Аннотация

Количество страниц – 69, без учета приложений

Количество таблиц – 6

Количество рисунков – 34

Количество используемых источников – 22

Объект исследования – ООО «Стройрем».

Предмет исследования – учет торговой и складской деятельности предприятия.

Цель работы – повышение эффективности деятельности предприятия с помощью доработанной информационной системы автоматизации торгового и складского учета

Методы и средства проектирования – Платформа 1С:Предприятие 8.3.

Результатом выполнения выпускной квалификационной работы Разработанное информационной системы, полностью подстроенное под нужды компании.

Содержание

Введение……4

1 Анализ существующей технологии оперативного учета предприятия и постановка задачи.....6

1.1 Современные тенденции автоматизации торгового и складского учета.....6

1.2 Выявление недостатков предыдущей технологии учета.....9

1.3 Описание систем разработки приложений и выбор системы, при помощи которой будет решаться задача.....11

2 Проектирование и реализация системы торгового и складского учета.....18

2.1 Краткая характеристика строительного магазина ООО"Стройрем" и автоматизируемого процесса.....18

2.2 Расширенная постановка задачи.....26

2.3 Функциональная структура разрабатываемой системы.....29

2.4 Разработка концептуальной (инфологической) модели данных.....39

2.5 Разработка структуры данных.....40

2.6 Клиент-серверный вариант работы платформы «1С:Предпритие» и особенности разработки приложения для этого варианта.....46

2.7 Создание прикладных объектов конфигурации.....47

2.8 Безопасность информационной базы и методы защиты.....55

2.9 Контрольный пример реализации.....57

3 Оценка и обоснование экономической эффективности проекта.....60

3.1 Выбор и обоснование методики расчета экономической эффективности 60

3.2 Расчет показателей экономической эффективности.....64

Заключение.....67

Список использованных источников.....68

Приложение 1. Текст программных модулей.....70

Введение

В качестве темы настоящего исследования выступает проведение работ, связанных с проектированием, а также непосредственной разработкой автоматизированной информационной систем (далее по тексту – АИС) по ведению торгового учета в ООО «Стройрем». Известно о том, что в настоящее время данное предприятие специализируется на продаже товаров для строительства, отделочных материалов, инструментов.

Актуальность работы заключается в том, что организовать эффективную работу торговой компании невозможно без современной системы автоматизации учета. Система автоматизации должна стать стержнем работы компании, помогая сотрудникам выполнять свою работу на всех этапах: от закупки продуктов до анализа работы компании за период.

Разрабатываемая автоматизированная система должна представлять из себя комплексную систему автоматизации и пронизывать все сферы работы торговой компании. Это позволит сделать прозрачной работу всех подразделений и автоматизировать учет всех операций торгового и складского учета в компании.

Система должна включать в себя подсистему ведения номенклатуры продукции, подсистему работы с клиентами, подсистему складского учета продаваемых продуктов, подсистему работы с поставщиками продуктов, а также подсистему отчетности, включающую в себя анализ продаж и предоставления услуг, отчет по остаткам на складе, отчет по взаиморасчетам с клиентами и поставщиками, а также наличию и движению денежных средств. Сложив все подсистемы в единое целое, мы получим в итоге комплексную систему автоматизации работы торгового предприятия.

Цель исследования – обосновать необходимость разработки и разработать систему автоматизации оперативного учета работы торгового предприятия.

Объект исследования – торговое предприятие ООО «Стройрем».

Предмет исследования – учет торговой и складской деятельности предприятия.

Задачи исследования, которые необходимо решить для достижения цели:

- анализ предметной области, изучение процессов и подпроцессов работы предприятия, которые предстоит автоматизировать;
- построение функциональной структуры разрабатываемой системы, выделение подсистем;
- проектирование системы учета работы торгового предприятия;
- разработка программного проекта решения задачи;
- тестирование полученного проекта, устранение недостатков.

Автоматизированную информационную систему, решающую задачу проекта, предполагается реализовать как прикладное решение для платформы «1С:Предприятие 8.3». Выбор этой платформы обусловлен, тем, что она является, во-первых, наиболее типичной системой решения задач автоматизации учета, а во-вторых, наиболее популярной на рынке систем разработки.

Практическая значимость работы состоит в том, что результатом работы будет являться готовая система, которую можно внедрить в торговой компании для автоматизации всех процессов ее работы.

1 Анализ существующей технологии оперативного учета предприятия и постановка задачи

1.1 Современные тенденции автоматизации торгового и складского учета

Складским учетом называется процесс учета и контроля за движением товаров на складах предприятия. Торговый учет – это более широкое понятие. Кратко торговый учет можно сформулировать так: это такой же складской учет, но касающийся не только остатков товаров на складах, но и операции торговли.

Когда говорят о торговом учете, зачастую понимают составление различных аналитических отчетов и контроль за следующими показателями работы торговой компании. Складской учет является всего лишь частью торгового учета, но выделяется в отдельный вид учета, так как является наиболее важной частью торгового учета.

Основная тенденция последних лет в области торгового и складского учета – это отход от ручного учета на бумаге или в разрозненных файлах и переход к автоматизации учета. Если кратко, то автоматизация нужна для того, чтобы упорядочить работу торговой компании и свести к минимуму участие человека. Все делает специальная компьютерная программа.

Основной принцип любой системы автоматизации торгового и складского учета – возможность в режиме реального времени отследить и проконтролировать текущий остаток и историю движения, любого учитываемого ресурса, будь то товар на складе, денежные средства в кассе или такой нематериальный ресурс как дебиторская задолженность. Анализ можно сделать не только на текущий момент, но и на любой момент времени. Анализ показывает не просто остатки и движения ресурсов, но и расшифровку этих движений: кто, когда и каким документом сделал это перемещение товаров или вывод денежных средств.

Второй важнейший принцип – автоматическое заполнение в документах всех данных, которые только возможны. Если в карточке контрагента записан его адрес, то в документ отгрузки он подставляется автоматически. Это касается абсолютно всех документов по работе с клиентами, поставщиками, государственными органами и т.д.

Любой системе автоматизации присуща развитая система отчетности. Анализ данных в различных разрезах, с наложением любых условий – обязательный атрибут любой системы автоматизации учета.

Облегчение ведения учета - еще один принцип автоматизации. В том числе это проявляется в большом количестве дополнительных опций. Даже базовые версии программ умеют работать со сканерами штрих-кодов, терминалами сбора данных, нередко имеют встроенные подсистемы CRM и многое друге.

Однако в последнее время стремительное развитее торговой сферы породило новую тенденцию к автоматизации торгового и складского учета – комплексная автоматизация учета. Комплексная автоматизация – это не просто наращивание функциональных возможностей существующих систем торгового и складского учета и введение в них новых «фишек» - это уже пройденный этап. Комплексная автоматизация подразумевает принципиально новый подход, основной идей которого является интеграция всех подсистем учета в единое целое и создание единого информационного пространства ведения учета всей организации.

С уверенностью можно сказать о том, что сегодня на рынке есть очень много готовых решений, посредством которых можно было бы решить поставленную задачу. И тут в качестве самого удачного варианта стоит выделить разработку корпорации 1С – «Управление торговлей 8» [4]. Считается, что это один из самых удачных инструментов, который комплексно работает на увеличение степени эффективности бизнеса той или иной компании.

«1С: Управление торговлей 8» дает возможность автоматизировать функции управленческого, а также оперативного учета, анализировать, планировать операции торговли; максимально продуктивно руководить торговой компанией.

На рисунке далее будет представлена схема, в которой выражается предметная область, автоматизируемая с применением продукта «1С: Управление торговлей 8» [1].

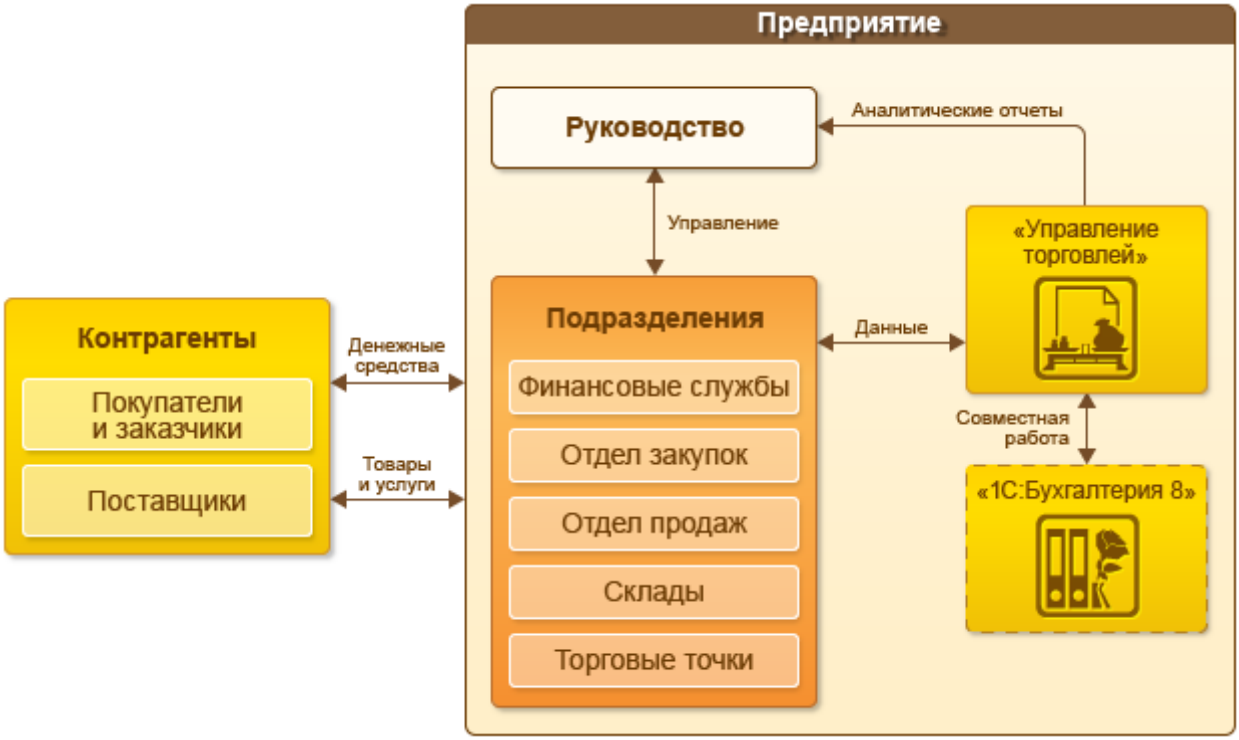

Рисунок Error: Reference source not found.1. Структурная схема анализируемого продукта

Итак, в целом анализируемый программный продукт дает возможность:

- полностью перевести в автоматизированный вид процедуру ведения документов в электронном формате по самым распространенным операциям, связанным с товарно-материальными ценностями;

- упростить в значительной мере процедуру введения справочной, нормативной информации;

- сформировать любой отчет по работе предприятия;

- гибко настроить уровень доступа к программному продукту.

Еще одна возможность для решения задачи настоящего проекта – использование программного продукта (ПП) «Торговля и склад». Эта разработка одного из первых интернет-сервиса в России – «Мой Склад». За счет наличия мощных возможностей по управлению складом, за счет аналитики, наличия модуля по работе с клиентами ПП обеспечивает максимальное удобство применения в компании, занимающейся оптовой торговлей. При этом удобная интеграция с интернет-магазином, ряд дополнительных возможностей – все это делает программу незаменимым инструментом для всех компаний.

Перечислим главные преимущества этого ПП:

- может применяться любая ОС для работы ПП;

- устанавливать, адаптировать программу к ОС – не требуется; интегрирована функция быстрого старта;

- распространяется программный продукт на бесплатной основе;

- открыта возможность работы в пробном периоде.

Однако, по итогам анализа популярных, распространенных решений для автоматизации работы компании, принимается решение о том, что нужно создать собственную АИС. Реализация нового продукта приведет к тому, что будут по максимуму задействованы программные, а также технические средства, которые уже есть в ООО «Стройрем». Позже, новые решения будут просто адаптированы к существующим решениям.

1.2 Выявление недостатков предыдущей технологии учета

Разумеется, сегодня во всех торговых компаниях может осуществляться большое количество операций. Для того, чтобы предприятие работало максимально успешно, должна быть в наличии информационная, слаженная система по ведению учета документации, товаров. Прежде всего, такая

система позволяет на практике автоматизировать все векторы работы субъекта хозяйствования. Сейчас система автоматизации в компании ООО «Стройрем» базируется на примитивной офисной программе MS Office. К слову сказать, началась работа с этой программой примерно 10 лет тому назад. Считалось в то время, что это значительный шаг вперед в сравнении с тем, что все документы велись на бумажных носителях. Суть автоматизации сейчас состоит в том, что файлы оформляются в «MS Excel» и «MS Word» на сервере торговой компании. В конечном итоге система функционирует для обслуживание торгового и складского учета, проводит процедуры взаимных расчетов с поставщиками, а также клиентами.

Совершенно очевиден тот факт, что автоматизация на базе системы, которая представлена в торговом предприятии, вряд ли может успешно справляться с документооборотом современного предприятия. По большому счету, как система автоматизации – продукт отсутствует. Имеется только малое количество примитивных шаблонов, с помощью которых документы оформляются немного быстрее, чем это выполнялось бы без имеющихся шаблонов.

Отмечая процедуру хранения справочных, нормативных документов в компании, можно подметить, что здесь дела обстоят немного лучше. Представлен прайс-лист, товарный каталог, список поставщиков, а также клиентов. Самое главное, что обозначенные документы имеют структуризацию, а также отсортированы. Вот только информация в них не имеет особой нагрузки, так как в оформляемые документы, в их шаблоны – она не вносится.

Куда хуже обстоят дела с ведением отчетности. В том варианте, который сейчас практикуется в компании – формирование отчетов на основании информации, которая разбросана по файлам «MS Excel», говорить о качественной системе формирования отчетности не приходится. Например, чтобы организовать аналитическую работу по продажам, предстоит найти данные в ручном режиме, внести их в таблицы, а только потом приступать к

формированию непосредственной отчетности. И в таком случае пакет «MS Excel» оказывается полностью бесполезным.

Итог закономерный: вести учет так, как это происходит в настоящее время в дальнейшем в компании нереально.

Когда мы говорим о реализации предстоящего проекта, стоит акцентировать внимание на том, что будут задействованы готовые технические, а также программные возможности анализируемой компании. Сейчас, пожалуй, не актуальным будет вопрос о том, какая АИС для учета деятельности предприятия будет куплена. Задача такова: выбрать соответствующую технологию, посредством которой будет создан собственный программный продукт. Разумеется, потребуется для той работы соответствующий язык программирования.

Становится очевидным, что будет разработан собственный ПП, на основании которого в перспективе компании сможет автоматизировать процедуру документооборота. Итак, создаваемый ПП – это конфигурация известной в России платформы – «1С: Предприятие 8.3».

Следует отметить, что разрабатываемую систему учета следует строить как комплексную , автоматизирующую все стороны и направления работы торговой компании: торговый учет по работе с клиентами, складской учет товаров, работу с поставщиками по заказу товаров, ведение взаиморасчетов с клиентами и поставщиками, построение аналитической отчетности.

1.3 Описание систем разработки приложений и выбор системы, при помощи которой будет решаться задача

Как показывает практика, в настоящее время в качестве одного из самых известных языков программирования выступает C++ . В данном случае реализация объектов осуществляется как бы классами. В целом, известно о том, что класс в конечном итоге определяет, как будут себя вести объекты конкретного типа, каким образом они появляются в системе, а также, как с нее

удаляются. Процедура написания программного продукта посредством применения классов предусматривает, что текст программы будет структурированным, а также понятным . Интегрированный аппарат объектно-направленного программирования позволяет описать тот или иной объект в рамках предметной сферы; создать программный пакет для пользовательских потребностей – как прикладного, так и системного использования.

По праву считается, что своеобразный последователь C++ - это ЯП C #. «C#» в настоящее время выступает в качестве инструмента, который специально создавался для разработки таких приложений, которые функционируют в среде «.NET Framework». Язык «C#» по своей структуре достаточно простой. За счет большого количества новаций в нем, можно быстро и без проблем разрабатывать нужный программный продукт.

В качестве примера можно привести самую известную разработку, которая написана на указанном ЯП. Речь идет в данном случае о «Microsoft Visual Studio». «Visual C#» — формат реализации языка «C#» корпорацией «Майкрософт».

Среду разработки «Microsoft Visual Studio» на практике достаточно часто применяют, кроме прочего, в качестве платформы для создания приложений на языке C++. Тем не менее, в большинстве случаев разработчики отдают предпочтение специализированным средам разработки. В качестве примера можно назвать «RAD C++ Builder».

Перспективное развитие идеи так называемого кросс-платформенного программирования в конечном итоге приводит к тому, что появляется язык Java. Приложения Java оформляются в особый байт-код, который потом выполняется на той или иной архитектуре посредством специализированной виртуальной Java-машины (JVM). Java является языком с Си-подобным синтаксисом и близок в этом отношении к C/C++ и C#.

Принимая во внимание тот факт, что JVM могут быть доступными практически на всех программных, а также аппаратных платформах, Java по своей природе может выступать в качестве ПО связующего типа, а также как

независимая платформа. Применение одинакового байт-кода для большинства известных платформ дает возможность описать Java как «скомпилировано однажды, запускается везде» - мечта кросс-платформенного программирования. Самая популярная платформа под написание программ на ЯП - Java Eclipse.

Еще одной популярной средой разработки приложений является платформа «1С:Предприятие 8.3». Считается, что сегодня популярной методологией проработки ИС является применение метаданных. Платформа «1С:Предприятие 8.3» является типичной представительницей такого подхода. Метаданные - это информация о прочих сведениях, которые непосредственным образом касаются дополнительной информации об объекте. Выделяется большое количество платформ разработки, которые построены на метаданных. Одной из самых популярных разработок можно назвать “1С: Предприятие”.

Здесь метаданные выступают в качестве комплекса проблемно направленных объектов, которые могут быть поддерживаемыми на уровне технологической платформы. Объектный состав – это итог длительного анализа предметных сфер, где задействована платформа. Главная задача, которая ставится перед разработчиком, состоит в том, чтобы собрать с имеющихся объектов требуемую структуру ПП, а потом – привести описание алгоритмов работы, взаимодействия таких объектов.С применением метаданных можно описать объекты из предметной сферы; модули на специальном языке могут реализовать любой функционал объектов. Комплекс представлений объектов, программных модулей формирует прикладные решения или конфигурации.

Далее нужно представить главные достоинства, которые можно получить от использования продукта “1С: Предприятие” 8.3. Прежде всего, выделим предметную направленность. Она гласит о том, что процедура программирования осуществляется в терминах, задействованных в предметной сфере. Разработчику нужно сконцентрироваться на реализации

функционала системы, а не на внутренних проблемах языка, а уж тем более – не на платформе.

Также “1С: Предприятие” позволяет быстро разработать требуемое приложение. Эта возможность открывается благодаря тому, что используются прикладные объекты, как компоненты конструктора. В целом, приложение может не иметь даже программного кода, но со своими задачами – справляться будет. Если нужно написать программный код, то для этого программист получает массу шаблонов, конструкторов.

Множество сервисных функций, которые дает платформа. Известно о том, что даже простейшее приложение с наличием таких функций становится мощным, солидным решением.

Соответственно, благодаря всем преимуществам, о которых было сказано ранее, «1С: Предприятие» становится одним из самых популярных инструментов для создания ПП не только в РФ, но и СНГ.

Так, можно использовать инструмент для учетных задач, которые в целом направлены на автоматизацию процессов в бизнесе. К слову сказать, задача, которая предусматривает автоматизацию складского и торгового учета предприятия – это учетная задача, в которой применение «1С: Предприятие» – оправданное, обоснованное решение.

В таблице () приведено сравнение инструментов разработки.

Таблица Error: Reference source not found.1. Сравнительная таблица инструментов разработки

IDE и язык	Лицензия	Windows	Другие платформы	Отладчик	Разработка GUI	Автодополнение	Анализ кода	Браузер классов
«RAD C++ Builder»	проприетарная	+	-	+	+	+	+	+
«Microsoft Visual Studio C#»	проприетарная	+	+	+	+	+	+	+
«Eclipse Java»	EPL	+	+	+	+	+	+	+
“1С:Предприятие 8.3»	проприетарная	+	+	+	+	+	+	+

Как показывает практика, написание ПП на базе первых трех языков из этой таблицы, а также на платформах для программирования на них – это процедура «классической» работы при разработке приложений. Тем не менее, сегодня активнее применяются новые способы и методологии программирования. И самой перспективной среди них является технология метаданных.

Еще одно неоспоримое преимущество, которым обладают программные решения 1С – это качественная поддержка пользователей, партнеров. На данный момент фирма «1С» имеет 7500 партнерских компаний по всей России, а также в странах СНГ.

Тем не менее, сегодня «1С: Предприятие 8.3» - это не просто продукт, посредством которого программисты могут создавать типовые решения под себя. Прежде всего – это платформа, на основании которой можно также получать уникальные прикладные приложения. Поэтому сейчас «1С: Предприятие 8.3» стоит по качеству, доступности в одном списке с такими разработками, как «Odoo», «Epicor IScala», «SAP» и др.

«1С: Предприятие 8.3» имеет уникальную возможность для разработчиков приложений. Она состоит в том, что сама платформа и получаемое приложение будут строго дифференцированы. Соответственно, новинки всегда будут открытыми, а сама платформа становится мощной средой разработки, посредством которой можно не только дорабатывать типовые решения, но и создавать новые продукты – как бы с нуля.

Далее нужно подчеркнуть, что «1С: Предприятие 8.3» имеет колоссальное количество инструментов, а также возможностей технического характера. Они представлены далее на рисунке (Рисунок Error: Reference source not found.). Тот или иной пункт этого рисунка - продуманный, комплексный механизм, который заслуживает особого внимания для анализа. Однако нам достаточно отметить, что в комплексе все инструменты образуют единую платформу, на которой создаются приложения класса «framework».

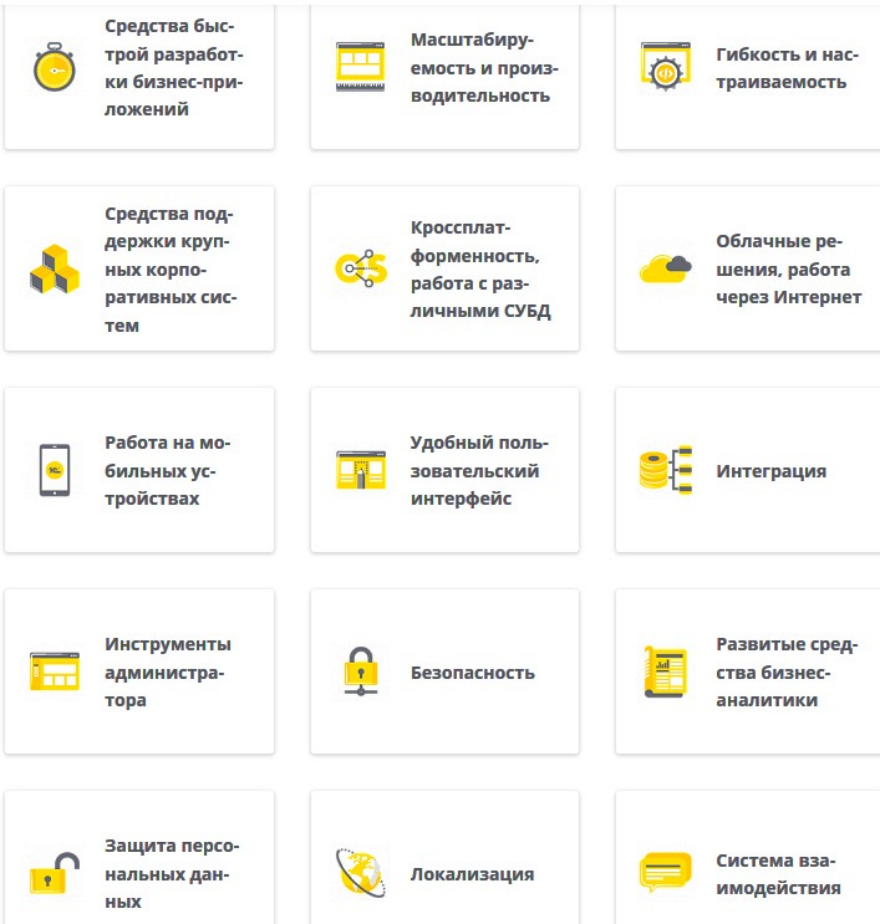

Рисунок Error: Reference source not found.2 Возможности платформы «1С:Предприятие 8.3»

Добавим, что конечно же «1С: Предприятие 8.3» – это не есть универсальная среда, на основании которой можно прорабатывать любые приложения. Тут функционал строго ограничен – создание приложений, которые позволяют автоматизировать задачи в бизнесе. Несмотря на имеющиеся недостатки, можно точно избрать эту технологию и методику программирования, а значит, исследуемая платформа становится лучшим решением для нашей задачи.

Таким образом в качестве платформы разработки системы выбирается «1С: Предприятие 8.3».

2 Проектирование и реализация системы торгового и складского учета

2.1 Краткая характеристика строительного магазина ООО"Стройрем" и автоматизируемого процесса

На данный момент известно о том, что предприятие, которое анализируется в рамках настоящей работы, уже более 2 лет ведет продажу строительных материалов и инструментов.

С каждым годом магазин старается все больше расширить задачи, которые выносятся на решения. Это могут быть не только комплексные определения потребностей заказчиков в плане требуемой продукции, но и операции по доставке товара. В исследуемой компании клиенты могут воспользоваться разными доставками: крупными и мелкими товарными партиями. Для этого предусматривается колоссальный ассортимент поставляемых товаров. Разумеется, есть возможность оформить скидку на крупную покупку, отсрочки в платежах. Вне зависимости от того, каким вариантом доставки, заказа продукции воспользуется клиент, предприятие гарантирует строгое соответствие стандартам, а также неукоснительное соблюдение норм в рамках продовольственной безопасности, а также обеспечение нужных документов на все поставки.

Центральными векторами в работе компании считаются: закупка продукции у крупных оптовых компаний и непосредственно у производителей; поставка клиентам разных товарных партий; организация складского хранения партий товаров, которые куплены по предоплате клиентами; проведение в разных формах взаимных расчетов компании и клиентов.

С уверенностью можно сказать о том, что процедура автоматизации торгового процесса в рамках исследуемого предприятия – важная и целесообразная процедура. Дело в том, что у предприятия на данный момент имеется мощная программная, техническая инфраструктура. Заново создавать ничего не придется; достаточно только адаптировать некоторые решения под создаваемую систему. Представим архитектуру системы.

В качестве базы ИТ-архитектуры предприятия выступает сервер «Dell R440». Это техническое, современное решение, которое позволяет обеспечивать доступ пользователей к принтерам, общим данным, файлам и иным сервисам. В частности, «Dell R440» имеет такие параметры:

- процессор: 2 × Intel Xeon Silver 4208 (8C 11M Cache 2.10 GHz);
- оперативная память: 4x DDR4 RDIMM 2133MHz REF 32 Гб, всего 128 Гб;
- контроллер RAID: Dell H330 (ZM);
- модуль удаленного управления iDRAC 9 Enterprise;
- интегрированная сетевая карта Dell 2port 1Gb;
- жесткий диск: Dell 1.2TB SAS 10k 2.5” G14;
- блок питания: 2x Dell 550w Hot Plug.

В ООО «Стройрем» указанный сервер исполняет функции контролера домена, сервера почты, контроля над доступом в Интернет. В некотором плане, это сервер приложений 1С и сервер БД. Работа его выстраивается на основе Windows Server Standard 2014 OEM. В качестве ИБП выступает «APC Smart-UPS RT 1000VA 230V».

У каждого сотрудника на рабочем месте стоит ПК с архитектурой «x86» или «x86-64».

Далее на рисунке (Рисунок 2.2.1) будет представлена техническая архитектура предприятия.



Рисунок 2.1. Техническая архитектура ИС предприятия

Кроме того, стоит добавить, что в компании установлены и успешно работают программные системы. Как и прочие компании, данное предприятие ведет бухгалтерский учет, формирует и передает налоговую, финансовую отчетность. Также осуществляется процедура учета работников компании, в том числе – учет зарплаты. Специально для этого используется ПП под названием «1С: Бухгалтерия 8, редакция 3.0», на основе «1С: Предприятие 8.3». На самом деле, у представленной системы есть множество возможностей, которые позволяют вести бухгалтерский учет, а также учет сотрудников.

«1С: Бухгалтерия 8, редакция 3.0» применяется на основе клиент-серверного варианта. Ранее было сказано о том, что в качестве физического сервера применяется «Dell R440», на нем работает сервер приложений 1С. Он осуществляет отсылку на СУБД «Microsoft SQL Server», который используется для хранения данных. «Microsoft SQL Server» развернут на том же физическом сервере «Dell R440». У сотрудников на рабочих местах стоит клиентская часть программы «1С: Бухгалтерия 8». На ПК установлена операционная система «Windows 7» или «Windows 10». Именно она дает возможность подключаться к сети, применять ее всевозможные ресурсы. На

рисунке (Рисунок 2.) далее будет представлена программная архитектура компании.

Известно о том, что файл-сервер компании выступает в качестве хранилища информации по учету. Посредством папки под названием «Торговый учет» ведется учет операций. В целом, это файлы в форматах «word» и «excel». Каждый участник процесса на основе шаблонов вносит в шапку «Торговый учет» на файловом сервере собственные данные, связанные с торговыми и складскими операциями. Тот же сервер «Dell R440» применяется в качестве файл-сервера.

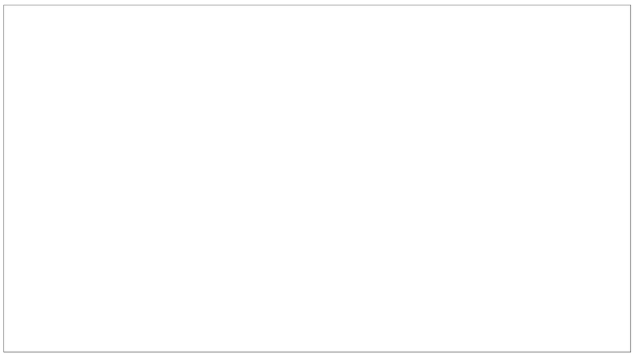

Рисунок 2.2. Программная архитектура ИС предприятия

Разумеется, сегодня во всех торговых компаниях может осуществляться большое количество операций. Для того, чтобы предприятие работало максимально успешно, должна быть в наличии информационная, слаженная система по ведению учета документации, товаров. Прежде всего, такая система позволяет на практике автоматизировать все векторы работы субъекта хозяйствования. Сейчас система автоматизации в компании ООО «Стройрем» базируется на примитивной офисной программе MS Office. К слову сказать, началась работа с этой программой примерно 10 лет тому назад. Считалось в то время, что это значительный шаг вперед в сравнении с тем, что все документы велись на бумажных носителях. Суть автоматизации сейчас состоит в том, что файлы оформляются в «MS Excel» и «MS Word» на сервере торговой компании. В конечном итоге система функционирует для обслуживание торгового и складского учета, проводит процедуры взаимных расчетов с поставщиками, а также клиентами.

Чтобы понять практически, как устроен процесс решения задачи в рамках проектируемой системы, мной были проработаны специальные «IDEF-диаграммы», на которых отражены процессы в работе компании. Итак, если обратить внимание на первую диаграмму, то она выступает в качестве отражения модели АИС в целом, без подробной детализации – рисунок (Рисунок 2.).

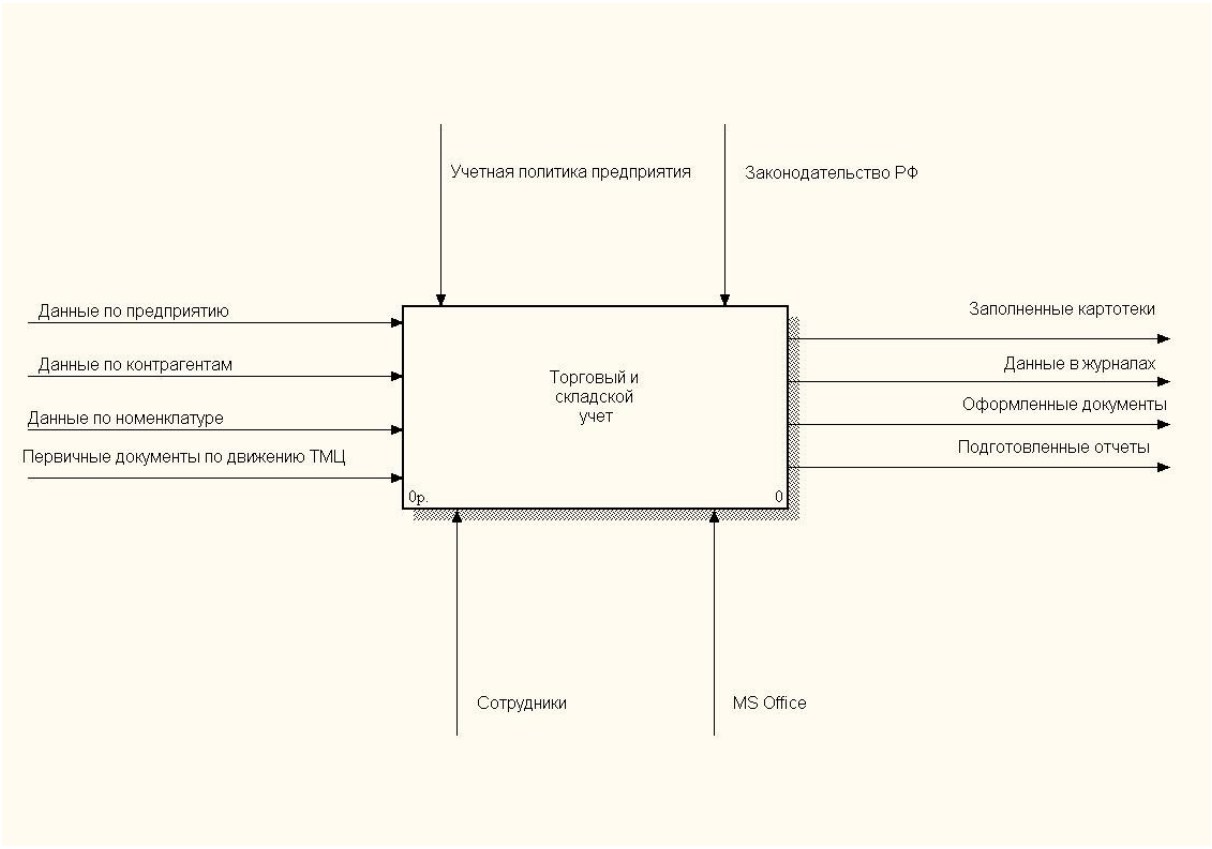

Рисунок 2.3. Диаграмма, посредством которой отражается нулевой уровень работы компании.

Первая декомпозиция системы показана на рисунке (Рисунок 2.).

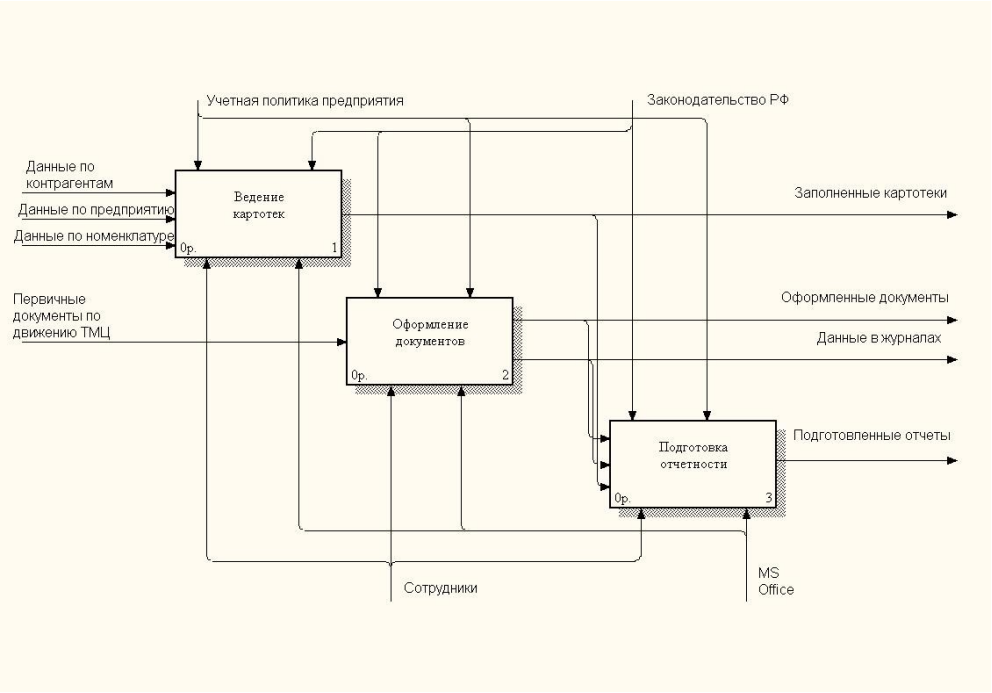

Рисунок 2.4. Декомпозиция процесса деятельности предприятия

Потом нами проводится декомпозиция той или иной задачи на более мелкие структурные единицы – подзадачи.Что касается здесь декомпозиции блока под названием «Ведение картотек», то она выражается посредством рисунка (Рисунок).

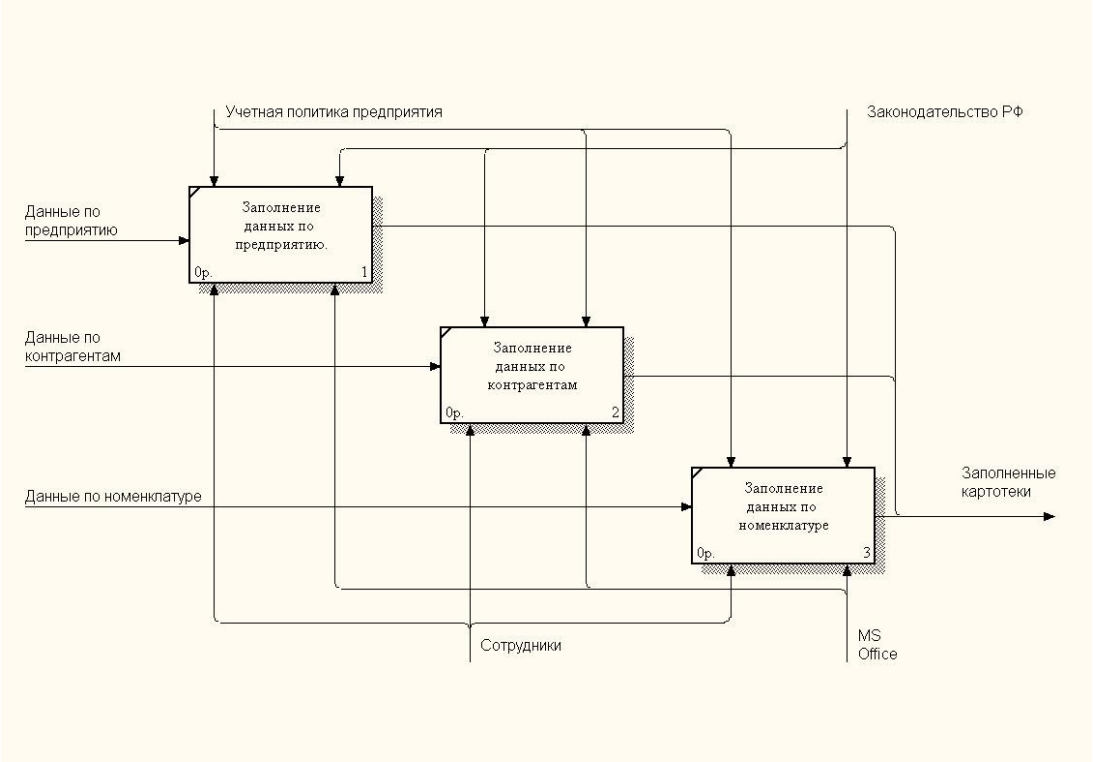

Рисунок 2.5. Декомпозиция задачи «Ведение картотек»

Ниже на рисунке (Рисунок 2.) показана декомпозиция блока, который называется «Оформление документации».

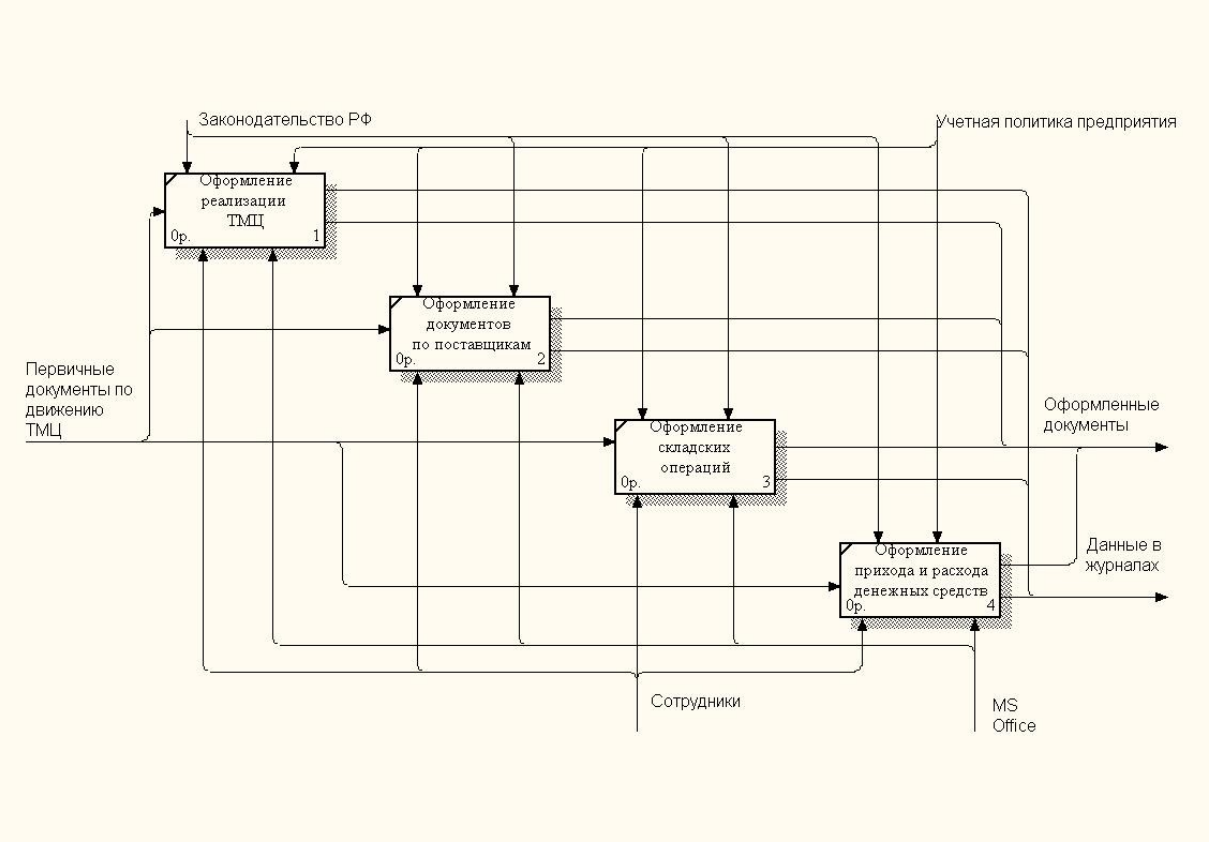

Рисунок 2.6. Декомпозиция процесса «Оформление документов»

При детальном рассмотрении блока «Аналитическая отчетность», можно сказать о том, что он специально был разработан для того, чтобы вести аналитическую работу компании в целом. Этот раздел можно дифференцировать на следующие блоки: «Отчеты по движению финансов», «Отчеты по остаткам средств на счетах», « Отчеты по проводимым взаимным расчетам». На рисунке (Рисунок 2.) ниже будет представлена декомпозиция блока, который носит название «Подготовка отчетности».

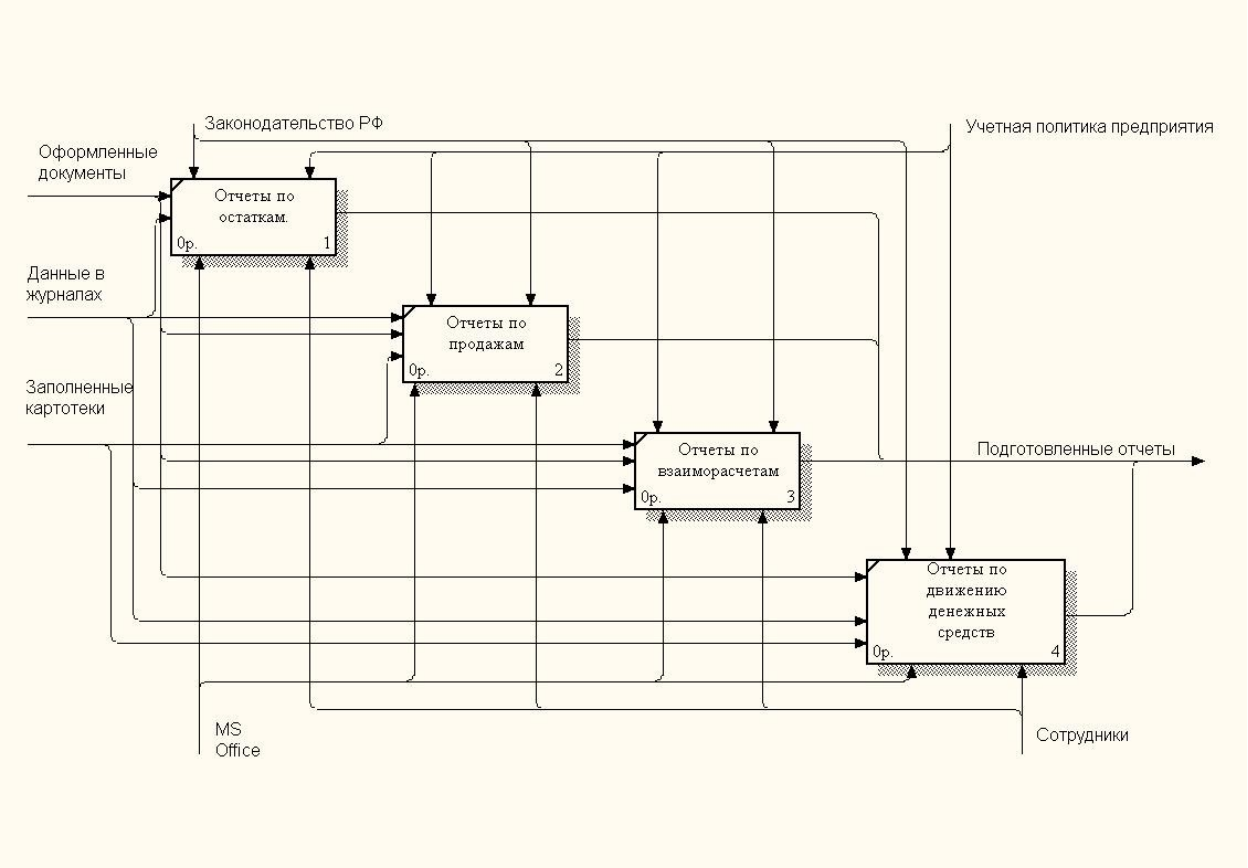

Рисунок 2.7. Декомпозиция задачи «Подготовка отчетности»

Детальным образом мы проанализировали все диаграммы, которые представлены выше. Эта работа дает возможность понять, что бизнес-процессы компании выстраиваются правильно. Единственный недостаток состоит в том, что неверно осуществлен выбор инструмента, посредством которого проводится автоматизация учета. Речь идет о программном пакете MS Office.

2.2 Расширенная постановка задачи

Назначение разрабатываемой ИС – комплексная автоматизация деятельности компании в части торгового учета и сопутствующих видов учета. Система позволяет решать все задачи учета, которые возникают в торговой деятельности компании, начиная от заказа товаров у поставщиков и заканчивая оплатой товаров, проданных клиентам. Торговая деятельность «тянет» за собой и все остальные направления работы компании: складской

учет товаров, взаиморасчеты с клиентами и поставщиками, аналитическая отчетность по продажам, остаткам и взаиморасчетам.

Структурно ИС можно разбить на такие подсистемы:

- Подсистема, на основании которой можно вести справочники;
- Подсистема, применяемая для целей торгового учета;
- Подсистема на складской учет;
- Подсистема, обеспечивающая возможность проведения взаимных расчетов с клиентами предприятия.

Добавим, что схематически каждый справочник допускается делить на группы, среди которых выделяются контрагенты, номенклатура, собственная компания. Например, что касается справочника группы «Номенклатура», то здесь он специально открывается для того, чтобы осуществлять процедуру введения, а также редактирования информации по товарной номенклатуре. Также справочники используются для ведения некоторых вспомогательных, дополнительных сведений. Характерная особенность, что выделяется при учете номенклатуры – учет в разных единицах измерения. Также открыты возможности для учета товаров по партиям, с учетом скидок и цен. Еще один бонус – презентация товаров с иллюстрациями.

Известно о том, что под конкретный вид продукции указывается несколько единиц измерения. Также система имеет учет по партиям. Любая товарная партия имеет свои свойства. В подсистеме назначаются цены, а также скидки для разных категорий клиентов.

Практика показывает, что к любой единице в номенклатуре можно прикреплять несколько файлов того или иного формата (видео и картинки, текст, таблицы в электронном виде и прочее).

Теперь можно перейти к анализу справочников из группы «Контрагенты». Здесь известно о том, что с их помощью должна быть возможность для ведения дополнительных, а также основных сведений по поставщикам предприятия, по группам клиентов. На основании заведенных

договоров с контрагентами должна быть возможность для гибкого ценообразования, для использования системы скидок.

Если говорить о справочниках из группы «Организация», то с их помощью должна быть возможность ведения дополнительных и основных сведений о предприятии: рабочие подразделения, конкретно взятые сотрудники, складское дело и прочее.

В целом можно констатировать, что подсистема по торговому учету на предприятиях создается для того, чтобы вести учет множественных торговых операций. Их учет, прежде всего, предусматривает аналитическую работу со следующими фактами хозяйственной активности:

- процедура предварительного приема заявок от покупателей;
- учет проданных по факту товаров клиентам;
- возврат бракованной, некачественной продукции от покупателей.

Выделяя формы пользовательского интерфейса, в котором ведется комплексная работа с документами, нужно отметить, что здесь все функции должны быть предельно понятными, удобными для введения данных и их просмотра.

Когда на предприятии торговли осуществляется расчет цен, важно принимать во внимание их тип, потенциальные скидки. К слову сказать, по отдельно взятым контрагентам могут формироваться дисконтные карты. Конечный результат работы подсистемы по торговому учету – это получение отчета, который называется «Продажи».

По складскому учету есть подсистема, которая создана для учета операций, проводимых на складе:

- оформление заказа для поставщика;
- поступление товарной партии от поставщика;
- возврат определенной продукции поставщикам;
- списание продукции, ее перемещение по складу;
- операция инвентаризации.

Считается, что центральная задача, которая поставлена для решения в рамках обозначенной подсистемы – это ведение изменения остатков по складу в автоматическом режиме. Важно, чтобы системой строго контролировались остатки по складу, вне зависимости от того, какой проводится документ. Закономерный исход работы подсистемы – формирование отчета, который называется «Остатки».

Наконец, подсистема по взаимным расчетам, по учету таких расчетов создана для того, чтобы фактически учитывать поступающие средства от клиентов. В системе обязательно должны быть реквизиты платежных документов, чтобы в любой момент времени можно было контролировать факт оплаты. Добавим, что с помощью представленной выше подсистемы должна открываться возможность – введение документов прихода, расхода финансов по счетам учета активов.

2.3 Функциональная структура разрабатываемой системы

На рисунке представлена функциональная структура приложения

Рисунок Error: Reference source not found.8. Функциональная структура приложения

Чтобы перейти к непосредственному показу функциональной структуры приложения, уместно осуществлять такую операцию посредством механизма, который имеется в рамках подсистем платформы «1С: Предприятие 8.3». Говоря о подсистемах, уточним, что они представляют собой механизм визуального, функционального, а также ролевого деления решения на мелкие, а также укрупненные блоки. Посредством таких компонентов в рамках общей платформы разработчик может создать командный интерфейс прикладного решения, а затем – визуально дифференцировать возможный его функционал. Практика показывает, что любая подсистема по своему строению может характеризоваться иерархией. Другими словами, одна подсистема в своем составе может иметь другие подсистемы, и в таком случае формируется так называемый «вложенный интерфейс».

Говоря о подсистеме «1С: Предприятие 8.3», отметим, что в ее составе включено сразу несколько объектов прикладного характера по объектам конфигурации. Добавим, что подсистемы создаваемой в настоящей работе конфигурации под цели автоматизации складского и торгового учета, отражаются на рисунке (Рисунок Error: Reference source not found.9). Если же подсистемы будут организованы максимально грамотно, то в автоматическом режиме удастся создать такой интерфейс, который как раз потребуется для решения нужных задач, для обеспечения пользователя приложения требуемым функционалом.

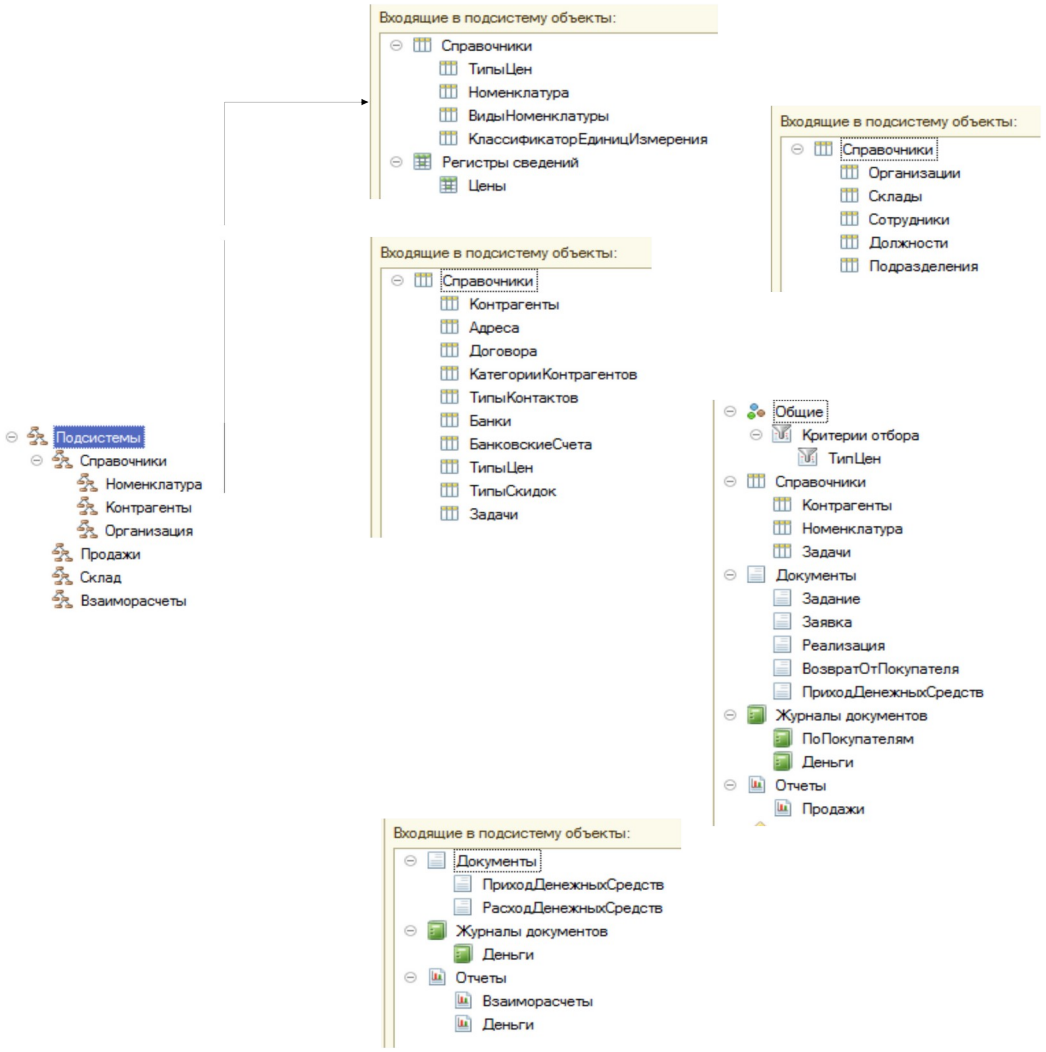

Рисунок Error: Reference source not found.9. Функциональная структура прикладного решения в виде подсистем

Итак, начать анализ нужно с подсистемы, которая называется «Справочники». В ней обычно организуется работа со справочной, а также нормативной документацией, которая только может быть представлена в рамках АИС. Указанная подсистема по структуре в своем составе имеет 3 прочих подсистемы. Все они позволяют организовать работу с другими, сопряженными справочниками системы:

- «Номенклатура» - подсистема, посредством которой практически выстраивается процедура ведения справочников товарной номенклатуры;

- «Контрагенты» - подсистема, за счет которой организуется ведение справочников с информацией о покупателях, а также поставщиках исследуемого предприятия;

- «Организация» - подсистема, в которой организуется процесс учета информации, связанной с компанией: о работниках, складах, функциональных подразделениях и так далее.

Если анализировать подсистему, которая называется «Склад», то можно подчеркнуть, что в ней собраны функциональные возможности, на основании которых ведется складской учет товарных остатков: формируются разные отчеты, создаются новые документальные формы и прочее.

Особого внимания заслуживает подсистема, которая называется «Взаиморасчеты». В ней группируется функционал, посредством которого можно осуществлять практический учет взаимных расчетов компании с контрагентами.

Считается, что базой для торгового учета любой компании считается ведение номенклатурного справочника. В нем собраны центральные данные по товарам. Особенностью прорабатываемой конфигурации выступает тот факт, что в справочнике номенклатуры имеются не только общие сведения, распространяющиеся на продукцию, но и другие данные о предприятии. Детально можно отразить такой функционал схемой, которая будет ниже представлена на рисунке (Рисунок Error: Reference source not found.10).



Рисунок Error: Reference source not found.10. Ведение номенклатуры

Практика показывает, что ведение списка, в котором зарегистрированы все контрагенты предприятия – это главная функция, которую должна исполнять создаваемая АИС по автоматизации складского, а также торгового учета. По праву центральный справочник, с применением которого осуществляется управление информацией по контрагентам, считается одноименный справочник – «Контрагенты». На самом деле, его реквизиты довольно масштабные.

Возвращаясь к анализу создаваемой конфигурации АИС, отметим, что характерная ее особенность будет состоять в гибкости и полноте справочника, в котором отражены контрагенты предприятия. Рисунок (Рисунок Error: Reference source not found.11) позволяет проанализировать схему, в рамках которой будут отражены функции исследуемой подсистемы.



Рисунок Error: Reference source not found.11. Ведение контрагентов

Если выделять третью подсистему в «Справочниках», то отметим, что ее назначение состоит в ведении информации, которая непосредственным образом относится к компании. Обычно подобные сведения применяются специалистами для формирования отчетов, выходных документов, а также в некоторых случаях – для операций складского учета. Схематически функциональный состав подсистемы «Организация» показан на рисунке (Рисунок Error: Reference source not found.12).



Рисунок Error: Reference source not found.12. Ведение данных организации

Особенно стоит отметить подсистему, которая называется «Продажи». Главное предназначение ее в рамках создаваемой АИС – введение, а также изменение документов, которые распространяются на операции по торговому учету. Представленная работа подразумевает принятие от клиентов заявок, создание разнарядки на отгрузку товаров на складе. Разумеется, могут быть возвраты товаров от покупателей, что также нужно учитывать в рассматриваемой подсистеме. Внесен в нее документ под названием «Задание персоналу». В качестве результирующего документа в рамках исследуемой подсистемы выступает отчет с названием «Продажи». Система показан на рисунке (Рисунок Error: Reference source not found.13) в виде схемы.

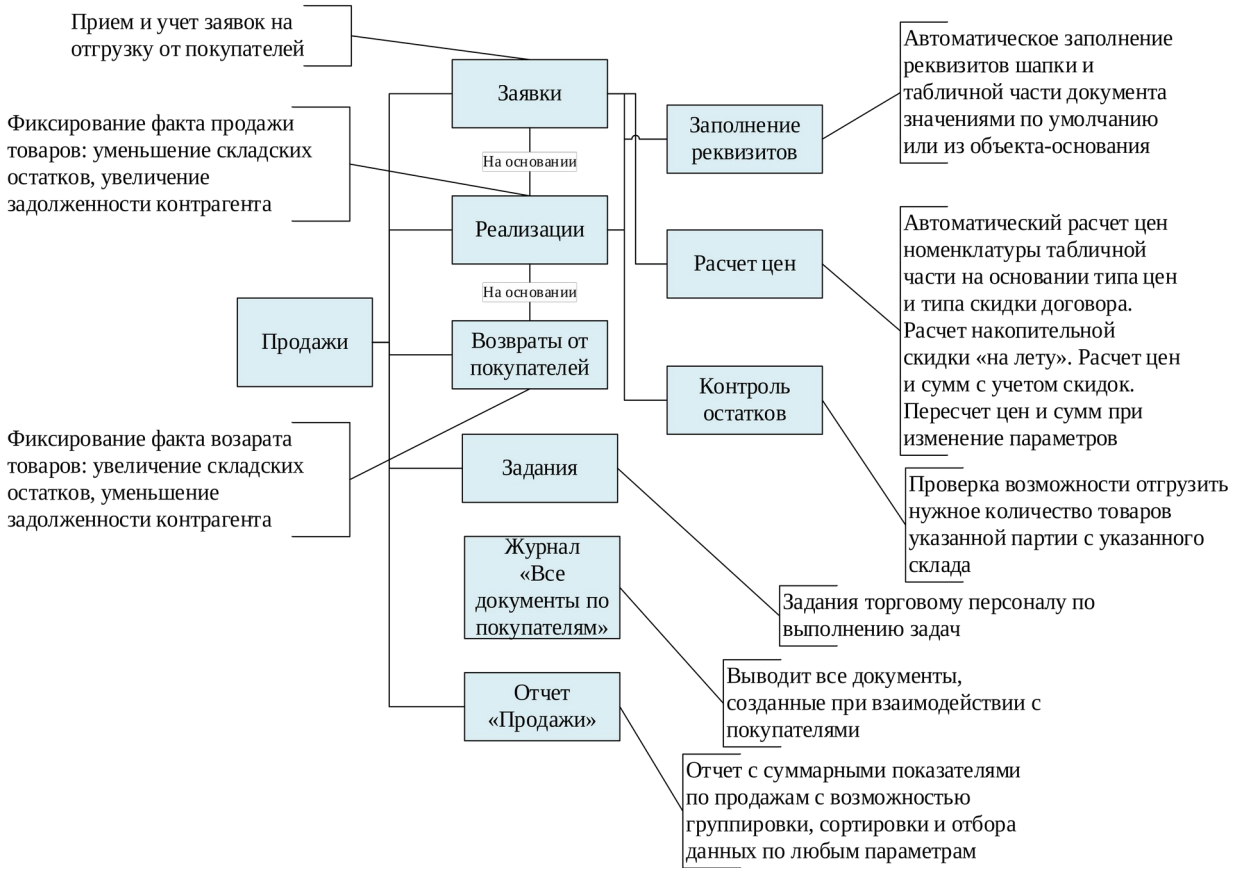

Рисунок Error: Reference source not found.13. Функции подсистемы «Продажи»

Подсистема, которая называется «Склад», разрабатывается для того, чтобы вести учет операций по складу компании. Напомним, что данный учет тесным образом связан с другим направлением работы, а именно – торговым учетом. Дело в том, что в случае возврата товара в компанию от покупателей, меняются и остатки продукции по складу. Однако заметим, что в составе функций подсистемы «Склад» есть учет взаимодействий с поставщиками. Тут предусматривается непосредственное формирование заказов, учет поступивших и возвращенных на склад предприятия товаров. «Склад» - это еще и одноименный отчет, результирующий файл по отчету «Остатки». Схематически система представлена на рисунке (Рисунок Error: Reference source not found.14).

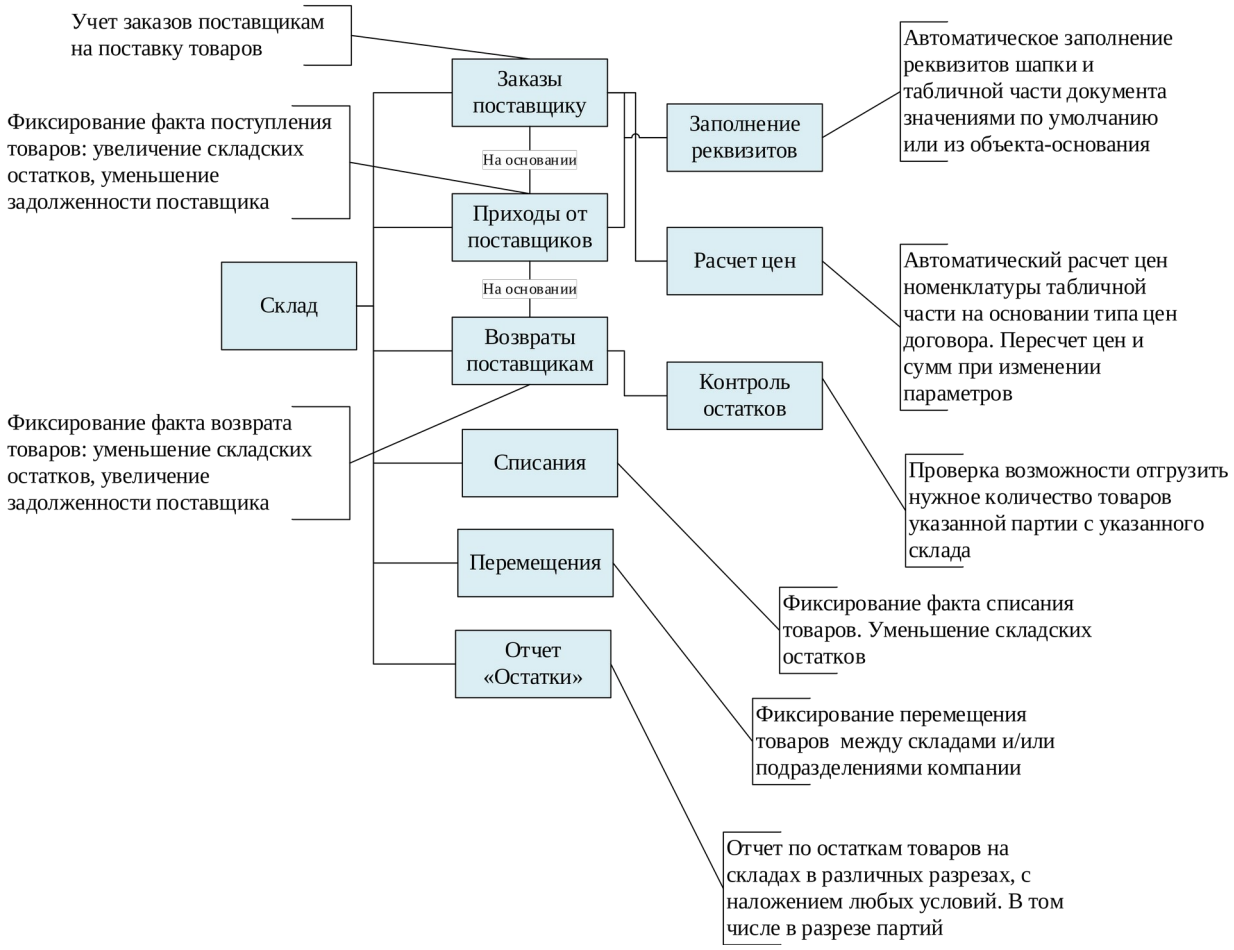

Рисунок Error: Reference source not found.14. Функции подсистемы «Склад»

Продолжая анализ подсистем, выделим «Взаиморасчеты». Это направление, посредством которого компания может осуществлять визуальное и функциональное группирование документов, которые несут сведения об оплате долгов за отгруженные покупателям товары; за оплату услуг – как наличным, так и безналичным методом расчета. На рисунке (Рисунок Error: Reference source not found.15) представлена функция такой подсистемы. В ней формируются документы – «Расход денежных средств», а также «Приход денежных средств». Их назначение расшифровывать нецелесообразно. Добавим лишь то, что на основании отчета, который называется «Взаиморасчеты», в целом можно контролировать кредиторский и дебиторский долг, долги предприятия перед поставщиками. Отчет под названием «Деньги» дает возможность контролировать в любое время

движение денег по счетам предприятия, как в наличном, так и безналичном их виде.



Рисунок Error: Reference source not found.15. Функции подсистемы «Взаиморасчеты»

В рамках создаваемой АИС, посредством которой будет решена задача складского и торгового учета, нужны следующие программные средства:

- операционная система «Windows», которая специально предназначается для серверного типа, что обеспечивает так называемое многопользовательское подключение сразу нескольких пользователей;

- ПП «1С: Предприятие 8.3».

Ранее мы говорили о том, что в компании уже есть ИТ инфраструктура, которая всецело отвечает программным, техническим требованиям новой системы. Почти все компоненты уже есть в компании:

- есть сервер, на котором работает ОС «Windows Server 2014 Standart Edition»;

- на сервере стоит сервер приложений «1С: Предприятие 8.3»;

- на сервере работает СУБД «MS SQL Server 2014»;

- у каждого сотрудника на рабочем месте есть ПК, на которых инсталлирована ОС Windows 7/10.

Принимая во внимание тот факт, что в компании уже работает «1С: Бухгалтерия 8», то покупать ее не требуется. Нужно только установить клиентскую часть платформы «1С: Предприятие 8.3» – на рабочее место каждого сотрудника. Потом их подключают к информационной базе новой АИС. Потребуется покупка только лицензий на работу новых пользователей «1С: Предприятие 8.3».

Таким образом, в нашей работе создается авторская конфигурация на базе «1С: Предприятие 8.3» под автоматизацию складского и торгового учета.

2.4 Разработка концептуальной (инфологической) модели данных

Схематически модель данных показана на рисунке (Рисунок Error: Reference source not found.16).

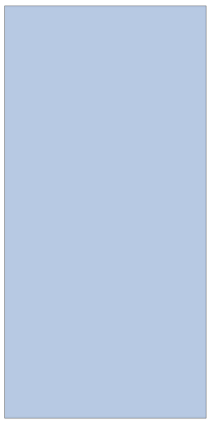

Рисунок Error: Reference source not found.16. Инфологическая модель

2.5 Разработка структуры данных

На рисунке (Рисунок Error: Reference source not found.17) схематично отражена структура справочников из подгруппы «Номенклатура».



Рисунок Error: Reference source not found.17. Справочники подгруппы «Номенклатура»

Далее перейдем к справочникам подгруппы «Контрагенты» (Рисунок Error: Reference source not found.28).



Рисунок Error: Reference source not found.28. Справочники подгруппы «Контрагенты»

И, наконец, справочники подгруппы «Организация» (Рисунок Error: Reference source not found.39).



Рисунок Error: Reference source not found.39. Справочники подгруппы «Организация»

В состав документов торгового учета входят документы «Реализация», «Заявки», «Возврат». Их структура приведена на рисунке (Рисунок Error: Reference source not found.20).



Рисунок Error: Reference source not found.20. Документы торгового учета

Структура регистров накопления приведена на рисунке (Рисунок Error: Reference source not found.1).



Рисунок Error: Reference source not found.21. Регистры накопления

2.6 Клиент-серверный вариант работы платформы «1С:Предпритие» и особенности разработки приложения для этого варианта

Клиент-серверный вариант работы — этот один из способов работы платформы «1С:Предприятие 8.3». Клиент-серверный способ функционирования предназначен для применения в рабочих группах или в масштабе предприятия. Он выполнен на основе трехуровневой архитектуры «клиент-сервер» (Рисунок Error: Reference source not found.22422).



Рисунок Error: Reference source not found.224 Трехуровневая архитектура

Работа в клиент-серверном варианте возможна как напрямую с кластером, так и через веб-сервер. При этом в случае непосредственного

подключения к кластеру толстый клиент и тонкий клиент применяют протокол TCP/IP. При подключении через веб-сервер тонкий клиент и веб-клиент применяют протокол HTTP или HTTPS (Рисунок Error: Reference source not found.2353).



Рисунок Error: Reference source not found.235 Клиент-серверный вариант работы

Вся работа с прикладными объектами, чтение и запись БД выполняется только на сервере. Функциональность форм и командного интерфейса также выполнена на сервере.

2.7 Создание прикладных объектов конфигурации

Считается, что самое главное отличие систем 1С от привычных систем прочих разработчиков – это возможность применения прикладных объектов. Под прикладным объектом принято рассматривать такой объект с платформы, который позволяет описать данные, реальное поведение объекта. В целом,

имеющаяся структура прикладного объекта дает возможность описать аналогичную структуру с предметной области. Формы же прикладного объекта определяют интерфейс для пользователя, программные модули.

Перед нами стоит задача – из существующих прикладных объектов платформы подобрать только такие объекты, которые будут в максимальной степени подходить для реализации объектов в нашей предметной сфере [7].

Для торгового учета в качестве основы выступает ведение товарной номенклатуры. База ведения номенклатуры – одноименный справочник, в котором сохранены сведения о продукции. Пример представлен на рисунке (Рисунок Error: Reference source not found.24624).

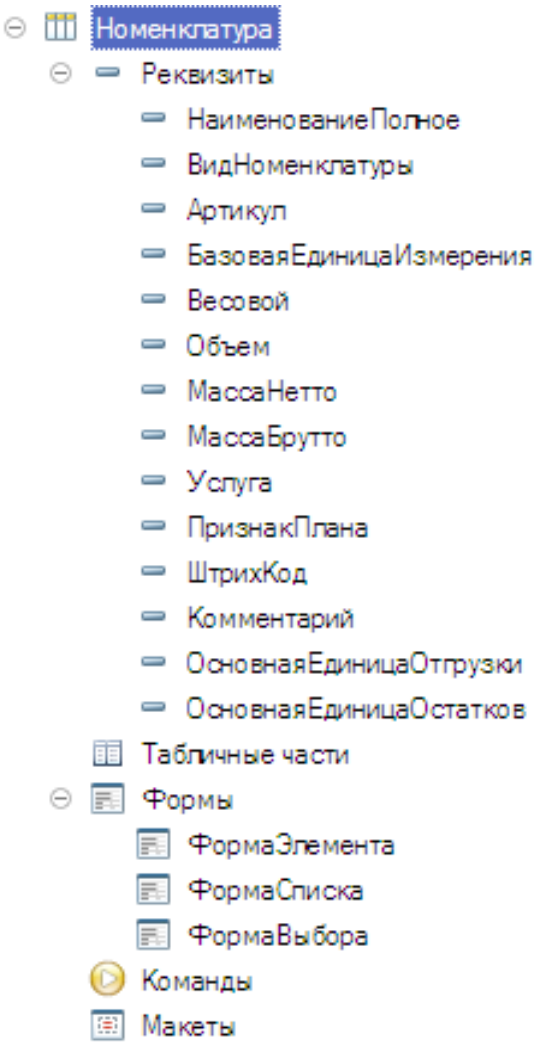

Рисунок Error: Reference source not found.246. Справочник «Номенклатура»

Для номенклатуры ведение цен организовано посредством периодического регистра данных, который называется «Цены». В нем обычно хранят значение цены на продукцию – в разрезе типов цен. Далее на рисунке (Рисунок Error: Reference source not found.25725) будут обозначены конфигурационные объекты, посредством которых реализуется механизм для учета цен.

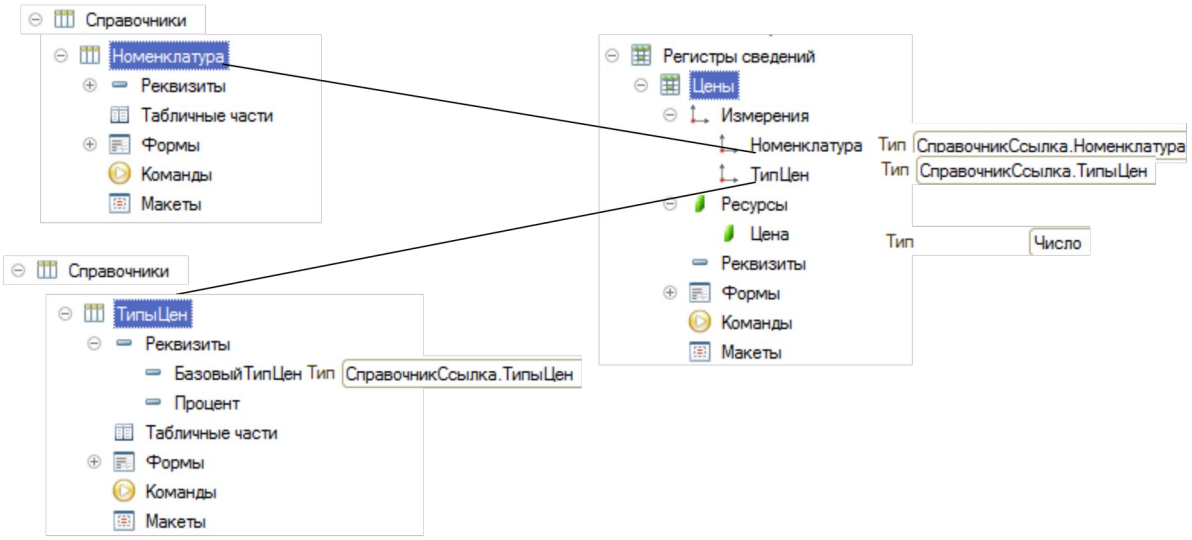

Рисунок Error: Reference source not found.257. Реализация механизма учета цен

Еще одна, не менее важная функция системы – это ведение списка контрагентов компании. Основной здесь выступает справочник, который называется «Контрагенты». В нем собраны основные сведения по всем контрагентам, с которыми сотрудничает предприятие.

Главная особенность АИС, которую мы прорабатываем в рамках настоящей работы, состоит в возможности хранения большого количества дополнительных свойств по контрагентам, часть с которых уже известна в момент начала разработки. Заметим, что свойства – это не просто текстовое, дополнительное описание объекта, а структурированная форма, которая может применяться в качестве основы для отбора, поиска и сортировки, группировки по справочникам, отчетам системы.

Известно о том, что структурированные параметры объектов группируются в справочнике, который называется «Наборы свойств». Он подчиняется исходному справочнику, в нашем случае – «Контрагенты», как показано на рисунке (Рисунок Error: Reference source not found.26826).

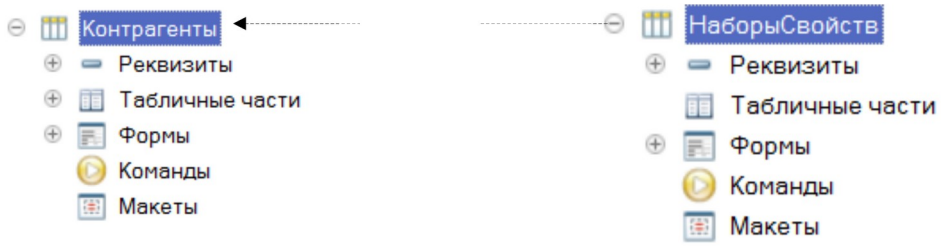

Рисунок Error: Reference source not found.268. Справочник «Наборы свойств»

Все операции, которые осуществляются в хозяйственной активности предприятия, в обязательном порядке заверяются документами. Структура их шапки, табличные части демонстрируют состав данных, посредством которых описывается конкретная операция. В новой конфигурации, которую мы прорабатываем, документы строго разделены по подсистемам. Так, к подсистеме «Продажи» можно отнести документы «Заявка», «Задание», «Реализация», «Возврат от покупателя». В подсистеме, которая называется «Склад», числятся документы «Заказ для поставщика», «Приход от поставщика», «Возврат поставщику» и так далее. В целом, описанный механизм будет приведен схематически на рисунке (Рисунок Error: Reference source not found.97).

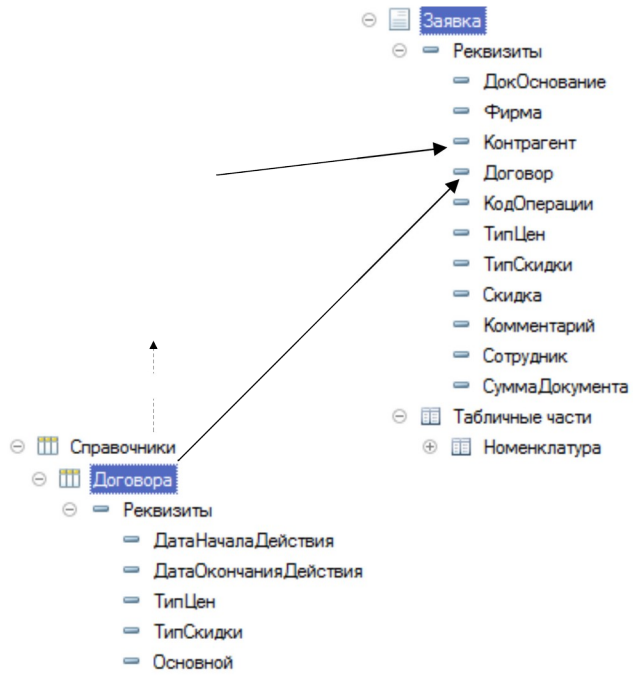

Рисунок Error: Reference source not found.97. Механизм ввода документов на основании

Пакет документов, которые созданы в «Конфигурации», позволяет исключительно зафиксировать факт осуществления операций хозяйствования. Тем не менее, каждая операция, как правило, влечет за собой трансформацию хранимой информации в базе. Хранение данных реализовано через инструменты механизма платформы – «Регистры накопления». Тут формируется основа учетного механизма по наличию, а также движению финансов. На рисунке (Рисунок Error: Reference source not found.108) будут отражены регистры в рамках конфигурации.

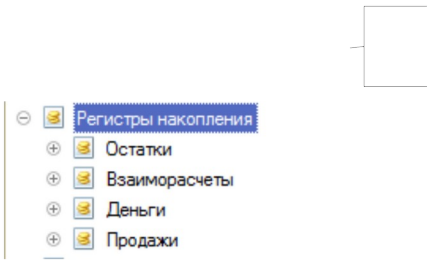

Рисунок Error: Reference source not found.108. Состав регистров конфигурации

9 показывает пример настройки движений документа «Реализация» по регистрам учета.

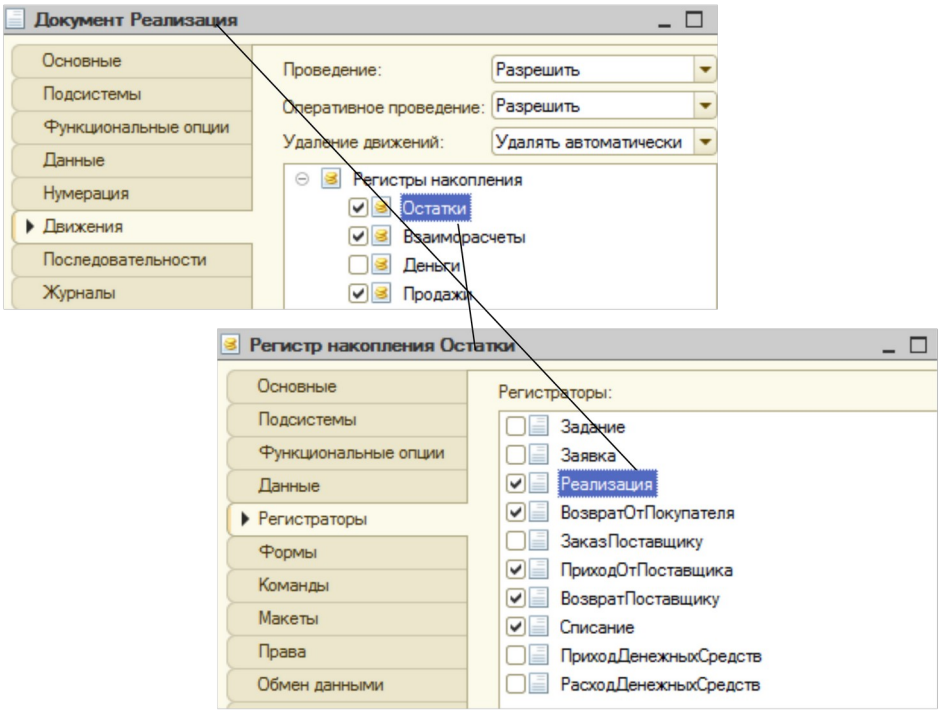

Рисунок Error: Reference source not found.119. Настройка движений документа

Далее на рисунке (Рисунок Error: Reference source not found.) показана схема, согласно которой осуществляется движение документов по регистрам.

Примечательно, что такой регистр накопления, как «Продажи» - оборотным, а потому по нему не может, ни расти, ни уменьшаться показатель ресурсов. Тут сугубо фиксируются обороты в разрезе операционных кодов.

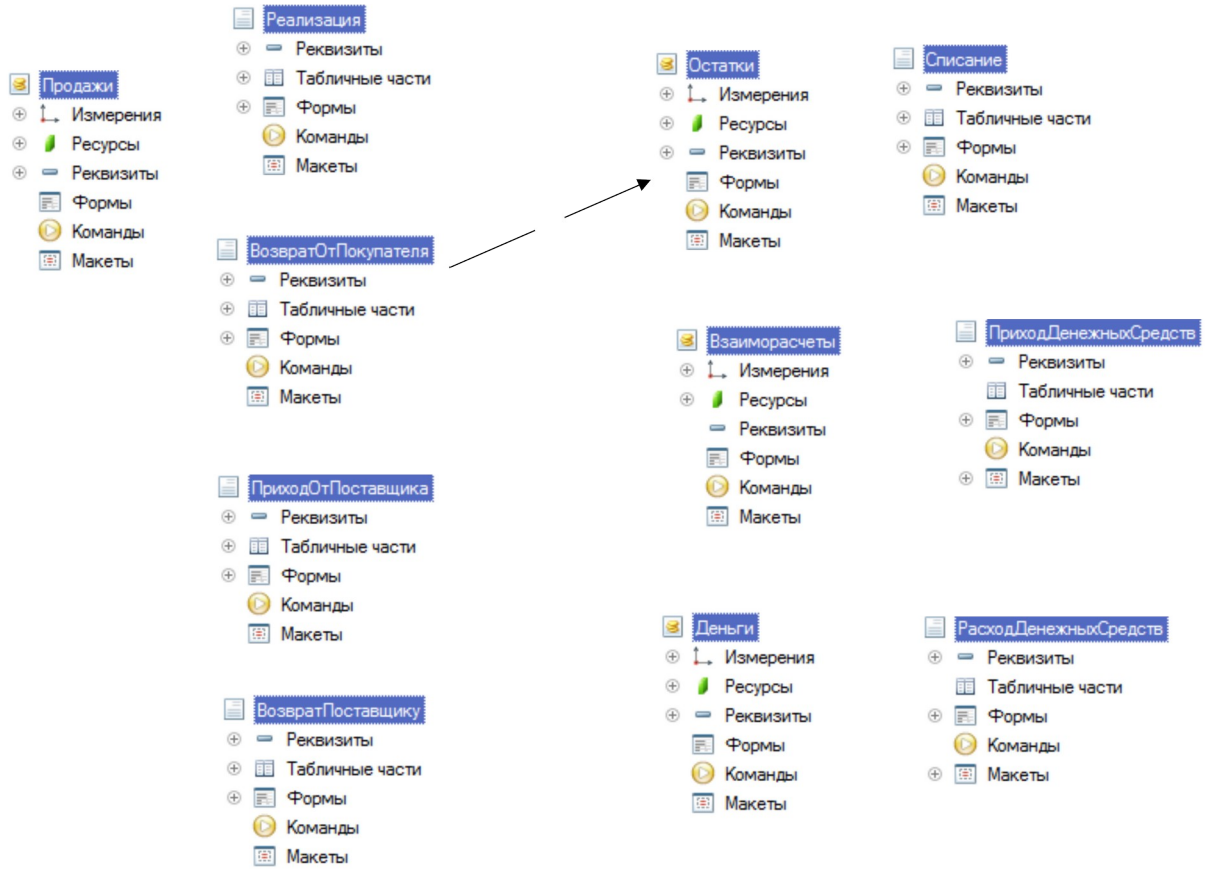

Рисунок Error: Reference source not found.30. Движения документов по регистрам

Практическим путем было установлено, что многочисленные отчеты в рамках конфигурации разрабатываются посредством механизма «системы компоновки данных». Ее схема показана на рисунке (Рисунок Error: Reference source not found.3112) ниже.

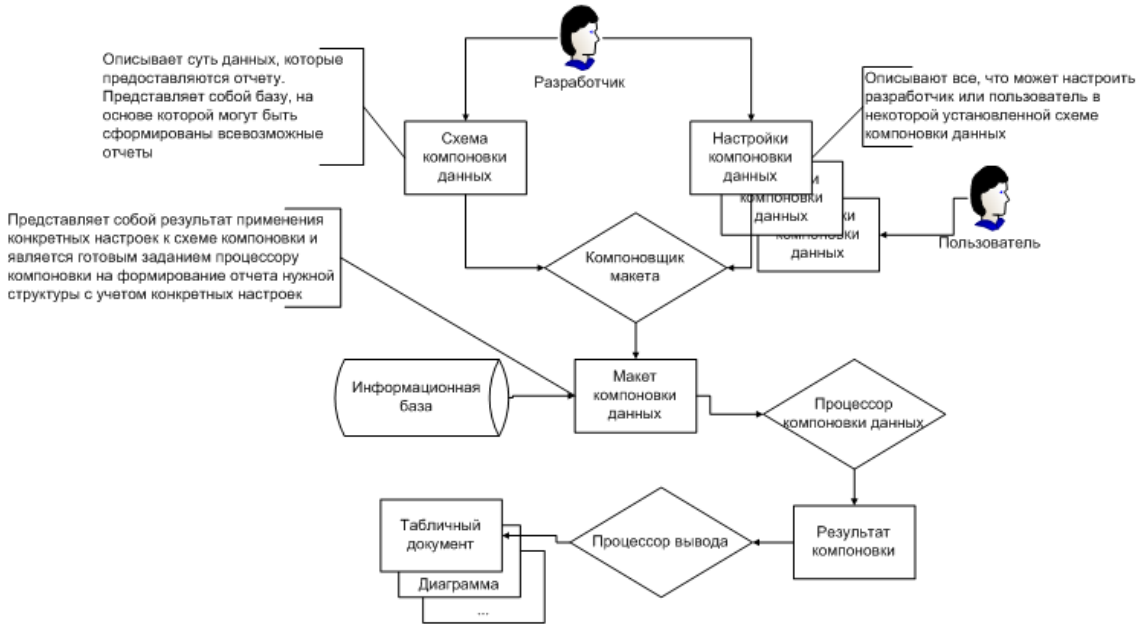

Рисунок Error: Reference source not found.3112. Система компоновки данных

Практика показывает, что формирование схемы осуществляется посредством конструктора. С его помощью можно в полной мере описать сведения нового отчета. Так, настройками задается разное количество отчетов; устанавливается структура вводимых сведений, прочее.

Если система работает в режиме «Предприятие», то можно сформировать отчет «Отчет по партиям».

Итак, мы установили, что новая система компоновки данных позволяет выстраивать прочие отчеты конфигурации. В списке они показаны на рисунке (Рисунок Error: Reference source not found.313).



Рисунок Error: Reference source not found.313. Состав отчетов системы

Наличие такого количества отчетов в конфигурации позволяет вести разнообразную отчетность.

2.8 Безопасность информационной базы и методы защиты

Информационная безопасность – задача комплексная, нацеленная на обеспечение безопасности, защиты информации от хищения, путем разработки системы безопасности и соблюдений правил пользования информационной базой.

Более ранние версии платформы 1с внимание безопасности не уделялось совсем, но с выходом 1С:Предприятия 8.3 отношение к защите информационной системе улучшилось.

Проблемы и потенциальные угрозы безопасности базы 1С:

-использование базы в файловом формате. Любой пользователь, который имеет право работать в файловой базе 1с, может удалить или скопировать базу.

- несанкционированный доступ к серверу. В случае незаконного доступа к серверу компании злоумышленники могут использовать этот доступ в целях

кражи или искажения информации. Сюда можно отнести передачу логинов/паролей другим пользователям, нарушение рулевой модели(полные права).

-Утечки информации.Например: потеря данных из базы(выгрузки),разглашение информации в прочей форме(социальные сети и тд.)

Существует перечень решения подобных проблем:

-использования лицензионной версией продукта 1С

-не хранить пароли в открытом виде, а также не использовать одинаковые пароли.

-ограничение обычных пользователей в административных правах в информационной системе

-политика соблюдений конфиденциальности данных уволенными сотрудниками и блокировок доступа их к информационным ресурсам

-изолировать внешних пользователей от основной сети

-запретить использование непроверенных внешних обработок

-запретить все виды соединений, кроме Тонкого клиента

-настроить профиль безопасности и администраторов кластера серверов 1С

-использования антивирусных систем

-своевременная установка всех обновлений операционных систем

Подводя итог всему вышесказанному, хотелось бы отметить, что главным правилом защиты информации является ограничение прав и возможностей пользователей, а также их контроль при использовании информационных систем. Чем меньше прав у пользователя при работе с информационной системой, тем меньше вероятность утечки или повреждения информации по злому умыслу или по неосторожности.

2.9 Контрольный пример реализации

В ходе испытания разработки проверяется ее работоспособность, каждая функция. Список таких функций отражается в таблице (Таблица Error: Reference source not found.2). Таблица Error: Reference source not found.2. Перечень проверяемых функций

№ п/п	Проверяемая функция	Примечание
1	Заполнение справочников	На примере справочников «Номенклатура»
2	Заполнение документов	На примере документа «Реализация»
3	Составление отчетов	Отчет «Остатки»

Описание проверки функции ведения справочников приведено в таблице (Таблица Error: Reference source not found.3).

Таблица Error: Reference source not found.3. Проверка функции «Ведение справочников»

Параметр	Значение
Функция	Ввод справочников
Предмет проверки	Формы на экране, а также их эргономичность Выведение нужных полей Поля, которые обязательны к заполнению
Используемые средства	Осуществление визуального контроля
Исходные данные	Программная БД, что заполняется вручную некоторыми информационными источниками
Искажение текста для установления внештатных случаев	Введение номенклатуры с не заполненным полем, которое называется «Основная единица измерения»
Ожидаемая реакция	Выведение сообщения о том, что нужно заполнить конкретное поле
Как определяются результаты	Осуществление визуального контроля

Описание проверки функции ведения документов приведено в таблице (Таблица Error: Reference source not found.4).

Таблица Error: Reference source not found.4. Проверка функции «Ведение документов»

Параметр	Значение

Функция	Введение документа под названием «Реализация»
Предмет проверки	Наложение запрета на проведение документа, в случае отсутствия требуемого количества товаров на запасах учета «Номенклатура», «Склад», «Партия»
Применяемые средства	Проведение визуального контроля
Исходные данные	Программная БД, которая заполняется вручную некоторыми сведениями
Искажения тестовой информации для имитации нештатных ситуаций	Попытка продать 60 товарных единиц в случае наличие на складе 50 единиц
Ожидаемая реакция	Появляется сообщение о том, что провести документ невозможно
Способ определения результатов	Осуществление визуального контроля

Описание проверки функции ведения документов приведено в таблице (Таблица Error: Reference source not found.5).

Таблица Error: Reference source not found.5. Проверка функции «Составление отчетов»

Параметр	Значение
Функция	Создание отчета «Отчет»
Предмет проверки	Выведение всех документов в отчет по движению
Используемые средства	Проведение визуального контроля
Исходные данные	Программная БД, что заполняется вручную
Искажения тестовой информации для имитации нештатных ситуаций	Проведение еще одного документа «Реализация»
Ожидаемая реакция	Проведение документа в отчет. Изменение конечного остатка
Способ определения результатов	Визуальный контроль

Испытание проводится выполнением контрольного примера, с выполнением функций, которые были обозначены нами ранее.

Проверка работы на основе справочника «Номенклатура» проводится по примеру конкретной позиции, что заимствуется со справочника. Когда записываются данные номенклатуры по позиции с незаполненным полем «Базовая единица измерения», возникает ошибка.

Теперь нужно проверить остатки номенклатуры на момент проверки самого документа. Чтобы выполнить эту работу, создается заявка от

покупателя, согласно которой отгружается товар, имеющийся на складе. Документ под названием «Заявка» проводится, ведь функции контроля остатка нет. Что касается документа под названием «Реализация», то в нем есть точно такое количество отгружаемой продукции, а потому выдается ошибка, когда предпринимается попытка проведения документа. Соответственно, система указывает на то, что количественные остатки продукции увести в минус невозможно.

Тестирование показывает, что система работает правильно; проверены все контрольные функции. Требуемая функциональность – реализована в полной мере.

3 Оценка и обоснование экономической эффективности проекта

3.1 Выбор и обоснование методики расчета экономической эффективности

Как правило, внедрение компьютерной технологии для обработки информации направлено на повышение эффективности производства за счет совершенствование управленческий процессов, что проявляется в показателях производственно-хозяйственной деятельности объекта управления. Поскольку реализация мер по использованию компьютерной техники, средств связи и оргтехники для целей управления требует значительных финансовых затрат, затраты на реализацию проектов по информационным и вычислительным услугам для пользователей должны быть окуплены как можно скорее. При создании автоматизированной системы управления компания несет единовременные затраты на ее разработку, приобретение необходимого аппаратного и программного обеспечения, текущие затраты на эксплуатацию системы, обучение и переподготовку персонала.

Методика расчета экономической эффективности данного проекта включает в себя расчет таких основных показателей, как годовая экономия, коэффициент экономической эффективности капитальных вложений, срок окупаемости капитальных вложений. Поскольку решение задачи "торговый и складской учет" был без использования компьютерной техники, расчет экономической эффективности будет осуществляться по следующей методике.

Годовая экономия от сокращения ручного труда по обработке информации рассчитывается по формуле:

$$Э = C_{p1} - C_{p2} \quad (20)$$

$Э$ – годовая экономия от сокращения ручного труда по обработки информации, руб.;

C_{p1} – годовые эксплуатационные затраты при ручной обработке информации, руб.

$$C_{p1}=\frac{Z_1 \cdot T_1}{Q}\cdot(1+\alpha)\cdot(1+\beta)\cdot 12 \quad (21)$$

Z_1 – месячная основная зар.плата маркетолога, руб.;

T_1 – месячные трудовые затраты на машинное решение задачи, человеко-дни;

Q – количество рабочих дней в месяц, среднее(дни);

α – коэффициент накладных расходов;

β – коэффициент дополнительной заработной платы (отчисление на социальное страхование, в различные фонды и т.п.).

$$C_{p2}=C_1+C_2+C_3 \quad (22)$$

C_1 – годовые затраты на решение задачи, руб.;

C_2 – годовые затраты на заполнение документов, анализ и корректировку данных (ручные операции), руб.;

C_3 – годовые затраты на настройку оборудования при решении задачи, руб.

$$C_1=12\cdot\sum_q S_q \cdot T_q \quad (23)$$

S_q – себестоимость часа работы оборудования q при решении задачи, руб.;

T_q – время работы оборудования q при решении задачи в течении месяца, машино-часы.

$$C_3=K_3 \cdot y \quad (24)$$

K_3 – годовые единовременные затраты на обучение персонала, адаптацию, настройку оборудования при решении задачи, руб.;

$У$ – коэффициент настройки оборудования.

$$K_3 = K_{3.1} + K_{3.2} + K_{3.3} \quad (25)$$

$K_{3.1}$ – годовые единовременные затраты по заработной плате персонала на обучение, адаптацию, настройку оборудования для решения задачи, руб.

$$K_{3.1} = \frac{Z_1' \cdot T_1'}{Q} \cdot (1+\alpha) \cdot (1+\beta) \cdot 12 \quad (26)$$

Z_1' – месячная основная заработная плата маркетолога, руб.;

T_1' – месячные трудовые затраты маркетолога на машинное решение задачи, человеко-дни.

$K_{3.2}$ – годовые единовременные затраты машинного времени, руб.

$$K_{3.2} = 12 \cdot \sum_q S_q \cdot T_q' \quad (27)$$

T_q' – время работы оборудования q на обучение персонала, адаптацию, настройку оборудования при решении задачи, машино-часы.

$K_{3.3}$ – прочие единовременные расходы, руб.

$$K_{3.3} = (K_{3.1} + K_{3.2}) \cdot h \quad (28)$$

h – коэффициент прочих расходов, к которым относятся расходы на приобретение машинных носителей, бумаги, краски и т.п.

Коэффициент экономической эффективности E_p рассчитывается по формуле:

$$E_p = \frac{Э}{K} \quad (29)$$

Если $E_p \geq E_{нвт}$, то технология является эффективной.

$E_{нвт}$ – нормативный коэффициент капитальных вложений. Он определяет нижнюю границу годовой экономии, которую можно получить на 1 рубль капитальных затрат.

T – срок окупаемости затрат на решение задачи:

$$T=\frac{K}{Э} \quad (30)$$

K – единовременные затраты на решение задачи, руб.

$$K=K_1+K_2+K_3 \quad (31)$$

K_1 – единовременные затраты на проектирование, руб.

$$K_1=\frac{\sum Z_2\cdot T_2\cdot(1+\alpha)\cdot(1+\beta)}{Q}\cdot n \quad (32)$$

T_2 – месячные трудовые затраты системного администратора (программиста) на проектирование решения задачи, человеко-часы;

n – длительность проектирования, год.

K_2 – затраты на оборудование, руб.

$$K_2=\frac{C_b\cdot\left(1-t\cdot\frac{r}{100}\right)\cdot T_q\cdot(1+\alpha_0)}{F} \quad (33)$$

C_b – балансовая стоимость комплекта техники или ПЭВМ, руб.;

t – длительность эксплуатации ПЭВМ до начала решения задачи, годы;

r – годовая норма на реновацию оборудования;

T_q – время работы оборудования q при решении задачи в течение месяца, машино-часы;

F – планируемый годовой фонд времени работы ПЭВМ (оборудования).

$$F=t_c\cdot T_c \quad (34)$$

t_c – среднесуточная фактическая загрузка ПЭВМ, часы;

T_c – среднее количество дней работы ПЭВМ в году, часы;

По приведённым выше формулам определим показатели экономической эффективности:

Годовая экономия $Э$;

Коэффициент экономической эффективности капитальных вложений E_p;

Срок окупаемости капитальных вложений T.

3.2 Расчет показателей экономической эффективности

Значения слагаемых для расчёта показателей экономической эффективности по описанной выше методике приведены в Таблице 1.

Таблица 1

Слагаемые для расчёта показателей экономической эффективности

Показатели	Обозначение	Единица измерения	Величина показателя	
			До внедрения	После внедрения
Месячная зарплата сотрудника отдела персонала	Z_1	руб.	20000	23000
Затраты труда сотрудника отдела персонала	T_1	чел./день	21	11
Месячная зарплата проектировщика	Z_2	руб.	–	30000
Затраты труда проектировщика	T_2	чел./день	–	9
Коэффициент накладных расходов	α	–	0,2	0,2
Коэффициент дополнительной зарплаты	β	–	0,3	0,3
Себестоимость часа работы ПЭВМ	S_q	руб./час	–	25
Время работы ПЭВМ для решения задачи	T_q	маш./час	–	10

Показатели	Обозначение	Единица измерения	Величина показателя	
			До внедрения	После внедрения
(мес)				
Время работы ПЭВМ на обучение, адаптацию, настройку оборудования	T_q'	маш./час	–	6
Коэффициент настройки оборудования	y	–	–	0,1
Длительность проектирования	n	год	–	0,208
Среднее количество рабочих дней в месяце	Q	дни	22	22
Коэффициент прочих расходов	h	–	0,1	0,1
Нормативный коэффициент эффективности капитальных вложений	$E_{нвт}$	–	–	0,3

Расчёт годовых эксплуатационных затрат при ручной обработке информации:

$$C_{p1}=\frac{(20000*21)*1,2*1,3}{22}*12=357381,82 \ руб.$$

Годовые эксплуатационные расходы при машинном варианте обработки экономической информации складываются из годовых затрат машинного времени на решение задачи (C_1), годовых затрат ручного труда (C_2), годовых затрат на настройку оборудования и обучение персонала (C_3):

$$C_1=12\cdot25\cdot10=3000 \ руб.$$

$$C_2=\frac{(23000\cdot11)\cdot1,2\cdot1,3}{22}\cdot12=17940 \ руб.$$

Единовременные затраты на обучение персонала, адаптацию и настройку оборудования при решении задачи K_3 определяем как сумму затрат $K_{3.1}$, $K_{3.2}$, $K_{3.3}$:

$$K_{3.1}=\frac{(30000\cdot 9)\cdot 1{,}2\cdot 1{,}3}{22}\cdot 12=229745{,}45\ руб.$$

$$K_{3.2}=12\cdot 25\cdot 6=1800\ руб.$$

$$K_{3.3}=(229745{,}45+1800)\cdot 0{,}1=23154{,}54\ руб.$$

$$K_3=229754{,}45+1800+23154{,}54=254708{,}99\ руб.$$

$$C_3=254708{,}99*0{,}1=25470{,}99\ руб.$$

$$C_{p2}=3000+17940+254780{,}99=275720{,}99\ руб.$$

Годовая экономия:

$$Э=357381{,}82-275720{,}99=81660{,}83\ руб.$$

Для расчета единовременных затрат на создание и внедрение системы (K) производится расчет затрат на проектирование (K_1), затрат на оборудование (K_2), единовременные затраты на обучение персонала, адаптацию и настройку оборудования (K_3) был рассчитан ранее:

$$K_1=\frac{(30000\cdot 9)\cdot 1{,}2\cdot 1{,}3}{22}\cdot 0{,}208=3982{,}25\ руб.$$

Затраты на новое оборудование отсутствуют, так как разработки ориентированы на использование существующей техники $(K_2=0)$.

$$K_3=237852{,}01\ руб.$$

$$K=3982{,}25+0+254708{,}99=258691{,}24\ руб.$$

$$E_p=\frac{81660.83}{258691{,}24}\approx 0{,}316$$

Нормативный коэффициент экономической эффективности равен 0,3, при расчете по данному проекту он составил 0,31, неравенство $0{,}316\geq 0{,}3$ выполняется, следовательно, решение задачи «разработка информационной системы торгового и складского учета» с использованием ЭВМ эффективно.

Срок окупаемости затрат на решение задач:

$$T = \frac{1}{0,316} = 3,16$$

Что составляет примерно 35 месяцев.

Результаты расчета экономической эффективности проекта – годовая экономия 81660 рублей.

Заключение

В выпускной квалификационной работе изложен порядок разработки информационной системы торгового и складского учета. Рассмотрены основные характеристики объекта управления, его функциональная и организационная структуры.

В результате анализа технологии решения данной проблемы на торговом предприятии были предложены изменения в технологии обработки информации, позволяющие своевременно принимать управленческие решения. Задачи исследования, которые были достигнуты:

- была изучена предметная область, изучены процессы и подпроцессы работы предприятия, которые автоматизировали
- построена функциональная структура разрабатываемой системы
- спроектирована системы учета работы торгового предприятия
- разработана программа проекта решения задачи
- протестирован полученный проект, устранены недостатки

Таким образом, можно сделать вывод, что несмотря на финансовые затраты, которые понесет торговое предприятие ООО «Стройрем» на проектирование и разработку информационной системы в будущем минимизирует затраты предприятия и увеличит прибыль предприятия, т.к. предлагаемая технология экономически эффективна.

Список использованных источников

1. “1С:Предприятие 8. Система программ [Электронный ресурс]. - http://v8.1c.ru/
2. 1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы (+диск). М.Г. Радченко, Е.Ю. Хрусталева. 968 с. 1С-Паблишинг, 2014 г.
3. Профессиональная разработка в системе «1С:Предприятие 8» (+DVD-ROM). Издание 2. Ажеронок В.А., Габец А.П., Гончаров Д.И., Козырев Д.В., Кухлевский Д.С., Островерх А.В., Радченко М.Г., Хрусталева Е.Ю. Под редакцией М. Г. Радченко. 870 с. 1С-Паблишинг, 2013 г.
4. Решение специальных прикладных задач в “1С:Предприятии 8.3”. Серия “1С:Профессиональная разработка”. Гончаров Д. И., Хрусталева Е. Ю. 935 с. 1С-Паблишинг, 2015 г.
5. Архитектура и работа с данными “1С:Предприятия 8.3”. Серия «1С:Профессиональная разработка». Радченко М. Г., Хрусталева Е. Ю. 650 с. 1С-Паблишинг, 2013 г.
6. Реализация прикладных задач в системе “1С:Предприятие 8.2” (+CD). Серия “1С:Профессиональная разработка”. Габец А.П., Козырев Д.В., Кухлевский Д.С., Хрусталева Е.Ю. 830 с. 1С-Паблишинг, 2014 г.
7. Разработка управляемого интерфейса (+CD). Серия “1С:Профессиональная разработка”. Ажеронок В.А., Островерх А. В., Радченко М. Г., Хрусталева Е. Ю. 740 с. 1С-Паблишинг, 2015 г.
8. 1С:Предприятие 8.2. Практическое пособие разработчика. Примеры и типовые приемы. Радченко М.Г., Хрусталева Е.Ю. 620 с. 1С-Паблишинг, 2012 г.
9. 1С:Предприятие 8.3. Коротко о главном. Новые возможности версии 8.2. Радченко М.Г. 690 с. 1С-Паблишинг, 2012 г.

10. 101 совет начинающим разработчикам в системе “1С:Предприятие 8”. Хрусталева Е. Ю. 214 с. 1С-Паблишинг, 2011 г.
11. Вендров А.М. CASE технологии Современные методы и средства проектирования информационных систем М.: Финансы и статистика, 1998. - 176 с.: ил.;
12. Методология функционального моделирования IDEF0, Руководящий документ, Госстандарт России.;
13. Диго С.М. Проектирование и использование баз данных Учебник. М.: Финансы и статистика. 1995 г;
14. Основы построения баз данных под ред. А.Д. Хомоненко Санкт-Петербург, 2004;
15. ГОСТ Р ИСО/МЭК 12207-99, Руководящий документ, Госстандарт России, Москва, 2004;
16. Котлер Ф. Маркетинг менеджмент / Пер. с англ. под ред. Л.А. Волковой, Ю.Н. Каптунеревского. – СПб.: Питер, 2002. – 756с.р
17. Матищев А.Н. Эффективность рекламы. – М.: Издательство «Финпресс», 2002. – 416с.
18. Новости и технологии торговли [Электронный ресурс]. – Режим доступа: http://www.torgrus.com
19. Программирование для всех [Электронный ресурс]. – Режим доступа: http://www.realcoding.net/
20. ГОСТ Р ИСО/МЭК 12207-99 «Информационная технология. Процессы жизненного цикла программных средств» (принят и введен в действие постановлением Госстандарта РФ от 23.12.1999 №675-ст).
21. ГОСТ Р ИСО/МЭК 15271-2002 «Информационная технология. Руководство по применению ГОСТ Р ИСО/МЭК 12207-99 (Процессы жизненного цикла программных средств)» (принят и введен в действие постановлением Госстандарта РФ от 05.06 2002 №227-ст).

22. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2000.-187с.

Приложение 1. Текст программных модулей

Общий модуль «Общий Срв»

```
Функция СоздатьОписаниеТипов(НазваниеТипа, Длина = "", Точность = "") Экспорт
	мас = Новый Массив;
	мас.Добавить(Тип(НазваниеТипа));
	Если Длина <> "" Тогда
		Если Точность = "" Тогда
			КвалифСтроки = Новый КвалификаторыСтроки(Длина, ДопустимаяДлина.Переменная);
		Иначе
			КвалифЧисла = новый КвалификаторыЧисла(Длина, Точность, ДопустимыйЗнак.Любой);
		КонецЕсли;
	КонецЕсли;
	Возврат Новый ОписаниеТипов(мас, КвалифСтроки, КвалифЧисла);
КонецФункции

Функция ПолучитьОсновнойТипЦен() Экспорт
	Возврат Константы.ОсновнойТипЦен.Получить();
КонецФункции

Функция ПолучитьОсновнойТипСкидки() Экспорт

	Возврат Константы.ОсновнойТипСкидок.Получить();
КонецФункции

// для одной строки документов при расчете в ней цены и суммы
Процедура РассчитатьСуммуСтроки(Дата, Номенклатура, Количество, Коэфф, ТипЦен, Скидка=0, Цена, ЦенаСоСкидкой=0, Сумма, СуммаСкидки=0) Экспорт

	Цена = ПолучитьЦену(Дата, Номенклатура, ТипЦен);
	ЦенаСоСкидкой = Цена - ((Цена*Скидка)/100);
	Сумма = ЦенаСоСкидкой * Количество*Коэфф;
	СуммаСкидки = Цена*Количество*Коэфф - ЦенаСоСкидкой*Количество*Коэфф;
КонецПроцедуры

// ПО КОНТРАГЕНТАМ
// получить ответсвенного сотрудника для контрагента
Функция ПолучитьОтветсвенного(КонтрагентСсылка) Экспорт

	Если Не КонтрагентСсылка.Пустая() Тогда
		Возврат КонтрагентСсылка.Отвественный;
```

```
    КонецЕсли;
    Возврат Неопределено;

КонецФункции

// получить основной договор контрагента
Функция ПолучитьОсновнойДоговор(ДляКонтрагента, НаДату, ТипЦен=0, ТипСкидки=0, Скидка=0) Экспорт

    глПустаяДата = Дата(01,01,0001,00,00,00);
    Договор = Неопределено;

    Если НаДату = глПустаяДата Тогда        // если дата еще не известна - то берем на текущую дату
        НаДату = ТекущаяДата();
    КонецЕсли;
    // запросом выбираем все договора этого контрагента с признаком Основной и действующие по дате
    Запрос = Новый Запрос;
    Запрос.Текст =
    "ВЫБРАТЬ
    |   Договора.Ссылка                          КАК Договор,
    |   Договора.Владелец                        КАК Владелец,
    |   Договора.Основной                        КАК Основной,
    |   Договора.ДатаНачалаДействия              КАК ДатаНачалаДействия,
    |   Договора.ДатаОкончанияДействия           КАК ДатаОкончанияДействия
    |ИЗ
    |   Справочник.Договора КАК Договора
    |   ГДЕ Владелец = &ДляКонтрагента И
    |  Основной = ИСТИНА И
    |   ДатаНачалаДействия    <= &НаДату
//  |   ДатаОкончанияДействия >= &НаДату
    |";

    Запрос.УстановитьПараметр("ДляКонтрагента", ДляКонтрагента);
    Запрос.УстановитьПараметр("НаДату", НаДату);
    Результат = Запрос.Выполнить();

    Выборка = Результат.Выбрать();
    Пока Выборка.Следующий() Цикл    // если в выборке - несколько - то выбираем первый попавшийся
        Договор = Выборка.Договор;
        Прервать;
    КонецЦикла;

    Если Договор = Неопределено Тогда
        ТипЦен    = Константы.ОсновнойТипЦен.Получить();        // по умолчанию
        ТипСкидки = Константы.ОсновнойТипСкидок.Получить();     // по умолчанию
    Иначе
        ТипЦен    = Договор.ТипЦен;             // из договора
```

```
            ТипСкидки = Договор.ТипСкидки;          // из договора
        КонецЕсли;
        Скидка = СкидкаПоТипуСкидок(НаДату, ТипСкидки, ДляКонтрагента);      // значение скидки

        Возврат Договор;

КонецФункции

// получить основной банковский счет контрагента/организации
Функция ПолучитьОсновнойСчет(ДляКонтрагента) Экспорт

        // запросом выбираем все договора этого контрагента с признаком Основной и действующие по дате
        Запрос = Новый Запрос;
        Запрос.Текст =
        "ВЫБРАТЬ
        |       Счета.Ссылка                     КАК Счет,
        |       Счета.Владелец                   КАК Владелец,
        |       Счета.Основной                   КАК Основной
        |ИЗ
        |       Справочник.БанковскиеСчета КАК Счета
        |       ГДЕ Владелец = &ДляКонтрагента И
        |  Основной = ИСТИНА
        |";

        Запрос.УстановитьПараметр("ДляКонтрагента", ДляКонтрагента);
        Результат = Запрос.Выполнить();

        Выборка = Результат.Выбрать();
        Пока Выборка.Следующий() Цикл    // если в выборке - несколько - то выбираем первый попавшийся
            Возврат Выборка.Счет;
        КонецЦикла;

        Возврат Неопределено;

КонецФункции


// получить тип цен из договора
Функция ПолучитьТипЦенИзДоговора(ДоговорСсылка) Экспорт

        Если Не ДоговорСсылка.Пустая()  Тогда
            Возврат ДоговорСсылка.ТипЦен;
        КонецЕсли;
        Возврат Неопределено;

КонецФункции

// получить тип скидки из договора
Функция ПолучитьТипСкидкиИзДоговора(ДоговорСсылка) Экспорт
```

```
	Если Не ДоговорСсылка.Пустая()  Тогда
		Возврат ДоговорСсылка.ТипСкидки;
	КонецЕсли;
	Возврат Неопределено;

КонецФункции

// рассчитать значение скидки (в %) по типу скидки (постоянная)  и объему продаж контрагента (накопительная)
Функция СкидкаПоТипуСкидок(НаДату, ТипСкидки, Контрагент) Экспорт

	Если ТипСкидки.Пустая() Тогда
		Возврат 0;
	КонецЕсли;
	Скидка = ТипСкидки.ПостояннаяСкидка;		// как минимум есть постоянная
	Если Не ТипСкидки.НакопительнаяСкидкаЕсть Тогда // если накопительной нет - то конец
		Возврат Скидка;
	КонецЕсли;
	Если ТипСкидки.Условия.Количество() = 0 Тогда	// если условия накопительной не заданы - то конец
		Возврат Скидка;
	КонецЕсли;
	СуммаПродаж = ПолучитьСуммуПродаж(НаДату, ТипСкидки.ДнейНакопления, Контрагент); // получить сумму продаж контрагента за период
	ТЗ = ТипСкидки.Условия.Выгрузить();
	ТЗ.Сортировать("Сумма");
	СкидкаНакопительная = 0;
	Для Каждого Строка Из ТЗ Цикл // найти ,в какую строку условий попадает сумма продаж
		Если СуммаПродаж >= Строка.Сумма Тогда
			СкидкаНакопительная = Строка.Процент;
		КонецЕсли;
	КонецЦикла;
	Возврат Скидка + СкидкаНакопительная;

КонецФункции

// рассчитать суммму продаж по контрагенту за период
Функция ПолучитьСуммуПродаж(ДатаОкончания, ДнейСчитать, ПоКонтрагенту)

	Запрос = Новый Запрос;
	Запрос.Текст =
	"ВЫБРАТЬ
	|	ПродажиОбороты.Контрагент	КАК Контрагент,
	|	ПродажиОбороты.СуммаОборот
	|ИЗ
	|	РегистрНакопления.Продажи.Обороты(&Начало, &Окончание) КАК ПродажиОбороты
	| ГДЕ Контрагент = &ПоКонтрагенту
	| СГРУППИРОВАТЬ ПО
```

```
        | ПродажиОбороты.Контрагент,
        | ПродажиОбороты.СуммаОборот
        |";

        Запрос.УстановитьПараметр("ПоКонтрагенту", ПоКонтрагенту);
        Запрос.УстановитьПараметр("Начало",
НачалоДня(ДобавитьМесяц(ДатаОкончания,-(ДнейСчитать/30))) );
        Запрос.УстановитьПараметр("Окончание", КонецДня(ДатаОкончания));

        Результат = Запрос.Выполнить();
        Выборка = Результат.Выбрать();
        Пока Выборка.Следующий() Цикл
                Возврат Выборка.СуммаОборот;
        КонецЦикла;

        Возврат 0;

КонецФункции

// ПО НОМЕНКЛАТУРЕ
Функция         ПолучитьОсновнуюЕдиницуОтгрузкиНоменклатуры(НоменклатураССылка)
Экспорт

        Если Не НоменклатураССылка.Пустая()  Тогда
                Возврат НоменклатураССылка.ОсновнаяЕдиницаОтгрузки;
        КонецЕсли;
        Возврат Неопределено;

КонецФункции

Функция ПолучитьКоэффициент(ЕдИзмеренияССылка) Экспорт

        Если Не ЕдИзмеренияССылка.Пустая()  Тогда
                Возврат ЕдИзмеренияССылка.Коэффициент;
        КонецЕсли;
        Возврат 0;

КонецФункции

// получить цену номенклатуры
Функция ПолучитьЦену(НаДату, ВыбНоменклатура, ВыбТипЦен) Экспорт

        Запрос = Новый Запрос;
        Запрос.Текст =
         "ВЫБРАТЬ
         |        Цены.Номенклатура    КАК Номенклатура,
         |        Цены.ТипЦен                КАК ТипЦен,
         |        Цены.Цена                  КАК Цена
         |ИЗ
         |        РегистрСведений.Цены.СрезПоследних(&ВыбДата) КАК Цены
         |ГДЕ
         |        ТипЦен                = &ВыбТипЦен И
```

```
        |        Номенклатура = &ВыбНоменклатура
        |
        |";
        Запрос.УстановитьПараметр("ВыбНоменклатура", ВыбНоменклатура);
        Запрос.УстановитьПараметр("ВыбТипЦен", ВыбТипЦен);
        Запрос.УстановитьПараметр("ВыбДата", НаДату);

        Результат = Запрос.Выполнить();
    Выборка = Результат.Выбрать();
    Пока Выборка.Следующий() Цикл
            Возврат Выборка.Цена;
        КонецЦикла;

        Возврат 0;

КонецФункции

// заполнить остатки номенклатуры
// в ТЗНоменклатура передается список запрашиваемой номенклатуры
// в ней же заполняется столбец "Количество"
Процедура ПолучитьОстаткиНоменклатуры(НаДату, ВыбФирма, ВыбСклад,
ТЗНоменклатура) Экспорт

        СЗНоменклатура = Новый СписокЗначений;
        Для Каждого ТекНоменклатура Из ТЗНоменклатура Цикл
            СЗНоменклатура.Добавить(ТекНоменклатура.Номенклатура);
        КонецЦикла;

        Запрос = Новый Запрос;
        Запрос.Текст =
         "ВЫБРАТЬ
         |    Остатки.Номенклатура    КАК Номенклатура,
         |    Остатки.Фирма            КАК Фирма,
         |    Остатки.Склад            КАК Склад,
         |    Остатки.КоличествоОстаток        КАК Количество
         |ИЗ
         |    РегистрНакопления.Остатки.Остатки(&НаДату) КАК Остатки
         |ГДЕ
         |    Номенклатура В (&СЗНоменклатура) И
         |    Фирма = &ВыбФирма И
         |    Склад = &ВыбСклад
         |
         |";
        Запрос.УстановитьПараметр("СЗНоменклатура", СЗНоменклатура);
        Запрос.УстановитьПараметр("ВыбФирма", ВыбФирма);
        Запрос.УстановитьПараметр("ВыбСклад", ВыбСклад);
        Запрос.УстановитьПараметр("НаДату", НаДату);

        Результат = Запрос.Выполнить();
        Выборка = Результат.Выбрать();
        Пока Выборка.Следующий() Цикл
```

```
            СтрокаТЗ = ТЗНоменклатура.Найти(Выборка.Номенклатура,
"Номенклатура");
            Если  СтрокаТЗ <> Неопределено Тогда
                  СтрокаТЗ.Количество = Выборка.Количество;
            КонецЕсли;
      КонецЦикла;

КонецПроцедуры

Справочник «Контрагенты»

Процедура ПриУстановкеНовогоКода(СтандартнаяОбработка, Префикс)
      Префикс = Обмен.ПолучитьПрефиксНомера();
КонецПроцедуры


Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)
      // ввод на основании адреса доставки контрагента-основания
      Если ТипЗнч(ДанныеЗаполнения) = Тип("СправочникСсылка.Контрагенты") Тогда
            // это создается адрес доставки для головного контрагента-основания
            ЮрФизЛицо = Перечисления.ЮрФизЛицо.АдресДоставки;
            ГоловнойКонтрагент  = ДанныеЗаполнения.Ссылка;
            Родитель =
Константы.ПапкаАдресовДоставки.Получить();
            // некоторые реквизиты берем из головного контрагента
            Наименование = ДанныеЗаполнения.Наименование;
            НаименованиеПолное  = ДанныеЗаполнения.НаименованиеПолное;
            Категория = ДанныеЗаполнения.Категория;
            Ответственный = ДанныеЗаполнения.Ответственный;
            ЮрАдрес = ДанныеЗаполнения.ЮрАдрес;
            ФактАдрес = ДанныеЗаполнения.ФактАдрес;
            ИНН = ДанныеЗаполнения.ИНН;
            КПП = ДанныеЗаполнения.КПП;
            ОКПО = ДанныеЗаполнения.ОКПО;

            // ?? надо
            ОГРН = ДанныеЗаполнения.ОГРН;
            ОКВЭД =
ДанныеЗаполнения.ОКВЭД;
            СвидетельствоСерияНомер =
ДанныеЗаполнения.СвидетельствоСерияНомер;
            СвидетельствоДатаВыдачи =
ДанныеЗаполнения.СвидетельствоДатаВыдачи;
            СтранаРегистрации =  ДанныеЗаполнения.СтранаРегистрации;

            Фамилия = ДанныеЗаполнения.Фамилия;

            Имя = ДанныеЗаполнения.Имя;

            Отчество = ДанныеЗаполнения.Отчество;
```

```
            ДокументУдостоверяющийЛичность                                    =
ДанныеЗаполнения.ДокументУдостоверяющийЛичность;

            ДатаРождения                        =
ДанныеЗаполнения.ДатаРождения;
            МестоРождения                       =
ДанныеЗаполнения.МестоРождения;
            ПФР                                         = ДанныеЗаполнения.ПФР;

            Комментарий                         =
ДанныеЗаполнения.Комментарий;

                        // контакты берем из головного контрагента
            Для Каждого ТекСтрокаТЧКонтакты Из ДанныеЗаполнения.ТЧКонтакты
Цикл
                  НоваяСтрока = ТЧКонтакты.Добавить();
                  НоваяСтрока.Комментарий = ТекСтрокаТЧКонтакты.Комментарий;
                  НоваяСтрока.Контакт = ТекСтрокаТЧКонтакты.Контакт;
                  НоваяСтрока.Основной = ТекСтрокаТЧКонтакты.Основной;
                  НоваяСтрока.ТипКонтакта = ТекСтрокаТЧКонтакты.ТипКонтакта;
            КонецЦикла;
      КонецЕсли;
КонецПроцедуры
```

Документ «Заявка»

```
Функция ПриПроведении()

      Если Фирма.Пустая() Тогда
            Сообщить("Зполните организацию");
            Возврат Ложь;
      КонецЕсли;
      Если Контрагент.Пустая() Тогда
            Сообщить("Зполните контрагента");
            Возврат Ложь;
      КонецЕсли;

      Если Номенклатура.Количество() = 0 Тогда
            Сообщить("Зполните список товаров");
            Возврат Ложь;
      КонецЕсли;

      Возврат Истина;

КонецФункции

Процедура ОбработкаПроведения(Отказ, Режим)

      Если Не ПриПроведении() Тогда // проверить возможность проведения
            Отказ = Истина;
            Возврат;
      КонецЕсли;
```

```
КонецПроцедуры

Процедура ОбработкаЗаполнения(Основание, Значение)

	// Если вводится новый из интерфейса
	Если Основание = Неопределено Тогда
		Фирма		= Константы.ОсновнаяФирма.Получить();
		Контрагент	= Константы.ОсновнойКонтрагент.Получить();
		Сотрудник	= Контрагент.Ответственный;
	// ответсвенный сотрудник
		Договор		= ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент,
Дата, ТипЦен, ТипСкидки, Скидка); // основной договор, цены, скидки
	КонецЕсли;

	// Если вводится на основании документа Задание
	Если ТипЗнч(Основание) = Тип("ДокументСсылка.Задание") Тогда
		Фирма		= Основание.Фирма;
		Контрагент	= Основание.Контрагент;		// основание - это
контрагент
		Сотрудник	= Контрагент.Ответственный;	// из контрагента
		Договор		= ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент,
Дата, ТипЦен, ТипСкидки, Скидка); // основной договор, цены, скидки
	КонецЕсли;

	// Если вводится на основании Контрагента
	Если ТипЗнч(Основание) = Тип("СправочникСсылка.Контрагенты") Тогда
		Фирма		= Константы.ОсновнаяФирма.Получить();	// по
умолчанию
		Склад		= Константы.ОсновнойСклад.Получить();	// по умолчанию

		Контрагент	= Основание;				// основание - это
контрагент
		Сотрудник	= Контрагент.Ответственный;	// из контрагента
		Договор		= ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент,
Дата, ТипЦен, ТипСкидки, Скидка); // основной договор, цены, скидки
	КонецЕсли;

	Если ТипЗнч(Основание) = Тип("Структура") Тогда //если вводлится новый из
списка с установленным отбором
		// то данные заполняются автоматически из отбора "Заполнять из данных
заполнения"
	КонецЕсли

КонецПроцедуры

Процедура ПередЗаписью()
	СуммаДокумента = Номенклатура.Итог("Сумма");
КонецПроцедуры
```

```
// изменился контрагент - изменить договор, тип цен, тип скидки, скидку и пересчитать цены
&НаКлиенте
Процедура КонтрагентПриИзменении(Элемент)

	Объект.Сотрудник = ОбщийСрв.ПолучитьОтветсвенного(Объект.Контрагент);
	Объект.Договор = ОбщийСрв.ПолучитьОсновнойДоговор(Объект.Контрагент, Объект.Дата);
	ДоговорПриИзменении(Элемент);
КонецПроцедуры

// изменился договор - изменить тип цен, тип скидки, скидку и пересчитаь цену
&НаКлиенте
Процедура ДоговорПриИзменении(Элемент)

	Если Не Объект.Договор.Пустая() Тогда
		Объект.ТипЦен = ОбщийСрв.ПолучитьТипЦенИзДоговора(Объект.Договор);
		Объект.ТипСкидки = ОбщийСрв.ПолучитьТипСкидкиИзДоговора(Объект.Договор);
	Иначе
		Объект.ТипЦен = ОбщийСрв.ПолучитьОсновнойТипЦен();
		Объект.ТипСкидки = ОбщийСрв.ПолучитьОсновнойТипСкидки();
	КонецЕсли;
	Объект.Скидка = ОбщийСрв.СкидкаПоТипуСкидок(Объект.Дата, Объект.ТипСкидки, Объект.Контрагент);
	ПересчитатьЦены();

КонецПроцедуры

// изхменился тип цен - персчитать цены (скидка не изменилась)
&НаКлиенте
Процедура ТипЦенПриИзменении(Элемент)

	ПересчитатьЦены();
КонецПроцедуры

// изменился тип скидки - персчитать скидку и пересчитать цены
&НаКлиенте
Процедура ТипСкидкиПриИзменении(Элемент)

	Объект.Скидка = ОбщийСрв.СкидкаПоТипуСкидок(Объект.Дата, Объект.ТипСкидки, Объект.Контрагент);
	ПересчитатьЦены();
КонецПроцедуры

// изменилась скидка вручную - персчитать цены
&НаКлиенте
Процедура СкидкаПриИзменении(Элемент)

	ПересчитатьЦены();
КонецПроцедуры
```

```
Процедура ПересчитатьЦены()

	Для Каждого ТекНоменклатура Из Объект.Номенклатура Цикл
		ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,
ТекНоменклатура.Номенклатура,

	ТекНоменклатура.Количество,

	ТекНоменклатура.Коэфф,


	Объект.ТипЦен,

	Объект.Скидка,

	ТекНоменклатура.Цена,

	ТекНоменклатура.ЦенаСкидка,

	ТекНоменклатура.Сумма,

	ТекНоменклатура.СуммаСкидки);
	КонецЦикла;
КонецПроцедуры

&НаКлиенте
Процедура НоменклатураНоменклатураПриИзменении(Элемент)

	ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество   = 1;
	ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф       = 1;
	ОсновнаяЕдиницаОтгрузки                                    =
ОбщийСрв.ПолучитьОсновнуюЕдиницуОтгрузкиНоменклатуры(ЭтотОбъект.Элементы.Н
оменклатура.ТекущиеДанные.Номенклатура);
	Если Не ОсновнаяЕдиницаОтгрузки.Пустая() Тогда
		ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЕдИзмерения         =
ОсновнаяЕдиницаОтгрузки;
		ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф               =
ОбщийСрв.ПолучитьКоэффициент(ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные
.ЕдИзмерения);
	КонецЕсли;

	ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,

	ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,

	ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,


	Объект.ТипЦен,
```

```
        Объект.Скидка,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);

КонецПроцедуры

&НаКлиенте
Процедура НоменклатураКоличествоПриИзменении(Элемент)

        ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,


        Объект.ТипЦен,

        Объект.Скидка,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);

КонецПроцедуры

&НаКлиенте
Процедура НоменклатураЕдИзмеренияПриИзменении(Элемент)

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф                    =
ОбщийСрв.ПолучитьКоэффициент(ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные
.ЕдИзмерения);

        ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,
```

```
        Объект.ТипЦен,

        Объект.Скидка,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);

КонецПроцедуры
```

Документ «Реализация»

```
Функция ПриПроведении()

        Если Фирма.Пустая() Тогда
                Сообщить("Зполните организацию");
                Возврат Ложь;
        КонецЕсли;
        Если Контрагент.Пустая() Тогда
                Сообщить("Зполните контрагента");
                Возврат Ложь;
        КонецЕсли;
        Если Склад.Пустая() Тогда
                Сообщить("Зполните склад");
                Возврат Ложь;
        КонецЕсли;

        Если Номенклатура.Количество() = 0 Тогда
                Сообщить("Зполните список товаров");
                Возврат Ложь;
        КонецЕсли;

        Возврат Истина;

КонецФункции

Функция ПроверитьОстатки()

        // если в таб части несколько строк с одной и той же номенклатурой/партией, то "простая" проверка не
        // будет правильно проверять остатки. Необходимо свернуть одинаковую номенклатуру/партии

        ТЗОстатки = Новый ТаблицаЗначений;
        ТипНоменклатуура = ОбщийСрв.СоздатьОписаниеТипов("СправочникСсылка.Номенклатура");
        ТЗОстатки.Колонки.Добавить("Номенклатура", ТипНоменклатуура);
        ТипКоличество = ОбщийСрв.СоздатьОписаниеТипов("Число",15,2);
```

```
	ТЗОстатки.Колонки.Добавить("Количество", ТипКоличество);

	Для Каждого ТекСтрокаНоменклатура Из Номенклатура Цикл
		НоваяСтрока = ТЗОстатки.Добавить();
		НоваяСтрока.Номенклатура = ТекСтрокаНоменклатура.Номенклатура;
		НоваяСтрока.Количество	= 0;
	КонецЦикла;
	ТЗОстатки.Свернуть("Номенклатура", "Количество");
	ОбщийСрв.ПолучитьОстаткиНоменклатуры(Дата, Фирма, Склад, ТЗОстатки); // остстки запросом

	ТЗЗапрашиваемаяНоменклатура = Новый ТаблицаЗначений;
	ТЗЗапрашиваемаяНоменклатура.Колонки.Добавить("Номенклатура", ТипНоменклатуура);
	ТЗЗапрашиваемаяНоменклатура.Колонки.Добавить("Количество", ТипКоличество);
	Для Каждого ТекСтрокаНоменклатура Из Номенклатура Цикл
		НоваяСтрока = ТЗЗапрашиваемаяНоменклатура.Добавить();
		НоваяСтрока.Номенклатура = ТекСтрокаНоменклатура.Номенклатура;
		НоваяСтрока.Количество = ТекСтрокаНоменклатура.Количество*ТекСтрокаНоменклатура.Коэфф;
	КонецЦикла;
	ТЗЗапрашиваемаяНоменклатура.Свернуть("Номенклатура", "Количество");

	Для Каждого ТекЗапрашиваемыйОстаток Из ТЗЗапрашиваемаяНоменклатура Цикл
		Остаток = ТЗОстатки.Найти(ТекЗапрашиваемыйОстаток.Номенклатура, "Номенклатура");
		Если Остаток = Неопределено Тогда
			Сообщить("По номенклатуре " + ТекЗапрашиваемыйОстаток.Номенклатура + " запрашиваемое количество меньше имеющегося остатка");
			Возврат Ложь;
		КонецЕсли;
		Если Остаток.Количество < ТекЗапрашиваемыйОстаток.Количество Тогда
			Сообщить("По номенклатуре " + ТекЗапрашиваемыйОстаток.Номенклатура + " запрашиваемое количество меньше имеющегося остатка");
			Возврат Ложь;
		КонецЕсли;
	КонецЦикла;

	Возврат Истина
КонецФункции

Процедура ОбработкаПроведения(Отказ, Режим)

	Если Не ПриПроведении() Тогда // проверить возможность проведения
		Отказ = Истина;
		Возврат;
	КонецЕсли;

	Если Склад.Контроль Тогда
```

```
            Если Не ПроверитьОстатки() Тогда // контроль остатков
                Отказ = Истина;
                Возврат;
            КонецЕсли;
        КонецЕсли;

        // регистр остатков Остатки
        Для Каждого ТекСтрокаНоменклатура Из Номенклатура Цикл
            Движение                    = Движения.Остатки.Добавить();
            Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
            Движение.Период     = Дата;
            Движение.Фирма              = Фирма;
            Движение.Склад              = Склад;
            Движение.КодОперации = КодОперации;
            Движение.Номенклатура = ТекСтрокаНоменклатура.Номенклатура;
            Движение.Партия              = ТекСтрокаНоменклатура.Партия;
            Движение.Количество                                                    =
ТекСтрокаНоменклатура.Количество*ТекСтрокаНоменклатура.Коэфф;
        КонецЦикла;

        // регистр остатков Взаиморасчеты
        Если Счет.Пустая() Тогда // если в документе указан счет-касса
            Движение = Движения.Взаиморасчеты.Добавить();
            Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
            Движение.Период          = Дата;
            Движение.Фирма                  = Фирма;
            Движение.Контрагент      = Контрагент;
            Движение.Сделка          = ЭтотОбъект.Ссылка;
            Движение.Сумма                  = Номенклатура.Итог("Сумма");
        КонецЕсли;

        // регистр остатков Деньги
        Если Не Счет.Пустая() Тогда // если в документе указан счет-касса
            Движение = Движения.Деньги.Добавить();
            Движение.ВидДвижения    = ВидДвиженияНакопления.Приход;
            Движение.Период          = Дата;
            Движение.Фирма                  = Фирма;
            Движение.Счет                   = Счет;
            Движение.КодОперации    = КодОперации;
            Движение.Сумма                  = Номенклатура.Итог("Сумма");
        КонецЕсли;


        // регистр оборотов Продажи
        Для Каждого ТекСтрокаНоменклатура Из Номенклатура Цикл
            Движение = Движения.Продажи.Добавить();
            Движение.Период                 = Дата;
            Движение.Фирма                          = Фирма;
            Движение.Контрагент             = Контрагент;
            Движение.Сотрудник                      = Сотрудник;
            Движение.КодОперации            = КодОперации;
```

```
		Движение.Номенклатура =
ТекСтрокаНоменклатура.Номенклатура;
		Движение.Количество =
ТекСтрокаНоменклатура.Количество*ТекСтрокаНоменклатура.Коэфф;
		Движение.Сумма = ТекСтрокаНоменклатура.Сумма;

	КонецЦикла;

КонецПроцедуры

Процедура ОбработкаЗаполнения(Основание, Значение)

	// Если вводится новый из интерфейса
	Если Основание = Неопределено Тогда
		Фирма = Константы.ОсновнаяФирма.Получить();
		Склад = Константы.ОсновнойСклад.Получить();
		Контрагент = Константы.ОсновнойКонтрагент.Получить();
		Сотрудник = Контрагент.Отвественный;
	// ответсвенный сотрудник
		Договор = ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент,
Дата, ТипЦен, ТипСкидки, Скидка); // основной договор, цены, скидки
		Счет = Константы.ОсновнойСчет.Получить();
	КонецЕсли;

	// Если вводится на основании Заявки
	Если ТипЗнч(Основание) = Тип("ДокументСсылка.Заявка") Тогда
		// Заполнение шапки
		ДокОснование = Основание; // документ-
основание
		Фирма = Основание.Фирма; // из основания
		Склад = Константы.ОсновнойСклад.Получить();
		Контрагент = Основание.Контрагент; // из основания
		Сотрудник = Контрагент.Отвественный; // из контрагента
		Договор = ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент,
Дата, ТипЦен, ТипСкидки, Скидка); // основной договор, цены, скидки
		Счет = Константы.ОсновнойСчет.Получить();
		// табличная часть
		Для Каждого ТекСтрокаНоменклатура Из Основание.Номенклатура Цикл //
по всем записям табличной части
			НоваяСтрока = Номенклатура.Добавить();
			НоваяСтрока.Номенклатура =
ТекСтрокаНоменклатура.Номенклатура;
			НоваяСтрока.Количество = ТекСтрокаНоменклатура.Количество;
			НоваяСтрока.ЕдИзмерения = ТекСтрокаНоменклатура.ЕдИзмерения;
			НоваяСтрока.Коэфф = ТекСтрокаНоменклатура.Коэфф;


			ОбщийСрв.РассчитатьСуммуСтроки(ТекущаяДата(),
ТекСтрокаНоменклатура.Номенклатура,
					ТекСтрокаНоменклатура.Количество,

	ТекСтрокаНоменклатура.Коэфф,
```

```
            ТипЦен,

            Скидка,

            НоваяСтрока.Цена,

            НоваяСтрока.ЦенаСкидка,

            НоваяСтрока.Сумма,

            НоваяСтрока.СуммаСкидки);
                КонецЦикла;
        КонецЕсли;

        // Если вводится на основании Контрагента
        Если ТипЗнч(Основание) = Тип("СправочникСсылка.Контрагенты") Тогда
                Фирма           = Константы.ОсновнаяФирма.Получить();       // по
умолчанию
                Склад           = Константы.ОсновнойСклад.Получить();   // по умолчанию

                Контрагент  = Основание;                                    // основание - это
контрагент
                Сотрудник   = Контрагент.Отвественный;      // из контрагента
                Договор         =   ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент,
Дата, ТипЦен, ТипСкидки, Скидка); // основной договор, цены, скидки
                Счет            = Константы.ОсновнойСчет.Получить();
        КонецЕсли;

        //Если вводится новый из списка с установленным отбором
        Если ТипЗнч(Основание) = Тип("Структура") Тогда
                // то данные заполняются автоматически из отбора "Заполнять из данных
заполнения"
        КонецЕсли

КонецПроцедуры

Процедура ПередЗаписью()
        СуммаДокумента = Номенклатура.Итог("Сумма");
КонецПроцедуры

// изменился контрагент - изменить договор, тип цен, тип скидки, скидку и пересчитать
цены
&НаКлиенте
Процедура КонтрагентПриИзменении(Элемент)

        Объект.Сотрудник = ОбщийСрв.ПолучитьОтветсвенного(Объект.Контрагент);
        Объект.Договор  =   ОбщийСрв.ПолучитьОсновнойДоговор(Объект.Контрагент,
Объект.Дата);
        //Если Не Объект.Договор = Неопределено Тогда
                ДоговорПриИзменении(Элемент);
        //КонецЕсли;
```

```
КонецПроцедуры

// изменился договор - изменить тип цен, тип скидки, скидку и пересчитаь цену
&НаКлиенте
Процедура ДоговорПриИзменении(Элемент)

	Если Не Объект.Договор.Пустая() Тогда
		Объект.ТипЦен = ОбщийСрв.ПолучитьТипЦенИзДоговора(Объект.Договор);
		Объект.ТипСкидки = ОбщийСрв.ПолучитьТипСкидкиИзДоговора(Объект.Договор);
	Иначе
		Объект.ТипЦен = ОбщийСрв.ПолучитьОсновнойТипЦен();
		Объект.ТипСкидки = ОбщийСрв.ПолучитьОсновнойТипСкидки();
	КонецЕсли;
	Объект.Скидка = ОбщийСрв.СкидкаПоТипуСкидок(Объект.Дата, Объект.ТипСкидки, Объект.Контрагент);
	ПересчитатьЦены();

КонецПроцедуры

// изхменился тип цен - персчитать цены (скидка не изменилась)
&НаКлиенте
Процедура ТипЦенПриИзменении(Элемент)

	ПересчитатьЦены();
КонецПроцедуры

// изменился тип скидки - персчитать скидку и пересчитать цены
&НаКлиенте
Процедура ТипСкидкиПриИзменении(Элемент)

	Объект.Скидка = ОбщийСрв.СкидкаПоТипуСкидок(Объект.Дата, Объект.ТипСкидки, Объект.Контрагент);
	ПересчитатьЦены();
КонецПроцедуры

// изменилась скидка вручную - персчитать цены
&НаКлиенте
Процедура СкидкаПриИзменении(Элемент)

	ПересчитатьЦены();
КонецПроцедуры

Процедура ПересчитатьЦены()

	Для Каждого ТекНоменклатура Из Объект.Номенклатура Цикл
		ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата, ТекНоменклатура.Номенклатура,

	ТекНоменклатура.Количество,
```

```
        ТекНоменклатура.Коэфф,


        Объект.ТипЦен,

        Объект.Скидка,

        ТекНоменклатура.Цена,

        ТекНоменклатура.ЦенаСкидка,

        ТекНоменклатура.Сумма,

        ТекНоменклатура.СуммаСкидки);
        КонецЦикла;
КонецПроцедуры

&НаКлиенте
Процедура НоменклатураНоменклатураПриИзменении(Элемент)

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество   = 1;
        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф        = 1;
        ОсновнаяЕдиницаОтгрузки                                     =
ОбщийСрв.ПолучитьОсновнуюЕдиницуОтгрузкиНоменклатуры(ЭтотОбъект.Элементы.Н
оменклатура.ТекущиеДанные.Номенклатура);
        Если Не ОсновнаяЕдиницаОтгрузки.Пустая() Тогда
                ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЕдИзмерения          =
ОсновнаяЕдиницаОтгрузки;
                ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф                =
ОбщийСрв.ПолучитьКоэффициент(ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные
.ЕдИзмерения);
        КонецЕсли;

        ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,


        Объект.ТипЦен,

        Объект.Скидка,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,
```

```
        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);

КонецПроцедуры

&НаКлиенте
Процедура НоменклатураКоличествоПриИзменении(Элемент)

        ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,


        Объект.ТипЦен,

        Объект.Скидка,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);

КонецПроцедуры

&НаКлиенте
Процедура НоменклатураЕдИзмеренияПриИзменении(Элемент)

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф                =
ОбщийСрв.ПолучитьКоэффициент(ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные
.ЕдИзмерения);

        ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,


        Объект.ТипЦен,

        Объект.Скидка,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,

        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,
```

```
        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,
        ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);
КонецПроцедуры
```