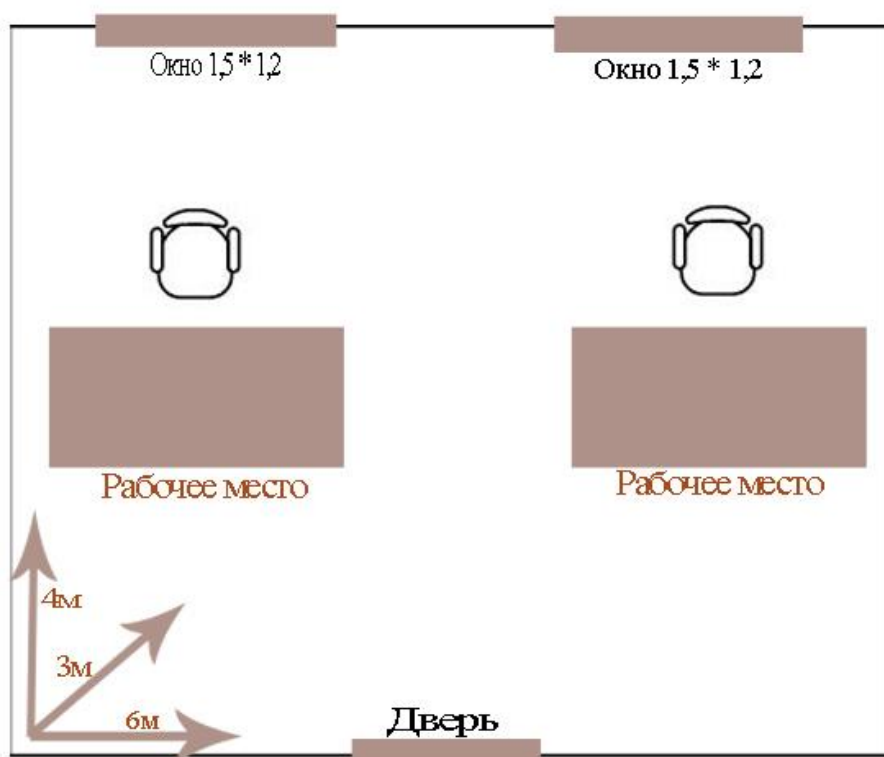


Рисунок 5.1 – План помещения

- Помещение имеет следующие параметры:
- находится на первом этаже одноэтажного здания;
- размеры помещения (комнаты): длина 6 м, ширина 4 м, высота 3м;
- вид свет пропускающего материала стекло листовое, двойное;
- вид переплета стальные, двойные, открывающиеся;
- солнцезащитные устройства жалюзи;
- два окна размером 1,5*1,2;
- внутренняя отделка стен– светлая краска;
- помещение по зрительным условиям работы относится к категории легких работ (легкая физическая, категория Ia, работа производится сидя и не требует физического напряжения) (ГОСТ 12.2.032–78);

Характеристики используемого в работе оборудования:

НОУТБУК LENOVO G580

- Тип процессора Core i3 / Pentium / Celeron
- Код процессора 2020M / 1005M / 3110M
- Ядро процессора Ivy Bridge
- Частота процессора 1900...2400 МГц
- Количество ядер процессора 2
- Объем кэша L2512 Кб
- Объем кэша L32 Мб / 3 Мб
- Размер оперативной памяти 2...4 Гб
- Тип памяти DDR3
- Частота памяти 1600 МГц
- Максимальный размер памяти 16 Гб
- Количество слотов памяти 2
- Размер экрана 15.6 дюйм
- Разрешение экрана 1366x768
- Широкоформатный экран есть
- Тип экрана глянцевый
- Сенсорный экран нет
- Мультиязык–экран нет
- Светодиодная подсветка экрана есть
- Поддержка 3D нет
- Тип видеоадаптера / встроенный / дискретный
- Видеопроцессор / Intel HD Graphics 4000 / NVIDIA GeForce 710M
- Два видеоадаптера нет
- электропитание: переменное напряжение 220–250 В, частотой 50 Гц и мощностью 90 Вт и 65 Вт соответственно;

НОУТБУК LENOVO G580

- Процессор
- Тип процессора Core i3 / Pentium / Celeron
- Код процессора 2020M / 1005M / 3110M

- Ядро процессора Ivy Bridge
- Частота процессора 1900...2400 МГц
- Количество ядер процессора 2
- Объем кэша L2 512 Кб
- Объем кэша L3 2 Мб / 3 Мб
- Память
- Размер оперативной памяти 2...4 Гб
- Тип памяти DDR3
- Частота памяти 1600 МГц
- Максимальный размер памяти 16 Гб
- Количество слотов памяти 2
- Размер экрана 15.6 дюйм
- Разрешение экрана 1366x768
- Широкоформатный экран есть
- Тип экрана глянцевый
- Сенсорный экран нет
- Мультиязычный экран нет
- Светодиодная подсветка экрана есть
- Поддержка 3D нет
- Тип видеоадаптера / встроенный / дискретный
- Видеопроцессор / Intel HD Graphics 4000 / NVIDIA GeForce 710M
- Два видеоадаптера нет
- электропитание: переменное напряжение 220–250 В, частотой 50 Гц и мощностью 90 Вт и 65 Вт соответственно;
- Принтер HP Color LaserJet Pro CP1025
- Общие характеристики
- Устройство принтер
- Тип печати цветная
- Технология печати лазерная
- Размещение настольный
- Область применения малый офис
- Количество страниц в месяц 15000
- Максимальный формат А4
- Количество цветов 4
- Максимальное разрешение для ч/б печати 600x600 dpi
- Максимальное разрешение для цветной печати 600x600 dpi
- Скорость печати 16 стр/мин (ч/б А4), 4 стр/мин (цветн. А4)
- Подача бумаги 150 лист. (стандартная)
- Вывод бумаги 50 лист. (стандартный)
- Плотность бумаги 176 г/м²

- Печать на карточках, пленках, этикетках, фотобумаге, глянцевой бумаге, конвертах, матовой бумаге
- Ресурс цветного картриджа/тонера 1000 страниц
- Ресурс ч/б картриджа/тонера 1200 страниц
- Количество картриджей 4
- Тип картриджа/тонера картриджи HP 126A: черный CE310A, голубой CE311A, желтый CE312A, малиновый CE313A
- Объем памяти 8 Мб, максимальный 8 Мб
- Процессор RISC
- Частота процессора 266 МГц
- Интерфейсы
- Интерфейсы USB 2.0
- Веб–интерфейс есть
- Телелефон ZyxEL V301–T1
- Тип VoIP–телефон
- Поддержка SIP есть, подключение к нескольким серверам
- Интерфейсы WAN, LAN
- Подключение гарнитуры есть
- Встроенная телефонная книга есть, 200 контактов
- Поддержка PoE есть
- Удержание, ожидание вызова есть
- Конференц–связь есть
- Громкая связь есть
- Web–интерфейс есть
- Поддержка NAT есть
- LCD–дисплей монохромный
- Размеры (ШхДхВ) 165x208x92 мм
- Вес 465 г

Электротехническое оборудование является потенциальным источником возникновения пожарной опасности. Оборудование малошумящее– вредность в качестве повышенного шума отсутствует.

5.2 Расчет естественного освещения

Естественное освещение не обеспечивает в течение всего рабочего времени необходимого освещения, так как может измениться погода, либо работы могут быть в позднее время, когда уже темнеет и естественного освещения может быть не достаточно, поэтому в рабочем помещении предусмотрена система искусственного общего освещения, состоящая из светильников с люминесцентными лампами. Нормативы на источники света приведены в таблице 5.1.

Таблица 5.1 – Рекомендуемые источники света при системе общего освещения (СНиП РК 2.04.–05–2002)

Характеристика зрительной работы по требованию к цветоразличию	Освещенность, лк	Диапазон цветов температуры источника света $T_c, ^\circ K$	Применяемый тип источника света
Различие цветных объектов при невысоких требованиях к цветоразличию	300, 400	3500– 5500	ЛД, ЛДЦ, ЛБ,
	150, 200	3000– 4500	ЛБ, (ЛХБ), НЛВД+МТЛ, ДРЛ

Здание относится к I степени огнестойкости (СНиП РК 2.02–05–2002) (Здания с несущими и ограждающими конструкциями из естественных или искусственных материалов, бетона или железобетона с применением листовых негорючих материалов). Рабочее помещение по вопросам пожарной безопасности относится к классу “Д”. В соответствии с типовыми правилами пожарной безопасности административные здания и отдельные помещения, и технологические установки обеспечиваются первичными средствами пожаротушения согласно нормативам.

Рассчитаем площадь боковых световых проемов помещения, необходимую для создания нормируемой освещенности на рабочем месте.

Помещение имеет размеры: длина $L = 6,5 \text{ м}$, ширина $B = 4 \text{ м}$, высота $H = 3 \text{ м}$. Высота рабочей поверхности над уровнем пола $h_{\text{рп}}$, $h_{\text{рп}} = 0,75 \text{ м}$, окно начинаются с высоты $h_{\text{но}}$, $h_{\text{но}} = 0,8 \text{ м}$, высота окна $h_{\text{о}}$, $h_{\text{о}} = 1,5 \text{ м}$. Рабочее помещение находится в IV часовом поясе– в г. Алматы (пояс светового климата– IV 50^0 северной широты и южнее).

Рабочие места расположены в $l_{\text{рт}}$, $l_{\text{рт}} = 0,5 \text{ м}$ от наружной стены помещения, где проектируем оконные проемы. Минимальная освещенность будет в точке, отстоящей на расстояние 4 м от оконного проема.

Общую площадь окон $S_0, \text{ м}^2$, определим по формуле:

$$100 \frac{S_0}{S_n} = \frac{e_n \eta_0}{\tau_0 r_1} k_{3д} k_3 \quad (5.1)$$

$$S_0 = \frac{S_n e_n \eta_0}{100 \tau_0 r_1} k_{3д} k_3 \quad (4.2)$$

где: S_n – площадь помещения м^2 ;

$$S_n = 24 \text{ м}^2;$$

e_n – нормированное значение КЕО, выбираемое по таблице 5.1.

Для высокой точности зрительных работ принимаем $e_n = 1,2$

m_N – коэффициент светового климата, который находится по таблице 5.2.

Учитывая заданный световой пояс, приняв ориентацию световых проемов на Север, определим: $m_N = 0,9$.

$$e_n^{IV} = e_n \cdot m \cdot c \quad (5.3)$$

$$m = 0,9;$$

$$c = 0,8 \text{ (в наружных стенах зданий);}$$

$$e_n = 1,2 \text{ для работ высокой точности III разряда зрительной работы;}$$

$$e_n^{IV} = 0,86;$$

Учитывая тип помещения, найдем коэффициент запаса по таблице 5.3
 $k_3 = 1,2$

τ_0 – общий коэффициент светопропускания равный $\tau_0 = \tau_1 \tau_2 \tau_3 \tau_4$;

$$\tau_1 = 0,5 \text{ (пустотелые стеклянные блоки);}$$

$$\tau_2 = 0,6 \text{ (деревянные двойные раздельные переплеты);}$$

$$\tau_3 = 0,8 \text{ (железобетонные фермы и арки);}$$

$$\tau_4 = 1 \text{ (убирающиеся регулируемые жалюзи и шторы);}$$

$$\tau_0 = 0,24;$$

η_0 – световая характеристика окон.

Определяем η_0 :

$$l = 3 \text{ м}$$

$$\frac{L}{l} = \frac{L}{B1} = \frac{6}{3} = 2$$

$$h_{\text{расч}} = h_{\text{но}} + h_o - h_{\text{рп}} \quad (5.4)$$

$$h_{\text{расч}} = 1,57$$

$$\frac{B}{h_{\text{расч}}} = \frac{4}{2,3} = 1,7$$

По таблице 5.2 определим $\eta_0 = 10$.

Γ_1 – коэффициент, учитывающий повышение КЕО при боковом освещении благодаря свету, отраженному от поверхностей помещения и подстилающего слоя, прилегающего к зданию.

Средний коэффициент отражения в помещении $\rho_{\text{СР}} = 0,5$, принимаем одностороннее боковое освещение.

$$\frac{l_{\text{рт}}}{B} = \frac{0,5}{4} = 0,125$$

Тогда $r_1 = 1,05$

$k_{\text{зд}}$ – коэффициент, учитывающий затенение окон противостоящими зданиями.

Поскольку затеняющих зданий поблизости нет, то $k_{\text{зд}} = 1$.

Вычислим общую площадь окон:

$$S_0 = 11,05 \text{ м}^2$$

Площадь световых проемов равна $S_{\text{сп}} = 11,05 \text{ м}^2$

Таким образом площадь световых проемов ($3,6 < 11,05$) не обеспечивает необходимых условий труда на рабочих местах. С целью создания наиболее благоприятных условий труда в помещении с параметрами $6 \times 4 \times 3$ для обеспечения нормированного значения КЕО, $e_N = 0,84$ при III характеристике зрительных работ, совместно с естественным освещением используется искусственное освещение.

Выводы

В данном разделе был произведён анализ условий труда в офисе компании и выполнен расчет параметров освещения с целью проверить их соответствие принятым стандартам. Уровень условий труда признан допустимым в случае использования дополнительного искусственного освещения дополнительно к естественному.

Расчёты показали, что два окна размером $1,5 \times 1,2 \text{ м}$ не соответствует нормативам естественного освещения рабочего помещения. Для обеспечения необходимой освещенности в дневное время суток используется искусственное освещение.

Заключение

Во время выполнения данного дипломного проекта были изучены основные принципы разработки конфигурации 1С Предприятия, а также требований, предъявляемых к ним. Изучены принципы построения и работы ПП для платформы 1С Предприятие. Помимо этого, было также установлено, что исследуемая тема является в наивысшей степени актуальной в реалиях современного рынка.

Также была разработана конфигурация для соответствующей платформы, которая, внедрена в работу компании «Интерфарма– К». Данное приложение отвечает всем стандартам и требованиям заказчика. Предоставляет удобный способ работы с информацией и позволяет гибко изменять его интерфейс под нужды пользователя. Имеет в своем наличии уникальные функции, которые не доступны в стандартных конфигурациях.

Была рассчитана общая стоимость продукта, в основе которого суммирование различных видов затрат, возможных при работе. Данный расчет соответствует реализации продукта для нужд заказчика. В процессе исследования аналогичных продуктов на рынке, установлено, что стоимость разработки полностью соответствует качеству разрабатываемой конфигурации и позволяет получить большой объем прибыли и сократить время работы.

В разделе «Безопасность жизнедеятельности» были рассчитаны необходимые условия для организации соответствующего всем установленным нормам, рабочего места для разработчиков, занятых разработкой проекта. Рассчитано необходимое естественное освещение, требующиеся для комфортной работы разработчиков.

Список используемой литературы

1. Сайт <http://v8.1c.ru/>
2. Сайт <http://1cprogrammistu.ru/>
3. Сайт www.1c.ru
4. Бадагуев Б.Т. Документация по охране труда в организации.– М.: Альфа–пресс, 2010.
5. Дюсебаев М.К., Бегимбетова А.С. Методические указания к выпускной работе (для студентов всех форм обучения специальностей 050719– Радиотехника электроника и телекоммуникации, 050704– Вычислительная техника и программное обеспечение).– Алматы: АИЭС, 2008.
6. СНиП РК 2.04–05–2002 «Естественное и искусственное освещение. Общие требования.– Астана, 2002.
7. Коробко, В.И. Охрана труда: Учебное пособие для студентов вузов / В.И. Коробко.– М.: ЮНИТИ–ДАНА, 2013.– 239 с.
8. Абдимуратов Ж.С, Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат.– Алматы: АИЭС, 2009.
9. Абдимуратов Ж. С., Маманбаева С. Е. Расчет производственного помещения.– Алматы: АУЭС, 2009.
10. Айдарханова М. Основы экономической теории.– М.: Фолиант, 2010.
11. Махотина М.В., Симоненко В.И. Экономика в схемах: Учебное пособие.– М.: Эксмо, 2011.
12. Носова С.С. Экономическая теория.– М.: Кнорус, 2010.
13. Артамонова В.С., Иванова С.А. Экономическая теория.– СПб.: Питер, 2010.
14. Камаева В.Д., Лобачевой Е.И. Экономическая теория: Учебник. – М.: Юрайт, 2010.
15. Выпускная работа бакалавров. Экономический раздел: методичка/ Под ред. Бабич А.А., Казыкен Б.Б., Сагира А.А.– Алматы: АУЭС, 2009.

Приложение А

Вход в 1С Предприятие смотрите рисунок А1.

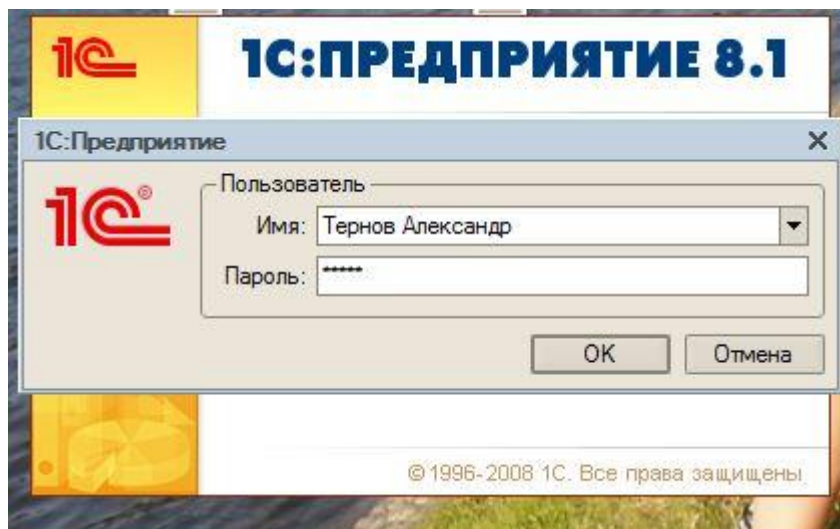


Рисунок А1

Анализ цен смотрите рисунок А2.

Анализ цен (29.05.2014)

На дату: 29.05.2014 в валюте USD

Данные на конец дня 29.05.2014
 Отбор: Строки: Номенклатура Иерархия, Хара...
 номенклатуры Элементы
 Колонки: Контрагент Элементы, Тип це...
 регистрации цены Элементы
 Показатели: Цена в USD, Отклонение от...
 %%

Характеристика номенклатуры	Центрическая Компания «Медсервис»		ООО «Л-Фарма»		ООО «ВИВА ФАРМА»		ИИКАР ООО		АО «Интерфарма.Кз»		ООО «Фирма "Санжар"»				
	Приходная		Приходная		Приходная		Приходная		Приходная		Приходная		Приходная		
	Имя	Цена в USD	Отклонение от цены компании в %	Цена в USD	Отклонение от цены компании в %	Цена в USD	Отклонение от цены компании в %	Цена в USD	Отклонение от цены компании в %	Цена в USD	Отклонение от цены компании в %	Цена в USD	Отклонение от цены компании в %	Цена в USD	Отклонение от цены компании в %
Аспаргиназа медак. 500мг/л. лиофилизат для приготовления раствора для в/в и в/м введения 500мг/л															
Бенлиста, лиофилизат для приготовления раствора для инфузий 120 мг															
Бенлиста, лиофилизат для приготовления раствора для инфузий 400 мг															
Винорелбин медак. концентрат для приготовления раствора для инфузий 10мг/мл 1 мл															
Гинипрал® таблетки 0,5 мг		23,50													
Гидроксимармид медак. капсулы, 500 мг		23,50													
		70,00													
Памидонат медак. концентрат для приготовления раствора для инфузий 90мг/30мл		38 064,00													
		38 064,00													
Темодол® капсулы, 140 мг		34 178,00													
		34 178,00													
Элигад® лиофилизат для приготовления раствора для подкожного введения 22,5 мг в комп. с б-тоном															
Элигад® лиофилизат для приготовления раствора для															

Рисунок А2

Продолжение приложения А

Ведомость товаров на складах смотрите рисунок А3.

Ведомость по товарам на складах (... - 30.05.2014)

Действия ▾ Сформировать Отбор Заголовок Настройка... ?

Период с: ... по: 30.05.2014

Ведомость по товарам на складах

Период: ... - 30.05.2014

Показатели: Количество (в базовых единицах)
 Группировки строк: Склад компании, Номенклатура
 Дополнительные поля: Базовая единица измерения

Номенклатура	Начальный остаток	Приход	Расход	Конечный остаток
Основной склад				
Актилизе®, пор. лиоф. для приг. р-ра для в/в инфузий 50 мг в компл. с р-телем, флакон		478,000		478,000
Аркоксиа®, таблетки, покрытые пленочной оболочкой, 90мг, таб		5 880,000		5 880,000
Аспарагиназа медак 5000ЕД, лиофилизат для приготовления раствора для в/в и в/м введения 5000ЕД, флакон		60,000		60,000
Асталин, аэрозоль для ингаляций, дозированный 100 мкг/доза, 200 доз, контейнер		2 196,000		2 196,000
Бенлиста, лиофилизат для приготовления раствора для инфузий 120 мг, флакон		8,000		8,000
Бенлиста, лиофилизат для приготовления раствора для инфузий 400 мг, флакон		3,000		3,000
Бипрол, таблетки, покрытые пленочной оболочкой, 5 мг, таб		39 870,000		39 870,000
Блоктран, таблетки, покрытые оболочкой 50 мг, таб		20 820,000		20 820,000
Валавекс, таблетки, покрытые оболочкой 500 мг, таб		1 830,000		1 830,000
Веро-Винкрестин, раствор для внутривенного введения 0,5 мг/мл 2мл, флакон		2 735,000		2 735,000
Винбластин-ЛЭНС, порошок для приготовления раствора для инъекций 5 мг, флакон		786,000		786,000
Винорелбин медак, концентрат для приготовления раствора для инфузий 10мг/мл 1 мл, флакон		270,000		270,000
Виреад, таблетки, покрытые пленочной оболочкой 300 мг, таб		12 390,000		12 390,000
Гидроксикарбамид медак, капсулы, 500 мг, капсул		44 200,000		44 200,000
Гинипрал®, таблетки 0,5 мг, таб		1 740,000		1 740,000
Губка гемоста. абсорб. желатиновая рассас. стер. Biosponge размер film 200x70x0.5мм, р-р200x70x0.5мм, штука		291,000		291,000
Губка гемоста. абсорб. желатиновая		1 099,000		1 099,000

Рисунок А3

Продолжение приложения А

Номенклатура смотрите рисунок А4.

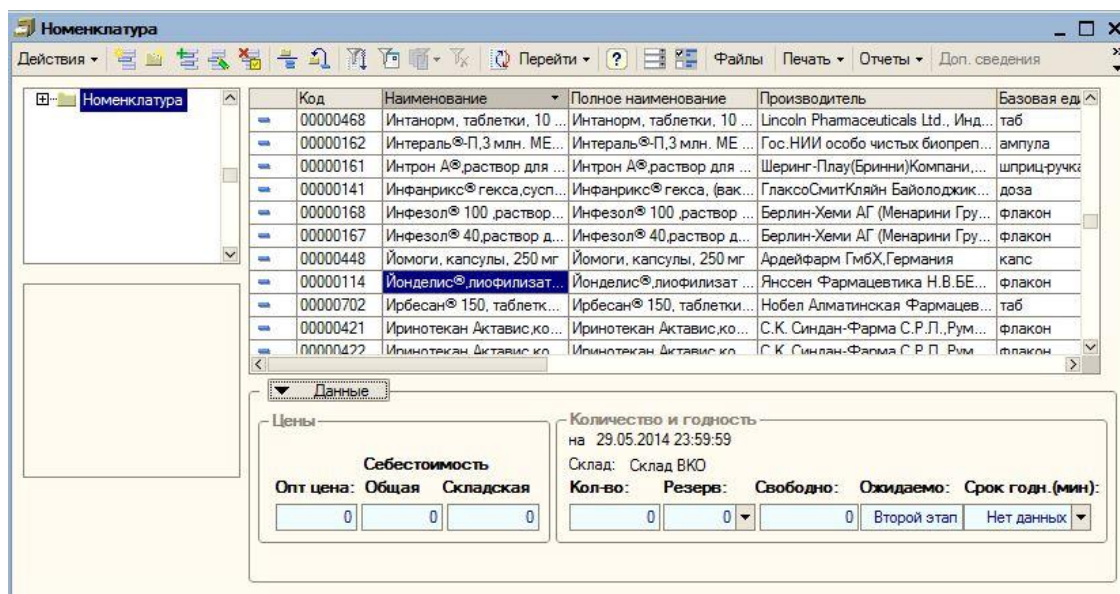


Рисунок А4

Детализация валюты смотрите рисунок А5.

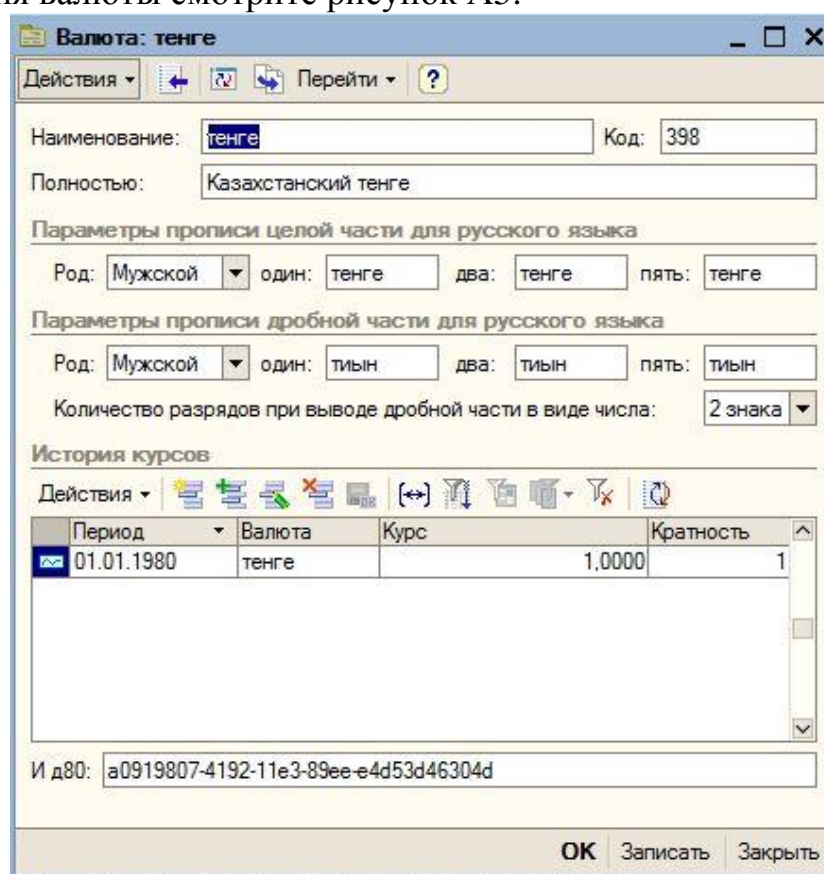


Рисунок А5

Продолжение приложения А

Детализация контрагента смотрите рисунок А6.

Контрагенты: Bharat Serums and Vaccines Limited, Индия

Наименование: Bharat Serums and Vaccines Limited, Индия Код: 00000055 Код СК: 0

Общие Дополнительно Контакты Счета и договоры События Свойства Категории Доступ

Юр. / физ. лицо: Юр. лицо Инопартнер VIP Покупатель Поставщик

Группа контрагентов: Производители

Полное наименование: Bharat Serums and Vaccines Limited, Индия

Код по ОКПО: КБе РНН: Проверить БИН: Проверить

Свидетельство о постановке на учет по НДС

Серия: №: Дата выдачи:

Расписание работы:

Входит в холдинг

Доп. описание:

Виды деятельности Как поставщик

Установить основным

Вид деятельности

Комментарий:

Филиал:

OK Записать Закрыть

Рисунок А6

Заказы поставщикам смотрите рисунок 7.

Дата	Номер	Сумма	Валюта	Контрагент	Организация	Подразделение	Ответственный	Комментарий
29.05.2014 9:47:13	00000000001	54 000,00	тенге	ТЕВАФарм.Пред...	АО "Интерфарма-К"		Тернов Александр	

Рисунок А7

Продолжение приложения А

Общий журнал документов смотрите рисунок А8.

УУ	БУ	Дата	Номер	Вид документа	Вид операции	Информация	Контрагент	Сумма докуме
		07.11.2013 22:18:57	00000000007	Доверенность		ЛС-14-059	ТОО «ВИВА ФАРМ»	
		07.11.2013 22:18:58	00000000006	Счет фактура полу...		Поступление товаров...		
		07.11.2013 22:18:59	00000000008	Доверенность		ЛС-14-060	ТОО "Фирма "Санжар"	
		08.11.2013 0:00:00	00000000001	Доверенность		ЛС-14-054	INKAR TOO	
✓	✓	08.11.2013 0:00:00	00000000009	Поступление това...	По ордеру	ЛС-14-003	ТОО "Фирма "Санжар"	1 505 750,0
✓	✓	08.11.2013 0:00:00	00000000010	Поступление това...	По ордеру	ЛС-14-001	ТОО «Казахская Фар...	8 756 048,0
✓	✓	08.11.2013 0:00:00	00000000011	Поступление това...	По ордеру	ЛС-14-001	ТОО «Казахская Фар...	28 577 850,0
✓	✓	08.11.2013 0:00:00	00000000013	Поступление това...	По ордеру	ЛС-14-037	ТОО «Казахская Фар...	4 674 950,0
✓	✓	08.11.2013 0:00:00	00000000014	Поступление това...	По ордеру	ЛС-14-037	ТОО «Казахская Фар...	58 638 186,0
✓	✓	08.11.2013 0:00:00	00000000015	Поступление това...	По ордеру	ЛС-14-037	ТОО «Казахская Фар...	3 166 023,0
✓	✓	08.11.2013 0:00:00	00000000012	Поступление това...	По ордеру	ЛС-14-001	ТОО «Казахская Фар...	151 506,0
✓	✓	08.11.2013 0:00:00	00000000016	Поступление това...	По ордеру	ЛС-14-037	ТОО «Казахская Фар...	53 281 631,0
✓	✓	08.11.2013 0:00:00	00000000017	Поступление това...	По ордеру	ЛС-14-033	ТОО «L-Фарма»	1 765 772,8
✓	✓	08.11.2013 0:00:00	00000000018	Поступление това...	По ордеру	ЛС-14-004	ТОО «L-Фарма»	15 114 594,1
✓	✓	08.11.2013 0:00:00	00000000001	Поступление това...	По ордеру	ЛС-14-054	INKAR TOO	9 093 039,4
		08.11.2013 12:18:10	00000000002	Приходный ордер ...	от поставщика	Основной склад		
		08.11.2013 17:14:52	00000000009	Доверенность		ЛС-14-060	ТОО "Фирма "Санжар"	
		08.11.2013 18:44:35	00000000010	Приходный ордер ...	от поставщика	Основной склад		
		08.11.2013 18:45:49	00000000007	Счет фактура полу...		Поступление товаров...		
		08.11.2013 18:54:47	00000000011	Приходный ордер ...	от поставщика	Основной склад		
		08.11.2013 18:57:50	00000000008	Счет фактура полу...		Поступление товаров...		

Рисунок А8

Заказы поставщикам смотрите рисунок А9

Цены и валюта... Действия ... Перейти ... Анализ Отправить по почте

Номер: 00000000001 от: 29.05.2014 0:00:00

Организация: АО "Интерфарма-К"

Контрагент: "ТЕВАФарм.Пред.Лтд.,Израиль.пр+ ... Договор: Основной договор

Поступление: ... Оплата: 29.05.2014 По договору с поставщиком нет долга

Склад: Основной склад Б/счет, касса: основной

Товары (0 поз.) Услуги (0 поз.) Тара (0 поз.) Дополнительно

№	Номенклатура	Количество	Ед.	К.	Цена	Сумма	%НДС	Сумма НДС

Тип цен: Не заполнено! Всего (тенге): 0,00
НДС (в т. ч.): 0,00

Комментарий:

Заказ поставщику Печать ... ОК Записать Закреть

Рисунок А9

Продолжение приложения А

Подбор из номенклатуры смотрите рисунок А10.

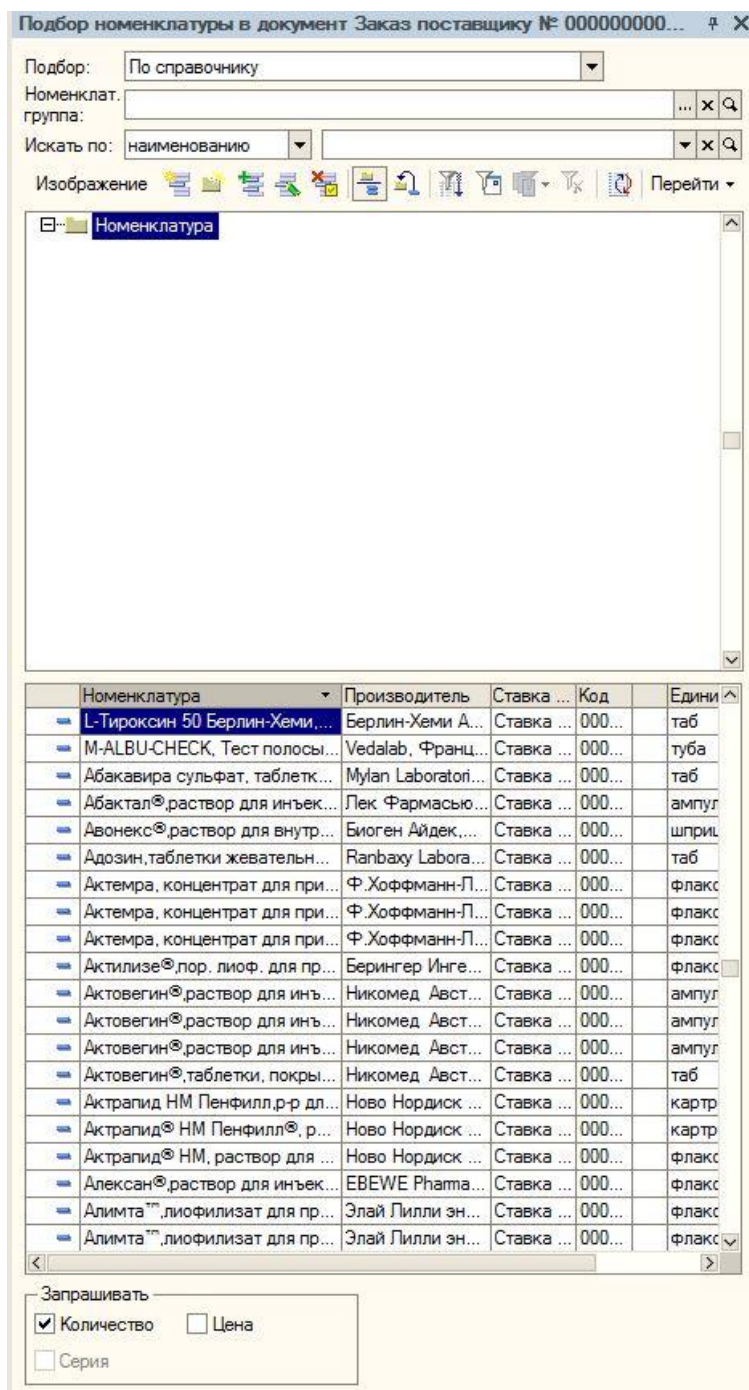


Рисунок А10