

Справочники, разработка форм

Посвящена конструированию справочников, вопросам разработки управляемых форм и выполнения различных действий в клиентских методах, в частности, рассмотрена методика вывода сообщений об ошибках в привязке к элементам управления.

Справочник Организации

Справочник можно сравнить с картотекой, с неким списком данных, каждая запись которого имеет определенную структуру. В организации – независимо от того, автоматизирован ли в ней учет или нет, присутствует множество таких списков. Это – списки сотрудников, клиентов, товаров.

В нашей конфигурации уже есть пара справочников. Один из них – это справочник **Организации**, который нужен для хранения списка организаций, по которым планируется вести учет. Справочник, сразу после его создания, имеет некоторые стандартные реквизиты. Это утверждение справедливо и для других объектов конфигурации. Для управления реквизитами объекта служит закладка **Данные** окна редактирования объекта, рис. 3.1.

Рис. 3.1. Настройка справочника

Ознакомиться со списком стандартных реквизитов можно, нажав на кнопку **Стандартные реквизиты** – появится окно, содержащее список таких реквизитов, рис. 3.2.

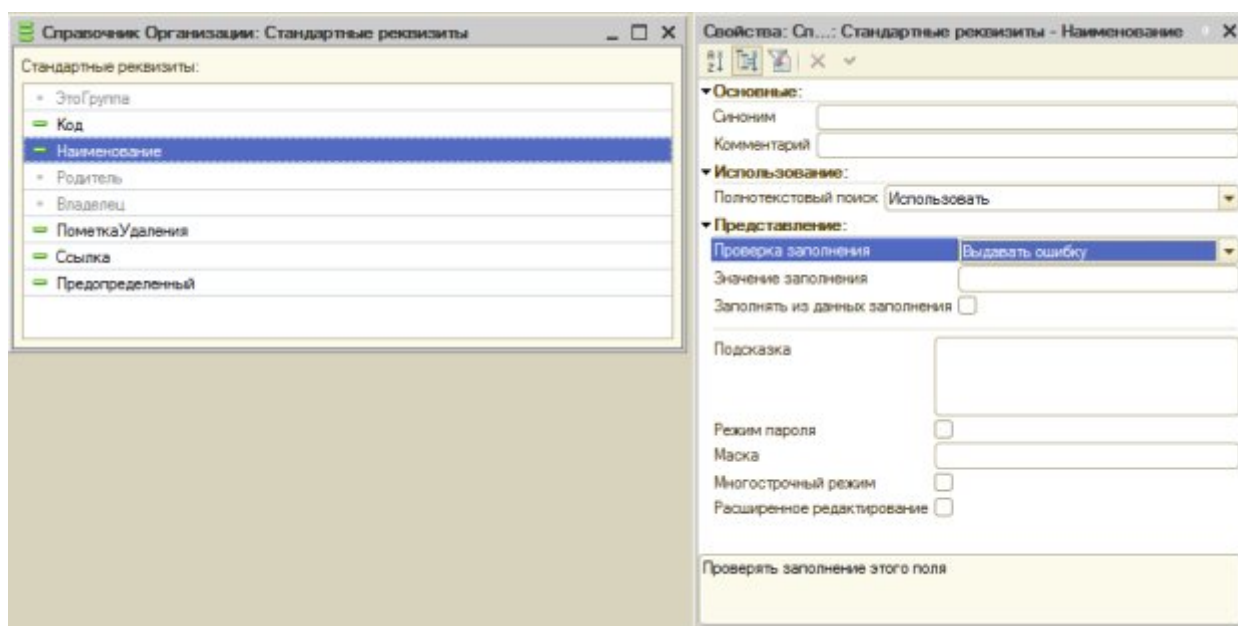


Рис. 3.2. Стандартные реквизиты справочника и их свойства

Стандартные реквизиты поддерживают настройку некоторых свойств – для доступа к свойствам стандартного реквизита, достаточно выделить его в окне и обратиться к палитре **Свойства**.

Нашему справочнику **Организации** не хватает, для полноты его использования в системе, реквизита, который содержал бы полное наименование организации. Добавим этот реквизит к справочнику – на вкладке **Данные** окна редактирования объекта, нажмем на кнопку **Добавить**, параметры реквизита будут следующими:

Имя: ПолноеНаименование

Тип: Строка, длина – 50.

Проверка заполнения: Выдавать ошибку

Свойство **Проверка заполнения** по умолчанию для новых реквизитов установлено в значение **Не проверять**. Оно позволяет автоматически проверять заполненность поля – если поле не заполнено – система выдаст ошибку. Если нам нужны особые алгоритмы проверки содержимого поля перед записью элемента справочника, мы можем реализовать эти алгоритмы самостоятельно.

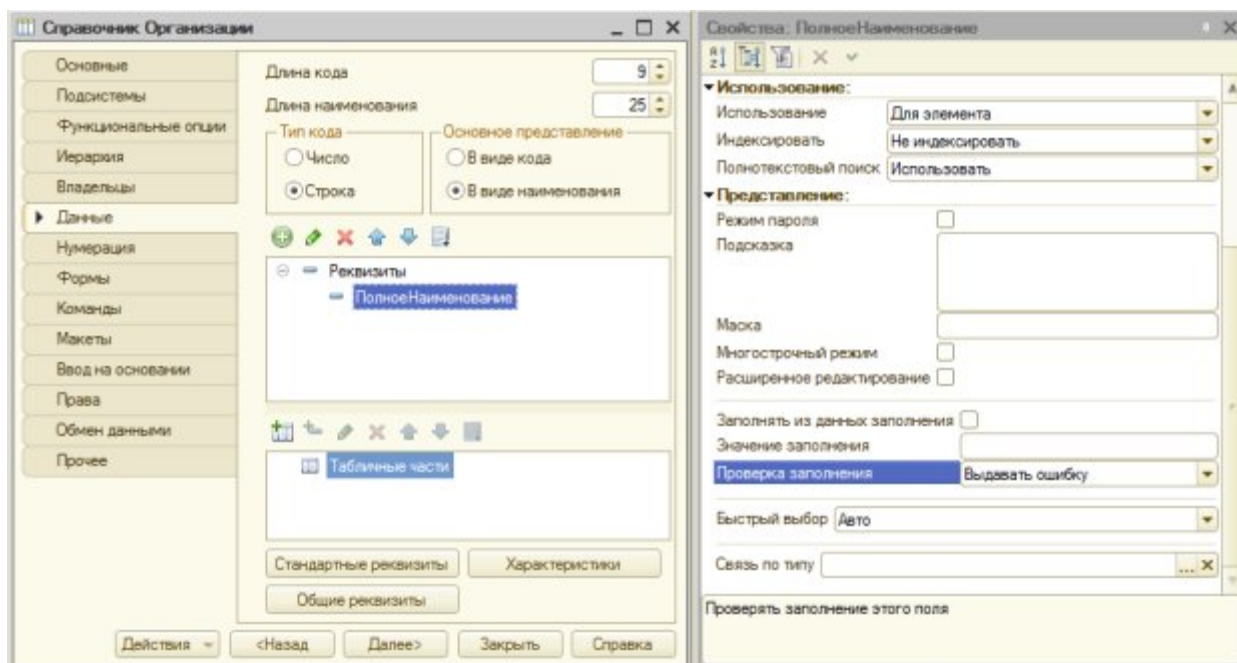


Рис. 3.3. Настройка нового реквизита справочника

Посмотрим на наш справочник в режиме 1С:Предприятие. Создадим новый элемент, дадим ему наименование **Салон красоты**, а полное наименование заполнять не будем, и попытаемся записать элемент, нажав на кнопку **Записать и закрыть**. Элемент не будет записан, мы увидим сообщение об ошибке – в виде сообщения и в виде всплывающей подсказки, рис. 3.4.

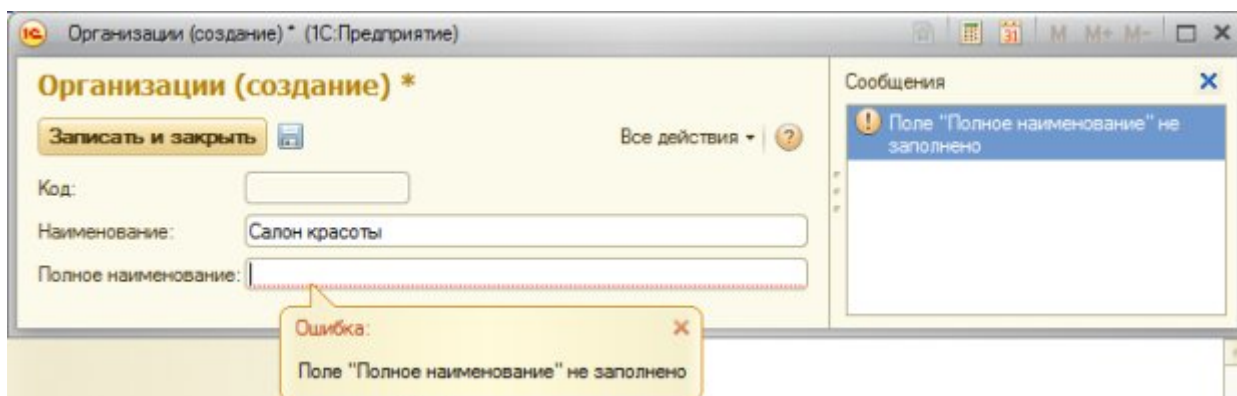


Рис. 3.4. Сообщение об ошибке при попытке записи элемента справочника

Введем в поле **Полное наименование** текст ООО "Салон красоты" - после этого можно будет записать и закрыть элемент справочника. Он отобразится в списке справочника в рабочей области окна программы, рис. 3.5.

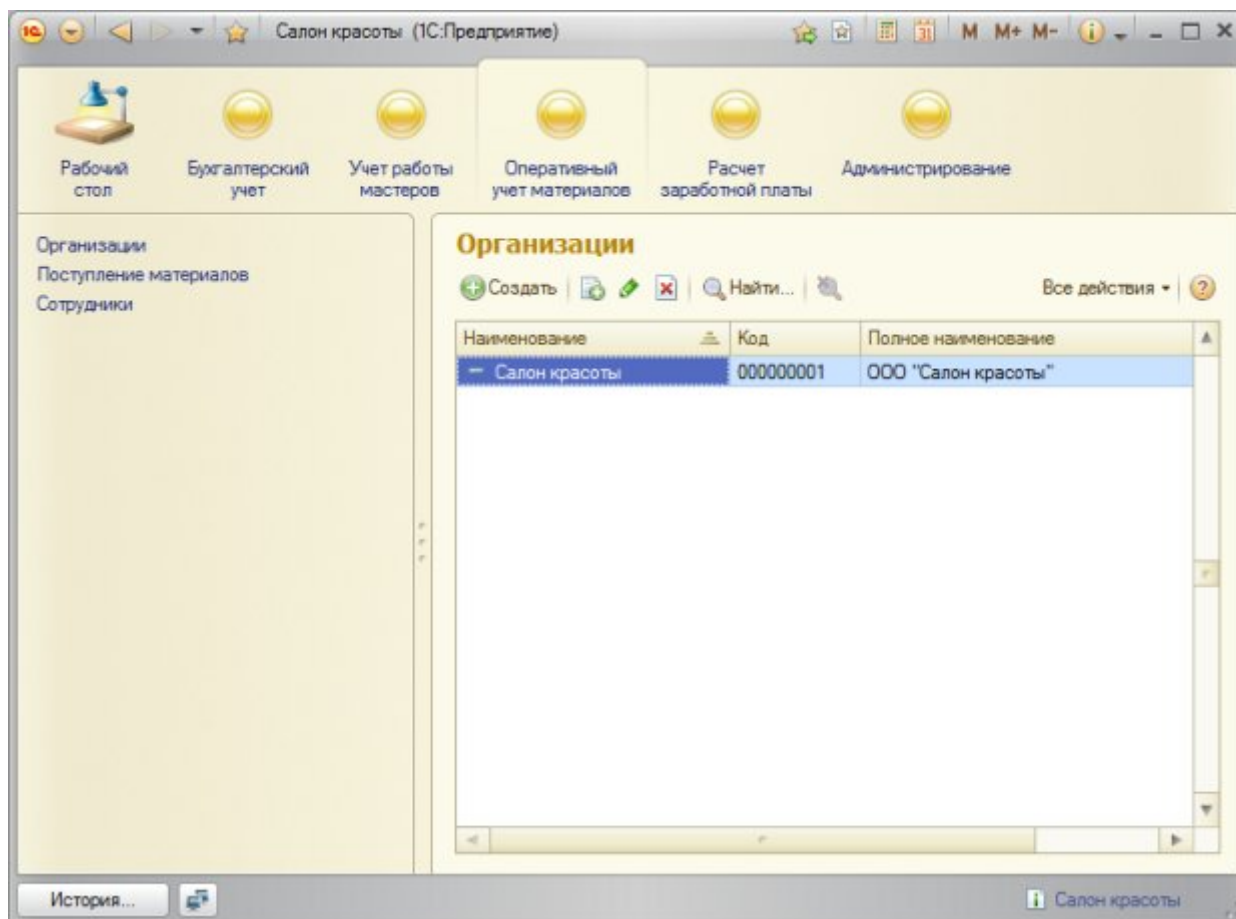


Рис. 3.5. Новый элемент справочника в списке

В информационной панели, которая расположена в нижней части окна программы, появится ссылка для доступа к только что созданному элементу и будет сообщено о его создании.

Код элементу справочника будет присвоен автоматически.

Справочники в 1С:Предприятие могут содержать predetermined elements. To their creation you can go from the **Прочее** (Other) tab, by the button **Предопределенные** (Predetermined).

Справочник ФизическиеЛица

Следующим нашим справочником будет справочник **ФизическиеЛица**. Он предназначен для хранения списка физических лиц и сведений о них. В частности, мы хотели бы хранить данные о самом физическом лице (Фамилия, Имя, Отчество, дата рождения, пол, район проживания), а так же об истории его трудовой деятельности. Для хранения данных о физическом лице хорошо подойдут обычные реквизиты справочника, которыми мы уже занимались выше. А вот для того, чтобы хранить историю трудовой деятельности, нам понадобится другая структура данных, а именно – **табличная часть** (Table Part).

Табличная часть – это таблица, состав и свойства полей (столбцов) которой мы задаем на этапе разработки. В пользовательском режиме создается необходимое количество строк. В нашем примере количество мест, в которых работало физическое лицо, заранее неизвестно.

Здесь надо отметить, что понятия "Сотрудник" и "Физическое лицо" - это разные вещи. Сотрудник – это тот, кто в настоящий момент работает в организации, и сотрудник обязательно является физическим лицом. А вот физическое лицо, сведения о котором могут храниться в базе данных организации, вполне может не являться сотрудником – например – это может быть кандидат на какую-либо должность, или, наоборот, уволенный сотрудник.

Создадим новый справочник, дадим ему имя **ФизическиеЛица**, включим его в состав подсистем **БухгалтерскийУчет**, **УчетРаботыМастеров** и **РасчетЗаработнойПлаты**.

На вкладке Данные создадим следующие реквизиты:

Имя: Фамилия

Тип: Строка, длина 30

Имя: Имя

Тип: Строка, длина 30

Имя: Отчество

Тип: Строка, длина 30

Имя: ДатаРождения

Тип: Дата, состав даты – Дата

Следующие реквизиты, которые мы планируем создать – это **Пол** и **РайонПроживания**. Строковые реквизиты, которые мы создавали выше, обычно заполняют вводом данных с клавиатуры. В случае же с указанием пола и района проживания заполнение с клавиатуры непременно приведет к появлению в базе различных наименований для одних и тех же показателей при использовании текстовых полей. Для мужского пола это вполне может быть, при ограничении длины строки одним символом, "М" и "м", для районов так же возможно различное написание. Для обеспечения единообразия при вводе подобных показателей рационально использовать для их хранения отдельные справочники или перечисления. Для хранения наименований пола мы воспользуемся перечислением.

Создадим новое перечисление, дадим ему имя **Пол**, включим в подсистему **РасчетЗаработнойПлаты**. На вкладке **Данные** окна редактирования объекта для перечисления задаются значения перечисления. Зададим два значения – **Мужской** и **Женский**, рис. 3.6.

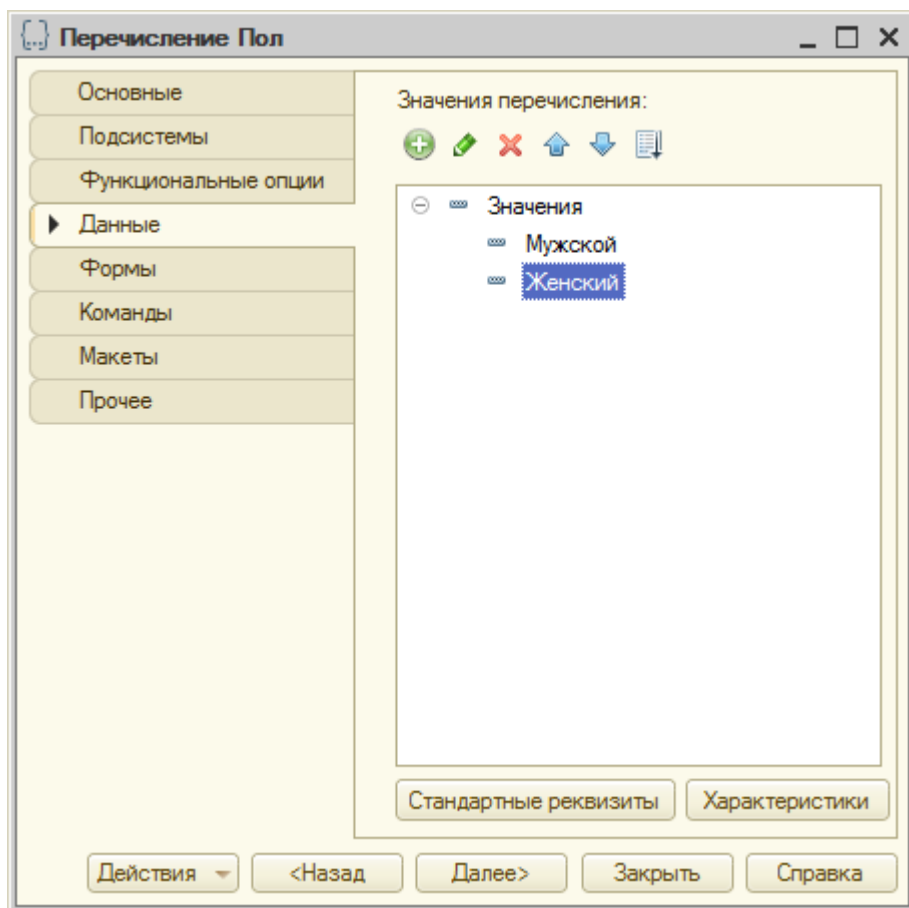


Рис. 3.6. Создание перечисления Пол

Теперь создадим новый справочник – дадим ему имя **Районы**, включим в состав подсистемы **РасчетЗаработнойПлаты**, на вкладке **Данные** изменим длину наименования до **100** символов, этот справочник не будет иметь дополнительных реквизитов, так же мы можем исключить его из состава общего реквизита **Организация**, рис. 3.7.

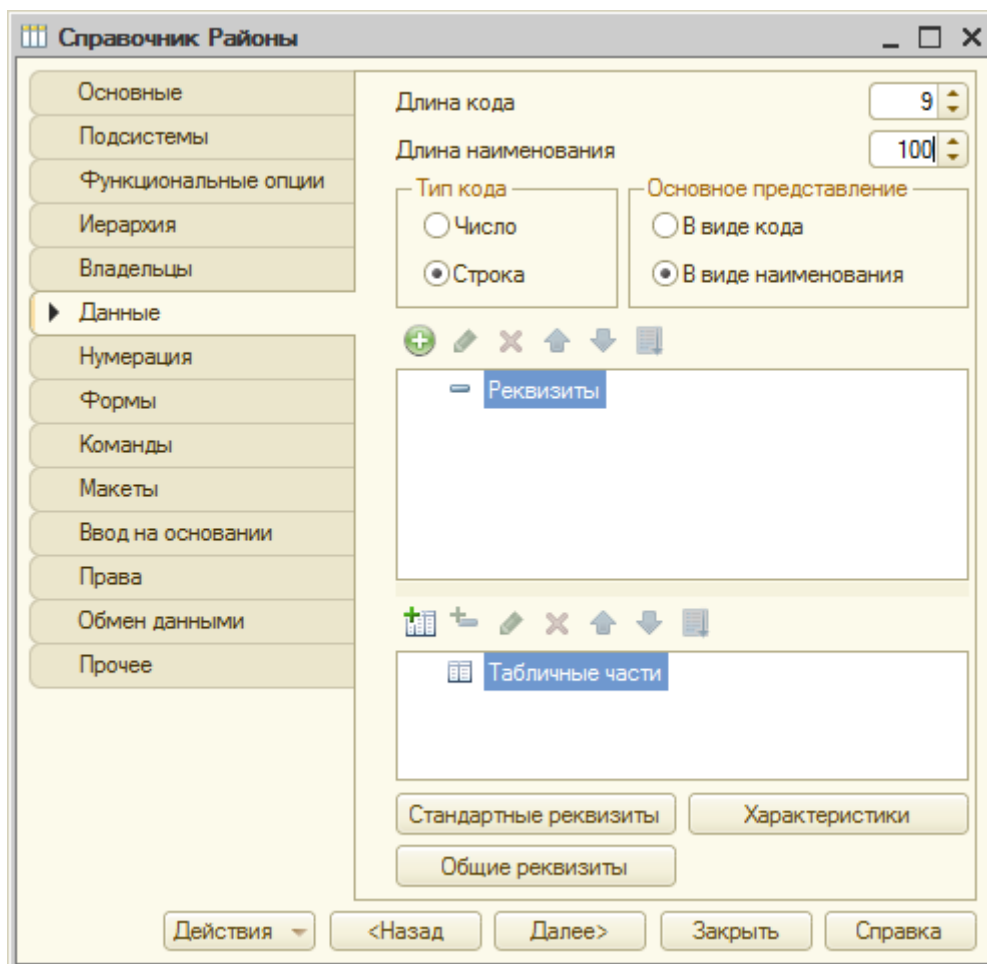


Рис. 3.7. Создание справочника Районы

Вернемся к настройке справочника **ФизическиеЛица**. Добавим еще два реквизита:

Имя: Пол

Тип: ПеречислениеСсылка.Пол

Имя: РайонПроживания

Тип: СправочникСсылка.Районы

Теперь займемся табличной частью справочника. При необходимости, справочники могут иметь несколько табличных частей. Сначала нажмем на кнопку **Добавить табличную часть**, зададим имя табличной части **ТрудоваяИстория**. В табличную часть добавим следующие реквизиты (поля), выделив табличную часть и нажав на кнопку **Добавить реквизит**:

Имя: Организация

Тип: Строка, длина 30

Имя: ДатаНачалаРаботы

Тип: Дата, состав даты – Дата

Имя: ДатаОкончанияРаботы

Тип: Дата, состав даты – Дата.

В итоге окно редактирования нашего справочника будет выглядеть так, как показано на рис. 3.8.

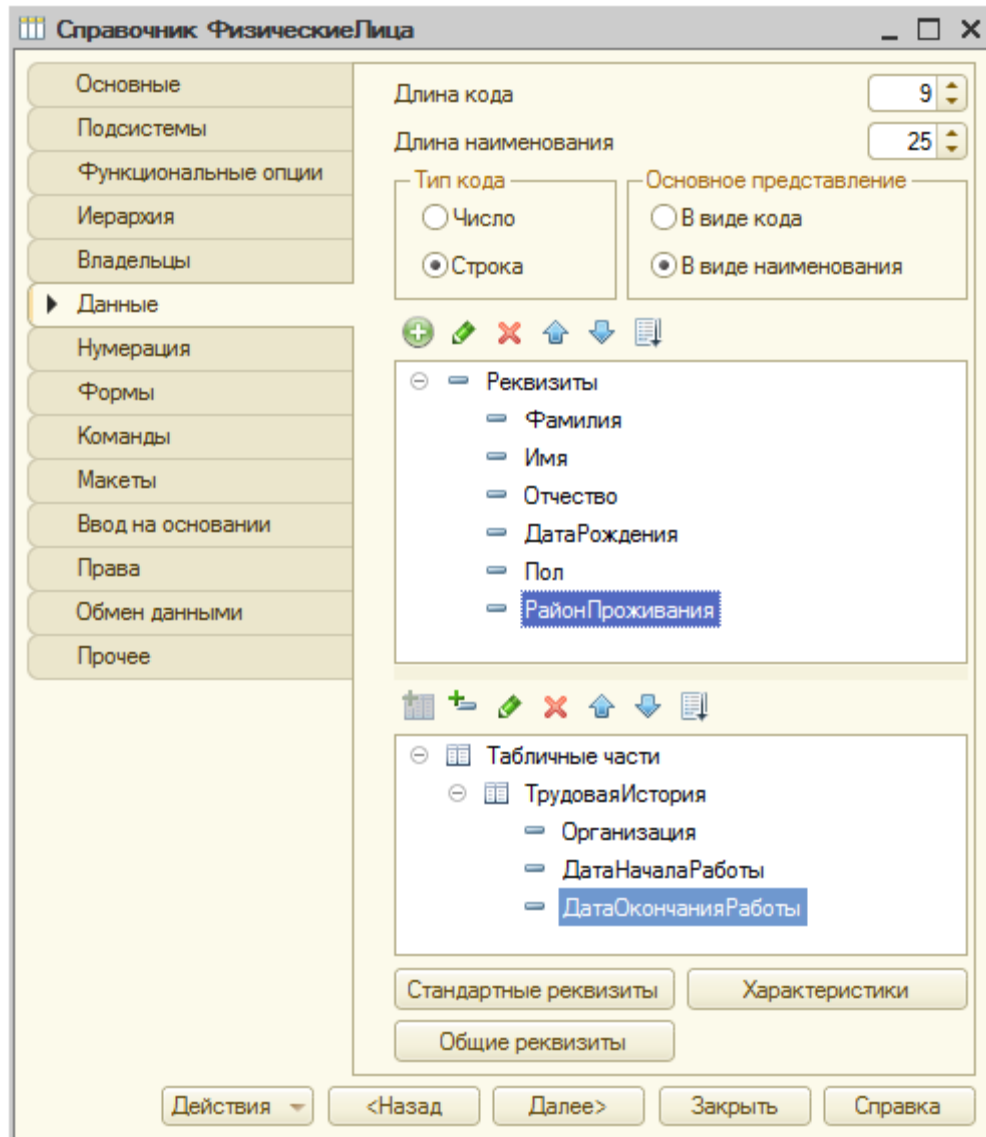


Рис. 3.8. Состав справочника ФизическиеЛица

В предыдущей лекции мы создавали общий реквизит **Организация**, который планировалось добавлять ко многим объектам конфигурации. Справочник **ФизическиеЛица** имеет смысл вести по всем организациям. Как вы уже видели, настроить состав общего реквизита можно в ветви **Общие реквизиты**. Сделать это можно и в окне редактирования объекта, нажав кнопку **Общие реквизиты** на вкладке **Данные**. Нажмем эту вкладку и установим для общего реквизита **Организация** значение **Не использовать**, рис. 3.9.

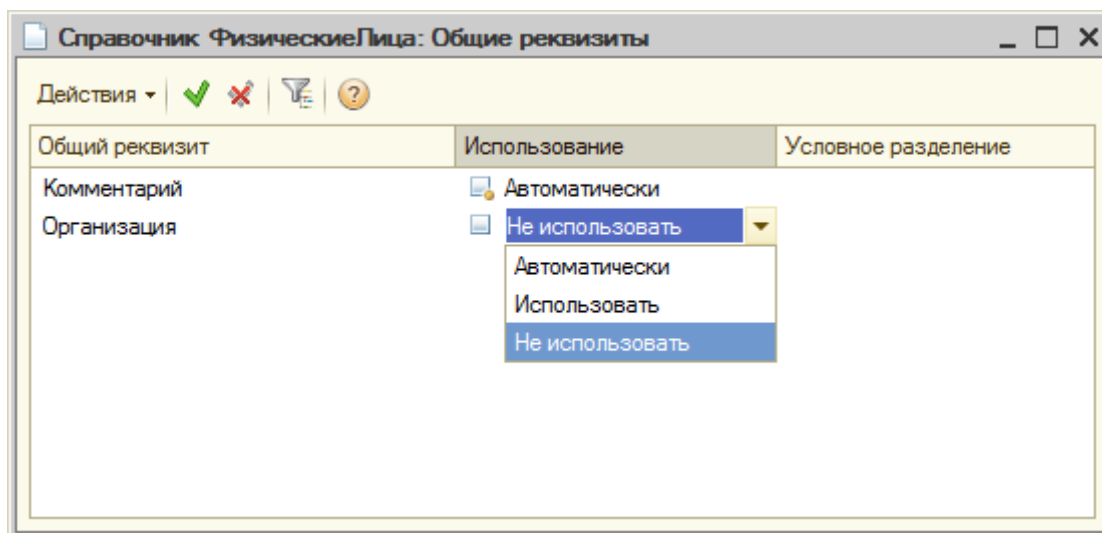


Рис. 3.9. Настройка общих реквизитов из окна редактирования объекта

Если мы попытаемся открыть справочник в режиме 1С:Предприятие – с ним можно будет работать, так как система автоматически сгенерирует его форму, рис. 3.10. – с автоматически созданными формами мы уже встречались ранее. Такие формы подходят в том случае, если мы не планируем каким-либо образом вмешиваться в функционирование формы из Конфигуратора.

Рис. 3.10. Форма справочника, сгенерированная автоматически

Если же решаемая нами задача требует каких-то особенных приемов работы с формой объекта, нам понадобится собственная форма. Например, это нам понадобится, если мы хотим автоматически заполнять поле **Наименование** на основе полей **Фамилия**, **Имя** и **Отчество**. А именно, мы хотели бы, чтобы наименование содержало фамилию и инициалы физического лица.

Разработка формы справочника ФизическиеЛица

Откроем закладку **Формы** окна редактирования справочника **ФизическиеЛица**. Можно отметить, рис. 3.11, что ни одной формы не задано – то есть все они создаются системой автоматически. Нам же нужна собственная форма элемента справочника.

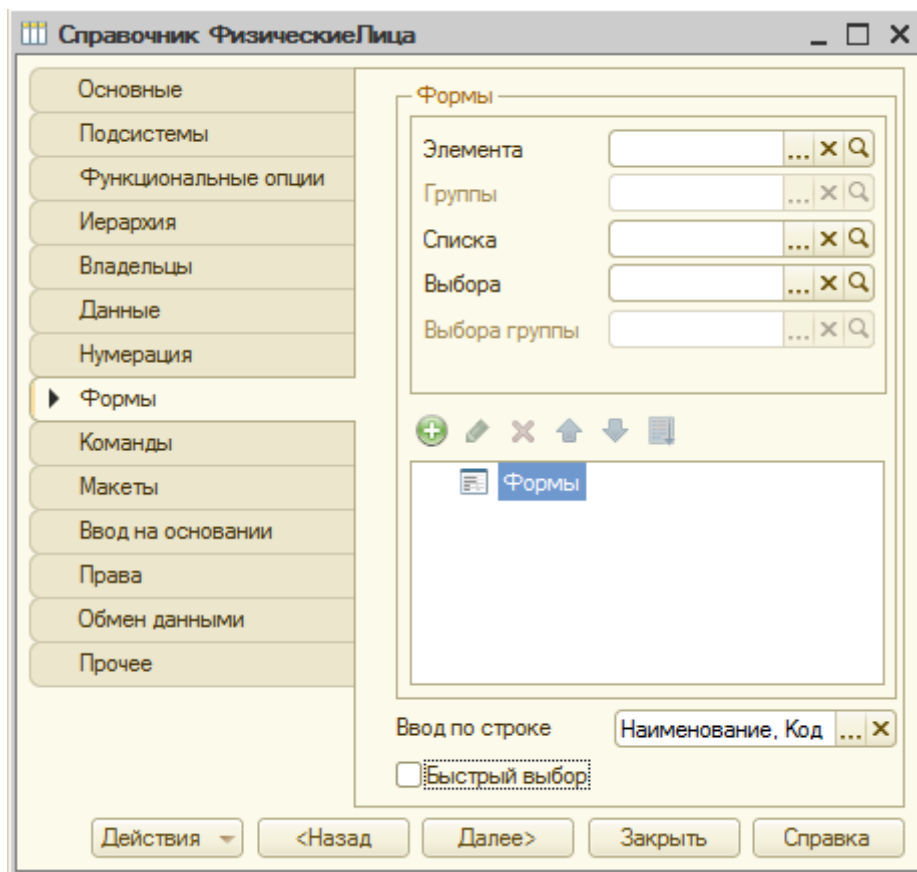


Рис. 3.11. Вкладка **Формы** окна редактирования объекта

Нажмем на кнопку с увеличительным стеклом напротив поля **Элемента** в группе **Формы**. Появится окно **Конструктора формы справочника**, в его первом окне оставим все по умолчанию – а именно – нас интересует **Форма элемента справочника**, рис. 3.12.

Конструктор формы справочника

Выберите тип формы:

- ☒ Форма элемента справочника
- ☐ Форма группы справочника
- ☐ Форма списка справочника
- ☐ Форма выбора справочника
- ☐ Форма выбора группы справочника
- ☐ Произвольная форма

☒ Назначить форму основной

☐ Основная форма элемента и группы

Имя: ФормаЭлемента

Синоним: Форма элемента

Комментарий:

< Назад Далее > Готово Отмена Справка

Рис. 3.12. Первое окно конструктора форм справочника

В следующем окне, рис. 3.13., мы можем указать состав реквизитов для расположения на форме, а так же указать количество колонок, которое нужно для расположения элементов управления на форме. Оставим здесь все так же по умолчанию и нажмем на кнопку **Готово**.

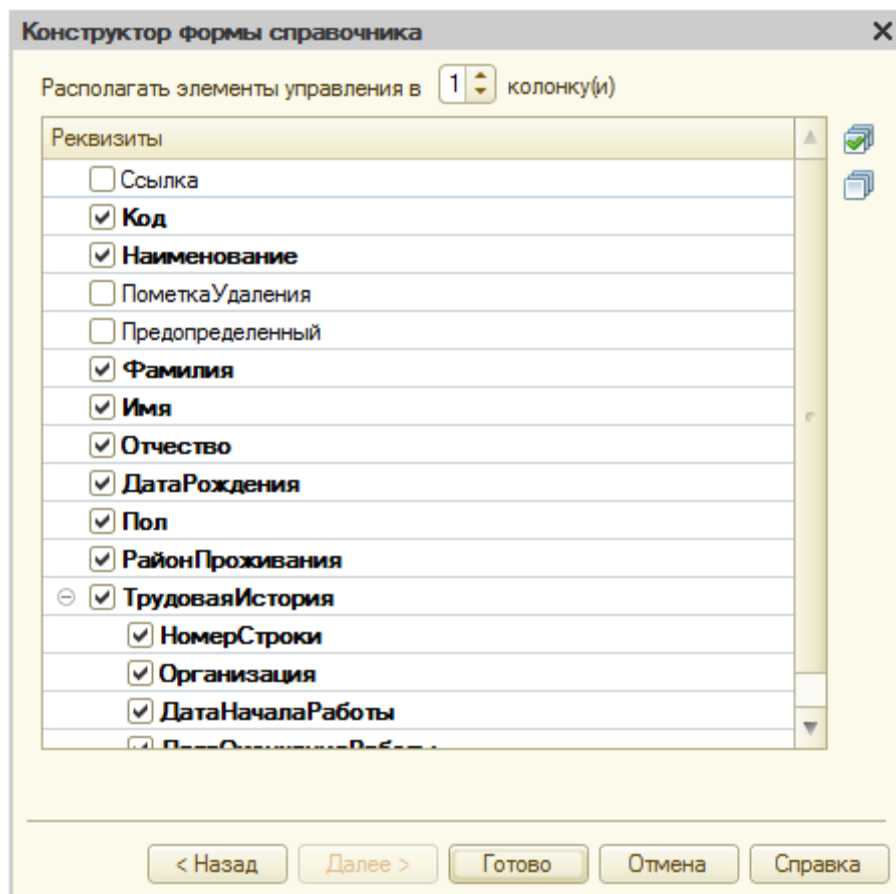


Рис. 3.13. Второе окно конструктора форм справочника

После этого нужно открыть окно редактора форм для формы элемента справочника, рис. 3.14. Ранее мы уже сталкивались с этим окном, теперь рассмотрим его подробнее.

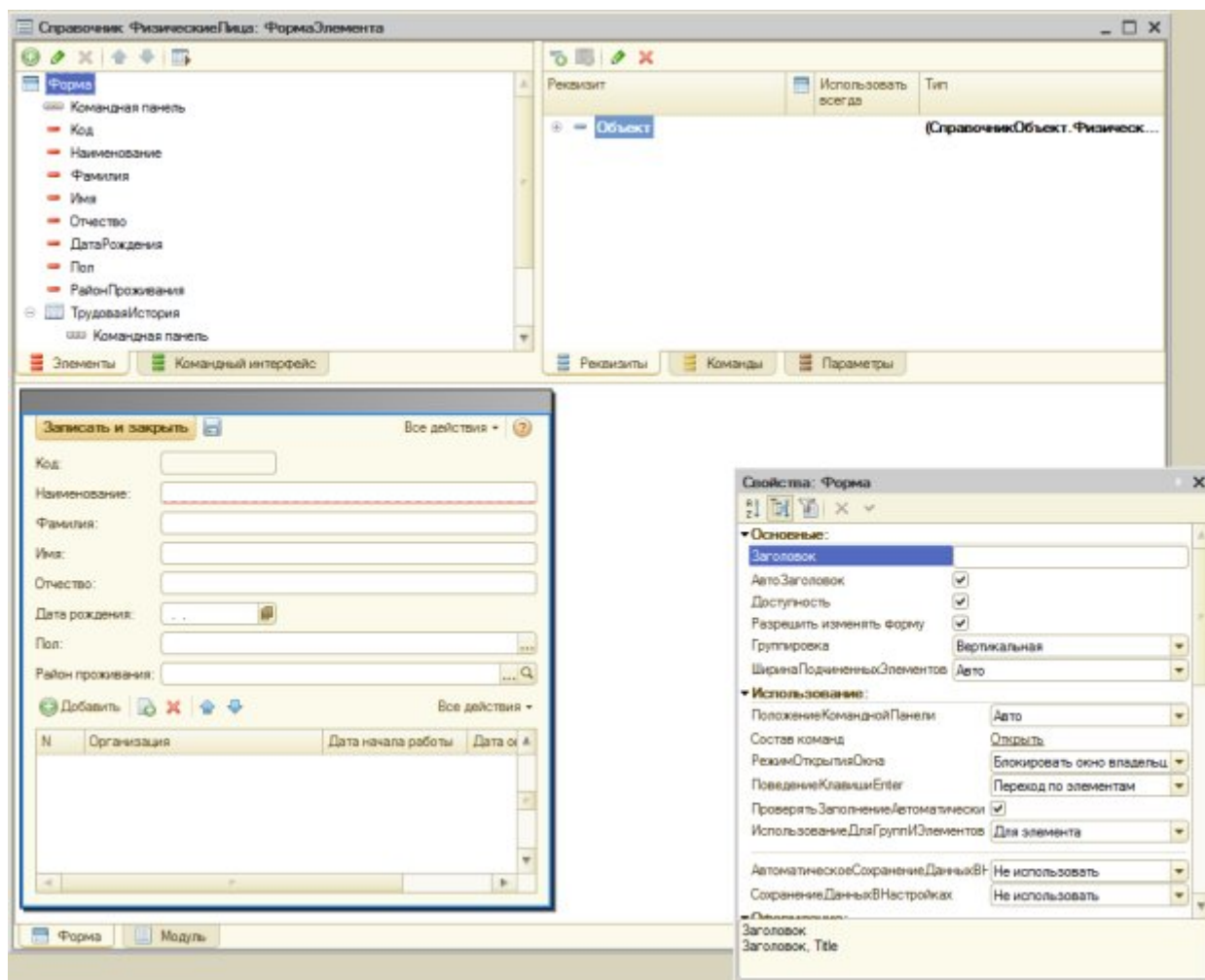


Рис. 3.14. Окно редактирования формы элемента справочника

На самом деле, это окно объединяет в себе несколько редакторов и окон. В частности, это следующие:

Редактор элементов формы (закладка **Элементы** в верхней левой части окна) – с его помощью можно контролировать элементы управления, которые будут расположены на форме. Выделив элемент в данном окне, мы можем настраивать его свойства в стандартной палитре свойств. Обратите внимание на кнопку **Проверить**, находящуюся в правой части командной панели закладки **Элементы**. Нажатие на нее приводит к выводу конструируемой формы в интерактивном виде, что позволяет лучше оценить ее внешний вид в пользовательском режиме, но, конечно, не дает возможности работать с данными информационной базы.

Окно просмотра формы (закладка **Форма** в нижней части окна) – здесь представлена форма в том виде, который она примет после настроек. Кроме того, выделяя элементы формы в данном окне, мы, не имея возможности, как это было ранее, произвольно перемещать их, можем вызывать их контекстное меню, рис. 3.14., с помощью которого можно перемещать элемент вверх или вниз (то же самое можно делать в окне **Элементы**), открывать окно его свойств, назначать обработчики событий (их можно назначать и в окне **Свойства**, открытом для данного элемента).

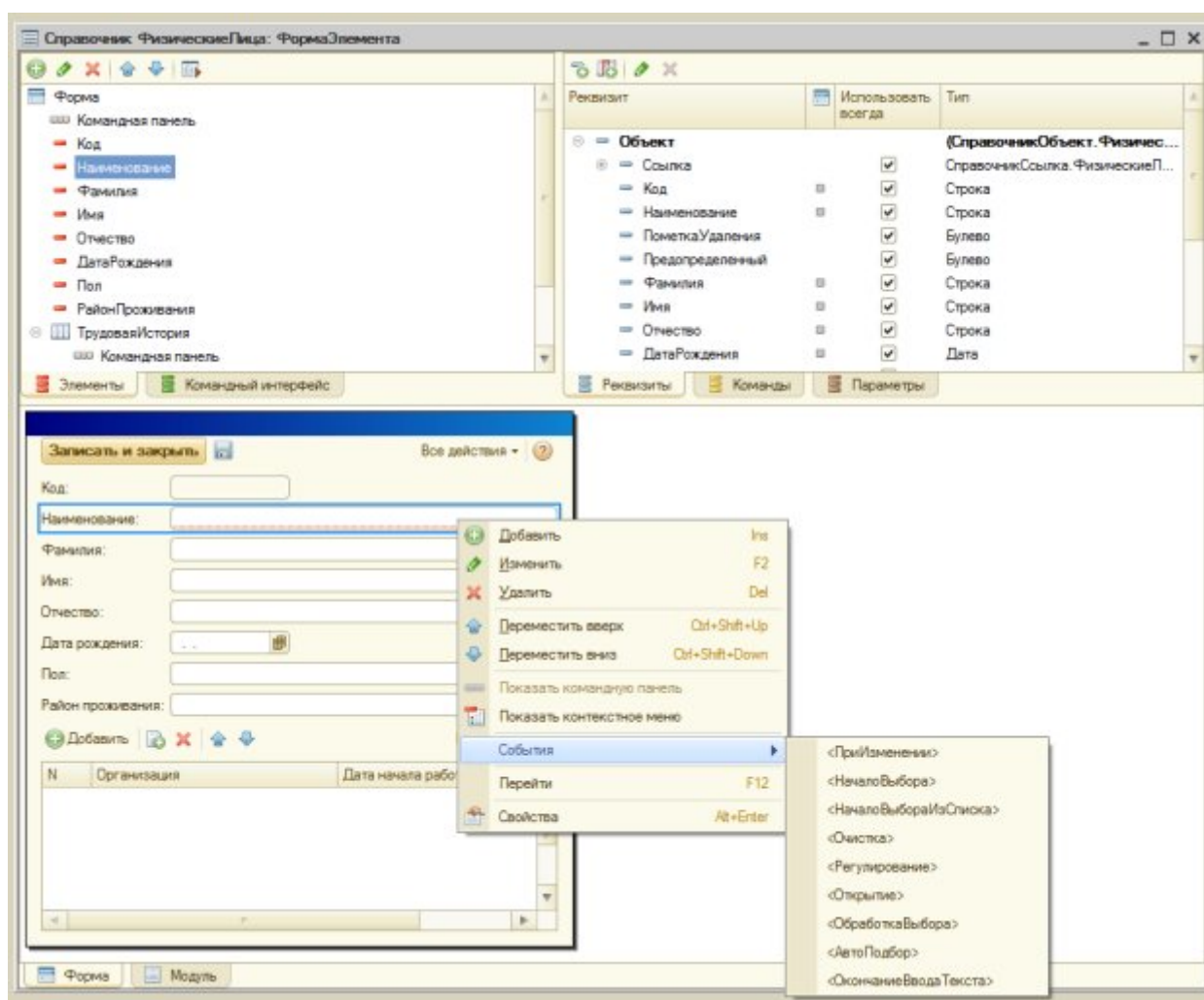


Рис. 3.15. Работа с элементами формы

Редактор реквизитов представлен вкладкой **Реквизиты** (рис. 3.15.). Для того, чтобы добавить реквизит объекта на форму (то есть – создать элемент управления, связанный с данным реквизитом), достаточно перетащить элемент из окна **Реквизиты** в окно **Элементы**. Реквизиты, уже присутствующие на форме, отмечены серым квадратиком.

Редактор команд можно открыть, нажав на вкладку **Команды**. Здесь доступны три дополнительные вкладки. Вкладка **Команды формы** (по умолчанию пустая) содержит команды формы, их можно сравнить с командными кнопками, которые в версии 1С:Предприятие 8.1. можно было размещать на форме. Теперь последовательность действий выглядит так – сначала создать команду формы, потом перетащить ее в окно **Элементы**, настроить свойства, задать обработчики событий. Вкладка **Стандартные команды** (рис. 3.16.) содержит стандартный набор команд – в нашем случае – стандартный для формы и табличного поля, размещенного на форме.

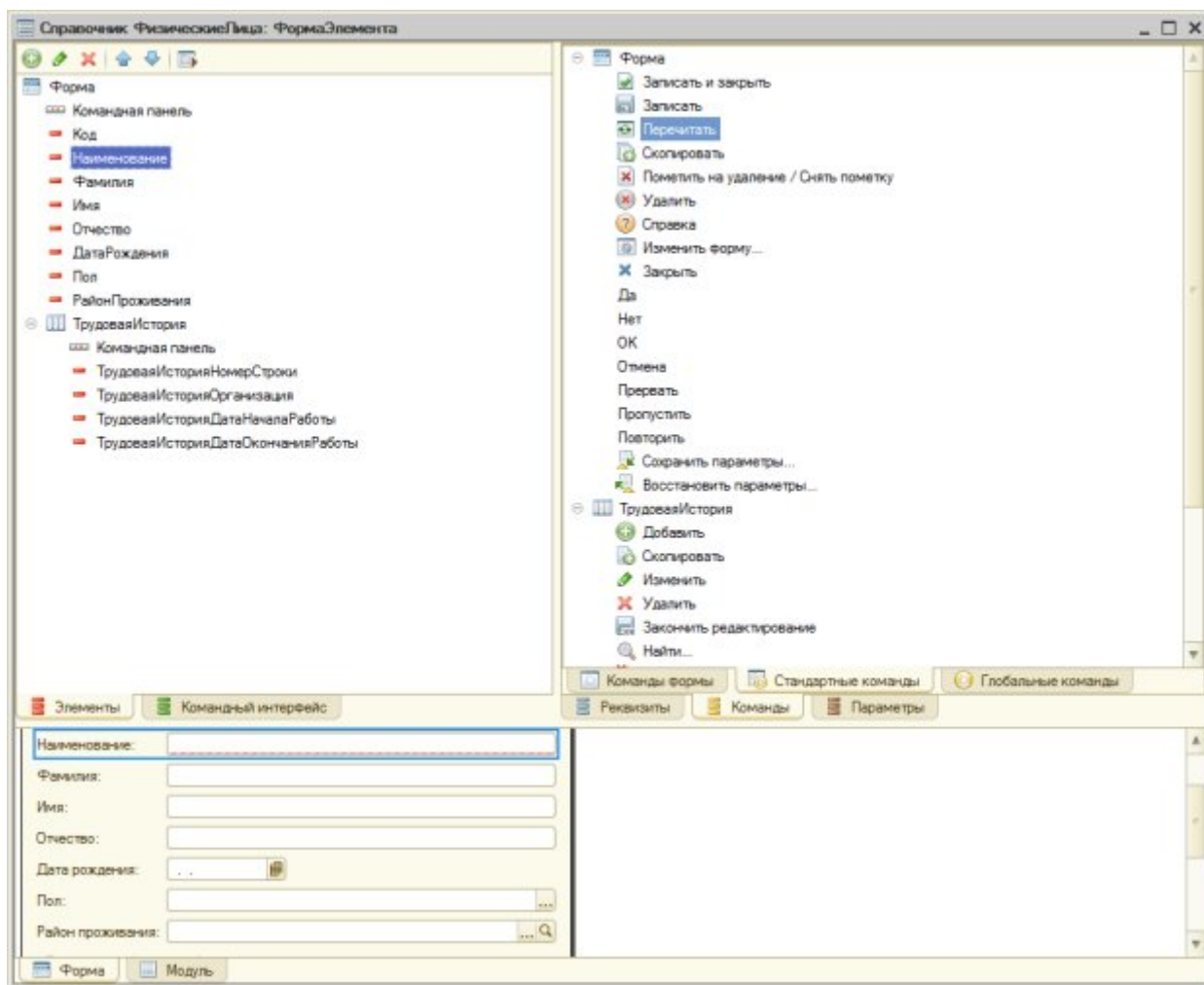


Рис. 3.16. Стандартные команды

Вкладка **Глобальные команды** содержит набор команд уровня прикладного решения.

Вкладка **Параметры** предоставляет доступ к редактору параметров.

Вкладка **Командный интерфейс** позволяет редактировать командный интерфейс.

Реализуем автоматическое заполнение поля **Наименование** на основе полей **Фамилия**, **Имя** и **Отчество**.

Для этого сначала настроим элемент управления, отображающий наименование на форме, таким образом, чтобы его нельзя было редактировать. Выделим элемент управления в панели **Элементы**, откроем окно его свойств и установим свойство **ТолькоПросмотр**, рис. 3.17.. Благодаря этому свойству пользователь не сможет отредактировать текст в поле ввода. Похожего эффекта можно достичь и другими способами, например, указав в свойстве **Вид элемента** элемента **Наименование** вместо **Поле ввода** – **Поле надписи**.

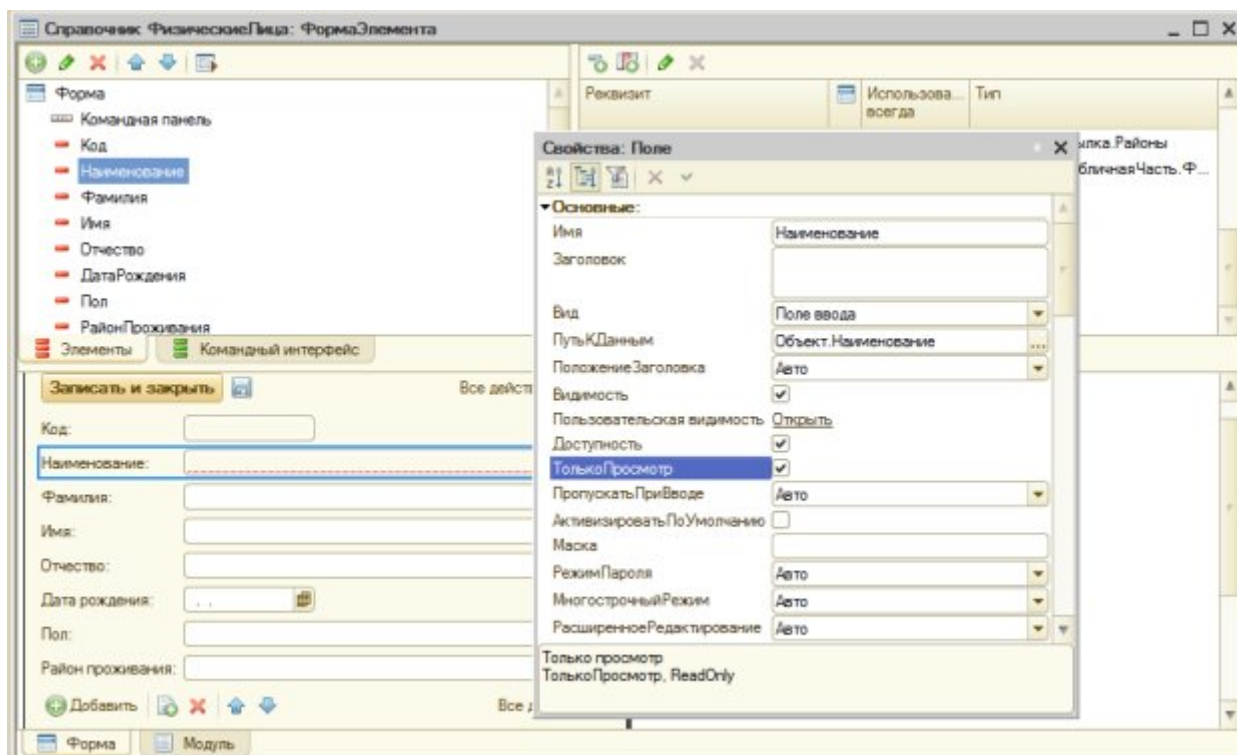


Рис. 3.17. Настройка элемента Наименование

Для правильного формирования наименования важно, чтобы пользователь ввел данные в поля **Фамилия**, **Имя** и **Отчество**.

Клиентские методы в модуле формы

Теперь перейдем к написанию кода, в котором будем формировать наименование. Для этого нам нужно понимать, что конфигурации 1С:Предприятие управляются событиями – и сейчас нас интересуют события формы.

Выделим форму в окне Элементы, откроем окно ее свойств и рассмотрим группу свойств **События**, рис. 3.18.. Наименование должно быть сформировано до того, как данные объекта будут записаны. Для достижения нашей цели нам вполне подойдет событие **ПередЗаписью**. Здесь же можно выполнить какие-либо пользовательские проверки полей перед формированием наименования. Хотя, если говорить о производительности решения, лучше подобные проверки производить на сервере, например, с помощью обработчика события **ОбработкаПроверкиЗаполнения**, который создается в модуле объекта.

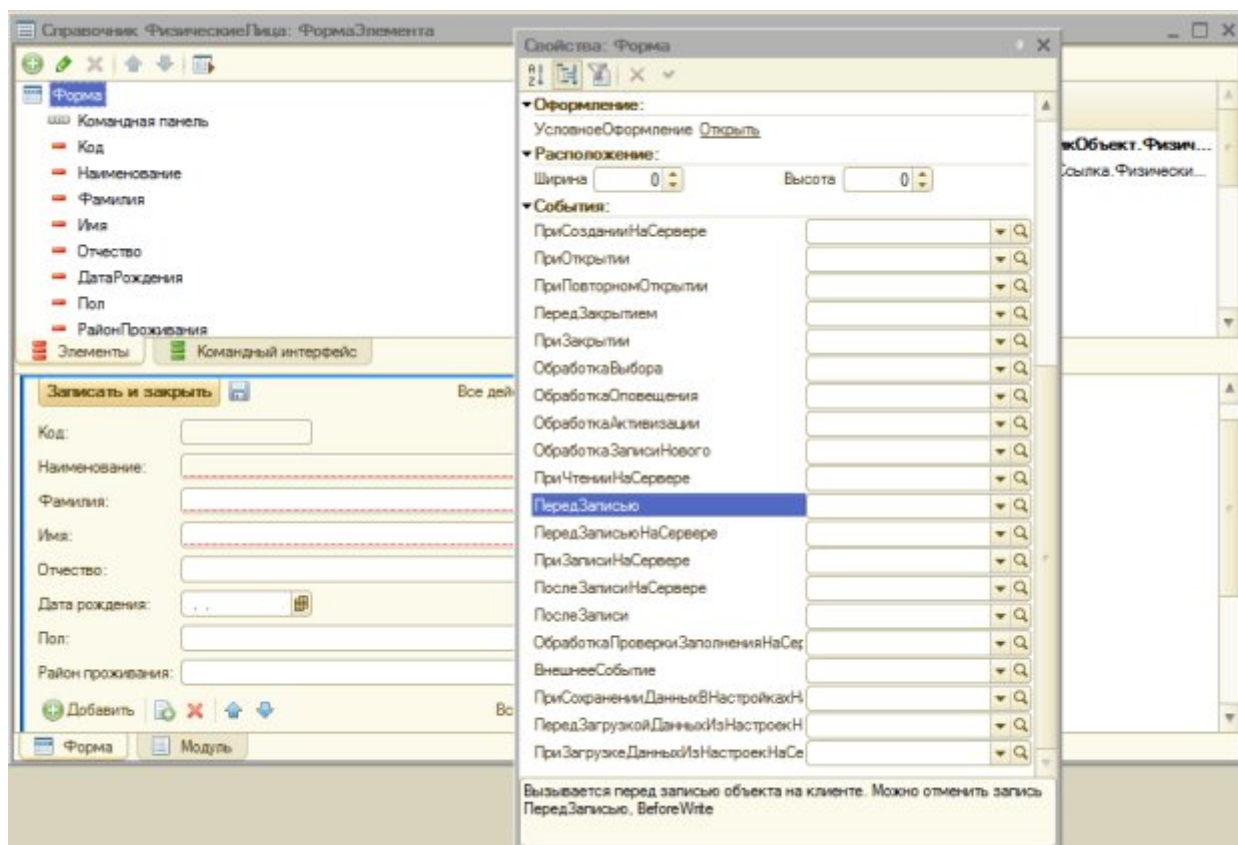


Рис. 3.18. Выбор события для выполнения запланированных действий

Нажмем на кнопку с увеличительным стеклом в поле события **ПередЗаписью** – автоматически будет открыт модуль формы и создан пустой обработчик события **ПередЗаписью**. Он имеет следующий вид:

```
&НаКлиенте
Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)
    // Вставить содержимое обработчика.
КонiecПроцедуры>
```

Из директивы компиляции **&НаКлиенте** понятно, что процедура это клиентская, она имеет два параметра – нас сейчас интересует параметр **Отказ** – благодаря этому параметру, а именно, установив его в значение **Истина**, мы можем отказаться от записи объекта в том случае, если выполняется какое-либо условие, препятствующее записи. В нашем случае записи объекта могут воспрепятствовать незаполненные или неправильно заполненные поля **Фамилия**, **Имя** или **Отчество**. Проверку на незаполненность реквизита мы можем доверить и системе – для этого можно установить свойство **Проверка заполнения** для нужных реквизитов в значение **Выдавать ошибку**, делается это в списке реквизитов объекта в окне редактирования объекта или в дереве конфигурации, рис. 3.19. Не будем включать проверку заполнения, выполним ее и еще некоторые проверки самостоятельно.

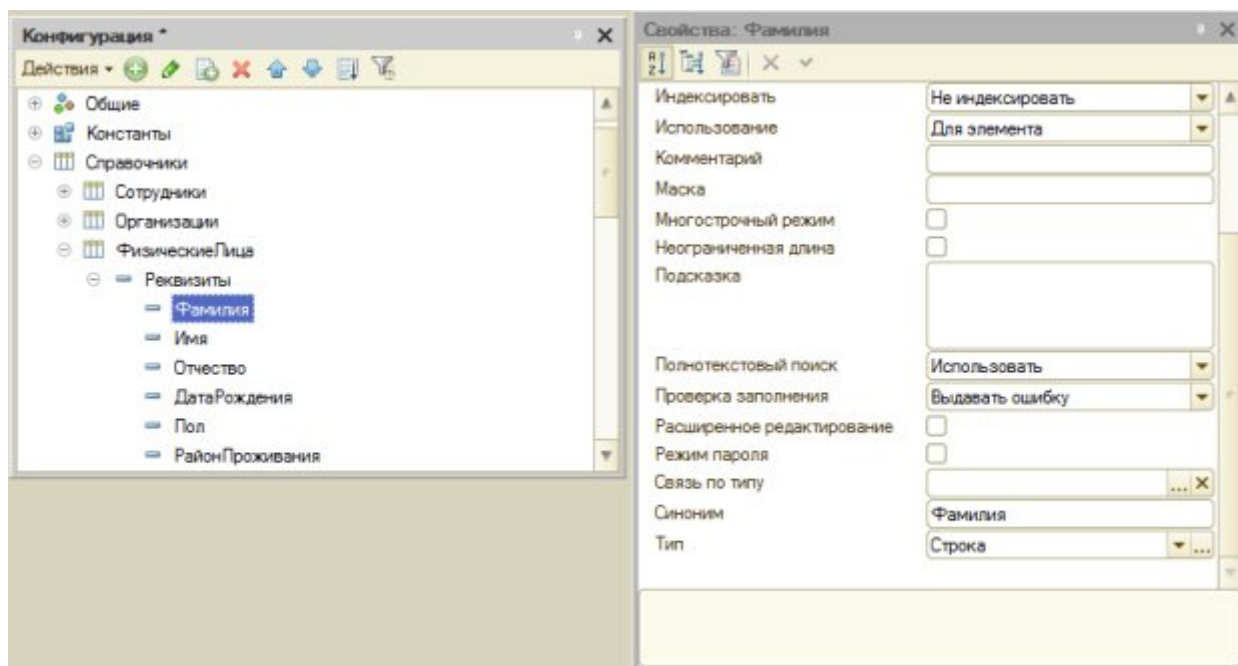


Рис. 3.19. Настройка проверки заполнения

Параметры и процедуры в системе 1С:Предприятие по умолчанию передаются по ссылке – передав в процедуру некую переменную, мы, на самом деле, передаем ссылку на нее, то есть – при модификации соответствующего этой переменной параметра внутри процедуры, фактически, происходит и модификация переменной. Вернемся к нашей процедуре **ПередЗаписью**.

В этой процедуре мы сначала проверим поля **Фамилия**, **Имя** и **Отчество** на заполненность (возможны и более сложные проверки), после чего, если хотя бы одно поле не заполнено – сообщим об этом пользователю и выйдем из процедуры, если все поля заполнены – сформируем наименование. Вот какой код позволяет реализовать эту задачу:

```
&НаКлиенте
Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)
//Переменная для хранения текста сообщения пользователю
Перем ТекстСообщения;
//Запишем пустую строку в переменную
ТекстСообщения="";
//Если не введена фамилия...
Если ПустаяСтрока(Объект.Фамилия) Тогда
//Формируем строку сообщения
ТекстСообщения=ТекстСообщения+"Не заполнено поле Фамилия;";
КонецЕсли;
//Если не введено имя...
Если ПустаяСтрока(Объект.Имя) Тогда
ТекстСообщения=ТекстСообщения+" Не заполнено поле Имя;";
КонецЕсли;
//Если не введено отчество...
Если ПустаяСтрока(Объект.Отчество) Тогда
ТекстСообщения=ТекстСообщения+" Не заполнено поле Отчество;";
КонецЕсли;
//Если строка сообщения не пуста, то есть - содержит
//сообщения о незаполненных полях
Если НЕ ПустаяСтрока(ТекстСообщения) Тогда
//Выводим сообщение
Сообщить(ТекстСообщения);
```

```

//Отказываемся от записи объекта
Отказ=Истина;
//Выходим из процедуры
Возврат;
КонецЕсли;
//Если все поля заполнены, выхода из процедуры не произошло,
//формируем наименование
Объект.Наименование=Объект.Фамилия+" "+ ВРег (Лев (Объект.Имя, 1) ) +".
"+ВРег (Лев (Объект.Отчество, 1) ) +". ";
КонецПроцедуры

```

Строковая функция **Лев** позволяет получить заданное количество символов из строки, начиная с самого левого. Строковая функция **ВРег** переводит символы в верхний регистр – на тот случай, если пользователь случайно ввел имя, фамилию или отчество с маленькой буквы. Конечно, здесь можно предусмотреть еще множество проверок и автоматических корректировок (например, можно исправить первую букву во введенных фамилии, имени и отчестве, если она случайно введена в нижнем регистре), мы ограничимся тем, что сделано сейчас.

В итоге мы получаем следующие сообщения об ошибках при незаполненности полей, рис. 3.20.

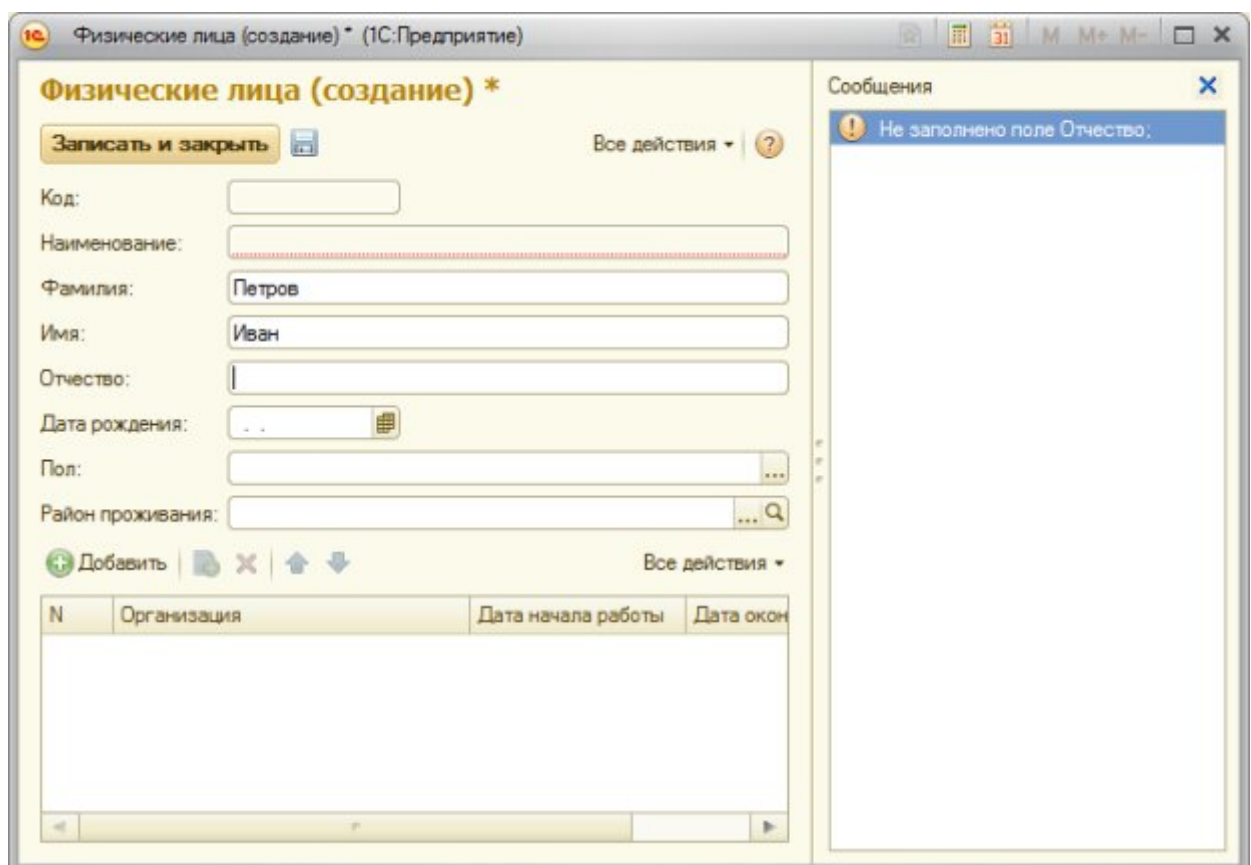


Рис. 3.20. Сообщение об ошибке

После успешного выполнения процедуры **ПередЗаписью**, наименование выглядит следующим образом. Мы намеренно ввели отчество с маленькой буквы – как было пояснено выше, наш код готов к такому повороту событий, рис. 3.21.

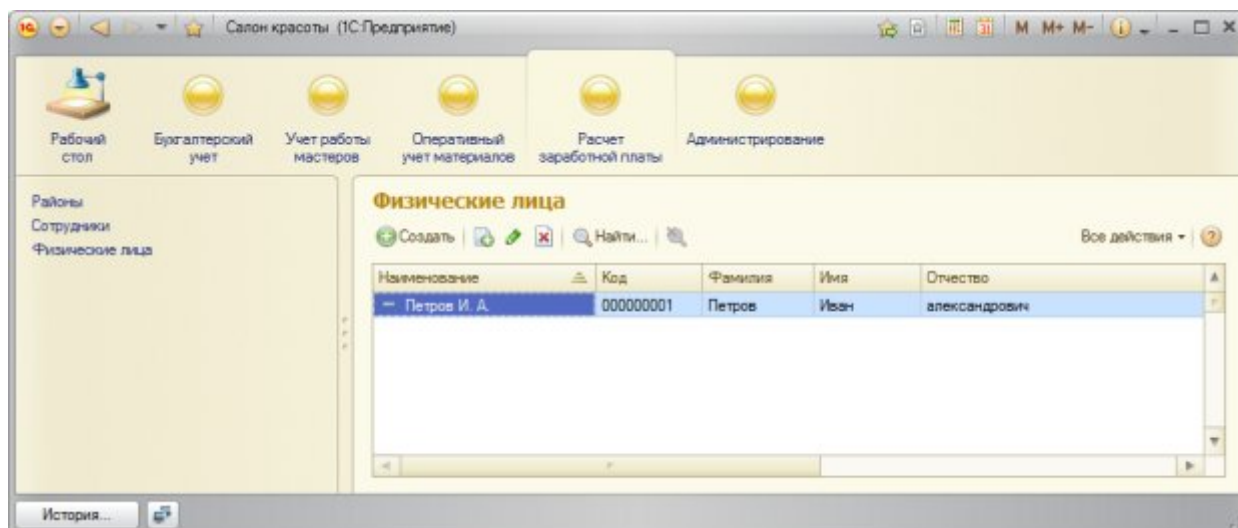


Рис. 3.21. Новая запись в справочнике Физические лица

Объект СообщениеПользователю

Обратите внимание на то, что здесь мы пользуемся обычным методом **Сообщить** – мы выводим в окно сообщения одно сообщение, содержащее необходимые сведения. В 1С:Предприятие 8.3. мы можем поступить по-другому – вывести сообщения об ошибках или другие сведения, "привязав" их к полям, которые вызвали ошибки. Для этого можно воспользоваться объектом **СообщениеПользователю**. Он, помимо прочих полезных возможностей, позволяет формировать сообщения и "привязывать" их к реквизитам формы. Перепишем код таким образом, чтобы сообщения об ошибках (то есть, о незаполненных полях **Фамилия**, **Имя**, или **Отчество**), выявленных в процедуре **ПередЗаписью**, выводились бы в привязке к соответствующим элементам формы. Вот какой код позволяет этого добиться:

```
&НаКлиенте
Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)
//Если не введена фамилия...
Если ПустаяСтрока(Объект.Фамилия) Тогда
    СообщитьПользователю("Объект.Фамилия", "Заполните поле Фамилия", Отказ);
КонецЕсли;
//Если не введено имя...
Если ПустаяСтрока(Объект.Имя) Тогда
    СообщитьПользователю("Объект.Имя", "Заполните поле Имя", Отказ);
КонецЕсли;
//Если не введено отчество...
Если ПустаяСтрока(Объект.Отчество) Тогда
    СообщитьПользователю("Объект.Отчество", "Заполните поле Отчество", Отказ);
КонецЕсли;
//Если флаг Отказ не был установлен - формируем наименование
Если НЕ Отказ Тогда
    Объект.Наименование=Объект.Фамилия+" "+ ВРег(Лев(Объект.Имя,1))+".
"+ВРег(Лев(Объект.Отчество,1))+".";
КонецЕсли;

КонецПроцедуры
```

```
&НаКлиенте
//Процедура, формирующая и выводящая сообщение с переданными ей параметрами
Процедура СообщитьПользователю(ПутьКРеквизиту, Текст, Отказ)
    Сообщение=Новый СообщениеПользователю;
```

```
Сообщение.Поле=ПутьКРеквизиту;  
Сообщение.Текст=Текст;  
Сообщение.Сообщить ( ) ;  
Отказ=Истина;  
КонецПроцедуры
```

Поясним приведенный код. Для начала, мы создали новую клиентскую процедуру **СообщитьПользователю**. Эта процедура принимает три параметра. Первый – **ПутьКРеквизиту** содержит строковый путь к полю, к которому должно быть привязано сообщение. Второй – **Текст** – содержит текст сообщения. Третий – **Отказ** – используется для установки в значение **Истина** параметра **Отказ** процедуры **ПередЗаписью** в том случае, если процедура **СообщитьПользователю** будет вызвана хотя бы один раз. А хотя бы однократный ее вызов означает, что одно из полей не заполнено, то есть наименование сформировать невозможно, соответственно, записать объект так же не получится.

Когда процедура вызывается, мы сначала создаем новый объект типа **СообщениеПользователю**. Затем его свойство **Поле** устанавливаем в значение параметра **ПутьКРеквизиту**. Этот параметр должен быть строковым и имеет, в нашем случае вид "Объект.Фамилия", "Объект.Имя", "Объект.Отчество" - это позволяет правильно "привязать" сообщение к полям формы. Свойство **Текст** объекта **СообщениеПользователю** содержит текст для вывода.

Мы, кроме того, полностью переработали процедуру **ПередЗаписью**. А именно, если проверка на заполнение поля указывает на то, что поле пустое, вызывается процедура **СообщитьПользователю**. По окончании проверок мы проверяем, установлен ли параметр **Отказ** в значение **Истина** – если не установлен – ни одна из проверок не завершилась обнаружением пустого поля и мы можем формировать наименование. Если установлен – наименование мы не формируем – и процедура заканчивает работу, а записи объекта, естественно, не происходит – пользователь видит лишь сообщения об ошибках.

Если было сформировано несколько сообщений типа **СообщениеПользователю** – пользователь видит одно окно сообщения около поля, но это окно снабжено кнопками для перемещения вперед и назад – щелчок по кнопке приводит к "переходу" сообщения от одного поля с ошибкой к другому, рис. 3.22, 3.23.

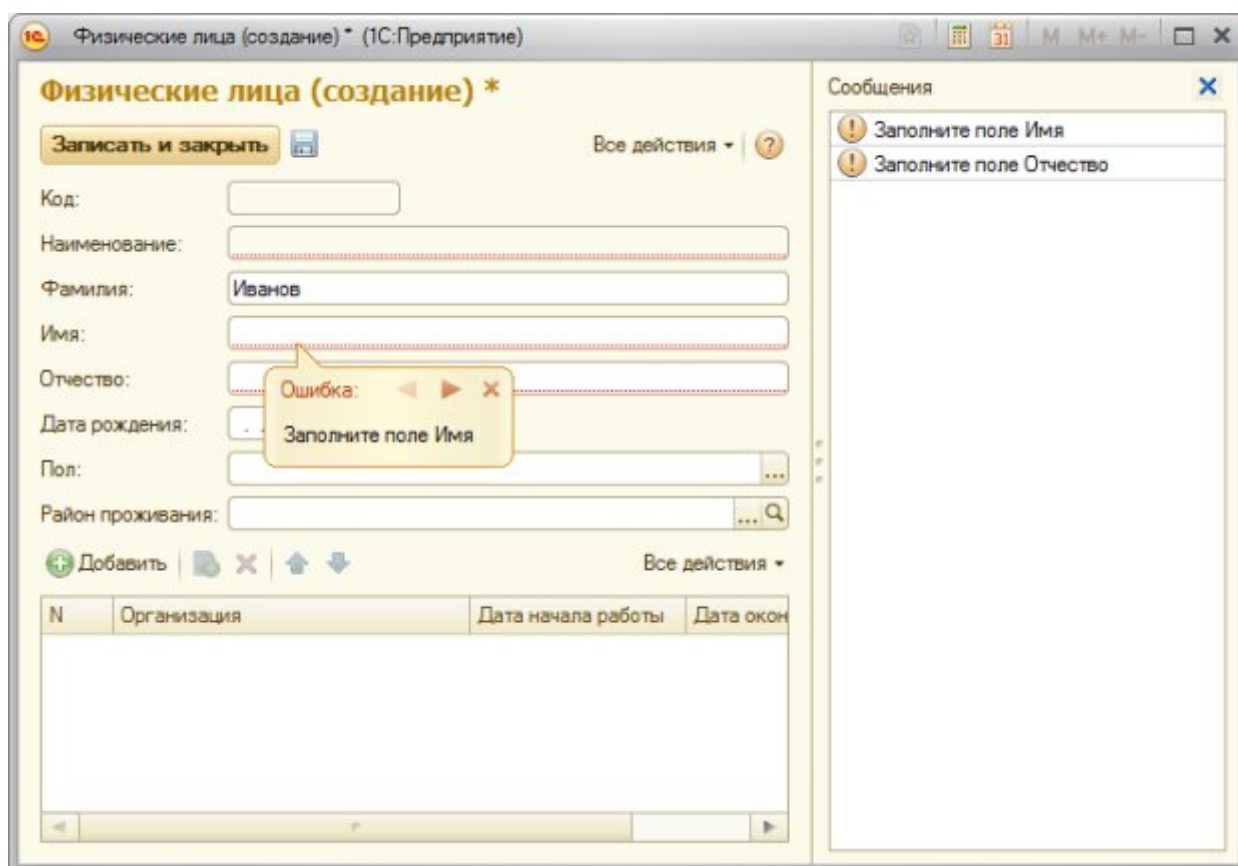


Рис. 3.22. Сообщение об ошибке, привязанное к полю Имя

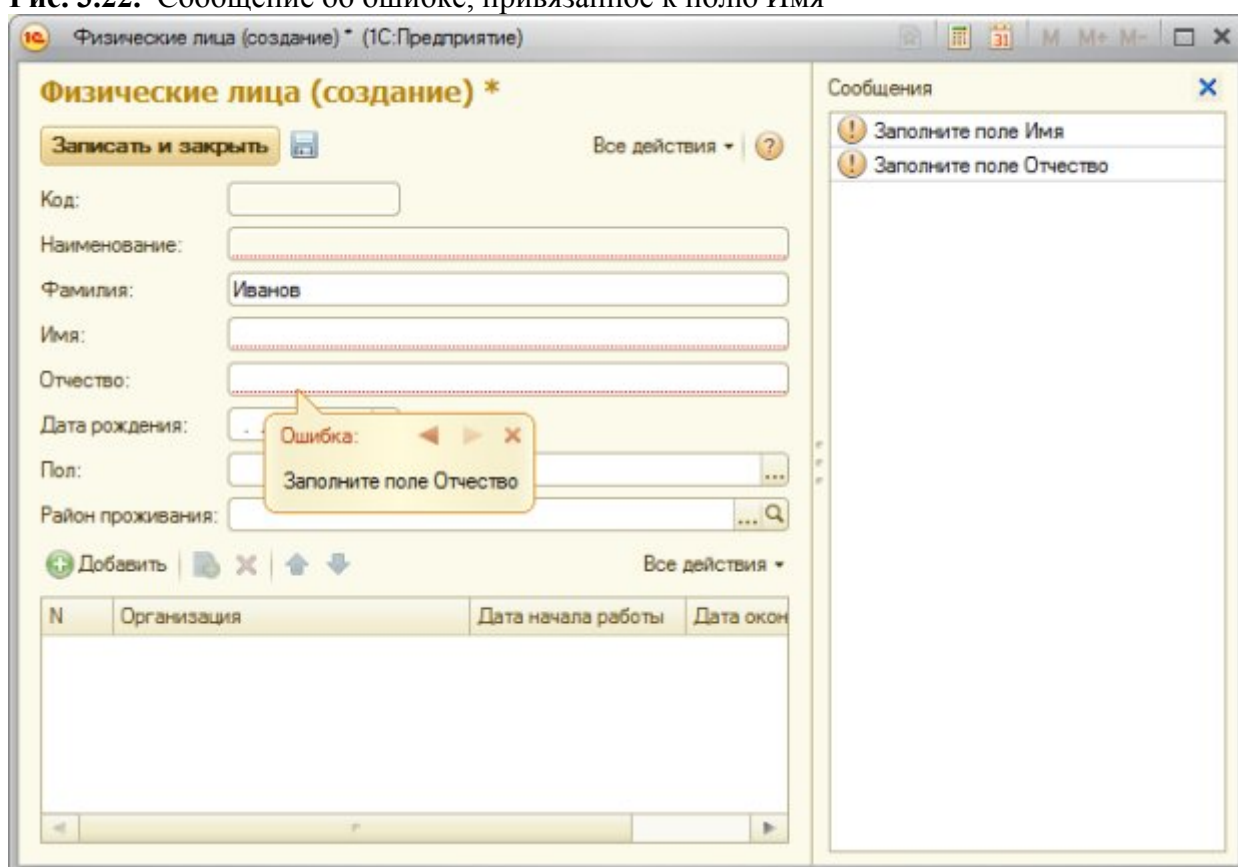


Рис. 3.23. Сообщение об ошибке, привязанное к полю Отчество

Доведем до логического завершения пример со справочником **ФизическиеЛица**. Для этого заполним справочник Районы и введем в информационную базу сведения о следующих физических лицах:

Фамилия	Имя	Отчество	Дата рождения	Пол	Район
Иванов	Иван	Иванович	27.02.1984	Мужской	Ленинский
Петров	Петр	Петрович	12.06.1985	Мужской	Ленинский
Васильев	Павел	Петрович	17.05.1985	Мужской	Ленинский
Расчетчиков	Александр	Иванович	12.03.1980	Мужской	Октябрьский
Александров	Александр	Александрович	17.09.1970	Мужской	Октябрьский
Бухгалтерова	Василиса	Владимировна	13.08.1976	Женский	Уральский

Обратите внимание на то, что справочник **ФизическиеЛица** – это пример справочника, с которым пользователям нашей информационной базы придется работать достаточно часто. В данный момент для того, чтобы создать новый элемент справочника, нам нужно выполнить несколько действий – перейти в раздел **Расчет заработной платы**, щелкнуть по ссылке, открывающей список справочника, после чего нажать на кнопку **Создать новый элемент списка**. Для того, чтобы сократить количество действий, необходимых для выполнения часто используемых операций, мы можем соответствующим образом настроить интерфейс нашего прикладного решения, в частности, поработать с панелью действий соответствующего раздела и с **Рабочим столом**.

Настройка командного интерфейса для ускорения доступа к справочнику

Добавим команду создания нового элемента справочника **ФизическиеЛица** в панель действий раздела **Расчет заработной платы**. Для этого откроем окно Все подсистемы командой контекстного меню ветви Подсистемы дерева конфигурации и установим флаг **Видимость** напротив команды **Физические лица: Создать** в области **Панель действий.Создать**, рис. 3.24.

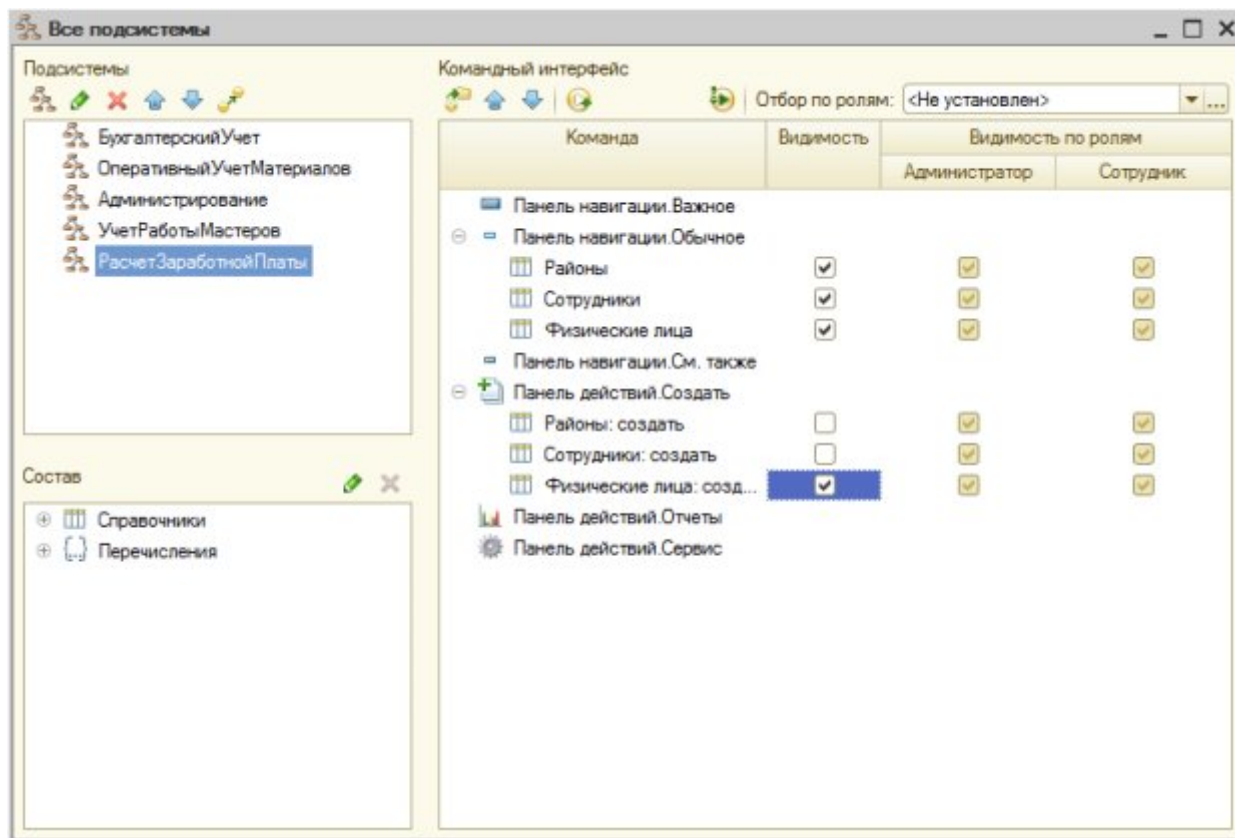


Рис. 3.24. Настройка панели действий раздела Расчет заработной платы

Мы можем включить команду добавления нового физического лица в командный интерфейс **Рабочего стола**.

Для этого выполним команду контекстного меню корневого элемента конфигурации **Открыть командный интерфейс рабочего стола**

Выделим в поле **Доступные команды** команду **Физические лица: создать**, в поле состава командного интерфейса – команду **Панель действий.Создать** и нажмем на кнопку со значком ">" (Добавить команду на рабочий стол), которая находится между полями, после чего установим флаг Видимость для добавленной команды, рис 3.25.

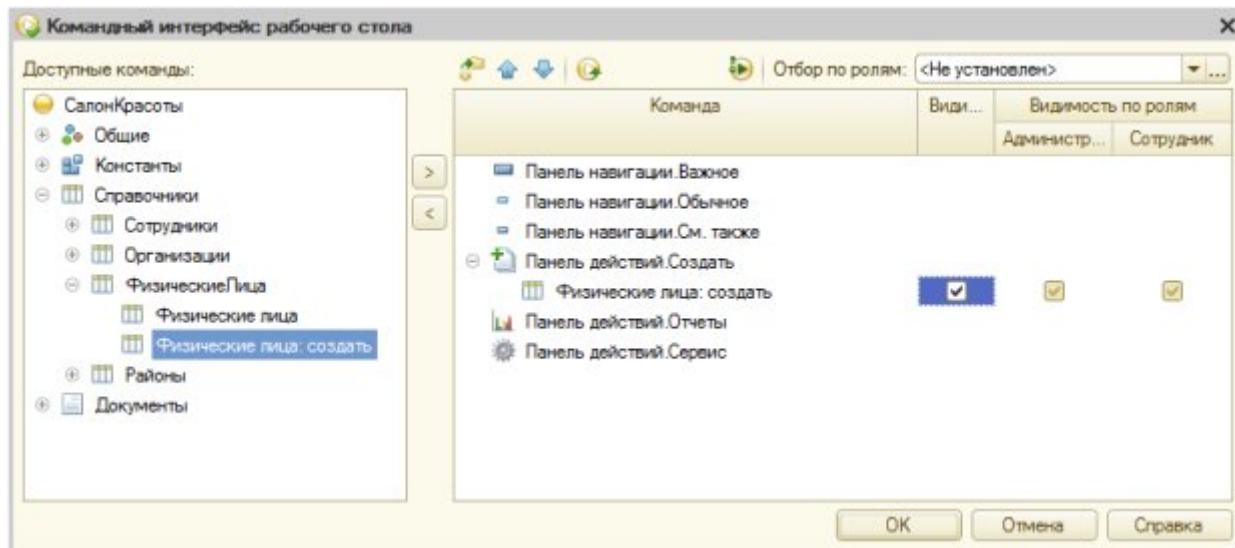


Рис. 3.25. Настройка панели действий Рабочего стола

Теперь, рис. 3.26., команда для быстрого создания элементов справочника **Физические Лица** добавлена в панель действий **Рабочего стола**, аналогичная команда появилась в разделе **Расчет заработной платы**.

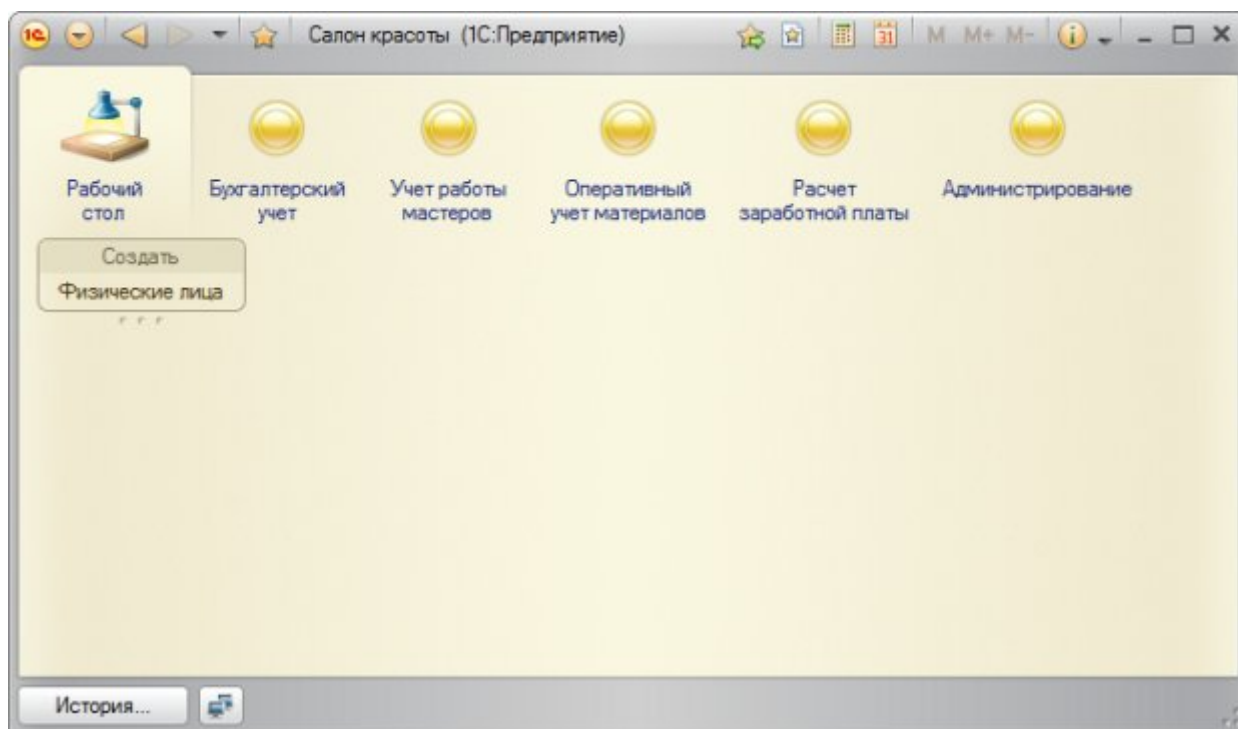


Рис. 3.26. Новая команда в панели действий рабочего стола