Проект Ананас

Дизайнер

Руководство пользователя

А.П. Паскаль

2005-2006

Аннотация

Краткое руководство предназначено для быстрого ознакомления с Дизайнером – программой настройки бизнес схем платформы автоматизации учета Ананас. В руководстве по шагам рассматривается создание простой бизнес схемы “Оперативный учет”. Руководство предназначено для прикладных программистов. Пользовательскую документацию по бизнес схеме “Оперативный учет”, входящей в дистрибутив Ананаса, смотрите в Руководстве пользователя бизнес схемы «Оперативный учет».

Оглавление

Введение....4
Получение и установка Ананаса....4
Установка на Linux....4
Установка на Windows....5
Первый шаг или с чего начать....5
Создание пользовательского интерфейса....16
Редактор главного меню....17
Действия....17
Редактор диалогов....18
Разработка первичных документов....18
Проектируем Приходную накладную....18
Реквизиты Приходной накладной....18
Реквизиты табличной части Приходной накладной....29
Экранная форма Приходной накладной....30
Модуль экранной формы....31
on_formstart()....31
on_formstop()....32
on_conduct()....32
on_button(name)....32
on_tabupdate(row,col,tableName)....32
on_tablerow(tableName) 32
Печатная форма Приходной накладной....40
Проектируем Расходную накладную....42
Реквизиты Расходной накладной 42
Реквизиты табличной части Расходной накладной....42
Экранная форма Расходной накладной....42
Печатная форма Расходной накладной....42
Разработка справочников....42
Проектируем Справочник Контрагентов....42
Определение реквизитов....42
Определение главного реквизита....46
Экранные формы справочника....46
Редактор групп справочника....47
Редактор элементов справочника....48
Проектируем Справочник товаров....49
Разработка журналов....49
Разработка отчетов....49
Исполнение бизнес схемы в Ананасе 50
Примеры кода Ананас-Скрипта....50
Динамические пользовательские диалоги. Программирование и обработка событий формы диалога 52
connect – задание собственных обработчиков событий 53
Список иллюстраций....55

Введение

В настоящее время повсеместное применение получило программное обеспечение автоматизации различных видов учета. Наиболее популярными видами учета в плане применения средств автоматизации на сегодня являются оперативный, управленческий, бухгалтерский, кадровый, торговый и производственный учет. Существуют различные подходы к решению задач автоматизации. Например, ПО автоматизации учета можно разделить на два группы, различающихся в подходе разработчиков к архитектуре создаваемого ими ПО. Первая группа – это монолитное ПО, вторая группа – это ПО разбитое на уровни реализации, когда есть базовый, мало изменяемый уровень – платформа, и построенный на его основе легко модифицируемый уровень – приложения. В каждом подходе можно найти как положительные моменты, так и недостатки. Однако, в настоящее время видна тенденция адаптации обоих подходов в различных рыночных сегментах. В то время как монолитное ПО получает наибольшее распространение среди небольших компаний и индивидуальных предпринимателей, многоуровневое ПО становится наиболее востребованным в крупных компаниях. Такое разделение по нишам объясняется в первую очередь низкой стоимостью сопровождения монолитного ПО и, в свою очередь, возможностью более гибкой адаптации многоуровневого ПО к специфике крупных компаний. Платформа Ананас и приложения на ее основе представляет собой ПО второй группы. Это как минимум значит, что при его изучении необходимо уделять внимание не только функциям, доступным конечному пользователю, но и вопросам применения платформы для наращивания, модификации и адаптации таких функций для отражения специфики автоматизируемого предприятия. «Дизайнер» - это составная часть платформы, обеспечивающая возможность управления доступными пользователю функциями, разработки, модификации и настройки используемых в повседневной деятельности пользователем приложений.

Получение и установка Ананаса

Последнюю версию дистрибутива системы всегда можно скачать с сайта проекта, расположенного по адресу http://ananas.lrn.ru Если вы пользуетесь дистрибутивом Linux, выпускаемым компанией ALT Linux, то вам крупно повезло, так как этот производитель любезно включает дистрибутив Ананаса в свой флагманский продукт ALT Linux Master и вам не потребуется тратить время на загрузку Ананаса из Сети, просто посмотрите rpm пакеты ananas-... на CD или DVD дисках вашего экземпляра дистрибутива ALT Linux.

Описываемый в этом разделе порядок установки, подразумевает использование Ананаса ровно на одном компьютере. Если вы собираетесь пользоваться одной и той же базой данных Ананаса с разных компьютеров, то есть эксплуатировать Ананас в многопользовательском режиме в локальной сети, вам потребуется провести установку Ананаса на каждый из компьютеров, а также на сервер и далее изменить настройки доступа к базе данных на компьютерах, выполняющих роль клиентов. Необходимо будет указать в поле «Адрес сервера базы данных», как показывает на странице , ip адрес сервера или его имя.

Установка на Linux

Для установки требуется следующий набор пакетов

ananas-engine-qt-0.9.1-1.i386.rpm
ananas-inventory-0.9.1-1.i386.rpm

Первый из вышеперечисленных пакетов является основным. Он позволяет исполнять приложения, разработанные на платформе Ананас. Второй пакет содержит одно из приложений – типовая бизнес схема “Оперативный учет”.

Команда установки

```
[app@app ~]$ rpm -ivh ananas-engine-qt-0.9.1-1.i386.rpm
```

может выдать сообщения о неудовлетворенных зависимостях. Если это произошло, установите требуемые пакеты до установки Ананаса.

Здесь мы не установили пакет ananas-inventory-0.9.1-1.i386.rpm, так как целью настоящего руководства является демонстрация разработки бизнес схемы Ананаса с самого начала. Для ознакомления с бизнес схемой “Оперативный учет” обратитесь к Руководству пользователя.

Установка на Windows

Для установки потребуется следующий файл

Ananas-Setup-0.9.3.exe

После его запуска на исполнение, достаточно следовать пошаговым инструкциям отображаемым на экране. Есть один нюанс, который, скорее всего, не касается обычного пользователя. Если до установки Ананаса на компьютере уже был установлен MySQL сервер, его следует остановить (выключить), дабы избежать конфликта по вопросу использования порта с сервером MySQL, устанавливаемым Ананасом.

Первый шаг или с чего начать.

Если вы решили разработать свою бизнес схему, отражающую специфику бизнес процессов принятых в компании заказчике, то вопрос, с чего начать, легко разрешим в следующем порядке. Самый быстрый и, по-видимому, общепринятый способ – доработать позаимствованную бизнес схему. Если такой подход неприемлем по той или иной причине, разработка бизнес схемы с нуля не является таким уж сверх трудоемким процессом, который следует избегать во что бы то ни стало. В любом случае для успешного достижения цели будет правильным пойти по пути определения требований к результату вашей работы – разрабатываемой бизнес схеме.

Здесь и дальше мы будем исходить из предположения о принятии решения о создании бизнес схемы с нуля.

Запуск Ананаса можно выполнить через меню или из командной строки. Второй способ может выручить вас, если Ананас не подаст признаков жизни или будет работать с ошибками, сообщения о которых появятся на консоли.[1] Если вы решите обратиться за помощью к кому-нибудь, включая разработчиков проекта Ананас, отсылайте вместе с вопросами эти сообщения об ошибках и информацию о версии ОС, на которую вы устанавливаете Ананас.

После успешного запуска командой

```
[app@app ~]$ ananas-designer
```

Ананас.Дизайнер предложит выбрать бизнес схему для работы из списка. Так как мы не установили пакет ananas-inventory, это список будет пуст. Нас это вполне устраивает.

1 В последних версиях Ананаса сообщения об ошибках заносятся в специальный файл – Файл логов. О том как его найти и настроить уровень детализации логов, вы можете узнать на сайте/форуме проекта и/или в пользовательской документации.

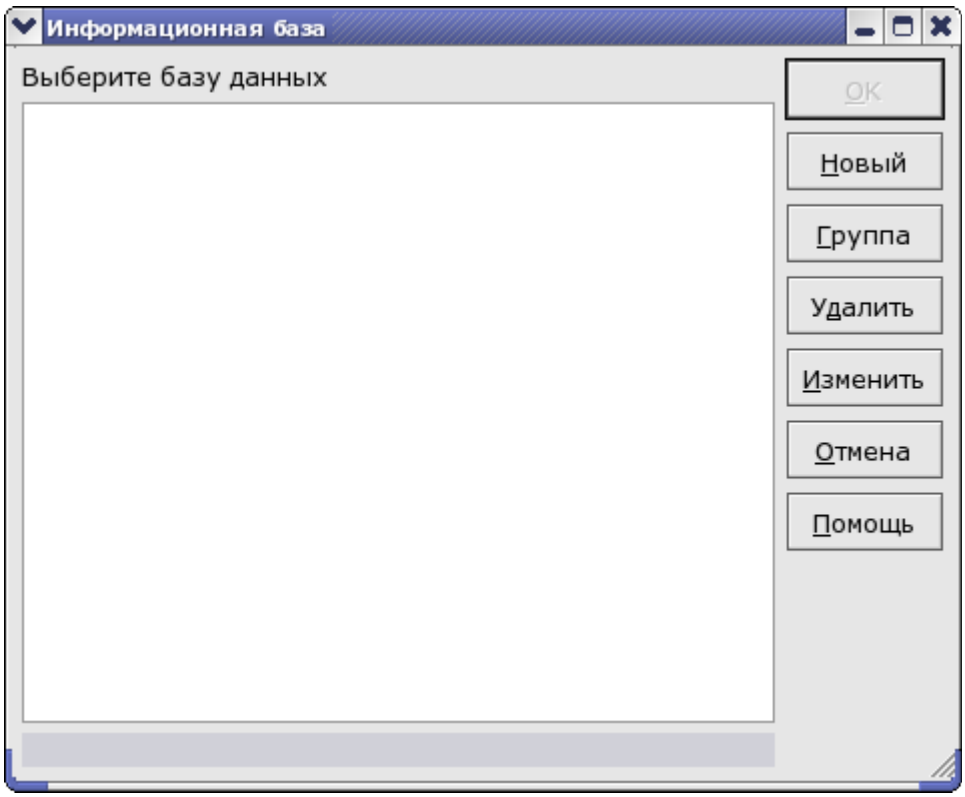


Иллюстрация 1. Диалог выбора базы данных. Список выбора пуст.

Первое, что необходимо сделать – создать группу. Название придумайте сами. Я вписал “В разработке” в поле названия Группы, имея в виду, что в группе будут находиться бизнес схемы для экспериментов. Создание группы осуществляется щелчком по кнопке “Группа”. Второе – создание бизнес схемы. Выберите группу и щелкните по кнопке “Новый”. Имеется ввиду новый элемент списка.

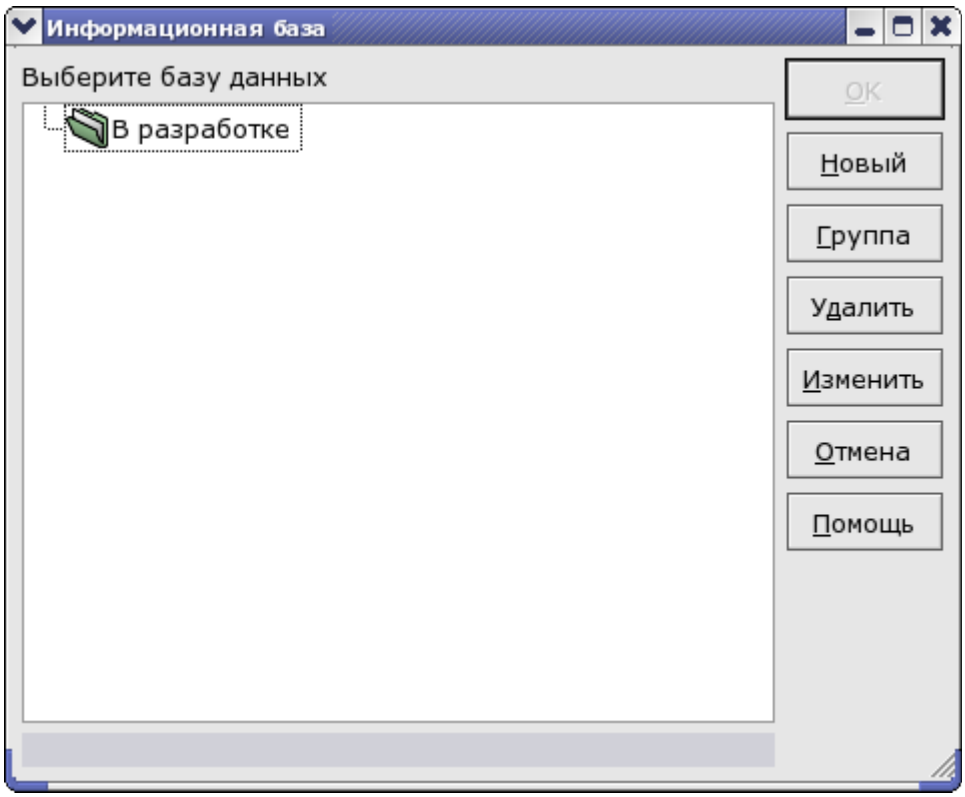


Иллюстрация 2. Диалог выбора базы данных. Создана группа "В разработке"

На экране появиться новое окно со множеством полей, предназначенных для ввода значений, смысл которых я сейчас поясню.

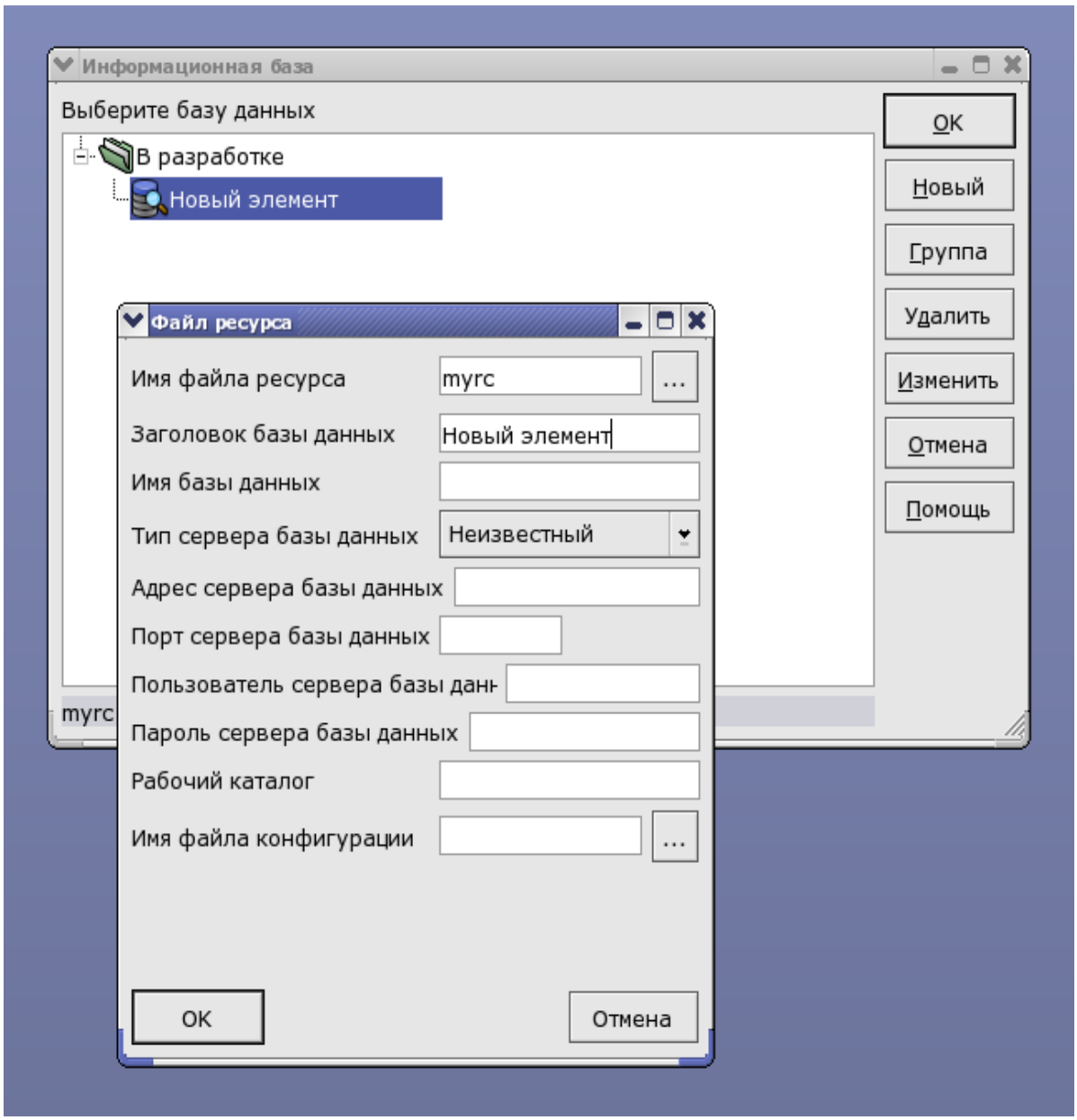


Иллюстрация 3. Диалог задания параметров новой базы данных.

Имя файла ресурса содержит ссылку на файл, в котором хранятся значения всех полей этого диалога, расположенных ниже. Имеется возможность выбора ранее созданного файла ресурса из файловой системы. Для этого достаточно вызвать диалог выбора файла щелчком по кнопке, расположенной справа от поля с именем файла ресурса. При открытии таким образом файла ресурса значения всех полей диалога будут заменены значениями из файла ресурса.

Заголовок базы данных предназначен для пользователя и отображается в списке доступных для выбора баз данных в диалоге “Информационная база”, появляющемся при старте Ананас.Дизайнера.

Имя базы данных содержит имя база данных, используемое системой при доступе к серверу базы данных.

Тип сервера базы данных позволяет выбрать один из поддерживаемых Ананасом серверов баз данных. На момент написания настоящего руководства Ананас поддерживает MySQL и PostgreSQL. Разумеется, следует убедиться, что соответствующий выбираемому вами типу сервер установлен и запущен, так как здесь вы задаете лишь драйвер для использования, который не будет работать, если не установлен соответствующий сервер.

Адрес сервера базы данных задает адрес компьютера, на котором установлен сервер баз данных выбранного выше типа. Разумеется, сервер баз данных может работать как на том же самом компьютере, где вы установили

Ананас, так и на любом другом, в том числе и удаленном компьютере. Если сервер баз данных работает локально, на одном компьютере с Ананасом, полю следует задать значение localhost. В остальных случаях следует узнать адрес сервера у системного администратора. Адрес может быть задан как символьным именем (например myhost.com), так и числовым значением (например 127.0.0.1) IP адреса.

Порт сервера базы данных по умолчанию не содержит ни какого значения. Его следует задавать лишь в случаях, когда используется нестандартный порт, что случается достаточно редко. По этому вопросу следует консультироваться с системным администратором, выполнявшим установку сервера баз данных. Если вы выполняли установку сервера баз данных самостоятельно и не меняли порт сервера баз данных, то поле **Порт сервера базы данных** следует оставить пустым.

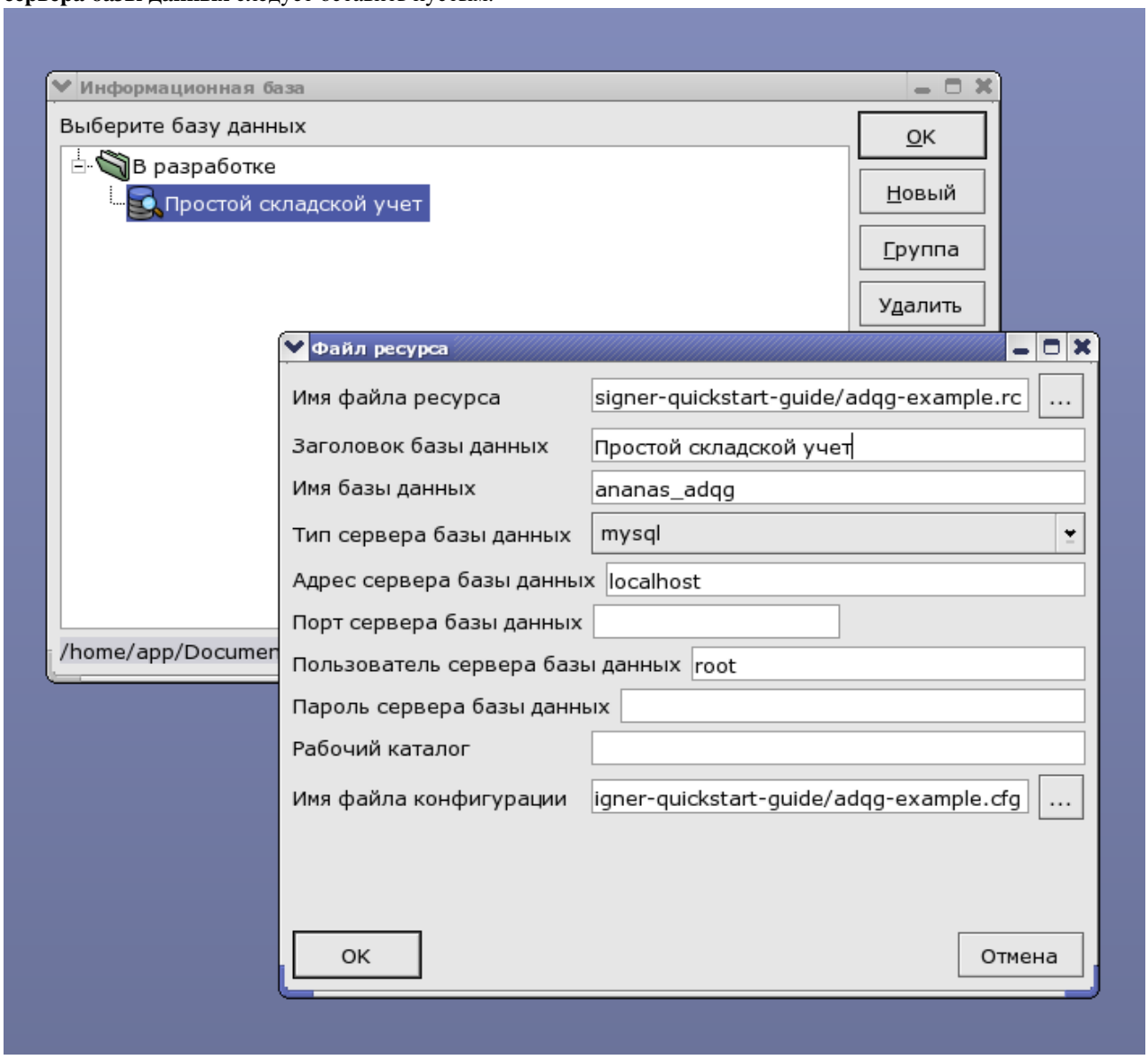


Иллюстрация 4: Диалог задания параметров базы данных. Значения заданы.

Пользователя и **пароль сервера базы данных** следует узнать у системного администратора. Если вы сами устанавливали MySQL сервер и не меняли пароль после установки, то пользователь root с пустым паролем обеспечат вам требуемый результат.

Значение поля **Рабочий каталог** можно оставить пустым. Понятия не имею для чего оно используется в настоящий момент.

Поле **Имя файла бизнес схемы** содержит ссылку на самый главный файл бизнес схемы – файл, где хранятся все результаты работы в Дизайнере, такие как экранные формы, алгоритмы бизнес логики на Ананас-Скрипте, описания структур данных бизнес объектов системы: каталогов, регистров, документов, используемые для создания и модификации схемы базы данных на SQL сервере. Более подробную информацию о содержании и структуре файла бизнес схемы следует искать в Руководстве хакера.

Задав таким образом значения полей, следует щелкнуть по Ok. Иллюстрация 5 показывает вид списка баз данных после того как вы выполните это действие. Теперь снова следует щелкнуть по кнопке Ok. После получения подтверждения о сохранении настроек локально, они будут сохранены в папке ~/.ananasrc и откроется главное окно Дизайнера. Не следует пугаться того, что оно пустое. Для открытия выбранной на предыдущем шаге бизнес схемы щелкните по кнопке Открыть схему (Иллюстрация 6 на странице 11)

Открытая в Дизайнере пустая бизнес схема дает представление о доступных для создания и конфигурирования типах прикладных объектов системы (Иллюстрация 7 на странице 12). Перечислим их.

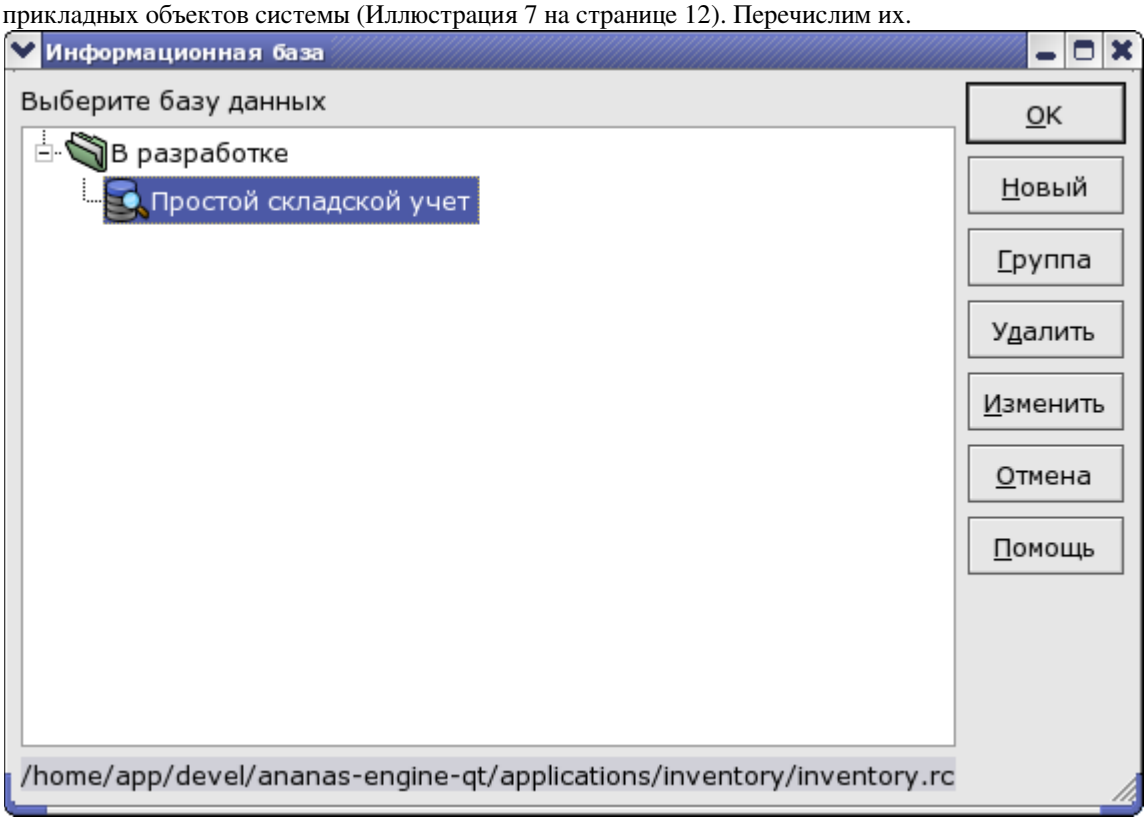


Иллюстрация 5. Диалог выбора базы данных. В списке одна группа с одной базой данных в ней.

Справочники предоставляют возможность ведения баз данных (картотек) произвольной структуры с возможностью классификации/разнесения элементов по иерархически организованным группам. Типичным примером использования этой возможности является ведение Каталога товаров, Справочника контрагентов, Реестра сотрудников. В бизнес схеме может быть создано более одного типа справочника. Идентификация справочников осуществляется по имени типа, заданном при их создании. Справочники характеризуются тем, что в системе не может быть более одного справочника определенного типа. То есть определяя тип справочника, мы по сути так же определяем экземпляр для хранения реальных данных. В процессе эксплуатации бизнес схемы определенные в ней справочники доступны сразу, то есть не требуют (и более того не допускают) операций по их

созданию. Создаются и редактируются лишь элементы того или иного справочника.

Документы определяют один из важных аспектов методологии работы Ананаса с данными - управление первичными учетными документами. Для отражения хозяйственных операций Ананас предусматривает их регистрацию с помощью электронных документов. Электронные документы могут иметь произвольную структуру, состоящую из набора полей и одной или нескольких табличных частей. Информация о структуре документов хранится в бизнес схеме. В отличие от справочников в бизнес схеме определяются не экземпляры документов, а их типы. Экземпляры документа любого из определенных типов создаются только уже при эксплуатации бизнес схемы.

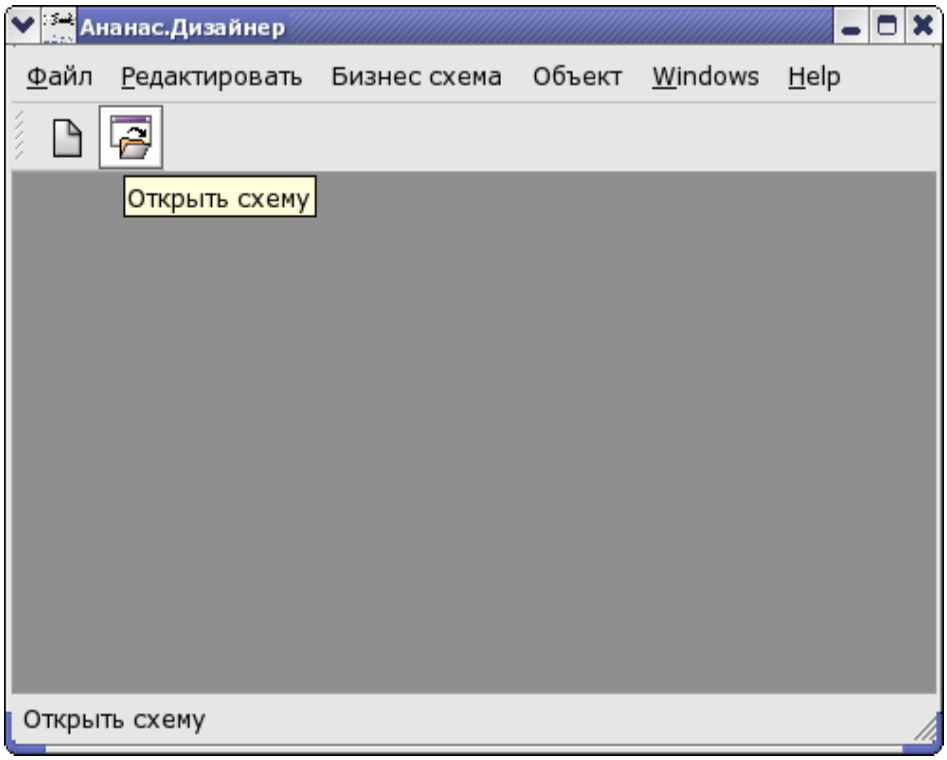


Иллюстрация 6. Вид Дизайнера сразу после запуска. Для доступа к бизнес схеме ее следует открыть.

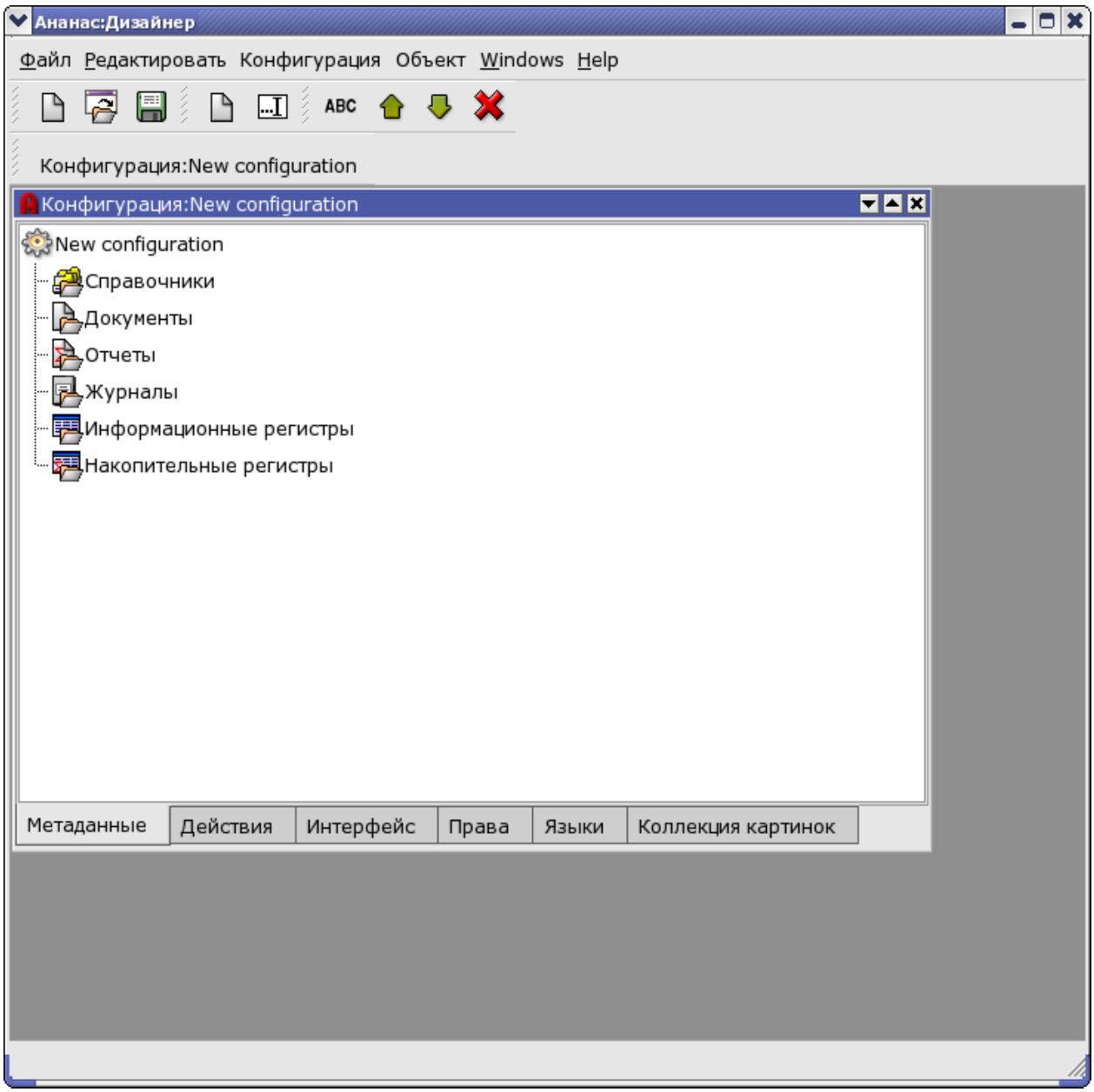


Иллюстрация 7. Новая, пустая бизнес схема открыта в Ананас.Дизайнере

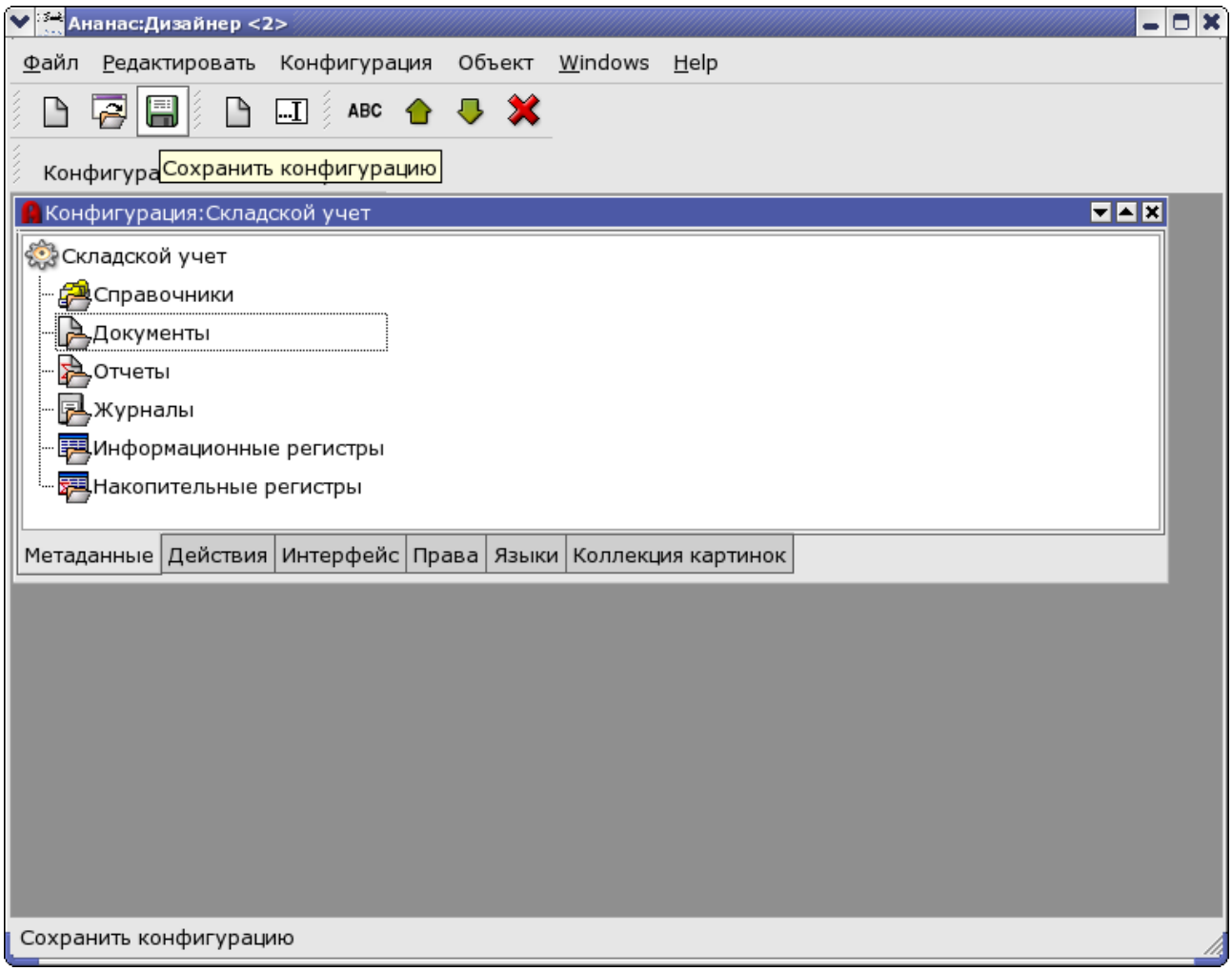


Иллюстрация 8: Сохранение новой пустой бизнес схемы

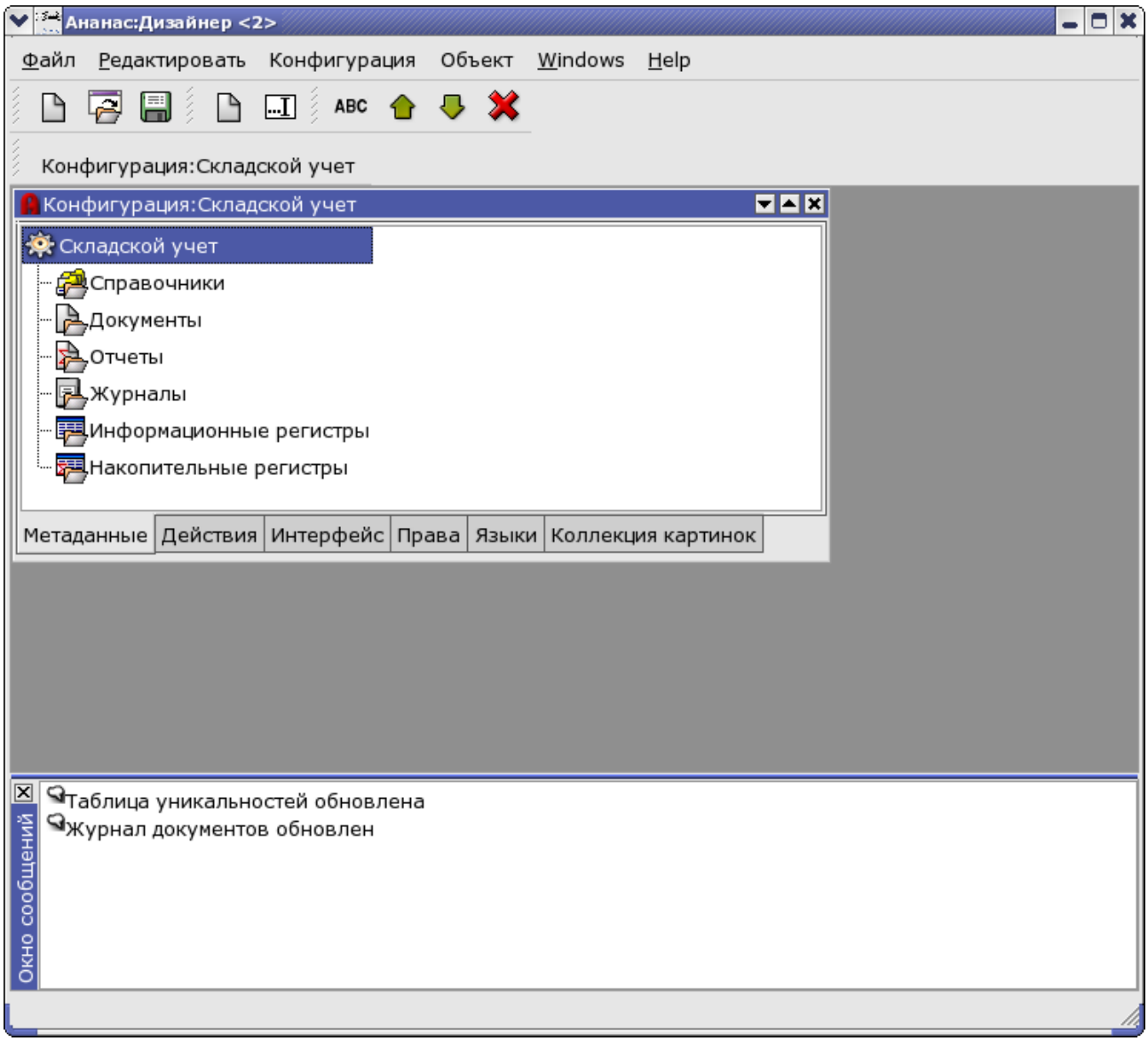


Иллюстрация 9: Сообщения об успешном сохранении пустой бизнес схемы.

Отчеты необходимы для подготовки печатных документов. Ананас позволяет формировать отчеты как в формате HTML так и в формате OpenOffice *.sxw, *.odt

Журналы обеспечивают доступ пользователя к экземплярам документов, созданных в процессе эксплуатации бизнес схемы. Журналы могут быть специализированными, обеспечивающими доступ только к документам одного типа или общими. Журналы идентифицируются по имени типа.

Информационные регистры так же как и накопительные регистры предназначены главным образом для накопления хронологической информации используемой отчетами и другими прикладными объектами системы. Занесением информации в регистры занимаются документы. Обычно пользователь не работает с регистрами напрямую.

Накопительный регистр, в отличие от информационного осуществляет расчет и хранение остатков по разрезам учета. Регистры так же как и справочники идентифицируются по имени типа и в системе существует ровно по одному регистру каждого типа. Типов регистров, разумеется, может быть больше одного.

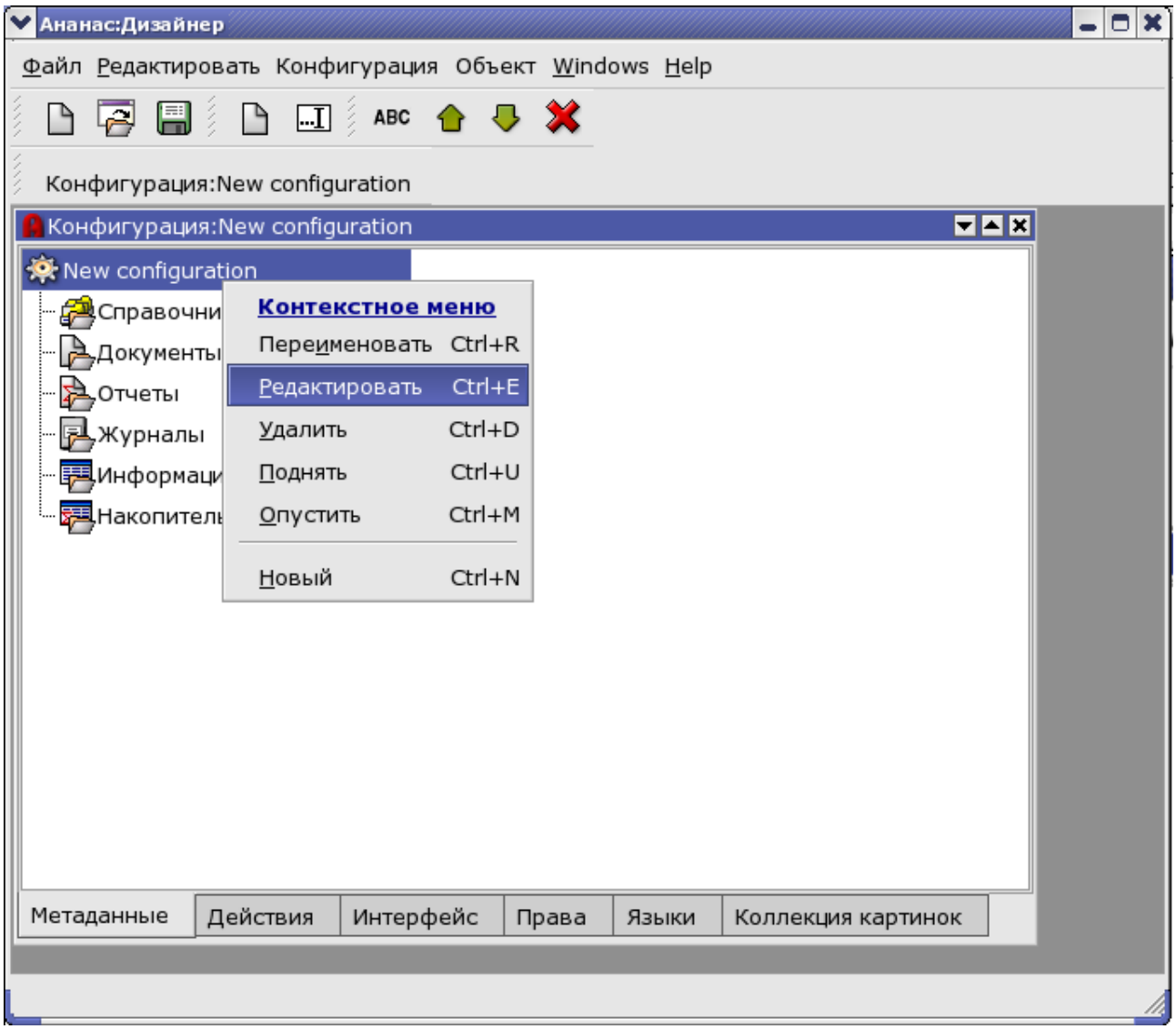


Иллюстрация 10: Вызов диалога редактирования параметров бизнес схемы

Вновь созданная пустая бизнес схема не содержит описания каких-либо бизнес объектов и бизнес логики, лишь собственные реквизиты, описывающие ее. Работу с реквизитами (параметрами) бизнес схемы отображает Иллюстрация 10: Вызов диалога редактирования параметров бизнес схемы на странице 15 и Иллюстрация 11: Диалог редактирования параметров бизнес схемы на странице 16. Тем не менее, даже для сохранения пустой бизнес схемы в файле задаваемом в поле «Имя файла бизнес схемы» (см. страницу 10) необходим доступ к SQL серверу, так как одновременно с записью бизнес схемы в файл, происходит создание (а в случае повторного сохранения - обновление) системных структур данных бизнес схемы. Об успешном сохранении бизнес схемы в файле и создании соответствующей схемы базы данных свидетельствуют сообщения, выведенные в окне сообщений, появляющемся в нижней части окна Дизайнера. В качестве примера приведена Иллюстрация 9: Сообщения об успешном сохранении пустой бизнес схемы. на странице 14.

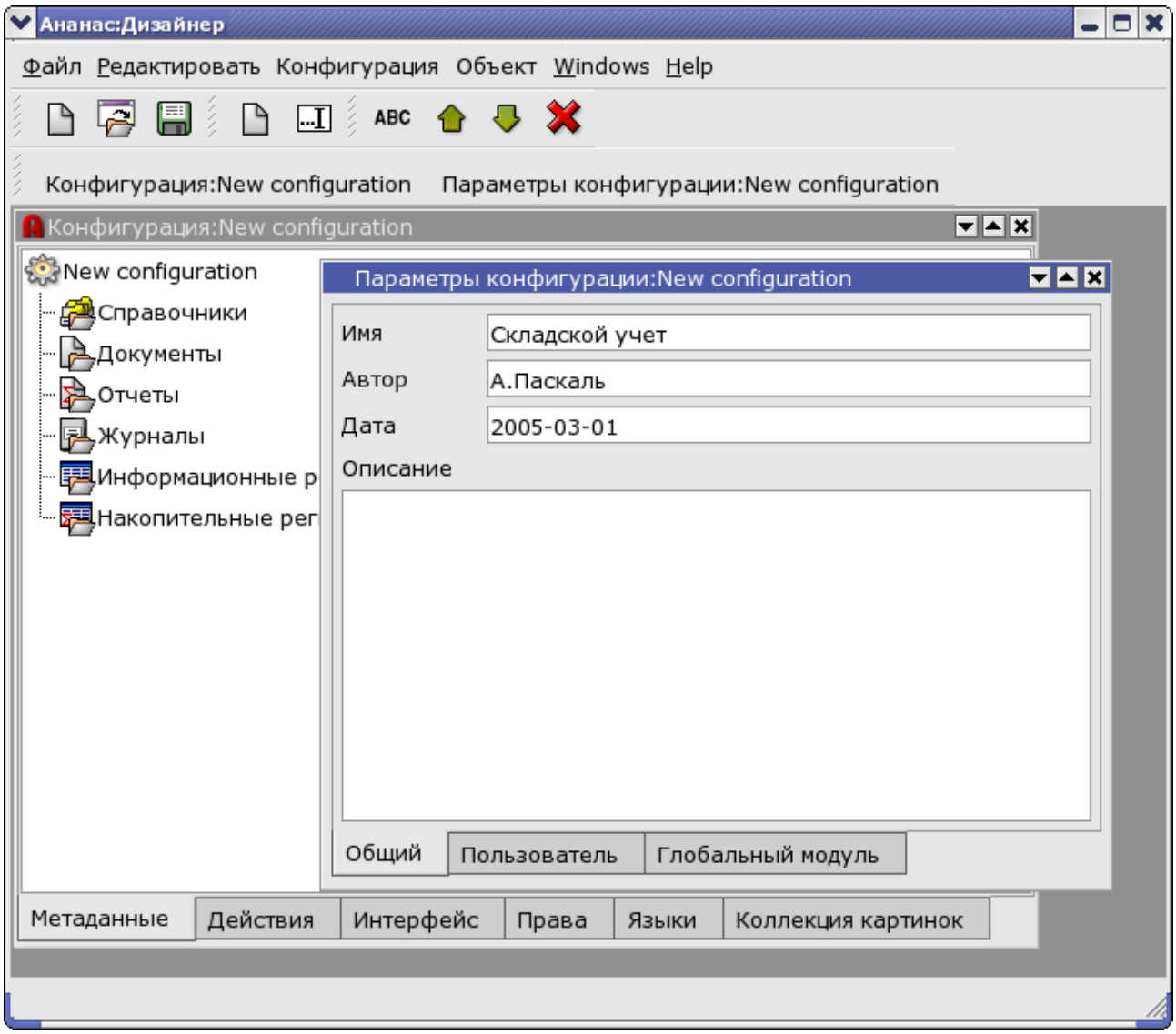


Иллюстрация 11: Диалог редактирования параметров бизнес схемы

Создание пользовательского интерфейса

Пользовательский интерфейс, иногда называемый Графическим Пользовательским Интерфейсом (английская аббревиатура GUI), это та часть программы, которая отвечает за непосредственный контакт с пользователем. Проще говоря, для пользователя пользовательский интерфейс – это и есть программа, точно так же как монитор компьютера является компьютером в глазах пользователя. Разумеется, проектированию именно этой части приложения стоит уделить особое внимание в стремлении создания комфортной среды для пользователя. Платформа Ананас в этом вопросе, как и во многих других, придерживается общепринятых взглядов и устоявшихся традиций и предлагает предоставить пользователю MDI интерфейс, хорошо всем знакомый и применяемый во множестве уже существующих приложений. Интерфейс этот предполагает работу пользователя с одним главным окном, с размещенным в нем главным меню, и одним или более окном, вложенных в главное окно, которые могут перекрывать частично или полностью друг друга, с возможностью быстрого переключения между ними. Главное меню приложения может быть совершенно различным у двух разных бизнес схем. Для

создания главного меню бизнес схемы в Дизайнере предусмотрен удобный в использовании Редактор главного меню.

Редактор главного меню

Меню, с которым работает конечный пользователь, целиком и полностью (за исключением пункта Справка) формируется при создании бизнес схемы. Дизайнер имеет специальные средства для создания меню – Редактор меню. Меню может состоять из пунктов, подменю и разделителей. Общий вид Редактора меню показывает Иллюстрация 12 на странице 17. Редактор меню, позволяет также формировать панели инструментов, добавляя туда необходимый набор кнопок.

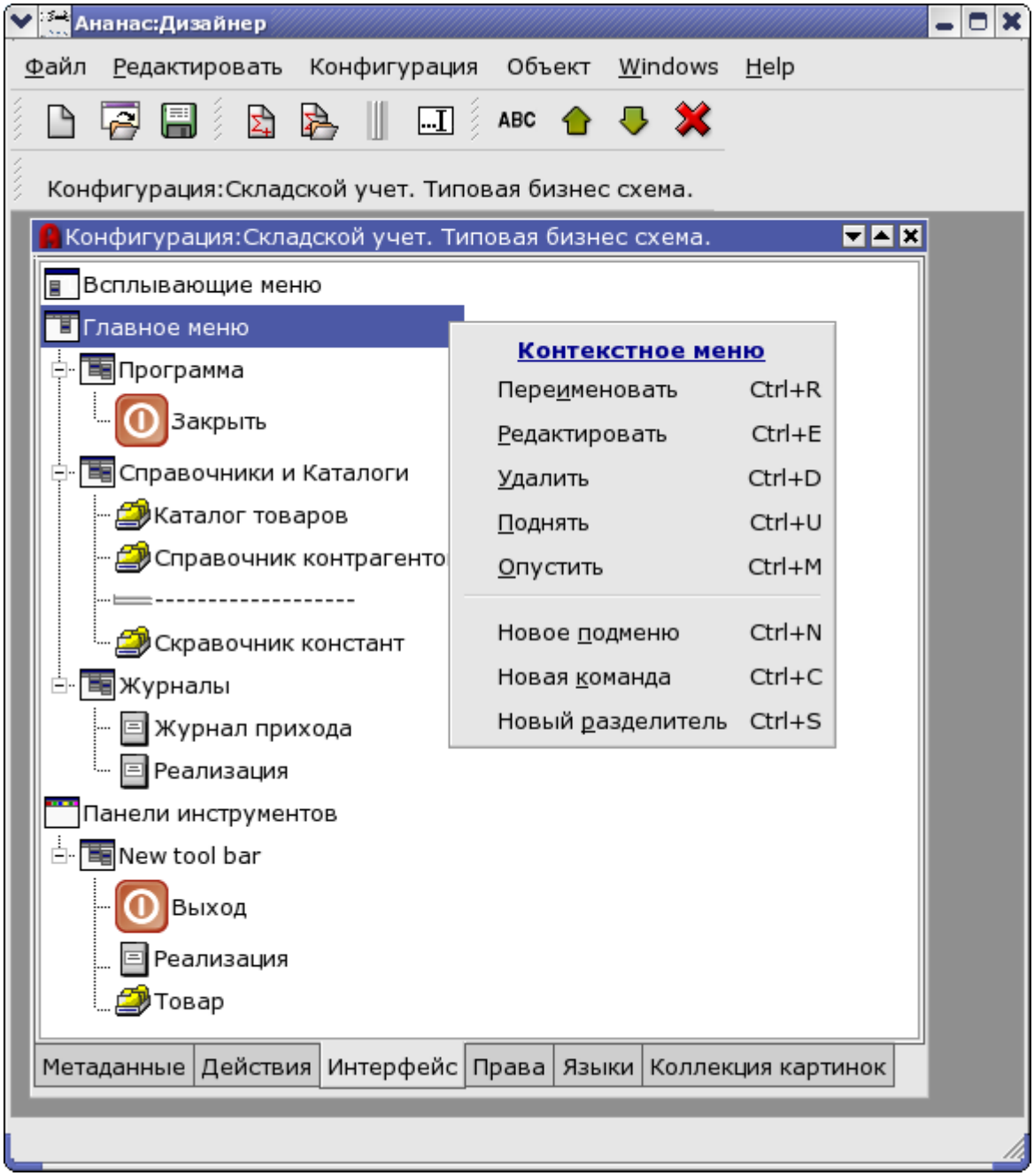


Иллюстрация 12: Редактор главного меню доступен через закладку Интерфейс

Действия

Бизнес схема позволяет определить именованные Действия над бизнес объектами. Действие определяется тремя

параметрами: бизнес объектом, операцией и экранной формой. Например, мы можем определить действие «Выписать счет» с параметрами «Документ.Счет», «Создать новый», «Предопределенная». Вызов такого действия приведет к открытию предопределенной экранной формы редактирования нового счета. Важно понимать, что конечному пользователю Действия не доступны напрямую. Они доступны только через пункты меню и кнопки панелей инструментов, к которым мы должны привязать действия, используя Редактор меню.

Для бизнес объектов доступны такие операции как «Создать новый», «Изменить», «Просмотреть», «Выполнить».

Редактор диалогов

Кроме главного меню, пользователь постоянно работает с экранными формами, позволяющими просматривать и редактировать данные, хранимые в справочниках, документах, журналах. То есть работая с каждым из бизнес объектов, будь то товар, контрагент или документ первичного учета, пользователь работает с его экранной формой. Но так как форма содержит не только поля для ввода значений, но и кнопки и другие динамические элементы, полное название такой экранной формы – экранная форма диалога пользовательского интерфейса. Так как форма с кнопками – это уже не просто бумажный бланк, переведенный в электронный вид, а средство диалога пользователя с программой.

Для создания экранных форм диалогов в Ананасе предусмотрен Редактор диалогов. Так как экранные диалоги связаны с бизнес объектами и могут создаваться при проектировании структуры бизнес объектов, Редактор диалогов вызывается по месту редактирования структуры бизнес объекта. В последующих главах показано как создать экранную форму первичного документа, справочника с помощью Редактора диалогов.

Разработка первичных документов

Далее будет показано определение бизнес объектов, используемых в разрабатываемой схеме. Так как наш учет строится на регистрации учетных событий в документах, нам требуется определить документы, их экранные и печатные формы.

Проектируем Приходную накладную

Реквизиты Приходн ой накладн ой

Наша Приходная накладная позволяет хранить номер накладной, дату ее составления, название продавца, выбранного из Справочника контрагентов, а также дополнительную информацию в поле Основание. Все эти реквизиты составляют шапку накладной. Разумеется нам требуется хранить в накладной информацию о приходуемых товарах их закупочной цене, количестве. Так как каждая накладная может содержать информацию более чем об одном товаре и их общее количество в общем случае различается для разных накладных, нам необходима табличка, позволяющая заносить в накладную, информацию для требуемого количества товаров.

Первым шагом создадим новый документ, вызвав для этого диалог создания нового документа, как показывает Иллюстрация 13 на странице 19, и зададим название «Приходная накладная», как показывает Иллюстрация 15 на странице 21. Описание задавать не обязательно.

Далее закроем диалог и откроем диалог создания атрибута шапки Приходной накладной как показывает на странице . Определим атрибут Номер, указав тип данных Строка, как показывает Иллюстрация 18 на странице 24. Закроем диалог и аналогично определим атрибуты Дата, Продавец, Основание, задав типы данных дата, Справочник Контрагентов, Строка соответственно. Получилось? Разумеется, не возможно задать тип данных Справочник Контрагентов, так как такого справочника еще нет. Поэтому, сделайте паузу в работе над Приходной накладной и займитесь справочником Контрагентов как объясняется в разделе Проектируем Справочник Контрагентов на странице 42. Однако, не обязательно сейчас полностью переключиться на Справочник Контрагентов, достаточно создать его, задав название. Определение реквизитов Контрагентов

можно оставить на потом. После создания Справочника Контрагентов полю Продавец можно, наконец, задать тип Справочник Контрагентов, что демонстрирует Иллюстрация 22 на странице 28.

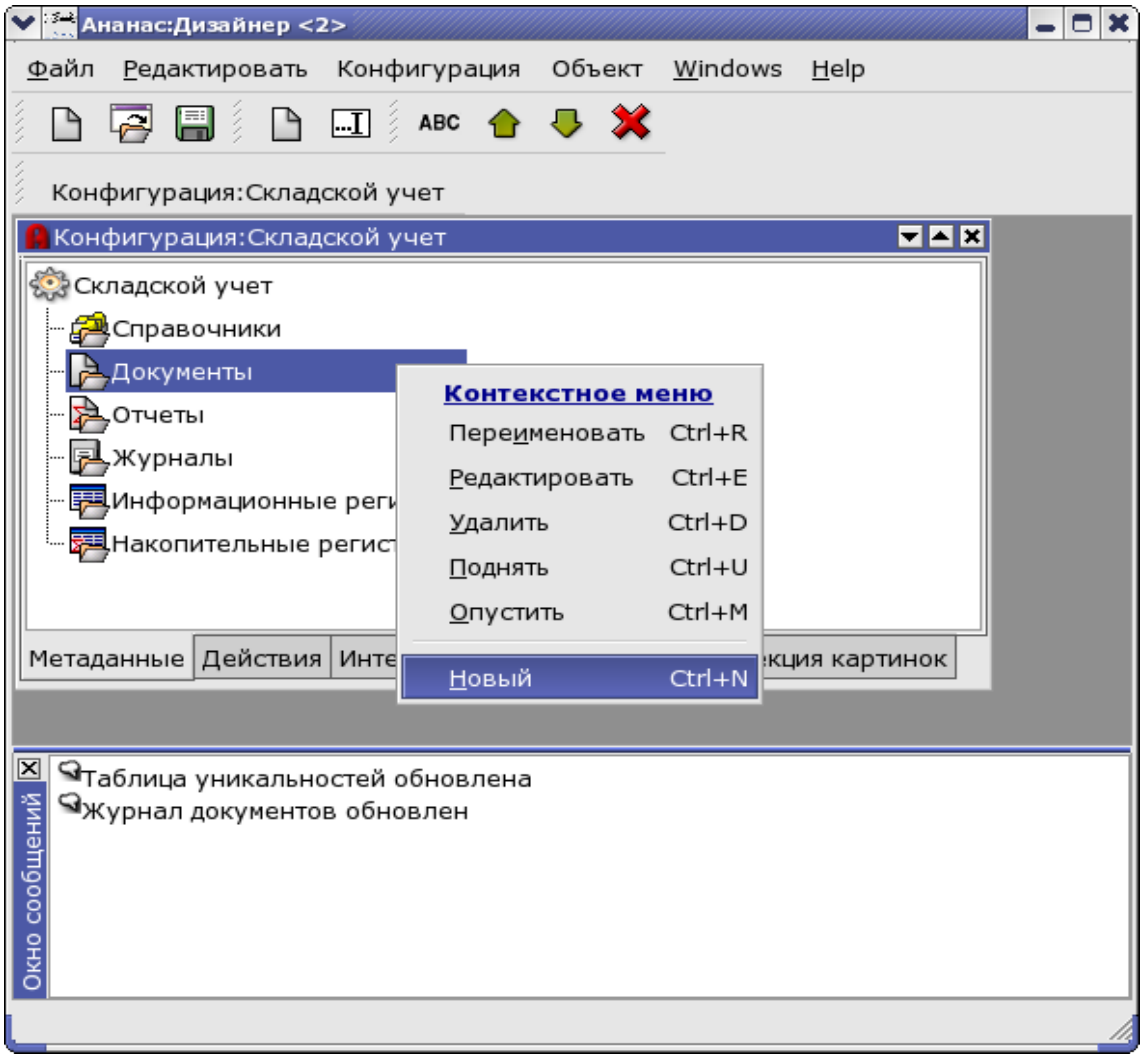


Иллюстрация 13: Вызов диалога создания нового документа

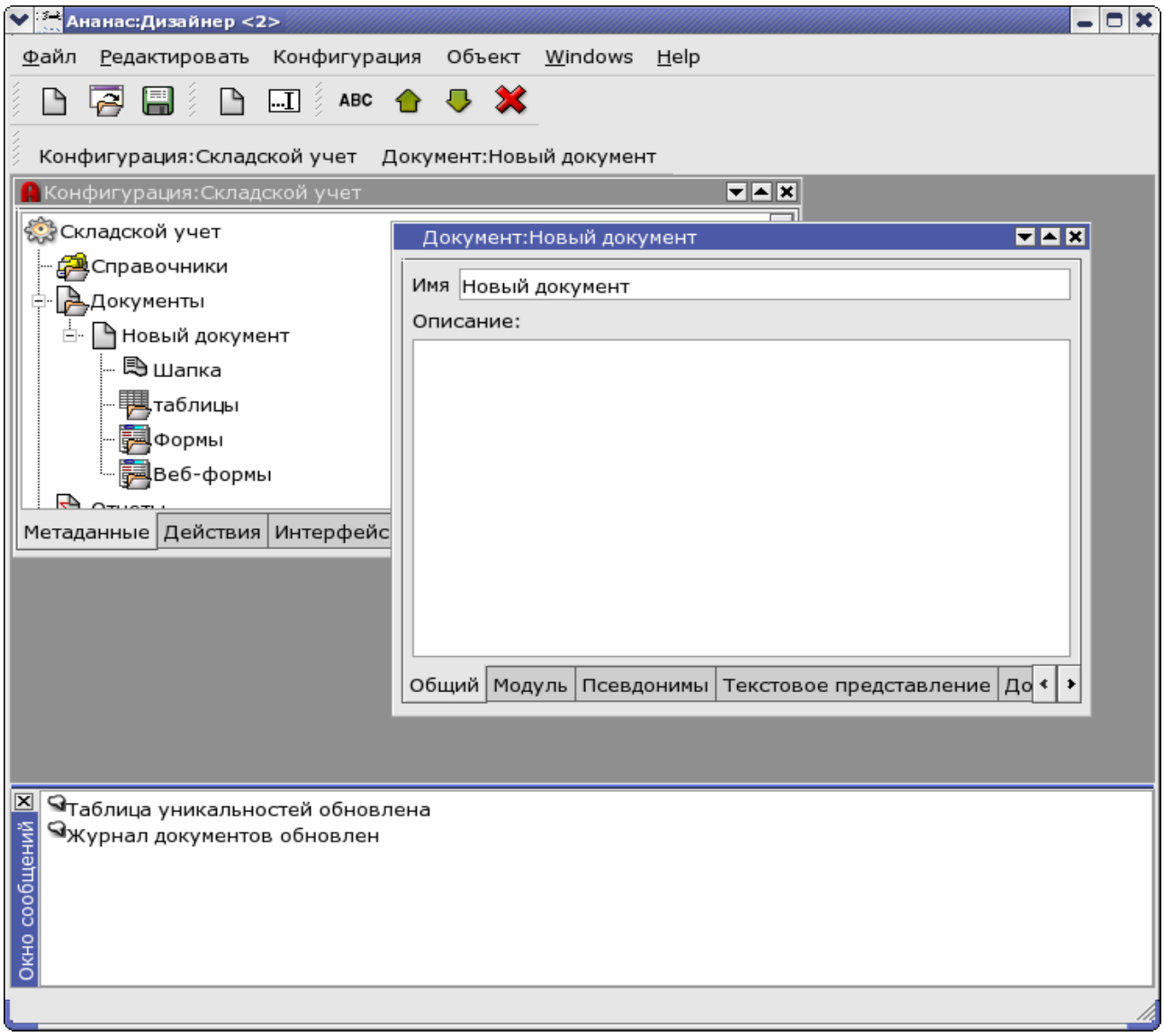


Иллюстрация 14: Открылся диалог создания нового документа

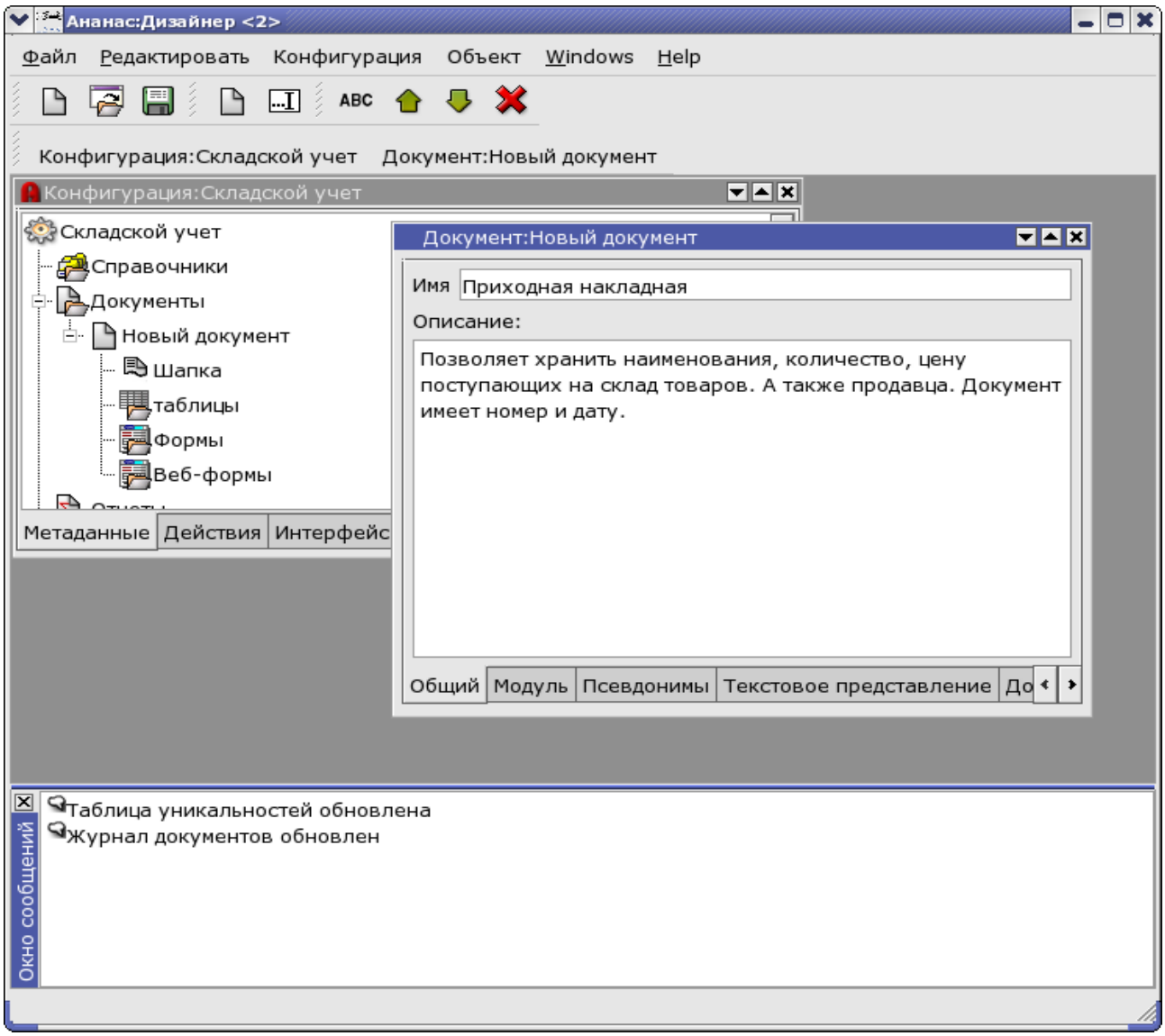


Иллюстрация 15: Задаем название нового документа

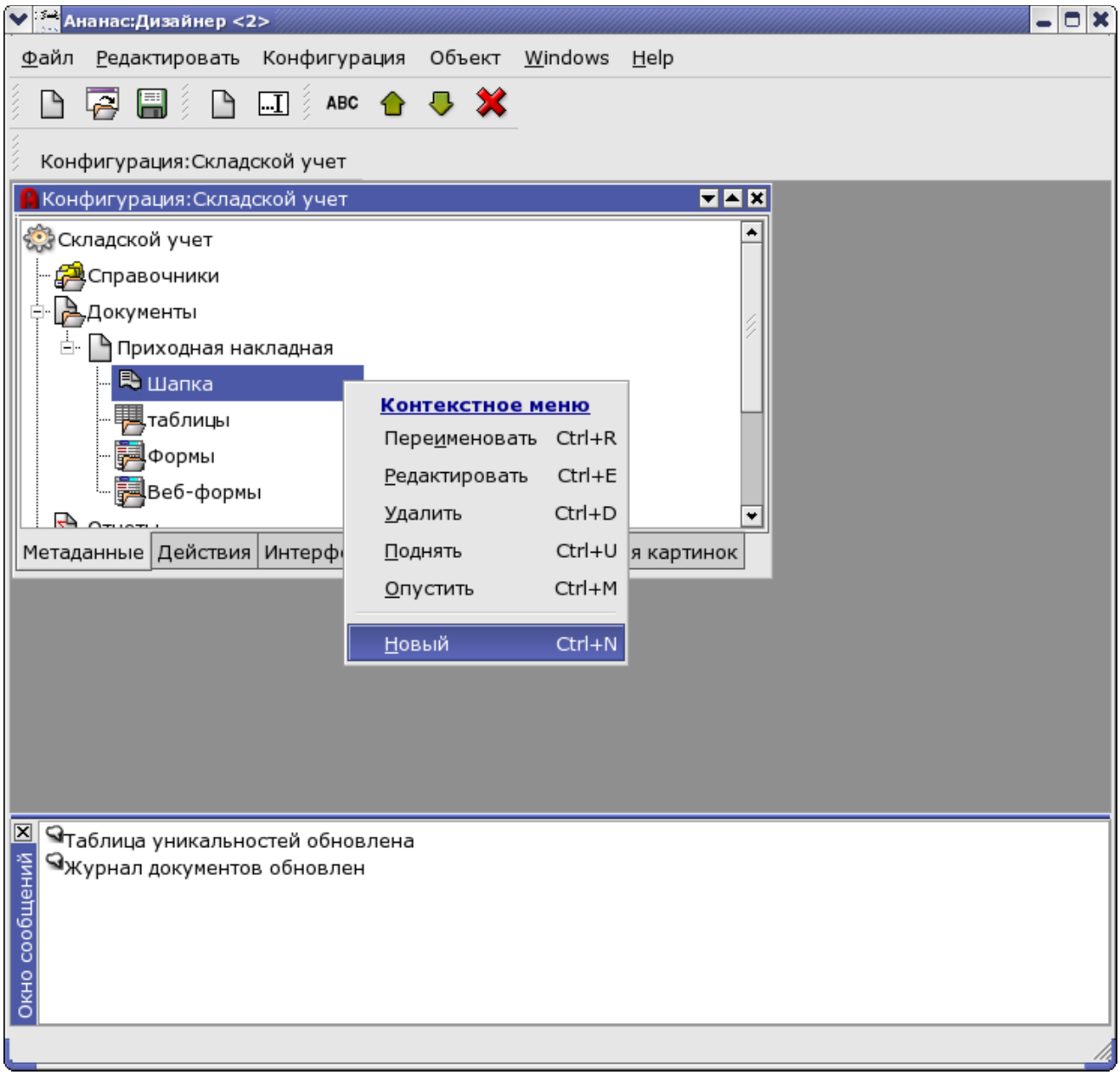


Иллюстрация 16: Вызов диалога редактирования нового реквизита шапки документа

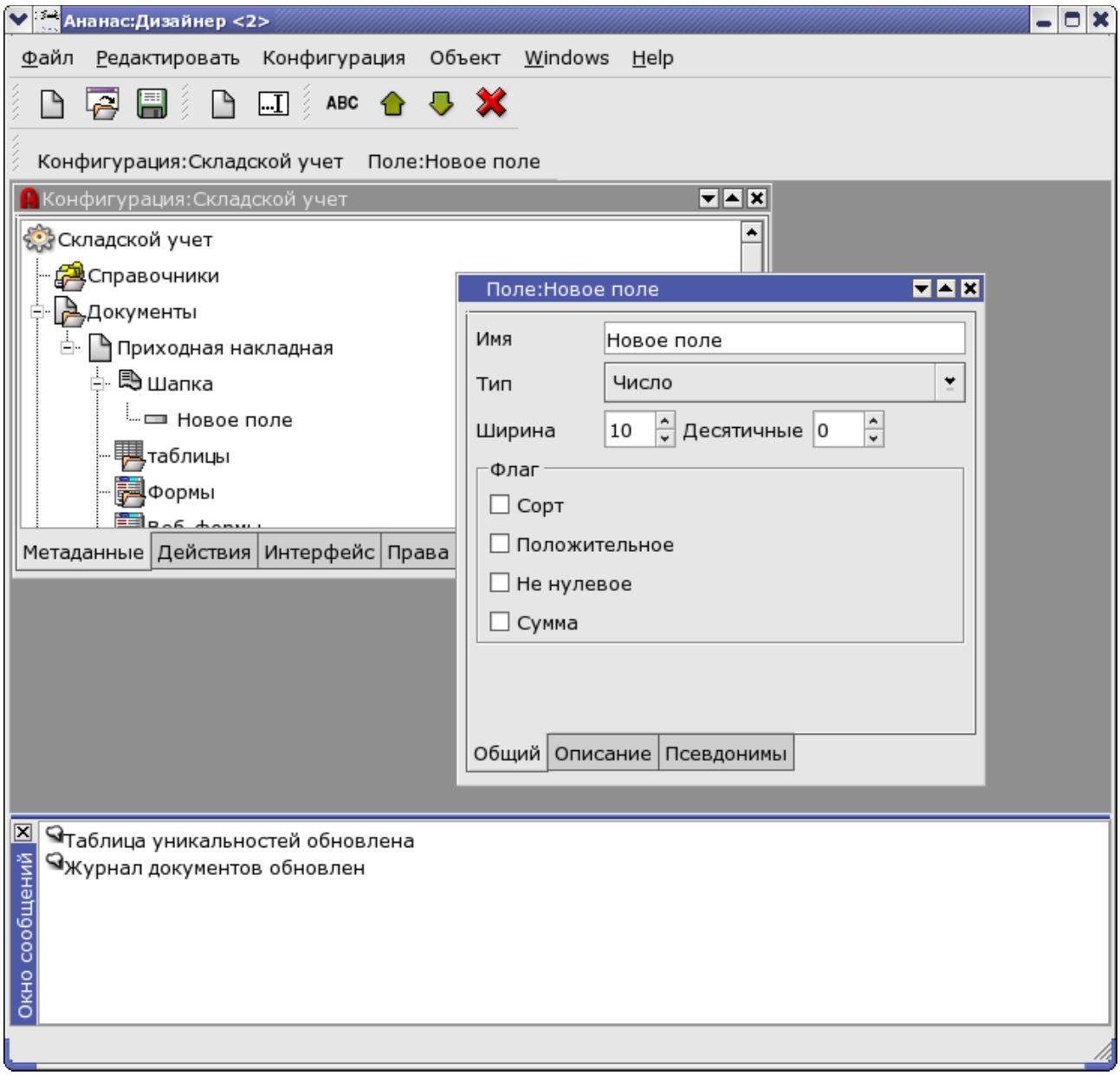


Иллюстрация 17: Открыт диалог редактирования нового реквизита шапки документа

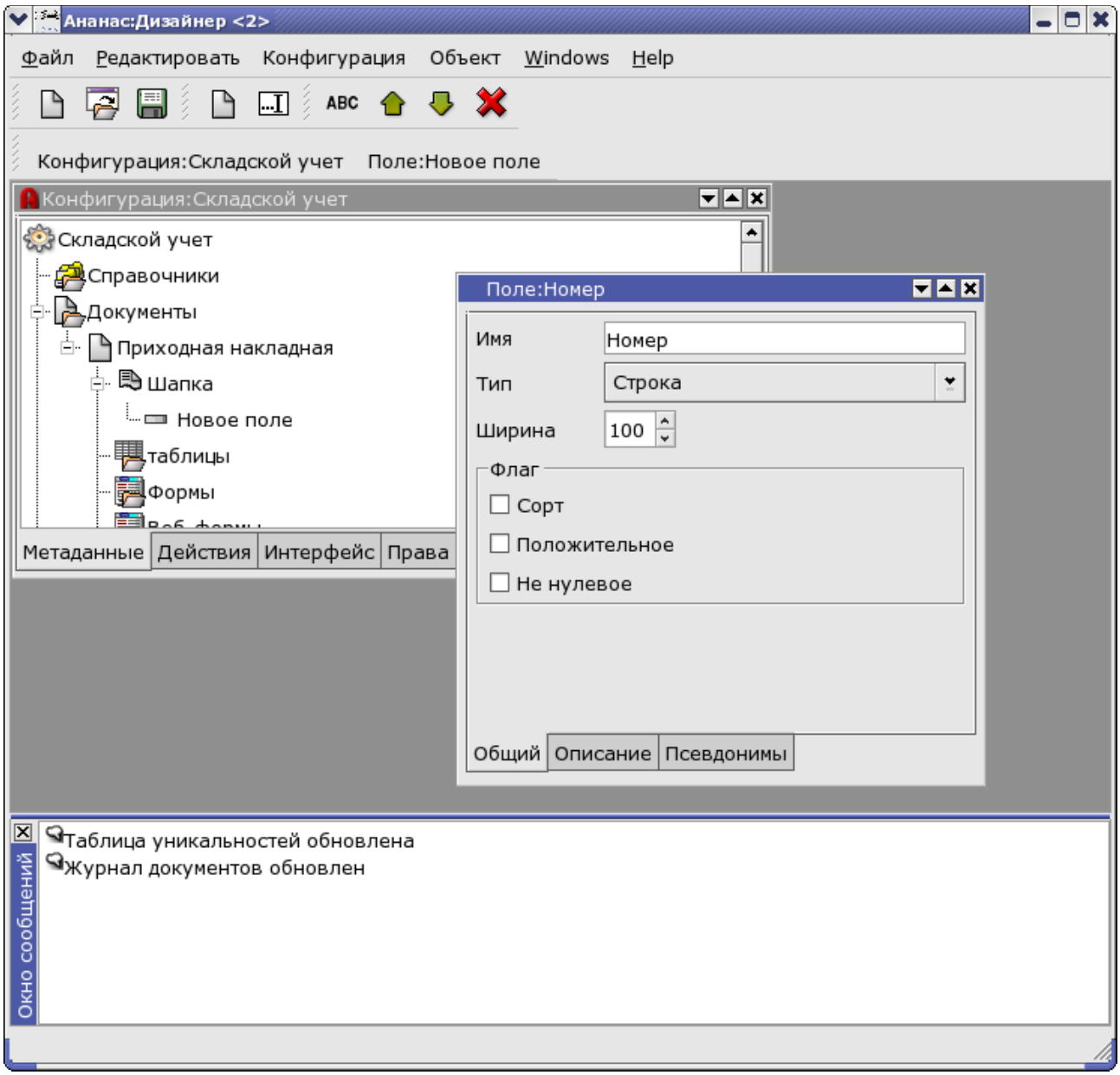


Иллюстрация 18: Определение реквизита Номер шапки документа

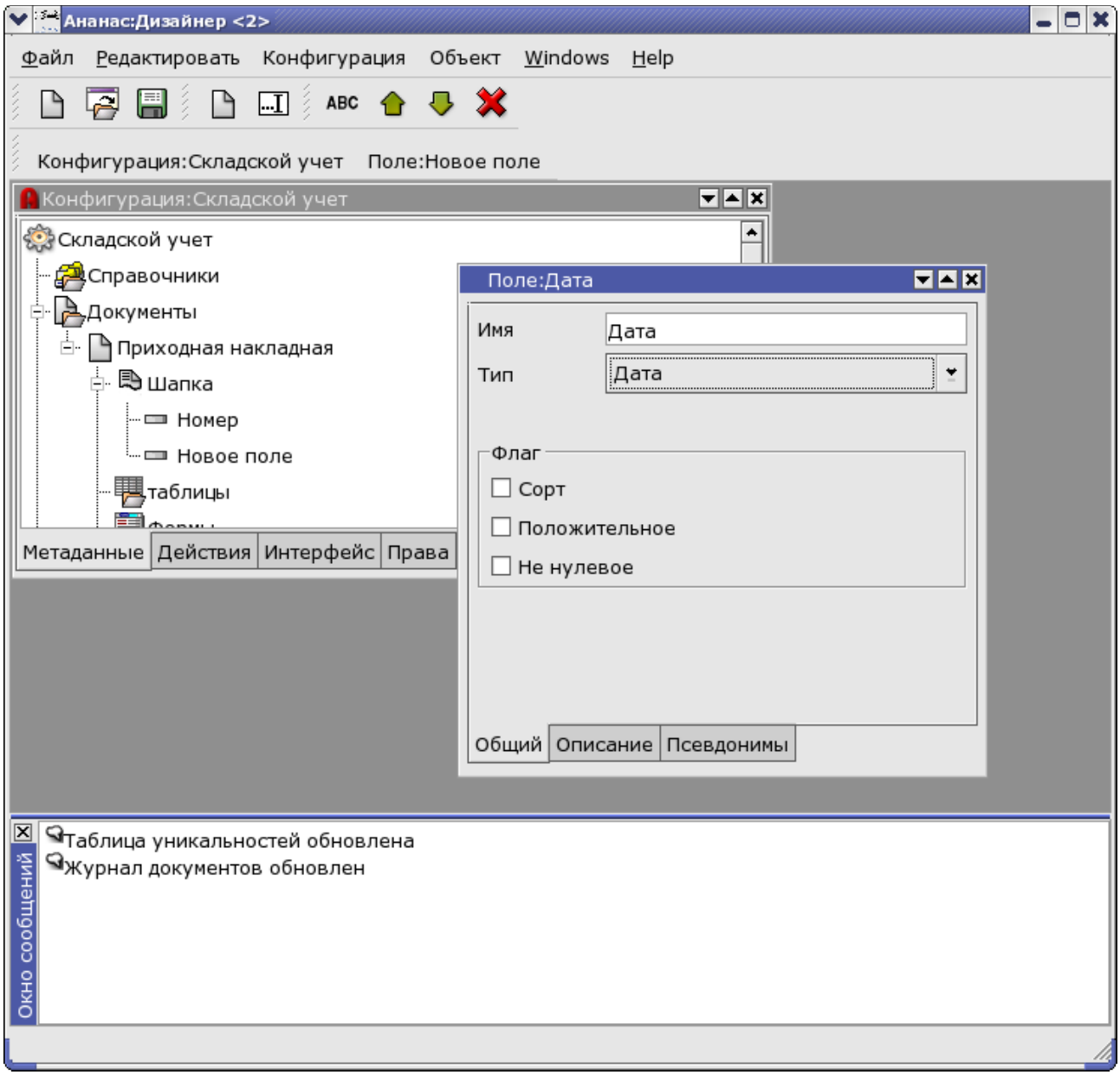


Иллюстрация 19: Определение реквизита Дата шапки Приходной накладной

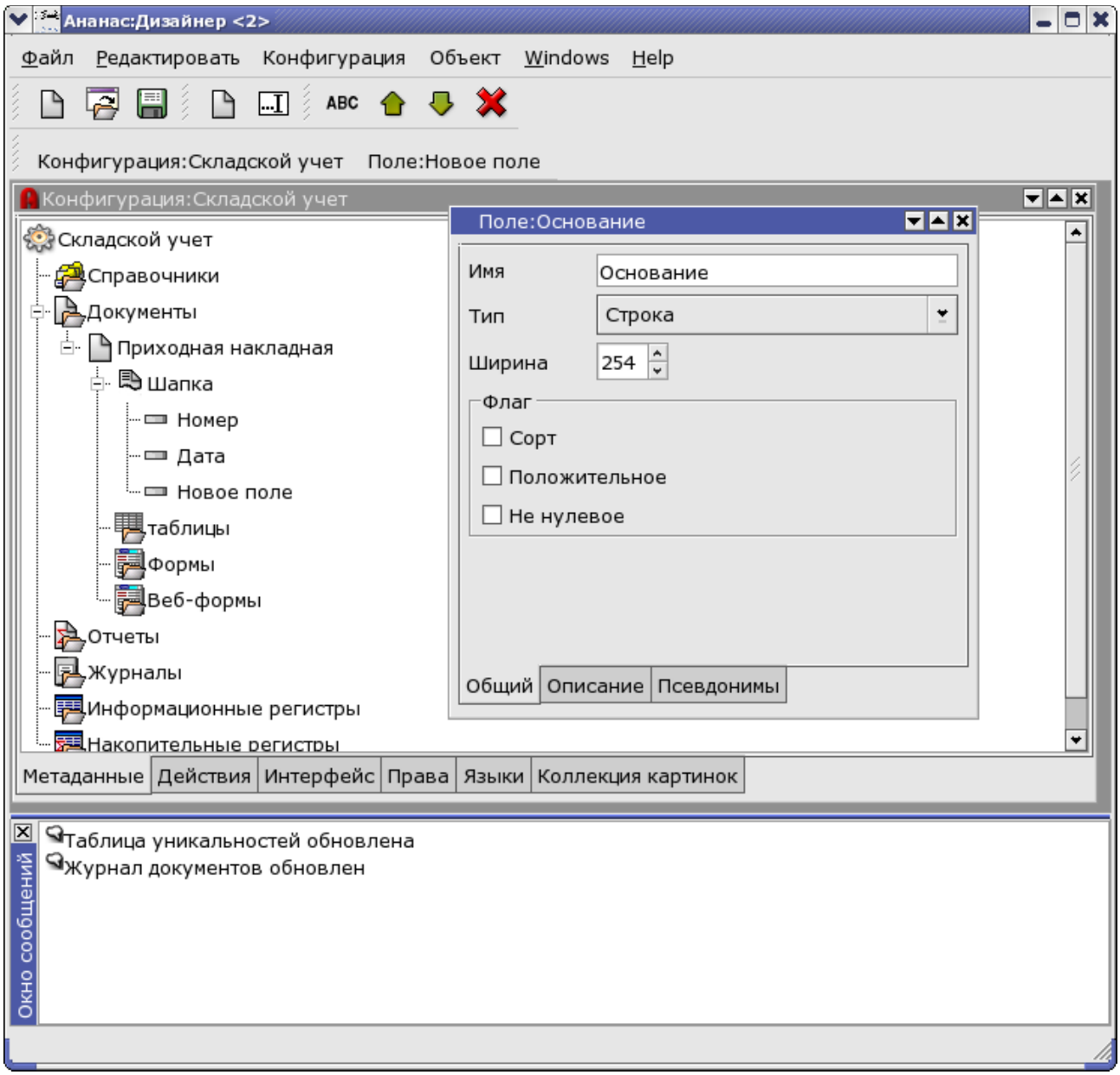


Иллюстрация 20: Определяем реквизит Основание шапки Приходной накладной

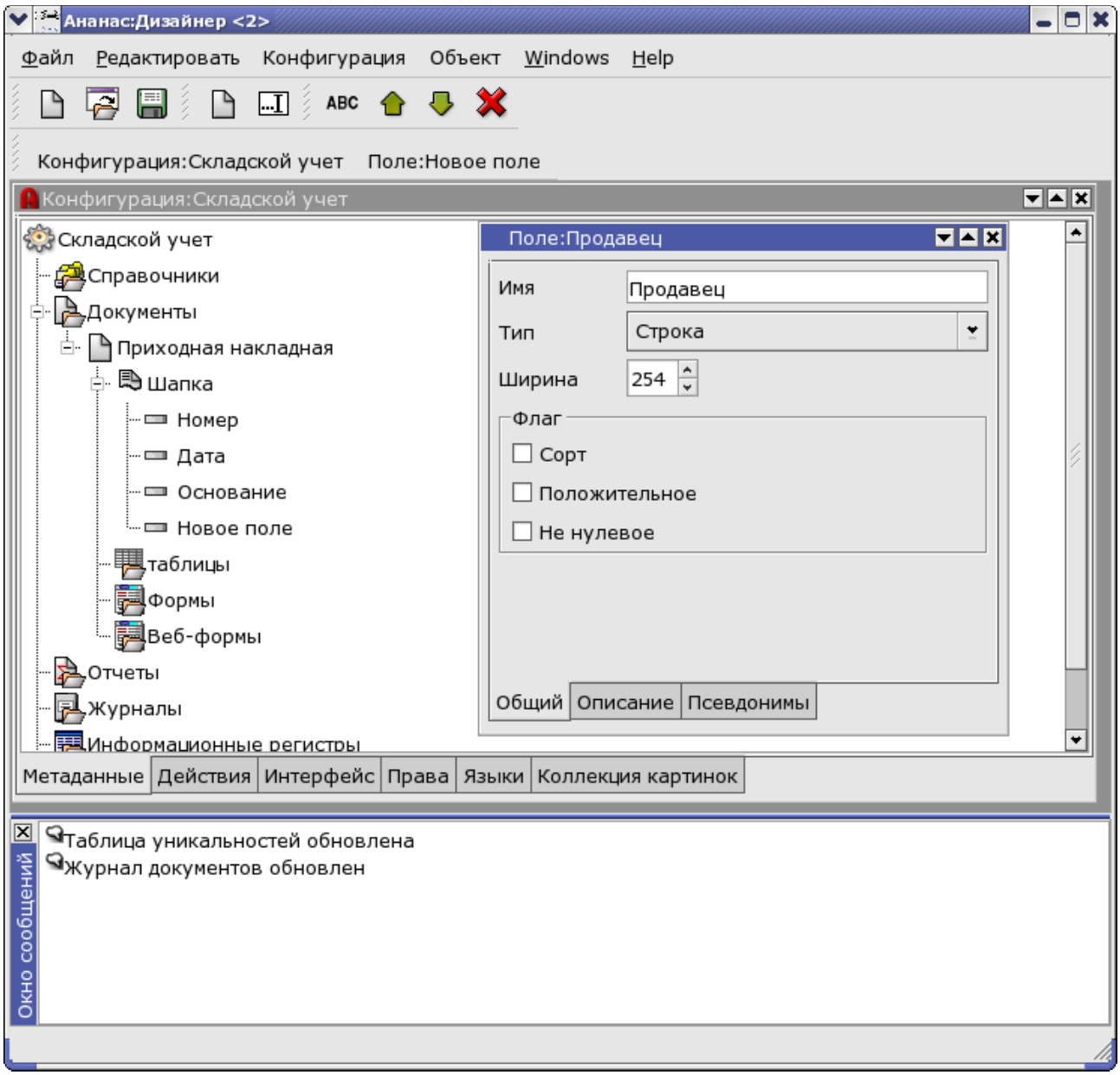


Иллюстрация 21: Определяем реквизит Продавец шапки Приходной накладной

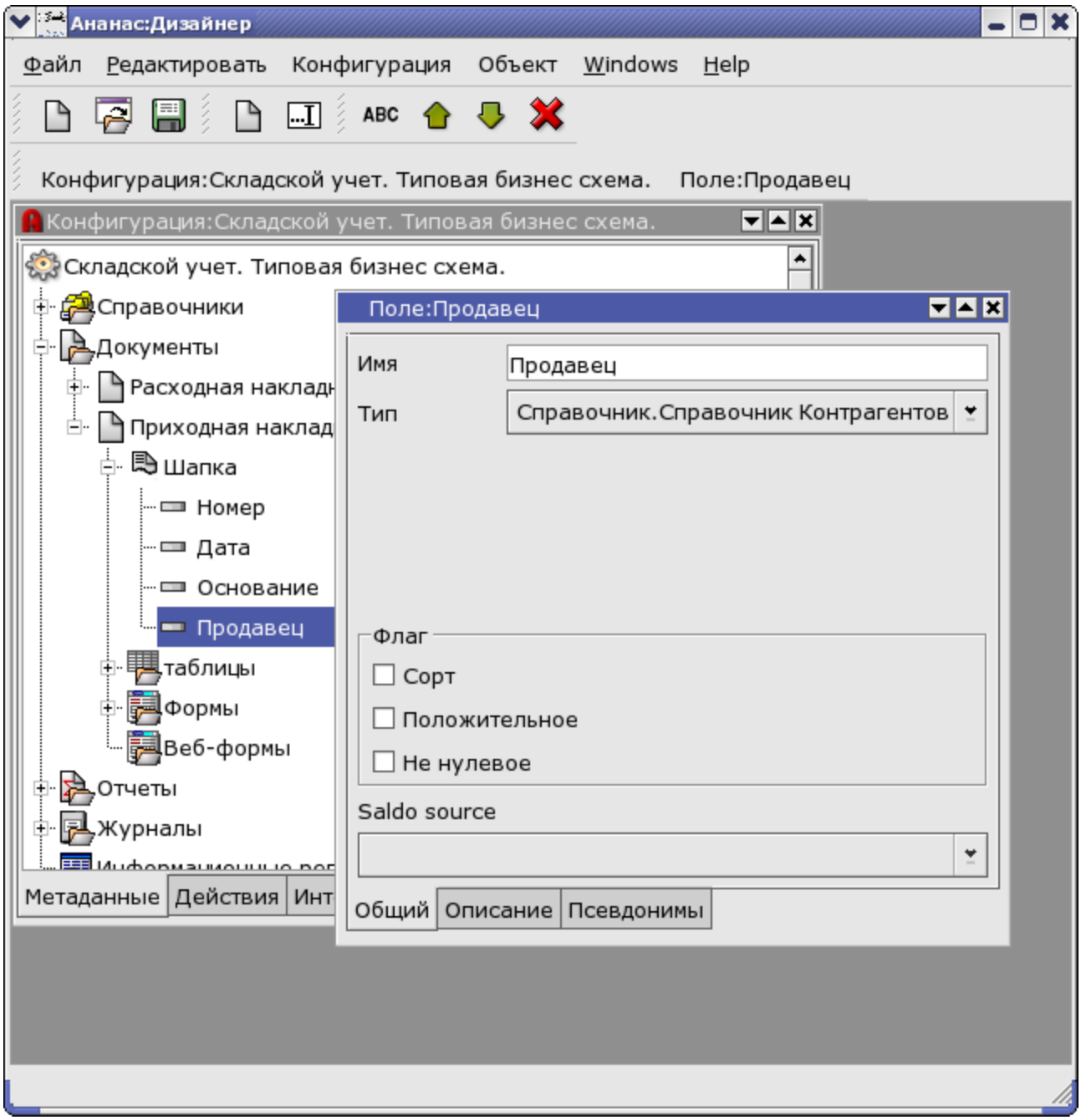


Иллюстрация 22: Реквизит Продавец имеет сложный тип данных

Следующим шагом определяем состав столбцов таблицы, в которую будет заноситься информация по товарам. Столбец Наименование имеет сложный тип данных, его значение задается выбором из Справочника товаров. Поэтому для задания типа этому столбцу необходимо в дереве метаданных уже иметь Справочник товаров. Создание Справочника товаров описано в разделе на странице .

Реквизиты табличной части Приходн ой накладн ой

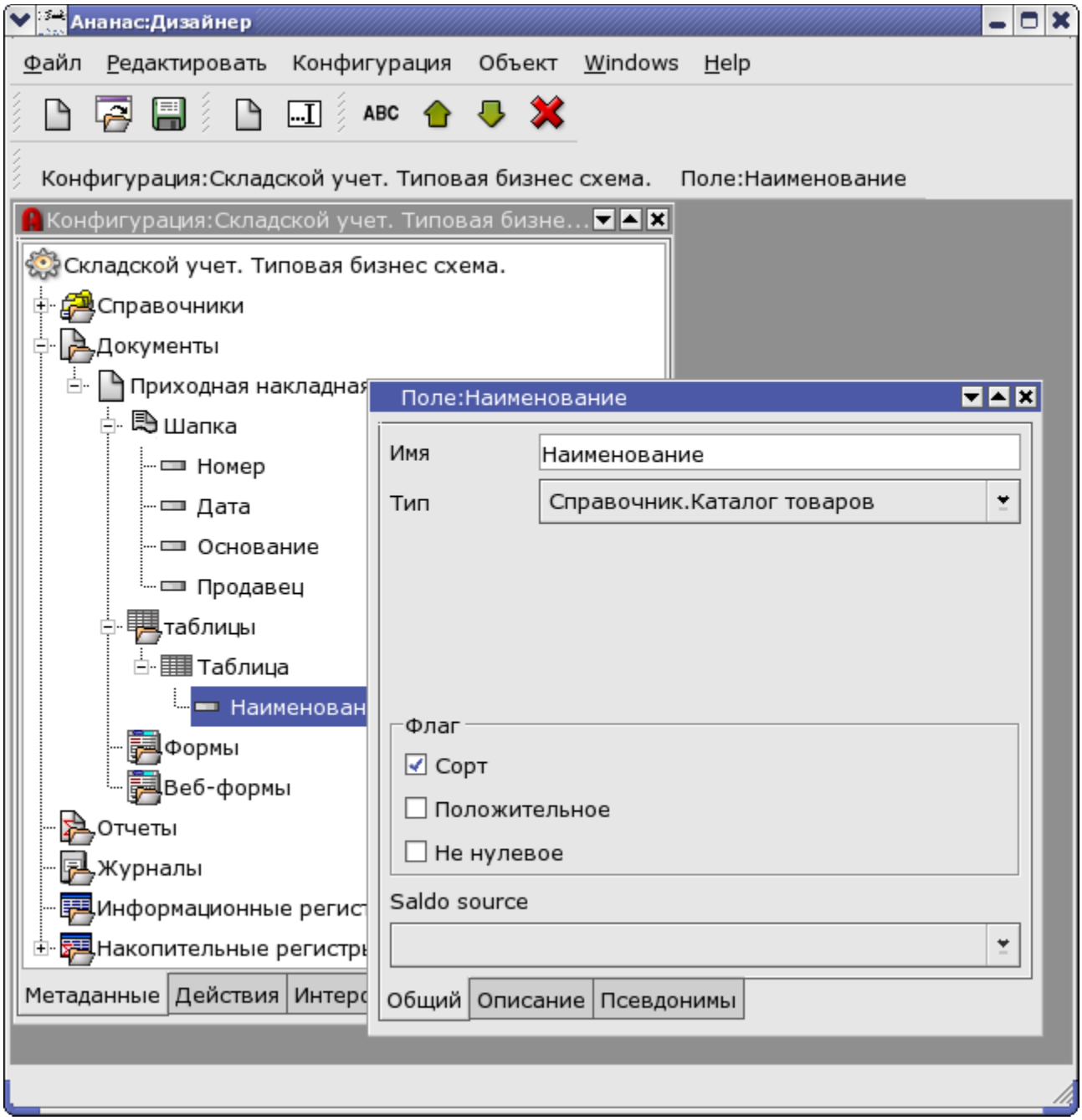


Иллюстрация 23: Реквизит Наименование товара табличной части Приходной накладной

Экранная форма Приходной накладн ой

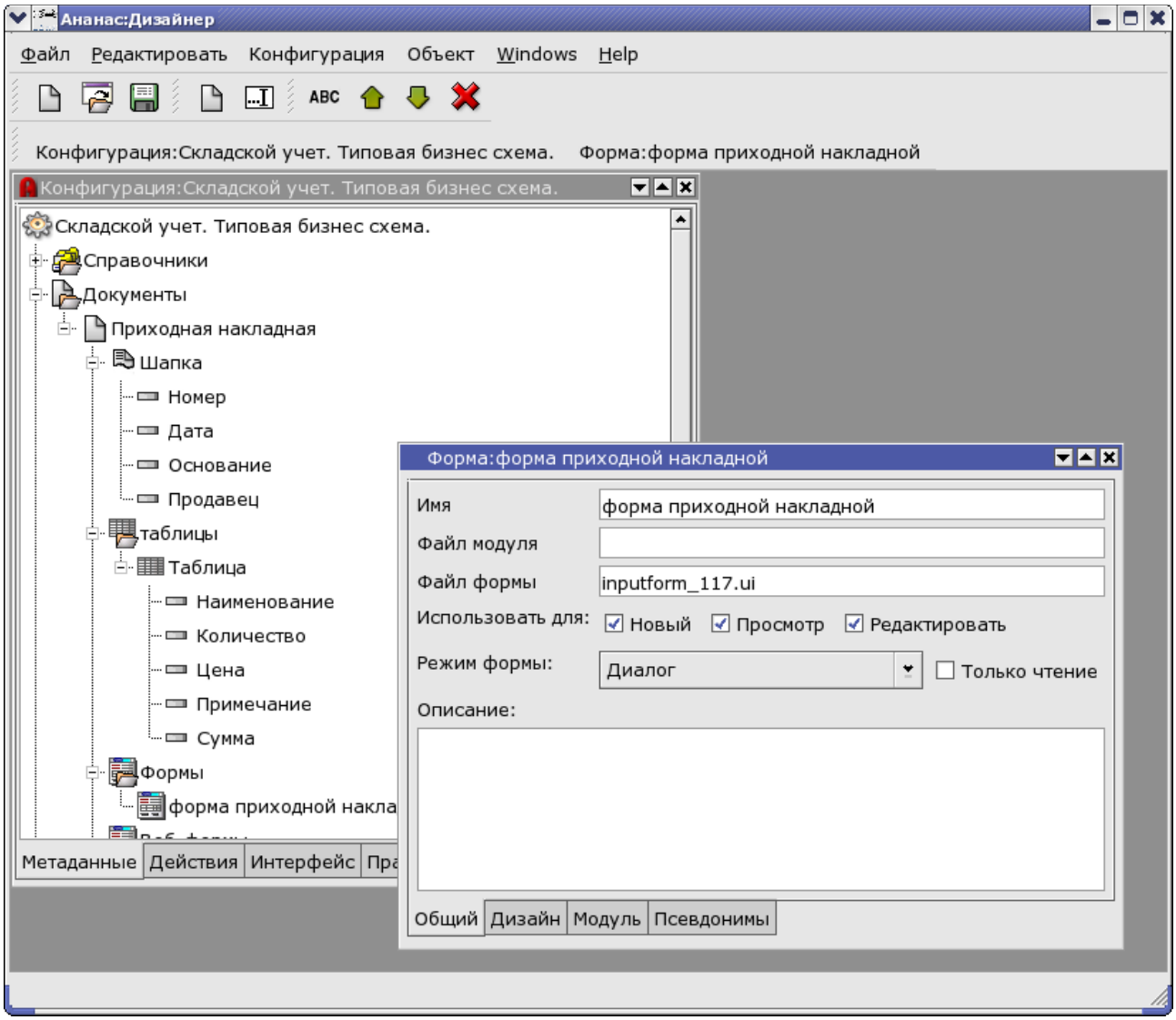


Иллюстрация 24: Диалог редактирования свойств экранной формы Приходной накладной

Ниже приведен код функции модуля экранной формы приходной накладной. Функция обрабатывает событие нажатия кнопки пользователем. Если пользователь нажал кнопку Печать, для получения печатной формы, происходит выполнение кода, приводящее к формированию документа на основе шаблона и открытие сформированного документа в редакторе OpenOffice Writer. Таким образом пользователь получает возможность распечатать на принтере и/или сохранить полученный документ в файловой системе.

```
/*
 * Copyright (c) Grigory Panow, Yoshkar-Ola, 2005
 * Ananas Project http://ananas.lrn.ru
 */
function on_button(button_name) // обработчик нажатия кнопки
{
    if(button_name!="print_button") return; // обрабатываем только нажатие кнопки Печать
    var p = new Report("Report1","1"); // создаем экземпляр отчета
    p.setTable("templ_prihod.sxw"); // связываем его с шаблоном
    // шаблоны ищутся в каталоге `Рабочий каталог'
    p.setValue("param",Value("Номер") ); // устанавливает значение для простого тега
```

```
    p.exec("Документ.Номер"); // применяет установленное значение для тега с именем
<:Документ.Номер:>
    p.setValue("param",Value("Основание") );
    p.exec("Документ.Основание");
    p.setValue("param",Value("Продавец"));
    p.exec("Документ.Продавец");
    p.setValue("param",Value("Дата") );
    p.exec("Документ.Дата");
    var sum=0;
    var countRow =  TabCount("wDBTable1");   // количество строк таблицы
    var i;
    for ( i =countRow-1; i>=0; i--)
    {
            p.setValue("npp", String(i+1)); // устанавливаем значение табличного тега с именем `npp'
            for ( j=1; j<=5; j++)
            {
                      p.setValue("f"+String(j), TabValue("wDBTable1",i,j-1));
            }
            p.exec("table_section"); // применяем установленные значения для строки
                                  //содерщей табличный тег [:table_section:]
    }
    sum = Value("lineEdit1");
    p.setValue("param",sum);
    p.exec("Итого");
    p.setValue("param", Propis(sum)); // записываем сумму прописью
    p.exec("Итого_прописью");
    p.setValue("param","Сидоров С.С");
    p.exec("Сдал");
    p.setValue("param","Петров П.П");
    p.exec("Принял");
    p.setValue("param",getConstant("Константы","Главный бухгалтер")); // получаем имя главного
бухгалтера из справочника констант
    p.exec("Организация.Главбух");
    p.show(); // запускаем OpenOfficce
    p.close(); // удаляем все временные файлы
}
```

Модуль экранной формы

Модуль экранной формы содержит код на Ананас-Скрипте. Код модуля может выполнять различные задачи. Например, обрабатывать нажатие пользователем кнопки экранной формы или перемещение между строками таблицы. В документах, таких как Приходная накладная, операции по заполнению регистров выполняются Ананас-Скриптом, хранящимся в модуле экранной формы. Модуль экранной формы имеет предопределенный набор именованных событий, обработчики которых должны задаваться функциями на Ананас-Скрипте.

on_formstart()

Событие, инициируемое непосредственно перед отображением формы на экране. Для его обработки необходимо определить функцию

```
function on_formstart(){
        //... some code here
}
```

on_formstop()

Событие, инициируемое непосредственно перед закрытием экранной формы. Для его обработки необходимо определить функцию

```
function on_formstop(){
        //... some code here
}
```

on_conduct()

Событие, инициируемое при сохранении документа для выполнения действий по его проведению. Например, для заполнения информационных регистров. Для его обработки необходимо определить функцию

```
function on_conduct(){
        //... some code here
}
```

on_button(name)

Событие, инициируемое при нажатии пользователем любой из кнопок экранной формы. В качестве параметра обработчику передается имя нажатой кнопки. Для его обработки необходимо определить функцию

```
function on_button(){
        //... some code here
}
```

on_tabupdate(row,col,tableName)

Событие, инициируемое при изменении значения ячейки таблицы экранной формы. Экранная форма документа может содержать более одной таблицы с данными, поэтому кроме номера строки и номера столбца ячейки в параметрах передается также имя таблицы. Для его обработки необходимо определить функцию

```
function on_tabupdate(row,col,tableName){
        //... some code here
}
```

on_tablerow(tableName)

Событие, инициируемое при перемещении пользователем фокуса ввода по строкам таблицы. В качестве параметра передается имя таблицы, так как экранная форма может содержать более одной таблицы с данными. Для его обработки необходимо определить функцию

```
function on_tablerow(tableName){
        //... some code here
}
```

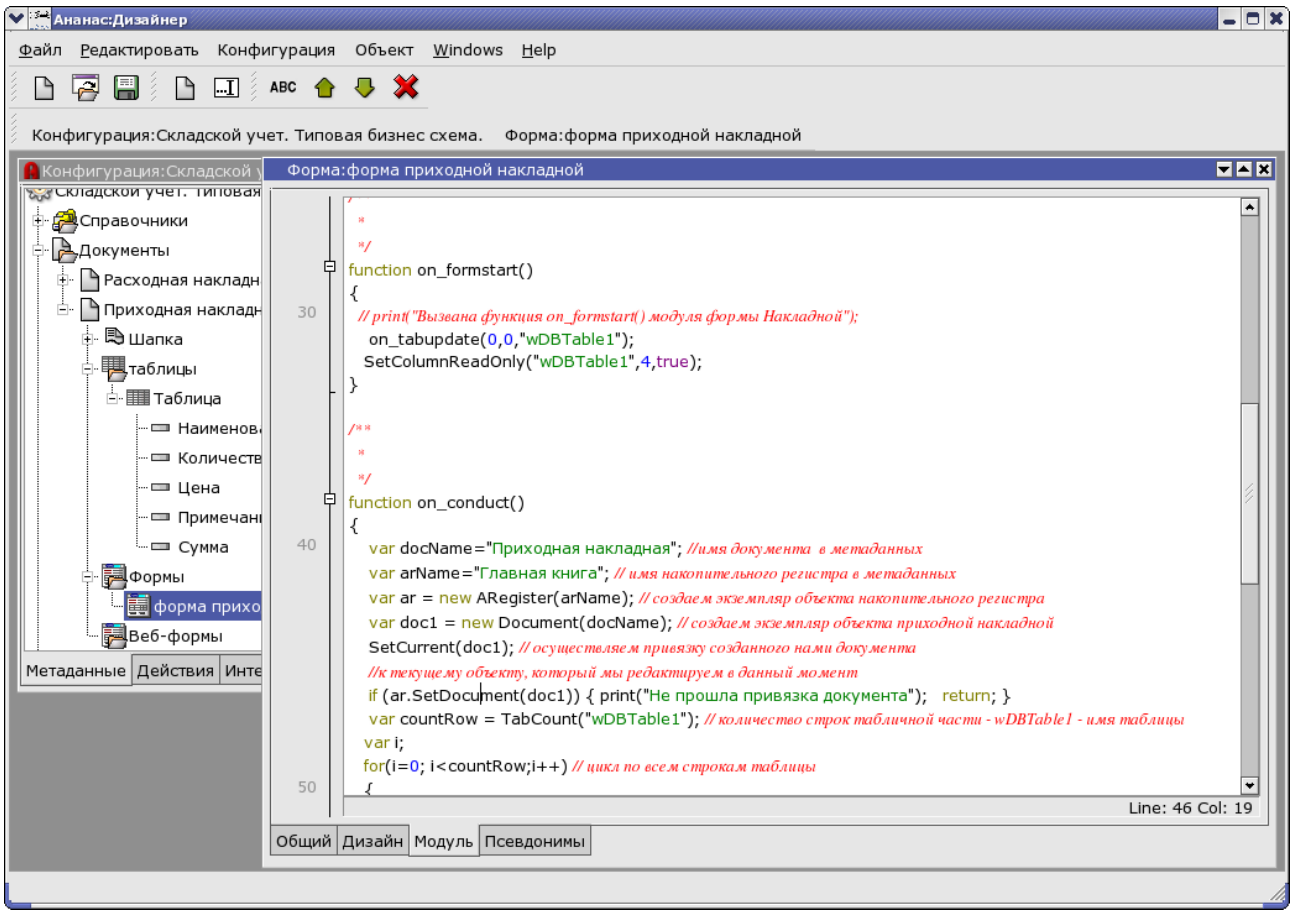


Иллюстрация 25: . Модуль экранной формы Приходной накладной

Код модуля исполняется сразу после создания экранной формы в памяти, но до ее появления на экране. Разумеется, что код выделенный в функции не будет выполнен, пока функции не будут вызваны либо экранной формой как обработчики событий, либо из кода модуля.

Диалог экранной формы

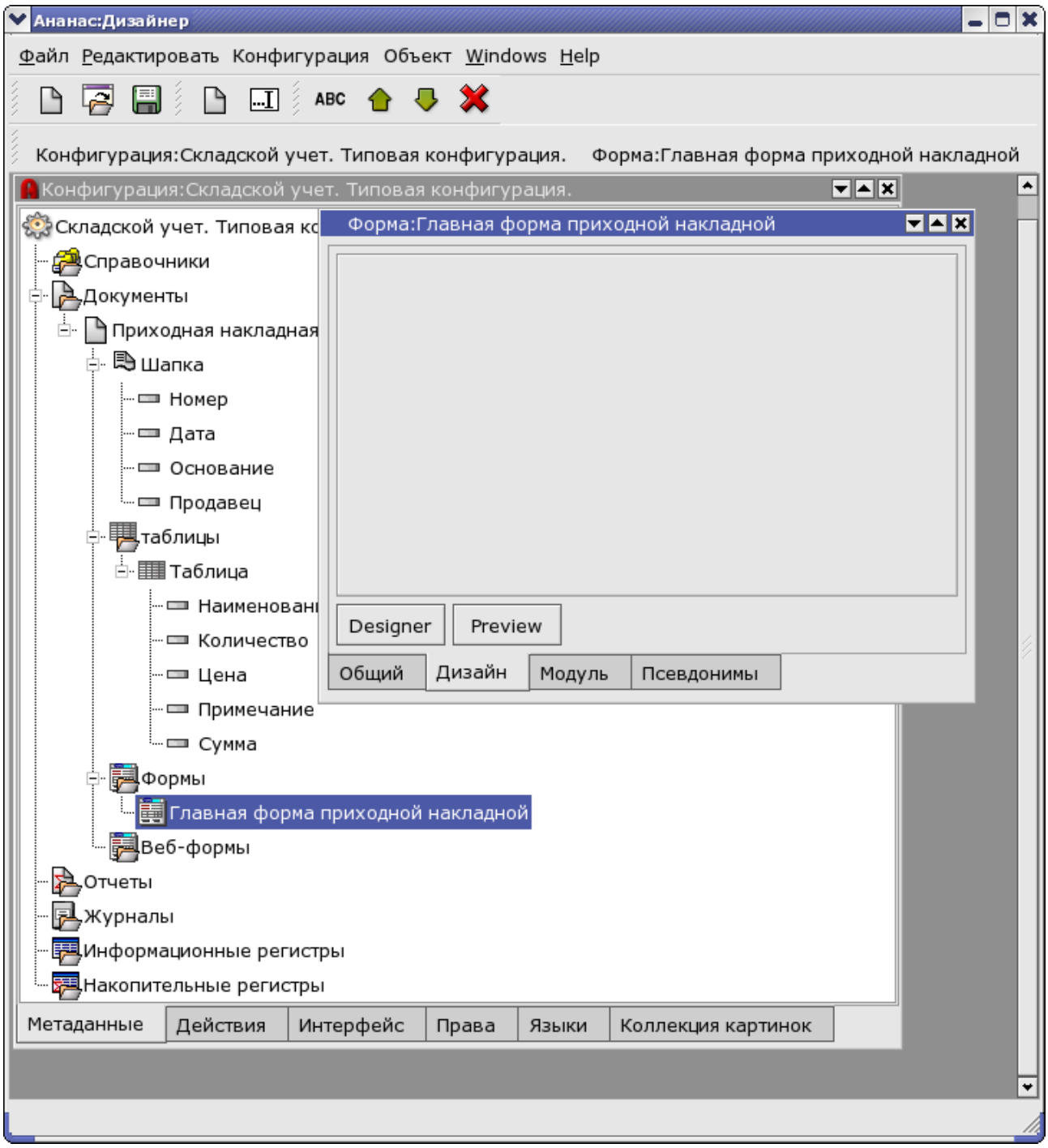


Иллюстрация 26: Открыта закладка для вызова редактора диалогов. Вызов производиться по нажатию кнопки Designer

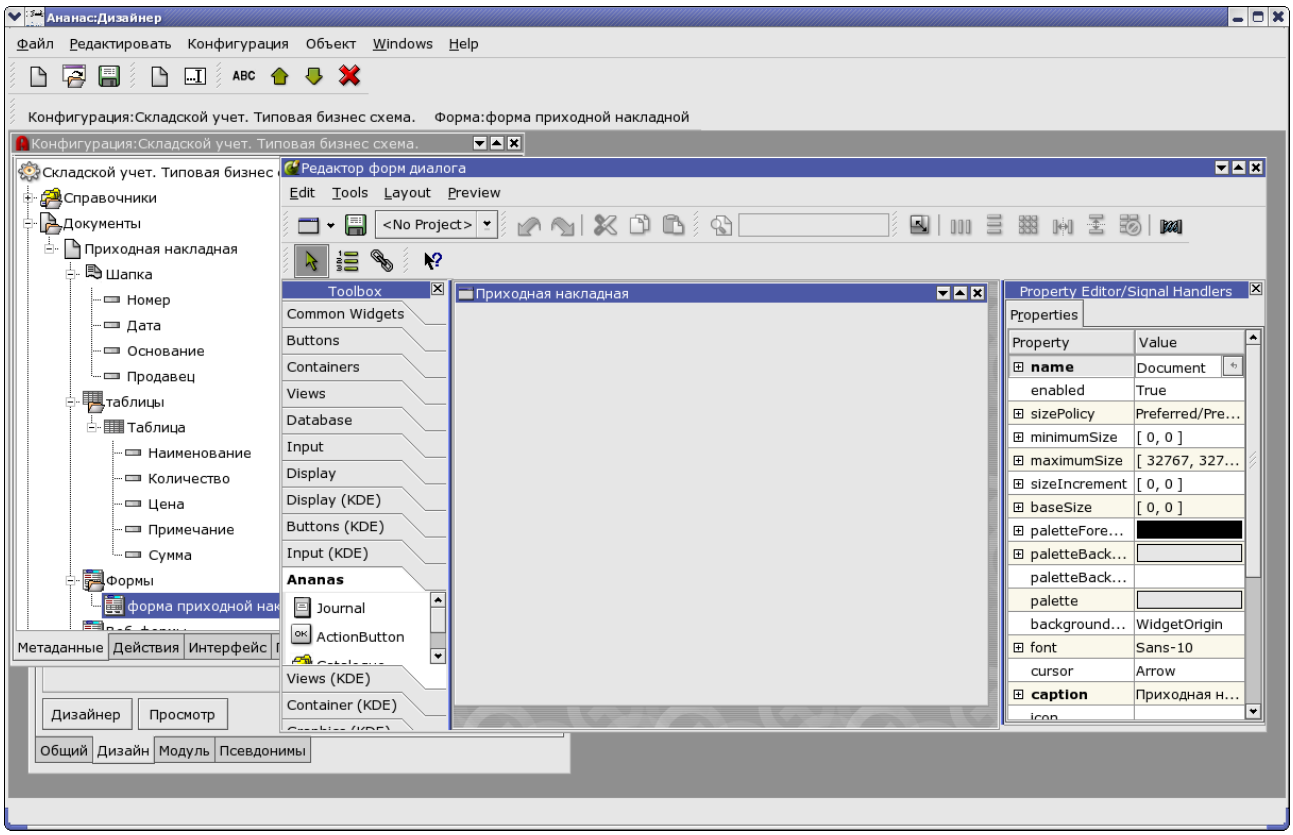


Иллюстрация 27: Открыт редактор диалогов для создания диалога экранной формы Приходной накладной

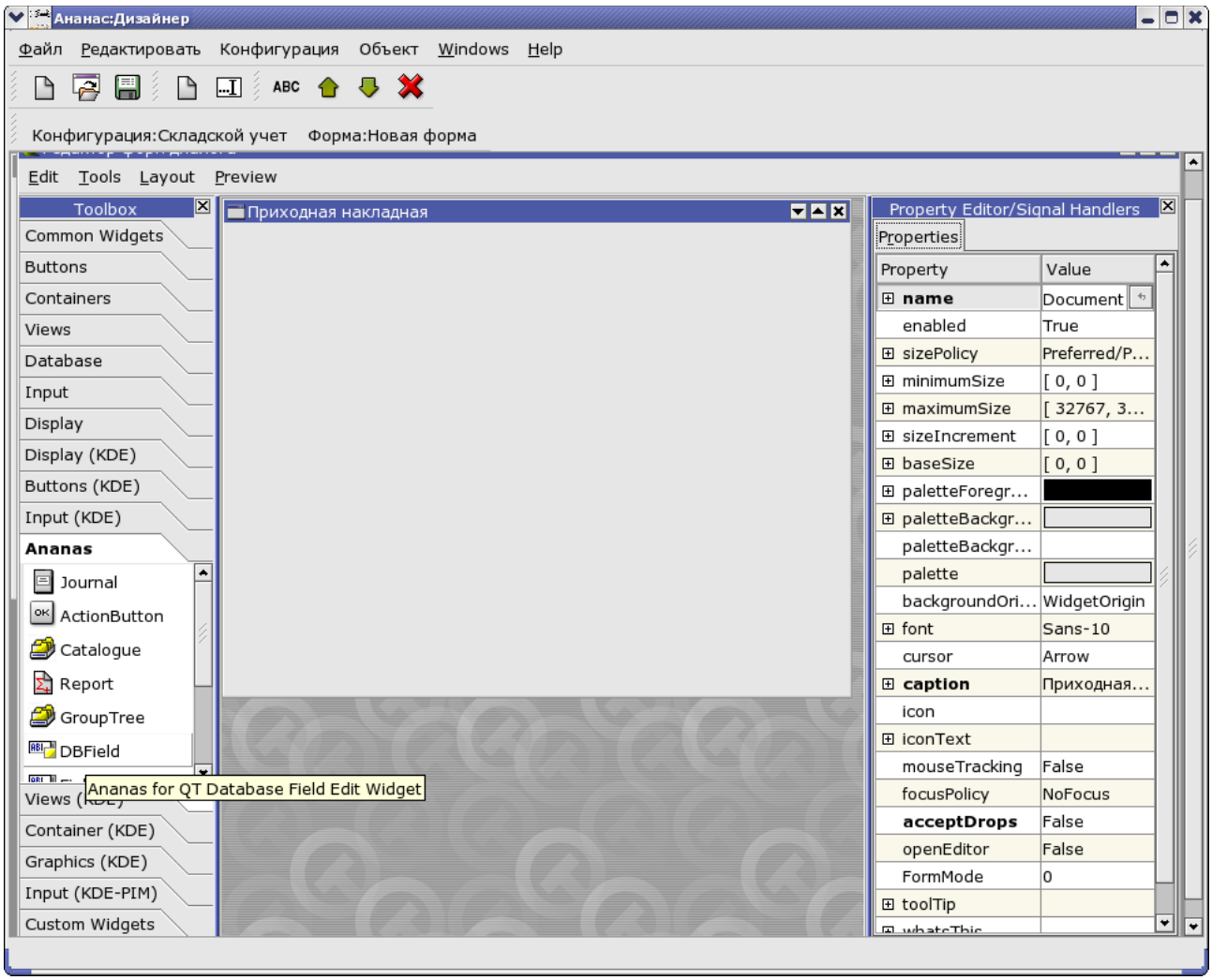


Иллюстрация 28: Выбор поля ввода из палитры виджетов Ананаса

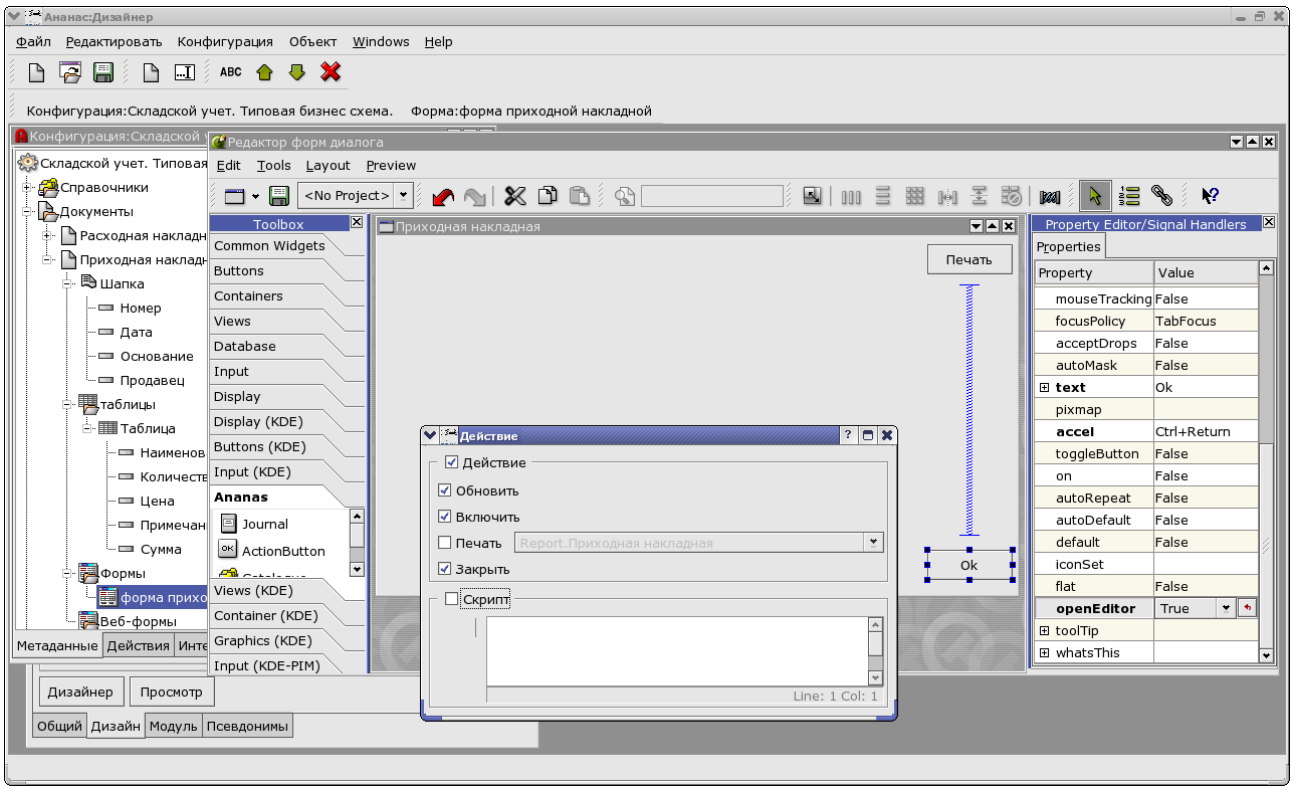


Иллюстрация 29: Задаем действие, выполняемое по нажатию кнопки Ок

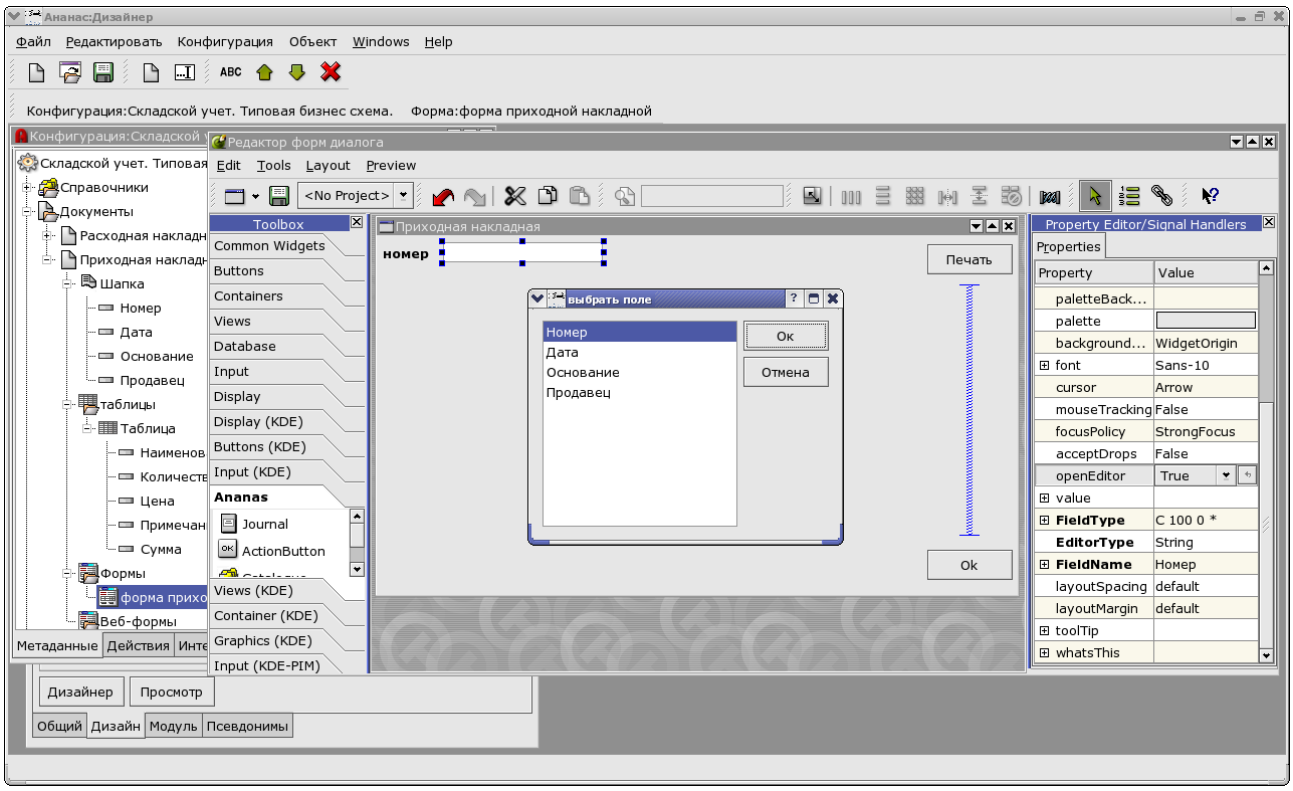


Иллюстрация 30: Задаем положение поля ввода и связываем его с полем "Номер" базы данных, которое мы определили ранее в метаданных

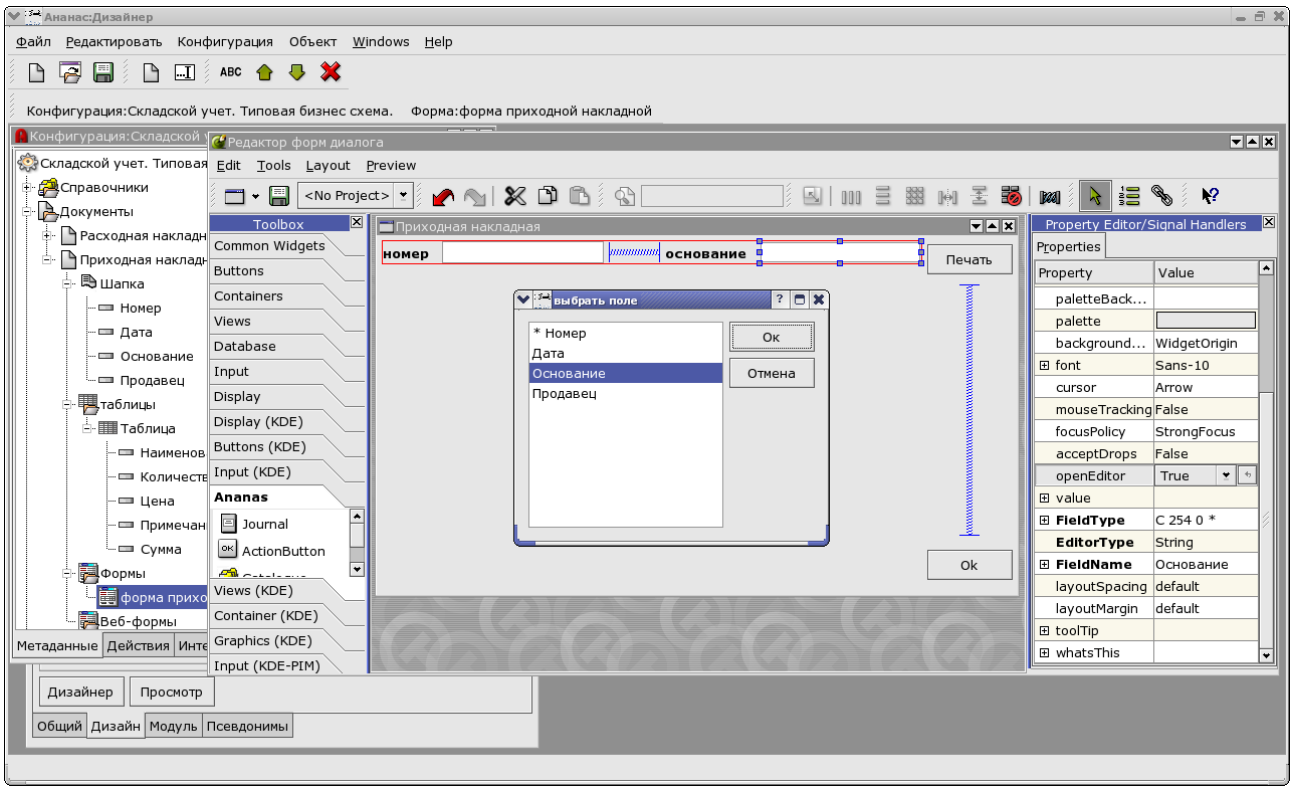


Иллюстрация 31: Размещаем поле ввода и связываем его с реквизитом «Основание» Приходной накладной в базе данных, который мы определили в метаданных ранее

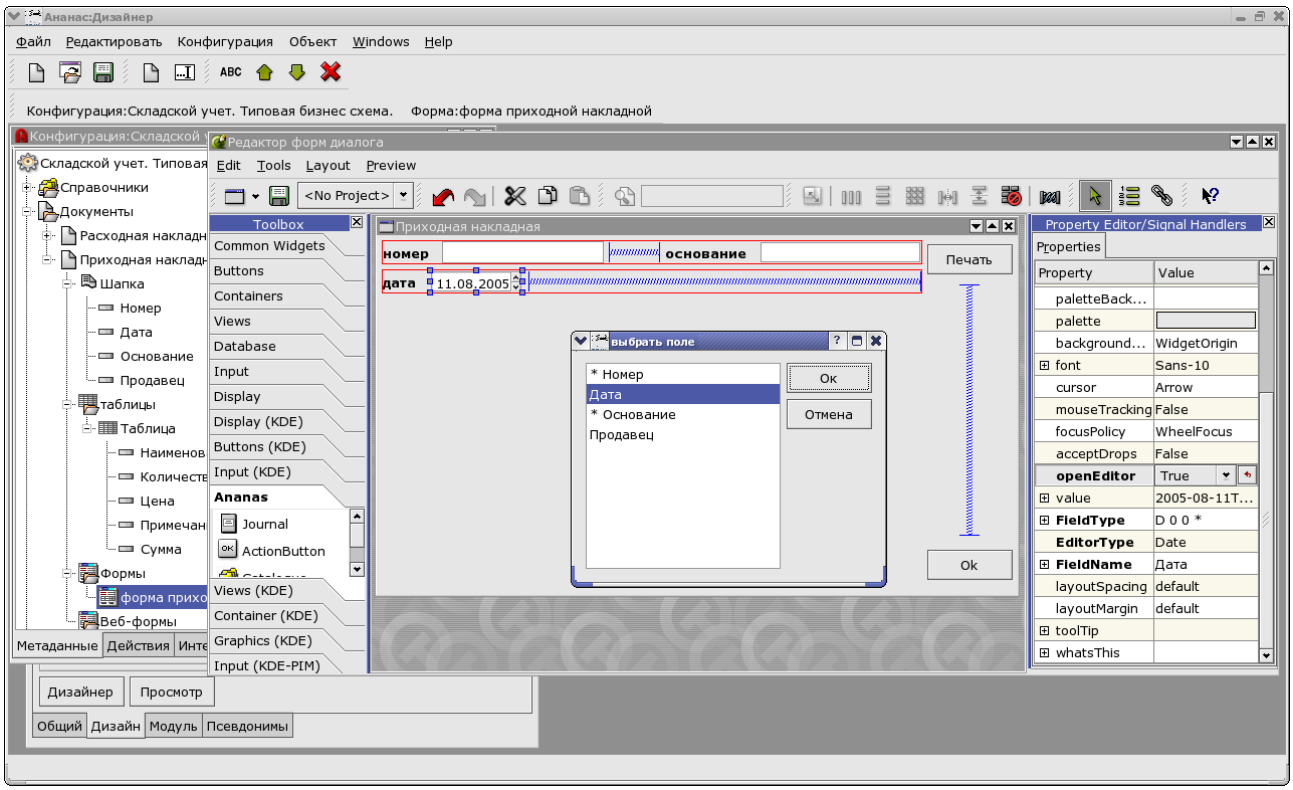


Иллюстрация 32: Размещаем поле ввода и связываем его с реквизитом «Дата» Приходной накладной в базе данных, который мы определили в метаданных ранее

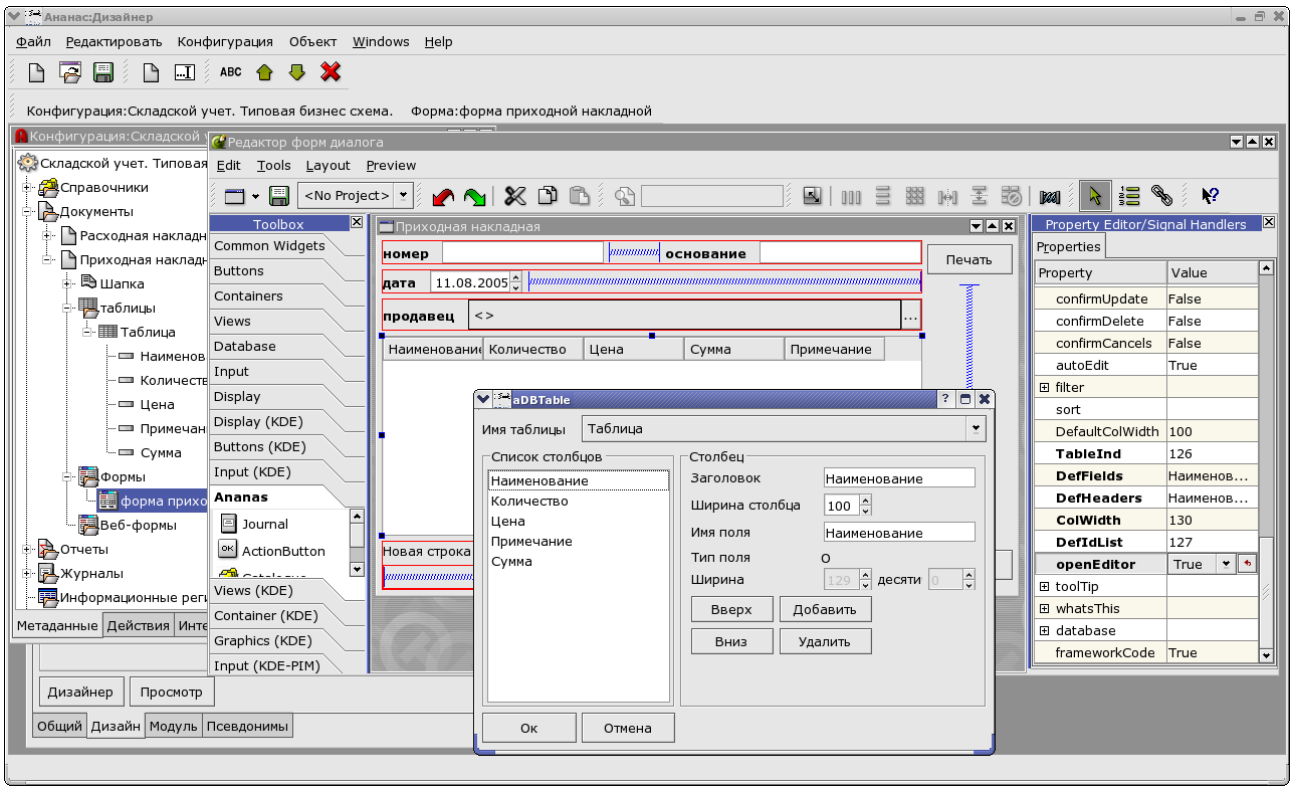
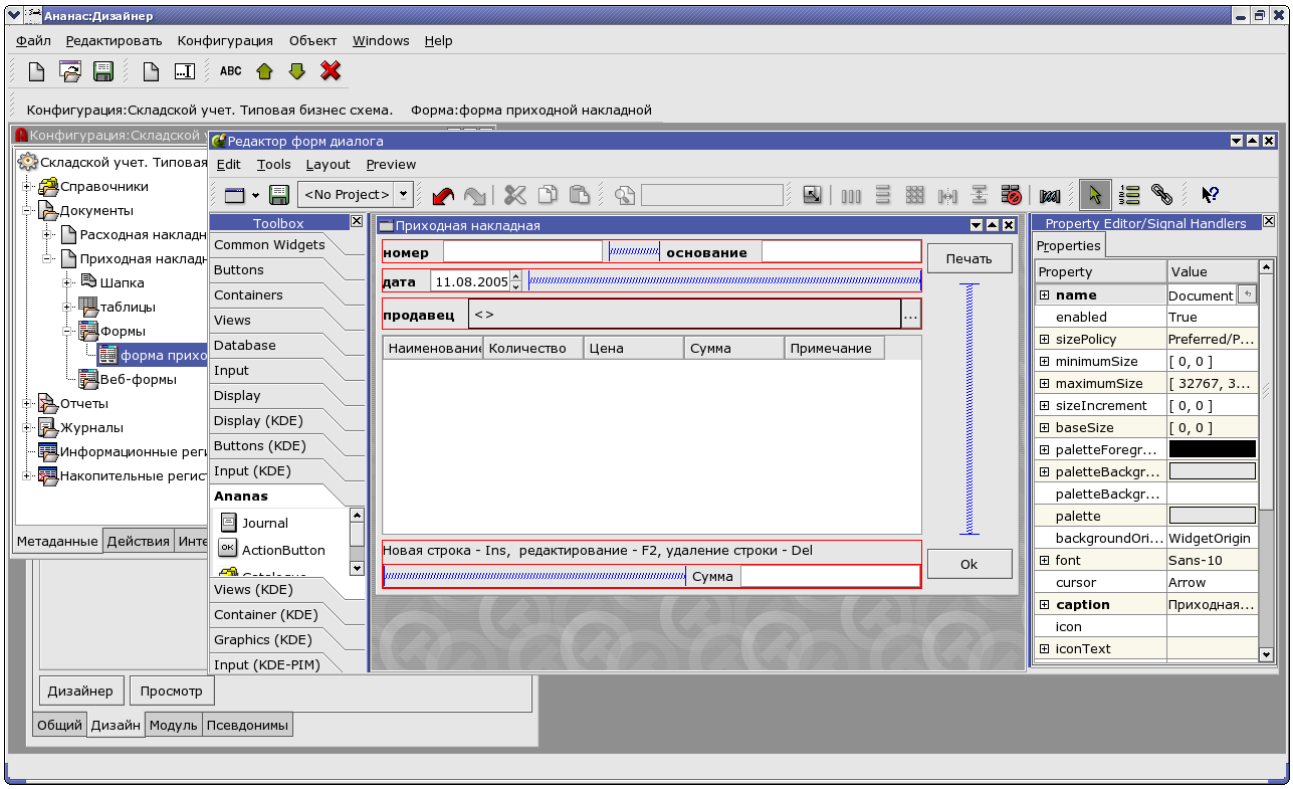


Иллюстрация 33: Размещаем таблицу ввода данных и связываем с ней табличную часть Приходной накладной, определенную ранее в метаданных



Печатная форма Приходн ой накладн ой

Ананас использует для подготовки печатных форм документов и отчетов возможности, предоставляемые пакетом офисных приложений OpenOffice. Точнее сказать, используется ровно одна возможность – самостоятельного формирования документов в формате OpenOffice, которой и пользуется Ананас. Формирование печатной формы осуществляет Ананас-Скрипт на основании имеющихся в его распоряжении данных и шаблона отчета. Шаблон отчета представляет собой документ OpenOffice Writer, с занесенными в него тегами специального формата. Процедура на Ананас-Скрипте подменяет тэги значениями, оставляя неизменным стилевое оформление, заданное для тэга. То есть стилевое оформление тэга становиться стилевым оформлением заменивших его данных. Иллюстрация 34 на странице 40 показывает шаблон приходной накладной. Каждый тэг в документе начинается с пары символов «<:» и заканчивается парой символов «:>», между которыми указано имя тэга. Имя тэга использует Ананас-Скрипт при формировании отчета.

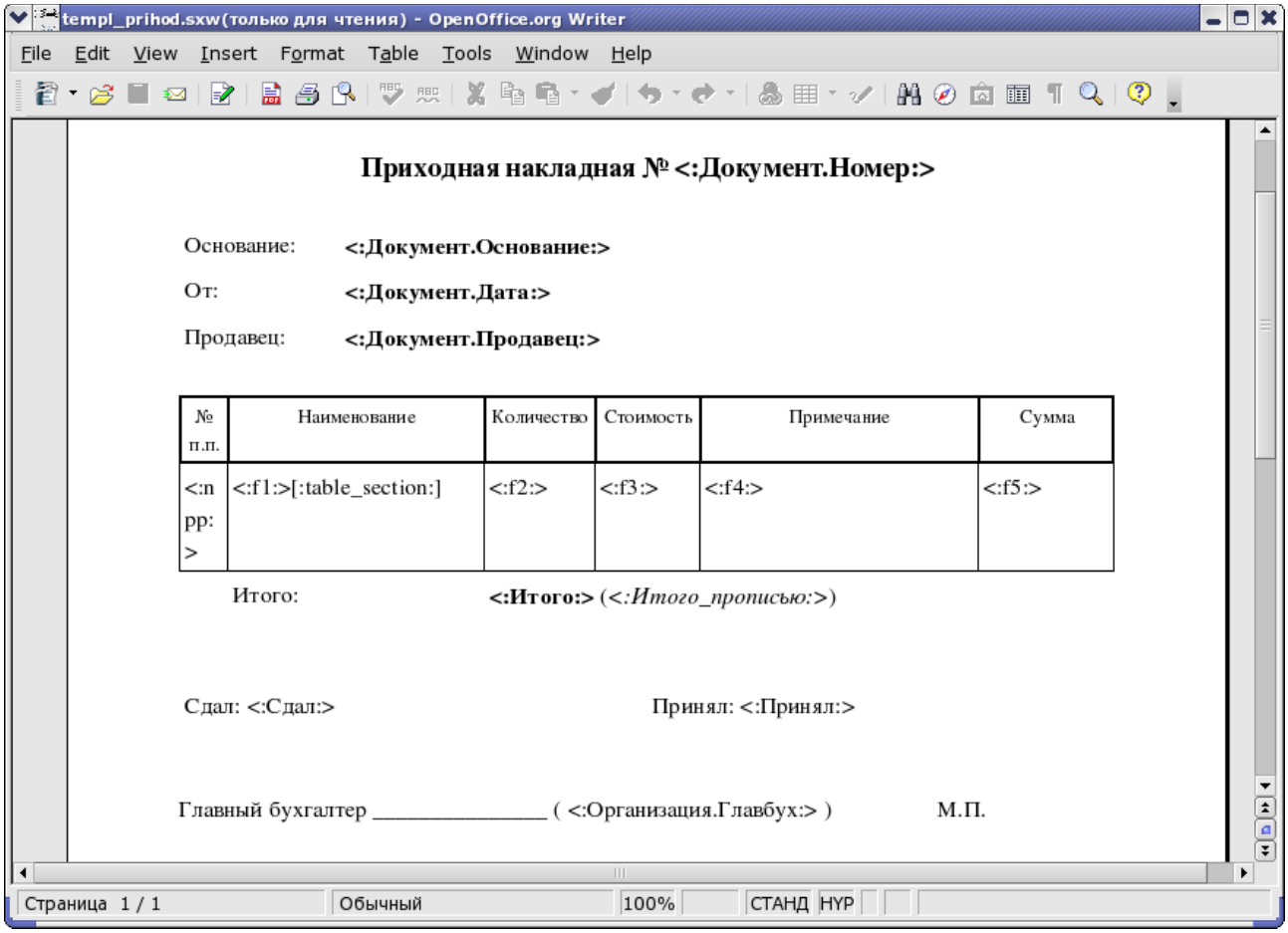


Иллюстрация 34: Вид шаблона печатной формы Приходной накладной

Приведем готовый код процедуры, осуществляющей формирование печатной формы Приходной накладной.

```
function on_button(button_name) // обработчик нажатия кнопки
{
    if(button_name!="print_button") return; // обрабатываем только нажатие кнопри Печать
    var p = new Report("Report1","1"); // создаем экземпляр отчета
    p.setTable("templ_prihod.sxw"); // связываем его с шаблоном
    // шаблоны ищутся в каталоге `Рабочий каталов'
    p.setValue("param",Value("Номер") ); // устанавливает значение для простого тега
    p.exec("Документ.Номер"); // применяет установленнок значение для тега с именем
<:Документ.Номер:>
    p.setValue("param",Value("Основание") );
    p.exec("Документ.Основание");
    p.setValue("param",Value("Продавец"));
    p.exec("Документ.Продавец");
    p.setValue("param",Value("Дата") );
    p.exec("Документ.Дата");
    var sum=0;
    var countRow =  TabCount("wDBTable1");   // количество строк таблицы
    var i;
    for ( i =countRow-1; i>=0; i--)
    {
                p.setValue("npp", String(i+1)); // устанавливаем значение табличного тега с именем
`npp'
                for ( j=1; j<=5; j++)
                {
                          p.setValue("f"+String(j), TabValue("wDBTable1",i,j-1));
                }
               p.exec("table_section"); // применяем установленные значения для строки
                                //содерщей табличный тег [:table_section:]
    }
    sum = Value("lineEdit1");
    p.setValue("param",sum);
    p.exec("Итого");
    p.setValue("param", Propis(sum)); // записываем сумму прописью
    p.exec("Итого_прописью");
    p.setValue("param","Сидоров С.С");
    p.exec("Сдал");
    p.setValue("param","Петров П.П");
    p.exec("Принял");
    p.setValue("param",getConstant("Константы","Главный бухгалтер")); // получаем имя главного
бухгалтера из справочника констант
    p.exec("Организация.Главбух");
    p.show(); // запускаем OpenOfficce
    p.close(); // удаляем все временные файлы
}
```

Этот код следует поместить в модуль экранной формы Приходной накладной, который демонстрирует

Проектируем Расходную накладную

Реквизиты Расходной накладной

Реквизиты табличной части Расходной накладной

Экранная форма Расходной накладной

Печатная форма Расходной накладной

Разработка справочников

Термин «справочник» в лексиконе разработчика бизнес схем соответствует таким понятиям как Номенклатура, Каталог товаров, Справочник товаров, Реестр контрагентов, Справочник складов и т.д. в лексиконе пользователя. Справочник служит для долговременного хранения информации, идентифицирующей объекты учета в форме простого списка или сгруппированного по разделам и подразделам иерархического представления (последнее часто называют деревом).

Ананас позволяет определить в бизнес схеме произвольное количество справочников, идентифицируемых по имени. Проектирование справочника, по сути, сводится к определению реквизитов элементов справочника, реквизитов групп справочника и построению трех (иногда двух) экранных форм: экранной формы работы со списком элементов и деревом групп (эта форма часто называется Редактор справочника), экранной формы редактирования реквизитов групп справочника (если справочник предусматривает работу с группами) и экранной формы редактирования реквизитов элементов справочника.

Определение реквизитов осуществляется в основном окне редактирования бизнес схемы. Для построения экранных диалогов используется специальный редактор диалогов. Редактор диалогов оперирует с реквизитами, которые были определены, поэтому построение экранных форм следует осуществлять после определения реквизитов.

Проектируем Справочник Контрагентов

Определение реквизитов

Практически в любом учете постоянно используется информация о контрагентах и товарах. Поэтому, рассмотрим их создание в первую очередь на следующем примере. Спроектируем справочник контрагентов с минимально необходимым набором реквизитов. Поняв, принципы определения справочника с небольшим количеством реквизитов вы без труда сможете добавить в него необходимые вам недостающие реквизиты.

Итак, определение справочника начинается с присвоения ему имени, которое в дальнейшем будет использовано в программировании бизнес логики в процедурах на языке программирования «Ананас.Скрипт».

В главном окне Дизайнера правой клавишей мыши щелкните по разделу «Справочники» и в появившемся меню выберите пунк «Новый».

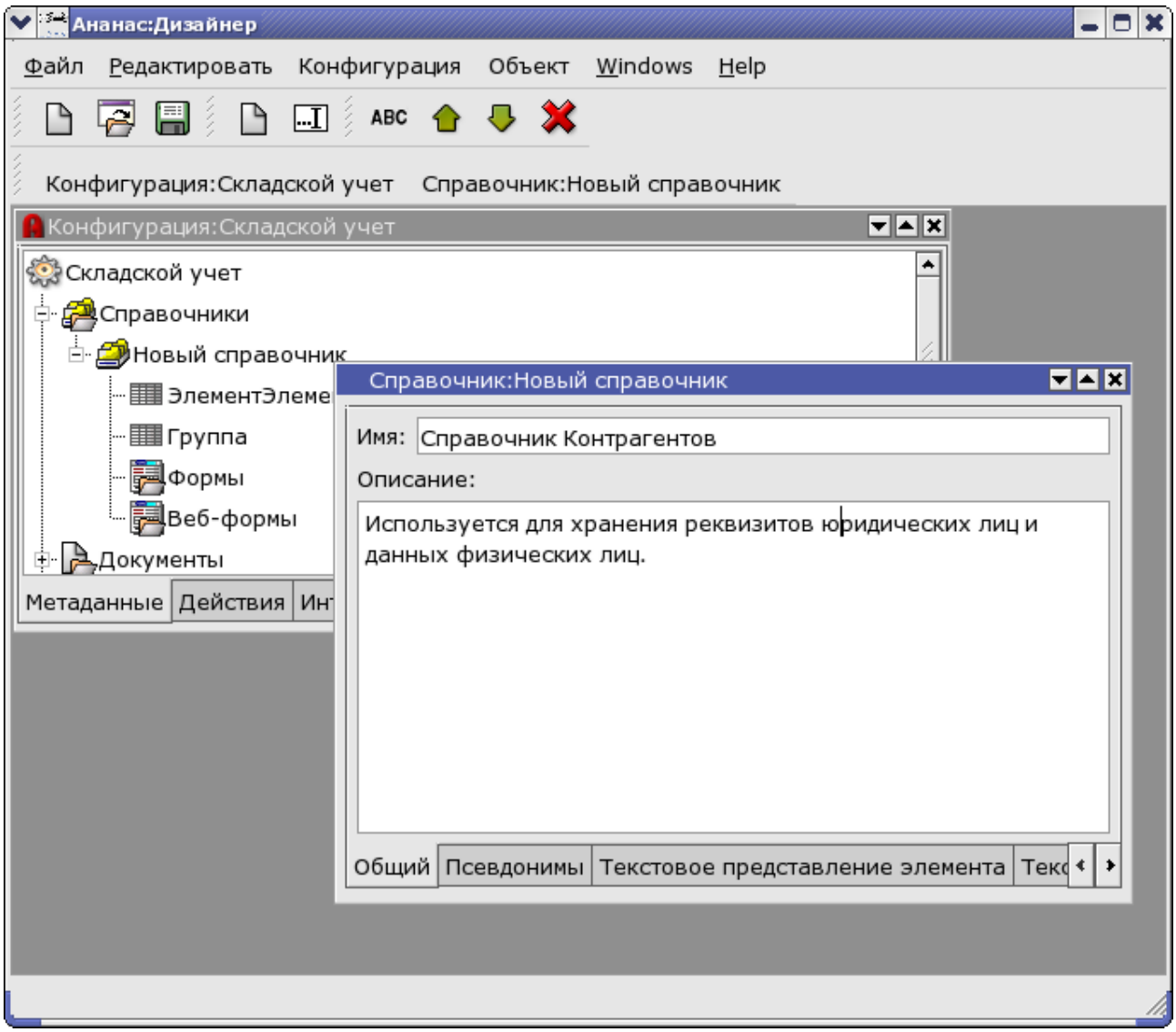


Иллюстрация 35: Задаем название нового справочника

Иллюстрация 35 на странице 43 показывает диалог, появляющийся в ответ на наше действие. Закладка «Общий» позволяет задать Имя справочника и заполнить поле «Описание». Больше ни чего в этом диалоге задавать не нужно на данном шаге. Следует его закрыть и перейти к определению реквизитов групп Справочника контрагентов. Разумеется, если группы в Справочнике контрагентов вам не нужны, то вы можете пропустить этот шаг безо всяких негативных для работоспособности вашей бизнес схемы последствий. Итак, как только вы закрыли диалог, где было задано имя справочника, в дереве бизнес схемы появилась папка с только что заданным названием, внутри которой видны подпапки Элемент, Группа, Формы. Для определения первого реквизита группы следует правой кнопкой мышки щелкнуть по папке Группа и выбрать пункт Новый появившегося меню. В открывшимся диалоге, как показывает Иллюстрация 36 на странице 36, необходимо задать такие свойства реквизита, как Наименование, Тип, Ширина. Прочие свойства задавать не обязательно, их назначение описано в Справочном руководстве по Дизайнеру. Закрыв диалог, следует повторить действия снова для определения второго, третьего и т.д. реквизитов группы. Определяйте столько реквизитов, сколько сочтете нужным. Однако, обычно группы содержат не больше одного-двух реквизитов для хранения Наименования и, например, Номера.

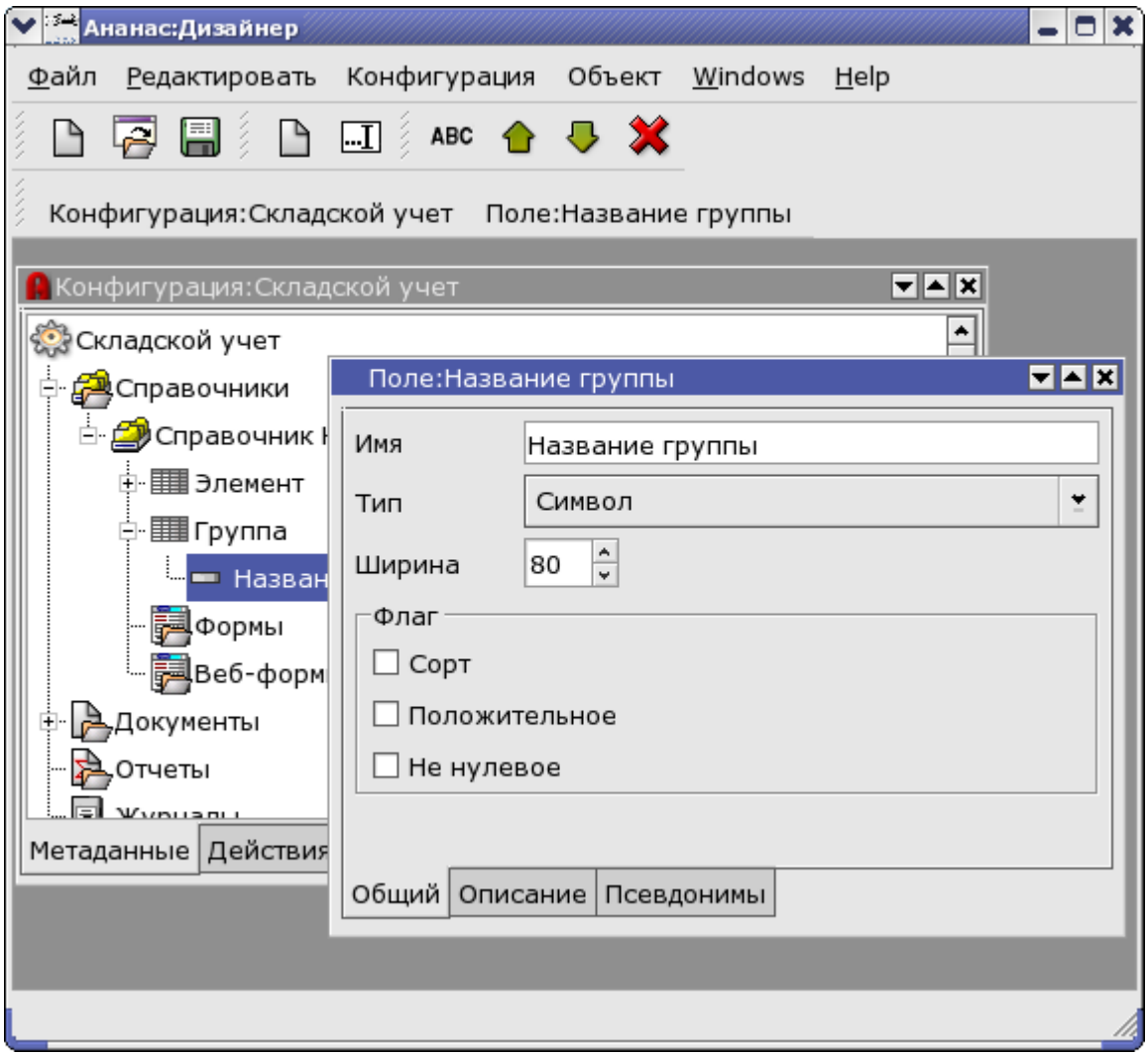


Иллюстрация 36: Определяем реквизит «Название группы» группы справочника Контрагентов

Основную содержательную нагрузку в справочниках, как правило, несут элементы, которые обычно содержат не один десяток реквизитов, а то и больше. Поэтому, именно определение состава и свойств реквизитов элементов справочника отнимает наибольшее время при проектировании справочника, если не считать создание экранных форм для редактирования значений реквизитов этих самых элементов. Порядок определения реквизитов элемента отличается от порядка определения реквизитов группы только тем, что мышиный щелчок необходимо направлять на папку Элемент. В том что диалог задания свойств реквизита элемента выглядит так же как и для задания свойств реквизита группы позволяет убедиться Иллюстрация 37 на странице 45.

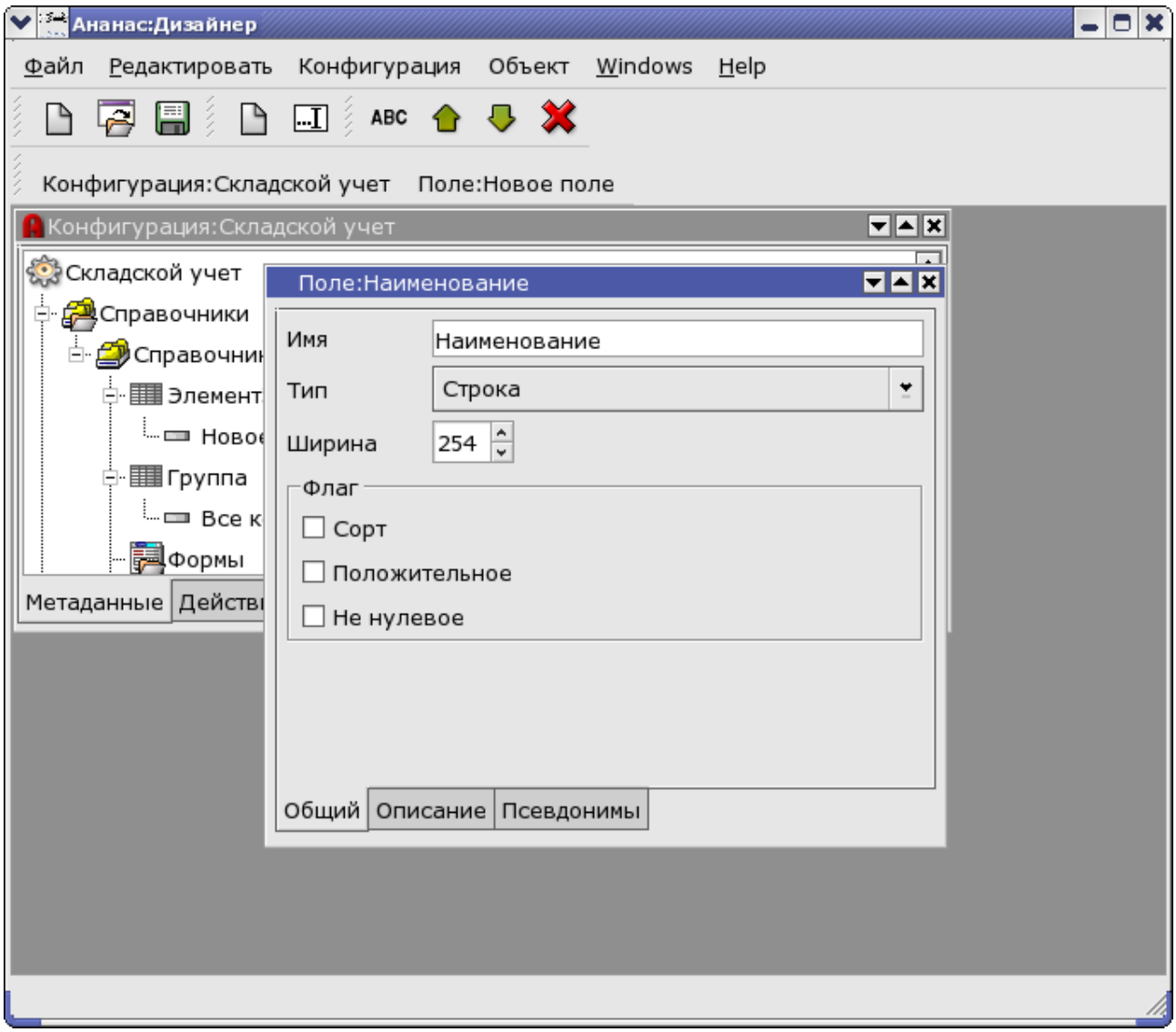


Иллюстрация 37: Определяем реквизит Наименование элемента Справочника контрагентов

Для нашего примера можно определить список реквизитов элементов Справочника контрагентов в следующем составе:

– Наименование;
– Почтовый адрес;
– ИНН;
– ОКОНХ;
– ОКПО;
– Телефон;
– Факс;
– E-Mail;
– Банк;
– БИК;
– Кор.Счет;
– Расч.Счет;
– Примечание.

После определения реквизитов Групп и Элементов справочника, работу по формированию структуры можно считать законченной.

Определение главного реквизита

Теперь настало время вернуться в диалог, с которого началось создание Справочника контрагентов и определить какой из определенных выше реквизитов будет отображать свое значение в различных экранных диалогах, содержащих поля, значения которых задаются путем выбора из Справочника контрагентов. Мы же не можем в одном поле такого диалога отображать значения стразу нескольких реквизитов элемента справочника! Поэтому мы должны выбрать ровно один реквизит. Разумеется в нашем случае выбор очевиден – это реквизит Наименование. Но если бы в нашем справочнике присутствовал реквизит Краткое наименование, выбор был бы уже не столь очевиден, согласитесь.

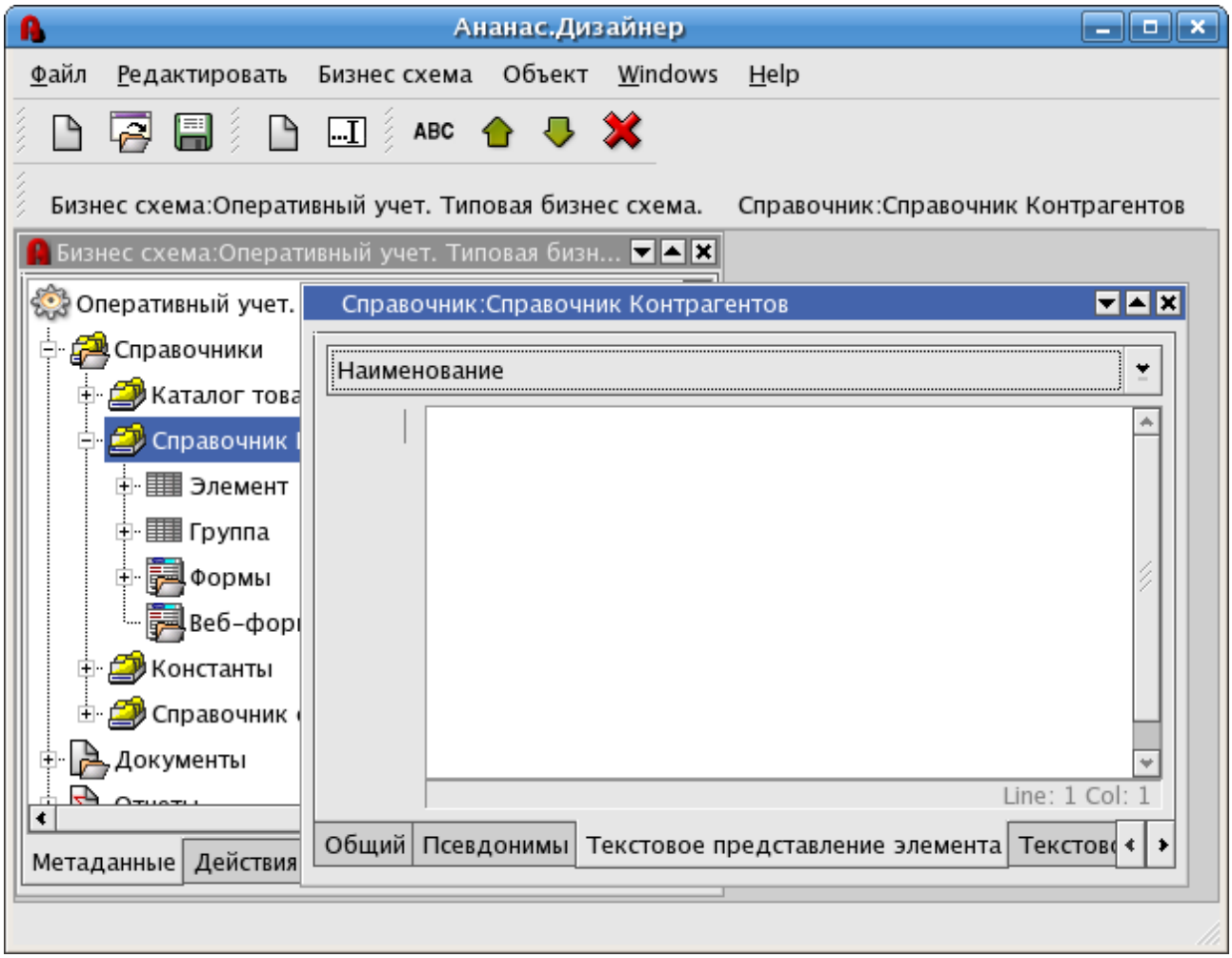


Иллюстрация 38: Определяем текстовое представление справочника

Экранные формы справочника

Для справочников в отличие от документов существует возможность выбора между использованием готовой (встроенной) главной экранной формы и созданием главной экранной формы в редакторе диалогов. Экранные формы для редактирования элементов справочника и для редактирования групп справочника необходимо создать с помощью редактора диалогов.

В настоящий момент времени рекомендуется использовать встроенный редактор справочника (главную экранную форму справочника) всегда, когда он вас устраивает, и создавать свой редактор справочника, когда возможностей встроенного редактора вам не достаточно.

Для использования встроенного редактора достаточно не определять свой собственный редактор. Так мы и поступим для Справочника контрагентов, перейдя сразу к определению экранной формы редактирования групп.

Пример создания собственного редактора справочника можно увидеть в разделе, посвященном Справочнику складов.

Редактор групп справочника

Если реквизиты групп справочника заданы, как описано выше, можно смело переходить к построению экранной формы для редактирования их значений. Для этого следует щелкнуть правой клавишей мыши по подпапке Формы папки Справочник контрагентов и выбрать пункт Новый появившегося меню.

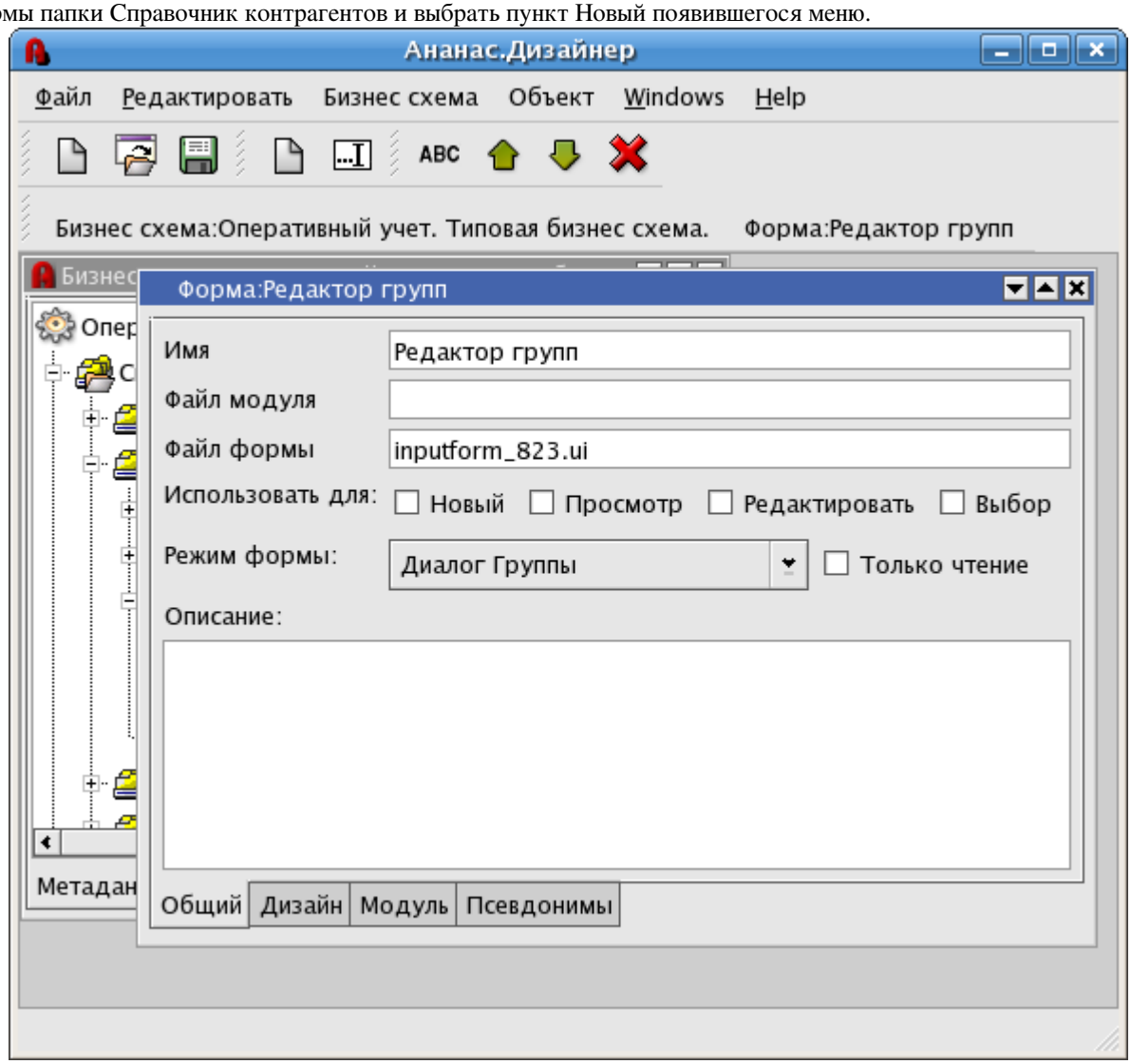


Иллюстрация 39: Настройка экранной формы редактора групп справочника

В открывшемся диалоге важно определить ровно один реквизит «Режим формы». Именно он отвечает за то, что форма будет использоваться в качестве редактора групп в справочнике контрагентов всякий раз, когда пользователь будет добавлять новую группу или редактировать уже имеющуюся. Для собственного удобства целесообразно заполнить поле «Имя» каким-нибудь осмысленным текстом, например, как демонстрирует Иллюстрация 39 на странице 47, вписать «Редактор групп», что бы отличать его от редактора элементов при последующей работе с бизнес схемой в Дизайнере.

Следующий очевидный шаг – вызов редактора диалогов для размещения в окне реквизитов по месту, где их ожидает увидеть пользователь. Для этого необходимо на вкладке «Дизайн» нажать кнопку «Редактор диалогов».

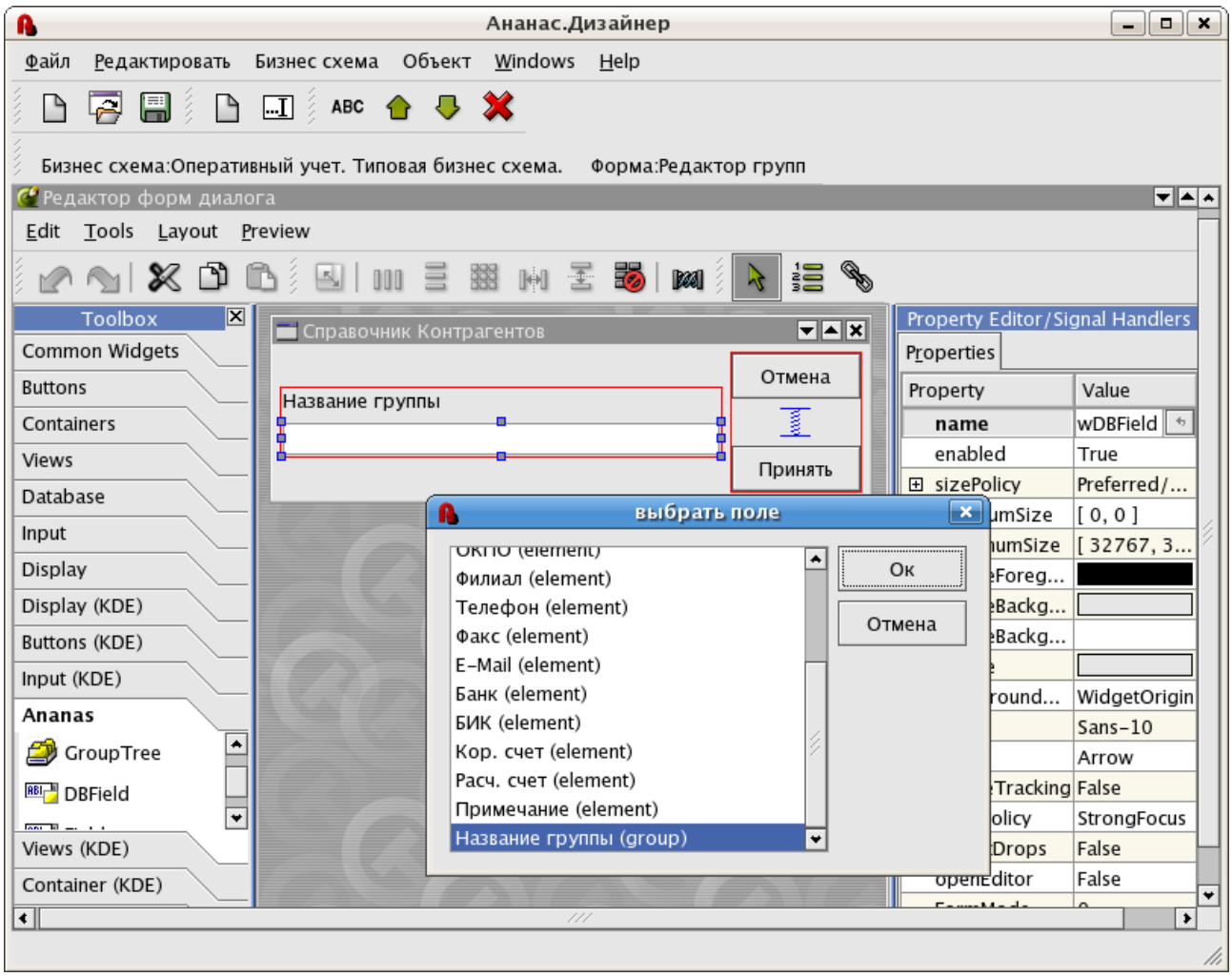


Иллюстрация 40: Экранная форма редактора групп Справочника контрагентов с открытым редактором свойств поля ввода

Иллюстрация 40 На странице 48 демонстрирует окончательный вид экранной формы редактора групп Справочника контрагентов. Предполагается, что группы в Справочнике контрагентов имею единственный реквизит – Название.

Порядок построения этой экранной формы аналогичен применяемому при создании экранной формы документов, описанному выше в разделах посвященных экранным формам документа. В качестве единственного поля ввода применен виджет DBField с закладки Ananas визуальных компонентов, отображаемых в левой части редактора диалогов. Связывание поля ввода DBField формы с реквизитом «Наименование» группы Справочника контрагентов выполнено путем выбора соответствующего элемента списка, вызванного двойным щелчком по полю формы.

Кнопки Принять и Отмена являются виджетами ActionButton, взятыми с той же закладки Ananas. В свойствах кнопки Принять отмечены чекбоксы «Простое действие», «Обновить», «Закрыть». В свойствах кнопки Отмена только «Простое действие» и «Закрыть».

Редактор элементов справочника

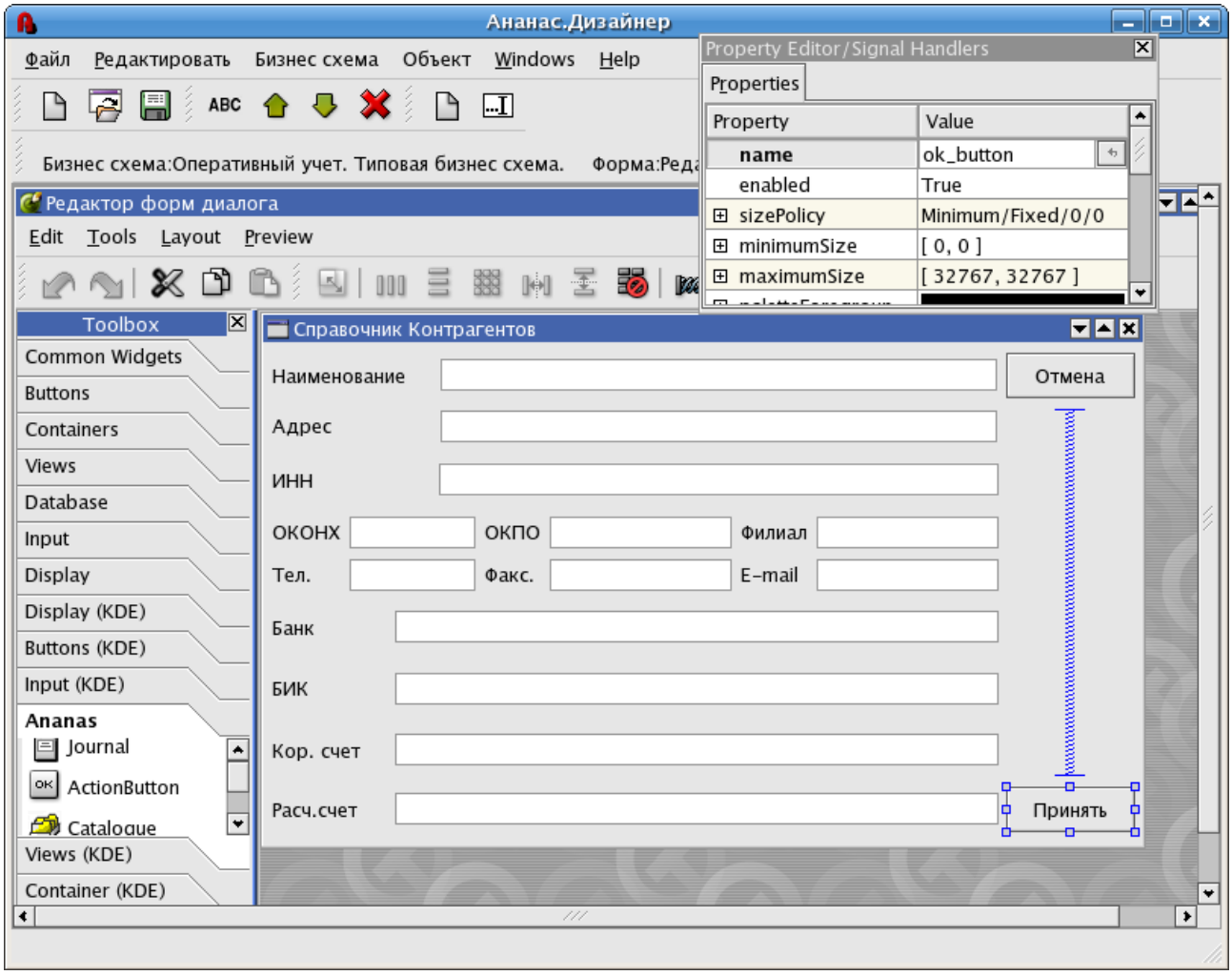


Иллюстрация 41: Экранная форма редактора элементов Справочника контрагентов

Проектируем Справочник товаров

Разработка журналов

Разработка отчетов

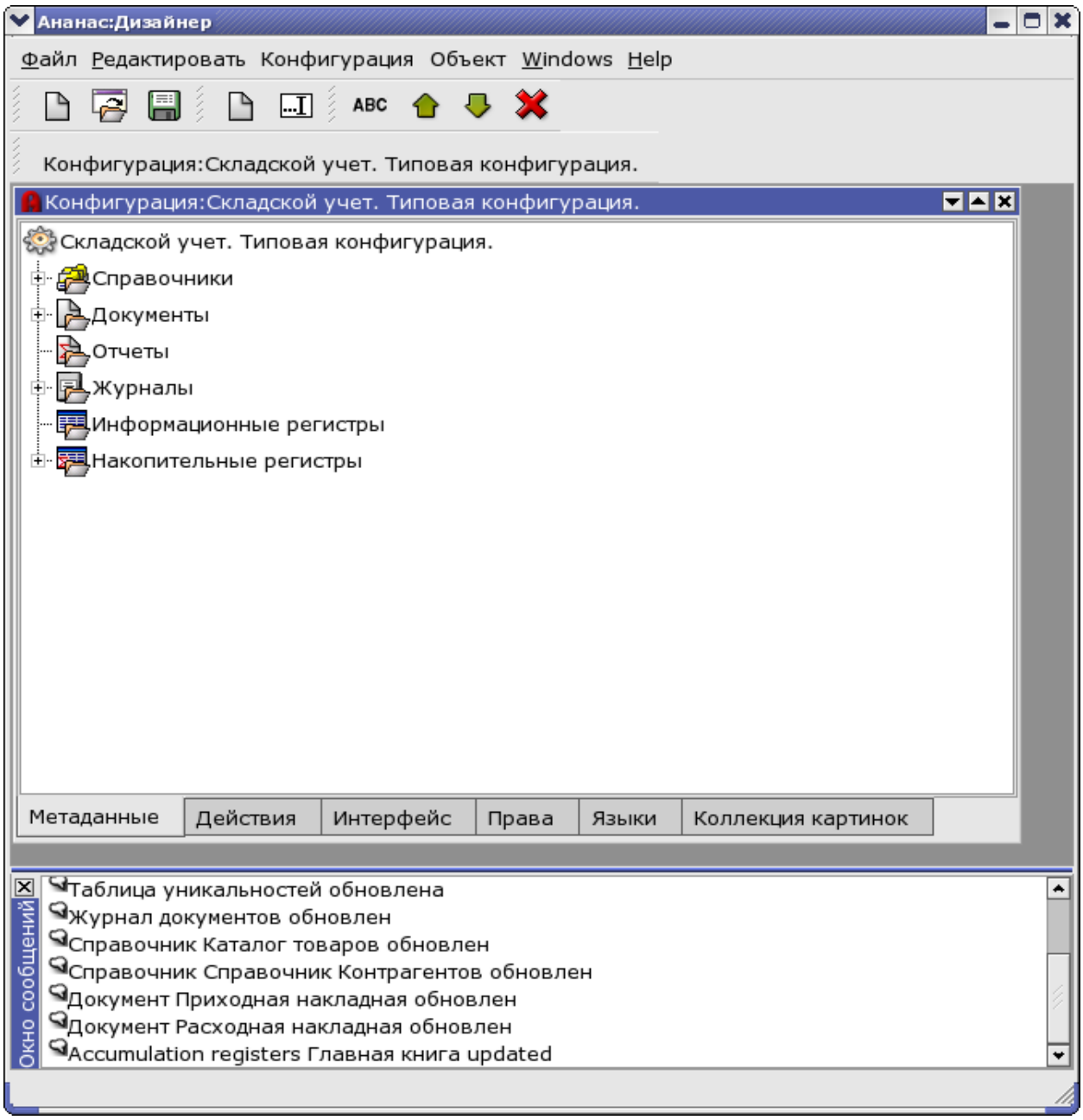


Иллюстрация 42. Сохранение на диск изменений в бизнес схеме. В окне сообщений - информация об успешном обновлении структур базы данных на SQL сервере

Исполнение бизнес схемы в Ананасе

Примеры кода Ананас-Скрипта

```
/**
*       Тестирование возможностей и производительности
*       Накопительного регистра
*       Код позволяет увидеть как происходит работа с такими бизнес объектами как Документ,
Справочник товаров, регистр
*/
function aaregistertest() {


    const maxlines = 15;
```

```
    var arName="Главная книга";
    var docName="Накладная";
    var catName="Каталог пром товаров";

    var ans = MessageBox.warning( "Начать тестирование?", MessageBox.Yes, MessageBox.No );
    if (ans == MessageBox.No) {
        print("завершили");
        return;
    }
    var ar = new ARegister(arName);
    var doc1 = new Document(docName);
    var catGoods = new Catalogue(catName);
    if (!catGoods) {  print("Ошибка доступа к каталогу "+catName); return;    }
    if (catGoods.GroupNew()) {       print("Ошибка при создании новой группы каталога "+catName);
return;    };
    if (catGoods.GroupSetValue("Название группы","Продукты питания")){    print("Ошибка задания
значения группе каталога"); return;    };
    if (catGoods.GroupSetValue("Код группы","ПрП")){       print("Ошибка задания значения Код
группы"); return;    };
    if (catGoods.Update()){  print("Ошибка обновления значений группы каталога"); return;    }
  //  print("Обновили значения группы каталога"); return;
    if (catGoods.New()) {print("Ошибка создания элемента каталога"); return;   }
    if (catGoods.SetValue("Наименование","Коньяк Дагестанский ***")) {print("Ошибка Наименование");
return;   }
    if (catGoods.SetValue("Производитель","р-ка Дагестан")) {print("Ошибка Производитель"); return;
}
    if (catGoods.SetValue("Цена",250)) {print("Ошибка Цена"); return;   }
    if ((rc=catGoods.Update())) {print("Ошибка обновления значений элемента каталога "+rc); return;
}



for (i=1; i< 2; i++) {
    if (doc1.New()) { print("Ошибка созания документа"); return; }
    doc1.SetValue("Номер","ПН/"+global.Date()+"/"+i);
//print("global.Date="+global.Date());
    doc1.SetValue("Дата",global.Date());
    doc1.SetValue("Основание","На основании договора N 23"+i+"/А от "+global.Date());
    doc1.SetValue("Создан на основании",doc1.Uid());

    if (doc1.TableNewLine("Таблица")) {     printf("Ошибка добавления строки документа");
return; }

    doc1.TableSetValue("Таблица","Наименование",catGoods.Uid());
    doc1.TableSetValue("Таблица","Количество",100);
    doc1.TableSetValue("Таблица","Цена",50);
    doc1.TableSetValue("Таблица","Примечание","без примечаний "+i);

//    print("Количество="+doc1.TableValue("Таблица","Количество"));
//    print("Цена="+doc1.TableValue("Таблица","Цена"));
```

```
    if (doc1.Update()) { print("Ошибка обновления документа в базе"); return; }

//    print("Номер="+ doc1.Value("Номер"));
//    print("Дата="+ doc1.Value("Дата"));
    if (ar.SetDocument(doc1)) { print("Не прошла привязка документа");   return; }
    if (ar.New()) {   print("Ошибка при добавлении в регистр"); return; }
   ar.SetValue("Счет по дебету","41.1");
   ar.SetValue("Счет по кредиту","60");
   ar.SetValue("Сумма",6340.14);
   ar.SetValue("Субконто Товары",catGoods.Uid());
   ar.SetValue("Примечание","Примечание в регистре");
   if (ar.Update()) { print("Ошибка обновления регистра"); return;}
   print("Добавили документ в Регистр. i="+i);
}



}
```

Динамические пользовательские диалоги. Программирование и обработка событий формы диалога

Независимо от типа бизнес объекта (документ, справочник, элемент справочника), все экранные формы диалогов работают одинаково. Экранные формы создаются для редактирования атрибутов бизнес объектов. За редактирование атрибутов отвечают специально созданные виджеты Ананаса. В обычной ситуации создание экранной формы для редактирования реквизитов бизнес объектов Ананаса сводится к размещению этих виджетов внутри формы с помощью редактора диалогов Ананаса и привязки каждого виждета к тому или иному атрибуту бизнес объекта.

В тех случаях, когда от экранного диалога требуется добиться какого-либо нестандартного поведения, когда недостаточно иметь простую возможность редактирования реквизитов бизнес объекта, следует использовать Ананас.Скрипт для обработки событий экранной формы и программирования требуемого поведения ее виджетов.

Так например, если требуется сделать доступность какого-либо виджета зависимой от определенных условий, то можно в модуле экранной формы диалога определить функцию on_formstart() и менять в ней состояние доступности виджета. Ананас будет искать в модуле экранной формы функцию с таким именем и вызовет ее перед тем как экранная форма станет доступна пользователю.

Функция on_formstart() будет вызвана ровно один раз за все время существования диалога на экране.

Пример показывает как можно задать состояние двум виджетам с именами «btnCancel» и «btnOk»

```
function on_formstart() {
       Widget("btnCancel").setEnabled( true );
       Widget("btnOk").setEnabled( false );
}
```

Другой вариант записи тех же действий

```
function on_formstart() {
        var w = Widget("btnCancel");
        w.setEnabled( true );
        w = Widget("btnOk");
        w.setEnabled( false );
}
```

Как мы видим в примере, доступ к виджету экранной формы осуществляется с помощью встроенной функции Widget() Ананаса, в качестве параметра которой передается имя виджета, назначенное ему при проектировании экранной формы в редакторе диалогов Ананаса.

connect – задание собственных обработчиков событий

В экранных формах Ананаса есть набор предопределенных событий с зарезервированными именами функций обработчиков, такими как on_formstart(). Кроме того есть возможность назначать с помощью вызова функции connect() обработчиков событий, генерируемых виджетами формы. Это очень гибкий и мощный инструмент.

Например напишем обработчика событий инициируемых виджетом CheckBox. Виджет инициирует события, как правило, когда пользователь меняет его состояние или начинает работать с виджетом, установив на него фокус. Обработчик будет менять состояние доступности кнопки btnOk в зависимости от состояния чек бокса. Так как в Ананасе нет зарезервированного названия функции для обработки события типа «Смена состояния чекбокса», мы воспользуемся встроенной функцией connect, позволяющей связать событие с функцией, выполняющей его обработку.

```
function on_formstart(){
        var w = Widget("checkBox1")
        connect(w, "clicked()", hideShow);
}
function hideShow() {

}
```

Здесь мы видим, что функции connect передается три параметра. Первый – ссылка на виджет, полученная при помощи функции Widget и записанная в переменную w; второй – имя обрабатываемого события - «clicked()», имя функции обработчика события «clicked()»

Привязка обработчика к событию происходит внутри функции on_formstart(). Это дает гарантию, что когда форма станет доступна пользователю, обработчик уже будет привязан к событию.

Осталось запрограммировать функцию hideShow, которая будет менять состояние доступности кнопки btnOk на противоположное.

```
function hideShow() {
        var Ok = Widget("btnOk");
        Ok.setEnabled( !Ok.enabled );
}
```

Функция setEnabled() делает виджет доступным пользователю, если в параметре было передано значение true и наоборот, делает виджет недоступным, если было передано значение false.

В нашем примере мы передаем значение противоположное текущему состоянию виджета. Текущее состояние виджета мы узнаем, обратившись к публичному свойству enabled этого виджета.

Библиотека экранных виджетов QT, используемая в Ананасе, располагает большим количеством виджетов и у каждого виджета есть свой набор генерируемых им событий. Справку по публичным методам и свойствам виджетов и их событиям смотрите в документации QT, для чтения которой существует удобная программа assistant.

Список иллюстраций

Иллюстрация 1. Диалог выбора базы данных. Список выбора пуст....5
Иллюстрация 2. Диалог выбора базы данных. Создана группа "В разработке"....6
Иллюстрация 3. Диалог задания параметров новой базы данных. 7
Иллюстрация 4. Диалог задания параметров базы данных. Значения заданы....8
Иллюстрация 5. Диалог выбора базы данных. В списке одна группа с одной базой данных в ней....9
Иллюстрация 6. Вид Ананас.Дизайнера сразу после запуска. Для доступа к бизнес схеме ее следует открыть....10
Иллюстрация 7. Новая, пустая бизнес схема открыта в Ананас.Дизайнере....11
Иллюстрация 8: Сохранение новой пустой бизнес схемы....12
Иллюстрация 9: Сообщения об успешном сохранении пустой бизнес схемы....13
Иллюстрация 10: Вызов диалога редактирования параметров бизнес схемы....14
Иллюстрация 11: Диалог редактирования параметров бизнес схемы....15
Иллюстрация 12: Редактор меню доступен через закладку Интерфейс....16
Иллюстрация 13: Вызов диалога создания нового документа....18
Иллюстрация 14: Открылся диалог создания нового документа....19
Иллюстрация 15: Задаем название нового документа....20
Иллюстрация 16: Вызов диалога редактирования нового реквизита шапки документа....21
Иллюстрация 17: Открыт диалог редактирования нового реквизита шапки документа....22
Иллюстрация 18: Определение реквизита Номер шапки документа....23
Иллюстрация 19: Определение реквизита Дата шапки Приходной накладной....24
Иллюстрация 20: Определяем реквизит Основание шапки Приходной накладной....25
Иллюстрация 21: Определяем реквизит Продавец шапки Приходной накладной....26
Иллюстрация 22: Реквизит Продавец имеет сложный тип данных....27
Иллюстрация 23: Реквизит Наименование товара табличной части Приходной накладной....28
Иллюстрация 24: Диалог редактирования свойств экранной формы Приходной накладной....29
Иллюстрация 25: . Модуль экранной формы Приходной накладной....32
Иллюстрация 26: Открыта закладка для вызова редактора диалогов. Вызов производиться по нажатию кнопки Designer....33
Иллюстрация 27: Открыт редактор диалогов для создания диалога экранной формы Приходной накладной....34
Иллюстрация 28: Задаем действие, выполняемое по нажатию кнопки Ок....34
Иллюстрация 29: Задаем положение поля ввода и связываем его с полем "Номер" базы данных, которое мы определили ранее в метаданных 35
Иллюстрация 30: Размещаем поле ввода и связываем его с реквизитом «Основание» Приходной накладной в базе данных, который мы определили в метаданных ранее....35
Иллюстрация 31: Размещаем поле ввода и связываем его с реквизитом «Дата» Приходной накладной в базе данных, который мы определили в метаданных ранее....36
Иллюстрация 32: Размещаем таблицу ввода данных и связываем с ней табличную часть Приходной накладной, определенную ранее в метаданных....36
Иллюстрация 33: Вид шаблона печатной формы Приходной накладной....38
Иллюстрация 34: Задаем название нового справочника....40
Иллюстрация 35: Определяем реквизит «Название группы» группы справочника Контрагентов....41
Иллюстрация 36: Определяем реквизит Наименование....42
Иллюстрация 37: Определяем реквизит Почтовый адрес....43
Иллюстрация 38: Определяем способ хранения банковских реквизитов....44
Иллюстрация 39. Сохранение на диск изменений в бизнес схеме. В окне сообщений - информация об успешном обновлении структур базы данных на SQL сервере....45