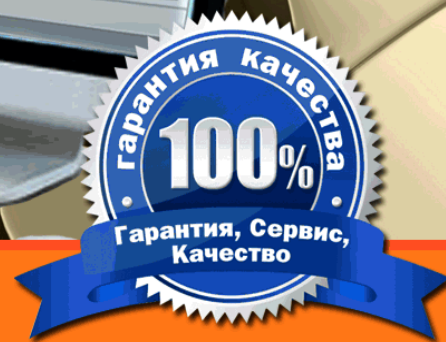


АЗБУКА РАСШИРЕНИЙ КОНФИГУРАЦИИ 1С

Низамутдинов Ильяс



9 ступеней к освоению технологии расширений

НИЗАМУТДИНОВ ИЛЬЯС

Азбука расширений конфигураций 1С

9 ступеней к освоению
технологии расширений

Ижевск, 2017 г

© Август 2017 г. Низамутдинов Ильяс. Все права защищены

Оглавление

Глава 1. Общие сведения	4
Глава 2. Заимствования объектов	14
Глава 3. Заимствования объектов (продолжение)	23
Глава 4. Контролируемые и модифицируемые свойства	35
Глава 5. Работа с программными модулями	46
Глава 6. Работа с формами	56
Глава 7. Работа с несколькими расширениями	66
Глава 8. Расширения и права доступа	80
Глава 9. Практическое использование механизма перехватов	92

Глава 1. Общие сведения

Прежде чем начать изучать работу с расширениями конфигурации, обращаю Ваше внимание, что работать с ними можно только на конфигурациях, у которых установлен режим совместимости не ниже 8.3.6.

Примеры к этой и последующим главам показаны на платформе 8.3.9 в конфигурации, у которой режим совместимости 8.3.8. («1С Бухгалтерия предприятия 3.0»).

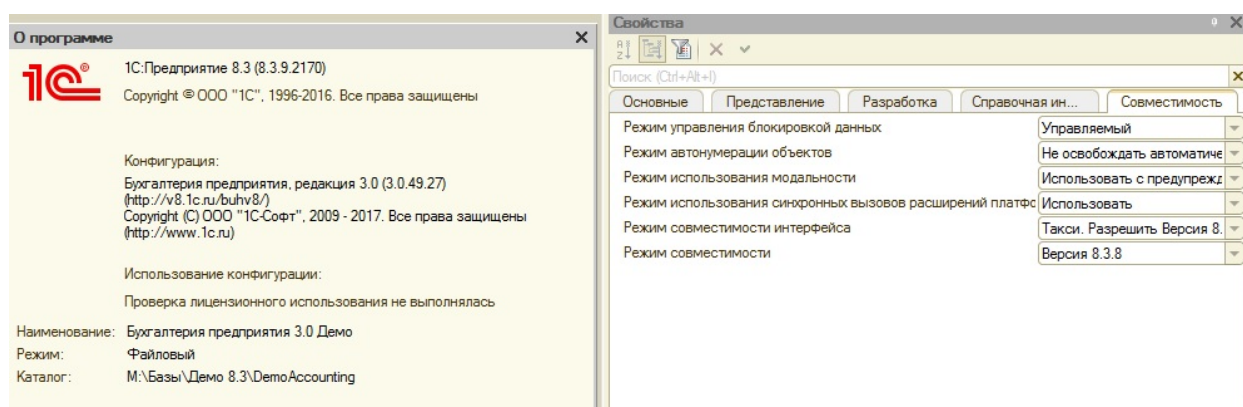


Рис. 1.1. Режим совместимости конфигурации

Что же такое расширение? Говоря простым языком, расширение конфигурации это конфигурация, которая как бы строится над нашей основной конфигурацией, причем она может содержать объекты, которые есть в основной конфигурации и которых нет в основной конфигурации.

Что очень удобно, когда мы добавляем расширение, то наша основная конфигурация остается на поддержке. Т.е. Вы ее можете спокойно обновлять, не боясь, что слетят ваши изменения.

Механизм этот постоянно дорабатывается фирмой 1С, и, к примеру, у платформы 8.3.9 работа с расширениями существенно отличается от платформы 8.3.7.

Но все по порядку.

В этой главе мы рассмотрим самые элементарные вещи: мы добавим расширение, создадим в ней новую подсистему, два отчета и обработку и добавим их в новую подсистему, а потом посмотрим, как это все работает.

У нас открыт конфигуратор «1С Бухгалтерия предприятия 3.0». Создадим расширение. Для этого идем в меню конфигурация – расширения конфигурации (см. рис. 1.2).

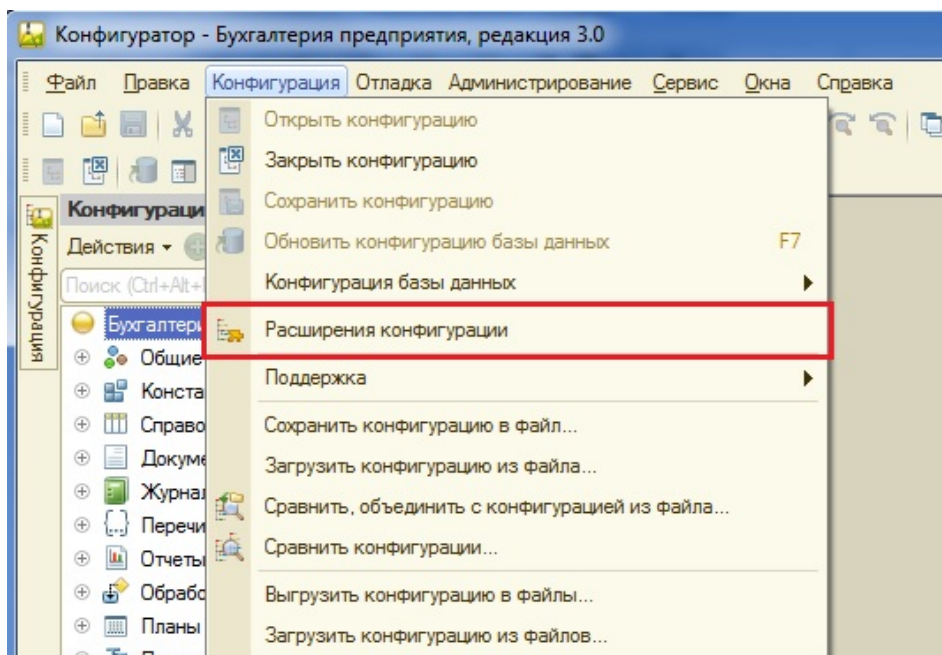


Рис. 1.2. Путь к расширениям конфигурации

Откроется форма списка расширений. Сейчас ничего нет, поэтому мы добавим новое расширение, нажав на кнопку «Добавить» (см. рис. 1.3).

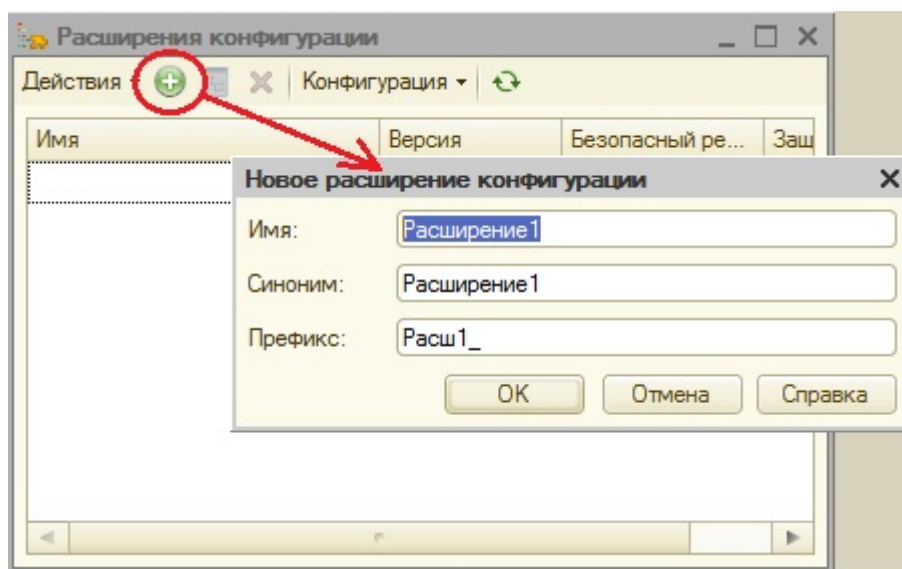


Рис. 1.3. Добавление нового расширения

Тут мы можем задать название, синоним и префикс расширения, с этого префикса будут начинаться новые объекты расширения. Назовем наше расширение

«Консолидированные отчеты», префикс будет КО. Нажмем кнопку ОК, и через некоторое время расширение будет создано (см. рис. 1.4.).

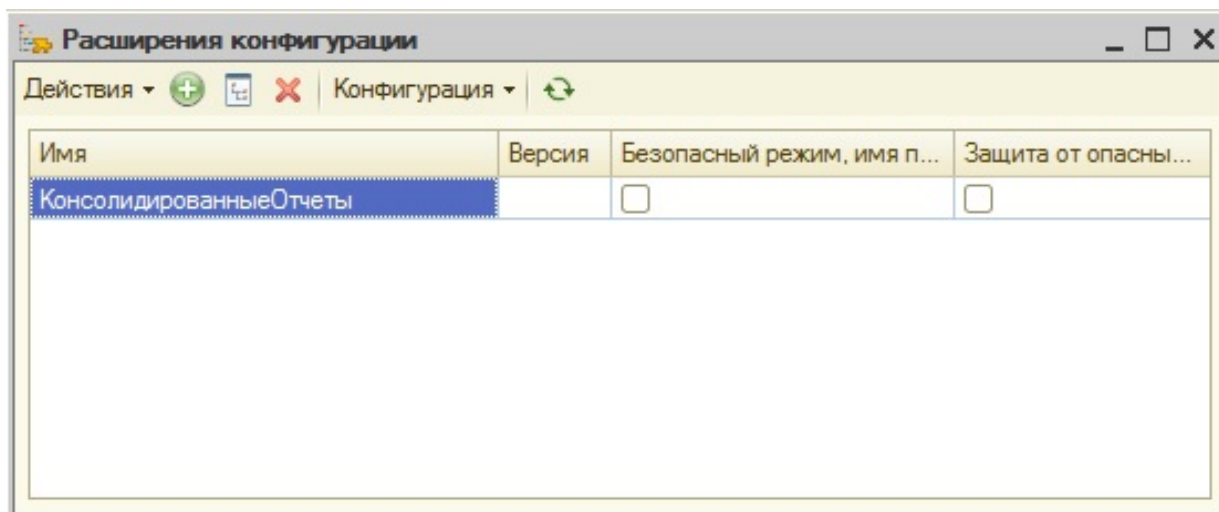


Рис. 1.4. Новое расширение «Консолидированные отчеты»

Теперь нажмем на кнопку «Открыть конфигурацию», после этого будет открыта конфигурация нового расширения (см. рис. 1.5).

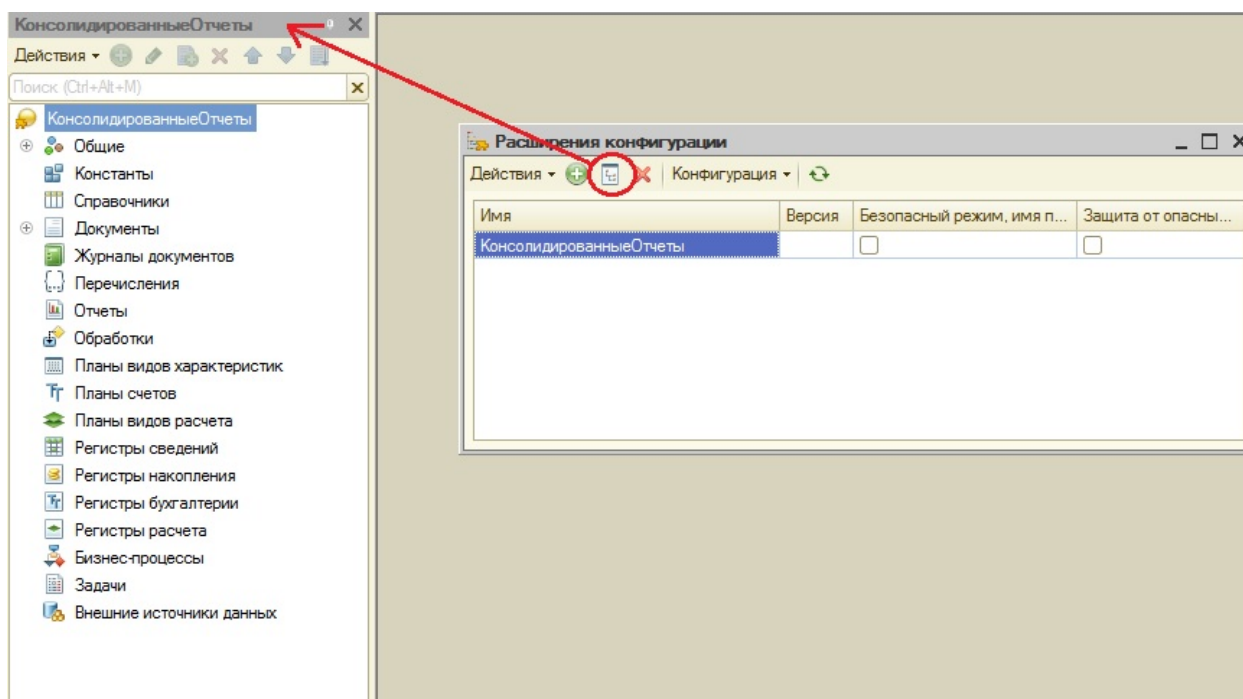


Рис. 1.5. Открытие конфигурации расширения

В эту конфигурацию Вы можете заимствовать объекты из основной конфигурации. Как это сделать, мы узнаем позже (см. глава 2, стр. 14). А пока создадим подсистему «Консолидированная отчетность» (см. рис. 1.6, 1.7).

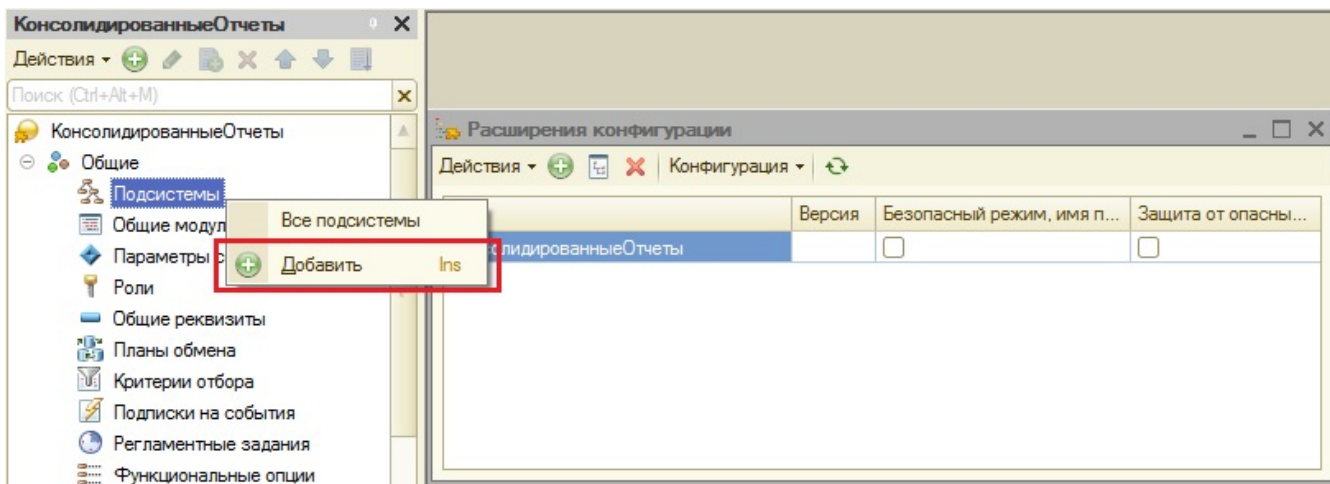


Рис. 1.6. Добавление новой подсистемы

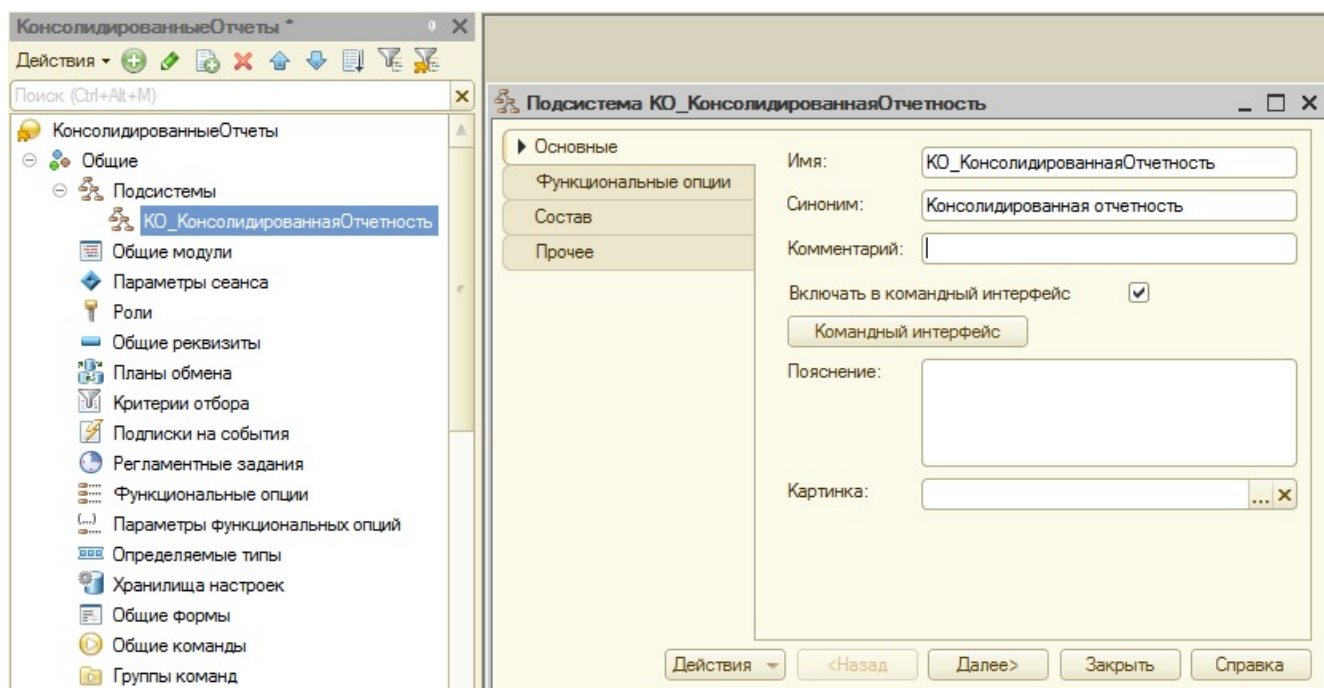


Рис. 1.7. Новая подсистема «Консолидированная отчетность»

Подсистема создана, заметьте, мы оставили флажок «Включать в командный интерфейс».

Перечислим объекты, которые можно создавать в расширении:

- Подсистемы
- Роли
- Общие модули
- Общие формы
- Общие команды

- Группы команд
- Общие макеты
- Общие картинки
- XDTO-пакеты
- Web-сервисы
- WS-ссылки
- Отчеты
- Обработки

Создадим два отчета и включим их в нашу новую подсистему (см. рис. 1.8, 1.9, 1.10, 1.11).

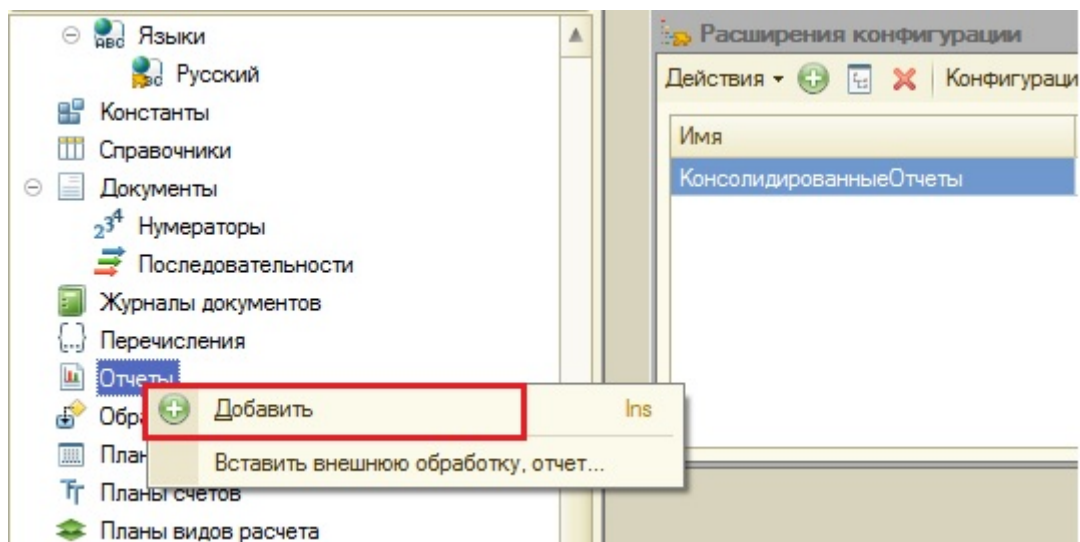


Рис. 1.8. Добавление отчета в расширении

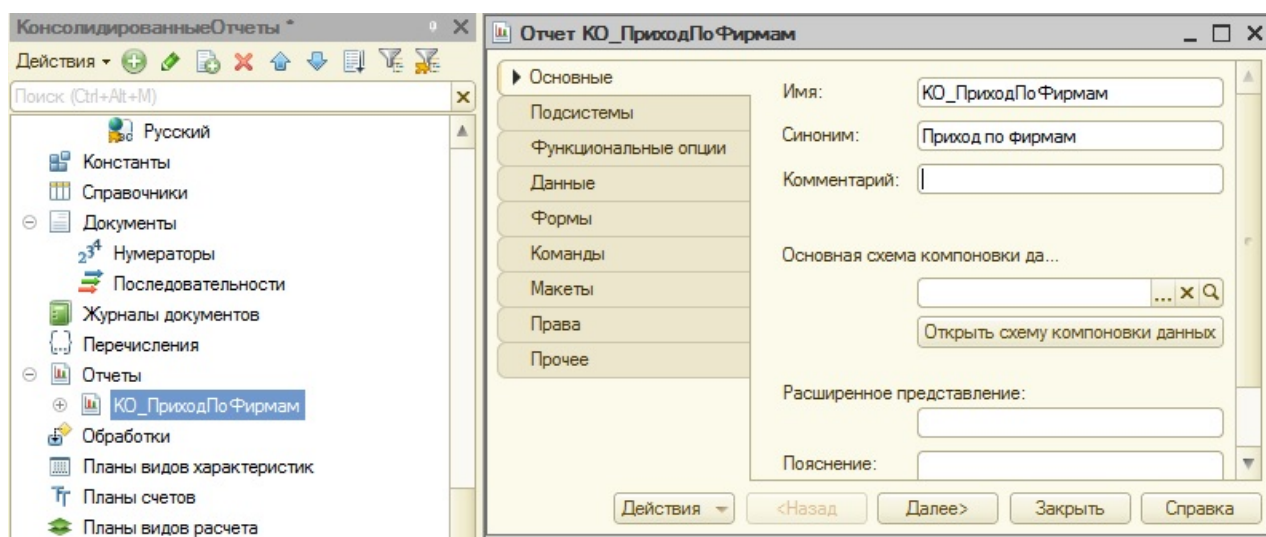


Рис. 1.9. Отчет «Приход по фирмам»

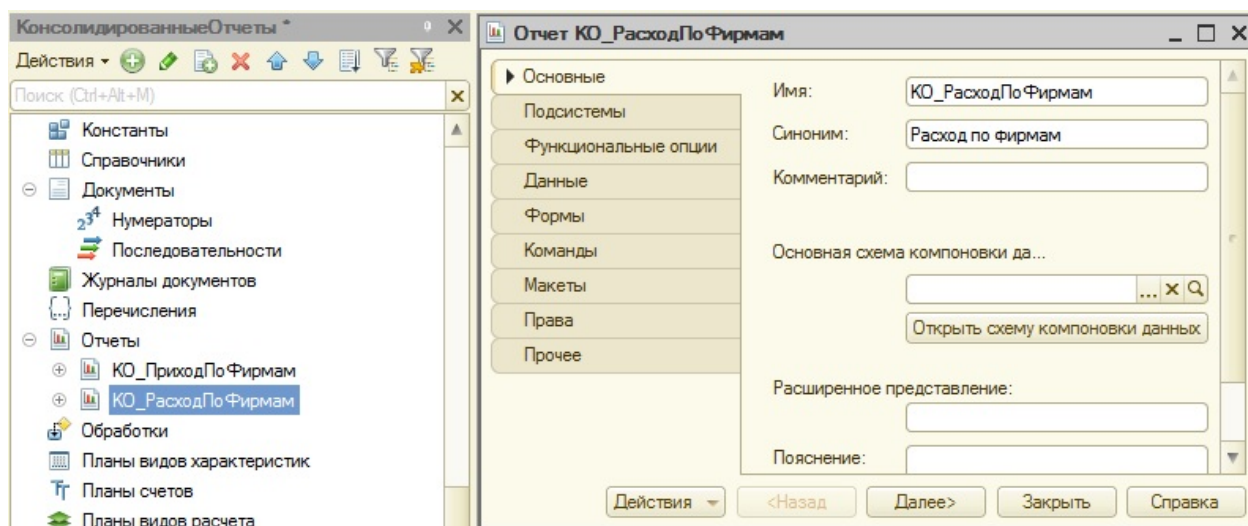


Рис. 1.10. Отчет «Расход по фирмам»

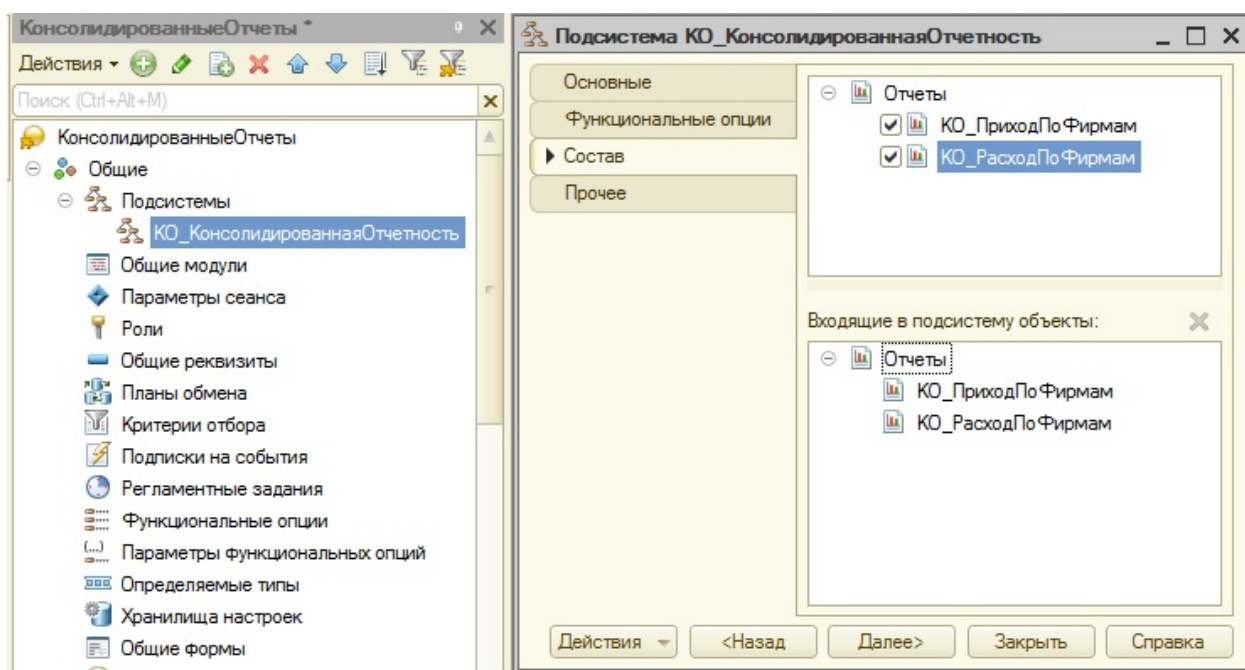


Рис. 1.11. Отчеты в составе подсистемы «Консолидированная отчетность»

Создадим обработку, назовем ее «Перемещение между фирмами» (см. рис.1.12) и включим её в подсистему.

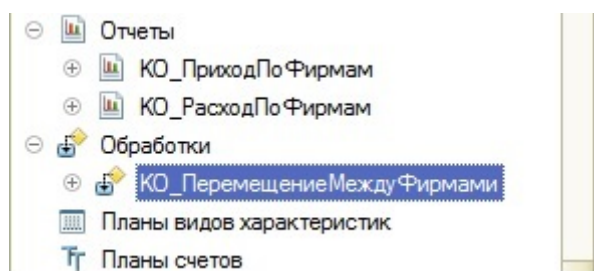


Рис. 1.12. Обработка «Перемещение между фирмами»

Добавим макет на отчеты и форму на обработку. Пусть они будут пустыми, это необходимо, чтобы мы увидели их в подсистеме (см. рис. 1.13, 1.14, 1.15).

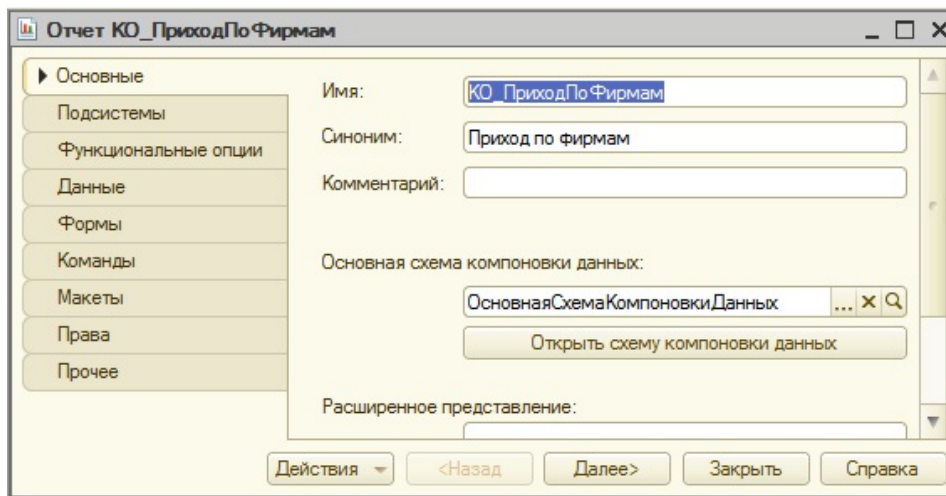


Рис. 1.13. Добавили схему компоновки в отчет «Приход по фирмам»

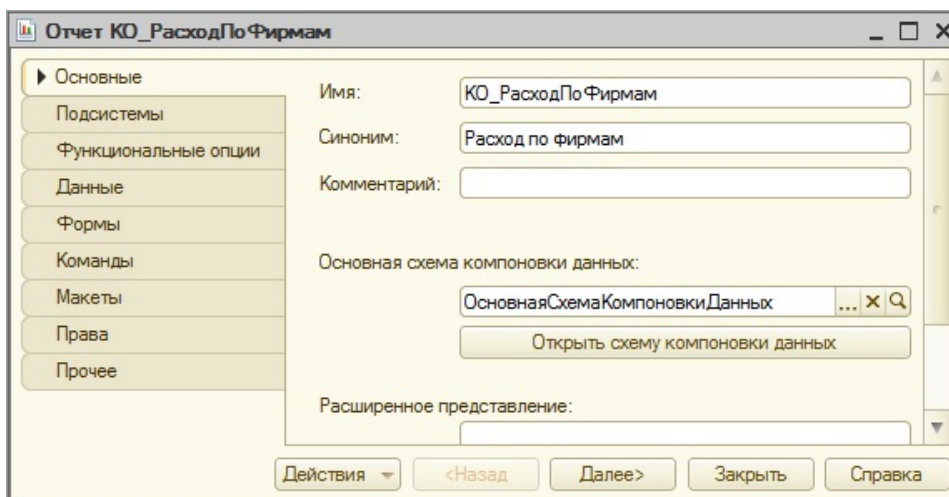


Рис. 1.14. Добавили схему компоновки в отчет «Расход по фирмам»

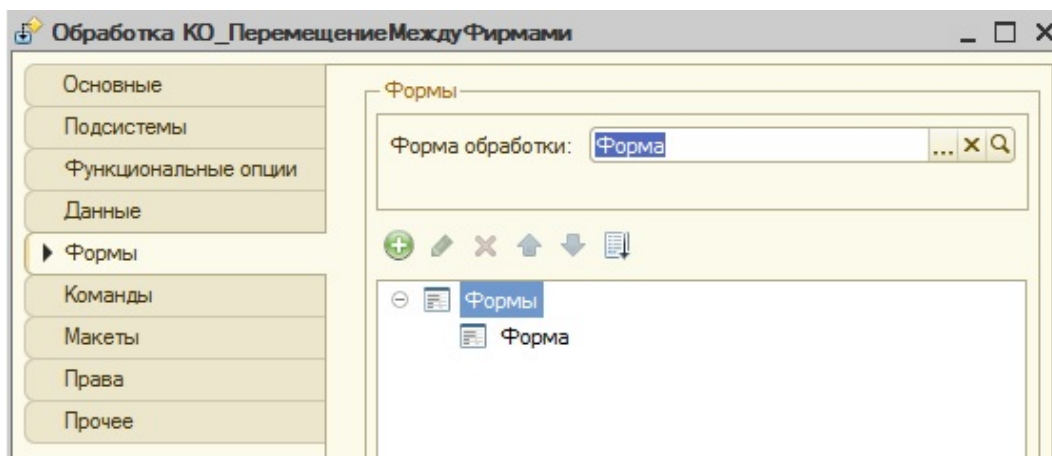


Рис. 1.15. Добавили основную форму обработки

Осталось внести поставщика и версию конфигурации в свойства конфигурации расширения (см. рис. 1.16, 1.17).

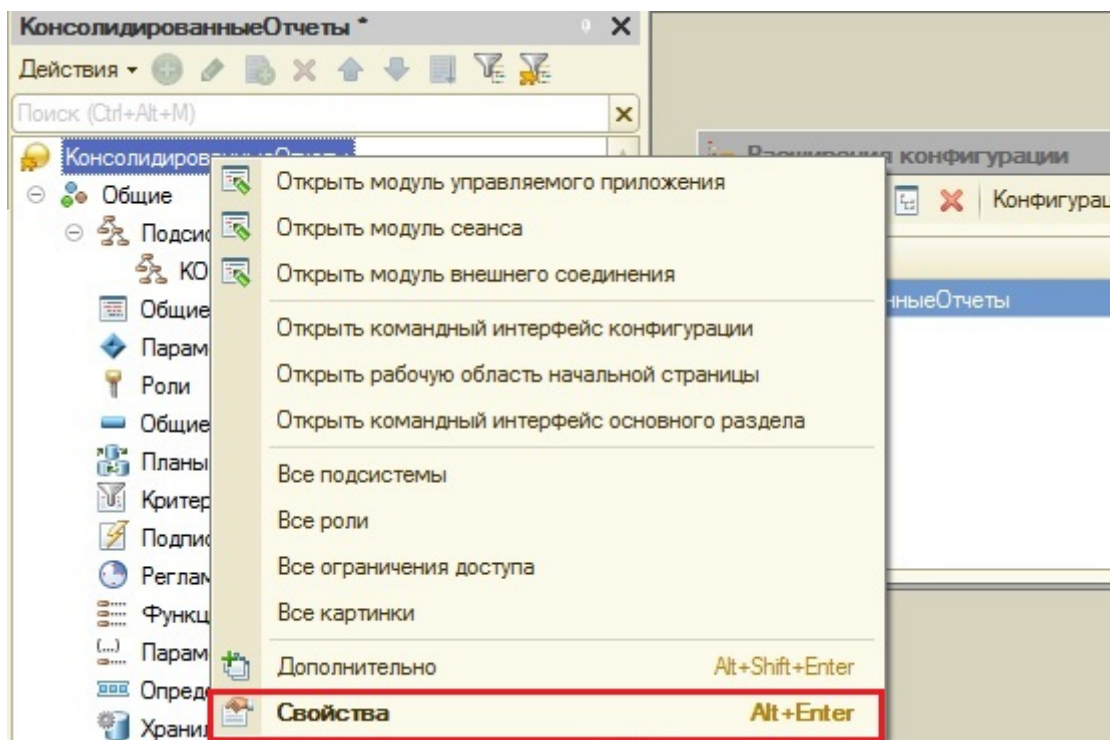


Рис. 1.16. Открываем палитру свойств конфигурации расширения

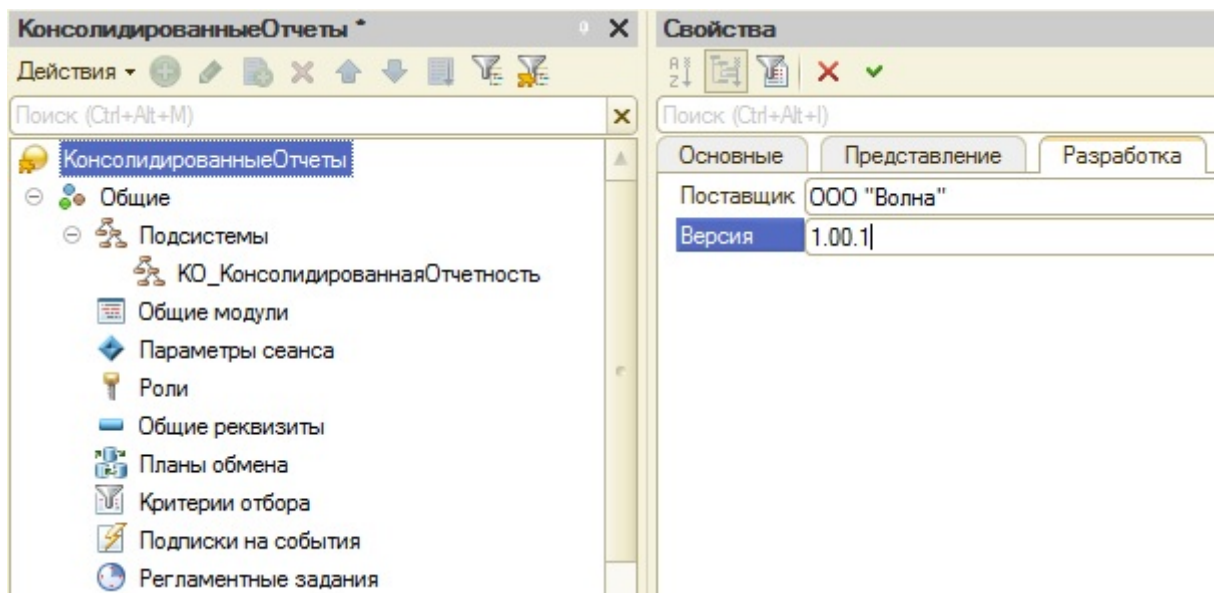


Рис. 1.17. Свойства «Поставщик» и «Версия»

На этом мы пока ограничимся. Сохраним нашу расширенную конфигурацию (см. рис. 1.18).

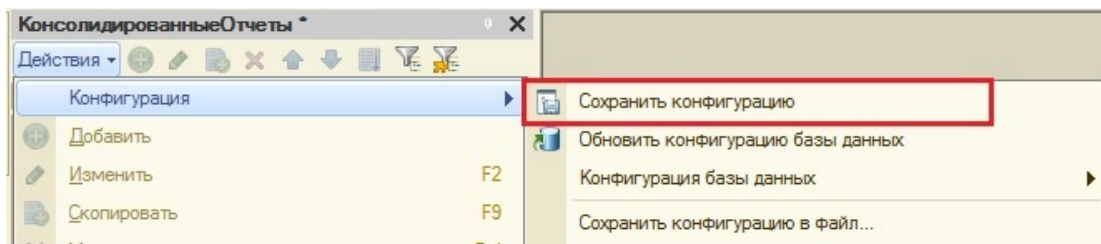


Рис. 1.18. Сохранение расширенной конфигурации

Обновим базу данных, чтобы наше расширение появилось в базе (см. рис. 1.19).

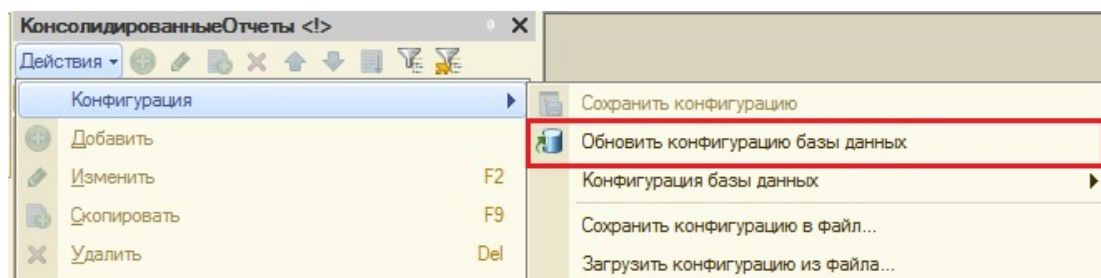


Рис. 1.19. Обновление конфигурации базы данных

Все, расширение создано, теперь посмотрим, как оно работает. Для этого запустим «1С: Предприятие» в пользовательском режиме.

Как видите, у нас добавилась новая закладка и три команды в ней, которые соответствуют новым отчетам и обработкам (см. рис. 1.20).

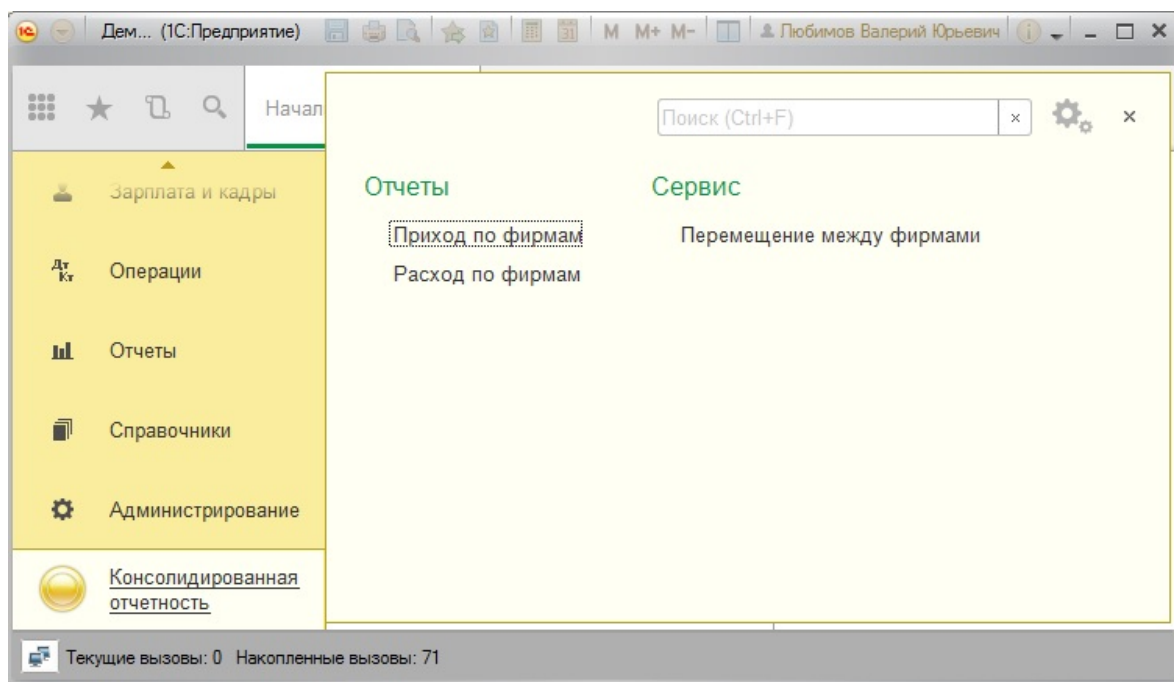


Рис. 1.20. Новая подсистема в конфигурации

Все отчеты пустые и ничего не возвращают. Для того, чтобы отчет как-то формировался, нужно заимствовать объекты из основной конфигурации в расширение. А как это сделать, мы узнаем в следующей главе.

Глава 2. Заимствование объектов

В прошлой главе мы остановились на том, что не смогли сформировать отчет в нашем расширении конфигурации. В этой главе мы с вами научимся заимствовать объекты в расширение из основной конфигурации.

Для тренировки решим простую задачу, на основании проведенных документов поступления товаров будем выводить отчет, в котором увидим какого товара, какой контрагент, на какую сумму нам продал. Понятно, что по уму это всё надо делать по регистрам, но наша задача сейчас в том, чтобы научиться работать с расширениями конфигурации, поэтому немного всё упростим.

Первым делом откроем расширение, созданное в предыдущей главе. Чтобы было удобно работать, я расположу расширяемую конфигурацию и расширенную рядом (см. рис. 2.1).

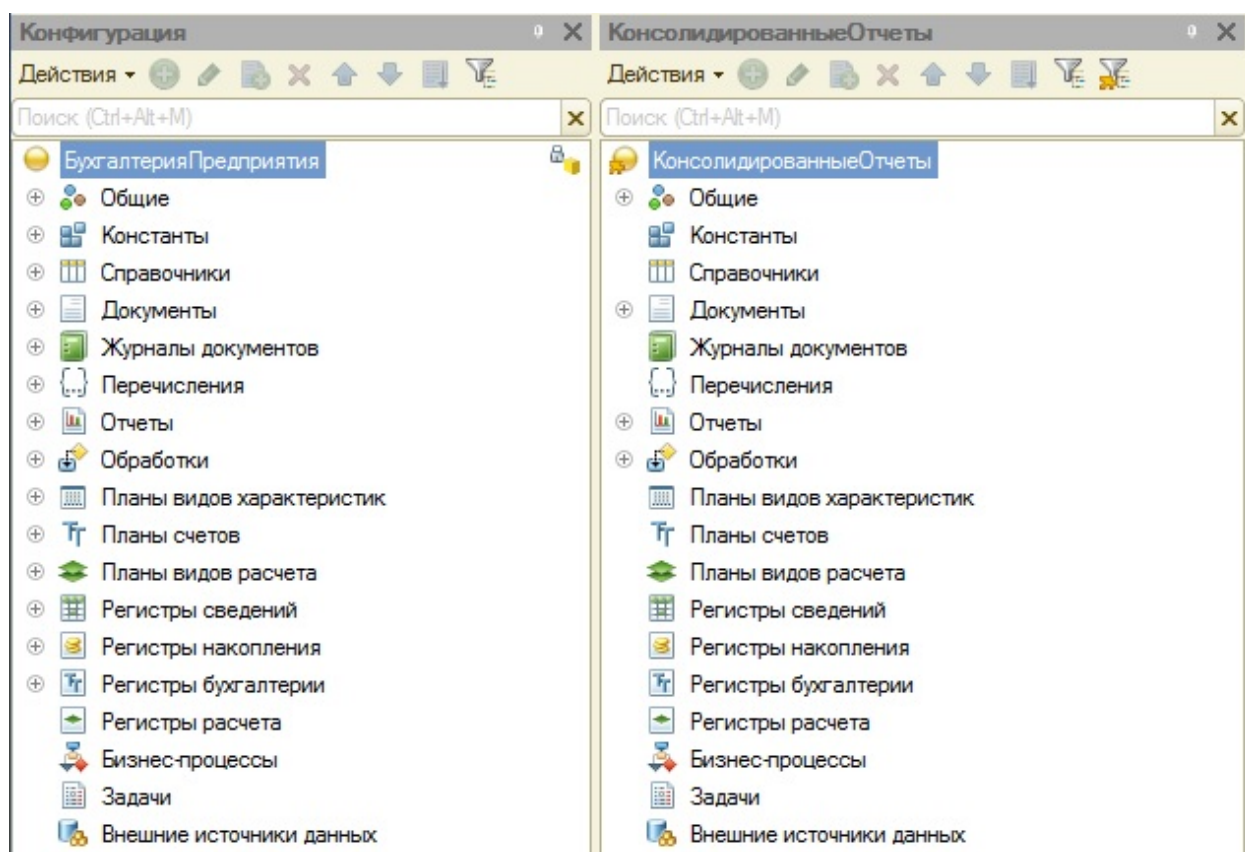


Рис. 2.1. Расширяемая и расширенная конфигурация

В это расширение нам нужно заимствовать документ «Поступление товаров и услуг». Для этого мы находим нужный документ в нашей расширяемой

конфигурации (слева), выделяем этот документ и нажимаем правую кнопку мышки. В контекстном меню у нас есть пункт – «Добавить в расширение» (см. рис. 2.2.).

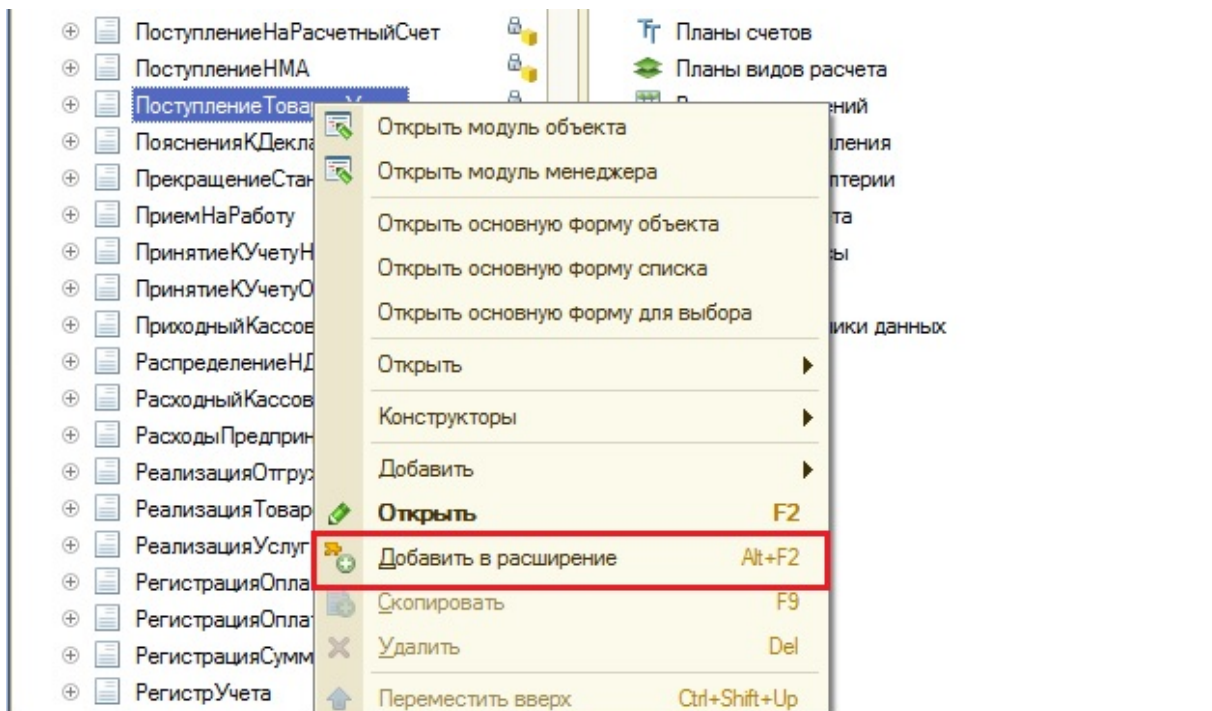


Рис. 2.2. Пункт меню «Добавить в расширение»

После того, как будет нажат этот пункт меню, документ появится в нашем расширении (см. рис. 2.3). Мы его заимствовали.

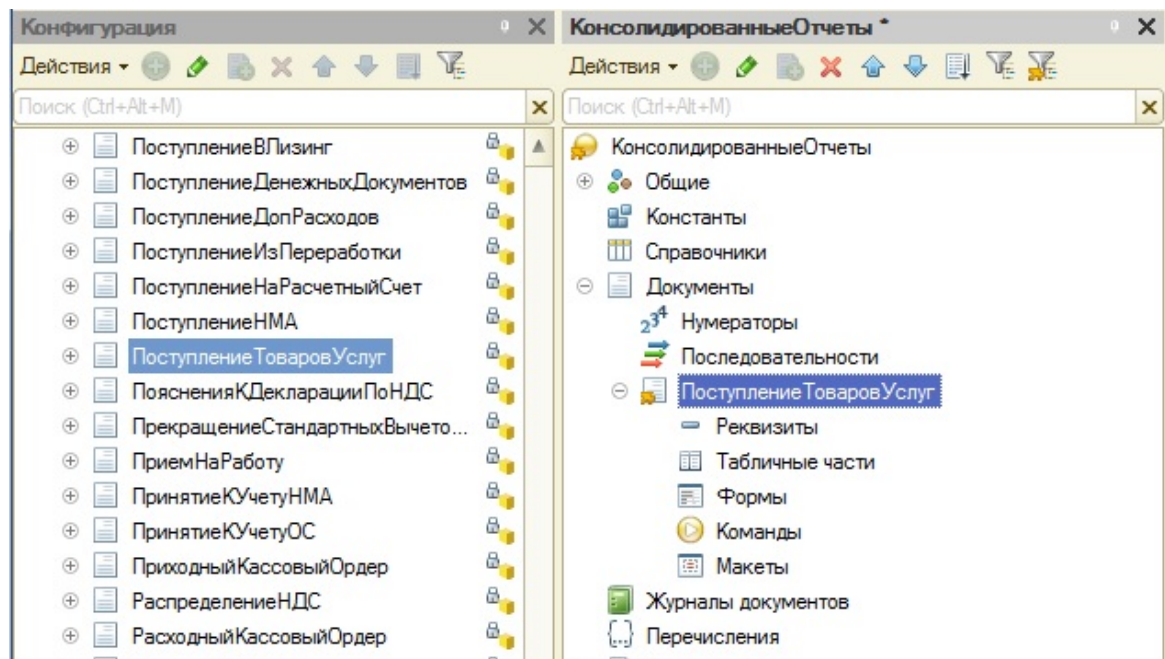


Рис. 2.3. Заимствованный документ «ПоступлениеТоваровУслуг» в расширении конфигурации

Из рисунка 2.3 мы видим, что перенесся только сам документ. Никакие подчиненные объекты справочника, будь то реквизиты, табличные части и формы, не были заимствованы.

Но в то же время, если мы добавляем в расширение какой-нибудь объект ссылочного типа, то также автоматически заимствуется объект, на который этот тип ссылается. Например, заимствуем реквизит «Контрагент» документа «Поступление товаров и услуг» (см. рис. 2.4, 2.5).

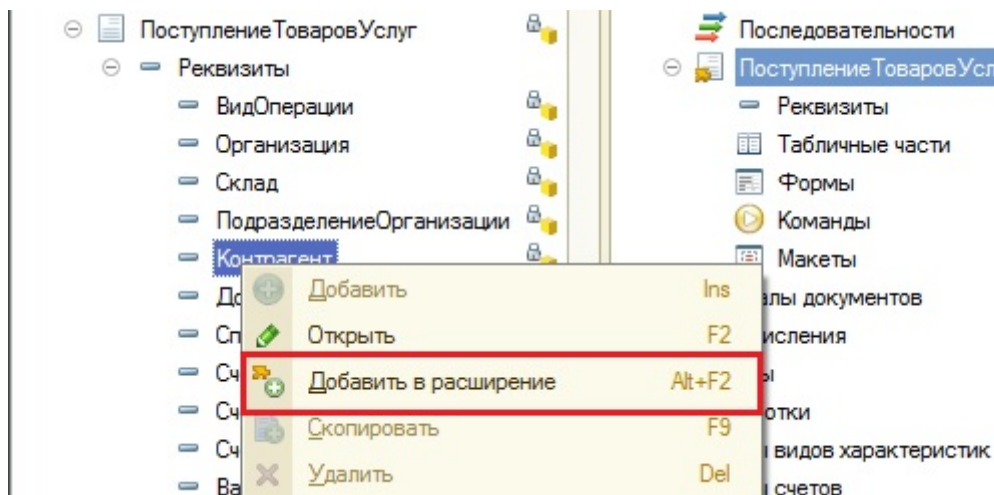


Рис. 2.4. Заимствование реквизита «Контрагент» документа «Поступление товаров и услуг»

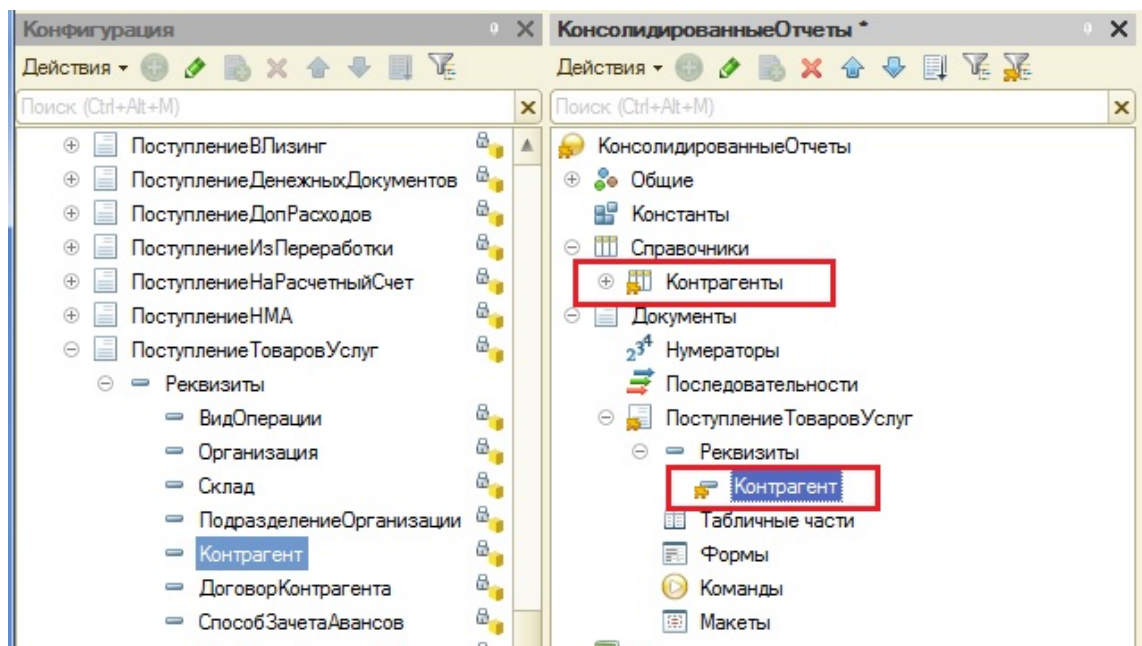


Рис. 2.5. Реквизит «Контрагент» документа «Поступление товаров и услуг» в расширении конфигурации

Как видно из рисунка 2.5, автоматически появился и сам справочник «Контрагент». Теперь заимствуем реквизит «Номенклатура» табличной части «Товары» (см. рис. 2.6).

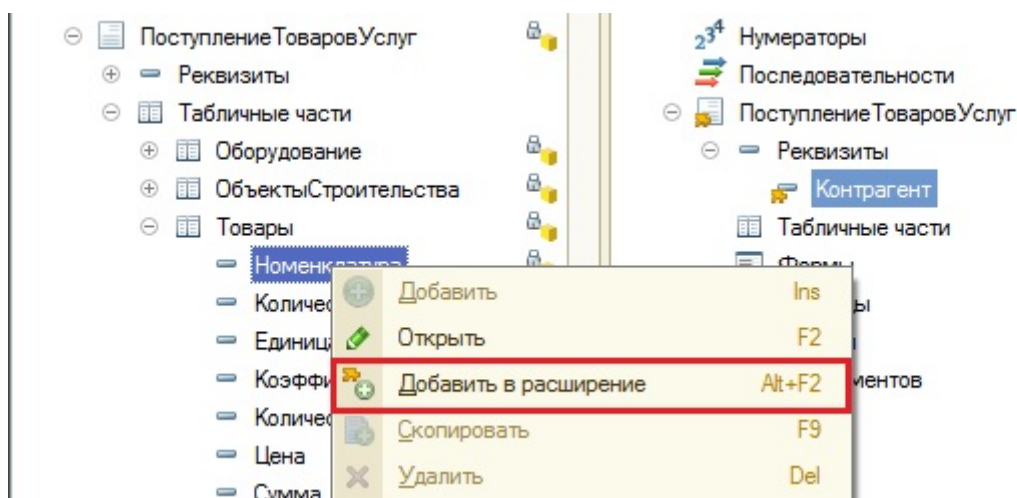


Рис. 2.6. Добавление реквизита «Номенклатура» табличной части «Товары» документа «Поступление товаров и услуг»

Табличная часть «Товары» в этом случае тоже заимствовалась, как и справочник «Номенклатура» (см. рис. 2.7).

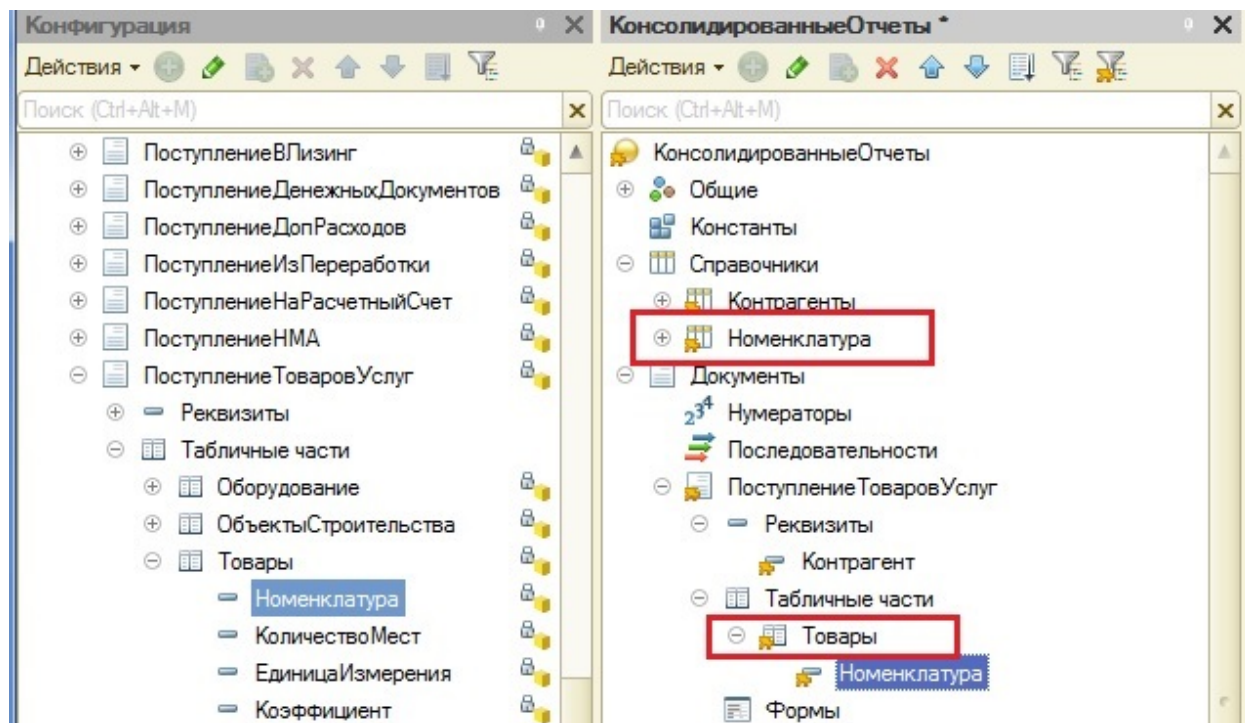


Рис. 2.7. После заимствования реквизита «Номенклатура» табличной части «Товары»

И напоследок, для отчета добавим в расширение конфигурации реквизит «Сумма» табличной части «Товары» (см. рис. 2.8).

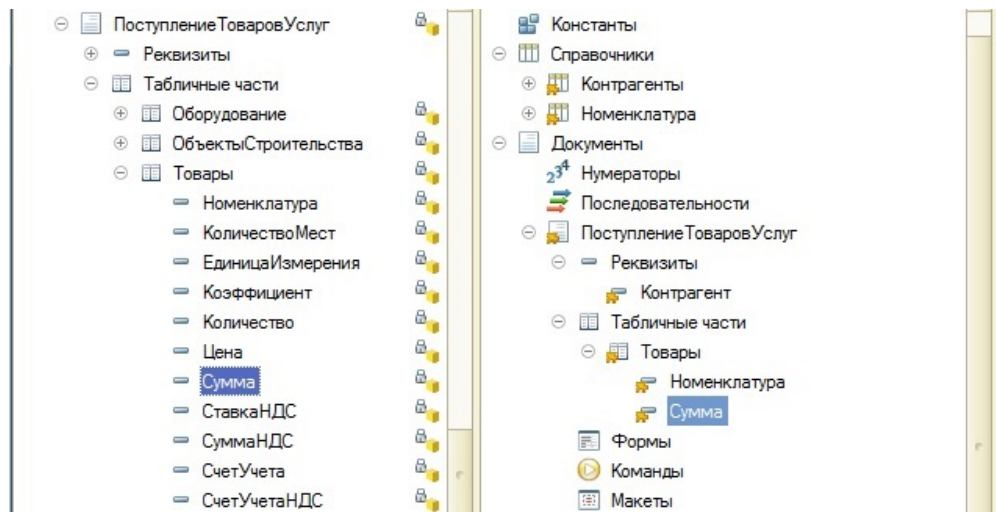


Рис. 2.8. Заимствовали реквизит «Сумма» табличной части «Товары»

Обратите внимание, мы добавили документ «Поступление товаров и услуг» и только один реквизит шапки «Контрагент». Вместе с этим документом добавились все его стандартные реквизиты, такие как дата, номер, пометка на удаление и прочие.

Все нужное для отчета «Приход по всем фирмам» заимствовано, теперь зайдем в схему компоновки данных отчета «Приход по всем фирмам», добавим набор данных с видом «Запрос», откроем конструктор запросов, в котором раскроем таблицу «Поступление товаров и услуг» (см. рис. 2.9).

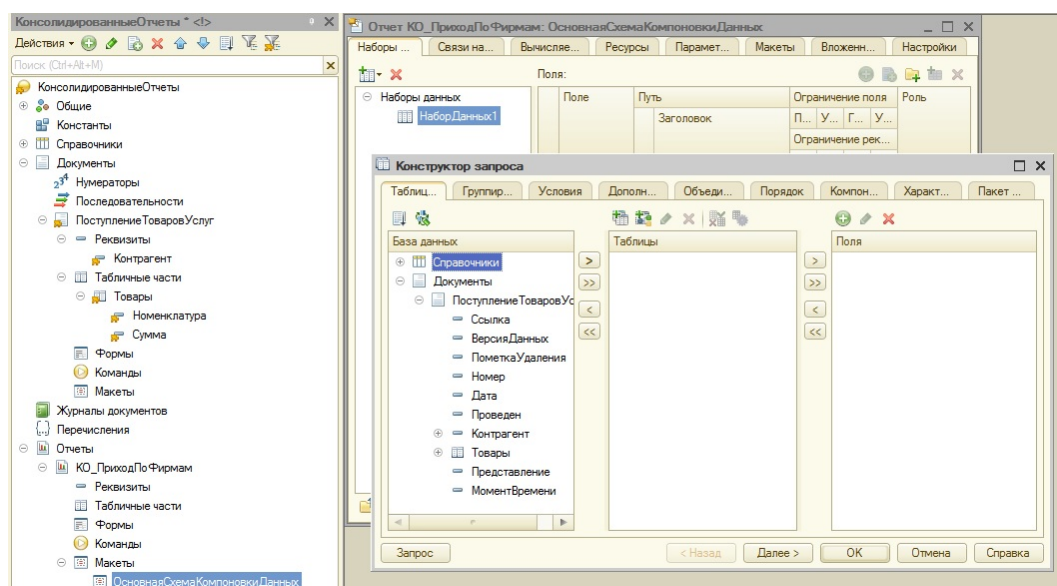


Рис. 2.9. Создали набор данных для отчета «Приход по всем фирмам»

Как Вы видите, мы можем выбрать любой стандартный реквизит документа. Мы же выберем табличную часть «Товары» (см. рис. 2.10).

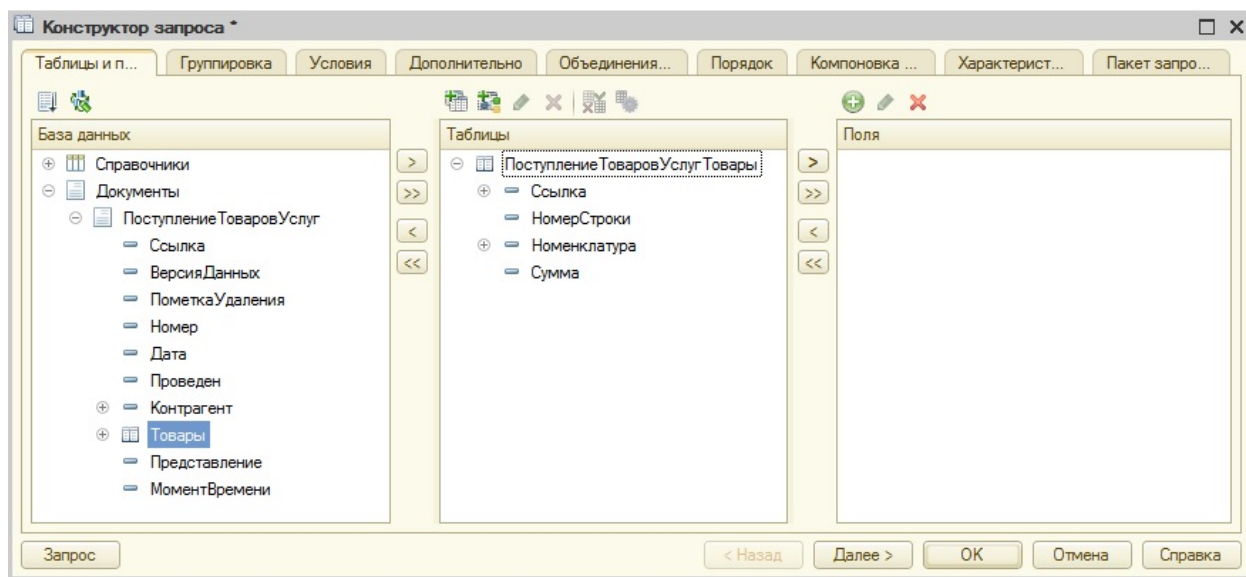


Рис. 2.10. Выбрали табличную часть «Товары» документа «Поступление товаров и услуг»

После этого раскроем реквизит «Ссылка» и выберем номер, дату и контрагента (см. рис. 2.11).

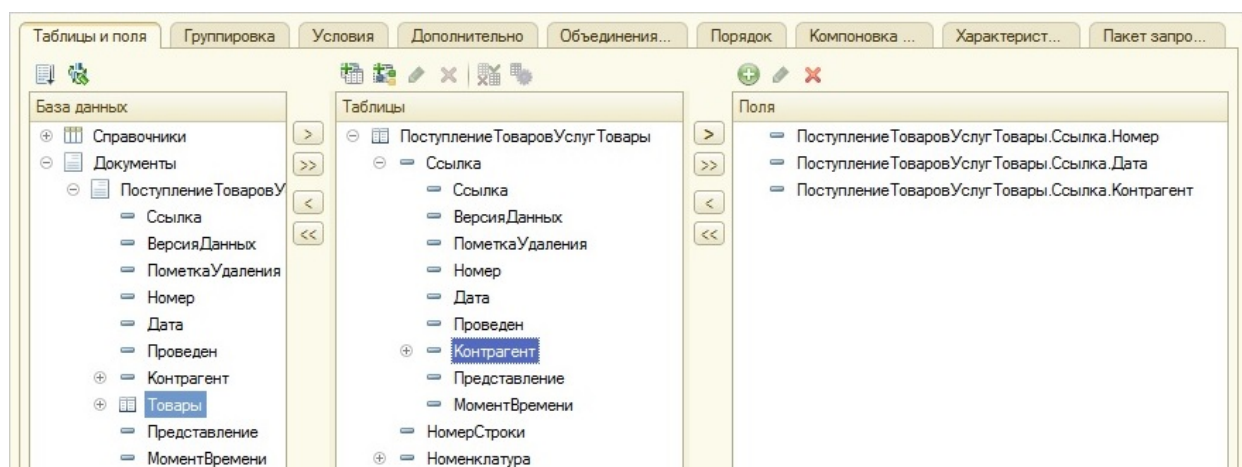


Рис. 2.11. Выбрали реквизиты шапки «Номер», «Дата», «Контрагент»

А из самой табличной части выберем номенклатуру и сумму (см. рис. 2.12).

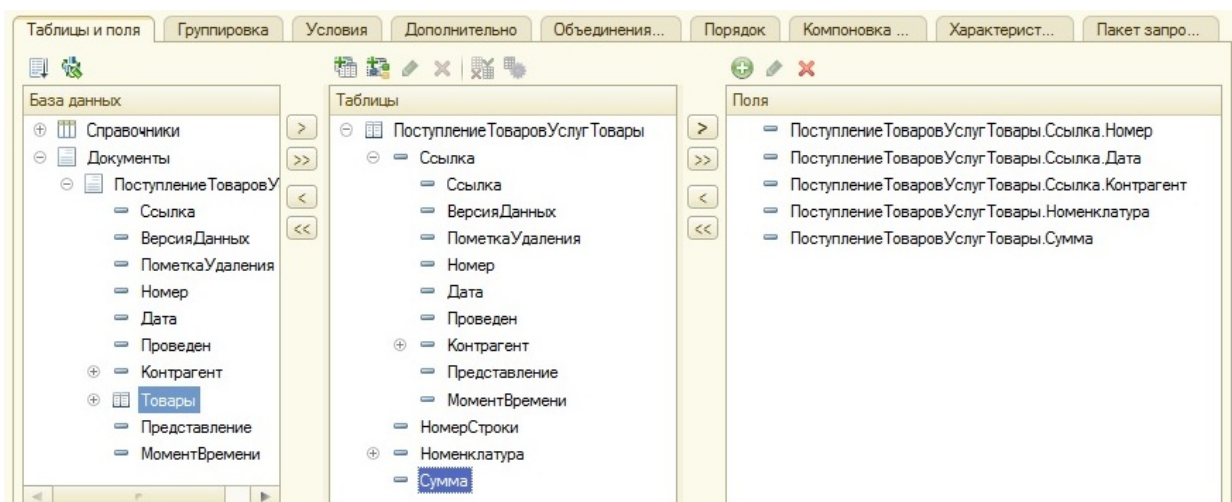


Рис. 2.12. Выбрали из табличной части «Номенклатура» и «Сумма»

Поставим условие, что документ должен быть проведен.

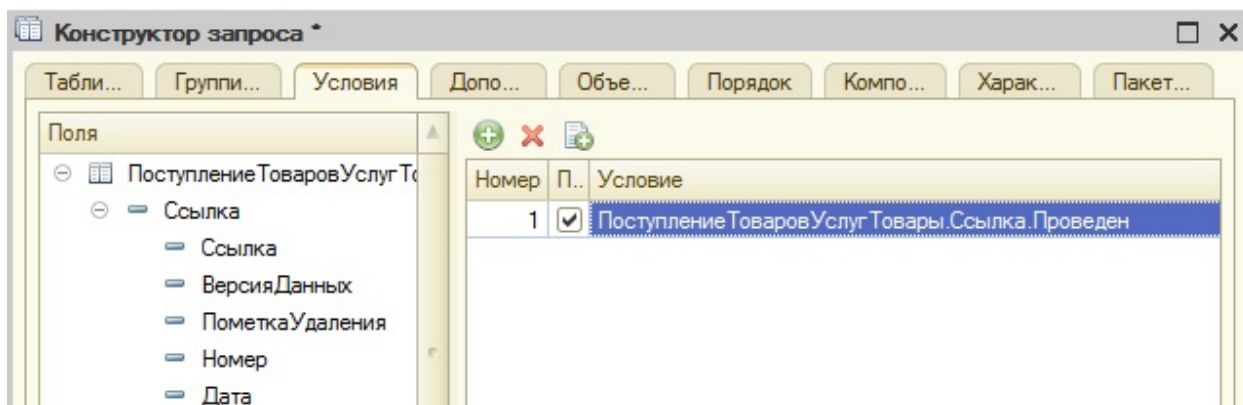


Рис. 2.13. Поставили условие

Также поставим условия на отбор по датам (см. рис. 2.14).

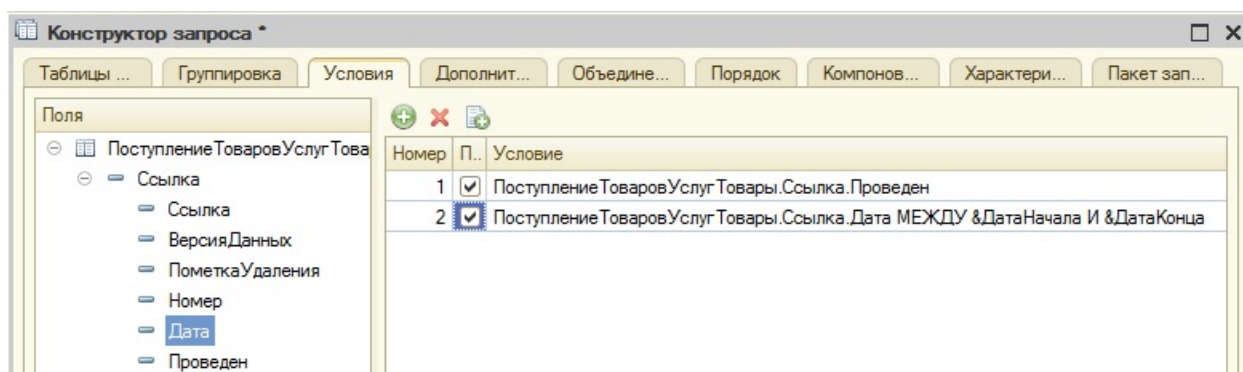


Рис. 2.14. Установили отбор по датам

Теперь сохраним запрос, добавим поле «Сумма» в ресурс (см. рис. 2.15) и создадим настройки компоновки данных (см. рис. 2.16).

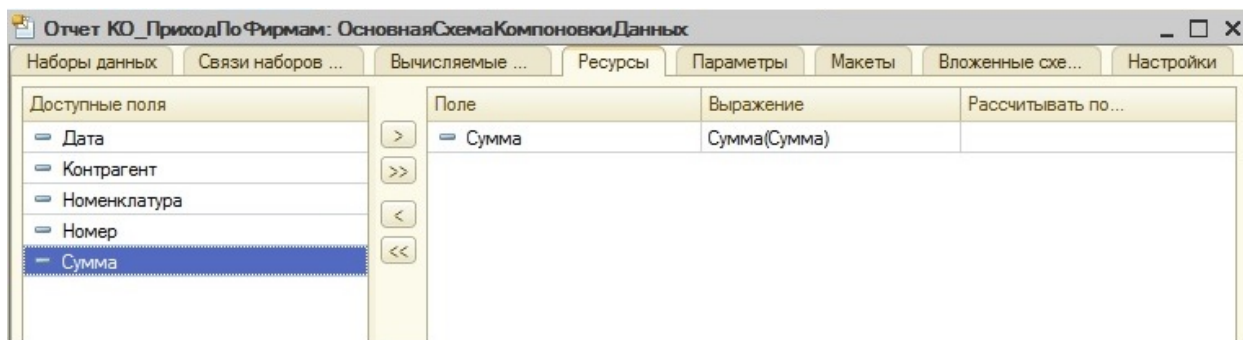


Рис. 2.15. Добавили поле «Сумма» в ресурс

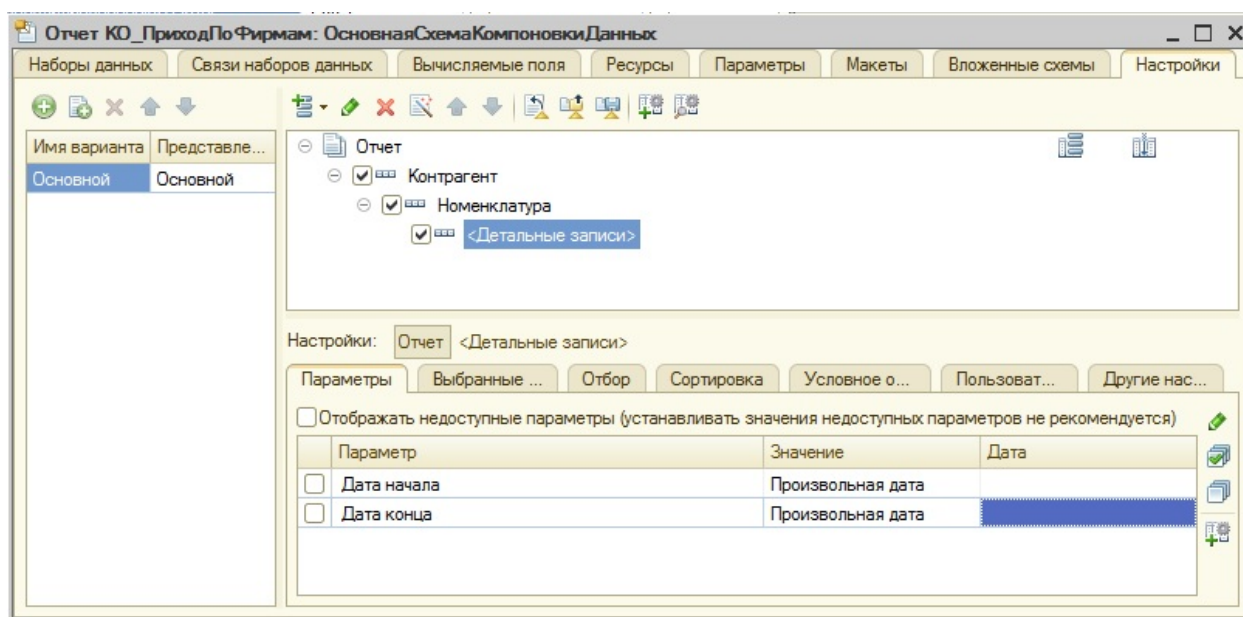


Рис.2.16. Настройки компоновки данных

Сохраним наше расширение и обновим базу (как это делается см. стр 12).

Осталось перезапустить базу данных и посмотреть, как работает отчет (см. рис. 2.17).

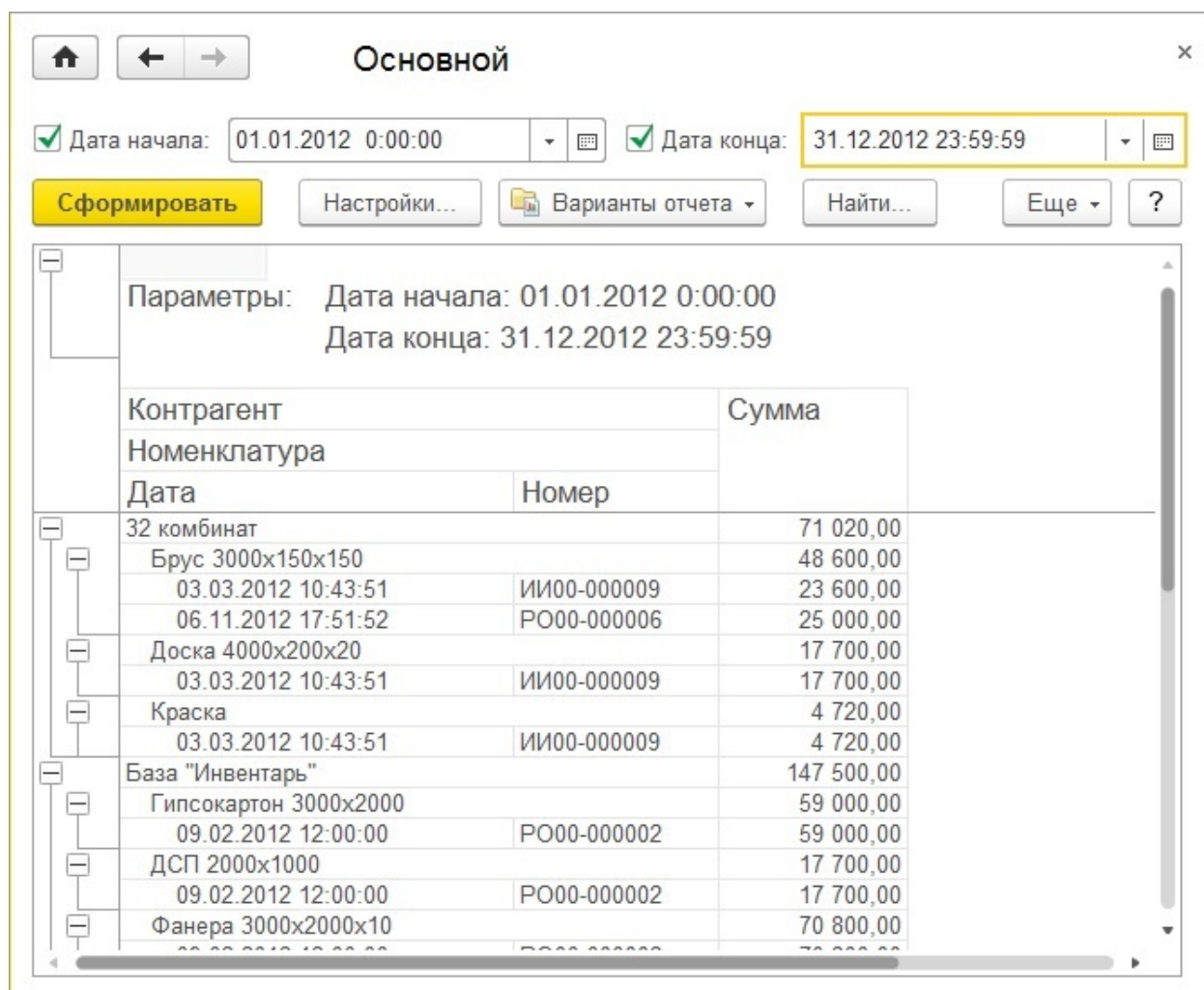


Рис. 2.17. Отчет «Приход по всем фирмам»

Точно так же делается отчет «Расход по всем фирмам»: добавляются нужные объекты в расширение и по ним формируется отчет в СКД.

На этом мы закончим нашу главу, где учились заимствовать объекты в расширение, в следующей главе мы заимствуем регистр бухгалтерии и научимся делать отчеты посложнее.

Глава 3. Заимствование объектов (продолжение)

В прошлой главе мы научились заимствовать объекты в расширение и сделали простой отчет, в котором вывели данные по документам. В этой главе мы сделаем отчет посложнее. Выведем весь приход по кассе. Чтобы такую информацию вывести, в запросе необходимо использовать виртуальные таблицы регистров бухгалтерии.

Прежде чем что-то делать в расширении, я сделаю запрос в консоли запросов 1С, который вернет нам оборот по дебету для всех 50-х счетов. Поскольку 50-й счет активный, то оборот по дебету этого счета и есть, по сути, приход по кассе (см. рис. 3.1).

Имя параметра	Тип	Значение
Счет50	Счет бухгалтерского учета	50

Текст запроса:

```

ВЫБРАТЬ
    ХозрасчетныйОбороты.Счет,
    ХозрасчетныйОбороты.Регистратор,
    ХозрасчетныйОбороты.Субконто1,
    ХозрасчетныйОбороты.Организация,
    ХозрасчетныйОбороты.СуммаОборотДт
ИЗ
    РегистрБухгалтерии.Хозрасчетный.Обороты(, , Регистратор,
        Счет В ИЕРАРХИИ (&Счет50),
        , , , ) КАК ХозрасчетныйОбороты
  
```

Рис. 3.1. Запрос по регистру «Хозрасчетный»

В параметрах запроса я устанавливаю 50-й счет. Выполним запрос и посмотрим, что у нас получится (см. рис. 3.2).

Запрос: РегистрБухгалтерии.Хозрасчетный.Обороты(, (Записей в результате: 32)

Счет	Регистратор	Субконто1	Организация	СуммаОборотДт
50.01	Отчет о розничных продажах МН00-000000	Приход денежных средств роз	Магазин 23	52 500
50.01	Поступление наличных МН00-000003	Приход денежных средств роз	Магазин 23	11 700
50.01	Поступление наличных ТД00-000003	Оплата от покупателя	Торговый дом "Комплексный"	73 605
50.01	Поступление наличных КС00-000001		Комфорт-сервис	70 000
50.01	Поступление наличных МН00-000001	Розничная выручка от НТТ	Магазин 23	73 300
50.03	Поступление денежных документов	Билет Москва-Санкт-Петербург	Иванов И. И. ИП	6 000
50.03	Поступление денежных документов	Билет авиа Москва-Санкт-Пет	Конфетпром	12 000
50.01	Поступление наличных ШИ00-000001	Розничная выручка от НТТ	Шилов С.А.	92 395
50.01	Поступление наличных СЛ00-000001	Получение наличных в банке	Сервислог	65 250

Рис. 3.2. Результат выполнения запроса с рис. 3.1.

«Субконто1» - это статья движения денежных средств.

Сможем ли мы сделать отчет по такому запросу в нашем расширении? Создадим в нашем расширении «Консолидированная отчетность» новый отчет «Приход по кассам», добавим ему схему компоновки и откроем конструктор запроса (см. рис. 3.3)

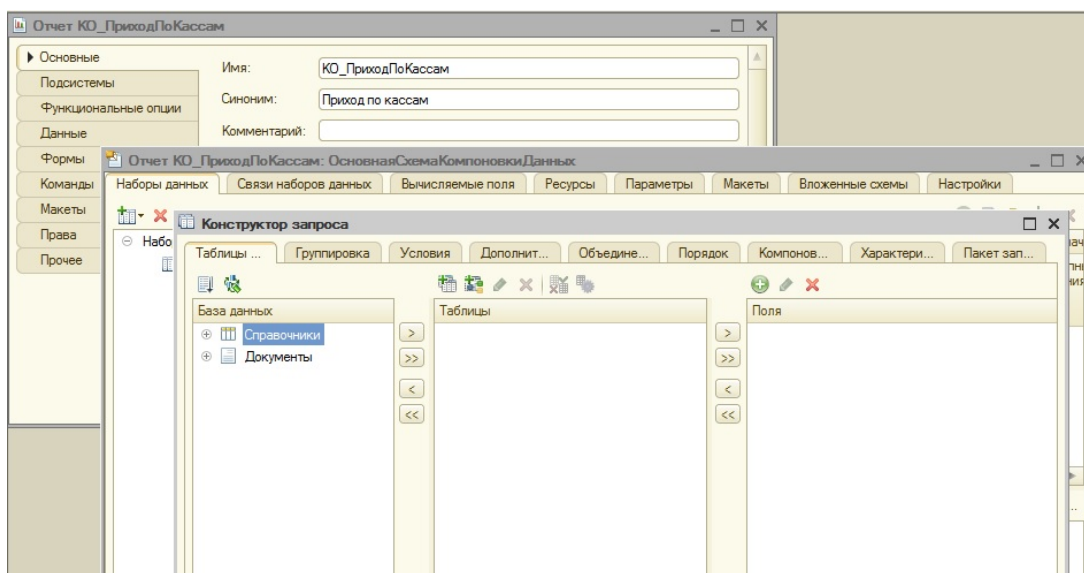


Рис. 3.3. Конструктор запросов у отчета «Приход по кассам»

Как видно из рисунка 3.3, никаких виртуальных таблиц у нас на данный момент нет. Т.е. первым делом нам нужно заимствовать регистр бухгалтерии (см. рис. 3.4).

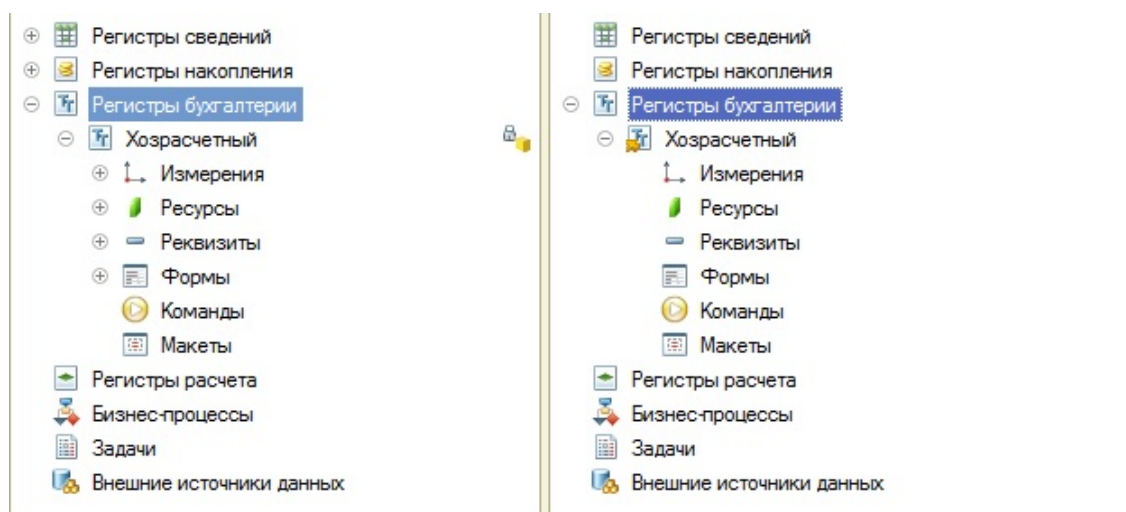


Рис. 3.4. Заимствование регистра бухгалтерии

Регистр бухгалтерии мы заимствовали, теперь зайдём обратно в схему компоновки отчета «Приход по кассам», откроем конструктор запросов и посмотрим, доступны ли станут виртуальные таблицы (см. рис. 3.5).

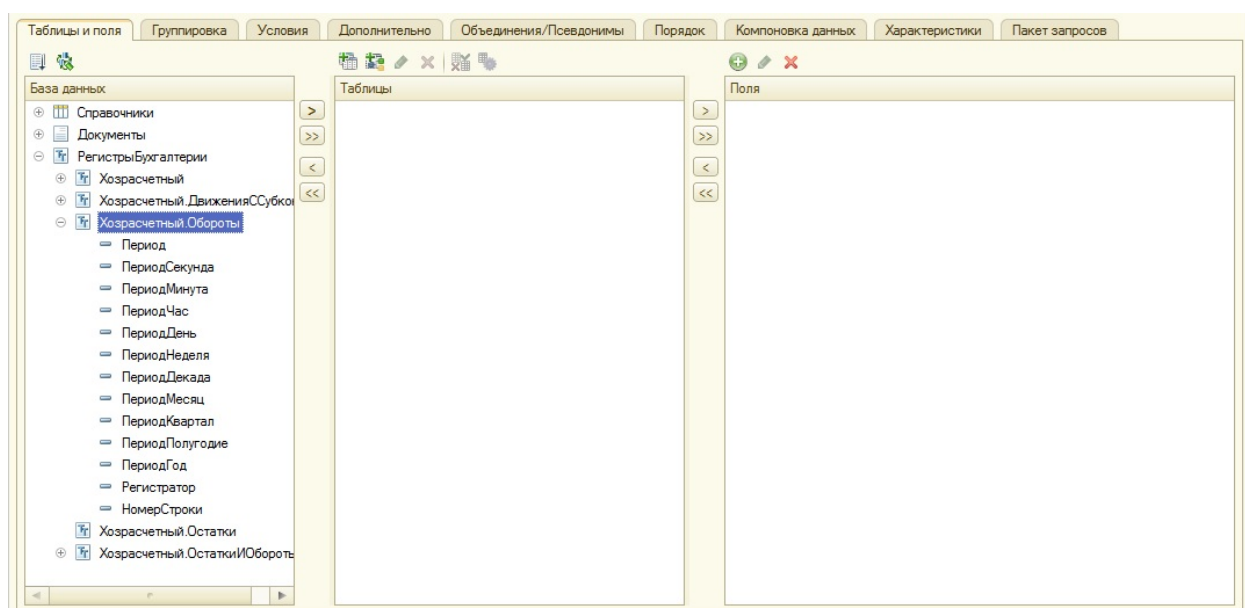


Рис. 3.5. Регистр бухгалтерии в конструкторе запросов

В целом нужная для нас таблица есть, но у нее нет ни счета, ни корсчета, ни ресурсов, ни субконто, которое мы используем в запросе.

Для того чтобы появился счет, необходимо заимствовать план счетов, который привязан к нашему регистру. В конфигурации он один – «Хозрасчетный», заимствуем его (см. рис. 3.6).

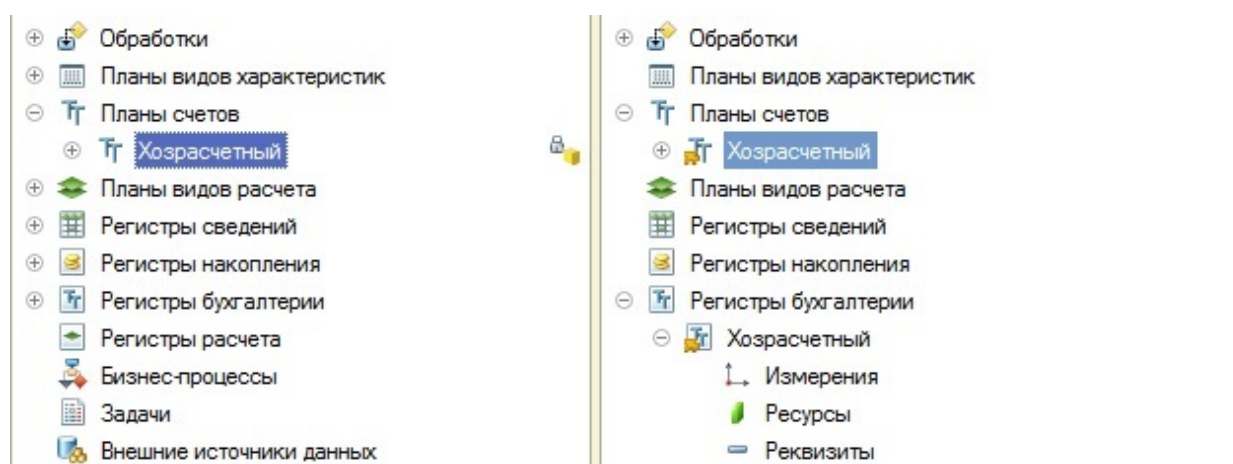


Рис. 3.6. Заимствуем план счетов

Теперь привяжем план счетов к нашему регистру. Для этого зайдём в свойства заимствованного регистра (см. рис. 3.7).

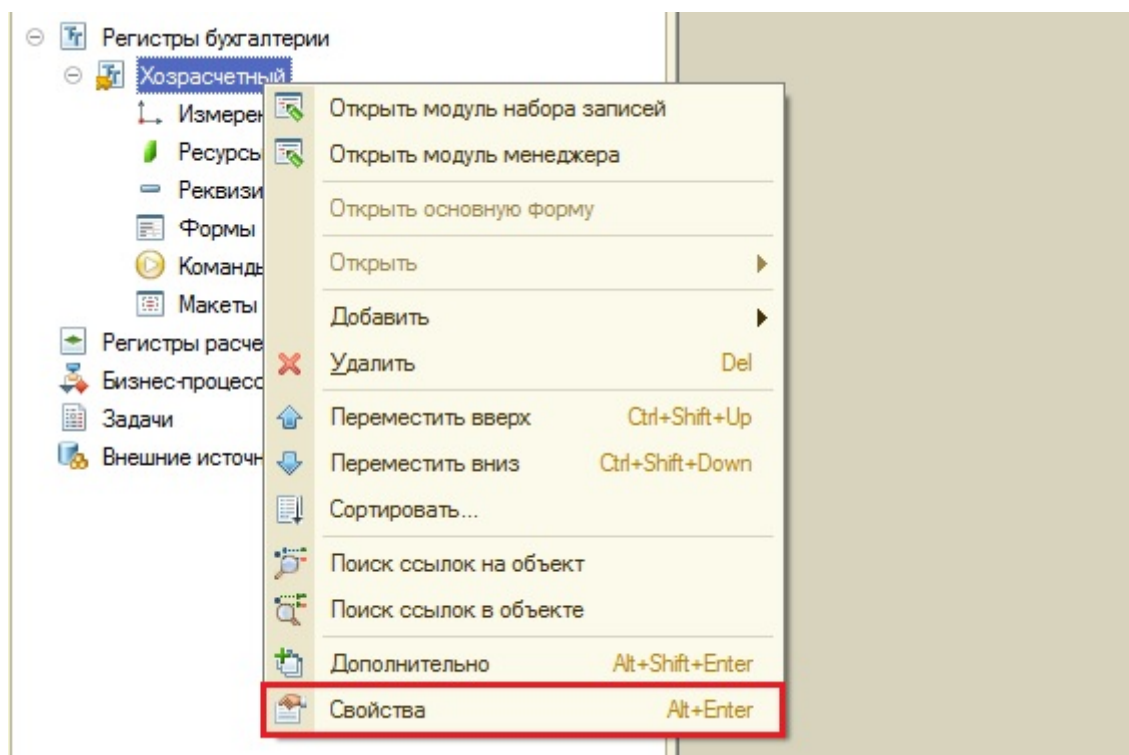


Рис. 3.7. Заходим в свойства заимствованного регистра

У свойства «ПланСчетов» (категория «Данные») поставим флаг, а после этого выберем план счетов «Хозрасчетный» (см. рис. 3.8).

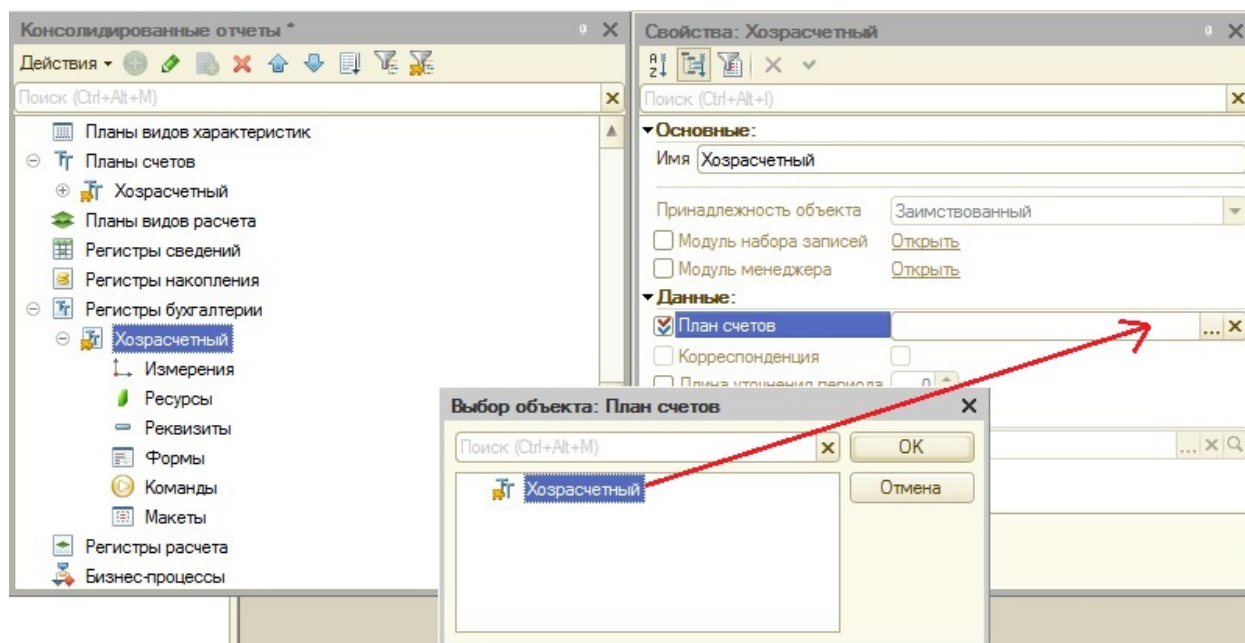


Рис. 3.8. Привязка плана счетов к регистру бухгалтерии в расширении

После того, как Вы привяжете план счетов к регистру, флаг «План счетов» можно снять (см. рис. 3.9).

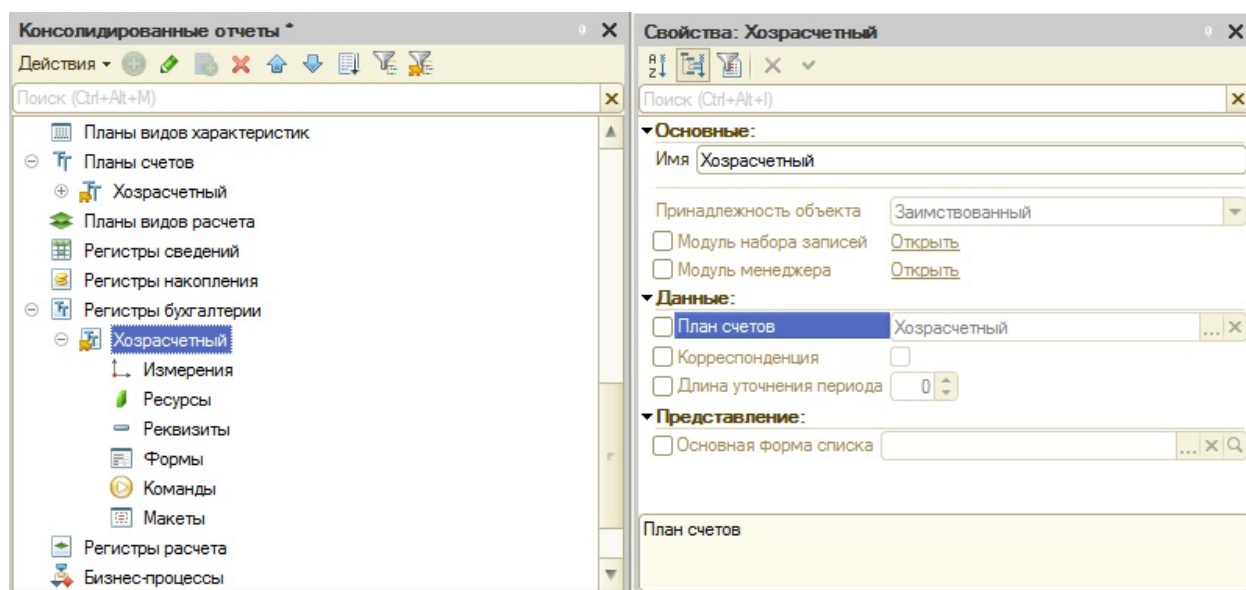


Рис. 3.9. После привязки плана счетов к регистру бухгалтерии

Свойства заимствованных объектов будем проходить в следующей главе (см. стр. 35).

Так же у регистра заимствуем ресурс «Сумма» и измерение «Организация» (см. рис. 3.10).

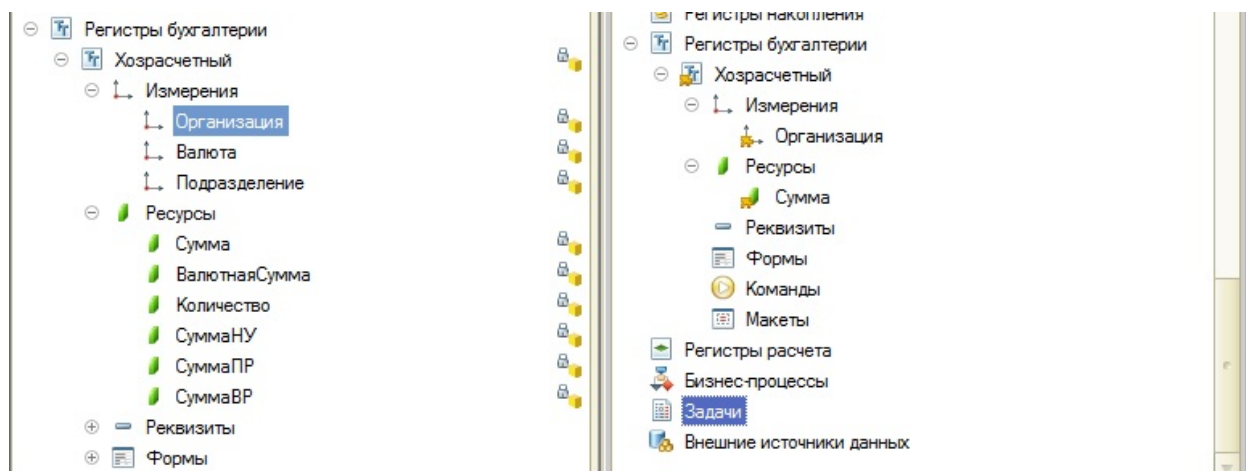


Рис. 3.10. Заимствовали измерение и ресурс

Зайдем опять в конструктор запросов схемы компоновки данных отчета «Приход по кассам» (см. рис. 3.11).

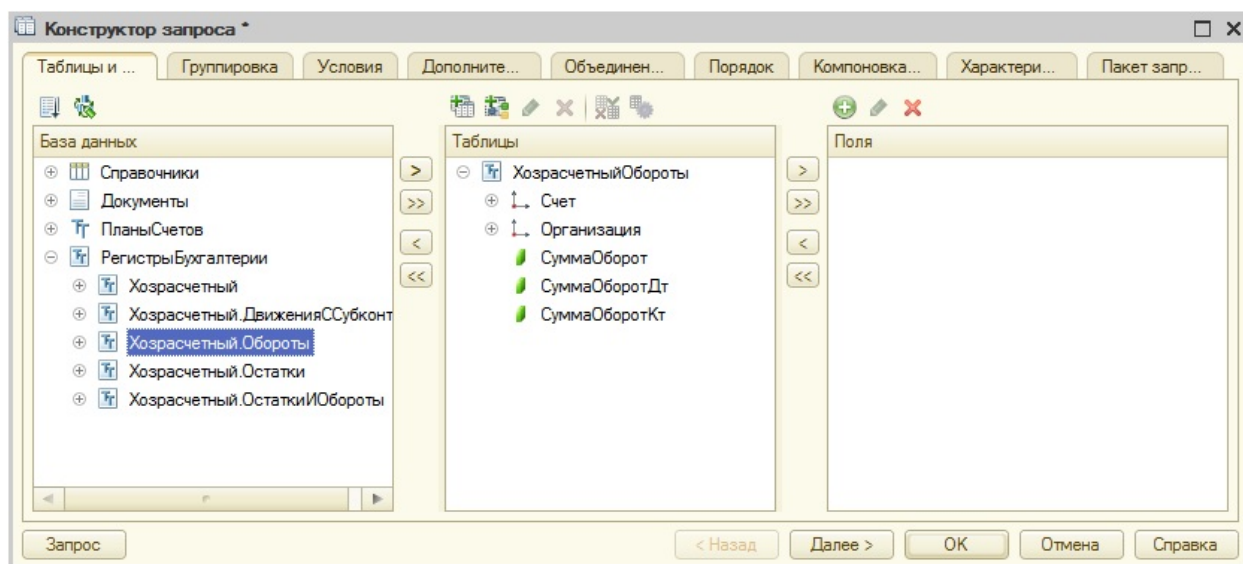


Рис. 3.11. Регистр бухгалтерии в конструкторе запросов

И видим, что у нас появился счет (можно сделать условие по счету), сумма, но нет субконто. Чтобы появилось субконто, нужно к плану счетов в расширении привязать план видов характеристик, который привязан к плану счетов в основной конфигурации. Но сначала нужно его добавить.

Смотрим, какой план видов характеристик у плана счетов основной конфигурации. Это план видов характеристик «ВидыСубконтоХозрасчетные» (см. рис. 3.12).

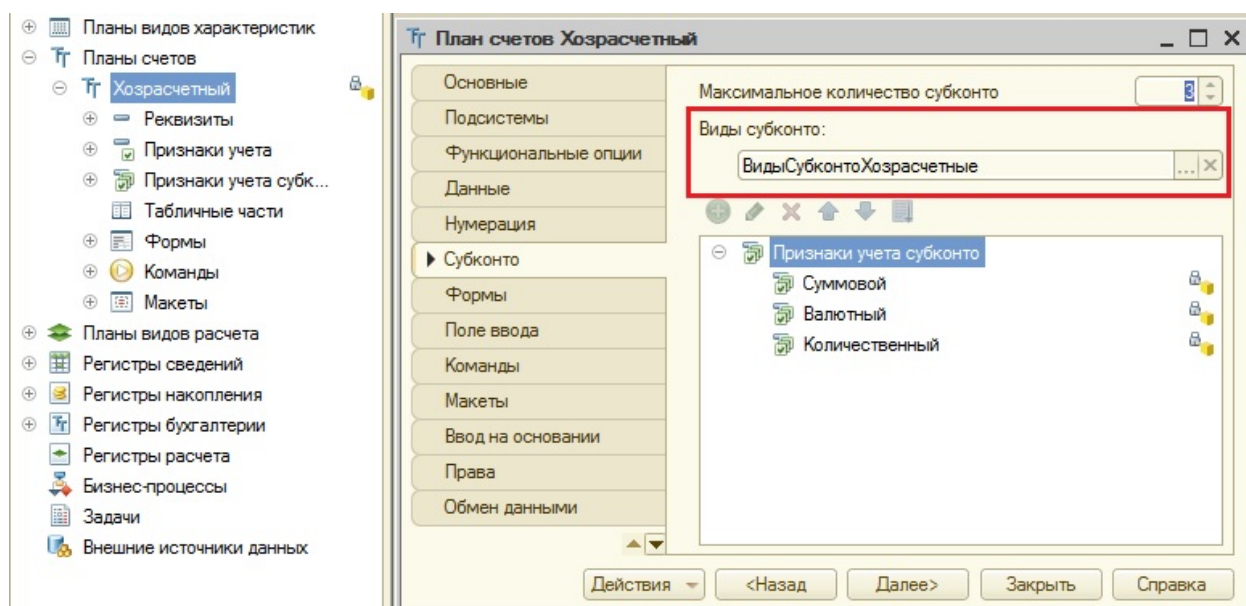


Рис. 3.12. Вид субконто у плана счетов «Хозрасчетный»

Заимствуем его (см. рис. 3.13).

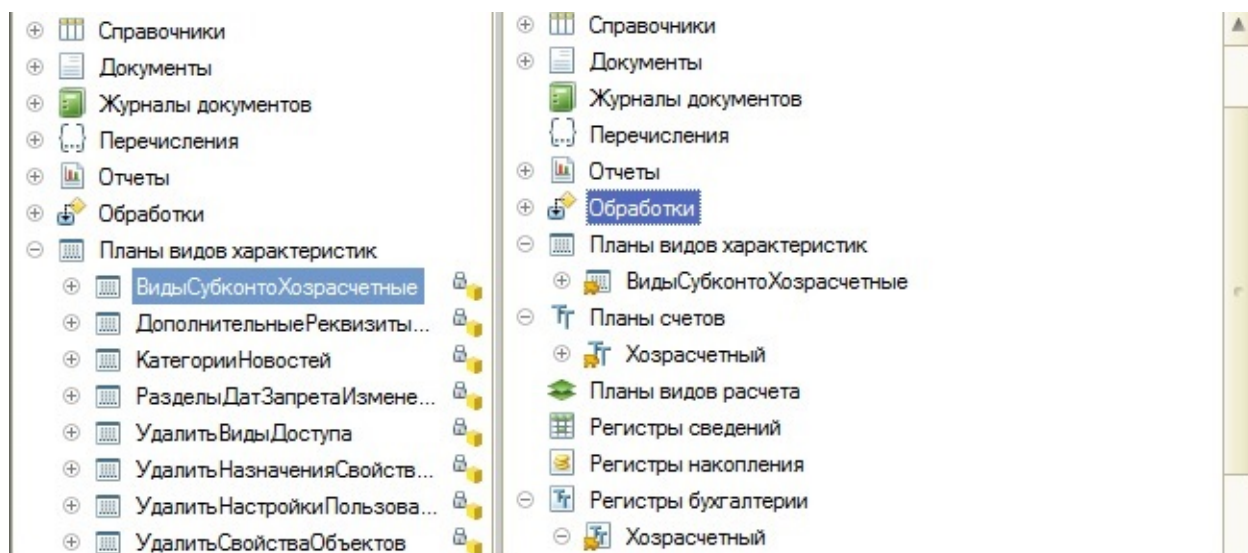


Рис. 3.13. Заимствовали план видов характеристик «Виды субконто хозрасчетные»

Теперь привяжем заимствованный план видов характеристик к нашему плану счетов. Для этого зайдём в свойства плана счетов «Хозрасчетный», поставим флаг рядом со свойством «Виды субконто» и выберем заимствованный план видов характеристик (см. рис. 3.14).

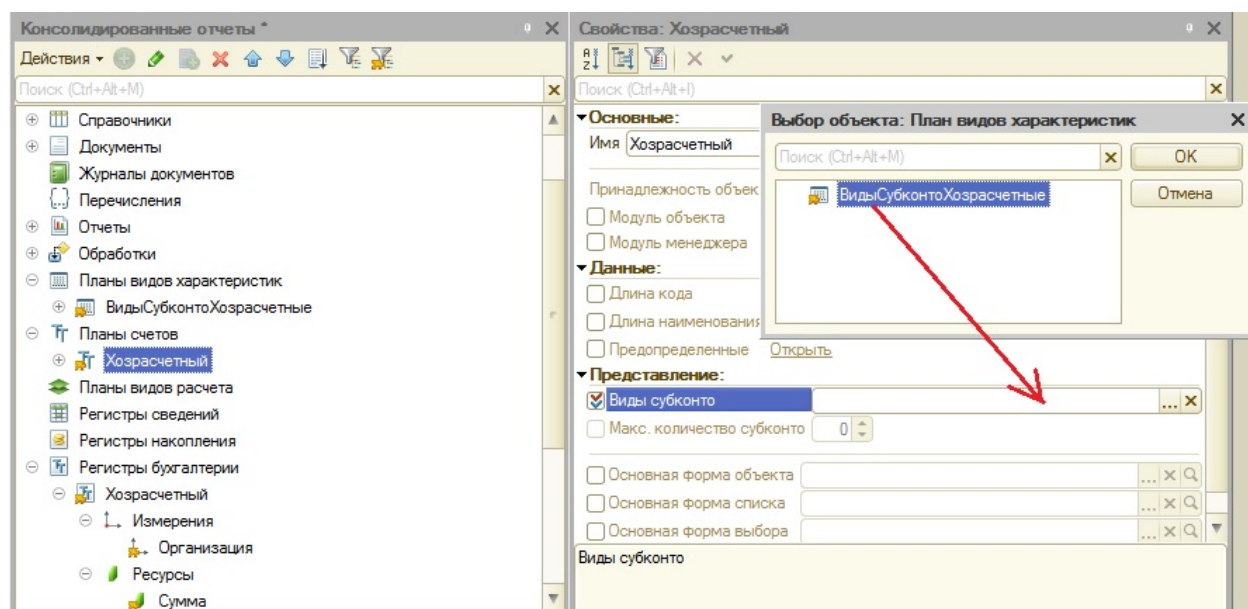


Рис. 3.14. Выбор вида субконто

После выбора плана видов характеристик флаг «Виды субконто» снимем (см. рис. 3.15).

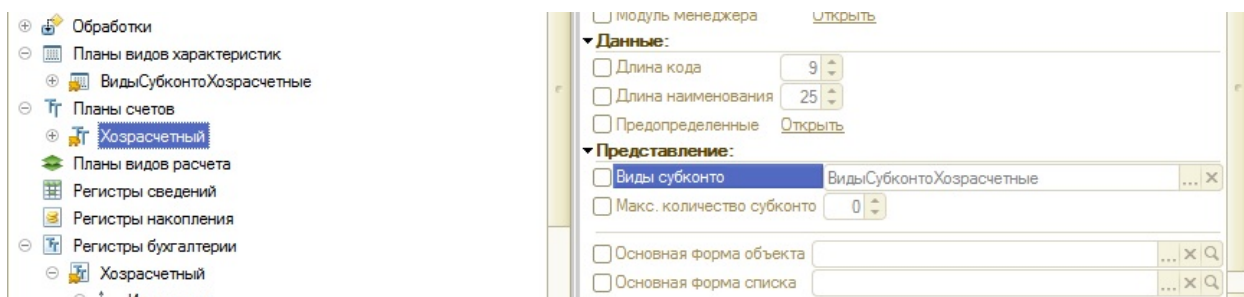


Рис. 3.15. Свойства плана счетов, после выбора субконто

Таким же образом установим максимальное количество субконто, как и у плана счетов в основной конфигурации – три (см. рис. 3.16).

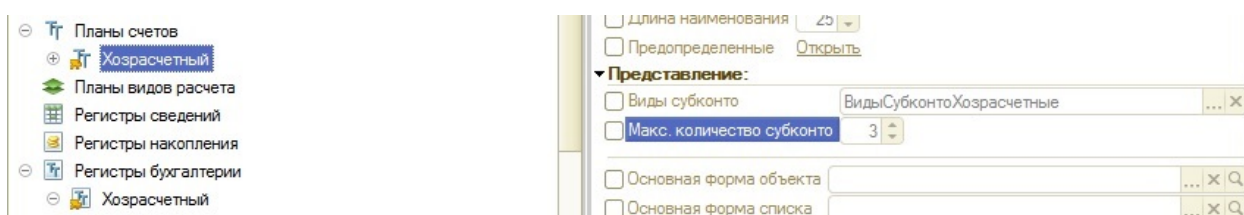


Рис. 3.16. Свойство «Макс. количество субконто»

Теперь зайдем в схему компоновки отчета «Приход по кассам» и попробуем сделать запрос, который был у нас в консоли (см. рис. 3.1).

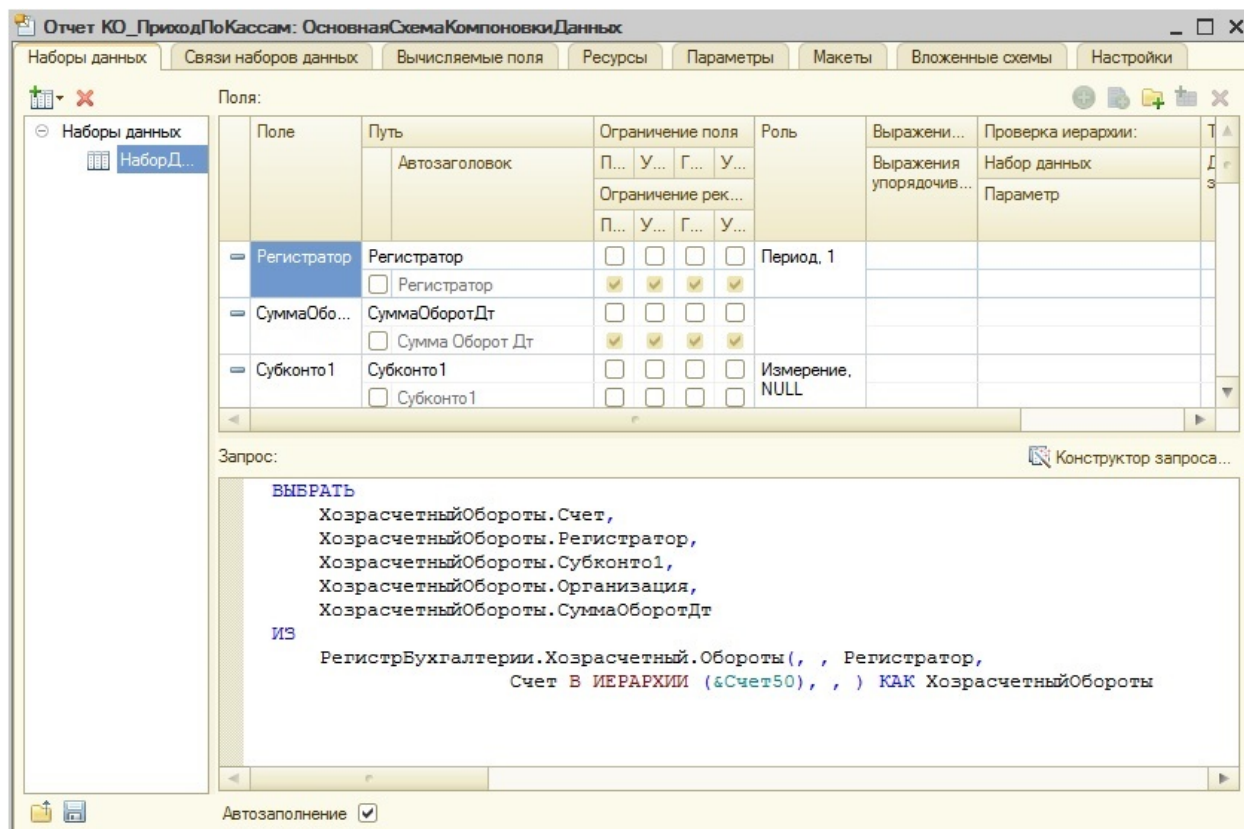


Рис. 3.17. Нужный запрос в СКД

Сделаем ресурсом поле «СуммаОборотДт»

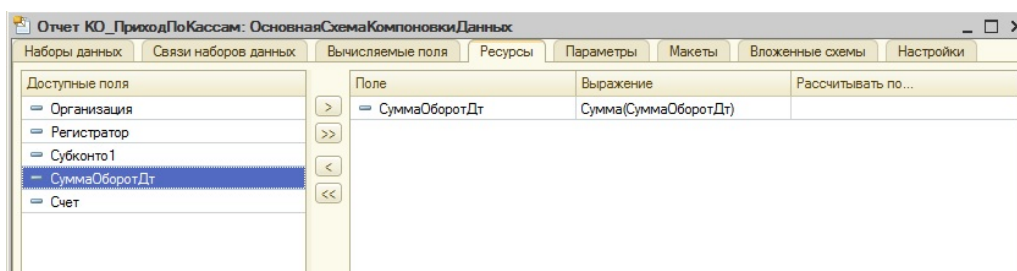


Рис. 3.18. Добавили поле «СуммаОборотДт» в ресурс

И нам стало задано параметр «Счет50» (см. рис. 3.19).

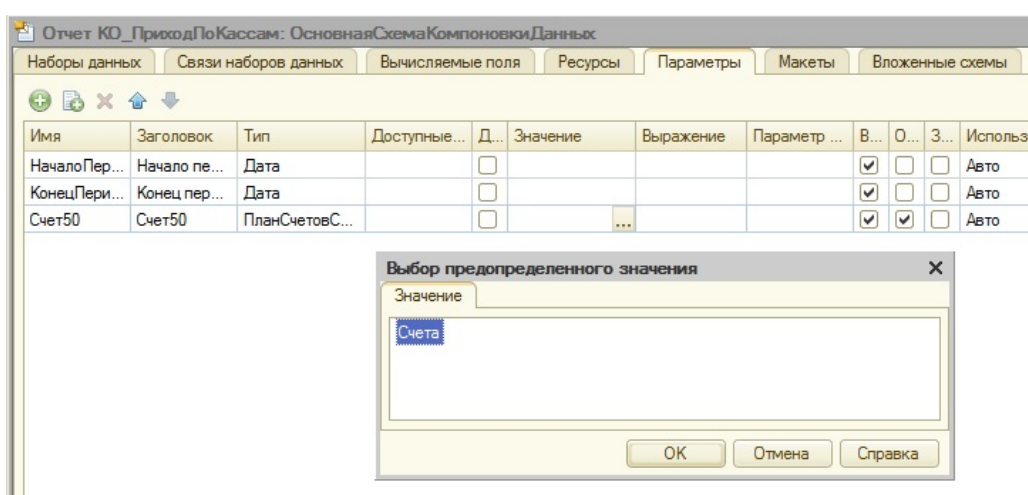


Рис. 3.19. Нужно добавить в параметр счет.

И мы видим, что нет счета, потому что мы не заимствовали счета. Надо это сделать.

Откроем predeterminedенные счета плана счетов «Хозрасчетный» основной конфигурации и заимствуем все 50-е счета (выделяем нужный счет и нажимаем кнопку «Заимствовать» панели predeterminedенных счетов, см. рис. 3.20).

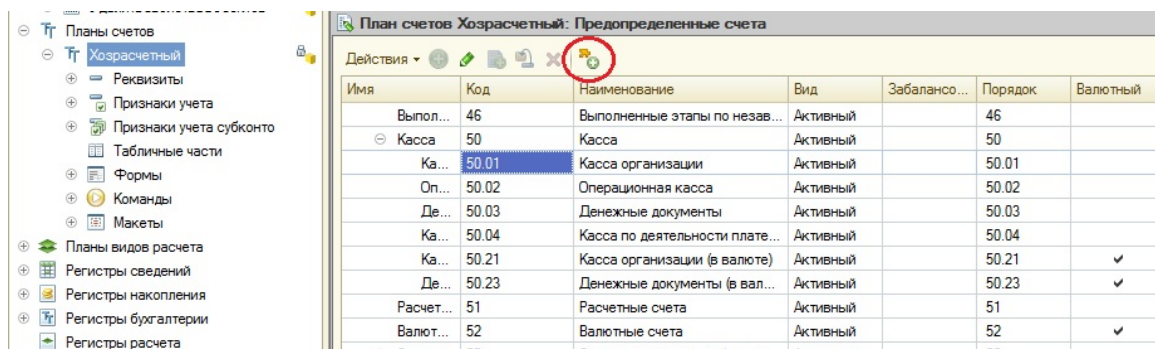


Рис. 3.20. Заимствование predeterminedенных счетов

Я заимствовал все субсчета 50-го счета. Их можно посмотреть, если зайти в свойства плана счетов и нажать на гиперссылку «Открыть» свойства «Предопределенные» (см. рис. 3.21).

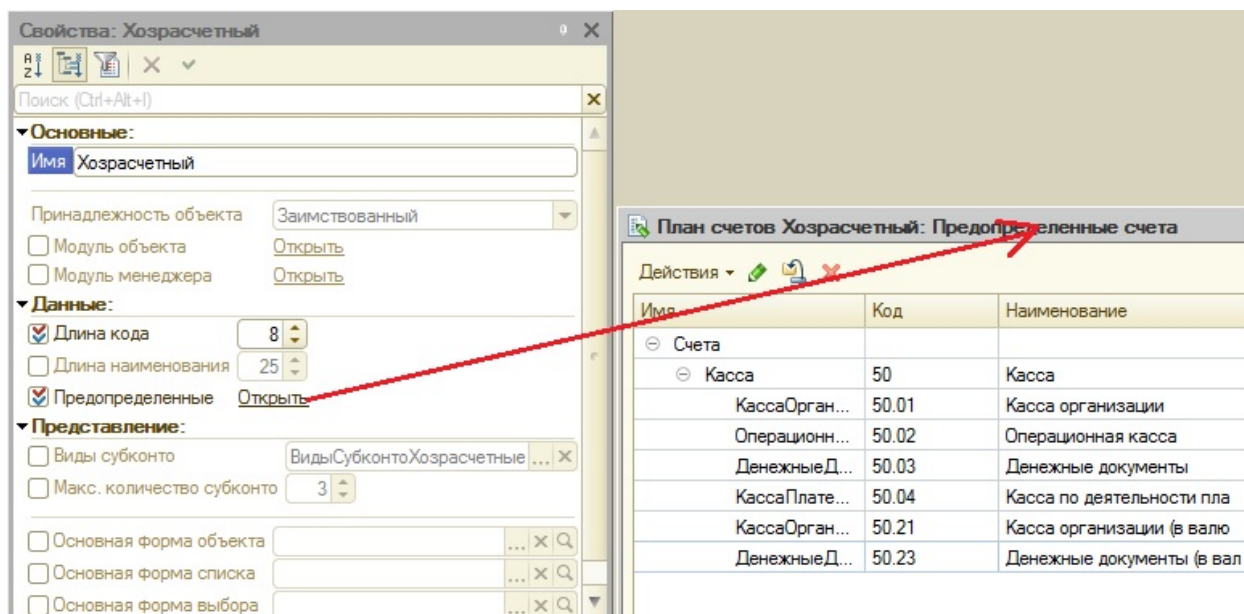


Рис. 3.21. Предопределенные счета заимствованного плана счетов

Флаг со свойства «Предопределенные» снимем, он нам е нужен.

Вернемся обратно в СКД отчета «Приход по кассам», и сейчас можно спокойно выбрать любой из заимствованных счетов. Мы выберем счет «Касса» (см. рис. 3.22).

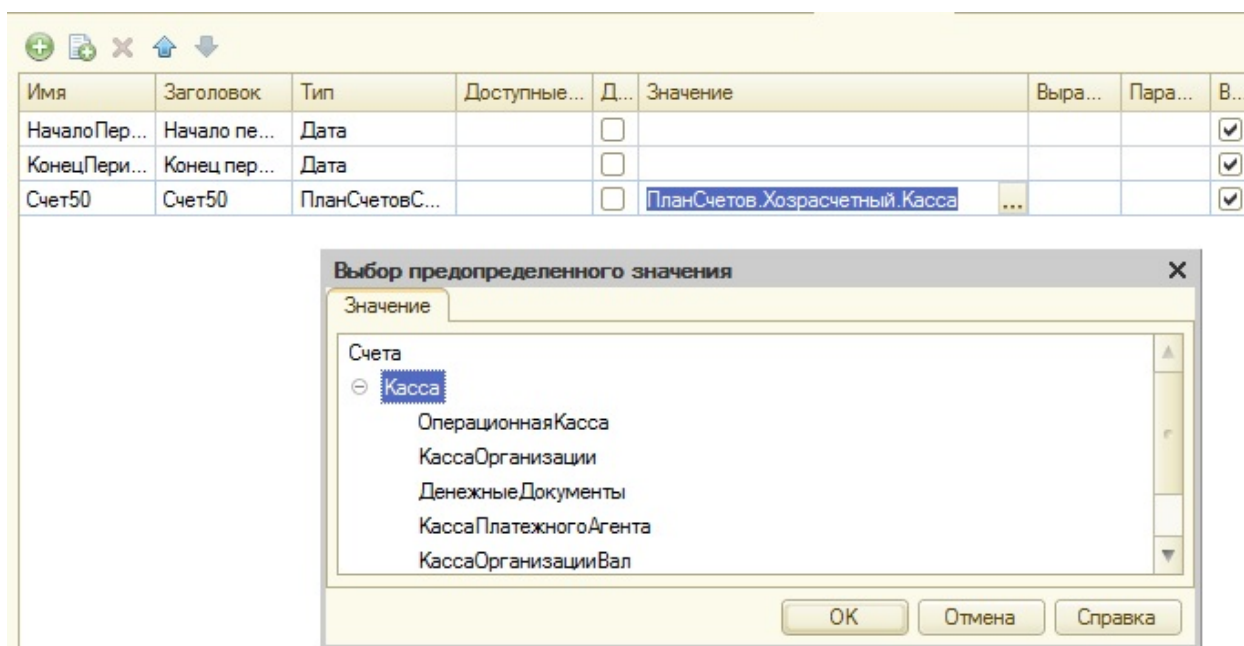


Рис. 3.22. Предопределенные счета для выбора

Нам осталось сделать период (см. рис. 3.23).

Имя	Заголовок	Тип	Д...	Д...	Значение	Выражение	Па...	В...	О...
НачалоПер...	Начало пе...	Дата		<input type="checkbox"/>		НачалоПериода(&Период.ДатаНачала,"День")		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
КонецПери...	Конец пер...	Дата		<input type="checkbox"/>		КонецПериода(&Период.ДатаОкончания,"День")		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Счет50	Счет50	ПланСчетовС...		<input type="checkbox"/>	ПланСчетов...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Период	Период	Стандартный...		<input type="checkbox"/>				<input checked="" type="checkbox"/>	<input type="checkbox"/>

Рис. 3.23. Задали параметр «Период».

Также поле «Субконто1» назовем «СДДС» – «Статья движения денежных средств». А поле «Сумма оборот дт» - «Сумма прихода».

Поле	Путь	Ограничение по...	Роль	Выражени...	Проверка иерархии:
Регистратор	Регистратор	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Период. 1		
СуммаОборотДт	СуммаОборотДт	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>			
Субконто1	Субконто1	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Измерение, NULL		
Субконто2	Субконто2	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>			
Счет	Счет	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Измерение		

Рис. 3.24. Изменили заголовки полей

Теперь сделаем какую-нибудь настройку (см. рис. 3.25).

Отчет КО_ПриходПоКассам: ОсновнаяСхемаКомпоновкиДанных

Наборы д... | Связи наб... | Вычисляе... | Ресурсы | Параметры | Макеты | Вложенн... | Настройки

Имя вари... | Представ

Основной | Основной

Отчет

- Организация
- Субконто1
- <Детальные записи>

Настройки: Отчет | Отчет

Параме... | Выбра... | Отбор | Сортир... | Условн... | Пользо... | Другие...

Отображать недоступные параметры (устанавливать значения недоступных парамет...

Параметр	Значение
Дата начала	Дата окончания
<input type="checkbox"/> Период	Произвольный период

Рис. 3.25. Настройка отчета «Приход по кассам»

Всё, включим наш новый отчет в подсистему «Консолидированная отчетность», сохраним расширение, обновим базу и посмотрим, как работает наш новый отчет (см. рис. 3.26).

Основной

Период: 01.01.2013 - 31.12.2013

Сформировать Настройки... Варианты отчета Найти...

Параметры: Период: 01.01.2013 - 31.12.2013

Организация	Документ	Счет	Сумма прихода
Комфорт-сервис			190 000,00
	Поступление наличных КС00-000001 от 04.02.2013 12:00:00	50.01	70 000,00
	Поступление наличных КС00-000002 от 04.03.2013 12:00:00	50.01	120 000,00
Конфетпром			513 480,00
	Билет авиа Москва-Санкт-Петербург-Москва		12 000,00
	Поступление денежных документов КП00-000001 от 10.03.2013 12:00:01	50.03	12 000,00
	Получение наличных в банке		501 480,00
	Поступление наличных КП00-000004 от 09.03.2013 12:00:00	50.01	152 250,00
	Поступление наличных КП00-000001 от 09.01.2013 11:52:58	50.01	150 000,00
	Поступление наличных КП00-000002 от 05.02.2013 12:00:01	50.01	46 980,00
	Поступление наличных КП00-000003 от 09.02.2013 12:00:00	50.01	152 250,00
Магазин 23			181 000,00
	Получение наличных в банке		43 500,00
	Поступление наличных МН00-000002 от 09.02.2013 12:00:01	50.01	43 500,00
	Приход денежных средств розничная выручка		64 200,00
	Отчет о розничных продажах МН00-000003 от 12.02.2013 12:00:00	50.01	52 500,00
	Поступление наличных МН00-000003 от 10.02.2013 11:00:00	50.01	11 700,00
	Розничная выручка от НТТ		73 300,00
	Поступление наличных МН00-000001 от 26.01.2013 12:00:01	50.01	73 300,00

Рис. 3.26. Отчет «Приход по кассам»

Всё, отчет получился. В следующей главе мы будем более подробно разбирать работу с заимствованными реквизитами.

Глава 4. Контролируемые и модифицируемые свойства

В прошлых главах мы научились заимствовать объекты в расширение и сделали простой отчет, в котором вывели данные по документам, также сделали сложный отчет, информацию в котором брали с регистров бухгалтерии.

В этой главе мы разберем свойства заимствованных объектов.

Откроем наше расширение, и зайдем в палитру свойств какого-нибудь объекта, например, в свойства документа «Поступление товаров и услуг» (см. рис. 4.1).

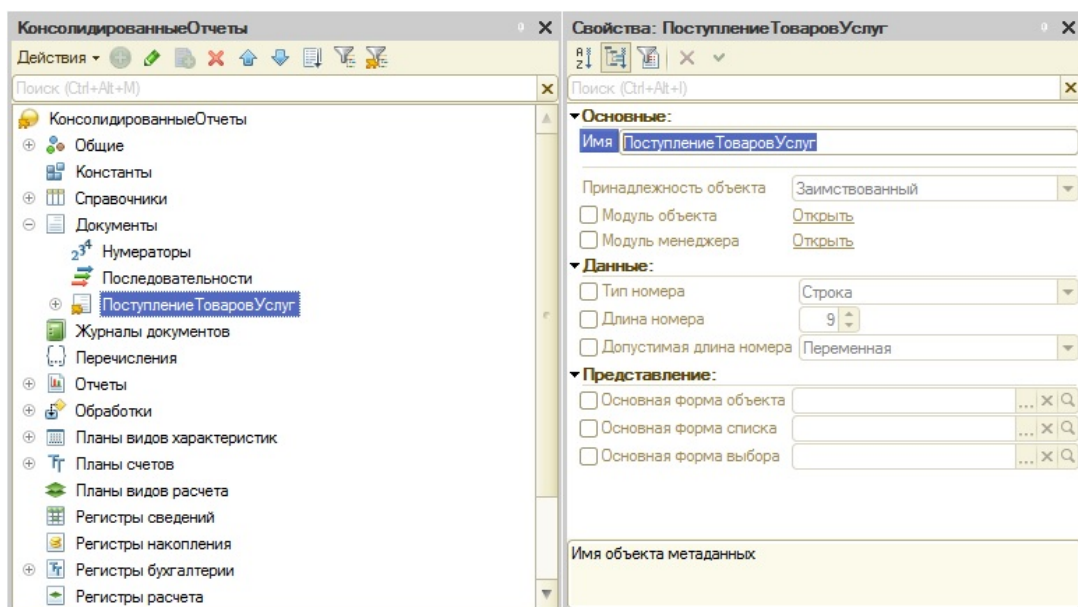


Рис.4.1. Палитра свойств заимствованного документа «Поступление товаров и услуг»

Смотрите, свойств у заимствованного документа гораздо меньше, чем свойств документа расширяемой конфигурации. Откроем палитру свойств документа «Поступление товаров и услуг» и посмотрим (см. рис. 4.2).

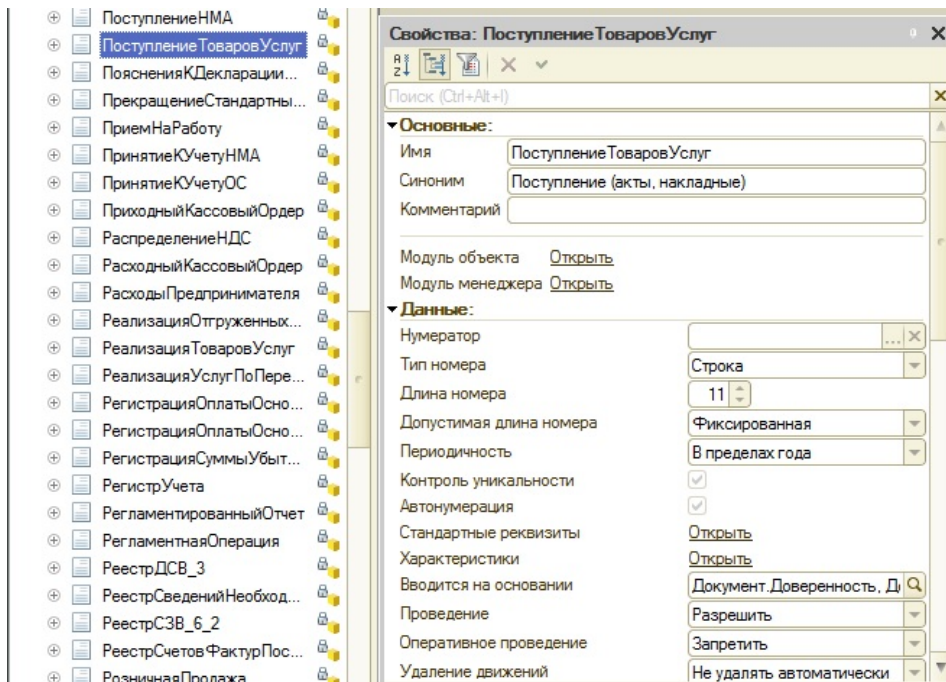


Рис. 4.2. Палитра свойств документа «Поступление товаров и услуг» расширенной конфигурации

Почему так? Потому что у объекта присутствуют только те свойства, которые могут быть *или контролируемыми, или модифицируемыми*.

Все свойства, которые показаны на рисунке 4.1, контролируемые или модифицируемые. Чем они отличаются, узнаем позже, а пока посмотрим, как их отличить.

Контролируемое свойство имеет вот такую пиктограмму (см. рис. 4.3).

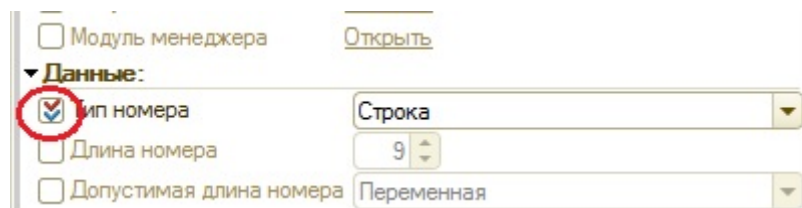


Рис. 4.3. Пиктограмма контролируемого свойства

А модифицированное свойство - вот такую пиктограмму (см. рис. 4.4.).

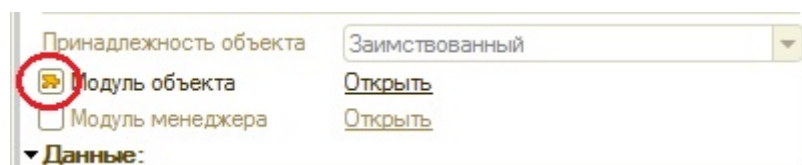


Рис. 4.4. Пиктограмма модифицированного свойства

Что такое контролируемое свойство? Если этот флаг установлен, это значит, что тип данного свойства будет контролироваться при подключении расширения. И если он не совпадает с типом расширяемой конфигурации, то расширение не будет подключено.

Например, на рисунке 4.3 мы установили контролировать свойство «Тип номера», у этого свойства можно выбрать два типа - или число, или строка (см. рис. 4.5).

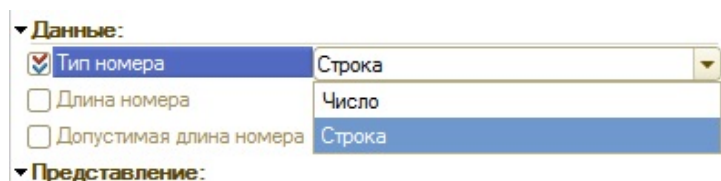


Рис. 4.5. Возможные значения свойства «Тип номера»

Если мы так и оставим тип номера – «Строка», то наше расширение подключится нормально, а если изменим тип номера на «Число» (см. рис. 4.6):

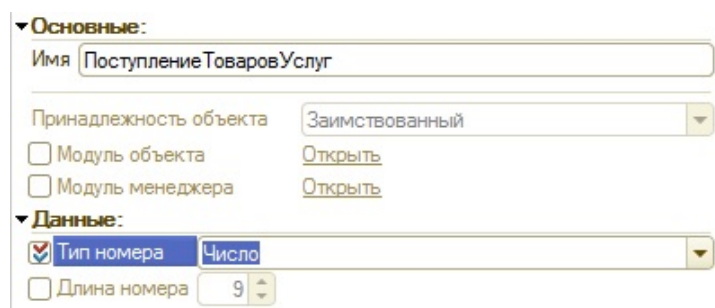


Рис. 4.6. Значение «Число» свойства «Тип номера»

- то, во-первых, выйдет ошибка при обновлении базы данных (см. рис. 4.7).

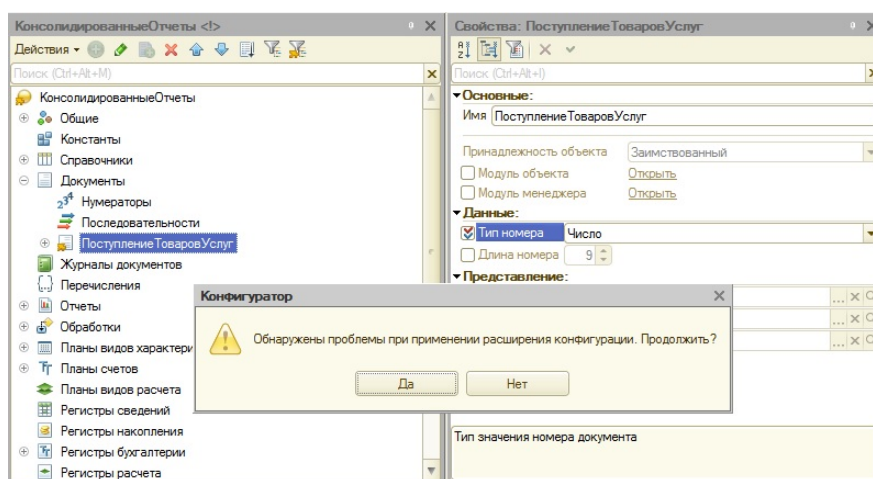


Рис. 4.7. Ошибка при обновлении базы данных

Но самое главное: расширение не подключится, когда мы запустим «1С: Предприятие» в пользовательском режиме, просто выйдет «1С: Предприятие» без нашего нового раздела «Консолидированная отчетность» (см. рис. 4.8).

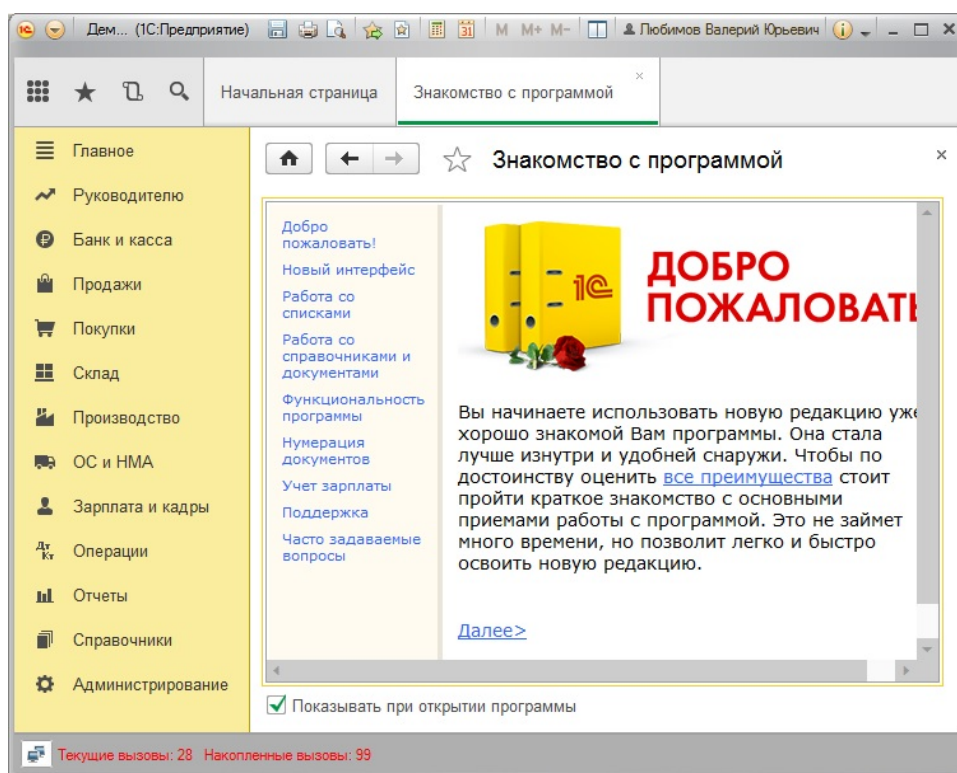


Рис. 4.8. «1С: Предприятие» без подключенного расширения

Теперь вернем обратно для свойства «Тип номера» значение *Строка* (см. рис. 4.9) и опять запустим расширение (см. рис. 4.10).

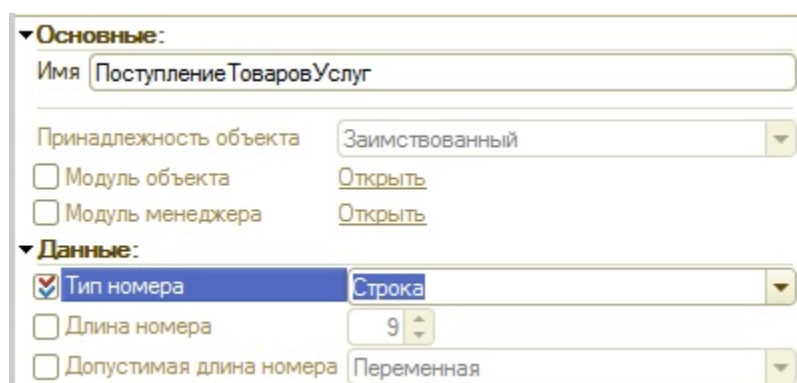


Рис. 4.9. Значение «Строка» свойства «Тип номера»

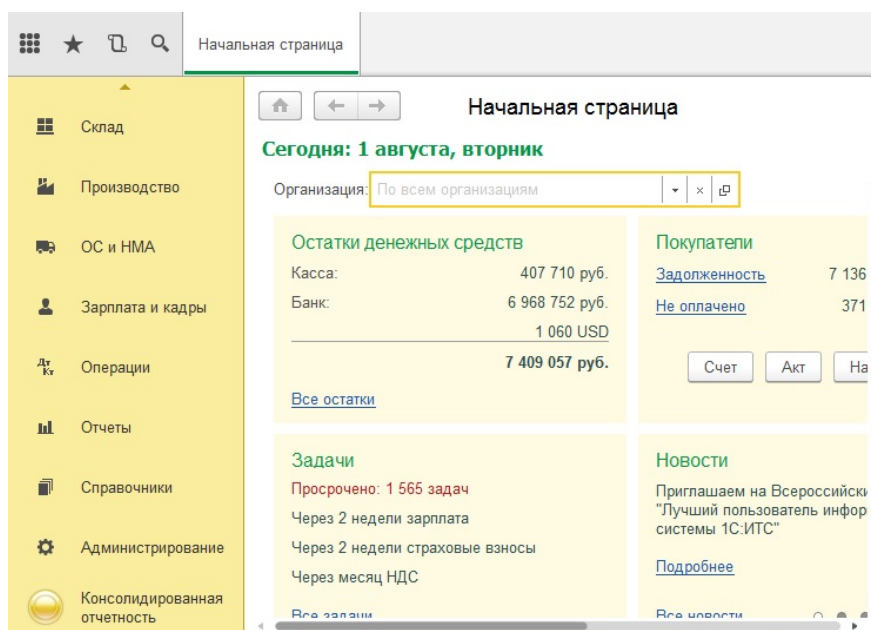


Рис. 4.10. 1С: Предприятие с подключенным расширением

В этот раз расширение подключилось нормально.

Также обращаю внимание, что имя объекта (см. рис. 4.11) тоже является *контролируемым* свойством. Поскольку платформа при подключении расширения сверяет объекты по имени, **поэтому – воздержитесь от изменения имен заимствованных объектов.**

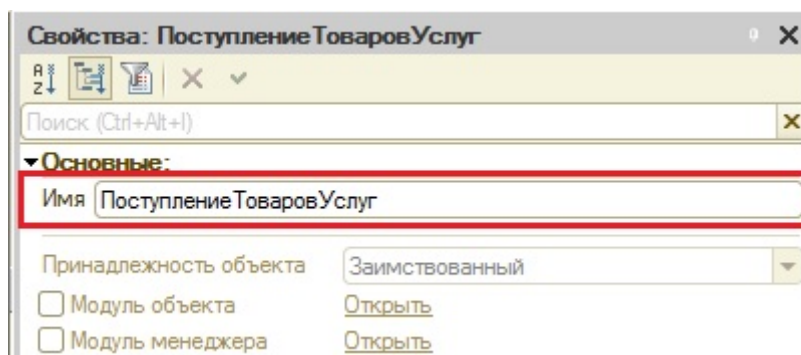


Рис. 4.11. Свойство «Имя» документа «Поступление товаров и услуг»

Следующий вид свойства - это модифицируемые свойства. Как правило, это формы и модули объекта. С модулями мы будем разбираться в следующей главе, а пока посмотрим, как у нас можно модифицировать форму списка документа «Поступление товаров и услуг». Для этого создадим в расширении собственную форму списка документа «Поступление товаров», где будут следующие поля – «Организация», «Контрагент» и «Сумма документа» (заимствуем эти поля, см. рис. 4.12).

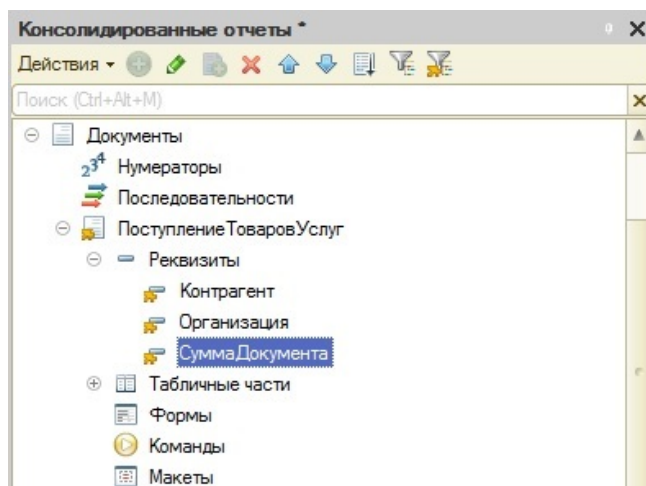


Рис. 4.12. Заимствованные реквизиты документа «Поступление товаров и услуг»

Добавим новую форму документа в расширении конфигурации (см. рис. 4.13).

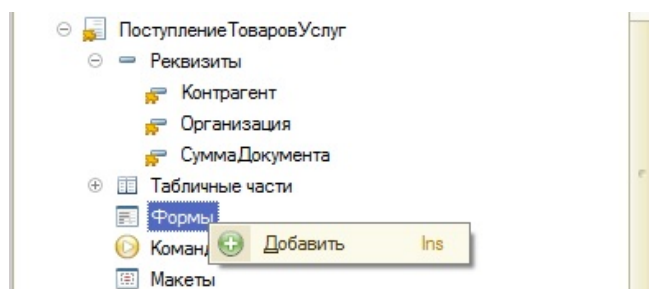


Рис. 4.13. Добавление новой формы документа

В открывшемся конструкторе установим переключатель на значение «Форма списка» и уберем признак «Назначить форму основной» (см. рис. 4.14).

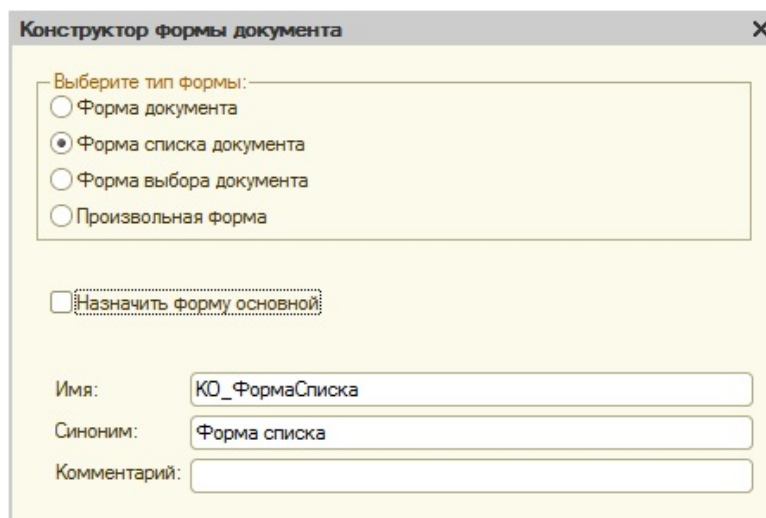


Рис. 4.14. Первая страница конструктора создания форм

На следующей странице конструктора форм отметим дополнительно поля «Контрагент», «Организация» и «Сумма документа» (см. рис. 4.15).

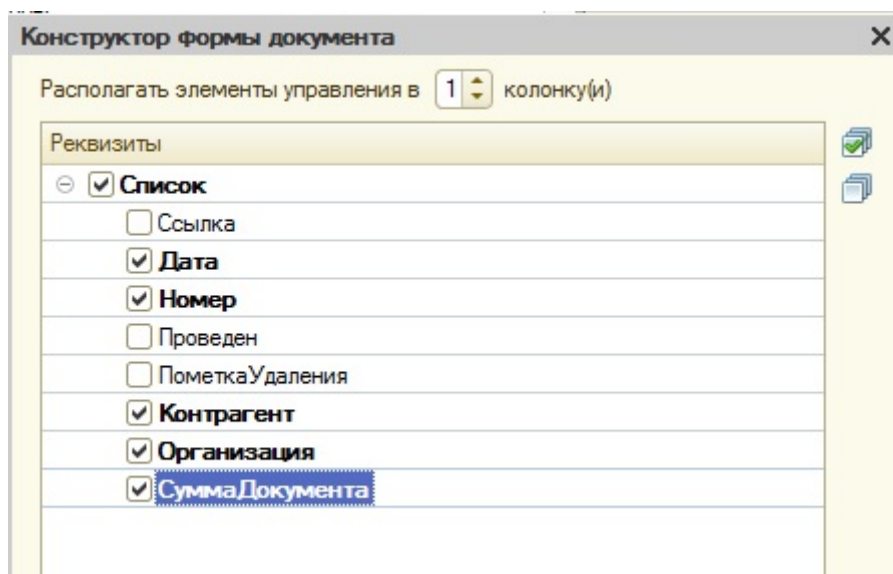


Рис. 4.15. Вторая страница конструктора создания форм

Нажмем кнопку «Готово» конструктора, и будет создана форма списка документа «Поступление товаров и услуг», которая существует только в расширении (см. рис. 4.16).

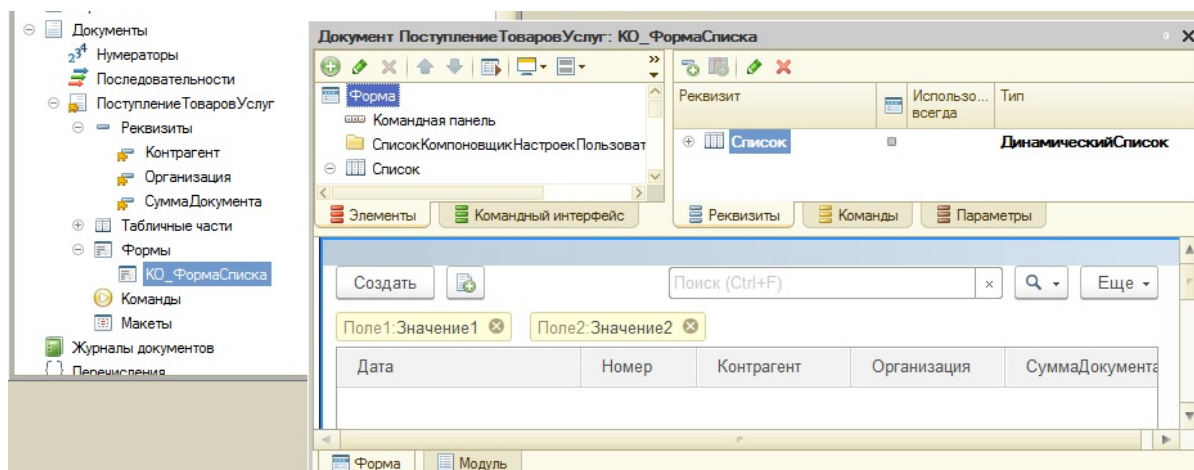


Рис. 4.16. Форма списка документа «Поступление товаров и услуг» в расширении конфигурации

Форму списка мы создали, я специально не назначил её основной, теперь перейдем обратно в свойства документа «Поступления товаров и услуг», и в свойстве «Форма списка» выберем нашу вновь созданную форму списка (см. рис. 4.17, 4.18).

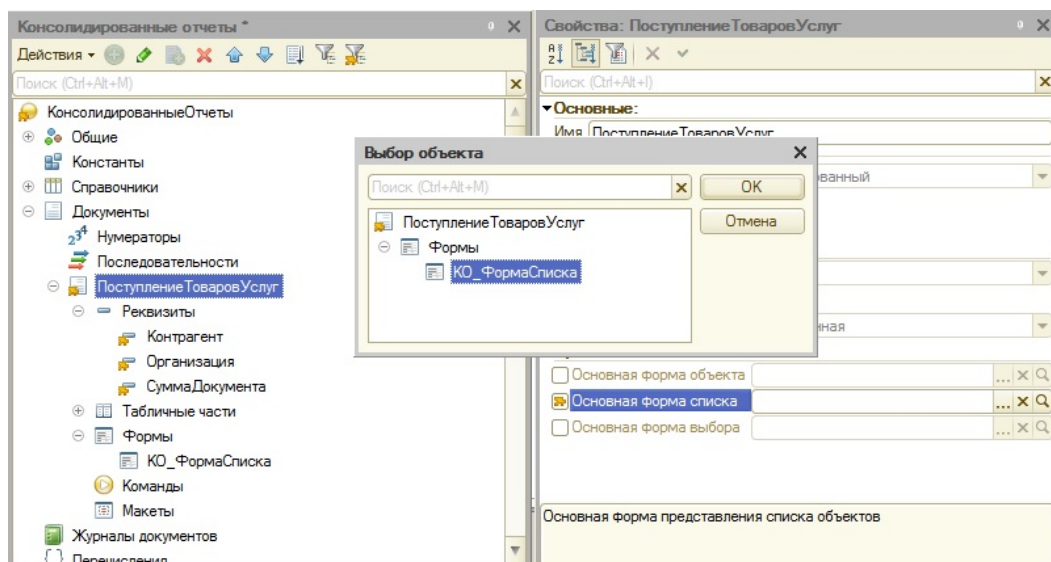


Рис. 4.17. Выбор формы списка в свойство «Основная форма списка»

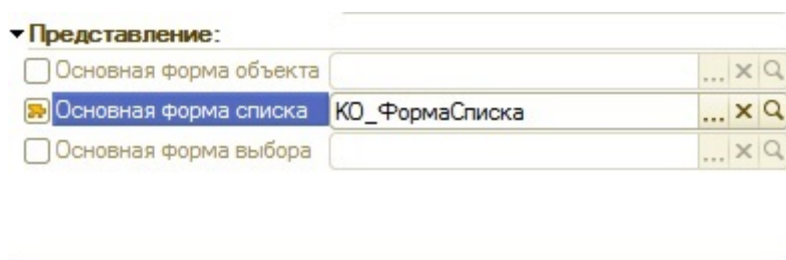


Рис. 4.18. После выбора формы списка в свойство «Основная форма списка»

Самое интересное, если вы установите флаг модифицированности свойства, но ничего не выберете, то в свойстве будет значение расширяемой (основной) конфигурации.

Замечу, что указать таким образом собственную форму расширения в модифицированном свойстве можно только тогда, когда режим совместимости расширения и режим совместимости конфигурации выше версии 8.3.7.

Сохраним расширение, обновим базу данных и запустим «1С: Предприятие». Зайдем в форму списка документа «Поступление товаров и услуг» (см. рис. 4.19).

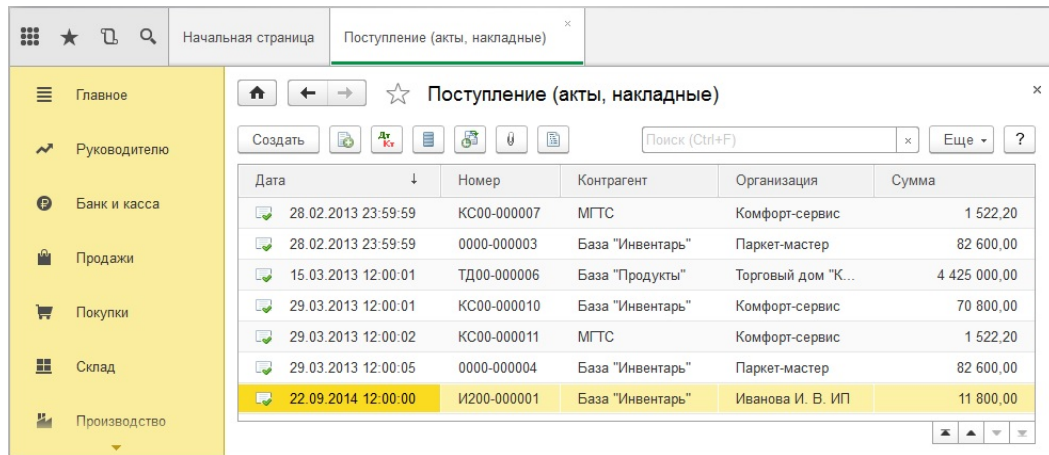


Рис. 4.19. Форма списка документа «Поступление...» из расширения конфигурации

Как видно из рисунка 4.19, форма стала такой, какой мы её сделали в расширение. Таким образом, с помощью модифицированных свойств можно переделывать интерфейс форм «1С: Предприятия» на свое усмотрение.

Рассмотрим несколько интересных моментов по работе с расширениями.

Вы заметили, что в расширении много объектов, которые подтянулись в качестве вспомогательных (т.е. мы их сознательно не добавляли из основной конфигурации). Можно ли отобразить только те объекты, которые или добавлены, или изменены? Да, можно. Для этого в панели расширения есть кнопка «Измененные или добавленные в расширение». Если её нажать, то останутся только те объекты, которые непосредственно участвуют в расширении (см. рис. 4.20).

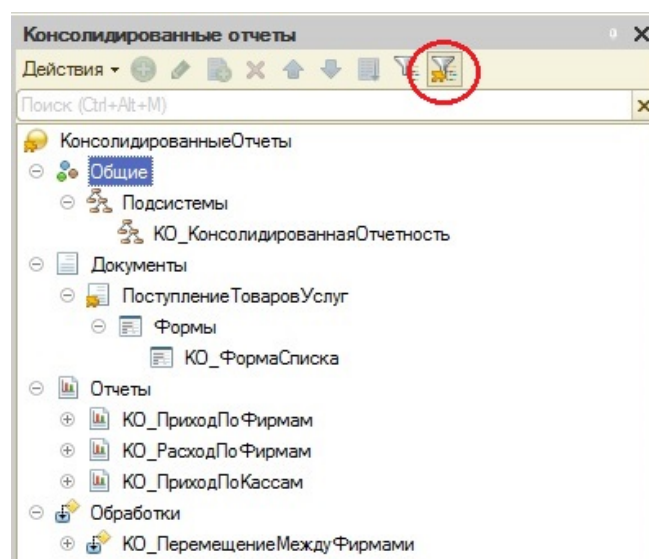


Рис. 4.20. После нажатия на кнопку «Измененные или добавленные в расширение»

Рассмотрим еще один практический момент. Во второй главе мы сделали отчет «Приход по всем фирмам», но нашим бухгалтерам не понравилось, что он расположен где-то в каком-то новом разделе. Они захотели, чтобы отчет был в разделе «Покупки». Сможем ли мы его туда поместить?

Да, и очень просто, но для начала надо заимствовать подсистему «Покупки» (см. рис. 4.21).

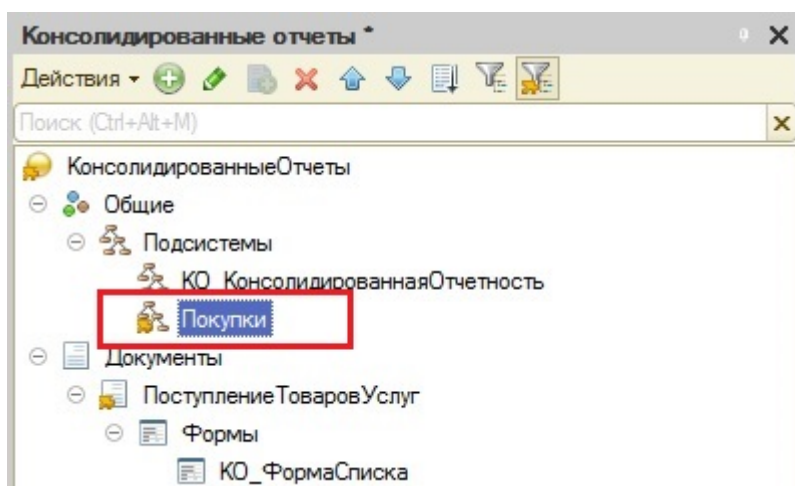


Рис. 4.21. Заимствовали подсистему «Покупки»

Теперь включим отчет «Приход по всем фирмам» в заимствованную подсистему (см. рис. 4.22).

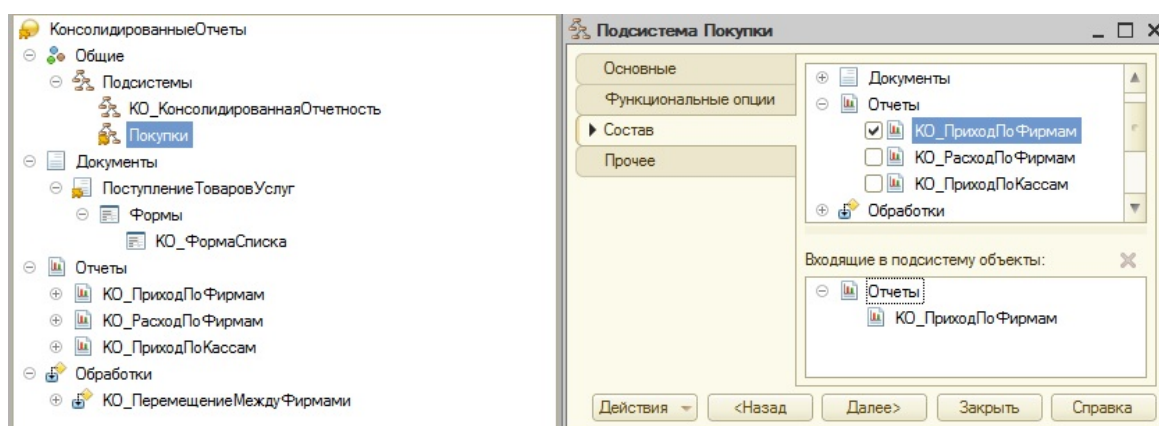


Рис. 4.22. Включили отчет «Приход по всем фирмам» в заимствованную подсистему

И всё. Посмотрим на всякий случай, куда в командном интерфейсе он попал: сохраним конфигурацию, обновим базу данных и перейдем в раздел «Покупки» (см. рис. 4.23).

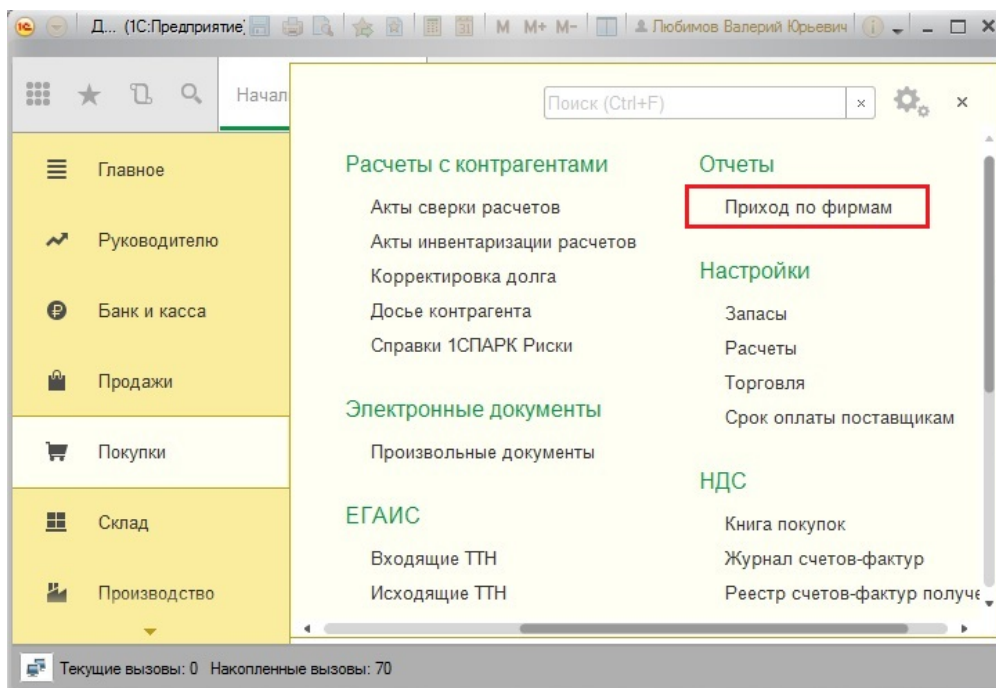


Рис. 4.23. Отчет «Приход по всем фирмам» в подсистеме «Получки»

Все получилось, как мы и хотели.

На этом я закончу 4-ю главу. В следующей главе мы будем разбираться с модулями.

Глава 5. Работа с программными модулями

В прошлых главах мы научились заимствовать объекты в расширение, сделали простой отчет, в котором вывели данные по документам, и сделали сложный отчет, информацию в котором брали с регистров бухгалтерии. А также изучили свойства заимствованных объектов.

В этой главе мы разберем работу с программными модулями. В расширение можно заимствовать практически все модули, кроме глобальных. Замечу, что модули объектов заимствуются вместе с объектами. И в них доступны все переменные и методы, которые есть в расширенных модулях.

Начиная с платформы 8.3.9, в расширениях появилась новая возможность: перехватывать методы объектов. А именно: выполнять код перед каким-то типовым методом, после этого метода и даже замещать выполнение этого метода собственным кодом. Замечу, что данная возможность работает только тогда, когда у расширяемой конфигурации свойство «РежимСовместимости» установлено в «Не использовать» (см. рис. 5.1).

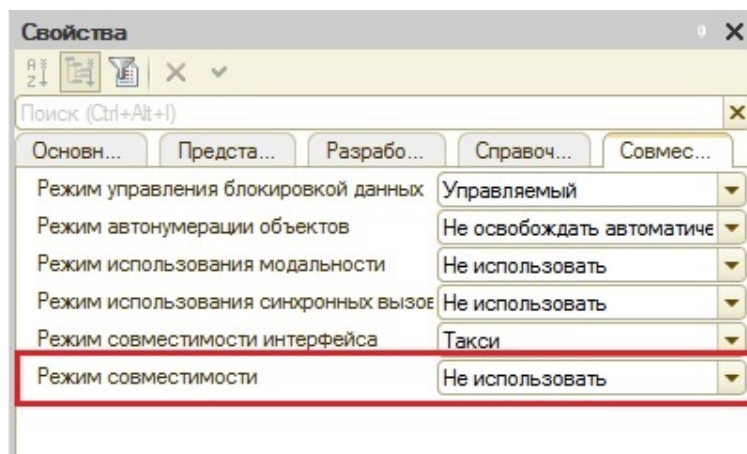


Рис. 5.1. Режим совместимости основной конфигурации

Точно такой же режим совместимости должен быть у расширения (см. рис. 5.2).

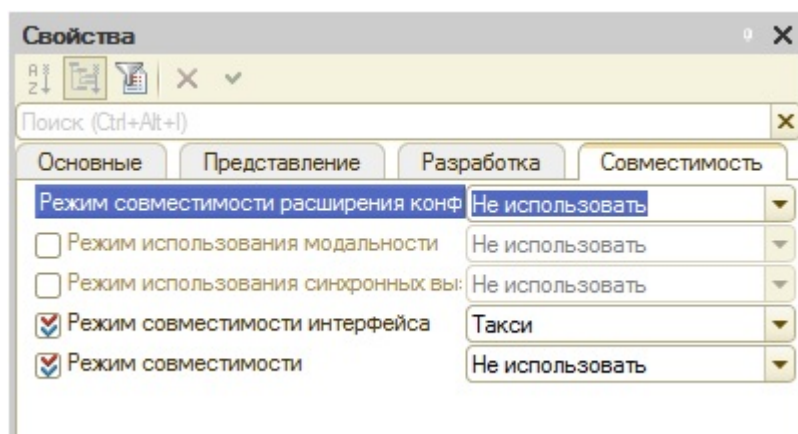


Рис. 5.2. Режим совместимости расширения

Поскольку на момент написания книги у основных конфигураций 1С режим совместимости не был в значении «Не использовать», то примеры к этой главе (и всем остальным) будут на демо-конфигурации «Управляемое приложение». Для этой конфигурации я создам расширение, которое назову «Для_Прихода_Товара» (см. рис. 5.3).

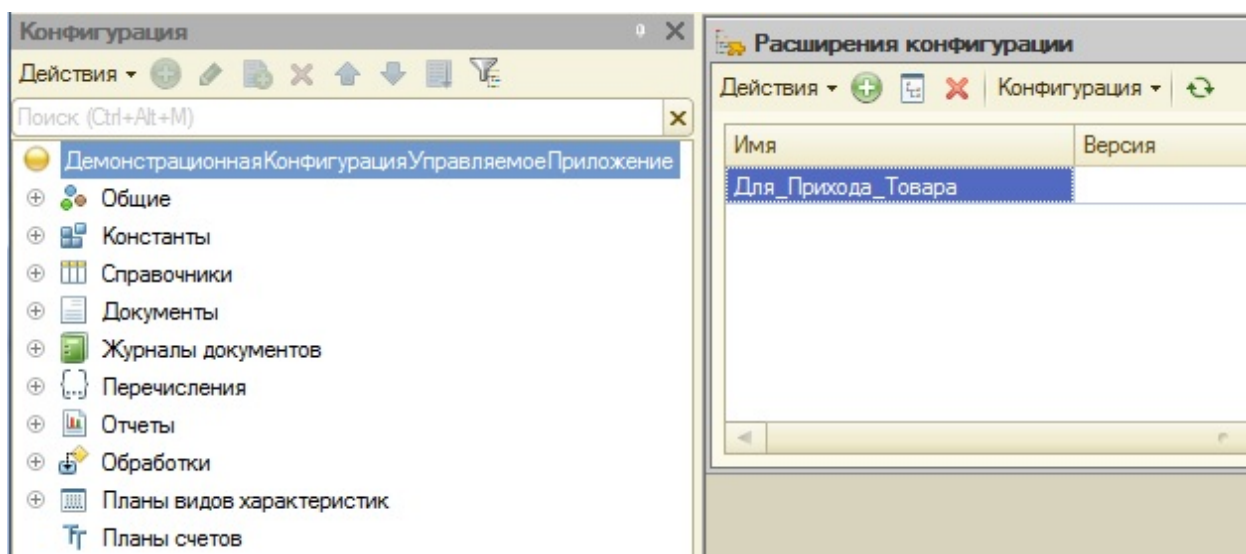


Рис. 5.3. Расширение «Для_Прихода_Товара»

Заимствуем в это расширение документ «Приход товара» (см. рис. 5.4).

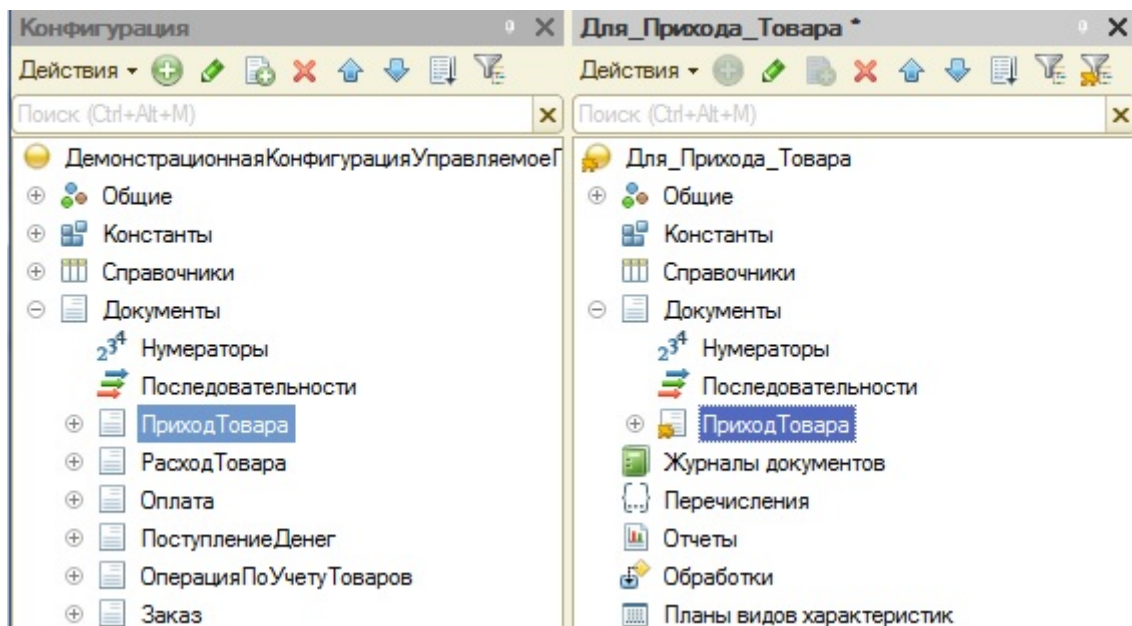


Рис. 5.4. Заимствовали документ «Приход товара»

Модуль объекта этого документа был заимствован вместе с документом.

Откроем модуль заимствованного документа (см. рис. 5.5) и модуль расширяемого документа (см. рис. 5.6).

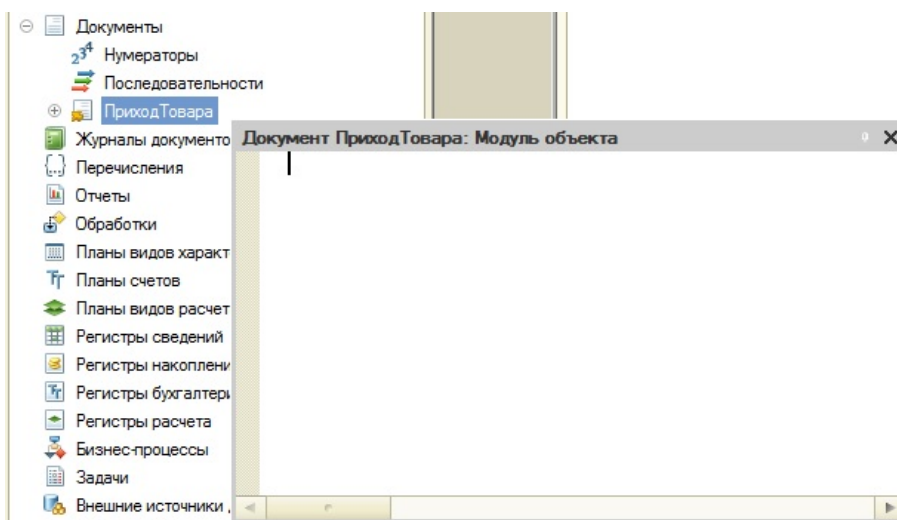


Рис. 5.5. Модуль заимствованного объекта

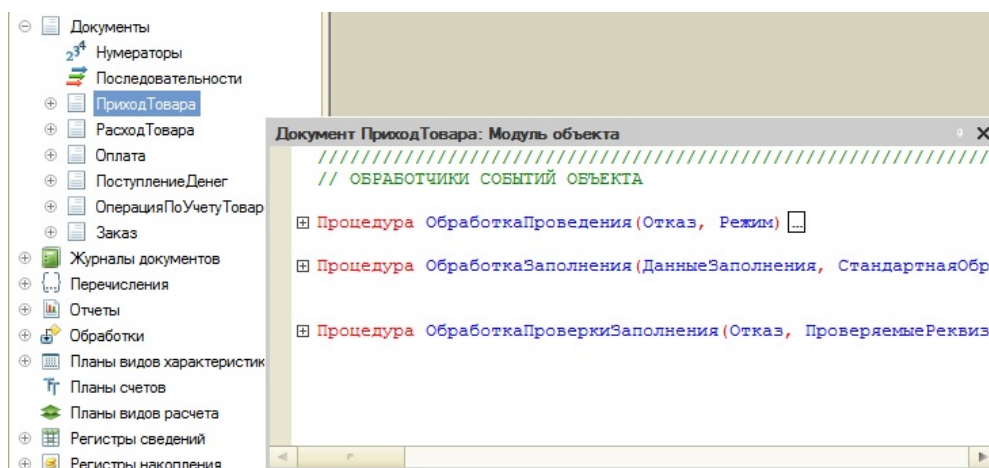


Рис. 5.6. Модуль расширенного объекта

Вы заметили, что в модуле объекта расширенной конфигурации нет никаких процедур и функций, но модуль расширения работает в контексте модуля основной (расширенной) конфигурации. Мы не можем в модуле расширения написать процедуру с тем же названием, как и в модуле расширенной конфигурации, будет сгенерирована ошибка (см. рис. 5.7).

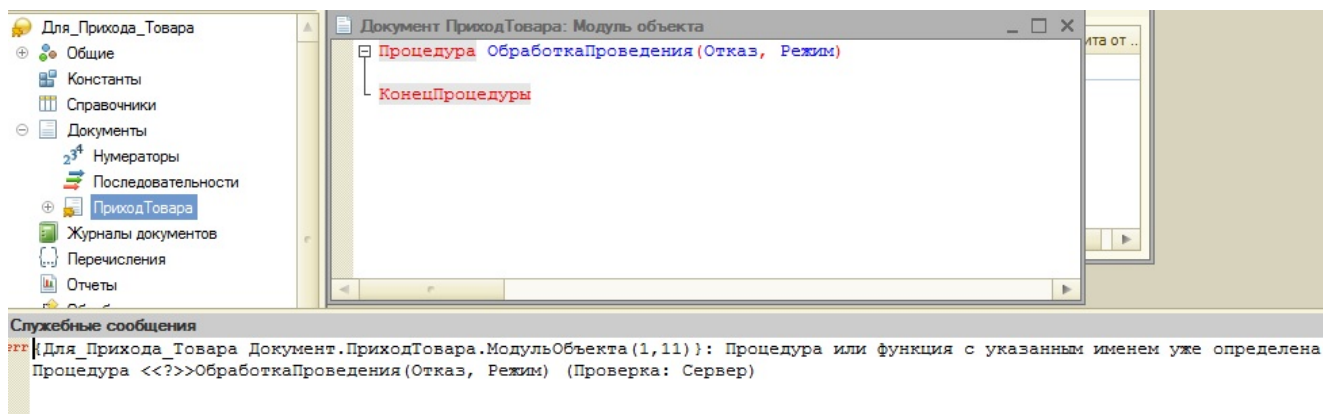


Рис. 5.7. Ошибка при задании процедуры с тем же названием, как и в расширенной конфигурации

В то же время, если мы будем писать свой код, то вполне сможем использовать методы и переменные основной конфигурации.

Научимся перехватывать методы расширенной конфигурации. Напишем процедуру, которая будет выполняться перед проведением документа «Приход товара». Для этого необходимо использовать аннотацию «Перед» и в скобках имя метода, перед которым будет срабатывать метод расширения (см. листинг. 5.1).

```
&Перед("ОбработкаПроведения")
```

Листинг 5.1. Аннотация «Перед»

Имя в кавычках обязательно должно совпадать с именем метода расширяемой конфигурации. Осталось написать процедуру, **с каким угодно названием, кроме названия оригинала**. У написанного метода обязательно должен быть тот же состав параметров, что и у оригинальной процедуры, в этой процедуре напишем простой код (см. листинг 5.2).

```
&Перед("ОбработкаПроведения")
```

```
Процедура У_Перед_ОбработкаПроведения(Отказ, Режим)
```

```
Сообщить("Выполнили перед проведением объекта");
```

```
КонецПроцедуры
```

Листинг 5.2. Перехват события перед проведением документа

Теперь перехватим событие после проведения документа. Для этого используем аннотацию «После» и в скобках пишем имя метода, после которого будет выполняться этот код, точно так же напишем процедуру с простым кодом (см. листинг 5.3).

```
&После("ОбработкаПроведения")
```

```
Процедура У_После_ОбработкаПроведения(Отказ, Режим)
```

```
Сообщить("Выполнили после проведения объекта");
```

```
КонецПроцедуры
```

Листинг 5.3. Перехват события после проведения документа

Сохраним конфигурацию, обновим базу данных, и посмотрим, как работает наше расширение (см. рис. 5.8).

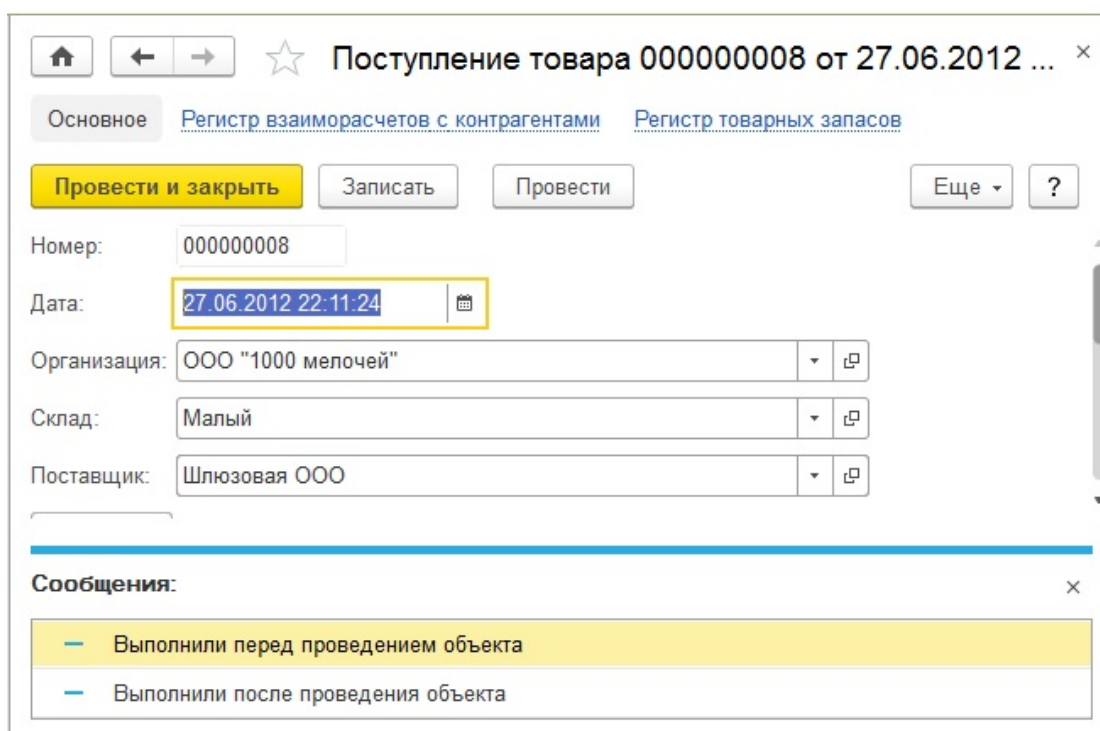


Рис. 5.8. Перехват событий перед проведением документа и после

Но имеется возможность вообще перехватить выполнение типовой процедуры, для этого нужно использовать аннотацию «Вместо». В качестве примера сделаем код немного посложнее: перехватим печать товара документом «Расход товара» и будем использовать свой макет.

Заимствуем в расширение документ «Расход товара», заимствуем макет «МакетПечати», а также заимствуем табличную часть, позже поймете почему (см. рис. 5.09).

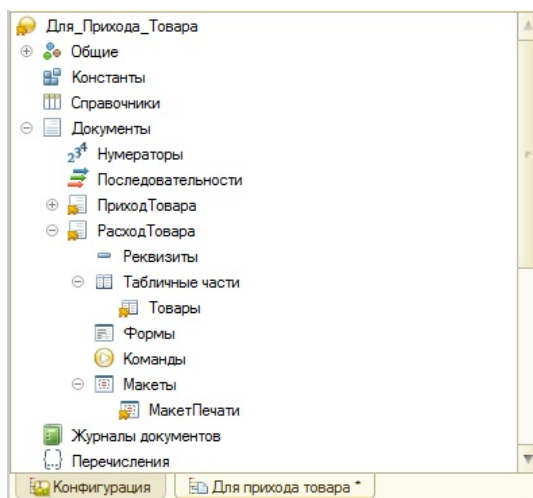


Рис. 5.9. Заимствованный документ «Расход товара»

Перехватим метод, в результате которого выполняется печать макета «МакетПечати», это процедура «ПечатнаяФорма», которая расположена в модуле документа «Расход товара» (см. рис. 5.10).

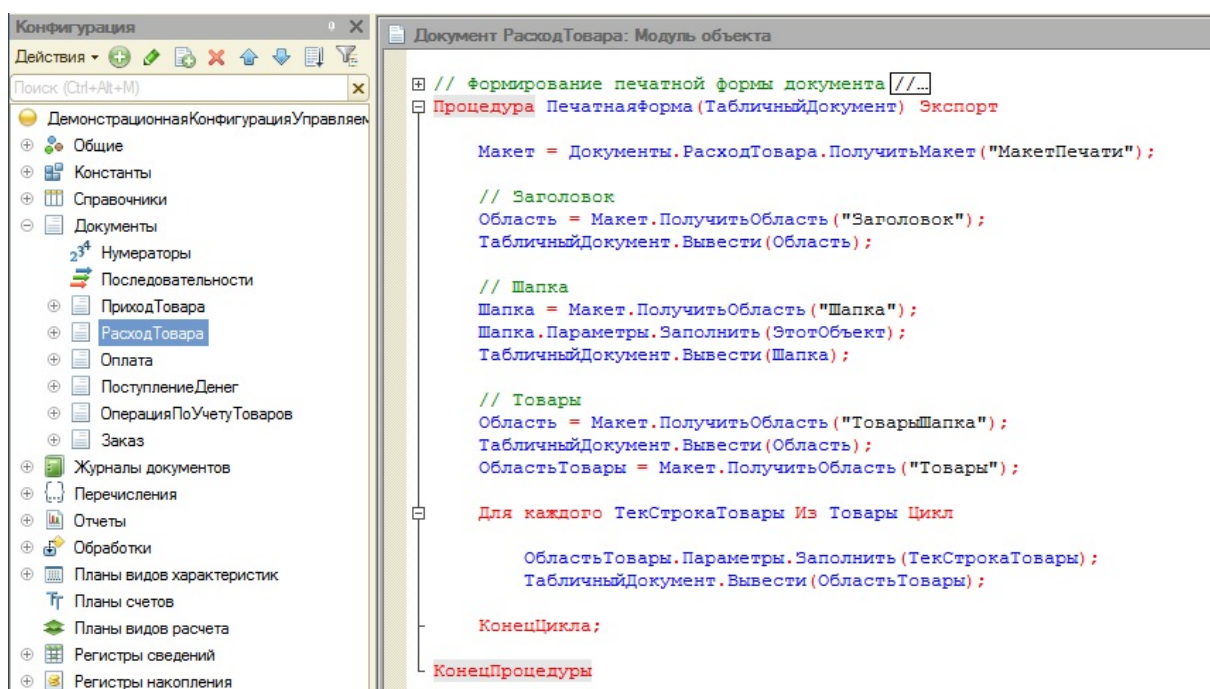


Рис. 5.10. Процедура «ПечатнаяФорма» модуля расширенного документа

Зайдем в модуль заимствованного документа «Расход товара» и напишем аннотацию «Вместо» вместе с именем метода, который будем менять (см. листинг 5.3).

&Вместо("ПечатнаяФорма")

Листинг 5.3. Аннотация «Вместо»

Процедура «ПечатнаяФорма» модуля расширенного документа имеет один параметр – «ТабличныйДокумент». Напишем под аннотацией «Вместо» свою процедуру в модуле расширенного документа, где укажем этот параметр (см. листинг 5.4).

&Вместо("ПечатнаяФорма")

Процедура Расш_У_ПечатнаяФорма(ТабличныйДокумент)

КонечПроцедуры

Листинг 5.4. Процедура в расширенном модуле документа «Расход товара»

Напомню, имя процедуры не должно совпадать с именем аналогичной процедуры в расширяемой конфигурации. Хотя оно может быть любым.

Осталось скопировать код из расширяемой конфигурации в процедуру расширения (см. рис. 5.11), обратите внимание на макет – «МакетПечати», и табличную часть – «Товары». Эти объекты берутся из расширения. Если бы мы их не заимствовали, то при сохранении расширения платформа выдала бы ошибку.

```

Документ РасходТовара: Модуль объекта
&Вместо ("ПечатнаяФорма")
Процедура Расш_У_ПечатнаяФорма (ТабличныйДокумент)

    Макет = Документы.РасходТовара.ПолучитьМакет "МакетПечати" ;

    // Заголовок
    Область = Макет.ПолучитьОбласть ("Заголовок");
    ТабличныйДокумент.Вывести (Область);

    // Шапка
    Шапка = Макет.ПолучитьОбласть ("Шапка");
    Шапка.Параметры.Заполнить (ЭтотОбъект);
    ТабличныйДокумент.Вывести (Шапка);

    // Товары
    Область = Макет.ПолучитьОбласть ("ТоварыШапка");
    ТабличныйДокумент.Вывести (Область);
    ОбластьТовары = Макет.ПолучитьОбласть ("Товары");

    Для каждого ТекСтрокаТовары Из Товары Цикл

        ОбластьТовары.Параметры.Заполнить (ТекСтрокаТовары);
        ТабличныйДокумент.Вывести (ОбластьТовары);

    КонечЦикла;

КонечПроцедуры
  
```

Рис. 5.11. Модуль расширенного документа «РасходТовара»

Изменим макет в расширении (см. рис. 5.12).

1	2	3	4	5	6
Заголовок	Расход товара (из расширения)				
Шапка	Номер	<Номер>			
	Дата	<Дата>			
	Покупатель	<Покупатель>			
	Склад	<Склад>			
ТоварыШаг	Товар	Цена	Количество	Сумма	
Товары	<Товар>	<Цена>	<Количество>	<Сумма>	

Рис. 5.12. Изменили макет печати в расширении

Сохраним конфигурацию, обновим базу данных и посмотрим, как работает наше расширение (см. рис. 5.13).

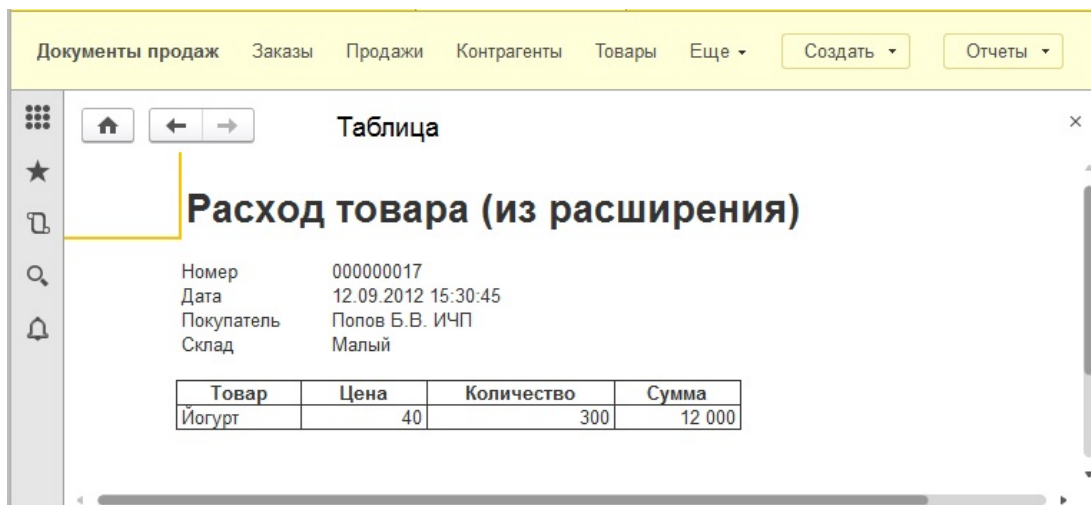


Рис. 5.13. Печать расходной накладной

Но самое интересное, типовой метод, который можно заменить, также можно продолжить. Для этого необходимо использовать метод «ПродолжитьВызов», в котором необходимо в качестве параметров указать параметры аналогичного метода в расширяемой конфигурации.

В модуле расширенного документа «Расход товара» напишем код, в котором перехватим обработку проведения документа «РасходТоваров», и потом продолжим вызов (см листинг 5.5).

```
&Вместо("ОбработкаПроведения")
```

```
Процедура Расш_У_ОбработкаПроведения(Отказ, Режим)
```

```
Сообщить("Перехватили обработку проведения");
```

```
ПродолжитьВызов(Отказ,Режим);
```

```
КонецПроцедуры
```

Листинг 5.5. Перехватили обработку проведения документа «Расход товара» и продолжили её выполнение

Сохраним конфигурацию, обновим базу данных и проверим, как работает этот перехват (см. рис. 5.14).

The screenshot shows a window titled "Продажа 000000017 от 12.09.2012 15:30:45". The window contains several tabs: "Основное", "Регистр взаиморасчетов с контрагентами", "Регистр продаж", and "Регистр товарных запасов". Below the tabs are buttons: "Провести и закрыть" (highlighted in yellow), "Записать", "Провести", "Создать на основании", "Печать", "Еще", and "?".

Fields include:

- Номер: 000000017
- Дата: 12.09.2012 15:30:45 (highlighted in yellow)
- Вид цен: Мелкооптовая
- Валюта взаиморасчетов: Рубли
- Организация: ООО "Все для дома"
- Покупатель: Попов Б.В. ИЧП
- Склад: Малый

Below the fields is a table with columns: N, Товар, Цена, Количество, Сумма. The table contains one row:

N	Товар	Цена	Количество	Сумма
1	Йогурт	40,00	300,00	12 000,00

At the bottom, there is a "Сообщения:" section with a yellow message: "Перехватили обработку проведения".

Рис. 5.13. Перехват обработки проведения

Заметьте, у нас вышло сообщение из расширения о проведении документа, в то же время и само проведение документа произошло. В 9-ой главе мы разберем некоторые практические случаи перехватов событий объектов. В этой главе мы научились перехватывать методы модулей объектов, в следующей главе будем разбираться с формами.

Глава 6. Работа с формами

В этой главе мы будем заимствовать формы и узнаем, как они работают в расширении. Все примеры в этой главе будем делать на форме документа «Приход товара» конфигурации «Управляемое приложение» (см. рис. 6.1).

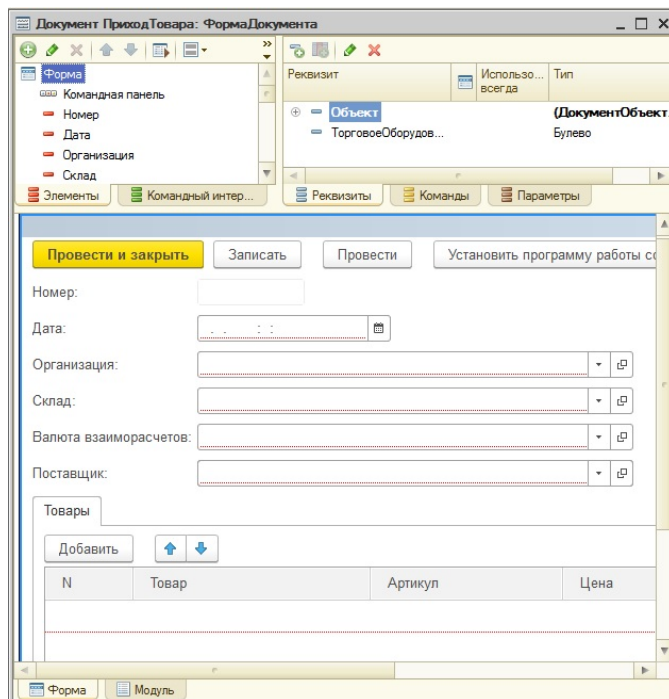


Рис. 6.1. Форма документа «Приход товара»

Добавим эту форму в наше расширение (см. рис. 6.2).

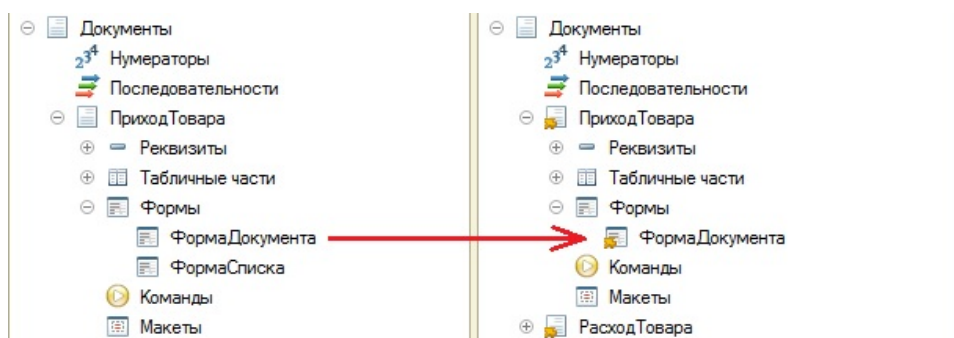


Рис. 6.2. Добавили форму документа в расширение конфигурации

Теперь в форме, которую мы добавили в расширение, создадим декорацию, которую назовем «Декорация расширения» (см. рис. 6.3).

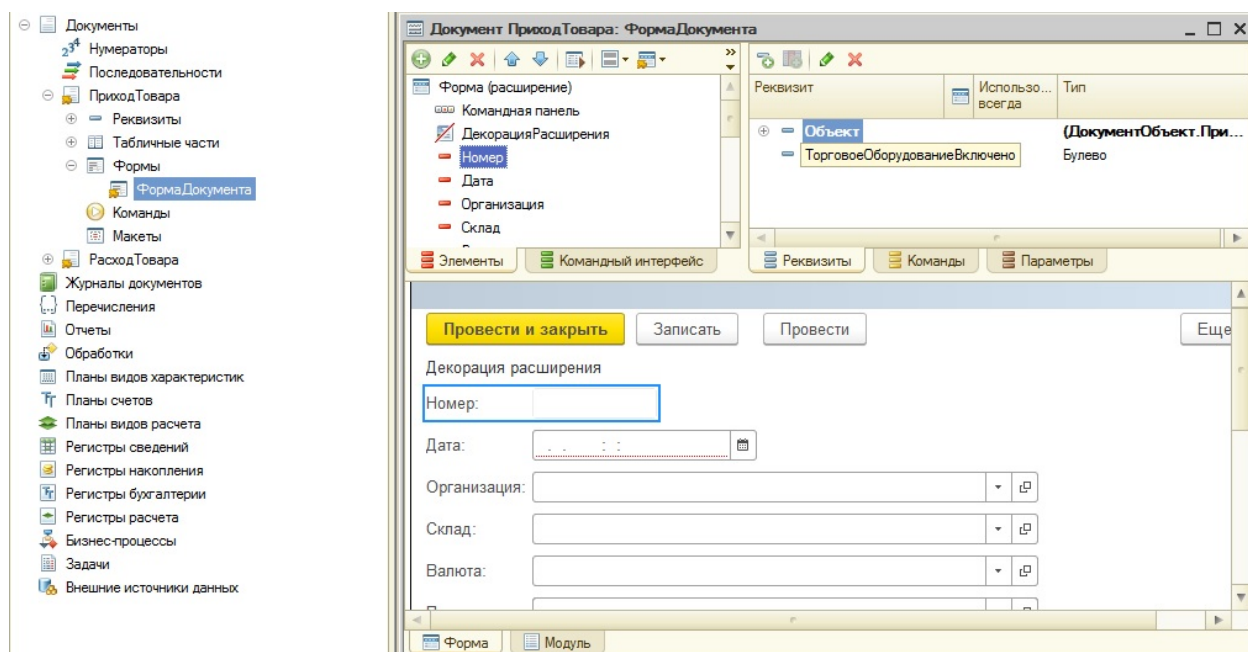


Рис. 6.3. Создали на форме расширения декорацию

Сохраним расширение, обновим базу данных и посмотрим, как запустится наша форма (см. рис. 6.4)

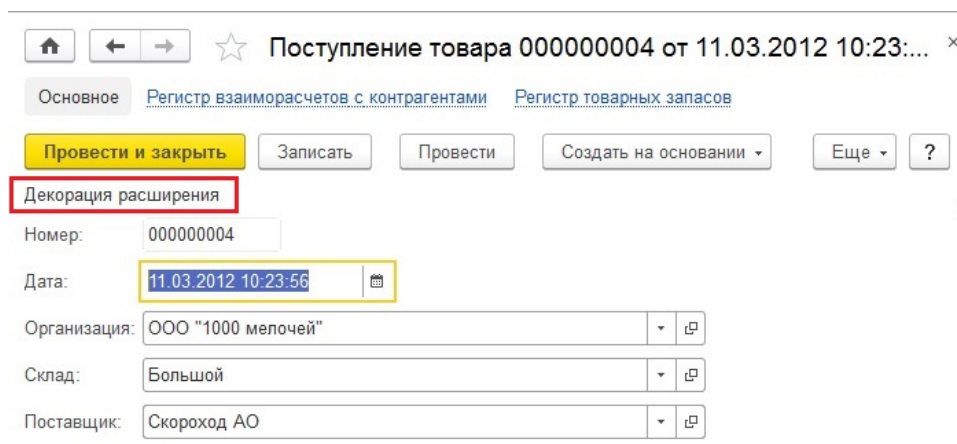


Рис. 6.4. Форма документа «Приход товара» в пользовательском режиме

Как видно из рисунков 6.3 и 6.4, наша форма расширенной конфигурации включила в себя и элементы основной формы (собственно все основные элементы), и элементы расширения.

Теперь добавим еще одну декорацию на расширяемую форму основной конфигурации (см. рис. 6.5).

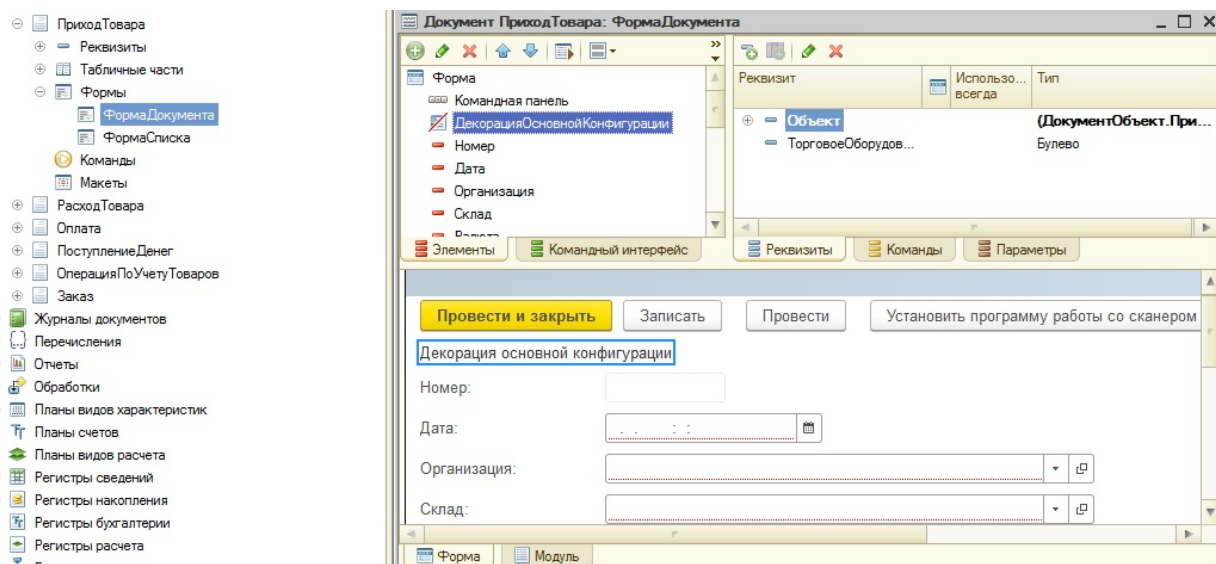


Рис. 6.5. Добавили декорацию на форму основной базы

Сохраним основную конфигурацию, обновим базу данных и посмотрим, как будет выглядеть наша форма документа в пользовательском режиме (см. рис. 6.6).

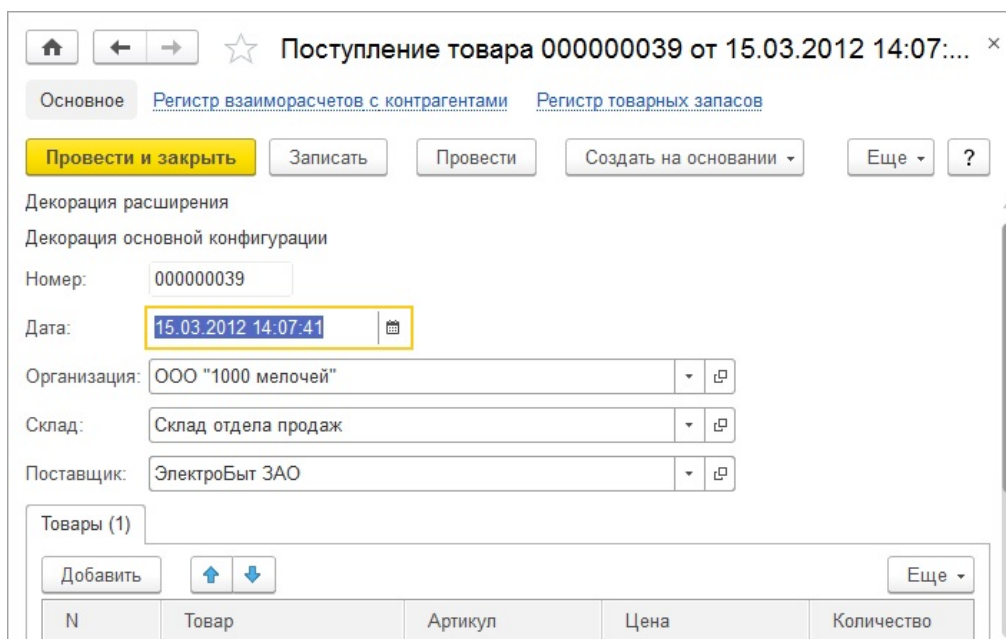


Рис. 6.6. Декорация документа «Приход товара»

Как видно из рисунка 6.6, форма в режиме «1С: Предприятия» объединила элементы и из расширяемой конфигурации, и из расширения, хотя новая декорация не попала в расширенную форму.

Сейчас мы разберем тонкости этого объединения. Для этого изменим заголовки некоторых полей в основной конфигурации (см. рис. 6.7), а также некоторых полей в расширенной конфигурации (см. рис. 6.8).

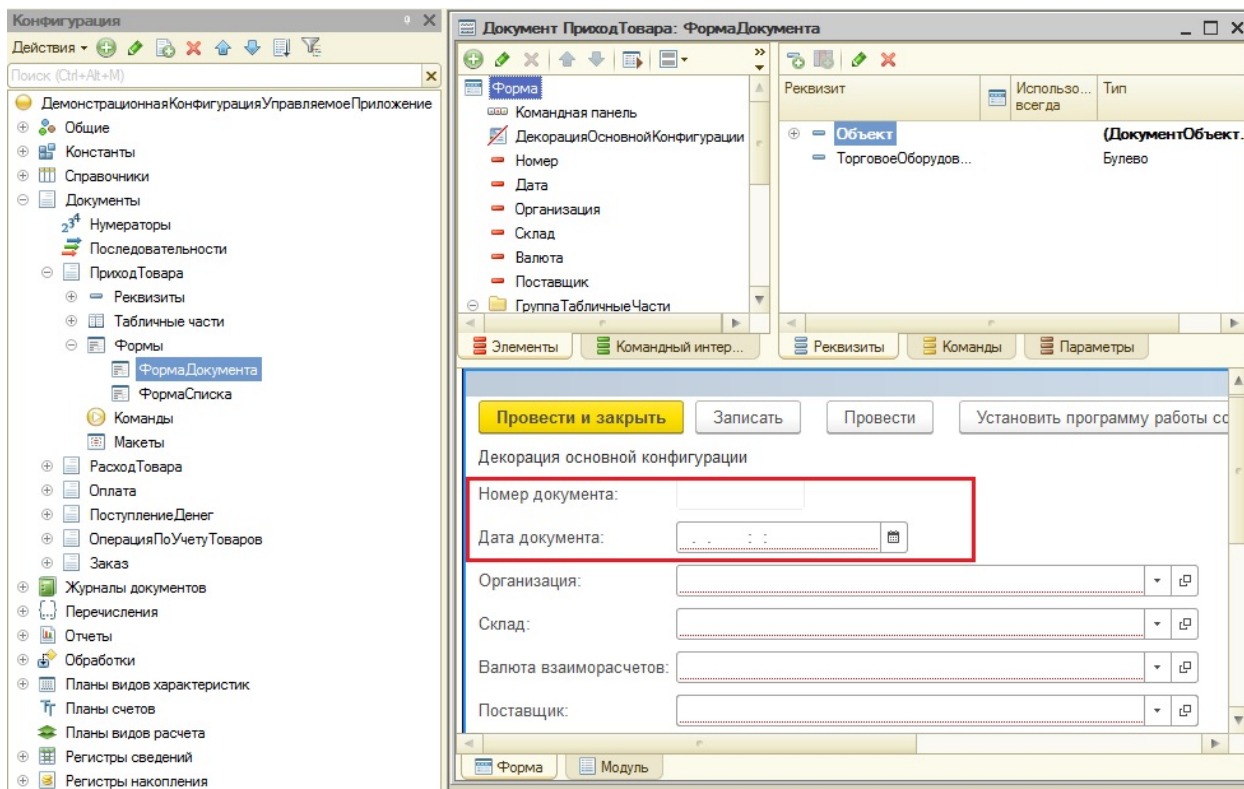


Рис. 6.7. Изменения на форме в основной конфигурации

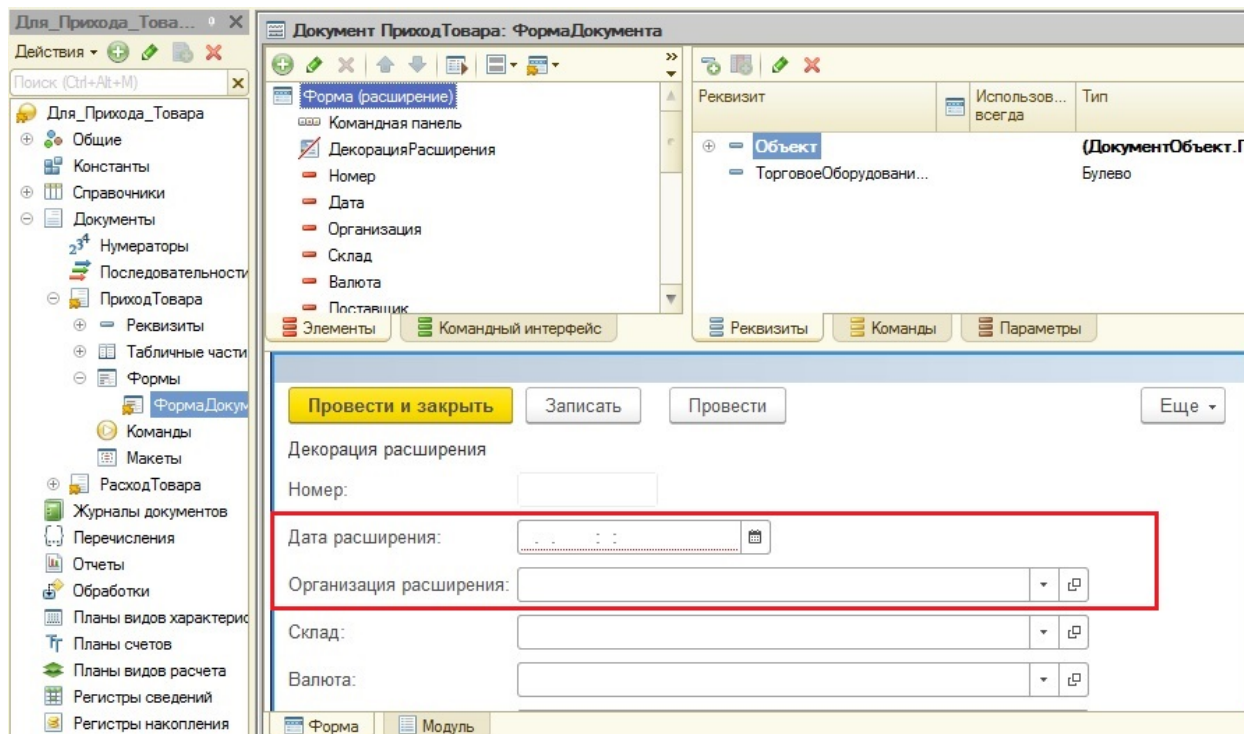


Рис. 6.8. Изменения на форме расширенной конфигурации

Сохраним расширение, обновим базу данных и посмотрим, как сейчас будет выглядеть наша форма (см. рис. 6.9).

Поступление товара 000000039 от 15.03.2012 14:07:...

Основное Регистр взаиморасчетов с контрагентами Регистр товарных запасов

Провести и закрыть Записать Провести Создать на основании Еще ?

Декорация расширения
Декорация основной конфигурации

Номер документа: 000000039

Дата расширения: 15.03.2012 14:07:41

Организация расширения: ООО "1000 мелочей"

Склад: Склад отдела продаж

Поставщик: ЭлектроБыт ЗАО

Товары (1)

Добавить ↑ ↓ Еще

Рис. 6.9. Форма документа «Приход товара» после изменений в основной и расширенной конфигурации

А теперь сформулируем правила объединения элементов формы:

Первое. Если элемент есть только в расширении, то он показывается в общей форме («Декорация расширения»).

Второе. Если элемент есть только в расширяемой конфигурации, то он показывается в общей форме («Декорация основной конфигурации»).

Третье. Если элемент изменен в основной конфигурации (расширяемой), то берется изменение из основной конфигурации («Номер документа»).

Четвертое. Если элемент изменен в расширении, то берется изменение из расширения («Организация расширения»).

Пятое. Если элемент изменен и в расширяемой конфигурации, и в расширении, то изменение берется из расширения («Дата расширения»).

Такие правила действуют не всегда. Они не действуют в случае условного оформления и командного интерфейса формы. Я не буду вдаваться в подробности этого объединения, отмечу только, что команды с одинаковым представлением могут дублироваться.

Перейдем к модулю расширенной формы. Как и в случае работы с модулями объектов, которые мы проходили в предыдущей главе, все процедуры и функции модуля формы заимствованы и к ним можно обращаться. Но они также *не показаны*. Работу с модулем расширенной формы следует выполнять через свойства модуля или свойства элемента модуля. В этом случае мы также можем задать возможные варианты исполнения кода: перед обработчиком, после обработчика или вместо обработчика.

Например, возьмем обработчик «ПриИзменении» поля «Организация». Он у нас изменен в основной конфигурации (см. рис. 6.10).

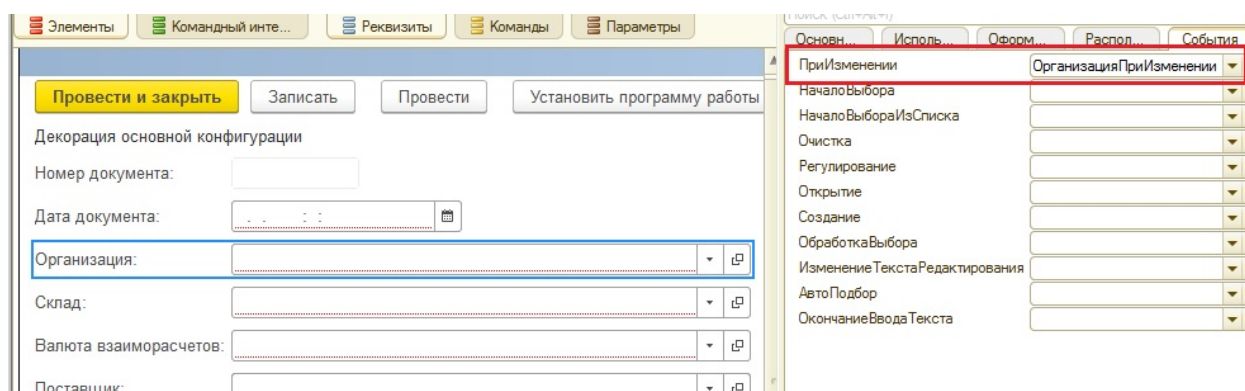


Рис. 6.10. Событие «ПриИзменении» поля «Организация» формы документа «Приход товара»

Теперь перейдем в палитру свойства этого элемента в расширении (см. рис. 6.11).

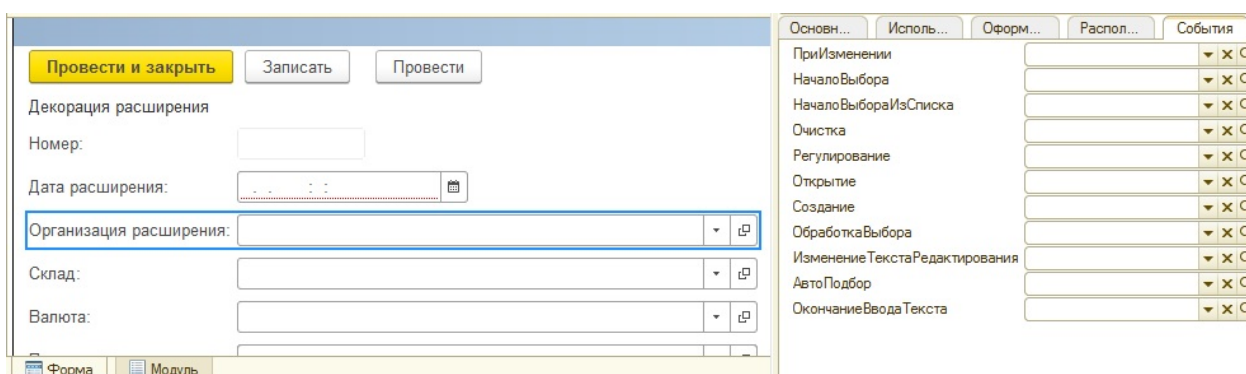


Рис. 6.11. Событие «ПриИзменении» поля «Организация» формы документа «Приход товара» в расширении

Как видите, все обработчики пустые, теперь нажмем на лупу рядом с обработчиком при изменении поля «Организации» формы документа в расширении (см. рис. 6.12).

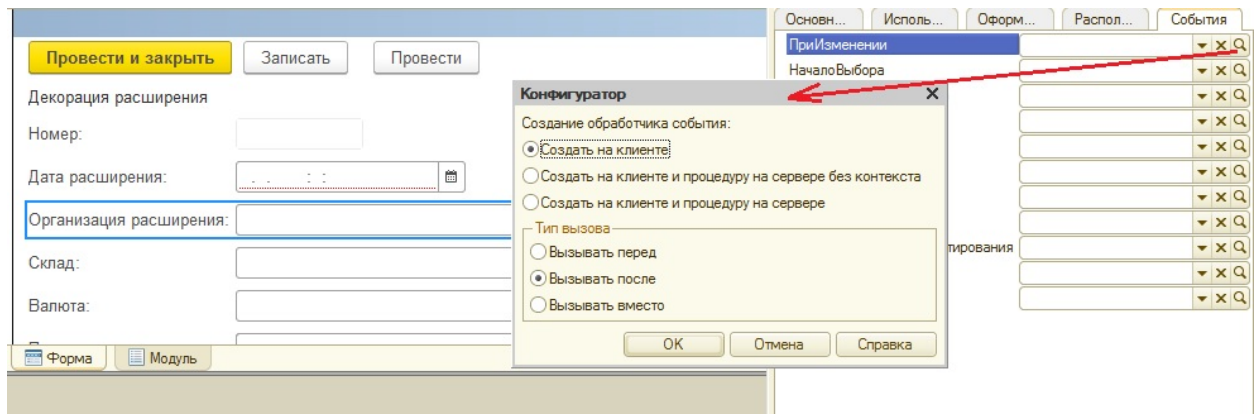


Рис. 6.12. Варианты создания обработчиков события в расширении

Как видно из рисунка 6.12, кроме стандартных вариантов расположения обработчика вышли еще три варианта типов вызова: «Перед», «После» и «Вместо».

Выберем вызов «Перед» и место «На клиенте» и создадим обработчик (см. рис. 6.13, 6.14).

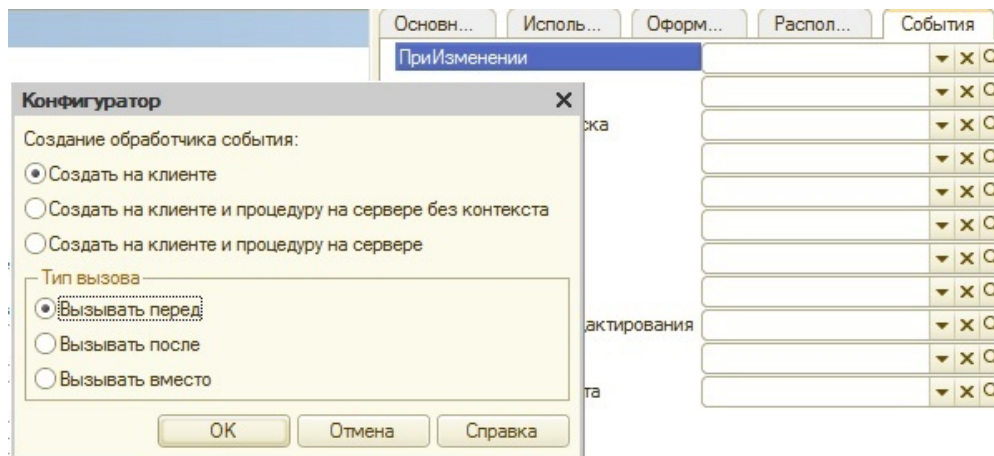


Рис. 6.13. Выбрали место обработчика и тип вызова

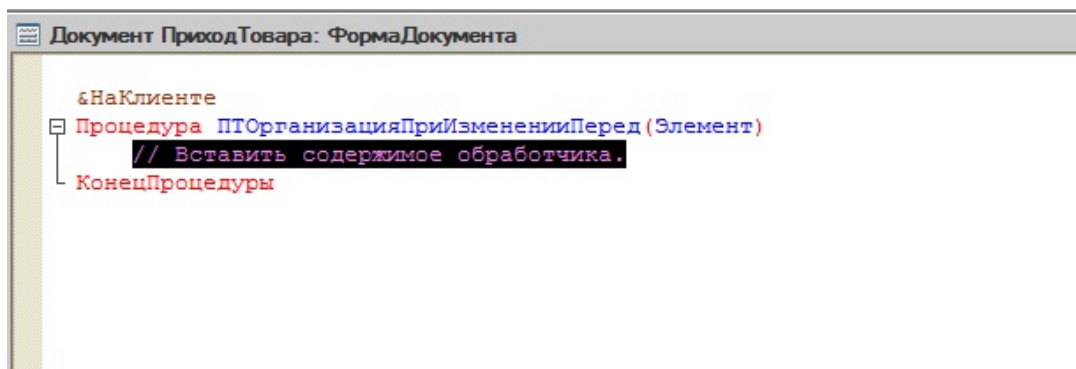


Рис. 6.14. Созданная процедура в форме документа

Как видно из рис. 6.13, в модуле нет аннотации «Перед», как в случае с модулем объектов. Перейдем обратно в свойство этого элемента (см. рис. 6.15).

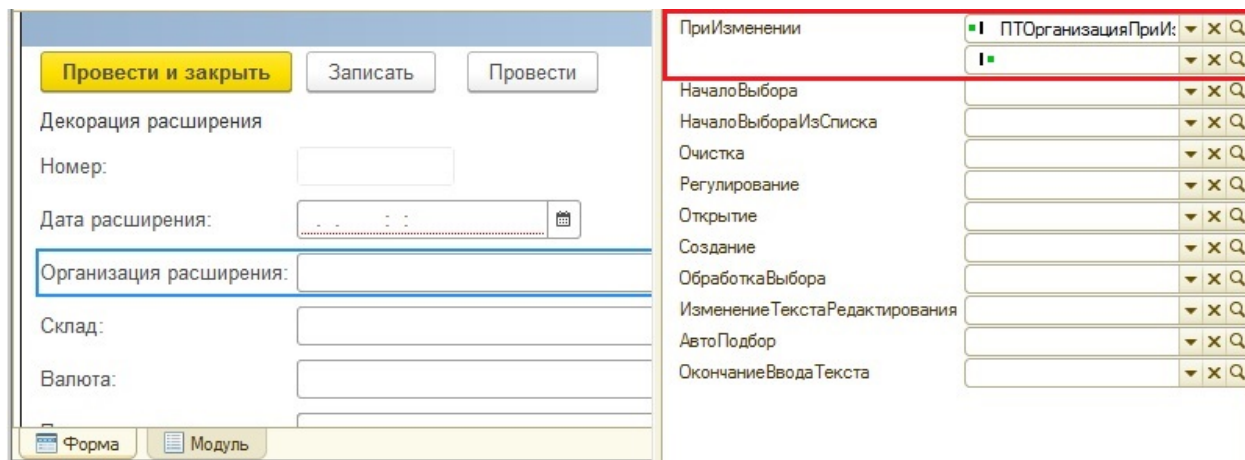


Рис. 6.15. Закладка «События» палитры свойств поля «Организация»

На рисунке 6.15 видно, что само свойство поменялось – появился знак, который означает, что процедура выполняется перед процедурой расширенной конфигурации.

Причем, сейчас мы можем создать обработчик, который будет выполняться только после процедуры расширенной конфигурации. Сделаем такой обработчик, нажав на лупу под уже созданным обработчиком «Перед» (см. рис. 6.16, 6.17).

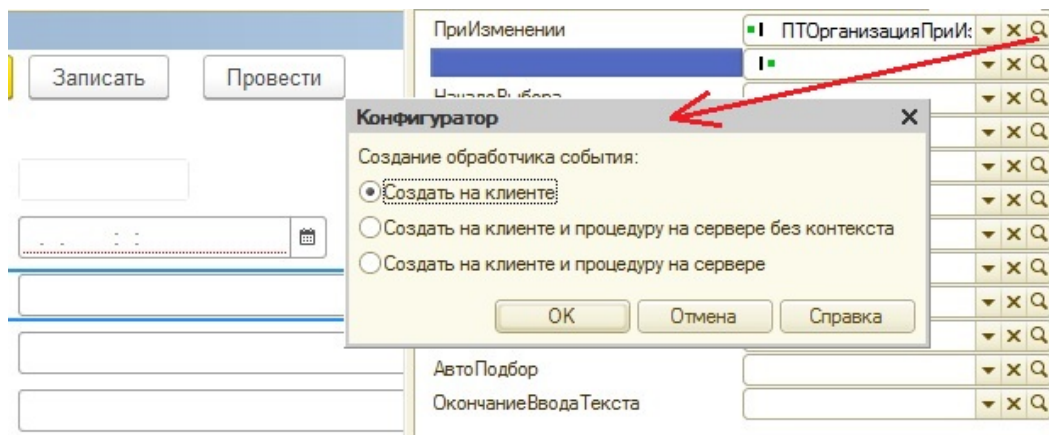


Рис. 6.16. Создаем обработчик, который будет выполняться после основного обработчика

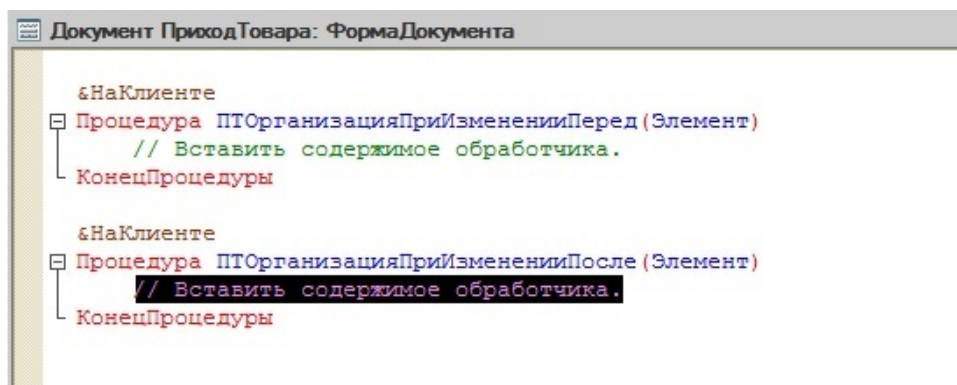


Рис. 6.17. Созданная процедура в форме документа

Если мы захотим сделать новую процедуру, которая будет выполняться вместо процедуры основной конфигурации, нам нужно или удалить имеющиеся обработчики, или пересоздать какой-то один, выбрав для него тип вызова «Вместо» (см. рис. 6.18, 6.19).

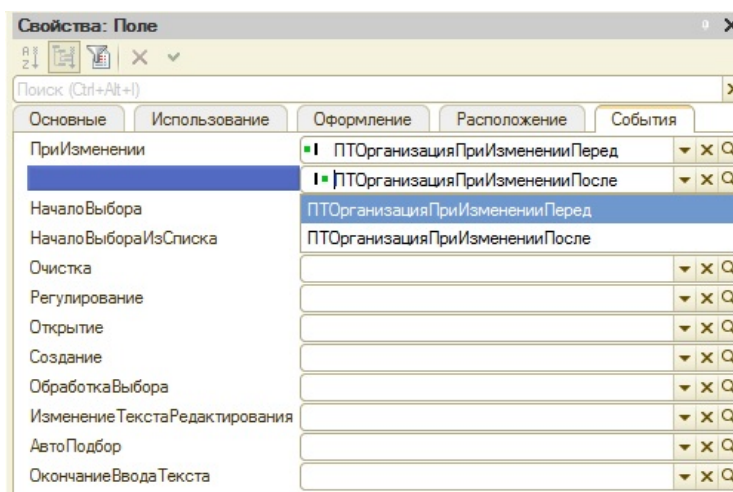


Рис. 6.18. Выбираем тот же самый обработчик, который уже есть

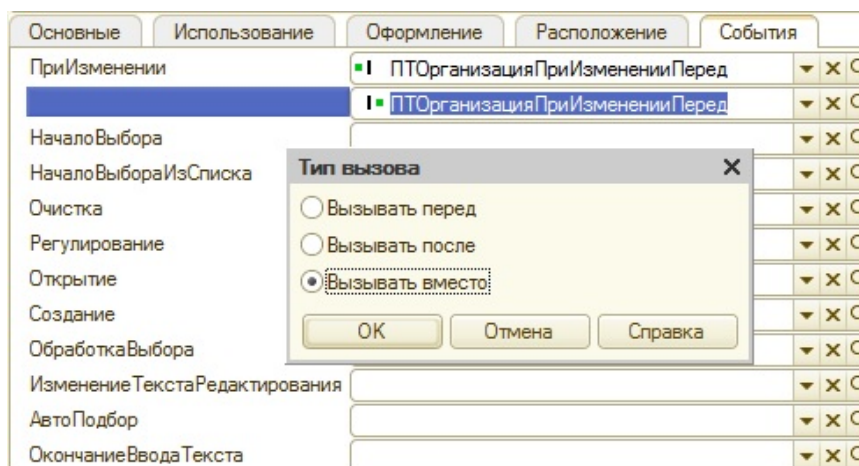


Рис. 6.19. После выбора платформа предложит новый тип вызова

В этом случае вариантов «Перед» и «После» уже не будет, ну они и не имеют смысла, и пиктограмма будет немного другая (см. рис. 6.20).

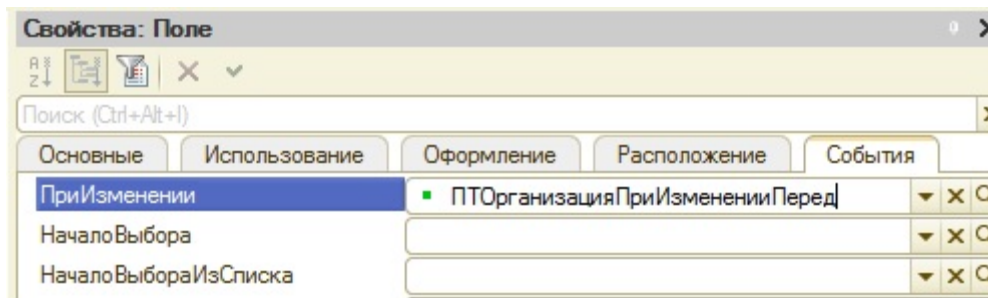


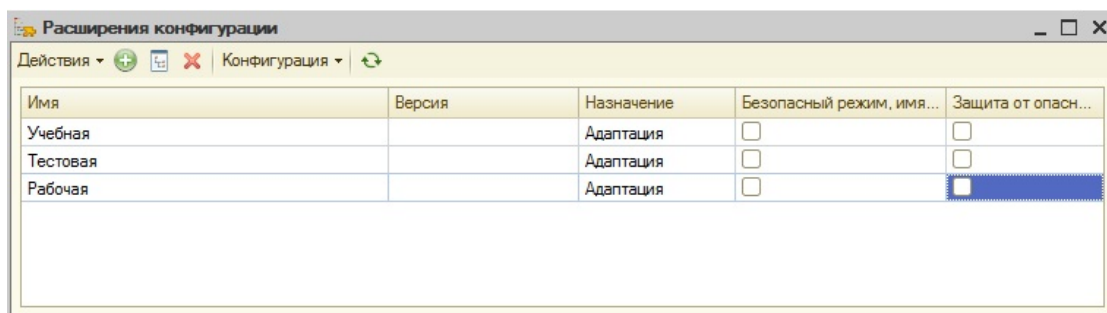
Рис. 6.20. Пиктограмма типа вызова «Вместо»

На этом закончим изучать работу с заимствованными формами. В следующей главе рассмотрим аспекты работы с несколькими расширениями.

Глава 7. Работа с несколькими расширениями

Платформа 1С 8.3. дает возможность на одну конфигурацию подключать несколько различных расширений конфигурации. В этой главе узнаем, как это осуществить и как расширения будут взаимодействовать между собой.

В конфигураторе учебной базы «Управляемое приложение» создадим несколько расширений с разными названиями и префиксами: «Учебная», «Тестовая» и «Рабочая» с префиксами соответственно «У», «Т», «Р» (см. рис. 7.1). Выключим у расширений безопасный режим и защиту, чтобы нам лишний раз не задавались вопросы.



Имя	Версия	Назначение	Безопасный режим, имя...	Защита от опасн...
Учебная		Адаптация	<input type="checkbox"/>	<input type="checkbox"/>
Тестовая		Адаптация	<input type="checkbox"/>	<input type="checkbox"/>
Рабочая		Адаптация	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 7.1. Расширения конфигурации

Создадим у них всех подсистемы с такими же названиями, как расширения. И просто обработки с аналогичными названиями и формами без всякого смысла, чтобы разместить в этих подсистемах. Сделаем мы это для того, чтобы подсистемы расширения отобразились в «1С:Предприятии». На рисунке 7.2. показано, как отобразились в пользовательском режиме подсистемы.

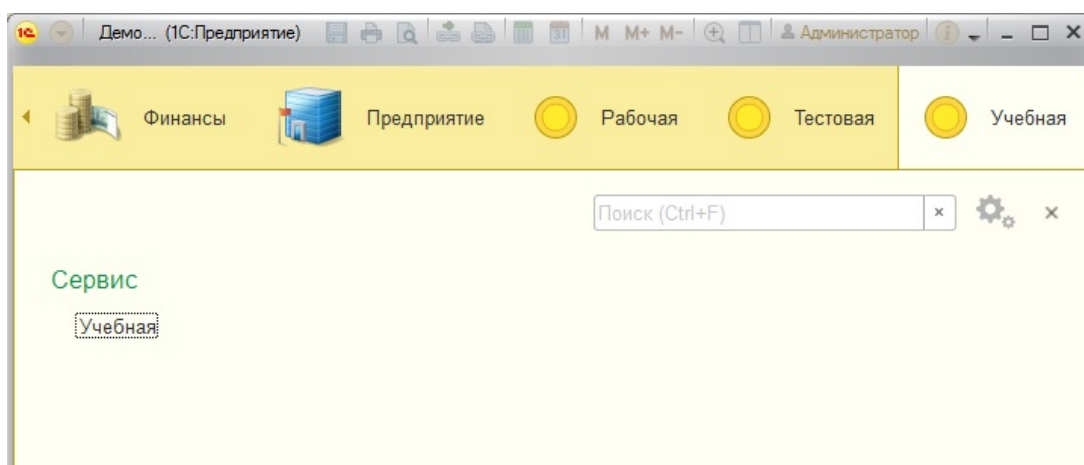


Рис. 7.2. Отображение подсистем расширений в пользовательском режиме

Они все подключились и даже есть какие-то обработки.

Узнаем, как можно подключать расширения и отключать. Для этого необходимо зайти в «Управление расширениями конфигурации». Делается это через меню «Все функции» (см. рис. 7.3, 7.4, 7.5).

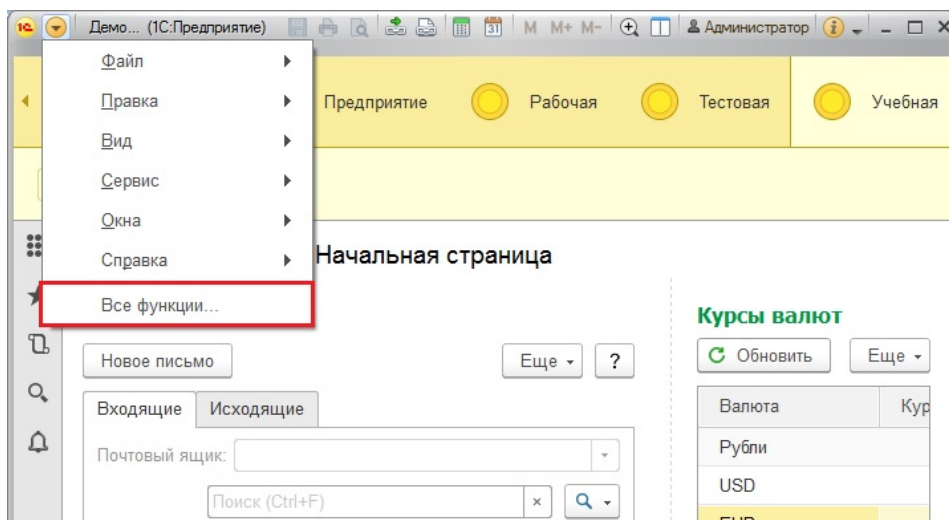


Рис. 7.3. Путь к меню «Все функции»

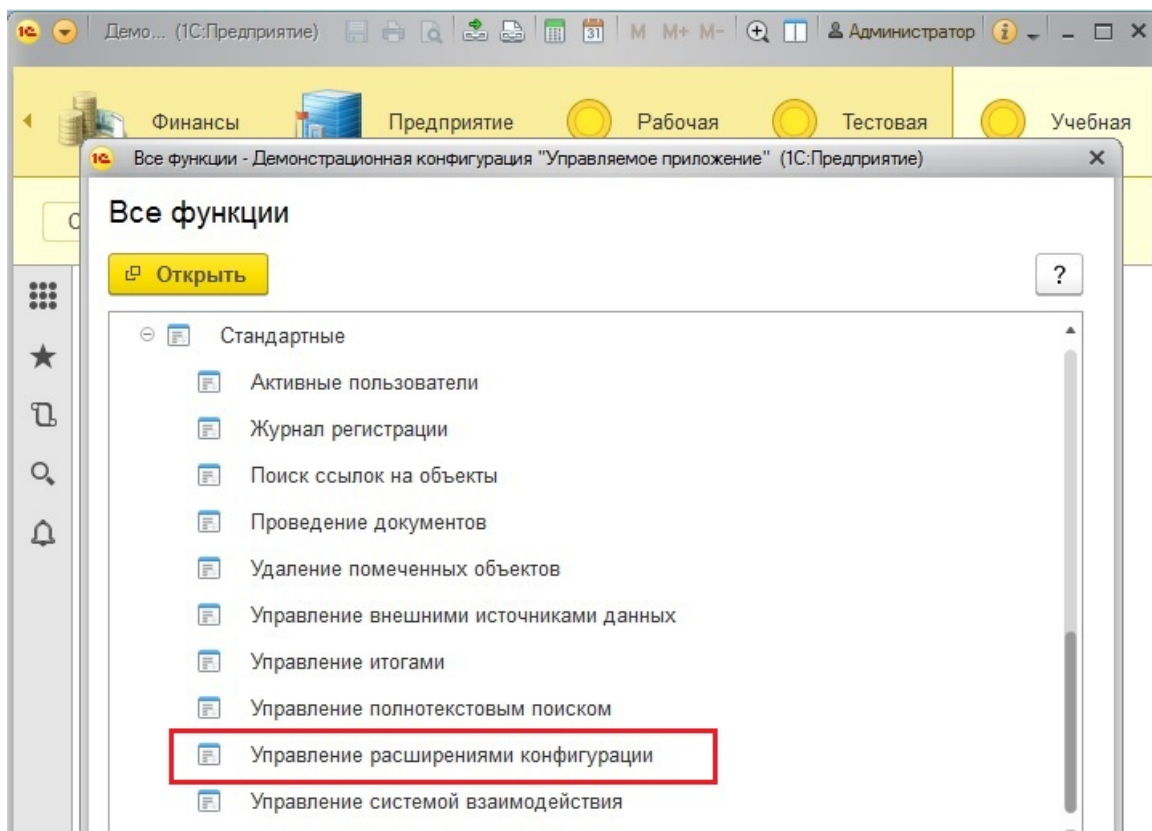


Рис. 7.4. «Управление расширениями конфигурации» в меню «Все функции»

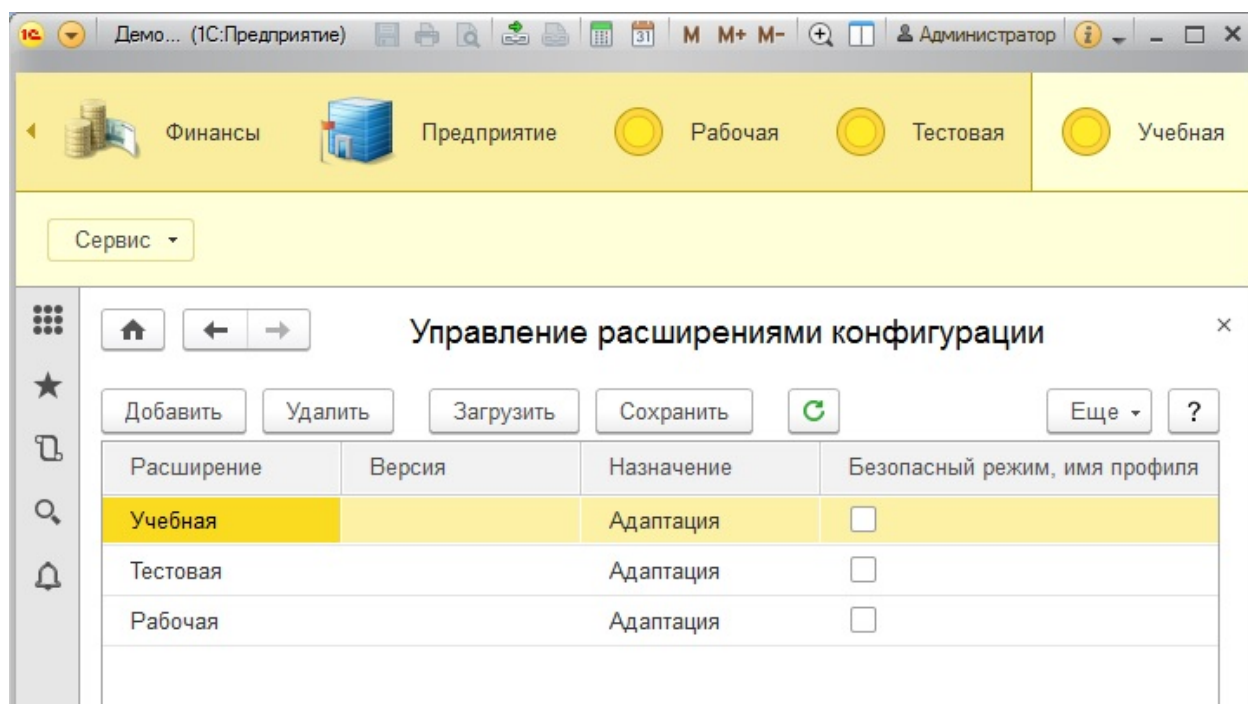


Рис. 7.5. «Управление расширениями конфигурации»

В этой форме мы можем удалять расширения и добавлять. **Но запомните! Если Вы удалите расширение и не сохраните, то оно исчезнет у Вас навсегда.** В конфигураторе его тоже не будет. Продемонстрируем это: сохраним расширение «Учебная» (оно нам еще пригодится), а потом удалим (см. рис 7.6, 7.7). **Конфигуратор должен быть закрыт, когда выполняете манипуляции с расширениями в пользовательском режиме.** Посмотрим, что будет с расширениями в конфигураторе (см. рис. 7.8).

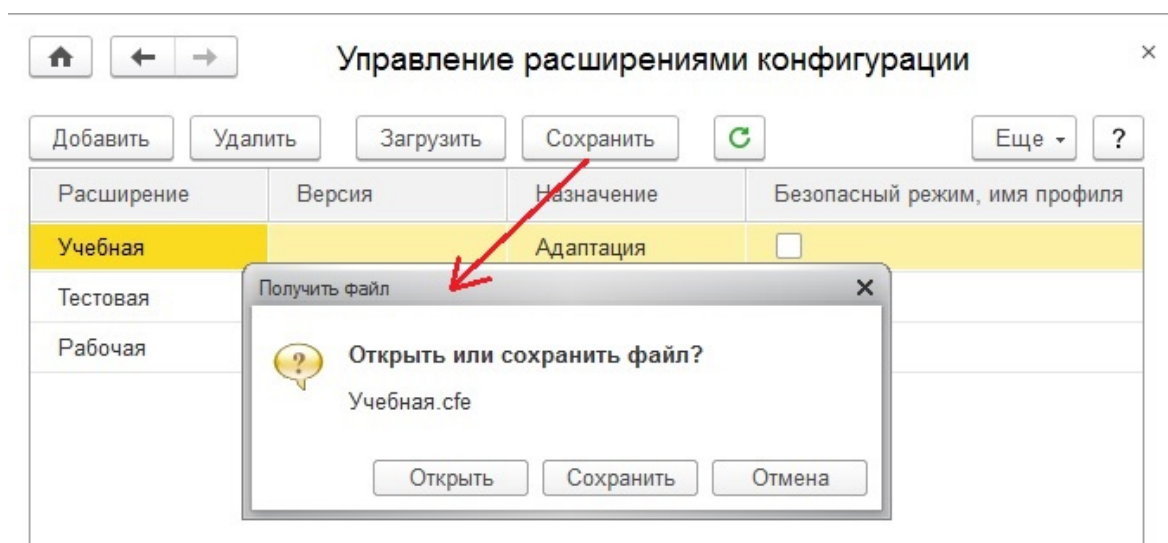


Рис. 7.6. Сохраняем расширение «Учебное»

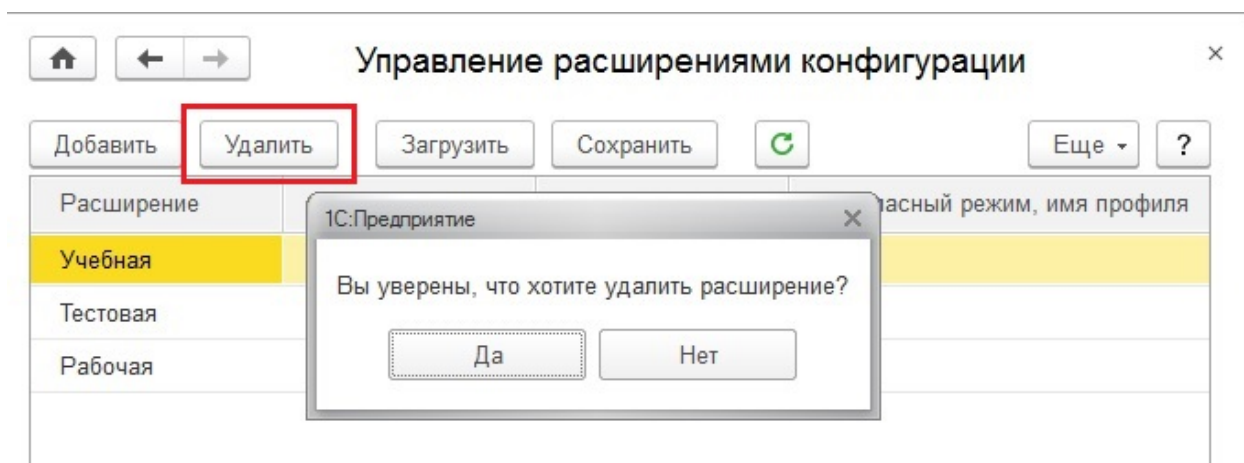


Рис. 7.7. Удаляем расширение «Учебное»

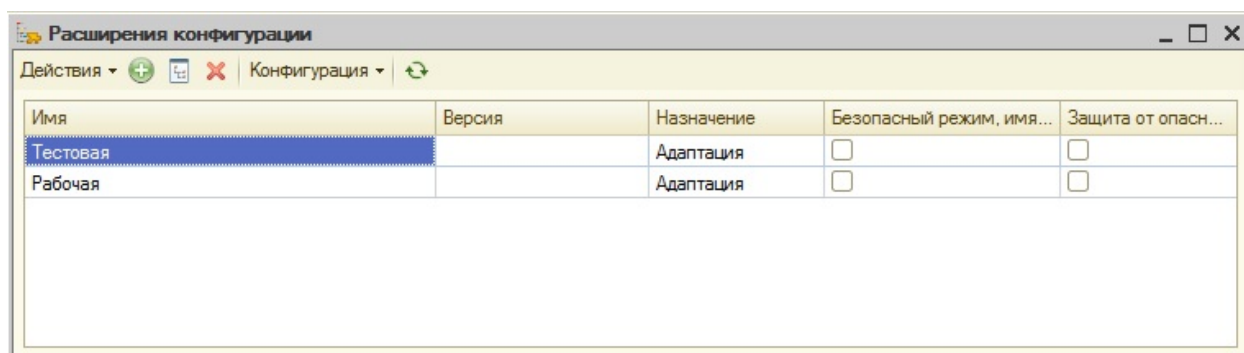


Рис. 7.7. Расширение «Учебное» исчезло из списка расширений в конфигураторе

Как видно из рисунка 7.7, расширение «Учебное» исчезло и списка расширений в конфигураторе.

Загрузим расширение обратно через «1С: Предприятие» (конфигуратор нужно закрыть). Для этого зайдём уже знакомым нам способом в «Управление расширениями конфигурации». И добавим новое расширение, нажав на кнопку «Добавить». В окне выбора файлов выберем сохранённый нами ранее файл с расширением *.cfe. (см. рис. 7.9). После этого конфигурация появится в списке (см. рис. 7.10).

Если мы после этих манипуляций перезапустим «1С: Предприятие», то увидим, что подсистема учебного расширения появилась.

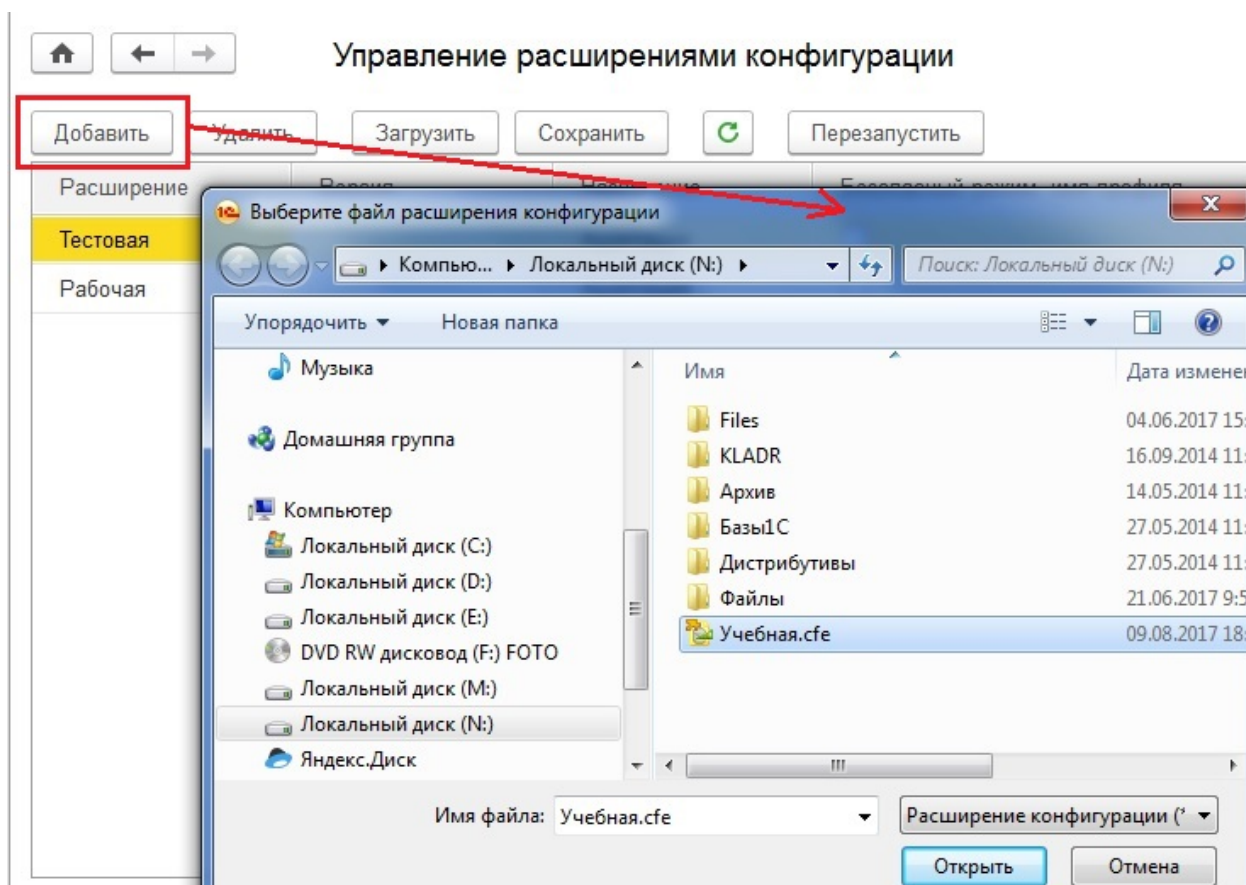


Рис. 7.9. Добавление нового расширения

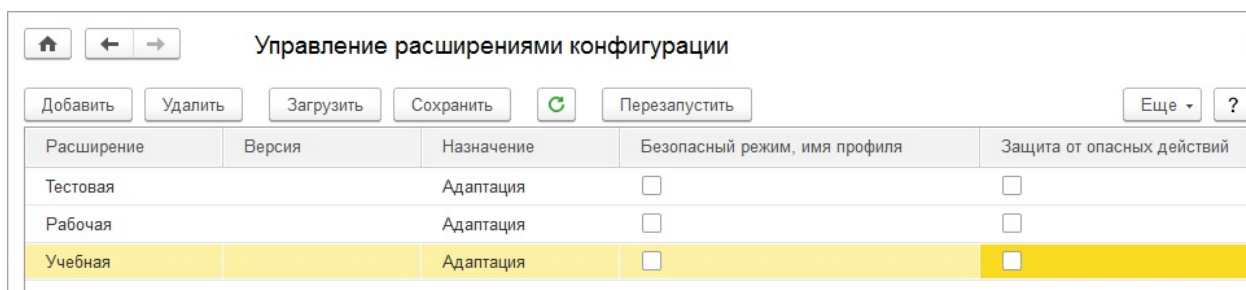


Рис. 7.10. Расширение появилось в списке

В предыдущих главах мы научились перехватывать методы. В этой главе мы узнаем, как работают перехваты методов, когда несколько расширений.

Откроем конфигуратор и на всякий случай проверим, установлен ли у новых расширений конфигурации режим совместимости «НеИспользовать» (см. рис. 7.11). Если не установлен, то его необходимо установить, как и у расширяемой (основной) конфигурации.

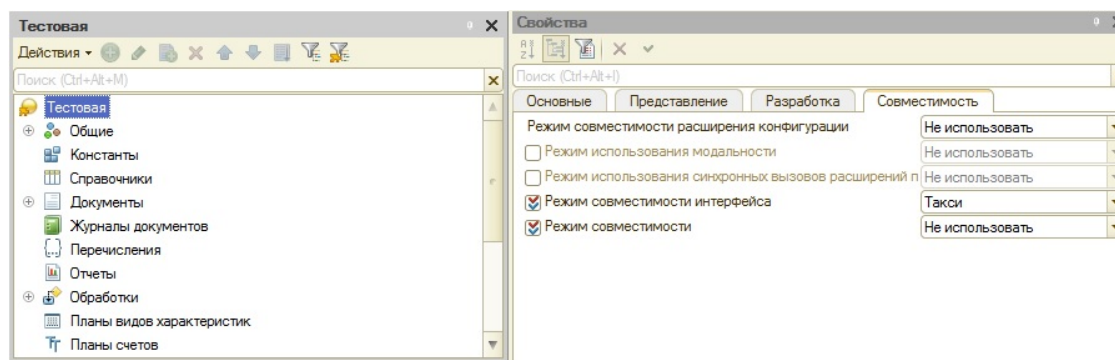


Рис. 7.11. Режим совместимости расширения конфигурации

Теперь заимствуем документ «Приход товара» во все три расширения (см. рис. 7.12, 7.13 и 7.14).

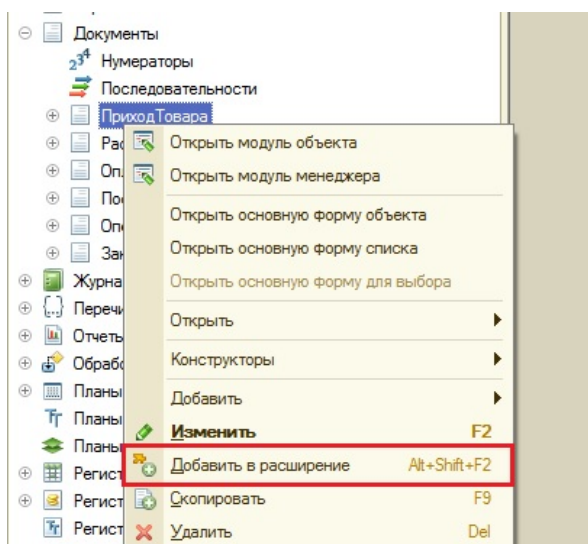


Рис. 7.12. Добавляем документ в расширение

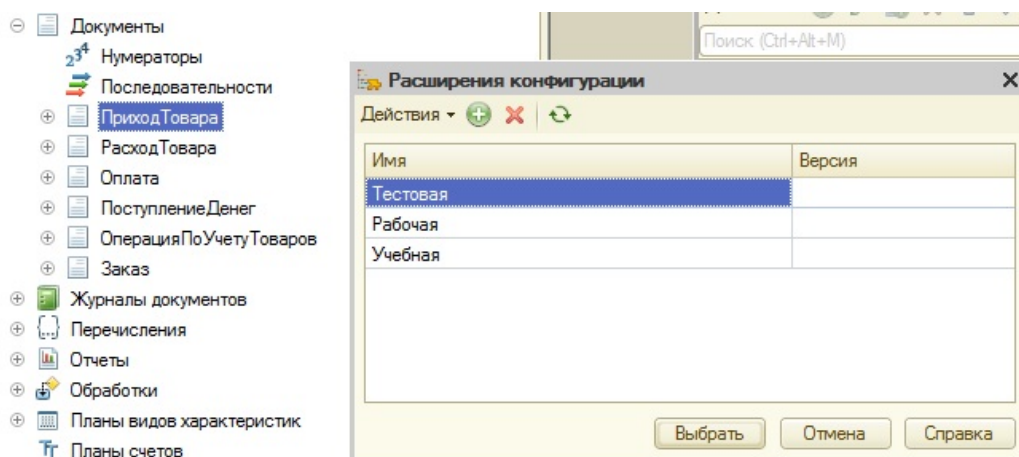


Рис. 7.13. Когда несколько расширений, нужно выбрать, в какое добавить документ

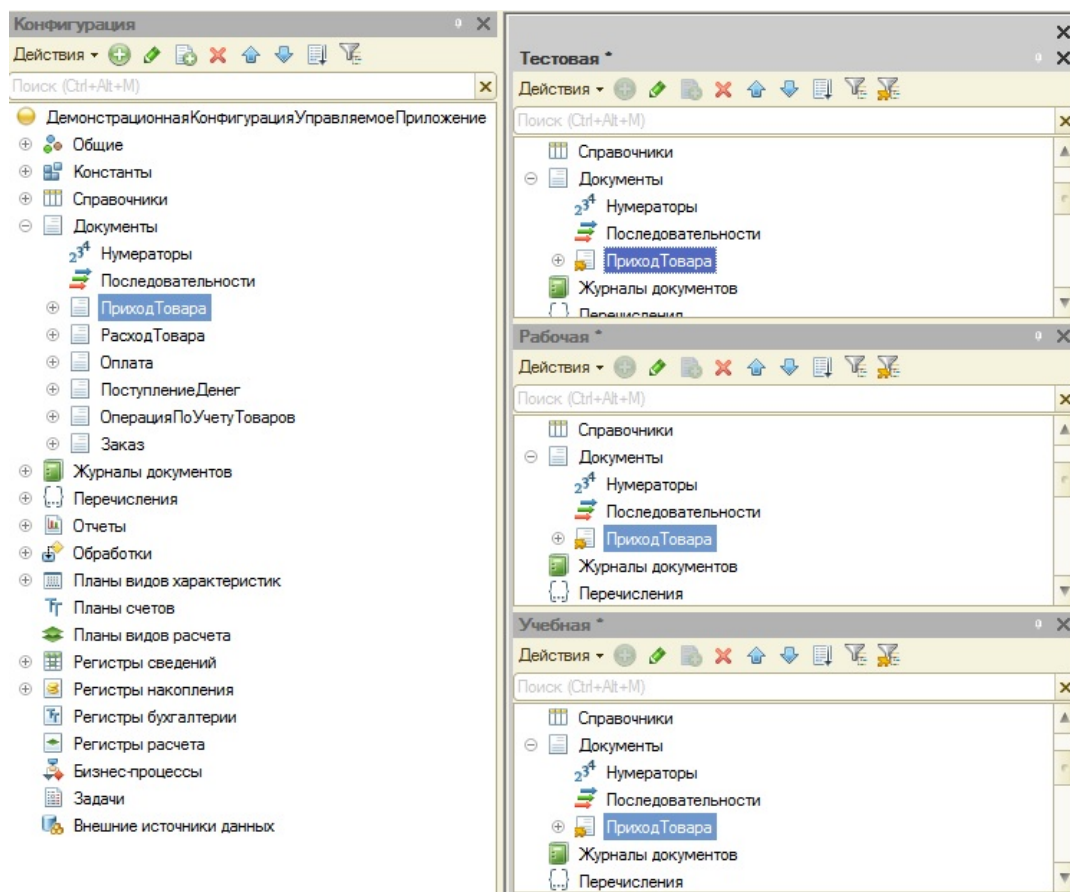


Рис. 7.14. Заимствовали документ «Приход товара» во все расширения

Сейчас во всех расширениях в модуле документа «Приход товара» создадим обработчики перед проведением документа (см. листинг 7.1, 7.2, 7.3)

```
&Перед("ОбработкаПроведения")
Процедура У_Перед_ОбработкаПроведения(Отказ, Режим)
    Сообщить("Сработало в расширении ""Учебная""");
КонецПроцедуры
```

Листинг 7.1. Код в модуле документа «Приход товара» в расширении «Учебная»

```
&Перед("ОбработкаПроведения")
Процедура Р_Перед_ОбработкаПроведения(Отказ, Режим)
    Сообщить("Сработало в расширении ""Рабочая""");
КонецПроцедуры
```

Листинг 7.2. Код в модуле документа «Приход товара» в расширении «Рабочая»


```
&Перед("ОбработкаПроведения")
Процедура Т_Перед_ОбработкаПроведения(Отказ, Режим)
    Сообщить("Сработало в расширении ""Тестовая""");
КонецПроцедуры
```

Листинг 7.3. Код в модуле документа «Приход товара» в расширении «Тестовая»

Обратите внимание: названия процедур у Вас должны быть разные во всех расширениях, поскольку при подключении расширений все процедуры модуля и расширений будут в одном контексте.

Сохраним все расширения, запустим «1С: Предприятие» и проведем документ «Приход товара», чтобы посмотреть, как отработают наши перехваты (см. рис. 7.15).

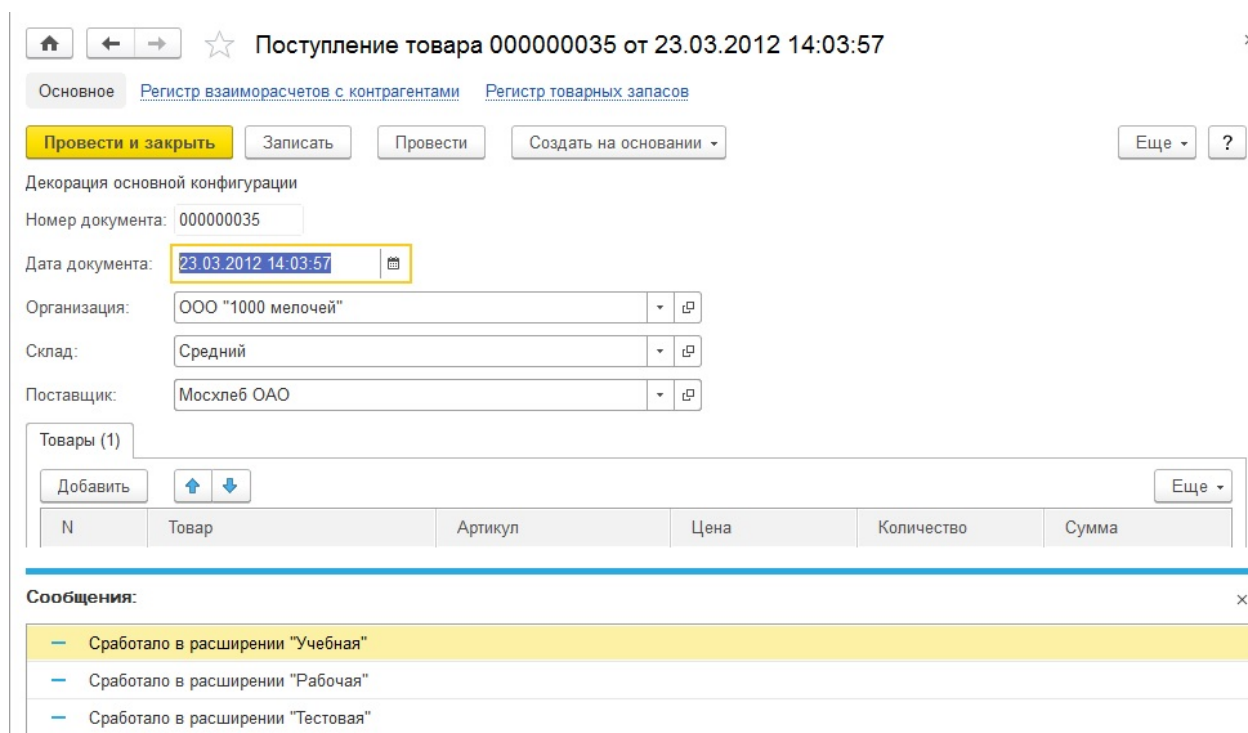


Рис. 7.15. После проведения документа «Приход товара»

Из рисунка 7.15 видно, что расширения отработали в каком-то порядке. Естественно у Вас возникнет вопрос: почему именно в таком? Для ответа на него удалим все расширения, предварительно сохранив, а потом подключим в таком порядке: **Рабочая, Учебная, Тестовая** (см. рис. 7.16). Как это делать, мы уже знаем.

Расширение	Версия	Назначение	Безопасный режим, имя профиля	Защита от опасных действий
Рабочая		Адаптация	<input type="checkbox"/>	<input type="checkbox"/>
Учебная		Адаптация	<input type="checkbox"/>	<input type="checkbox"/>
Тестовая		Адаптация	<input type="checkbox"/>	<input type="checkbox"/>

Рис. 7.16. Новый порядок расширений

Перезайдем в «1С:Предприятие» и заново проведем документ «Приход товара» (см. рис. 7.17).

Поступление товара 000000034 от 25.03.2012 13:16:59

Основное [Регистр взаиморасчетов с контрагентами](#) [Регистр товарных запасов](#)

Провести и закрыть | Записать | Провести | Создать на основании ▾

Декорация основной конфигурации

Номер документа: 000000034

Дата документа: 25.03.2012 13:16:59

Организация: ООО "Все для дома" ▾

Склад: Большой ▾

Валюта взаиморасчетов: Рубли ▾

Поставщик: Пантера АО ▾

Товары (1)

N	Товар	Артикул	Цена	Количество
---	-------	---------	------	------------

Сообщения:

- Сработало в расширении "Тестовая"
- Сработало в расширении "Учебная"
- Сработало в расширении "Рабочая"

Рис. 7.17. После проведения документа «Приход товара»

На рисунке 7.17 видно, что расширения выполнились в следующем порядке (сверху вниз): сначала сработал код в «Тестовой», потом в «Учебной», а потом в «Рабочей».

Посмотрим, в каком порядке будут выполняться расширения, если мы будем перехватывать событие после проведения документа, допишем в модулях соответствующих расширений следующий код (см. листинг 7.4, 7.5, 7.6):

```
&После("ОбработкаПроведения")  
Процедура У_После_ОбработкаПроведения(Отказ, Режим)  
    Сообщить("Сработало после проведения в расширении ""Учебная""");  
КонецПроцедуры
```

Листинг 7.4. Код в модуле документа «Приход товара» в расширении «Учебная»

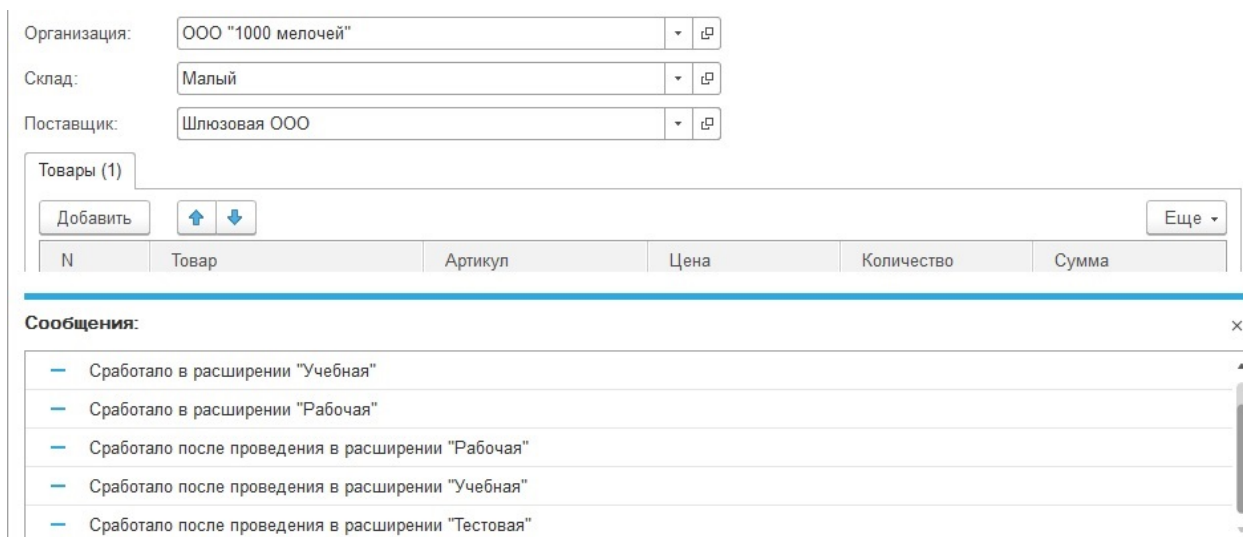
```
& После ("ОбработкаПроведения")  
Процедура Р_После_ОбработкаПроведения(Отказ, Режим)  
    Сообщить("Сработало после проведения в расширении ""Рабочая""");  
КонецПроцедуры
```

Листинг 7.5. Код в модуле документа «Приход товара» в расширении «Рабочая»

```
&После("ОбработкаПроведения")  
Процедура Т_После_ОбработкаПроведения(Отказ, Режим)  
    Сообщить("Сработало после проведения в расширении ""Тестовая""");  
КонецПроцедуры
```

Листинг 7.6. Код в модуле документа «Приход товара» в расширении «Тестовая»

Сохраним все, и опять проведем документ «Приход товара», чтобы посмотреть, в каком порядке обработались события после расширения (см. рис. 7.18).



После проведения сначала сработал код в «Рабочей», потом в «Учебной», а потом в «Тестовой».

Таким образом, выполнение расширений можно представить в виде такой картинки (см. рис. 7.19):

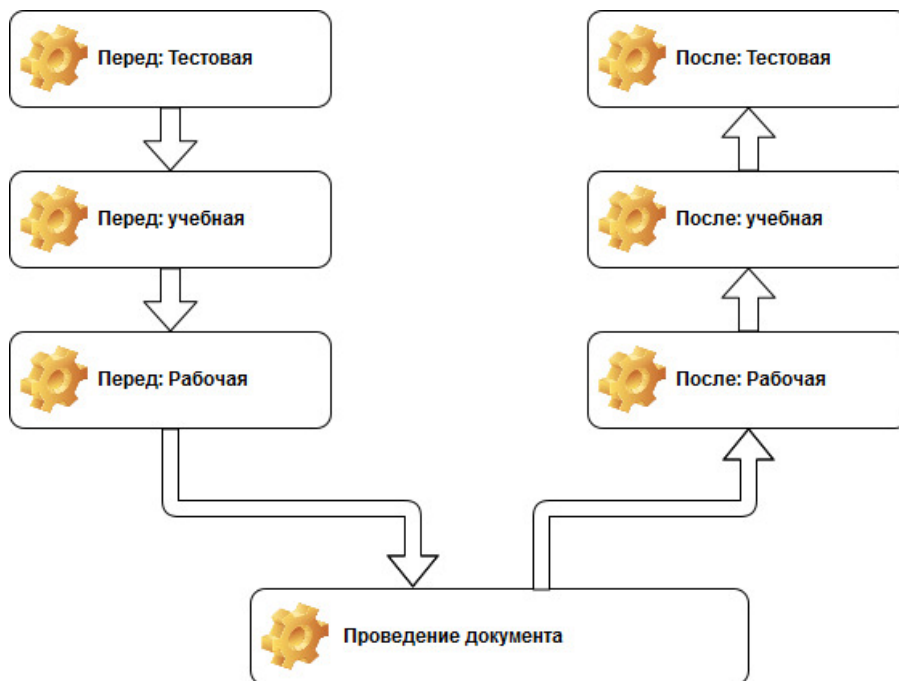


Рис. 7.19. Этапы выполнения наших расширений

На этой картинке мы видим, что ближе всего к ядру, т.е. к процедуре расширенной конфигурации, у нас находится расширение, которое мы подключили самым первым.

Соответствуют ли этой схеме вызовы перехватчиков событий форм? Для того, чтобы это узнать, заимствуем форму документа «Приход товаров» во всех расширениях (см. рис. 7.20).

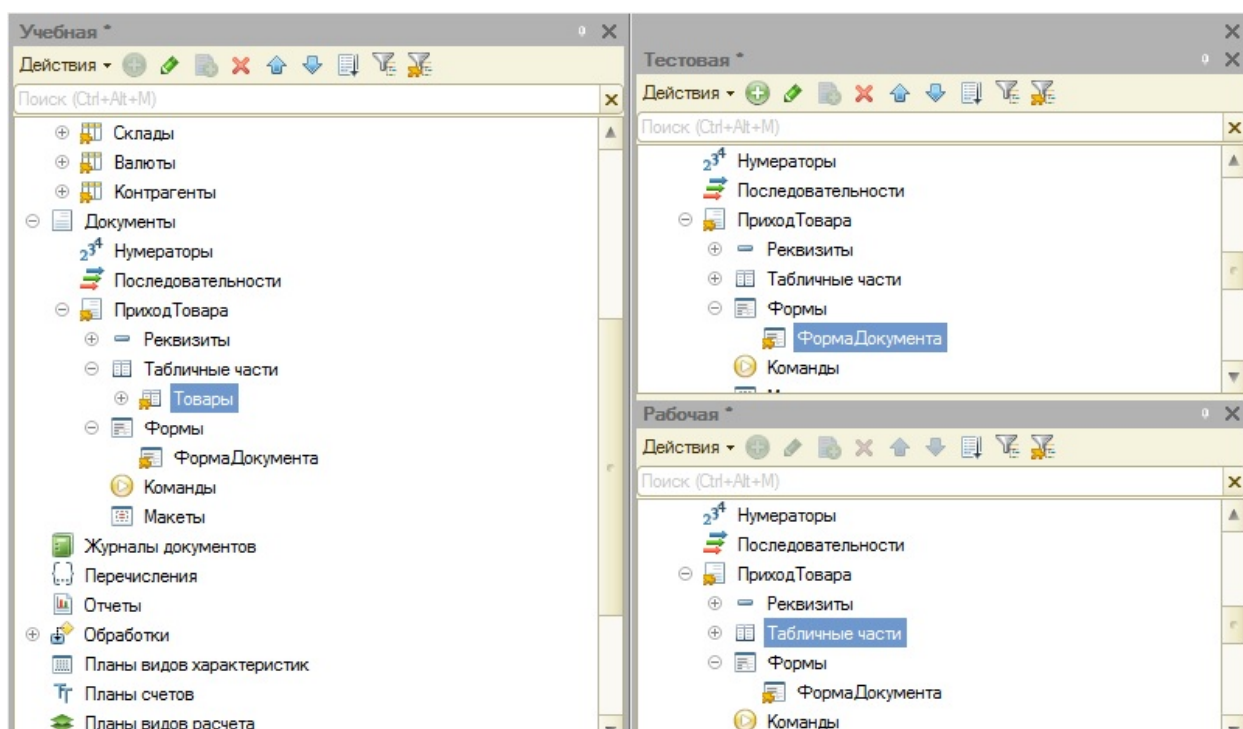


Рис. 7.20. Добавили форму документа «Приход товара» во все расширения

Во всех расширениях вызовем перехватчики события формы «При создании на сервере» перед и после события (см. рис.7.21) у всех расширений и пропишем в них соответствующие сообщения (см. листинги 7.7, 7.8, 7.9)

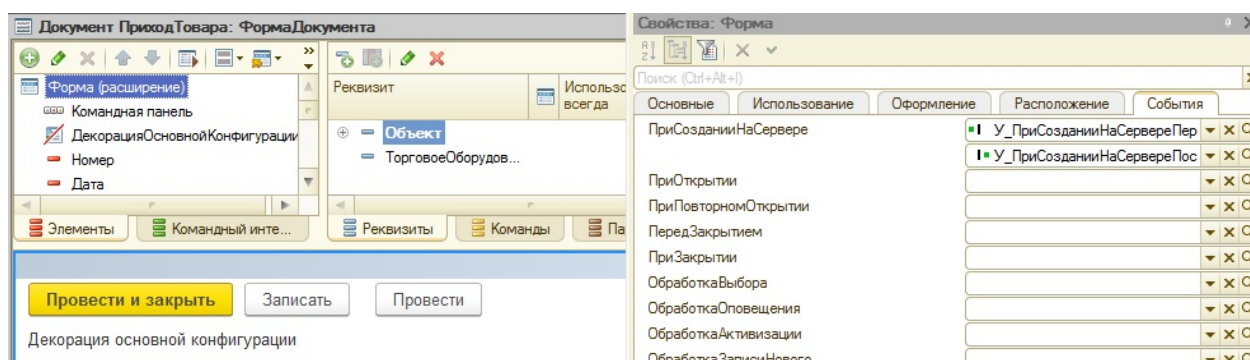


Рис. 7.21. События на форме документа «Приход товара» в расширении «Учебная»

```
&НаСервере
Процедура У_ПриСозданииНаСервереПеред(Отказ, СтандартнаяОбработка)
    Сообщить("Вызвали перед событием в учебном расширении");
КонецПроцедуры
```

```
&НаСервере
Процедура У_ПриСозданииНаСервереПосле(Отказ, СтандартнаяОбработка)
    Сообщить("Вызвали после события в учебном расширении");
КонецПроцедуры
```

Листинг 7.7. Обработка событий формы документа
«Приход товара» в расширении «Учебная»

```
&НаСервере
Процедура Р_ПриСозданииНаСервереПеред(Отказ, СтандартнаяОбработка)
    Сообщить("Вызвали перед событием в рабочем расширении");
КонецПроцедуры
```

```
&НаСервере
Процедура Р_ПриСозданииНаСервереПосле(Отказ, СтандартнаяОбработка)
    Сообщить("Вызвали после события в рабочем расширении");
КонецПроцедуры
```

Листинг 7.8. Обработка событий формы документа «Приход
товара» в расширении «Рабочая»

```
&НаСервере
Процедура Т_ПриСозданииНаСервереПеред(Отказ, СтандартнаяОбработка)
    Сообщить("Вызвали перед событием в тестовом расширении");
КонецПроцедуры
```

```
&НаСервере
Процедура Т_ПриСозданииНаСервереПосле(Отказ, СтандартнаяОбработка)
    Сообщить("Вызвали после события в тестовом расширении");
КонецПроцедуры
```

Листинг 7.8. Обработка событий формы документа
«Приход товара» в расширении «Тестовая»

Сохраним все расширения, обновим базу данных и посмотрим, как будут они подключаться при создании формы (см. рис. 7.22).

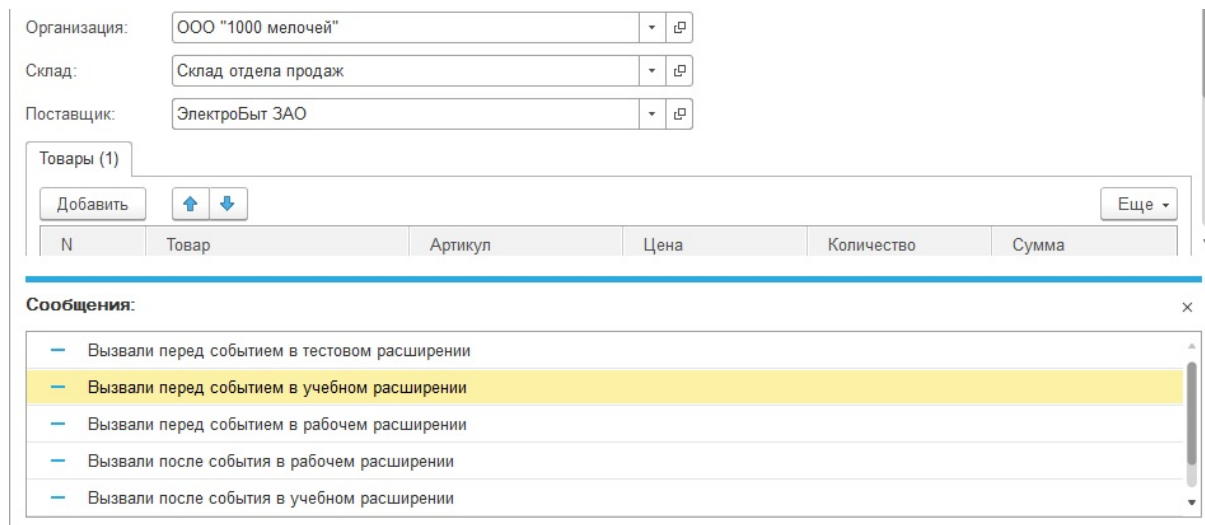


Рис. 7.22. Отработка событий при открытии формы документа

Все сработало согласно картинке 7.19.

На этом мы закончим главу, которая посвящена работе с несколькими расширениями. В следующей главе разберем работу прав доступа в расширении.

Глава 8. Расширения и права доступа

Механизм платформы 1С 8.3. позволяет заимствовать в расширения уже созданные роли основной (расширяемой) конфигурации, а также создавать собственные роли в расширении. В этой главе мы рассмотрим некоторые особенности работы инструмента расширений с ролями.

Научимся заимствовать роли. В учебной базе «Управляемое приложение» создадим специальное расширение для работы с ролями конфигурации, которое назовем «Правка ролей» (см. рис. 8.1).

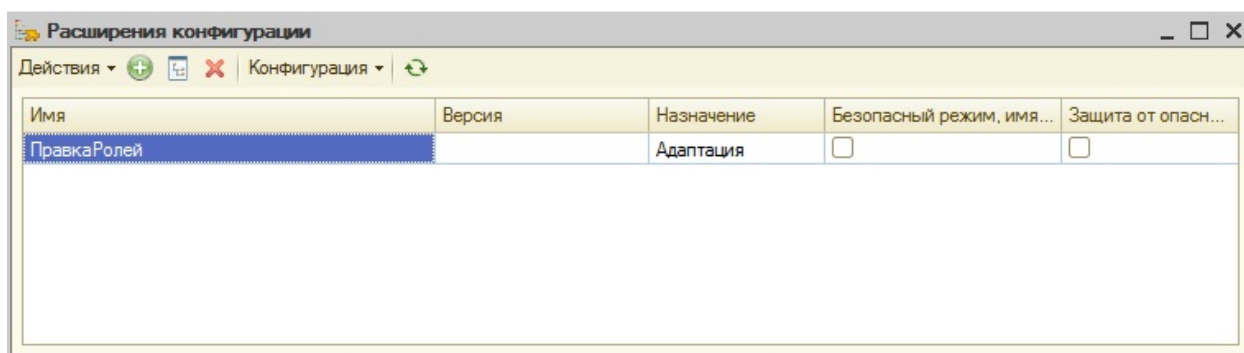


Рис. 8.1. Расширение «Правка ролей»

Выполним небольшую задачу, а на её основе разберем механизм заимствования ролей. В расширяемой конфигурации есть роль «Менеджер по закупкам» и у этой роли есть право на перепроведение (сохранение проведенного) документа «Приход товара», которое называется «Интерактивное изменение проведенных» (см. рис. 8.2).

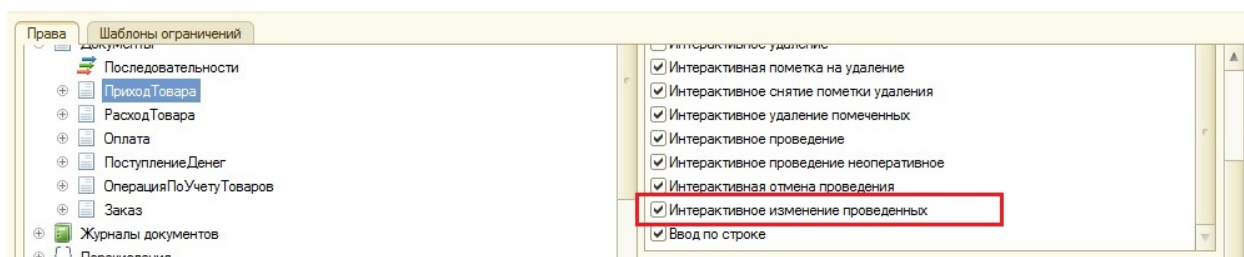


Рис. 8.2. Право на перепроведение документа «Приход товара»

Уберем право на интерактивное перепроведение документа «Приход товара» у роли «Менеджер по закупкам». Сделаем это посредством расширения конфигурации «Правка ролей».

Заимствуем в наше расширение роль «Менеджер по закупкам» (см. рис. 8.3).

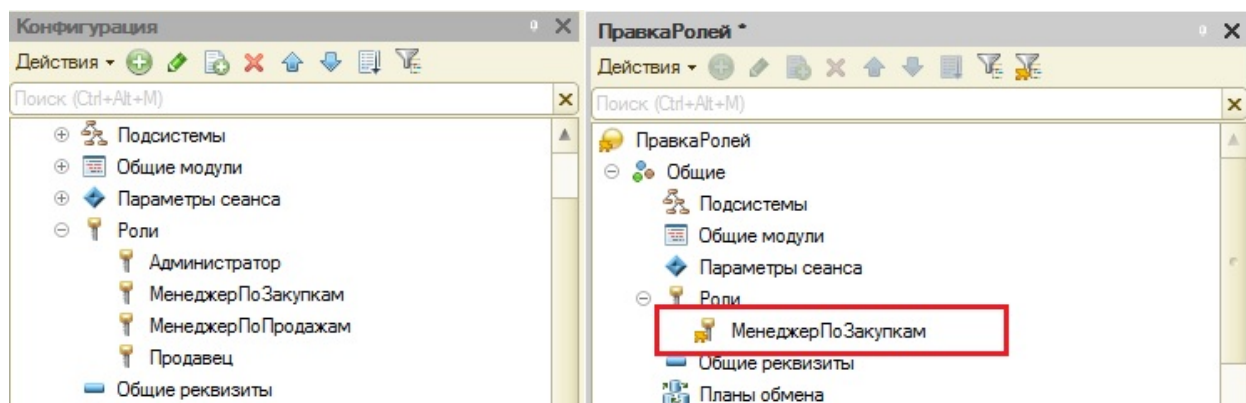


Рис. 8.3. Заимствовали роль «Менеджер по закупкам»

Если мы сейчас откроем эту роль и посмотрим на её состав, то заметим, что никаких метаданных у этой роли нет (см. рис. 8.4).

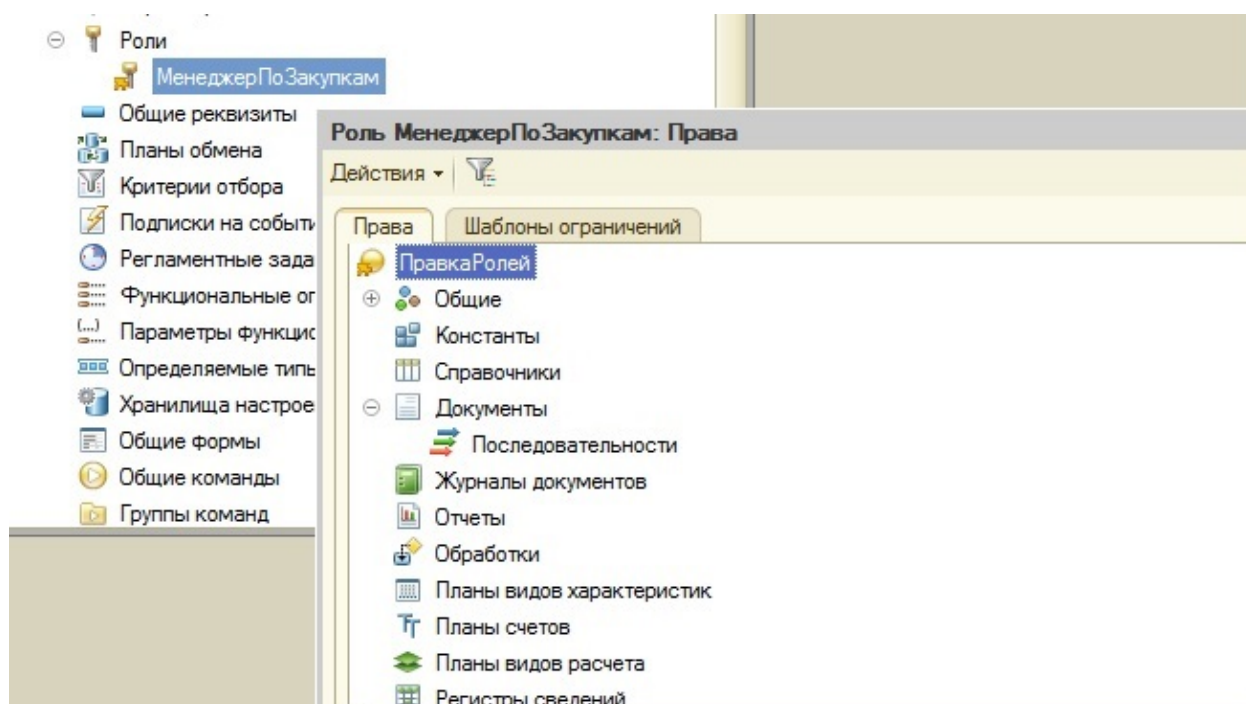


Рис. 8.4. Состав роли «Менеджер по закупкам» расширения конфигурации

Поскольку мы хотим изменить принцип работы с документом «Приход товара», то необходимо заимствовать этот документ в расширение (см. рис. 8.5), тогда он автоматически появится в составе роли (см. рис. 8.6).

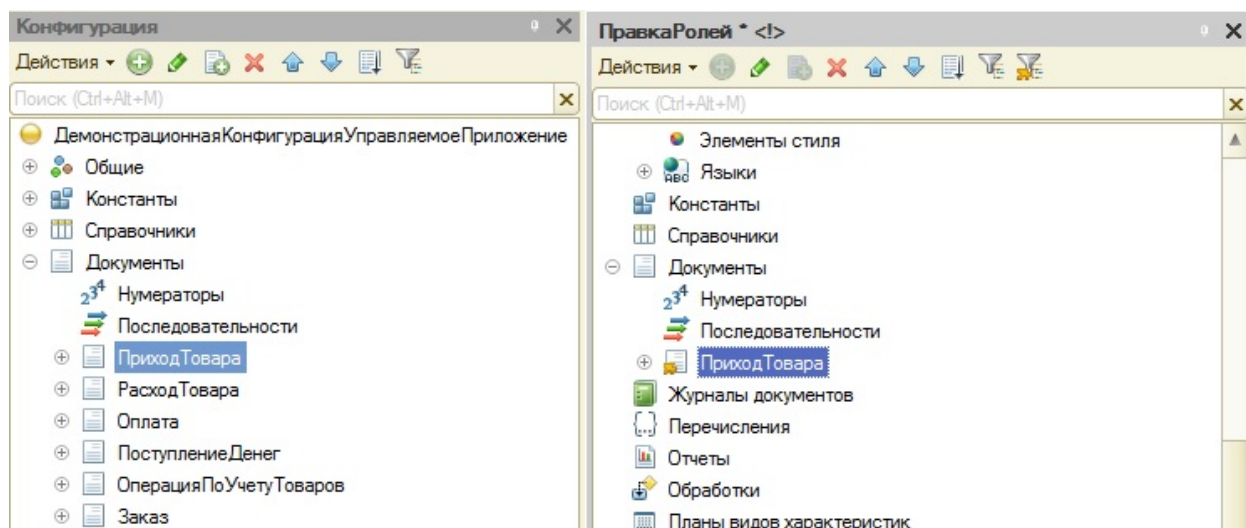


Рис. 8.5. Добавили документ «Приход товара»

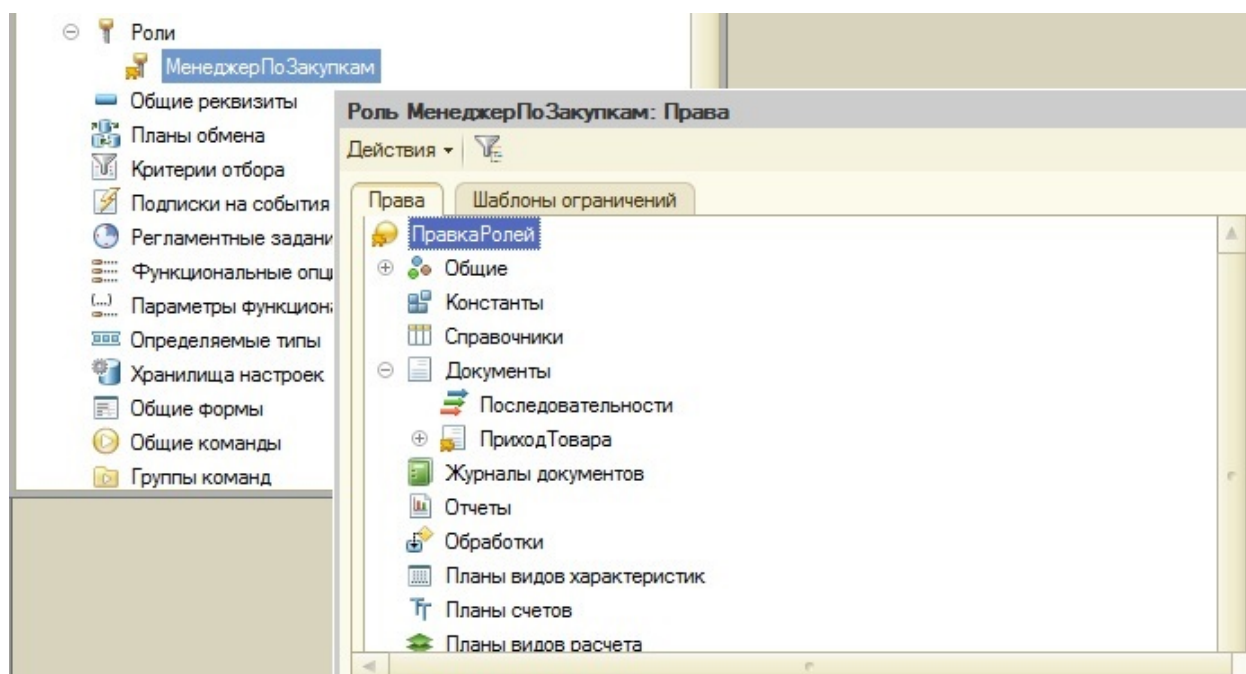


Рис. 8.6. Документ «Приход товара» в составе роли «Менеджер по закупкам»

Если мы сейчас посмотрим на права документа «Приход товара» в заимствованной роли, то заметим, что все права у данной роли «затемнены» (см. рис. 8.7), в отличие от прав для роли расширяемой конфигурации (см. рис. 8.8). «Затемненный» флаг права - это третье состояние, которое есть только у прав в расширениях конфигурации, которое означает, что значение права берется из основной (расширяемой) конфигурации. Без разницы, установлено право или нет. Если право для данного объекта стоит в роли, то значит и в расширенной роли у данного объекта это право будет установлено. Если право снято, то и в расширенной роли это право будет снято.

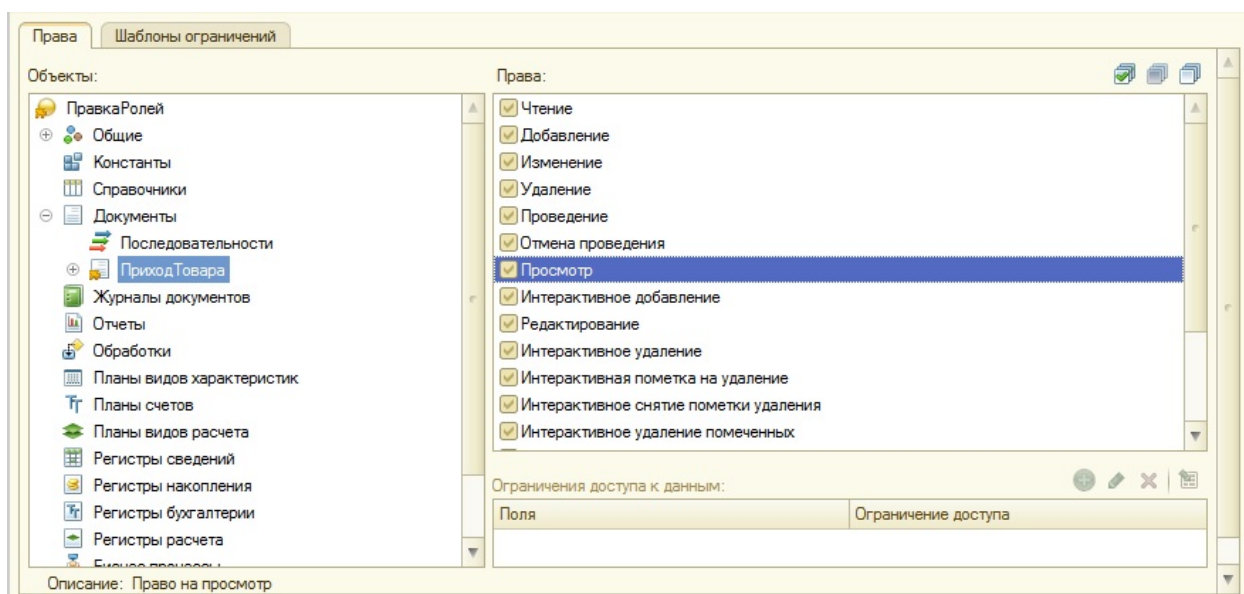


Рис. 8.7. Набор прав у роли в расширении конфигурации

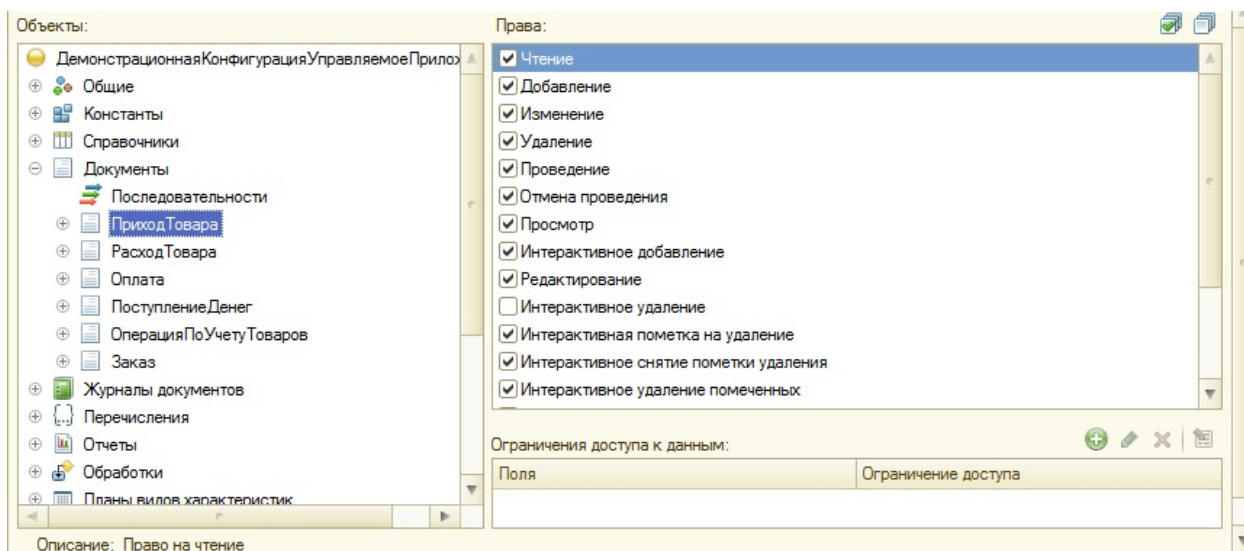


Рис. 8.8. Набор прав у роли в основной конфигурации

Если мы хотим, чтобы у роли в расширении какое-то право было установлено отличное от роли основной (расширяемой) конфигурации, то необходимо либо поставить, либо снять флаг у нужной роли. Мы хотели запретить пользователю интерактивно проводить документ «Приход товара», поэтому у роли в расширении снимем флаг с права «Интерактивное изменение проведенных» (см. рис. 8.9).

Теперь если расширение будет подключено, то пользователь с ролью «Менеджер по закупкам» не сможет перепроводить этот документ.

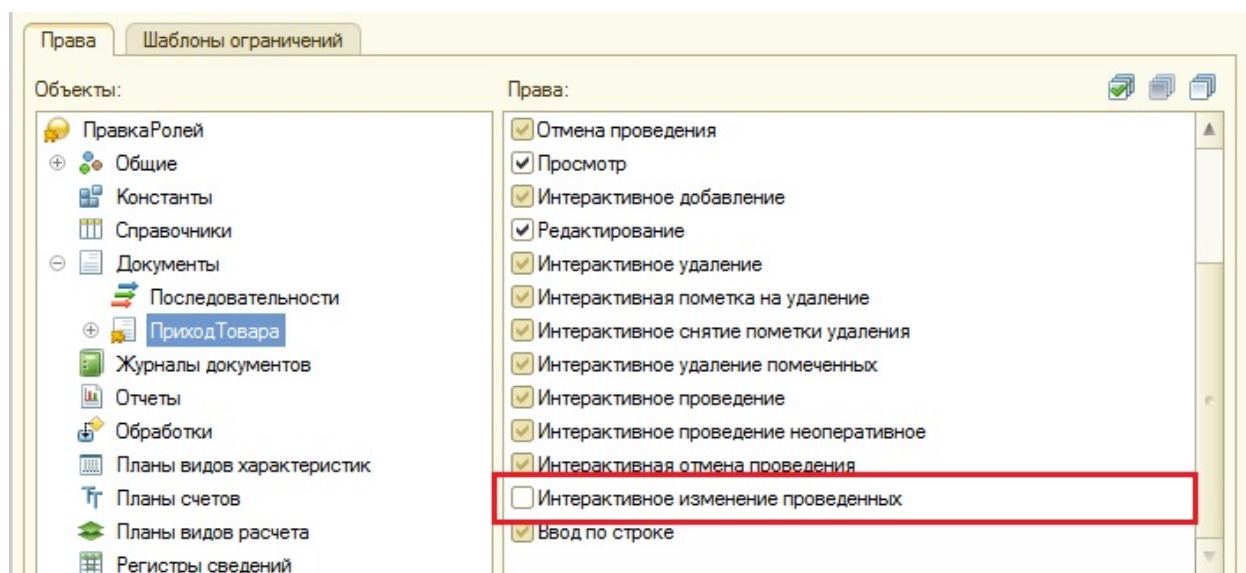


Рис. 8.9. Сняли прав «Интерактивное изменение проведенных» в заимствованной роли

Таким образом можно резюмировать: когда в роли расширения установлен серый флаг, то при подключенном расширении право определяется из роли основной (расширяемой) конфигурации. А когда флаг установлен или снят, то право определяется уже из роли расширения конфигурации.

Можно для всех прав объекта заимствованной роли установить третье состояние («затемненное»), нажав на кнопку «Затемнить все флажки» (см. рис. 8.10).

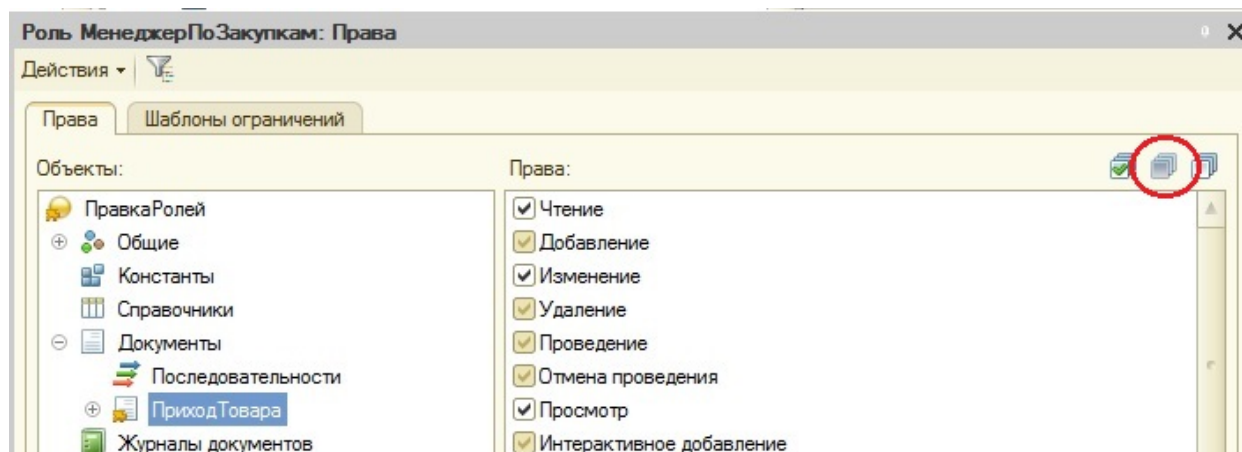


Рис. 8.10. Кнопка «Затемнить все флажки»

Мы научились использовать роли из расширения конфигурации, но в расширении можно создавать собственные роли. Правда, добавить эту роль к пользователю можно только программным способом. Для того чтобы понять, как

это делать, реализуем следующую задачу: создадим в новом расширении «Правка ролей» отчет, который назовем «Отчет для менеджера по закупкам» (схема и форма будут пустая, он нужен только для того, чтобы понять, как работает этот механизм). Чтобы этим отчетом можно было пользоваться, создадим в расширении собственную роль «Доступ к отчету», также в расширении создадим обработку «Установить роль для пользователя», в которой будем выбранному пользователю добавлять собственную роль из расширения.

Приступим. Первым делом создадим отчет, который назовем «Отчет менеджера по закупкам» (см. рис. 8.11). Для этого отчета создадим схему компоновки без набора данных и форму, для того, чтобы он отобразился в подсистеме.

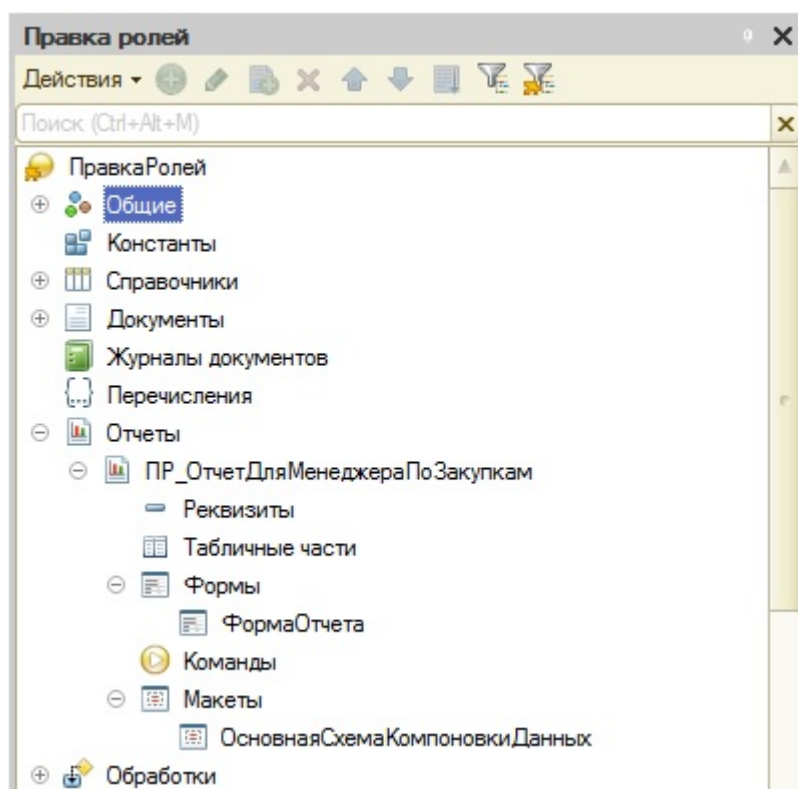


Рис. 8.11. Создали отчет «Отчет менеджера по закупкам»

Пусть этот отчет отображается в подсистеме «Закупки», для этого заимствуем эту подсистему в наше расширение и добавим вновь созданный отчет в её состав (см. рис. 8.12).

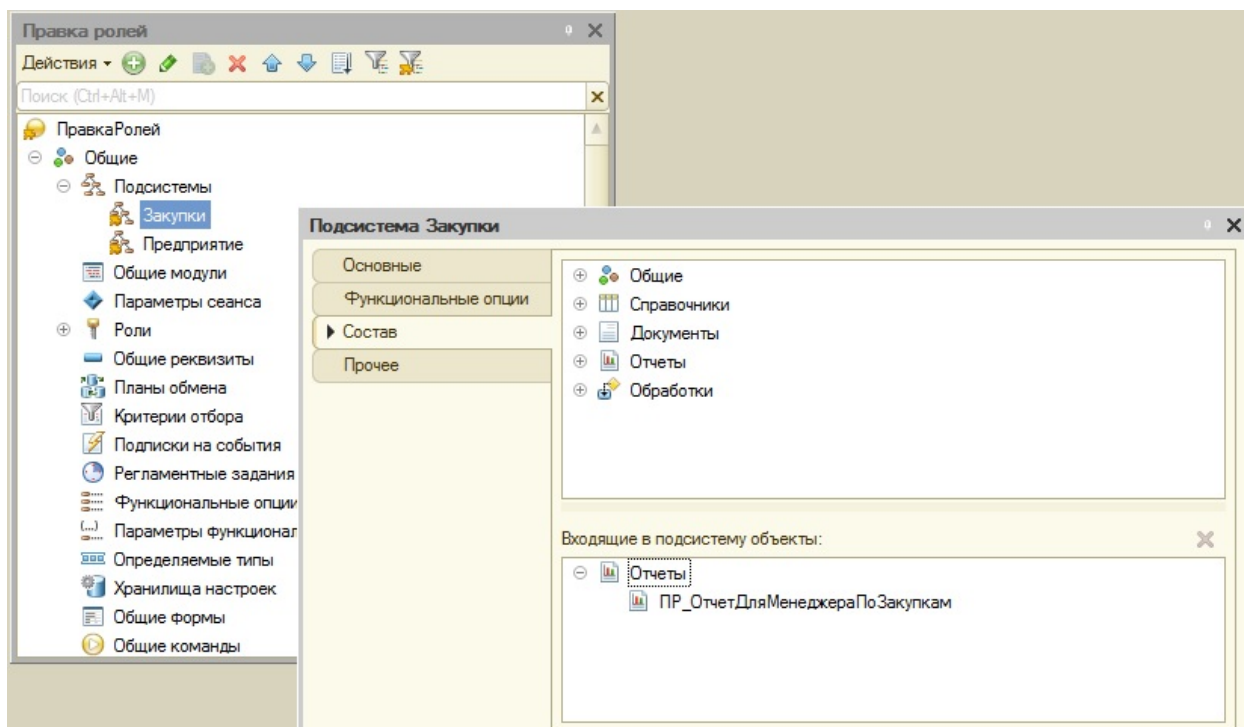


Рис. 8.12. Заимствовали подсистему «Закупки» и добавили в ее состав новый отчет

Теперь создадим роль «Доступ к отчету» и у этой роли установим право на использование и просмотр вновь созданного отчета (см. рис. 8.13).

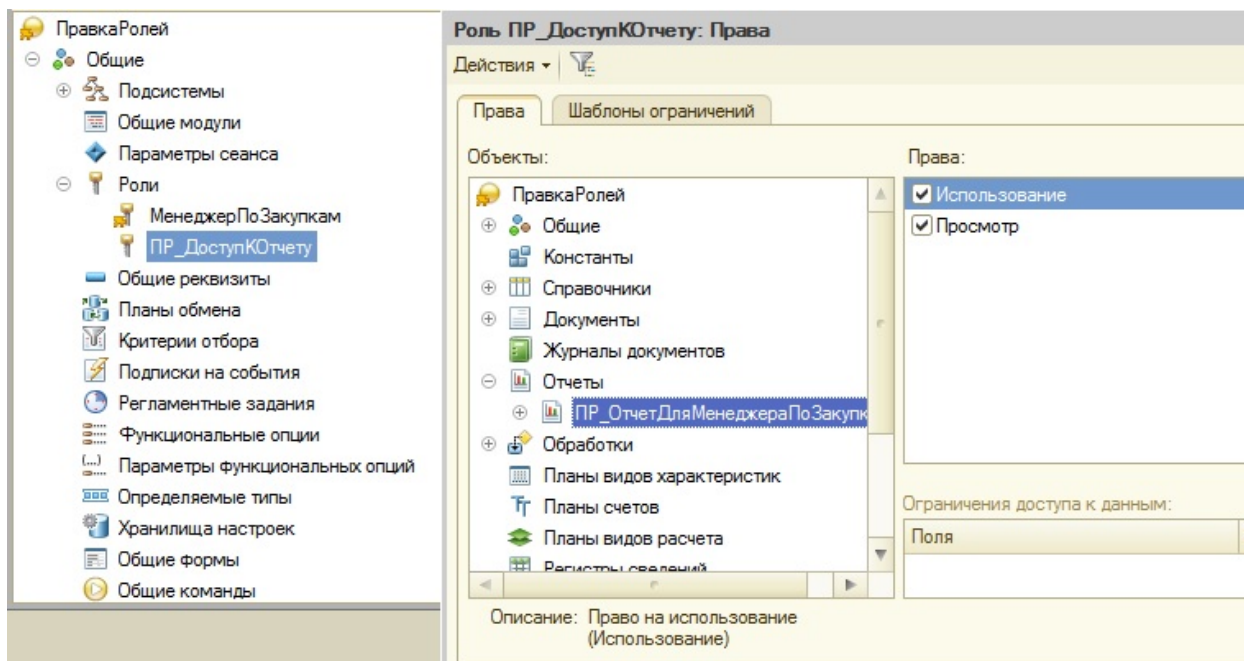


Рис. 8.13. Создание новой роли «Доступ к отчету»

Все мы создали отчет и роль, которая дает доступ к этому отчету, но мы не сможем в конфигураторе привязать эту роль к нужному пользователю, её просто не будет в списке, после того как мы обновим базу данных. Привязать роль к нужному пользователю можно только при помощи встроенного языка. Для этого необходимо создать обработку, в которой будем выбирать нужного пользователя, и привязывать новую роль «Доступ к отчету» к этому пользователю. Создадим эту обработку и форму для неё, где разместим реквизит «Пользователь» и команду «Установить» (см. рис. 8.14).

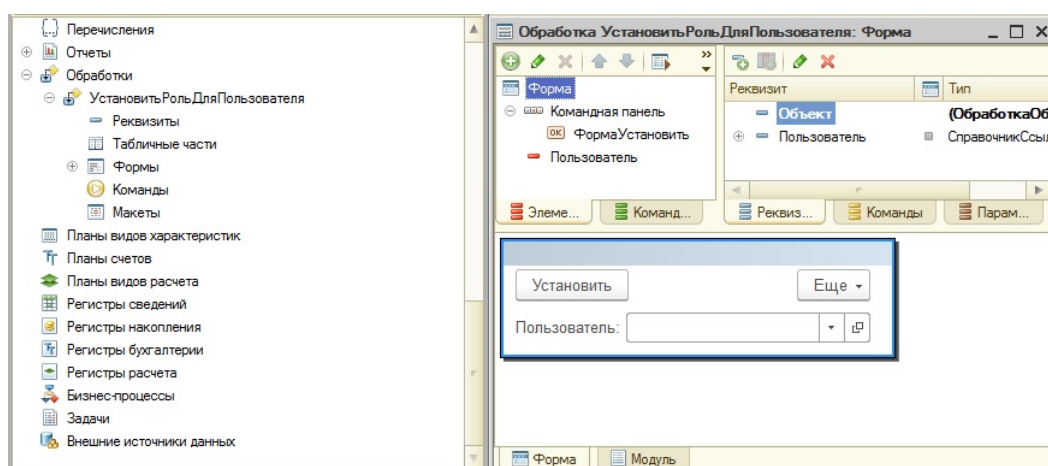


Рис. 8.14. Создание новой обработки

В обработчике команды «Установить», напишем код, где будем добавлять новую роль из расширения выбранному пользователю (см. листинг 8.1).

&НаСервере

Процедура УстановитьНаСервере()

```

ПользовательДляУстПрав =
ПользователиИнформационнойБазы.НайтиПоИмени(Пользователь.Наименование);
РольДоступКОтчету = Метаданные.Роли.Найти("ПР_ДоступКОтчету");
Роли = ПользовательДляУстПрав.Роли;
Роли.Добавить(РольДоступКОтчету);
ПользовательДляУстПрав.Записать();

```

КонецПроцедуры

&НаКлиенте

Процедура Установить(Команда)

УстановитьНаСервере());

КонецПроцедуры

Листинг 8.1. Обработчик команды «Установить»

Осталось сделать так, чтобы новая обработка отображалась в «1С:Предприятии» в пользовательском режиме. Для этого заимствуем роль «Администратор» и добавим в состав этой роли право на выполнение обработки (см. рис. 8.15), а также заимствуем подсистему «Предприятие» и добавим в состав подсистемы новую обработку (см. рис. 8.16).

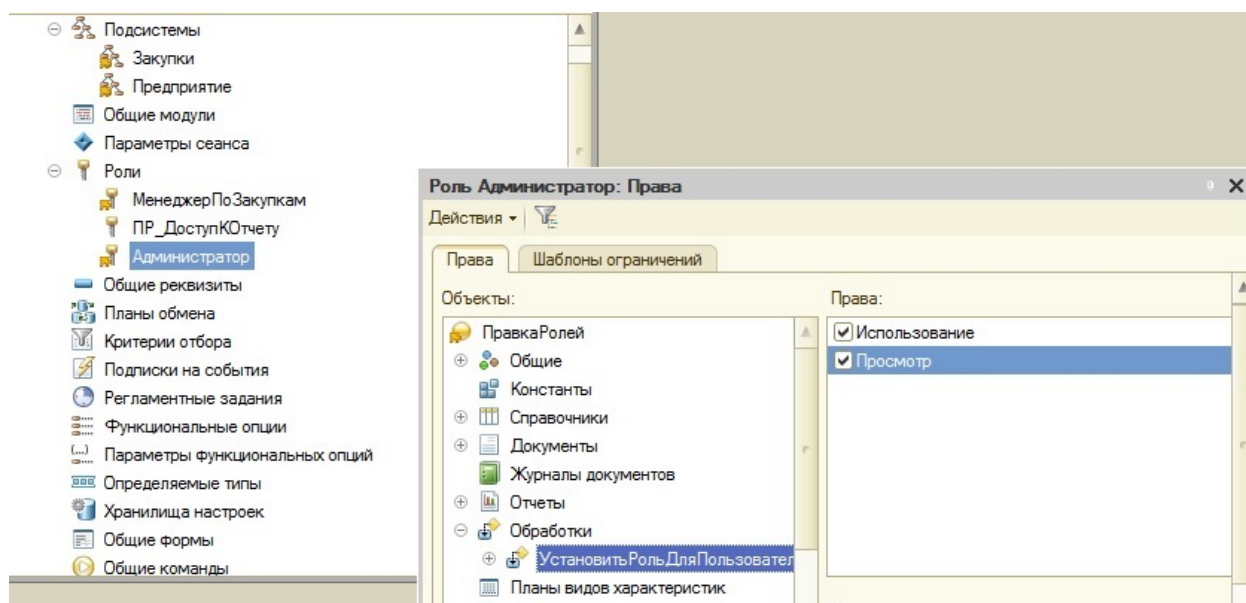


Рис. 8.15. Заимствовали роль «Администратор» и привязали к ней новую обработку

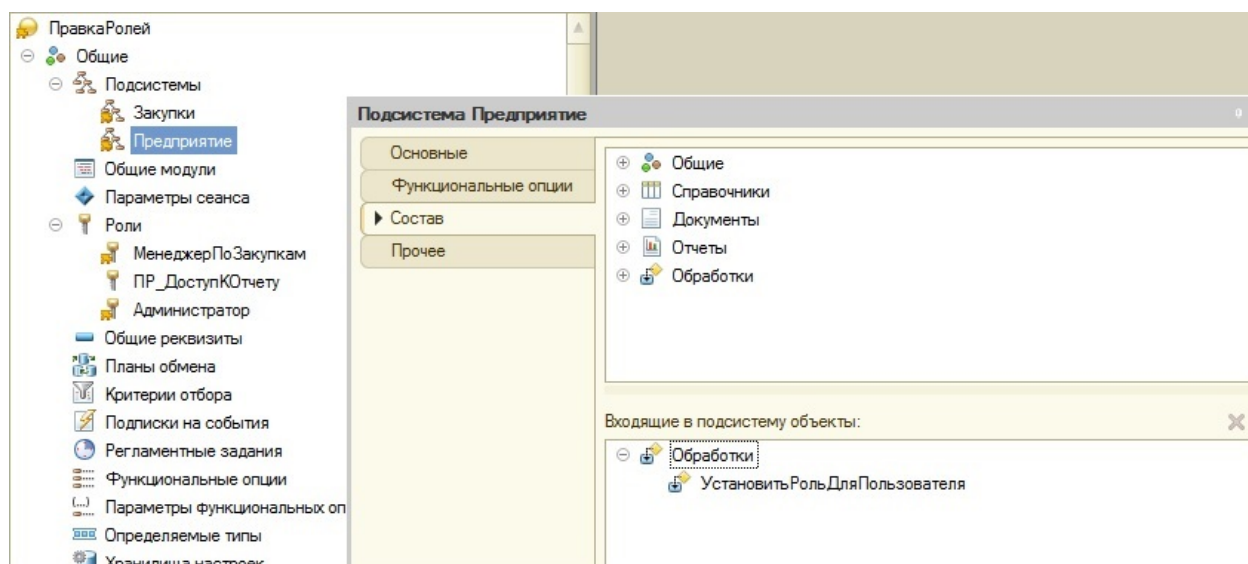


Рис. 8.16. Заимствовали подсистему «Предприятие» и добавили в её состав новую обработку

Сохраним расширение и обновим базу данных. После запуска «1С: Предприятия» в пользовательском режиме мы обнаружим обработку в разделе «Предприятие» (см. рис. 8.17).

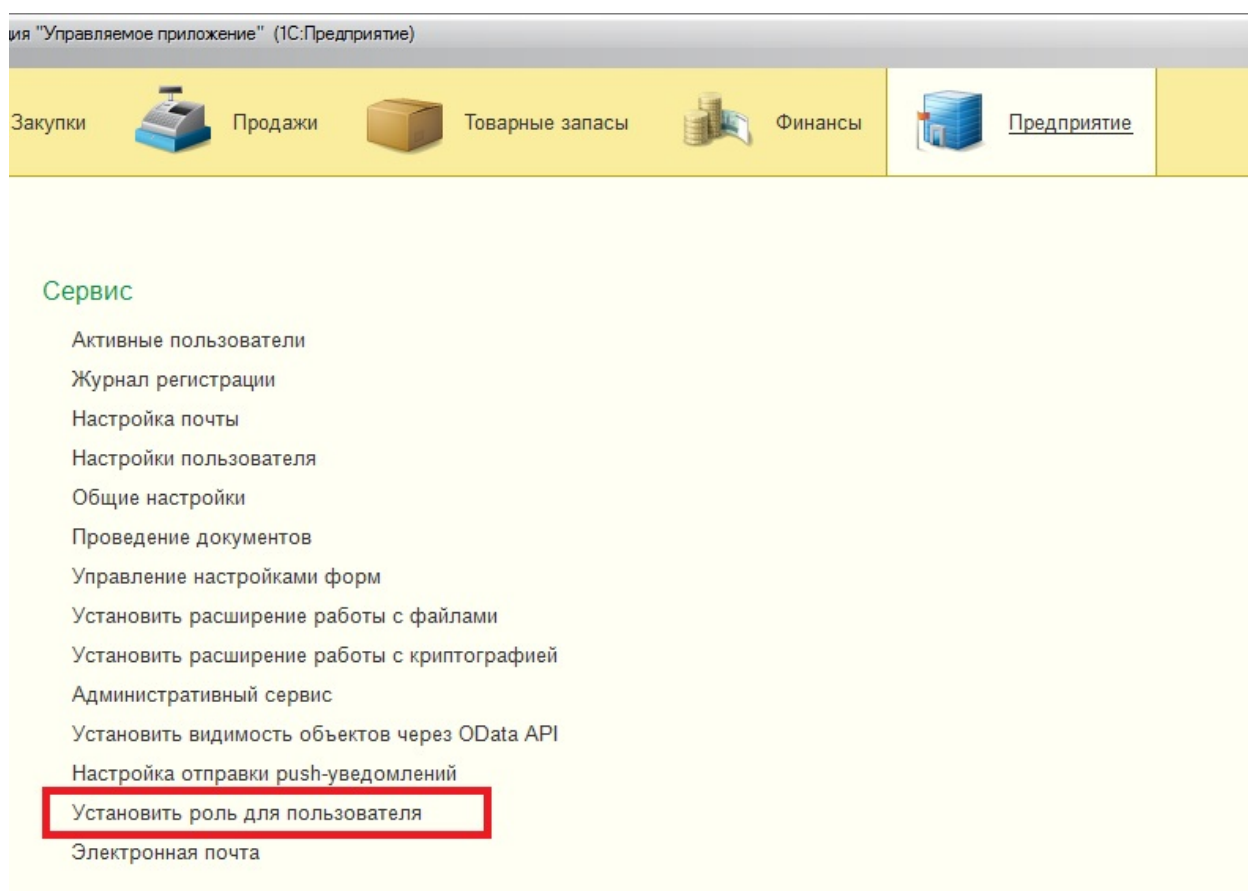


Рис. 8.17. Обработка в меню раздела «Предприятие»

Откроем обработку, выберем пользователя «Администратор» и выполним команду «Установить» (см. рис. 8.18).

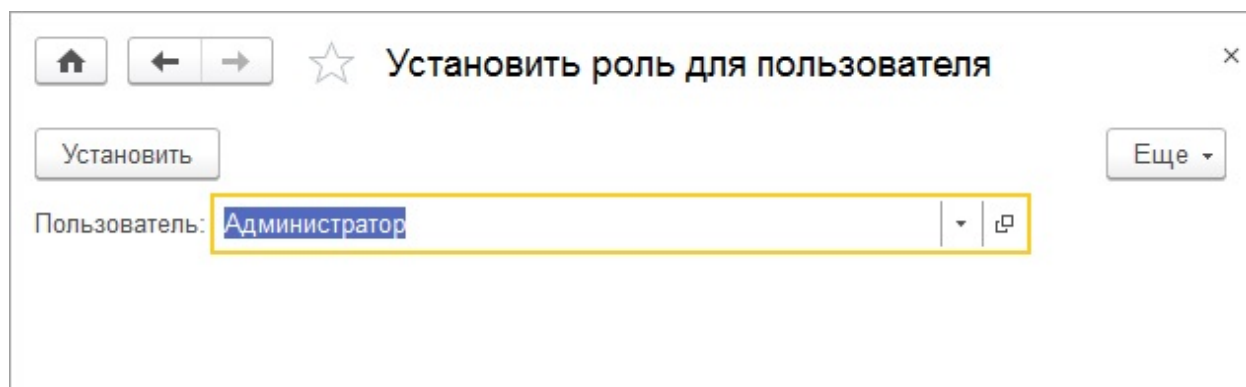


Рис 8.18. Устанавливаем роль пользователю «Администратор»

После перезапуска «1С: Предприятия» в разделе «Закупки» появится новый отчет (см. рис. 8.19).

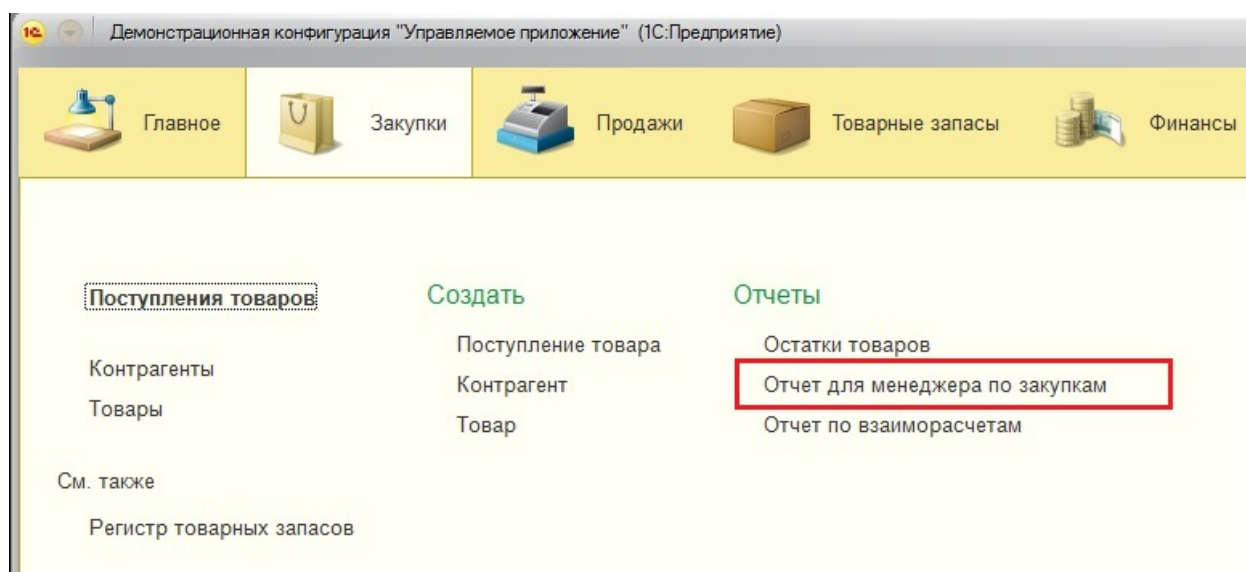


Рис. 8.19. Новый отчет в разделе «Закупки»

А если мы посмотрим на пользователя «Администратор» в списке пользователей, то обнаружим, что у него появился специфичный знак, и в списке ролей добавилась соответствующая надпись (см рис. 8.20).

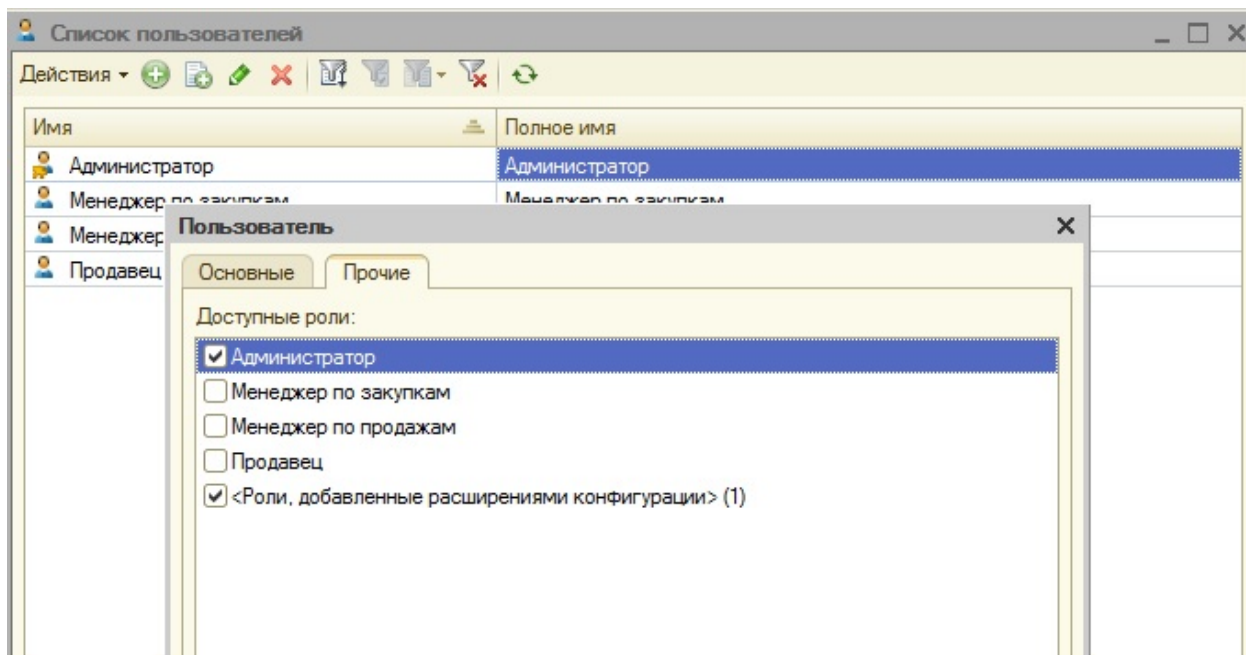


Рис. 8.20. Список ролей пользователя «Администратор»

На этом мы закончим изучать работу расширений с ролями. Как Вы уже знаете, в расширения можно заимствовать имеющиеся роли, а также создавать новые, правда, в этом случае придется программно обрабатывать их добавление пользователю.

Глава 9. Практическое использование механизмов перехватов

В этой главе я разберу некоторые практические примеры, где можно использовать механизм перехвата событий расширений.

Все примеры будут показаны на демонстрационной конфигурации «Управляемое приложение», априори предполагается, что её нельзя изменять.

Напомню Вам, что перехват событий и методов работает только тогда, когда режим совместимости установлен в «Не использовать» (так было для платформы 8.3.10.1981). Как Вы должны помнить из предыдущих глав, в расширении конфигурации можно перехватывать типовые процедуры модуля объекта. Имеется возможность сделать три варианта перехвата «типовой» процедуры: «перехватить» событие перед выполнением метода (с помощью аннотации «&Перед»), «перехватить» событие после выполнения метода (с помощью аннотации «&Перед»), а также выполнить вместо типовой процедуры процедуру расширения (с помощью аннотации «&Вместо»).

В задачах ниже мы разберем работу «перехватчиков» модуля объекта.

Задача: В расширении создадим собственный макет печати документа «Расход товара». И добавим команду на форму расходной накладной «Печать расходной накладной (расширение)», при выполнении которой будет печататься макет из расширения.

Для этого создадим расширение конфигурации «ДляРасходаТовара», заимствуем в это расширение документ «Расход товара», а также форму (см. рис. 9.1).

Скопируем (именно копируем, а не заимствуем) из расширяемой конфигурации макет «МакетПечати», переименуем его и изменим сам макет, чтобы можно было отличать (см. рис. 9.2).

Теперь нам нужно скопировать (тоже именно скопировать, а не заимствовать) команду документа «Печать расходной накладной», но группа, в которую входит команда, у нас не копируется, поэтому перед этим необходимо заимствовать группу «Печать» (см. рис. 9.3).

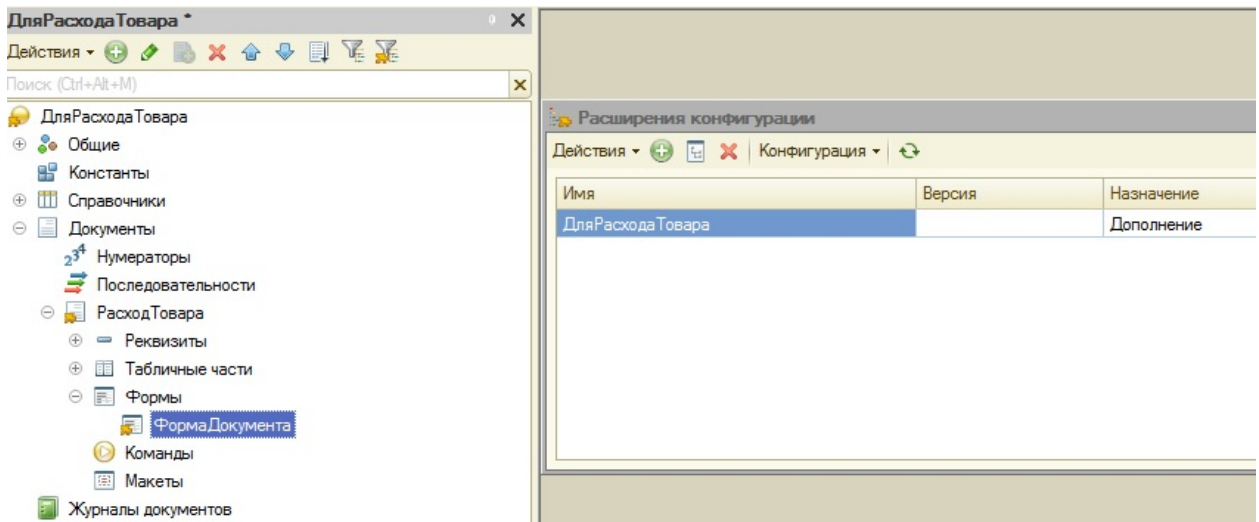


Рис. 9.1. Создание расширения и заимствования документа «Расход товара» с формой

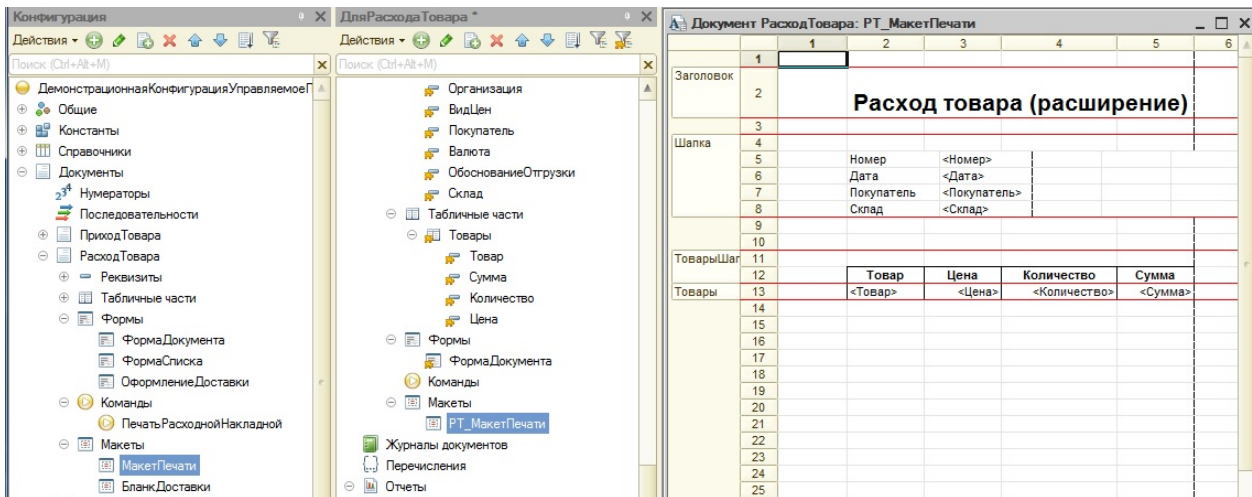


Рис. 9.2. Копирование макета «Макет печати»

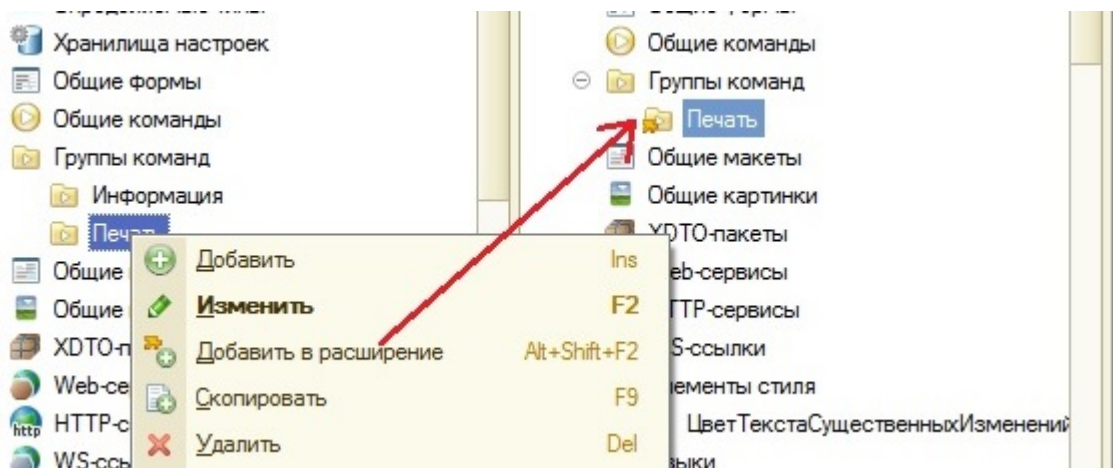


Рис. 9.3. Заимствование группы команд

Следующим шагом копируем команду «Печать расходной накладной», и также переименовываем, и не забудем указать тип параметра команды – ссылку на документ «Расход товаров».

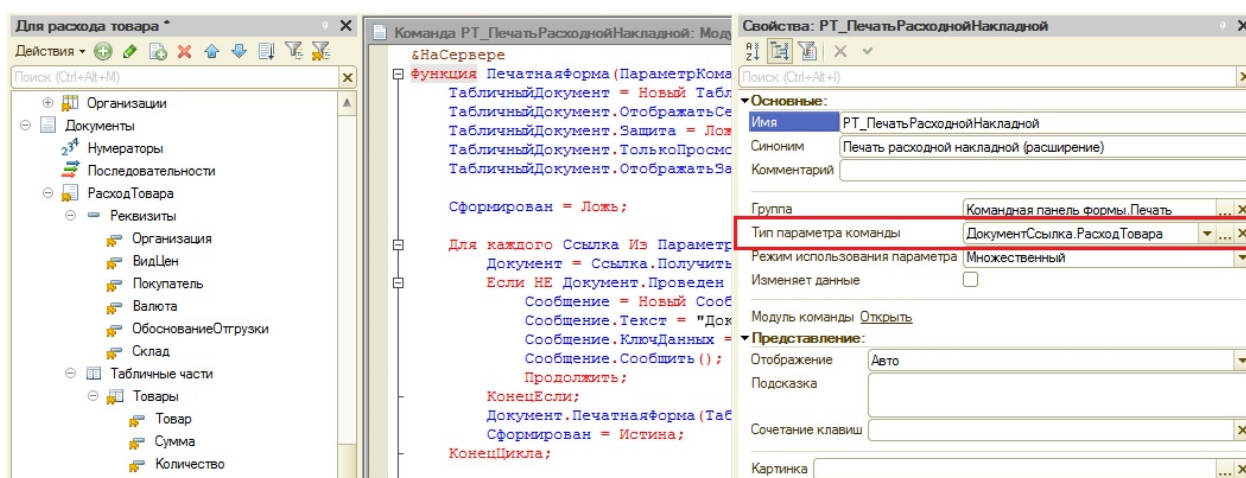


Рис. 9.4. Копирование команды «Печать расходной накладной»

Если мы на этом этапе сохраним расширение и обновим базу данных, то при открытии формы документа увидим, что команда добавилась.

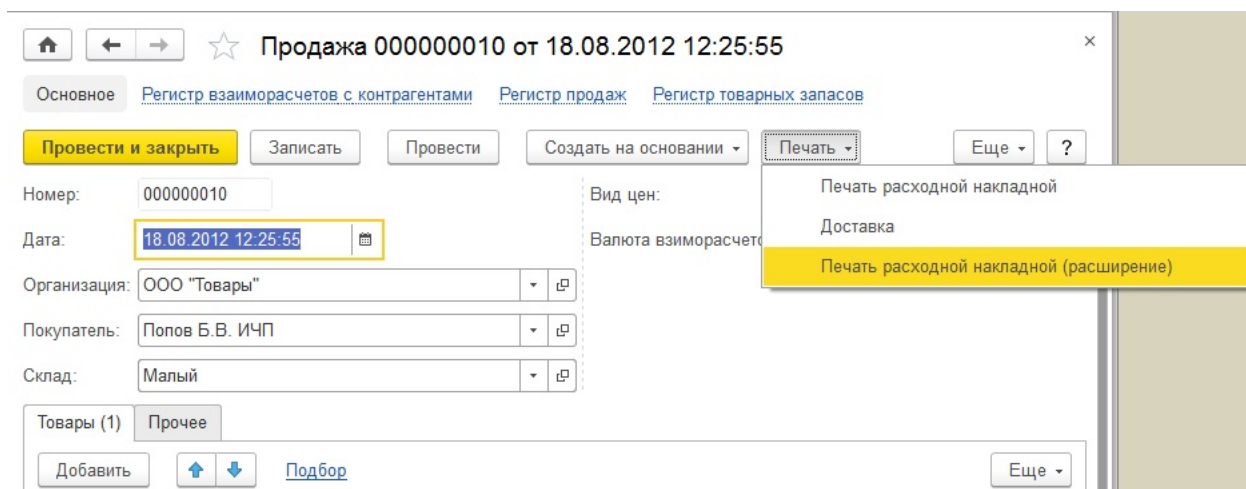


Рис. 9.5. Команда добавилась в интерфейс

Но если мы сейчас попробуем её выполнить, то выйдет обычная печатная форма, не из расширения.

Всё из-за того, что в команде расширения используется метод документа «ПечатнаяФорма», в котором и «подтягивается» нужный макет (см. рис. 9.6).

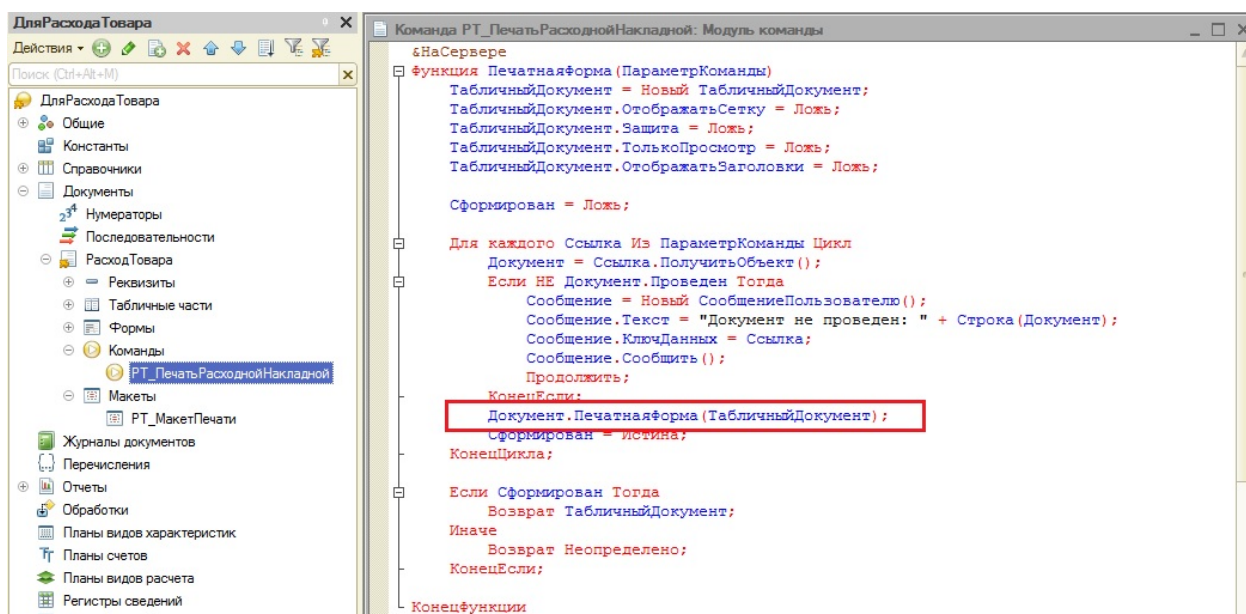


Рис. 9.6. Метод «Печатная форма» документа «Расход товара»

Перехватим процедуру «ПечатнаяФорма» модуля объекта, и переделаем её. Для того чтобы понимать, что мы используем табличный документ из расширения, будем из расширения передавать в перехваченную процедуру структуру с табличным документом. А потом в перехваченной процедуре «ПечатнаяФорма» в зависимости от типа параметра будем использовать тот или иной макет: если параметр имеет тип «ТабличныйДокумент», то будем использовать типовой макет, а если параметр имеет тип «Структура», то будем использовать макет из расширения. Для того, чтобы перехватить процедуру или функцию в модуле объекта необходимо написать аннотацию «&Вместо» и после неё в скобочках название перехваченного метода.

Напишем в модуле документа «Расход товара» расширения следующий код, см. листинг 9.1.

```
&Вместо("ПечатнаяФорма")
Процедура RT_ПечатнаяФорма(ТабличныйДокумент)
КонiecПроцедуры // ПечатнаяФорма(ТабличныйДокумент)
```

Листинг 9.1. Код в модуле документа «Расход товара»

Обратите внимание на название метода, он не должен совпадать с основным, поскольку при работе процедуры модулей расширяемой конфигурации и расширений работают в одном контексте.

Теперь скопируем код из процедуры «ПечатнаяФорма» расширенной конфигурации и переделаем, как описано выше (см. листинг 9.2).

```
&Вместо("ПечатнаяФорма")
Процедура РТ_ПечатнаяФорма(ТабличныйДокумент)

Если ТипЗнч(ТабличныйДокумент) = тип("Структура") Тогда
    ТабличныйДокумент = ТабличныйДокумент.ТабДок;
    Макет = Документы.РасходТовара.ПолучитьМакет("РТ_МакетПечати");
иначе
    Макет = Документы.РасходТовара.ПолучитьМакет("МакетПечати");
КонецЕсли;
//-----основной код из расширенной конфигурации-----//
// Заголовок
Область = Макет.ПолучитьОбласть("Заголовок");
ТабличныйДокумент.Вывести(Область);

// Шапка
Шапка = Макет.ПолучитьОбласть("Шапка");
Шапка.Параметры.Заполнить(ЭтотОбъект);
ТабличныйДокумент.Вывести(Шапка);

// Товары
Область = Макет.ПолучитьОбласть("ТоварыШапка");
ТабличныйДокумент.Вывести(Область);
ОбластьТовары = Макет.ПолучитьОбласть("Товары");

Для каждого ТекСтрокаТовары Из Товары Цикл

    ОбластьТовары.Параметры.Заполнить(ТекСтрокаТовары);
    ТабличныйДокумент.Вывести(ОбластьТовары);

КонецЦикла;

КонецПроцедуры // ПечатнаяФорма(ТабличныйДокумент)
```

Листинг 9.2. Перехват процедуры «ПечатнаяФорма»

Нам осталось переделать код в команде расширения «РТ_ПечатьРасходнойНакладной» (см. листинг 9.3).

&НаСервере

Функция ПечатнаяФорма(ПараметрКоманды)

ТабличныйДокумент = Новый ТабличныйДокумент;

ТабличныйДокумент.ОтображатьСетку = Ложь;

ТабличныйДокумент.Защита = Ложь;

ТабличныйДокумент.ТолькоПросмотр = Ложь;

ТабличныйДокумент.ОтображатьЗаголовки = Ложь;

Сформирован = Ложь;

///создаем структур, для передачи в процедуру

СтруктураПередачи = Новый Структура("ТабДок",ТабличныйДокумент);

Для каждого Ссылка Из ПараметрКоманды Цикл

Документ = Ссылка.ПолучитьОбъект();

Если НЕ Документ.Проведен Тогда

Сообщение = Новый СообщениеПользователю();

Сообщение.Текст = "Документ не проведен: " + Строка(Документ);

Сообщение.КлючДанных = Ссылка;

Сообщение.Сообщить();

Продолжить;

КонецЕсли;

Документ.ПечатнаяФорма(СтруктураПередачи); //передали структуру

Сформирован = Истина;

КонецЦикла;

Если Сформирован Тогда

Возврат ТабличныйДокумент;

Иначе

Возврат Неопределено;

КонецЕсли;

КонецФункции

&НаКлиенте

Процедура ОбработкаКоманды(ПараметрКоманды,
ПараметрыВыполненияКоманды)

ТабличныйДокумент = ПечатнаяФорма(ПараметрКоманды);

Если ТабличныйДокумент <> Неопределено Тогда

ТабличныйДокумент.Показать();

КонецЕсли;

КонецПроцедуры

Листинг 9.3. Код в команде «РТ_ПечатьРасходнойНакладной»

После того, как сохраним расширение, то спокойно сможем распечатать макет расширения.

Задача: Перед проведением документа «Расход товара» будем проверять, внес ли контрагент аванс (должен быть больше суммы продажи), если нет, то не проведем документ.

Для решения этой задачи также будем использовать аннотацию «&Вместо», но в этот раз нам нужно продолжить выполнение основного метода (в нашем случае процедура «ОбработкаПроведения»).

Напишем в модуле расширения документа «Расход товара» следующий код:

&Вместо("ОбработкаПроведения")

Процедура РТ_ОбработкаПроведения(Отказ, Режим)

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ВзаиморасчетыОстатки.СуммаОстаток КАК СуммаОстаток

| ИЗ

| РегистрНакопления.Взаиморасчеты.Остатки(

| &МоментВремени,

| Контрагент = &Контрагент

| И Валюта = &Валюта) КАК ВзаиморасчетыОстатки";

Запрос.УстановитьПараметр("МоментВремени",?(Режим =

```
РежимПроведенияДокумента.Оперативный,ТекущаяДата(),МоментВремени());  
Запрос.УстановитьПараметр("Контрагент", Ссылка.Покупатель);  
Запрос.УстановитьПараметр("Валюта",Ссылка.Валюта);
```

```
Блокировка = Новый БлокировкаДанных;  
ЭлБлокировки = Блокировка.Добавить("РегистрНакопления.Взаиморасчеты");  
ЭлБлокировки.Режим = РежимБлокировкиДанных.Исключительный;  
ЭлБлокировки.УстановитьЗначение("Контрагент",Ссылка.Покупатель);  
ЭлБлокировки.УстановитьЗначение("Валюта",Ссылка.Валюта);  
Блокировка.Заблокировать();
```

```
Результат = Запрос.Выполнить();
```

```
Выборка = Результат.Выбрать();  
Выборка.Следующий();
```

```
Если Ссылка.Товары.Итог("Сумма") > Выборка.СуммаОстаток Тогда
```

```
СуммаНеДобора = Ссылка.Товары.Итог("Сумма") + Выборка.СуммаОстаток;
```

```
Сообщение = Новый СообщениеПользователю;
```

```
Сообщение.Текст = "Авансов у покупателя недостаточно!" + Символы.ПС + "Не  
хватает " + СуммаНеДобора ;
```

```
Сообщение.Поле = "Покупатель";
```

```
Сообщение.УстановитьДанные(ЭтотОбъект);
```

```
Сообщение.Сообщить();
```

```
Отказ = Истина;
```

```
КонецЕсли;
```

```
ПродолжитьВызов(Отказ,Режим);
```

```
КонецПроцедуры
```

Листинг 9.3. Перехват события модуля документа «ОбработкаПроведения»

В этом коде мы получаем остаток по регистру взаиморасчетов по покупателю в разрезе валюты. И если в документе сумма всех товаров больше чем остаток по взаиморасчету, то присваиваем параметру «Отказ» значение «Истина» и выводим сообщение. А в конце процедуры вызываем метод «ПродолжитьВызов» с теми же параметрами, что и у нашей процедуры. Подробно про особенности метода «ПродолжитьВызов» можно прочитать в синтаксис-помощнике, отмечу только: чтобы не путаться, лучше называть параметры в методе-перехватчике так же, как и в методе расширенной конфигурации.

Посмотрим, как сработает наш перехват события (см. рис. 9.7).

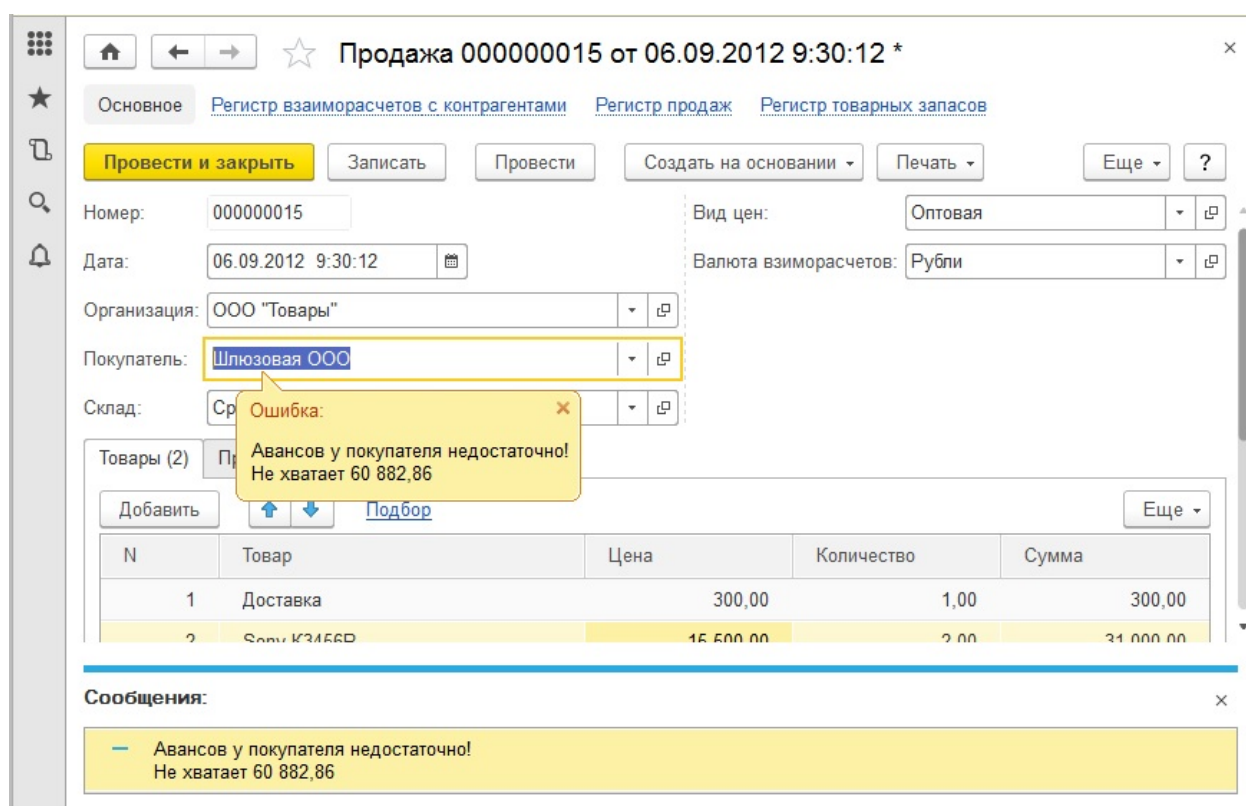


Рис. 9.7. Перехват проведения документа

Кроме того, что мы можем написать метод, который сработает взамен метода расширенной конфигурации, у нас есть возможность написать процедуру (функцию), которая выполнится перед или после какой-нибудь «типового» метода модуля объекта. Для этого нужно использовать аннотации «&Перед» и «&После».

Задача: При создании на основании контрагента документ «Расход товара» поле «Вид цены» должно оставаться пустым.

В конфигурации «Управляемое приложение» на основании контрагента нужно можно создать документ «Расход товара», в котором заполнится Контрагент и поле «Вид цены». Допустим, нам не нужно автоматическое заполнение вида цены, поэтому мы перехватим событие после выполнения процедуры «ОбработкаЗаполнения» в модуле расширения документа «Расход товара», где присвоим полю «Вид цены» пустую ссылку. Получиться вот такой код (см. листинг 9.4):

```
&После("ОбработкаЗаполнения")
```

```
Процедура РТ_ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)
```

```
Если ТипЗнч(ДанныеЗаполнения) = Тип("СправочникСсылка.Контрагенты") Тогда
```

```
    ВидЦен = Справочники.ВидыЦен.ПустаяСсылка();
```

```
КонецЕсли;
```

```
КонецПроцедуры
```

Листинг 9.4. Перехват обработки заполнения

Если мы сейчас создадим на основании контрагента документ «Продажи», то поле «Вид Цены» будет пустым.

Перейдем к формам. Платформа позволяет перехватывать в расширении события формы и элементов формы. Так же, как и для модуля объекта, существует три варианта перехвата события: перед событием, после события и вместо события. Но задаются они по-другому: посредством указания типа вызова при создании обработчика (см. рис. 9.8).

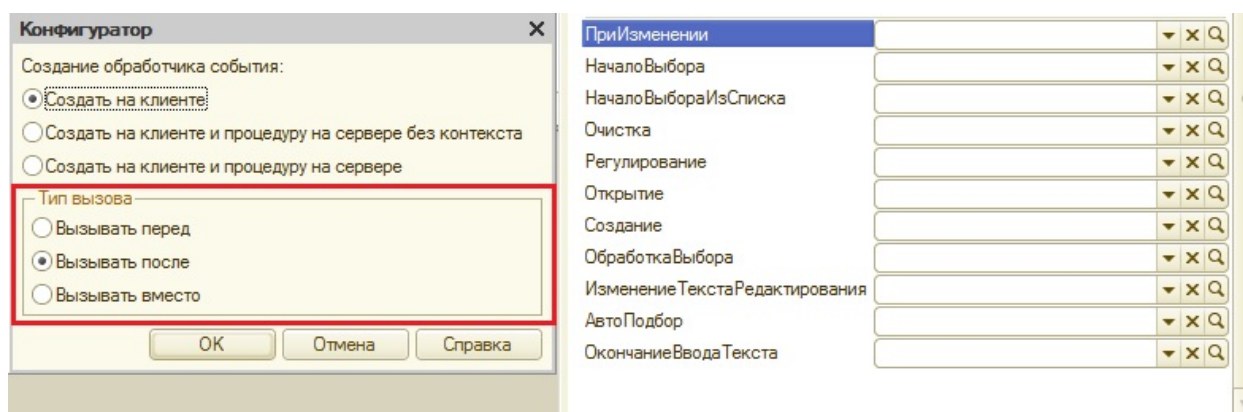


Рис. 9.8. Перехват события формы

Разберем работу перехвата событий форм на ряде примеров.

Задача: В документе «Расход товара» после выбора контрагента в окно «Обоснование отгрузки» будем записывать информацию из реквизита контрагента «Дополнительная информация».

Для этого перехватим событие после изменения поля «Покупатель». Зайдем в палитру свойств поля «Покупатель» формы документа «Расход товара» нашего расширения. И нажмем на пиктограмму «Лупа» рядом с событием «При изменении», после этого выйдет форма, где помимо знакомых вариантов «Клиент», «Сервер» и т.д., появятся варианты вызова события: «Перед», «После» и «Вместо». Выберем «На клиенте» и «После» (см. рис. 9.9).

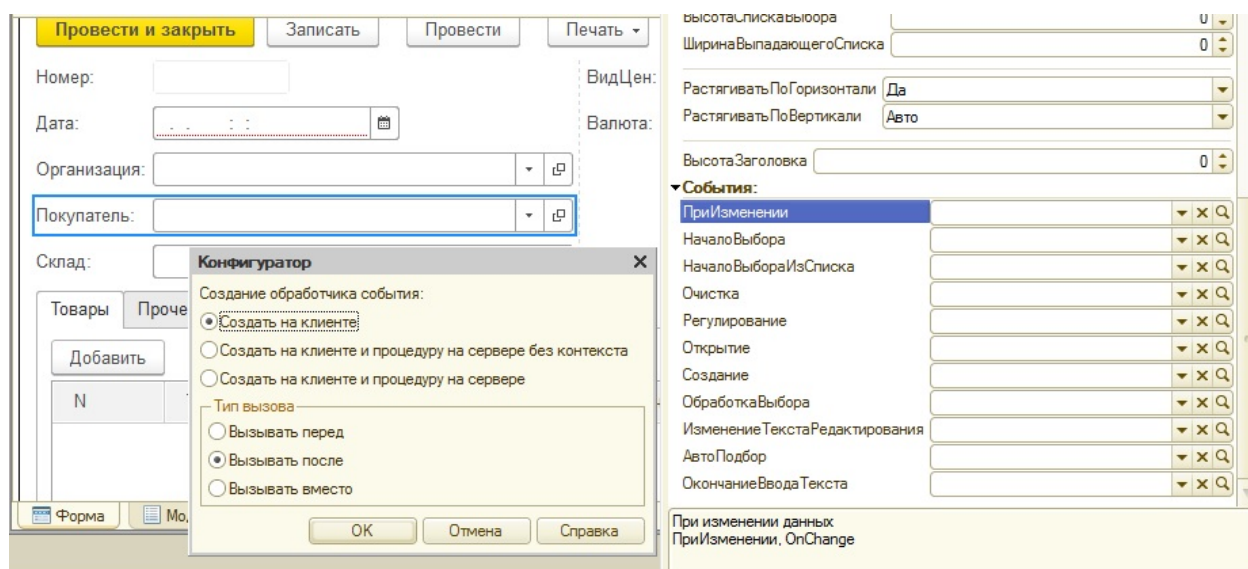


Рис. 9.9. Перехват события «ПриИзменении»

После выбора в палитре свойств рядом с созданным событием появилась соответствующая пиктограмма (см. рис. 9.10):

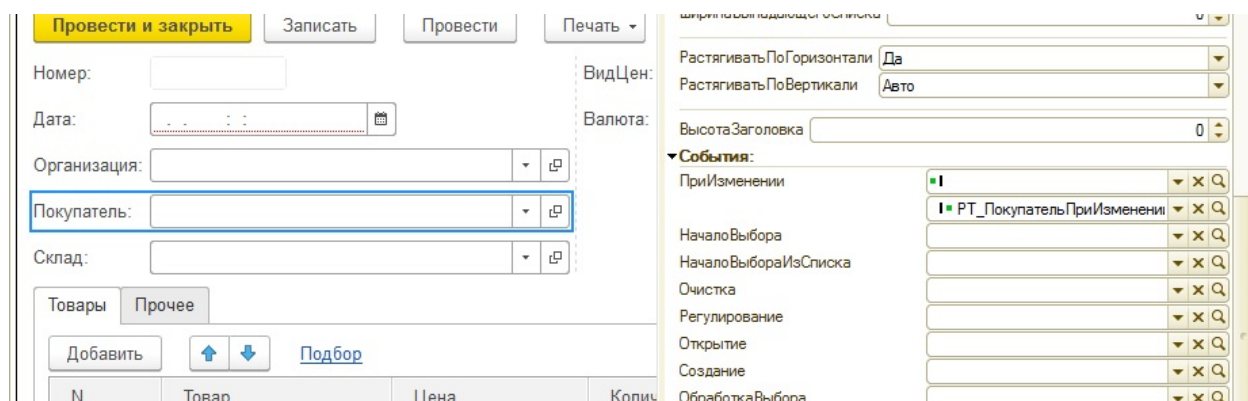


Рис. 9.10. После выбора места обработчика и типа вызова

В модуле формы обработка создана. Напишем следующий код (см. листинг 9.5):

&НаСервереБезКонтекста

Функция ПолучитьДопИнформацию(Контрагент)

Возврат Контрагент.ДополнительнаяИнформация;

КонецФункции

&НаКлиенте

Процедура РТ_ПокупательПриИзмененииПосле(Элемент)

Объект.ОбоснованиеОтгрузки = ПолучитьДопИнформацию(Объект.Покупатель);

КонецПроцедуры

Листинг 9.5. Код в перехвате события «ПриИзменении»

Теперь, если мы выберем контрагента, то заполнится информация в поле «Обоснование отгрузки»

И последняя задача будет немного поинтереснее: в документе «Расход товара» в табличной части «Товары» сделаем возможность выбора только номенклатуры с видом «Товар».

В демонстрационной конфигурации есть товары с видом «Услуга». Они показаны на рисунке 9.11.

Наименование	Код	Артикул	Поставщик	Вид
Услуги	000000036			
Доставка	000000037			Услуга
Ремонт	000000038			Услуга

Рис. 9.11. Товары с видом «Услуга»

Доработаем расширение конфигурации таким образом, чтобы в форме документа «Расход товара» при выборе товара в табличной части выходила форма выбора справочника «Товары» с отбором по виду товара.

Для этого я заимствую форму выбора справочника товары в мое расширение (см. рис. 9.12). И у этой формы создам обработчик события «ПриСозданииНаСервере», который будет выполняться вместо типового (см. рис. 9.13).

Почему *вместо* типового? Потому что типового обработчика события «ПриСозданииНаСервер» формы выбора справочника «Товары» нет, и я решил сделать вместо. Хотя можно сделать после.

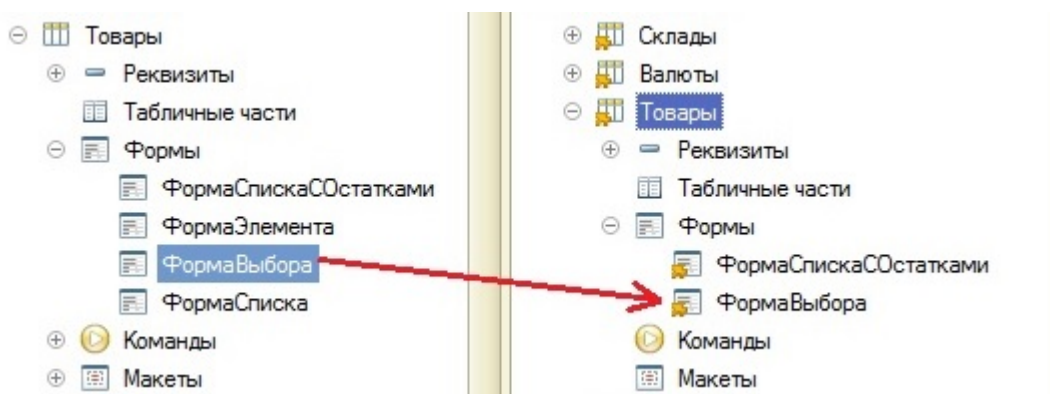


Рис. 9.12. Заимствовали форму выбора

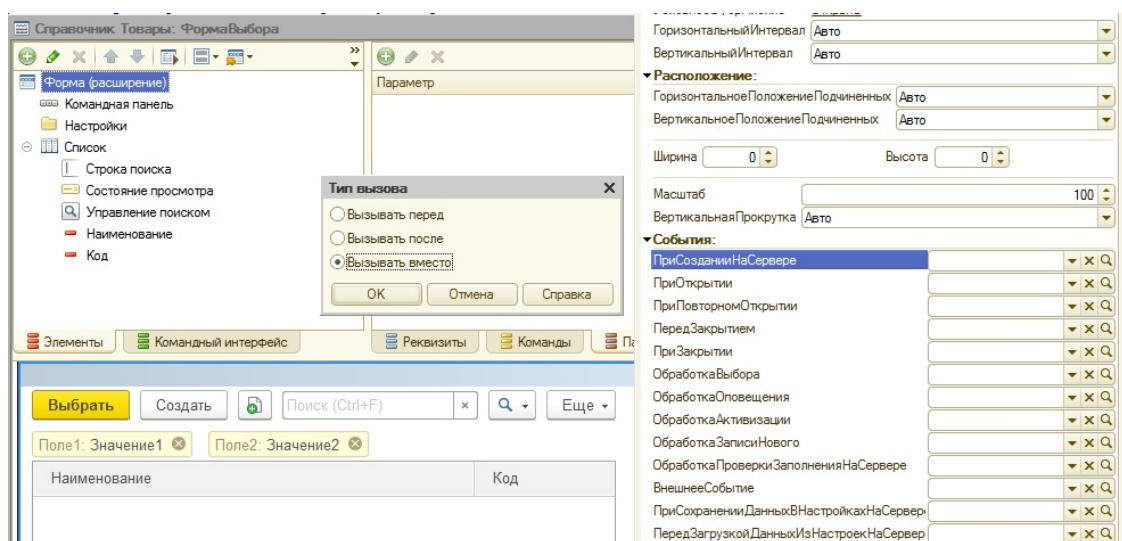


Рис. 9.13. Создадим перехват события «ПриСозданииНаСервере»

Теперь напишем код, в котором будем устанавливать отбор на динамический список «Список» формы выбора (см. листинг 9.6).

&НаСервере

Процедура РТ_ПриСозданииНаСервереВместо(Отказ, СтандартнаяОбработка)

Если Параметры.Свойство("Вид") Тогда

Отбор = Список.Отбор;

НовыйЭлОтбора =

Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));

НовыйЭлОтбора.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("Вид");

НовыйЭлОтбора.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;

НовыйЭлОтбора.ПравоеЗначение = Параметры.Вид;

КонецЕсли;

КонецПроцедуры

Листинг. 9.6. Перехват события при создании на сервере формы выбора справочника «Товары»

Все. С формой выбора мы покончили и можем начать работать с заимствованной формой документа «Расход товара». Первым делом создадим событие «НачалоВыбора» поля «Товар» таблицы «Товары», которое будем выполнять вместо типового.

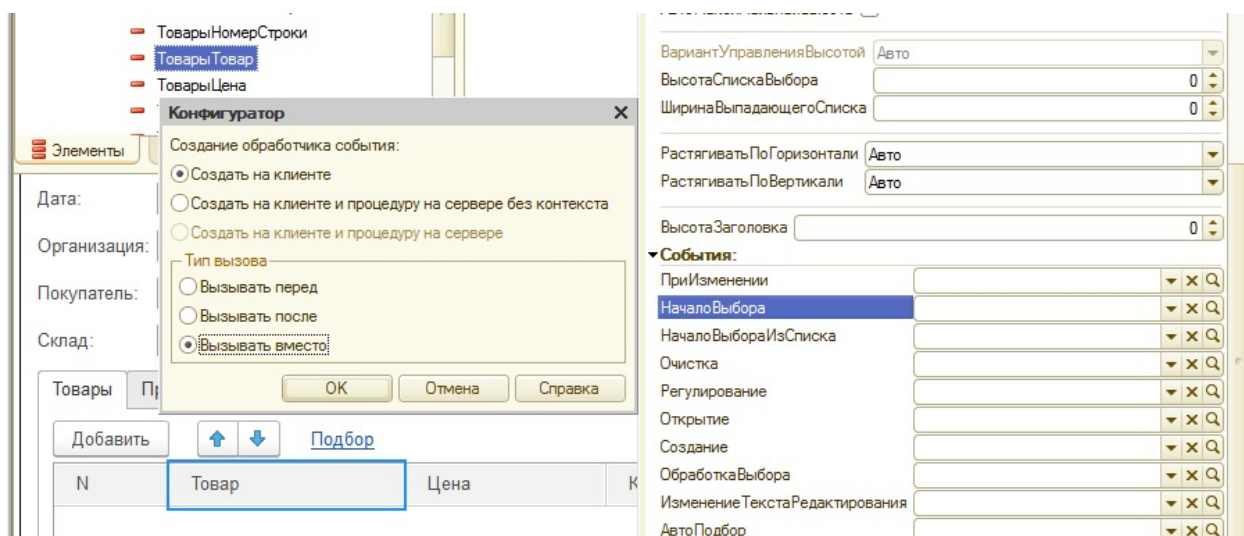


Рис 9.14. Перехват события «НачалоВыбора»

В этом обработчике мы напишем код, в результате которого будет открываться форма выбора справочника «Товары», и при открытии в эту форму будем передавать в качестве параметра вид товара – «Товар» (см. листинг 9.7).

&НаКлиенте

Процедура РТ_ТоварыТоварНачалоВыбораВместо(Элемент, ДанныеВыбора, СтандартнаяОбработка)

//отменим стандартный вызов

СтандартнаяОбработка = Ложь;

//передадим параметры на форму

ПараметрыОткрытия = Новый Структура;

ПараметрыОткрытия.Вставить("Вид",ПредопределенноеЗначение("Перечисление.ВидыТоваров.Товар"));

//опишем оповещение о закрытии

ОповещениеОЗакрытииВыбораТовара = Новый

ОписаниеОповещения("ВыполнитьПослеЗакрытияВыбораТовара",ЭтотОбъект);

ОткрытьФорму("Справочник.Товары.ФормаВыбора",ПараметрыОткрытия

,ЭтаФорма

,

УникальныйИдентификатор

""

ОповещениеОЗакрытииВыбораТовара

,);

КонецПроцедуры

Листинг 9.7. Перехват обработчика начало выбора

Не забудем написать процедуру, которая должна сработать после закрытия формы выбора (см. листинг 9.8):

&НаКлиенте

Процедура

ВыполнитьПослеЗакрытияВыбораТовара(РезультатЗакрытия,ДополнительныеПараметры) Экспорт

Если Не ЗначениеЗаполнено(РезультатЗакрытия) Тогда

Возврат;

КонецЕсли;

Идентификатор = Элементы.Товары.ТекущаяСтрока;

СтрТовара = Объект.Товары.НайтиПоИдентификатору(Идентификатор);

СтрТовара.Товар = РезультатЗакрытия;

КонецПроцедуры

Листинг 9.8. Процедура оповещения

Теперь при выборе товаров из документа «Расход товара» каталог «Услуги» будет пуст (см. рис. 9.15).

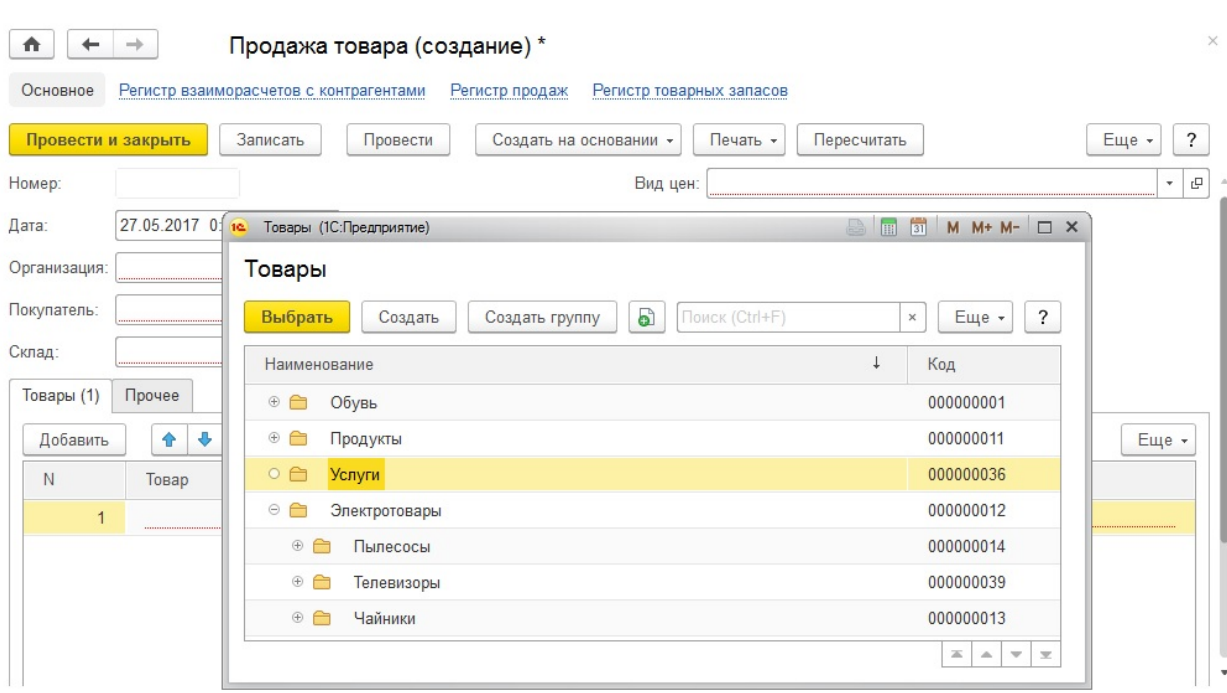


Рис. 9.15. Список услуг пуст

Но, заметьте, можно будет подобрать услугу, вводя вручную её название в поле «Товар» (см. рис. 9.16).

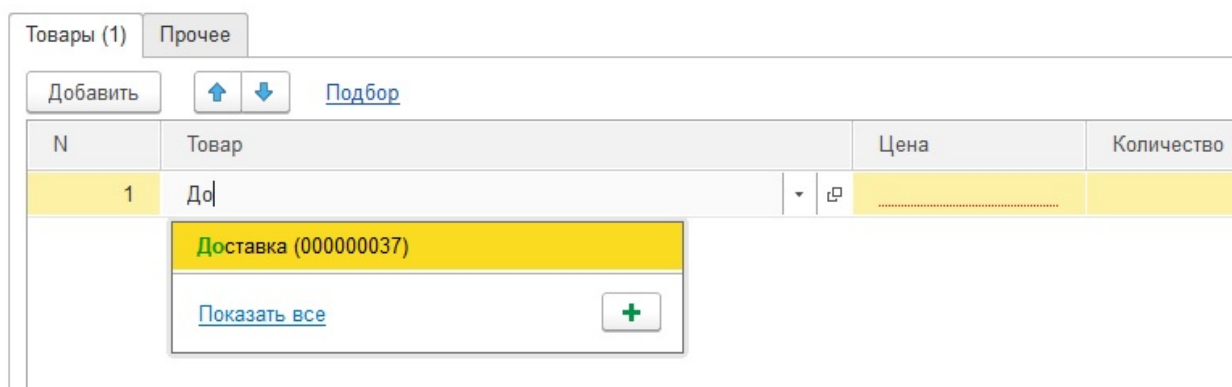


Рис. 9.16. Можно подобрать услугу с клавиатуры

Чтобы такого не было, мы перехватим событие формы «АвтоПодбор», наше событие будет выполняться вместо события расширенной конфигурации (см. рис. 9.17).

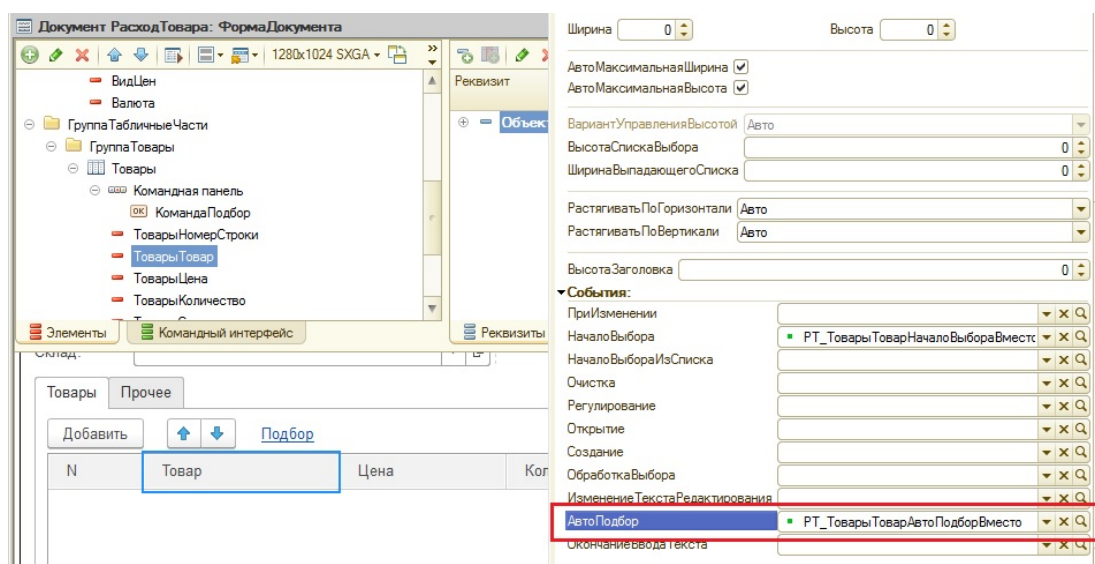


Рис. 9.17. Перехват события «АвтоПодбор»

В этом обработчике будем накладывать отбор в параметрах получения данных (см. листинг 9.9).

&НаКлиенте

Процедура `PT_ТоварыТоварАвтоПодборВместо(Элемент, Текст, ДанныеВыбора, ПараметрыПолученияДанных, Ожидание, СтандартнаяОбработка)`

Отбор = `ПараметрыПолученияДанных.Отбор;`

```
Отбор.Вставить("Вид",ПредопределенноеЗначение("Перечисление.ВидыТоваров.Товар"))
```

КонецПроцедуры

Листинг 9.9. Код в перехвате события «АвтоПодбор»

Теперь при подборе с клавиатуры услуги не будут выходить (см. рис. 9.18).

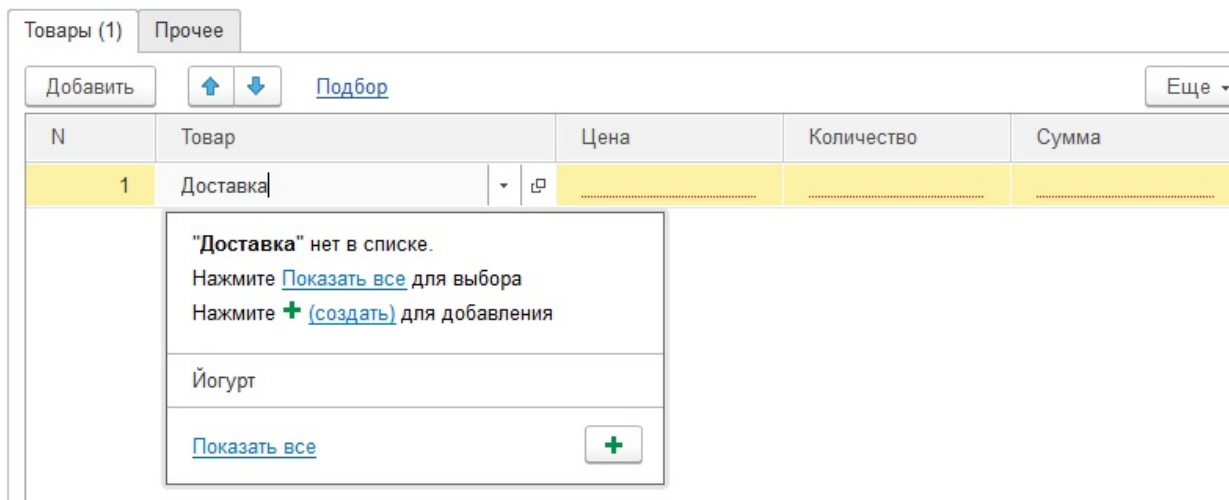


Рис. 9.18. Нельзя подобрать услугу с клавиатуры

Это последняя практическая задача, которая описана в этой главе.

На этой главе мы закончим изучать работу с расширениями конфигурации. Этим знаний Вам вполне хватит, чтобы начать работать с расширениями, изменять конфигурации без снятия с поддержки. Фирма 1С постоянно изменяет и обновляет этот инструмент, и я надеюсь, нас в будущем ждут интересные новшества.