

Лабораторная работа 7

Проведение расходного документа, журналы документов, программная работа с документами

Посвящена конструированию процедуры проведения расходного документа, особое внимание уделено созданию сложных запросов с помощью консоли запросов. Рассмотрена работа с объектом Журнал документов, а так же – методы программной работы с документами.

Проведение расходного документа

Мы уже настроили проведение документа, который отражает приход материалов. Теперь займемся проведением документа, отвечающего за их списание. Это – документ **ОтпускМатериаловМастеру**. Для начала сформулируем цели, которых мы хотим достичь при проведении данного документа. Если в случае с документом поступления материалов в нашей конфигурации все довольно просто – нам нужно лишь отразить поступление материалов без каких-либо дополнительных проверок или расчетов, то списание материалов даже в нашем, достаточно простом случае, превращается в непростую задачу.

Во-первых, мы хотим, чтобы система не позволяла списать больше материалов, чем числится за конкретным ответственным лицом. Это означает, что перед формированием движений мы должны сверить данные, введенные в табличную часть документа с данными по остаткам материалов, хранящимися в нашей базе, и, в том случае, если материалов нам не хватит – отказать в проведении документа и сообщить пользователю об ошибке.

Во-вторых, списывая материалы, мы должны придерживаться какой-либо политики оценки. Наиболее простая и широко используемая политика – это списание материалов по средней стоимости.

Предположим, мы приняли две партии одного и того же товара – в первой было 10 единиц по 9 рублей единица (общая стоимость 90 рублей), во второй – 20 единиц по 12 рублей единица (общая стоимость 240 рублей). В регистре мы храним общую стоимость материалов и их общее количество. Поэтому, списывая по методу средней стоимости, например, 15 единиц товара, мы сначала должны найти среднюю стоимость единицы имеющихся товаров, разделив их общую стоимость на общее количество – в нашем случае это $(90+240)/(10+20)=11$ рублей – и умножить полученную стоимость на количество списываемых товаров – то есть, $11*15=165$ рублей. Таким образом, мы сможем оценивать стоимость списанных товаров и оценивать стоимость остатков.

Существуют и другие методы списания материальных ценностей, в частности – это ФИФО, ЛИФО. Вполне можно организовать учет каждой единицы материальных ценностей и списание их по индивидуальной себестоимости. Мы реализуем списание материалов по средней стоимости.

Определившись с нашими двумя основными задачами – реализации списания материалов по средней стоимости и контроля остатков, приступим к работе над процедурой для проведения нашего документа.

Перейдем в модуль объекта документа **ОтпускМатериаловМастеру**, с помощью панели инструментов **Модуль** создадим процедуру **ОбработкаПроведения**. Данные из

табличной части мы будем получать с помощью запроса – в дальнейшем мы будем развивать этот запрос для получения необходимых сведений об остатках номенклатуры.

При создании запроса очень удобно пользоваться консолью запросов, которая позволяет в режиме 1С:Предприятие тут же проверять результаты, возвращаемые запросом. Подобные обработки можно найти на дисках ИТС, на различных Интернет-ресурсах, в частности, мы использовали в примерах консоль запросов для управляемого приложения, которую можно найти в приложении к лабораторной работе. Итак, обработку консоли запросов следует открыть командой **Главное меню > Файл > Открыть**. Начнем конструировать запрос, выбрав все поля из таблицы документа **ОтпускМатериаловМастеру**, рис. 7.1.

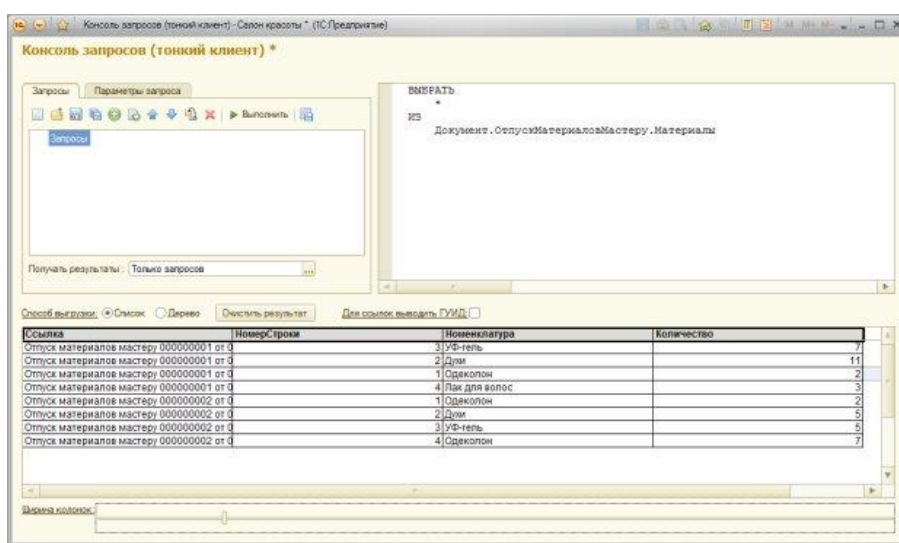


Рис. 7.1. Конструирование запроса с помощью консоли запросов

Наш запрос имеет такой вид:

```
ВЫБРАТЬ
*
ИЗ
Документ.ОтпускМатериаловМастеру.Материалы
```

Мы выбрали все поля из таблицы, не вводя никаких ограничений. Мы собираемся получать данные из таблицы документа, проведением которого мы занимаемся. В запросе же мы получили данные по всем документам. Ограничим наш запрос по документу. Модифицируем запрос в консоли таким образом:

```
ВЫБРАТЬ
*
ИЗ
Документ.ОтпускМатериаловМастеру.Материалы
ГДЕ
Ссылка=&Ссылка
```

Для того, чтобы задать параметр запроса в консоли запросов, перейдем на вкладку **Параметры запроса**, нажмем на кнопку **Заполнить**, после чего в поле **Значение параметра** выберем нужное его значение, в нашем случае – это будет один из документов **ОтпускМатериаловМастеру**, рис. 7.2.

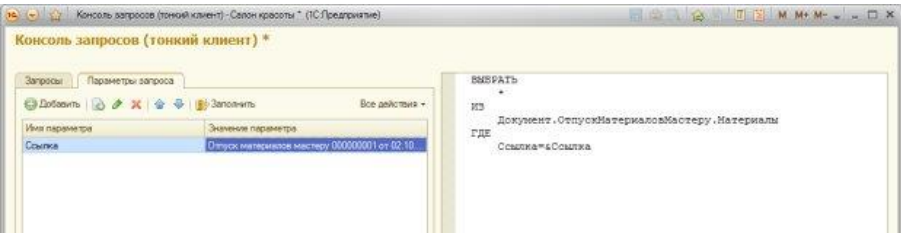


Рис. 7.2. Настройка параметра запроса

Выполнив этот запрос, получим таблицу **Материалы** из указанного документа. Теперь подумаем над тем, какие именно данные нас интересуют. Нам нужны, во-первых, сведения о номенклатуре (поле **Номенклатура**), во-вторых – о количестве номенклатуры, которую мы хотим списать (поле **Количество**). Модифицируем запрос следующим образом:

```
ВЫБРАТЬ
    ДокМ.Номенклатура,
    ДокМ.Количество
ИЗ
    Документ.ОтпускМатериаловМастеру.Материалы КАК ДокМ
ГДЕ
    Ссылка=&Ссылка
```

Этот запрос даст нам такой результат:

| Номенклатура | Количество |
|---------------|------------|
| Одеколон | 2 |
| Духи | 11 |
| УФ-гель | 7 |
| Лак для волос | 3 |
| Одеколон | 3 |

В документе мы намеренно смоделировали ситуацию, в которой пользователь, заполняя его, два раза ввел одну и ту же номенклатурную позицию. Сгруппируем теперь результаты запроса по полю **Номенклатура** – придем к такому тексту запроса:

```
ВЫБРАТЬ
    ДокМ.Номенклатура,
    СУММА (ДокМ.Количество) КАК Количество
ИЗ
    Документ.ОтпускМатериаловМастеру.Материалы КАК ДокМ
ГДЕ
    Ссылка=&Ссылка
ГРУППИРОВАТЬ ПО ДокМ.Номенклатура
```

Здесь мы применили функцию **СУММА** к количественному показателю и с помощью выражения **СГРУППИРОВАТЬ ПО** сгруппировали результаты по полю **Номенклатура**. Это привело к такому результату:

| Номенклатура | Количество |
|---------------|------------|
| Духи | 11 |
| Одеколон | 5 |
| УФ-гель | 7 |
| Лак для волос | 3 |

Теперь все данные, которые нужны нам для проведения документа, мы получили. Следующим этапом работы над запросом будет добавление в него команд для выбора нужных данных из регистра накопления **ОстаткиМатериалов**. Мы приходим к такому запросу:

```
ВЫБРАТЬ
    ДокМ.Номенклатура,
    СУММА (ДокМ.Количество) КАК Количество,
    МАКСИМУМ (ОстМ.КоличествоОстаток) КАК КоличествоОстатков,
    МАКСИМУМ (ОстМ.СуммаОстаток) КАК СуммаОстатков
ИЗ
    Документ.ОтпускМатериаловМастеру.Материалы КАК ДокМ
ЛЕВОЕ СОЕДИНЕНИЕ
    РегистрНакопления.ОстаткиМатериалов.Остатки(&МоментВремени) Как ОстМ
ПО
    ДокМ.Номенклатура = ОстМ.Номенклатура
ГДЕ
    Ссылка=&Ссылка
СГРУППИРОВАТЬ ПО ДокМ.Номенклатура
```

Добавлено примечание ([Г1]): Не работает запрос ругается на момент времени чтобы решить проблему перейдем на закладку параметры, щелкаем Заполнить и вносим ссылки на конкретный док, момент времени(дату проведения документа) и отв лицо (выбираем то, на которое списываются материалы.)

Здесь мы соединили таблицу регистра остатков с полученной таблицей документа по полю **Номенклатура**. К числовым полям, полученным из регистра, мы применили функцию **МАКСИМУМ** – иначе запрос будет выполняться неверно. В частности, если бы выше мы не выполнили группировку результатов запроса по полю **Номенклатура**, то в результатах запроса мы получили бы несколько полей с одной и той же номенклатурой, к каждому из которых было бы присоединено одно и то же поле из таблицы регистра.

Поля в таблице документа несут значимую информацию – если даже номенклатура повторяется, это подразумевает то, что пользователь сознательно ввел ее несколько раз (возможно для того, чтобы, выведя печатную форму документа, добавить какие-то дополнительные комментарии). А присоединение к этим данным одних и тех же данных из регистра накопления нас не устраивает. Поэтому мы суммируем данные по количеству материалов из табличной части, а к полям таблицы регистров применяем функцию **МАКСИМУМ**. Применение функции **МАКСИМУМ** здесь не принципиально – с тем же успехом можно было бы применить и функцию **МИНИМУМ**, и другие подходящие – их суть сводится к переходу от нескольких одинаковых значений к одному.

Вышеописанный запрос привел к такому результату:

| Номенклатура | Количество | КоличествоОстатков | СуммаОстатков |
|--------------|------------|--------------------|---------------|
| Духи | 11 | 36 | 4 899 |
| Одеколон | 5 | 18 | 3 510 |

| | | | |
|---------------|---|------|-------|
| УФ-гель | 7 | 11 | 3 350 |
| Лак для волос | 3 | NULL | NULL |

Здесь нас не устраивают два момента. Во-первых, в полях **КоличествоОстатков** и **СуммаОстатков** показаны данные по всем ответственным лицам – а нам нужно знать данные лишь по тому ответственному, с которого мы материалы списываем. Во-вторых, по номенклатурной позиции, по которой данных в регистре **ОстаткиМатериалов** не имеется, в полях находится значение **NULL**. Для того, чтобы попытка работать с этим значением не привела в будущем к возникновению ошибок, обработаем поля, полученные из регистра, функцией **ЕСТЬNULL**.

Модифицируем запрос в соответствии с последними соображениями.

```

ВЫБРАТЬ
    ДокМ.Номенклатура,
    СУММА (ДокМ.Количество) КАК Количество,
    МАКСИМУМ (ЕСТЬNULL(ОстМ.КоличествоОстаток, 0)) КАК КоличествоОстатков,
    МАКСИМУМ (ЕСТЬNULL(ОстМ.СуммаОстаток, 0)) КАК СуммаОстатков
ИЗ
    Документ.ОтпускМатериаловМастеру.Материалы КАК ДокМ
ЛЕВОЕ СОЕДИНЕНИЕ
    РегистрНакопления.ОстаткиМатериалов.Остатки (&МоментВремени ,
    ОтветственныйСотрудник = &ОтвСотр) Как ОстМ
ПО
    ДокМ.Номенклатура = ОстМ.Номенклатура
ГДЕ
    Ссылка=&Ссылка
ГРУППИРОВАТЬ ПО ДокМ.Номенклатура

```

Здесь мы добавили отбор из регистра только записей, относящихся к заданному ответственному сотруднику (**ОтветственныйСотрудник = &ОтвСотр**) и применили к показателям количества и суммы, полученным из регистра, функцию **ЕСТЬNULL**. Если в поле находится **NULL**, мы заменяем это значение нулем.

Результат запроса теперь выглядит так:

| Номенклатура | Количество | КоличествоОстатков | СуммаОстатков |
|---------------|------------|--------------------|---------------|
| Духи | 11 | 15 | 1 860 |
| Одеколон | 5 | 2 | 400 |
| УФ-гель | 7 | 3 | 900 |
| Лак для волос | 3 | 0 | 0 |

На данном этапе запрос возвращает нам исключительно те данные, которые нужны для решения задачи. А именно – эти данные позволяют нам понять, хватит ли материалов, числящихся за ответственным лицом, для списания запрошенного в документе количества материалов. Так же, этими данными мы сможем воспользоваться при формировании движений по регистру накопления – эти движения можно будет формировать в цикле обхода результатов запроса.

Скопируем полученный текст запроса в буфер обмена и перейдем в **Конфигуратор**. В процедуре **ОбработкаПроведения** документа **ОтпускМатериаловМастеру**. Процедура пока пуста, щелкнем в ней правой кнопкой мыши и вызовем из контекстного меню команду **Конструктор запроса с обработкой результата**. В ответ на вопрос

Добавлено примечание ([Г2]): Эта обработка проведения документа вставляется на закладке прочие модуль документа, щелкаем на добавление процедуры и выбираем «Обработка проведения». Запрос вставляем.

конструктора о создании нового запроса, ответим утвердительно, после чего, в окне конструктора нажмем на кнопку **Запрос** и вставим в пустое поле для текста запроса полученный текст запроса (предварительно нажав на кнопку **Редактировать запрос** в окне **Запрос**), рис. 7.3.

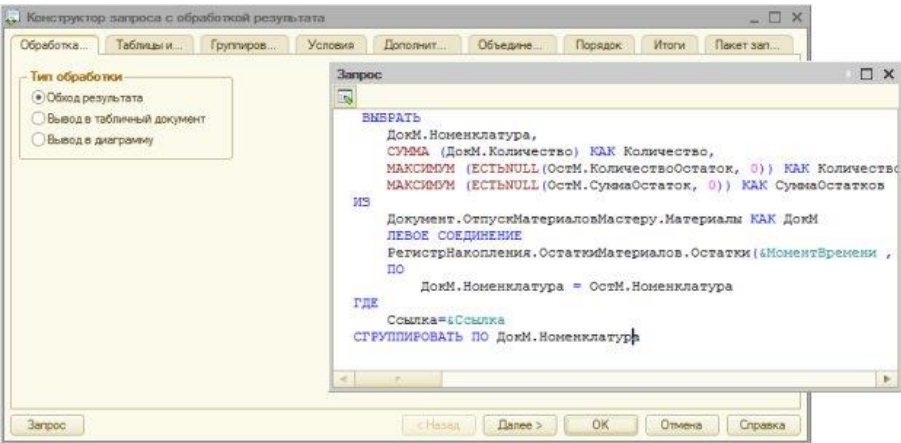


Рис. 7.3. Добавление сформированного текста запроса в конструктор

На закладке **Обработка** окна **Конструктор запроса** оставим переключатель **Тип обработки** в положении **Обход результата**. После закрытия окна **Запрос** конструктор автоматически разберет запрос, "разложит" по закладкам своего окна, при необходимости, его можно будет редактировать, пользуясь инструментами, расположенными на этих закладках. Наш запрос устраивает – поэтому мы можем нажимать в окне конструктора **ОК** и переходить к дальнейшему редактированию кода, рис. 7.4.

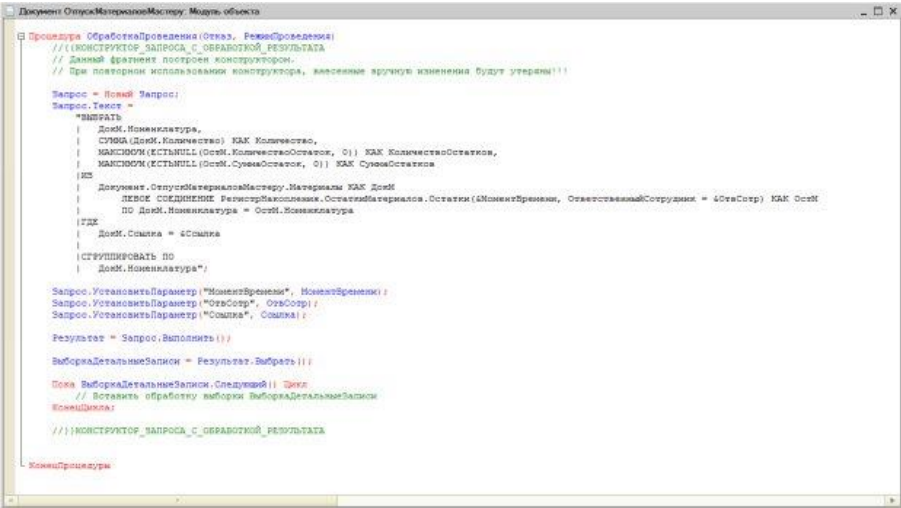


Рис. 7.4. Добавление сформированного текста запроса в конструктор

Здесь нас, в первую очередь, не устраивает автоматическое заполнение параметров запроса

Заменяем код:

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

На код:

```
Запрос.УстановитьПараметр ("МоментВремени", МоментВремени());
Запрос.УстановитьПараметр ("ОтвСотр", ОтветственныйСотрудник);
```

Здесь мы, во-первых, вызвали метод **МоментВремени()**, возвращающий момент времени для нашего документа (то есть – для того, в модуле объекта которого мы сейчас работаем). Во-вторых, мы обратились к реквизиту документа **ОтветственныйСотрудник** для установки параметра **ОтвСотр**.

Проверим работу созданного механизма, запустив систему в режиме отладки и установив в коде модуля точку останова после получения выборки из результатов запроса.

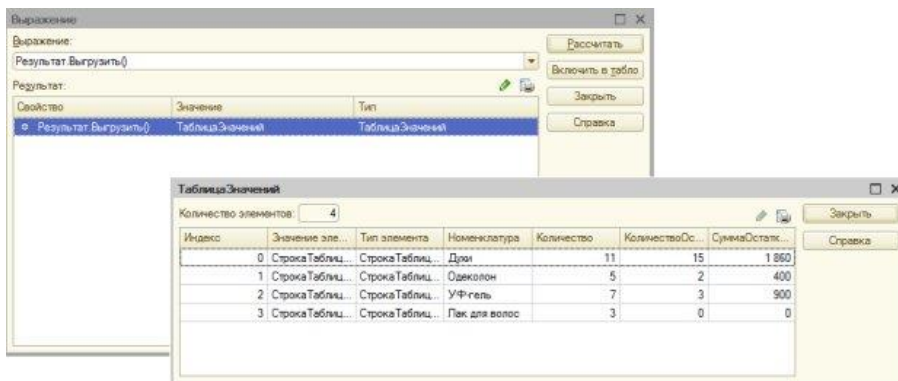


Рис. 7.5. Анализ результата выполнения запроса в коде процедуры проведения документа

Здесь мы выполнили метод **Выгрузить()** для результата выполнения запроса (переменная **Результат**), получили таблицу значений, которую можно проанализировать. Результат нас устраивает, поэтому мы принимаемся за дальнейшую работу над процедурой. В итоге у нас получился следующий код:

```
Процедура ОбработкаПроведения(Отказ, РежимПроведения)
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| ДокМ.Номенклатура,
| СУММА (ДокМ.Количество) КАК Количество,
```

```

| МАКСИМУМ(ЕСТЬNULL(ОстМ.КоличествоОстаток, 0)) КАК КоличествоОстатков,
| МАКСИМУМ(ЕСТЬNULL(ОстМ.СуммаОстаток, 0)) КАК СуммаОстатков
| ИЗ
| Документ.ОтпускМатериаловМастеру.Материалы КАК ДокМ
| ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ОстаткиМатериалов.Остатки(&МоментВремени,
ОтветственныйСотрудник = &ОтвСотр) КАК ОстМ
| ПО ДокМ.Номенклатура = ОстМ.Номенклатура
| ГДЕ
| ДокМ.Ссылка = &Ссылка
|
| СГРУППИРОВАТЬ ПО
| ДокМ.Номенклатура";

```

```

Запрос.УстановитьПараметр("МоментВремени", МоментВремени());
Запрос.УстановитьПараметр("ОтвСотр", ОтветственныйСотрудник);
Запрос.УстановитьПараметр("Ссылка", Ссылка);

```

```

Результат = Запрос.Выполнить();

```

```

ВыборкаДЗ = Результат.Выбрать();

```

```

Движения.ОстаткиМатериалов.Записывать=Истина;

```

```

Пока ВыборкаДЗ.Следующий() Цикл
Если ВыборкаДЗ.Количество>ВыборкаДЗ.КоличествоОстатков Тогда
Сообщить("Недостаточное количество товара "+ВыборкаДЗ.Номенклатура
+", необходимо "+ВыборкаДЗ.Количество+", в наличии "
+ВыборкаДЗ.КоличествоОстатков);
Отказ=Истина;
Движения.ОстаткиМатериалов.Записывать=Ложь;
КонецЕсли;

```

```

Если Отказ Тогда
Продолжить;
КонецЕсли;

```

```

Движение=Движения.ОстаткиМатериалов.Добавить();
Движение.ВидДвижения=ВидДвиженияНакопления.Расход;
Движение.Период=Дата;
Движение.Номенклатура=ВыборкаДЗ.Номенклатура;
Движение.Количество=ВыборкаДЗ.Количество;

```

```

Движение.Сумма=ВыборкаДЗ.Количество*ВыборкаДЗ.СуммаОстатков/ВыборкаДЗ.Количес
твоОстатков;
Движение.ОтветственныйСотрудник=ОтветственныйСотрудник;
Движение.ПолучательМатериалов=ПолучательМатериалов;
КонецЦикла;

```

```

КонецПроцедуры

```

Здесь мы, для краткости, переименовали переменную **ВыборкаДетальныеЗаписи в ВыборкаДЗ**, добавили команду, включающую запись движений по регистру **Остатки Материалов**, после чего занялись кодом, располагающимся в цикле обхода полученных запросом данных.

Сначала мы сравнили требуемое и реальное количество материалов. Если нужно больше, чем имеется за конкретным ответственным лицом, мы формируем сообщение для пользователя, устанавливаем параметр **Отказ** в значение **Истина** (то есть указываем системе на то, что документ мы проводить не будем), и отключаем запись данных в регистр накопления. После проверки количества мы выполняем еще одну проверку – на

состояние переменной **Отказ**. Если эта переменная установлена в значение **Истина** – мы переходим к следующей итерации цикла, не выполняя оставшиеся команды. Если же **Отказ** установлен в значение **Ложь**, мы формируем запись по регистру **ОстаткиМатериалов** с использованием текущих данных. Обратите внимание на то, как рассчитывается поле **Сумма** – для его получения мы умножаем количество материалов, которые нужно списать, на результат деления общей суммы остатков материалов на общее количество остатков.

Проверим результаты работы нашего кода в режиме 1С:Предприятие. Если документ верно реагирует на попытку списания материалов, количество которых превышает имеющееся, и если анализ состава регистра накопления после проведения показывает, что списано нужное количество материалов и их стоимость определена верно – можно считать, что мы справились с поставленной задачей.

Теперь в нашей конфигурации присутствует документ, движения которого списывают данные из регистра накопления. Для анализа состояния дел полезно иметь отчет, который способен показывать не только остатки материалов на определенную дату, но и обороты – приход и расход материалов.

Отчет – приход и расход материалов

Мы собираемся построить отчет, который выводил бы сведения о начальном и конечном остатке материалов за определенный временной интервал, а так же – сведения о приходе и расходе материалов за этот период.

Создадим новый отчет, назовем его **Материалы**, включим в подсистему **ОперативныйУчетМатериалов**, добавим основную схему компоновки данных, создадим новый набор данных – Запрос. В конструкторе запроса выберем из виртуальной таблицы регистра накопления **ОстаткиМатериалов. ОстаткиИОбороты** следующие поля, рис. 7.6.:

- Номенклатура
- ОтветственныйСотрудник
- КоличествоНачальныйОстаток
- КоличествоПриход
- КоличествоРасход
- КоличествоКонечныйОстаток

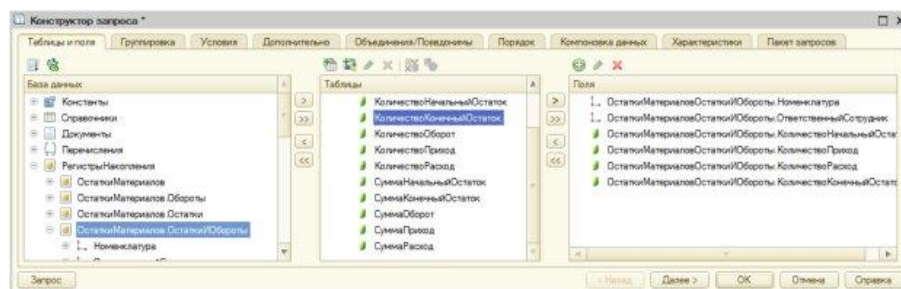


Рис. 7.6. Настройка запроса для отчета

Нажмем **ОК** в окне конструктора запроса, перейдем на закладку **Ресурсы** окна редактора СКД, добавим все количественные поля в состав ресурсов, рис. 7.7.

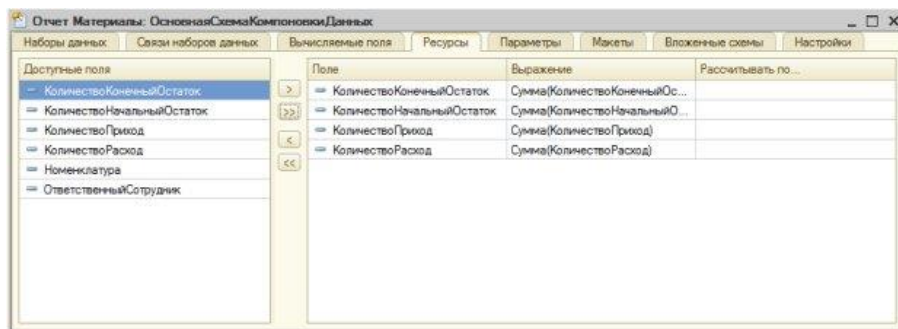


Рис. 7.7. Настройка состава ресурсов

На закладке **Настройки** воспользуемся конструктором настроек. Выберем табличный тип отчета, нажмем **Далее**, в окне настройки состава и порядка следования полей, которые будут отображаться в отчете, расположим поля следующим образом:

- Номенклатура
- ОтветственныйСотрудник
- КоличествоНачальныйОстаток
- КоличествоПриход
- КоличествоРасход
- КоличествоКонечныйОстаток

На следующем этапе укажем, что группировка строк будет осуществляться по полю **Номенклатура**, колонок – по полю **ОтветственныйСотрудник**.

На этапе настройки упорядочения укажем упорядочение по возрастанию по полю **Номенклатура**.

На этом настройка таблицы завершена.

На верхнем уровне настроек отчета укажем, что параметры **Начало периода** и **Конец периода** следует включать в пользовательские настройки.

Отчет готов, нам осталось лишь проверить его работу в режиме 1С:Предприятие, рис. 7.8.

Рис. 7.9. Настройка журнала документов

В режиме 1С:Предприятие наш журнал позволит просматривать список документов разных типов, включенных в него, рис. 7.10.

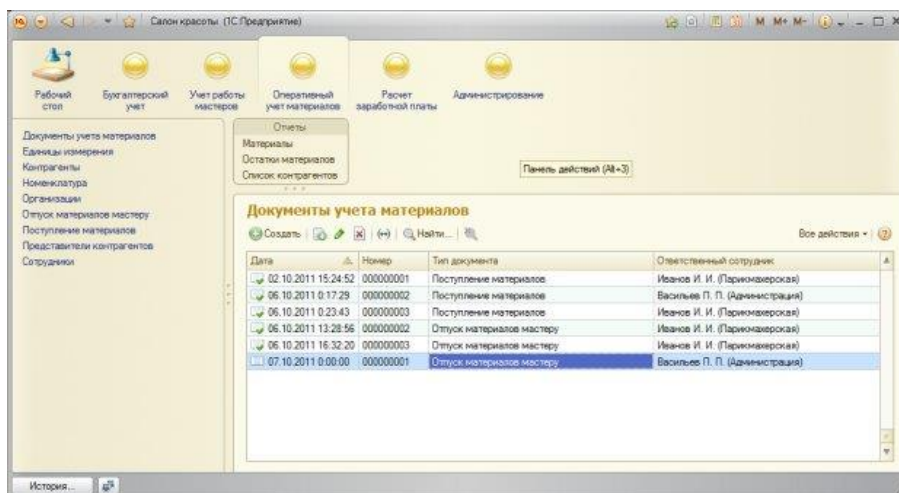


Рис. 7.10. Журнал документов в режиме 1С:Предприятие

Программная работа с документами

С документами в пользовательском режиме можно работать не только интерактивно – так, как мы это делали до сих пор, но и программно. Рассмотрим объекты 1С:Предприятие, которые используются для программной работы с документами.

ДокументыМенеджер – этот объект доступен через свойство глобального контекста **Документы**. Он позволяет управлять документами, давая доступ к объекту **ДокументМенеджер** для конкретного документа

ДокументМенеджер – позволяет управлять документом, в частности, создавать новые документы, производить поиск документов, помечать документы на удаление, получать выборы.

ДокументСсылка – идентификатор документа. Используется обычно в реквизитах других объектов для указания ссылки на определенный документ.

ДокументОбъект – позволяет работать с отдельным документом – читать и редактировать его данные.

ДокументВыборка – предназначен для работы с выборкой документов, произведенной по определенным условиям.

ДокументСписок – позволяет управлять списком документов, расположенных в табличном поле.

Для доступа к метаданным документа можно воспользоваться свойством **Документы** свойства глобального контекста **Метаданные**.

Создадим новую обработку, назовем ее **РаботаСДокументами**. Включим в подсистему **Администрирование**.

Добавим в обработку команду с именем **ВывестиСписокВидовДокументов**, зададим обработчик для этой команды, выведем ее на форму обработки.

Сейчас мы воспользуемся свойством глобального контекста **Метаданные** для того, чтобы вывести пользователю список синонимов существующих в конфигурации документов. Для подобных действий нам понадобится серверная процедура, которую мы вызовем из клиентской процедуры обработчика ранее созданной команды. Выполнить запланированное можно с помощью следующего кода:

```
&НаКлиенте
Процедура ВывестиСписокВидовДокументов (Команда)
    ВывестиСинонимыДокументов ();
КонецПроцедуры

Процедура ВывестиСинонимыДокументов ()
    Для каждого Документ из Метаданные.Документы Цикл
        Сообщить (Документ.Синоним) ;
    КонецЦикла;
КонецПроцедуры
```

Результат выполнения показан на рис. 7.11.

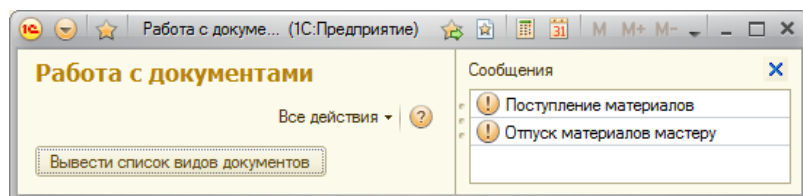


Рис. 7.11. Вывод списка синонимов документов

С помощью коллекции **Метаданные** (тип **ОбъектМетаданныхКонфигурация**) мы можем получить доступ к структуре конфигурации, в частности, к коллекции **Документы** (тип **КоллекцияОбъектовМетаданных**), состоящая из объектов типа **ОбъектМетаданных: Документ**. Обратившись к документу, как к объекту метаданных, мы можем узнать имя, синоним и другие метаданные.

Теперь рассмотрим следующую задачу. Нужно программно создать новый документ заданного вида, заполнить его некоторыми данными, сохранить в базе данных, приняв решение о том, проводить его или не проводить на основе анализа флага **ПроводитьДокумент**, который имеется на форме обработки, после чего открыть его форму для проверки пользователем.

Здесь нам понадобится начинать "поиски" нужного объекта от объекта **ДокументыМенеджер**. С его помощью мы можем получить объект **ДокументМенеджер** для нужного вида документов, и уже с помощью этого объекта мы сможем создать новый документ.

Добавим в нашу обработку новую команду – **СоздатьДокументПоступлениеМатериалов**. Так же добавим новый реквизит – **ПроводитьДокумент**, поместим его на форму, рис. 7.12.

Мы зададим все данные, в том числе – и тип документа для создания – в коде.

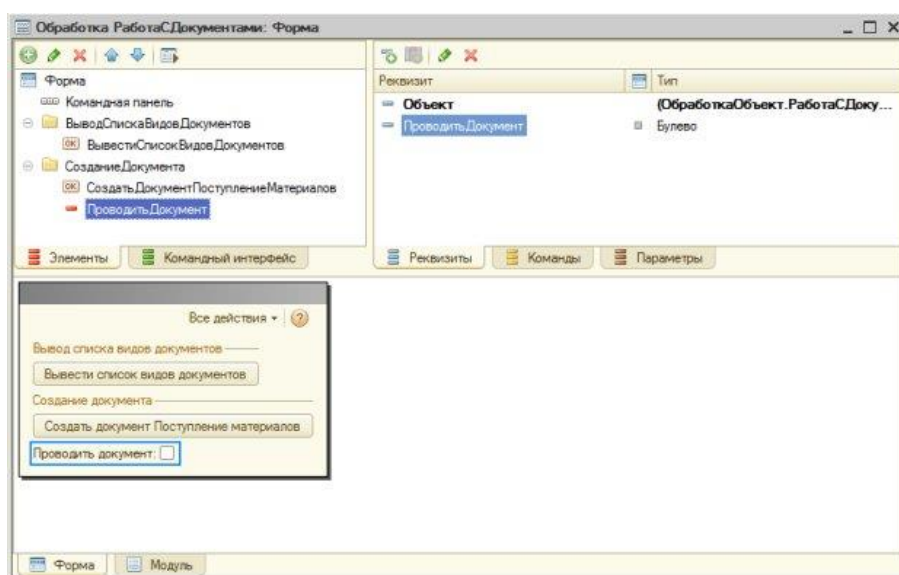


Рис. 7.12. Модификация формы обработки

Решить поставленную задачу можно с помощью следующего кода:

```

&НаКлиенте
Процедура СоздатьДокументПоступлениеМатериалов (Команда)
    //Настраиваем режим записи нового документа
    РежимЗаписи=РежимЗаписиДокумента.Запись;
    Если ПроводитьДокумент Тогда
        РежимЗаписи=РежимЗаписиДокумента.Проведение;
    КонецЕсли;
    //В функции будет создан новый документ
    //Она возвратит ссылку на него
    Документ=СоздатьДокумент (РежимЗаписи) ;
    //Открываем форму документа
    ОткрытьЗначение (Документ) ;
КонецПроцедуры

Функция СоздатьДокумент (РежимЗаписи)
    //Создаем новый документ
    Документ = Документы.ПоступлениеМатериалов.СоздатьДокумент () ;

```

```

//Заполняем его реквизиты
Документ.Дата=ТекущаяДата();

Документ.ОтветственныйСотрудник=Справочники.Сотрудники.НайтиПоКоду("000000003");

Документ.Контрагент=Справочники.Контрагенты.НайтиПоРеквизиту("КонтактныеСведения", "ул. Береговая, д. 2, телефон 3-34-34");
Документ.Комментарий="Документ создан автоматически";
//Заполняем строку табличной части
НоваяСтрокаТЧ=Документ.Материалы.Добавить();

НоваяСтрокаТЧ.Номенклатура=Справочники.Номенклатура.НайтиПоНаименованию("Духи");
НоваяСтрокаТЧ.Количество=10;
НоваяСтрокаТЧ.Цена=200;
НоваяСтрокаТЧ.Сумма=10*200;
//Записываем документ
Документ.Записать(РежимЗаписи);
//Возвращаем ссылку на документ
Возврат(Документ.Ссылка);
КонецФункции

```

Проведение документа, либо его запись, либо – отмена проведения – производятся с помощью метода документа **Записать()**. Этот метод принимает параметр, который и отвечает за то, что будет произведено при записи документа. В начале клиентской процедуры **СоздатьДокументПоступлениеМатериалов** мы анализируем флаг **ПроводитьДокумент**, предварительно установив переменную **РежимЗаписи** в значение **РежимЗаписиДокумента.Запись**. Если флаг установлен – то есть мы хотим не только записать, но и провести документ, значение переменной меняется на **РежимЗаписиДокумента.Проведение**.

При вызове функции **СоздатьДокумент()**, мы планируем воспользоваться возвращенной ей ссылкой на созданный документ для открытия его формы с помощью метода **ОткрытьЗначение()**.

Функция **СоздатьДокумент** принимает один параметр – **РежимЗаписи**, который мы используем после создания документа для задания режима его записи.

В переменной **Документ**, после работы метода **СоздатьДокумент()**, мы имеем объект типа **ДокументОбъект**, новый чистый документ, с которым можем работать. Мы заполняем дату с помощью функции **ТекущаяДата()**. Заполняя реквизиты мы пользуемся различными методами поиска данных в справочнике. Метод **НайтиПоКоду()** позволяет найти элемент справочника по заданному коду, **НайтиПоРеквизиту()** ищет по значению заданного реквизита. Оба эти метода возвращают нам ссылку на элемент справочника.

В реквизит **Комментарий** мы записываем строку.

Следующим этапом работы с документом становится заполнение его табличной части. Мы выполняем заполнение построчно, добавляя в табличную часть строку – она представлена в коде переменной **НоваяСтрокаТЧ** с типом **Строка табличной части**. Когда строка добавлена, мы обращаемся к ее реквизитам, заполняя их данными.

После того, как документ сформирован, мы записываем его (или – и записываем и проводим – в зависимости от значения параметра **РежимЗаписи**) и возвращаем ссылку на него в точку вызова функции **СоздатьДокумент**.

Вот, как выглядит документ, созданный программно с помощью нашего кода, рис. 7.13.

Поступление материалов 000000008 от 07.10.2011 16:27:28

Провести и закрыть Провести Все действия ?

Номер: 000000008

Дата: 07.10.2011 16:27:28

Контрагент: ООО "Полет" ...Q

Ответственный сотрудник: Иванов И. И. (Парикмахерская) ...Q

Комментарий: Документ создан автоматически

Организация: ...Q

Сумма (итог): 2 000,00

+ Добавить - Удалить + Добавить - Удалить Все действия

| N | Номенклатура | Цена | Количество | Сумма |
|---|--------------|--------|------------|----------|
| 1 | Духи | 200,00 | 10,000 | 2 000,00 |

Рис. 7.13. Документ, созданный автоматически

Решим теперь следующую задачу. Нужно пометить на удаление все документы типа **ПоступлениемМатериалов**, которые созданы автоматически – их реквизит **Комментарий** содержит текст "Документ создан автоматически".

Добавим в форму обработки новую команду, назовем ее **ПометитьНаУдаление**. Поставленную задачу можно реализовать с помощью следующего кода:

```
&НаКлиенте
Процедура ПометитьНаУдаление (Команда)
    ПометитьДокументыНаУдаление ();
    Предупреждение ("Были помечены на удаление документы поступления
материалов");
КонецПроцедуры

Процедура ПометитьДокументыНаУдаление ()
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        | ПоступлениеМатериалов.Ссылка
        | ИЗ
        | Документ.ПоступлениеМатериалов КАК ПоступлениеМатериалов
        | ГДЕ
        | ПоступлениеМатериалов.Комментарий = &Комментарий";

    Запрос.УстановитьПараметр ("Комментарий", "Документ создан автоматически");
```



```

Результат = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = Результат.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Документ=ВыборкаДетальныеЗаписи.Ссылка.ПолучитьОбъект();
    Документ.УстановитьПометкуУдаления(Истина);
КонецЦикла;
КонецПроцедуры

```

Здесь мы, в серверной процедуре **ПометитьДокументыНаУдаление()**, получаем с помощью запроса список ссылок на документы, реквизит **Комментарий** которых равен нужному нам значению. После этого в цикле обхода выборки запроса переходим от ссылки на объект к объекту (тип **ДокументОбъект**) и устанавливаем у объектов пометки удаления.

При завершении серверной процедуры, мы, на клиенте, показываем пользователю окно сообщения, рис. 7.14.

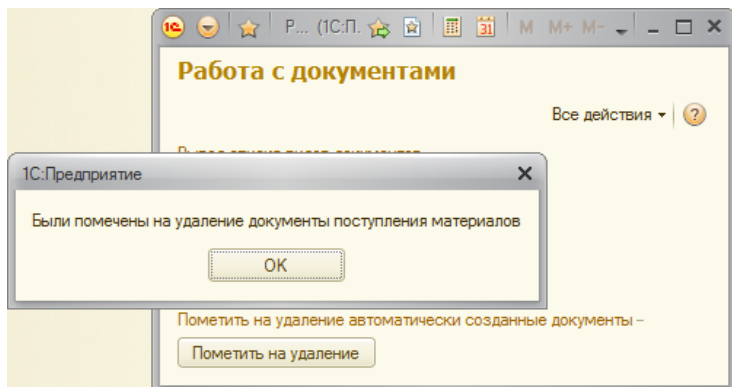


Рис. 7.14. Сообщение пользователю о пометке документов на удаление

Нашей следующей задачей будет вывод пользователю списка документов за заданный пользователем период. Добавим в форму обработки команду **ВыводСпискаДокументовЗаПериод** и два реквизита – **ДатаНачала** и **ДатаОкончания** – тип **Дата**, состав даты – **Дата и время**. Дата документа содержит сведения о дате и времени создания документа, поэтому для выбора периода, в который должны попасть искомые документы, нам понадобятся значения даты с датой и временем.

Решение задачи может выглядеть так:

```

&НаКлиенте
Процедура ВыводСпискаДокументовЗаПериод(Команда)
    Сообщить("Обнаружены следующие документы за период с "+ДатаНачала+" по "+ДатаОкончания);
    ВыводСписка();
КонецПроцедуры

Процедура ВыводСписка()
    Выборка=Документы.ПоступлениеМатериалов.Выбрать(ДатаНачала, ДатаОкончания);
    Пока Выборка.Следующий() Цикл

```

Сообщить (Выборка.Ссылка) ;
КонецЦикла
КонецПроцедуры

Здесь мы пользуемся методом **Выбрать** с параметрами, устанавливающими дату начала и дату окончания для выборки документов. Полученную выборку перебираем в цикле и сообщаем пользователю о найденных документах, рис. 7.15.

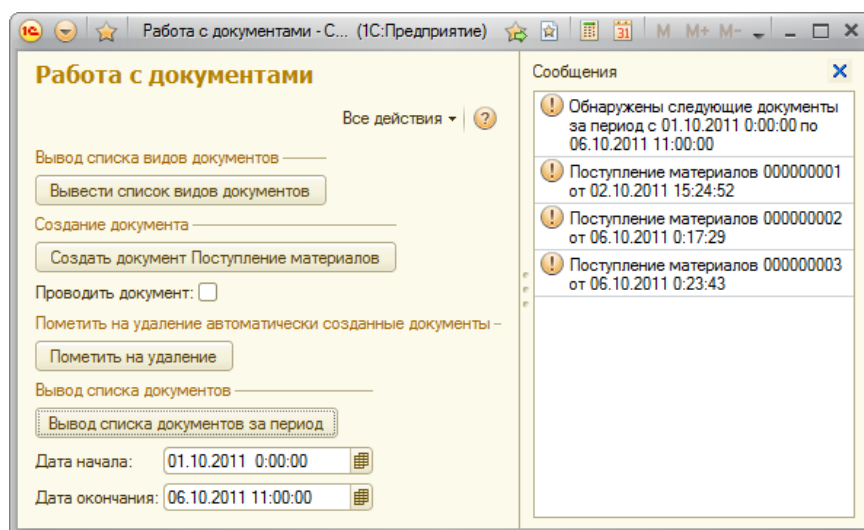


Рис. 7.15. Вывод списка документов, принадлежащих периоду, заданному пользователем