

Константы, основы клиент-серверного программирования, общие реквизиты

Содержит описание работы с константами, обзор и демонстрацию основ клиент-серверного программирования в 1С:Предприятие 8.3, а так же – пример использования общих реквизитов.

Константы

Константы в 1С:Предприятие используются для хранения информации, которая либо не меняется никогда, либо меняется – но очень редко. Константы содержатся в ветви дерева конфигурации **Константы**. Создадим новую константу (рис. 2.1.), заполним ее параметры следующим образом:

Имя: ТекстСообщения

Тип: Строка

Длина: 50

Включим константу в состав подсистемы **УчетРаботыМастеров**. Предполагается, что данная константа будет использоваться для показа сообщения пользователям, входящим в систему.

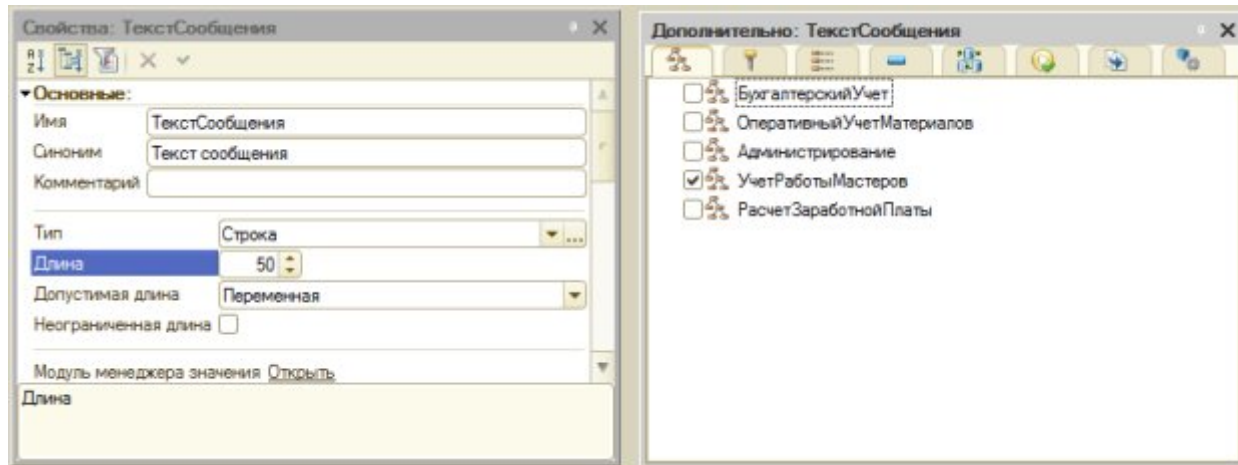


Рис. 2.1. Настройка параметров новой константы

Посмотрим, как включение константы в подсистему **УчетРаботыМастеров**, отразится на интерфейсе нашего приложения в режиме 1С:Предприятие. Видно, рис. 2.2., что в разделе **Учет работы мастеров**, под панелью разделов, появилась еще одна панель. Она называется **панелью действий**. В панель действий автоматически включаются команды, разбитые на группы – **Сервис**, **Создать**, **Отчеты**. Группы в панели действий можно создавать и самостоятельно. В нашем случае в панели действий видна группа **Сервис**, содержащая команду для работы с только что созданной константой.

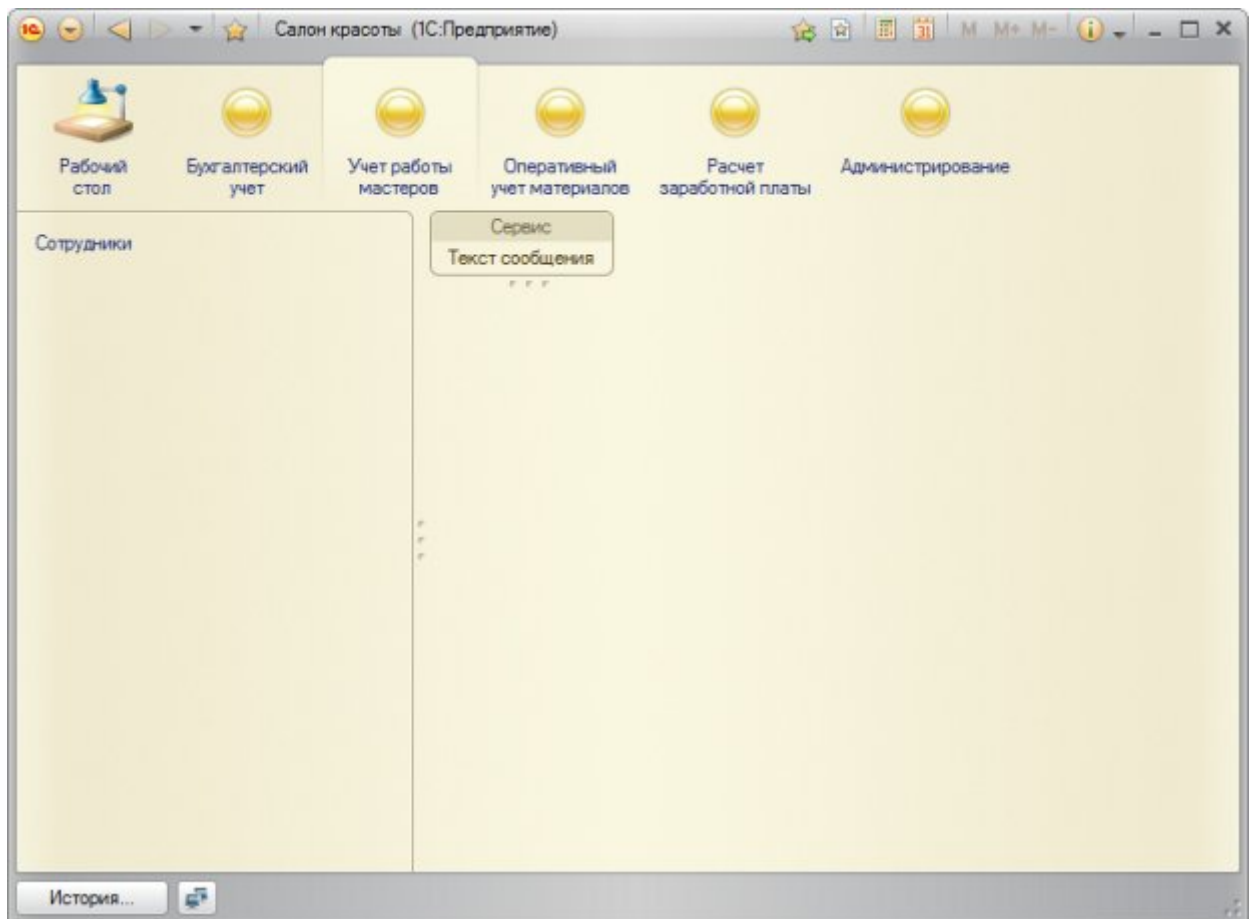


Рис. 2.2. Константа в панели действий в разделе Учет работы мастеров

В левой части окна программы можно видеть еще одну панель – она называется **панелью навигации**. Сейчас она отображает ссылку для доступа к справочнику **Сотрудники**, который мы создавали в предыдущей лекции. Свободная часть окна – это рабочая область, в которой, например, открываются списки справочников.

Щелкнем по команде **Текст сообщения** в панели действий. Отобразится окно, которое позволяет нам редактировать константу **ТекстСообщения**. Введем в поле **Текст сообщения** строку "Здравствуйте, уважаемый пользователь!", рис. 2.3. и нажмем на кнопку **Записать и закрыть**.

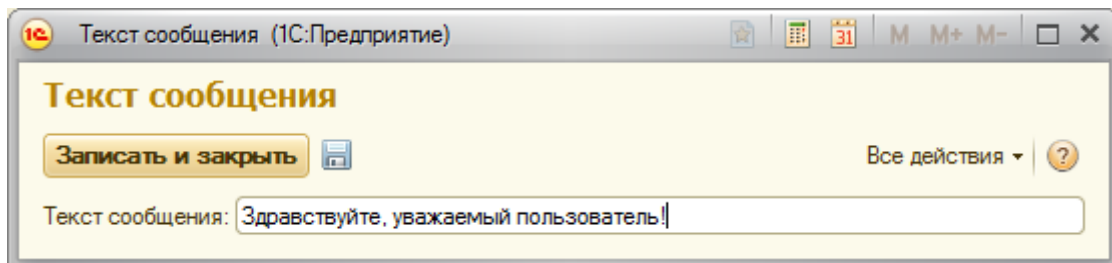


Рис. 2.3. Форма редактирования константы Текст сообщения

Если мы не хотим сохранять внесенные изменения, можно просто закрыть окно с помощью стандартной кнопки **Заккрыть**, для записи изменений без закрытия формы служит кнопка **Записать объект**.

Для того, чтобы воспользоваться дополнительными возможностями по работе с формой, можно использовать меню **Все действия**, рис. 2.4.

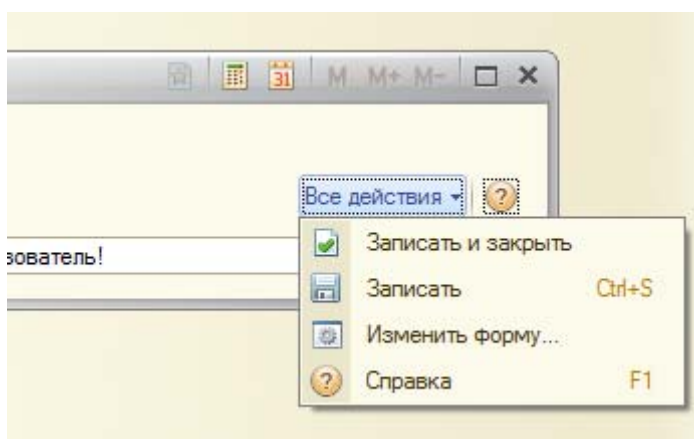


Рис. 2.4. Меню Все действия

Отчасти оно дублирует кнопки, имеющиеся на форме, в нем так же имеется одна специфичная для платформы 1С:Предприятие 8.3. команда. А именно, речь идет о команде **Изменить форму**.

Форма, которую мы видим, сформирована автоматически. Однако, в режиме 1С:Предприятие мы можем вносить в нее некоторые изменения. Выполним команду **Изменить форму**, появится окно Настройка формы, рис. 2.5.

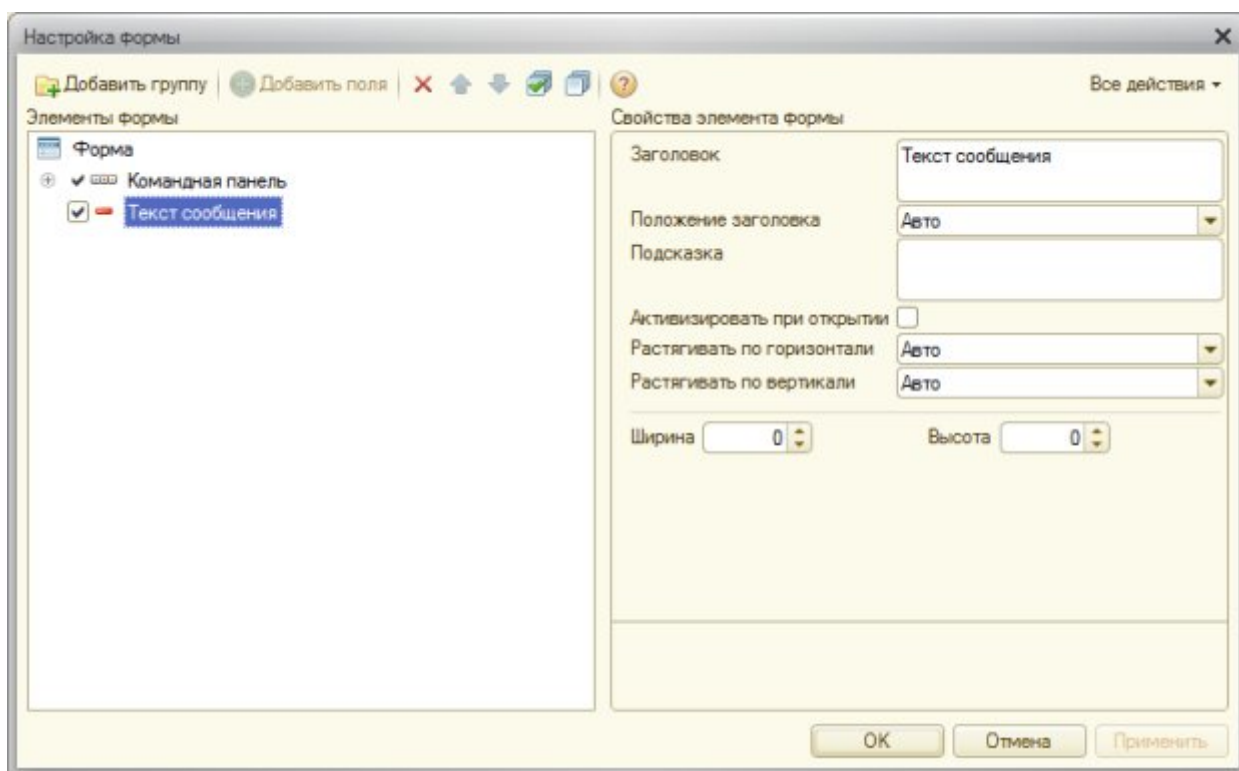


Рис. 2.5. Окно Настройка формы

Нужно учитывать, что пользователь сможет настраивать внешний вид форм в том случае, если для него установлено право **Сохранение данных пользователя**. Это право можно

настраивать, как и другие права, в роли пользователя, рис. 2.6. В нашем случае оно установлено.

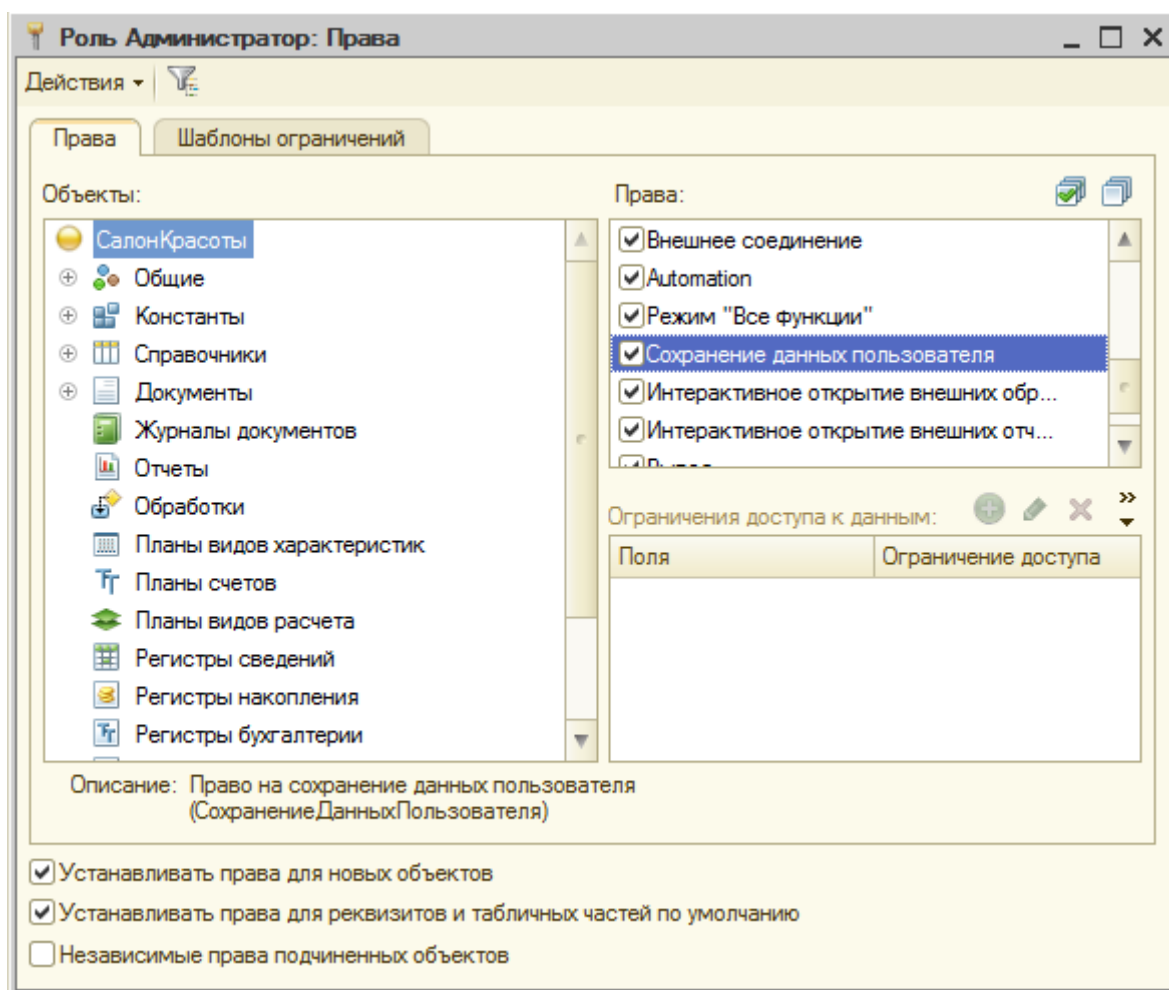


Рис. 2.6. Право Сохранение данных пользователя

В нашем случае, рис. 2.5., в группе **Элементы формы** выделен элемент **Текст сообщения**, в группе **Свойства элемента формы** мы можем настраивать его свойства. Изменим свойство **Заголовок**, вместо "Текст сообщения" введем "Текст сообщения для пользователей", в итоге форма будет выглядеть так, как показано на рис. 2.7.

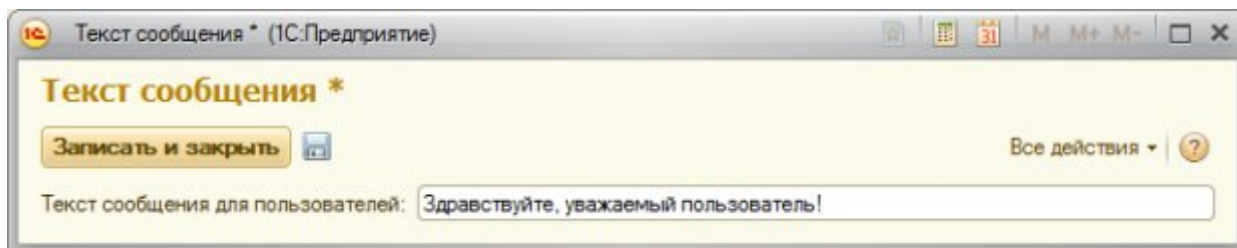


Рис. 2.7. Отредактированный заголовок объекта

Перейдем в режим конфигурирования, создадим еще одну константу (она пригодится нам позже):

Имя: ПрефиксНомера

Тип: Строка

Длина: 2

Включим эту константу в подсистему **Администрирование**. В режиме 1С:Предприятие доступ к этой константе будет организован в группе **Сервис** панели действий раздела **Администрирование**. Кроме того, мы можем организовать доступ к константам из других мест нашего приложения. Мы можем самостоятельно включить команду для вызова формы просмотра и редактирования константы, отредактировав командный интерфейс, можем так же создать специальную форму, называемую **формой констант**.

Форма констант

Для создания формы констант нужно вызвать контекстное меню ветви **Константы** дерева конфигурации и выбрать в нем команду **Создать форму констант**. В появившемся окне **Конструктор общих форм**, рис. 2.8., нужно оставить тип формы в значении **Форма констант**, при необходимости заполнить другие поля и нажать на кнопку **Далее**.

Рис. 2.8. Конструктор общих форм

В появившемся окне, рис. 2.9. можно настроить состав формы констант, в нашем случае нас устраивает то, что в нее включены обе созданные в конфигурации константы, поэтому нажмем на кнопку **Готово**.

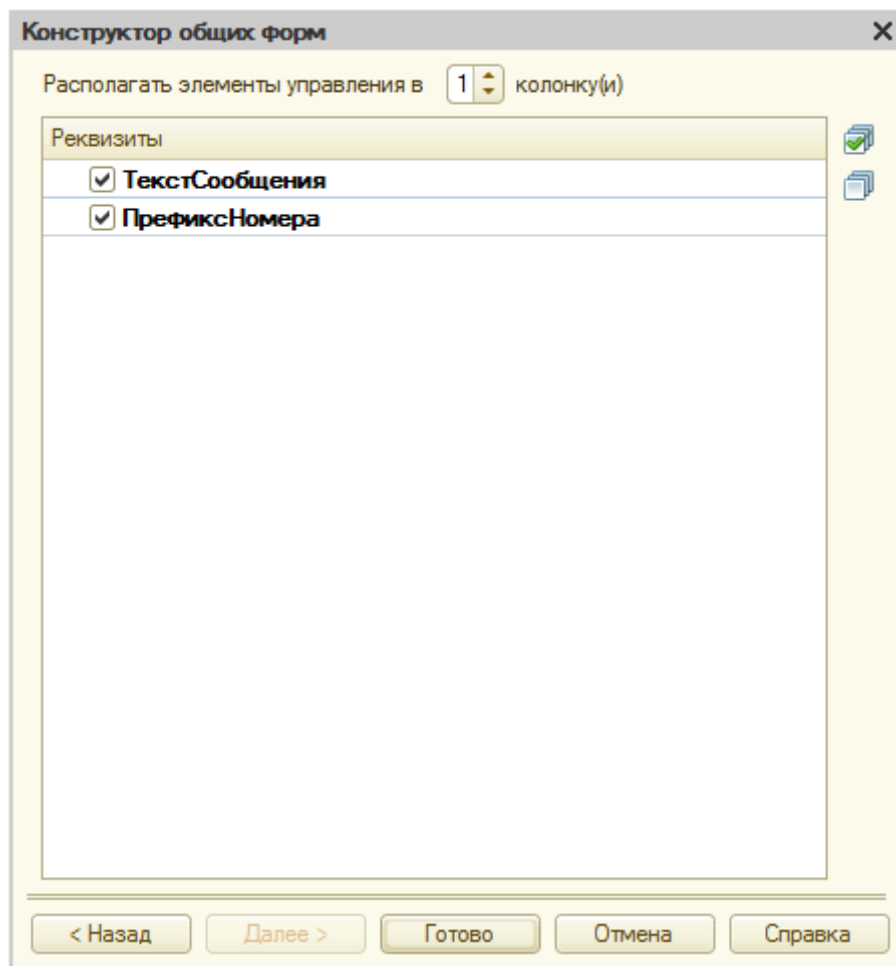


Рис. 2.9. Конструктор общих форм, состав формы констант

В ветви **Общие формы** появится новая форма с именем **ФормаКонстант**, будет открыто окно редактирования формы, рис. 2.10.

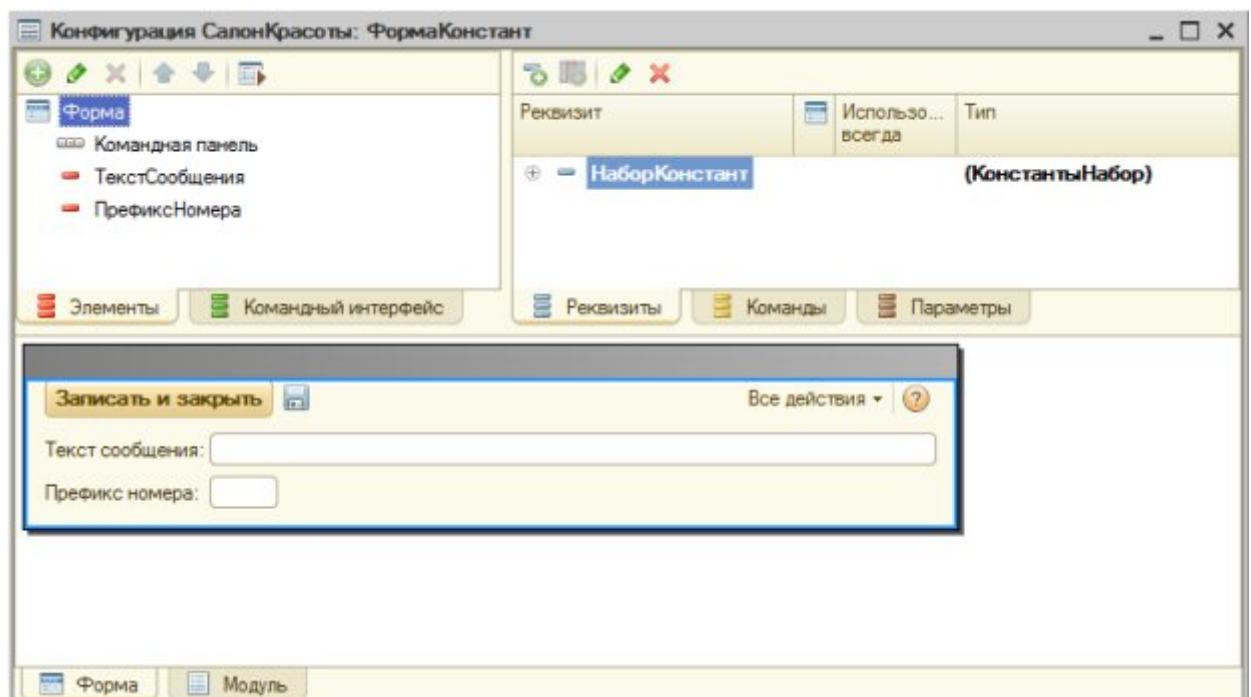


Рис. 2.10. Окно редактирования формы

При разработке командного интерфейса в 1С:Предприятие 8.3. используется концепция **декларативного описания**. Это означает, что разработчик задает схематичное описание элементов интерфейса, их группировку, свойства. При построении интерфейса для конкретного пользователя система создает его, опираясь на описание, сделанное разработчиком, учитывая различные дополнительные факторы, например, такие как права пользователя и пользовательские настройки интерфейса.

Форму констант так же нужно включить в одну из подсистем. Включим ее в подсистему **Администрирование**, посмотрим, что у нас получилось, рис. 2.11.

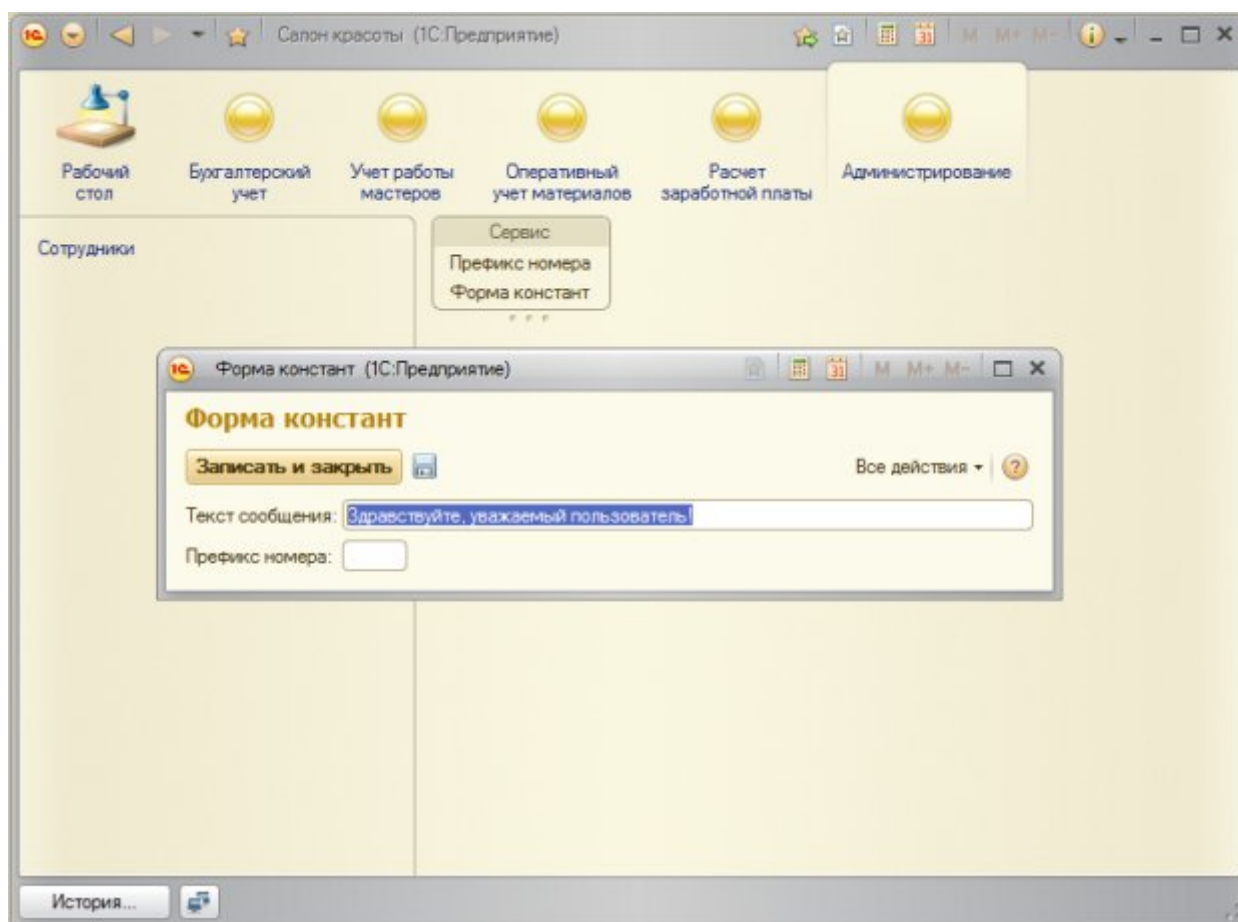


Рис. 2.11. Окно редактирования формы

Мы видим, что форма констант доступна в группе **Сервис** панели действий раздела **Администрирование**. В текущей ситуации наличие в той же группе команды вызова окна константы **Префикс номера** может показаться избыточным. Для того чтобы убрать эту команду из панели действий, нам понадобится отредактировать командный интерфейс. Для этого мы можем выполнить команду **Открыть командный интерфейс** подсистемы **Администрирование** и в появившемся окне, рис. 2.12., снять флаг **Видимость** для команды **Префикс номера** группы **Сервис** панели действий.

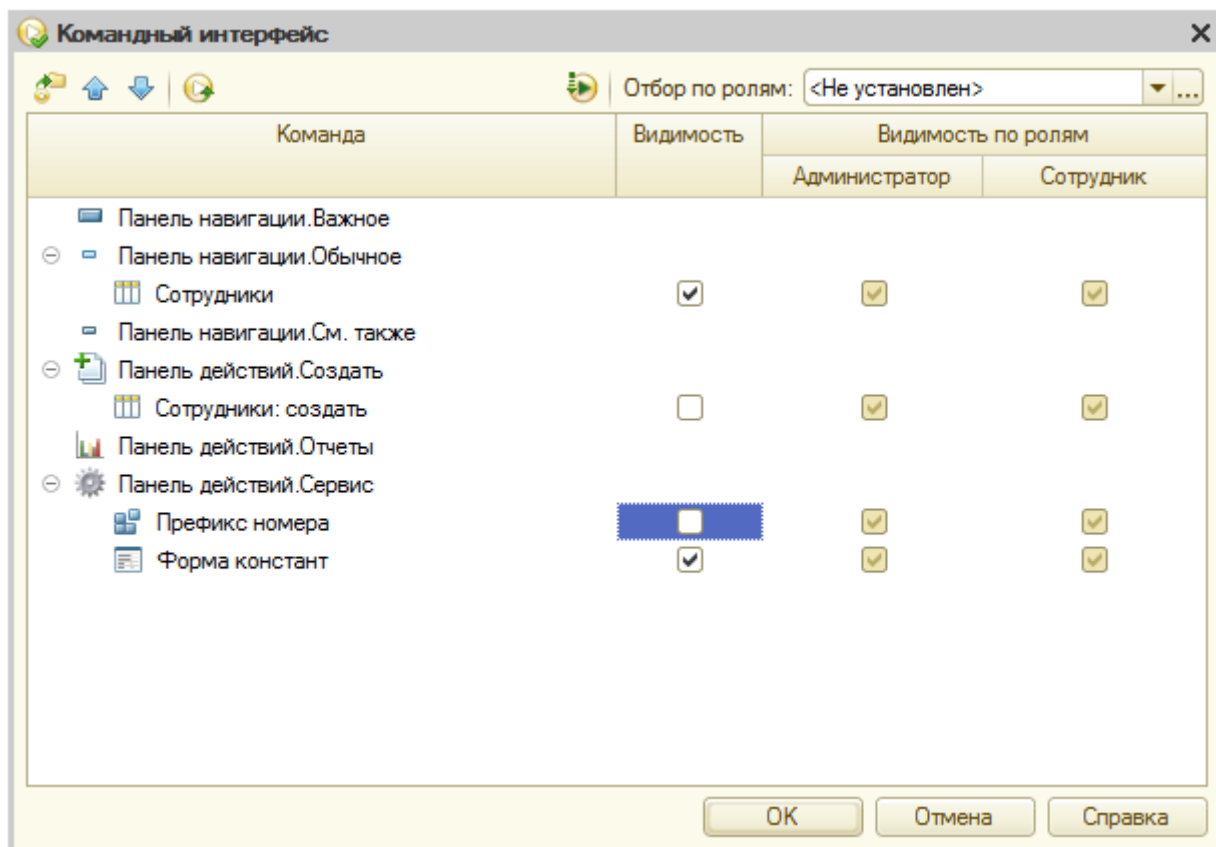


Рис. 2.12. Настройка панели действий

Теперь при запуске в режиме 1С:Предприятие ненужная команда отображаться не будет.

Выше мы создавали константу **Текст сообщения**, предполагая выводить заданный в ней текст в качестве сообщения для пользователей, входящих в систему. Реализуем эту функциональность. Для этого нам понадобится написать код в модуле управляемого приложения. Для того, чтобы открыть этот модуль, нужно воспользоваться командой **Открыть модуль управляемого приложения** корневого элемента конфигурации. Для этого модуля предусмотрено несколько стандартных обработчиков событий, которые можно найти в панели инструментов **Модуль**, рис. 2.13. Нас интересует обработчик **ПриНачалеРаботыСистемы**.

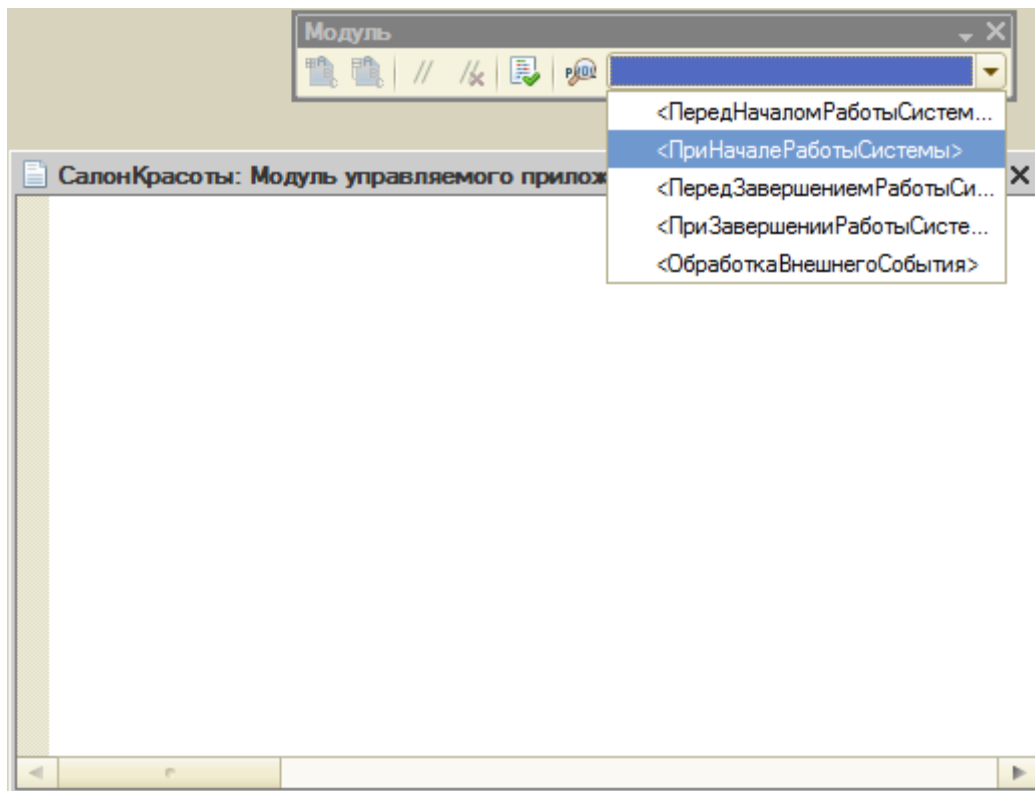


Рис. 2.13. Выбор обработчика ПриНачалеРаботыСистемы

В модуле появится пустое тело обработчика, в которое нам нужно ввести команду для вывода сообщения пользователям. Если попытаться обратиться к константе напрямую из модуля управляемого приложения – мы столкнемся с ошибкой. Дело в том, что исполнение модуля управляемого приложения происходит на клиенте, в контексте которого нет доступа к константам. Поэтому нам понадобится код, который выполняется на сервере и возвращает значение константы. В данном примере мы можем возложить на серверную часть примера и вывод сообщения (функция **Сообщить**, которой можно здесь воспользоваться, работает и на клиенте, и на сервере). Но гораздо полезнее, в плане перспектив повторного использования кода серверной процедуры, "вытащить" серверные данные в метод, который выполняется на клиенте.

Создадим новый общий модуль (в ветви **Общие модули** дерева конфигурации), назовем его **СерверныеФункции**. Проследим за тем, чтобы в его свойствах были установлены флаги **Сервер** и **Вызов сервера**, рис. 2.14.

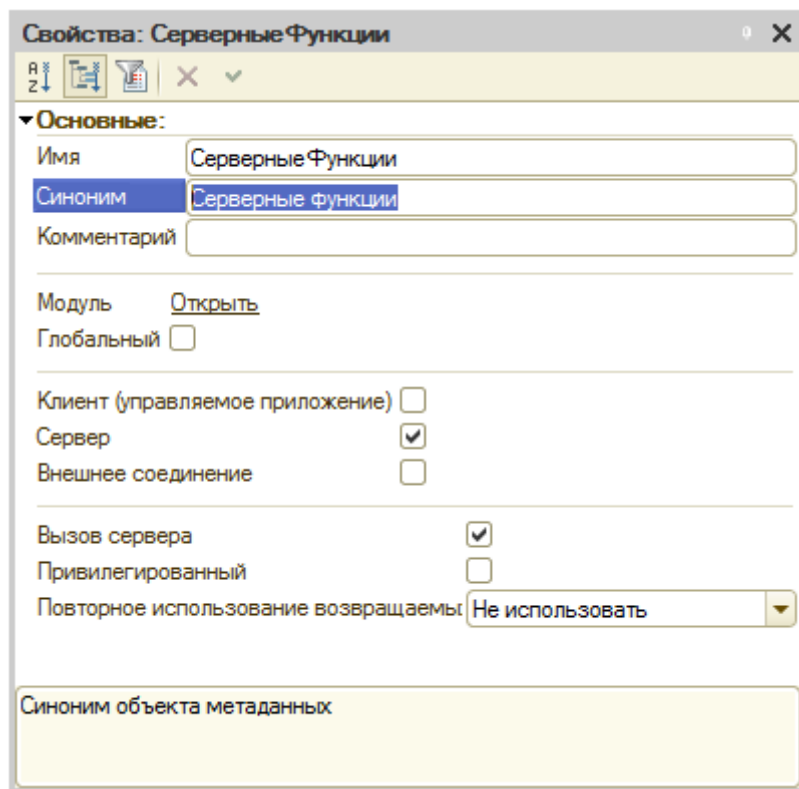


Рис. 2.14. Общий модуль СерверныеФункции, свойства

Откроем редактор кода для кода модуля (например, двойным щелчком по модулю в дереве конфигурации) и введем следующий код, рис. 2.15.:

```
//Экспортная функция для вызова из других модулей
Функция ПолучитьКонстанту() Экспорт
    //Возвращаем полученное значение константы
    Возврат (Константы.ТекстСообщения.Получить());
КонецФункции
```

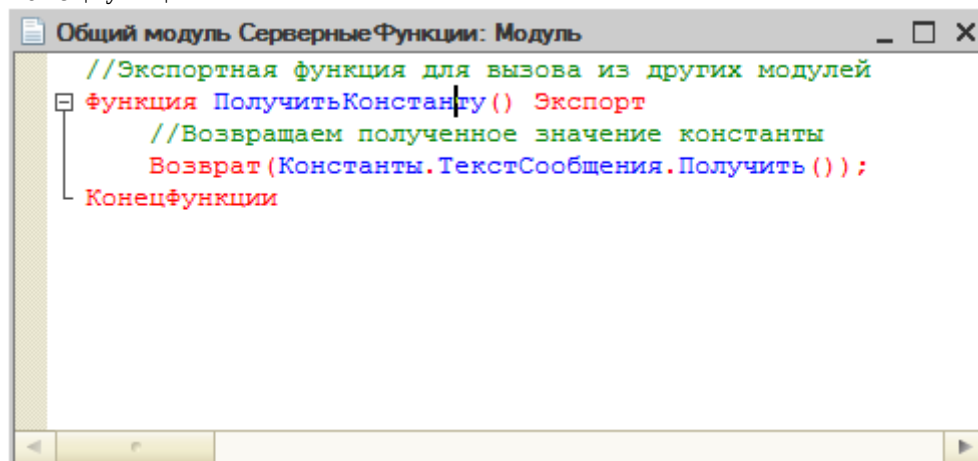


Рис. 2.15. Общий модуль СерверныеФункции, код

Теперь нам нужно вызвать эту функцию в подходящем месте кода обработчика события **ПриНачалеРаботыСистемы** в модуле управляемого приложения. Например, это можно сделать так:

```
Процедура ПриНачалеРаботыСистемы()
    //Выводим сообщение пользователю
```

```
Сообщить (СерверныеФункции.ПолучитьКонстанту ( ) ) ;  
КонецПроцедуры
```

В результате при входе в систему мы получим сообщение следующего вида, рис. 2.16.

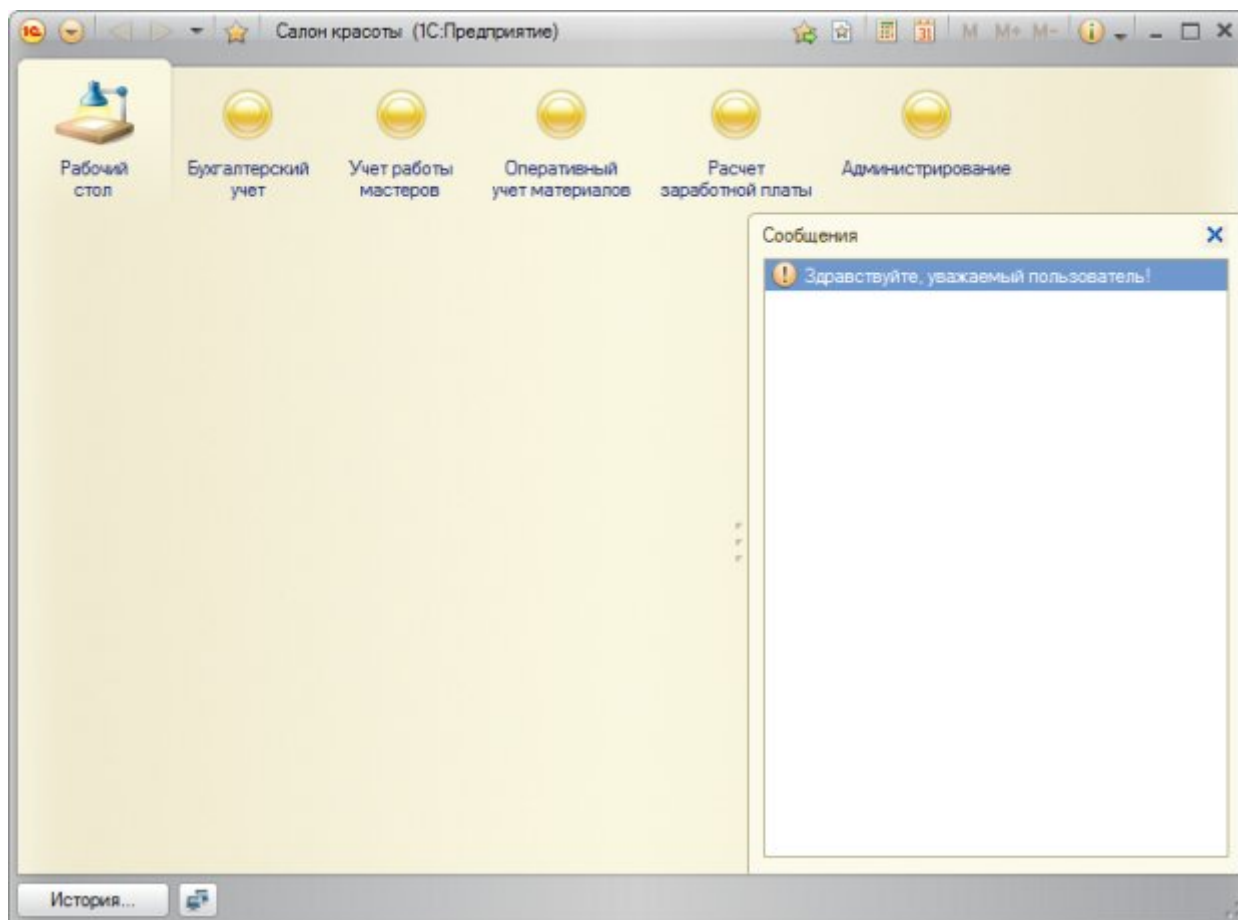


Рис. 2.16. Вывод сообщения пользователю

Обратите внимание на то, что сообщение выводится в область **Сообщения** основного рабочего окна. Если сообщение вызвано из модуля какого-либо отдельного окна, например, из модуля формы констант, которая создана ранее, то, по умолчанию, сообщение будет выведено в этом окне.

Посмотрим на этот механизм в действии. Откроем окно редактирования формы констант (**Общие формы > ФормаКонстант**), перейдем на вкладку **Модуль**, на панели инструментов **Модуль** выберем стандартный обработчик события **ПриОткрытии**, отредактируем тело обработчика, чтобы оно приняло следующий вид, рис. 2.17.:

```
&НаКлиенте  
Процедура ПриОткрытии(Отказ)  
    Сообщить ("Вы открыли форму констант!");  
КонецПроцедуры
```

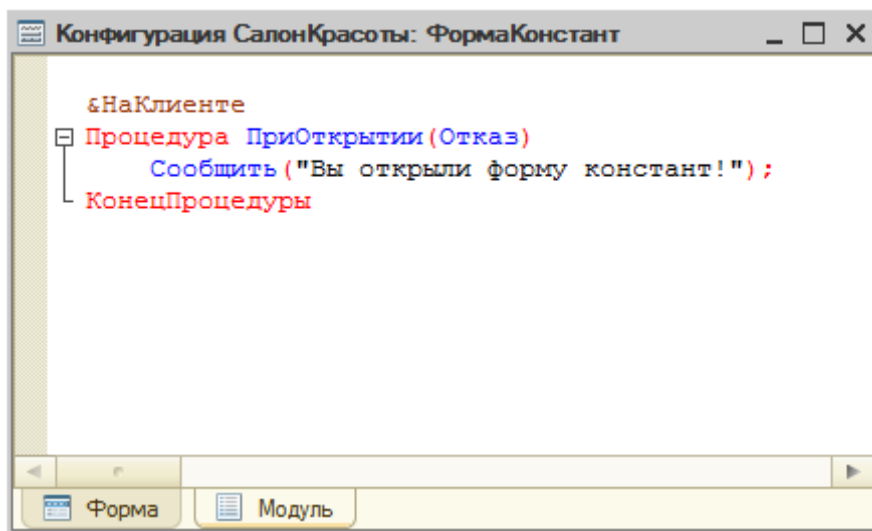


Рис. 2.17. Вывод сообщения пользователю из модуля формы констант

Благодаря этому коду при открытии формы констант будет появляться следующее сообщение, рис. 2.18.

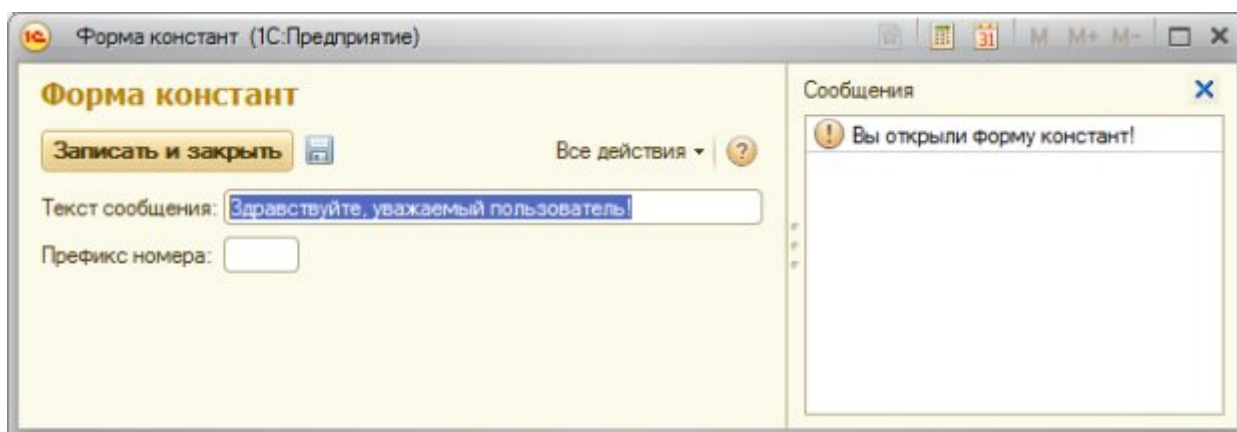


Рис. 2.18. Вывод сообщения в форму констант

Основы клиент-серверного программирования

Обратите внимание на директиву компиляции **&НаКлиенте**, которая, в вышеописанном участке кода, автоматически размещена перед описанием процедуры **ПриОткрытии()**. Создавая решение для 1С:Предприятие 8.3 разработчик должен четко разграничивать код, исполняемый в клиентской и серверной частях приложения. Причем, на клиенте (в контексте клиента) и на сервере (в контексте сервера) доступны различные объекты, различные программные механизмы. Основная задача серверного кода заключается во взаимодействии с базой данных, клиентский код занимается отображением этих данных и взаимодействием с пользователем. А задача разработчика заключается в том, чтобы создать код с учетом клиент-серверного взаимодействия.

Если перед описанием процедуры, функции или переменной в модуле формы отсутствует директива компиляции, по умолчанию считается, что код будет исполняться на сервере. В явном виде это задается указанием директивы **&НаСервере**.

Попытаемся в нашем модуле формы вывести в окно сообщения значение константы. Для этого мы можем добавить в модуль функцию, возвращающую значение константы, которая должна выполняться в контексте сервера. Например, это можно сделать одним из следующих способов – ниже приведена дополненная процедура ПриОткрытии и еще пара процедур, заданных в коде модуля формы:

```
&НаКлиенте
Процедура ПриОткрытии(Отказ)
    Сообщить ("Вы открыли форму констант!");
    Сообщить (ПолучитьКонстанту() + " - из функции модуля формы без директивы");
    Сообщить (СерверныеФункции.ПолучитьКонстанту() + " - из общего модуля");
    Сообщить (ПолучитьКонстантуНаСервере() + " - из функции модуля формы с
директивой &НаСервере");
КонецПроцедуры

//По умолчанию функция считается серверной
Функция ПолучитьКонстанту()
    //Возвращаем полученное значение константы
    Возврат (Константы.ТекстСообщения.Получить());
КонецФункции

//Директива компиляции задана явно
&НаСервере
Функция ПолучитьКонстантуНаСервере()
    //Возвращаем полученное значение константы
    Возврат (Константы.ТекстСообщения.Получить());
КонецФункции
```

Здесь мы создали пару функций – одну назвали **ПолучитьКонстанту()**, при ее описании директиву компиляции мы не указывали. Вторую назвали **ПолучитьКонстантуНаСервере()** – при ее описании была указана директива **&НаСервере**. Мы вызвали эти функции для вывода сообщения в клиентской процедуре **ПриОткрытии()**. У нас уже есть серверная функция в общем модуле **СерверныеФункции** – здесь показан пример ее использования, в подобном случае, возникшем при реальной разработке, если действия, которые выполняются в серверной функции модуля формы, совпадают с действиями функции, описанной в общем модуле, можно и даже нужно пользоваться функцией общего модуля.

На рис. 2.19. вы можете видеть вывод сообщений, выполненный вышеприведенным кодом.

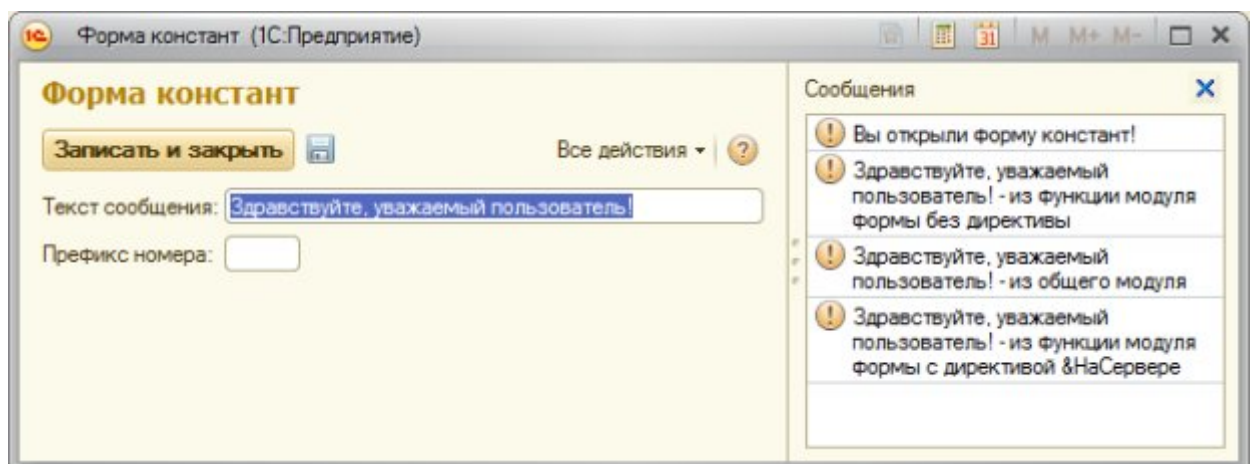


Рис. 2.19. Вывод сообщения в форму констант, разные варианты работы с серверными данными

Вернемся к директивам компиляции. Они могут использоваться не только в модулях форм, но и в общих модулях, и в модулях команд. Таблица 2.1. содержит описание возможности применения директив компиляции в различных модулях

Таблица 2.1. Применение директив компиляции в модулях и переменных				
Директивы компиляции	Модули			Переменные
	Модуль формы	Модуль команды	Общий модуль	
&НаКлиенте	Да	Да	Да	Да
&НаСервере	Да	Да	Да	Да
&НаКлиентеНаСервере	Нет	Да	Нет	Нет
&НаСервереБезКонтекста	Да	Нет	Нет	Нет
&НаКлиентеНаСервереБезКонтекста	Да	Нет	Нет	Нет

Опишем основные особенности их применения:

&НаКлиенте – эта директива предназначена для клиентских процедур и функций. Из такого метода могут быть вызваны любые процедуры и функции. Кроме того, с данной директивой можно объявлять переменные – их называют клиентскими. Такая переменная существует столько же, сколько существует клиентская часть формы. Из метода с данной директивой доступны другие клиентские переменные модуля формы.

&НаСервере – эта директива предназначена для серверных процедур и функций. Из такой процедуры могут быть вызваны серверные, серверные и клиент-серверные внеконтекстные методы, а так же методы неглобальных серверных общих модулей. Допустимо объявление переменных с данной директивой – такие переменные существуют во время выполнения вызова сервера. Из серверных методов доступны серверные переменные, объявленные в модуле формы.

&НаКлиентеНаСервере – предназначена для описания процедур и функций, выполняемых на клиенте и на сервере. Такие процедуры и функции могут вызывать клиентские и серверные процедуры общих модулей. Не подходит для объявления переменных.

&НаСервереБезКонтекста – такая процедура или функция выполняется на сервере вне контекста формы, в модуле которой она описана. Из нее можно вызывать лишь другие внеконтекстные процедуры или функции. Использование этой директивы позволяет сократить объем данных, передаваемых между сервером и клиентом. Не подходит для объявления переменных.

&НаКлиентеНаСервереБезКонтекста – такая процедура или функция может выполняться на клиенте и на сервере, без доступа к контексту формы. Не подходит для объявления переменных.

Помимо директив компиляции в модуле управляемой формы можно пользоваться инструкциями препроцессору. Инструкции препроцессору обрабатываются до того, как будут обработаны директивы компиляции.

Обсудив основы работы с константами и основные положения клиент-серверного программирования, перейдем к работе со справочниками. Обычно разработка системы справочников относится к начальному этапу разработки любой конфигурации, так как на типах данных, заданных справочниками, основываются другие механизмы системы. Но прежде чем начать разговор, о, собственно, справочниках, давайте поговорим об общих реквизитах.

Общие реквизиты

Общие реквизиты являются новой возможностью, которая была добавлена в 14-й релиз платформы 1С:Предприятие 8.3. Общие реквизиты можно использовать двумя способами. Первый из них заключается в использовании их, как, собственно, реквизитов, которые присутствуют у всех (или у достаточно большого количества) объектов конфигурации. Второй способ предусматривает использование общих реквизитов в механизме разделения данных в качестве разделителей данных. В базе, использующей механизм разделения данных, могут работать несколько пользователей, набор данных каждого из которых не пересекается с набором данных других пользователей, то есть – каждый из них считает, что база содержит лишь "его" данные и ничего больше.

Сейчас мы создадим общие реквизиты, которые планируется использовать именно как реквизиты для других объектов конфигурации. В частности, общий реквизит может быть "подключен" к следующим объектам. Правильным будет и утверждение о том, что эти объекты могут "входить в состав" общего реквизита, так как, собственно, от настройки состава общего реквизита зависит его появление в других объектах. Итак, речь идет о следующих объектах:

- планы обмена
- справочники
- документы
- планы видов характеристик
- планы счетов
- планы видов расчета
- регистры сведений
- регистры накопления
- регистры бухгалтерского учета
- регистры расчета
- бизнес-процессы
- задачи

В нашем учебном примере мы собираемся вести в базе данных учет по нескольким организациям. Для этого нам понадобится, чтобы все объекты конфигурации, для которых уместен данный реквизит, содержали бы реквизит **Организация**, который содержит ссылку на организацию. Например, каждый документ будет оформляться от лица определенной организации, каждый элемент справочника будет относиться к той или иной организации, и так далее. Для того, чтобы не усложнять наши примеры, мы не будем в дальнейших лекциях курса развивать тему многофирменного учета в одной базе данных. Однако, в любом случае, общие реквизиты позволяют снизить трудоемкость разработки.

Второй реквизит, который предназначен для документов, будет использоваться для ввода комментариев к документу.

Прежде чем продолжать работу над общими реквизитами, создадим следующие объекты конфигурации, не настраивая их дополнительных свойств – справочник с именем **Организация**, и документ с именем **ПоступлениеМатериалов**. Включим их в подсистему **ОперативныйУчетМатериалов**.

Создадим новый **общий реквизит** со следующими параметрами, рис. 2.20.:

Имя: Комментарий

Тип: Строка, длина 50

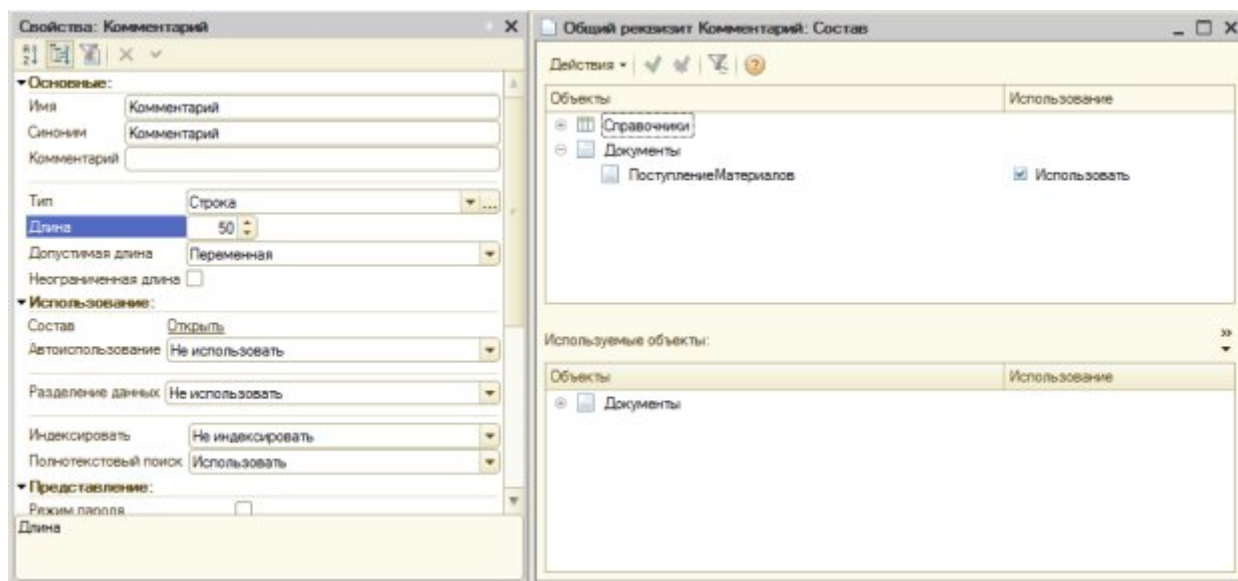


Рис. 2.20. Настройка общего реквизита

Обратите внимание на параметр **Автоиспользование**. В данном случае мы оставляем его в значении по умолчанию – **Не использовать**. То есть – состав общего реквизита мы будем настраивать вручную. Этот общий реквизит мы планируем добавить ко всем документам, поэтому найдем свойство **Состав**, нажмем на ссылку **Открыть**, в появившемся окне выберем вариант **Использовать** для документа **ПоступлениеМатериалов**. При создании других документов мы сможем самостоятельно включать их в состав общего реквизита. Быстро проверить состав используемых объектов общего реквизита можно в нижней части окна настройки состава.

Преимущества использования общих реквизитов напоминают использование процедур и функций в общих модулях, к которым обращаются из многих других методов. Если возникает необходимость в изменении общего реквизита – например – в процессе работы в конфигурации оказалось, что длину комментария нужно увеличить – достаточно изменить параметры типа общего реквизита, и это изменение затронет все объекты конфигурации, включенные в его состав.

Создадим **второй** общий реквизит:

Имя: Организация

Тип: СправочникСсылка.Организации

Автоиспользование: Использовать

Этот реквизит мы планируем добавить ко всем объектам, допускающим использование общих реквизитов, за исключением справочника **Организации** и некоторых других. Перейдем в окно настройки состава общего реквизита и установим свойство **Использование** у справочника **Организации** в значение **Не использовать**, рис. 2.21.

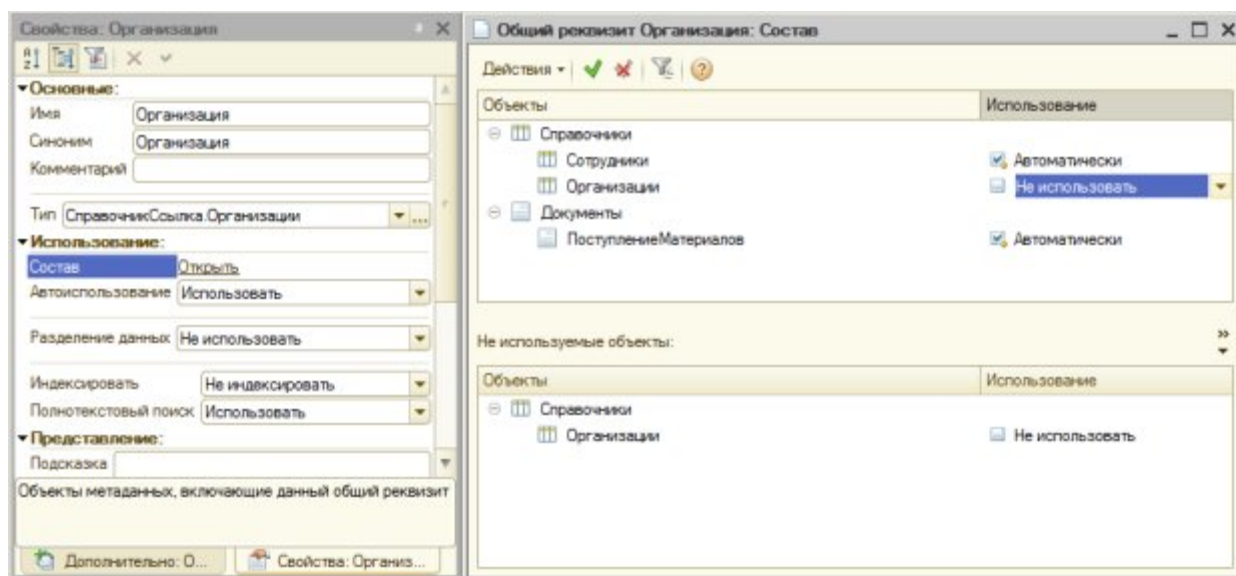


Рис. 2.21. Настройка общего реквизита

Откроем нашу конфигурацию в режиме 1С:Предприятие и посмотрим, как выглядит документ **ПоступлениеМатериалов** и справочники **Организации** и **Сотрудники**.

Для начала перейдем на вкладку **Оперативный учет материалов**. Обратите внимание на то, что в панель навигации раздела были автоматически добавлены ссылки для доступа к только что созданному справочнику **Организации** и к документу **Поступление материалов**. Щелкнем по ссылке **Организации**. В рабочей области окна появится список справочника. На данный момент он пуст, так как мы пока не заполняли справочник организациями, по которым будет вестись учет в базе. Щелкнем по кнопке **Создать**, которая расположена на командной панели списка – появится отдельное окно для заполнения свойств элемента справочника, рис. 2.22. Можно отметить, что помимо стандартных реквизитов (**Наименование**, **Код**) данный справочник не содержит ничего другого – это неудивительно, мы исключили его из состава общего реквизита **Организация**.

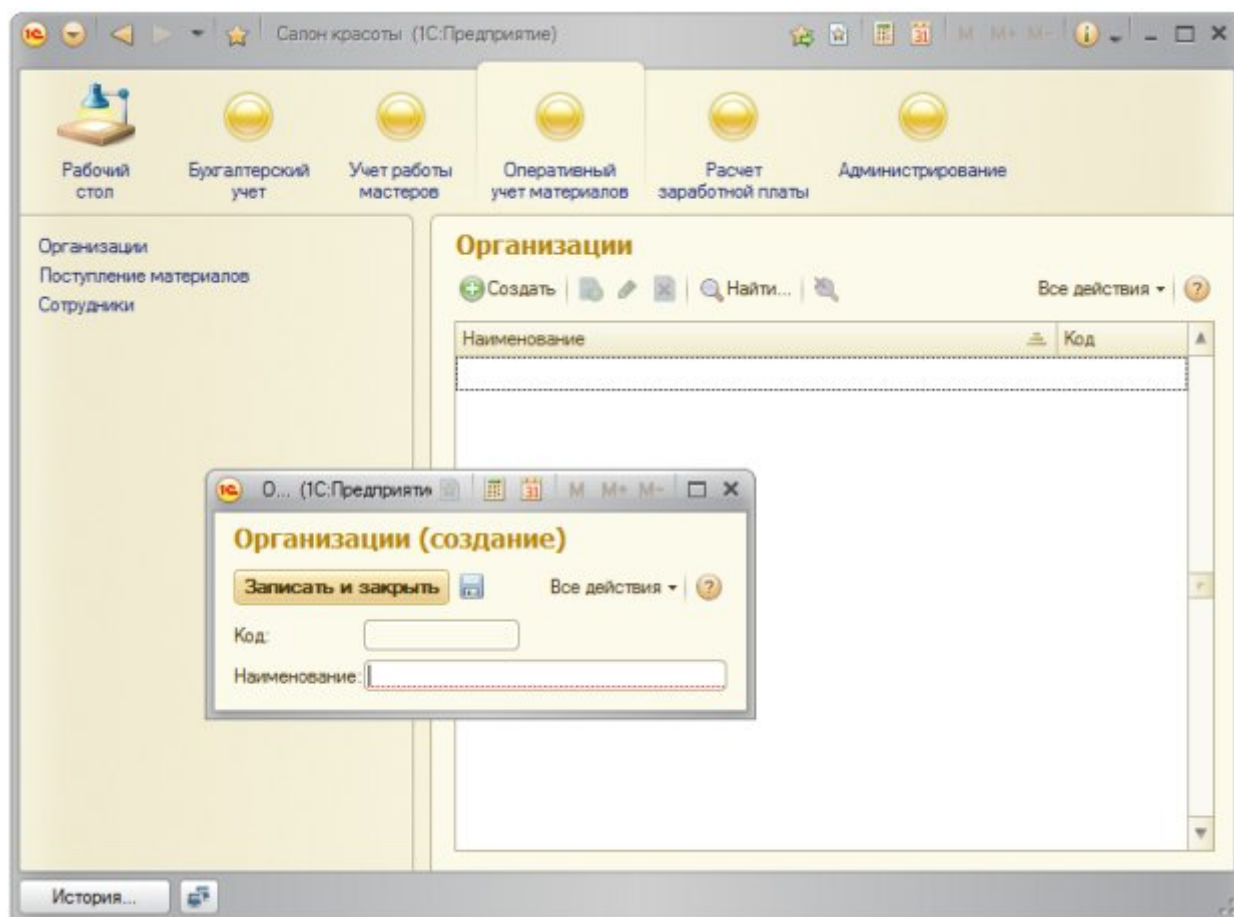


Рис. 2.22. Справочник Организации

Теперь откроем список справочника **Сотрудники** и нажмем на кнопку **Добавить**. Общий реквизит **Организация** у данного справочника присутствует, рис. 2.23.

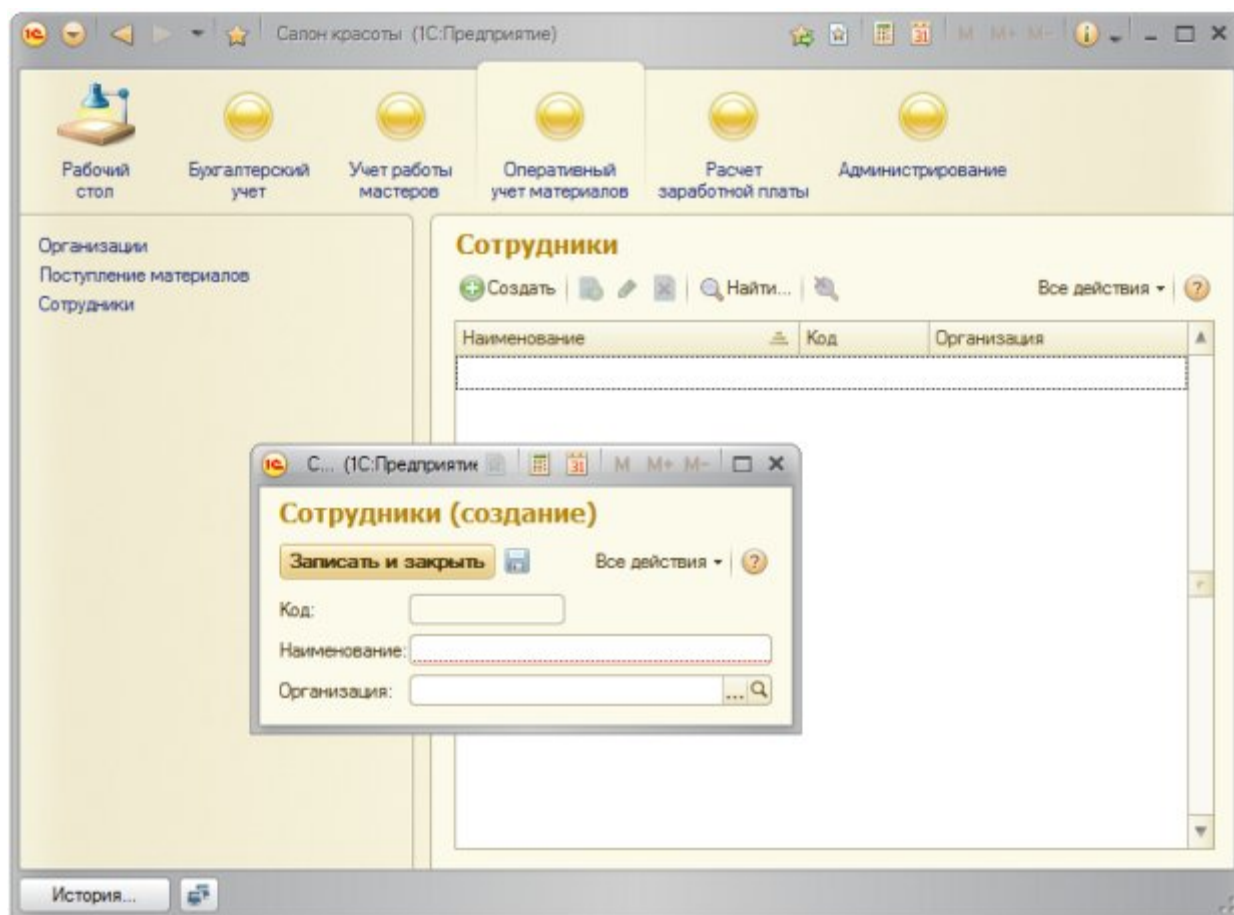


Рис. 2.23. Справочник Сотрудники

Откроем, наконец, окно создания документа **ПоступлениеМатериалов**. Здесь мы видим два общих реквизита – **Комментарий** и **Организация**, рис. 2.24.

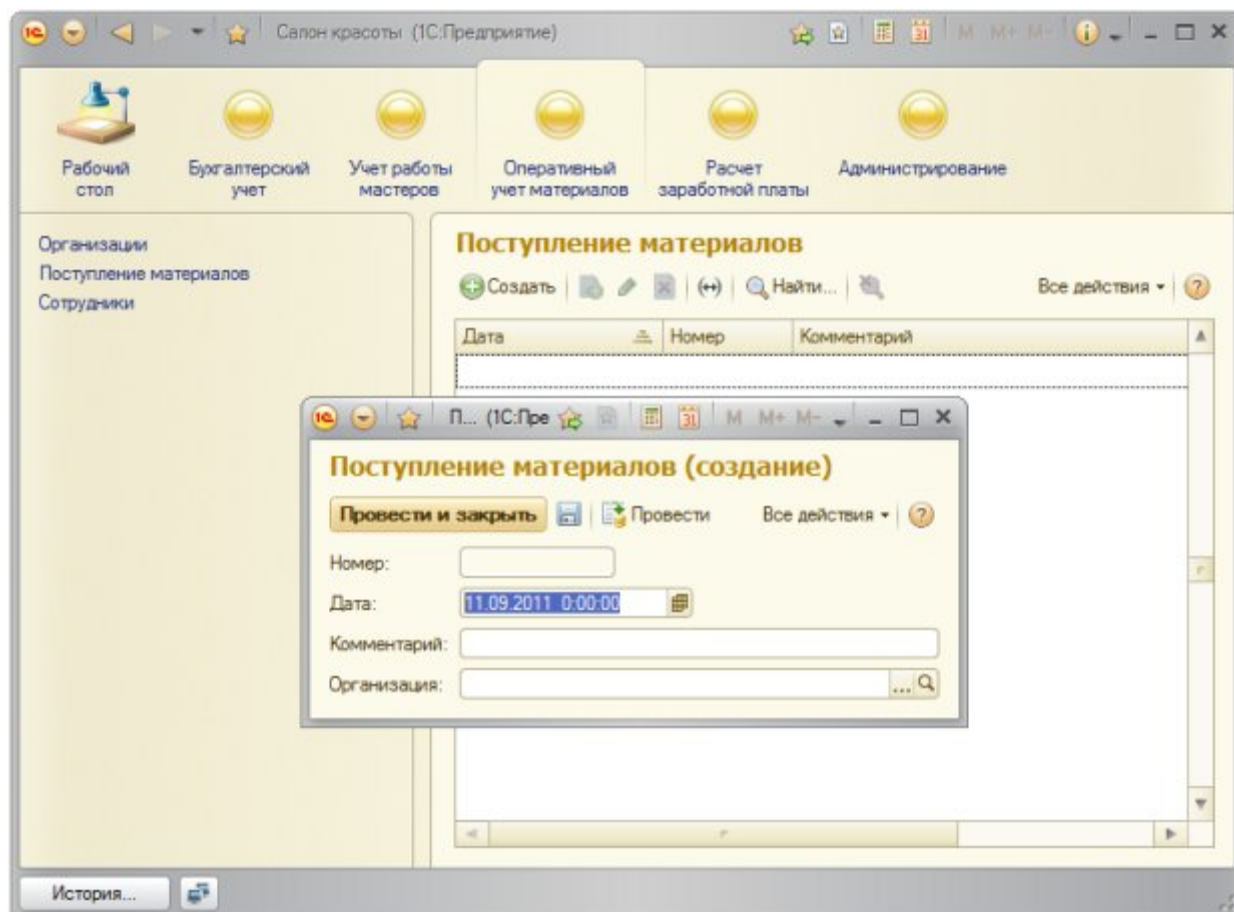


Рис. 2.24. Документ ПоступлениеМатериалов