


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КубГУ»)

Кафедра прикладной математики

КУРСОВАЯ РАБОТА


РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ  
ПРОИЗВОДСТВЕННОГО ПРЕДПРИЯТИЯ

Работу выполнил  \_\_\_\_\_ Н.Ю. Колесниченко  
(подпись, дата) (инициалы, фамилия)

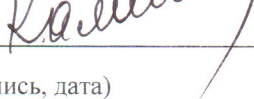
Факультет компьютерных технологий и прикладной математики курс 3

Направление 09.03.03 Прикладная информатика

Научный руководитель,

Д-р физ.-мат. наук  \_\_\_\_\_ М.Х Уртенов  
(подпись, дата) (инициалы, фамилия)

Нормоконтролер,

К. физ.-мат. наук  \_\_\_\_\_ Г.В. Калайдина  
(подпись, дата) (инициалы, фамилия)

Краснодар 2018

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	3
1 Прикладные решения 1С .....	4
1.1 Вступление .....	4
1.2 Виды конфигураций .....	4
1.3 Типовые конфигурации 1С.....	6
1.4 Особенности разработки прикладных решений.....	9
2 Создание конфигурации.....	13
ЗАКЛЮЧЕНИЕ .....	58

## ВВЕДЕНИЕ

На данный момент 1С:Предприятие, а в точности 1С:Бухгалтерия является актуальным программным продуктом. В связи с развитием рыночной экономики, необходимо совершенствовать систему учета на предприятии. С помощью данного программного продукта можно оптимизировать ведение бухгалтерского учета на предприятии, подготавливать отчетность в электронном виде и выполнять другие процессы. Это экономит время и средства.

Целью курсовой работы является разработка автоматизированной системы производственного предприятия, а именно разработка конфигурации для автоматизации работы и учета на производственном предприятии.

Основные задачи курсовой работы – изучить некоторые аспекты разработки конфигураций в 1С и создание собственной конфигурации.

Курсовая работа состоит из двух глав, введения, заключения и списка использованной информации.

В первой главе подробно описываются основные типы конфигураций и их применение и особенности разработки.

Во второй главе описывается поэтапное создание собственной конфигурации и настройка информационной базы на основе этой конфигурации.

В заключении приведены основные источники курсовой работы.

## 1 Прикладные решения 1С

### 1.1 Вступление

Прикладное решение – комплекс прикладных программ, обеспечивающий автоматизацию задач конкретной предметной области. Включает в себя платформу исполнения (Платформа 1с, например, платформа 8.1) и конфигурацию (написанную средствами внутреннего языка 1с), обеспечивающую бизнес-логику прикладного решения.

Конфигурация — это набор форм и алгоритмов, написанных на платформе 1С, для ведения учета хозяйственной деятельности предприятия.

Конфигураций 1С на Российском рынке – очень много. Большинство из них разработаны различными компаниями франчайзи 1С. Типовые конфигурации 1С – это универсальные конфигурации 1С, разработанные компанией 1С. Наиболее распространенных типовых конфигураций 1С – сравнительно немного – менее десяти. Дополнительно они делятся на версии – базовая/профессиональная/корпоративная.

Типовые конфигурации 1С разрабатывает сама фирма 1С. Франчайзи разрабатывает собственные тиражные конфигурации 1С, которые могут также продаваться через сеть 1С, но они не являются «типовыми». Каждая компания может разработать собственную оригинальную конфигурацию 1С «с нуля» и распространять ее самостоятельно. Если разработанная конфигурация 1С содержит часть типовой конфигурации 1С или является модификацией типовой, то для распространения такой конфигурации 1С требуется получить разрешение компании 1С. Для получения такого разрешения необходимо пройти сертификацию «1С:Совместимо».

### 1.2 Виды конфигураций

Базовые (специализированные) конфигурации предназначены для работы на базовых версиях платформы (для версии 7.x – с соответствующей «компонентой»). Базовые конфигурации поставляются только фирмой 1С, не

допускают внесения изменений, но могут быть преобразованы в типовые. Все неспециализированные, «небазовые» конфигурации, в том числе и типовые от 1С, допускают изменение (конфигурирование) пользователем. Тиражные решения сторонних фирм часто бывают защищены «системой защиты конфигураций» (СЗК), предлагаемой 1С или аналогичной, и предусматривают наличие дополнительного ключа, распространяемого с конфигурацией. Без этого ключа программа работает в ограниченном или демонстрационном режиме (как рекомендует 1С) или работа вовсе невозможна (даже при наличии ключа). Указанные ключи СЗК не заменяют ключей к технологической платформе, те продолжают оставаться необходимыми.

Конфигурации 7.x могут использовать базовые объекты и объекты других «Стандартных компонент». Для использования объекта какой-либо компоненты необходимо наличие соответствующей компоненты в Технологической платформе. Если конфигурация написана с использованием только базовых объектов («компонент-независимая»), то для её использования необходима Технологическая платформа 7.x с любой хотя бы одной «компонентой», не менее чем «Бухгалтерский учёт (стандартная)».

Следует различать понятия «базовые объекты» и «базовые конфигурации»: так, неспециализированная конфигурация, использующая только базовые объекты, не будет работать на базовой платформе, а базовые конфигурации не ограничиваются использованием базовых объектов.

Конфигурации часто делятся на две группы – «типовые» и «нетиповые». «Типовая» конфигурация – это тиражное решение, которое распространяется «в коробке». Также встречается сленговый термин «Правленая» конфигурация – имеется в виду изменённая под нужды компании типовая конфигурация, которая в силу внесённых в неё изменений уже не является типовой, однако очень похожа. «Самописная» или конфигурация «с нуля» – конфигурация, разработанная независимо от фирмы 1С, обычно под нужды конкретного заказчика.

### 1.3 Типовые конфигурации 1С

Типовые прикладные решения фирмы «1С» предназначены для автоматизации типовых задач учета и управления предприятий. При разработке типовых прикладных решений учитывались как современные международные методики управления, так и реальные потребности предприятий, не укладывающиеся в стандартный набор функциональности этих методик.

Состав функциональности, включаемой в типовые решения, тщательно проработан. Фирма «1С» анализирует опыт пользователей, применяющих программы системы «1С:Предприятие» и отслеживает изменение их потребностей.

Для использования на российских предприятиях фирма «1С» предлагает следующие прикладные решения:

1) «1С:Бухгалтерия 8» (включая версию КОРП, «1С:Упрощенка 8» и «1С:Предприниматель 8»)

«1С:Бухгалтерия 8» предназначена для автоматизации бухгалтерского и налогового учета, включая подготовку регламентированной отчетности в организациях, осуществляющих любые виды коммерческой деятельности: оптовую и розничную торговлю, комиссионную торговлю, оказание услуг, производство и т. д. Бухгалтерский и налоговый учет ведется в соответствии с действующим законодательством Российской Федерации.

2) «1С:Управление небольшой фирмой 8»

«1С:Управление небольшой фирмой 8» (УНФ) - это готовое решение для автоматизации оперативного управления на предприятиях малого бизнеса. В программе реализовано все самое необходимое для ведения оперативного учета, контроля, анализа и планирования на предприятии. Решение не перегружено излишним функционалом, его можно легко настроить под особенности организации управления и учета в компании - это обеспечивает возможность «быстрого старта» и удобство ежедневной работы. УНФ помогает повысить эффективность работы компании, предоставляя

владельцам и руководителям широкий спектр инструментов для управления, а сотрудникам – новые возможности для продуктивной ежедневной работы.

3) «Управление торговлей»

«1С:Управление торговлей 8» - это современный инструмент для повышения эффективности бизнеса торгового предприятия. «1С:Управление торговлей 8» позволяет в комплексе автоматизировать задачи оперативного и управленческого учета, анализа и планирования торговых операций, обеспечивая тем самым эффективное управление современным торговым предприятием.

4) «1С:Зарплата и Управление Персоналом 8»

В «1С:Зарплата и управление персоналом 8» поддерживаются все основные процессы управления персоналом, а также процессы кадрового учета, расчета зарплаты, исчисления налогов, формирования отчетов и справок в государственные органы и социальные фонды, планирование расходов на оплату труда. Учтены требования законодательства, реальная практика работы предприятий и перспективные мировые тенденции развития подходов к управлению персоналом.

5) «1С:Комплексная автоматизация 8»

В единой информационной базе ведется управленческий, бухгалтерский и налоговый учет как одной, так и нескольких организаций. Это существенно снижает трудоемкость ведения учета за счет использования общих массивов информации. Управленческий учет по предприятию может вестись в любой выбранной валюте, бухгалтерский и налоговый учет ведется в национальной валюте. Регламентированная отчетность для каждой организации формируется отдельно.

6) «1С:Управление производственным предприятием»

«1С:Управление производственным предприятием» является комплексным прикладным решением, охватывающим основные контуры управления и учета на производственном предприятии. Решение позволяет организовать комплексную информационную систему, соответствующую

корпоративным, российским и международным стандартам и обеспечивающую финансово-хозяйственную деятельность предприятия. Прикладное решение создает единое информационное пространство для отображения финансово-хозяйственной деятельности предприятия, охватывая основные бизнес-процессы. В то же время четко разграничивается доступ к хранимым сведениям, а также возможности тех или иных действий в зависимости от статуса работников. На предприятиях холдинговой структуры общая информационная база может охватывать все организации, входящие в холдинг. Это существенно снижает трудоемкость ведения учета за счет повторного использования разными организациями общих массивов информации.

7) «1С:Платежные документы 8»

Программа предназначена для подготовки, печати и хранения основных бухгалтерских документов:

- а. платежные поручения;
- б. платежные требования;
- в. счета;
- г. акты об оказании услуг;
- д. товарная накладная (ТОРГ-12);
- е. товарно-транспортная накладная (1-Т);
- ж. счета-фактуры;
- з. приходные и расходные кассовые ордера;
- и. авансовые отчеты;
- к. доверенности.

8) «1С:Налогоплательщик 8»

Программа предназначена для подготовки и представления отчетности в государственные органы.

9) «1С:Розница 8»



Прикладное решение предназначено для автоматизации торговой деятельности магазинов и других розничных торговых точек, в том числе объединенных в торговую сеть.

10) «1С:Документооборот 8»

«1С:Документооборот 8» позволяет:

- а. упорядочить работу сотрудников с документами, исключить возможность утери версий или пересечения фрагментов при одновременной работе;
- б. сократить время поиска нужной информации и суммарное время коллективной обработки документов;
- в. повысить качество готового материала (проектов, документации и пр.) за счет решения большого количества спорных вопросов и упорядочивания работы пользователей.
- г. Программа не имеет отраслевой специфики и может эффективно использоваться как в бюджетном секторе, так и на коммерческих предприятиях.

#### 1.4 Особенности разработки прикладных решений

«1С:Предприятие» как предметно-ориентированная среда разработки имеет определенные преимущества. Поскольку круг задач более точно очерчен, то и набор средств и технологий можно подобрать с большей определенностью. В задачу платформы входит предоставление разработчику интегрированного набора инструментов, необходимых для быстрой разработки, распространения и поддержки прикладного решения для автоматизации бизнеса. При этом отдельные «детали» могут уступать по функциональности универсальным средствам разработки и специализированным средствам управления жизненным циклом, используемым разработчиками. Однако эффект достигается благодаря общему набору средств и их тесной интеграции.

Платформа «1С:Предприятие» содержит такие инструменты для выполнения поставленных задач, как визуальное описание структур данных, написание программного кода, визуальное описание запросов, визуальное описание интерфейса, описание отчетов, отладка программного кода, профилирование. В ее составе: развитая справочная система, механизм ролевой настройки прав, инструменты создания дистрибутивов, удаленного обновления приложений, сравнения и объединения приложений, ведения журналов и диагностики работы приложения, создания Web-приложений и приложений для КПК, а также поддержка коллективной разработки, версионирования и пр.

Разработка в «1С:Предприятии» строится на основе общей модели работы приложения, предлагаемой платформой «в обязательном порядке», т. е. основные и наиболее сложные архитектурно-технологические решения (такие, как механизм трехуровневой архитектуры, вопросы взаимодействия компонентов, аутентификация пользователей и т. д.) предлагаются разработчикам в готовом виде.

В «1С:Предприятии» процесс написания программного кода – не центральный элемент разработки ПО. Приложение разрабатывается прежде всего, как структура метаданных. Код пишется в определенных узлах приложения «по необходимости», чтобы переопределить, если это нужно, стандартное поведение или написать ту часть бизнес-логики, которая требует именно алгоритмической формулировки, например расчет себестоимости. То есть имеется framework, задающий общий облик приложения, а приложение определяется как набор различных артефактов, которые функционируют в данном framework-е. Идея разработки на основе метаданных (metadata-driven) начинает активно использоваться и в универсальных системах, но в предметно-ориентированной среде разработки она дает существенно больший эффект, так как структура метаданных четко ориентирована на круг решаемых системой задач.

Один из моментов, обычно вызывающих споры – принятое в «1С:Предприятии» построение основной части приложения на основе стандартных прототипов (patterns) прикладных объектов. Действительно, эта модель отличается от классического подхода (объектно-ориентированного программирования и работы с таблицами базы данных или отображаемыми в базу данных произвольными сущностями). Фактически система предоставляет не один базовый класс для построения прикладных объектов приложения, а несколько, каждый из которых имеет специализированную функциональность и предназначен для отображения в приложении объектов предметной области, обладающих схожими свойствами и ролью в бизнес-логике. Разработчик использует эти прототипы для создания объектов приложения, которые уже являются финальными (описывающими конкретные бизнес-сущности).

Прототипы применяются с некоторой параметризацией, определяющей необходимые в конкретном случае свойства и особенности поведения. Например, справочник может быть «плоским» или иерархическим. Такой подход фактически обеспечивает построение приложения на основе определенной прикладной модели, в которой каждый объект играет определенную роль, и система хорошо знает эту роль, что позволяет ей автоматически выполнять существенную часть операций. Выше мы говорили о предоставляемой технологической модели приложения. Заметим, что эти две модели (технологическая и прикладная), с одной стороны, идеологически связаны, но с другой – имеют достаточно высокий уровень независимости в части реализации. Такой подход позволяет использовать в платформе новые архитектурные и технологические решения без смены прикладной модели и без изменения приложения.

Еще одна особенность «1С:Предприятия» как предметно-ориентированной среды разработки – особое отношение к подбору технологических возможностей, предоставляемых разработчику. Разумеется, в «1С:Предприятии» есть возможность подключать другие (внешние) программные модули. Но платформа ориентирована на то, чтобы актуальные

для задач автоматизации бизнеса технологии предоставить разработчику в готовом виде. Причем высокая степень «готовности» включает и простоту освоения, и «гладкость» интеграции с общей функциональностью и другими технологическими возможностями системы. Фактически платформа позволяет разработчику прикладных решений задействовать необходимые и современные технологии своевременно, максимально просто и без радикальных изменений в своем приложении.

Примеров этому можно привести много. Одно из сильных конкурентных преимуществ платформы «1С:Предприятия» (по общей оценке разработчиков приложений) – система обмена данными, позволяющая достаточно просто реализовать синхронизацию данных на основе XML как для создания территориально распределенных приложений, так и для целей интеграции (с другими приложениями «1С:Предприятия» и с другими системами).

Построение системы на основе технологической модели работы приложения, метаданных и прикладной модели работы приложения позволяет существенно упростить и ускорить разработку.

Во-первых, опираясь на метаданные и единую модель, все компоненты системы взаимодействуют между собой без существенных усилий со стороны разработчика. Фактически они знают, «что нужно делать» с теми или иными объектами бизнес-логики и типами данных.

Во-вторых, существенная часть технологических сложностей реализации решается платформой и не «наваливается» на разработчика решения

## 2 Создание конфигурации

Чтобы создать новую конфигурацию для начала следует создать информационную базу без конфигурации, процесс создания базы без конфигурации представлен на рисунках 1,2.

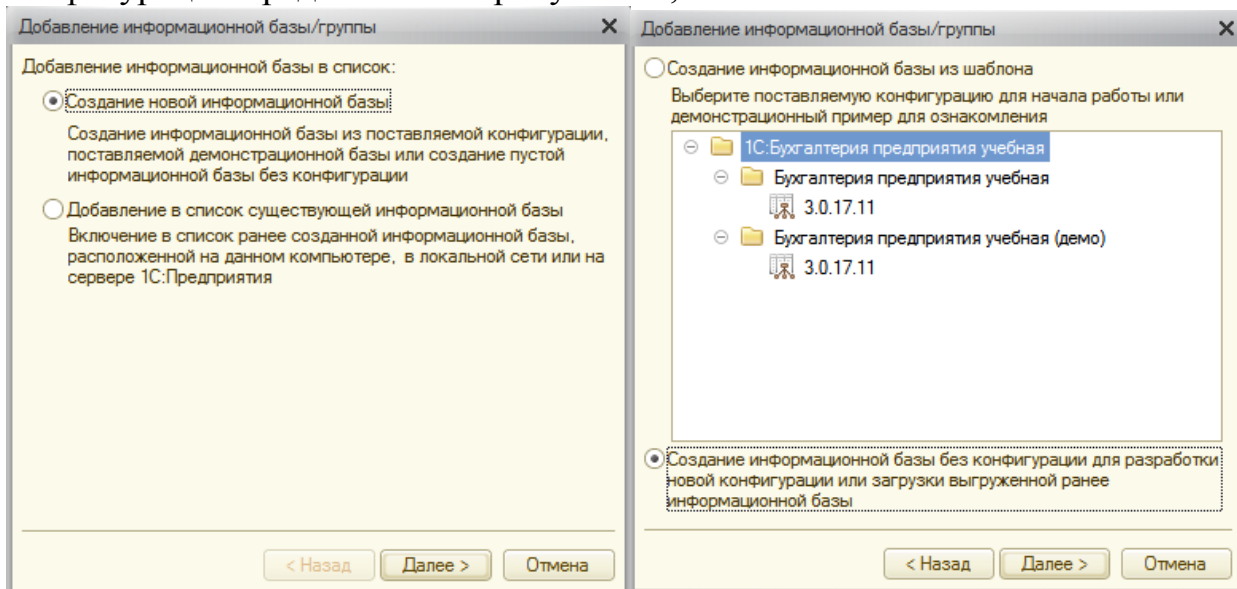


Рисунок 1,2 – Создание новой базы

После создания новой базы, мы можем открыть конфигуратор, внешний вид окна конфигуратора можно увидеть на рисунке 3.

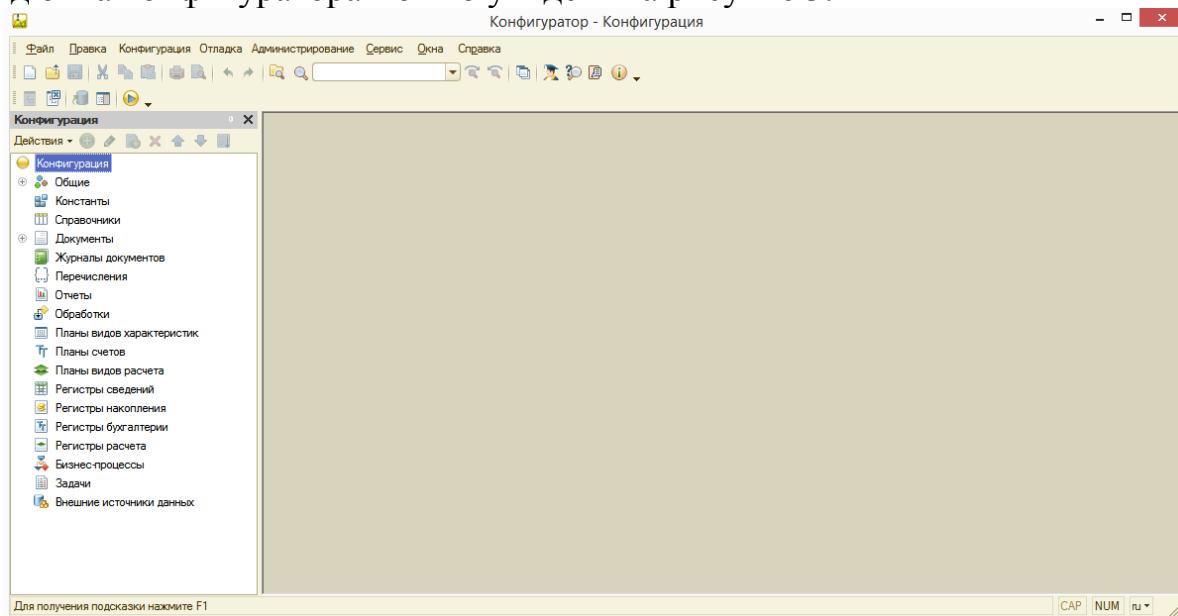


Рисунок 3 – Окно конфигуратора.

Далее откроем палитру свойств конфигурации, зададим имя конфигурации «ПособиеДляНачинающих» и «открепим» палитру свойств. Настройки палитры свойств представлены на рисунке 4.

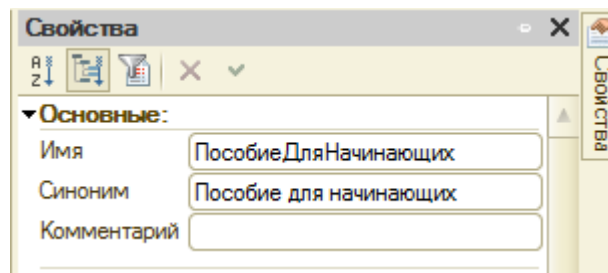


Рисунок 4 – Палитра свойств

Запустим отладку в режиме 1С:Предприятие. Запуск отладки можно увидеть на рисунке 5.

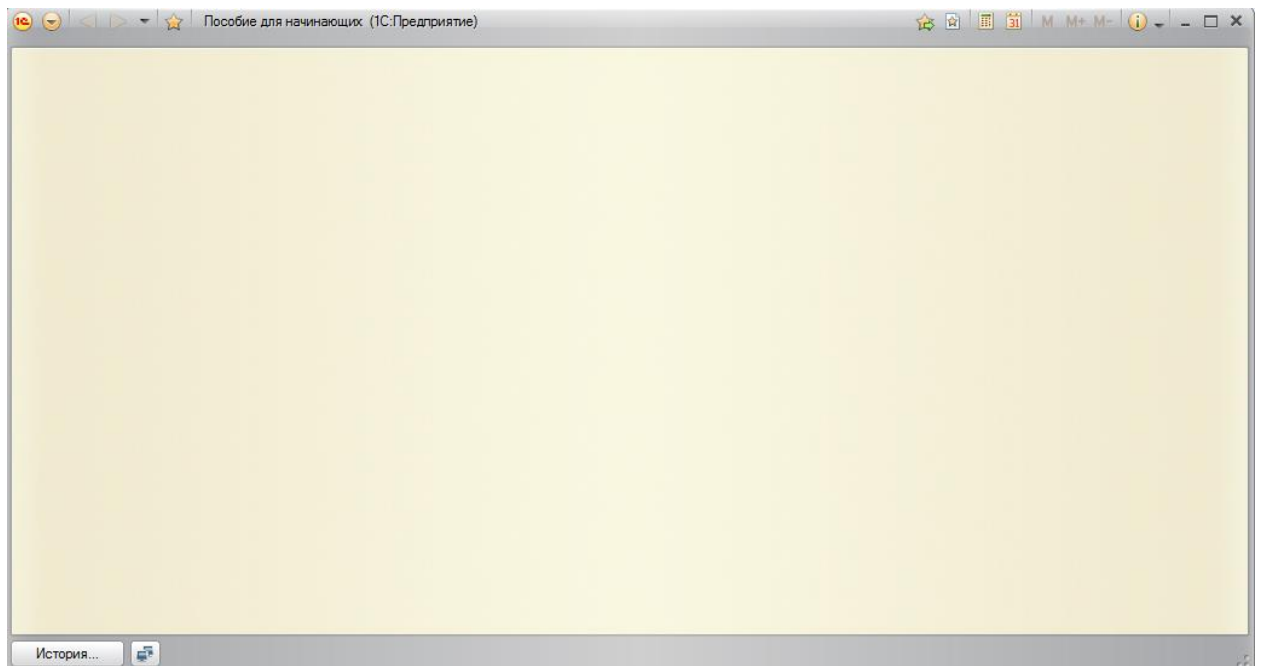


Рисунок 5 – Отладка в режиме 1С:Предприятие

Добавим в режиме Конфигуратор подсистему Бухгалтерия. Добавление подсистемы представлено на рисунке 6.

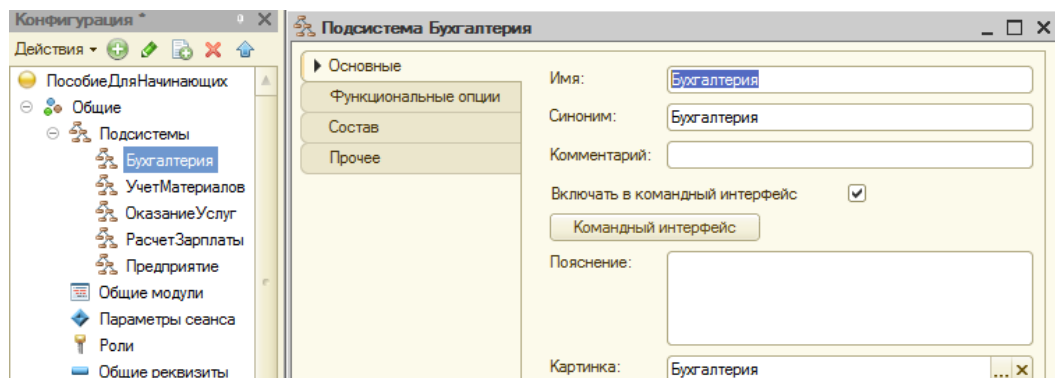


Рисунок 6 – Добавление подсистемы Бухгалтерия

Таким же образом добавим подсистемы УчетМатериалов, ОказаниеУслуг, РасчетЗарплаты и Предприятие. Добавление подсистем представлено на рисунке 7.

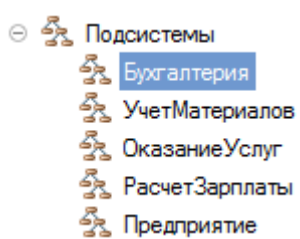


Рисунок 7 – Добавленные подсистемы

Откроем конфигурацию в режиме отладки. Конфигурация представлена на рисунке 8.

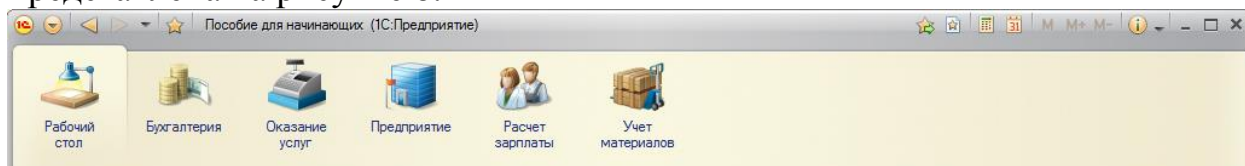


Рисунок 8 – Конфигурация с новыми подсистемами

Изменим порядок расположения разделов. Изменение порядка представлено на рисунке 9.



Рисунок 9 – Измененный порядок

Снова запустим конфигурацию в режиме отладки. Измененная конфигурация представлена на рисунке 10.

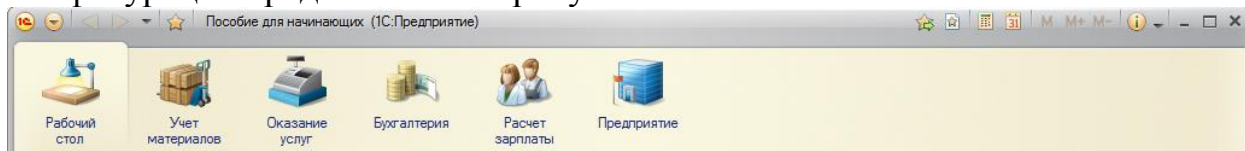


Рисунок 10 –Измененная конфигурация

Создадим новый объект конфигурации – справочник «Клиенты».

Укажем основные свойства справочника для правильного представления его в конфигурации. Заданные свойства представлены на рисунке 11.

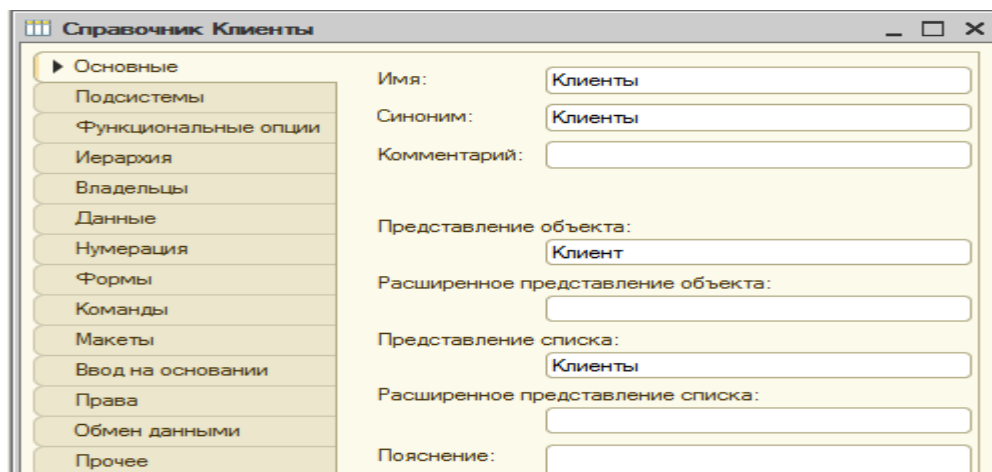


Рисунок 11 – Основные свойства объекта

Определим в каких подсистемах будет отображаться данный справочник. Задание принадлежности объекта к подсистемам представлено на рисунке 12.

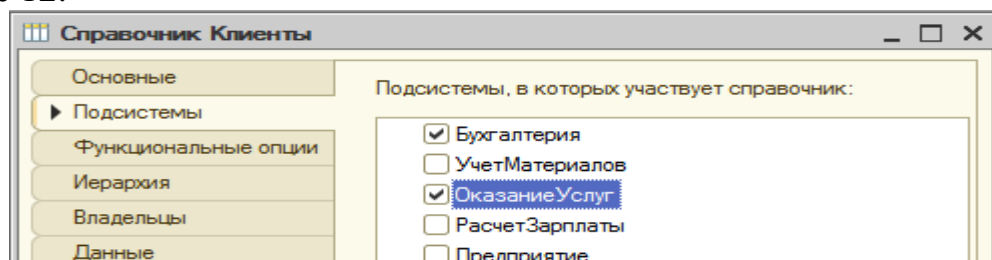


Рисунок 12 – Представление объекта в подсистемах

Укажем длину кода и наименования справочника. Заданные числа представлены на рисунке 13.

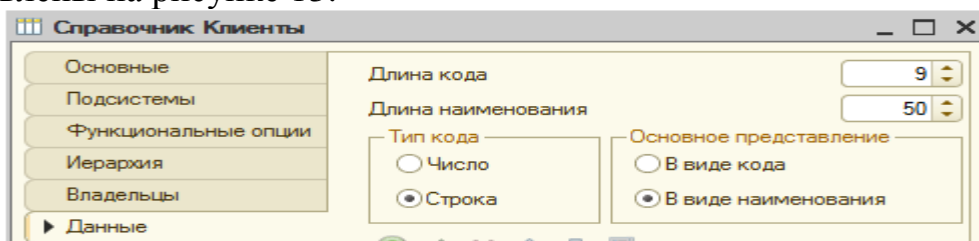


Рисунок 13 – Длина кода и наименования справочника

Сделаем доступной в панели действия ОказаниеУслуг стандартную команду для создания новых клиентов. Включение доступности команды создания новых клиентов представлено на рисунке 14.

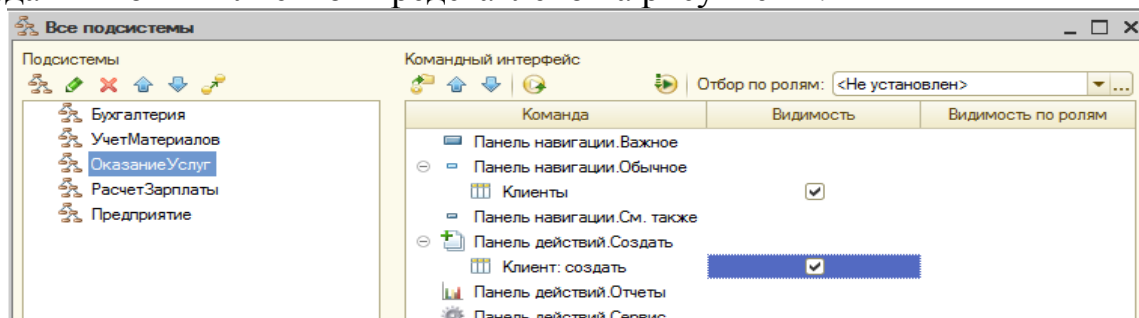


Рисунок 14 – Включение доступности команды создания новых клиентов



В режиме «1С:Предприятие» внесём в список «Клиенты» новые наименования клиентов. Внесение новых данных представлено на рисунке 15.

Наименование	Код
Иванов Михаил Юрьевич	000000001
Роман	000000002
Спиридонова Галина	000000003

Рисунок 15 – Новые данные в справочнике «Клиенты»

Создадим новый справочник «Сотрудники». Заданные параметры представлены на рисунках 16,17,18.

Рисунок 16 – Задание параметров нового справочника

Рисунок 17 – Задание параметров нового справочника

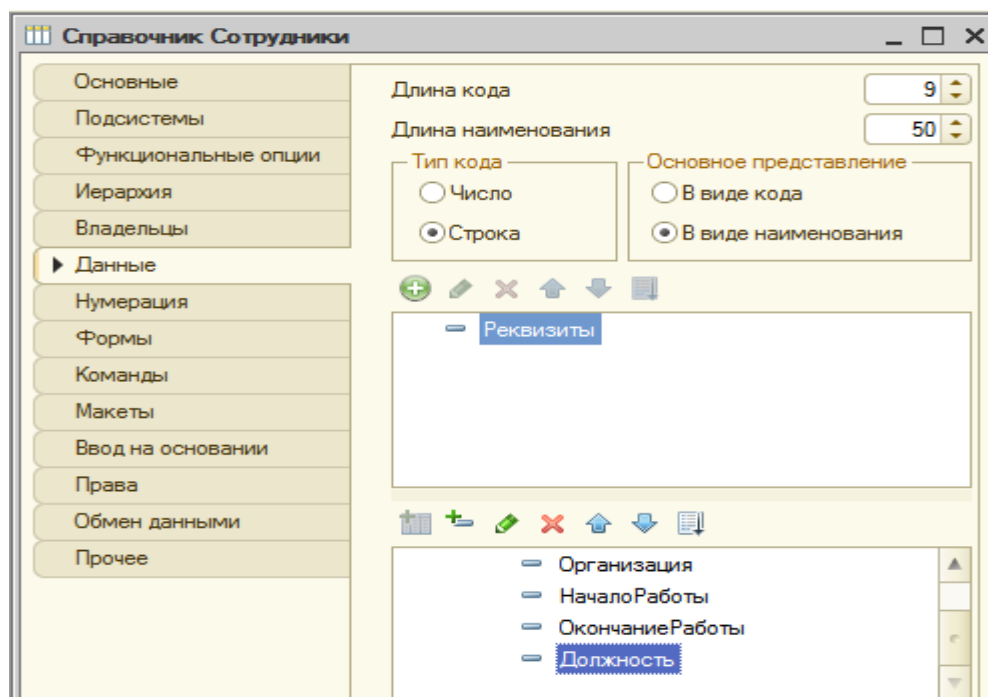


Рисунок 18 – Задание параметров нового справочника

В режиме «1С:Предприятие» заполним справочник «Сотрудники». Заполненный справочник представлен на рисунке 19.

Список сотрудников

Наименование Код

Гусakov Николай Дмитриевич	000000001
Деловой Иван Сергеевич	000000002
Симонов Валерий Михайлович	000000003

Рисунок 19 – заполненный справочник «Сотрудники»

Сделаем видимой в панели действий подсистемы РасчетЗарплаты команду добавления новых сотрудников.

Иерархический справочник

Создадим справочник «Номенклатура». Заданный при создании параметры представлены на рисунках 20-23.

Рисунок 20 – Заданные параметры справочника «Номенклатура»

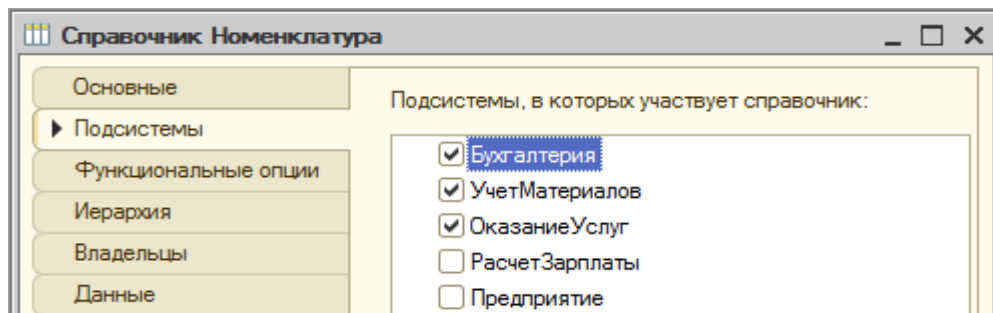


Рисунок 21 – Заданные параметры справочника «Номенклатура»

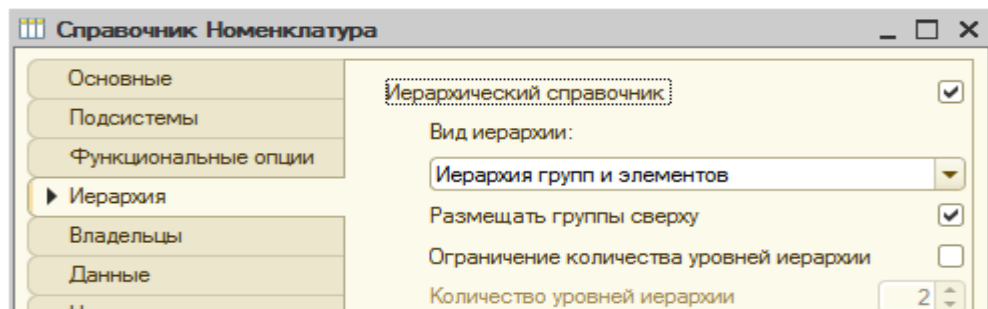


Рисунок 22 – Заданные параметры справочника «Номенклатура»

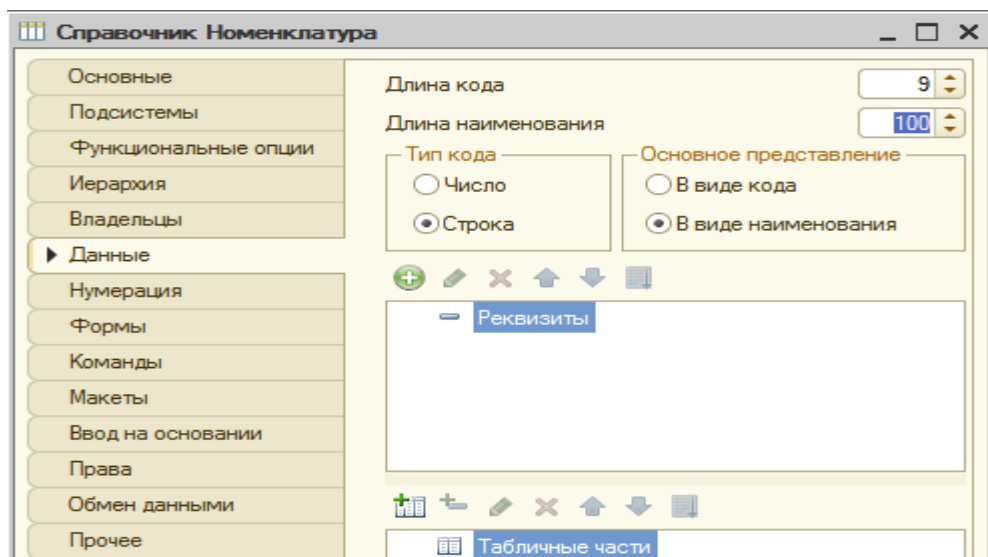


Рисунок 23 – Заданные параметры справочника «Номенклатура»

Сделаем видимой в панели действий разделов УчётМатериалов и ОказаниеУслуг команду создания новых элементов списка номенклатуры.

Заполним справочник «Номенклатура». Заполненный справочник представлен на рисунке 24.

Номенклатура	
Создать             Найти...	Все действия ▾
Наименование	Код
[-] Материалы	000000001
[-] Кабель электрический	000000007
[-] Строчный трансформатор GoldStar	000000004
[-] Строчный трансформатор Samsung	000000003
[-] Транзистор Phillips 2N2369	000000005
[-] Шланг резиновый	000000006
Наименование	Код
[-] Услуги	000000002
[-] Диагностика	000000008
[-] Подключение воды	000000011
[-] Подключение электричества	000000012
[-] Ремонт импортного телевизора	000000010
[-] Ремонт отечественного телевизора	000000009

Рисунок 24 – Заполненный справочник «Номенклатура»

Разнесем услуги по двум смысловым группам: услуги по ремонту телевизоров и услуги по установке стиральных машин, а материалы на радиодетали и прочее. Результат представлен на рисунке 25.

[-] Номенклатура	
[-] Материалы	000000001
[-] Прочее	000000016
[-] Кабель электрический	000000007
[-] Шланг резиновый	000000006
[-] Радиодетали	000000015
[-] Строчный трансформатор GoldStar	000000004
[-] Строчный трансформатор Samsung	000000003
[-] Транзистор Phillips 2N2369	000000005
[-] Услуги	000000002
[-] Стиральные машины	000000013
[-] Подключение воды	000000011
[-] Подключение электричества	000000012
[-] Телевизоры	000000014
[-] Диагностика	000000008
[-] Ремонт импортного телевизора	000000010
[-] Ремонт отечественного телевизора	000000009

Рисунок 25 – разделенные по группам элементы «Номенклатуры»

Создадим справочник «Склады». Заданные параметры справочника «Склады» представлены на рисунке 26.

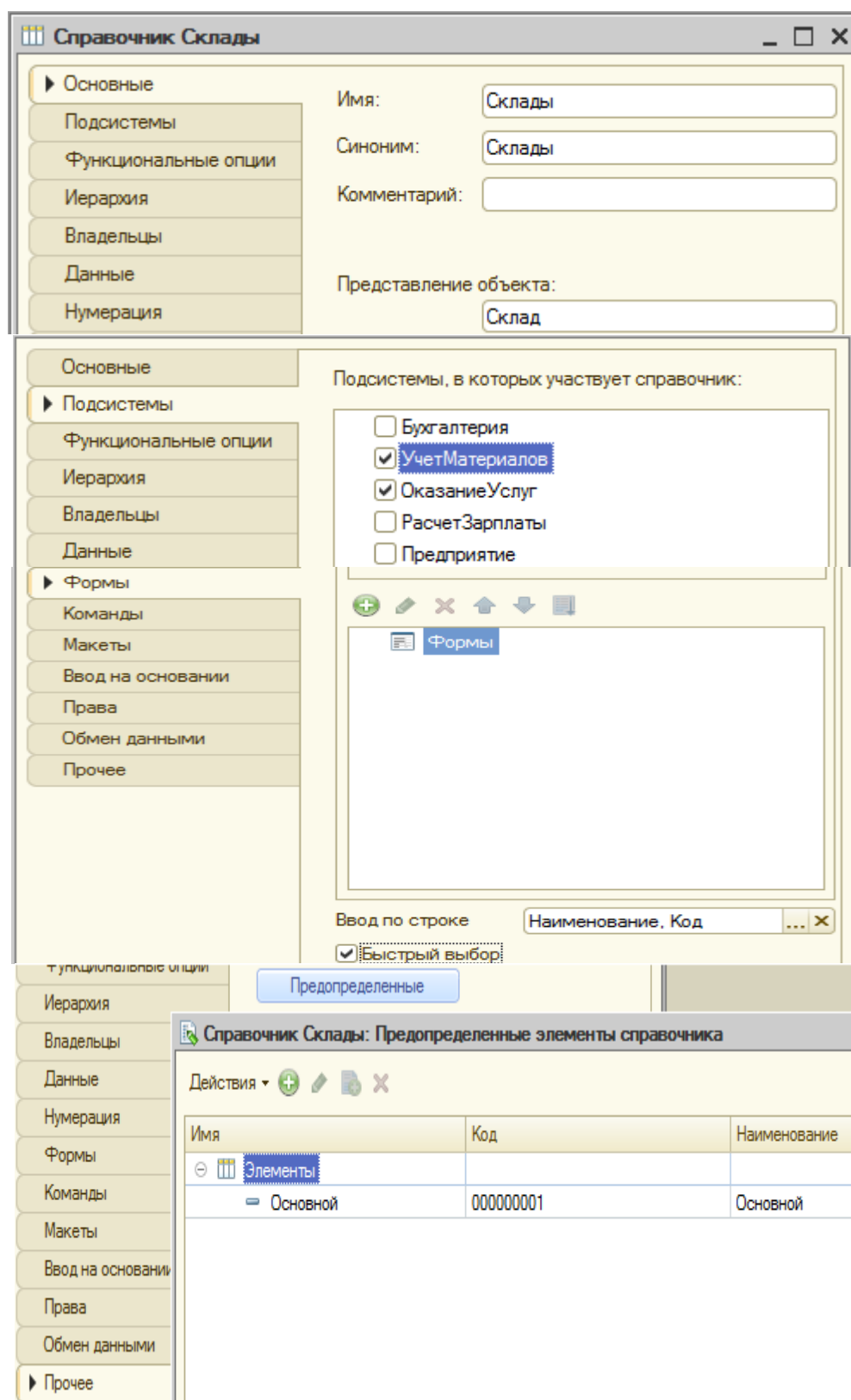


Рисунок 26 – Параметры справочника «Склады»

В панели действий раздела УчётМатериалов сделаем доступной команду создания новых складов.

Создадим новый элемент «Розничный» в справочнике «Склады». Справочник «Склады» представлен на рисунке 24.

<b>Склады</b>	
<div> <span>Создать</span> <span>Найти...</span> </div>	
Наименование	Код
Основной	000000001
Розничный	000000002

Рисунок 27 – справочник «Склады»

Создадим документ «Приходная накладная». Заданные параметры документа «Приходная накладная» представлены на рисунке 28.

Документ ПриходнаяНакладная

- Основные
- Подсистемы
- Функциональные опции
- Данные
- Нумерация
- Движения
- Последовательности
- Журналы
- Формы
- Команды

Имя: ПриходнаяНакладная

Синоним: Приходная накладная

Комментарий:

Представление объекта:

Расширенное представление объекта:

Представление списка: Приходные накладные

Основные

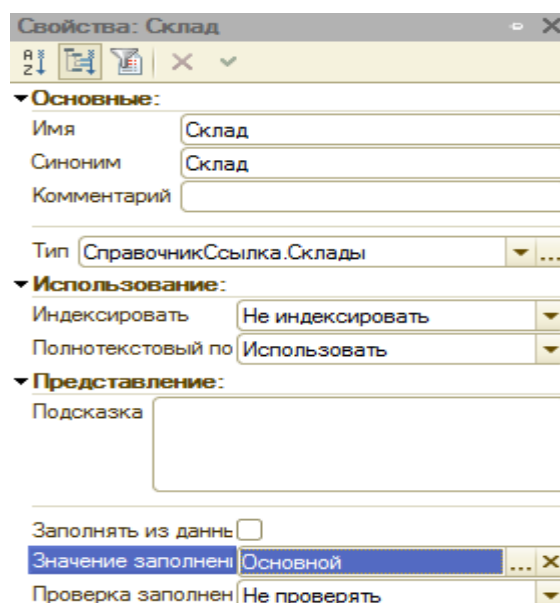
- Подсистемы
- Функциональные опции
- Данные
- Нумерация
- Движения

Подсистемы, в которых участвует документ:

- ☒ Бухгалтерия
- ☒ УчетМатериалов
- ☐ ОказаниеУслуг
- ☐ РасчетЗарплаты
- ☐ Предприятие

Рисунок 28 – Параметры документа «Приходная накладная»

Создадим реквизит документа с именем «Склад». Параметры созданного реквизита представлены на рисунке 29.



**Свойства: Склад**

▼ **Основные:**

Имя: Склад

Синоним: Склад

Комментарий:

Тип: СправочникСсылка.Склады

▼ **Использование:**

Индексировать: Не индексировать

Полнотекстовый по: Использовать

▼ **Представление:**

Подсказка:

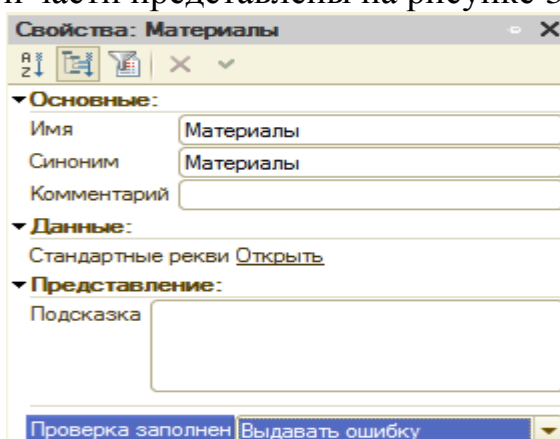
Заполнять из даннь: ☐

Значение заполнен: Основной

Проверка заполнен: Не проверять

Рисунок 29 – Параметры реквизита «Склад»

Добавим в документ табличную часть с именем «Материалы». Параметры табличной части представлены на рисунке 30.



**Свойства: Материалы**

▼ **Основные:**

Имя: Материалы

Синоним: Материалы

Комментарий:

▼ **Данные:**

Стандартные рекви: Открыть

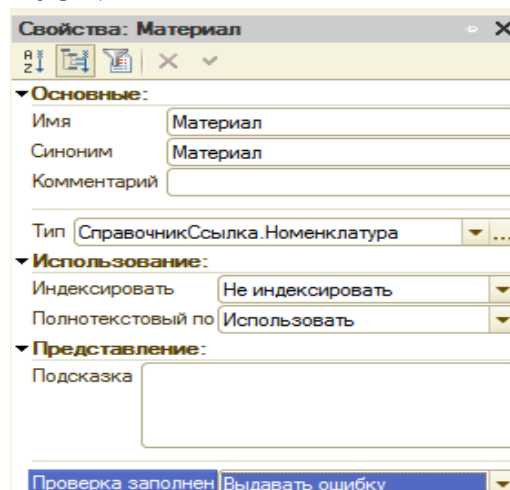
▼ **Представление:**

Подсказка:

Проверка заполнен: Выдавать ошибку

Рисунок 30 – Параметры табличной части документа

Создадим реквизиты табличной части «Материалы». Реквизиты представлены на рисунке 31.



**Свойства: Материал**

▼ **Основные:**

Имя: Материал

Синоним: Материал

Комментарий:

Тип: СправочникСсылка.Номенклатура

▼ **Использование:**

Индексировать: Не индексировать

Полнотекстовый по: Использовать

▼ **Представление:**

Подсказка:

Проверка заполнен: Выдавать ошибку

Рисунок 31 – Реквизиты табличной части «Материалы»

Отредактируем командный интерфейс так, чтобы в подсистеме «Учет материалов» была доступна команда создания новых документов.

Создадим несколько «Приходных накладных». Созданные накладные представлены на рисунке 32.

N	Материал	Количество	Цена	Сумма
1	Строчный трансформатор GoldStar	10,000	270,00	2 700,00
2	Строчный трансформатор Samsung	10,000	600,00	6 000,00
3	Транзистор Phillips 2N2369	10,000	3,00	30,00

Рисунок 32 – Приходные накладные

Сделаем автоматический пересчет суммы в строках документа. Автоматический пересчет представлен на рисунке 33.

```

<НаКлиенте
  Процедура МатериалыКоличествоПриИзменении (Элемент)
  СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;
  СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество * СтрокаТабличнойЧасти.Цена;
  КонечПроцедуры

```

Рисунок 33 – Автоматический пересчет суммы

В «Общих модулях» создадим модуль «РаботаСДокументами» и пропишем в нём код, который представлен на рисунке 34.

```

Процедура.РассчитатьСумму (СтрокаТабличнойЧасти) Экспорт
  СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество * СтрокаТабличнойЧасти.Цена;
  КонечПроцедуры

```

Рисунок 34 – Код процедуры «Рассчитать сумму»

Теперь в модуле нашей формы изменим текст обработчика. Новый текст обработчика представлен на рисунке 35.

```

<НаКлиенте
  Процедура МатериалыКоличествоПриИзменении (Элемент)
  СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;
  РаботаСДокументами.РассчитатьСумму (СтрокаТабличнойЧасти);
  КонечПроцедуры

```

Рисунок 35 – Текст обработчика



Аналогично созданию документа «Приходная накладная» создадим документ «Оказание услуги».

Заполним новый документ «Оказание услуги». Заполненный документ представлен на рисунке 36.

N	Номенклатура	Количество	Цена	Сумма
1	Транзистор Phillips 2N2369	1,000	3,00	3,00

Рисунок 36 – Заполненный документ «Оказание услуги»

Добавим регистр накопления «ОстаткиМатериалов». Заданные параметры представлены на рисунке 37.

Рисунок 37 – Параметры регистра накопления «ОстаткиМатериалов»

Настроим движение документа. Заданные параметры представлены на рисунке 38.

Рисунок 38 – Параметры движения документа

В итоге конструктор сформировал текст в модуле документа «ПриходнаяНакладная» представленный на рисунке 39.

```

Процедура ОбработкаПроведения(Отказ, Режим)
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!

// регистр ОстаткиМатериалов Приход
Движения.ОстаткиМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
    Движение = Движения.ОстаткиМатериалов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Материал = ТекСтрокаМатериалы.Материал;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаМатериалы.Количество;
КонецЦикла;

//}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

Рисунок 39 – Текст сформированный конструктором

Отредактируем командный интерфейс, чтобы в подсистемах «Бухгалтерия», «Оказание услуг» и «Учет материалов» была доступна ссылка для просмотра записей регистра накоплений.

Сделаем возможность вызова из формы документа список регистра, в котором показаны движения, произведенные только этим документом. Настройки представлены на рисунке 40.

Рисунок 40 – Настройки документа

Аналогичным образом создадим движение документа «Оказание услуги».

Создадим отчет, который будет нам показывать приход, расход и остатки материалов.

Зададим схему компоновки данных. Компоновка данных представлена на рисунках 41 - 45.

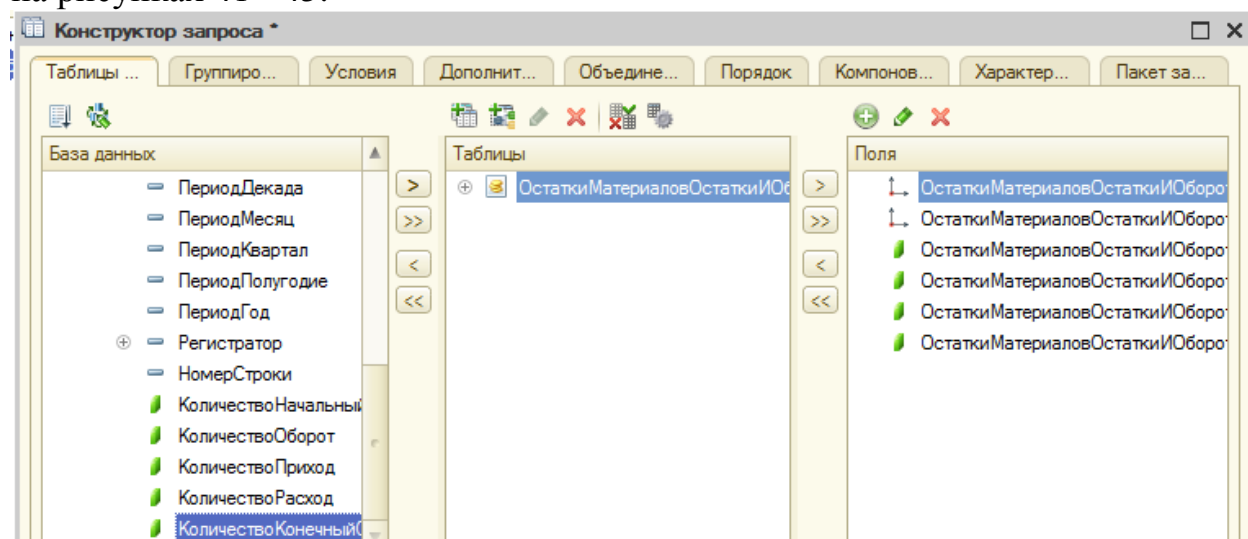


Рисунок 41 – Задание схемы компоновки данных

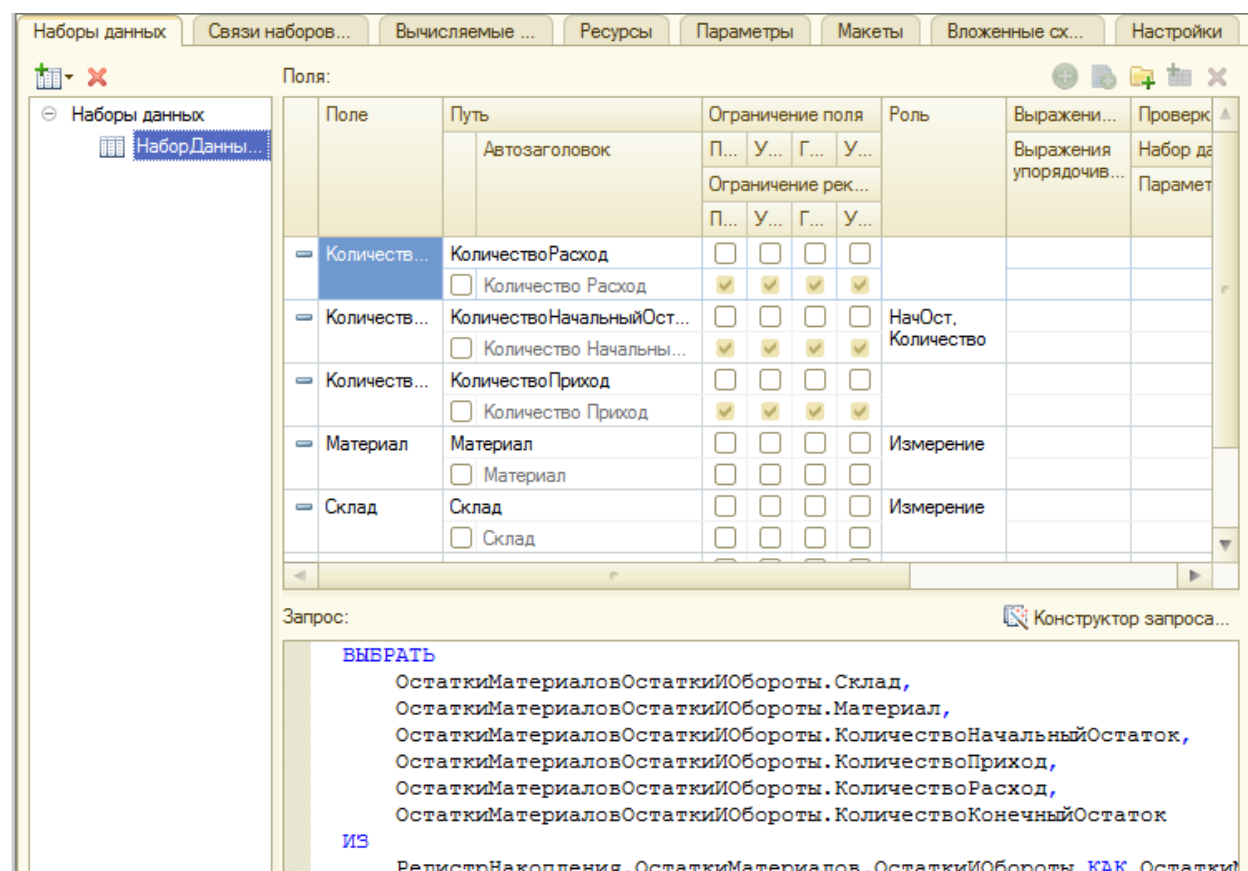


Рисунок 42 – Задание схемы компоновки данных

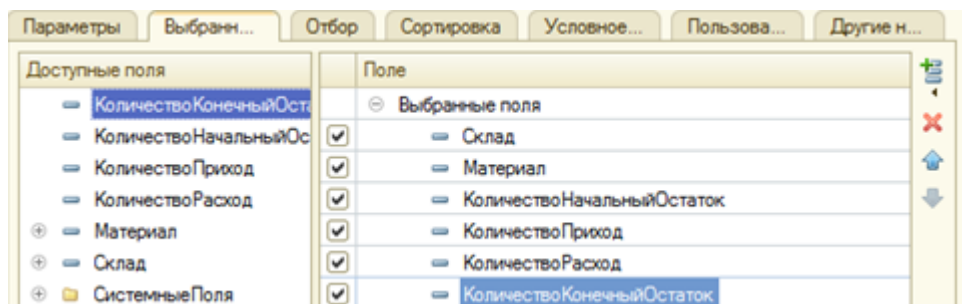


Рисунок 43 – Задание схемы компоновки данных

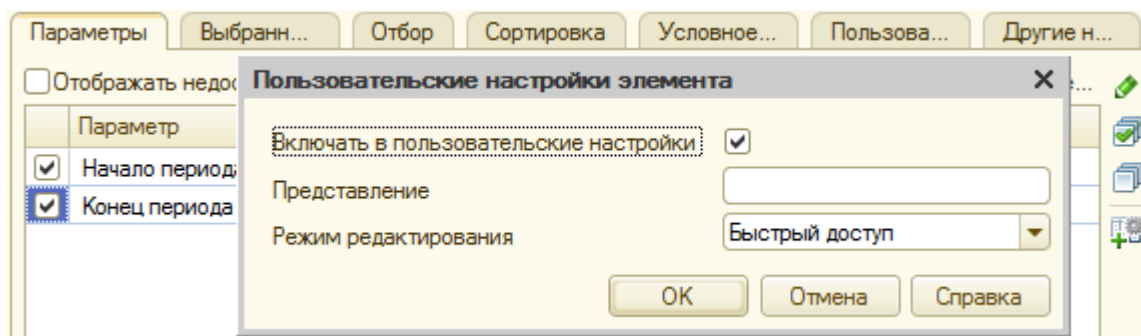


Рисунок 44 – Задание схемы компоновки данных

Зададим подсистемы в которых участвует отчет: «Бухгалтерия», «УчётМатериалов», «ОказаниеУслуг».

Далее создадим макет печатной формы документа «Оказание услуги». Задание параметров конструктора печати представлено на рисунках 45, 46.

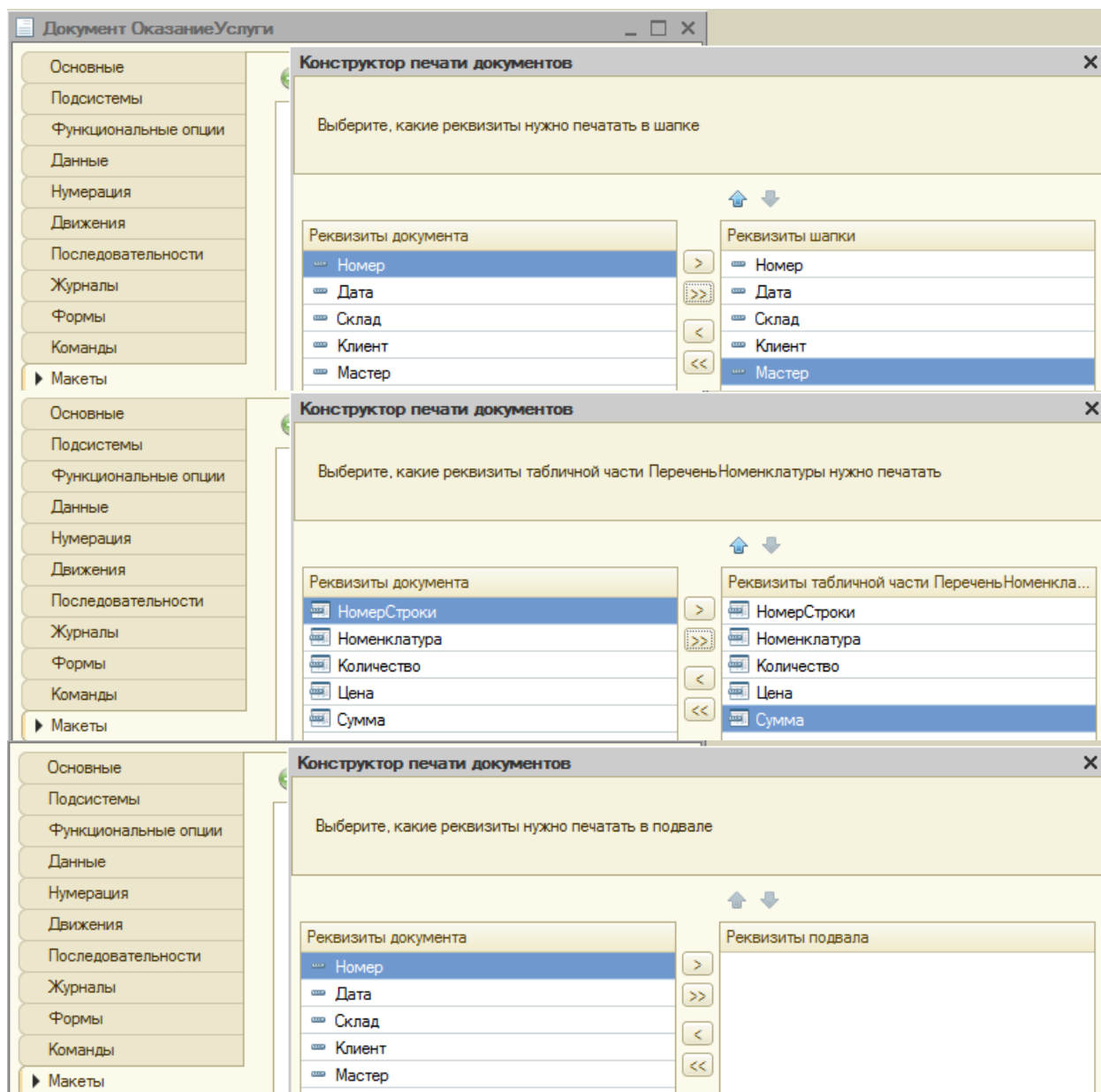


Рисунок 45 – Задание параметров конструктора печати

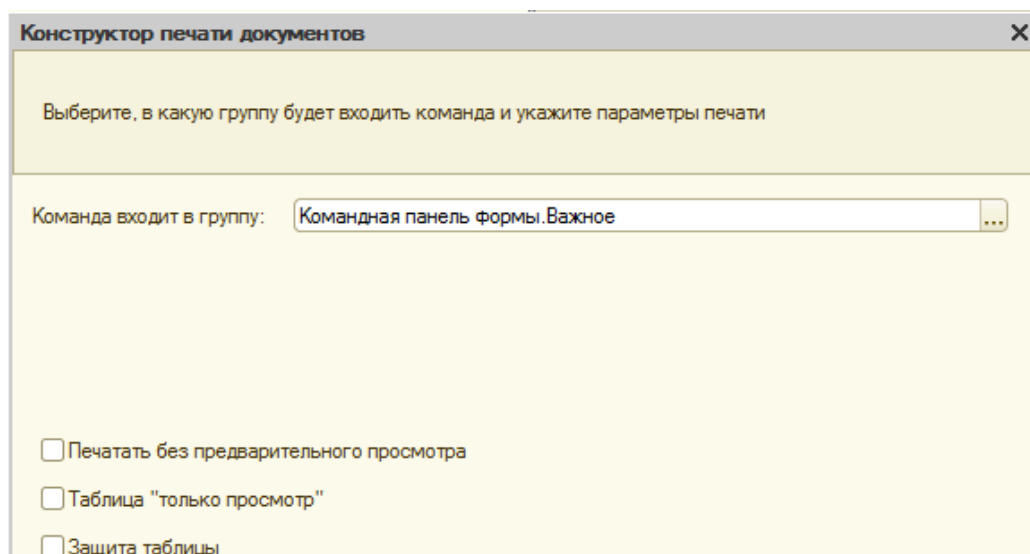


Рисунок 46 – Задание параметров конструктора печати

Добавим итоговую сумму в печатную форму документа. Добавление представлено на рисунке 47.

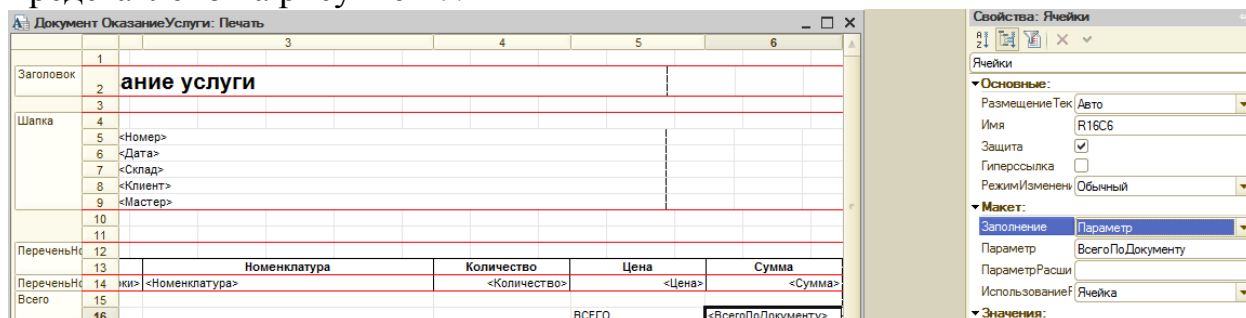


Рисунок 47 – Добавление итоговой суммы в печатную форму документа

Добавим строки, выделенные знаком «>>>» в тело процедуры «Печать» в модуле менеджера документа «Оказание услуги». Измененная процедура представлена на рисунке 48.

```

ОбластьЗаголовок = Макет.ПолучитьОбласть ("Заголовок");
Шапка = Макет.ПолучитьОбласть ("Шапка");
ОбластьПереченьНоменклатурыШапка = Макет.ПолучитьОбласть ("ПереченьНоменклатурыШапка");
ОбластьПереченьНоменклатуры = Макет.ПолучитьОбласть ("ПереченьНоменклатуры");
ОбластьИтог = Макет.ПолучитьОбласть ("Всего");
>>> ТабДок.Очистить ();

ВставляяРазделительСтраниц = Ложь;
Пока Выборка.Следующий() Цикл
    Если ВставляяРазделительСтраниц Тогда
        ТабДок.ВывестиГоризонтальныйРазделительСтраниц ();
    КонецЕсли;

    ТабДок.Вывести(ОбластьЗаголовок);

    Шапка.Параметры.Заполнить(Выборка);
    ТабДок.Вывести(Шапка, Выборка.Уровень());

    ТабДок.Вывести(ОбластьПереченьНоменклатурыШапка);
    ВыборкаПереченьНоменклатуры = Выборка.ПереченьНоменклатуры.Выбрать();
>>> СуммаИтог = 0;
    Пока ВыборкаПереченьНоменклатуры.Следующий() Цикл
        ОбластьПереченьНоменклатуры.Параметры.Заполнить(ВыборкаПереченьНоменклатуры);
        ТабДок.Вывести(ОбластьПереченьНоменклатуры, ВыборкаПереченьНоменклатуры.Уровень());
>>> СуммаИтог = СуммаИтог + ВыборкаПереченьНоменклатуры.Сумма;
    КонецЦикла;

>>> ОбластьИтог.Параметры.ВсегоПоДокументу = СуммаИтог;
>>> ТабДок.Вывести(ОбластьИтог);

```

Рисунок 48 – Процедура «Печать»

Сделаем возможность просмотра суммы документа не печатая его. Для этого добавим «подвал» у документа и отредактируем свойства полей «Цена», и «Сумма». Результат представлен на рисунке 49.

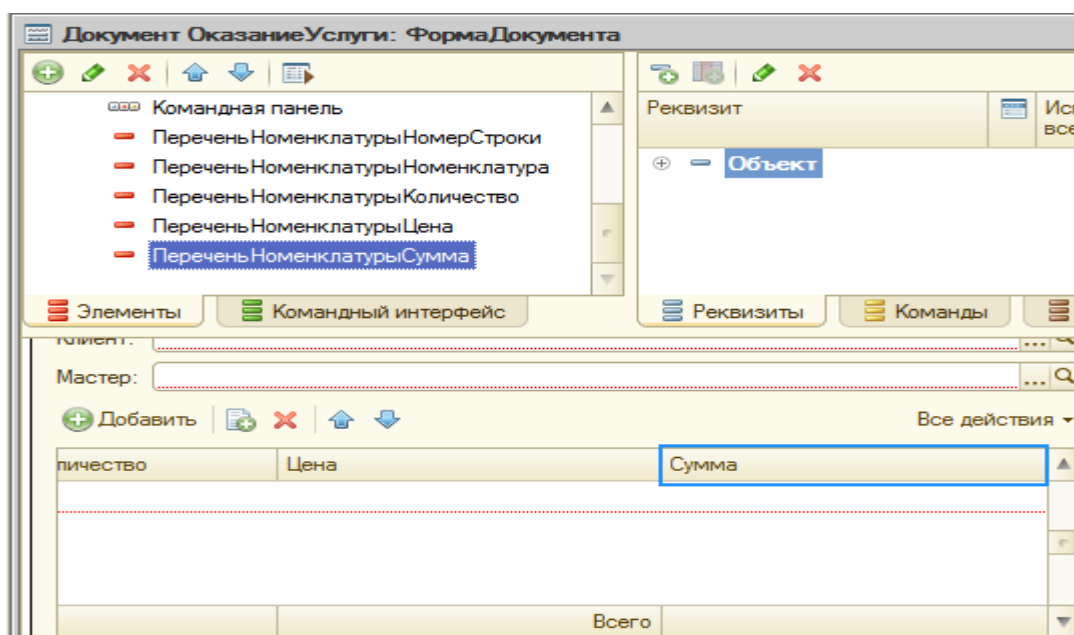


Рисунок 49 – Измененная форма документа

Создадим периодический регистр сведений, который будет хранить развернутые во времени розничные цены материалов и стоимости услуг. Заданные при создании свойства представлены на рисунке 50.

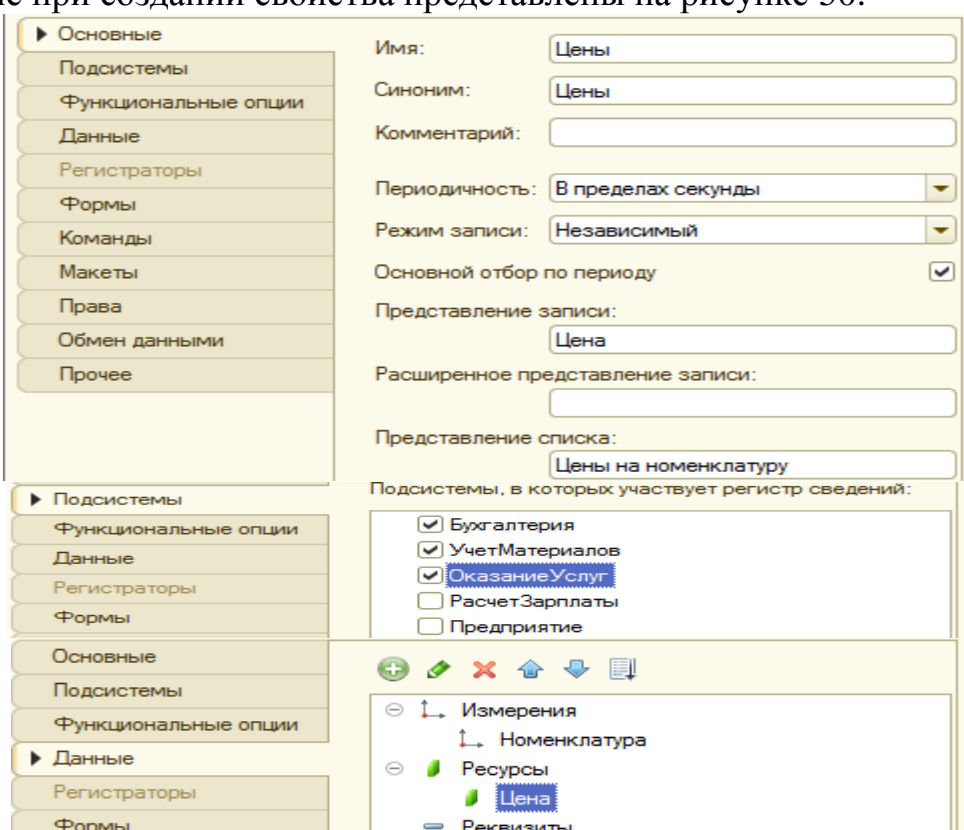


Рисунок 50 – Свойства периодического регистра сведений

Автоматическая подстановка цены в документ при выборе номенклатуры

Создадим общий модуль «РаботаСоСправочниками». Создание представлено на рисунке 51.



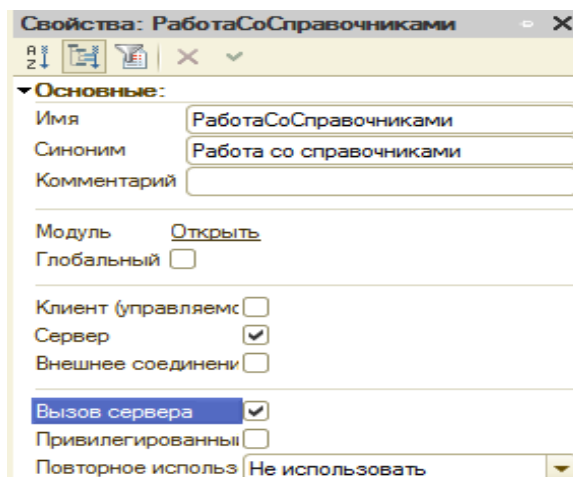


Рисунок 51 – Общий модуль «РаботаСоСправочниками»

В модуле «РаботаСоСправочниками» прописываем следующую функцию «РозничнаяЦена()» представленную на рисунке 52.

```

Функция РозничнаяЦена(АктуальнаяДата, ЭлементНоменклатуры) Экспорт
    //Создать вспомогательный объект Отбор
    Отбор = Новый Структура("Номенклатура", ЭлементНоменклатуры);

    //Получить актуальные значения ресурсов регистра
    ЗначенияРесурсов = РегистрыСведений.Цены.ПолучитьПоследнее(АктуальнаяДата, Отбор);
    Возврат ЗначенияРесурсов.Цена;

Конецфункции
  
```

Рисунок 52 – Функция «РозничнаяЦена()»

Создадим новый объект конфигурации Перечисление с именем «ВидНоменклатуры». Свойства нового объекта представлены на рисунке 53.

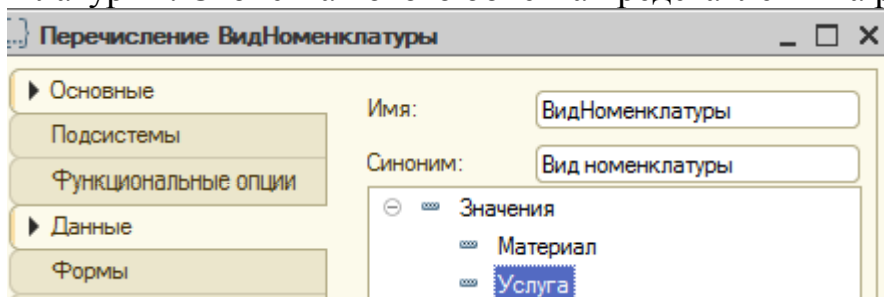


Рисунок 53 – Свойства объекта «ВидНоменклатуры»

Привязка номенклатуры к значениям перечисления «ВидНоменклатуры»

В справочнике «Номенклатура» добавим реквизит «ВидНоменклатуры». Свойства реквизита «ВидНоменклатуры» представлены на рисунке 54.

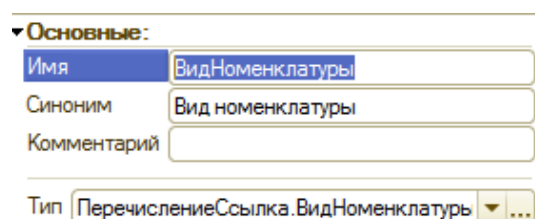


Рисунок 54 – Реквизит «ВидНоменклатуры»



Для регистрации только той номенклатуры, которая является материалом, добавим в процедуру «ОбработкаПроведения» строки отмеченные знаком «>>>» на рисунке 55.

```

>>> Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
>>> Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры = Перечисления.ВидНоменклатуры.Материал Тогда
    Движение = Движения.ОстаткиМатериалов.Добавить ();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
>>>| КонечЕсли;
    КонечЦикла;

```

Рисунок 55 – Измененная процедура «Обработка проведения»

По аналогии с регистром «ОстаткиМатериалов» создадим новый регистр накопления «СтоимостьМатериалов».

Изменим процедуру проведения. Отметим что документ будет создавать движение и по регистру «СтоимостьМатериалов». Измененная процедура «ОбработкаПроведения» представлена на рисунке 56.

```

Процедура ОбработкаПроведения(Отказ, Режим)
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

// регистр ОстаткиМатериалов Приход
Движения.ОстаткиМатериалов.Записывать = Истина;
Движения.СтоимостьМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
    Движение = Движения.ОстаткиМатериалов.Добавить ();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Материал = ТекСтрокаМатериалы.Материал;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаМатериалы.Количество;

//регистр СтоимостьМатериалов Приход
Движение = Движения.СтоимостьМатериалов.Добавить ();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаМатериалы.Материал;
Движение.Стоимость = ТекСтрокаМатериалы.Сумма;
КонечЦикла;

//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонечПроцедуры

```

Рисунок 56 – Процедура «ОбработкаПроведения»

Отредактируем командный интерфейс документа, чтобы иметь возможность переходить к списку записей регистра «СтоимостьМатериалов».

Для проведения документа «Оказание услуги» по двум регистрам сначала добавим в табличную часть документа новый реквизит «Стоимость». Затем изменим процедуру проведения документа. Измененная процедура проведения документа представлена на рисунке 57.

```

Процедура ОбработкаПроведения(Отказ, Режим)
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

// регистр ОстаткиМатериалов Расход
Движения.ОстаткиМатериалов.Записывать = Истина;
Движения.СтоимостьМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры = Перечисления.ВидНоменклатуры.Материал Тогда
        Движение = Движения.ОстаткиМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
        Движение.Склад = Склад;
        Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

        //регистр СтоимостьМатериалов Расход
        Движение = Движения.ОстаткиМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
        Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Количество * ТекСтрокаПереченьНоменклатуры.Стоимость;
    КонецЕсли;
КонецЦикла;

//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

Рисунок 57 – Процедура проведения документа

Отредактируем командный интерфейс документа, чтобы иметь возможность переходить к списку записей регистра «СтоимостьМатериалов».

Создадим регистр накопления «Продажи», указав его тип как «Оборотный».

Отредактируем командный интерфейс документа, чтобы в подсистемах «Бухгалтерия», «Оказание услуг» и «Учет материалов» была доступна ссылка для просмотра оборотного регистра накопления.

Для проведения документа «Оказание услуги» по трем регистрам укажем, что документ «Оказание услуги» будет создавать движение и по регистру «Продажи».

Изменим процедуру проведения документа. Измененная процедура представлена на рисунке 58.

```

Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры = Перечисления.ВидНоменклатуры.Материал Тогда
    Движение = Движения.ОстаткиМатериалов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

    //регистр СтоимостьМатериалов Расход
    Движение = Движения.СтоимостьМатериалов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Количество * ТекСтрокаПереченьНоменклатуры.Стоимость;
КонецЕсли;

//регистр Продажи
Движение = Движения.Продажи.Добавить();
Движение.Период = Дата;
Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Клиент = Клиент;
Движение.Мастер = Мастер;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
Движение.Выручка = ТекСтрокаПереченьНоменклатуры.Сумма;
Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Стоимость * ТекСтрокаПереченьНоменклатуры.Количество;
конецЦикла;

```

Рисунок 58 – Процедура проведения

Отредактируем командный интерфейс документа, чтобы иметь возможность переходить к списку записей регистра «Продажи».

Создадим отчет «Реестр документов оказание услуги». После задания параметров в конструкторе запроса получим запрос представленный на рисунке 59.

```

ВЫБРАТЬ
    ОказаниеУслуги.Склад,
    ОказаниеУслуги.Мастер,
    ОказаниеУслуги.Клиент,
    ОказаниеУслуги.Ссылка КАК Документ
ИЗ
    Документ.ОказаниеУслуги КАК ОказаниеУслуги

УПОРЯДОЧИТЬ ПО
    Документ
    
```

Рисунок 59 – Запрос, сформированный конструктором запросов

Зададим настройки, определяющие, как будет выводиться информация в отчет. Настройки представлены на рисунке 60.

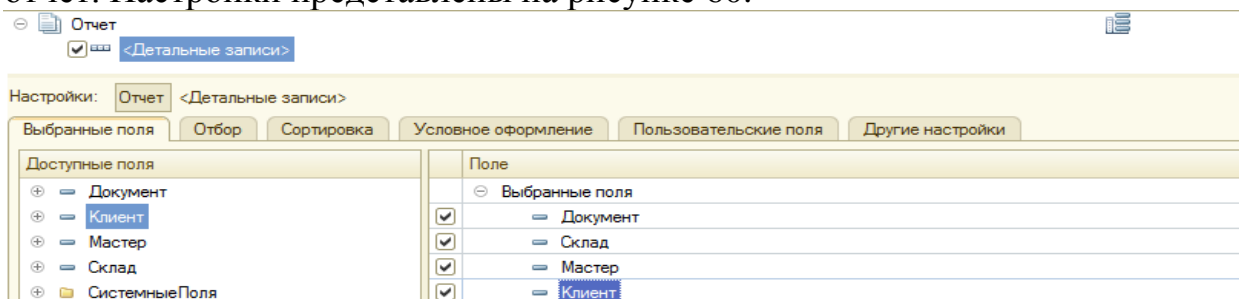


Рисунок 60 – Настройки отображения информации

Отметим на закладке «Подсистемы» подсистему «ОказаниеУслуг» для того, чтобы ссылка на отчет попала на панель действий подсистемы.

Создадим отчет «Рейтинг услуг», который будет отражать список услуг, принесших наибольшую прибыль в указанном периоде. Параметры создания отчета представлены на рисунках 61 - 69.

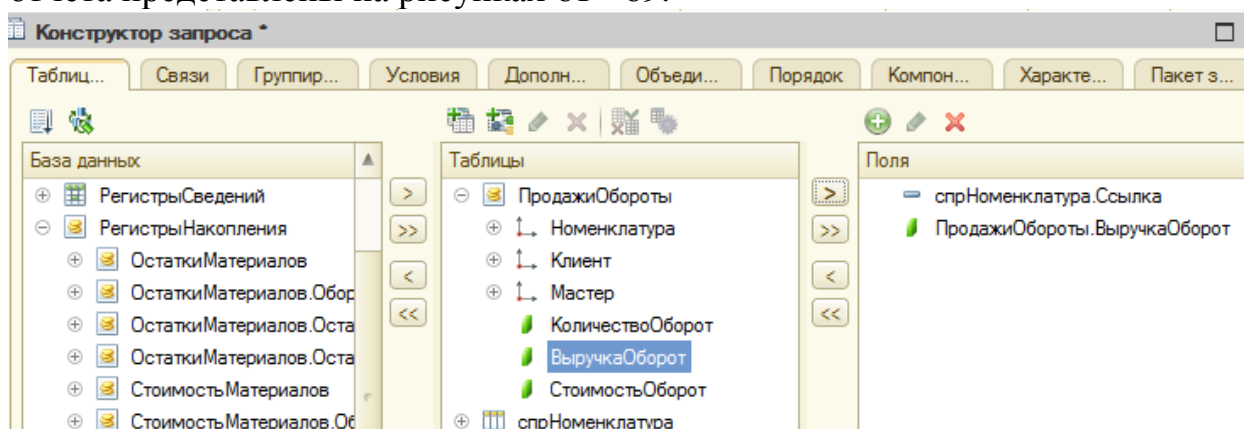


Рисунок 61 – Параметры отчета

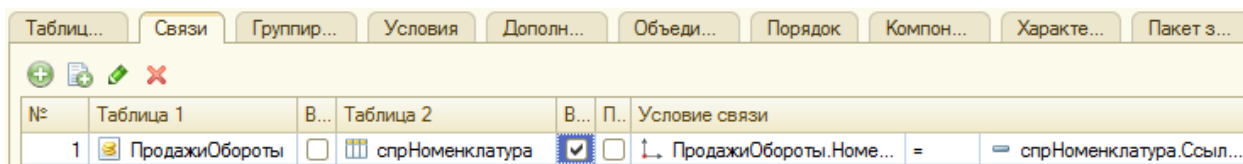


Рисунок 62 – Параметры отчета

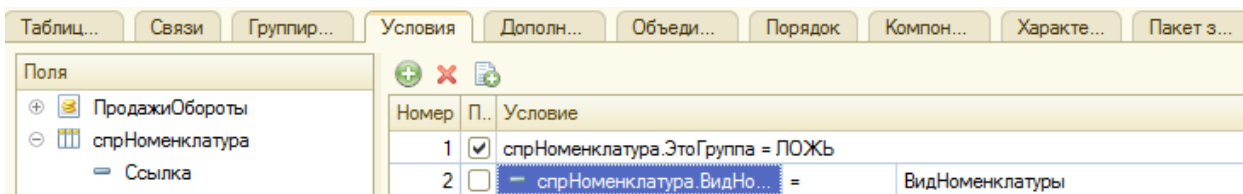


Рисунок 63 – Параметры отчета

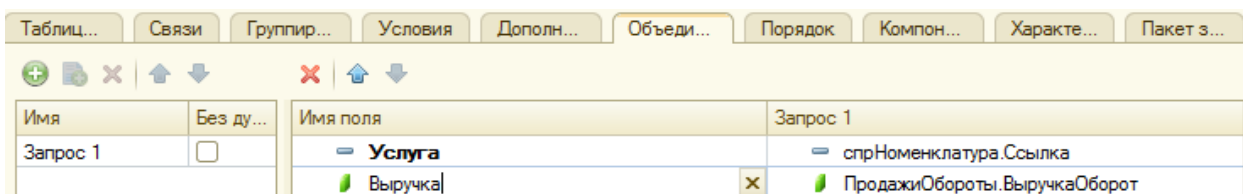


Рисунок 64 – Параметры отчета

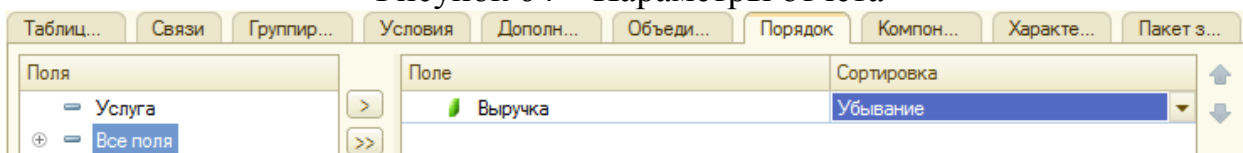


Рисунок 65 – Параметры отчета

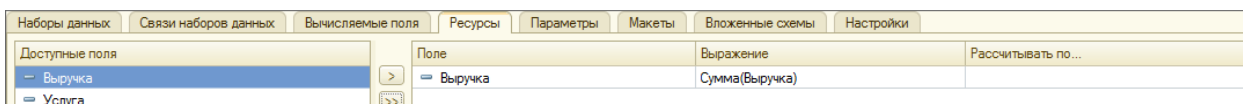


Рисунок 66 – Параметры отчета

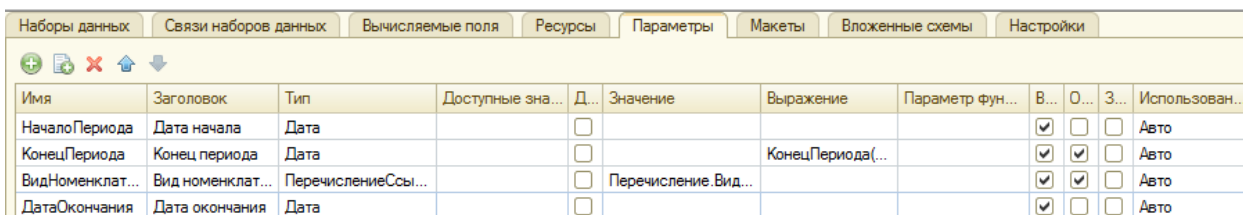


Рисунок 67 – Параметры отчета

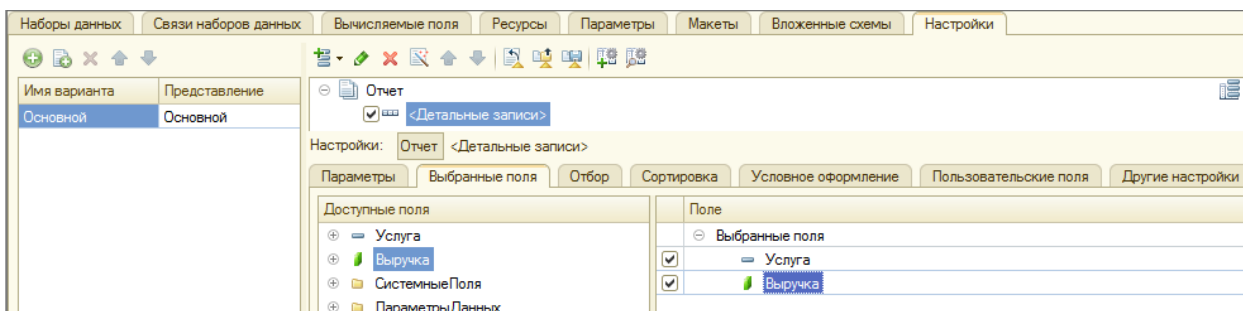


Рисунок 68 – Параметры отчета

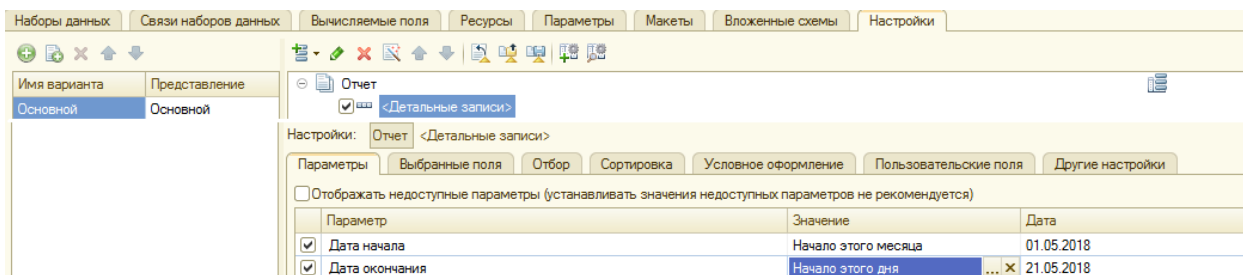


Рисунок 69 – Параметры отчета

Зададим параметры условного форматирования. Заданные параметры представлены на рисунке 70.

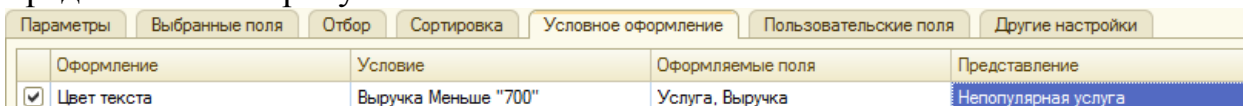


Рисунок 70 – Параметры условного форматирования

Предоставим пользователю часть полных настроек, которые задает разработчик. Предоставление представлено на рисунке 71.

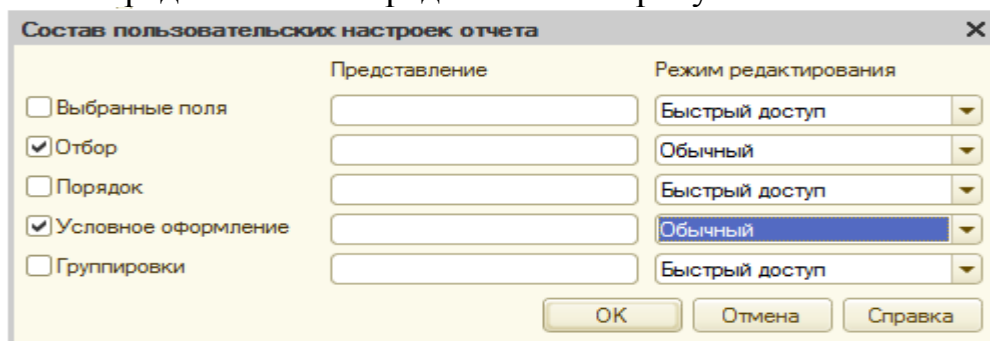


Рисунок 71 – Состав пользовательских настроек отчета

Создадим настройку отбора в отчете. Параметры представлены на рисунке 72.

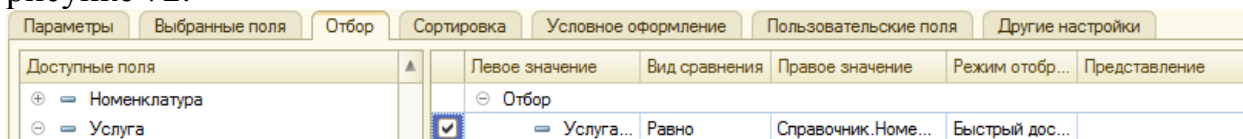


Рисунок 72 – Параметры отбора

Создадим отчет «ВыручкаМастеров» по подобию прошлого отчета, но укажем периодичность в один день. Задание периодичности представлено на рисунках 73 и 74.

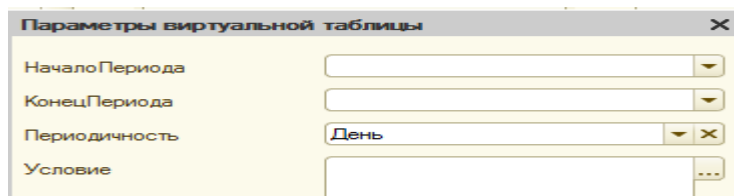


Рисунок 73 – Задание периодичности

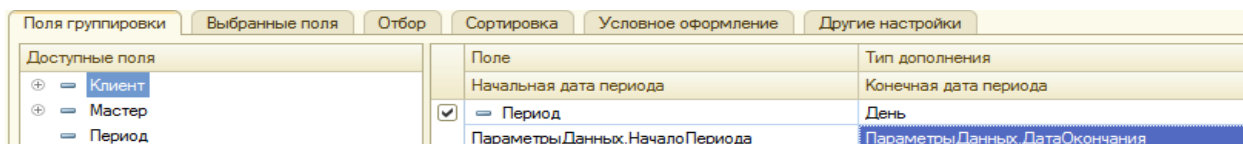


Рисунок 74 – Задание периодичности

Создадим отображение отчета в виде диаграммы. Заданные параметры при создании диаграммы представлены на рисунках 75, 76 и 77.

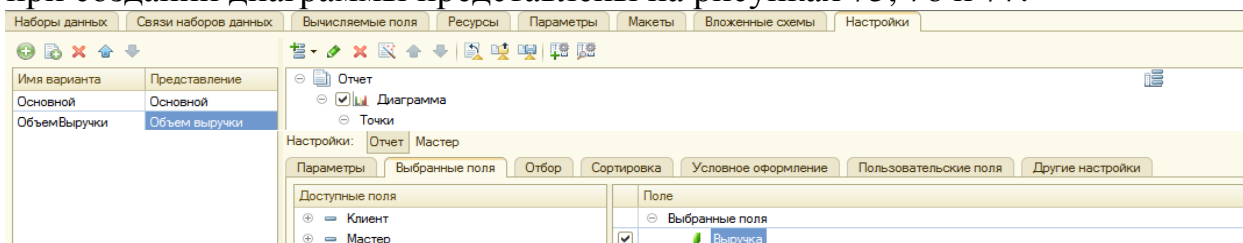


Рисунок 75 – Параметры диаграммы

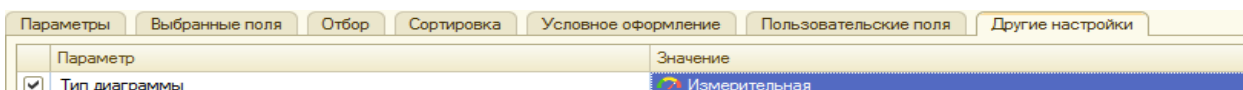


Рисунок 76 – Параметры диаграммы

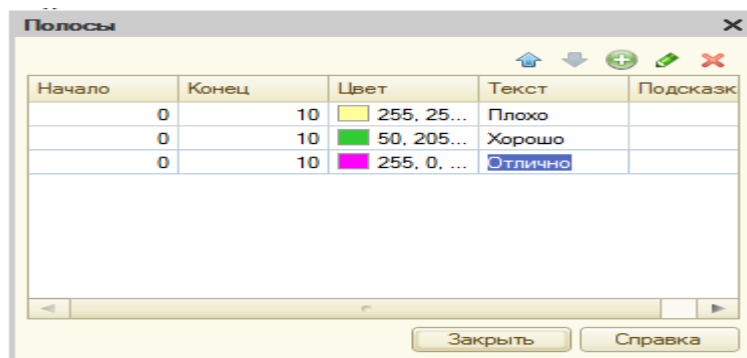


Рисунок 77 – Параметры диаграммы

Создадим отчет «Перечень услуг», содержащий информацию о том, какие услуги и по какой цене оказывает ООО «На все руки мастер».

Зададим параметры виртуальной таблицы. Задание параметров представлено на рисунке 78.

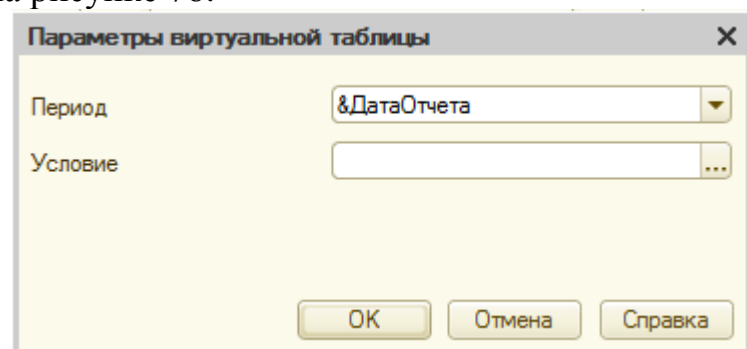


Рисунок 78 – Параметры виртуальной таблицы

Создадим отчет «Рейтинг клиентов».



Добавим вычисляемое поле. Параметры вычисляемого поля представлены на рисунке 79.

Наборы данных    Связи наборов данных <b>Вычисляемые поля</b> Ресурсы    Параметры    Make						
<div> <div>+</div> <div>+</div> <div>×</div> </div>						
Путь к данным	Выражение	Заголовок	Ограничение доступности			
			Поле	Усло...	Группа	Упор...
Доход	Выручка - Стоимость	Доход	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 79 – Параметры вычисляемого поля

По подобию прошлых отчетов создадим новый отчет «Универсальный».

Перейдем в конструктор запроса с обработкой результата и укажем необходимые параметры. Параметры представлены на рисунке 80.

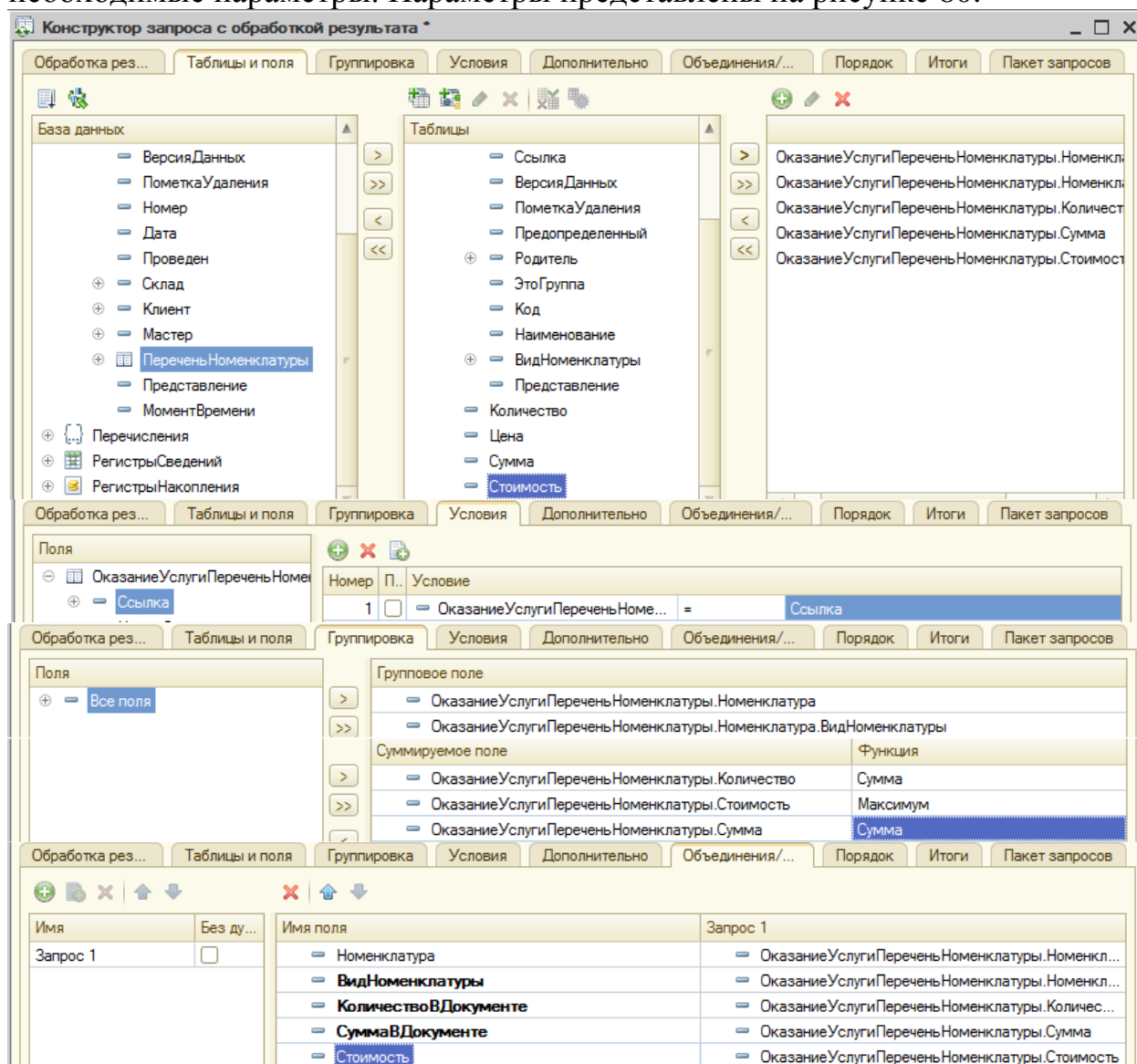


Рисунок 80 – Параметры запроса с обработкой результата

В результате процедура проведения примет вид, представленный на рисунке 81.

```

} Процедура ОбработкаПроведения(Отказ, Режим)
  {{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны

  // регистр ОстаткиМатериалов Расход
  Движения.ОстаткиМатериалов.Записывать = Истина;
  Движения.СтоимостьМатериалов.Записывать = Истина;
  Движения.Продажи.Записывать = Истина;
  {{{_КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны

  Запрос = Новый Запрос;
  Запрос.Текст =
    "ВЫБРАТЬ
    | ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
    | ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК ВидНоменклатуры,
    | СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество) КАК КоличествоВДокументе,
    | СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе,
    | МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость) КАК Стоимость
    | ИЗ
    | Документ.ОказаниеУслуги.ПереченьНоменклатуры КАК ОказаниеУслугиПереченьНоменклатуры
    | ГДЕ
    | ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка
    |
    | СГРУППИРОВАТЬ ПО
    | ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
    | ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";

  Запрос.УстановитьПараметр("Ссылка", Ссылка);

  Результат = Запрос.Выполнить();

  ВыборкаДетальныеЗаписи = Результат.Выбрать();

  Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Если ВыборкаДетальныеЗаписи.Номенклатура.ВидНоменклатуры = Перечисления.ВидНоменклатуры.Расход
    Тогда
      Движение = Движения.ОстаткиМатериалов.Добавить();
      Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
      Движение.Период = Дата;
      Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
      Движение.Склад = Склад;
      Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

      //регистр СтоимостьМатериалов Расход
      Движение = Движения.СтоимостьМатериалов.Добавить();
      Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
      Движение.Период = Дата;
      Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
      Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе * ВыборкаДетальныеЗаписи.Стоимость;
    КонецЕсли;

    //регистр Продажи
    Движение = Движения.Продажи.Добавить();
    Движение.Период = Дата;
    Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;
    Движение.Клиент = Клиент;
    Движение.Мастер = Мастер;
    Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;
    Движение.Выручка = ВыборкаДетальныеЗаписи.СуммаВДокументе;
    Движение.Стоимость = ВыборкаДетальныеЗаписи.Стоимость * ВыборкаДетальныеЗаписи.КоличествоВДокументе;

  КонецЦикла;
}

```

Рисунок 81 – Процедура проведения



Сначала удалим реквизит табличной части «Стоимость» документа «ОказаниеУслуги» и соответствующее поле из таблицы «ПереченьНоменклатуры».

Создадим менеджер временных таблиц и укажем, что этот запрос будет использовать менеджер временных таблиц. Создание менеджера временных таблиц представлено на рисунке 82.

```
//Создать менеджер временных таблиц.
МенеджерВТ = Новый МенеджерВременныхТаблиц;

Запрос = Новый Запрос;
//Укажем, какой менеджер временных таблиц использует этот запрос
Запрос.МенеджерВременныхТаблиц = МенеджерВТ;
```

Рисунок 82 – Создание менеджера временных таблиц

В конструкторе запроса изменим запрос. Изменение представлено на рисунке 83.

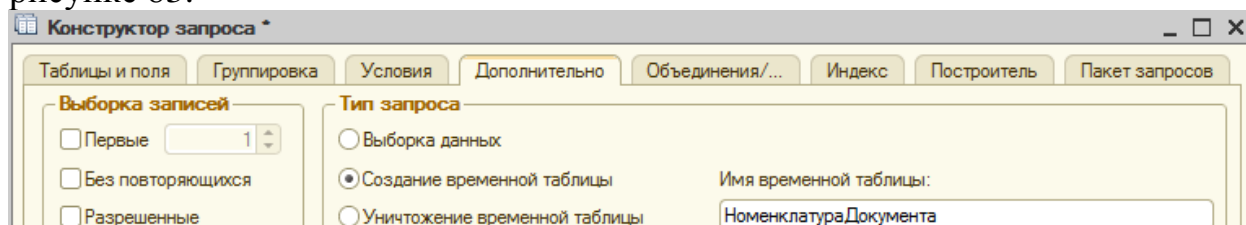


Рисунок 83 – Настройки запроса

Создадим новый запрос и сконструируем его в «Конструкторе запроса». Далее соединим виртуальную таблицу левым соединением с таблицей остатков. Параметры запроса представлены на рисунке 84.

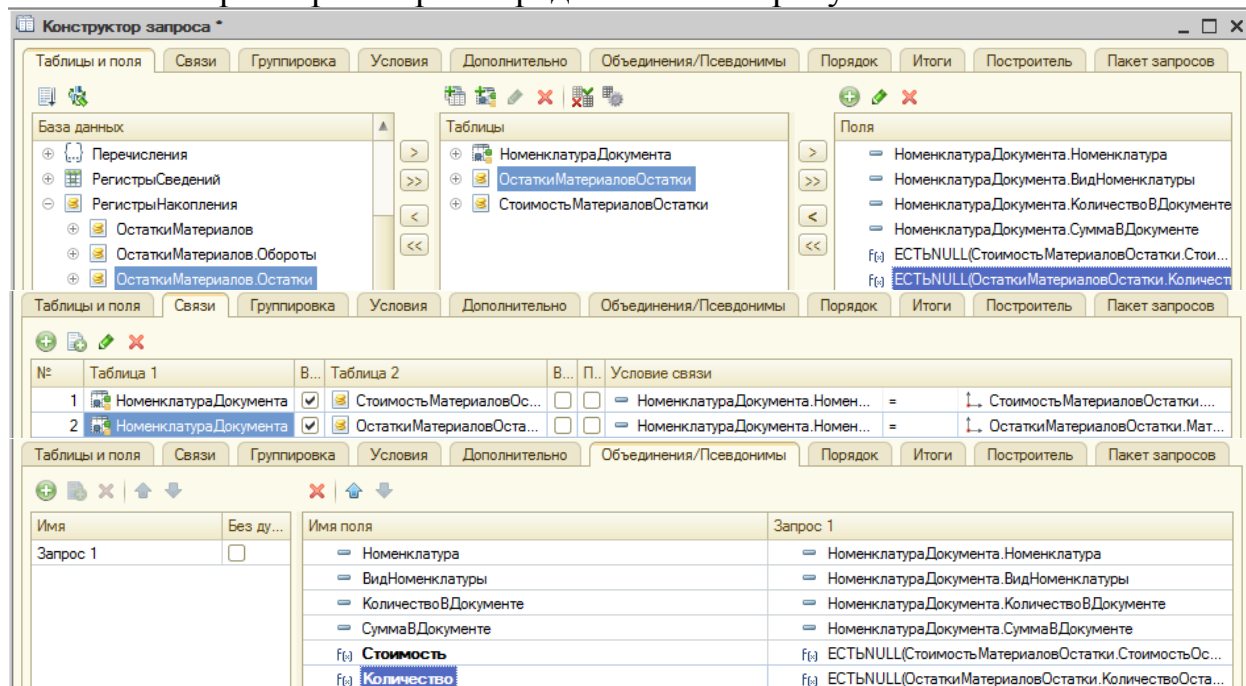


Рисунок 84 – Параметры запроса

Если у нас есть номенклатура, но она еще не поступала в нашу фирму, то в запросе лучше сразу избавиться от нее. Избавление представлено на рисунке 85.

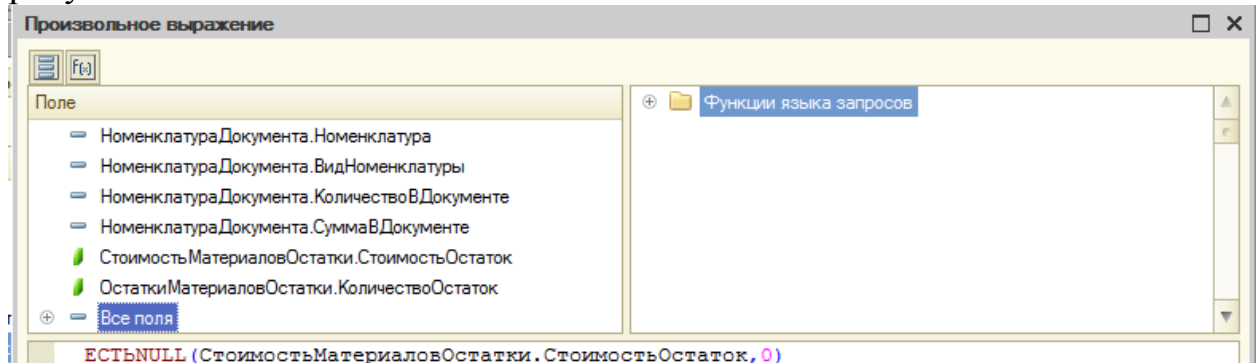


Рисунок 85 – Избавление от пустых значений

Полученный текст запроса представлен на рисунке 86.

```

ВЫБРАТЬ
НоменклатураДокумента.Номенклатура,
НоменклатураДокумента.ВидНоменклатуры,
НоменклатураДокумента.КоличествоВДокументе,
НоменклатураДокумента.СуммаВДокументе,
ЕСТЬNULL (СтоимостьМатериаловОстатки.СтоимостьОстаток, 0) КАК Стоимость,
ЕСТЬNULL (ОстаткиМатериаловОстатки.КоличествоОстаток, 0) КАК Количество
ИЗ
НоменклатураДокумента КАК НоменклатураДокумента
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.СтоимостьМатериалов.Остатки (
        /
        Материал В
        (ВЫБРАТЬ
            НоменклатураДокумента.Номенклатура
        ИЗ
            НоменклатураДокумента)) КАК СтоимостьМатериаловОстатки
    ПО НоменклатураДокумента.Номенклатура = СтоимостьМатериаловОстатки.Материал
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОстаткиМатериалов.Остатки (
        /
        Материал В
        (ВЫБРАТЬ
            НоменклатураДокумента.Номенклатура
        ИЗ
            НоменклатураДокумента)) КАК ОстаткиМатериаловОстатки
    ПО НоменклатураДокумента.Номенклатура = ОстаткиМатериаловОстатки.Материал

```

Рисунок 86 – Текст запроса

Добавим изменения в текст процедуры. Изменения представлены на рисунках 87 и 88.

```

>>> Движения.СтоимостьМатериалов.Записать ();
>>> Движения.ОстаткиМатериалов.Записать ();

>>> Результат = Запрос2.Выполнить ();

```

Рисунок 87 – Изменения в тексте процедуры

```

>>> Если ВыборкаДетальныеЗаписи.Количество = 0 Тогда
    СтоимостьМатериала = 0;
Иначе
    СтоимостьМатериала = ВыборкаДетальныеЗаписи.Стоимость/ВыборкаДетальныеЗаписи.Количество;
>>> КонецЕсли;

```

Рисунок 88 – Изменения в тексте процедуры

Создадим «Запрос3» с параметрами, представленными на рисунке 89.

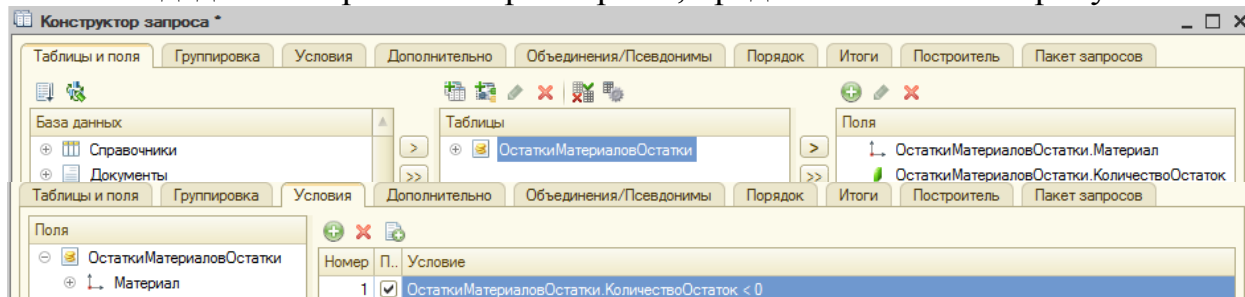


Рисунок 89 – Параметры «Запрос3»

Добавим в текст процедуры строки для проверки отрицательных остатков. Добавленные строки представлены на рисунке 90.

```
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
Сообщение = Новый СообщениеПользователю();
Сообщение.Текст = "Не хватает" + Строка(ВыборкаДетальныеЗаписи.КоличествоОстаток) + " единиц материала" + ВыборкаДетальныеЗаписи.Материал + " ";
Сообщение.Сообщить();
Отказ = Истина;
КонецЦикла;
```

Рисунок 90 – Добавленные строки

Для блокирования данных установим управляемую блокировку в значение «истина». Изменение значения представлено на рисунке 91.

```
//Установим необходимость блокировки данных в регистрах СтоимостьМатериалов и ОстаткиМатериалов
Движения.СтоимостьМатериалов.БлокироватьДляИзменения = Истина;
Движения.ОстаткиМатериалов.БлокироватьДляИзменения = Истина;
```

Рисунок 91 – Изменение значения управляемой блокировки

Вначале создадим справочник «ВариантыНоменклатуры» и в списке его владельцев укажем справочник «Номенклатура». Затем создадим справочник «ДополнительныеСвойстваНоменклатуры».

Создадим план видов характеристик «СвойстваНоменклатуры». Параметры плана видов характеристик представлены на рисунке 92.

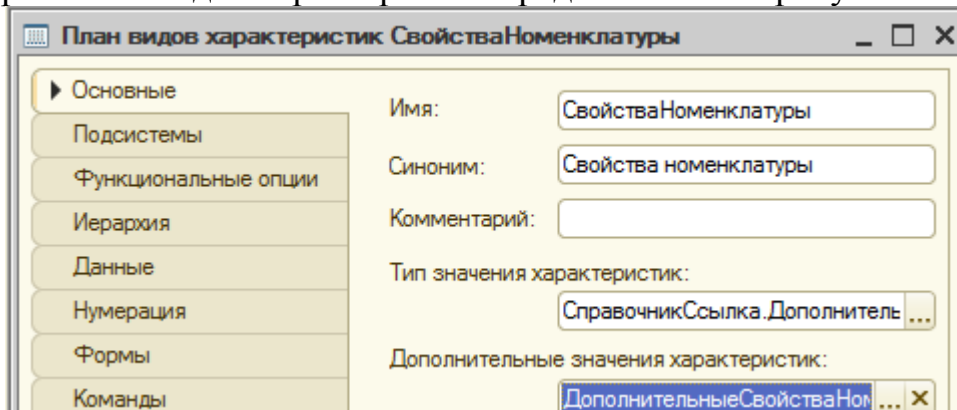


Рисунок 92 – Параметры плана видов характеристик

У справочника «ДополнительныеСвойстваНоменклатуры» укажем в качестве владельца план видов характеристик «СвойстваНоменклатуры».

Создадим регистр сведений «ЗначенияСвойствНоменклатуры». Заданные параметры представлены на рисунке 93.

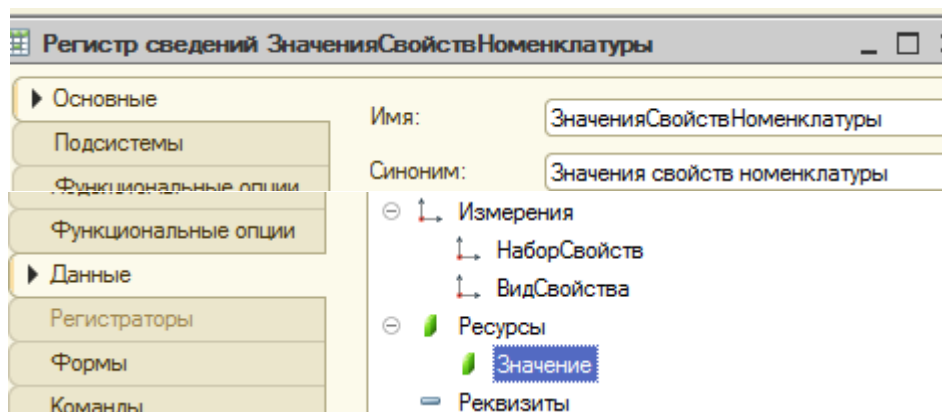


Рисунок 93 – Параметры регистра сведений

В справочнике «Номенклатура» заменим синоним стандартного реквизита «Родитель» на «Группа номенклатуры». Замена представлена на рисунке 94.

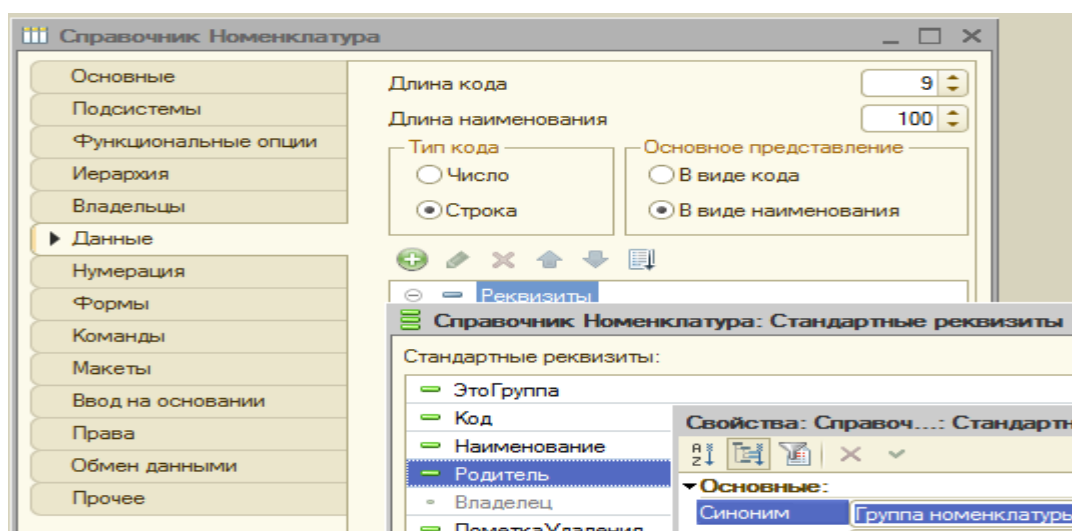


Рисунок 94 – Замена синонима реквизита

В справочнике «Варианты номенклатуры» создадим основную форму списка и в событии «ПриСозданииНаСервере» пропишем код представленный на рисунке 95.

```

НаСервере
Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)
    Если Параметры.Отбор.Свойство ("Владелец") Тогда
        Элементы.Код.Видимость = Ложь;
    КонецЕсли;
КонецПроцедуры

```

Рисунок 95 – Код процедуры «ПриСозданииНаСервере»

Добавим основное свойство «Представление объекта» как «Вариант номенклатуры».

Создадим основную форму элемента с параметрами, представленными на рисунке 96.

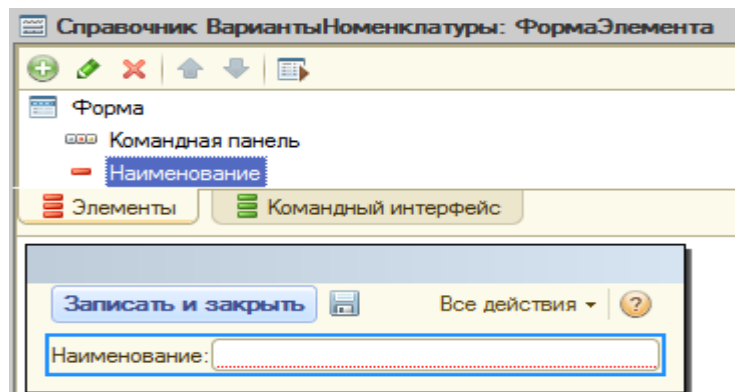


Рисунок 96 – Форма элемента

Заменяем синоним реквизита «Наименование» как «Название». Укажем в качестве представления списка регистра сведений «ЗначенияСвойстваНоменклатуры» «Состав варианта номенклатуры».

В регистре сведений «ЗначенияСвойствНоменклатуры» измерению «ВидСвойства» укажем синоним «Свойство».

Для того, чтобы убрать поле «Набор свойств» создадим форму списка и в процедуре «ПриСозданииНаСервере» пропишем код представленный на рисунке 97.

```

НаСервере
Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)
|   Если Параметры.Отбор.Свойство ("НаборСвойств") Тогда
|       Элементы.НаборСвойств.Видимость = Ложь;
|   КонецЕсли;
КонецПроцедуры

```

Рисунок 97 – Процедура «ПриСозданииНаСервере»

Далее в регистре сведений «ЗначенияСвойствНоменклатуры» зададим «Представление записи» как «Свойство и значение», а потом создадим основную форму записи и удалим поле «НаборСвойств».

Доработаем регистр «Остатки материалов». Создадим в нем новое измерение «НаборСвойств».

Доработаем документ «ПриходнаяНакладная». Добавим реквизит табличной части «НаборСвойств» и отобразим этот реквизит в форме документа. В конце добавим в тексте процедуры «ОбработкаПроведения» добавим строку, представленную на рисунке 98.

```

>>>| Движение.Материал = ТекСтрокаМатериалы.Материал;
|     Движение.НаборСвойств = ТекСтрокаМатериалы.НаборСвойств;
|     Движение.Склад = Склад;
|     Движение.Количество = ТекСтрокаМатериалы.Количество;

```

Рисунок 98 – Добавление строки в процедуру «ОбработкаПроведения»

Аналогичным образом доработаем документ «Оказание услуги». Отчет, использующий характеристики

Добавим новый отчет «ОстаткиМатериаловПоСвойствам». Заданные параметры представлены на рисунках 99 и 100.

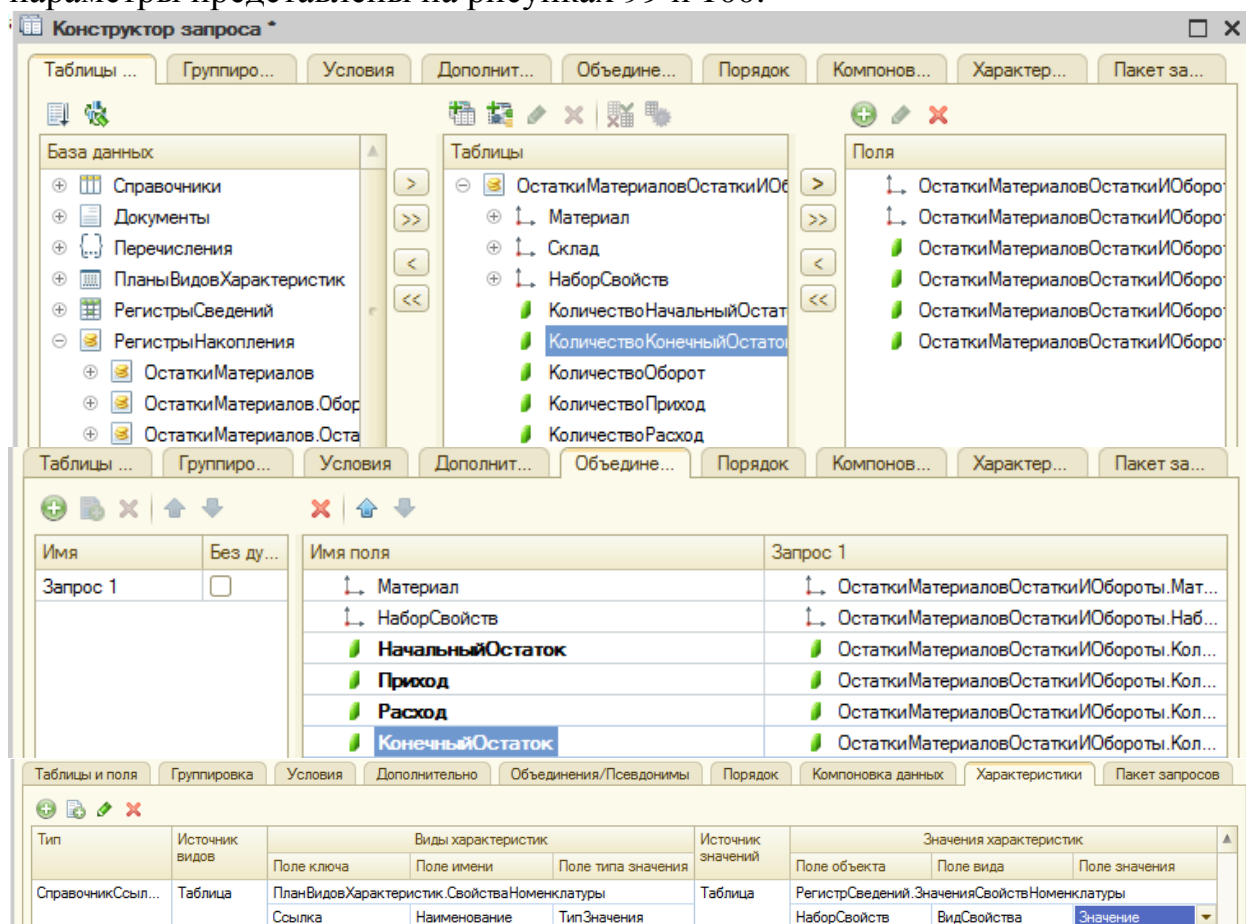


Рисунок 99 – Параметры отчета «ОстаткиМатериаловПоСвойствам»

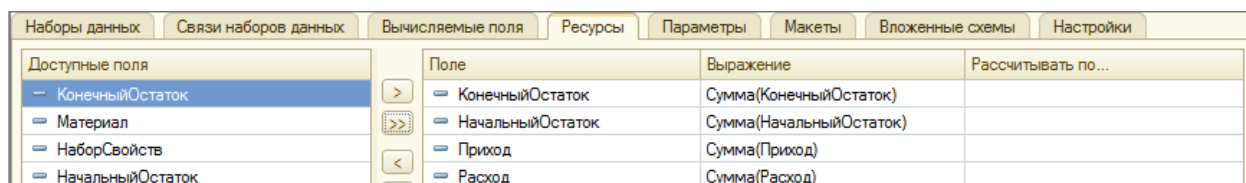


Рисунок 100 – Параметры отчета «ОстаткиМатериаловПоСвойствам»

Создадим план видов характеристик «ВидыСубконто».

Создадим справочник «Субконто» и в качестве владельца справочника укажем план видов характеристик «ВидыСубконто».

Параметры при создании плана видов характеристик «ВидыСубконто» представлены на рисунках 101 и 102.



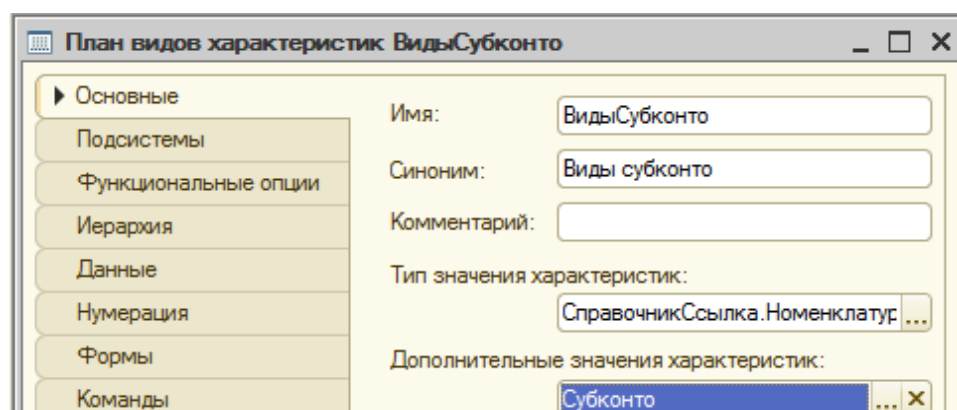


Рисунок 101 – параметры плана видов характеристик «ВидыСубконто»

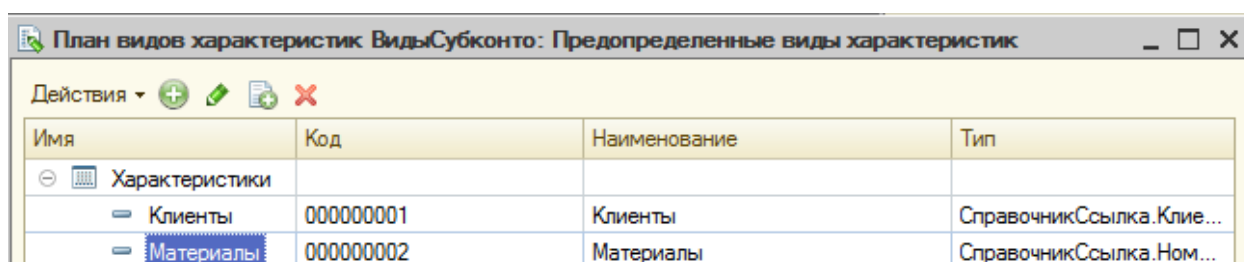


Рисунок 102 – параметры плана видов характеристик «ВидыСубконто»

Создадим план счетов «Основной». Заданные параметры при создании представлены на рисунках 103 – 106.

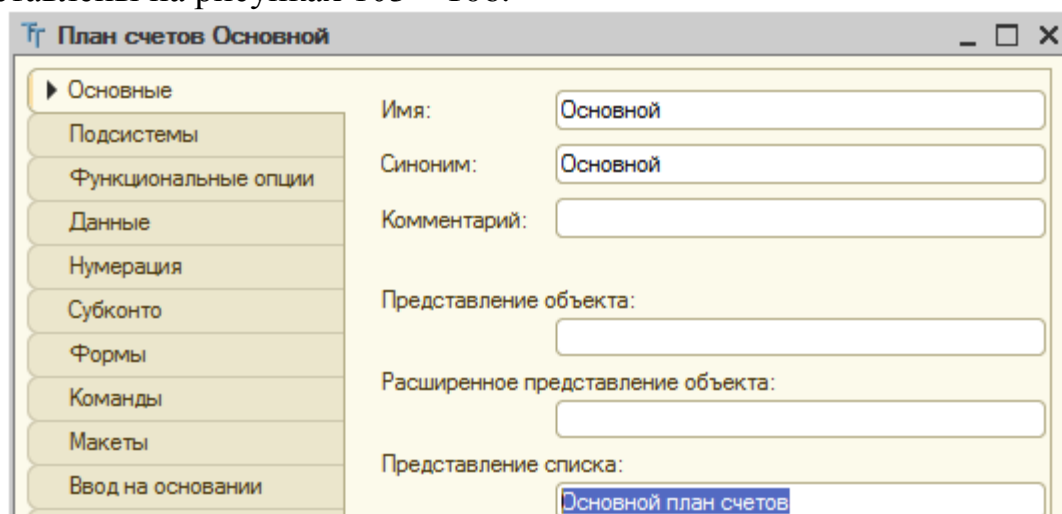


Рисунок 103 – Параметры плана счетов «Основной»

Рисунок 104 – Параметры плана счетов «Основной»

Рисунок 105 – Параметры плана счетов «Основной»

План счетов Основной: Предопределенные счета

Имя	Код	Наименование	Вид	Забалансо...	Порядок	Количеств...	Субконто 1	Субконто
Счета								
Товары	41	Товары	Активный			✓	Материалы	
Расчет...	60	Расчеты с поставщиками	Активный/...					
Дебит...	62	Дебиторская задолженность	Активный/...				Клиенты	
Капитал	90	Капитал	Активный/...					

Рисунок 106 – Параметры плана счетов «Основной»

Создадим регистр бухгалтерии «Управленческий». Заданные параметры представлены на рисунках 107 и 108.



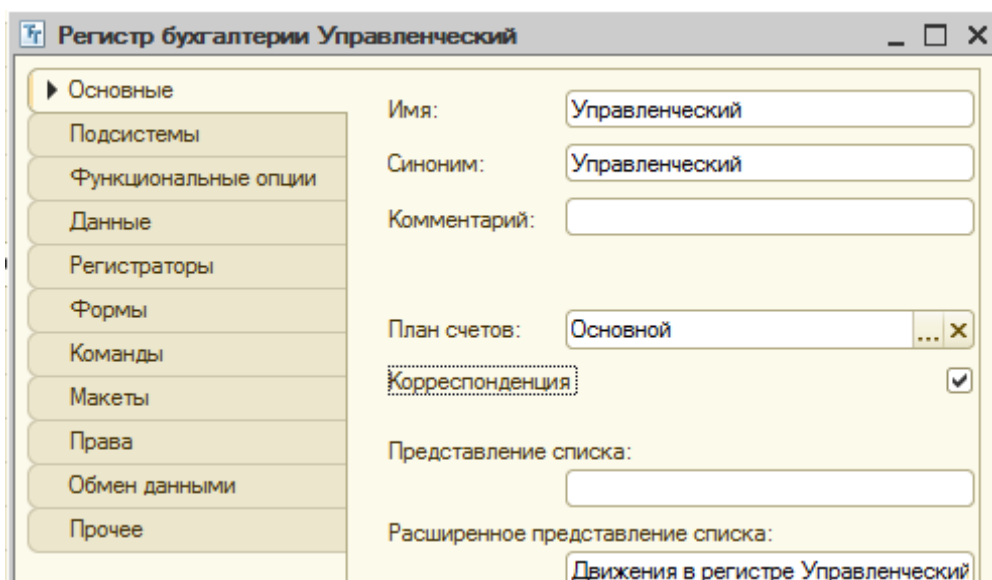


Рисунок 107 – Параметры регистра бухгалтерии «Управленческий»

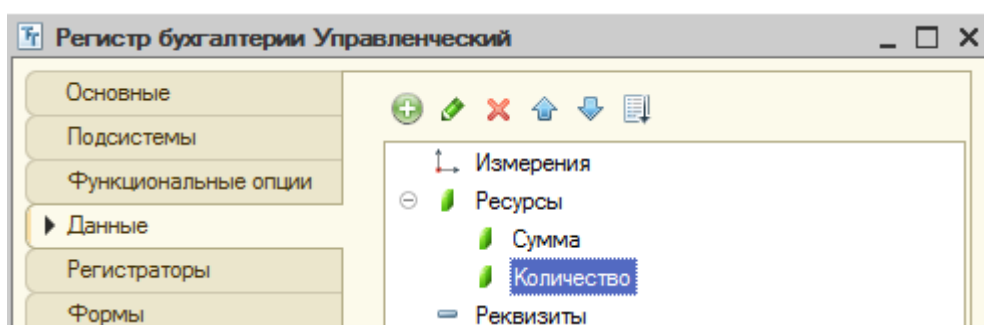


Рисунок 108 – Параметры регистра бухгалтерии «Управленческий»

Изменим процедуру проведения документа «Приходная накладная». После изменения процедура выглядит как представлено на рисунке 109.

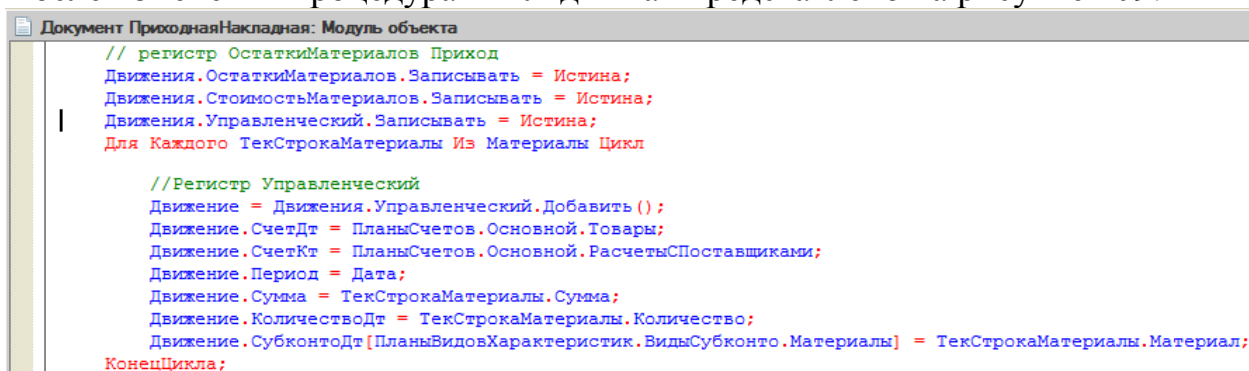


Рисунок 109 – Процедура проведения

Доработаем документ «Оказание услуги». В процедуру проведения документа добавим код представленный на рисунке 110.

```

//Регистр Управленческий
//Первая проводка: Д 62 (ДебиторскаяЗадолженность) - К 90 (Капитал) Розничная сумма
Движение = Движения.Управленческий.Добавить ();
Движение.СчетДт = ПланыСчетов.Основной.ДебиторскаяЗадолженность;
Движение.СчетКт = ПланыСчетов.Основной.Капитал;
Движение.Период = Дата;
Движение.Сумма = ВыборкаДетальныеЗаписи.СуммаВДокументе;
Движение.СубконтоДт [ПланыВидовХарактеристик.ВидыСубконто.Клиенты] = Клиент;
//Вторая проводка Д 90 (Капитал) - К 41 (Товары) - себестоимость
Движение = Движения.Управленческий.Добавить ();
Движение.СчетДт = ПланыСчетов.Основной.Капитал;
Движение.СчетКт = ПланыСчетов.Основной.Товары;
Движение.Период = Дата;
Движение.Сумма = СтоимостьМатериала * ВыборкаДетальныеЗаписи.КоличествоВДокументе;
Движение.КоличествоКт = ВыборкаДетальныеЗаписи.КоличествоВДокументе;
Движение.СубконтоКт [ПланыВидовХарактеристик.ВидыСубконто.Материалы] = ВыборкаДетальныеЗаписи.Номенклатура;

```

Рисунок 110 – Процедура проведения документа

Создадим оборотно-сальдовую ведомость. Построим запрос. Параметры запроса представлены на рисунке 111.

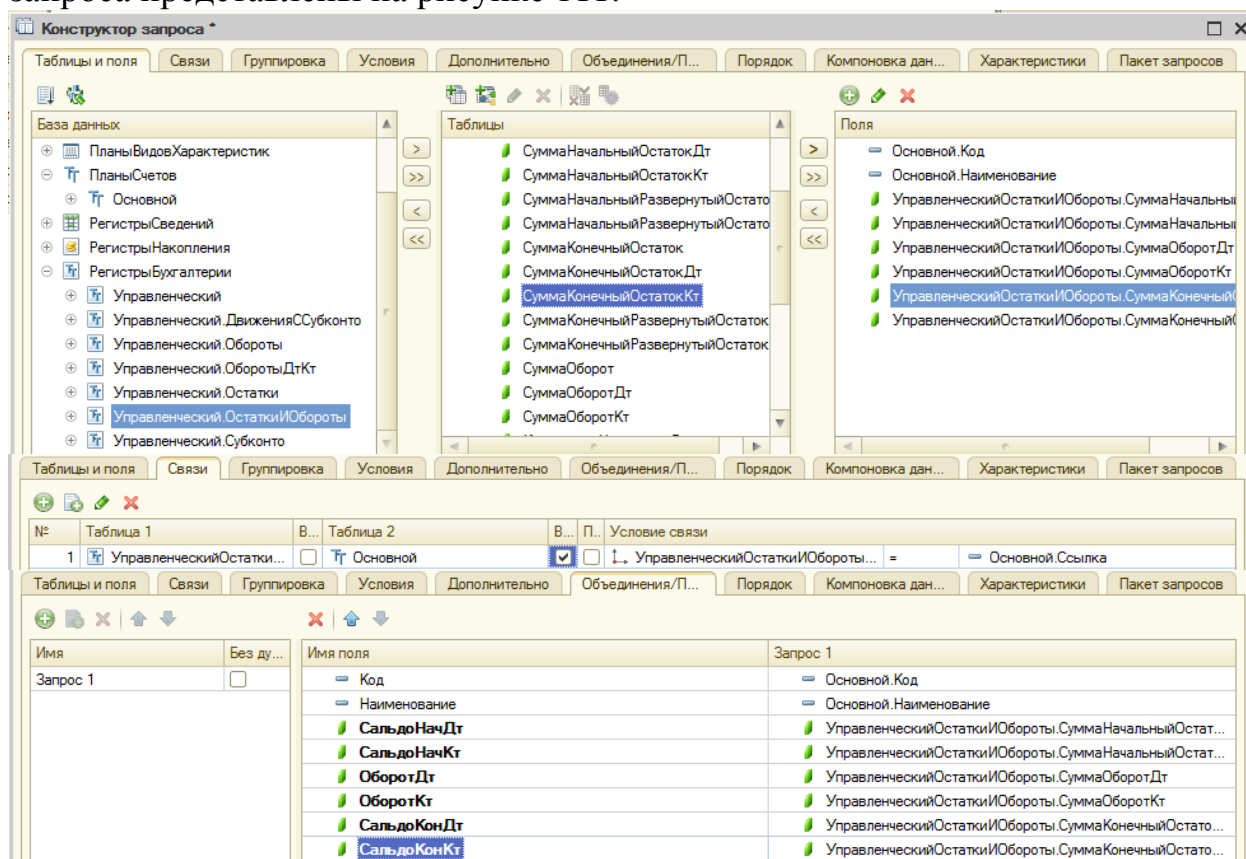


Рисунок 111 – Параметры оборотно-сальдовой ведомости

Добавим план видов расчета. Заданные при создании параметры представлены на рисунках 112.1 и 112.2.

Рисунок 112.1 – Параметры плана видов расчета

Имя	Код	Наименование
Невыход	Невыход	Невыход
Оклад	Оклад	Оклад
Премия	Премия	Премия

Рисунок 112.2 – Параметры плана видов расчета

Создадим новый справочник «ВидыГрафиковРаботы». Параметры справочника представлены на рисунке 113.

Имя	Код	Наименование
Элементы		
ГрафикАдминистрации	000000001	График администрации
ГрафикМастеров	000000002	График мастеров

Рисунок 113 – Параметры справочника «ВидыГрафиковРаботы»

Создадим регистр сведений «ГрафикиРаботы». Параметры регистра представлены на рисунке 114.

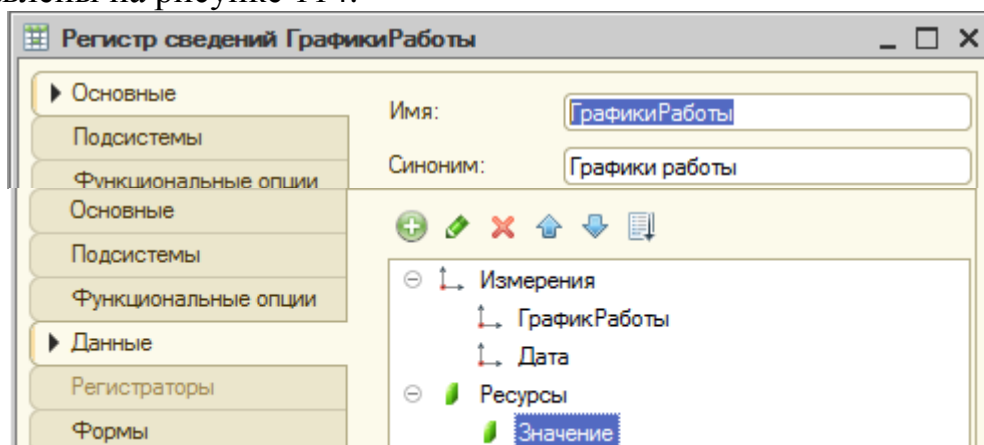


Рисунок 114 – Параметры регистра «ГрафикиРаботы»

Создадим регистр расчета «Начисления». Параметры представлены на рисунке 115.

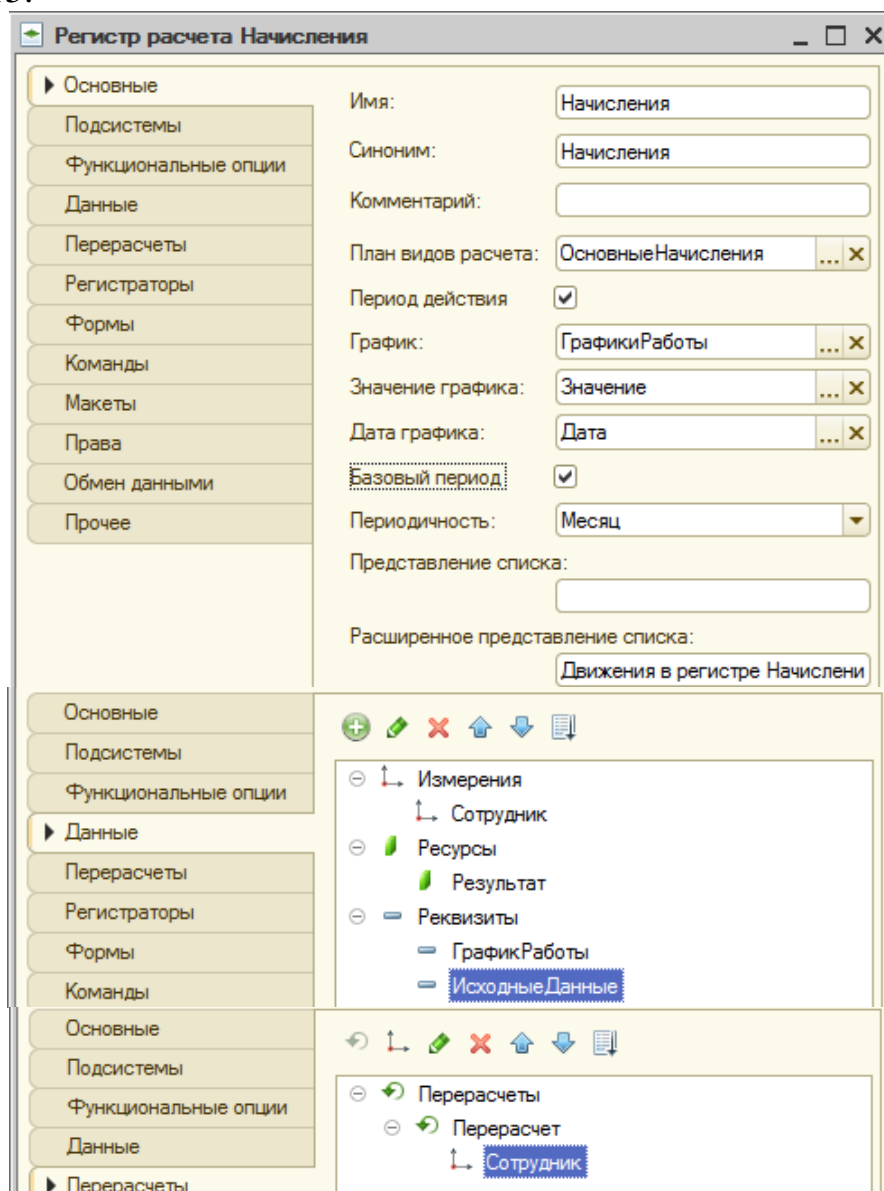


Рисунок 115 – Параметры регистра расчета «Начисления»

Создадим документ «НачисленияСотрудникам». Сконструируем движение. Движение представлено на рисунке 116.

```

1  Процедура ОбработкаПроведения(Отказ, Режим)
    ///{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
    // Данный фрагмент построен конструктором.
    // При повторном использовании конструктора, внесенные вручную изменения будут !

    // регистр Начисления
    Движения.Начислять = Истина;
    Для Каждого ТекСтрокаНачисления Из Начисления Цикл
        Движение = Движения.Начисления.Добавить();
        Движение.Сторно = Ложь;
        Движение.ВидРасчета = ТекСтрокаНачисления.ВидРасчета;
        Движение.ПериодДействияНачало = ТекСтрокаНачисления.ДатаНачала;
        Движение.ПериодДействияКонец = КонецДня(ТекСтрокаНачисления.ДатаОкончания);
        Движение.ПериодРегистрации = Дата;
        Движение.БазовыйПериодНачало = ТекСтрокаНачисления.ДатаНачала;
        Движение.БазовыйПериодКонец = КонецДня(ТекСтрокаНачисления.ДатаОкончания);
        Движение.Сотрудник = ТекСтрокаНачисления.Сотрудник;
        Движение.ГрафикРаботы = ТекСтрокаНачисления.ГрафикРаботы;
        Движение.ИсходныеДанные = ТекСтрокаНачисления.Начислено;
    КонецЦикла;

    ///}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

Рисунок 116 – Процедура проведения документа

Создадим отчет по перерасчетам. Параметры отчета представлены на рисунке 117.

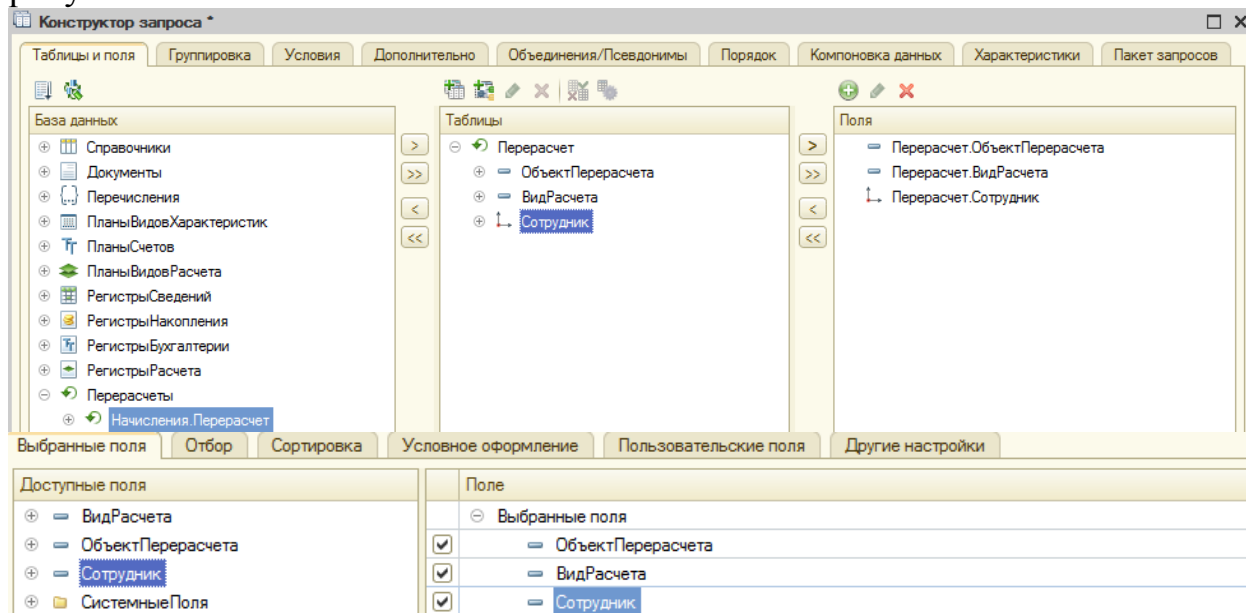


Рисунок 117 – Параметры отчета

Допишем процедуру проведения в документе «Начислениясотрудникам». Процедура представлена на рисунке 118.

```

//Записываем движение регистров.
Движения.Начисления.Записать();

//Получим список всех сотрудников, содержащихся в документе.
Запрос = Новый Запрос(
"ВЫБРАТЬ РАЗЛИЧНЫЕ
| НачисленияСотрудникамНачисления.Сотрудник
| ИЗ
| Документ.НачисленияСотрудникам.Начисления
| КАК НачисленияСотрудникамНачисления
|
| ГДЕ
| НачисленияСотрудникамНачисления.Ссылка = <ТекущийДокумент">;

Запрос.УстановитьПараметр("ТекущийДокумент", Ссылка);

//Сформируем список сотрудников
ТаблЗнач = Запрос.Выполнить().Выгрузить();
МассивСотрудников = ТаблЗнач.ВыгрузитьКолонку("Сотрудник");

//Вызов процедуры РассчитатьНачисления из общего модуля.
ПроведениеРасчетов.РассчитатьНачисления(Движения.Начисления, ПланыВидовРасчета.ОсновныеНачисления.Оклад, МассивСотрудников);
Движения.Начисления.Записать(, Истина);

```

Рисунок 118 – Процедура проведения документа

Создадим новый общий модуль «ПроведениеРасчетов». Параметры модуля представлены на рисунке 119.

Свойства: ПроведениеРасчетов

Основные:

Имя: ПроведениеРасчетов

Синоним: Проведение расчетов

Комментарий:

Модуль: Открыть

Глобальный: ☒

Клиент (управл): ☐

Сервер: ☒

Внешнее соеди: ☐

Вызов сервера: ☒

Рисунок 119 – Модуль «ПроведениеРасчетов»

В модуле напомним процедуру «РассчитатьНачисления», а также изменим процедуру «ОбработкаПроведения» в документе «НачисленияСотрудникам».

Создадим новый отчет «НачисленияСотрудникам». Параметры отчета представлены на рисунке 120.

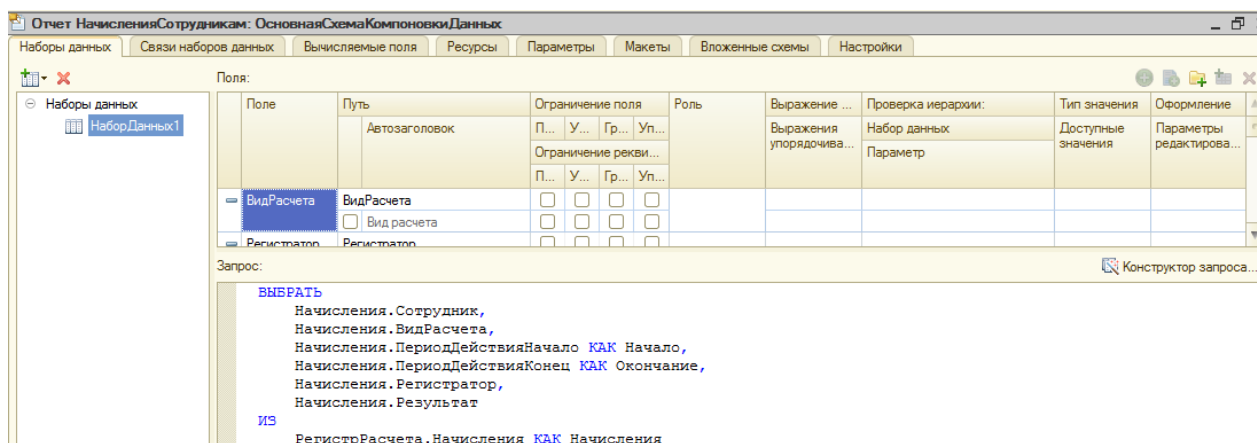


Рисунок 121 – Параметры отчета «НачисленияСотрудникам»

Создадим новый отчет «ДиаграммаГанта». Параметры отчета представлены на рисунке 122.

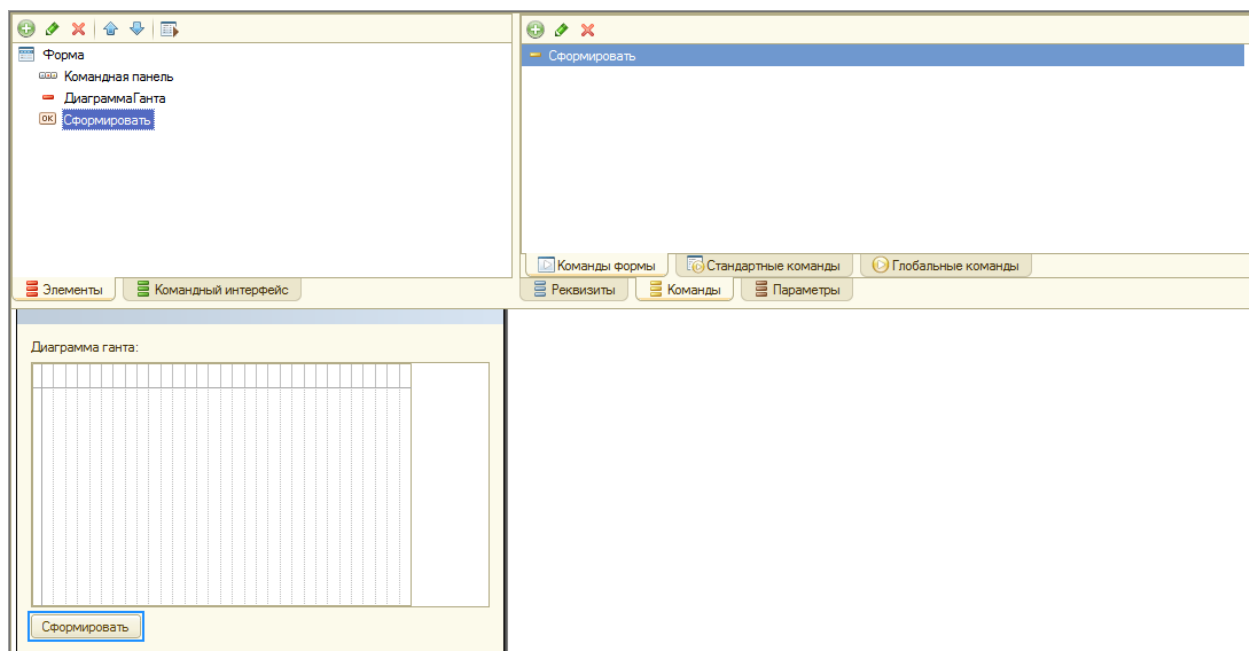


Рисунок 122 – Параметры отчета

Создадим документ «ВводНачальныхОстатковНоменклатуры». Параметры документа представлены на рисунке 123.

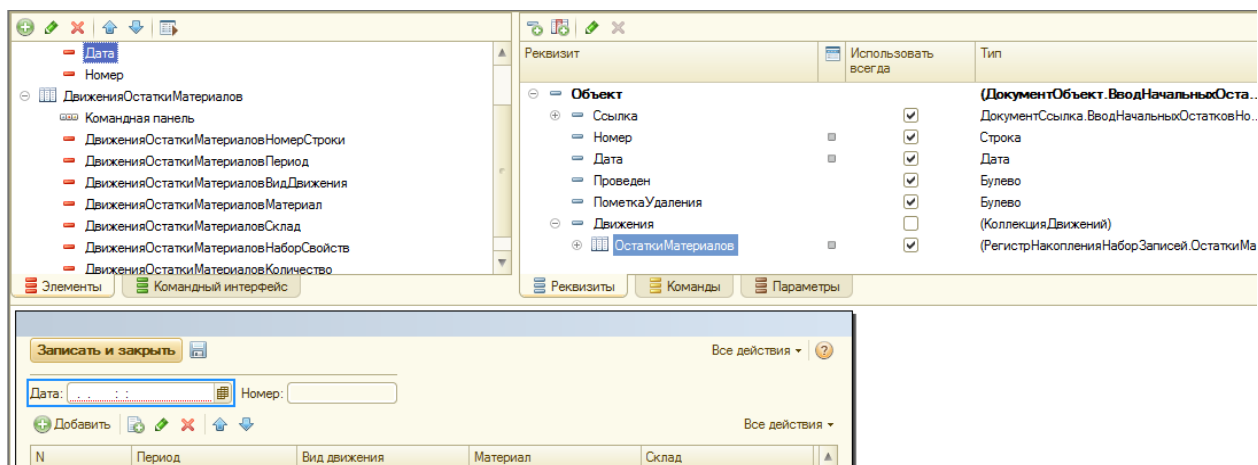


Рисунок 123 – Параметры документа

Изменим командный интерфейс подсистемы «УчетМатериалов». Изменения представлены на рисунке 124.

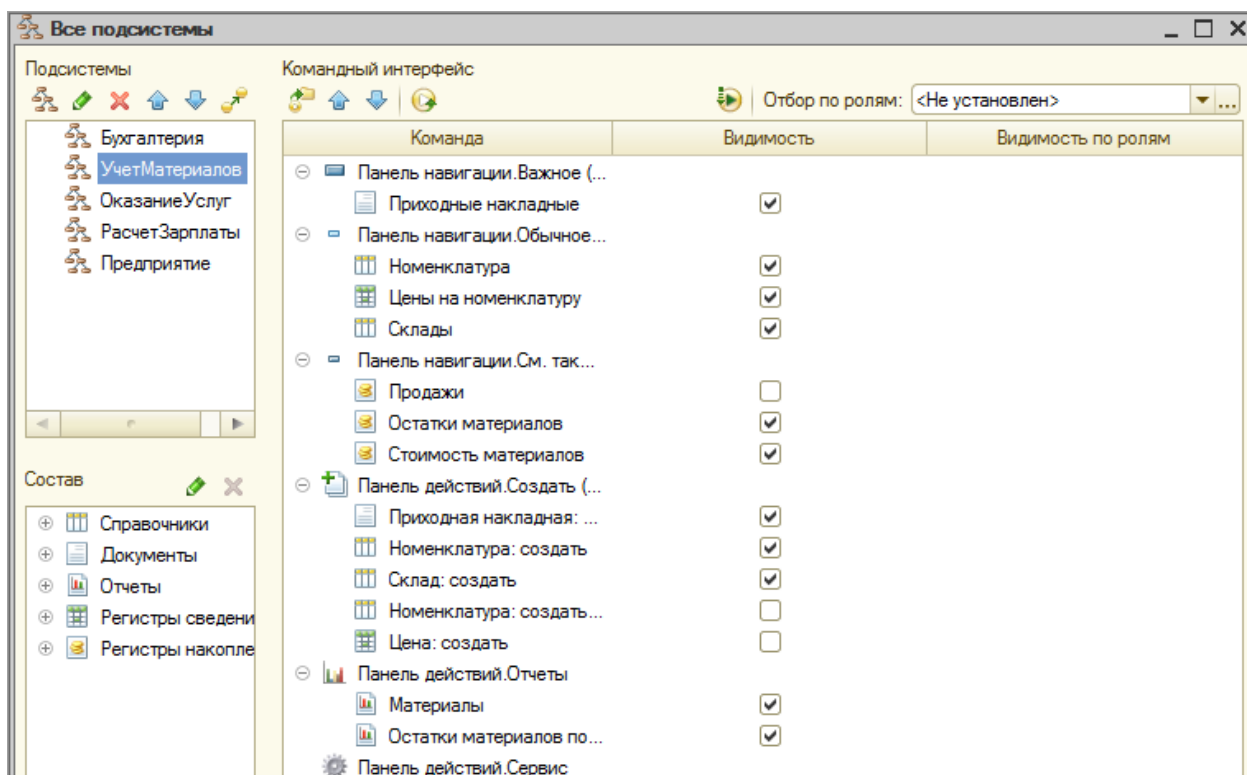


Рисунок 124 – Командный интерфейс подраздела «УчетМатериалов»

Таким же образом изменим командный интерфейс и в остальных подразделах.



Настроим рабочий стол. Добавим в левую колонку информацию об оказании услуг, а в правую информацию о приходных накладных и клиентах. Настройки рабочего стола представлены на рисунке 125.

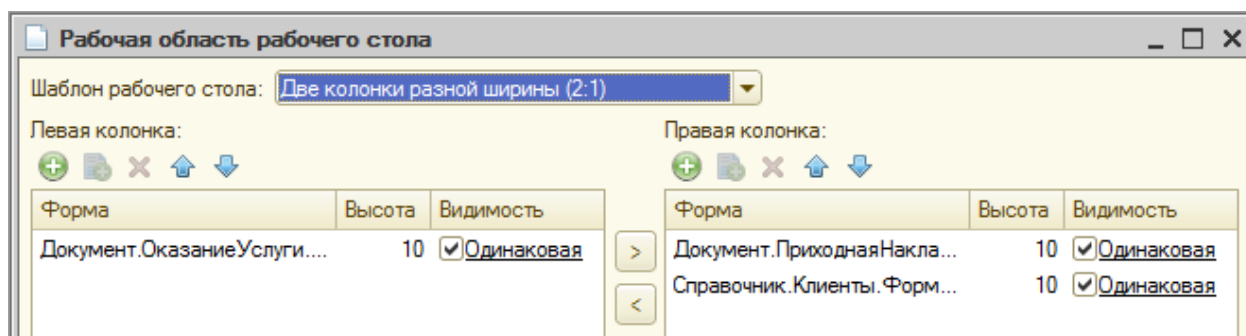


Рисунок 125 – Параметры рабочего стола

## ЗАКЛЮЧЕНИЕ

На данный момент 1С:Предприятие, а в точности 1С:Бухгалтерия является актуальным программным продуктом. В связи с развитием рыночной экономики, необходимо совершенствовать систему учета на предприятии. С помощью данного программного продукта можно оптимизировать ведение бухгалтерского учета на предприятии, подготавливать отчетность в электронном виде и выполнять другие процессы. Это экономит время и средства.

Целью курсовой работы является разработка автоматизированной системы предприятия, а именно разработка собственной конфигурации для автоматизации производства предприятия.

Основные задачи курсовой работы – изучить некоторые аспекты разработки конфигураций в 1С и создать собственную конфигурацию.

Созданная нами конфигурация содержит справочники (Клиенты, Сотрудники, Номенклатура, Склады, Варианты номенклатуры, Дополнительные свойства номенклатуры, Субконто, Виды графиков работы), документы (Приходная накладная, Оказание услуги, Начисления сотрудникам, Ввод начальных остатков номенклатуры), отчеты (Материалы, Реестр документов оказания услуг, Рейтинг услуг, Выручка мастеров, Перечень услуг, Рейтинг клиентов, Универсальный, Остатки материалов по свойствам, Оборотно-сальдовая ведомость, Перерасчет, Начисления сотрудникам, Диаграмма начислений), планы видов характеристик (Свойства номенклатуры, Виды субконто), план счетов (Основной), планы видов расчета (Основные начисления), регистры сведений (Цены, Значения свойств номенклатуры, Графики работы), регистры накопления (Остатки материалов, Стоимость материалов, Продажи), регистры бухгалтерии (Управленческий), регистры расчета (Начисления).

Таким образом, типовая конфигурация системы 1С обладает большими возможностями в плане настройки и оптимизации документооборота, однако,

для спецификации конфигурации для определенного предприятия разработчику приходится вносить изменения самому, а иногда и создавать конфигурацию с нуля.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Радченко М.Г., Е.Ю. Хрусталева Архитектура и работа с данными «1С:Предприятия 8.2».
- 3 1С:Предприятие 8. URL: [http://v8.1c.ru/overview/Term\\_000000035.htm](http://v8.1c.ru/overview/Term_000000035.htm).  
(Дата обращения: 10.12.17).
- 4 Что такое конфигурация 1С | ЭвриСофт  
URL: <http://evrysoft.ru/blog/article/26/>
- 5 Конфигурации 1С URL: <http://howknow1c.ru/kupit-1s/konfiguracii-1s.html>
- 6 Обзор типовых решений «1С»  
URL: [http://www.vnedriupp.ru/stat/?ELEMENT\\_ID=8295](http://www.vnedriupp.ru/stat/?ELEMENT_ID=8295)
- 7 Радченко М.Г., Хрусталева Е.Ю. 1С:Предприятие 8.2. Практическое пособие разработчика.