

## **Занятие № 6**

### *Регистры накопления.*

На этом занятии мы познакомимся с объектом конфигурации Регистр накопления. Вы узнаете, для чего используется этот объект, какой структурой он обладает и каковы его отличительные особенности.

Затем мы создадим с вами один из регистров накопления, который будет использоваться в нашей конфигурации и отражать изменение данных в процессе работы ранее созданных нами документов.

### **Зачем нужен регистр накопления**

Итак, мы с вами подошли к одному из главных моментов разработки любой конфигурации – созданию механизма учета накопления данных.

Казалось бы, все необходимое мы с вами уже создали: у нас есть что расходовать и приходовать (справочники), и у нас есть чем расходовать и приходовать (документы). Осталось только построить несколько отчетов, и автоматизация ООО «На все руки мастер» будет закончена.

Однако это не так.

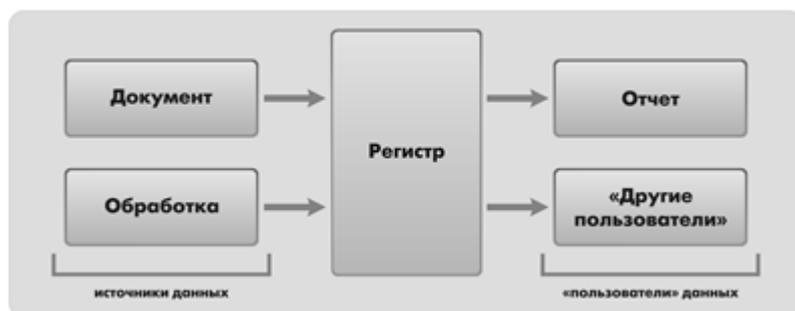
*Во-первых*, путем анализа документов можно, конечно, получить требуемые нам выходные данные. Но представьте, что завтра ООО «На все руки мастер» решит немного изменить свои бизнес-процессы, и нам потребуется ввести в конфигурацию еще один документ (или несколько документов).

Например, сейчас мы полагаем, что товары поступают в ООО «На все руки мастер» и затем расходуются. Руководство захотело усилить материальный контроль и решило приходовать товары на основной склад организации и затем выдавать их материально ответственными лицам. В этом случае нам придется добавить в конфигурацию еще один документ, который будет фиксировать перемещение материалов между основным складом и материально ответственными лицами. И очевидно, нам придется

переработать все отчеты, которые были нами созданы к этому моменту, с тем, чтобы они учитывали изменения, вносимые новым документом. А представьте, если в нашей конфигурации не два, а двадцать документов?!

*Во-вторых*, отчеты, анализирующие документы, будут работать довольно медленно, что будет вызывать раздражение пользователей и недовольство руководителей.

Поэтому в системе «1С:Предприятие» есть несколько объектов конфигурации, которые позволяют создавать в базе данных структуры, предназначенные для накопления информации в удобном для последующего анализа виде. Использование таких хранилищ данных позволяет нам, с одной стороны, накапливать в них данные, поставляемые различными документами (или другими объектами базы данных), а с другой стороны, легко создавать нужные нам отчеты или использовать эти данные в алгоритмах работы конфигурации (рис. 6.1).



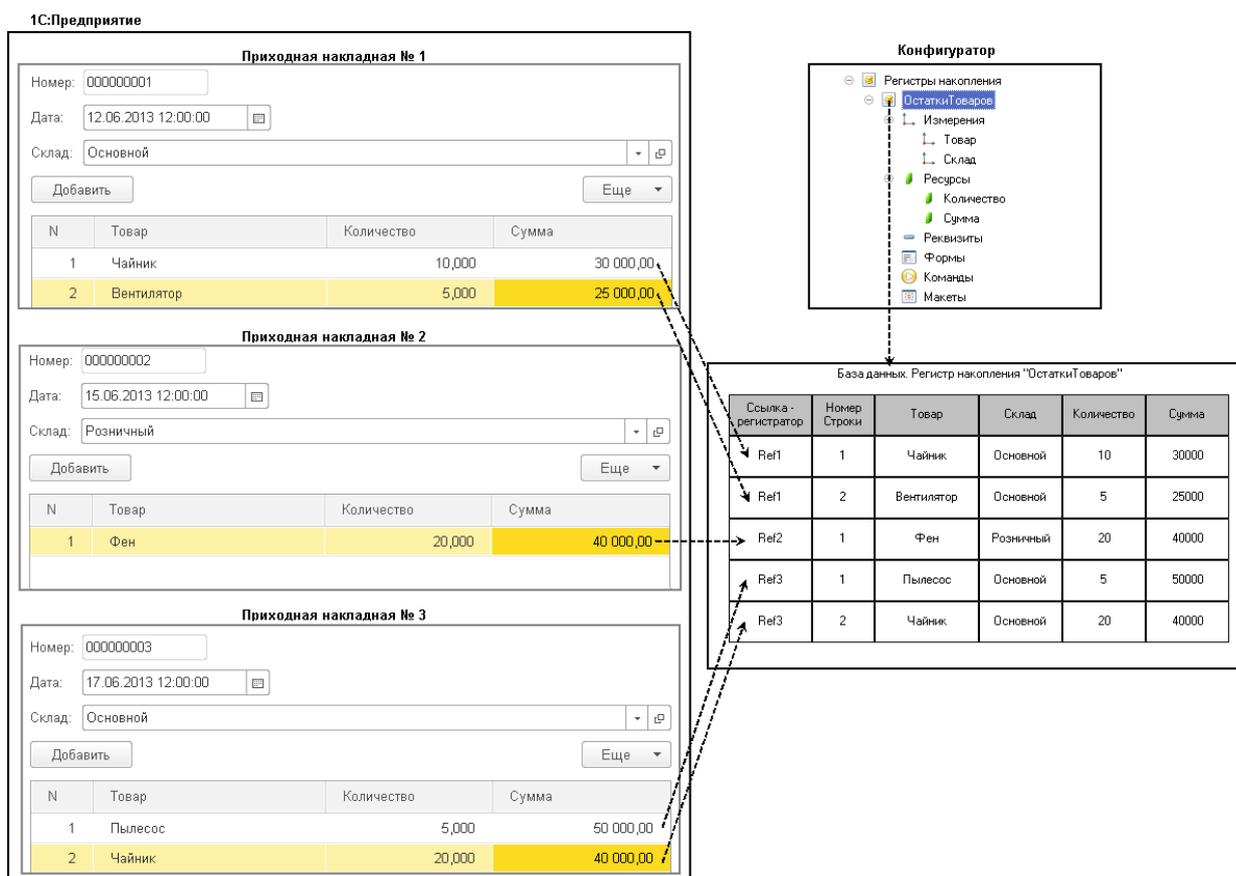
**Рис. 6.1.** Алгоритм работы конфигурации

В конфигурации существует несколько объектов, называемых *регистрами*, для описания подобных хранилищ. Сейчас мы рассмотрим один из них.

### **Что такое регистр накопления**

Объект конфигурации *Регистр накопления* предназначен для описания структуры накопления данных. На основе объекта конфигурации *Регистр накопления* платформа создает в базе данных таблицы, в которых будут накапливаться данные, поставляемые различными объектами базы данных.

Эти данные будут храниться в таблицах в виде отдельных записей, каждая из которых имеет одинаковую заданную в конфигураторе структуру (рис. 6.2).



**Рис. 6.2.** Регистр накопления «Остатки товаров» в конфигураторе и в базе данных

На основании таблицы движений регистра накопления система рассчитывает таблицу итогов регистра, которая хранит в базе данных итоги на момент времени последнего движения (актуальные итоги).

Отличительной особенностью регистра накопления является то, что он не предназначен для интерактивного редактирования пользователем.

Разработчик может при необходимости предоставить пользователю возможность редактировать регистр накопления. Но предназначение регистра накопления заключается в том, чтобы его модификация производилась на основе алгоритмов работы других объектов базы данных, а не в результате непосредственных действий пользователя.

Основным назначением регистра накопления является накопление числовой информации в разрезе нескольких *измерений*, которые описываются разработчиком в соответствующем объекте конфигурации *Регистр накопления* и являются подчиненными объектами конфигурации.

Виды числовой информации, накапливаемой регистром накопления, называются *ресурсами*, также являются подчиненными объектами и описываются в конфигураторе.

Например, регистр накопления может накапливать информацию о количестве и сумме товаров на складах. В этом случае он будет иметь измерения *Товар* и *Склад* и ресурсы *Количество* и *Сумма* (см. рис. 6.2).

Изменение состояния регистра накопления происходит, как правило, при проведении документа и заключается в том, что в регистр добавляется некоторое количество записей. Каждая запись содержит значения измерений, значения приращений ресурсов, ссылку на документ, который вызвал эти изменения (регистратор), и «направление» приращения (приход или расход). Такой набор записей называется *движениями* регистра накопления. Каждому движению регистра накопления всегда должен соответствовать *регистратор* – объект информационной базы (как правило, документ), который произвел эти движения.

Кроме этого, регистр накопления может хранить дополнительную информацию, описывающую каждое движение. Набор такой дополнительной информации задается разработчиком при помощи *реквизитов* объекта конфигурации *Регистр накопления*.

### **Добавление регистра накопления**

Теперь, когда мы знаем, для чего предназначены регистры накопления, посмотрим, как можно их использовать в нашем примере.

Прежде всего, нас интересует информация о том, сколько и каких материалов есть у нас на складах. Для накопления такой информации мы создадим регистр *ОстаткиМатериалов*.

### **В режиме «Конфигуратор»**

Откроем в конфигураторе нашу учебную конфигурацию и добавим новый объект конфигурации *Регистр накопления*.

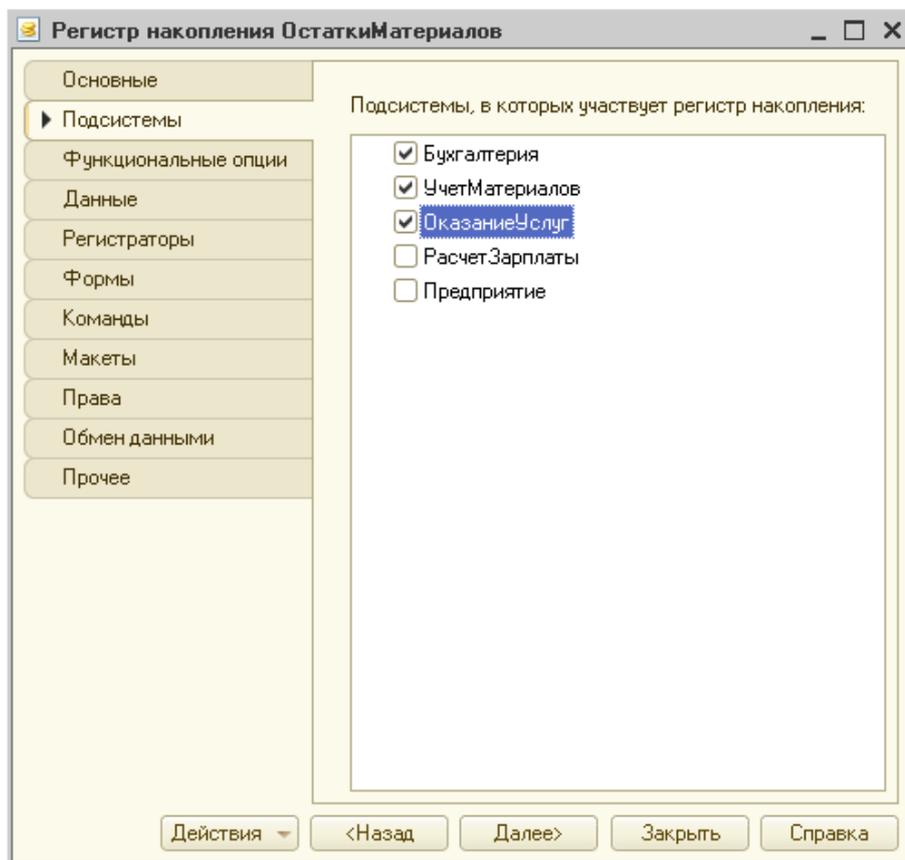
Для этого выделим в дереве объектов конфигурации ветвь *Регистры накопления* и нажмем кнопку *Добавить* в командной панели окна конфигурации.

В открывшемся окне редактирования объекта конфигурации на закладке *Основные* зададим имя регистра – *ОстаткиМатериалов*.

Также зададим и *Расширенное представление списка* как *Движения по регистру Остатки материалов*. Этот заголовок будет отображаться в окне списка записей регистра.

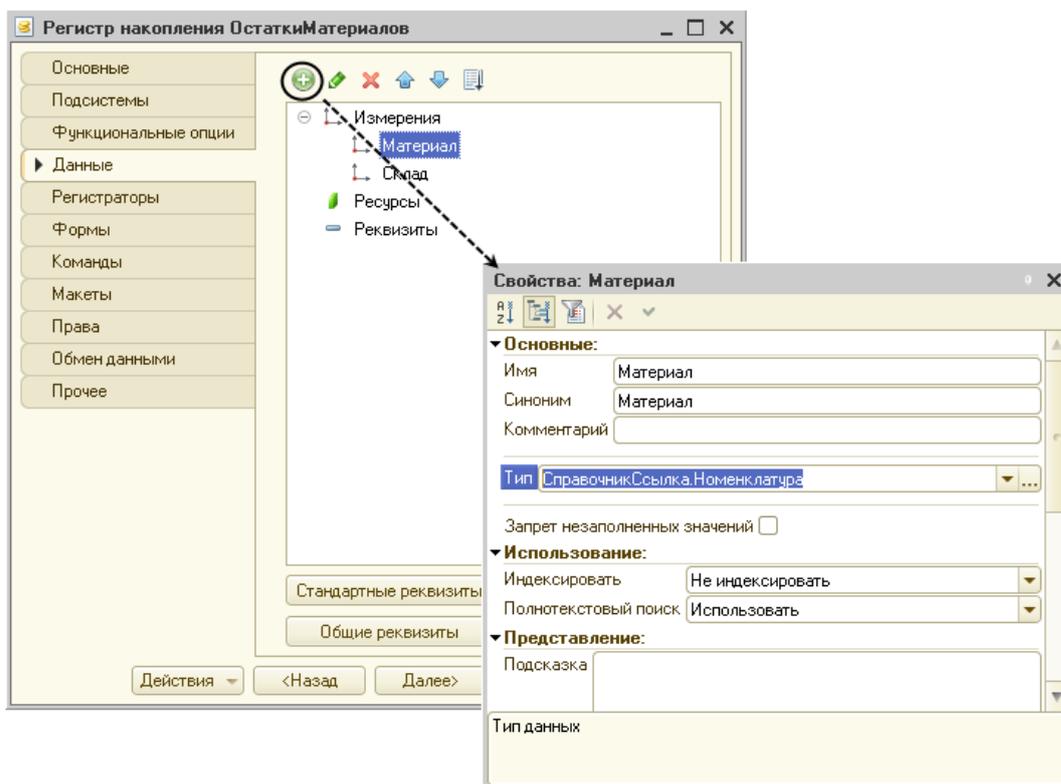
Нажмем *Далее* и перейдем на закладку *Подсистемы*.

По логике нашей конфигурации данный регистр должен быть доступен в разделах *Учет материалов*, *Оказание услуг* и *Бухгалтерия*. Поэтому отметим в списке подсистем эти подсистемы (рис. 6.3).



**Рис. 6.3.** Определение списка подсистем, в которых будет отражаться регистр

Выделим закладку *Данные* и перейдем к созданию структуры регистра.



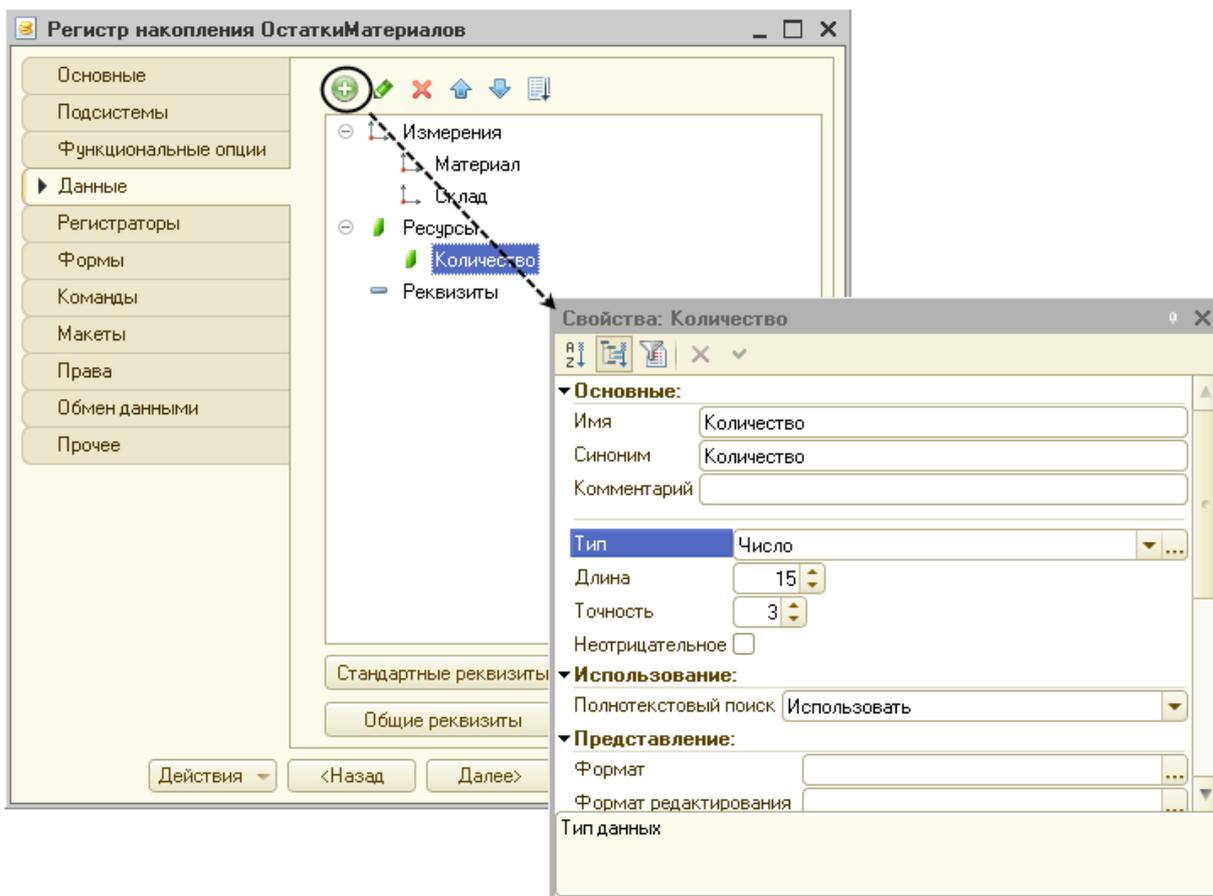
**Рис. 6.4.** Создание измерений регистра

Создадим измерения регистра:

- *Материал*, тип *СправочникСсылка.Номенклатура*;
- *Склад*, тип *СправочникСсылка.Склады*.

Для этого выделим ветвь *Измерения* и нажмем кнопку *Добавить* в командной панели окна (рис. 6.4).

Затем создадим ресурс *Количество* с длиной 15 и точностью 3. Для этого выделим ветвь *Ресурсы* и нажмем кнопку *Добавить* в командной панели окна (рис. 6.5).

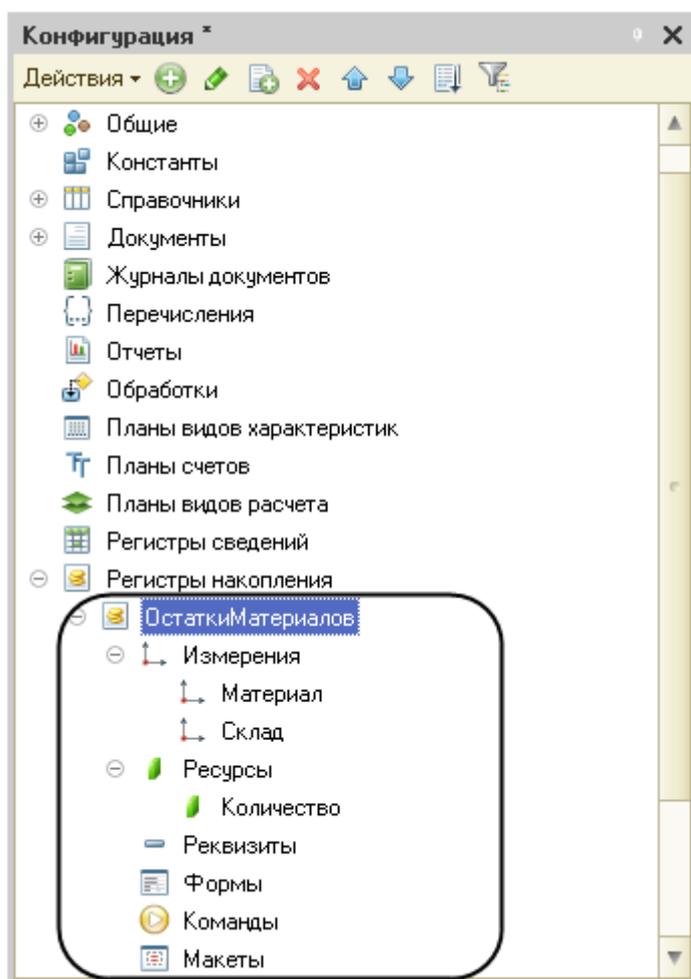


**Рис. 6.5.** Создание ресурсов регистра

В результате этих действий регистр *ОстаткиМатериалов* должен иметь следующий вид (рис. 6.6).

Если вы сейчас попытаетесь запустить «1С:Предприятие» в режиме отладки, то система выдаст сообщение об ошибке: «РегистрНакопления.ОстаткиМатериалов: Ни один из документов не является регистратором для регистра». Это сообщение еще раз подтверждает

тот факт, что назначение регистра накопления в том, чтобы аккумулировать данные, поставляемые различными документами.



**Рис. 6.6.** Регистр «ОстаткиМатериалов»

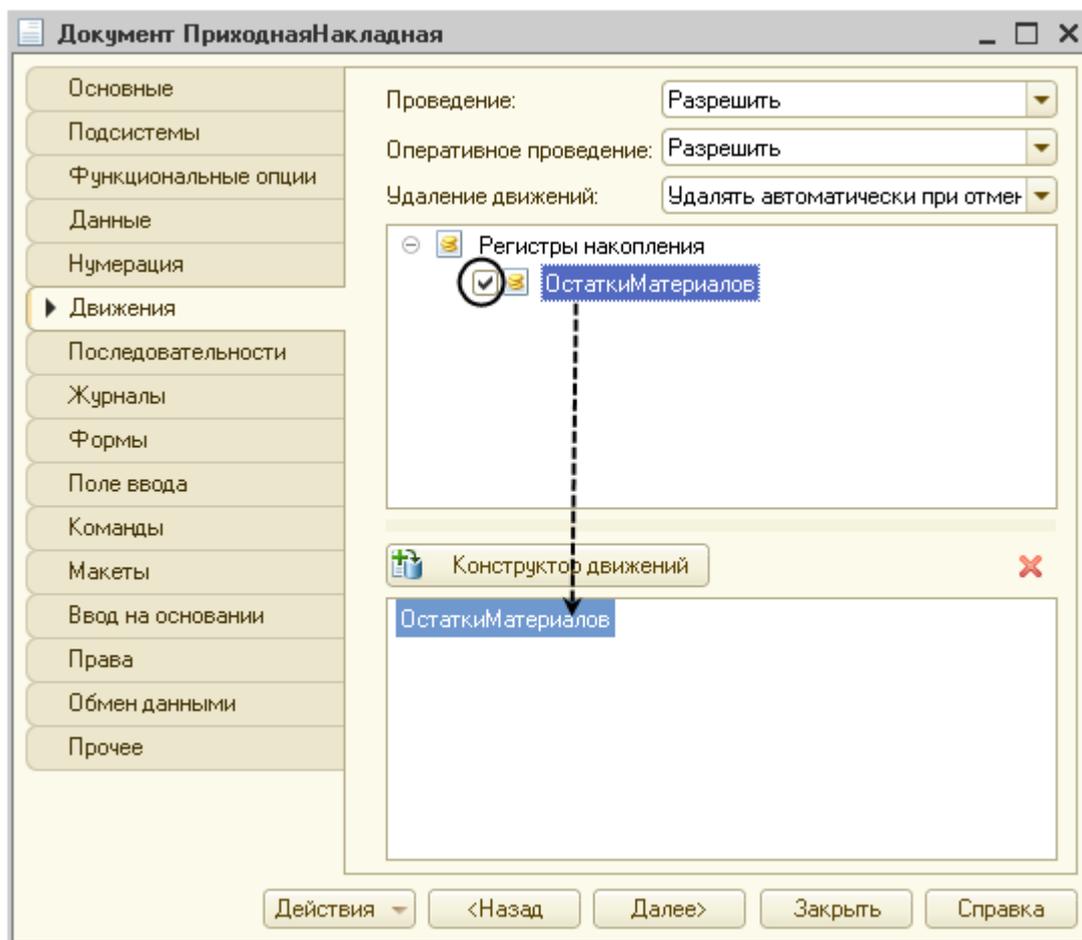
Поэтому мы сформируем движения регистра накопления *ОстаткиМатериалов* в процессе проведения двух созданных нами документов *ПриходнаяНакладная* и *ОказаниеУслуг*.

### **Движения документа**

*Движения документа* – это записи в регистрах, которые создаются в процессе проведения документа и отражают изменения, производимые документом.

Откроем окно редактирования объекта конфигурации *Документ ПриходнаяНакладная*. Перейдем на

закладку *Движения*, раскроем список *Регистры накопления* и отметим регистр накопления *ОстаткиМатериалов* (рис. 6.7).



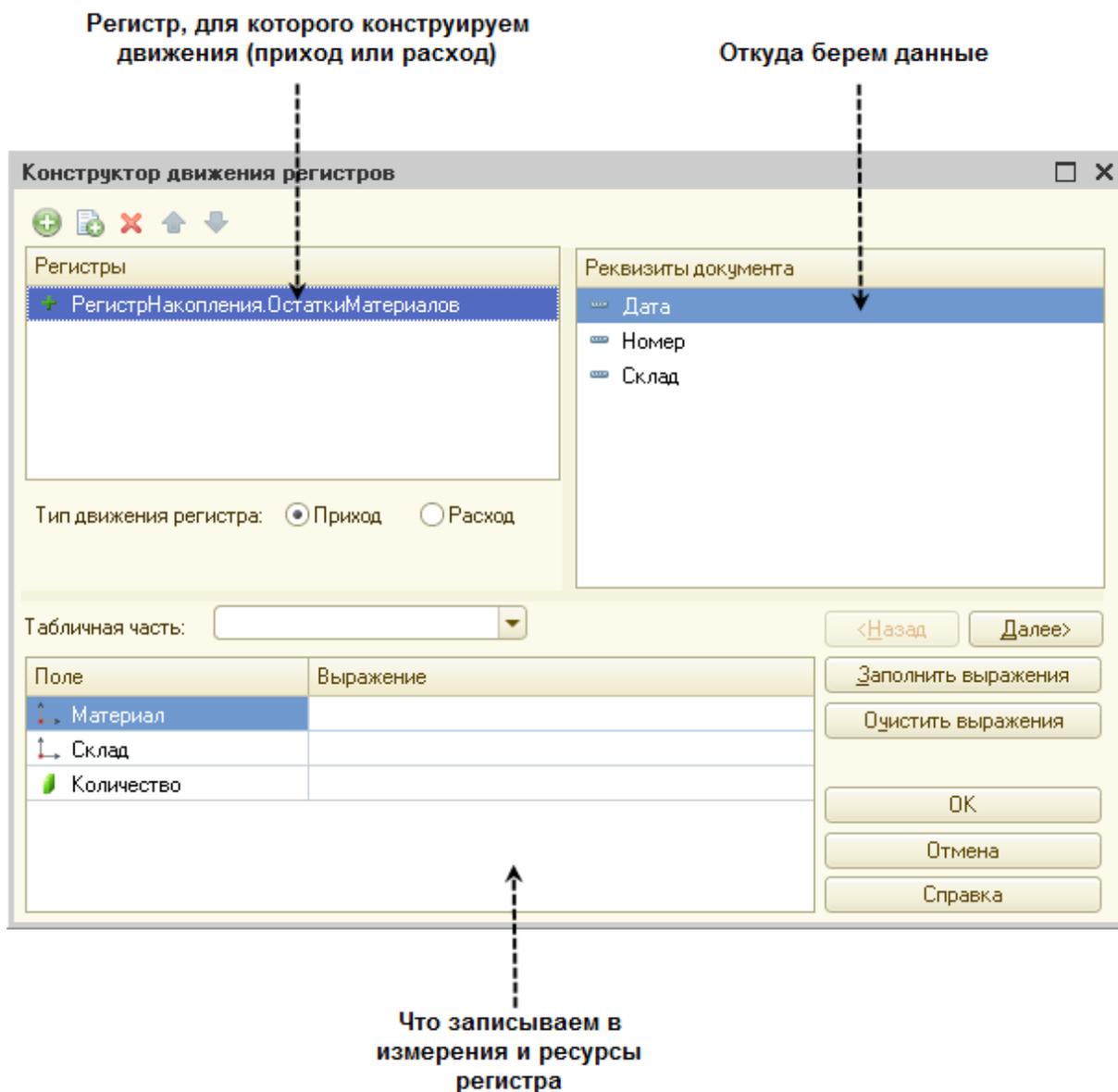
**Рис. 6.7.** Создание движений документа «ПриходнаяНакладная» в регистре «ОстаткиМатериалов»

Обратите внимание, что сразу после отметки выбранного регистра становится доступной кнопка *Конструктор движений*. Нажмем ее и воспользуемся этим конструктором.

Конструктор устроен просто. В списке *Регистры* перечислены регистры, в которых документ может создавать движения. В нашем случае там пока один регистр *ОстаткиМатериалов*.

В списке *Реквизиты документа* должны находиться исходные данные для создания движений – реквизиты документа *ПриходнаяНакладная*.

А в таблице *Поле – Выражение* должны быть заданы формулы, по которым будут вычисляться значения измерений и ресурсов регистра при записи движений (рис. 6.8).



**Рис. 6.8.** Конструктор движений

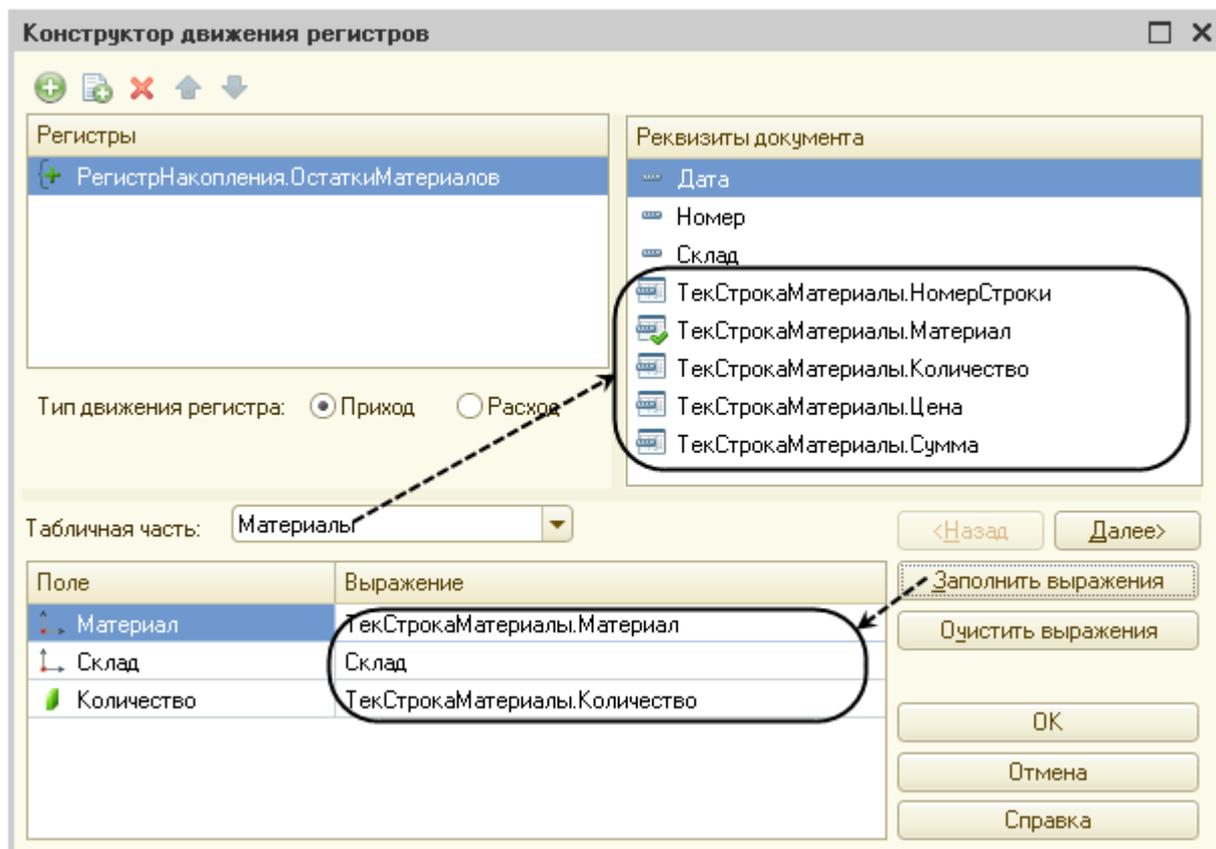
Обратите внимание, что по умолчанию конструктор предлагает нам создавать движения прихода (*Тип движения регистра* – *Приход*, символ + рядом с названием регистра) по регистру *ОстаткиМатериалов*. Это нас вполне устраивает, ведь документ *ПриходнаяНакладная* и должен приходить материалы.

В поле выбора *Табличная часть* выберем табличную часть нашего документа – *Материалы*.

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

Теперь нажмем кнопку *Заполнить выражения*.

В нижнем окне сформируется соответствие полей (измерений и ресурсов) регистра и выражений для их расчета (рис. 6.9).



**Рис. 6.9.** Выбор табличной части документа и заполнение выражений для расчета движений регистра

Как видите, конструктор движений установил соответствия подходящим образом: в качестве материала в регистр будет записан материал из табличной части документа, в качестве склада – склад, указанный в шапке документа, а в качестве количества – количество из табличной части документа.

Нажмем кнопку *OK* и посмотрим, какой текст сформировал конструктор в модуле документа *ПриходнаяНакладная* (листинг 6.1).

## Листинг 6.1. Процедура «ОбработкаПроведения()»

```
Процедура ОбработкаПроведения(Отказ, Режим)
  //{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора внесенные вручную
изменения будут утеряны!!!
  // регистр ОстаткиМатериалов Приход
Движения.ОстаткиМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
  Движение = Движения.ОстаткиМатериалов.Добавить ();
  Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
  Движение.Период = Дата;
  Движение.Материал = ТекСтрокаМатериалы.Материал;
  Движение.Склад = Склад;
  Движение.Количество = ТекСтрокаМатериалы.Количество;
КонецЦикла;
  //}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

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

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

Поясним текст процедуры обработчика.

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

К конкретному набору записей этой коллекции можно обратиться, указав через точку имя регистра, которому принадлежит этот набор записей. Например, *Движения.ОстаткиМатериалов*.

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

В первой строке процедуры мы 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.

Внутри обработчика расположен цикл *Для Каждого ... Из ... Цикл*. Он предназначен для перебора строк табличной части нашего документа.

В цикле обращение к табличной части документа происходит по имени (*Материалы*). Переменная *ТекСтрокаМатериалы* содержит объект с данными текущей строки табличной части документа. Эта переменная создается в начале цикла и меняется по мере его прохождения.

В первой строке тела цикла, используя метод *Добавить()*, мы добавляем к набору записей, который создает наш документ в регистре, новую запись. Тем самым мы создаем объект *РегистрНакопленияЗапись* и сохраняем его в переменной *Движение*.

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

Причем *Движение.Материал*, *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, мы обращаемся к полям табличной части, указав

имя поля через точку от переменной *ТекСтрокаМатериалы* (например, *ТекСтрокаМатериалы.Материал*).

Заметим, что *Склад* – это реквизит шапки документа, а *Дата* – стандартный реквизит документа, который создается автоматически. Причем в цикле меняются только значения полей табличной части документа – *ТекСтрокаМатериалы.Материал* и *ТекСтрокаМатериалы.Количество*. Остальные поля не меняются, так как относятся к документу в целом и не зависят от текущей строки табличной части документа.

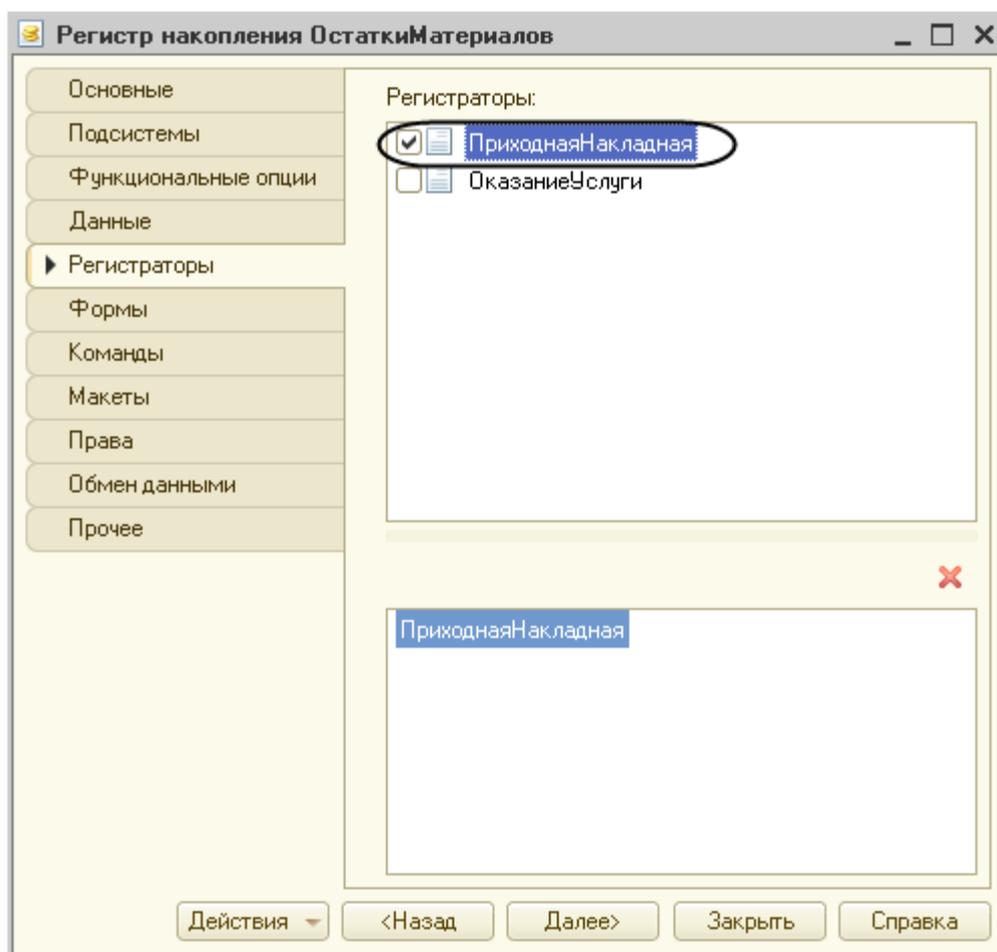
*ВидДвиженияНакопления.Приход* – это значение системного перечисления, которое определяет вид движения регистра накопления как *Приход*.

Таким образом, мы присваиваем нужные значения всем полям новой записи. После перебора всех строк документа (после завершения цикла) в этом наборе записей (*Движения.ОстаткиМатериалов*) будет содержаться столько записей, сколько строк в табличной части проводимого документа.

Если мы теперь откроем окно редактирования объекта конфигурации *Регистр накопления ОстаткиМатериалов* и перейдем на закладку *Регистраторы*, то в списке документов, созданных в конфигурации, мы увидим отмеченный документ *ПриходнаяНакладная*, так как мы задали в модуле этого документа формирование движений в регистре *ОстаткиМатериалов* (рис. 6.10).

В заключение отредактируем командный интерфейс, чтобы в разделах *Бухгалтерия*, *Оказание услуг* и *Учет материалов* была доступна ссылка для просмотра записей нашего регистра накопления.

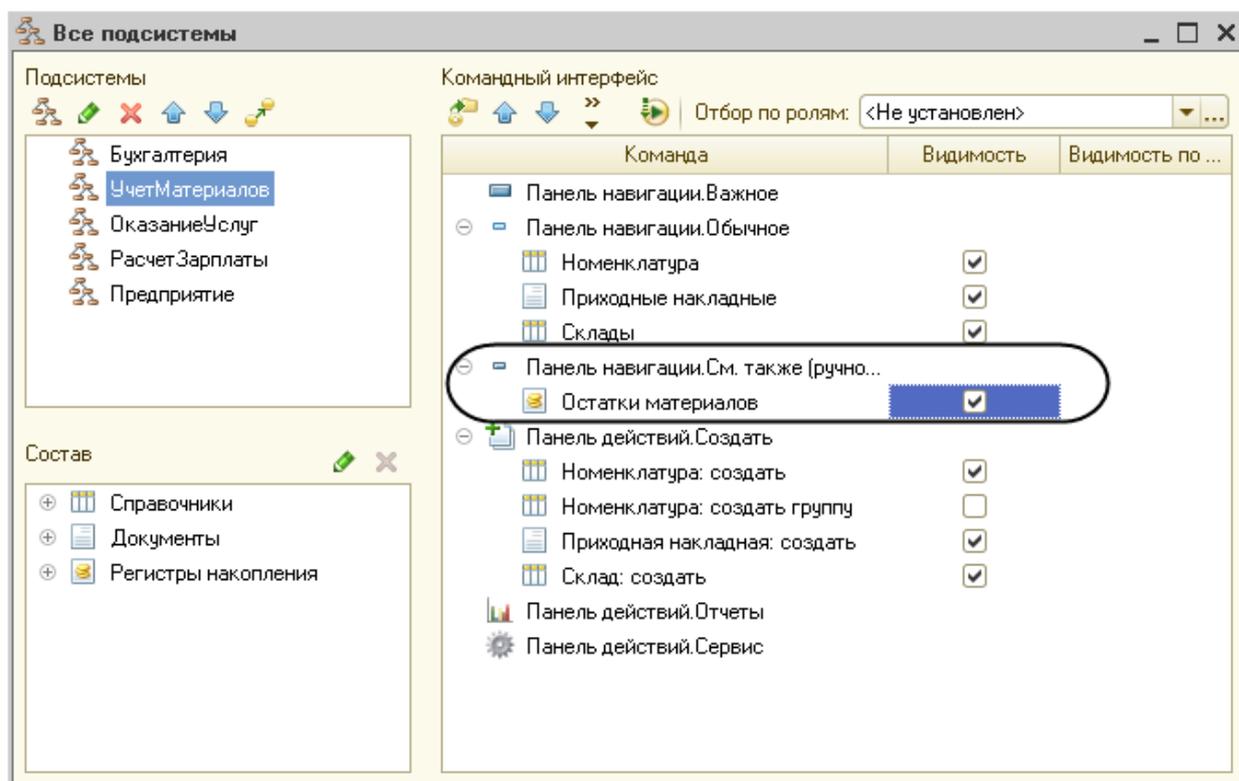
Дело в том, что команды открытия регистров также добавляются в панель команд разделов, но по умолчанию они невидимы, в отличие от команд открытия справочников и документов.



**Рис. 6.10.** Регистраторы регистра «ОстаткиМатериалов»

В дереве объектов конфигурации выделим ветвь *Подсистемы*, вызовем ее контекстное меню и выберем пункт *Все подсистемы*. В открывшемся окне слева в списке *Подсистемы* выделим подсистему *УчетМатериалов*. Справа в списке *Командный интерфейс* отразятся все команды выбранной подсистемы.

В группе *Панель навигации.Обычное* включим видимость у команды *Остатки материалов* и мышью перетащим ее в группу *Панель навигации.См.также* (рис. 6.11).



**Рис. 6.11.** Настройка командного интерфейса подсистем

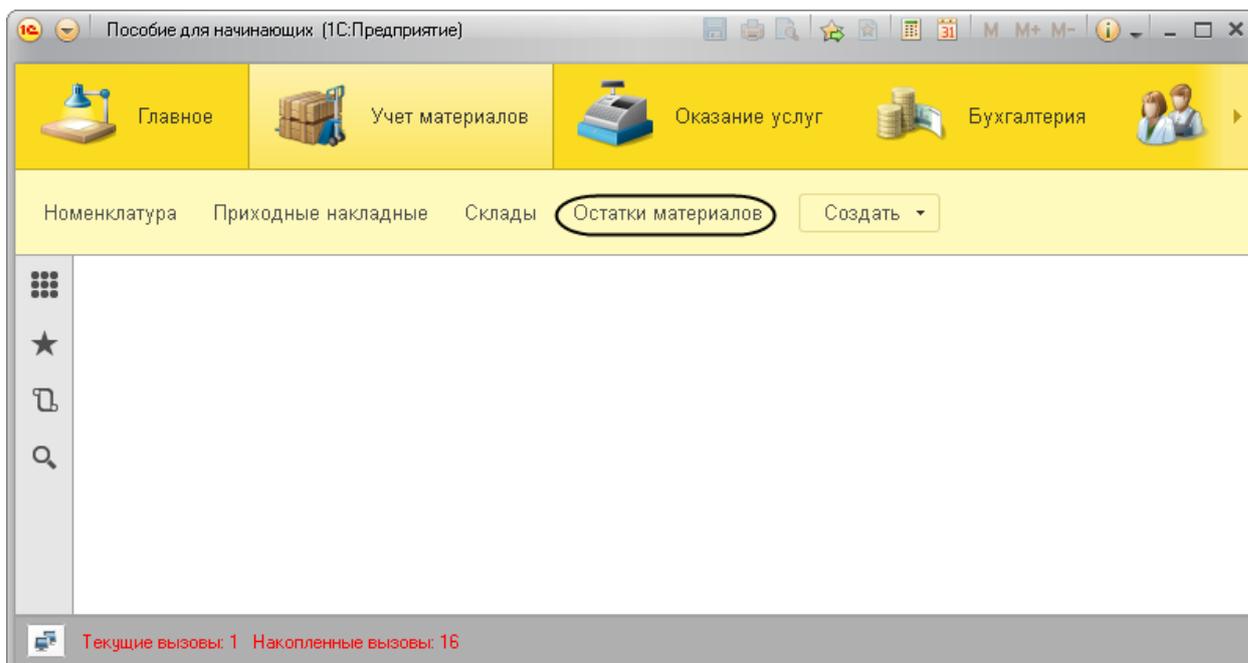
Тем самым мы переносим команды открытия списка регистров накопления в конец списка подобных команд раздела, так как эти команды не так часто используются и их приоритет невысок.

Аналогично, выделив подсистемы *Оказание Услуг* и *Бухгалтерия*, в группе *Панель навигации.Обычное* включим видимость у команды *Остатки материалов* и перенесем ее в группу *Панель навигации См.также*.

### **В режиме «1С:Предприятие»**

Запустим «1С:Предприятие» в режиме отладки и протестируем внесенные нами изменения.

В открывшемся окне «1С:Предприятия» мы видим, что в панели команд разделов *Бухгалтерия*, *Оказание услуг* и *Учет материалов* в конце списка команд для открытия различных списков, перед подменю *Создать*, *Отчеты* (если они есть) появилась команда для открытия списка регистра *Остатки материалов* (рис. 6.12).

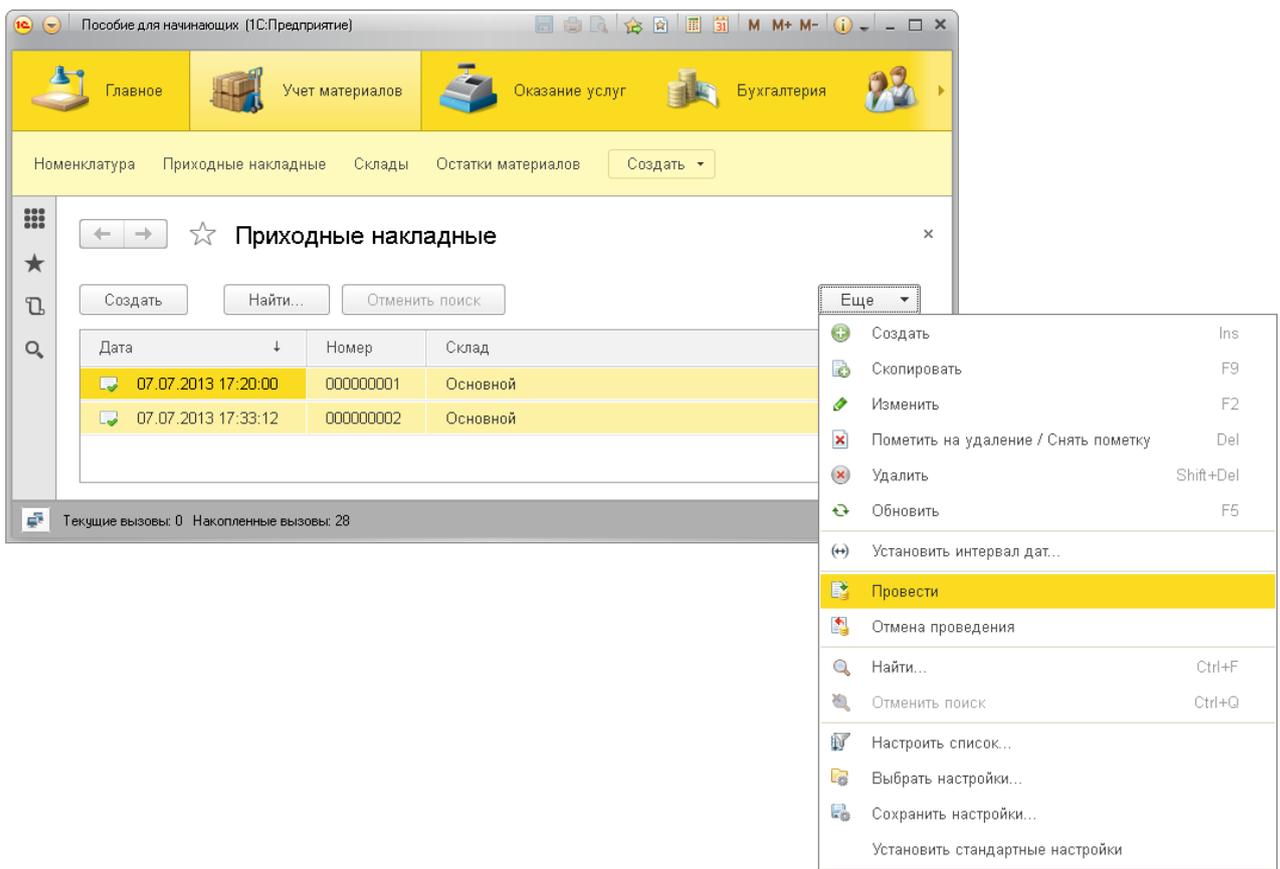


**Рис. 6.12.** Список регистра накопления «Остатки Материалов»

Чтобы проследить связь между проведением документа и накоплением информации в регистре, откроем список приходных накладных, выполнив команду *Приходные накладные* в разделе *Учет материалов*.

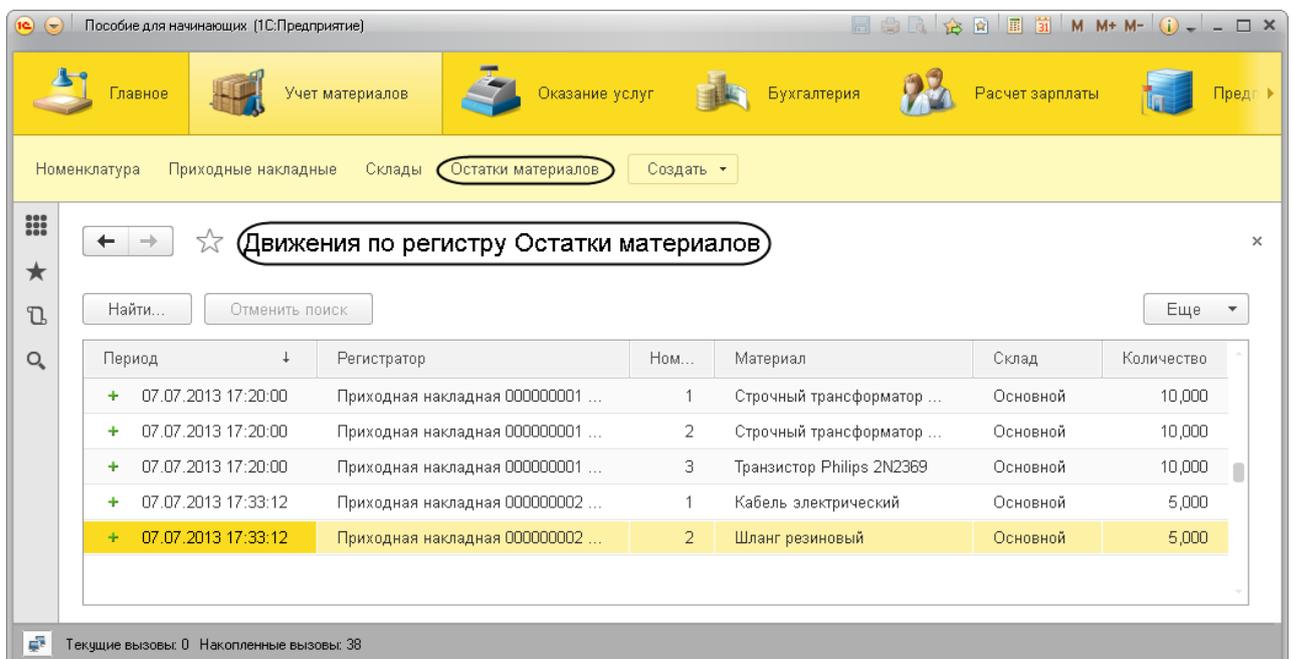
Откроем *Приходную накладную № 1* и нажмем *Провести и закрыть*, то есть перепроведем ее. То же самое сделаем для *Приходной накладной № 2*.

Перепровести документы можно и не открывая документов. Для этого следует выделить нужный документ в списке (или выделить мышью группу документов, удерживая клавишу *Ctrl*), раскрыть подменю *Еще* в командной панели формы списка и выбрать пункт *Провести* (рис. 6.13).



**Рис. 6.13.** Проведение документа

Теперь выполним команду *Остатки материалов* и откроем список нашего регистра накопления (рис. 6.14).



**Рис. 6.14.** Список регистра накопления «ОстаткиМатериалов»

Мы видим, что при проведении приходных накладных появляются соответствующие записи в регистре накопления *Остатки материалов*. Обратите внимание, что добавилось пять записей – первые три после проведения первого документа, что соответствует количеству строк в его табличной части, и последние две после проведения второго документа.

Все поля регистра заполнились данными документов так, как мы задали в обработчике проведения документа *ПриходнаяНакладная*. Пиктограмма со знаком + слева от каждой записи указывает на тип движения – *Приход*.

Как мы видим, заголовок формы списка записей регистра соответствует заданному нами в свойстве *Расширенное представление списка* для этого регистра.

### **Команда перехода к движениям в форме документа**

#### **В режиме «Конфигуратор»**

При реальной работе записей в регистре *ОстаткиМатериалов* будет много, и будет трудно понять, какие записи относятся к определенному документу.

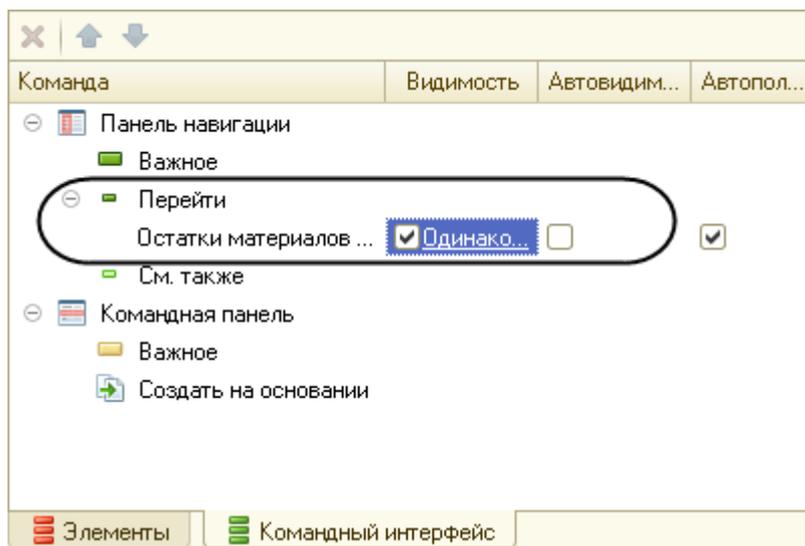
Поэтому наряду с общим списком регистра хотелось бы иметь возможность вызывать из формы документа список регистра, в котором показаны движения, произведенные только этим документом.

Чтобы реализовать такую возможность, вернемся в конфигуратор и откроем форму документа *ПриходнаяНакладная*.

В левом верхнем окне перейдем на закладку *Командный интерфейс*.

В разделе *Панель навигации* раскроем группу *Перейти* и увидим команду для открытия списка регистра накопления *Остатки материалов*. Эта команда была автоматически помещена в панель навигации формы документа, так как он является регистратором, то есть создает движения в нашем регистре.

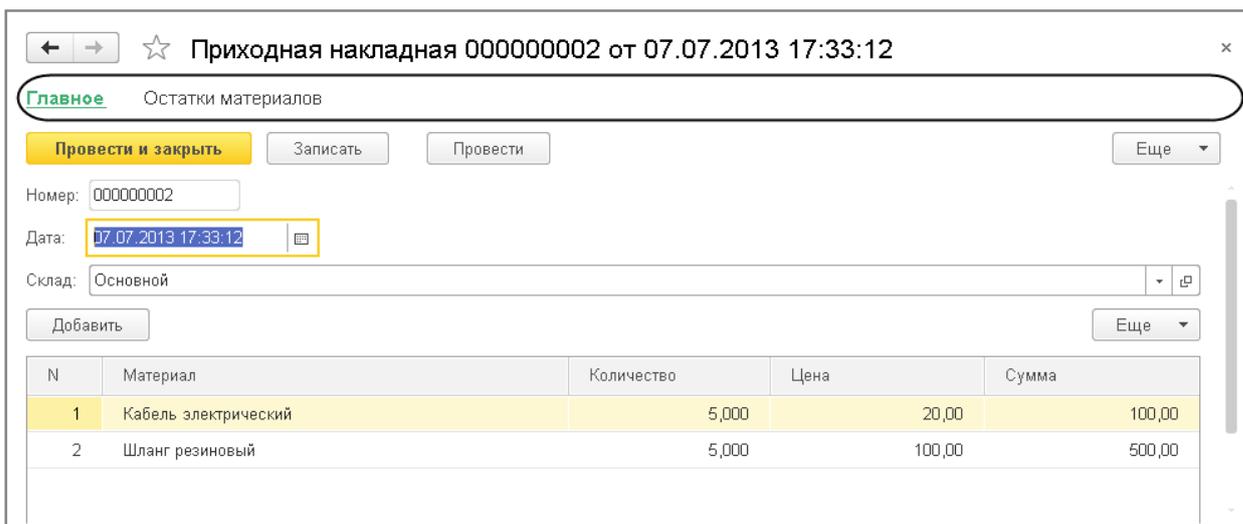
Установим свойство *Видимость* для этой команды (рис. 6.15).



**Рис. 6.15.** Настройка командного интерфейса формы документа

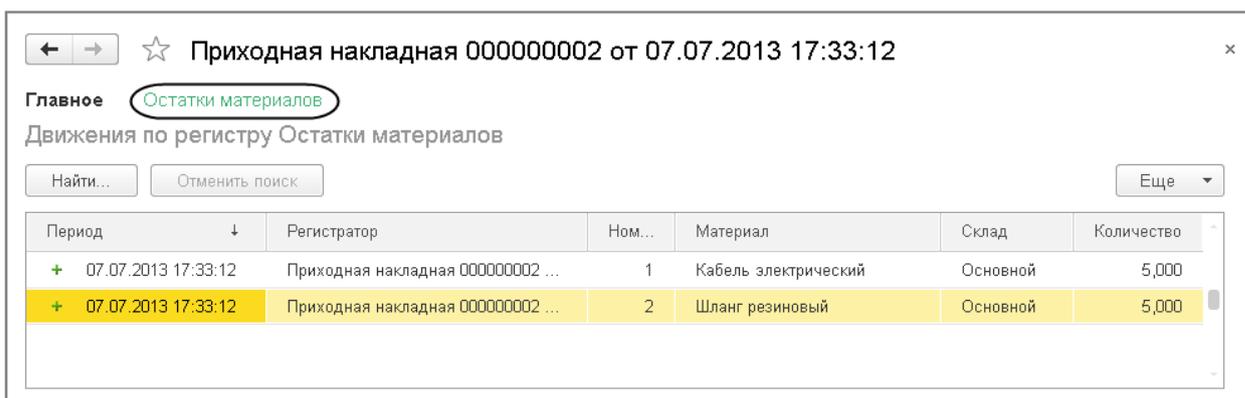
### В режиме «1С:Предприятие»

Запустим «1С:Предприятие» в режиме отладки и откроем *Приходную накладную № 2* (рис. 6.16).



**Рис. 6.16.** Панель навигации формы документа «Приходная накладная»

Под заголовком формы документа появилась панель навигации, в которой мы можем переходить к списку записей регистра *Остатки Материалов*, связанному с документом (рис. 6.17), и обратно к содержимому документа (ссылка *Главное*).



**Рис. 6.17.** Переход к регистру накопления из формы документа

Обратите внимание, что до этого панель навигации в форме приходной накладной была не видна, так как в ней не было отображено ни одной команды.

### Движения документа «Оказание услуги»

Теперь мы аналогичным образом создадим движения документа *ОказаниеУслуги*. Для этого потребуется выполнить уже знакомые нам действия.

### В режиме «Конфигуратор»

Откроем окно редактирования объекта конфигурации *Документ ОказаниеУслуги*.

Перейдем на закладку *Движения* и в списке регистров конфигурации отметим регистр накопления *ОстаткиМатериалов*.

Нажмем кнопку *Конструктор движений*.

В открывшемся окне конструктора изменим тип движения регистра на *Расход*, так как документ *ОказаниеУслуги* должен расходовать материалы. Пиктограмма слева от названия регистра изменится на знак —. В поле выбора *Табличная часть* выберем табличную часть нашего документа — *ПереченьНоменклатуры*.

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

Теперь нажмем кнопку *Заполнить выражения*.

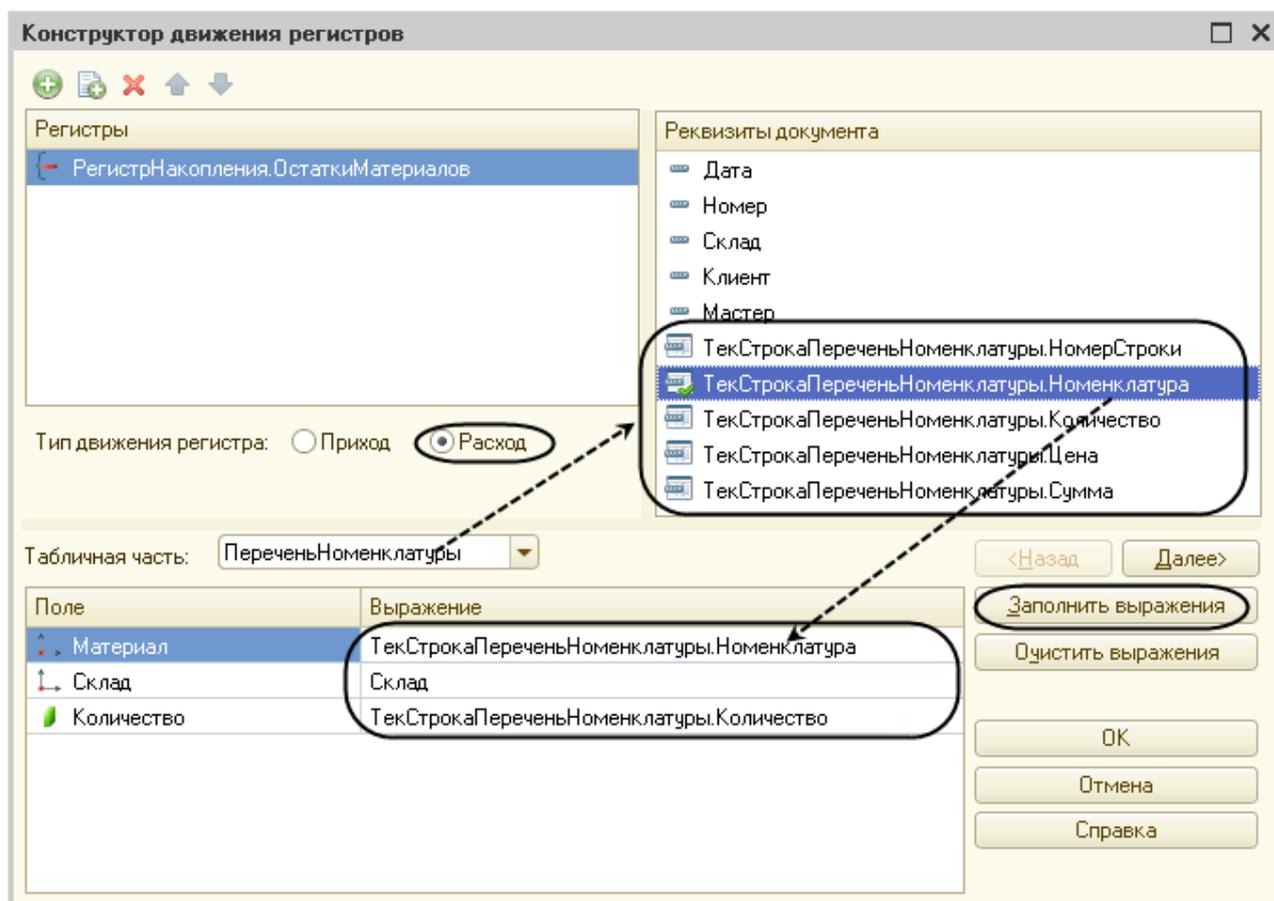
В нижнем окне сформируется соответствие полей (измерений и ресурсов) регистра и выражений для их расчета. Однако при автоматическом заполнении поле *Материал* не заполнится.

Так происходит потому, что имя поля табличной части *Номенклатура* не совпадает с именем измерения регистра – *Материал*. Если мы оставим это так, как есть, то в регистре накопления в строках с типом *Движение регистра – расход* номенклатура фиксироваться не будет.

Чтобы избежать этого, нужно выделить поле регистра *Материал* и в окне *Реквизиты документа* дважды щелкнуть по строке *ТекСтрокаПереченьНоменклатуры.Номенклатура*.

Таким образом, номенклатура для движений регистра накопления будет выбираться из табличной части документа (рис. 6.18).

Нажмем кнопку *ОК*.



**Рис. 6.18.** Выбор табличной части документа и заполнение выражений для расчета движений регистра

Конструктор создал обработчик события *ОбработкаПроведения* объекта конфигурации *Документ ОказаниеУслуги* и поместил его в модуль объекта (листинг 6.2).

### Листинг 6.2. Процедура «ОбработкаПроведения()»

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{{ __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора внесенные вручную
изменения будут утеряны!!!
// регистр ОстаткиМатериалов Расход
Движения.ОстаткиМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Движение = Движения.ОстаткиМатериалов.Добавить ();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
КонецЦикла;
//}} __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

Обратите внимание, что строка *Движение.ВидДвижения = ВидДвиженияНакопления.Расход* определяет вид движения регистра накопления, производимый этим документом как *Расход*, а в остальном процедура обработчика *ОбработкаПроведения* документа *ОказаниеУслуги* идентична обработчику документа *ПриходнаяНакладная* (см. листинг 6.1), подробно разобранному нами ранее.

В заключение отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра *Остатки Материалов*, связанному с документом.

Для этого откроем форму документа *ОказаниеУслуги*. В левом верхнем окне перейдем на закладку *Командный интерфейс*.

В разделе *Панель навигации* раскроем группу *Перейти* и установим видимость для команды открытия регистра накопления *Остатки материалов*.

### В режиме «1С:Предприятие»

Запустим «1С:Предприятие» в режиме отладки. В разделе *Оказание услуг* откроем документ *Оказание услуги № 1* и нажмем *Провести и закрыть*, то есть перепроведем его.

Теперь выполним команду *Остатки материалов* и откроем список нашего регистра накопления (рис. 6.19).

Период	Регистратор	Ном...	Материал	Склад	Количество
+ 07.07.2013 17:20:00	Приходная накладная 000000001 ...	1	Строчный трансформатор ...	Основной	10,000
+ 07.07.2013 17:20:00	Приходная накладная 000000001 ...	2	Строчный трансформатор ...	Основной	10,000
+ 07.07.2013 17:20:00	Приходная накладная 000000001 ...	3	Транзистор Philips 2N2369	Основной	10,000
+ 07.07.2013 17:33:12	Приходная накладная 000000002 ...	1	Кабель электрический	Основной	5,000
+ 07.07.2013 17:33:12	Приходная накладная 000000002 ...	2	Шланг резиновый	Основной	5,000
- 08.07.2013 15:10:55	Оказание услуги 000000001 от 08...	1	Транзистор Philips 2N2369	Основной	1,000

**Рис. 6.19.** Список регистра накопления «ОстаткиМатериалов»

Мы видим, что в регистре накопления *Остатки материалов* появилась еще одна запись, что соответствует количеству строк в табличной части проведенного документа.

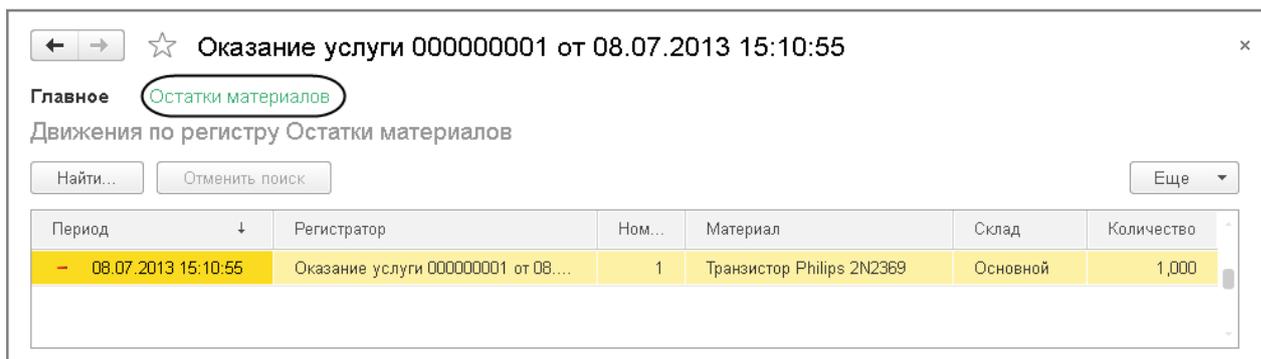
Все поля регистра заполнились данными документа так, как мы задали в обработчике проведения документа *Оказание услуги*.

Пиктограмма со знаком минус слева от записи указывает на тип движения – *Расход* (см. рис. 6.19).

Сейчас мы видим весь список движений регистра. Открыв этот список из формы документа, мы можем отфильтровать движения по документу-регистратору.

Для этого откроем еще раз документ *Оказание услуги № 1*.

В форме документа появилась панель навигации, в которой мы можем переходить к списку записей регистра *Остатки материалов*, связанному с документом, и обратно к содержимому документа (рис. 6.20).



**Рис. 6.20.** Переход к регистру накопления из формы документа

Однако сформированные таким образом движения этого документа будут не совсем правильны.

Дело в том, что в документе *Оказание услуги*, в отличие от документа *Приходная накладная*, могут содержаться не только расходуемые материалы, но и услуги. Поэтому в регистр *Остатки материалов* будут попадать записи и о расходуемых услугах, что неправильно.

Пока мы ничего не будем делать с движениями, которые сформировал конструктор, но как только познакомимся с перечислениями, мы внесем в обработчик проведения необходимые изменения.

### **Теория: способы работы с коллекцией**

В процессе формирования движений документов, когда в цикле обходили табличные части документов *ПриходнаяНакладная* и *ОказаниеУслуги*, мы столкнулись с одним из объектов встроенного языка, который является коллекцией.

Многие объекты встроенного языка являются коллекциями. Коллекция представляет собой совокупность объектов. Существуют общие принципы работы с любой коллекцией.

Во-первых, доступ к каждому объекту коллекции возможен путем перебора элементов коллекции в цикле. Для этого используется конструкция языка *Для Каждого Из ... Цикл ...* (листинг 6.3).

### Листинг 6.3. Перебор элементов коллекции в цикле

```
Для Каждого СтрокаТабличнойЧасти Из ТабличнаяЧасть Цикл
Сообщить (СтрокаТабличнойЧасти.Услуга) ;

КонецЦикла ;
```

В этом примере *ТабличнаяЧасть* – это коллекция строк табличной части объекта конфигурации. При каждом проходе цикла в переменной *СтрокаТабличнойЧасти* будет содержаться очередная строка из этой коллекции.

Во-вторых, существует доступ напрямую к элементу коллекции, без перебора коллекции в цикле. Здесь возможны различные комбинации двух обращений.

1. Во встроенном языке бывают именованные коллекции. То есть коллекции, в которых каждый элемент имеет некоторое уникальное имя. В этом случае обращение к элементу коллекции возможно по этому имени (листинг 6.4).

### Листинг 6.4. Обращение к элементу коллекции

```
Справочники.Сотрудники;
Справочники["Сотрудники"] ;
```

В этом примере *Справочники* – это коллекция менеджеров всех справочников, содержащихся в конфигурации. Так как каждый справочник конфигурации имеет свое уникальное имя, то к конкретному элементу этой коллекции (к менеджеру конкретного справочника) можно обратиться, указав имя этого справочника: *Справочники["Сотрудники"]*.

2. Если нет смысла в «персонификации» элементов коллекции (коллекция неименованная), тогда обращение к элементу коллекции возможно по индексу (индекс первого элемента коллекции – ноль), листинг 6.5.

#### **Листинг 6.5.** Обращение к элементу коллекции по индексу

```
ТабличнаяЧасть [0];
```

В этом примере *ТабличнаяЧасть* – это коллекция строк табличной части объекта конфигурации. И мы обращаемся к первому элементу этой коллекции, указывая его индекс – 0.

Следует отметить, что существуют коллекции, сочетающие оба вида обращений. Например, к коллекции колонок таблицы значений можно обращаться как по именам колонок, так и по индексу.

#### **Контрольные вопросы**

- *Для чего предназначен объект конфигурации «Регистр накопления»?*
- *Почему следует использовать регистры, хотя необходимая информация содержится в других объектах?*
- *Для чего нужны измерения регистра, ресурсы и реквизиты?*
- *Что такое движения регистра, и что такое регистратор?*
- *Как создать новый регистр накопления и описать его структуру?*
- *Как создать движения документа с помощью конструктора движений?*
- *Как средствами встроенного языка обойти табличную часть документа и обратиться к ее данным?*
- *Как показать команды открытия списка регистра в интерфейсе конфигурации и в интерфейсе формы?*