**Лабораторная работа №16**

**Тема**: Справочники

**Цель работы:** Исследовать работу со справочниками конфигурации и рассмотреть некоторые программные механизмы работы с ними

**Теоретические сведения**

### О различных видах справочников

В этой лабораторной работе мы продолжим работу над справочниками нашей конфигурации, попутно рассмотрим некоторые программные *механизмы* работы с ними.

В частности, в предыдущей работе мы создавали обычные справочники – такие справочники содержат простые списки элементов. Однако, справочники в 1С:Предприятие 8 могут иметь более сложную структуру.

Многие справочники, используемые на практике, являются иерархическими. Это означает, что каждому из элементов справочника может быть сопоставлен другой элемент, который называется родителем.

Возможна *иерархия* различных видов, в частности, при иерархии групп и элементов в справочник, помимо обычных элементов можно включать группы, которые, в свою *очередь*, могут включать в себя другие группы и элементы.

Примером такого справочника является справочник, хранящий списки товаров, материалов, услуг. Обычно такой справочник носит название Номенклатура. Как правило, подобный справочник имеет множество групп – например – Товары – для хранения записей о товарно-материальных ценностях, и Услуги – для хранения списков услуг.

Второй вид иерархии – это *иерархия* элементов. Хорошим примером справочника, для которого естественна *иерархия* элементов, является справочник для хранения сведений о подразделениях организаций. Такой справочник обычно так и называется – *Подразделения*. Логика иерархии элементов заключается в том, что отдельные элементы справочника, описывающие отдельные *подразделения* организации, могут являться родителями для других элементов – так же подразделений. Например, подразделение Администрация вполне может включать в себя *подразделения* Бухгалтерия, Отдел кадров и так далее.

Справочники, в сущности, являются хранилищами аналитических признаков учета. Например, в случае со справочником Номенклатура можно организовать учет таким образом, чтобы иметь сведения об остатках каких-либо номенклатурных позиций на складе, об их стоимости. В случае со справочником, поддерживающим иерархию элементов, любой элемент справочника можно использовать как аналитический "разделитель" учета (то есть, например, выбирать данные элементы при заполнении табличных частей других объектов, например, документов). А вот при иерархии групп и элементов использовать в качестве аналитического разделителя группу не получится.

Если родительские отношения существуют внутри справочника, такие справочники называются иерархическими. Существует еще один вид взаимоотношений между справочниками, который называется подчинением. Предположим, у нас имеется справочник Контрагенты, содержащий *список* организаций, с которыми наша организация имеет какие-то взаимоотношения. Каждый из контрагентов, описанных в этом справочнике, имеет некоторое количество контактных лиц – представителей контрагента. Можно сказать, что каждый представитель контрагента "принадлежит" определенному контрагенту. Такие взаимоотношения между справочниками реализуются при помощи механизма указания владельцев справочника и настройки подчинения.

**Ход работы:**

Рассмотрим работу с *иерархическими справочниками*.

### Иерархические справочники

Создадим новый справочник **Единицы измерения**, зададим следующие его параметры:

**Имя**: ЕдиницыИзмерения

**Длина наименования**: 100 символов

**Подсистемы**: БухгалтерскийУчет, ОперативныйУчетМатериалов

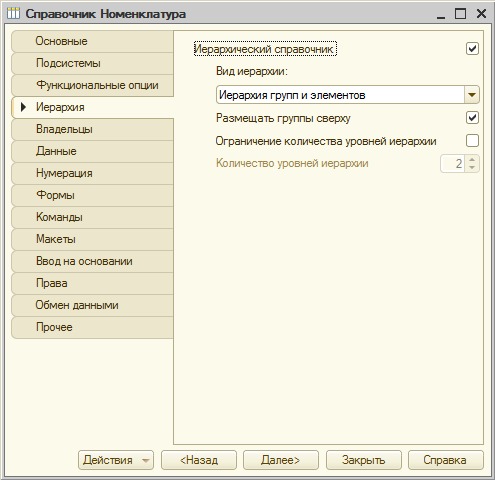
Это будет очень простой справочник, стандартный *реквизит* которого **Наименование** будет использоваться для хранения информации о наименовании единицы измерения.

Теперь создадим очередной справочник – **Номенклатура**. Зададим следующие параметры:

**Имя**: Номенклатура

**Подсистемы**: БухгалтерскийУчет, ОперативныйУчетМатериалов

На вкладке окна редактирования объекта **Иерархия**, [рис. 4.1](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=1#image.4.1)., установим следующие параметры:



**Рис. 4.1.**Настройка иерархического справочника

**Иерархический справочник**: Установлено

**Вид иерархии**: *Иерархия* групп и элементов.

Этот *параметр* может принимать *значение* **Иерархия элементов**. В нашем случае справочник сможет содержать отдельные элементы, собранные, в зависимости от их вида, в группы. Эту структуру можно сравнить с папками и файлами в файловой системе компьютера. Группы – это папки, отдельные элементы – это файлы.

На вкладке **Данные**, [рис. 4.2](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=1#image.4.2)., добавим следующие реквизиты:

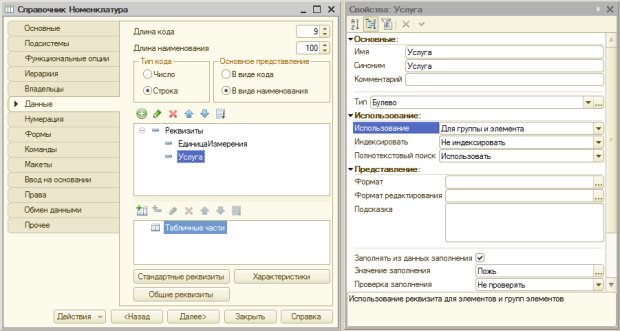
**ЕдиницаИзмерения**: Тип СправочникСсылка.ЕдиницыИзмерения.

**Услуга**: **Тип** Булево, **Использование**: Для группы и элемента. Эта установка позволит задавать данный *реквизит* и для элементов и для групп.

**Заполнять из данных заполнения**: *Истина*

Отдельные группы нашего справочника планируется использовать для хранения исключительно услуг, и подобная установка (в частности, истинность параметра **Заполнять из данных заполнения**) позволит нам реализовать автоматический механизм заполнения данного реквизита для элементов, входящих в группы.

**Длина наименования**: 100

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/4/files/04_02.jpg)

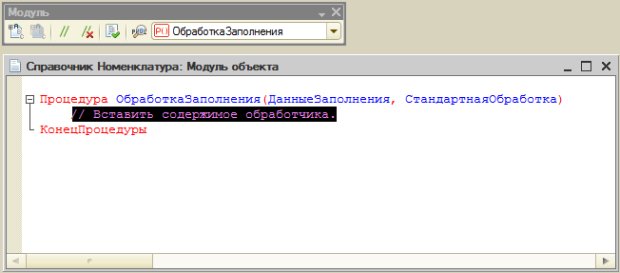
**Рис. 4.2.**Состав реквизитов справочника Номенклатура

Таким образом, при создании элемента справочника мы будем задавать название элемента в стандартном *реквизите* **Наименование**, указывать единицу измерения, а так же, для услуг, устанавливать флаг **Услуга**, причем, установка этого флага для группы будет означать, что в ней хранятся списки услуг, а для элемента – то, что он является услугой.

### Автозаполнение реквизитов

Реализуем функцию автоматического заполнения реквизита **Услуга** для элементов, входящих в группы. Нам нужно, чтобы элемент, создаваемый в группе с установленным флагом **Услуга**, при его создании, автоматически бы получал установленный флаг **Услуга**, соответственно, если данный флаг у группы не установлен, у элемента он так же не должен быть установлен. При этом нам нужно предусмотреть ситуацию, когда элемент создается вне группы – на верхнем уровне справочника **Номенклатура**. Для решения этой задачи мы можем воспользоваться обработчиком события **ОбработкаЗаполнения**, его процедура располагается в модуле объекта.

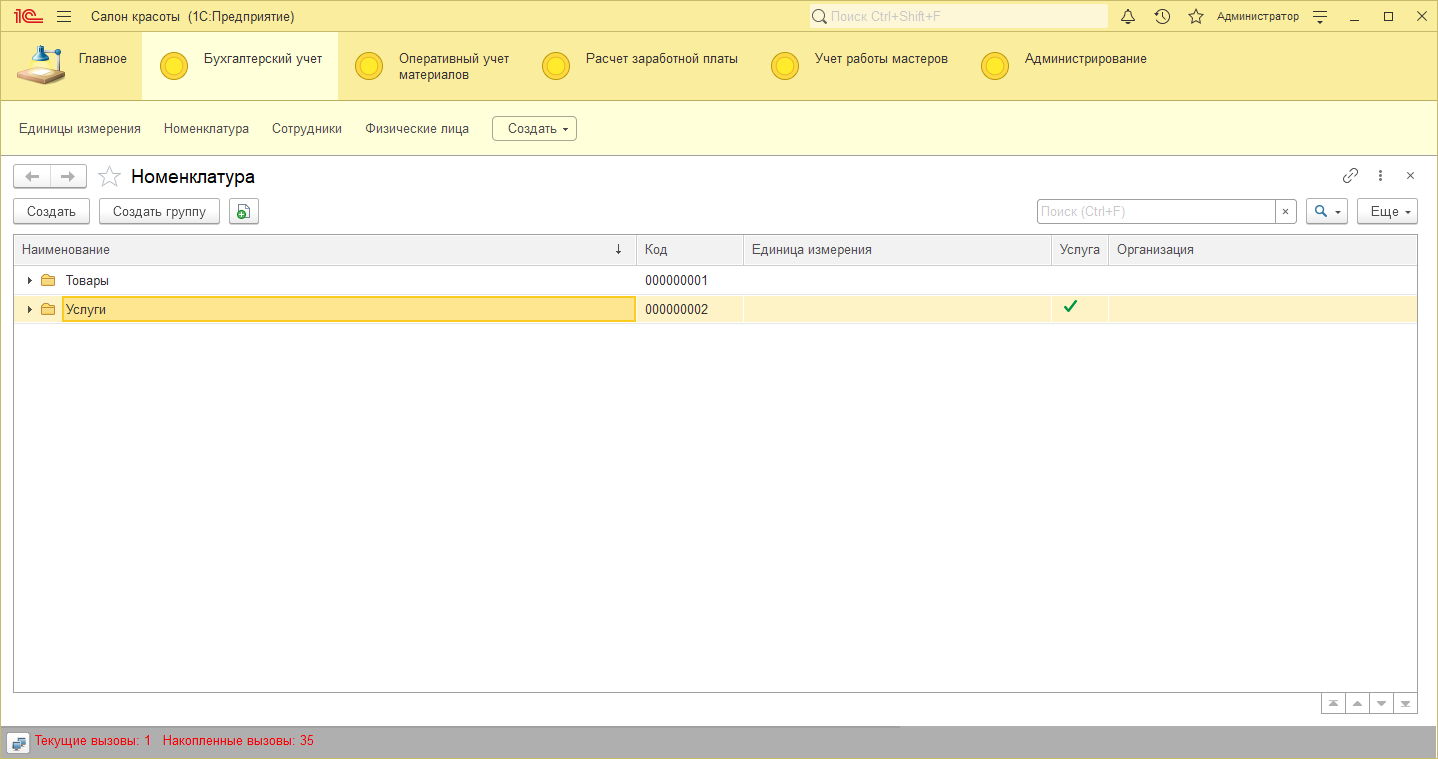
Перейдем в *модуль* объекта (кнопка **Модуль объекта** на закладке **Прочие** окна редактирования объекта), из списка процедур и выберем **ОбработкаЗаполнения**, [рис. 4.3](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=2#image.4.3).



**Рис. 4.3.**Процедура обработки заполнения справочника

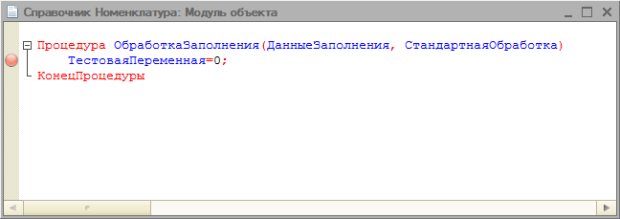
Процедура будет исполняться на стороне сервера, причем, вызываться она будет при различных способах создания элемента справочника – например, при интерактивном создании пользователем, при копировании, при программном создании. *Параметр* процедуры **СтандартнаяОбработка** позволяет включать или отключать стандартную обработку процесса заполнения реквизитов, *параметр* **ДанныеЗаполнения** содержит данные, которые система использует для заполнения элемента.

В режиме 1С:Предприятие откроем справочник **Номенклатура**, создадим две группы – **Товары** – флаг **Услуги** в этой группе не устанавливаем, и **Услуги** – флаг установлен, [рис. 4.4](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=2#image.4.4).



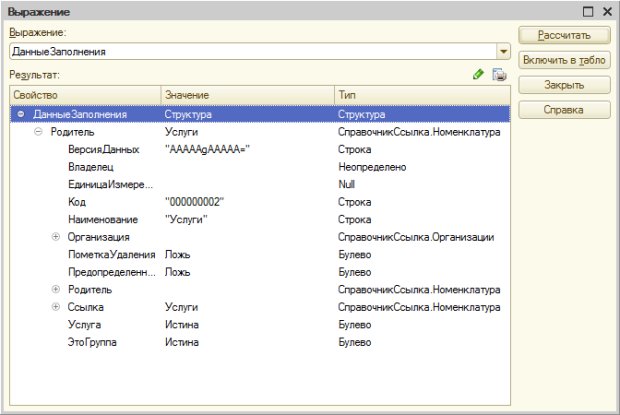
**Рис. 4.4.**Две группы в справочнике Номенклатура

Исследуем процедуру обработки заполнения, прежде чем продолжать работу над ней. Для этого вставим в нее какую-нибудь команду, например: **ТестоваяПеременная = 0;** и установим на строку с данной командой точку останова. Для этого либо выполним *двойной щелчок* левой кнопкой мыши на сером *поле* слева от команды, либо, установив *курсор* в строку с командой, выполним команду **Отладка > Точка останова**, либо – установив *курсор* в нужную строку, нажмем **F9**. Для установки точки останова нужно, чтобы строка, на которую мы пытаемся ее установить, присутствовала в *конфигурации базы данных*, то есть – написав код, нужно нажать на кнопку **Обновить конфигурацию** *базы данных*. В итоге у нас должно получиться следующее, [рис. 4.5](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=2#image.4.5).



**Рис. 4.5.**Точка останова в процедуре

Запустим конфигурацию в режиме отладки (кнопка **Начать отладку**, *команда* *меню* **Отладка > Начать отладку**, или клавиша **F5** на клавиатуре). Перейдем в группу **Услуги** и создадим в ней новый элемент. Когда управление будет передано в **Конфигуратор**, установим *курсор* на имя параметра **ДанныеЗаполнения**, вызовем контекстное *меню* и выберем команду **Вычислить выражение**. Появится окно *Выражение*, из которого можно понять, что *переменная* **ДанныеЗаполнения** – это структура, в которой присутствуют сведения о родителе создаваемого элемента – то есть – о группе **Услуги**, [рис. 4.6](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=2#image.4.6).



**Рис. 4.6.**Структура ДанныеЗаполнения

Структура – это *таблица*, которая содержит пары вида *Ключ* – *Значение*.

В нашем случае, если процедура **ОбработкаЗаполнения** отработает – она заполнит лишь *поле* **Родитель** для создаваемого элемента. А нам хотелось бы установить и флаг **Услуга** в соответствии с данными родителя.

Рассмотрим некоторые составляющие данных, к которым мы можем получить *доступ* посредством структуры.

**Родитель** – здесь хранится родитель элемента – в нашем случае – *группа* **Услуги** типа **СправочникСсылка.Номенклатура**. То есть, при заполнения поля **Родитель** создаваемого элемента, окажется, что он будет хранить ссылку на другой элемент (в нашем случае – группу), входящий в справочник **Номенклатура**.

**Владелец** – данное *поле* у нашей группы, находящейся в справочнике, имеет *значение* **Неопределено**. Такое *значение* присваивается тем свойствам, которые, в принципе, могут быть установлены, но в данном случае значения не имеют.

**ЕдиницаИзмерения** имеет *значение* **Null**. При настройке состава реквизитов справочника **Номенклатура**, мы указали, что **ЕдиницаИзмерения** может задаваться только для элемента. Но в структуре справочника в *информационной базе* подобное *поле*присутствует и у группы. Однако значения оно содержать не может – поэтому в качестве типа значения мы видим **Null**. Типы значений **Неопределено** и **Null** кажутся похожими, но это – разные вещи. *Значение* с типом **Неопределено** может быть задано, а *значение***Null** не может быть задано в принципе.

Свойство **Услуга** установлено в *значение* **Истина** – этот флаг мы устанавливали при создании группы **Услуги**.

Свойство **ЭтоГруппа** так же истинно – оно устанавливается в истинность для групп.

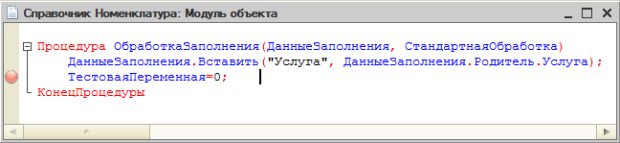
Для того чтобы установить свойство **Услуга** у создаваемого элемента, мы могли бы напрямую обратиться к свойству элемента **Услуга** и установить его в *значение* флага Услуга у его родителя. Выглядеть это может, например, так:

Услуга = ДанныеЗаполнения.Родитель.Услуга.

Однако процедура ОбработкаЗаполнения предусматривает автоматический механизм заполнения реквизитов на основе переданной структуры. Так как стандартный механизм нас вполне устраивает, мы можем поступить *по*-другому. А именно, для установки свойства **Услуга** нам нужно лишь дополнить структуру необходимой записью. Сделать это можно с помощью стандартных операций *по* работе со структурой. А именно, следующим образом:

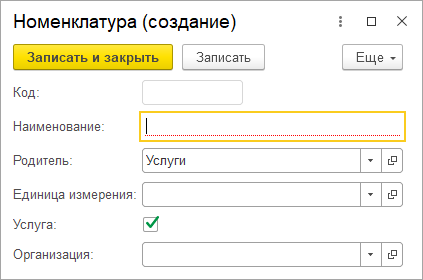
ДанныеЗаполнения.Вставить("Услуга", ДанныеЗаполнения.Родитель.Услуга);

В итоге у нас получается такой код, [рис. 4.7](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=2#image.4.7).



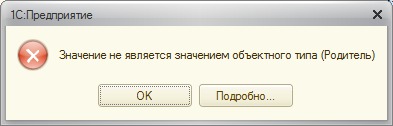
**Рис. 4.7.**Заполнение реквизита Услуга на основании параметров элемента родителя

Здесь мы установили точку останова для того, чтобы посмотреть, как изменится структура при выполнении данной процедуры. Опробуем решение в пользовательском режиме, можно заметить, что, во-первых, структура **ДанныеЗаполнения** действительно теперь содержит *ключ* **Услуга** со значением **Истина**, а так же то, что элементы, создаваемые в группе с установленным флагом **Услуга**, имеют данный *реквизит* в установленном положении, [рис. 4.8](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=2#image.4.8).



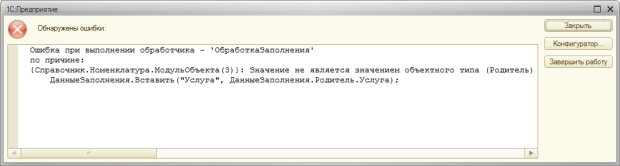
**Рис. 4.8.**Результат заполнения реквизита Услуга на основании параметров элемента родителя

Подойдет ли *созданная процедура* для практического использования? На этот вопрос может ответить ее тестирование. А именно, попробуем создать еще одну группу в корневой части справочника. Очевидно, что у такой группы родителя не будет. То же самое касается создания элемента. Попытка приводит к появлению ошибки, [рис. 4.8](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=2#image.4.8).



**Рис. 4.9.**Ошибка при попытке создать элемент, у которого нет родителя

Нажав на кнопку **Подробно**, видим, что ошибка произошла при попытке добавить в структуру новую *запись*, [рис. 4.10](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=2#image.4.10).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/4/files/04_10.jpg)

**Рис. 4.10.**Более подробное описание ошибки

Нажав на кнопку **Конфигуратор**, мы попадаем в *Конфигуратор*. Прежде чем обращаться к элементу структуры Родитель, нужно убедиться в том, что **Родитель** в структуре присутствует. Если **Родителя** нет – в структуру не нужно ничего добавлять, если есть – можно добавить. Попробуем такой код:

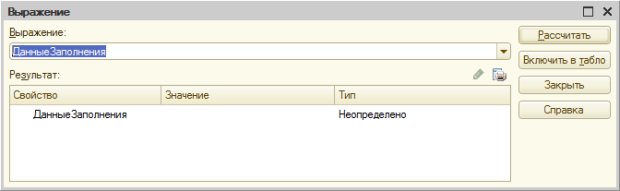
Если ДанныеЗаполнения.Свойство("Родитель") Тогда

ДанныеЗаполнения.Вставить("Услуга", ДанныеЗаполнения.Родитель.Услуга);

КонецЕсли;

Как кажется, все должно работать правильно – если в структуре обнаружилось *поле* Родитель – мы можем обращаться к свойству Услуга. Но попытка выполнить эту процедуру снова приводит к ошибке.

Эта ошибка возникает при проверке условия на наличие в структуре свойства Родитель. Здесь у нас возникает вопрос о том, чем является передаваемый *параметр* **ДанныеЗаполнения** при создании элемента или группы на верхнем уровне справочника. Для ответа на этот вопрос мы можем снова прибегнуть к отладке. Как видно, *значение* параметра не определено, [рис. 4.11](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=2#image.4.11).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/4/files/04_11.jpg)

**Рис. 4.11.**Параметр ДанныеЗаполнения при создании элемента или группы на верхнем уровне справочника

Тип **Неопределено** говорит нам о том, что перед нами лишь *переменная*, тип которой программе не известен. Мы не можем обращаться к ней как к структуре, поэтому, прежде чем проверять, есть ли в структуре **ДанныеЗаполнения** свойство **Родитель**, нам нужно проверить, является ли передаваемый *параметр* **ДанныеЗаполнения** структурой. Мы знаем, что этот *параметр*, когда он заполнен данными, имеет тип **Структура**, следовательно, нам нужно исключить вариант, когда его тип равняется **Неопределено**.

Вышеприведенные рассуждения приводят нас к следующему коду:

Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)

Если ДанныеЗаполнения<>Неопределено Тогда

Если ДанныеЗаполнения.Свойство("Родитель") Тогда

ДанныеЗаполнения.Вставить("Услуга", ДанныеЗаполнения.Родитель.Услуга);

КонецЕсли;

КонецЕсли;

КонецПроцедуры

В данной редакции обработчика события **ОбработкаЗаполнения** все работает верно.

Чисто теоретически (предположим, при изменении кем-либо нашего кода) возможна ситуация, когда **ДанныеЗаполнения** будут являться структурой и в этой структуре, в то же время, не будет свойства **Родитель**. Поэтому наряду с проверкой на неопределенность значения мы оставляем и проверку на наличие свойства **Родитель**.

Мы обсудили и проиллюстрировали родительские отношения в справочнике, рассмотрим теперь пример работы с *подчиненными справочниками*.

### Подчиненные справочники

Создадим новый справочник, назовем его **Контрагенты**.

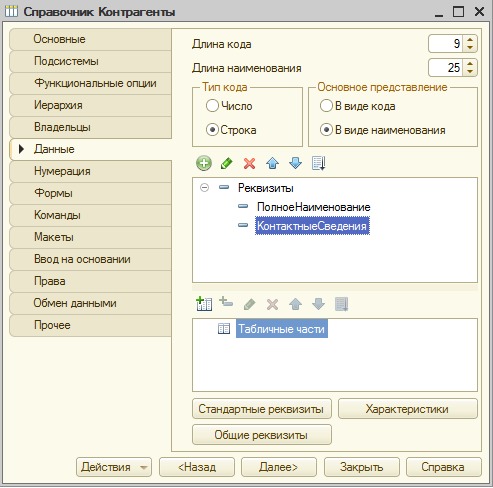
Добавим его в подсистемы **БухгалтерскийУчет** и **ОперативныйУчетМатериалов**.

Справочник будет иерархическим, с иерархией групп и элементов.

В состав реквизитов справочника добавим следующие ([рис. 4.12](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=3#image.4.12).):

**Имя**: ПолноеНаименование, тип – Строка, *длина* 100.

**Имя**: КонтактныеСведения, тип – Строка, *длина* 100



**Рис. 4.12.**Справочник Контрагенты

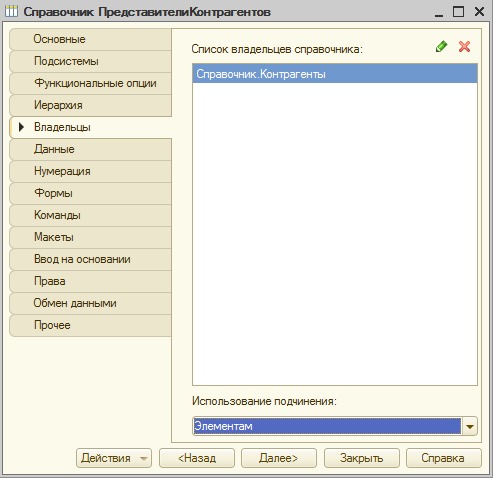
Создадим еще один справочник. Назовем его **ПредставителиКонтрагентов**. Главная черта этого справочника – то, что он подчинен справочнику **Контрагенты**. Для настройки подчинения используется вкладка окна настройки объекта конфигурации **Владельцы**. Здесь мы должны добавить в **Список владельцев справочника** справочники-владельцы, в нашем случае – справочник **Контрагенты**. После того, как владелец добавлен в этот *список*, мы можем настроить для него *параметр* **Использование подчинения**. Здесь возможны три варианта:

**Элементам** – элементы *подчиненного справочника* подчинены элементам справочника-владельца. В нашем случае выбор этого параметра означает, что в качестве "владельца" представителя контрагента выступает сам *контрагент*.

**Группам** – подчинение группам справочника-владельца. Нас это не устроит – так как группы справочника будут содержать контрагентов, сгруппированных *по* достаточно общим показателям (Покупатели, Заказчики и т.д.), и сопоставление одного контактного лица нескольким разным, например, покупателям, смысла не имеет. В другой ситуации эта установка могла бы быть вполне оправданной.

**Группам и элементам** – подчинение как группам, так и элементам справочника-владельца.

Мы укажем в параметре **Использование подчинения** вариант **Элементам**, [рис. 4.13](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=3#image.4.13).



**Рис. 4.13.**Настройка подчинения

Добавим справочник в состав подсистем **БухгалтерскийУчет** и **ОперативныйУчетМатериалов**.

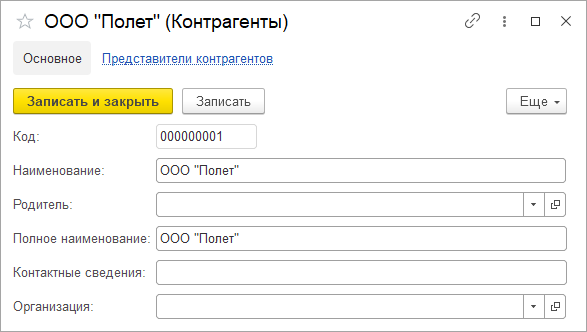
В состав реквизитов справочника добавим следующие:

**Имя**: ФИО, тип – Строка, *длина* 100

**Имя**: КонтактныеСведения, тип – Строка, *длина* 100.

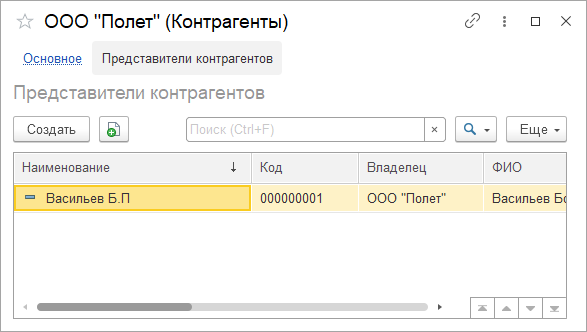
**Имя**: ПредставительРаботает, тип – Булево.

Посмотрим теперь, как выглядит работа с созданными справочниками в режиме **1С:Предприятие**. Особенность здесь заключается в том, что, открывая карточку контрагента, можно найти ссылку для перехода в справочник **ПредставителиКонтрагентов**, [рис. 4.14](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=3#image.4.14).



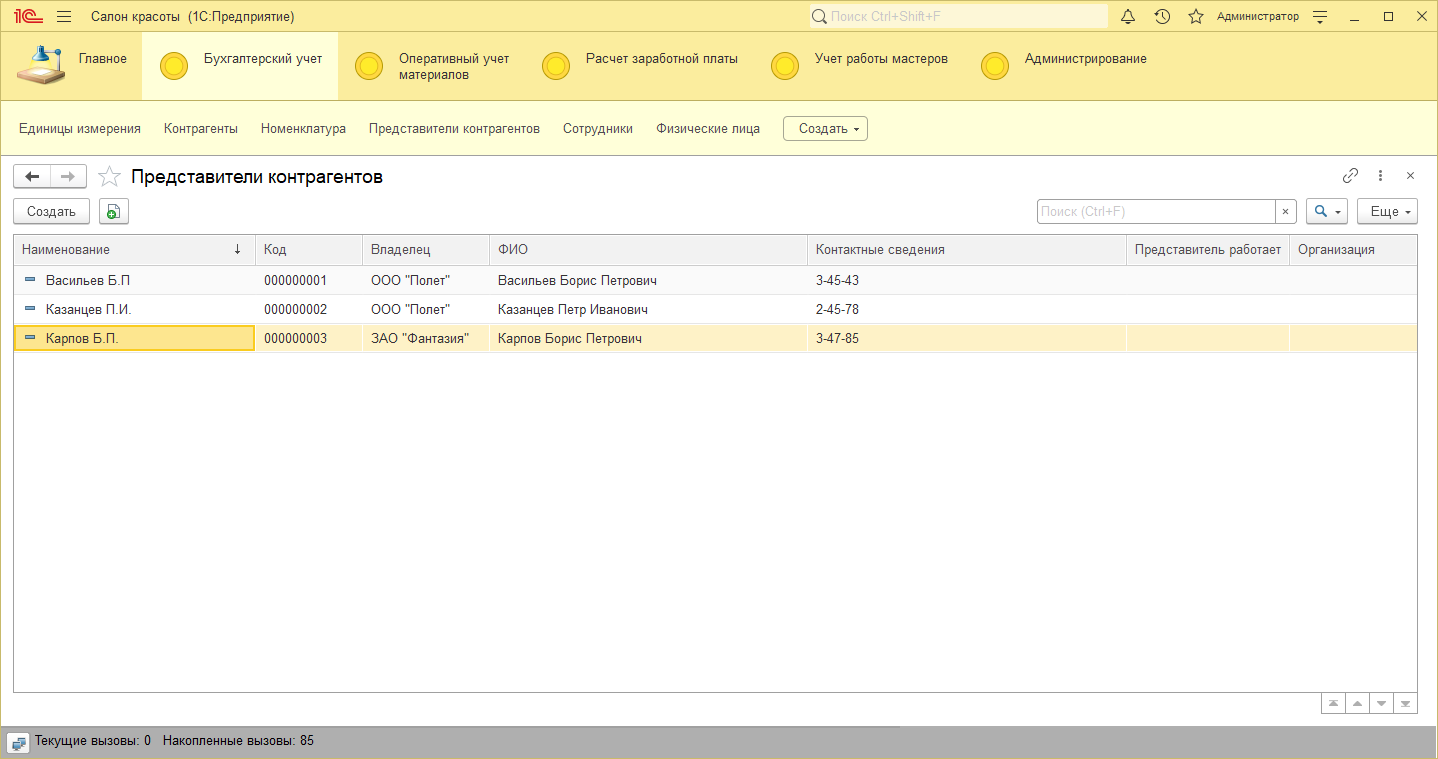
**Рис. 4.14.**Форма элемента справочника Контрагенты

При переходе в этот справочник мы будем видеть в открывшемся окне лишь тех представителей, которые относятся к контрагенту, с которым мы в данный момент работаем. При создании новой записи о представителе он автоматически будет "привязываться" к тому контрагенту (*поле* владелец будет заполнено должным образом), из формы элемента которого мы перешли в справочник **ПредставителиКонтрагентов**. В форме списка справочника будет отображаться *ссылка* для перехода к форме элемента справочника-владельца, [рис. 4.15](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=3#image.4.15).



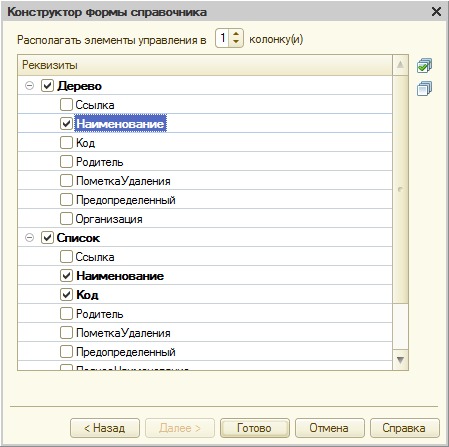
**Рис. 4.15.**Формы списка и элемента справочника ПредставителиКонтрагентов

Мы можем создавать элементы справочника **ПредставителиКонтрагентов** и непосредственно перейдя в него, тогда нам придется самостоятельно указывать его владельца – элемент справочника **Контрагенты**. При переходе в *подчиненный справочник* не из формы элемента справочника-владельца, мы можем просматривать все его элементы, [рис. 4.16](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=3#image.4.16).



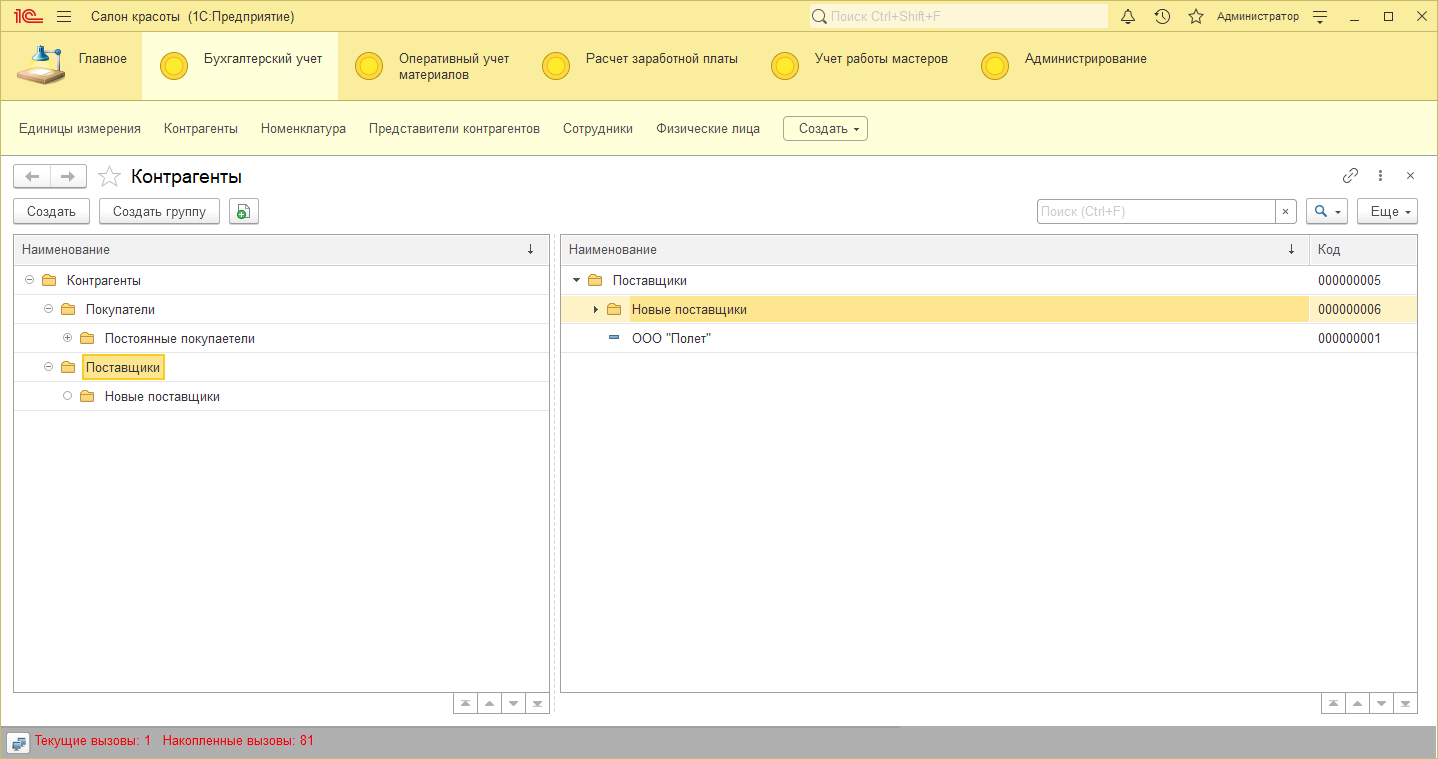
**Рис. 4.16.**Просмотр формы списка справочника ПредставителиКонтрагентов

Перейдем в окно редактирования объекта конфигурации справочника **Контрагенты**, перейдем на его закладку **Формы**, создадим новую форму списка. При работе с конструктором форм можно заметить, что на закладке управления реквизитами присутствуют два элемента – **Дерево** и **Список**. *Список* мы с вами уже видели, а элемент **Дерево** характерен для *иерархических справочников*, он позволяет облегчить навигацию *по* большим справочникам, выводя их иерархическую структуру в *дополнение* к списку. Установим флаг в *поле* **Дерево**, из списка реквизитов, отображаемых в дереве элементов, выберем **Наименование**, [рис. 4.17](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=3#image.4.17).



**Рис. 4.17.**Конструктор формы справочника Контрагенты

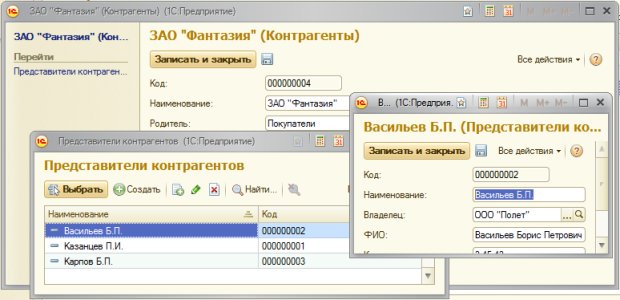
Вот как будет выглядеть форма списка справочника в режиме 1С:Предприятие, [4.18](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=3#image.4.18).



**Рис. 4.18.**Форма справочника со списком и деревом элементов

### Проверка заполнения реквизита справочника, фильтрация

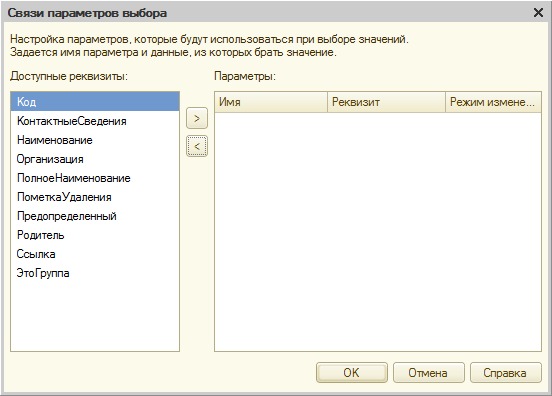
Расширим справочник **Контрагенты**, добавим в состав его реквизитов еще один – назовем его **ОсновноеКонтактноеЛицо**, тип – **СправочникСсылка.ПредставителиКонтрагентов**. Смысл этого поля заключается в хранении ссылки на представителя контрагента, который является "основным" для данного контрагента. Если нужно связаться с контрагентом, можно открыть его карточку и тут же увидеть, какой представитель является основным. Создадим форму элемента справочника **Контрагенты** и посмотрим на нее, попытавшись установить новое *поле* – **Основное контактное лицо**, [рис. 4.18](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=3#image.4.18).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/4/files/04_18.jpg)

**Рис. 4.19.**Попытка заполнения реквизита Основное контактное лицо

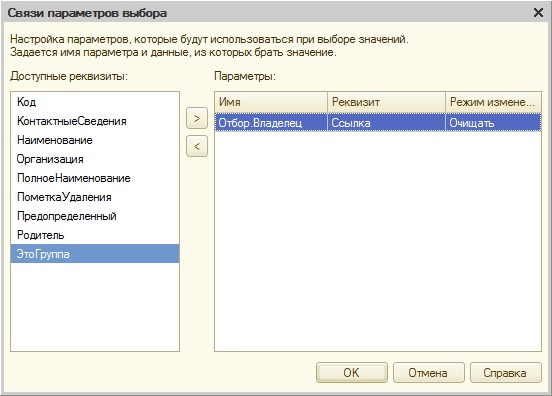
Видно, что при попытке подбора элемента в данное *поле* нам показывают не только те элементы справочника **ПредставителиКонтрагентов**, владельцем которых является редактируемый элемент, но и все остальные. Так работать неудобно – это значит, что нам нужно настроить фильтрацию выводимых элементов. Для того, чтобы это сделать, удобнее всего будет воспользоваться свойством **Связи параметров выбора** реквизита **ОсновноеКонтактноеЛицо**. Для открытия палитры свойств реквизита мы можем сделать *двойной щелчок* *по* реквизиту в окне редактирования объекта конфигурации, в *дереве конфигурации*, или воспользоваться командой контекстного *меню* Свойства.

В открывшейся палитре свойств найдем свойство **Связи параметров выбора** и нажмем на кнопку с тремя точками около этого поля. Появится окно **Связи параметров выбора**, [рис. 4.20](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=4#image.4.20).



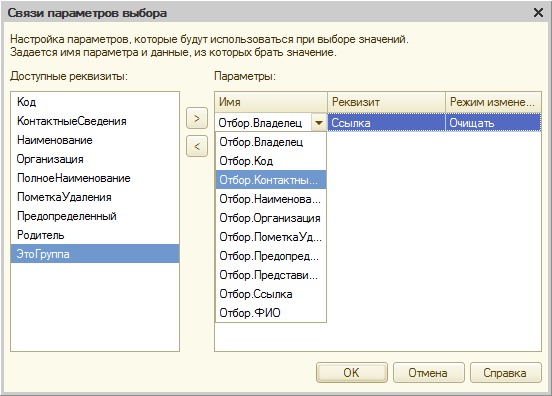
**Рис. 4.20.**Окно Связи параметров выбора

В левой части окна можно видеть доступные реквизиты (это реквизиты открытого элемента справочника **Контрагенты**), в правом – параметры, влияющие на отбор элементов в появляющемся окне выбора элементов при заполнении поля представителя контрагента. Выделим *реквизит* **Ссылка** и нажмем на кнопку **Добавить выбранный реквизит в параметры выбора**(она находится между полями). Окно примет следующий вид, [рис. 4.21](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=4#image.4.21).



**Рис. 4.21.**Окно Связи параметров выбора с настроенным параметром

В данном случае в строке области **Параметры** отображается как раз то, что нам нужно – нам нужно, чтобы отбор в раскрывающемся списке происходил *по* владельцу, а именно – *по* текущему открытому элементу справочника **Контрагенты**, на который и указывает *реквизит* **Ссылка**. В *поле* имя можно выбрать другие варианты отбора, [рис. 4.21](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=4#image.4.21).



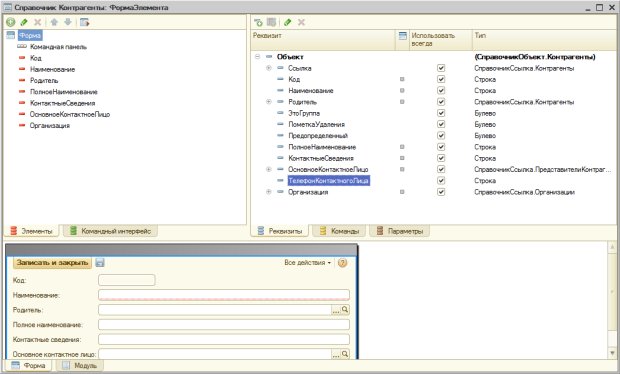
**Рис. 4.22.**Настройки в окне Связи параметров выбора

После того, как эта настройка выполнена, мы можем нажать **ОК** в окне **Связи параметров выбора**и проверить функциональность решения – при заполнении поля **ОсновноеКонтактноеЛицо** *список* выбора ограничивается подчиненными элементами.

Добавим в справочник **Контрагенты** еще один *реквизит* – **ТелефонКонтактногоЛица**. Зададим тип – **Строка**, *длина* – **100**. Этот *реквизит* соответствует реквизиту **КонтактныеСведения** справочника **ПредставителиКонтрагентов**. Он нужен нам исключительно для удобства – для того, чтобы, когда в форме контрагента указано основное контактное лицо, пользователю не пришлось бы, для поиска телефона контактного лица, заглядывать в его карточку.

После добавления реквизита в справочник **Контрагенты**, запустим режим 1С:Предприятие и откроем форму одного из элементов этого справочника. Если присмотреться к этой форме на данном этапе работы, окажется, что реквизита **ТелефонКонтактногоЛица**на ней не наблюдается. Все дело в том, что, создав собственную форму элемента для справочника, мы отказываемся от автоматического механизма создания форм, который, если бы не наша, самостоятельно созданная ранее форма, автоматически построил бы форму с новым реквизитом.

Добавим элемент управления для реквизита **ТелефонКонтактногоЛица** на форму. Откроем форму элемента справочника Контрагенты для редактирования и перетащим *реквизит* **ТелефонКонтактногоЛица** с вкладки **Реквизиты** на вкладку **Элементы**, [рис. 4.23](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=4#image.4.23).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/4/files/04_22.jpg)

**Рис. 4.23.**Реквизит ТелефонКонтактногоЛица нужно переместить со вкладки Реквизиты на вкладку Элементы

В справочнике **ПредставителиКонтрагентов** есть *реквизит*, который указывает на то, что представитель контрагента работает в организации-контрагенте – это *реквизит* *логического типа* **ПредставительРаботает**. Нам нужно реализовать следующий функционал. Если *пользователь*, заполняя карточку элемента справочника **Контрагенты** выбирает в качестве реквизита **ОсновноеКонтактноеЛицо** сотрудника, у которого флаг **ПредставительРаботает** не установлен – мы предупреждаем пользователя об этом, выводя сообщение.

Для этого нам понадобится *перехватить событие* изменения поля **ОсновноеКонтактноеЛицо**, после чего проверить, установлен ли у выбранного контактного лица флаг **ПредставительРаботает**, и если такой флаг не установлен – вывести сообщение пользователю.

Из контекстного *меню* элемента формы **ОсновноеКонтактноеЛицо** выберем событие **ПриИзменении**, откроется *редактор кода*, в котором уже будет создана пустая клиентская процедура для перехвата этого события. К этому моменту *реквизит***ОсновноеКонтактноеЛицо** уже будет содержать выбранного представителя контрагента. Нам понадобится серверная процедура, которая обратится к реквизиту этого представителя **ПредставительРаботает** и вернет нам его *значение*. После того, как мы получим с сервера сведения о том, работает ли представитель, мы примем решение – выводить ли пользователю сообщение или нет.

Все это реализовано с помощью нижеприведенного кода:

&НаКлиенте

Процедура ОсновноеКонтактноеЛицоПриИзменении(Элемент)

Если НЕ ПроверитьЗаполнениеРеквизита() Тогда

Сообщить("Выбранное контактное лицо, "+Объект.ОсновноеКонтактноеЛицо+", не работает у контрагента.");

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПроверитьЗаполнениеРеквизита()

Возврат (Объект.ОсновноеКонтактноеЛицо.ПредставительРаботает);

КонецФункции

Кроме того, зададим автоматический механизм переноса в *реквизит* **ТелефонКонтактногоЛица** сведений из элемента справочника **ПредставителиКонтрагентов**, который указан в *поле* **ОсновноеКонтактноеЛицо**. В частности, дополним обработчик события для поля **ОсновноеКонтактноеЛицо** **ПриИзменении():**

Подготовим серверную процедуру, которая будет работать с реквизитами объекта, она будет иметь следующий вид:

&НаСервере

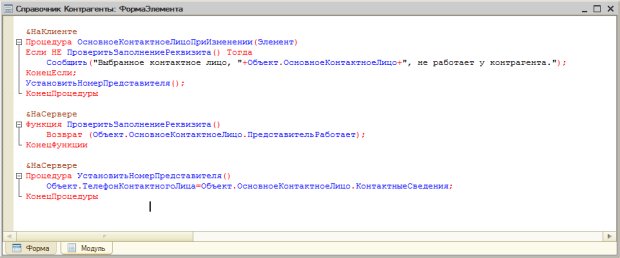
Процедура УстановитьНомерПредставителя()

Объект.ТелефонКонтактногоЛица=Объект.ОсновноеКонтактноеЛицо.КонтактныеСведения;

КонецПроцедуры

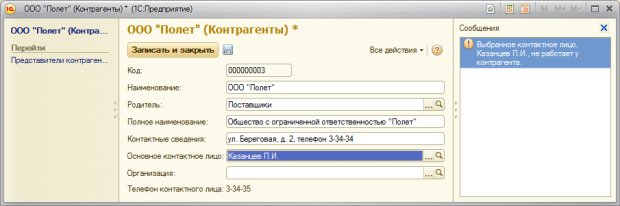
Вызовем эту процедуру в уже существующем обработчике **ПриИзменении**, код модуля приобретет вид, показанный на [рис. 4.24](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=4#image.4.24).

Кроме того, в окне его свойств отредактируем свойство **Вид** – выберем его *значение* **Поле надписи**. *Пользователь* не будет ничего в это *поле* вводить самостоятельно, поэтому *поле* надписи нас вполне устроит.

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/4/files/04_23.jpg)

**Рис. 4.24.**Код модуля формы элемента справочника Контрагенты

При выборе "неподходящего" представителя окно элемента справочника Контрагенты примет следующий вид, [рис. 4.25](https://www.intuit.ru/studies/courses/2318/618/lecture/13470?page=4#image.4.25).

[](https://www.intuit.ru/EDI/13_12_15_4/1449958784-19792/tutorial/1046/objects/4/files/04_24.jpg)

**Рис. 4.25.**Сообщение о выборе неподходящего контактного лица

Предположим, нам нужно, чтобы полное наименование организации было бы не короче 5-ти символов.

В предыдущей лекции, работая с формами, мы рассматривали проверку заполнения реквизитов справочника на стороне клиента. Такой подход возможен, но если говорить о производительности решения, предполагая работу конфигурации на сравнительно маломощном клиенте, лучше выполнять подобные проверки на стороне сервера. Как мы уже говорили, для проверки заполненности реквизита можно использовать стандартное свойство **Проверка заполнения**. Но если вам нужны более сложные проверки, если их достаточно много (например, при проверке производится манипулирование с табличными частями, получение каких-то дополнительных данных из базы, при этом алгоритмы достаточно сложны), лучше всего для проверки заполнения реквизитов воспользоваться процедурой **ОбработкаПроверкиЗаполнения ()**, которая объявляется на уровне модуля объекта.

Объявим эту процедуру, открыв *модуль* объекта (закладка **Прочее** окна редактирования свойств объекта, кнопка **Модуль объекта**) выбором из списка **Процедуры и функции панели инструментов***модуль* процедуры ОбработкаПроверкиЗаполнения().

Эта процедура работает на сервере, мы можем напрямую обращаться к реквизитам объекта.

Процедура ОбработкаПроверкиЗаполнения(Отказ, ПроверяемыеРеквизиты)

Если СтрДлина(ПолноеНаименование) < 5 И НЕ ЭтотОбъект.ЭтоГруппа Тогда

Отказ=Истина;

Сообщить("Полное наименование организации должно быть не короче 5-ти символов");

КонецЕсли;

КонецПроцедуры

Передаваемый в процедуру *параметр* **Отказ** можно установить в *значение* **Истина** для того, чтобы показать, что проверка не пройдена. **ПроверяемыеРеквизиты** – это *массив*, он содержит реквизиты для автоматической проверки, в частности, там находятся те реквизиты, для которых включена автоматическая проверка заполнения. *По* умолчанию это – *реквизит* **Наименование**. В этом можно убедиться, просмотрев содержимое переменной в отладчике.