

**Министерство образования и молодёжной политики Свердловской области
Государственное автономное профессиональное образовательное учреждение
Свердловской области
«Белоярский многопрофильный техникум»
(ГАПОУ СО «БМТ»)**

**Методические указания к практическим работам
«Разработка прикладного решения средствами 1С: Предприятие 8.3»
ОП.12 «Программирование в 1С»
по специальности среднего профессионального образования
09.02.04 Информационные системы (по отраслям)**

Ладейщикова М.Н.

Разработка прикладного решения средствами 1С: Предприятие 8.3. Учебное пособие.

Учебное пособие по выполнению практических работ по ОП.12 «Программирование в 1С» предназначено для студентов средне специальных учебных заведений дневной и заочной форм обучения по специальности Информационные системы (по отраслям).

Цель данного пособия – формирование навыков программирования в среде «1С: Предприятие 8.3». В учебном пособие последовательно даны практические задания, в результате выполнения которых должна быть разработана автоматизированная информационная система готовая к внедрению на предприятии. Для закрепления навыков разработки экономических информационных систем и программирования в среде «1С: Предприятие 8.3» в заключительной части пособия представлены варианты заданий для самостоятельного выполнения.

Учебное пособие разработано на основе учебного пособия Карпухиной Н.Н. Экономические информационные системы: Разработка прикладного решения средствами 1С: Предприятие 8.2. Учебное пособие (лабораторный практикум). – М.: МГУПИ, 2012.

Содержание

Введение	4
Практическая часть	6
ЧАСТЬ 1. Разработка простого прикладного решения	6
I. Создание новой информационной базы	6
II. Создание справочников	11
III. Создание документов	15
IV. Регистры и отчеты	19
V. Ввод на основании	35
VI. Предопределенные элементы	39
VII. Рабочий стол и настройка командного интерфейса	40
ЧАСТЬ 2. Доработка приложения	53
I. Добавление подсистемы	53
II. Панель навигации и панель разделов	59
III. Проверка заполнения стандартных реквизитов в справочнике	60
IV. Иерархический справочник	61
V. Иерархический справочник с табличной частью и связь с документом	66
VI. Одна процедура для обработки нескольких событий	73
VII. Документ «Оказание услуги»	75
VIII. Периодический регистр сведений	77
IX. Автоматическая подстановка цены в документ при выборе номенклатуры	80
X. Перечисления	83
XI. Оборотные регистры накопления	84
XII. Отчет – выбор данных из двух таблиц	85
XIII. Видимость итоговой суммы в документах	99
Задание на диф.зачёт	103
Приложение А. Примерный перечень предметных областей АИС	106
Приложение Б. Анализ предметной области АИС. Проектирование АИС	108

Введение

Платформа «1С: Предприятие 8.3» поставляется со средством разработки, с помощью которого создаются новые или изменяются существующие прикладные решения. Это средство разработки называется «**конфигуратор**». Так как он включен в стандартную поставку 1С: Предприятия, то пользователь может самостоятельно разработать или модифицировать прикладное решение (адаптировать его под себя), возможно, с привлечением сторонних специалистов.

Конфигуратор включает следующие удобные инструменты:

- дерево конфигурации,
- окно свойств,
- различные редакторы (форм, интерфейсов, программных модулей и т.д.),
- конструкторы,
- отладчик,
- синтакс-помощник
- и другие инструменты.

Структура прикладного решения в конфигураторе создается **визуальными средствами** (буквально мышкой!). Разработчик создает различные объекты и настраивает их взаимосвязи друг с другом. С помощью визуальных редакторов создаются таблицы, экранные формы, макеты отчетов и печатных форм документов. Встроенный язык используется для описания различных алгоритмов, например, расчета налогов, исчисления себестоимости, алгоритмов проведения документов и формирования отчетов.

В лабораторной работе приведем **пример разработки прикладного решения**. Это поможет понять суть разработки и получить навыки для дальнейших самостоятельных действий. За небольшое время можно получить готовую к использованию систему, облегчающую работу пользователей и помогающую управлению бизнесом.

Создание объектов и настройка взаимосвязей между ними производится визуальными средствами. Ниже приводится общая схема данного прикладного решения (конфигурации) (рис.1).

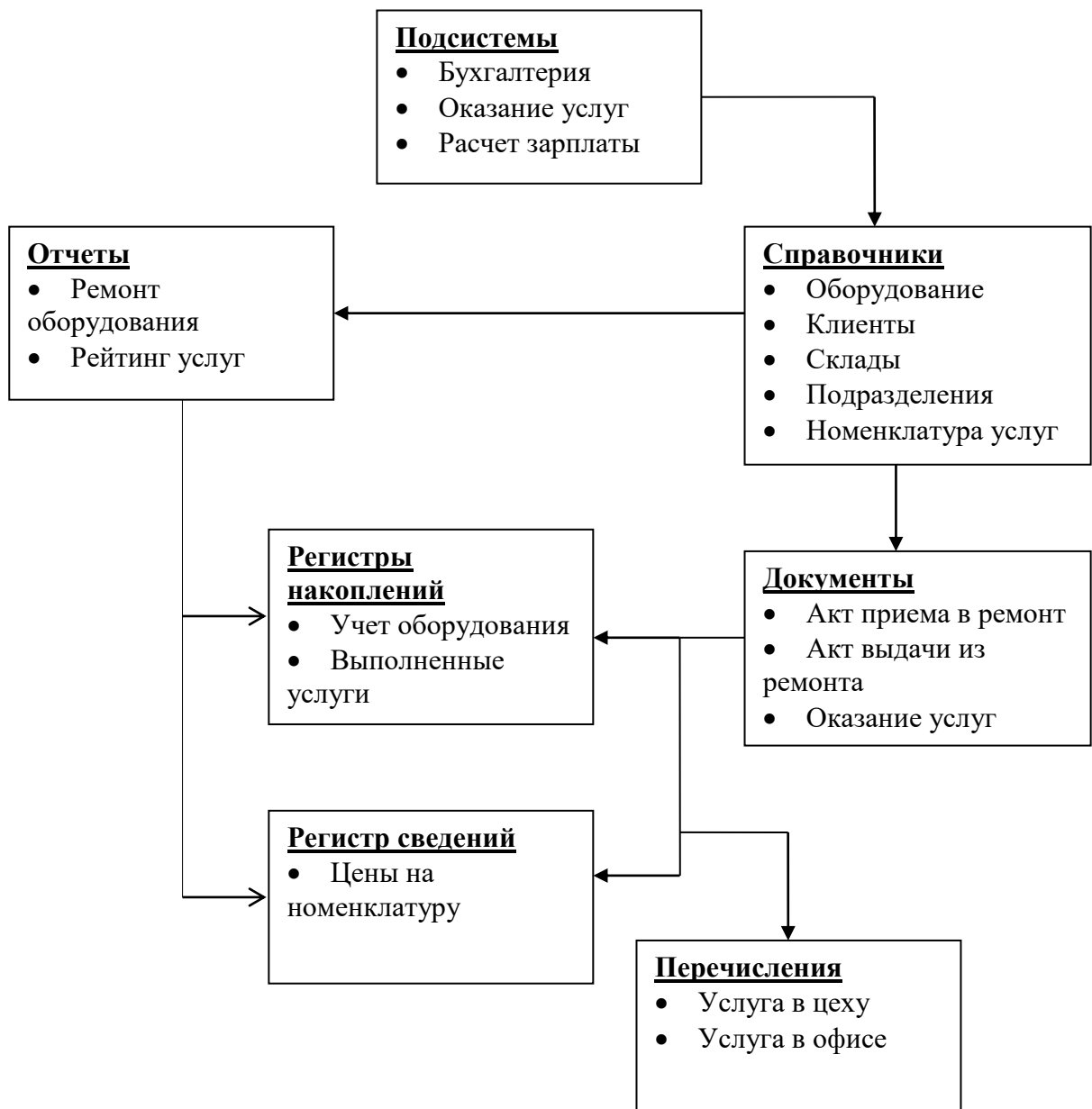


Рис.1. Общая схема будущей программы

Все эти объекты создаются с помощью визуальных средств.

Практическая часть

ЧАСТЬ 1. Разработка простого прикладного решения

I. СОЗДАНИЕ НОВОЙ ИНФОРМАЦИОННОЙ БАЗЫ

Прежде чем начинать работу необходимо создать новую информационную базу.

1. Создать на рабочем столе новую папку «ФИО студента».
2. Запустите программу 1С: Предприятие 8.2 (Учебная версия). На экране появится окно Запуск 1С: Предприятие (рис. 2).

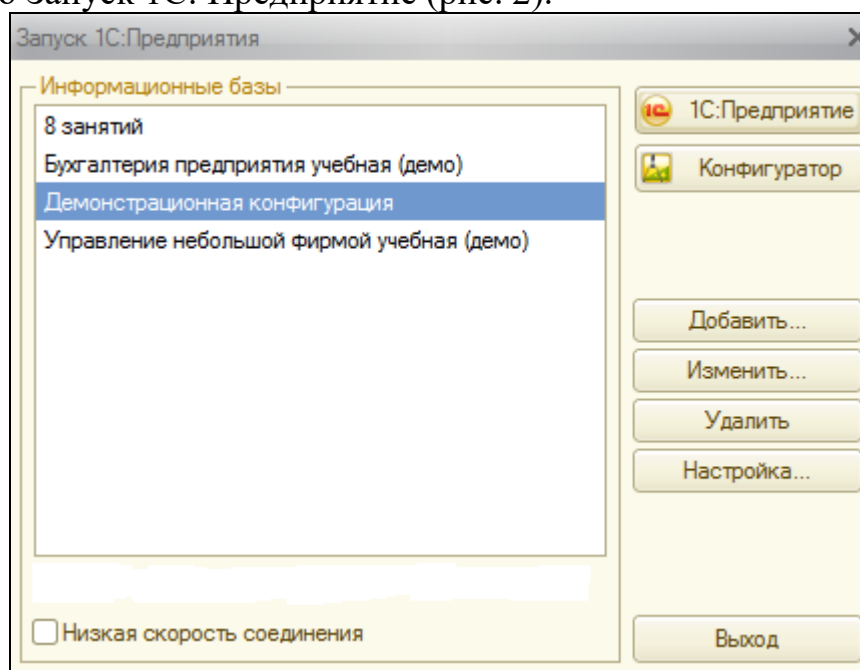


Рис.2. Запуск 1С: Предприятие

3. Нажмите на кнопку **Добавить** новую базу.
На формах (рис. 3-5а) введите следующие данные:

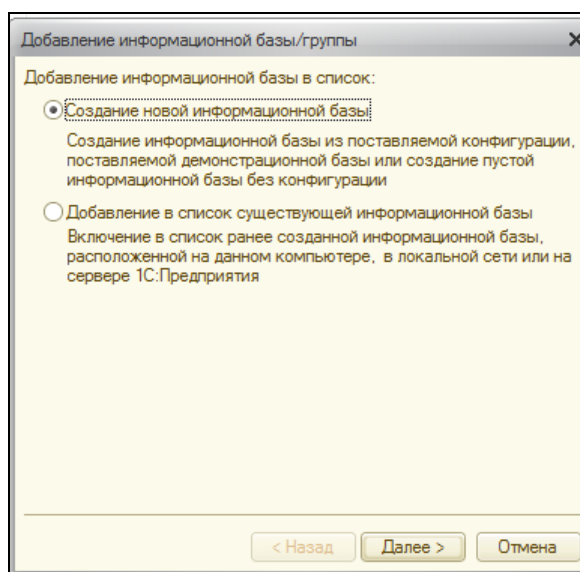


Рис. 3.Добавление новой информационной базы

Создаем информационную базу без конфигурации (рис. 4).

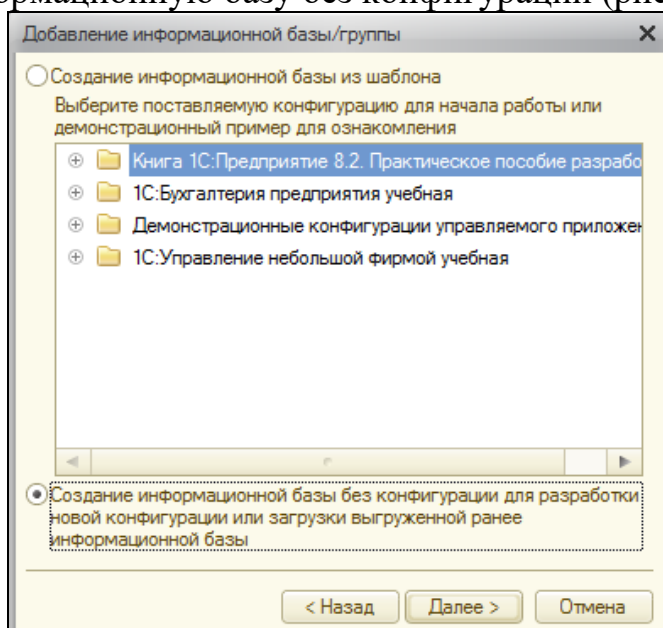


Рис. 4. Добавление новой информационной базы

В поле Укажите наименование информационной базы назовите имя новой базы Ремонт насосов (рис. 4а).

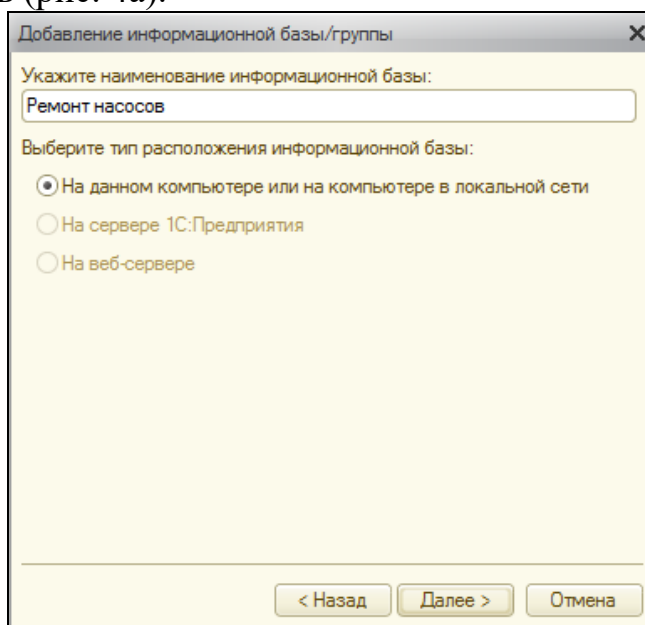


Рис. 4а. Добавление наименование новой информационной базы

В поле Каталог информационной базы укажите ту папку, которая была создана в п.1.

Добавление информационной базы/группы

Укажите параметры информационной базы:

Каталог информационной базы:

D:\ФИОСтудента

Язык (Страна):

русский (Россия)

< Назад Далее > Отмена

Рис. 5. Добавление новой информационной базы

Добавление информационной базы/группы

Укажите параметры запуска:

Вариант аутентификации (определения пользователя):

☒ Выбирать автоматически

☐ Запрашивать имя и пароль

Скорость соединения:

Обычная

Дополнительные параметры запуска:

Основной режим запуска:

☒ Выбирать автоматически

☐ Тонкий клиент

☐ Веб-клиент

☐ Толстый клиент

Версия ТС:Предприятия:

8.2

< Назад Готово Отмена

Рис. 5а. Добавление новой информационной базы

4. Выберите только что созданную информационную базу и нажмите на кнопку Конфигуратор (рис. 6).

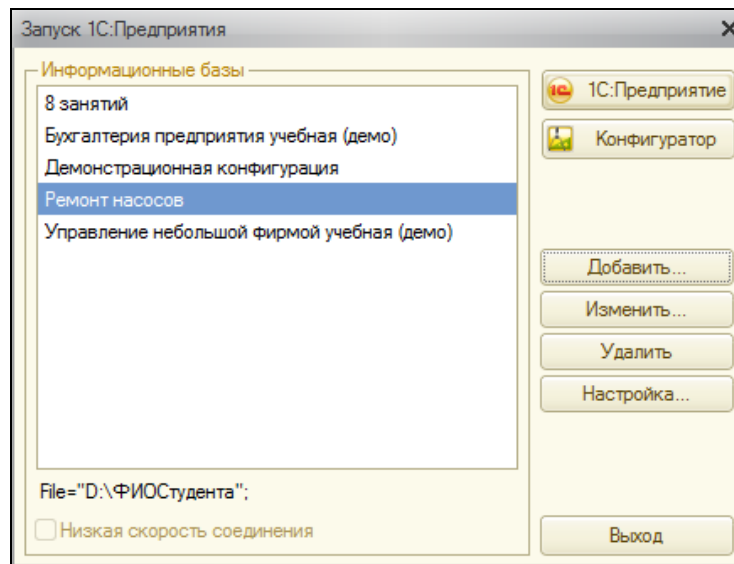



Рис. 6. Запуск 1С: Предприятие

Таким образом, была создана новая пустая информационная база, наполнением которой произведем далее.

После запуска Конфигуратора и создания новой информационной базы нужно открыть дерево конфигурации при помощи кнопки на панели инструментов  (рис.7). Можно начать создание объектов конфигурации.

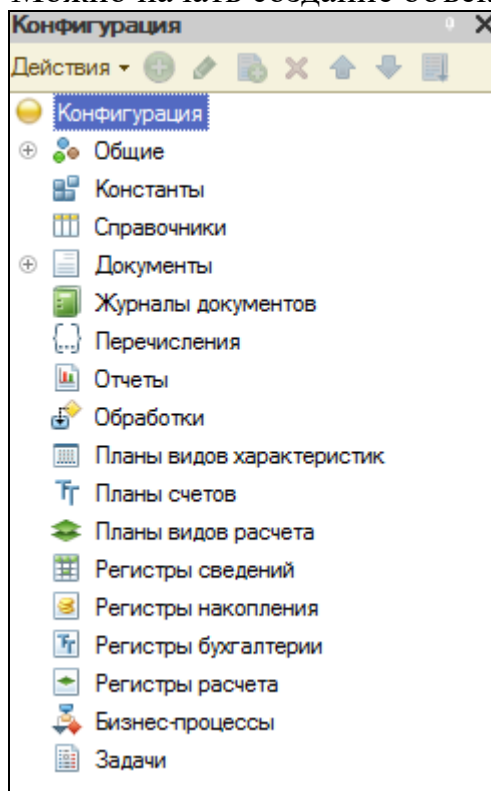


Рис. 7. Окно «Конфигурация»

Конфигурация представляет собой дерево метаданных. Метаданные описывают все аспекты работы приложения. Они представляют собой структурированное декларативное его описание. Метаданные образуют иерархию объектов, из которых формируются все составные части конфигурации и которые определяют все аспекты ее поведения. Фактически, при

работе бизнес-приложения платформа «проигрывает» (интерпретирует) метаданные, обеспечивая всю необходимую функциональность.

Метаданными описываются структуры данных, состав типов, связи между объектами, особенности их поведения и визуального представления, система разграничения прав доступа, пользовательский интерфейс и т.д. В метаданных, фактически, сосредоточены сведения не только о том, «что хранить в базе данных», но и о том, «зачем» хранится та или иная информация, какова ее роль в системе и как связаны между собой информационные массивы. Использование языка программирования ограничено в основном решением тех задач, которые действительно требуют алгоритмического описания, например, расчета налогов, проверки корректности введенных данных.

Дадим конфигурации Имя. На корне Конфигурации (корень дерева конфигурации) щелкаем левой кнопкой мыши. Появляется контекстное меню (рис. 8), выбираем Свойства и задаем название Конфигурации «Ремонт оборудования» (рис. 9).

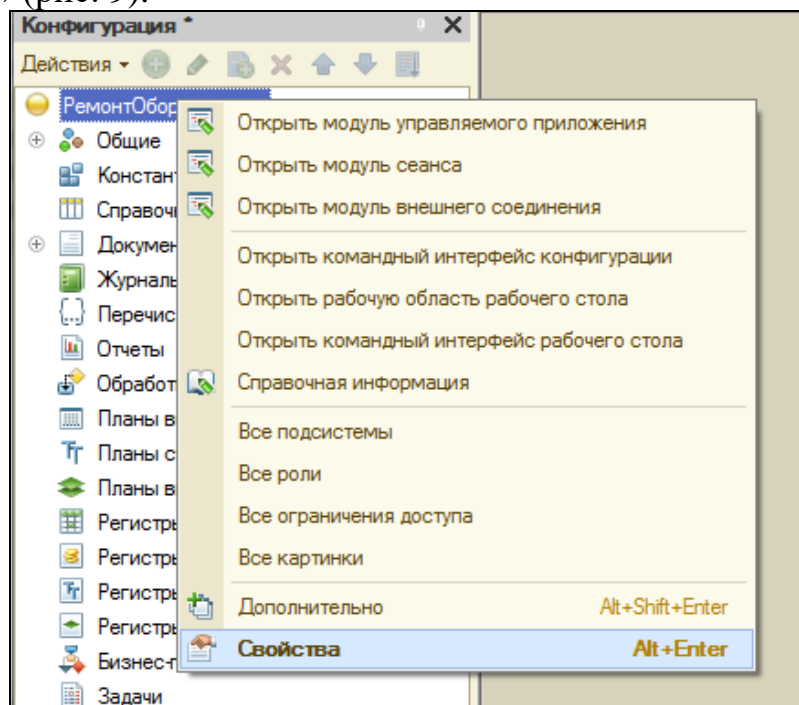


Рис. 8. Контекстное меню корня конфигурации

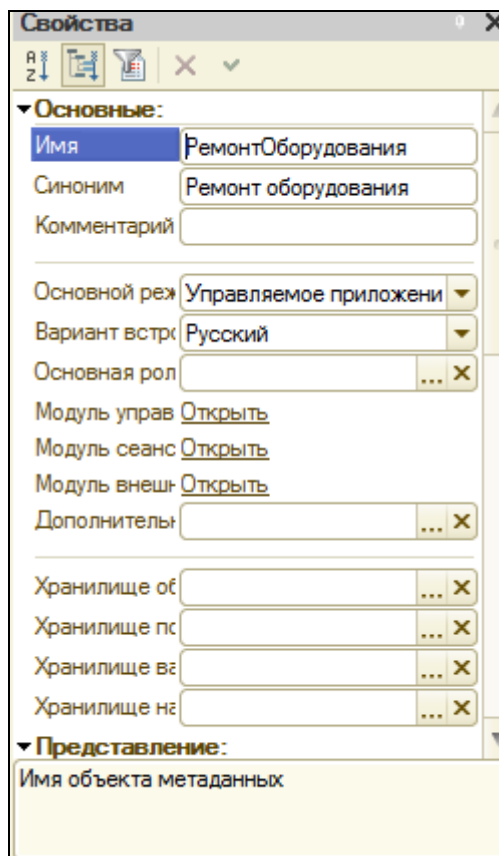


Рис. 9. Окно Свойства корня конфигурации

Задайте название новой конфигурации **РемонтОборудования**.

Автоматически сформировался синоним названия конфигурации **Ремонт оборудования**, который будет показан в заголовке основного окна в пользовательском режиме.

Проверим в работе выполненные выше действия, запустим приложение в режиме 1С: Предприятие. Выполним команду Сервис – 1С: Предприятие. Сохраним изменения. Запустится 1С: Предприятие.

Приступим к созданию объектов метаданных типа Справочник.

II. СОЗДАНИЕ СПРАВОЧНИКОВ

Любой справочник в системе «1С:Предприятие» имеет встроенные реквизиты «Код» и «Наименование». Встроенный реквизит «Код» будет присваиваться автоматически системой.

Создание справочника «Оборудование»

Данный справочник будет хранить информацию по оборудованию, которое сдается в ремонт клиентами.

1. Правой кнопкой мыши щелкнем по **Справочники** в конфигураторе (рис. 10) и выберем **Добавить**.

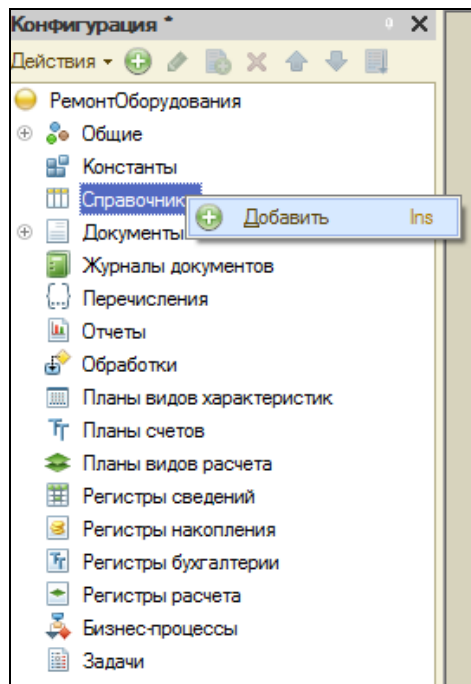


Рис. 10. Добавление нового Справочника

2. На экране появится форма **Конструктор справочника** и **Свойства**. Заполните их следующими данными (рис. 11):

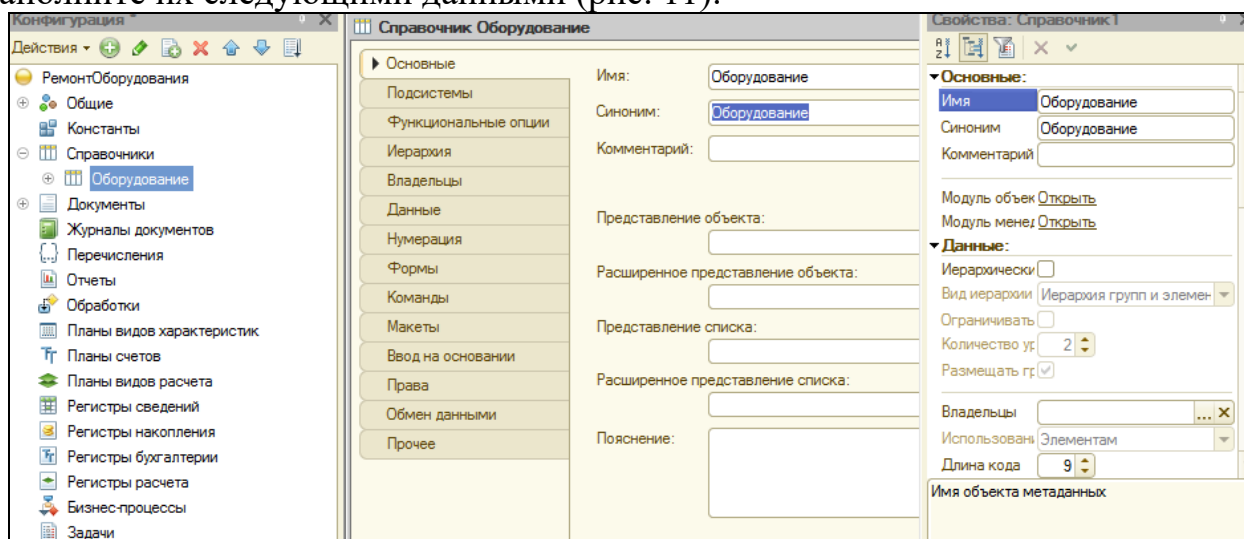


Рис. 11. Конструктор Справочника и окно Свойства

Заполните Имя – **Оборудование**.

3. На вкладке **Данные** зададим **Длину наименования** – **100** символов (рис. 12).

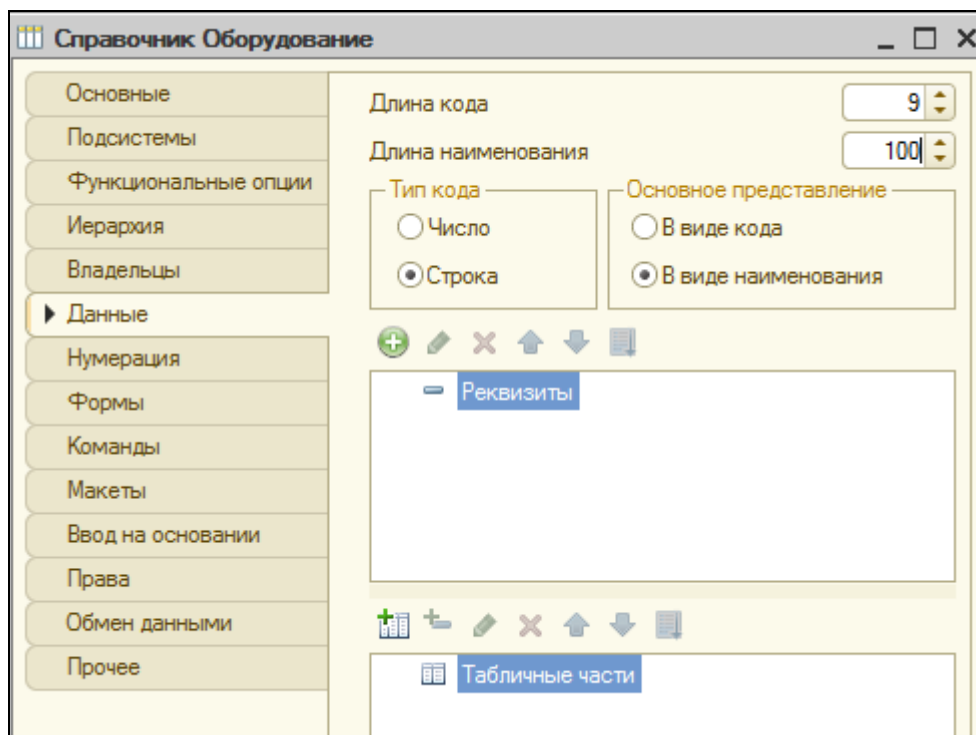


Рис. 12. Конструктор Справочника и вкладка Данные

Остальная стандартная реализация этого объекта полностью удовлетворяет заданию, поэтому все остальные параметры оставим без изменения. (Внимательно изучите данные формы).

Закройте форму (рис. 12).

Создание справочника «Клиенты»

Создадим новый справочник (рис. 13): имя – Клиенты.

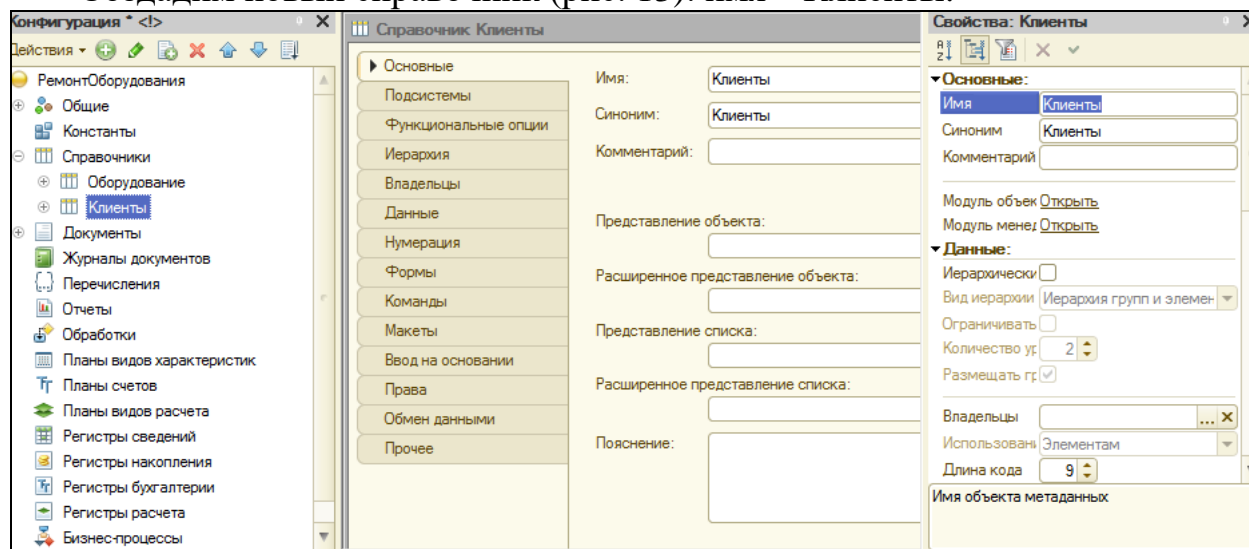


Рис. 13. Справочник Клиенты

Стандартная реализация этого объекта полностью удовлетворяет заданию, поэтому все остальные параметры оставим без изменения.

Закройте форму (рис. 13).

Создание справочника «Склады»

Ремонтная фирма, кроме центрального офиса, имеет еще филиал, расположенный в регионе. Поэтому необходимо знать, на какой склад приняли насос в ремонт и с какого склада клиент будет забирать оборудование после ремонта. Для этого создадим еще справочник Склады (рис. 14).

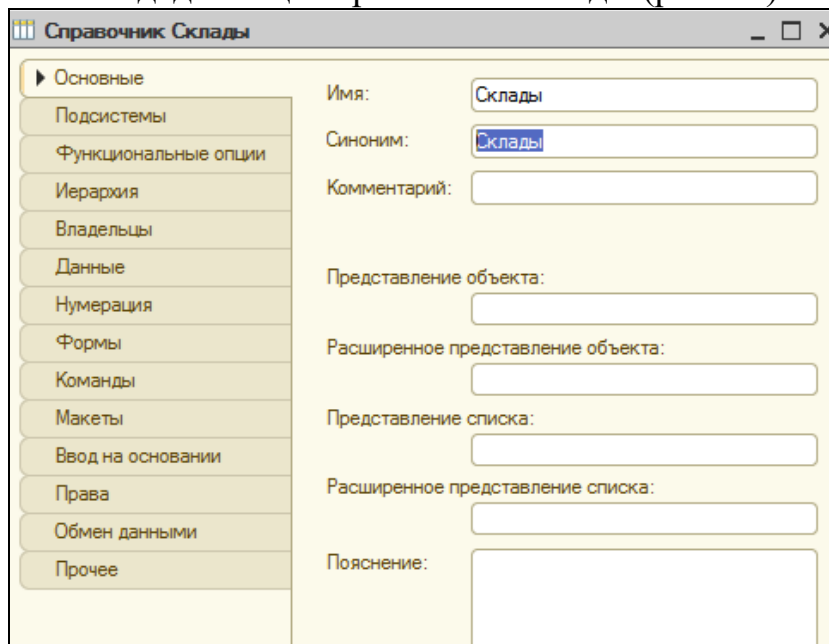


Рис. 14. Справочник Склады

Совершенно точно известно, что центральный офис существовал и будет далее существовать, потому что без него вся работа теряет смысл. А вот филиалы могут открываться и закрываться.

1. Добавим элемент **Главный склад** в справочник **Склады**. Элементы, добавляемые в конфигураторе, называются предопределенные.

Если обычные элементы пользователь может добавлять и удалять по своему желанию, то предопределенные элементы пользователь удалить не может. Эти элементы существуют всегда. Благодаря этому на них могут быть завязаны какие-либо программные алгоритмы. (Эти возможности рассмотрим позже).

2. Переходим на закладку **Прочее** (рис. 15) и нажмем кнопку **Предопределенные**.

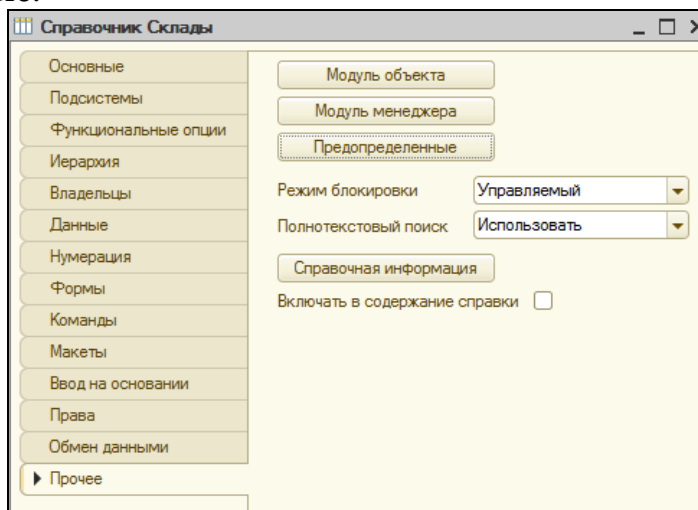



Рис. 15. Справочник Склады закладка Прочее

3. Добавим  новый предопределенный элемент **ОсновнойСклад** (рис. 16, 16а).

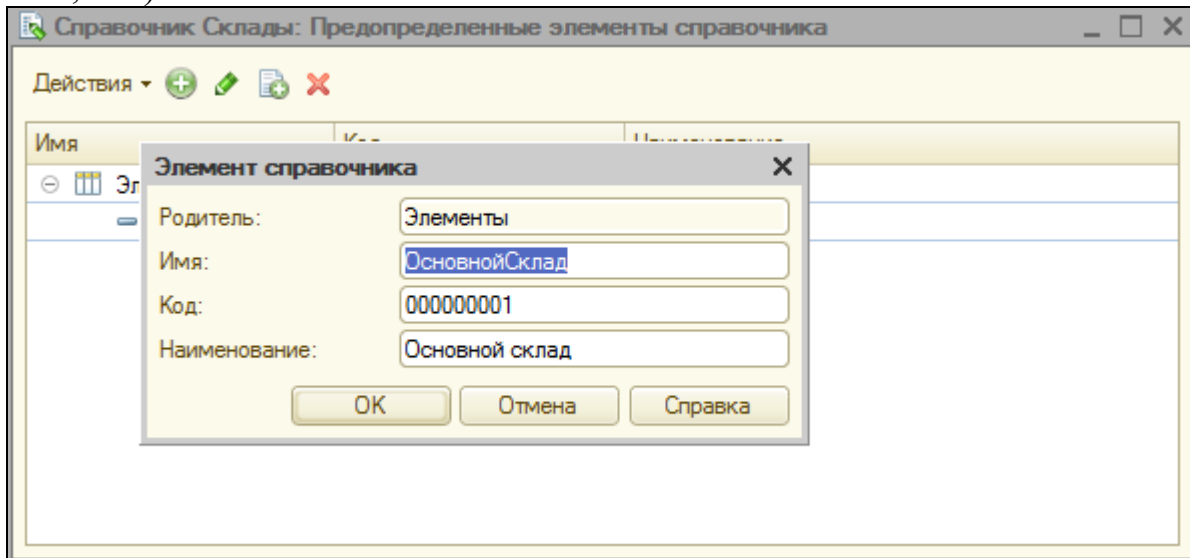


Рис. 16. Предопределенные элементы справочника

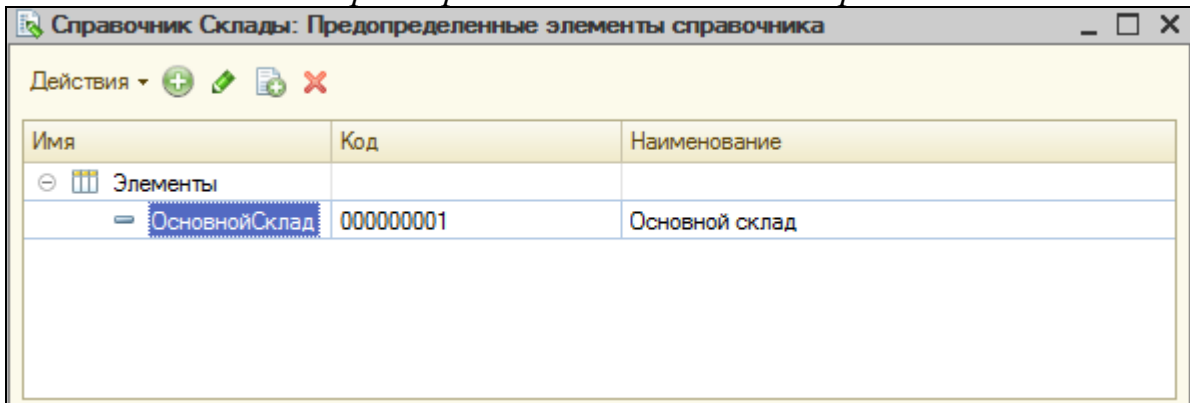


Рис. 16а. Предопределенные элементы справочника

III. СОЗДАНИЕ ДОКУМЕНТОВ

Нужно как-то фиксировать выполняемые менеджером действия:

1. принять насос в ремонт,
2. выдать клиенту отремонтированный насос.

Создадим два документа:

1. Акт приема в ремонт.
2. Акт выдачи из ремонта.

1. На закладке Документы в Дереве конфигуратора добавим документ **АктПриемаВРемонт** (рис. 17).

Рис. 17. Документ Акт приема в ремонт

2. Что необходимо знать, когда оборудование принимается в ремонт:

1. кто клиент,
2. какое оборудование,
3. на какой склад,
4. в чем неисправность.

Для этого переходим на закладку **Данные**. Добавляем новый реквизит – **Клиент**, тип данных – **Справочник.Ссылка.Клиенты**. Для этого нажмем на кнопку **Добавить** и введем необходимые данные(рис. 18, 18а).

Рис. 18. Документ Акт приема в ремонт закладка Данные

Рис. 18а. Документ Акт приема в ремонт закладка Данные Свойства реквизита Клиент

3. Аналогично создайте еще три новых **Реквизита** документа:

- Имя – **Оборудование**, Тип данных –

Справочник.Ссылка.Оборудование

- Имя – **Склад**, Тип данных – **Справочник.Ссылка.Склады**

- Имя – **ОписаниеНеисправности**, Тип данных – **Строка**, Длина – 250.

Чтобы удобнее было записывать, укажем **Многострочный режим**, **Расширенное редактирование**. (Это значит, что в форме этот реквизит будет отображаться как поле из нескольких строк. При нажатии кнопки Ввод будет происходить переход к следующей строке этого поля) (рис. 19а). Этот реквизит необходим, чтобы со слов клиента зафиксировать неисправность.

Свойства: ОписаниеНеисправно...

Основные:

Имя: ОписаниеНеисправности

Синоним: Описание неисправности

Комментарий:

Тип: Строка

Длина: 250

Допустимая дл: Переменная

Неограниченн: ☐

Использование:

Индексировать: Не индексировать

Полнотекстов: Использовать

Представление:

Режим пароля: ☐

Подсказка:

Маска:

Многострочный: ☒

Расширенное р: ☒

Рис. 19а. Документ Акт приема в ремонт закладка Данные Свойства реквизита ОписаниеНеисправности

На рис. 19б представлены созданные реквизиты Документа АктПриемаВРемонт.

Документ АктПриемаВРемонт

Основное меню:

- Основные
- Подсистемы
- Функциональные опции
- Данные**
- Нумерация
- Движения
- Последовательности
- Журналы
- Формы
- Команды
- Макеты
- Ввод на основании
- Права
- Обмен данными
- Прочее

Действия:

Реквизиты:

- Клиент
- Оборудование
- Склад
- ОписаниеНеисправности**

Табличные части:

Стандартные реквизиты

Рис. 19б. Документ Акт приема в ремонт закладка Данные

4. Создадим второй документ **Акт выдачи из ремонта**. В основном он будет аналогичен предыдущему. Необходимо указать клиента, оборудование, склад и какие ремонтные работы были выполнены. Т.е. те же реквизиты, что в Акте приема в ремонт. Для быстрого получения нового объекта воспользуемся копированием. Для копирования и вставки документа в конфигурацию используем стандартные клавиатурные команды.

Присвоим новому скопированному документу имя – **АктВыдачиИзРемонта**.

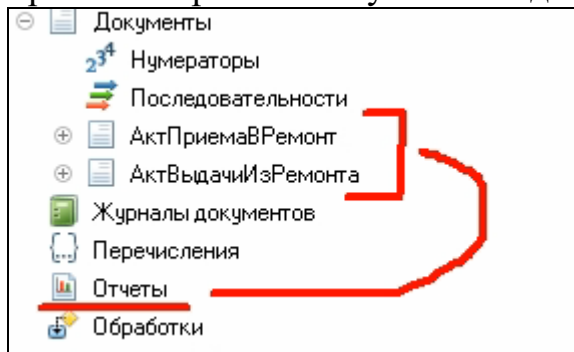
Реквизиты Клиент, Оборудование, Склад – остаются без изменений, реквизит Описание неисправностей – переименуем в **ВыполненныеРаботы**.

Проверим в работе созданные объекты, запустим приложение в режиме 1С: Предприятие. Выполним команду Сервис – 1С: Предприятие. Сохраним изменения. Запустится 1С: Предприятие.

IV. РЕГИСТРЫ И ОТЧЕТЫ

Т.о. можно создавать клиентов, записать насосы, принятые в ремонт, отремонтировать их и вернуть клиенту. Но через некоторое время руководителя понадобятся отчетные данные. Например, сколько насосов было отремонтировано в прошлом месяце, какое оборудование находится сейчас в ремонте и т.д. Для ответа на эти вопросы служат **Отчеты**.

Теоретически такие отчеты можно построить на основе тех документов, которые уже имеются: проанализировать от кого и сколько приняли в ремонт, затем проанализировать кому и что выдали из ремонта.



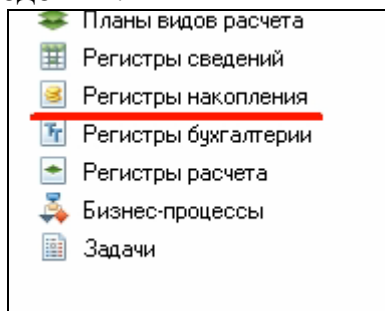
Это будет работать, пока пользователь не решит усовершенствовать существующую конфигурацию и вести более точный отчет.

Например, реализуем в отчете два фактора:

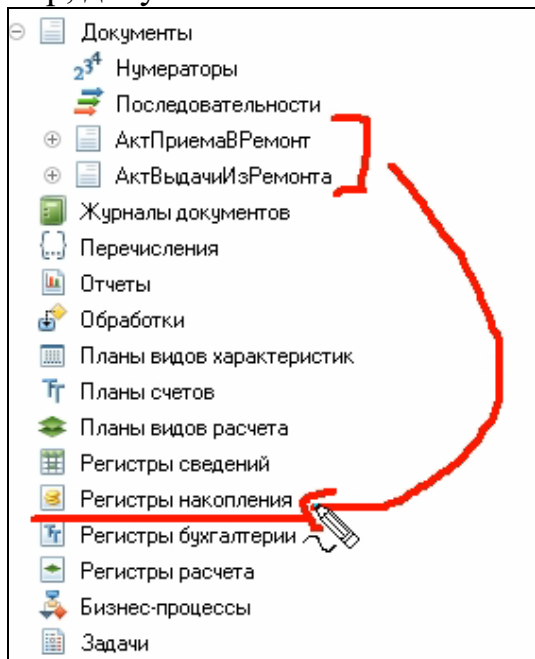
1. передачу принятых насосов в ремонтное подразделение (это может быть не в тот же день, когда было принято оборудование, а в другой день в зависимости от загрузки сотрудников),
2. завершение ремонта (клиент приезжает не в тот же день, когда было отремонтировано оборудование, а через несколько дней).

Для этого придется в конфигурацию добавить еще два документа. А это значит, что придется переделывать отчет, т.к. он учитывает существующие ранее два документа. Чтобы избежать такой ситуации и чтобы не собирать долго итоговые данные по многим документам, а получить их быстро, в 1С:

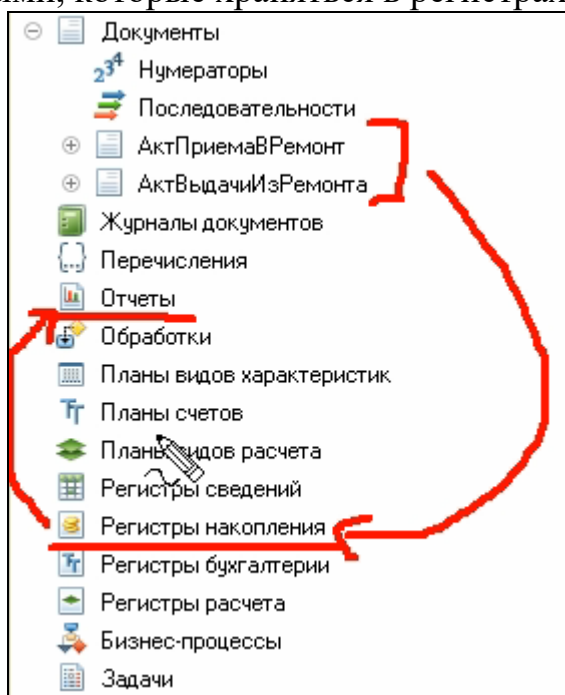
Предприятия существуют **Регистры**, а документы имеют возможность **Проведения**.



Регистры специальным образом хранят учетные данные, поставляемые, например, документами.



При проведении документ, как раз записывает свои данные в регистр. А отчеты оперируют не теми данными, которые хранятся в документах, а теми данными, которые хранятся в регистрах.



Таки образом дело отчета выбрать из регистра в нужном разрезе нужные данные, дело документа – поместить свои данные в регистрах в соответствии с прикладной логикой. Регистры специальным образом обрабатывают данные, которые в них помещают документы так, чтобы отчеты могли быстро получить им итоговые значения. Добавление в конфигурацию новых документов сводится к тому, чтобы правильно поместить данные документы в нужные регистры. При этом имеющиеся отчеты остаются без изменений.

Переходим назад в **Конфигуратор**.

1. Добавляем **Регистр накоплений – Учет оборудования**.

Правой кнопкой мыши щелкнем по Регистры накопления в конфигураторе и выберем **Добавить**.

В появившейся форме введем имя регистра **УчетОборудования** (рис. 20):

Рис. 20. Регистр накопления Учет оборудования

2. Далее опишем **структуру учета**, откроем закладку **Данные**. Регистр предполагает описание многомерной системы учета, имеющей один или несколько ресурсов в точке пересечения измерений.

Измерения регистра – это разрезы регистра, в которых учитываются хранимые данные.

Создадим три измерения:

- Имя – **Оборудование**, Тип данные – **Справочник.Ссылка.Оборудование**.
- Имя – **Клиент**, Тип данные – **Справочник.Ссылка.Клиенты**.
- Имя – **Склад**, Тип данные – **Справочник.Ссылка.Склады**.

Ресурсы регистра – это сами учитываемые показатели. В данном случае учитываемыми показателями является **количество**.

Так как каждый насос является уникальным – у каждого насоса есть серийный номер, то количество может быть либо 1, либо 0 (насос отремонтировали и отдали).

Создадим ресурс **Количество**, тип данных – **Число**.

3. Далее дважды щелкнем на регистр **Учет оборудования**, на закладке **Регистраторы** укажем, что данные в это регистр будут записывать два документа **Акт приема в ремонт** и **Акт выдачи из ремонта** (рис. 21).

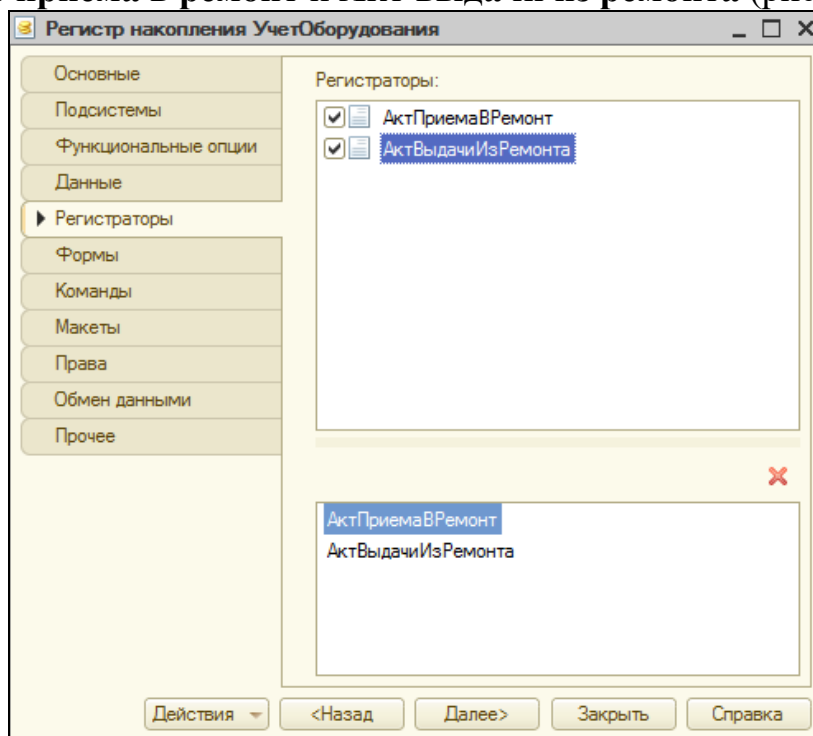


Рис. 21. Регистр накопления Учет оборудования закладка Регистры

4. Осталось для каждого из этих документов **описать алгоритмы**, в соответствии с которыми они будут помещать свои данные в этот регистр. Данные документ, помещенные в регистр, называются Движениями.

Дважды щелкнем на документ **Акт приема в ремонт**. Перейдем на закладку **Движения**. Воспользуемся **Конструктором движения** (рис. 22).

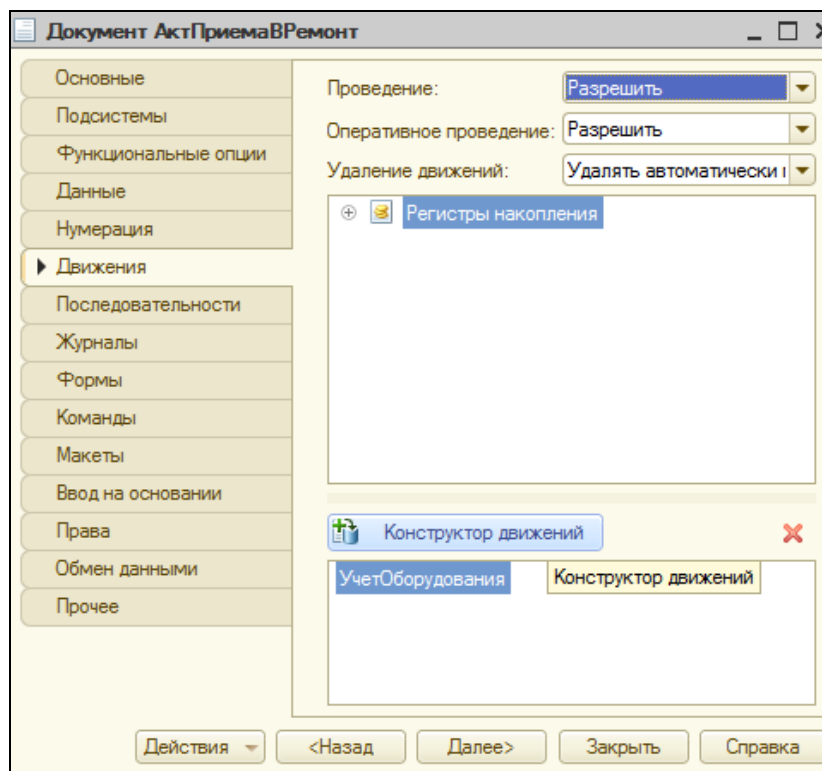


Рис. 21. Документ Акт приема в ремонт закладка Движения

Конструктор движения регистров (рис. 22) содержит:

- слева указано, что движение будем формировать для регистра Учет оборудования;
- ниже указано, что формировать будем движение Прихода, т.е. такие движение, которые увеличивают количество в регистре;
- справа перечислены доступные реквизиты документа, т.е. то, что может быть записаны в этот регистр;
- слева в низу перечислены реквизиты регистров, которые должны что-то записать.

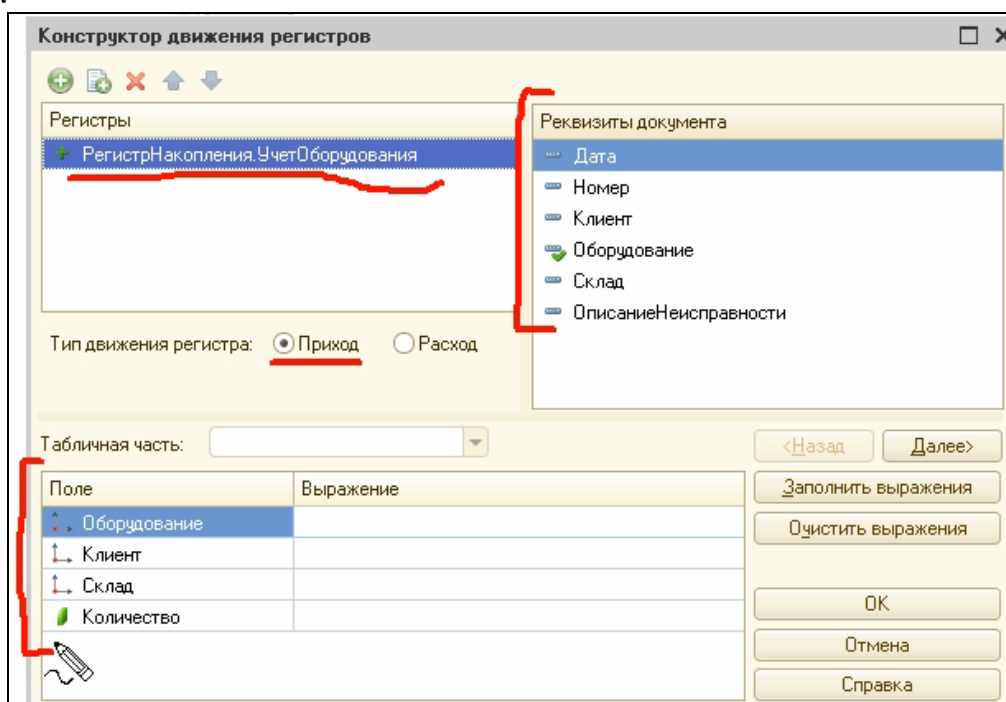


Рис. 21. Конструктор движения регистров

Для того, чтобы записать в эти регистры воспользуемся кнопкой **Заполнить выражение**. Конструктор предложил записать в измерения Оборудование, Клиент, Склад. А что записать в ресурс Количество, он не знает (рис. 22).

Поле	Выражение
Оборудование	Оборудование
Клиент	Клиент
Склад	Склад
Количество	

Рис. 22. Конструктор движения регистров (заполненный)

В ресурс количество запишем – 1, т.е. по акту проведения работ – это означает, что в ремонте появилось новое оборудование. Нажмем **ОК**.

Откроется модуль документа (рис. 23), в котором конструктор сформировал процедуру проведения этого документа.

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изм
Движения.УчетОборудования.Записывать = Истина;
// регистр УчетОборудования Приход
Движение = Движения.УчетОборудования.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Оборудование = Оборудование;
Движение.Клиент = Клиент;
Движение.Склад = Склад;
Движение.Количество = 1;

//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```


Поясним текст процедуры обработчика.

Конструктор создал обработчик события **ОбработкаПроведения** объекта конфигурации **Документ Акт приема в ремонт**, поместил его в модуль объекта и открыл текст модуля.

Событие **ОбработкаПроведения** является одним из важнейших событий, связанных с документами. Это событие возникает при проведении документа. Основное назначение обработчика – генерация движений по документу. Выполнение различных операций с данными в процедуре обработчика влияет на состояние учета. Т.о., именно в эту процедуру разработчик должен поместить собственные алгоритмы преобразования данных, выполняемые в момент проведения документа.

Объект встроенного языка **ДокументОбъект** имеет свойство **Движения**. Оно возвращает объект **КоллекцияДвижений**, содержащий коллекцию наборов записей регистров, по которым этот документ может формировать движение.

К конкретному набору записей этой коллекции можно обратиться, указав через точку имя регистра, которому принадлежит этот набор записей (**ДвижениеУчетОборудования**).

Затем через точку можно использовать различные методы набора записей регистра (**Движения.УчетОборудования.Добавить()**). Метод **Добавить()** добавляет новую запись в набор записей.

В первой строке процедуры устанавливаем свойство **Записывать** набора записей регистра в значение **Истина**. То есть в явном виде указываем, что после завершения обработки проведения платформа должна будет записать этот набор записей в базу данных.

Далее, используя метод **Добавить()**, идет добавление к набору записей, который создает документ в регистре, новую запись. Тем самым создаем объект **РегистрНакопленияЗапись** и сохраняем его в переменной **Движение**.

Используя этот объект, можно обратиться к полям этой записи, указав имя поля через точку от этой переменной (например, **Движение.Период**).

Причем **Движение.Оборудование**, **Движение.Клиент**, **Движение.Склад** – это измерения регистра, **Движение.Количество** – это ресурс регистра, а **Движение.ВидДвижения** и **Движение.Период** – стандартные реквизиты регистра, которые создаются автоматически.

Чтобы присвоить полям новой записи регистра соответствующие значения полей документа, обращаемся к полям табличной части, указав имя поля через точку от переменной **ТекСтрокаМатериалы** (например, **ТекСтрокаМатериалы.Материал**) (это свойство будем использовать в обратном регистре).

ВидДвиженияНакопления.Приход – это значение системного перечисления, которое определяет вид движения регистра накопления как **Приход**.

Т. о., присваиваем нужные значения всем полям новой записи. После перебора всех строк документа (после завершения цикла) в этом наборе записей

(Движения.УчетОборудования) будет содержаться столько записей, сколько строк в табличной части проводимого документа.

5. Аналогичным образом поступим со вторым документом. Разница будет в том, что **Акт выдачи из ремонта** должен уменьшать количество насосов в регистре, т.к. происходит выдача насоса клиенту.

Выполним: Документ **Акт выдачи из ремонта** вкладка **Движение**, далее Конструктор движения. Указываем Тип движения – **Расход**. Нажимаем – **Заполнить выражение**. Количество – **1**. Нажимаем ОК.

В результате откроется модуль документа (рис. 24), в котором конструктор сформировал процедуру проведения этого документа.

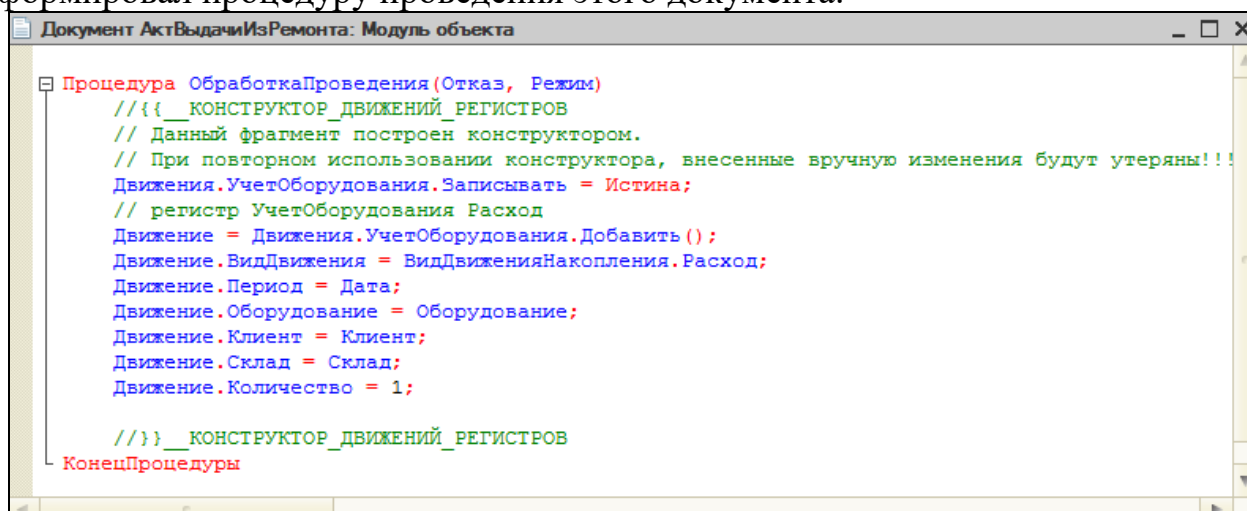


Рис. 24. Модуль документа Акт выдачи из ремонта

В результате в регистре должно стать на один насос меньше.

6. Далее добавим **отчет**, который покажет «что обстоит с ремонтом насосов на предприятии в общем».

Для того, чтобы добавить отчет, в Дереве конфигурации выберите Отчет **Добавить** (рис. 25).

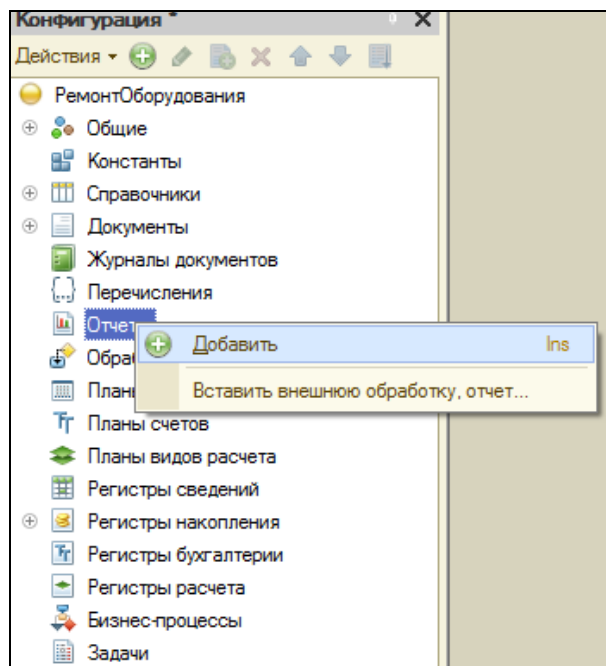


Рис. 25. Дерево конфигурации Добавить отчет

Назовем отчет **Ремонт оборудования** (рис. 26). Основу любого отчета составляет **Схема компоновки данных**.

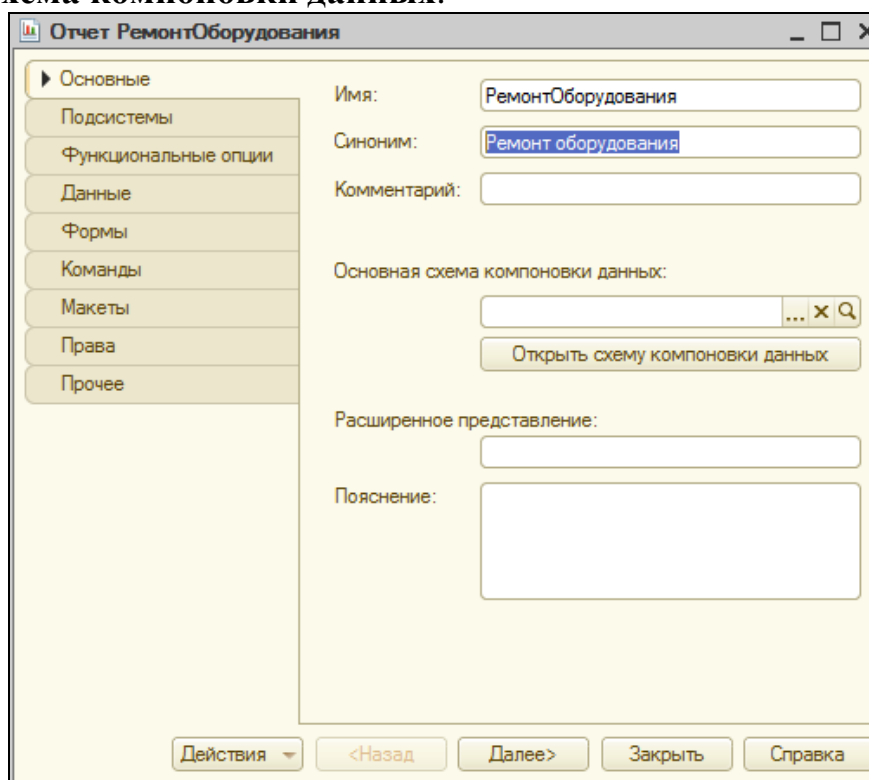
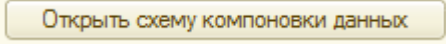


Рис. 26. Форма создания отчета

7. Открываем **Схему компоновки данных** . В **Конструкторе макета** нажмите на кнопку **Готово**.

Любая схема компоновки данных должна знать, откуда необходимо брать данные для отчета. Поэтому добавим **Набор данных – запрос** (рис. 27).

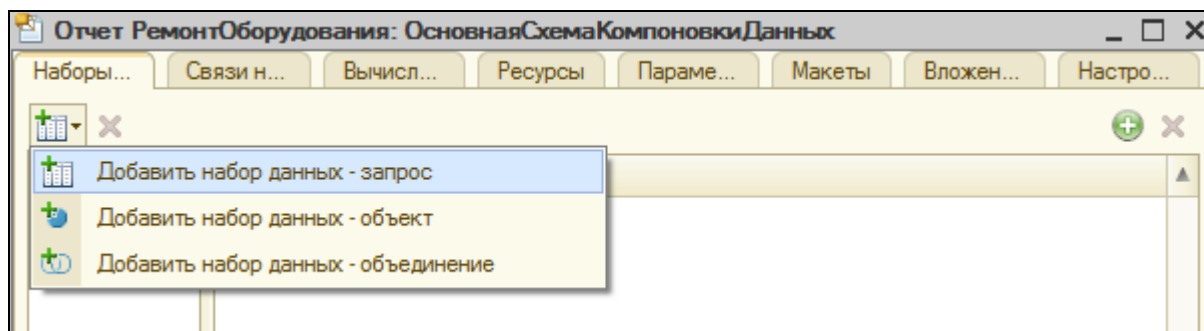


Рис. 27. Основная схема компоновки данных

8. Такой набор данных получает данные из базы данных 1С: Предприятия с помощью запроса. Какие именно данные получать, описывается на языке запросов. Нажмем кнопку **Конструктор запроса** (рис. 28). Конструктор позволяет не писать запрос вручную, а сконструировать его с помощью мыши.

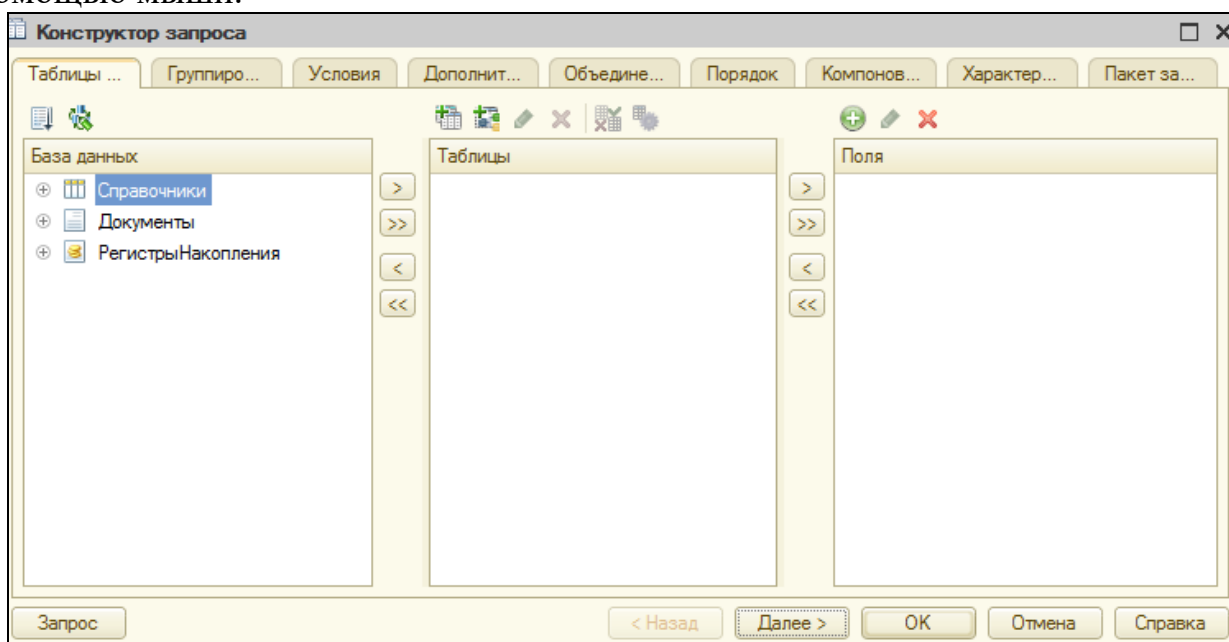


Рис. 28. Конструктор запроса

Раскроем Регистры накопления. Перенесем ОстаткиИОборы в Таблицы. Из этой Таблице возьмем все Поля, кроме поля Оборот (рис. 28а).

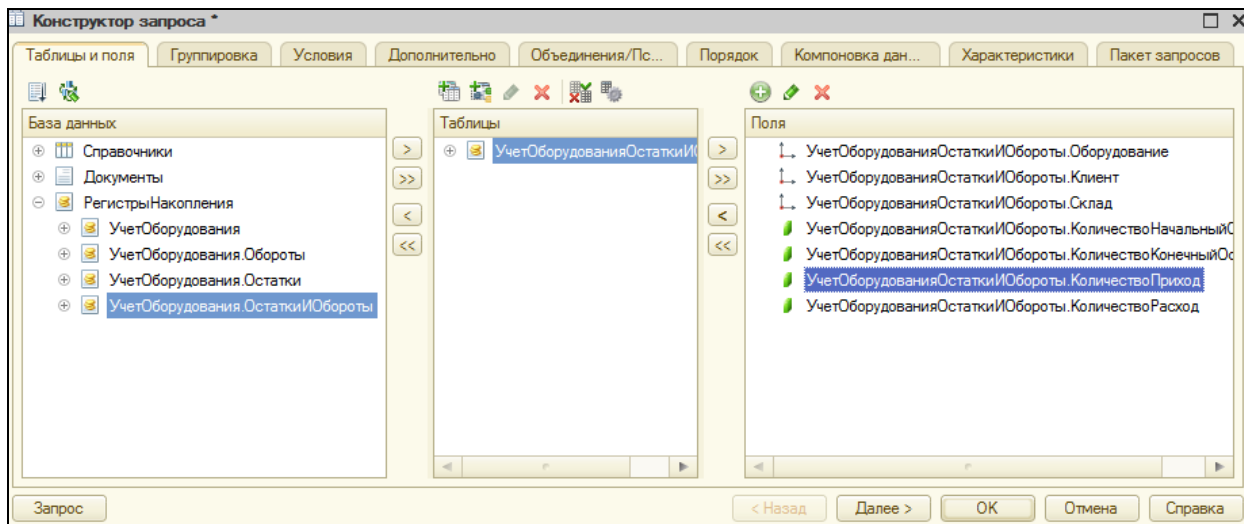


Рис. 28а. Конструктор запроса

На этом формирование данного запроса закончено. Нажмем ОК и закроем Конструктор.

9. Сформировался текст запроса и заполнились поля Схемы компоновки данных (рис. 29).

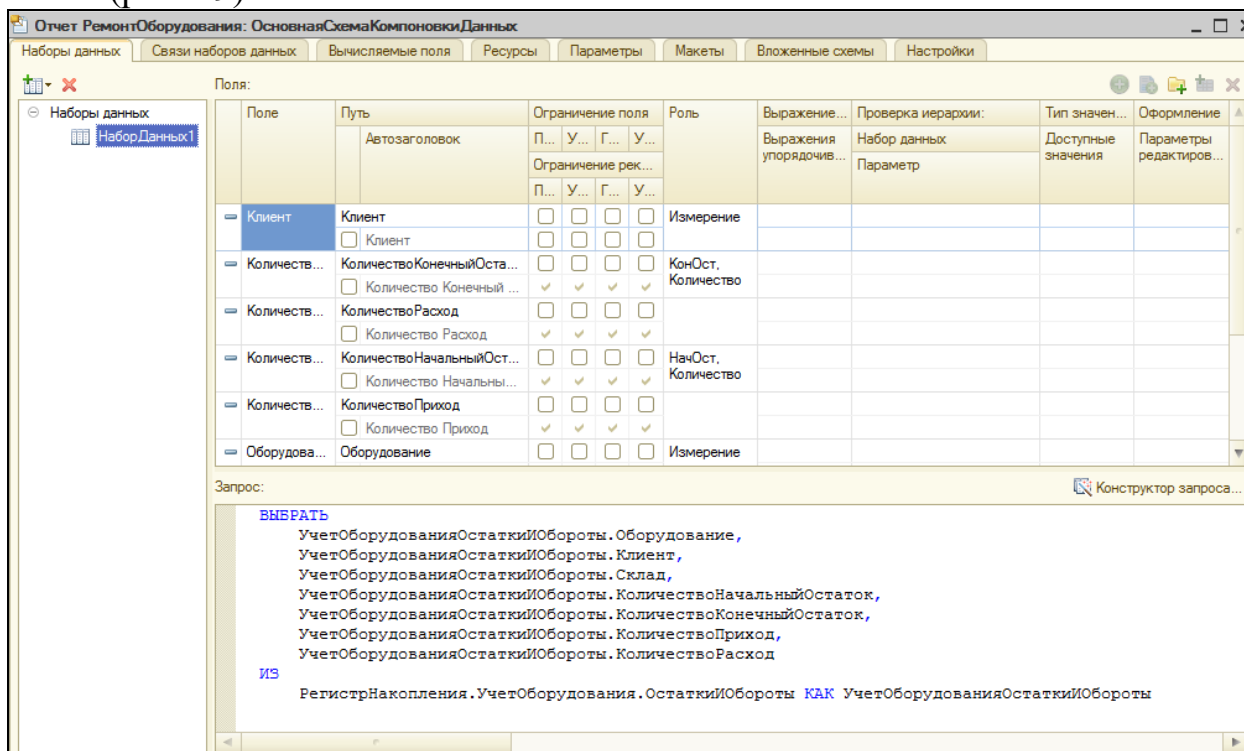


Рис. 29. Схема компоновки данных Отчета

Изменим заголовки нескольких полей, чтобы отчет был читабельным.

Поле КоличествоКонечныйОстаток — заменим на Остаток на конец периода (рис. 30, 30а).

Поля:						
Поле	Путь	Ограничение поля				Роль
		Автозаголовок				
		П...	У...	Гр...	У...	
		Ограничение рекв...				
		П...	У...	Гр...	У...	
Клиент	Клиент	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Измерение
	<input type="checkbox"/> Клиент	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Количество...	КоличествоКонечныйОстаток	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	КонОст, Количество
	<input type="checkbox"/> Количество Конеч...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Количество...	КоличествоРасход	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/> Количество Расход	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Количество...	КоличествоНачальн...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	НачОст, Количество
	<input type="checkbox"/> Количество Начал...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Количество...	КоличествоПриход	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/> Количество Приход	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Оборудова...	Оборудование	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Измерение

Рис. 30. Фрагмент схемы компоновки данных Отчета

Поля:							
Поле	Путь	Ограничение поля				Роль	
		Автозаголовок	П...	У...	Гр...		У...
			Ограничение рекв...				
			П...	У...	Гр...		У...
Клиент	Клиент	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Измерение	
	<input type="checkbox"/> Клиент	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Количество...	[Остаток на конец периода]	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	КонОст, Количество	
	<input type="checkbox"/> Количество Конеч...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Количество...	КоличествоРасход	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	<input type="checkbox"/> Количество Расход	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Количество...	КоличествоНачальн...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	НачОст, Количество	
	<input type="checkbox"/> Количество Начал...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Количество...	КоличествоПриход	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	<input type="checkbox"/> Количество Приход	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Оборудова...	Оборудование	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Измерение	

Рис. 30а. Фрагмент схемы компоновки данных Отчета

Изменим также заголовки еще нескольких полей:

- Поле КоличествоРасход – заменим на Выдано клиенту.
- Поле КоличествоНачальныйОстаток – заменим на Остаток на начало периода.
- Поле КоличествоПриход – заменим на Принято в ремонт.

10. Далее на Схеме компоновки данных Отчета переходим на закладку **Ресурсы**. Все поля, итоговые значения которых можно вычислить в результате выполнения отчета, добавим в ресурсы Схемы компоновки данных (Рис. 31).

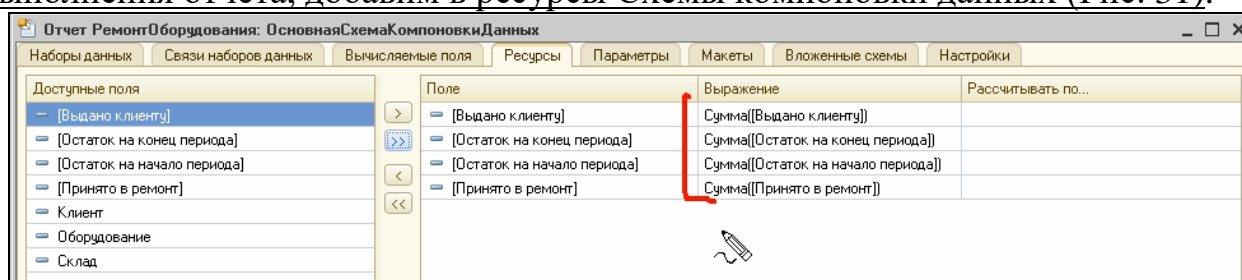


Рис. 31. Фрагмент схемы компоновки данных Отчета. Вкладка Ресурсы

Платформа для вычисления этих полей автоматически использует функцию **суммирования**, что необходимо для отчета (рис. 31).

11. Переходим на закладку **Настройки** и создадим структуру отчета (т.е. его внешний вид).

Необходимо, чтобы отчет показывал суммарные данные по отдельным клиентам, которые можно развернуть и посмотреть более подробные сведения: какие именно насосы ремонтировались для этого клиента.

Поэтому сначала добавим группировку Клиент (рис. 32а, б, в).

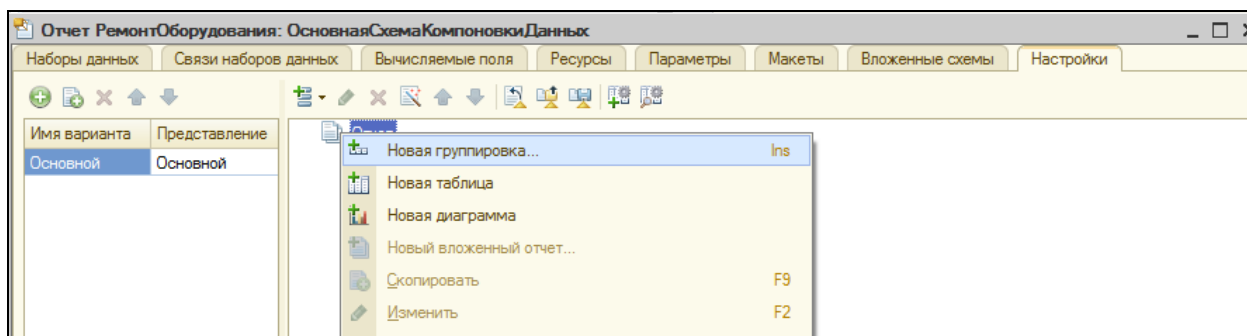


Рис. 32а. Фрагмент схемы компоновки данных Отчета. Вкладка Настройки.
Создание новой группировки

Новая группировка – поле Клиент – ОК (рис. 32б).

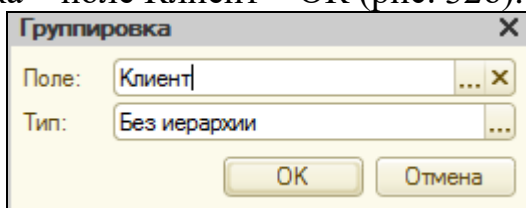


Рис. 32б. Создание новой группировки Клиент

Далее в группировку **Клиент** добавим группировку **Детальная запись**. Для этого нет необходимости указывать группировочное поле (название оставляем пустым) (рис. 32в).

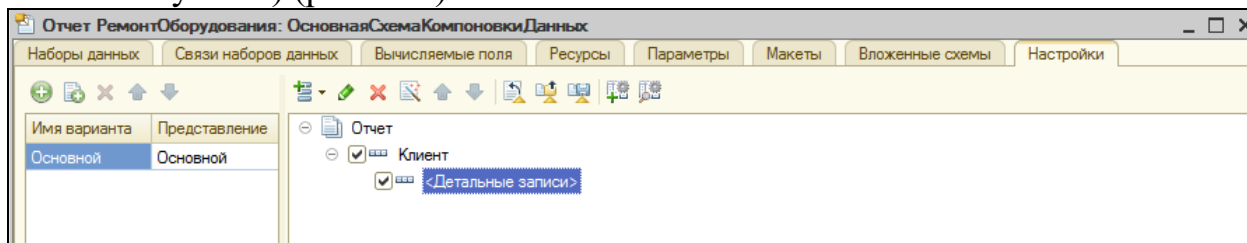

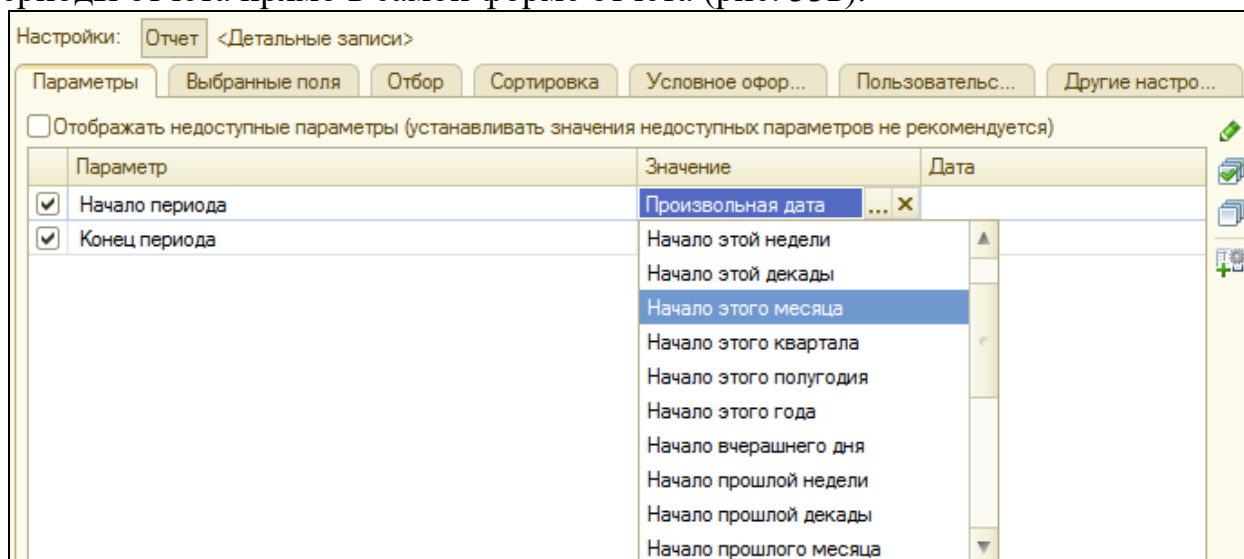


Рис. 32в. Фрагмент схемы компоновки данных Отчета. Вкладка Настройки.

Укажем, что в качестве

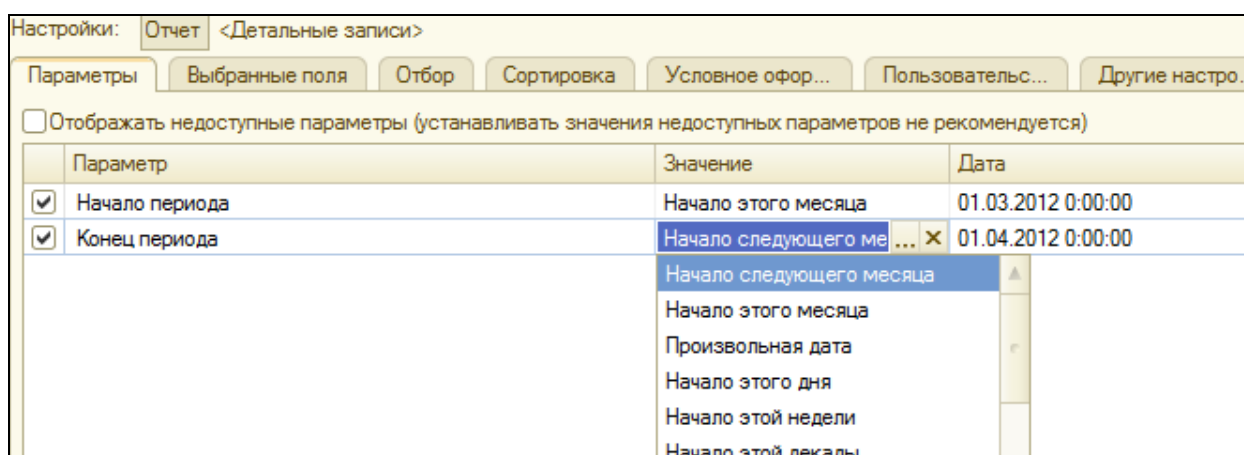
- начала периода необходимо использовать дату начала этого месяца (рис. 33а),
- конца периода – дату начала следующего месяца.

Оба параметра включим в быстрые пользовательские настройки (кнопка  справа на форме отчета), чтобы пользователь при желании мог поменять периоды отчета прямо в самой форме отчета (рис. 33в).



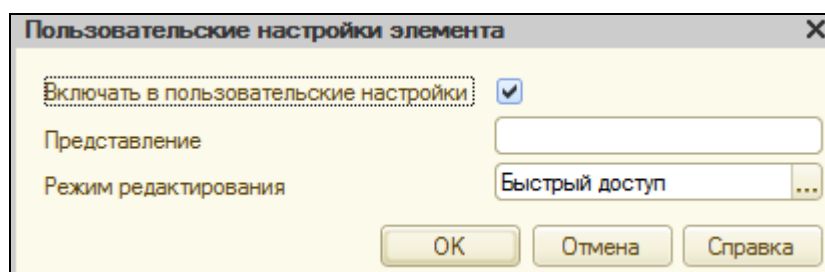
Параметр	Значение	Дата
<input checked="" type="checkbox"/> Начало периода	Произвольная дата	
<input checked="" type="checkbox"/> Конец периода	Начало этой недели	
	Начало этой декады	
	Начало этого месяца	
	Начало этого квартала	
	Начало этого полугодия	
	Начало этого года	
	Начало вчерашнего дня	
	Начало прошлой недели	
	Начало прошлой декады	
	Начало прошлого месяца	

Рис. 33а. Фрагмент схемы компоновки данных Отчета. Вкладка Параметры. Начало периода



Параметр	Значение	Дата
<input checked="" type="checkbox"/> Начало периода	Начало этого месяца	01.03.2012 0:00:00
<input checked="" type="checkbox"/> Конец периода	Начало следующего ме...	01.04.2012 0:00:00
	Начало следующего месяца	
	Начало этого месяца	
	Произвольная дата	
	Начало этого дня	
	Начало этой недели	
	Начало этой декады	

Рис. 33б. Фрагмент схемы компоновки данных Отчета. Вкладка Параметры. Конец периода



Пользовательские настройки элемента

☒ Включать в пользовательские настройки

Представление:

Режим редактирования: Быстрый доступ

OK Отмена Справка

Рис. 33в. Пользовательские настройки элемента отчета

12. Перейдем на вкладку **Выбранные поля** и выберем поля, которые будут отражаться в отчете в следующей последовательности:

- Клиент,
- Оборудование,
- Склад,

- Остаток на начало периода,
- Принято в ремонт,
- Выдано клиенту,
- Остаток на конец периода.

Для этого необходимо два раза щелкнуть мышью на необходимое поле в разделе доступные поля. В результате схема компоновки данных Отчета. Вкладка Выбранные поля должна выглядеть следующим образом (рис 34):

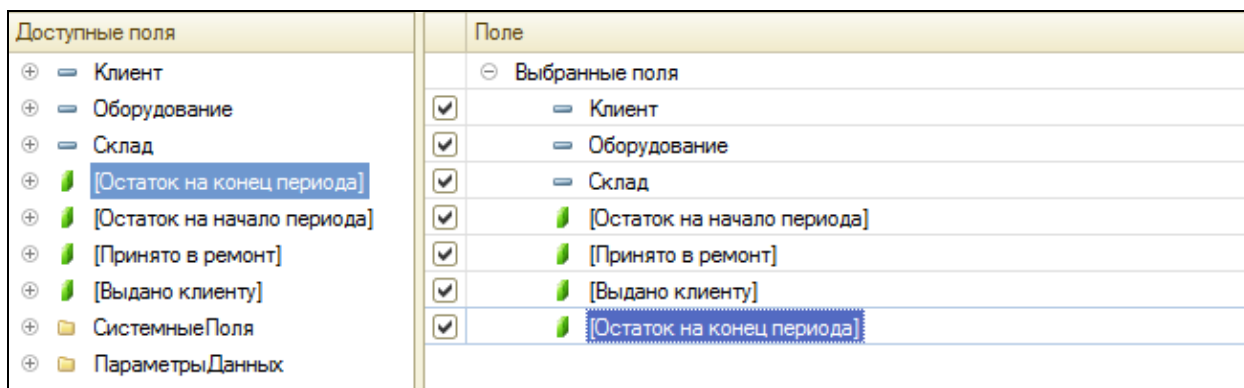


Рис. 34. Фрагмент схемы компоновки данных Отчета. Вкладка Выбранные поля

Проверим в работе выполненные выше действия, запустим приложение в режиме 1С: Предприятие. Поработаем с разработанной конфигурацией.

ВВОД ДАННЫХ В БД

1. Клиент принес в ремонт насос на основной склад.

Выполним:

1. В список Клиентов добавим нового клиента – **Насосный завод** (рис. 35а).

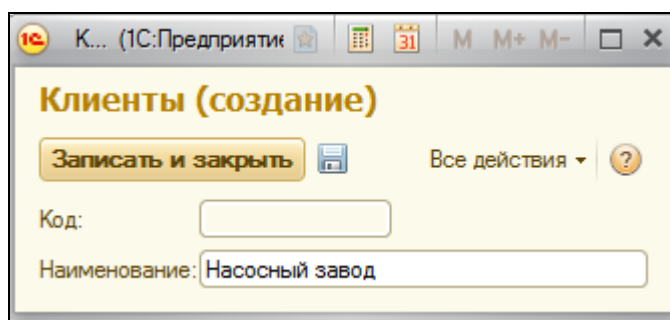


Рис. 35а. Форма создание нового клиента

2. В список оборудования добавим новый насос – **Насос ЦМНШ-80.5, с/н 987254** (рис. 35б).

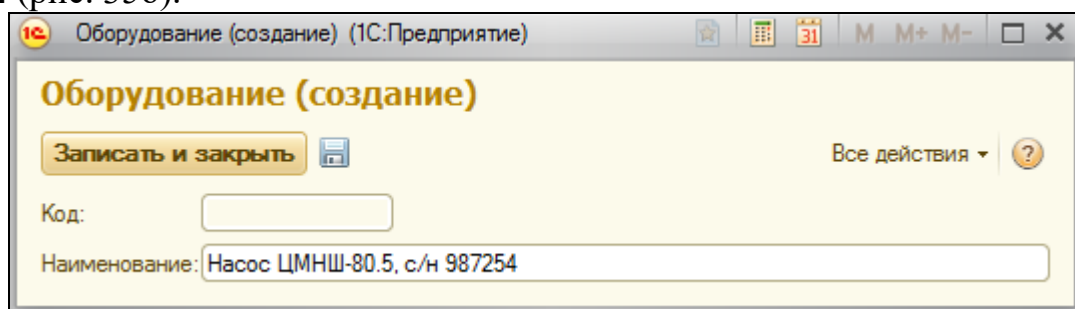


Рис. 35б. Форма создание нового оборудования

3. В список **Актов приема в ремонт** добавим новый документ: Клиент – Насосный завод. Оборудование – Насос ЦМНШ-80.5, с/н 987254. Склад – Основной склад (он уже есть в конфигурации, потому что при разработке был обозначен как предопределенный). Неисправность – Не работает совсем. Дата – 5.11.2011 г. Провести. Закреть (рис. 35в).

Акт приема в ремонт (создание) *

Провести и закрыть Провести Все действия ?

Номер:

Дата: 05.11.2011 0:00:00

Клиент: Насосный завод

Оборудование: Насос ЦМНШ-80.5, с/н 987254

Склад: Основной склад

Описание неисправности:

Не работает совсем

Рис. 35в. Форма создание нового документа Акт приема в ремонт

4. Далее откроем список **Актов выдачи из ремонта**. Отметим, что 11.11.2011 г. насос был отремонтирован и отдан клиенту. Выполненные работы – Замена упругой втулки муфты. Провести. Закреть (рис. 36).

Акт выдачи из ремонта 000000001 от 11.11.2011 ...

Провести и закрыть Провести Все действия ?

Номер: 000000001

Дата: 11.11.2011 12:00:00

Клиент: Насосный завод

Оборудование: Насос ЦМНШ-80.5, с/н 987254

Склад: Основной склад

Выполненные работы:

Замена упругой втулки муфты

Рис. 36. Форма создание нового документа Акт выдачи из ремонта

Отметим недостатки в разработанной конфигурации:

– Когда создается Акт выдачи из ремонта, то снова придется выбирать Клиента, Оборудование. Эти реквизиты такие же, как в Акте приема в ремонт.

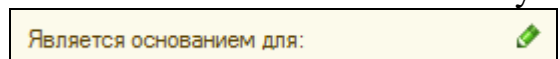
– Когда выбираем склад, то он выбирается в отдельном окне. Все это не совсем удобно, т.к. большого количества складов наверняка не будут.

Вернемся в конфигуратор и сделаем несколько доработок.

V. ВВОД НА ОСНОВАНИИ

Укажем, что на основании насоса будет выводиться Акт приема в ремонт.

1. В справочнике Оборудование откроем вкладку Ввод на основании. Нажмем на кнопку **Является основанием для**



Выбираем – Документ **АктПриемаВРемонт** (рис. 37а).

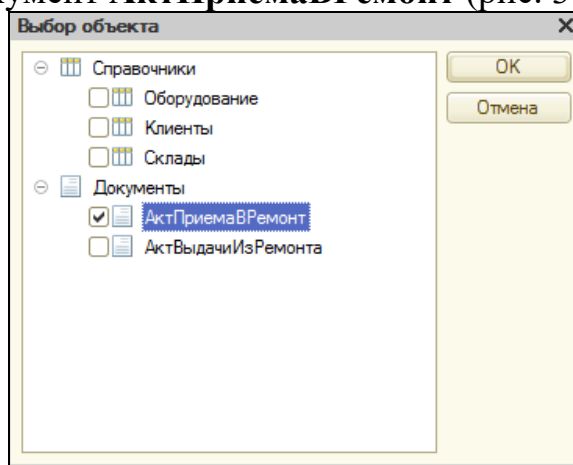


Рис. 37а. Ввод на основании. Выбор объекта

2. На основании **АктПриемаВРемонт** будет вводится **АктВыдачиИзРемонта**.

Выполните данные действия для документа (открыть документ **АктПриемаВРемонт** – Вкладка Ввод на основании – Является основанием для **АктВыдачиИзРемонта**) (рис. 37б).

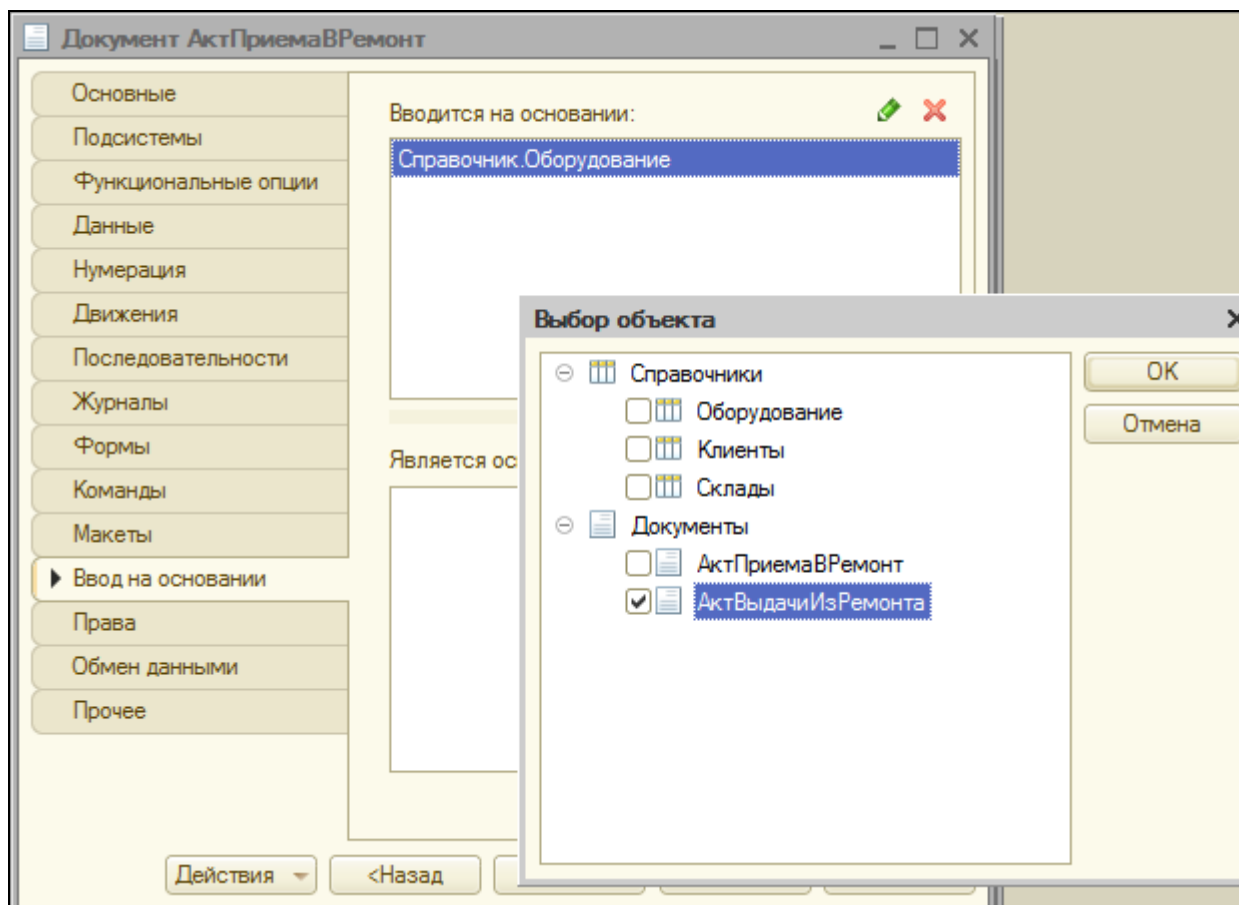


Рис. 37б. Ввод на основании. Выбор объекта для Акта приема в ремонт

3. Далее **создадим процедуры**, описывающие какие именно реквизиты должны заполняться при **Вводе на основании**.

Для документа **АктПриемаВРемонт** откроем его **Модуль** (нажатием правой кнопкой мыши открываем контекстное меню документа **АктПриемаВРемонт** – выбираем **Открыть модуль объекта**) (рис. 38).

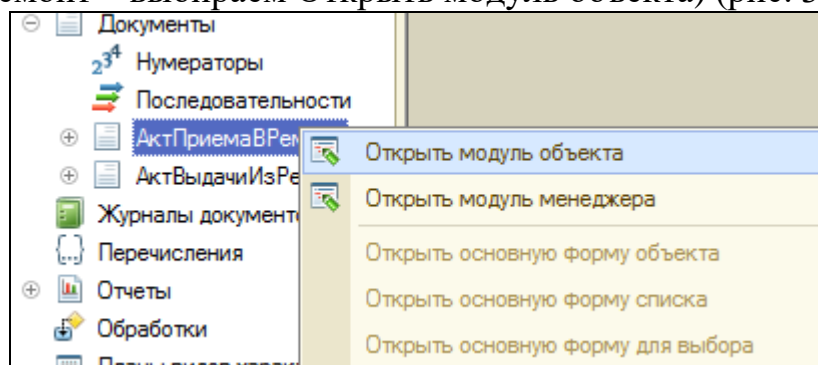


Рис. 38. Контекстное меню Документа

Создадим обработчик событий Обработка заполнения (рис. 39). В модуле появилась новая запись (рис. 40а).

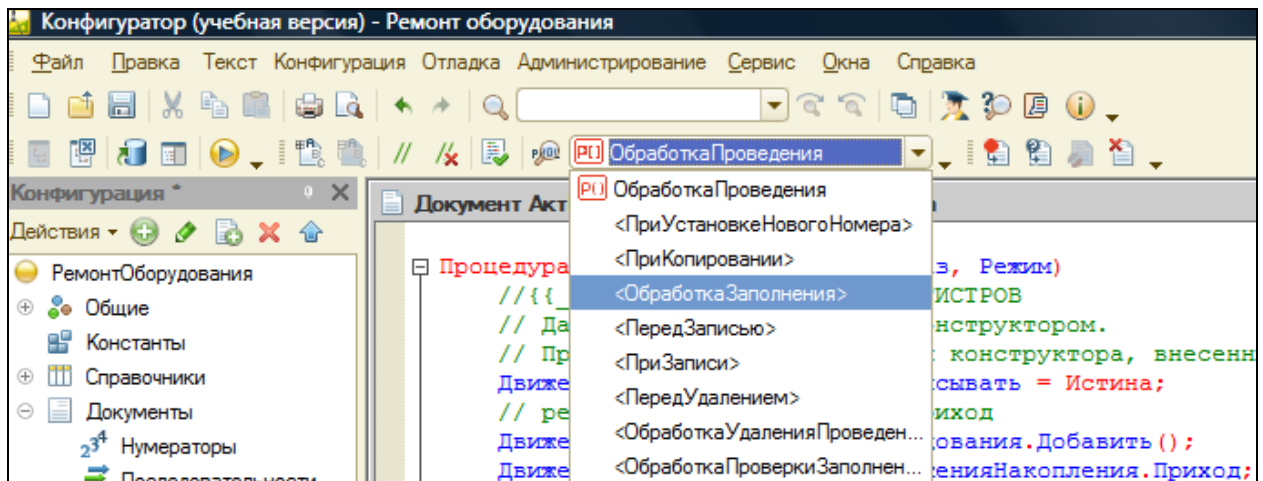


Рис. 39. Обработчик событий Обработка заполнения

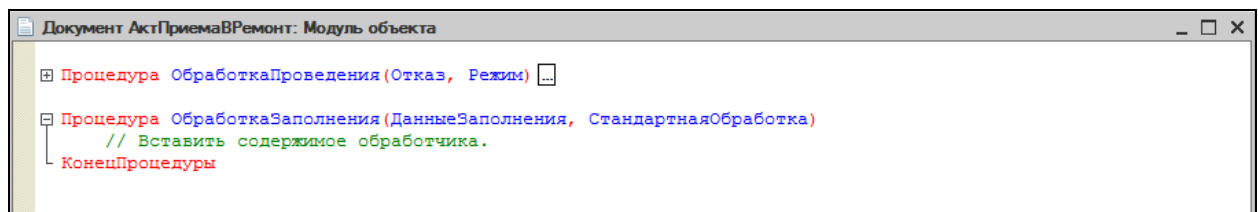


Рис. 40а. Модуль документа Акт приема в ремонт

По условию задачи Акт приема в ремонт вводится на основании насоса, поэтому в параметры ДанныеЗаполнения будет передана ссылка на этот самый насос.

Реквизиту документа АктПриемаВРемонт Оборудование присвоим значение ДанныеЗаполнения. Для это вводим в процедуру

Оборудование = ДанныеЗаполнения;

(рис. 40б).

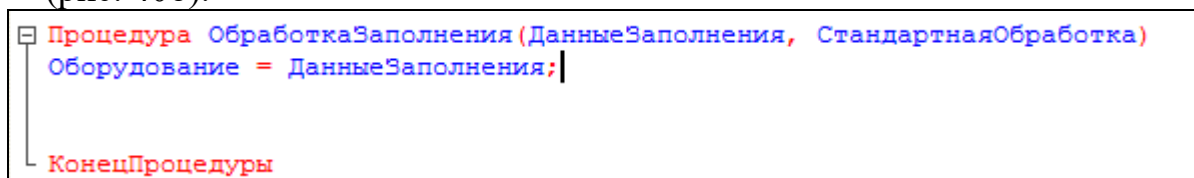


Рис. 40б. Модуль документа Акт приема в ремонт

Примечание. Также нужный реквизит можно просто перетащить мышью из дерева конфигурации, чтобы не писать реквизиты вручную. А при вводе имен переменных и операторов встроенного языка можно воспользоваться контекстной подсказкой, т.е. ввести начало имени и нажать ctrl+пробел. Платформа сама предскажет или автоматически дополнит напечатанное имя.

4. Далее для **Акта выдачи из ремонта** также опишем алгоритм заполнения при вводе на основании.

Для этого аналогично откроем **Модуль объекта документа**. Создадим **обработчик событий Обработка заполнения**.

Данный документ вводится на основании Акта приема в ремонт. Потому в данных заполнения будет ссылка на АктПриемаВРемонт. Значит

– в реквизит Клиент необходимо подставить ДанныеЗаполнения.Клиент,

- в реквизит Оборудование – ДанныеЗаполнения.Оборудования
- в реквизит Склад – ДанныеЗаполнения.Склад (рис. 40в).

```

Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)
    Клиент = ДанныеЗаполнения.Клиент;
    Оборудование = ДанныеЗаполнения.Оборудование;
    Склад = ДанныеЗаполнения.Склад;
КонецПроцедуры

```

Рис. 40в. Модуль документа Акт выдачи из ремонта

5. Также для справочника Склады укажем, что справочник будет использовать свойство **Быстрый выбор**. Это значит, что теперь Склад будет выбираться не из отдельной формы, открывающейся в отдельном окне, а из маленького списка, который выпадает прямо из поля ввода (открыть Свойства справочника Склады – отметить свойство Быстрый выбор) (рис. 41).

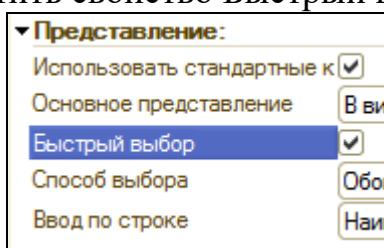


Рис. 41. Фрагмент Свойства справочника. Быстрый выбор

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие.

ВВОД ДАННЫХ В БД

1. Насосный завод принес еще один насос в ремонт. Запишем его в справочник – КМ 0-32-180/2-5, с/н7744.

Обратите внимание, что в командной панели появилась кнопка **Создать на основании** Создать на основании ▾.

Создадим на основании Акт приема в ремонт.

Оборудование, необходимое, уже подставилось (рис. 42). Клиент – Насосный завод. Склад – Основной склад (появился маленький список). Дата- 12.11.2011 г. Провести. Закрыть.

Рис. 42. Ввод Акта приема в ремонт

Отметим **недостатки в разработанной конфигурации**:

- Хорошо было бы, когда приносят насос в ремонт, то хорошо бы иметь список насосов, чтобы сразу проверить не ремонтировался ли у уже этот насос раньше, чтобы не добавлять его много раз.
- Список ссылок, представленный слева, выглядит не очень красиво (рис. 43).
- В подавляющем большинстве случаев насосы будут приниматься именно на основном складе. Поэтому хотелось бы, чтобы в новом документе поле склад сразу же было заполнено значением Основной склад.

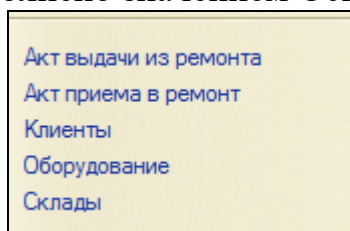


Рис. 43. Список ссылок меню системы

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие.

VI. ПРЕДОПРЕДЕЛЕННЫЕ ЭЛЕМЕНТЫ

Выполним, чтобы значение поля Склад в документах заполнялось автоматически.

1. В свойствах документа **АктПриемаВРемонт**, укажем реквизит Значение заполнения Предопределенный элемент справочника Склады, который был создан в конфигурации. В свойствах Склад выбираем Значение заполнения, в открывшемся списке выбираем Основной склад (рис. 44).

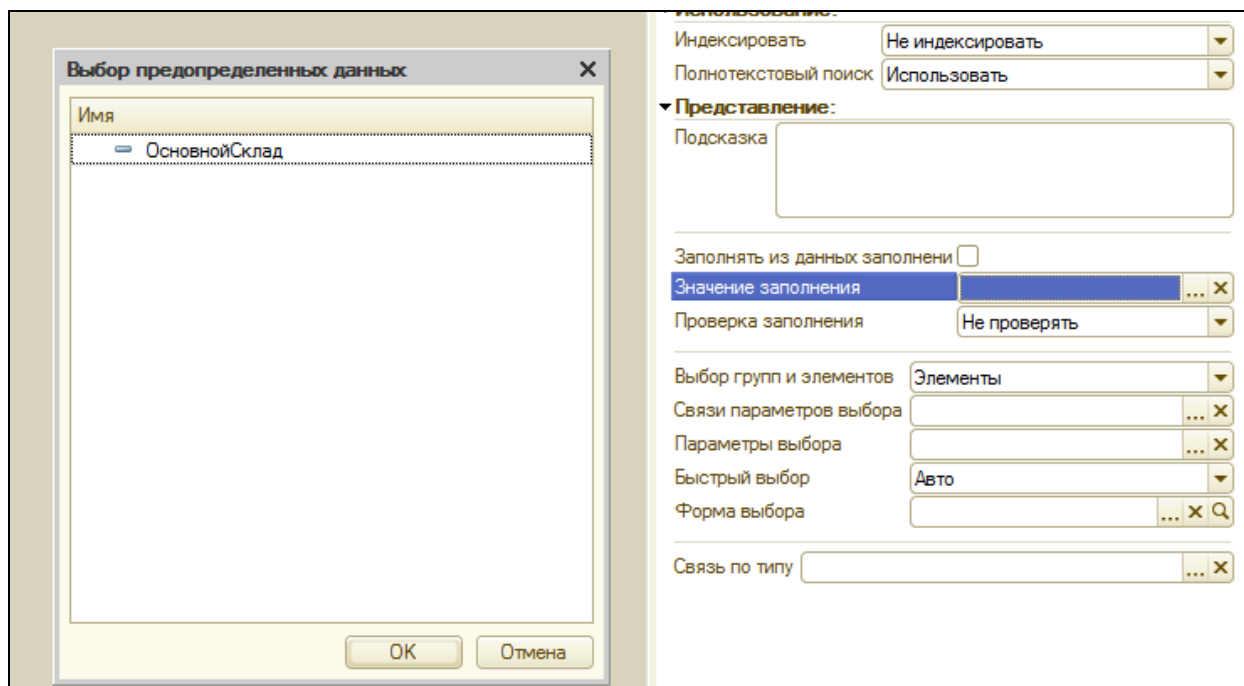


Рис. 44. Свойства документа *Значение заполнения*. Выбор predeterminedенных данных

Конфигурация\АктПриемаВРемонт\Данные\Склад\Свойства\Значение заполнения\ОсновнойСклад.

2. То же самое выполним в другом документе **АктВыдачиИзРемонта**.

Таким образом, в новых документах сразу будет заполняться **Основной склад**.

VII. РАБОЧИЙ СТОЛ И НАСТРОЙКА КОМАНДНОГО ИНТЕРФЕЙСА

Для решения остальных задач создадим **Рабочий стол** и определим его **Командный интерфейс**.

На рабочем столе пользователю необходимо видеть Список насосов. Для этого создадим **Форму списка справочника Оборудования**.

1. В справочнике **Оборудование** добавляем новую форму – **Форма списка справочника** (рис. 45а, 45б, 45в). Ничего в форме не меняем.

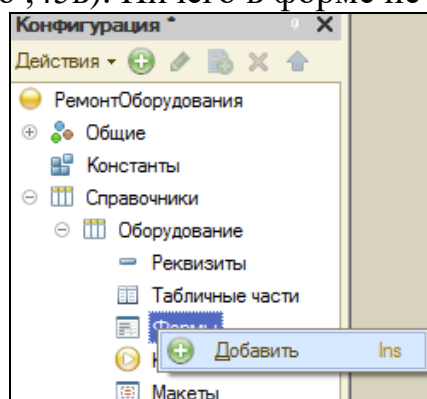


Рис. 45а. Выбор в дереве конфигурации формы справочника

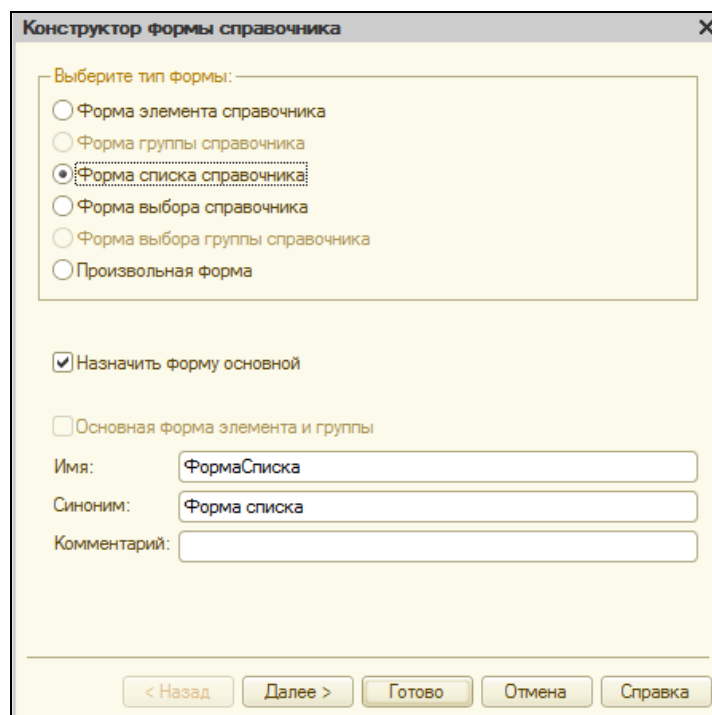


Рис. 45б. Конструктор формы справочник

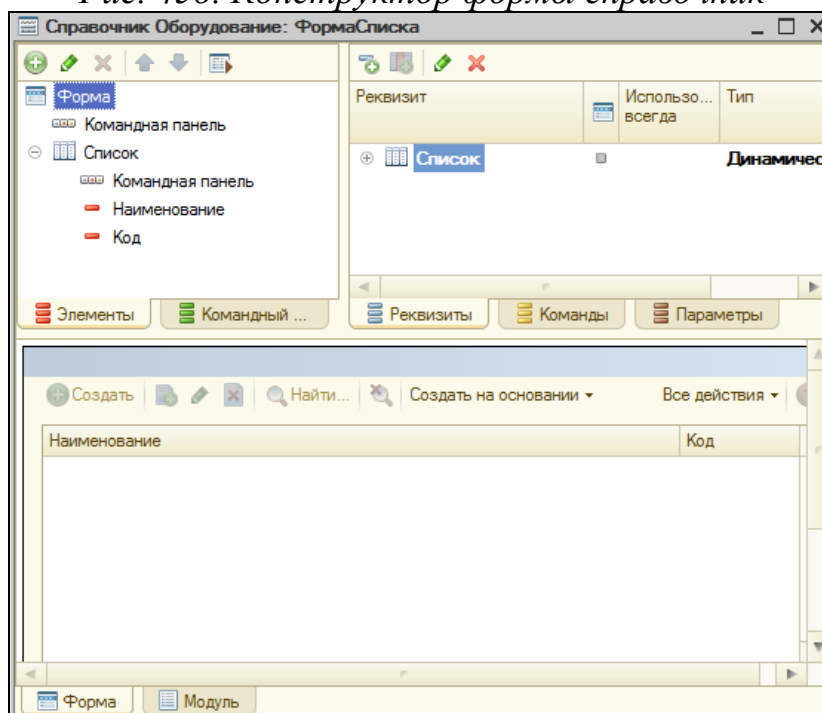


Рис. 45в. ФормаСписка справочника

2. Далее поместим эту форму на Рабочий стол. Откроем **Редактор рабочей области начальной страницы**.

Выполним: Контекстное меню конфигурации Контекстное меню конфигурации – Открыть рабочую область рабочего стола – Рабочая область начальной страницы – Шаблон рабочего стола – Одна колонка – Новая форма – Справочники – Оборудование – Форма списка (рис. 46а, 46б, 46в, 46г).

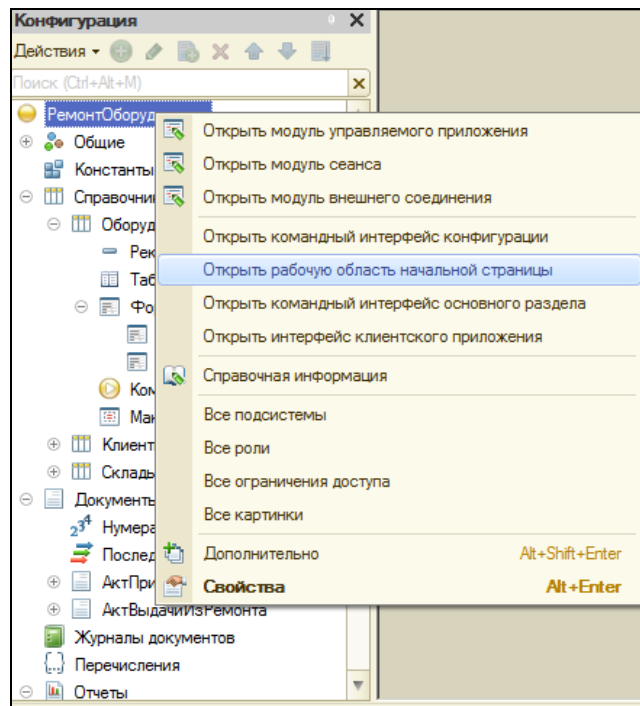


Рис. 46а. Контекстное меню конфигурации – Открыть рабочую область начальной страницы

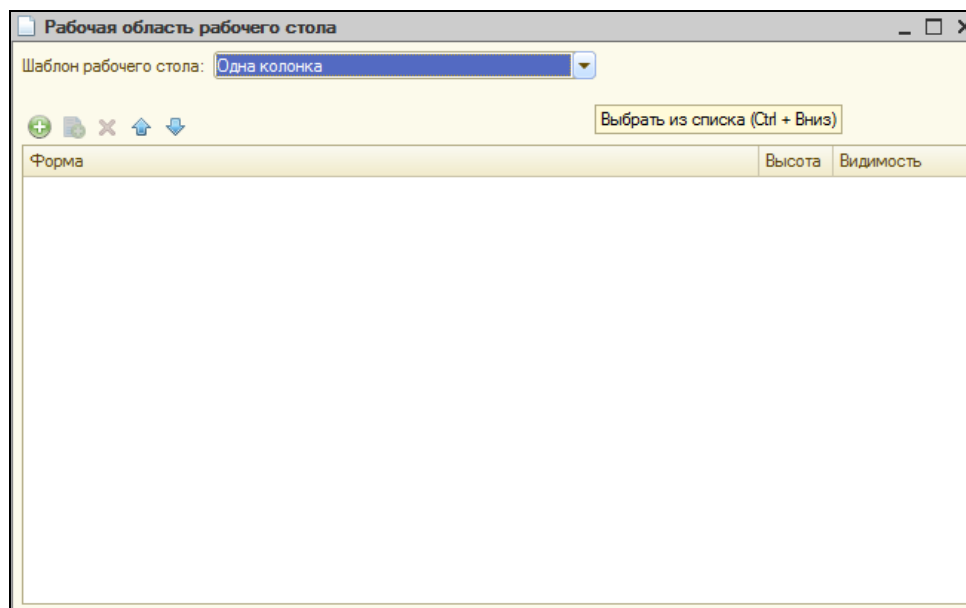


Рис. 46б. Рабочая область рабочего стола. Выбор шаблона рабочего стола

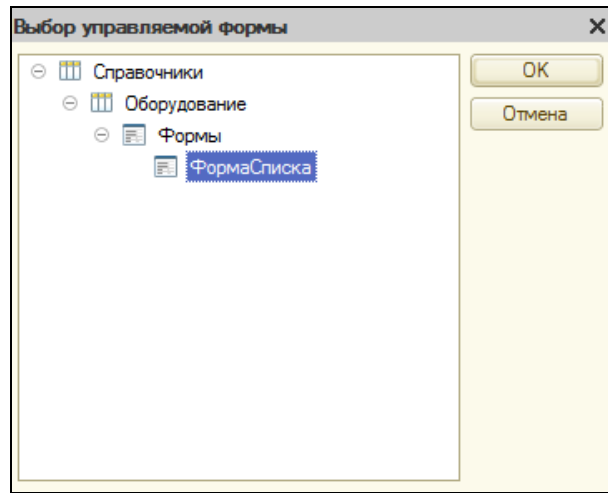


Рис. 47в. Выбор управляемой формы Рабочего стола

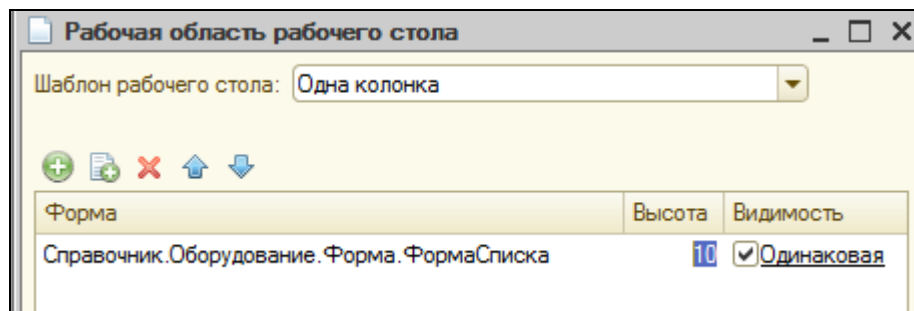


Рис. 47в. Рабочая область Рабочего стола после внесенных изменений

3. Далее отредактируем командный интерфейс Рабочего стола.

Выполним: Контекстное меню конфигурации – Открыть командный интерфейс основного раздела (рис. 48).

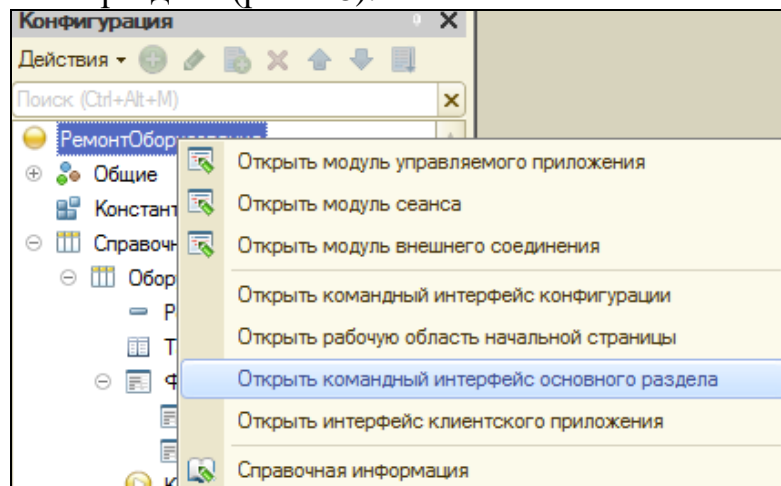


Рис. 48. Контекстное меню конфигурации – Открыть командный интерфейс основного раздела

Выполним:

– Из представленного списка удалим Список оборудования т.к., этот список уже есть на рабочем столе.

– Клиенты и Склады мышью переместим в группу **Панель навигации**.Смотри также.

- Поменяем местами АктПриемаВРемонт и АктВыдачиИзРемонта. Первый должен быть АктПриемаВРемонт, а за ним – АктВыдачиИзРемонта.
- На **Панель действий.Создать** добавим **Склады: создать**.
Выполненные выше действия должны быть представлены следующим образом (рис. 49):

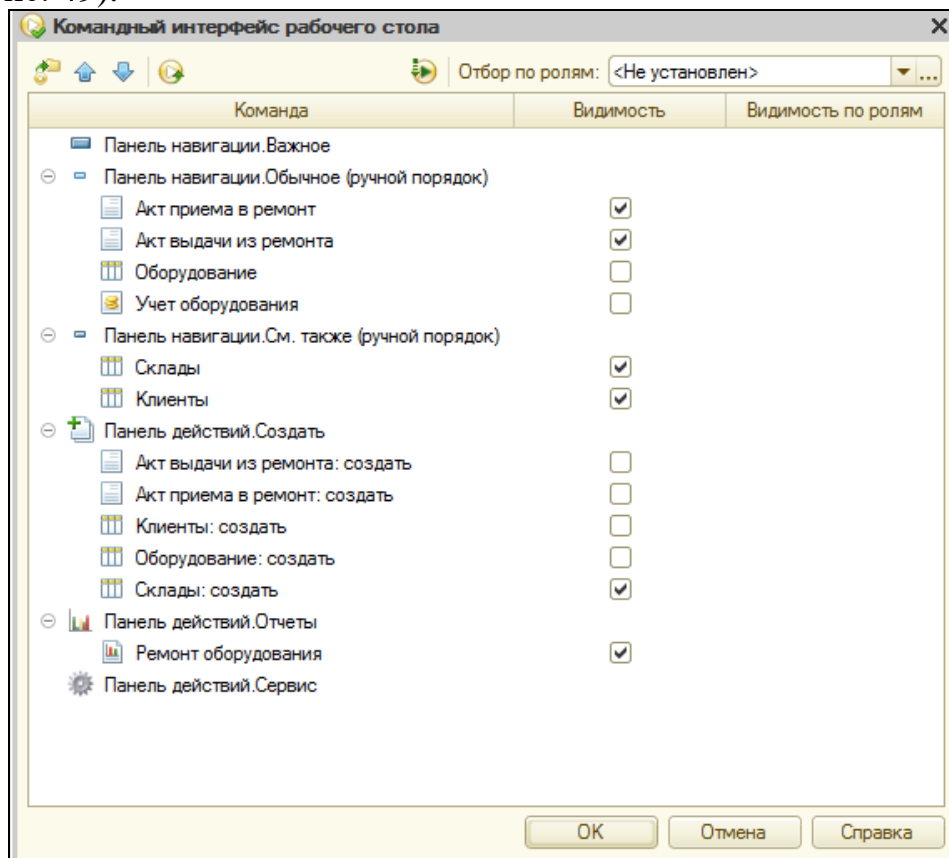
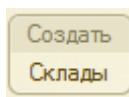


Рис. 48. Командный интерфейс рабочего стола

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие.



Создать склады звучит довольно странно.
Вернемся в конфигуризатор.

4. Откроем окно редактирования свойств справочника **Склады**. Зададим Представление объекта – Склад (рис. 49).

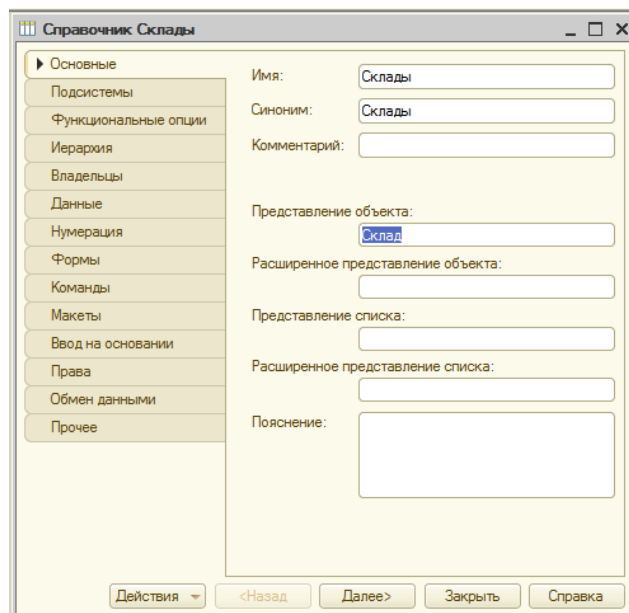
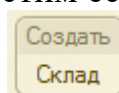


Рис. 49. Справочник Склады – Представление объекта

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие.



Кнопка поменяла свое название на

Внесем дальнейшие изменения.

Слева представлены Акт приема в ремонт и Акт выдачи из ремонта. На самом деле это два списка. Поэтому поменяем названия.

Вернемся в конфигуратор и сделаем несколько доработок.

5. Откроем окно редактирования свойств документа **АктПриемаВРемонт**. Зададим Представление списка – Прием в ремонт – Расширенное представление списка – Акты приема в ремонт (рис. 50а).

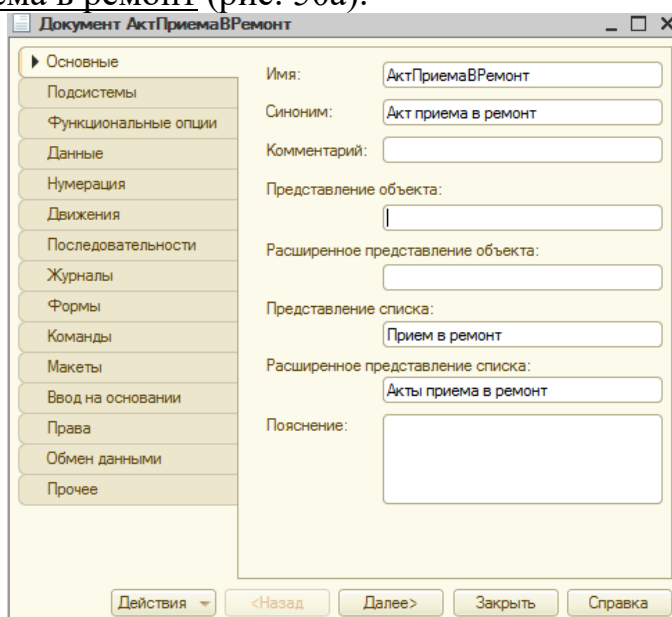


Рис. 50а. Документ Акт приема в ремонт – Представление списка

6. Тоже самое выполним для **АктВыдачиИзРемонта** (Выдача из ремонта – Акты выдачи из ремонта) (рис. 50б).

Рис. 50б. Документ Акт выдачи из ремонта – Представление списка

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие. Списки поменяли название – Прием в ремонт, Выдача из ремонта.

Обратим внимание на Рабочий стол, именно с него начинается работа. На Рабочем столе представлен список Оборудования, теперь можно осуществлять поиск по серийным номерам.

ВВОД ДАННЫХ В ДБ

В Загородный филиал привезли насос с серийным номером 984. Проверим, есть ли он в базе.


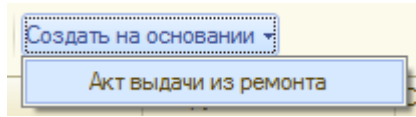
Нажимаем на кнопку . В появившейся форме выбираем и вводим данные (рис. 51):

Рис. 51. Форма поиска в справочнике Оборудование

Такого насоса в справочнике нет, поэтому вводим новый объект в справочник Оборудование – **Grundfos SPO 3-75A, с/н 984**.

Далее создадим на основании **Акт приема в ремонт**. Клиент – **Семен Семенович** (создать нового клиента). Склад – **Загородный филиал** (для этого есть команда **Создать склад**). Дата – **06.11.2011 г.** Провести. Закрыть.

Далее отремонтируем насос. Через неделю отдадим его клиенту (11.11.2011 г). Создадим **Акт выдачи из ремонта**. В Акты приема в ремонт отмечаем мышью необходимый документ. Далее нажимаем кнопку

, выбираем необходимое оборудование в справочнике Оборудование и вводим на основании Акт выдачи из ремонта (рис.52а, 52б).

Акты приема в ремонт

Создать Найти... Создать на основании Все действия

Дата	Номер	Клиент	Оборудование	Склад	Описание неисп.
05.11.2011 12:00:00	000000001	Насосный завод	Насос ЦМНШ-80.5, с/н 987254	Основной склад	Не работает с
06.11.2011 12:00:00	000000003	Семен Семенович	Grundfos SPO 3-75A, с/н 984	Загородный филиал	
12.11.2011 12:00:00	000000002	Насосный завод	КМ 0-32-180/2-5, с/н 7744	Основной склад	

Рис. 52а. Акты приема в ремонт – Ввод на основании Акт выдачи из ремонта

Акт выдачи из ремонта (создание) *

Провести и закрыть Провести Все действия

Номер:

Дата:

Клиент:

Оборудование:

Склад:

Выполненные работы:

Рис. 52б. Акт выдачи из ремонта (Ввод на основании Акта приема в ремонт)

Но на следующий рабочий день Семен Семенович вернул насос обратно, т.к. он опять сломался. Примем оборудование в ремонт еще раз (12.11.2011 г.) (рис. 53).

Рис. 53 Форма Акт приема в ремонт

Далее создадим Отчет.

Запустим отчет

Выбрать вариант отчета – Основной.

Поменять даты на

Сформировать отчет: Кнопка

Посмотрим в отчете только первый уровень группировки (вызовем контекстное меню и выберем Уровень 1) (рис. 54).

Параметры данных: Начало периода = 01.11.2011 0:00:00
Конец периода = 30.11.2011 0:00:00

Клиент	Склад	Количество Начальный остаток	Количество Приход	Количество Расход	Количество Конечный остаток
	Основной склад		2	1	1
	Основной склад		1	1	1
	Основной склад		2	1	1
	Загородный филиал		1	1	
	Основной склад		1		1
			4	2	2

Рис. 54. Контекстное меню – Выбор Уровня группировки отчета

В результате сформировался следующий отчет (рис. 55а):

Клиент		Количество Начальный остаток	Количество Приход	Количество Расход	Количество Конечный остаток
Оборудование	Склад				
Насосный завод			2	1	1
Семен Семенович			2	1	1
Итого			4	2	2

Рис. 55а. Отчет Приход – Расход

Из отчета видно, что за месяц было отремонтировано по 1 насосу для Насосного завода и для частного клиента, но у них было принято в ремонт по 2 насоса. Т.о. из отчета видно, что 2 насоса до сих пор остались в ремонте.

Если раскрыть группировки, то можно увидеть более подробную информацию: для Насосного завода предприятие ремонтировало два разных насоса, а для Семен Семеновича один и тот же насос (рис. 55б).

Клиент		Количество Начальный остаток	Количество Приход	Количество Расход	Количество Конечный остаток
Оборудование	Склад				
Насосный завод			2	1	1
Насос ЦМНШ-80.5, с/н 987254	Основной склад		1	1	
КМ 0-32-180/2-5, с/н7744	Основной склад		1		1
Семен Семенович			2	1	1
Grundfos SPO 3-75A, с/н 984	Загородный филиал		2	1	1
Итого			4	2	2

Рис. 55б. Отчет Приход – Расход

Примечание. Обратите внимание, что начало и конец периода можно выбирать прямо в отчете. Что очень удобно. Т.к. эти параметры были добавлены в быстрые пользовательские настройки.

По полученному отчету еще можно добавить, что он несет много информации для руководителя. Но для мастера гораздо важнее видеть, какие насосы сейчас находятся в ремонте.


Создадим второй вариант данного отчета.

Вернемся в конфигуратор.

1. Откроем **Схему компоновки данных Отчета**. Перейдем на закладку **Настройки**. Слева виден имеющийся вариант данного отчета. Сейчас вариант

один – **Основной**

Имя варианта	Представление
Основной	Основной

Чтобы не создавать для нового отчета все настройки заново. **Скопируем Основной вариант** (кнопка  на панели вверху слева схемы компоновки данных) и переименуем его в **Оборудование в ремонте**

Имя варианта	Представление
Основной	Основной
Оборудование в ремонт	Оборудование в рем...

2. Внесем некоторые изменения для данного варианта отчета. Отключим параметр – **Начало периода**. Уберем этот параметр из быстрых

пользовательских настроек (кнопка  справа на схеме компоновки данных) (рис. 56а).

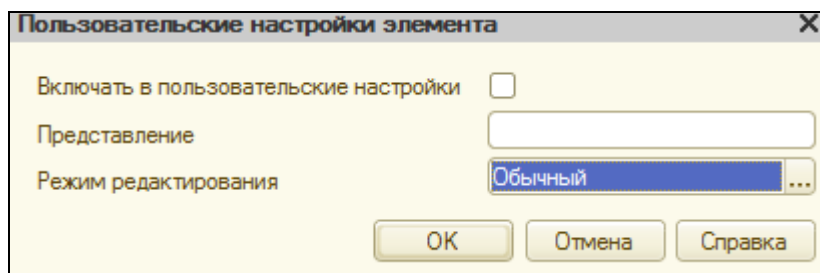



Рис. 56а. Пользовательские настройки элемента Начало периода

3. Для параметра **Конец периода** установим значение **Начало завтрашнего дня** (рис. 56б). Также уберем этот параметр из быстрых пользовательских настроек (кнопка  справа на схеме компоновки данных) (рис. 56а).

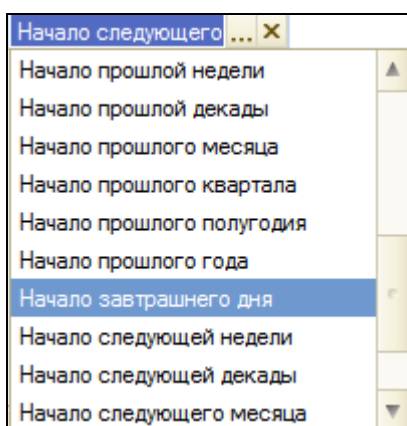



Рис. 56б. Настройка значения параметра Конец периода

4. Откроем вкладку **Выбранные поля**. Из Выбранных ресурсов удалим (кнопкой ) все, кроме **Остаток на конец периода** (рис. 57а).

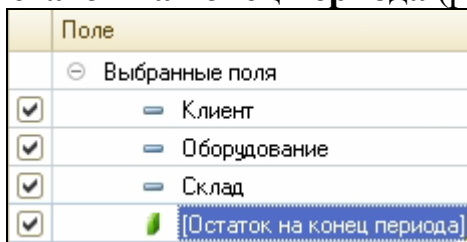


Рис. 57а. Настройка выбранных полей для отчета

5. Правой кнопкой мыши вызовем контекстное меню этого ресурса и выберем **Установить Заголовок** (рис. 57б) – **Количество** (рис. 57в). Именно так это поле будет показано в отчете (рис. 57г).

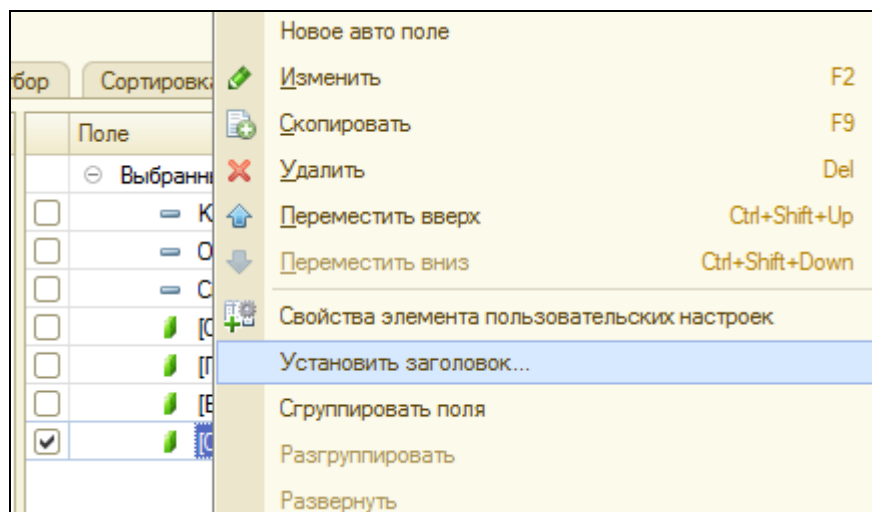


Рис. 57б. Настройка выбранных полей для отчета – Установить заголовок

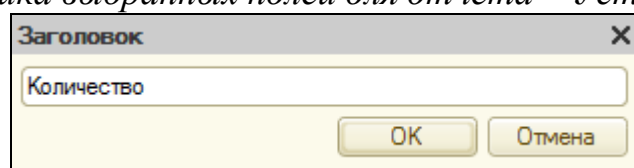


Рис. 57в. Настройка выбранных полей для отчета – Установить заголовок- Ввод названия заголовка

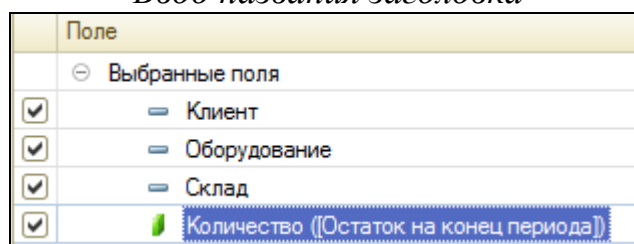


Рис. 57г. Настройка выбранных полей для отчета после внесенных изменений

6. Далее зададим **Отбор** (откроем вкладку Отбор). Укажем, что **Остаток на конец периода** должен быть **не равен 0** (рис. 57а, 57б).

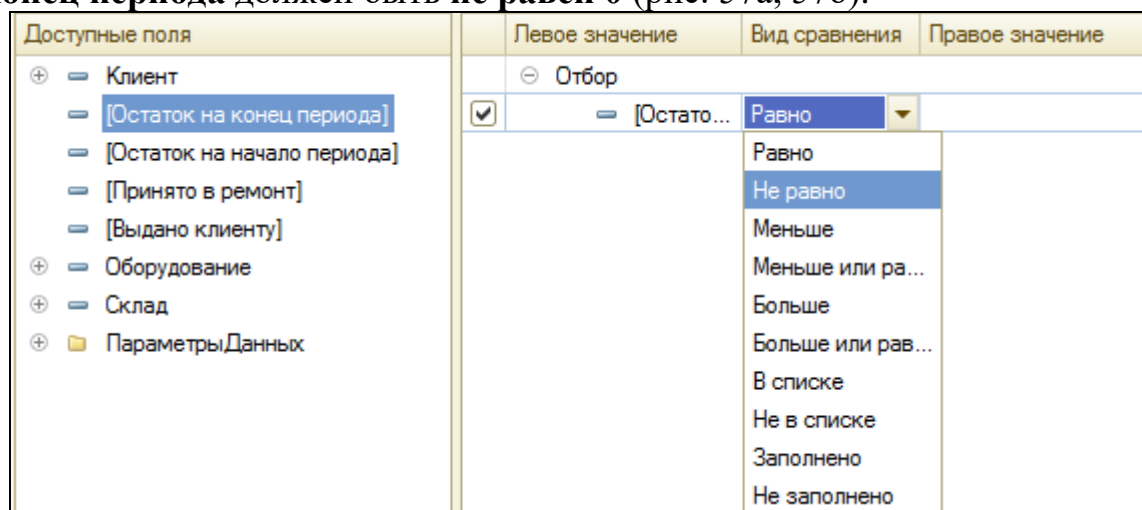


Рис. 57а. Значение параметров отбора для доступных полей

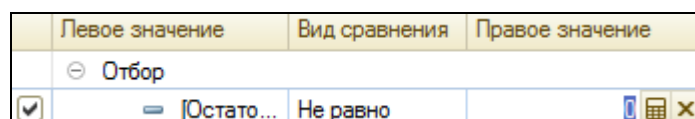


Рис. 57б. Значение параметров отбора для доступных полей

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие.

Запустить **Отчет**. Выбрать вариант отчета **Оборудование в ремонте**. Кнопка **Сформировать** (рис. 58).

Параметры данных: Конец периода = 01.04.2011 0:00:00	
Отбор: Количество Конечный остаток Не равно "0"	
Клиент	Количество
Оборудование	Склад
Насосный завод	1
КМ 0-32-180/2-5, с/н7744	Основной склад
Семен Семенович	1
Grundfos SPO 3-75A, с/н 984	Загородный филиал
Итого	2

Рис. 58. Отчет Оборудование в ремонт

Внесем некоторые изменения в отчет.

В полученном отчете не совсем хорошо, что также выводятся параметры данных и отбор. Кроме того, у отчета нет заголовка.

Вернемся в конфигуратор.

7. Откроем **Схему компоновки данных Отчета**. Перейдем на закладку **Другие настройки**. Укажем, что для отчета Оборудование в ремонте (обратите

на это внимание) **Выводить параметры данных и Выводить отбор не нужно**. Заголовок будет называться **Оборудование в ремонте** (рис. 59а).

Имя варианта	Представление
Основной	Основной
Оборудование в ремонт	Оборудование в рем...

<input checked="" type="checkbox"/>	Выводить заголовок	Выводить
<input checked="" type="checkbox"/>	Заголовок	Оборудование в ремонте
<input checked="" type="checkbox"/>	Выводить параметры данных	Не выводить
<input checked="" type="checkbox"/>	Выводить отбор	Не выводить

Рис. 59а. Выбор других настроек отчета Оборудование в ремонте

8. Выберем Макет оформления отчета – **Море** (рис. 59б).

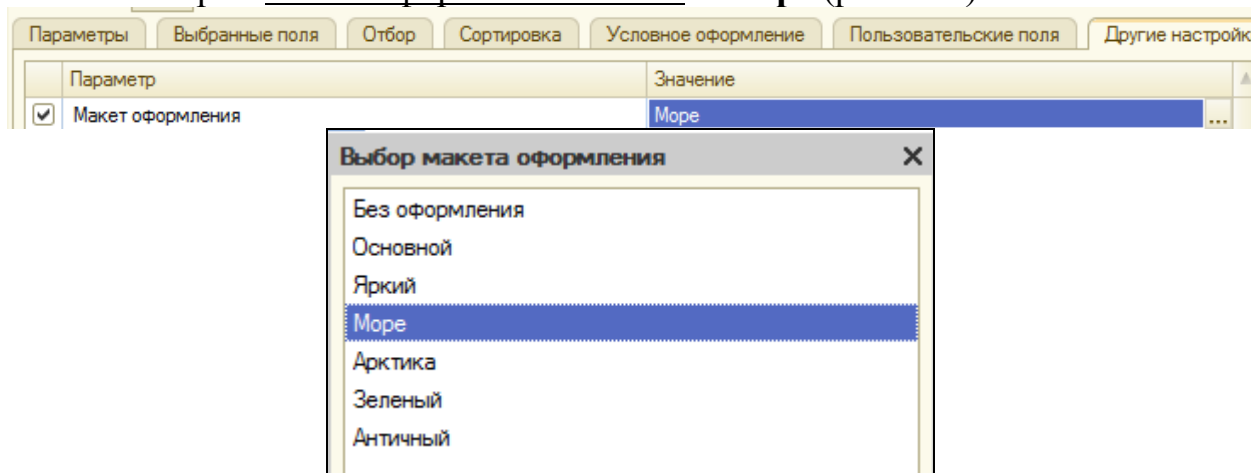


Рис. 59а. Выбор других настроек отчета Оборудование в ремонте – Макет оформления - Море

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие.

Выполним Отчет (рис. 60).

Оборудование в ремонте		
Клиент		Количество
Оборудование	Склад	
Насосный завод		1
КМ 0-32-180/2-5, с/н7744	Основной склад	1
Семен Семенович		1
Grundfos SPO 3-75A, с/н 984	Загородный филиал	1
Итого		2

Рис. 60. Отчет Оборудование в ремонт

Таким образом, было разработано простое приложение!!!

В целом прикладное решение получилось удобным.

Система была создана в визуальном режиме с использованием встроенных механизмов 1С: Предприятия – конструкторов движения, конструкторов форм, конструкторов запросов, других механизмов.

Применение этих механизмов позволяет в значительной степени отвлечься от деталей технологии создания форм и управляющих элементов, от написания кода для выполнения движений регистров и запросов, и решения других рутинных задач, и сосредоточить свое внимание на предметной области.

ЧАСТЬ 2. Доработка приложения

1. ДОБАВЛЕНИЕ ПОДСИСТЕМЫ

Подсистемы – это основные элементы для построения интерфейса 1С:Предприятия. Поэтому первое, с чего следует начинать разработку конфигурации, – это проектирование подсистем.

Наличие подсистем определяет весь пользовательский интерфейс, позволяет «рассортировать» документы, справочники ,отчеты по логически связанными с ними разделами и сотрудниками.

В простых прикладных решениях можно не использовать подсистемы. Т.к. в этой части будет рассмотрен пример более сложного приложения, разработаем несколько подсистем, чтобы разграничить функциональные задачи между разделами и сотрудниками.

Создадим в прикладном решении **три подсистемы:**

1. **Бухгалтерия** (для расчетов с клиентами),
2. **Оказание услуг** (для формирования документов приема и выдачи оборудования, закрепления мастера за конкретным заказом),
3. **Расчет зарплаты** (для расчета зарплаты сотрудникам).

Эти объекты располагаются в ветке объектов Общие и позволяют строить древовидную структуру, состоящую из подсистем и подчиненных подсистем.

Создадим подсистемы.

Вернемся в конфигуратор.

1. Правой кнопкой мыши щелкнем по **Общие** выбираем **Подсистемы** и нажимаем кнопку **Добавить** (рис. 61а).

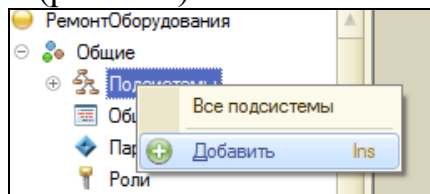


Рис. 61а. Фрагмент Дерево объектов конфигуратора – Добавление Подсистемы

2. Зададим имя подсистемы – **Бухгалтерия** (рис. 61б).

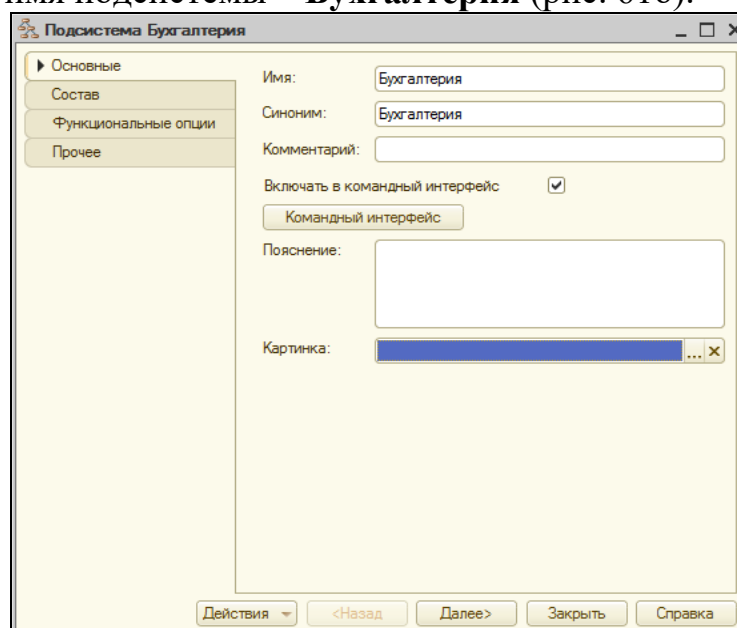


Рис. 61б. Установка имени и синонима подсистемы

3. Также зададим картинку для отображения подсистемы (рис. 61б). Для этого необходимо нажать на кнопку выбора в поле

Картинка

В окне **Выбор картинки** (рис. 61в) необходимо добавить картинку в список на закладке **Имя конфигурации**. Для этого нажмите на кнопку **Добавить** (рис. 61в).

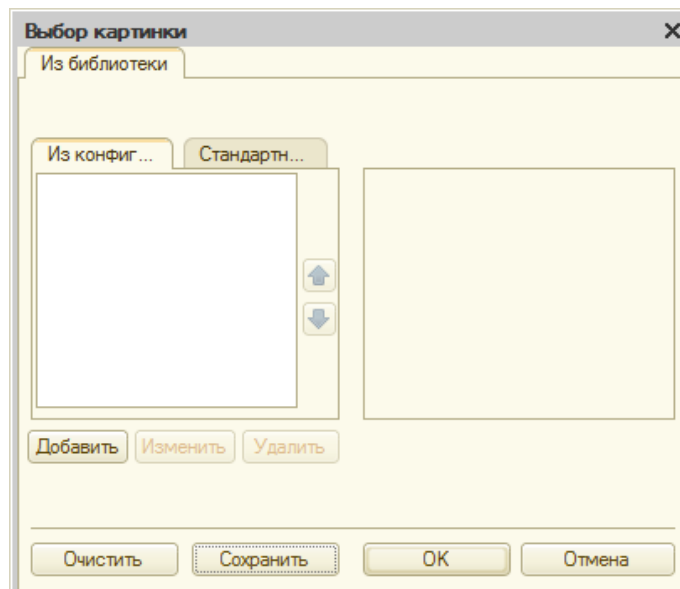


Рис. 61в. Выбор картинки для представления подсистемы

4. Дадим картинке имя **Бухгалтерия**. Чтобы задать картинку нажмите на кнопку **Выбрать из файла** (рис. 61г).

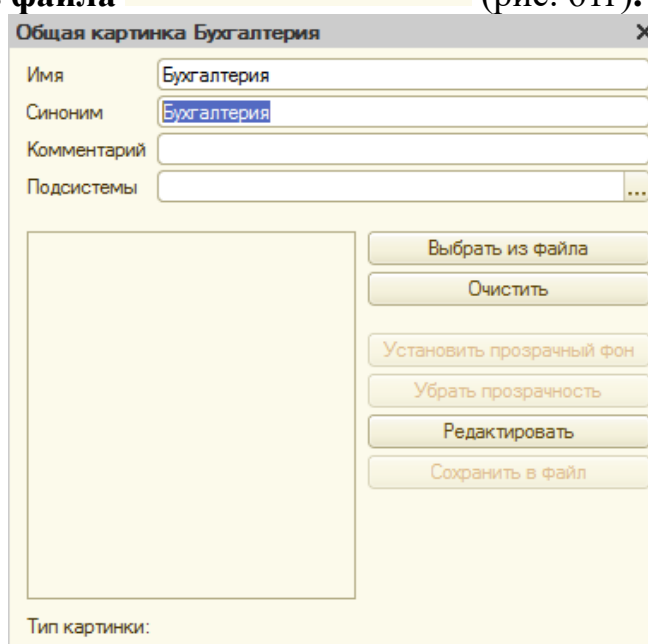


Рис. 61г. Окно редактирования объекта конфигурации

5. В папке Мои документы найдите папку **Image-1С**. Отметьте файл **Бухгалтерия** и нажмите кнопку **Открыть** (рис. 61д).

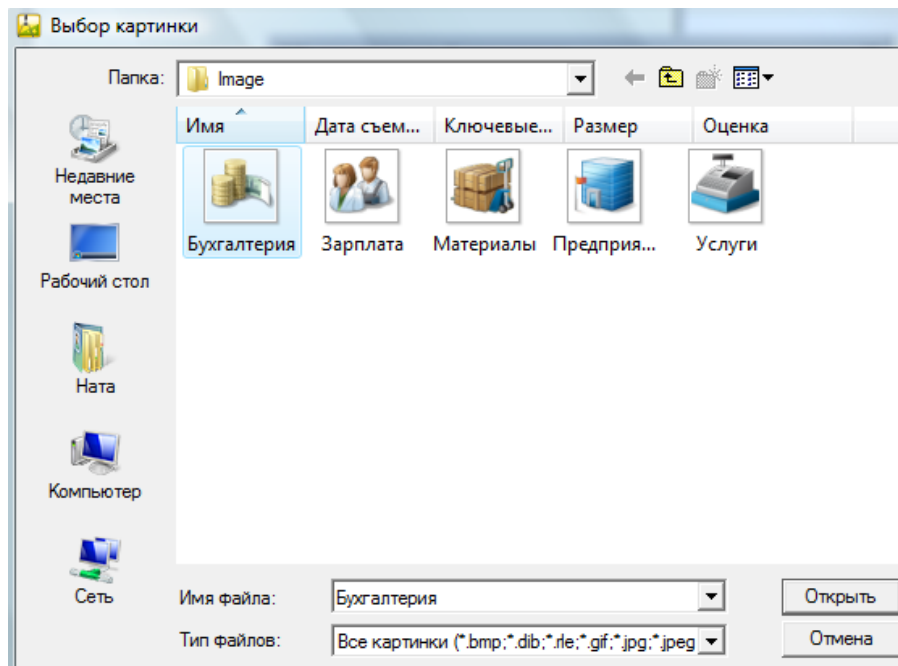


Рис. 61г. Выбор картинки для представления подсистем

Выбранная картинка должна появиться в списке картинок на закладке **Из конфигурации** (рис. 61д).

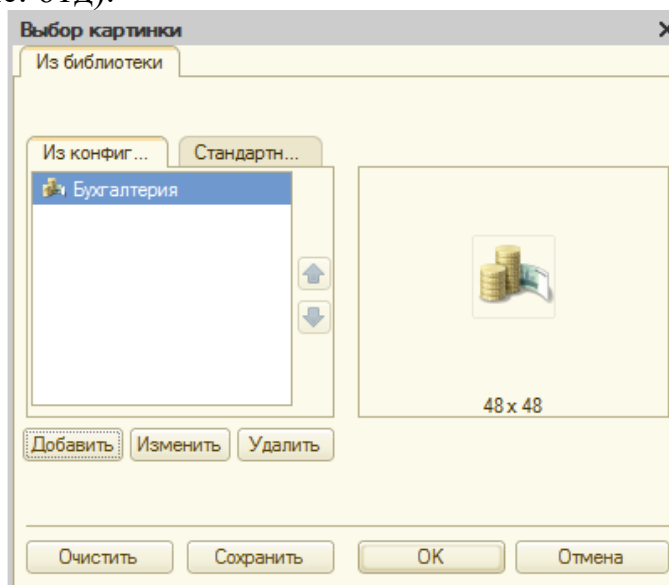


Рис. 61д. Выбор картинки для представления подсистемы

7. Нажать **ОК**.

Т.о. в дереве объектов конфигурации появилась новая подсистема

8. Аналогично создайте еще две подсистем **Оказание услуг**, **Расчет зарплаты**, выбрав для них картинки **Услуги**, **Зарплата**.

9. Изменим порядок подсистем. Выделите корень дерева объектов конфигурации и выберите пункт **Открыть командный интерфейс конфигурации** (рис. 61е).

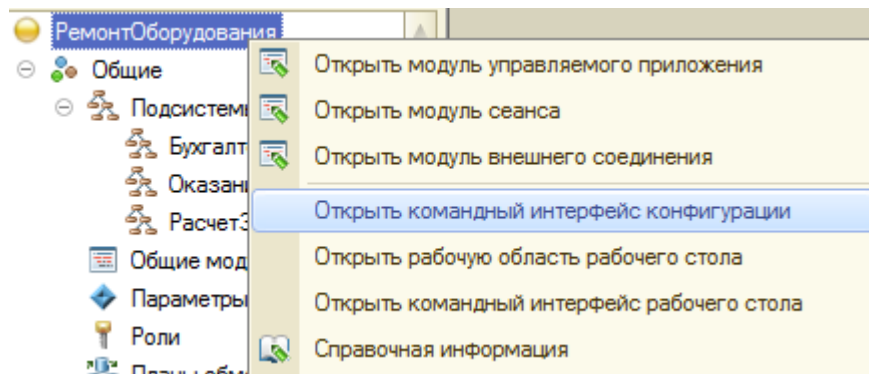



Рис. 61е. Вызов окна настройки командного интерфейса конфигурации

В открывшемся окне **Командный интерфейс** с помощью кнопок  изменим порядок, как представлено на рис. 61ж.

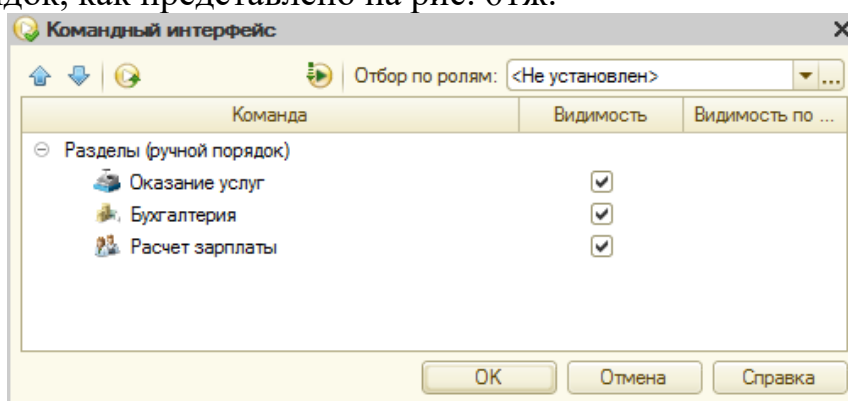


Рис. 61ж. Окно настройки подсистем

10. Определим принадлежность созданных ранее объектов подсистемам. Это можно сделать двумя способами:

- в дереве объектов конфигурации выбрать конкретную подсистему и на вкладке **Состав** отметить все необходимые объекты для этой подсистемы (рис. 61з).

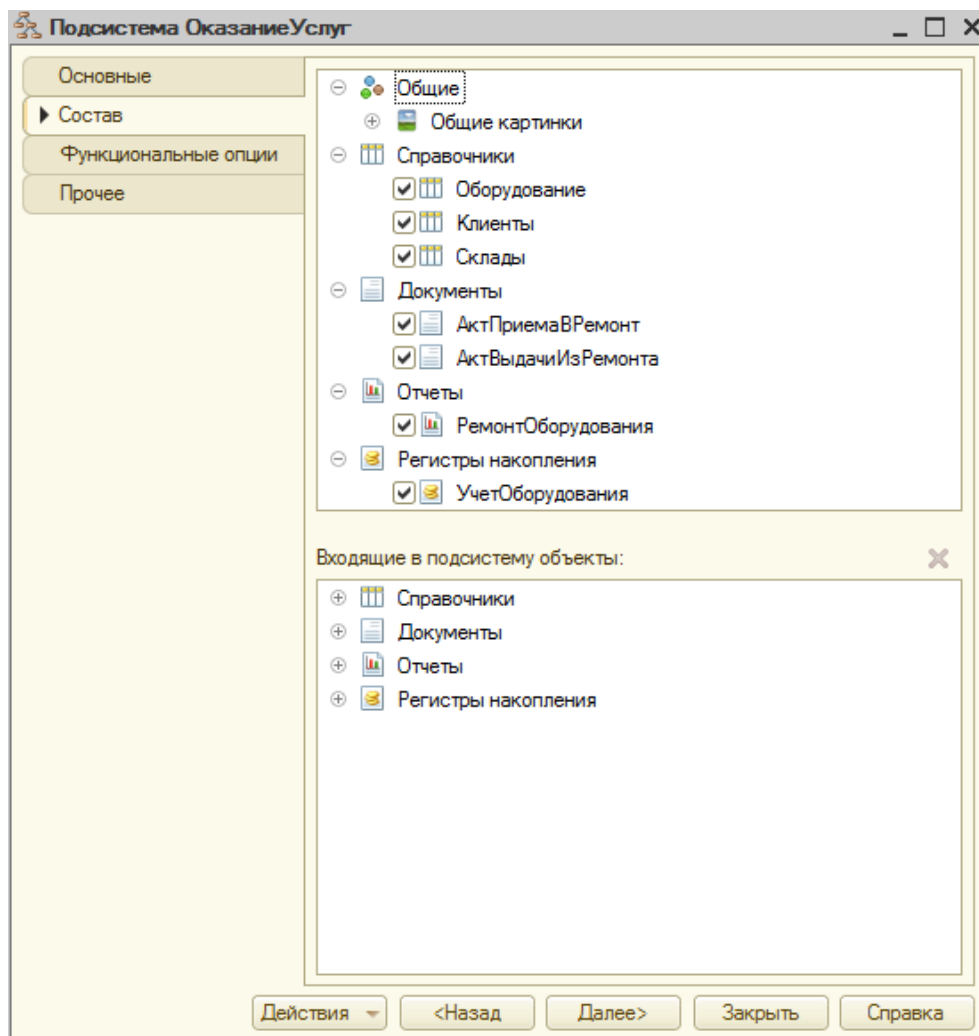


Рис. 61з. Настройка подсистемы

– в дереве объектов конфигурации выбрать **Подсистемы** далее **Все подсистемы** (рис. 61и).

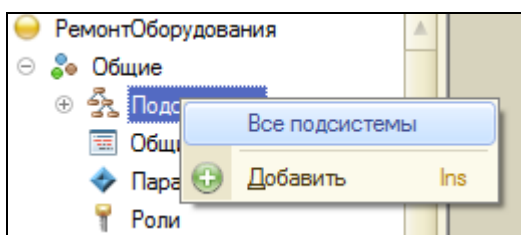


Рис. 61и. Фрагмент Дерево объектов конфигулятора – Общие – Подсистемы – Все подсистемы

Выберите подсистему **Оказание услуг**. Справой стороны формы **Все подсистемы** отметьте видимость команд для подсистемы **Оказание услуг**, как показано на рис. 61к.

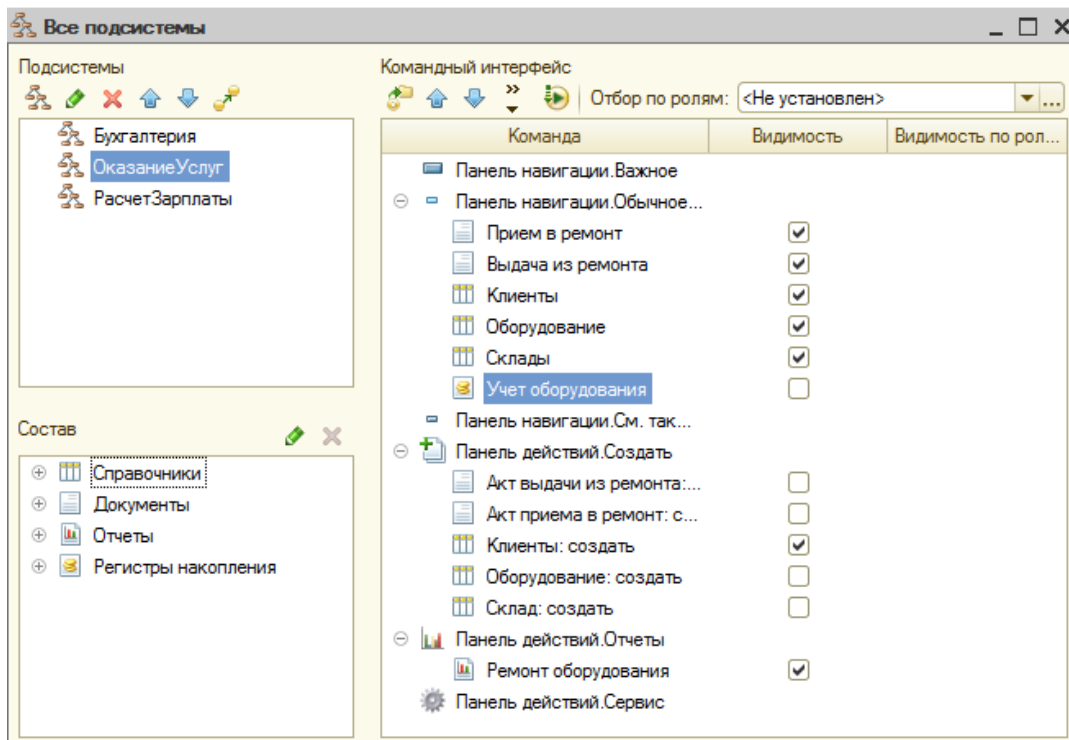


Рис. 61к. Окно настройки подсистемы

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие. На рис. 62. представлен вид разрабатываемого приложения.

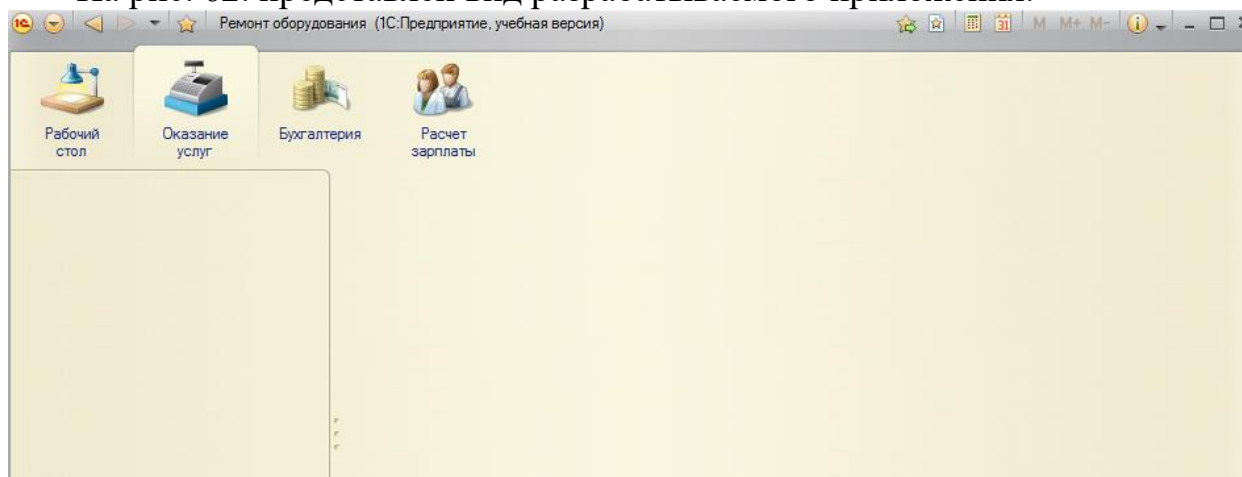


Рис. 62. Разделы прикладного решения

II. ПАНЕЛЬ НАВИГАЦИИ И ПАНЕЛЬ РАЗДЕЛОВ

На рис. 63. представлено Окно 1С: Предприятия раздел Оказание услуг. Слева в вертикальной области окна представлена **панель навигации**, которая отображает структуру выбранного раздела. Панель навигации предназначена для быстрого перехода к различным спискам в пределах выбранного раздела программы.

Кроме этого Окно 1С: Предприятия содержит **панель действий**. Панель действий содержит команды, которые соответствуют текущему разделу, выбранному в **панели разделов**. Эти команды объединены в **стандартные группы**: Создать, Отчеты, Сервис и группы, созданные разработчиком. Группа

Создать включает команды создания новых объектов базы, например, документов или элементов справочника.



Рис. 63. Окно 1С: Предприятие

Далее произведем доработку системы – усложним ее.

III. ПРОВЕРКА ЗАПОЛНЕНИЯ СТАНДАРТНЫХ РЕКВИЗИТОВ В СПРАВОЧНИКЕ

Вернемся в конфигуратор.

Откроем Справочник *Клиенты*, на закладке *Данные* нажмем кнопку **Стандартные реквизиты**. Выделим в списке реквизитов реквизит *Наименование*, вызовем его контекстное меню и выберем пункт *Свойства* (рис. 64а).

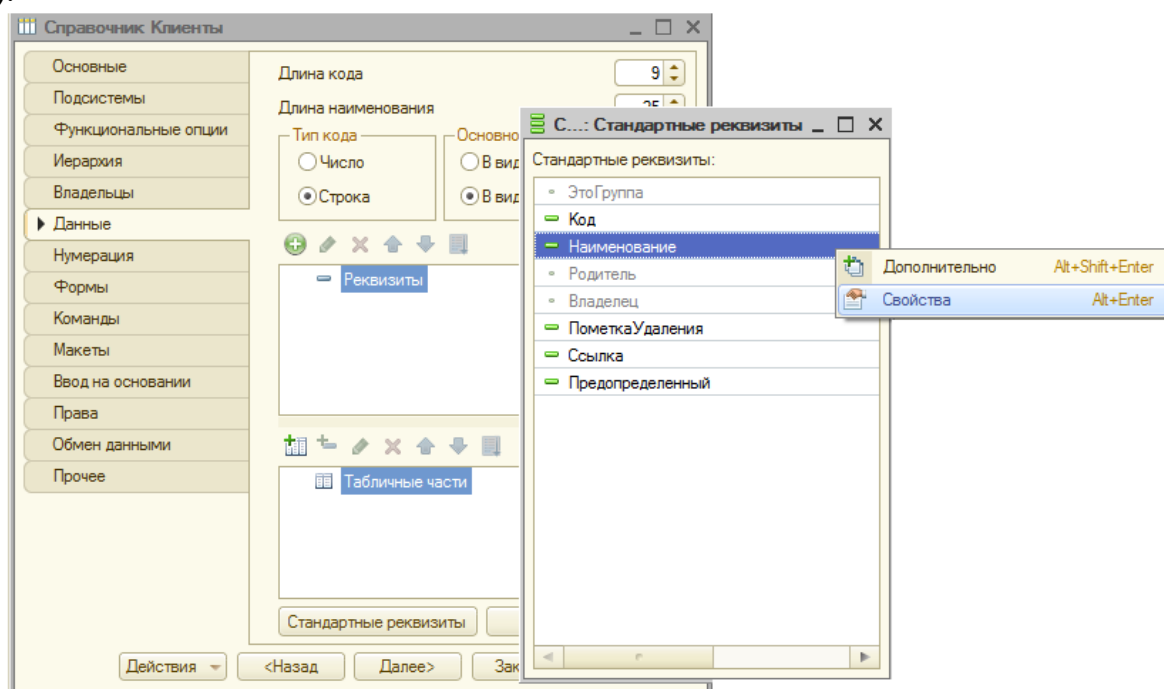


Рис. 64а. Состав стандартных реквизитов справочника

В палитре свойств стандартного реквизита **Наименование** видим, что свойство **Проверка заполнения** по умолчанию установлено значение **Выдавать ошибку** (рис. 64б). Это означает, что если реквизит Наименование не заполнен, то будет выведено сообщение об ошибке.

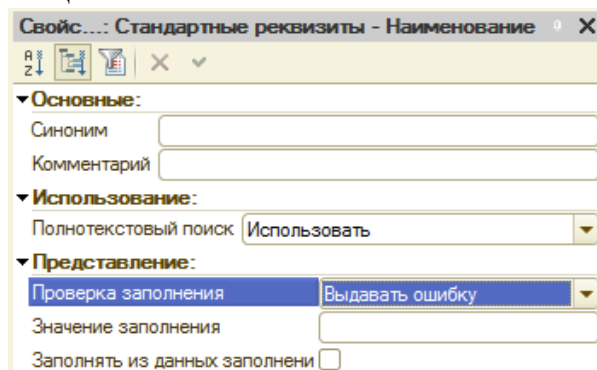


Рис. 64б. Палитра свойств стандартного реквизита Наименование

IV. ИЕРАРХИЧЕСКИЙ СПРАВОЧНИК

Для хранения информация по сотрудникам создадим новый справочник, который будет также указывать на принадлежность сотрудника к определенной структурной единице (отделу, подразделению, цеху и т.д.).

1. Создадим иерархический справочник с реквизитами с наименованием **Подразделения** (рис. 65а).

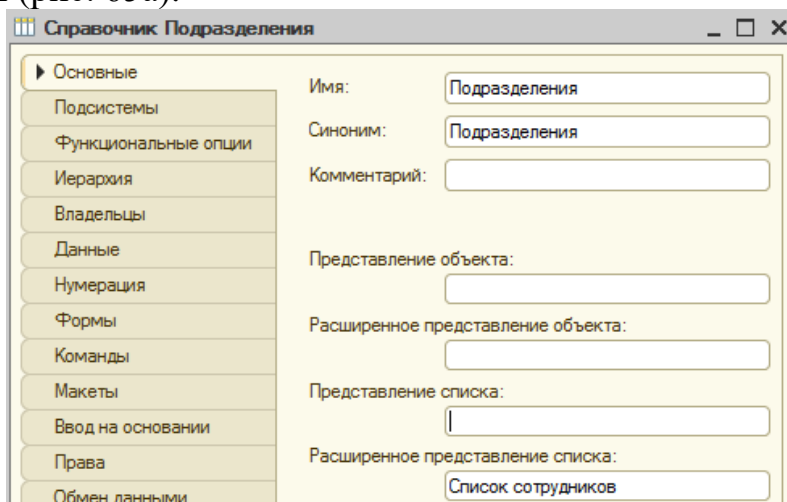


Рис. 65а. Установка наименования справочника Подразделения

2. Отметим, что справочник участвует во всех подсистемах Бухгалтерия, Оказание услуг, Расчет зарплаты (рис. 65б).

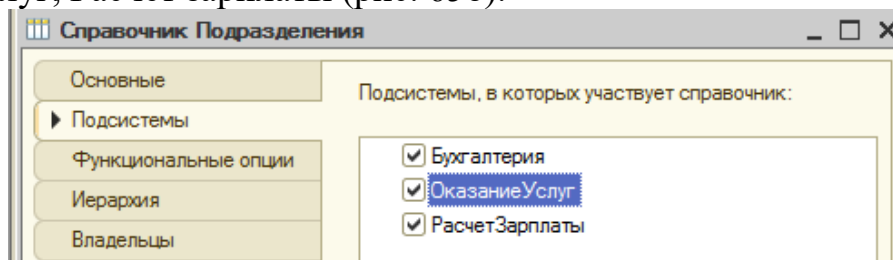


Рис. 65б. Определение списка подсистем, в которых участвует справочник

3. На закладке **Иерархия** установим флажок **Иерархический справочник**, Количество уровней иерархии – 2, ограничение количества уровней иерархии (рис. 65в).

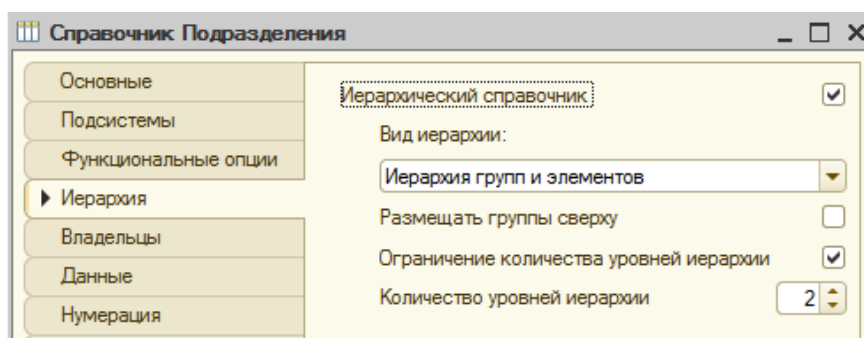


Рис. 65в. Установка признака иерархического справочника

4. Установим видимость создания объектов справочника в подсистеме **Расчет зарплаты** (рис. 65г).

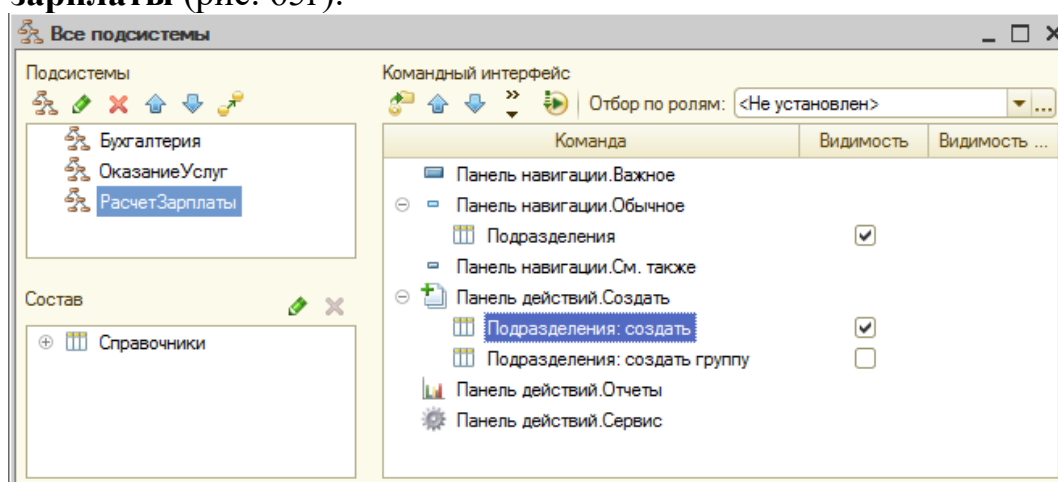


Рис. 65г. Окно настройки подсистем

5. Кроме типовых реквизитов справочник должен содержать другие данные (рис. 65д). На закладке **данные** создайте новые реквизиты:

- наименование – Должность, тип данных – строка, длина – 20;
- наименование – дата рождения, тип данных – дата;
- наименование – оклад, тип данных – число, длина – 6, неотрицательное.

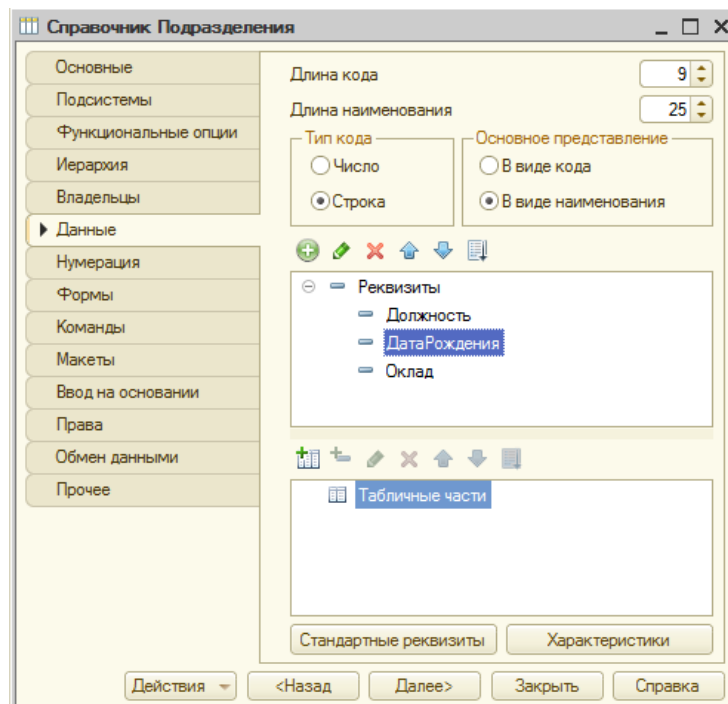


Рис. 65д. Добавление новых реквизитов в справочник

6. При создании нового справочника автоматически создаются два реквизита: код и наименование. Реквизит Код необходим, а вот наименование не совсем логичное название реквизита. Заменяем название реквизита Наименование на реквизит **ФИО**. Для этого откроем вкладку **Формы** и добавим **Форму элемента справочника** (рис. 65е).

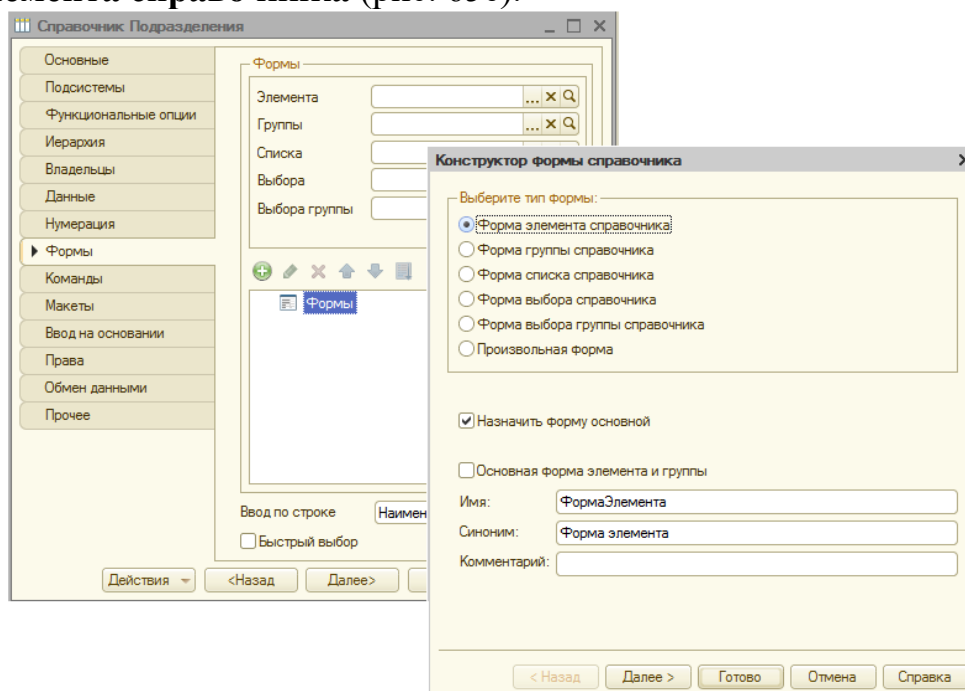


Рис. 65е. Конструктор формы справочника

Нажимаем кнопку **Далее**. На экране откроется список реквизитов формы (рис. 65ж). В списке ничего не меняем. Нажимаем на кнопку **ОК**.

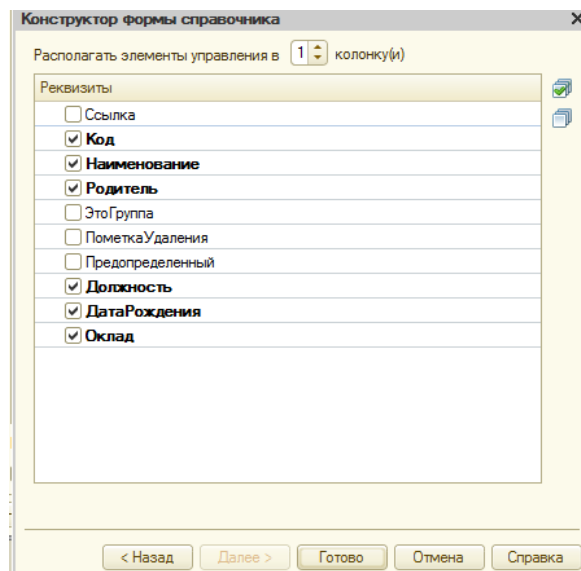


Рис. 65ж. Список элементов (реквизитов) формы справочника

7. На экране откроется Окно редактирования форм, которое имеет три области: реквизиты, типы реквизитов, внешний вид формы (рис. 65з).

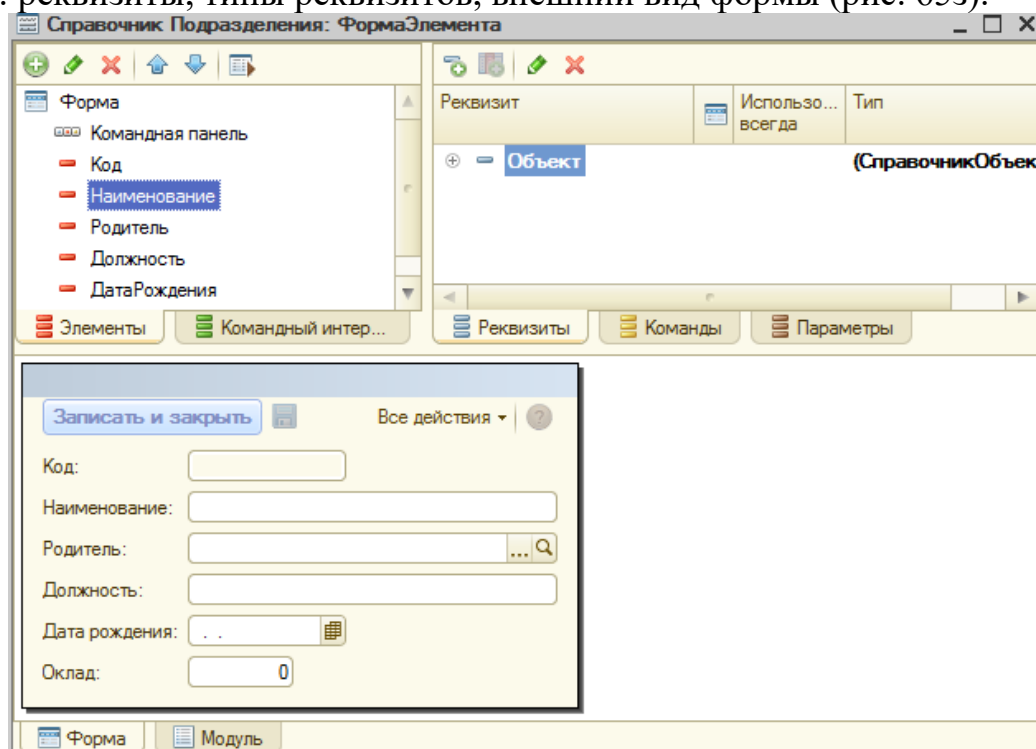


Рис. 65з. Окно редактирования форм

В верхнем левом поле, где представлены реквизиты, выделите реквизит Наименование. Откройте свойства данного поля и замените заголовок на ФИО (рис. 65и).

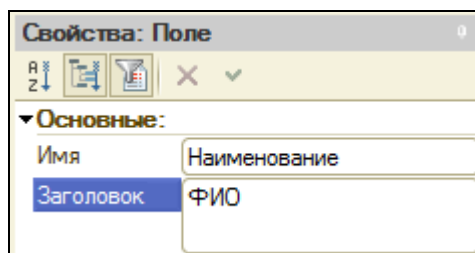



Рис. 65и. Фрагмент Свойства реквизита Наименование Формы справочника Подразделения

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие.

Созданный справочник имеет иерархическую структуру:

- вначале создаем подразделения нажатием на кнопку  (рис. 65к);

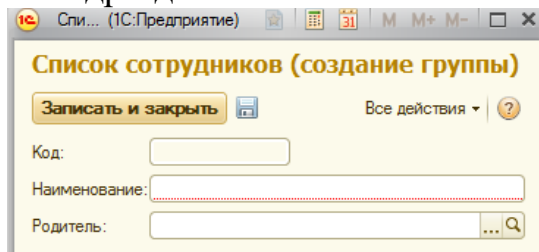


Рис. 65к. Форма создания новой группы в справочнике Подразделения

- потом заносим информацию по сотруднику (рис. 65л).

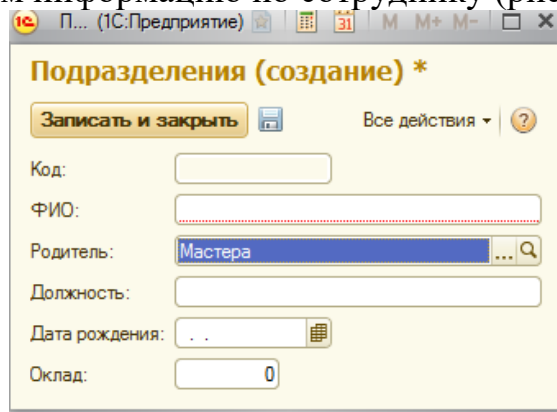


Рис. 65л. Форма создания нового сотрудника в справочнике Подразделения

ВВОД ДАННЫХ В БД

1. Введем в разработанный справочник 4 подразделения: Дирекция, Бухгалтерия, Мастера, Склад (рис. 65м).

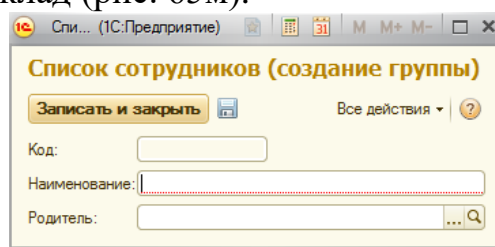


Рис. 65м. Форма создания группы справочника Подразделения

2. В каждое подразделение введем данные по сотрудникам: Дирекция – 3 человека, Бухгалтерия – 2 человека, Мастера – 8 человек, Склад – 2 человека. Данные для ввода сформулируйте самостоятельно по смыслу разрабатываемого решения (рис. 65н).

Рис. 65н. Форма создания элемента справочника Подразделения

V. ИЕРАРХИЧЕСКИЙ СПРАВОЧНИК С ТАБЛИЧНОЙ ЧАСТЬЮ И СВЯЗЬ С ДОКУМЕНТОМ

Создадим еще один справочник **Номенклатура услуг**. Данный справочник будет содержать информацию об услугах, которые оказывает предприятие, и о тех мастерах, которые их выполняют.

Данный справочник будет сложный.

Кроме этого свяжем справочник Номенклатура услуг с документом Акт приема в ремонт. Создадим у данного документа табличную часть так, чтобы наименование услуги можно было выбрать из справочника.

1. Введите необходимые данные по справочнику, как показано на рис.

Рис. 66а. Установка наименования справочника Номенклатура услуг

Рис. 66б. Определение списка подсистем, в которых участвует справочник

Рис. 66в. Установка признака иерархического справочника

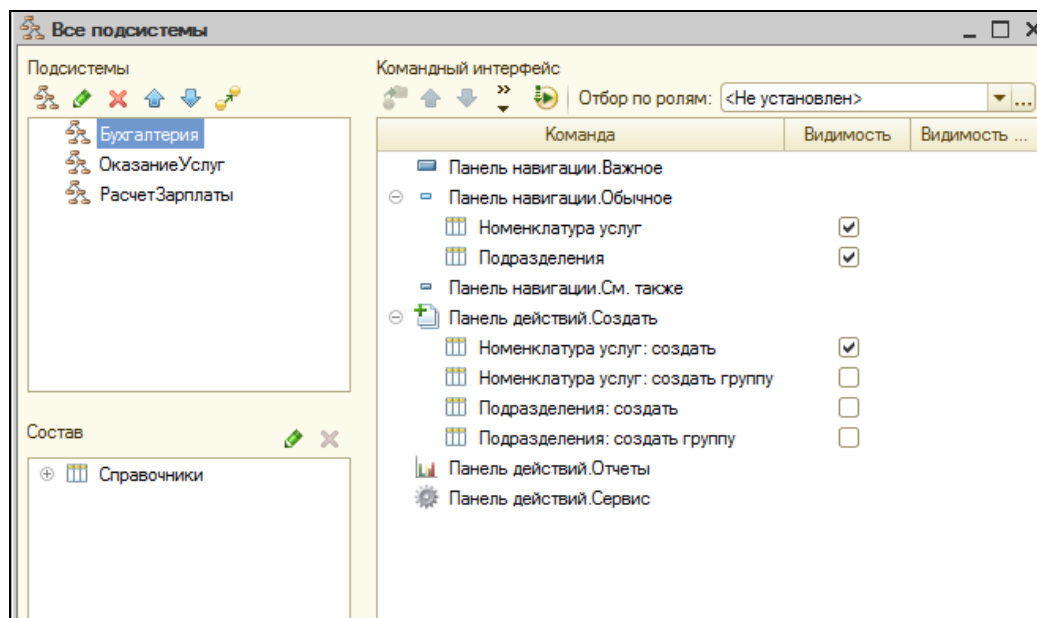


Рис. 66г. Окно настройки подсистем

2. Откроем вкладку **Данные** документа **АктПриемаВРемонт** и добавим в табличную часть **Услуги**. Создадим следующие реквизиты табличной части для Услуги (рис. 66д):

- Имя – **Услуга**, Тип данных – **СправочникСсылка.НоменклатураУслуг** (рис. 66е).
- Имя – **Цена**, Тип данных – **Число**, Флаг – **неотрицательное**, Точность – **2** (рис. 66ж).
- Имя – **Количество**, Тип данных – **Число**, Флаг – **неотрицательное**, Точность – **0** (рис. 66з).
- Имя – **Сумма**, Тип данных – **Число**, Флаг – **неотрицательное**, Точность – **2** (рис. 66и).

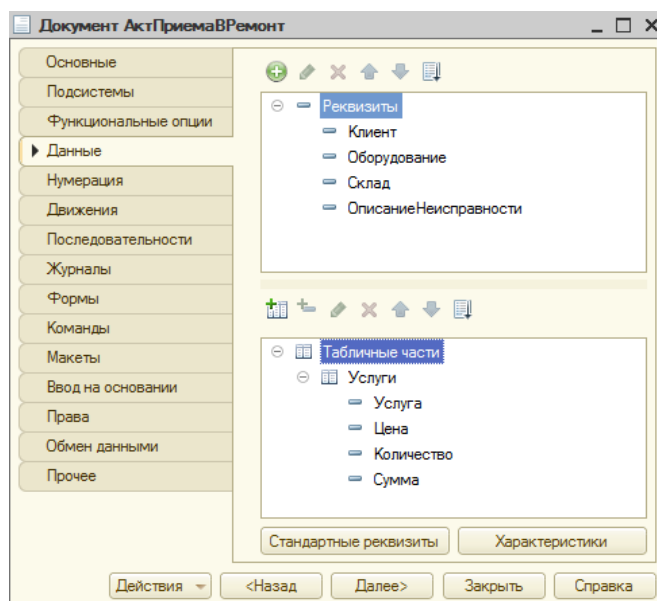


Рис. 66д. Создание реквизитов табличной части

Рис. 66е. Свойства реквизита табличной части Услуга

Рис. 66ж. Свойства реквизита табличной части Цена

Рис. 66з. Свойства реквизита табличной части Количество

Рис. 66и. Свойства реквизита табличной части Сумма

3. Для ввода данных нам понадобится форма документа. Нажмем на закладку **Формы**. Нажимаем на кнопку **Просмотр поля** Форма документа. На экране появилось окно **Конструктор формы документа** (рис. 67а). Выбираем Форма документа. Нажимаем на кнопку **Далее**. Открывается окно с

реквизитами создаваемой формы. Ничего не меняем. Нажимаем на кнопку Готово (рис. 67б).

Рис. 67а. Конструктор формы документа

Рис. 67б. Конструктор формы документа – Реквизиты формы

4. На экране откроется **Окно редактирования форм**, которое имеет три области: реквизиты, типы реквизитов, внешний вид формы (рис. 67в).

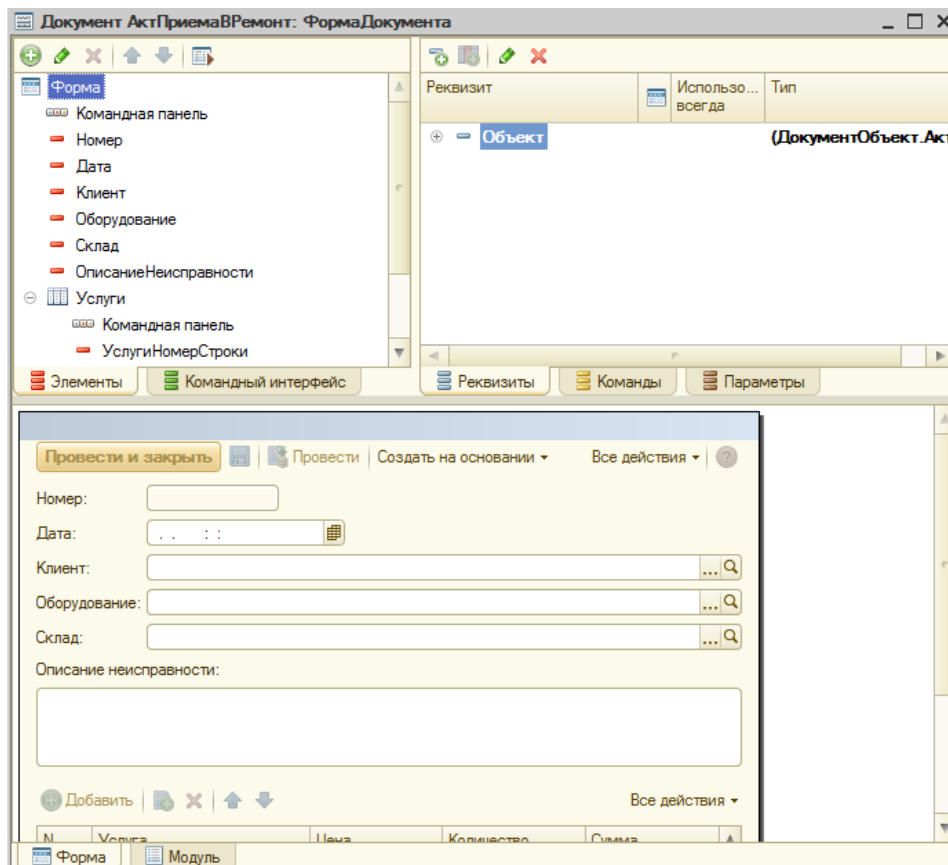


Рис. 67в. Окно редактирования формы документа

В верхнем левом поле представлены реквизиты документа Акт приема в ремонт (рис. 67г).

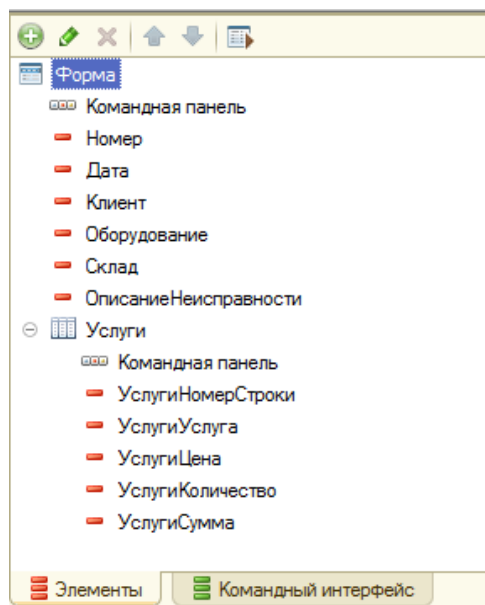


Рис. 67г. Элементы документа Акт приема в ремонт

При заполнении документа приходится вводить сумму в каждой строке. Это неудобно, и возникает естественное желание автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества.


Т.е. необходимо написать на встроенном языке команду, похожую на $\text{Сумма} = \text{Количество} * \text{Цена}$, которая будет выполняться при изменении значения поля Количество или Цена.

Создание процедуры обработки события в модуле формы.

У системы существуют события, которые связаны с самыми различными моментами ее «стандартного» функционирования. В том числе события, связанные с функционированием форм и элементов, расположенных в этих формах.


Используя встроенный язык, разработчик может «вклиниться» в эти события и описать собственный алгоритм того, что должно происходить при наступлении этого события.

5. Дважды щелкните на элемент формы **УслугиКоличество** (рис. 67г) и откройте для него **палитру свойств** (пункт контекстного меню Свойства). Прокрутите список до конца, и вы обнаружите перечень событий, которые могут быть связаны с этим полем ввода.

Прокрутите список до конца, из перечня событий выберите События ПриИзменении . Нажмите на значок лупы в поле ввода. Система создаст шаблон процедуры обработки этого события в модуле формы и откроет закладку Модуль редактирования формы (рис. 68).

```
НаКлиенте
Процедура УслугиКоличествоПриИзменении(Элемент)
    // Вставить содержимое обработчика.
КонiecПроцедуры
```

Рис. 68. Шаблон обработчика события «ПриИзменении» поля табличной части «Количество»

Для перехода между формой и модулем воспользуйтесь закладками , расположенными в нижней части Окна редактирования формы документа (рис. 67в).

Модуль – это «хранилище» для текста программы на встроенном языке. В данном случае это модуль формы, т.к. обработчики всех интерактивных событий, связанных с элементами формы, помещаются именно в модуль формы.

До процедуры УслугиКоличествоПриИзменении напомним процедуру расчета **Суммы**, которую необходимо будет вызывать при изменении Количества или Цены.

```
НаКлиенте
Процедура РассчитатьСумму(СтрокаТабличнойЧасти)
    СтрокаТабличнойЧасти.Сумма=СтрокаТабличнойЧасти.Цена*СтрокаТабличнойЧасти.Количество;
КонiecПроцедуры
```

В первой части строки создадим переменную СтрокаТабличнойЧасти, в которую будет помещен объект, содержащий данные, находящиеся в строке табличной части, которую необходимо пересчитать. Здесь не надо объявлять

переменные заранее. Создаем переменную прямо по ходу работы, и ее тип определяется типом значения, которое она содержит.

Данная процедура должна вызываться каждый раз при изменении **Количества**.

В модуль формы рис. 68 добавим следующий текст для **УслугиКоличествоПриИзменении**:

```
НаКлиенте
Процедура УслугиКоличествоПриИзменении (Элемент)
    РассчитатьСумму (Элементы.Услуги.ТекущиеДанные) ;
КонечПроцедуры
```

Т.к. сейчас мы находимся в модуле формы, в нем доступны все свойства и методы объекта встроенного языка. В данном случае мы обращаемся к коллекции элементов формы, используя свойство **Элементы**.

Коллекция элементов формы является объектом встроенного языка **ВсеЭлементыФормы**, содержащим все элементы формы. Т.е. это программный аналог корня дерева элементов формы.

Каждый элемент формы можно получить, указав его имя в качестве свойства этого объекта, т.е. через точку от него. В данном случае обращение идет к табличной части Акт приема в ремонт (Элементы.Услуги).

Табличная часть документа представляет собой объект встроенного языка **ТаблицаФормы**. Получив ту строку ,в которой в настоящее время осуществляется редактирование ,можно при помощи свойства программного объекта **ТаблицаФормы** – **ТекущиеДанные** (Элементы.Услуги.ТекущиеДанные).

6. Аналогично создадим обработчик событий **ПриИзменении** для **УслугиЦенаПриИзменении**:

```
НаКлиенте
Процедура УслугиЦенаПриИзменении (Элемент)
    РассчитатьСумму (Элементы.Услуги.ТекущиеДанные) ;
КонечПроцедуры
```

Данная процедура должна вызываться каждый раз при изменении **Цены**.

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие. Проверим работу внесенных изменений.

ВВОД ДАННЫХ В БД

1. В справочнике **НоменклатураУслуг** ведем 3 группы услуг: Кузовной цех, Малярный цех, ТрейдИН (рис. 67д).

Рис. 67д. Форма создания группы справочника **Номенклатура услуг**

2. В каждую группу услуг ведем не менее 6 записей. Данные для ввода сформулируйте самостоятельно по смыслу разрабатываемого решения (рис. 67е).

Рис. 67е. Форма создания элемента справочника Номенклатура услуг

3. Внесите не менее 10 Актв приема в ремонт (рис. 67ж).

Рис. 67ж. Форма для ввода данных по документу Акт приема в ремонт

VI. ОДНА ПРОЦЕДУРА ДЛЯ ОБРАБОТКИ НЕСКОЛЬКИХ СОБЫТИЙ

В рассмотренном выше примере два раза был использован один и тот же программный код при обработке событий.

В дальнейшем также может пригодиться процедура расчета суммы в других документах. Поэтому поместим расчет суммы в некоторое «общедоступное» место, чтобы разные документы, имеющие аналогичные реквизиты в табличной части, могли использовать этот алгоритм.

1. В дереве объектов конфигурации выбрать **Общие** далее **Общие модули**, кнопка **Добавить** (рис. 68а).

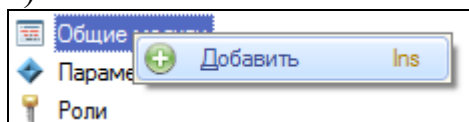


Рис. 68а. Выбор объекта конфигурации Общие модули

2. Назовем новый общий модуль РаботаСДокументами (рис. 68б). Установим в свойствах флажок Клиент (управляемое приложение), а флажок Сервер снимем. Это значит, что экземпляры этого модуля будут скомпилированы в контексте тонкого клиента и в контексте веб-клиента.

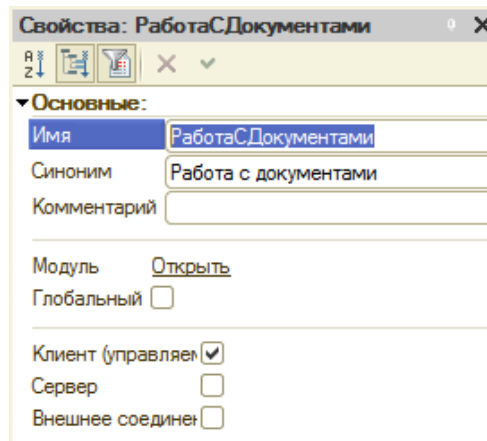


Рис. 68б. Свойства объекта конфигурации Общие модули

3. Внесем в модуль следующий текст:

```

&НаКлиенте
[ Процедура РассчитатьСумму (СтрокаТабличнойЧасти) Экспорт
    СтрокаТабличнойЧасти.Сумма=СтрокаТабличнойЧасти.Количество*СтрокаТабличнойЧасти.Цена;
КонецПроцедуры

```

В процедуру РассчитатьСумму() передаем переменную СтрокаТабличнойЧасти, которая была определена в обработчике событий ПриИзменении полей Количество и Цена документа Акт приема в ремонт.

Теперь, используя эту процедуру, можно получить доступ к данным колонок табличной части и рассчитывать сумму как произведение цены на количество.

Ключевое слово Экспорт в заголовке процедуры указывает, что эта процедура может быть доступна из других программных модулей.

4. Внесем изменения в Модуль Формы документа Акт приема в ремонт для полей Количество и Цена.

```

&НаКлиенте
[ Процедура УслугиКоличествоПриИзменении (Элемент)
    СтрокаТабличнойЧасти=Элементы.Услуги.ТекущиеДанные;
    РаботаСДокументами.РассчитатьСумму (СтрокаТабличнойЧасти);
КонецПроцедуры

&НаКлиенте
[ Процедура УслугиЦенаПриИзменении (Элемент)
    СтрокаТабличнойЧасти=Элементы.Услуги.ТекущиеДанные;
    РаботаСДокументами.РассчитатьСумму (СтрокаТабличнойЧасти);
КонецПроцедуры

```

В результате выполнения первой строки процедуры обработчика переменная СтрокаТабличнойЧасти будет содержать объект ДанныеФормыСтруктура. Этот объект содержит данные ,находящиеся в текущей строке табличной части документа (Элементы.Услуги.ТекущиеДанные).

Во второй строке вместо непосредственного расчета суммы вызываем процедуру РассчитатьСумму() из общего модуля РаботаСДокументами и передаем ей в качестве параметра текущую строку табличной части.

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие.

ВВОД ДАННЫХ В БД

1. Внесите еще 2 записи в Акты приема в ремонт для того, чтобы проверить работу приложения после изменений (рис. 67ж).

VIII. ДОКУМЕНТ «ОКАЗАНИЕ УСЛУГИ»

Создадим документ, в котором будут храниться сведения о мастере, который выполняет услугу клиенту, и цена услуги.

1. Добавим новый Документ – **ВыполнениеУслуги**. Введем данные, как показано на рис. 69. Обратите внимание на Представление списка – **Оказание услуги**, так документ будет называться в приложении.

Рис. 69а. Установка основных свойств документа

2. Отметим необходимые **подсистемы** (рис. 69б).

Рис. 69б. Определение списка подсистем

3. На закладке **Данные** создадим реквизиты документа (рис. 69в):

– **Клиент** – СправочникСсылка.Клиенты. Установим свойство **Проверка заполнения** в значение **Выдавать ошибку**;

– **Оборудование** – Справочник.Ссылка.Оборудование. Установим свойство **Проверка заполнения** в значение **Выдавать ошибку**;

- Склад – СправочникСсылка.Склады. Выбрем для свойства Значение заполнение predetermined элемент Основной склад справочника.
- Мастер – Справочник.Ссылка.Подразделения. Установим свойство **Проверка заполнения** в значение **Выдавать ошибку**.

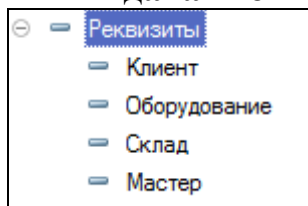


Рис. 69в. Реквизиты документа *ВыполнениеУслуги*

В табличной части документа создадим **ПереченьНоменклатуры** с реквизитами (рис. 69г):

- Услуга, тип СправочникСсылка.НоменклатураУслуг;
- Количество, тип Число, длина 15, точность 3, неотрицательное;
- Цена, тип Число, длина 15, точность 2, неотрицательное;
- Сумма, тип Число, длина 15, точность 3, неотрицательное.

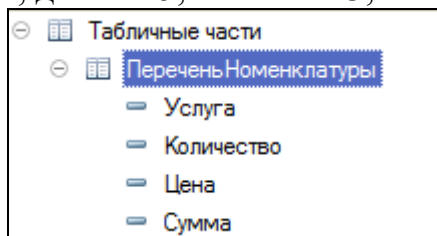
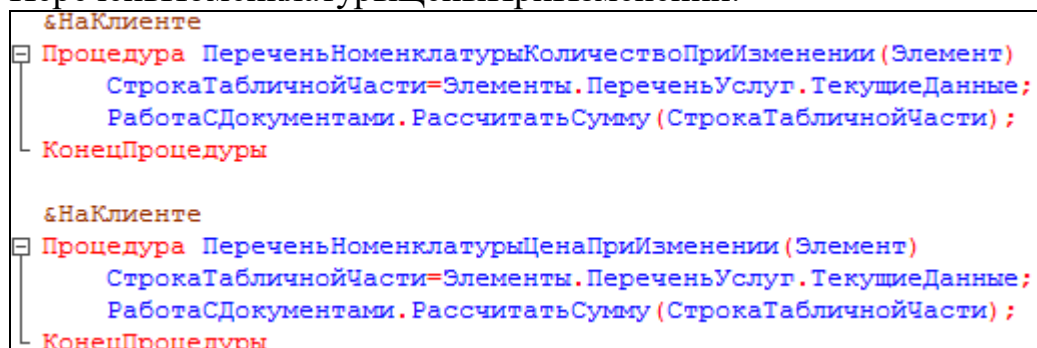


Рис. 69г. Реквизиты табличной части документа *ВыполнениеУслуги*

4. На закладке **Формы** создайте основную форму документа аналогично Форме АктПриемаВРемонт.

Для поля **ПереченьНоменклатурыКоличество** создадим обработчик события **ПриИзменении**, в котором будем вызывать процедуру **РассчитатьСумму** из общего модуля **РаботаСДокументами**.

Необходимо создать два обработчика событий **ПереченьНоменклатурыКоличествоПриИзменении** и **ПереченьНоменклатурыЦенаПриИзменении**.



5. Также отредактируем **Командный интерфейс**, чтобы подсистеме Оказание услуг была доступна команда создания новых документов.

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие.

ВВОД ДАННЫХ В БД

1. Внесите не менее 10 документов Оказание услуги для того, чтобы проверить работу приложения после изменений.

Рис. 69д. Форма ввода данных по документу Оказание услуги

VIII. ПЕРИОДИЧЕСКИЙ РЕГИСТР СВЕДЕНИЙ

В документе **Оказание услуги** каждый раз приходится вводить цену услуги вручную. Но очевидно, что существует перечень услуг, который определяет стоимость (цену) каждой услуги. Т.к. стоимость услуги является неотъемлемым свойством самой услуги.

В тоже время стоимость услуг имеет особенность меняться со временем. И может сложиться такая ситуация, когда потребуется внести изменения или уточнения в один из ранее проведенных документов Оказание услуги. В этом случае нельзя будет получить правильную стоимость услуги, т.к. в реквизите справочника будет храниться последнее введенное значение.

В дальнейшем может возникнуть потребность анализа изменений стоимости услуг во времени.

Для этого воспользуемся Регистром сведений.

Регистр сведений предназначен для описания структуры хранения данных в разрезе нескольких измерений. На основе регистра сведений платформа создает в БД таблицу, в которой может храниться произвольная информация, «привязанная» к набору измерений.

Каждое движение регистра сведений устанавливает новое значение ресурса, поэтому он может хранить любые данные (не только числовые).

При необходимости регистр сведений хранит данные с привязкой ко времени: и актуальную информацию, и историю изменений во времени. Такие регистры называют **Периодический регистр сведений**.

Создадим регистр сведений, который будет хранить развернутую во времени информацию по стоимости услуг, оказываемых предприятием.

1. Добавить новый **Регистр сведений** (рис. 70а).

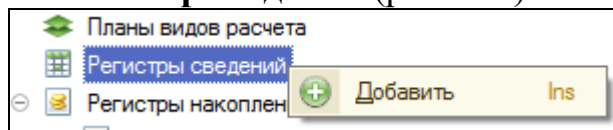


Рис. 70а. Выбор объекта конфигурации Регистр сведений

2. Присвоим имя Регистру сведений – **Цены**. Установим периодичность **В пределах секунды** (т.к. необходимо отслеживать цены несколько раз в течении дня). Также установим Представление записи как **Цена**, а Представление списка как **Цены на номенклатуру** (рис. 70б).

Рис. 70б. Основные свойства регистра сведений Цены

Обратите внимание на свойство **Режим записи**. По умолчанию оно имеет значение – **Независимый**, т.е. в созданный независимый регистр сведений в дальнейшем можно будет вводить данные без использования регистратора», вручную».

3. На закладке **Подсистемы** по логике конфигурации укажем, что регистр будет доступен в разделах **Бухгалтерия**, **Оказание услуг**.

4. На закладке **Данные** создадим измерение **НоменклатураУслуг** с типом **СправочникСсылка.НоменклатураУслуг**. Укажем, что это измерение будет **Ведущим** (рис. 70в).

Свойство **Ведущее** имеет смысл использовать только тогда, когда измерение имеет тип ссылки на объект базы данных. В этом случае запись регистра сведений представляет интерес, пока существует тот объект, ссылка на который выбрана в качестве значения этого измерения в этой записи. При удалении объекта, все записи регистра сведений по этому объекту будут автоматически удалены. Кроме этого в форме элемента справочника **Номенклатура услуг**, в панели навигации в группе **Перейти**, появится ссылка. По ней возможен переход к записям этого регистра, которые содержат в измерении **НоменклатураУслуг** ссылку на этот элемент справочника.

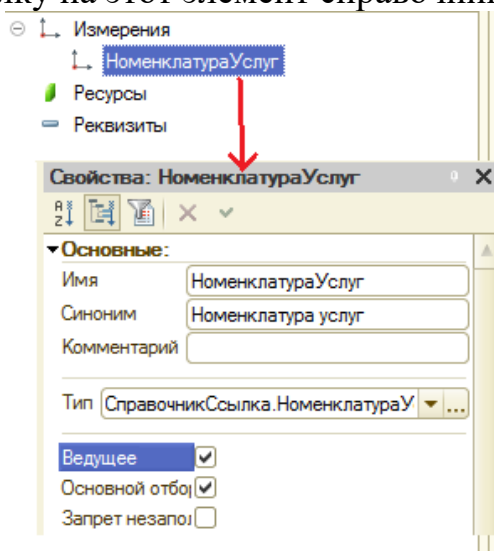


Рис. 70в. Создание ведущего измерения регистра сведений

4. Далее создадим ресурс **Цена**, тип **Число**, длина 15, точность 2, неотрицательное (рис. 70г).

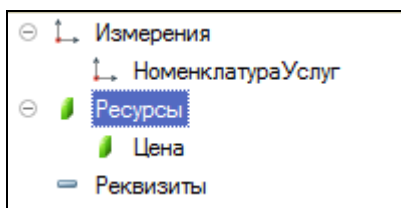


Рис. 70г. Ресурс регистра сведений

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие.

ВВОД ДАННЫХ В БД

1. Задайте несколько вариантов цен от 2-х до 5-ти по каждой услуги из справочника на разные даты при помощи формы (рис. 70д).

Рис. 70г. Форма для ввода цен на услуги в регистре сведений

IX. АВТОМАТИЧЕСКАЯ ПОДСТАНОВКА ЦЕНЫ В ДОКУМЕНТ ПРИ ВЫБОРЕ НОМЕНКЛАТУРЫ

Решим следующую задачу. Необходимо, чтобы при создании или изменении документа ОказаниеУслуги и при добавлении в табличную часть какой-нибудь номенклатуры услуг, одновременно подставлялась бы сразу актуальная цена, полученная из регистра сведений и соответствующая дате документа.

Для этого необходимо:

1. написать функцию **РозничнаяЦена**, которая будет возвращать актуальную цену;
2. вызвать функцию, когда в документ будет добавляться новая номенклатура.

Поскольку такая функция может понадобиться в дальнейшем, напомним ее в «общедоступном» месте – в общем модуле.

1. Создать функцию **РозничнаяЦена** в ветке Общие – Общие модули с названием **РаботаСоСправочниками**. Установить флаг **Вызов сервера** (чтобы экспортные процедуры и функции этого модуля можно было бы вызвать с клиента) (рис. 71а).

Рис. 71а. Свойства общего модуля

2. Поместите в модуль следующий текст.

```
Функция РозничнаяЦена(АктуальнаяДата,ЭлементНоменклатурыУслуг)Экспорт
//Создать вспомогательный объект Отбор
Отбор = Новый Структура("НоменклатураУслуг",ЭлементНоменклатурыУслуг);

//Получить актуальные значения ресурсов регистра
ЗначенияРесурсов=РегистрыСведений.Цены.ПолучитьПоследнее(АктуальнаяДата,Отб
ор);
Возврат ЗначенияРесурсов.Цена;
КонецФункции
```

```
Функция РозничнаяЦена (АктуальнаяДата, ЭлементНоменклатурыУслуг) Экспорт
//Создать вспомогательный объект Отбор
Отбор = Новый Структура ("НоменклатураУслуг", ЭлементНоменклатурыУслуг) ;

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

КонецФункции
```

Поясним функцию.

Для получения розничной цены необходимо в функцию передать два параметра:

- **АктуальнаяДата** – параметр типа Дата, определяет точку на оси времени, в которой нас интересует значение розничной цены.
- **ЭлементНоменклатурыУслуг** – ссылка на элемент справочника НоменклатураУслуг, для которого необходимо получить розничную цену.

Эта структура содержит **Отбор** по измерениям регистра. С его помощью определяется искомое измерение регистра НоменклатураУслуг, которое равно переданной в функцию ссылке на элемент справочника.

Имя ключа структуры («НоменклатураУслуг») должна совпадать с именем измерения регистра, заданного в конфигураторе, а значение элемента структуры (**ЭлементНоменклатурыУслуг**) задает отбираемое по данному измерению значение.

Во второй строке идет обращение к менеджеру регистра сведений Цены (**РегистрыСведений.Цены**) и выполнение метода **ВыполнитьПоследнее()**, который возвращает значение ресурсов самой последней записи регистра, соответствующей переданной в функцию дате (**АктуальнаяДата**) и значениям измерений регистра **Отбор**.

Метод **ПолучитьПоследнее** возвращает структуру, содержащую значения ресурсов, которая сохраняется в переменной **ЗначенияРесурсов**.

В следующей строке получаем искомую розничную цену, указав имя нужного ресурса регистра через точку (**ЗначенияРесурсов.Цена**) и возвращаем ее при выполнении функции.

Т.о. данную функцию можно вызывать в некоторый момент времени.

3. Далее решим задачу подстановки актуальной цены в документ.

Необходимо открыть **Форму документа ВыполнениеУслуги** (она была создана ранее). В **Модуль формы** необходимо дописать **Процедуру ПриИзменении**.

Для этого дважды щелкните на элемент формы **ПереченьНоменклатурыУслуга** (рис. 716).

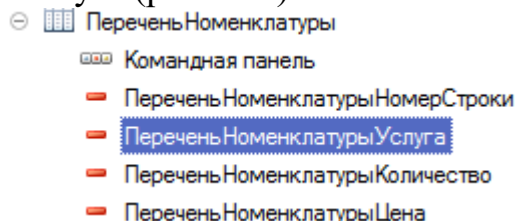


Рис. 716. Элементы документа *Выполнение услуги*

В окне свойств в событии **ПриИзменении** создайте обработчик события, в который внесем следующий текст:

```
&НаКлиенте
Процедура ПереченьНоменклатурыУслугаПриИзменении(Элемент)
    //Получить следующую строку табличной части
    СтрокаТабличнойЧасти=Элементы.ПереченьНоменклатуры.ТекущиеДанные;

    //Установить цену
    СтрокаТабличнойЧасти.Цена=РаботаСоСправочниками.РозничнаяЦена(Объект.Дата,С
    трокаТабличнойЧасти.Услуга);

    //Пересчитать сумму
    РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
КонецПроцедуры
```

```
&НаКлиенте
Процедура ПереченьНоменклатурыУслугаПриИзменении (Элемент)
    //Получить следующую строку табличной части
    СтрокаТабличнойЧасти=Элементы.ПереченьНоменклатуры.ТекущиеДанные;

    //Установить цену
    СтрокаТабличнойЧасти.Цена=РаботаСоСправочниками.РозничнаяЦена(Объект.Дата,СтрокаТабличнойЧасти.Услуга);

    //Пересчитать сумму
    РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
КонецПроцедуры
```

Поясним функцию.

Первая строка уже известна: сначала получаем текущую строку табличной части документа, т.к. она понадобится в дальнейшем, и сохраним ее в переменной **СтрокаТабличнойЧасти**.

Затем вызываем функцию **РозничнаяЦена()** из общего модуля **РаботаСоСправочниками**.

Первым параметром в функцию передаем дату документа, на которую необходимо получить цену. Дату документа получаем из основного реквизита формы – **Объект.Дата**.

Вторым параметром передаем ссылку на элемент справочника **НоменклатураУслуг**, который содержится в текущей строке табличной части документа (**СтрокаТабличной.Услуга**).

Функция возвращает последнее значение цены, и это значение присваивается полю Цена в текущей строке табличной части документа (**СтрокаТабличнойЧасти.Цена**).

Здесь вызывается процедура **РассчитатьСумму** из общего модуля **РаботаСДокументами**.

Обратите внимание, что процедура **ПереченьНоменклатурыУслугаПриИзменении()** начинает работать в модуле формы на стороне клиента, т.к. это обработчик интерактивного события формы. Создавая заготовку этой процедуры, платформа автоматически разместит перед описанием процедуры директиву компиляции **&НаКлиенте**.

Затем вызываем функцию **РозничнаяЦена()**. Т.к. эта функция не будет найдена на стороне клиента, то исполнение будет передано в общий модуль **РаботаСоСправочниками**, который выполняется на сервере. После завершения функции программный код продолжит исполняться на клиенте.

Проверим в работе выполненные выше действия, запустим приложение в режиме 1С: Предприятие. Поработаем с разработанной конфигурацией.

ВВОД ДАННЫХ В БД

1. Введите в приложение не менее 8-ми документов Оказание услуги.

Х. ПЕРЕЧИСЛЕНИЯ

Объект конфигурации **Перечисления** предназначен для описания структуры хранения постоянных наборов значений, не изменяемых в процессе работы конфигурации. На основе Перечислений платформа в БД создает таблицу, в которой можно хранить набор некоторых постоянных значений.

Набор возможных значений перечислений задается при конфигурировании системы, и пользователь не может изменять, удалять или добавлять их (например, «включая НДС», «без НДС»).

Важная особенность перечислений – значения перечисления не «обезличены» для конфигурации, на них могут описаться алгоритмы работы программы.

1. Создайте новый объект конфигурации Перечисления с именем **ВидыНоменклатурыУслуг** (рис. 72а).



Рис. 72а. Выбор объекта конфигурации Перечисления

2. На закладке **Данные** добавить два значения перечисления (рис. 72б):

1. **УслугаВЦеху**,
2. **УслугаВОфисе**.

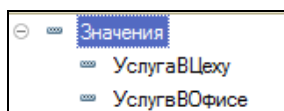


Рис. 72б. Данные перечисления Виды номенклатуры

3. Далее привяжем номенклатуру услуг к значениям перечисления.

В режиме Конфигуратор создадим новый реквизит справочника **НоменклатураУслуг** – **ВидыНоменклатурыУслуг** с типом **ПеречислениеСсылка.ВидыНоменклатурыУслуг** (рис. 72в).

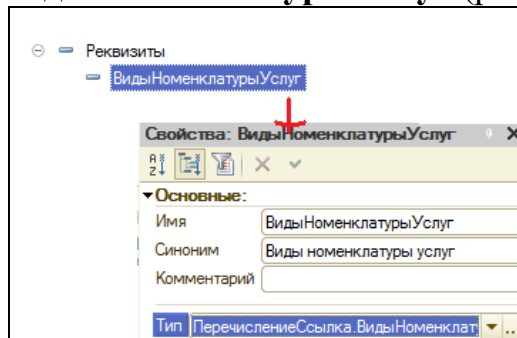


Рис. 72в. Данные справочника НоменклатураУслуг

Обновим конфигурацию БД и запустим ее в режиме 1С: Предприятие.

Для каждого элемента справочника **Номенклатура** услуг задайте соответствующее значение реквизита Вид номенклатуры.

4. Проверим в работе выполненные выше действия, запустим приложение в режиме 1С: Предприятие.

XI. ОБОРОТНЫЕ РЕГИСТРЫ НАКОПЛЕНИЯ

Создадим еще один регистр накопления, который бы учитывал сумму ,оказанной услуги, количество услуг ,которые были оказаны за определенный промежуток времени. Кроме этого интересно было бы посмотреть рейтинг услуг, какому клиенту оказывались услуги (например ,чтобы представить ему скидку) ,какой мастер предоставлял услуги (чтобы начислить ему зарплату).

Для реализации этих задач создадим «хранилище» данных – оборотный регистр накопления.

Оборотный регистр накопления очень похож на знакомый регистр остатков, но для него понятие «остаток» не имеет смысла. Оборотный регистр накапливает только обороты, остатки ему «безразличны». Поэтому единственной виртуальной таблицей, которую будет создавать система для такого регистра, будет таблица оборотов. В остальном оборотный регистр ни чем не отличается от регистра остатков.

Следует сказать об одной особенности конструирования регистров накопления, напрямую связанной с возможностью получения остатков.

При создании оборотного регистра накопления нет особой сложности в определении того, какие именно параметры должны являться измерениями регистра – можно назначить в качестве его измерений любые нужные параметры.

Совсем иная ситуация в случае регистра накопления поддерживающего накопление остатков. Для него выбор измерений должен выполняться исходя из того, что движения регистра могут быть осуществлены «в две стороны»: приход

и расход. Таким образом, в качестве измерений нужно выбирать те параметры, по которым движения точно будут осуществляться как в одну, так и в другую сторону.

Для разработки оборотного регистра выполним следующие действия.

1. Создать новый регистр накопления **Выполненные услуги** и определить вид регистра – **Обороты** (рис. 73а).

Рис. 73а. Создание оборотного регистра накопления

2. На закладке **Данные** создадим *измерения* регистра:

- Услуга, тип СправочникСсылка.НоменклатураУслуг;
- Клиент, тип СправочникСсылка.Клиент;
- Мастер, тип СправочникСсылка.Подразделения.

У регистра будет *два ресурса*:

- Количество, Тип данных – Число, Длина – 15, Флаг – неотрицательное, Точность – 0;
- Стоимость, Тип данных – Число, Длина – 15, Флаг – неотрицательное, Точность – 2.

После создания регистр **Выполненные услуги** должен выглядеть в дереве конфигурации так, как показано на рис. 73б.

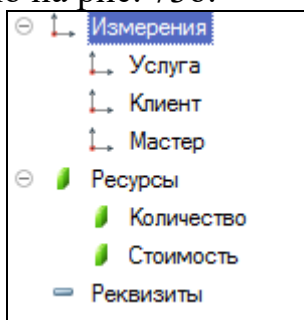


Рис. 73б. Оборотный регистр накопления Выполненные услуги – Измерения и ресурсы

3. Отредактируем командный интерфейс. На закладке **Подсистемы** отметим, что регистр будет отображаться во всех подсистемах: **Бухгалтерия, Оказание услуг, Расчет зарплаты.**

Откроем вначале подсистему **Бухгалтерия** (рис. 73в).

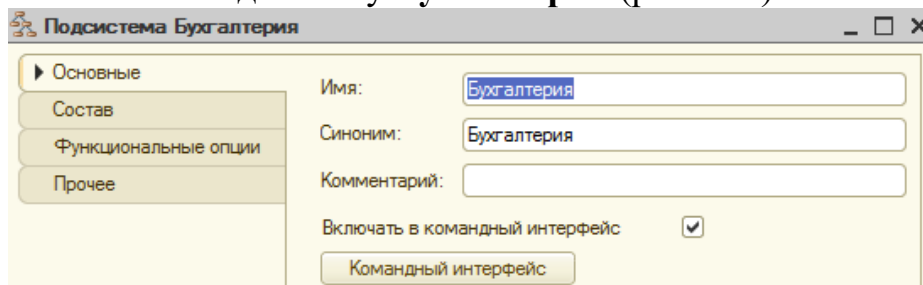
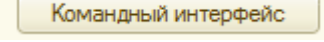


Рис. 73в. Подсистема Бухгалтерия

При нажатии на кнопку  отразятся команды выбранной подсистемы. В группе **Панель навигации.Обычное** включить видимость у команды **Выполненные услуги** и мышью перетащим ее в группу **Панель навигации.См.также** (рис. 73г).

Тоже самое выполните для остальных подсистем.

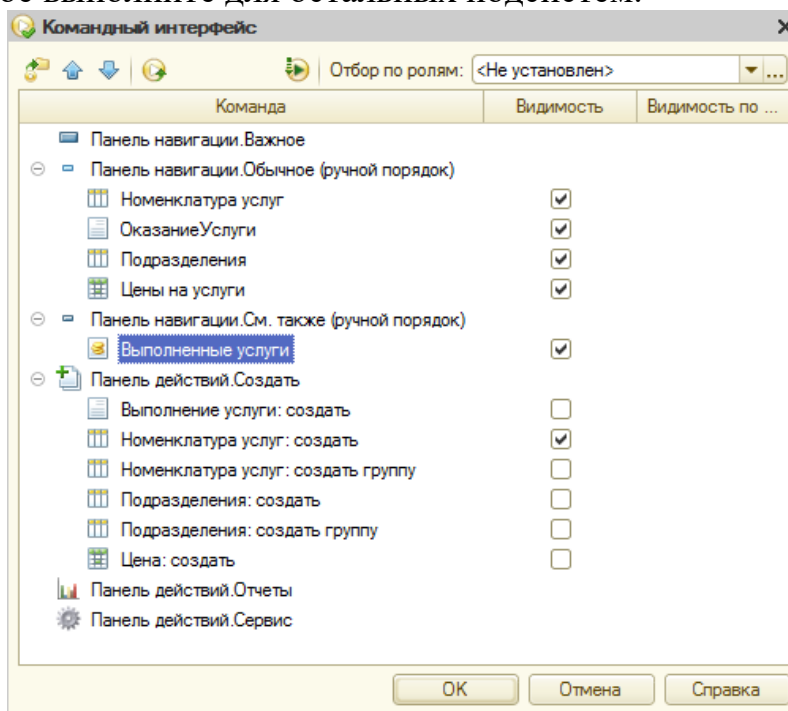


Рис. 73г. Командный интерфейс подсистемы Бухгалтерия

Далее выполним проведение документа по регистру.

Сначала изменим процедуру проведения документа **Выполнение услуги**.

4. Открыть окно редактирования объекта конфигурации документа **ОказаниеУслуги** и на закладке **Движения** укажем, что этот документ будет создавать движение по регистру **Выполненные услуги** (рис. 74а).

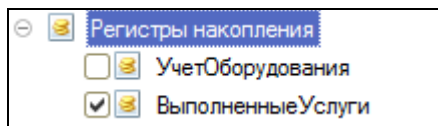



Рис. 74а. Создание движений документа *Выполнение услуги в регистре*
Выполненные услуги

5. Обратите внимание, что сразу после отметки выбранного регистра становится доступной кнопка  **Конструктор движений**. Нажмите на кнопку. На экране отразится **Конструктор движения регистров**.

В табличной части выберите **Перечень Номенклатуры**.

Список реквизитов документа, который уже заполнен реквизитами шапки документа, автоматически дополнится реквизитами табличной части.

Нажмите на кнопку **Заполнить выражения**.

В нижнем окне сформируется соответствии полей (измерений и ресурсов) регистра и выражений для их расчета (рис. 74б).

Конструктор движений установил необходимое соответствие регистра и документа.

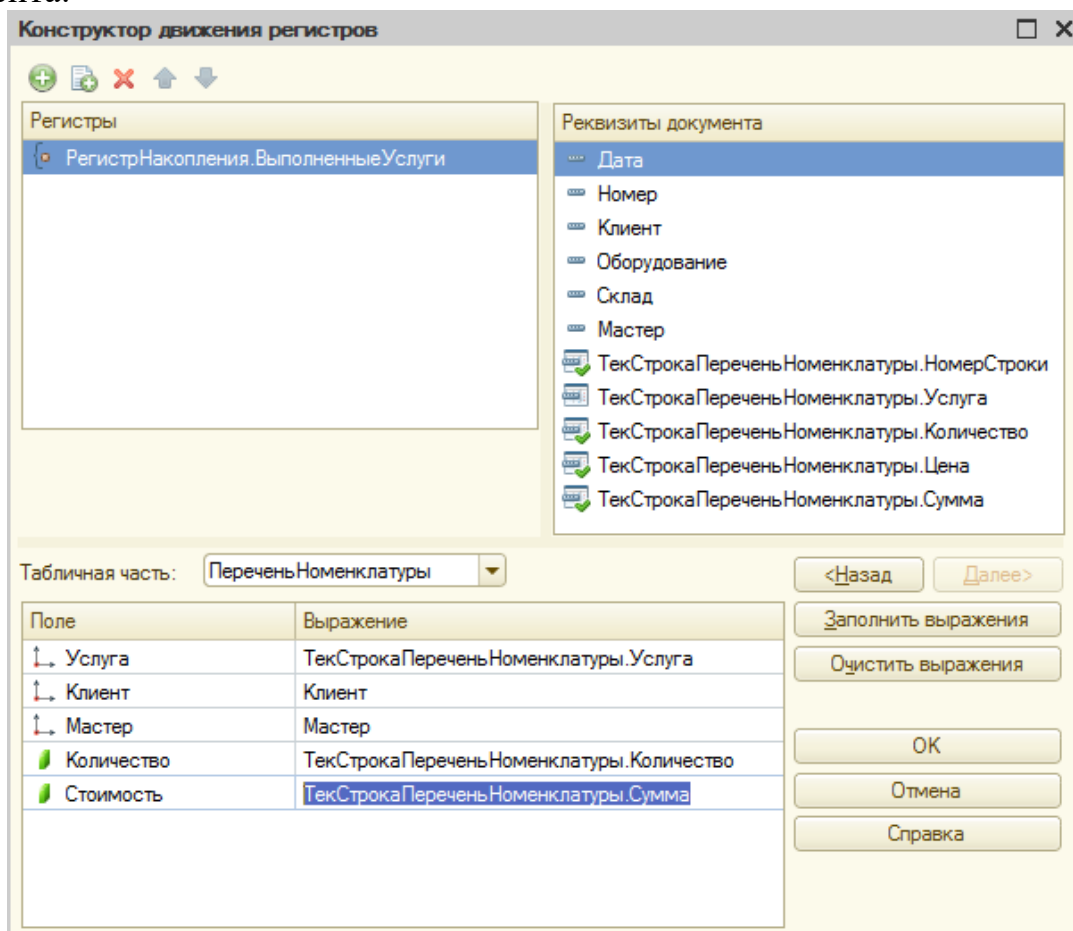


Рис. 74б. Выбор табличной части документа и заполнение выражений для
расчета движений регистра

Далее необходимо нажать на кнопку **ОК**.

6. На экране откроется текст модуля документа *Выполненные услуги* (рис. 74в).


```

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

  // регистр ВыполненныеУслуги
  Движения.ВыполненныеУслуги.Записывать = Истина;
  Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Движение = Движения.ВыполненныеУслуги.Добавить ();
    Движение.Период = Дата;
    Движение.Услуга = ТекСтрокаПереченьНоменклатуры.Услуга;
    Движение.Клиент = Клиент;
    Движение.Мастер = Мастер;
    Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
    Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Сумма;
  КонечЦикла;

  ///}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  КонечПроцедуры

```

Рис. 74в. Процедура документа Выполненные услуги ОбработкаПроведения

Проверим в работе выполненные выше действия, запустим приложение в режиме 1С: Предприятие.

ХII. Отчет – ВЫБОР ДАННЫХ ИЗ ДВУХ ТАБЛИЦ

Создадим отчет **Рейтинг услуг**, который будет содержать информацию о том, выполнение каких работ принесло наибольшую прибыль предприятию.

На примере отчета **Рейтинг услуг** проиллюстрируем, как отбирать данные в некотором периоде, как задавать параметры запроса, как использовать в запросе данные из нескольких таблиц и как включать в результат запроса все данные одного из источников, как работать с параметрами системы компоновки данных, как использовать стандартные даты, и познакомимся с быстрыми пользовательскими настройками отчетов, как более детально настраивать отбор и условное оформление в отчетах.

1. В режиме Конфигуратор добавим новый объект конфигурации **Отчет**. Назовем его **РейтингУслуг** и запустим конструктор **схемы компоновки данных** (рис. 75а).

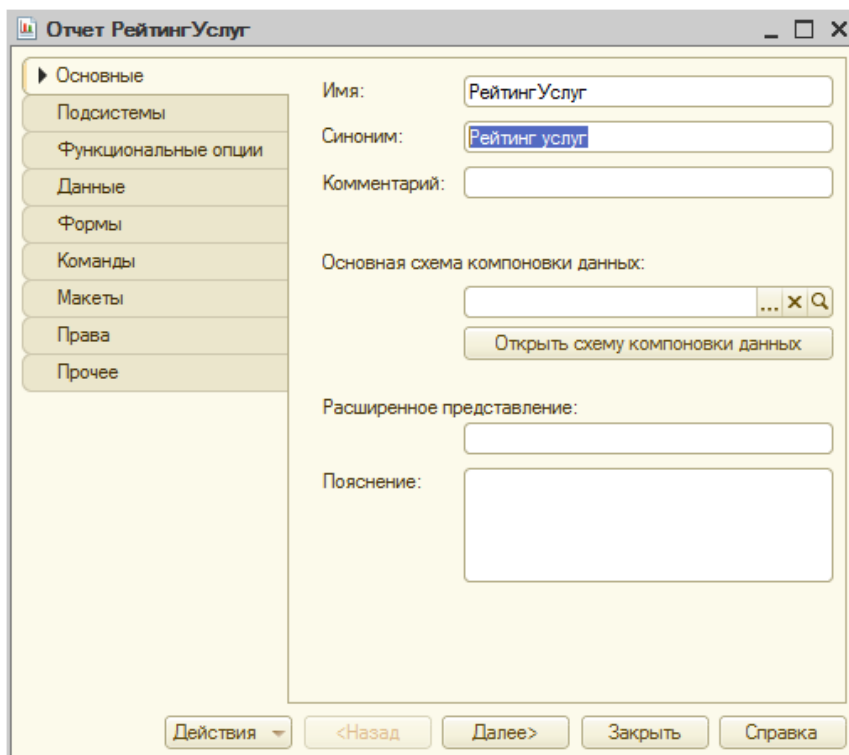


Рис. 75а. Основные свойства отчета

2. В схеме компоновки данных **Добавить набор данных – запрос** (рис. 75б) и нажать на кнопку **Конструктор запроса...**.

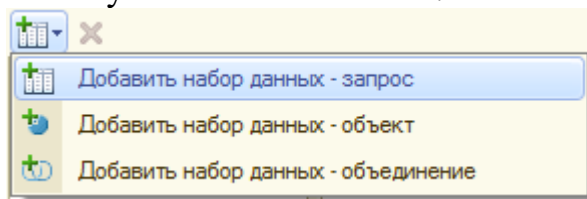


Рис. 75б. Добавление набора данных в Схему компоновки данных

3. В качестве источника данных для запроса выберем объектную (ссылочную) таблицу **НоменклатураУслуг** и виртуальную таблицу регистра накопления **ВыполненныеУслуги.Обороты** (рис. 75в).

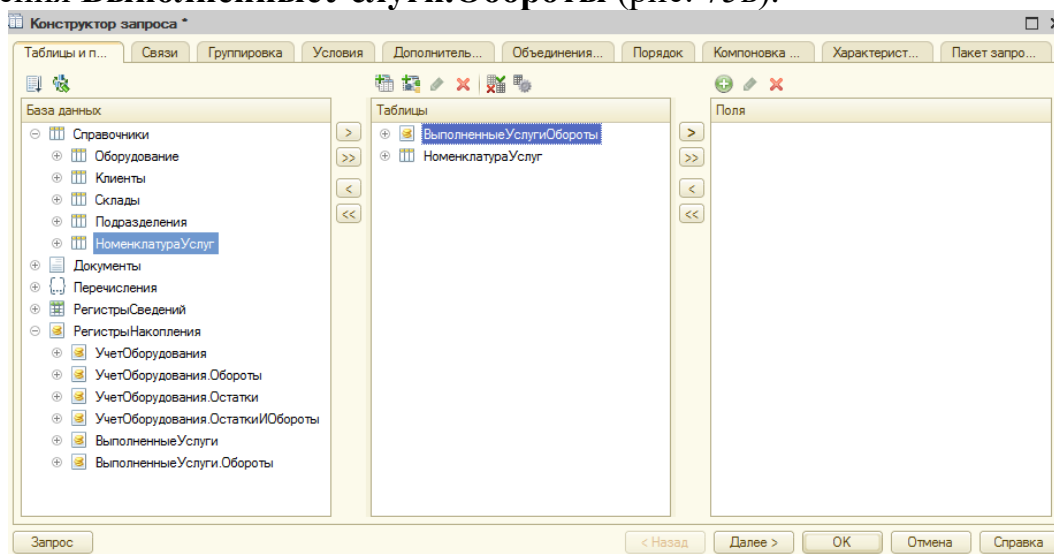


Рис. 75в. Конструктор запроса – Выбор таблиц

4. Чтобы исключить неоднозначность имен в запросе, переименуем таблицу НоменклатураУслуг в спрНоменклатураУслуг. Для этого выделим ее в списке Таблицы, вызовем ее контекстное меню и выберем пункт Переименовать таблицу (рис. 75г).

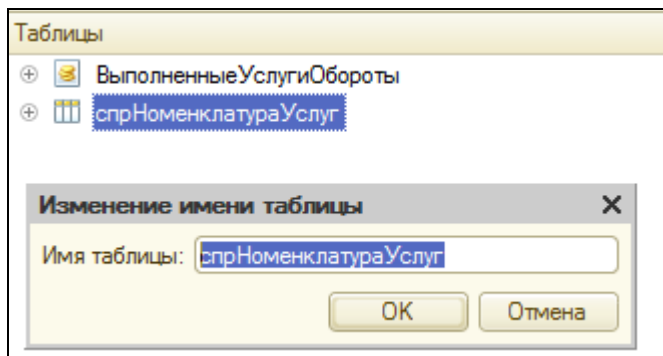


Рис. 75г. Переименование таблицы в запросе

5. В список полей перенесем поля СпрНоменклатураУслуга.Ссылка и ВыполненныеУслугиОбороты.СтоимостьОборот из этих таблиц (рис. 75д).

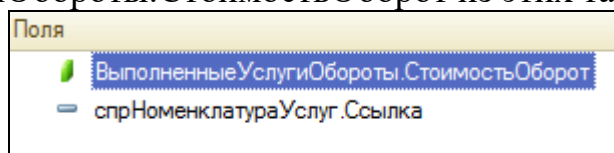


Рис. 75д. выбранные поля

6. Перейдем на закладку **Связи**. Так как в запросе теперь участвуют несколько таблиц, требуется определить связь между ними.

По умолчанию платформой уже будет создана связь по полю **НоменклатураУслуг**. То есть значение измерения **НоменклатураУслуг** регистра **ВыполненныеУслуги** должно быть равно ссылке на элемент справочника **НоменклатураУслуг**.

Необходимо снять флажок **Все** у таблицы **ВыполненныеУслуги.Обороты** и установить его у таблицы **спрНоменклатураУслуг** (рис. 75е).

№	Таблица 1	В...	Таблица 2	В...
1	ВыполненныеУслуги...	<input type="checkbox"/>	спрНоменклатураУслуг	<input checked="" type="checkbox"/>

Рис. 75е. Установление связи между таблицами

Тем самым задаем тип связи как **Левое соединение**, то есть в результат запроса будут включены все записи справочника **НоменклатураУслуги** те записи регистра **ВыполненныеУслуги**, которые удовлетворяют условию связи по полю **НоменклатураУслуг**. Например (рис. 75ж):

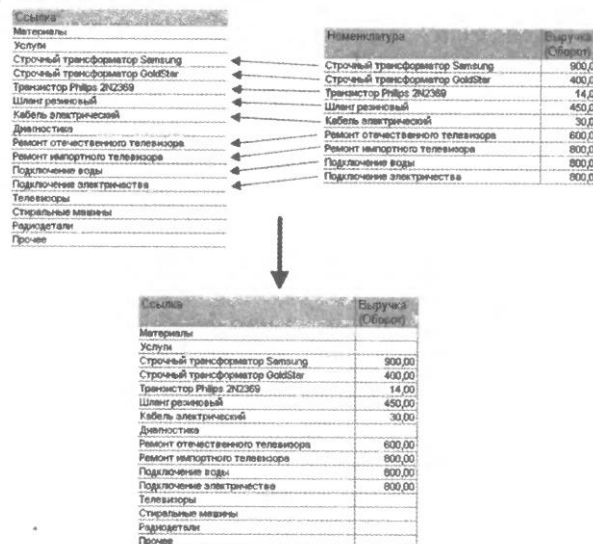


Рис. 75ж. связь таблиц в запросе

7. Определим **Условие отбора записей**. На закладке Условия установим отбор, чтобы группы справочника НоменклатураУслуги не попадали в отчет.

Для этого раскроем таблицу **спрНоменклатураУслуги**, перетащим мышью поле **ЭтоГруппа** в список условий, установим флажок Произвольное и напомним в поле **Условие** следующий текст (рис. 75з):

спрНоменклатура.ЭтоГруппа=ЛОЖЬ

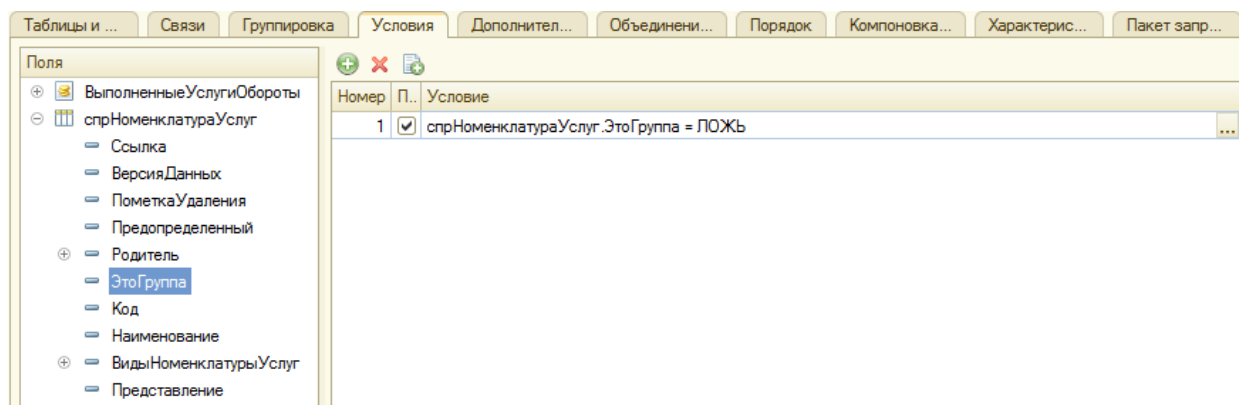


Рис. 75з. Условие запроса

Тем самым было указано, что из базы данных нужно выбрать только те записи справочника НоменклатураУслуг, которые не являются группами.

Вторым условием должно быть то, что выбранный элемент является услугой. Это – **Простое условие**. Чтобы его создать, перетащим мышью поле **ВидНоменклатурыУслуг** в список условий.

Платформа автоматически сформирует условие, согласно которому вид номенклатуры услуг должен быть равен значению параметра **ВидНоменклатурыУслуг** (рис. 75и).

В дальнейшем перед выполнением запроса передадим в параметр ВидНоменклатурыУслуг значение перечисления – **Услуга**.

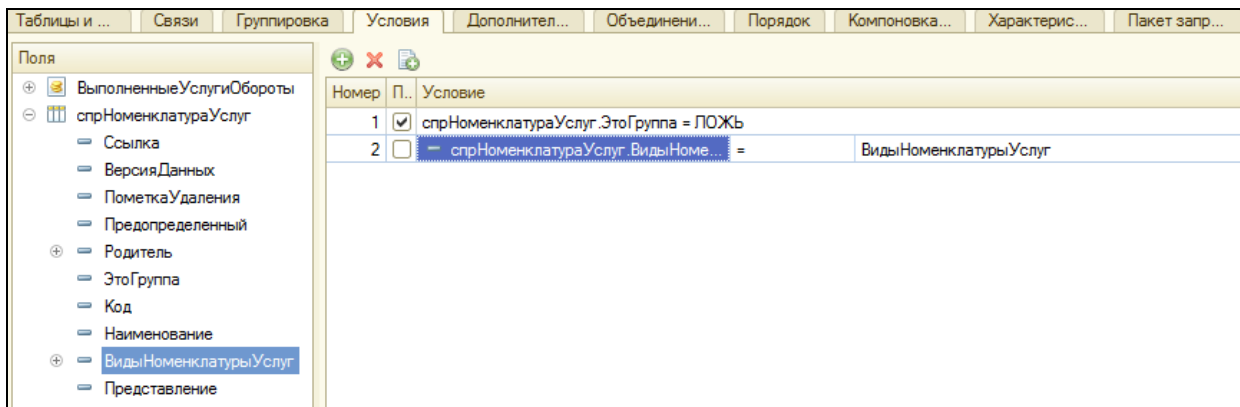


Рис. 75и Создание условия запроса

8. Обозначим **Псевдонимы полей**. На закладке **Объединения/Псевдонимы** укажем, что представление элемента справочника (поле Ссылка) будет иметь псевдоним **Услуга**, а поле регистра будет иметь псевдоним **Стоимость** (рис. 75к).

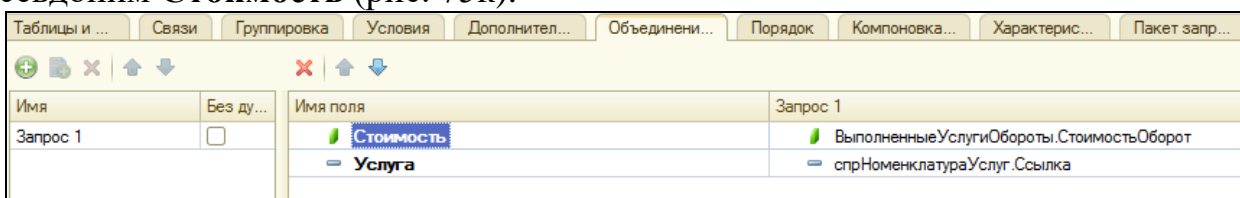


Рис. 75к. Установка псевдонимов полей запроса

8. Далее определим **Порядок записей**. На закладке **Порядок** укажем, что результат запроса должен быть отсортирован по убыванию значения поля **Сумма** (рис. 75л).

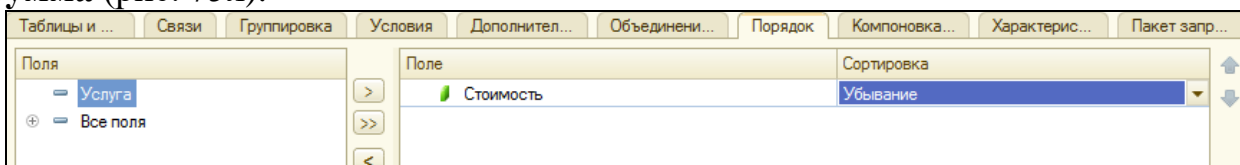
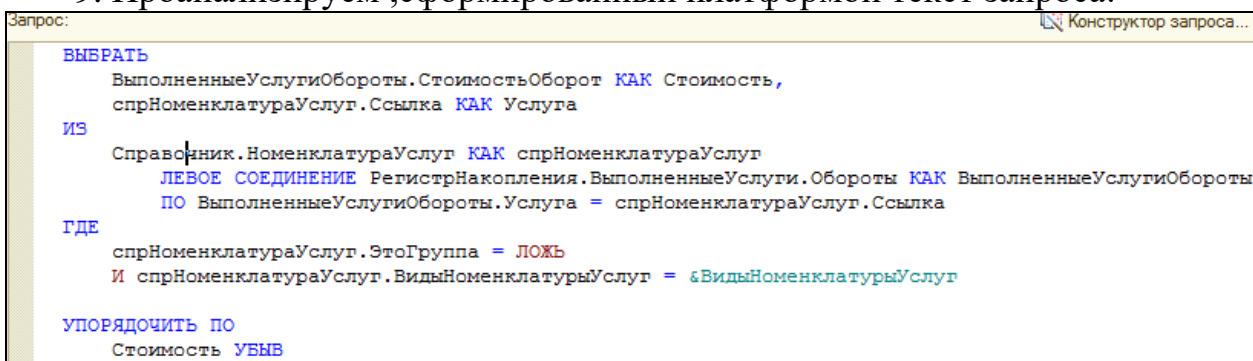


Рис. 75л. Порядок записей запроса

Создание запроса закончено, нажмем кнопку **ОК**. Вернемся в конструктор схемы компоновки данных.

9. Проанализируем ,сформированный платформой текст запроса:



Сначала, как обычно, идет часть описания запроса, и в ней есть новые для конструкции.

При описании источников запроса (после ключевого слова **ИЗ**) использована возможность определения нескольких источников запроса.

В данном случае выбираются записи из двух источников: **СпрНоменклатураУслуга** и **ВыполненныеУслуги**, причем ключевым предложением **ЛЕВОЕ СОЕДИНЕНИЕ ... ПО** описан способ, которым будут скомбинированы между собой записи этих двух источников.

ЛЕВОЕ СОЕДИНЕНИЕ означает, что в результат запроса нужно включить комбинации записей из обоих источников, которые соответствуют указанному после ключевого слова **ПО** условию. Кроме этого, в результат запроса нужно включить еще и записи из первого (указанного слева от слова **СОЕДИНЕНИЕ**) источника, для которых не найдено соответствующих условию записей из второго источника.


В части описания запроса есть еще одна новая конструкция – задание условий отбора данных из исходных таблиц.

Условию отбора всегда предшествует ключевое слово **ГДЕ**. После него описывается само условие. Обратите внимание, что поля исходных таблиц, на которые накладывается условие, могут и не входить в список выборки (как в нашем случае). Кроме того, в нашем условии использован параметр запроса **ВидНоменклатурыУслуг** (перед именем параметра указывается символ **&** – амперсанд).

Вернемся в конструктор схемы компоновки данных.

Предполагается, что в отчете будут представлены итоговые значения полученной предприятием суммы (в руб.) для каждой услуги. Для этого необходимо определить поля ресурсов отчета.

Под **ресурсами** в системе компоновки данных подразумеваются поля, значения которых рассчитываются на основании детальных записей, входящих в группировку. По сути, ресурсы являются групповыми или общими итогами отчета.

10. Итоговые данные формируются на закладке **Ресурсы**. Перейдем на эту закладку и нажмем кнопку , чтобы конструктор выбрал все доступные ресурсы, по которым можно вычислять итоги.

В нашем случае это единственный ресурс **Сумма**.

Платформа автоматически предложит рассчитывать сумму значений этого поля, что и необходимо (рис. 76а).

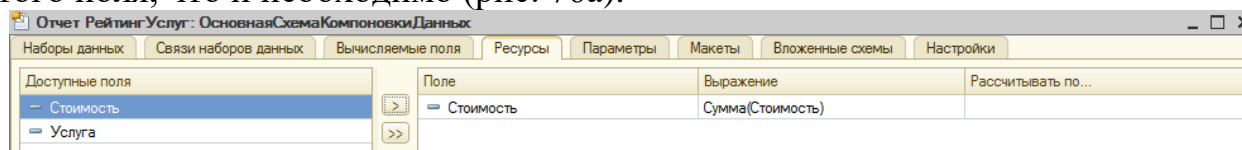


Рис. 76а. Ресурсы схемы компоновки данных

11. Установим **Параметры**. Пользователя, как правило, интересуют данные о хозяйственной деятельности за определенный период. Поэтому

практически в любом отчете используются параметры, задающие начало и конец отчетного периода.

Параметры отчета задают условия отбора записей в отчет. В схеме компоновки данных параметры отчета задаются на закладке Параметры.

На этой закладке представлено три параметра: **НачалоПериода**, **КонецПериода** и **ВидНоменклатурыУслуг**.

Почему параметра три, хотя в запросе был задан всего один – **ВидНоменклатурыУслуг**?

Все дело в том, что система компоновки данных самостоятельно анализирует текст запроса и помимо тех параметров, которые указаны в нем в явном виде (ВидНоменклатурыУслуг), предоставляет возможность настроить также и параметры виртуальных таблиц, которые участвуют в запросе.

Таковыми параметрами являются **НачалоПериода** и **КонецПериода**. Это первые два параметра виртуальной таблицы **РегистрНакопления.ВыполненныеУслуги.Обороты**, которая была использована в запросе, в левом соединении.


Следует помнить, что если передается в качестве этих параметров дата (а в нашем случае так и будет), то дата содержит и время с точностью до секунды.

Допустим, заранее известно, что пользователя не будут интересовать результаты работы отчета в периодах, указанных с точностью до секунды. В этом случае следует учесть две особенности:

- пользователя нужно избавить от необходимости указывать время при вводе даты периода, за который формируется отчет.
- по умолчанию время в дате установлено 00:00:00. Поэтому если пользователь задаст период отчета с 01.07.2009 по 14.07.2009, итоги регистра будут рассчитаны с начала дня 01.07.2009 00:00:00 по начало дня 14.07.2009 00:00:00.

12. Изменим существующее описание типа для параметра **НачалоПериода**.

Вернемся на закладку **Параметры схемы компоновки данных** и дважды щелкнем в ячейке **Тип**, соответствующей параметру **НачалоПериода**.

Затем нажмем кнопку выбора  и в нижней части окна редактирования типа данных установим **Состав даты** в значение **Дата**. Нажмем **ОК** (рис. 766).

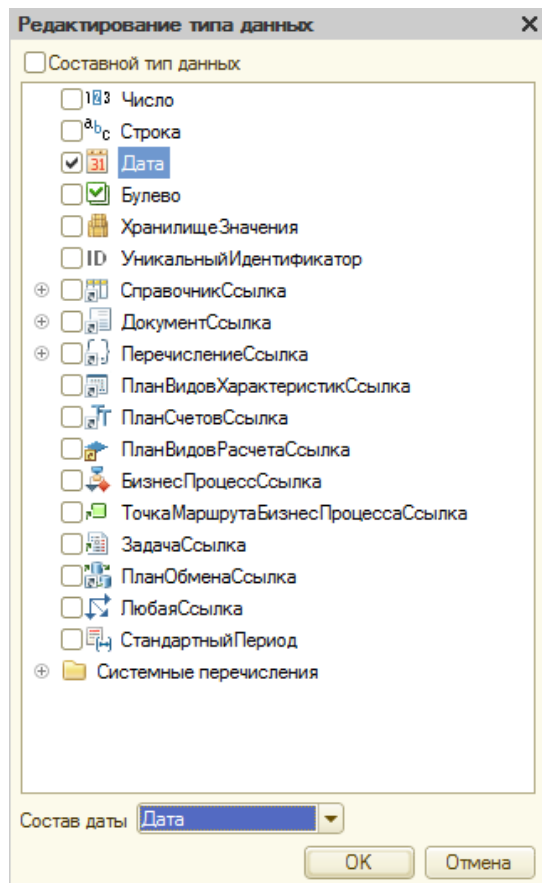


Рис. 76б. Редактирование состава даты

13. Еобавим еще один параметр, в который пользователь будет вводить дату окончания. А значение параметра **КонецПериода** будем рассчитывать автоматически таким образом, чтобы оно указывало на конец дня даты, введенной пользователем.

Поэтому для параметра **КонецПериода** установим флажок Ограничение доступности (рис. 76в).

Если этот флажок не установлен, то параметр будет доступен для настройки пользователем. Если же установить этот флажок, то пользователь не увидит этот параметр.

Имя	Заголовок	Тип	Доступные значе...	Д...	Значение	Выражение	Параметр функц...	В...	О...	Параметры реда...
НачалоПериода	Начало периода	Дата		<input type="checkbox"/>				<input checked="" type="checkbox"/>	<input type="checkbox"/>	
КонецПериода	Конец периода	Дата		<input type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
ВидыНоменклат...	Виды номенклат...	ПеречислениеСсылк...		<input type="checkbox"/>	Перечисление.Виды...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Рис. 76в. Ограничение доступности для параметра Конец периода

14. Далее с помощью кнопки **Добавить** в командной панели добавим новый параметр с именем **ДатаОкончания**. Для этого параметра платформа автоматически сформирует заголовок – **Дата окончания**. Оставим его без изменений. Тип значения параметра – **Дата**. Как и для параметра **Начало Периода**, укажем состав даты – **Дата**. А также для параметра **НачалоПериода** зададим заголовок, который будет отображаться пользователю, – **Дата начала**. По умолчанию добавленный нами параметр доступен для пользователя (ограничение доступности в колонке снято) (рис. 76г)

Отчет РейтингУслуг: ОсновнаяСхемаКомпоновкиДанных									
Наборы данных Связи наборов данных Вычисляемые поля Ресурсы Параметры Макеты Вложенные схемы Настройки									
Имя	Заголовок	Тип	Доступные значе...	Д...	Значение	Выражение	Параметр функц...	В...	О...
НачалоПериода	Дата начала	Дата		<input type="checkbox"/>				<input checked="" type="checkbox"/>	<input type="checkbox"/>
КонецПериода	Конец периода	Дата		<input type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ВидыНоменклат...	Виды номенклат...	ПеречислениеСсылк...		<input type="checkbox"/>	Перечисление.Виды...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ДатаОкончания	Дата окончания	Дата		<input type="checkbox"/>				<input checked="" type="checkbox"/>	<input type="checkbox"/>

Рис. 76г. Добавления параметра Дота окончания

14. Для параметра **КонецПериода** установили флажок **Ограничение доступности**, поскольку значение этого параметра собираемся вычислять на основании значения, установленного пользователем для параметра **ДатаОкончания**.

Чтобы задать формулу, по которой будет вычисляться значение параметра **КонецПериода**, воспользуемся языком выражений системы компоновки данных.

В нем есть функция **КонецПериода()**, которая позволяет получить дату, соответствующую концу какого-либо периода, например, указанного дня.

В ячейке **Выражение** зададим для параметра **КонецПериода** следующее выражение (рис. 76д):


КонецПериода(&ДатаОкончания,"День")

Отчет РейтингУслуг: ОсновнаяСхемаКомпоновкиДанных									
Наборы данных Связи наборов данных Вычисляемые поля Ресурсы Параметры Макеты Вложенные схемы Настройки									
Имя	Заголовок	Тип	Досту...	Д...	Значение	Выражение	Параметр...	В...	О...
НачалоПериода	Дата начала	Дата		<input type="checkbox"/>				<input checked="" type="checkbox"/>	<input type="checkbox"/>
КонецПериода	Конец периода	Дата		<input type="checkbox"/>		КонецПериода(&ДатаОкончания,"Дата")		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ВидыНоменклат...	Виды номенклат...	ПеречислениеСсылк...		<input type="checkbox"/>	Перечисление.ВидыНо...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ДатаОкончания	Дата окончания	Дата		<input type="checkbox"/>				<input checked="" type="checkbox"/>	<input type="checkbox"/>

Рис. 76д. Параметры системы компоновки

15. Настроим параметр **ВидНоменклатурыУслуг**. Поскольку отчет должен отображать суммы, на которую были оказаны услуги, значение параметра ВидНоменклатурыУслуг пользователь изменять не должен. Оно должно быть задано непосредственно в схеме компоновки как, например, **Перечисление.ВидыНоменклатурыУслуг.УслугаВЦеху** (таким образом будут проанализированные услуги, оказываемые в цехах).

Флажок ограничения использования у параметра **ВидНоменклатурыУслуг** платформа установила по умолчанию, поэтому необходимо указать только нужное значение перечисления **ВидыНоменклатурыУслуг** в ячейке **Значение**, соответствующей параметру **ВидНоменклатурыУслуг**.

Воспользуемся кнопкой выбора  и выберем это значение из списка перечисления видов номенклатуры (рис. 76е).

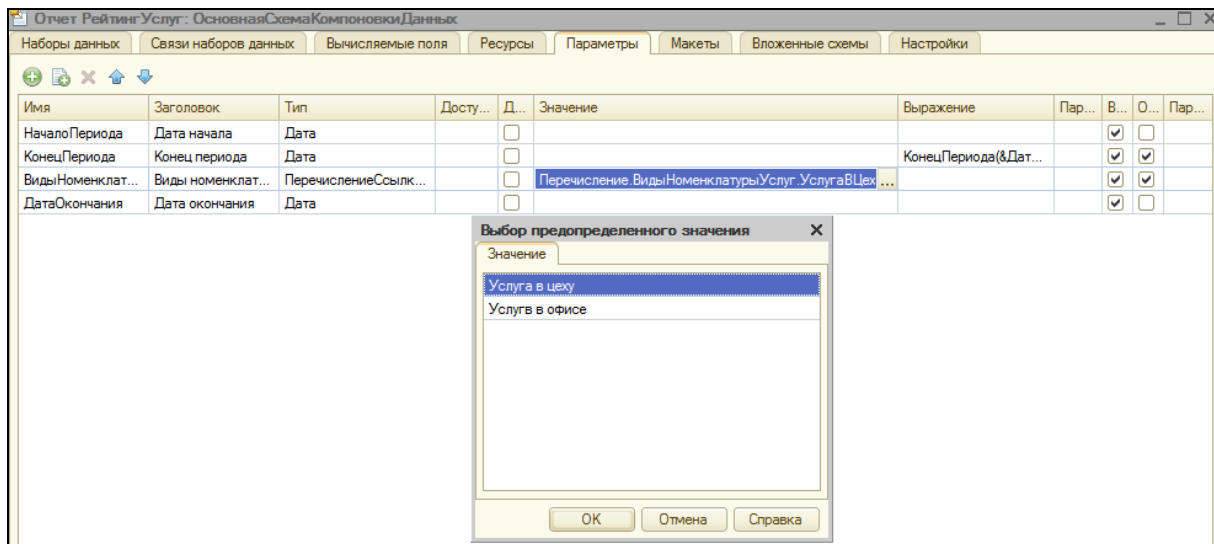


Рис. 76е. Установка значения параметра «ВидНоменклатурыУслуг»

16. Перейдем к формированию структуры отчета. На закладке **Настройки** добавим группировку и снова не укажем поле группировки (рис. 76ж).

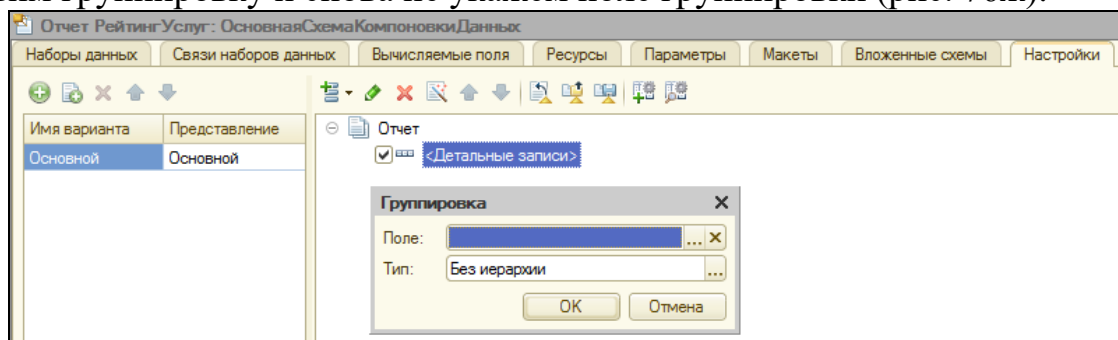


Рис. 76ж. Добавление группировки в отчет

На закладке **Выбранные поля** укажем поля Услуга и Стоимость (рис. 76з).

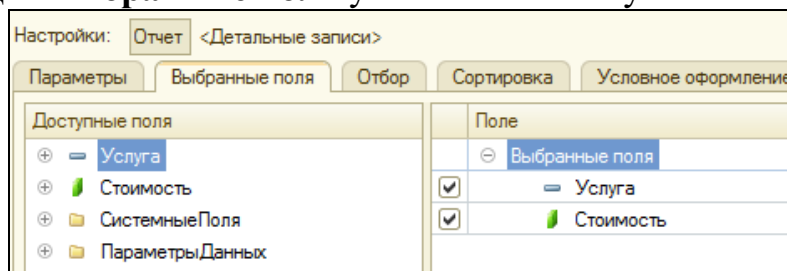


Рис. 76з. Структура отчета Рейтинг услуг

Затем перейдем на закладку **Другие настройки** и зададим заголовок отчета – Рейтинг услуг (рис. 76и).

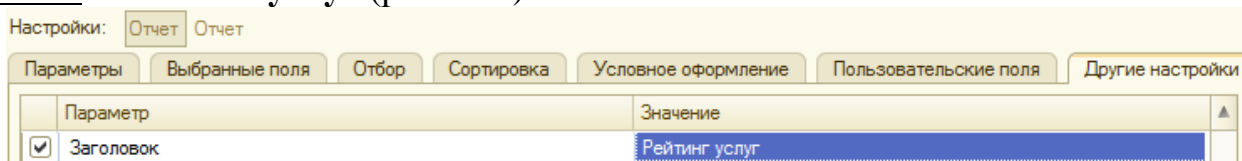


Рис. 76и. Установка заголовка отчета

17. Зададим Быстрые пользовательские настройки. Т.е. предоставим пользователю возможность задавать отчетный период перед формированием

отчета. То есть параметры Дата начала и Дата окончания включим в состав пользовательских настроек.

Причем поскольку задавать отчетный период требуется практически всегда, эти настройки должны находиться непосредственно в форме отчета.

На закладке **Параметры** (рис. 77а) выделим по очереди каждый из параметров и нажмем кнопку **Свойства элемента пользовательских настроек**, расположенную в правом нижнем углу окна настроек.

Установим флажок **Включать в пользовательские настройки** (рис. 77б) и оставим предложенное по умолчанию для свойства Режим редактирования значение Быстрый доступ.

Флажок **Включать в пользовательские настройки** означает, что эта настройка будет доступна пользователю в отдельном окне при нажатии кнопки Настройка (то есть такая настройка, которой он может пользоваться, но не очень часто).

А режим редактирования, установленный в значение **Быстрый доступ**, означает, что эта настройка также будет автоматически отображаться непосредственно в отчетной форме. Это быстрая пользовательская настройка – такая настройка, которая нужна пользователю постоянно, чуть ли не при каждом запуске отчета. Поэтому она всегда на виду.

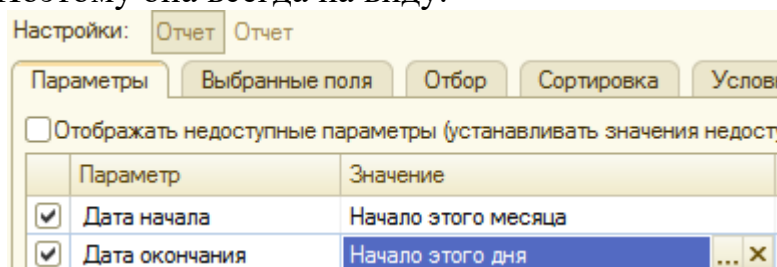


Рис. 77а. Закладка Параметры отчета

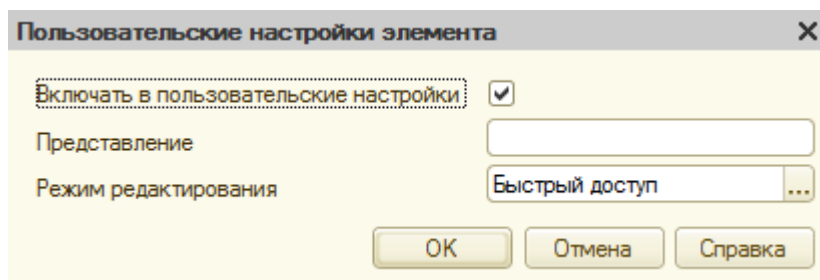


Рис. 77б. Установка параметров Пользовательских настроек элементов

18. Определим, в каких подсистемах будет отображаться отчет (рис. 77в)

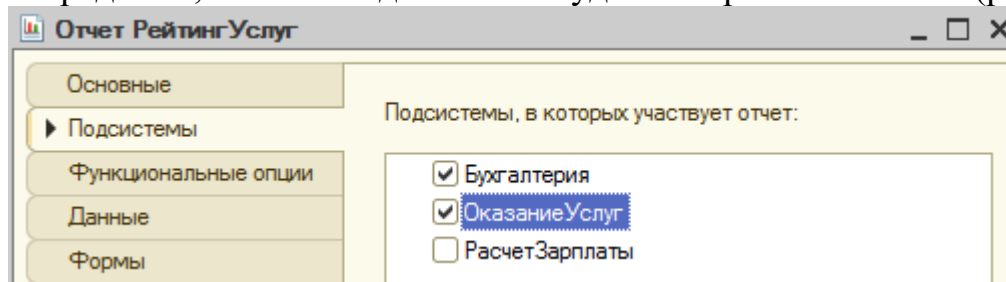



Рис. 77в. Подсистемы, в которых отображается отчет


Проверим в работе выполненные выше действия, запустим приложение в режиме 1С: Предприятие.

ВВОД ДАННЫХ В БД

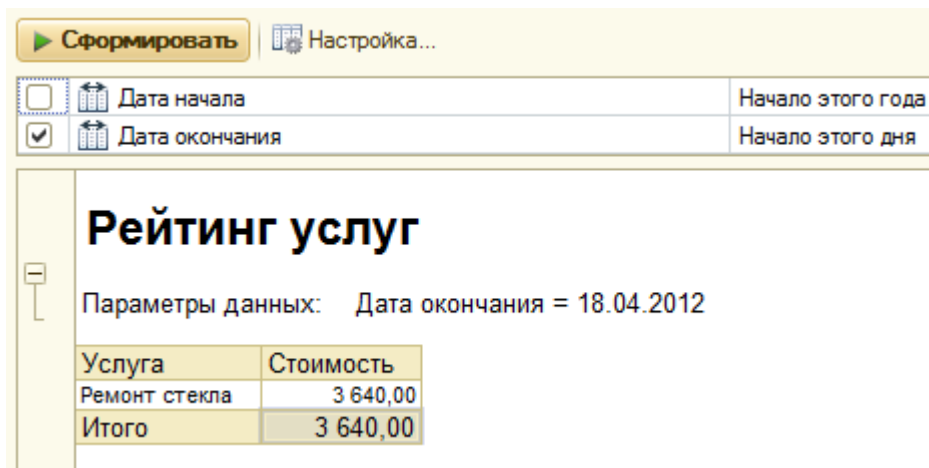
Сформируем отчет Рейтинг услуг.

1. Через форму отчета, автоматически сформированную системой, сформируйте отчет по тем параметрам, которые изначально создавались

☒  Дата начала

☒  Дата окончания

2. Измените дату окончания или дату начала и сформируем отчет (рис. 77г).



Услуга	Стоимость
Ремонт стекла	3 640,00
Итого	3 640,00

Рис. 77г. Отчет Рейтинг услуг

ХIII. ВИДИМОСТЬ ИТОГОВОЙ СУММЫ В ДОКУМЕНТАХ

Произведем дальнейшее совершенствование разрабатываемого решения.

Возникает естественное желание видеть итоговую сумму в форме документа. Чтобы в процессе его создания можно было оперативно, не печатая документ, знать итоговую сумму по документу.

Для этого внесем небольшие изменения в форму документа **ВыполнениеУслуги**.

1. Для редактирования формы документа откроем конфигуратор, раскроем дерево документа **ВыполнениеУслуги** и дважды щелкнем на форме **ФормаДокумента**.

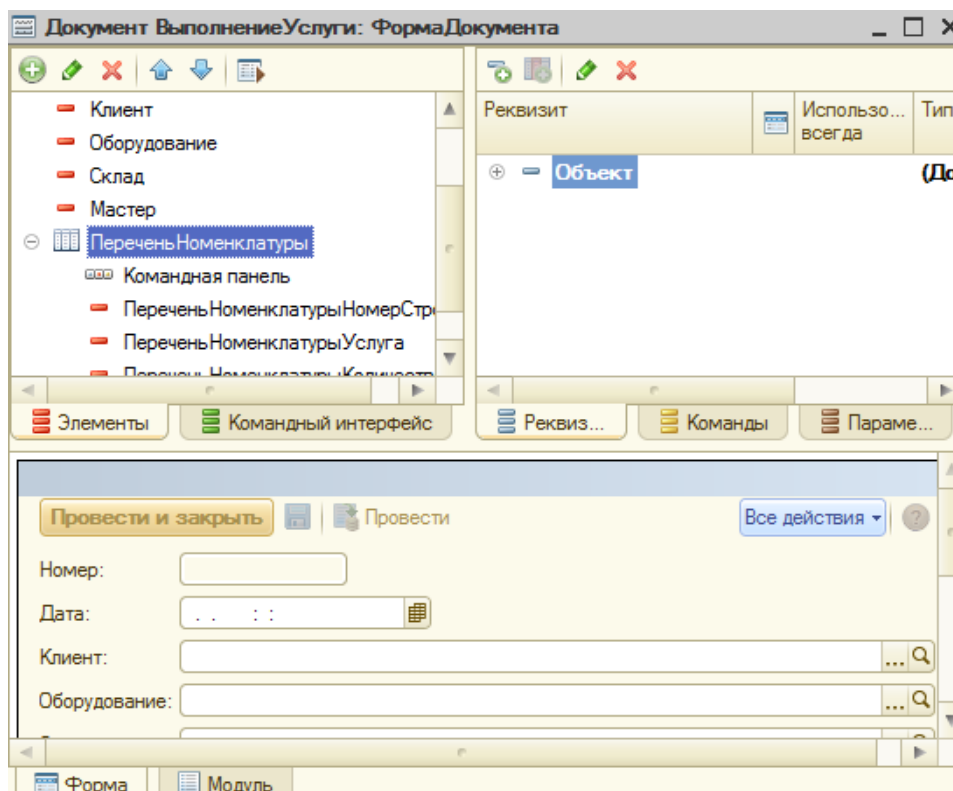


Рис. 78а. Форма документа

2. Дважды щелкнем на элементе **Перечень Номенклатуры** в дереве элементов формы (рис. 78а) или правой кнопкой мыши откроем для него палитру свойств (пункт контекстного меню **Свойства**). Установим свойство **Подвал**, которое определяет наличие подвала у таблицы формы (рис. 78б).

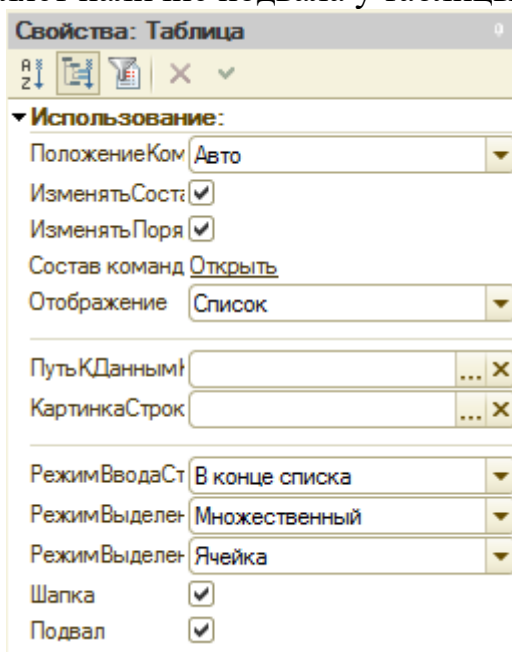


Рис. 78б. Установка видимости подвала у табличной части

3. Откроем свойства элемента **Перечень Номенклатуры Сумма** (рис. 78в), установим Горизонтальное положение в подвале – **Право** (рис. 78г). В свойстве Шрифт подвала тоже изменим начертание на **Жирный** (рис. 78д).

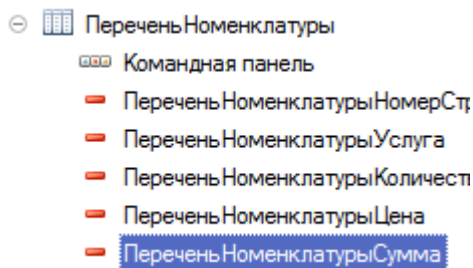


Рис. 78в. Выбор элемента табличной части

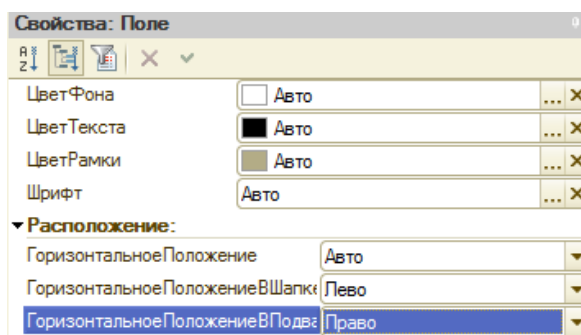


Рис. 78г. Выбор данных подвала – Положение

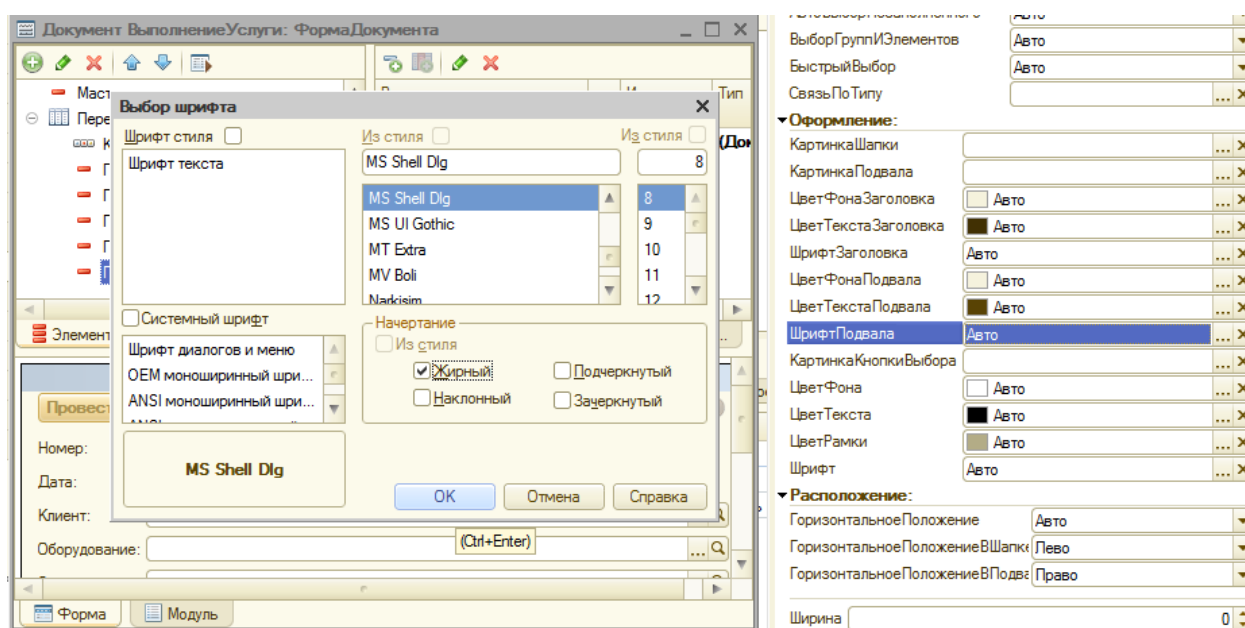


Рис. 78д. Выбор данных подвала – Шрифт

4. Для того чтобы в подвале колонки **Сумма** отображался итог по ней, нажмем кнопку выбора в поле **Путь К Данным Подвала** (рис. 78е).

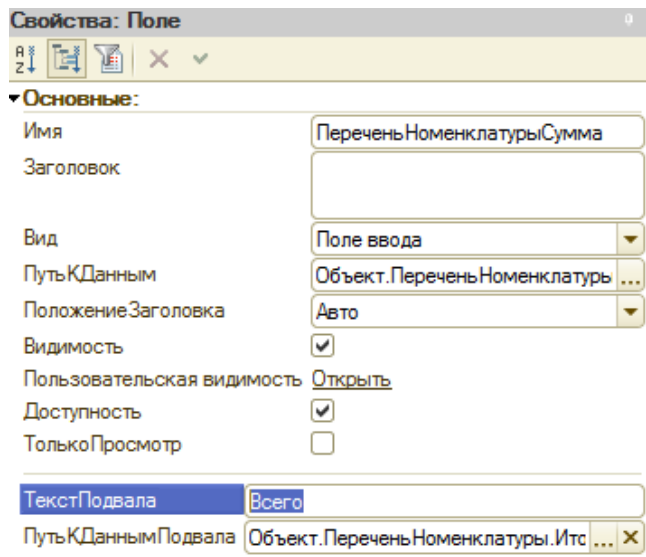


Рис. 78е. Выбор данных подвала для колонки Сумма

Аналогично добавьте строку **Итого** в документ **Акт приема в ремонт**,

Проверим в работе выполненные выше действия, запустим приложение в режиме 1С: Предприятие.

ВВОД ДАННЫХ В БД

1. Перепроведите все введенные ранее документы **Выполнение услуги** и **Акт приема в ремонт**.

Задание на диф. зачёт

Задание предполагает создание средствами 1С: Предприятия автоматизированной информационной системы (АИС).

План выполнения работы:

1 этап. Выбор предметной области АИС.

Предметная область АИС может быть выбрана из предложенных вариантов (приложение А) или сформулирована студентом самостоятельно.

2 этап. Анализ предметной области АИС. Проектирование АИС.

Разработка функциональной модели IDEF0, построение DFD-диаграммы, разработка ER-диаграммы будущей АИС (приложение Б).

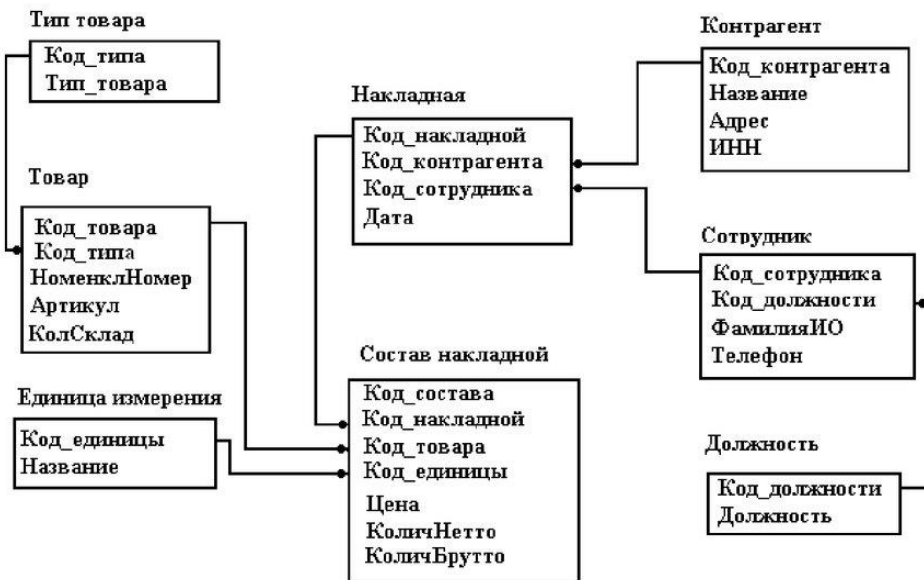


Пример диаграммы DFD



Рис. 3.1. Пример диаграммы DFD

Пример ER – диаграммы для предметной области «Движение товара в фирме»



Для наглядного представления диаграмм возможно использование программ **BPWin**, **ERWin** и др.

3 этап. Разработка АИС.

Разработка программы средствами «1С: Предприятие 8»: разработка программного кода. Разработанные на втором этапе схемы воплощаются в реальные объекты.

Программа должна содержать (минимальные требования):

- 2-3 подсистемы;
- 5 справочников, как минимум один иерархический и один с табличной частью;
- 2 вида документов (заполняются по форме);
- 2 регистра накопления;
- регистр сведений
- 2 вида отчетов;
- перечисление;
- планы видов характеристик;
- бухгалтерский учет и расчет зарплаты (регистры бухгалтерии, план видов расчета, регистр расчета)
- собственный рабочий стол и пользовательский интерфейс.

4 этап. Создание отчета по выполненной работе.

Результаты работы представляются в виде пояснительной записки. Пояснительная записка оформляется на листах формата А4 и должна содержать следующие элементы:

- Титульный лист.
- Описание задачи.
- Описание проведенного анализа предметной области и проектирования АИС средствами структурного подхода.
- Описание структуры программы.

К пояснительной записке должен прилагаться диск с разработанной ЭИС.

Примерный перечень предметных областей АИС

№ варианта	Предметная область	Примерные задачи
1	Компьютерный салон	Учет компьютеров и комплектующих в компьютерном салоне.
2	Учет деятельности фирмы	Учет клиентов; ведение договоров; ввод и хранением информации о наличии и состоянии аппаратного и программного обеспечения клиента; учет сотрудников; учет планирования и выполнения этапов работ; контроль оплаты данных работ; определением заработной платы сотрудников как процента от выполненных работ.
3	Автоматизация сервисного центра	Учет ремонтов оргтехники, сборки компьютеров (комплектов), выдачи гарантийных талонов, ведения склада, учет движения денежных средств, формирование отчетов.
4	Внедрение и техническое обслуживание	Для фирм занимающихся внедрением программного обеспечения: БД потенциальных клиентов с возможностью регистрации проведенных работ по их окучиванию (для менеджеров). БД клиентов (учет уст.программ, техника клиента, договора). Учет проведенных работ внедренцами и системотехниками. Учет доработок в конфигурациях и программах.
5	Кредиты	Учет входящих (получаемых) кредитов. Начисление пени за просроченные платежи в текущей реализации не предусмотрено. Учет исходящих (выдаваемых) кредитов.
6	Домашняя бухгалтерия	Учет ведется в разрезе статей расхода и дохода и принадлежности расходов. Вывод напоминаний по важным событиям и датам. Планирование доходов и расходов.
7	Частный предприниматель	Рознично-оптовая продажа, перемещение товаров, остатки товаров, ведение списка сотрудников и начисление зарплаты в разрезе торговых точек.
8	Абонентский отдел для кабельного телевидения	Ведение базы абонентов кабельного телевидения, поступление денежных средств от абонентов, начисление абонентская плата, формирование счет-фактур, формирование отчетов по задолженностям абонентов, возможность скидок.
9	Учет компьютеров на предприятии	Учет рабочих мест, компьютеров, пользователей ответственных за эти компьютеры. Учет комплектующих на рабочих местах и складах по серийным номерам, гарантии, количеству, цене. Возможность идентификации любой конкретной комплектующей. Работа с комплектами и инвентарными номерами. Отчеты по рабочим местам, ремонту, складам, комплектующим. Учет расходных материалов. Учет заявок пользователей.
10	Учет оргтехники на предприятии	Учет техники, комплектующих, ремонтов, списков доступа, паролей, расходных материалов, уникальность каждой единицы техники и комплектующей.
11	Командировки	Печать командировочных удостоверений и приказов к ним, реестра командировок.
12	Медосмотр сотрудников	Сбор и обработка информации о сотрудниках: прививках, лабораторных анализах, флюорографии и обследовании специалистами. Формирование отчетов о результатах ежегодного обследования сотрудников, о сотрудниках подлежащих обследованию, прививке, флюорографии. Ведение медицинской карты сотрудников.
13	Табель	Заполнение отработанного времени сотрудников.
14	Меню	Расчет стоимости блюд по раскладке, печать калькуляционных карт, меню, хранение кулинарной книги.
15	Кафе+Бар	Учет полуфабрикатов, автоматическое списание полуфабрикатов со склада при создании блюд и продажи.
16	Служба такси	
17	Турфирма	
18	Салон красоты	Учет материалов, учет услуг, учет заработной платы сотрудников, учет записи к специалистам, учет денежных средств, учет скидок по дисконтным картам.

19	Санаторно-курортный комплекс	Учет отдыхающих и услуг.
20	Кадровое агентство	Договор. Заказ от работодателя. Отчеты (зарплата, спрос вакансий от работодателя, спрос вакансий от соискателя).
21	Подписка и подписчики	Учет по подписчикам, текущей подписке и оплате, отслеживание долгов по подписке. Учет по доставке.
22	Сервисный центр	Автоматизации работы сервисного центра: прием заказов, учет движения материалов, инструментов.
23	Рынок-Аренда	Работ с арендаторами на рынке: аренда мест на рынке.
24	Учет сим карт	Учет сим-карт и договоров (контрактов) операторов связи: оформление и печать договоров.
25	Риэлтор	
26	Ломбард	
27	Видеопрокат	
28	Рекламная фирма	Учет рекламных заказов в фирме, занимающейся агентской деятельностью (прием заказов у клиентов и выставление рекламы (журналы, газеты, ТВ)).
29	Органайзер руководителя	Ведение списка заданий сотрудников.
30	Автоматизация работы секретаря	Учет входящих, исходящих документов, командировок и контроль за исполнением.
31	Движение акций внутри предприятия	Учет движения акций предприятия.
32	Обувной магазин	Складской учет товара в розничном магазине, торгующем обувью.
33	Контингент техникума	Для учебного отдела среднего специального учебного заведения: заносить личные данные студентов, определять состав групп, движение по группам, информацию по учебным предметам, курсовым и дипломным проектам, вносить сведения об успеваемости, выводить среднюю оценку.
34	Медицинское учреждение (регистратура)	Ведение регистрации больных в мед. Учреждениях: регистратура в поликлинике и прием пациента специалистом, учет начислений по приему пациентов, учет выписки льготных рецептов.
35	Рабочее место стоматолога	Планировка расписания клиентов. Ведение истории болезни, истории посещений. Отчет о взаиморасчетах.
36	Мебельное производство	
37	Производство окон-ПВХ	
38	Типография	Составление калькуляции по заказу. Отчет по остаткам на складах.
39	Производство полуфабрикатов	Учет при производстве полуфабрикатов (котлет, пельменей и т.п.) и реализации.
40	Магазин	Складской учет товара в розничном магазине, торгующем одеждой.

Анализ предметной области АИС. Проектирование АИС

В области создания систем автоматизации деятельности доминируют структурные подходы, так как они максимально приспособлены для взаимодействия с пользователями, не являющимися специалистами в области информационных технологий. Сущность структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции. Система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и т.д. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны.

Адекватными инструментальными средствами, поддерживающими структурный подход к созданию информационных систем, являются так называемые **CASE**-системы автоматизации проектирования.

Основополагающая концепция состоит в построение совокупности логических моделей предметной области при помощи графических методов структурного анализа, которые дали бы возможность пользователям, аналитикам и разработчикам получить ясную общую картину проекта, а также обеспечили бы естественный переход к логической модели будущей ИС.

В основе деятельности по созданию и использованию ИС лежит понятие жизненного цикла.

Жизненный цикл – это модель создания и использования ИС, отражающая ее различные состояния, начиная с момента возникновения необходимости в данном комплексе средств и заканчивая моментом его полного выхода из употребления у пользователей.

Опыт создания и использования заказных ИС позволяет условно выделить следующие основные этапы их жизненного цикла:

- анализ – определение того, что должна делать система;
- проектирование – определение того, как система будет делать то, что она должна делать. Проектирование это, прежде всего, спецификация подсистем, функциональных компонентов и способов их взаимодействия в системе;
- разработка – создание функциональных компонентов и подсистем по отдельности, соединение подсистем в единое целое;
- тестирование – проверка функционального и параметрического соответствия системы показателям, определенным на этапе анализа;
- внедрение – установка и ввод системы в действие;
- сопровождение – обеспечение штатного процесса эксплуатации системы на предприятии заказчика.

Этапы разработки, тестирования и внедрения ИС обозначаются единым, объемлющим термином – реализация.

Существующие модели жизненного цикла, определяют порядок исполнения этапов в процессе создания ИС, а также критерии перехода от этапа

к этапу. В соответствии с этим наибольшее распространение получила каскадная модель, которая предполагает переход на следующий этап после полного завершения работ предыдущего этапа (характерна для военно-технических проектов).

Главная особенность современной индустрии ИС состоит в концентрации усилий на двух начальных этапах ее жизненного цикла – **анализе** и **проектировании**, при относительно невысокой сложности и трудозатратах на последующих этапах.

Этап анализа предполагает подробное исследование бизнес-процессов (функций, определенных на этапе выбора стратегии) и информации, необходимой для их выполнения (сущностей, их атрибутов и связей (отношений)). На этом этапе создается **информационная модель**, а на следующем за ним этапе проектирования — **модель данных**. Целью **анализа** является преобразование общих, расплывчатых знаний об исходной предметной области в точные определения и спецификации, а также генерация функционального описания системы.

Структурный анализ начинается с исследования того, как организована система управления предприятием, с обследования функциональной и информационной структуры системы управления. По результатам обследования аналитик на первой стадии анализа строит обобщенную логическую модель исходной предметной области, отображающую ее функциональную структуру, особенности основной деятельности и информационное пространство, в котором эта деятельность осуществляется. Используя специальную терминологию, можно сказать, что аналитик строит модель «**как есть**».

Вторая стадия работы, к которой привлекаются заинтересованные представители заказчика, а при необходимости и независимые эксперты, состоит в анализе модели «как есть», выявлении ее недостатков и узких мест, определение путей совершенствования системы управления на основе выделенных критериев качества.

Третья стадия анализа, содержащая элементы **проектирования**, – создание усовершенствованной обобщенной логической модели, отображающей реорганизованную предметную область или ее часть, которая подлежит автоматизации. Эту модель можно назвать моделью «**как надо**».

Заканчивается процесс разработкой «карты автоматизации», представляющей собой модель реорганизованной предметной области, на которой обозначены «границы автоматизации».

В качестве средств структурного анализа, наиболее распространение получила нотация **IDEF0**. Для новых IDEF0 применяется для определения требований (функций) для разработки системы, реализующей выделенные функции. Для уже существующих – IDEF0 может быть использована для анализа функций, выполняемых системой. Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Вершина этой древовидной структуры, представляющая собой самое общее описание системы. После описания системы в целом проводится разбиение ее на крупные фрагменты (функциональная декомпозиция).

Модели IDEF0 наиболее удобны при построении функциональных моделей. Они наглядно отражают функциональную структуру объекта: производимые действия, связи между этими действиями. Таким образом, четко прослеживается логика и взаимодействие процессов организации. Главным достоинством нотации является возможность получить полную информацию о каждой работе, благодаря ее жестко регламентированной структуре. С ее помощью можно выявить все недостатки, касающиеся как самого процесса, так и то, с помощью чего он реализуется: дублирование функций, отсутствие механизмов, регламентирующих данный процесс, отсутствие контрольных переходов и т.д.

Следующей нотацией, которую используют при реализации этапов анализа и проектирования, является диаграмма потоков данных (**DFD**). DFD-диаграммы являются основным средством моделирования функциональных требований проектируемой системы. С их помощью эти требования разбиваются на функциональные компоненты (процессы) и представляются в виде сети, связанной потоками данных. Главная цель таких средств – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

DFD моделируют функции, которые система должна выполнять, но ничего (или почти ничего) не сообщают об отношениях между данными – для этих целей методологии используют диаграммы инфологических моделей «сущность-связь» (**ER-диаграмма**). ER-диаграмма строится на этапе проектирования. ER-диаграммы используются для разработки данных и представляют собой стандартный способ определения данных и отношений между ними. Таким образом, осуществляется детализация хранилищ данных. ER-диаграмма содержит информацию о сущностях системы и способах их взаимодействия, включает идентификацию объектов, важных для предметной области (сущностей), свойств этих объектов (атрибутов) и их отношений с другими объектами (связей). На этапе проектирования формируется модель данных. Конечным продуктом этапа проектирования является схема базы данных (на основании ER-модели, разработанной на этапе анализа). Информационная база 1С является реляционной, она состоит из набора таблиц формата DBF, связанных между собой.