**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное учреждение высшего образования**
**«Севастопольский государственный университет»**

**Институт информационных технологий и управления в технических системах**

# «ИССЛЕДОВАНИЕ ОБЪЕКТНЫХ ТИПОВ ДАННЫХ В СИСТЕМЕ 1С:ПРЕДПРИЯТИЕ 8»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

часть 2

по дисциплине

«Платформа 1С»

для студентов дневной формы обучения специальности 09.03.02

«Информационные системы и технологии»

Севастополь
2017

УДК 004.42

Методические указания к лабораторной работе «Исследование объектных типов данных в системе 1С:Предприятие 8» по дисциплине «Платформа 1С» для студентов дневной формы обучения. Сост. С.П. Шишкевич В.Е., Лагуткина Т.В., Савин А.Ю.– Севастополь: Изд-во СГУ, 2017. – 76 с.

Целью настоящих указаний является оказание помощи студентам в изучении приемов описания объектов предметной области в конфигураторе программного комплекса 1С:Предприятие.

Методические указания рассмотрены и утверждены на заседании кафедры Информационных систем (протокол №10 от 21.04 2017 года).

Допущено учебно-методическим центром СевГУ в качестве методических указаний.

Рецензент: к.т.н., доцент кафедры «Информационные технологии и компьютерные системы» ИИТиУвТС СевГУ Кожаев Е.А.

# СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА №3
«Исследование объектных типов данных в системе 1С:Предприятие» ..... 4
1.1. ЦЕЛЬ РАБОТЫ ..... 4
1.2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ ..... 4
1.2.1. Основные объекты конфигурации в «1С:Предприятие» ..... 4
1.2.2. Объектные и табличные данные. Общие сведенья ..... 5
1.2.2.1. Подходы к манипулированию данными. «1С:Предприятие» обеспечивает два способа доступа к данным: ..... 7
1.1.2.2. Объектные типы данных ..... 8
1.2.2.3. Справочники ..... 9
1.2.2.4. Документы ..... 19
1.3. ПРИМЕР ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ ..... 29
1.4. ПРОГРАММА ВЫПОЛНЕНИЯ РАБОТЫ ..... 35
1.5. СОДЕРЖАНИЕ ОТЧЕТА ..... 36
1.6. КОНТРОЛЬНЫЕ ВОПРОСЫ ..... 37
ЛАБОРАТОРНАЯ РАБОТА №4. «Исследование табличных типов данных в системе 1С:предприятие.» ..... 38
2.1. ЦЕЛЬ РАБОТЫ ..... 38
2.2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ ..... 38
2.2.1. Табличные данные. Основные сведенья. ..... 38
2.2.2. Регистр накопления ..... 39
2.2.3. Регистр сведений ..... 48
2.3. ПРИМЕРЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ ..... 57
2.3.1. Регистры сведений ..... 57
2.3.2. Регистры накоплений ..... 61
2.4. ПРОГРАММА ВЫПОЛНЕНИЯ РАБОТЫ ..... 66
2.5. СОДЕРЖАНИЕ ОТЧЕТА ..... 67
2.6. КОНТРОЛЬНЫЕ ВОПРОСЫ ..... 68
БИБЛИОГРАФИЧЕСКИЙ СПИСОК ..... 69
ПРИЛОЖЕНИЕ А ..... 71
ПРИЛОЖЕНИЕ В ..... 73

# ЛАБОРАТОРНАЯ РАБОТА №3

# «Исследование объектных типов данных в системе 1С:Предприятие»

## 1.1. ЦЕЛЬ РАБОТЫ

Целью данной лабораторной работы является исследование объектных типов данных в системе 1С:Предприятие.

## 1.2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

### 1.2.1. Основные объекты конфигурации в «1С:Предприятие»

***Объекты конфигурации*** – это составные элементы, "детали", из которых складывается любое прикладное решение. Они представляют собой настраиваемые объекты (объекты с настраиваемыми свойствами), поддерживаемые на уровне технологической платформы. По большому счету задача разработчика заключается в том, чтобы собрать из этих объектов, как из конструктора, необходимую структуру прикладного решения и затем описать специфические алгоритмы функционирования и взаимодействия этих объектов, отличающиеся от их типового поведения.

Состав объектов, поддерживаемых технологической платформой, является результатом анализа предметных областей использования 1С:Предприятия, и выделения и классификации используемых в этих областях сущностей. В результате этого анализа разработчик может оперировать такими объектами как справочники, документы, регистры сведений, планы счетов и пр.

Для того чтобы стандартизировать и упростить процесс разработки и модификации прикладных решений, разработчику предоставляется графический интерфейс, с помощью которого он имеет возможность описать состав объектов, используемых в конкретном прикладном решении.

На основании этого описания технологическая платформа создаст в базе данных соответствующие информационные структуры, и определенным образом будет работать с данными, хранящимися в этих структурах. Разработчику нет необходимости заботиться о том, в каких таблицах, например, должны размещаться данные, каким образом они будут модифицироваться или представляться пользователю. Все эти действия платформа будет выполнять автоматически, исходя из типового поведения используемых объектов.

Таким образом, разработчик оперирует метаданными – "данными о данных", или *объектами* конфигурации. Добавляя в структуру

прикладного решения очередной объект конфигурации, разработчик, по сути, добавляет описание того, как будут размещаться соответствующие данные, и как они будут взаимодействовать с другими данными, хранящимися в информационной базе.

Состав объектов, которые может использовать разработчик, фиксирован и определен на уровне платформы. Разработчик не может создавать собственные виды объектов, он может оперировать только тем набором объектов, который имеется. Подобный подход к разработке прикладных решений позволяет, во-первых, стандартизировать процесс разработки, а во-вторых – обеспечить простую и быструю модификацию прикладных решений другими разработчиками или пользователями. На рисунке 2.1. представлены различные группы объектов конфигурации.




Рисунок 2.1 – Взаимосвязь различных групп объектов.

### 1.2.2. Объектные и табличные данные. Общие сведенья

В «1С:Предприятие» принято разделять все прикладные данные на те, которые имеют объектную природу (**объектные** данные) и не имеют объектной природы (необъектные или **табличные** данные).

С точки зрения системы ***объектные данные*** состоят из отдельных объектов. Каждый из этих объектов обладает внутренним уникальным идентификатором, благодаря наличию которого к некоторой совокупности значений, хранящихся в базе данных, можно обращаться как к единому целому – объекту. Например, объектом является *элемент справочника* или

*документ*. Каждый объект, помимо того что он является совокупностью значений некоторых полей, имеет также определенную значимость сам по себе. Например, элемент справочника *ФизическиеЛица* – это некое физическое лицо, которое имеет набор характеризующих его значений:

- имя, фамилия,
- отчество,
- паспортные данные и т. д.

У него может поменяться, например, фамилия или паспортные данные, но при этом с точки зрения системы он останется тем же самым физическим лицом – объектом.

Удаление какого-либо объекта из системы приводит к тому, что состояние базы данных с точки зрения прикладного решения изменяется. Даже создав новый элемент справочника *ФизическиеЛица* с теми же самыми значениями реквизитов, получим уже другое состояние базы данных, поскольку это будет уже другой объект с другим уникальным идентификатором.

В отличие от объектных данных, ***необъектные (табличные) данные*** не имеют собственной ценности и полностью описываются значениями своих полей. Необъектные данные представляют собой **записи**, которые хранятся в базе данных. Для записей не поддерживаются внутренние уникальные идентификаторы. Удалив некоторую запись и создав новую с точно такими же значениями полей, получим то же самое состояние базы данных, которое было до удаления записи.

Могут измениться любые данные объекта, но это будет все тот же объект. Удалив объект, его нельзя создать заново. Даже если для нового объекта будут установлены те же данные, это будет уже другой объект с точки зрения «1С:Предприятия», обладающий другим уникальным идентификатором.

К **объектным** данным относятся данные *справочников, документов, планов видов характеристик, планов счетов, планов видов расчета, бизнес-процессов, задач, планов обмена.*

К **необъектным** данным относятся данные *регистров сведений, регистров накопления, регистров расчета, перерасчетов, регистров бухгалтерии и последовательностей*. Также к необъектным данным относятся *константы*.

Деление на объектные и необъектные определяют различные подхода к *работе с данными*.

**1.2.2.1. Подходы к манипулированию данными**. «1С:Предприятие» обеспечивает два способа доступа к данным:

- объектный;
- табличный.

В *объектной* модели разработчик оперирует объектами встроенного языка. В этой модели обращения к объекту, например документу, происходят как к единому целому – он полностью загружается в память, вместе с вложенными таблицами, к которым можно обращаться средствами встроенного языка как к коллекциям записей и т.д. (рис.2.2.)

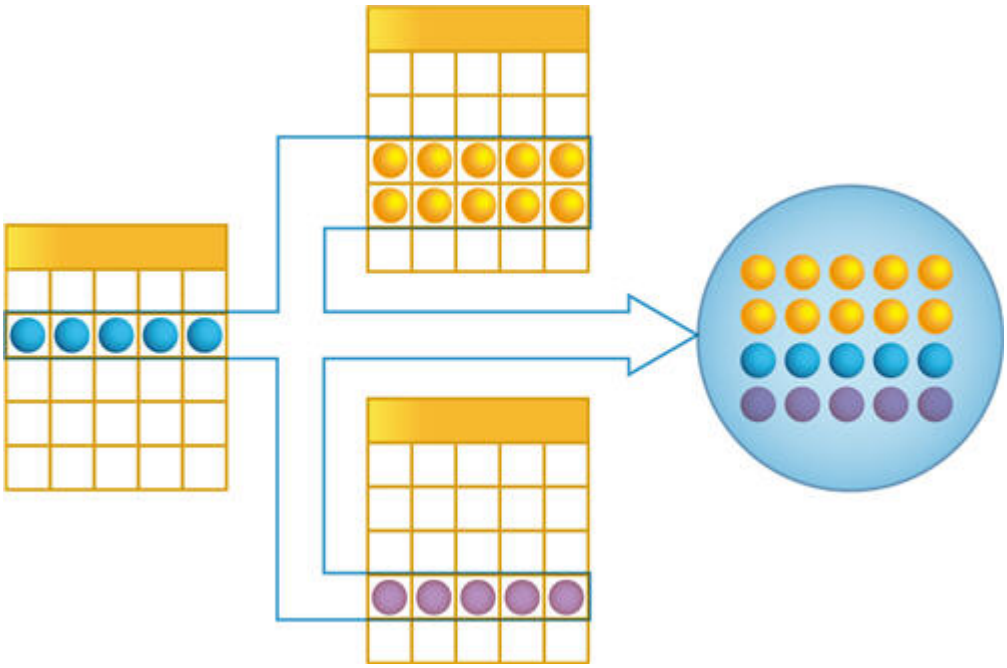


Рисунок 2.2 – Объектный доступ к данным.

При манипулировании данными в объектной модели обеспечивается сохранение целостности объектов, кэширование объектов, вызов соответствующих обработчиков событий и т.д.

В *табличной* модели все множество объектов того или иного класса представляется как совокупность связанных между собой таблиц, к которым можно обращаться при помощи запросов – как к отдельной таблице, так и к нескольким таблицам во взаимосвязи (рис.2.3):

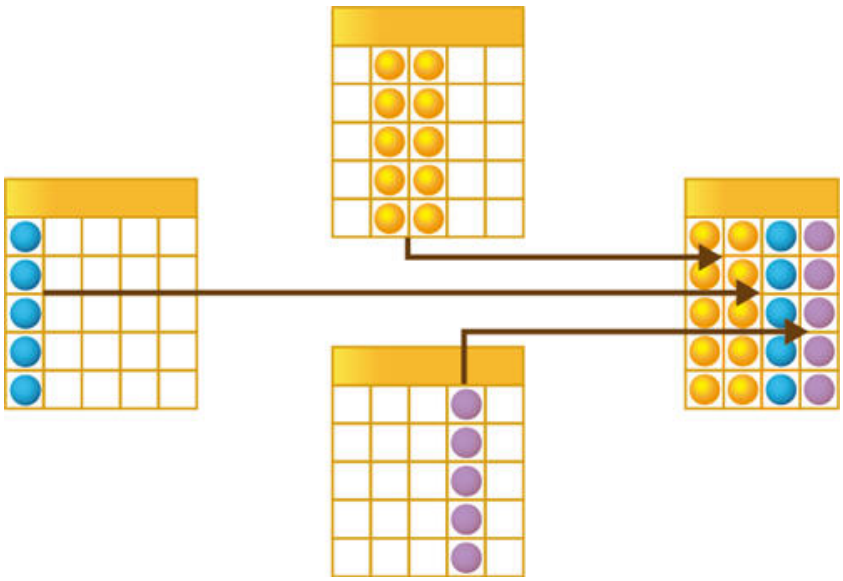


Рисунок 2.3 – Табличный доступ к данным.

В этом случае разработчик получает доступ к данным сразу нескольких объектов, что очень удобно для анализа больших объемов данных, например, при создании отчетов. Однако в силу того, что данные, выбираемые таким способом, содержат не все, а лишь некоторые реквизиты анализируемых объектов, табличный способ доступа не позволяет изменять эти данные.

Сочетание табличного и объектного доступа к данным позволяет, с одной стороны, сделать разработку прикладных решений простой и наглядной, а с другой стороны, получать сколь угодно сложные выборки данных и использовать возможности агрегирования полученных данных.

Рассмотрим работу с объектными данными более подробно.

### 1.1.2.2. Объектные типы данных

К объектным данным в «1С:Предприятие» относятся данные следующих объектов конфигурации:

- Справочник,
- Документ,
- План видов характеристик,
- План счетов,
- План видов расчета,
- План обмена,
- Бизнес-процесс,
- Задача.

Далее будет подробно рассмотрена работа с двумя объектами конфигурации: *Справочник*, *Документ*.

Это два основных объекта, которые лежат в основе реализации любой задачи на платформе 1С: Предприятие.

При проектировании прикладного решения формируется:

- конечный перечень объектов предметной области;
- перечень операций, которые совершают пользователи в процессе работы в рамках рассматриваемой предметной области.

Полученные объекты реализуются в виде Справочников в терминах платформы 1С, а операции в виде Документов.

### 1.2.2.3. Справочники

Справочники 1С – специализированный объект, который служит для хранения статичной (условно-постоянной) информации справочного характера. Например, в типовых конфигурациях можно увидеть следующие виды: Контрагенты, Номенклатура, Сотрудники, Основные средства и т.д.

#### Структура справочника

Справочник состоит из *элементов*. Например, для справочника Сотрудники элементом является сотрудник.

В базе данных каждый элемент справочника представляет собой отдельную запись в основной таблице, хранящей информацию из этого справочника. Каждая запись имеет уникальный идентификатор, который однозначно определяет конкретный элемент. В дальнейшем этот идентификатор мы будем называть Ссылкой на элемент справочника.

Каждый элемент справочника, как правило, содержит некоторую дополнительную информацию, которая подробнее описывает этот элемент. Набор такой информации является одинаковым для всех элементов справочника, и для описания такого набора используются ***реквизиты*** объекта конфигурации, которые также, в свою очередь, являются объектами конфигурации

Большинство реквизитов разработчик создает самостоятельно, однако у каждого объекта конфигурации Справочник по умолчанию существует набор стандартных реквизитов: Код и Наименование и пр. Код элемента справочника может быть как числовым, так и текстовым. Доступность стандартных реквизитов зависит от свойств справочника.

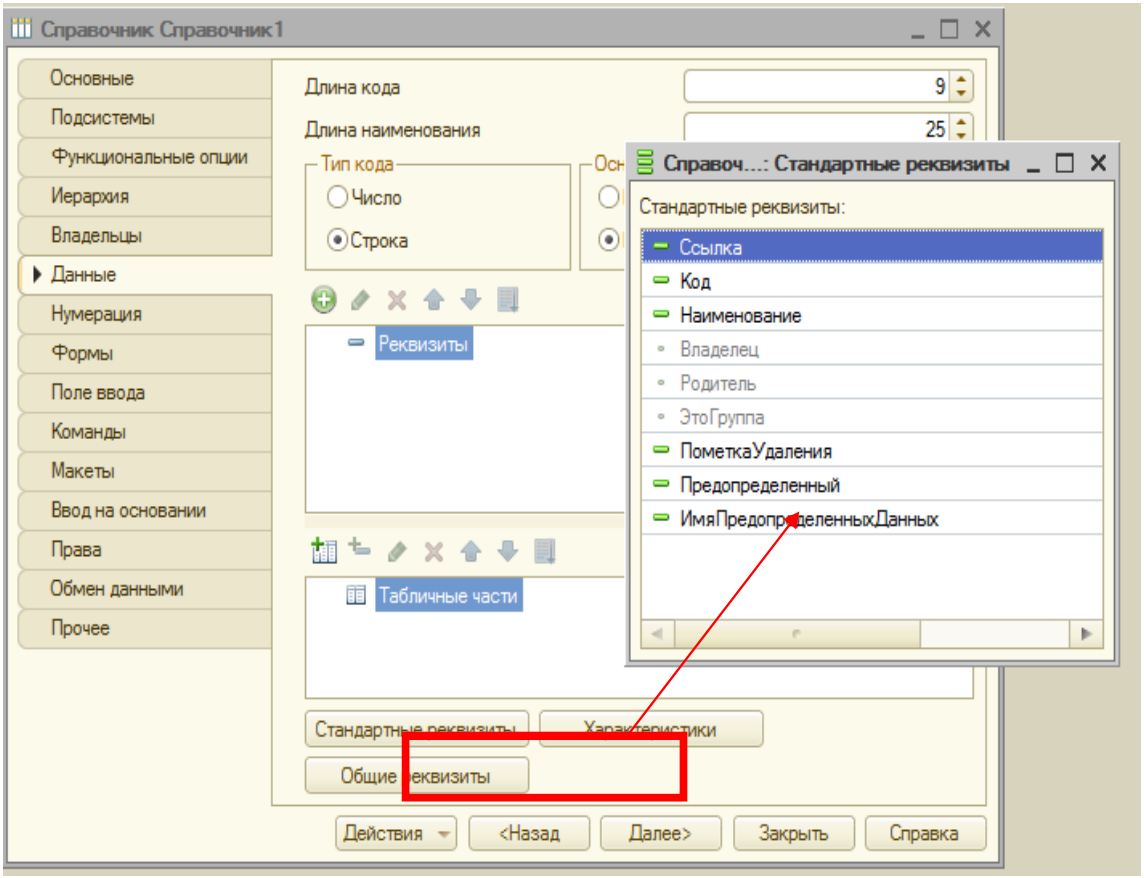



Рисунок 2.4 – Панель инструментов “Стандартные реквизиты”.

*ЭтоГруппа* – реквизит с типом булево, показывающий группа это или элемент. Доступен только в иерархическом справочнике.

*Код* – реквизит, тип число или строка (как правило, строка). Номер, присваиваемый системой автоматически. Как правило, рассчитывается как (предыдущий код + 1). Рекомендую использовать именно строковый тип, потому как сортировка числовых значений происходит не так, как нужно. Можно использовать как представление справочника в списке и в полях ввода. Как правило, используется для поиска элемента при вводе по строке. Если вам нужно убрать поле Код – укажите в длине строки ноль.

*Наименование* – реквизит, обязательный к заполнению, строкового типа. Максимальная длина строки – 150 символов. Можно использовать как представление справочника в списке и в полях ввода. Как правило, используется для поиска элемента при вводе по строке. Если вам нужно убрать поле Наименование – укажите в длине строки ноль.

*Родитель* – реквизит, имеющий тип СправочникСсылка.<ИмяТекущегоСправочника>. Доступен только в иерархическом справочнике. Указывает на вышестоящего родителя в иерархии. Если Элемент или Группа находятся в корне справочника – указывается значение Справочник.<ИмяТекущегоСправочника>.ПустаяСсылка.

*Владелец* – ссылка на элемент-владелец текущего элемента (группы) справочника. Доступен только в подчиненном справочнике.

*ПометкаУдаления* – реквизит с типом булево. Отвечает за отображение «пометки удаления» в системе. Помеченный на удаление элемент считается не пригодным к использованию, однако на нём могут оставаться старые движения в документах

*Ссылка* – поле, строкового типа. В этом реквизите хранится уникальный идентификатор объекта – GUID. То, что в системы видно в визуальном отображении под название «ссылка» – это всего лишь представление объекта. Невозможно изменить.

*Предопределенный* – тип булево, отображает, является ли элемент предопределенным. Невозможно изменить.

Можно изменить обозначение стандартного реквизита (например, Наименование на Ф.И.О) в интерфейсе приложения. Для этого необходимо в окне свойств реквизита изменить значение поля Синоним, т.к. в интерфейсе приложения отображаются синонимы объектов.

Система поддерживает режим автоматической нумерации элементов, при котором она самостоятельно может генерировать код для нового элемента справочника. Кроме этого система позволяет осуществлять контроль уникальности кодов справочника, не разрешая создавать элементы с одинаковыми кодами:

Как уже говорилось выше, ни Код, ни Наименование **не являются** первичным ключом элемента справочника.

Помимо кода и наименования, каждый элемент справочника, как правило, содержит некоторую дополнительную информацию, которая подробно описывает этот элемент. Например, для товара это может быть информация об артикуле, номере государственной таможенной декларации, стране происхождения и т.п. Набор такой информации является одинаковым для всех элементов конкретного справочника, и для ее хранения служат реквизиты справочника:

Кроме этого, каждый элемент справочника может содержать некоторый набор информации, которая одинакова **по своей структуре**, **но различна по количеству**, для разных элементов справочника. Например, для каждого сотрудника в справочнике **Физические лица** это может быть контактная информация или информация о составе семьи, образовании. Для описания подобной информации могут быть использованы *табличные части* объекта конфигурации Справочник, являющиеся подчиненными ему объектами конфигурации. В этом случае в базе данных будут созданы дополнительные таблицы для хранения табличных частей, связанных по внешнему ключу с основной таблицей справочника.

Для каждого реквизита справочника необходимо задать тип данных, например, "число", "строка", "дата", булево (Истина или Ложь).

Это базовые типы, но можно указать и сложные типы данных (ссылочный тип данных). Например, реквизит Должность имеет тип данных Должности. В этом случае, значения этого реквизита будут выбираться из ранее созданного **справочника** Должности. Так реализуется

простейшая связь между справочниками, когда значения реквизитов одного справочника выбираются из элементов другого справочника.

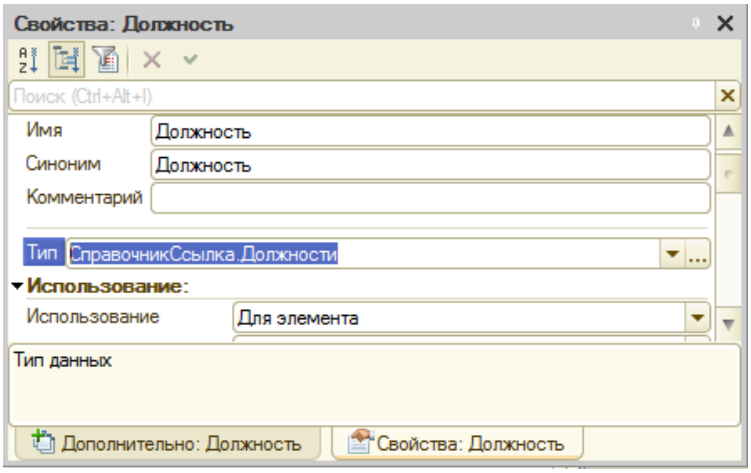



Рисунок 2.5 – “Свойства - Должность”.

Справочники могут поддерживать иерархическое расположение элементов. Например, в справочнике Бытовая техника могут быть созданы следующие *группы*: Холодильники, Телевизоры, Стиральные машины и т. д. Возможность создания таких групп в справочнике задается свойством **Иерархический** объекта конфигурации Справочник. В этом случае элемент справочника, представляющий собой группу, будет являться *родителем* для всех элементов и групп, входящих в эту группу. Такой вид иерархии называется ***иерархией групп и элементов***

В справочниках возможен и другой вид иерархии – *иерархия элементов*, при элементы справочника будут относиться не к группам, а к другим элементам этого же справочника. Такой вид иерархии может использоваться, например, при создании справочника **Подразделения**, где одно подразделение может включать в себя несколько других подразделений.

В первом случае родителем (папкой) может быть только папка (группа), а во втором случае родителем может быть и элемент.

*Размещать группы сверху* – флаг отвечает за отображение групп в форме списка.

Так же в настройках можно ограничить количество групп иерархии справочника, соответствующей настройкой.

Кроме того группы справочника могут включать в себя и другие группы, создавая тем самым многоуровневую иерархическую структуру.

Использование вида иерархии определяется текущей задачей. Если использование иерархии определено необходимостью визуально отделять и группировать данные в списке, то используется иерархия групп и элементов. Как правило, при такой иерархии группы не используются как значения в реквизитах других объектов. А если необходимо описать

вложенную структуру, где узлы могут быть значениями в данных, то используется иерархия элементов.

Разные справочники могут находиться в состоянии подчинения, т.е. элементы одного справочника могут быть подчинены элементам или группам другого справочника. В терминах реляционных баз данных, между таблицами устанавливается связь "один-ко-многим". В этом случае каждый элемент подчиненного справочника будет связан с одним из элементов справочника-владельца. Иногда можно сказать, что элементы одного справочника принадлежат элементам другого. Например, справочник **Кассы** может быть подчинен справочнику **Организации**. Тогда при оформлении кассовых документов для некоторой организации можно будет выбрать кассу не среди всех имеющихся в программе касс, а среди касс, существующих только в этой организации.

Справочники допускают также создание **предопределенных элементов**, которые существуют в справочнике всегда, вне зависимости от действий пользователя. Такие элементы справочника создаются разработчиком при разработке прикладного решения и не могут быть удалены или перемещены пользователем.

Например, если в основном прикладное решение будет внедряться на предприятиях одного государства, России, в справочник **Страны мира** сразу же можно добавить предопределенный элемент **Россия**, поскольку он понадобится подавляющему большинству пользователей.

В интерфейсе предопределенные элементы справочника помечены специальной пиктограммой  :

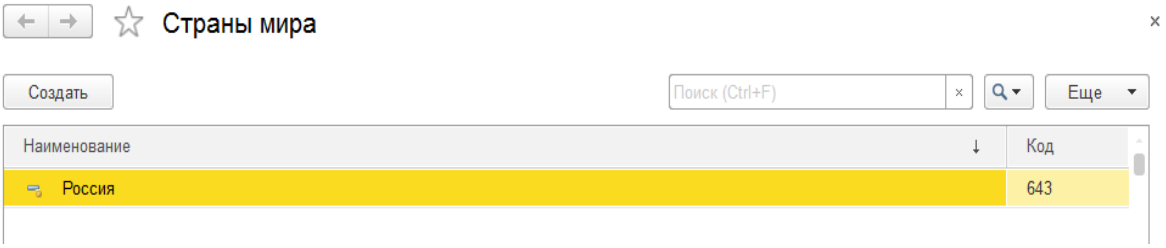



Рисунок 2.6 – Интерфейс “Переопределенные элементы”.

## Формы справочника

Для того, чтобы пользователь мог просматривать и изменять данные, содержащиеся в справочнике, система поддерживает несколько форм представления справочника. Система может автоматически генерировать все нужные формы справочника. Наряду с этим разработчик имеет возможность создать собственные формы, которые система будет использовать вместо форм по умолчанию.

## Форма списка

Для просмотра данных, содержащихся в справочнике, используется форма списка. Она позволяет выполнять навигацию по справочнику, добавлять, помечать на удаление и удалять элементы и группы справочника, перемещать элементы и группы. Форма списка может представлять данные в иерархическом и не иерархическом виде и позволяет выполнять сортировку и отбор отображаемой информации по нескольким критериям:

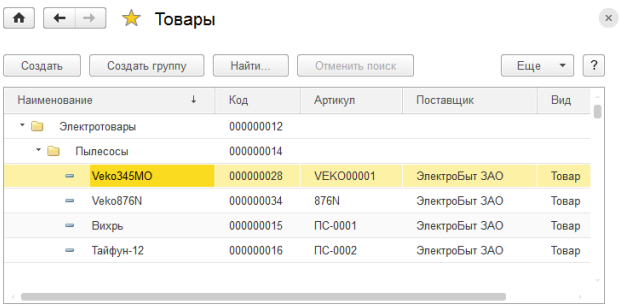



Рисунок 2.7 – Просмотр данных в Справочнике.

## Форма элемента

Для просмотра и изменения данных отдельных элементов справочника используется форма элемента. Как правило, она представляет данные в удобном для восприятия и редактирования виде:

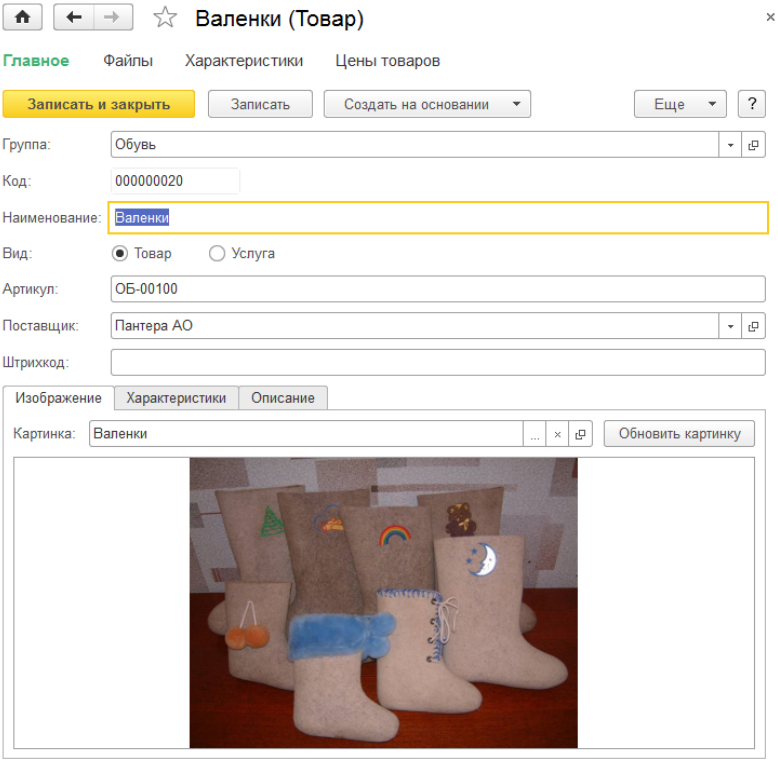



Рисунок 2.8 – Форма Элемента.

## Форма группы

Система поддерживает отдельный вид формы для групп справочника. Это связано с тем, что, как правило, состав информации, относящейся к группе справочника, намного меньше, чем состав данных, имеющих отношение к отдельным элементам справочника:

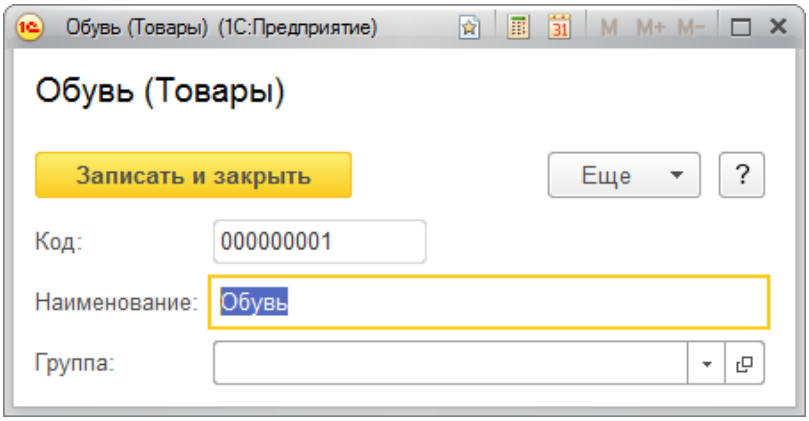



Рисунок 2.9 – Форма Элемента.

## Форма выбора, форма выбора группы

При различных действиях пользователя в прикладном решении возникает необходимость выбора конкретных элементов справочников. Это может потребоваться, например, при создании приходной накладной, в которой должны быть перечислены определенные товары. В этом случае система использует форму выбора справочника, которая, обычно, содержит минимальный набор информации, необходимой для выбора того, или иного элемента:

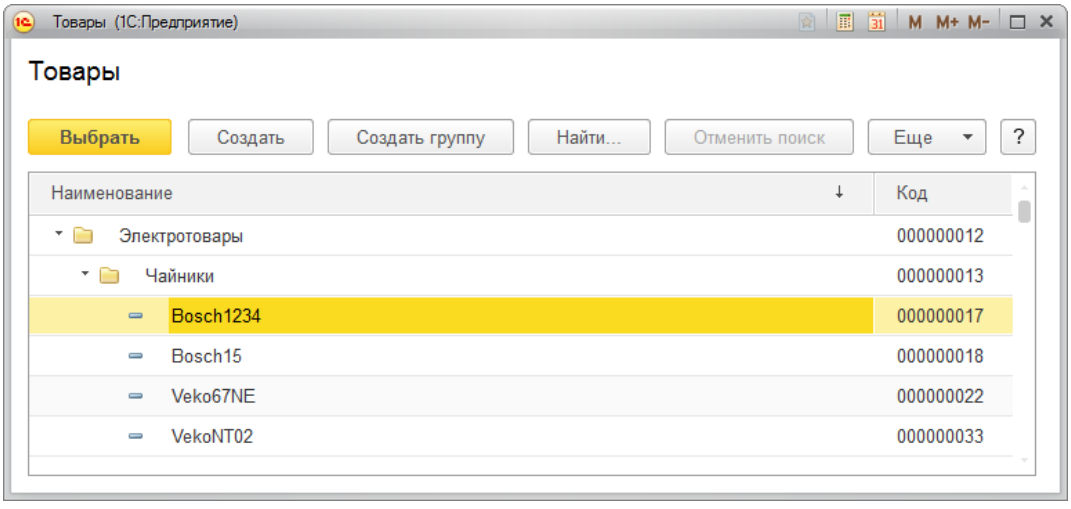



Рисунок 2.10 – Просмотр товаров  и их кодов.

## Программный код для работы со справочниками

На следующей схеме изображено взаимодействие объектов встроенного языка для работы со справочниками. Заливкой выделен объект манипулирования данными.

Ниже приведены примеры использования объектов встроенного языка для работы со справочниками.

Данные примеры программного кода должны размещаться в процедурах и функциях (или модулях),выполняемых на стороне сервера.

```
// 1. Глобальный контекст
//    Справочники

// Пример: вывести все типы ссылок на элементы справочников, существующих в
конфигурации.
Массив = Справочники.ТипВсеСсылки().Типы();
Для Каждого ОчереднойТип из Массив Цикл
    Сообщить(ОчереднойТип);
КонецЦикла;

// 2. объект СправочникиМенеджер
//    .<имя справочника>
//    [<имя справочника>]
//    Для Каждого ... Из ... Цикл ... КонецЦикла;

// Пример: создать новую группу справочника "Номенклатура".
НоваяГруппа = Справочники.Номенклатура.СоздатьГруппу();
НоваяГруппа.Наименование = "Моя новая группа";
НоваяГруппа.Записать();

// Пример: получить ссылку на справочник "Номенклатура".
Справочники["Номенклатура"].ПолучитьСсылку();

// 3. объект СправочникиМенеджер.<Имя справочника>
//    НайтиПоКоду()
//    НайтиПоНаименованию()
//    НайтиПоРеквизиту()
//    ПустаяСсылка()
//    ПолучитьСсылку()
//    .<имя предопределенного элемента справочника>

// Пример: проверить, помечен ли на удаление элемент справочника
"Номенклатура" с кодом 13.
Если Справочники.Номенклатура.НайтиПоКоду(13).ПометкаУдаления Тогда
    Сообщить("Элемент с кодом 13 помечен на удаление");
КонецЕсли;

// Пример: является ли элемент справочника "Номенклатура" с наименованием
"Услуги" группой.
Если Справочники.Номенклатура.НайтиПоНаименованию("Услуги",
Истина).ЭтоГруппа Тогда
    Сообщить("Элемент Услуги является группой");
КонецЕсли;

// Пример: проверить, что для всех элементов задан вид номенклатуры.
ПустаяСсылкаПеречисления = Перечисления.ВидыНоменклатуры.ПустаяСсылка();
Если Не Справочники.Номенклатура.НайтиПоРеквизиту("ВидНоменклатуры",
ПустаяСсылкаПеречисления).Пустая() Тогда
    Сообщить("Есть элементы, для которых не задан вид номенклатуры");
КонецЕсли;
```

```
// Пример: передать пустую ссылку в параметр метода.
Выборка =
Справочники.Номенклатура.Выбрать(Справочники.Номенклатура.ПустаяСсылка());

// 4. объект СправочникиМенедежер.<Имя справочника>
//    Выбрать()
//    ВыбратьИерархически()

// Пример: вывести список элементов, расположенных в корне справочника.
Выборка =
Справочники.Номенклатура.Выбрать(Справочники.Номенклатура.ПустаяСсылка());
Пока Выборка.Следующий() Цикл
    Если Не Выборка.ЭтоГруппа Тогда
        Сообщить(Выборка);
    КонецЕсли;
КонецЦикла;

// Пример: удалить все элементы иерархического справочника.
Выборка = Справочники.Номенклатура.ВыбратьИерархически();
Пока Выборка.Следующий() Цикл
    Выборка.Удалить();
КонецЦикла;

// 5. объект СправочникМенеджер.<Имя справочника>
//    СоздатьГруппу()
//    СоздатьЭлемент()

// Пример: создать новый элемент справочника "Сотрудники".
НовыйЭлемент = Справочники.Сотрудники.СоздатьЭлемент();
НовыйЭлемент.Наименование = "Смирнов Андрей Анатольевич";
// Заполнить табличную часть "ТрудоваяДеятельность".
НоваяСтрокаТабличнойЧасти = НовыйЭлемент.ТрудоваяДеятельность.Добавить();
НоваяСтрокаТабличнойЧасти.Организация = "ООО НТЦ";
НоваяСтрокаТабличнойЧасти.НачалоРаботы = Дата(2013,02,01);
НоваяСтрокаТабличнойЧасти.ОкончаниеРаботы = Дата(2013,08,31);
НоваяСтрокаТабличнойЧасти.Должность = "Программист";
НовыйЭлемент.Записать();

// 6. объект СправочникОбъект.<Имя справочника>, СправочникСсылка.<Имя
справочника>
//    Владелец
//    Родитель
//    Ссылка

// Пример: запретить изменение подчиненных элементов, если у
// владельца установлено соответствующее свойство
// "ИзмененияЗапрещены" в модуле формы элемента справочника.
Процедура ПередЗаписью(Отказ)
    Если Владелец.ИзмененияЗапрещены Тогда Отказ = Истина;
    КонецЕсли;
КонецПроцедуры

// 7. объект СправочникСсылка.<Имя справочника>
//    ПолучитьОбъект()
//    СправочникиОбъект.<Имя справочника>
//    Скопировать()

// Пример: изменить наименование элемента справочника.
Элемент = Справочники.Номенклатура.НайтиПоКоду(10).ПолучитьОбъект();
Элемент.Наименование = "Мое новое наименование";
Элемент.Записать();

// Пример: заполнить справочник тестовыми данными.
```

```
Элемент = Справочники.Номенклатура.СоздатьЭлемент();
Элемент.Наименование = "Тестовый элемент";
Элемент.Записать();
Для ш = 1 по 1000 Цикл
    НовыйЭлемент = Элемент.Скопировать();
    НовыйЭлемент.Записать();
КонецЦикла;

// 8. объект СправочникВыборка.<Имя справочника>
//    Ссылка

// Пример: заполнить табличную часть документа
// "ПриходнаяНакладная" всеми элементами из указанной группы справочника
"Номенклатура".
Выборка = Справочники.Номенклатура.ВыбратьИерархически(ПолеВвода1);
Пока Выборка.Следующий() Цикл
    ССылкаНаНоменклатуру = Выборка.Ссылка;
    Если ССылкаНаНоменклатуру.ЭтоГруппа Тогда Продолжить;
    КонецЕсли;
    НоваяСтрока = Материалы.Добавить();
    НоваяСтрока.Материал = ССылкаНаНоменклатуру;
КонецЦикла;

// 9. объект СправочникВыборка.<Имя справочника>
//    ПолучитьОбъект()

// Пример: пометить все элементы неиерархического справочника на удаление.
Выборка = Справочники.Клиенты.Выбрать();
Пока Выборка.Следующий() Цикл
    Выборка.ПолучитьОбъект().УстановитьПометкуУдаления(Истина);
КонецЦикла;
```

Последовательность событий при записи элемента справочника из формы элемента (записать и закрыть). Заливкой выделены события, выполняющиеся в транзакции записи.

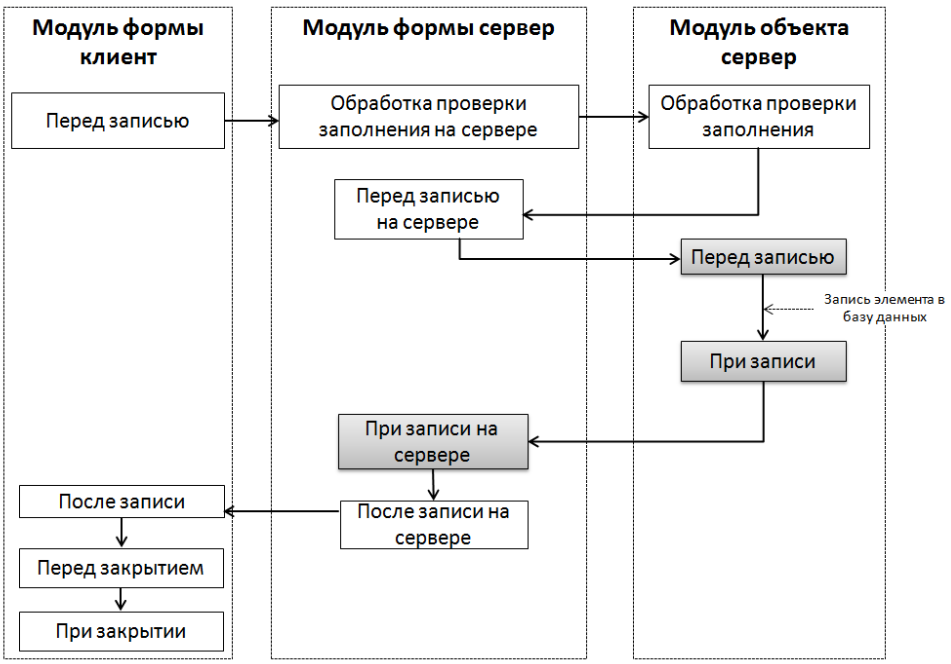



Рисунок 2.11 – Последовательность событий при записи элемента справочника.

Информация в справочниках, как правило, часто не изменяется. Справочники в дальнейшем используются практически во всех объектах учета как разрез учета или справочная информация. *Информация из справочника используется, например, в документах.*

### 1.2.2.4. Документы

Объект конфигурации *Документ* предназначен для описания информации о совершенных хозяйственных операциях или о событиях, произошедших в жизни организации вообще. Свойства и структура этих документов описываются в объектах конфигурации Документ, на основе которых платформа создает в базе данных таблицы для хранения информации из этих документов. Например, документом является *Экзаменационная ведомость*, которая содержит информацию о дате проведения экзамена, наименовании предмета, экзаменаторе и об экзаменационной оценке конкретного студента.

В большинстве своем документы, которые создаются в процессе настройки конфигурации задачи, являются электронными аналогами стандартных бумажных документов, являющихся основаниями для тех или иных учетных действий или расчетов. Тем не менее, использование этого типа данных может выходить далеко за рамки простой фиксации изменений, внесенных в учетные регистры.

Логика работы документов отличается от логики работы других объектов конфигурации. Документ обладает способностью *проведения*. Факт проведения документа означает, что событие, которое он отражает, повлияло на состояние учета.

Документ может иметь три состояния:

- **создан;**
- **помечен на удаление;**
- **проведен**.

До тех пор, пока документ не проведен, состояние учета неизменно, и документ не более чем черновик, заготовка. Как только документ будет проведен, изменения, вносимые документом в учет, вступят в силу и состояние учета будет изменено.

Система 1С дает возможность изменять проведенный документ, отменять его проведение.

**Структура документа**

Структура каждого конкретного вида документа определяется при его создании в конфигураторе. У любого вида документа существуют реквизиты, которые создаются автоматически – это "Дата" и "Номер". Номер создается, если при конфигурировании длина номера указана больше 0. Другие реквизиты документа определяются в конфигураторе отдельно для каждого создаваемого вида документа.

Также как и элемент справочника каждый документ определяется уникальным идентификатором в базе данных – **Ссылкой**. Номер идентификатором документа не является и может быть измен для конкретного объекта.

**Проведен** – булевское свойство, содержащее информацию о том, проведен документ или нет.

Пометка удаления это булевский признак, содержащий информацию о том, помечен документ на удаление или нет.

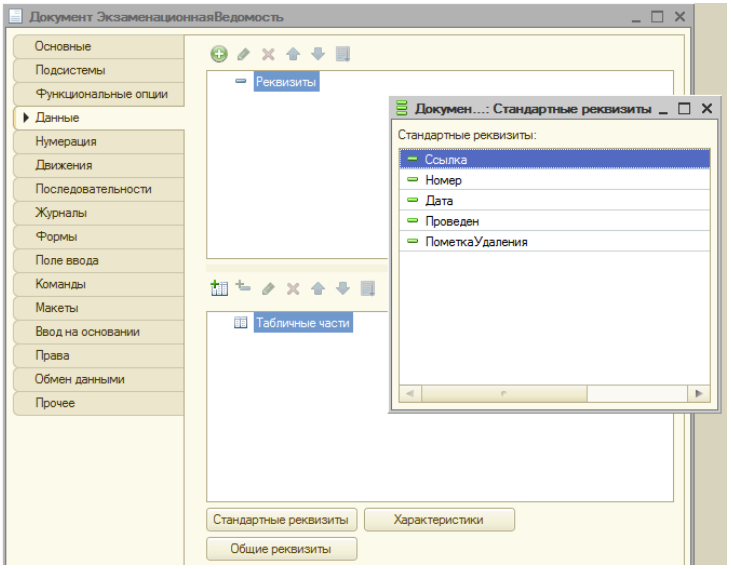



Рисунок 2.12 – “Документ – Экзаменационная ведомость”.

Система поддерживает режим автоматической нумерации документов, при котором она самостоятельно может генерировать номер для нового документа. Кроме этого система позволяет осуществлять контроль уникальности номеров документов, не разрешая создавать документы с одинаковыми номерами:

Система автоматически поддерживает режим, при котором уникальность номеров и автоматическая нумерация могут выполняться в пределах определенного периода (день, месяц, квартал, год). Например, если периодичность установлена год, то с нового года система опять начнет нумерацию указанных документов с 1.

Важными характеристиками документа являются дата и время. Они позволяют установить строгую временную последовательность совершения операций. Таким образом, документы могут отличаться друг от друга не только номером, но и своим положением на временной оси. В результате всегда можно сказать, какая из двух операций была совершена раньше.

Помимо номера, даты и времени, каждый документ, как правило, содержит некоторую дополнительную информацию, которая подробно описывает этот документ. Например, для документа **Поступление товаров и услуг** это может быть информация о поставщике товаров (контрагенте), складе, на который приходуются товары и т.п. Набор такой информации

является одинаковым для всех документов конкретного вида, и для ее хранения служат реквизиты документа:

Кроме этого, каждый документ может содержать некоторый набор информации, которая одинакова по своей структуре, но различна по количеству, для разных документов. Например, для документа **Поступление товаров и услуг** это может быть информация о товарах, поступивших на предприятие (наименование, количество, и т.д.), серийных номерах и возвратной таре. Для хранения подобных данных служат табличные части документа.

Для рассматриваемого выше документа Экзаменационная ведомость реквизитами документа, помимо стандартных, будут являться:

Кроме этого, документ будет содержать некоторый набор информации, которая одинакова по своей структуре, но различна по количеству (табличная часть):

Для некоторых реквизитов указан ссылочный тип данных. Этот тип стал доступен в конфигурации после создания объекта конфигурации Справочник.

## Проведение документа

Как уже отмечалось ранее, важным свойством документа является возможность его проведения. Если документ проводится, то он может изменить состояние тех или иных учитываемых данных. Если же документ не является "проводимым" это значит, что событие, которое он отражает, не влияет на состояние учета, который ведется в данном прикладном решении.

Например, документ **Поступление товаров и услуг** при своем проведении может вносить изменения в состояние расчетов с поставщиками, в учет остатков товаров, в состояние заказов покупателей и другие учетные данные.

Для документа **Коммерческое предложение контрагенту** напротив, проведение может быть запрещено, потому что этот документ никак не влияет на состояние учета.

## Конструктор движений

Алгоритм, на основании которого документ вносит те или иные изменения в состояние учетных данных при своем проведении, описывается средствами встроенного языка на этапе разработки прикладного решения. Система содержит **конструктор движений**, который помогает разработчику создавать алгоритмы проведения документа.

## Формы документа

Для того, чтобы пользователь мог просматривать и изменять данные, содержащиеся в документах, система поддерживает несколько форм

представления документа. Система может автоматически генерировать все нужные формы документа. Наряду с этим разработчик имеет возможность создать собственные формы, которые система будет использовать вместо форм по умолчанию:

### Форма списка

Для просмотра документов одного вида используется форма списка. Она позволяет выполнять навигацию по списку документов, задавать временной интервал отображаемых документов, добавлять, помечать на удаление и удалять документы. Форма списка позволяет выполнять сортировку и отбор отображаемой информации по нескольким критериям:

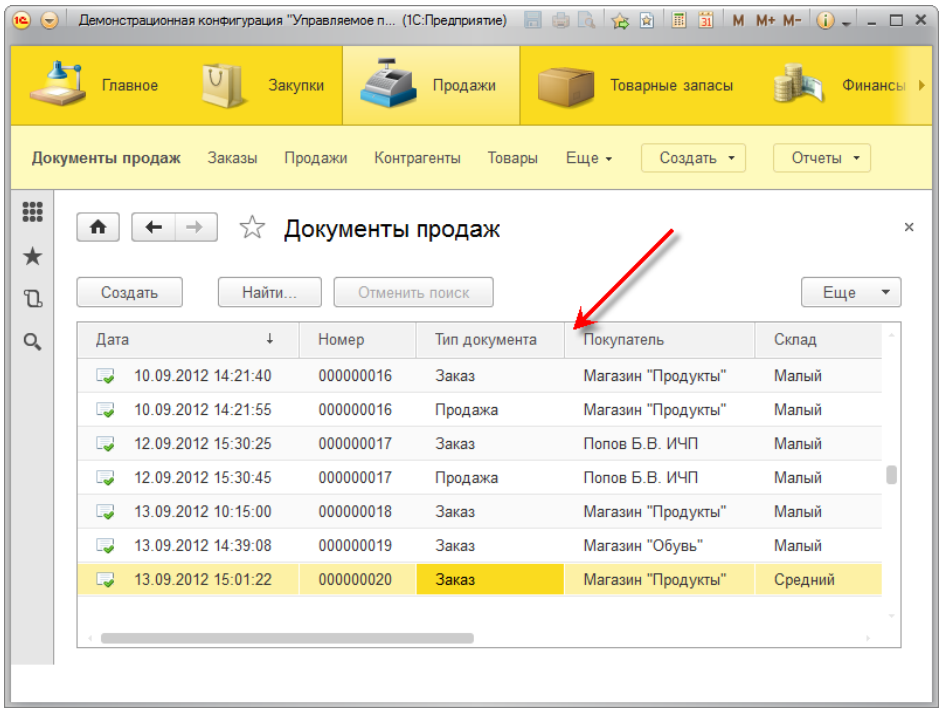



Рисунок 2.13 – Формы списка.

### Форма документа

Для просмотра и изменения данных отдельных документов используется форма документа. Как правило, она представляет данные в удобном для восприятия и редактирования виде:

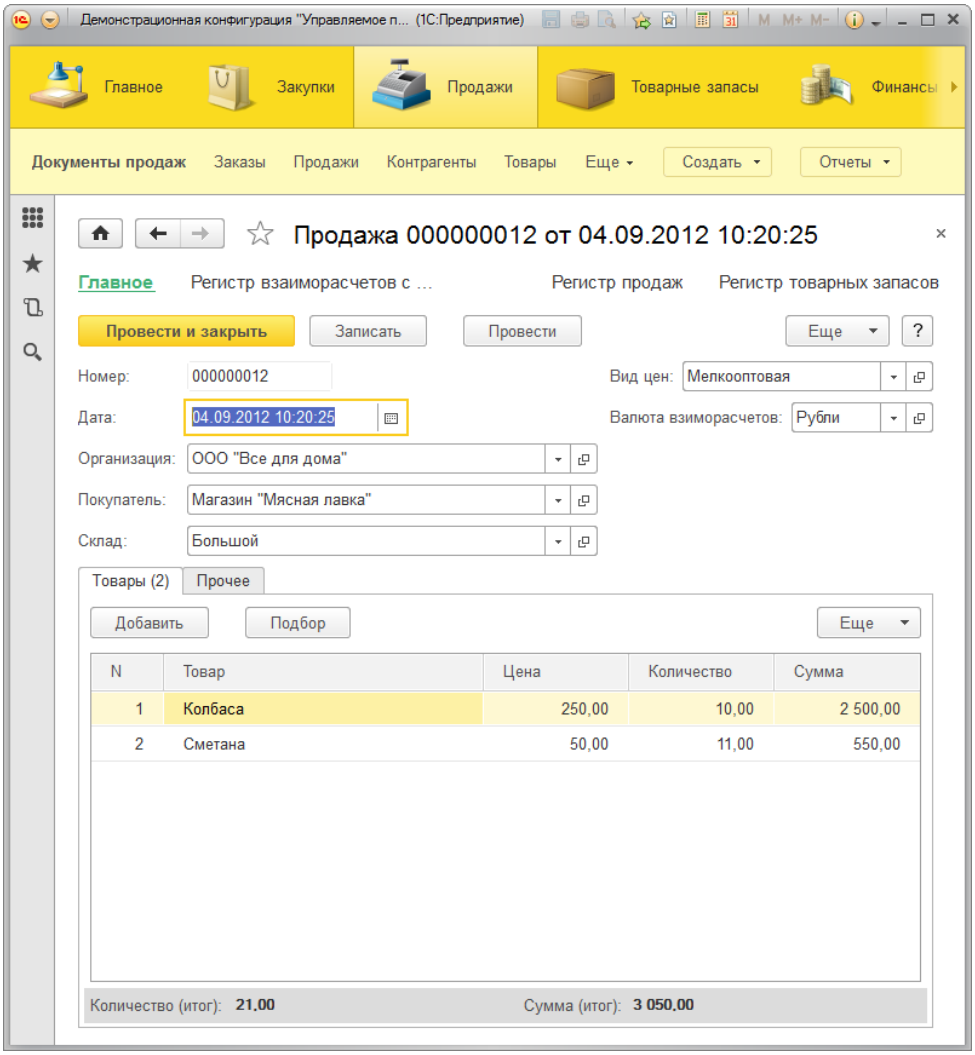



Рисунок 2.14 – Форма документа.

### Форма выбора

Кроме этих двух форм для документа поддерживается форма выбора конкретных документов из списка. Она, обычно, содержит минимальный набор информации, необходимой для выбора того или иного документа и может использоваться, например, при создании документа **Поступление товаров и услуг**, в котором должен быть указан заказ поставщику, на основании которого формируется поступление товаров.

Программный код для работы с документами

На следующей схеме изображено взаимодействие объектов встроенного языка для работы с документами. Заливкой выделен объект манипулирования данными.

Метод объекта, от которого идет стрелка, приводится в далее листинге под соответствующей цифрой, а объект, к которому идет стрелка, – это тип объекта, возвращаемого методом.

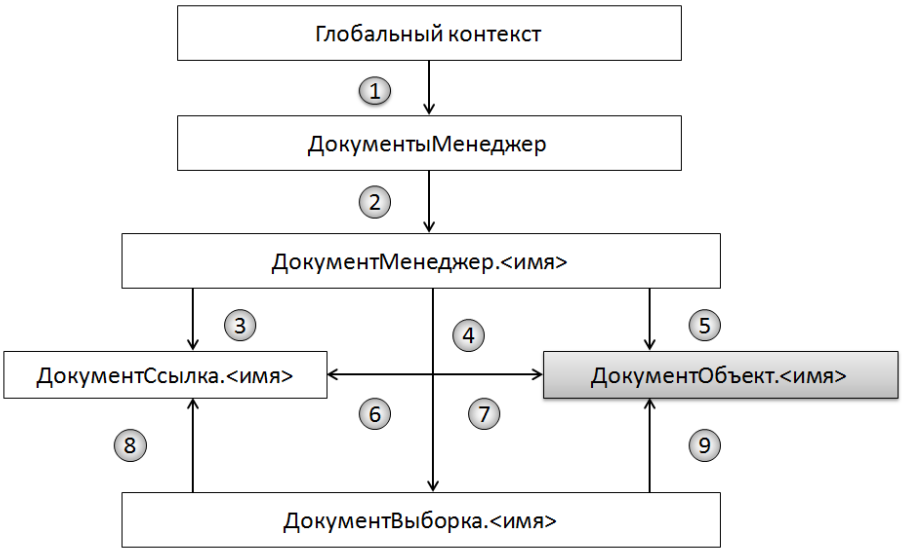



Рисунок 2.15 – Взаимодействие объектов встроенного языка для работы с документами.

Ниже приведены примеры использования объектов встроенного языка для работы с документами.

```
// 1. Глобальный контекст
//   Документы

// Пример: вывести все типы ссылок на элементы документов, существующие в конфигурации.
Массив = Документы.ТипВсеСсылки().Типы();
Для Каждого ОчереднойТип из Массив Цикл
   Сообщить(ОчереднойТип);
КонецЦикла;

// 2. объект ДокументМенедежер
//   .<имя документа>
//   [<имя документа>]
//   Для Каждого ... Из ... Цикл ... КонецЦикла;

// Пример: получить макет для печати документа "Оказание услуги".
Макет = Документы["ОказаниеУслуги"].ПолучитьМакет("Печать");
```

```
// Пример: получить ссылку на каждый из документов, существующих в конфигурации.
Для Каждого ОчереднойДокумент Из Документы Цикл
    Ссылка = ОчереднойДокумент.ПолучитьСсылку();

    ...
КонецЦикла;

// 3. объект ДокументМенеджер.<Имя документа>
//    НайтиПоНомеру()
//    НайтиПоРеквизиту()
//    ПустаяСсылка()

// Пример: проверить, проведен ли документ ПриходнаяНакладная с номером 3.
Если Документы.ПриходнаяНакладная.НайтиПоНомеру(3).Проведен Тогда
    Сообщить("Документ с номером 3 проведен");
КонецЕсли;

// Пример: проверить, что во всех документах ПриходнаяНакладная заполнен реквизит Склад.
ПустаяСсылкаСклада = Справочники.Склады.ПустаяСсылка();
Если Не Документы.ПриходнаяНакладная.НайтиПоРеквизиту("Склад", ПустаяСсылкаСклада).Пустая() Тогда
    Сообщить("Есть документы, у которых не заполнен реквизит Склад");
КонецЕсли;

// 4. объект ДокументыМенедежер.<Имя документа>
//    Выбрать()

// Пример: выбрать все документы ПриходнаяНакладная за текущий месяц.
Выборка = Документы.ПриходнаяНакладная.Выбрать(НачалоМесяца(ТекущаяДата()), КонецМесяца(ТекущаяДата()));
Пока Выборка.Следующий() Цикл
    Сообщить(Выборка);
КонецЦикла;

// 5. объект ДокументМенеджер.<Имя документа>
//    СоздатьДокумент()

// Пример: создать новый документ ПриходнаяНакладная.
НовыйДокумент = Документы.ПриходнаяНакладная.СоздатьДокумент();
```

```
НовыйДокумент.Дата = ТекущаяДата();
НовыйДокумент.Склад = Справочники.Склады.Основной;
// заполнить табличную часть Материалы
НоваяСтрокаТабличнойЧасти =
НовыйДокумент.Материалы.Добавить();
НоваяСтрокаТабличнойЧасти.Материал =
Справочники.Номенклатура.НайтиПоКоду(6);
НоваяСтрокаТабличнойЧасти.Количество = 10;
НоваяСтрокаТабличнойЧасти.Цена = 22.5;
НоваяСтрокаТабличнойЧасти.Сумма = 225;
НовыйДокумент.Записать();

// 6. объект ДокументОбъект.<Имя документа>, объект
ДокументСсылка.<Имя документа>
// Ссылка

// Пример: в модуле объекта вызвать процедуру проверки заполнения
реквизитов документа.
Если Не ПроверитьЗаполнениеРеквизитов(ЭтотОбъект.Ссылка) Тогда
   Сообщить("Реквизиты документа не заполнены!");
КонецЕсли;

// 7. объект ДокументСсылка.<Имя документа>, объект
ДокументОбъект.<Имя документа>
// ПолучитьОбъект()
// Скопировать()

// Пример: пометить документ на удаление.
НенужныйДокумент =
Документы.ОказаниеУслуги.НайтиПоНомеру(13).ПолучитьОбъект();
НенужныйДокумент.УстановитьПометкуУдаления(Истина);

// 8. объект ДокументВыборка.<Имя документа>
// Ссылка

// Пример: сформировать список документов ПриходнаяНакладная за
текущий месяц.
СписокНакладных = Новый СписокЗначений;
Выборка =
Документы.ПриходнаяНакладная.Выбрать(НачалоМесяца(ТекущаяДата()
), КонецМесяца(ТекущаяДата()));
Пока Выборка.Следующий() Цикл
   СписокНакладных.Добавить(Выборка.Ссылка);
КонецЦикла;
```

```
// 9. объект ДокументВыборка.<Имя документа>
//    ПолучитьОбъект()

// Пример: удалить все документы ПриходнаяНакладная.
Выборка = Документы.ПриходнаяНакладная.Выбрать();
Пока Выборка.Следующий() Цикл
    Выборка.ПолучитьОбъект().Удалить();
КонецЦикла;
```

Последовательность событий при записи документа из формы документа. Заливкой выделены события, выполняющиеся в транзакции записи.

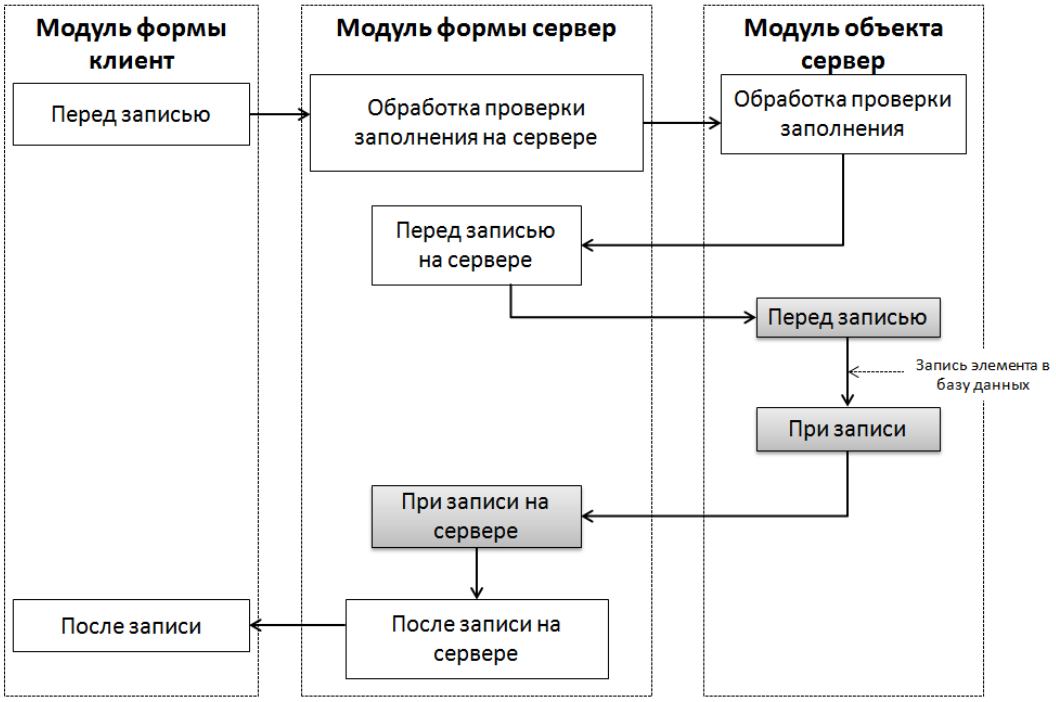



Рисунок 2.16 – Последовательность событий при записи документа из формы документа.

Последовательность событий при проведении документа из формы документа (провести и закрыть). Заливкой выделены события, выполняющиеся в транзакции записи.

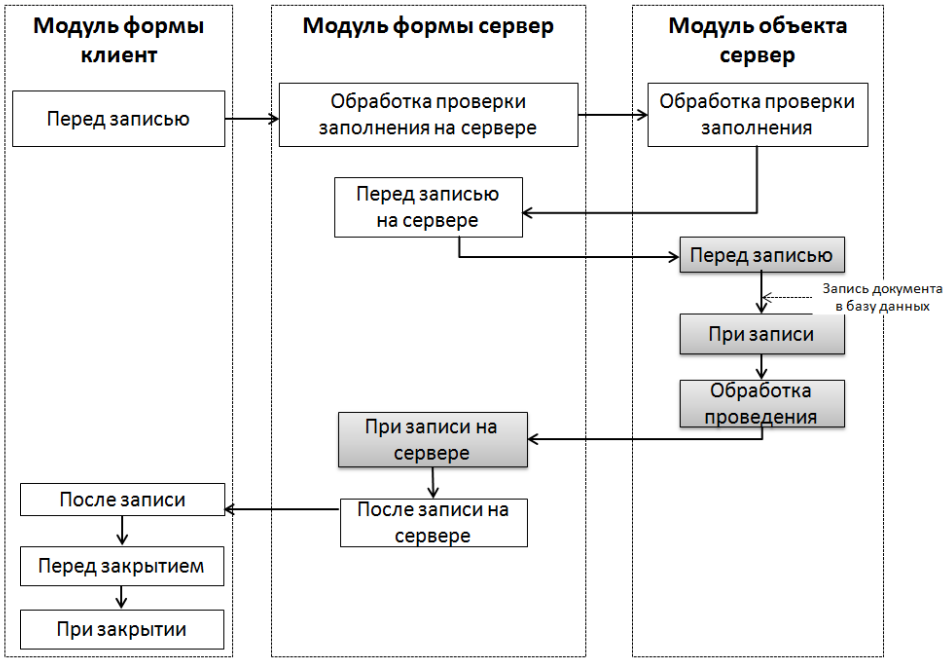



Рисунок 2.17 – Последовательность событий при записи документа из формы документа.

Последовательность событий при отмене проведения документа из формы документа. Заливкой выделены события, выполняющиеся в транзакции записи.

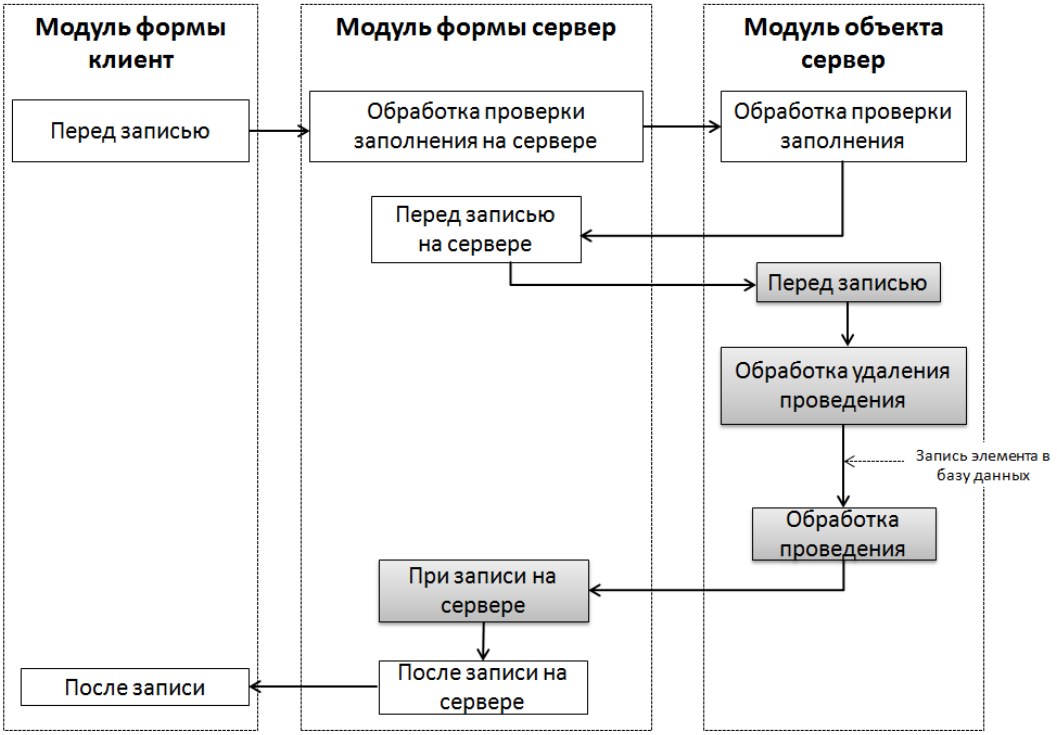



Рисунок 2.18 – Последовательность событий при отмене проведения документа.

## 1.3. ПРИМЕР ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

### Система кадрового учета

*Постановка задачи*. Необходимо создать систему кадрового учета сотрудников на предприятии, которая позволит принимать и увольнять сотрудников, а также работать с кадровыми документами.

*Ход работы:*

1. Откройте дерево конфигурации.
2. Установите курсор на строку «Справочники» и нажмите кнопку . При этом откроется **Окно редактирования объекта конфигурации.**
3. На первом шаге (вкладке) задайте «Имя» справочника (идентификатор) – «Должности».
4. Перейдите на вкладку «Данные». Задайте длину наименования – 30 символов.

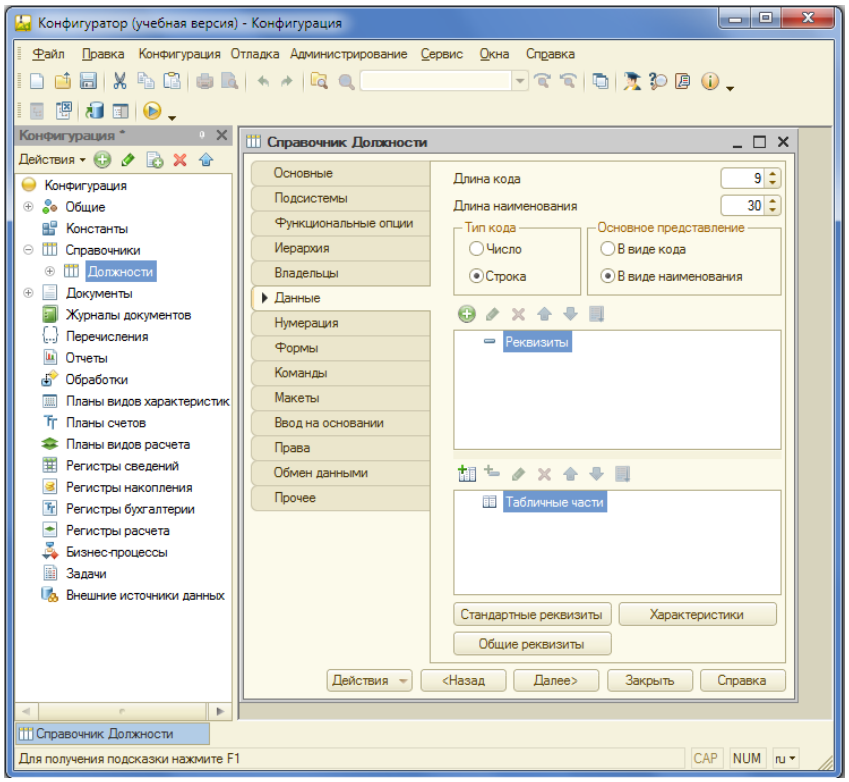



Рисунок 3.2 – Конфигуратор “Справочник – Должности - Данные”.

5. Закройте Окно редактирования справочника «Должности».
6. Создайте новый справочник «Сотрудники».

7. Задайте длину наименования 100 символов. В наименовании будет храниться ФИО сотрудника.

В этом справочнике кроме стандартных реквизитов (Код и Наименование) будут заданы еще несколько.

9. Задайте в палитре свойств имя реквизита – «Должность». Укажите тип реквизита – «СправочникСсылка.Должности», который был создан ранее. Это говорит о том, что значения данного реквизита будут выбираться из справочника «Должности».

11. Создайте реквизит «Оклад» (тип Число, Длина 10, Точность 2). У значения типа «Число» точность указывает на количество знаков после запятой.

12. Создайте реквизиты «ДатаПриема» и «ДатаУвольнения» (тип Дата).

13. Обновите конфигурацию базы данных.

14. Запустите систему в режиме 1С:Предприятие.

15. Откройте справочник «Должности».

16. Введите несколько должностей, используя кнопку  или клавишу INS. Например, директор, бухгалтер, программист.

Заметьте, что коды присваиваются автоматически, хотя их можно редактировать.

По умолчанию установлена сортировка по Наименованию. Это позволяет набрать первые буквы названия должности, и курсор автоматически перейдет на нужную должность. Можно установить и другой тип сортировки: по коду или по реквизиту. Для этого достаточно щелкнуть левой кнопкой мышки по названию (заголовку) реквизита.

17. Откройте справочник «Сотрудники».

18. Введите нескольких сотрудников. Например, Иванов Иван Иванович – директор, Петров Петр Петрович – бухгалтер, Сидоров Сергей Сергеевич – программист. Заполните только реквизиты Наименование (ФИО) и Должность. Заметьте, что при указании должности открывается справочник Должности.

19. Для сохранения нового элемента справочника (новой записи) нажмите «Записать и закрыть».

20. Вернитесь в режим Конфигуратора.

21. Создайте новый документ «ПриемНаРаботу».

22. Перейдите на закладку документа Данные.

23. Добавьте реквизит «ФИО», укажите ему тип значения – Строка, длина – 100 символов.

Далее нужно создать реквизиты:

- реквизит «Должность» – тип «СправочникСсылка.Должности»;
- реквизит «Оклад» – тип Число, длина 10, точность 2;
- реквизит «ДатаПриема» – тип Дата.

Теперь в этот документ можно будет вводить данные о новом сотруднике (в режиме 1С:Предприятие).

При нажатии кнопки «Провести и закрыть» документ сначала будет записан, потом проведен, а потом закрыта его форма.

При проведении выполнится предопределенная процедура **ОбработкаПроведения()**. Именно в этой процедуре записывается

программный код на языке 1С, который и совершает все изменения в системе. Данная процедура хранится в «Модуле объекта».

25. Для открытия модуля объекта перейдите на закладку «Прочее» и нажмите кнопку «Модуль объекта». Другой способ перейти в модуль объекта – выбрать в дереве конфигурации документ «ПриемНаРаботу» и кликнув правой кнопкой мышки выбрать «Открыть модуль объекта».

В результате откроется окно для редактирования текста программы.

26. На панели инструментов найдите пиктограмму «Процедуры и функции»  и выберете «ОбработкаПроведения».

В результате в модуле объекта появилась процедура **ОбработкаПроведения**.

27. Напишите там следующий текст:

```
СпрСотр = Справочники.Сотрудники.СоздатьЭлемент();
СпрСотр.Наименование = ФИО;
СпрСотр.Должность = Должность;
СпрСотр.Оклад = Оклад;
СпрСотр.ДатаПриема = ДатаПриема;
СпрСотр.Записать();
```

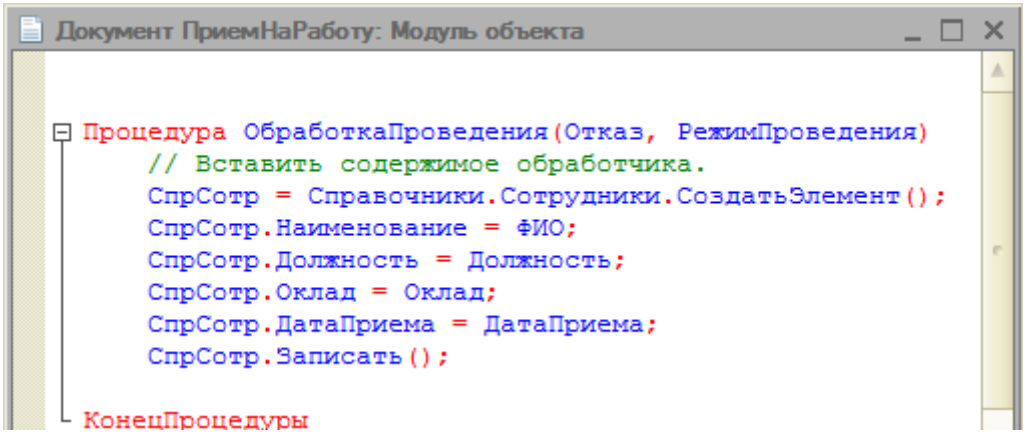



Рисунок 3.11 – Документ “Прием на работу”.

Если не вызвать метод **Записать()**, то изменения не сохранятся в базе.

Обратите внимание, что каждый оператор заканчивается символом «;». После слова **КонецПроцедуры** точки с запятой нет, т.к. это не оператор, а операторная скобка. Точки с запятой нет также после слов **Цикл** и **Тогда**, но после слов **КонецЦикла** и **КонецЕсли** обязательно ставится точка с запятой.

28. Закройте окно модуля и сохраните конфигурацию.

29. Запустите 1С в режиме «1С:Предприятие».

30. Создайте документ «Прием на работу». Номер документа присвоится автоматически после его записи, дата документа – сегодняшнее число.

Введите все данные: ФИО сотрудника, укажите ему должность, оклад и дату приема на работу.

31. Нажмите кнопку «Провести и закрыть». Сначала документ будет записан, потом проведен, а затем его форма будет закрыта. При проведении сработает процедура **ОбработкаПроведения** и в справочник Сотрудники добавится новая строка.

32. Вернитесь в режим Конфигуратора.

33. Создайте в Конфигураторе новый документ «Увольнение». Укажите реквизиты «Сотрудник» тип «СправочникСсылка.Сотрудники» и «ДатаУвольнения» тип Дата.

34. Напишите следующий текст в модуле документа в процедуре ОбработкаПроведения():

```
СпрСотр=Сотрудник.ПолучитьОбъект();
СпрСотр.ДатаУвольнения=ДатаУвольнения;
СпрСотр.Записать();
```

В первой строке кода из базы данных будет получена запись с сотрудником, который будет уволен. МетодПолучитьОбъект() позволяет не только прочить значения реквизитов данного объекта, но и изменять их. Следующая строка устанавливает ему дату увольнения. И последняя строка записывает сделанные изменения в справочник «Сотрудники».

35. Создайте журнал «КадровыеДокументы». Для этого установите курсор на ветку «Журналы документов» в дереве конфигурации и щелкните «Добавить». Задайте имя журнала.

На закладке «Данные» укажите регистрируемые документы: «ПриемНаРаботу» и «Увольнение».

Создайте новые графы:

- Сотрудник – для документа «ПриемНаРаботу» это реквизит «ФИО», а для документа «Увольнение» – это реквизит «Сотрудник»;
- ДатаПриемаУвольнения – для документа «ПриемНаРаботу» это реквизит «ДатаПриема», а для документа «Увольнение» – это реквизит «ДатаУвольнения».

36. Сохраните конфигурацию и запустите ее в режиме «1С:Предприятие».

37. Проверьте работу документа "Увольнение", а затем откройте журнал "Кадровые документы".

38. Вернитесь в режим Конфигуратора.

Далее необходимо сделать так, чтобы данные о работающих и уволенных сотрудниках хранились в разных группах.

39. В дереве конфигурации выделите справочник «Сотрудники, откройте Окно редактирования справочника.

40. Перейдите на закладку «Иерархия».

41. В окне редактирования справочника «Сотрудники» на закладке «Иерархия» укажите, что справочник иерархический. Задайте ограничение количества уровней иерархии – 2.

Это позволит на 1 уровне создать две группы «Работающие» и «Уволенные», а уже в этих группах (на 2 уровне) размещать сведения о соответствующих сотрудниках.

42. Запустите систему в режиме 1С:Предприятие.

43. Откройте справочник «Сотрудники». Создайте 2 группы:

- «Работающие»
- «Уволенные».

Для создания новых групп можно воспользоваться горячими клавишами **Ctrl + F9**, либо использовать пиктограмму на командной панели.

44. С помощью контекстного меню перенесите сотрудников по соответствующим группам: работающих – в группу «Работающие», уволенных – в группу «Уволенные».

45. Вернитесь в режим Конфигуратора.

Для того, чтобы при создании нового документа «Прием на работу» новый сотрудник создавался в группе «Работающие», необходимо внести правки в код процедуры **ОбработкаПроведения()** модуля объекта документа «Прием на работу».

Первой строкой кода будет:

```
Работающие = Справочники.Сотрудники.НайтиПоНаименованию("Работающие");
```

В этой строке переменной работающие присваивается значение – ссылку на группу «Работающие» справочника «Сотрудники». Для того, чтобы присвоить ссылку, вначале необходимо найти этот элемент по его наименованию.

Затем идет прежняя первая строка кода:

```
СпрСотр = Справочники.Сотрудники.СоздатьЭлемент();
```

После нее вставьте еще одну новую строку:

```
СпрСотр.Родитель = Работающие;
```

Здесь реквизиту Родитель присваивается ссылка на группу «Работающие».

В результате процедура **ОбработкаПроведения()** будет выглядеть так:

```
Работающие = Справочники.Сотрудники.НайтиПоНаименованию("Работающие");
СпрСотр = Справочники.Сотрудники.СоздатьЭлемент();
```

```
СпрСотр.Родитель = Работающие;
СпрСотр.Наименование = ФИО;
СпрСотр.Должность = Должность;
СпрСотр.Оклад = Оклад;
СпрСотр.ДатаПриема = ДатаПриема;
СпрСотр.Записать();
```

46. Запустите систему в режиме «1С:Предприятие» и убедитесь, что при создании нового документа «Прием на работу» новый сотрудник создается в группе «Работающие».

47. По аналогии внесите изменения в код процедуры **ОбработкаПроведения()** модуля объекта документа «Увольнение», чтобы при увольнении сотрудник перемещался в группу «Уволенные»:

```
Уволенные = Справочники.Сотрудники.НайтиПоНаименованию("Уволенные");
СпрСотр.Родитель = Уволенные;
```

В результате процедура **ОбработкаПроведения()** будет выглядеть так:

```
СпрСотр=Сотрудник.ПолучитьОбъект();
Уволенные = Справочники.Сотрудники.НайтиПоНаименованию("Уволенные");
СпрСотр.Родитель = Уволенные;
СпрСотр.ДатаУвольнения=ДатаУвольнения;
СпрСотр.Записать();
```

48. Если необходимо реализовать проверку – не был ли уже уволен выбранный сотрудник, то процедура **ОбработкаПроведения()** будет выглядеть так:

```
Процедура ОбработкаПроведения(Отказ, РежимПроведения)
СпрСотр = Сотрудник.ПолучитьОбъект();
Уволенные = Справочники.Сотрудники.НайтиПоНаименованию("Уволенные");
Если СпрСотр.Родитель <> Уволенные       Тогда
СпрСотр.Родитель = Уволенные;
СпрСотр.ДатаУвольнения=ДатаУвольнения;
СпрСотр.Записать();
Иначе
Сообщить("Сотрудник "+СпрСотр +" уже уволен!!!");
Отказ = Истина;  //ОТКАЗ - это параметр нашей процедуры
ОбработкаПроведения()
КонецЕсли;

КонецПроцедуры
```

49**.** Запустите систему в режиме «1С:Предприятие» и убедитесь, что увольнение сотрудника выполняется корректно.

## 1.4. ПРОГРАММА ВЫПОЛНЕНИЯ РАБОТЫ

В ходе выполнения работы студенту предлагается проявить творчество в пределах варианта *(под руководством преподавателя)* и представить себя начальником либо владельцем некой фирмы (отдела фирмы).

*Постановка задачи.* Имеется некий документооборот, который следует автоматизировать с помощью системы «1С:Предприятие».

Необходимо определить:

- деятельность фирмы, отдела фирмы;
- хранение какой информации необходимо организовать;
- связи документов (аналогично примеру);
- возможность получения какой информации необходима.

1. На основе указанных выше требований необходимо создать таблицу отношений и организовать хранение информации. Создать в конфигураторе справочники и документы (согласно варианту).
2. В режиме «1С:Предприятие»:
   - создать несколько справочников и заполнить их;
   - создать несколько документов, заполнить их и провести.

*Примечание*: организуйте документооборот таким образом, чтобы пользователю было удобно работать с программой и не приходилось вводить информацию дважды.

Выполнение задания подразумевает продолжение **ЛР №1** (см.Приложение А).

## 1.5. СОДЕРЖАНИЕ ОТЧЕТА

1. Цель работы.
2. Описание предметной области.
3. Описание процесса создания объектов конфигурации.
4. Скриншоты созданных и заполненных справочников и документов (режим «1С:Предприятие»).
5. Выводы.

## 1.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что представляет собой объект конфигурации?
2. Какие группы объектов конфигурации Вы знаете?
3. Где представлены все объекты конфигурации?
4. Какие объекты относятся к объектным типам данных?
5. Какие объекты относятся к необъектным типам данных?
6. Что произойдет при удалении какого-либо объекта из системы?
7. Что произойдет при удалении записи из системы?
8. Перечислите существующие подходы к работе с данными. В чем их отличие?
9. Для чего предназначен объект конфигурации «Справочник»?
10. Каковы характерные особенности справочника?
11. Для чего используются реквизиты и табличные части справочника?
12. Зачем нужны иерархические справочники, и что такое родитель?
13. Зачем нужны подчиненные справочники, и что такое владелец?
14. Какие основные формы существуют у справочника?
15. Что такое предопределенные элементы?
16. Чем с точки зрения конфигурации отличаются обычные элементы справочника от предопределенных элементов?
17. Как создать объект конфигурации «Справочник» и описать его структуру?
18. Как задать синоним стандартного реквизита?
19. Как добавить новые элементы в справочник?
20. Как создать группу справочника?
21. Как переместить элементы из одной группы справочника в другую?
22. Какие стандартные реквизиты имеет справочник?
23. Перечислите объекты встроенного языка для работы с объектом Справочник.
24. Для чего предназначен объект конфигурации «Документ»?
25. Какими характерными особенностями обладает документ?
26. Для чего предназначены реквизиты и табличные части документа?
27. Какие существуют основные формы документа?
28. Что такое проведение документа?
29. Как создать объект конфигурации «Документ» и описать его основную структуру?
30. Как создать новый документ и заполнить его данными?
31. Перечислите объекты встроенного языка для работы с объектом Документ

# ЛАБОРАТОРНАЯ РАБОТА №4. «Исследование табличных типов данных в системе 1С:Предприятие.»

## 2.1. ЦЕЛЬ РАБОТЫ

Целью данной лабораторной работы является исследование табличных типов данных в системе 1С:Предприятие.

## 2.2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Задача любой системы – хранение и оперативное отображение информации пользователю. С помощью полученных данных, как правило, принимаются управленческие решения.

Предположим, что есть 1000 разных документов, и каждый из документов изменяет количество определенного товара на складе. Чтобы получить информацию о текущем количестве на складе, необходимо перебрать все документы: какие-то увеличивают количество товара, какие-то уменьшают, какие-то могут и увеличивать и уменьшать. Такая система очень ресурсоёмкая.

Для упрощения данного процесса, разработчики создали особые объекты конфигурации – регистры.

Сведения из регистров используются для формирования отчетов. Классическая схема использования регистров в 1С:Предприятие выглядит следующим образом:

**Документы → Регистры → Отчеты**

1) документами регистрируются события, приводящие к изменению значений показателей;
2) сами значения показателей хранятся в регистрах;
3) посредством отчетов пользователи получают информацию о состоянии показателей и проводят ее анализ.

### 2.2.1. Табличные данные. Основные сведенья.

К табличным данным в «1С:Предприятие» относятся данные следующих объектов конфигурации:

- регистр сведений,
- регистр накопления,
- регистр бухгалтерии,
- регистр расчета,
- перерасчет,
- последовательность,

– константа.

Табличные данные с точки зрения «1С:Предприятие» представляют собой некоторый набор записей, которые хранятся в таблице. Каждая из этих записей полностью описывается значениями своих полей. Для системы эти записи не обладают какой-либо значимостью, кроме того, что в их полях содержатся некоторые значения. Запись можно удалить, а затем создать новую, с такими же значениями полей. Состояние базы данных при этом не изменится (с точки зрения логики прикладного решения). Это принципиально отличает табличные данные от объектных (см.ЛР№3): **объект нельзя создать дважды, он ценен сам по себе, самим фактом своего существования**.

Второе важное отличие заключается в том, что, изменив значения полей записи, мы получаем другую запись, в то время как изменение значений полей объекта не влечет за собой появление нового объекта.

Важным свойством всех записей является их подчинение регистратору. ***Регистратор*** – это некоторый документ, с которым связаны записи табличных (необъектных) данных. Не может существовать записей, не относящихся к какому-либо регистратору. Исключение составляет лишь ***регистр сведений***, когда для него выбран независимый режим записи (без подчинения регистратору). Поэтому любая запись всегда содержит обязательное поле **Регистратор**.

Записи, относящиеся к одному регистратору, называются **движениями** этого регистратора.

Движения могут создаваться при проведении документов, при записи документов, могут формироваться интерактивно или в виде набора записей регистра (с отбором по регистратору).

Движения не могут существовать отдельно от своих регистраторов. Поэтому при удалении документов или отмене проведения (если свойство Удаление движений установлено в значение, отличное от Не удалять) система производит поиск движений этого документа в таблицах всех регистров, для которых данный документ может быть регистратором, и удаляет их.

Кроме этого, при удалении документа его движения всегда удаляются.

### 2.2.2. Регистр накопления

***Регистр накопления*** – объект конфигурации, предназначенный для хранения итоговых (накопленных) значений показателей и хранения движений показателей. Помимо итоговых значений показателей, регистр может хранить и промежуточные итоги, рассчитанные для указанных периодов.

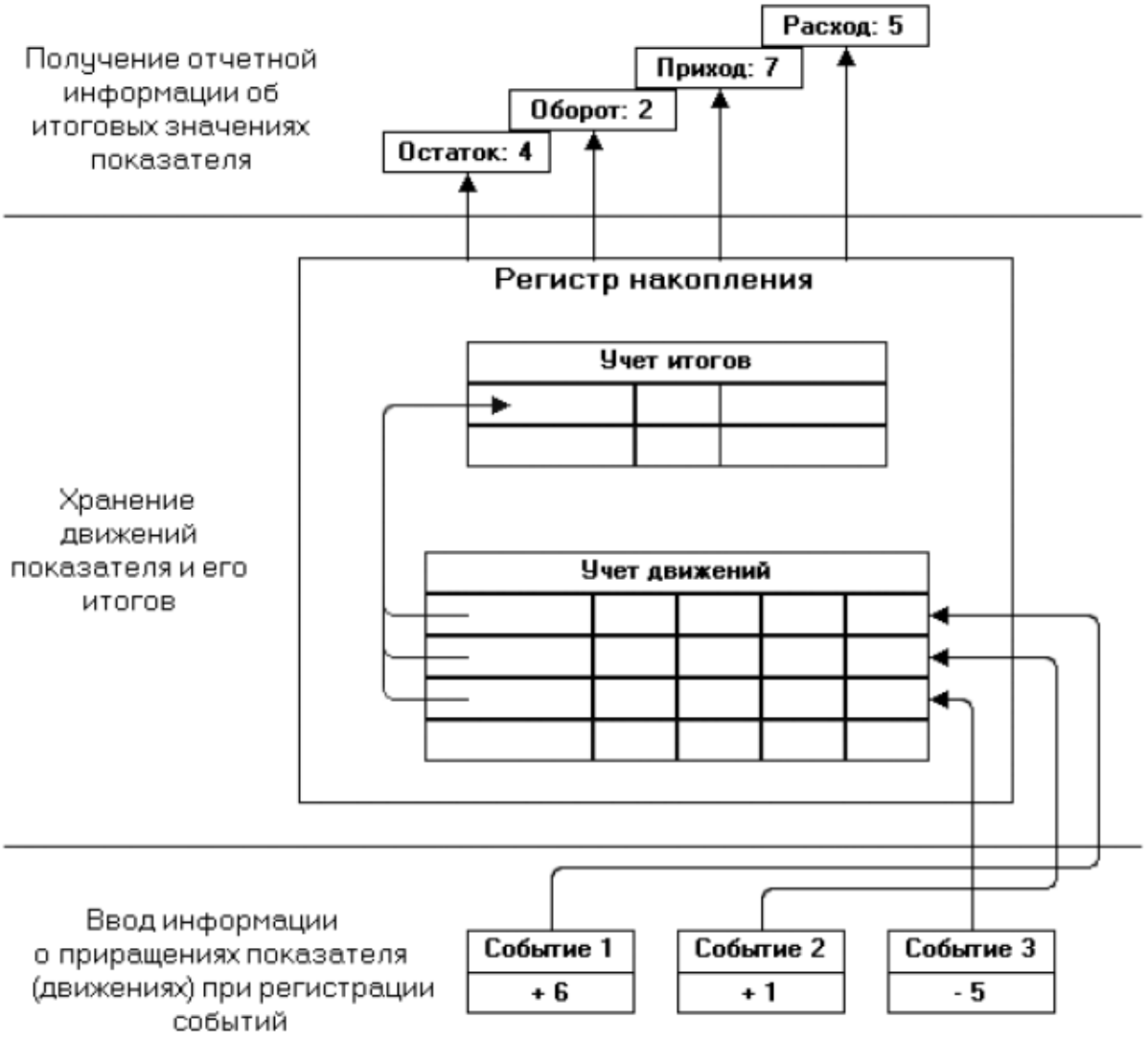



Рисунок 2.1. – Упрощенная схема регистра накопления.

Данные каждого регистра накопления хранятся в **двух таблицах** базы данных:

- таблице движений регистра накопления,
- таблице итогов регистра накопления.

Объект конфигурации *Регистр накопления* предназначен для описания структуры накопления данных. На основе объекта конфигурации Регистр накопления платформа создает в базе данных таблицы, в которых будут накапливаться данные, поставляемые различными объектами базы данных. Эти данные будут храниться в таблицах в виде отдельных записей, каждая из которых имеет одинаковую заданную в конфигураторе структуру.

На основании таблицы движений регистра накопления система рассчитывает таблицу итогов регистра, которая хранит в базе данных итоги на момент времени последнего движения.

Информация о движениях показателей (они могут быть как положительными, так и отрицательными) вносится в регистр накопления посредством наборов записей регистра, подчиненных документу-регистратору.

Этим обеспечивается обоснованность регистрации изменений показателей – регистрацией событий, приводящих к этим изменениям.

Отличительной особенностью регистра накопления является то, что он не предназначен для интерактивного редактирования пользователем. Пользователи смотрят результаты в отчетах.

*Преимущества* регистра накоплений:

- позволяет получить нужные итоги на любой момент времени или документ;
- все необходимые изменения в итогах, связанные с отменой проведения документа, изменением проведенного документа и удалением проведенного документа рассчитываются системой автоматически (если надо – можно прописать вручную);
- обеспечивает контроль уникальности записей, хранящихся в регистре накопления. Благодаря этому в регистре накоплений не может находиться двух записей, относящихся к одной и той же строке одного и того же документа.

**Структура регистра накопления**

В конфигураторе регистры расположены в соответствующих ветках.

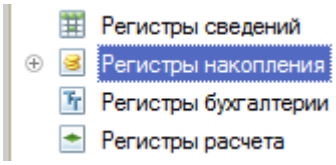



Рисунок 2.2. – Структура регистра накопления.

Закладка Основные:

1) Название регистра 1С в конфигураторе (Имя) и в режиме «1С:Предприятие» (Синоним).

2) Вид регистра 1С:

- для регистра накопления 1С – остатки или обороты;
- для регистра сведений 1С – периодичность (см.п.2.3).

Существует **два** вида регистра накопления – **обороты и остатки**.

*Регистры накопления остатков* позволяют получать итоговые значения показателей остатков и, кроме того (суммируя приращения этих показателей за периоды), позволяют получать обороты.

Итоги *показателей остатков* отражают значения учитываемых показателей на некоторый момент времени.

Изменения показателя *Деньги в кассе* фиксируются посредством документов Приходный кассовый ордер (ПКО) и Расходный кассовый ордер (РКО). Оформление поступления денег в январе повлияет на остатки, как на начало февраля, так и начало марта и т. д.

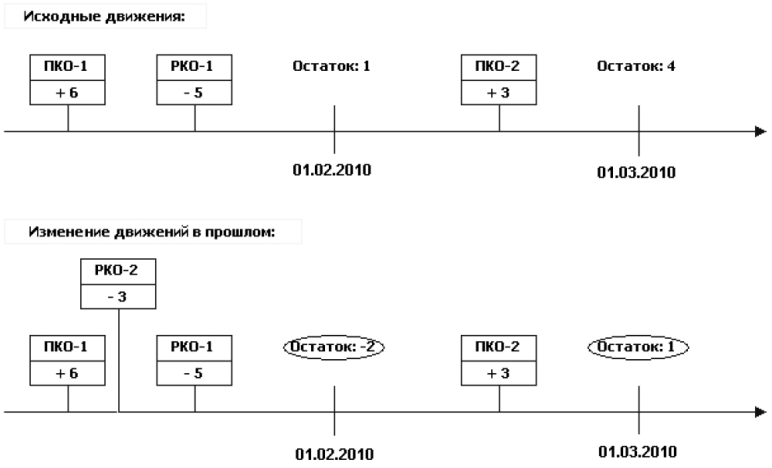



Рисунок 2.3. –Изменения движений в прошлом.

Если же для некоторых сущностей накопление остатков смысла не имеет и требуется накапливать только обороты, тогда следует использовать *оборотные регистры накопления*.

Итоги *оборотных показателей* отражают значения совокупного изменения показателя за временной интервал между двумя моментами времени (период).

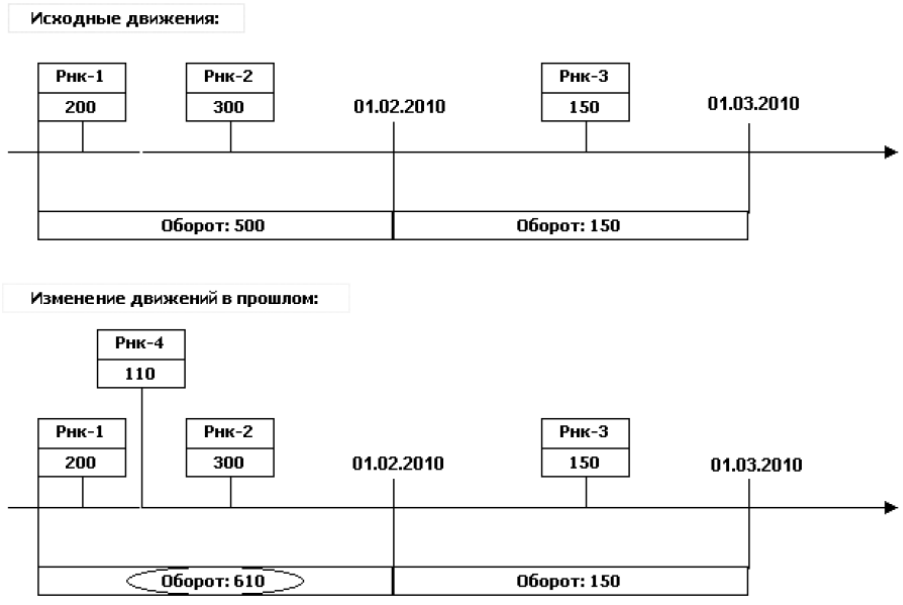



Рисунок 2.4. –Изменения движений в прошлом.

Изменения показателя *Сумма проданного* регистрируются посредством документа Расходная накладная (Рнк). Введенные изменения в объемах продаж января будут касаться только января, но никак не коснутся февральских объемов продаж.

В состав регистра накопления как объекта конфигурации входят ***измерения***, ***ресурсы*** и ***реквизиты***.

*Ресурсы* используются для хранения информации как о движениях, так и о самих значениях показателей.

В регистрах накопления информационные разрезы данных реализуются с помощью *измерений*.

В состав регистра можно включить более одного измерения, поэтому в общем случае регистр накопления можно представить как *n*-мерную систему разрезов учета (измерений), в узлах которой хранятся совокупные данные ресурсов (итоги).

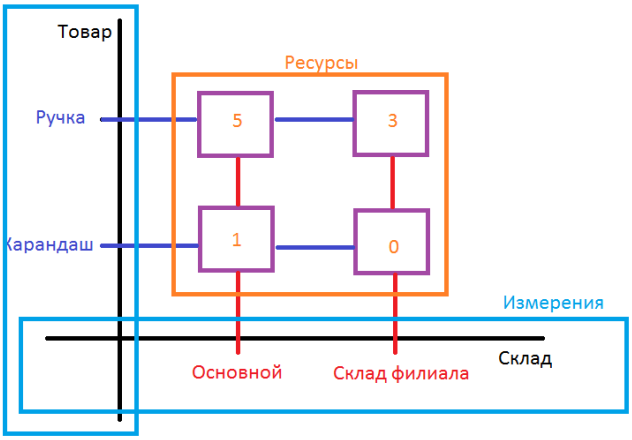



Рисунок 2.5. – Схема взаимодействия.

На представленной схеме – двумерная модель хранимой в регистре информации по итогам.

Измерения – Товар (ручка, карандаш) и Склад (основной, склад филиала). Узлами пересечений этих разрезов учета будут все возможные комбинации значений номенклатурных позиций на складах.

Итог показателя Количество в каждом узле пересечения отражает данные: сколько (в количественном выражении) конкретной номенклатуры находится на конкретном складе.

Данные этой графической двумерной модели можно представить в виде табличной модели (табл. 2.2).

Таблица 2.2. - Табличное представление данных регистров

| Товар | Склад | Количество |
|---|---|---|
| Ручка | Основной | 5 |
| | Склад филиала | 3 |
| Карандаш | Основной | 1 |
| | Склад филиала | 0 |

*Реквизиты* – дополнительные характеристики движений, то есть первичных записей регистра. Реквизиты не влияют на итоги, хранимые в регистре. На вышеприведенной схеме, отражающей итоги, нет места реквизитам. Однако

реквизиты могут быть использованы при решении задач анализа выполненных движений. То есть таких задач, в которых работа идет только с приращениями показателей, безотносительно их итоговых значений.

Например, если при формировании движений регистра ТоварыНаСкладах заполнять соответствующие значения: поступление, реализация в розницу, реализация оптом, списание, перемещение и т. д. для реквизита ВидОперации, то впоследствии можно будет анализировать, какая из операций производится чаще других, какая реже; какие операции не выполняли в течение последнего квартала и т. п.

3) Как уже говорилось ранее, данные каждого регистра накопления (*остатки* или *обороты*) хранятся в **двух таблицах** базы данных:

- таблице движений регистра накопления,
- таблице итогов регистра накопления.

Для регистра накопления **остатков** состав колонок **таблицы движений** следующий:

*Период* – дата записи. Совместно с полями Регистратор и НомерСтроки определяет положение данной записи на временной оси;

*Регистратор* – ссылка на документ, которому подчинена данная запись;

*НомерСтроки* – уникальный номер данной записи в наборе записей регистра, подчиненных документу, указанному в поле Регистратор;

*ВидДвижения* – значение системного перечисления ВидДвиженияНакопления, обозначающее направление приращения указанных в записи ресурсов на итоги по этим ресурсам (Приход или Расход);

*Активность* – тип Булево. Содержит признак влияния записи на итоги регистра;

<*Измерение*> – значение измерения. Количество таких полей равно количеству измерений, определенных в данных регистра как объекта конфигурации;

<*Ресурс*> – значение ресурса. Количество таких полей равно количеству ресурсов, определенных в данных регистра как объекта конфигурации;

<*Реквизит*> – значение реквизита. Количество таких полей равно количеству реквизитов, определенных в данных регистра как объекта конфигурации.

Таблица движений регистра хранит данные о выполненных по регистру движениях.

| Период | Регистратор | Номер строки | Вид движения | Активность | Номенклатура | Склад | Количество | Сумма |
|---|---|---|---|---|---|---|---|---|
| 03.04.2010 15:00:00 | Поступление товаров № 1 13.05.2010 15:00:00 | 1 | Приход | Истина | Пульт VH | Главный | 1 | 10 |
| 13.05.2010 15:00:01 | Поступление товаров № 2 13.05.2010 15:00:01 | 1 | Приход | Истина | Пульт VH | Главный | 9 | 90 |
| 13.05.2010 15:00:01 | Поступление товаров № 2 13.05.2010 15:00:01 | 2 | Приход | Истина | Пульт PW | Главный | 7 | 490 |
| 23.05.2010 15:30:45 | Поступление товаров № 3 23.05.2010 15:30:45 | 1 | Приход | Ложь | Пульт VH | Фили-2 | 5 | 60 |
| 27.05.2010 12:30:45 | Реализация № 1 27.05.2010 12:30:45 | 1 | Расход | Истина | Пульт VH | Главный | 8 | 80 |
| 31.05.2010 23:59:59 | Поступление товаров № 4 31.05.2010 23:59:59 | 1 | Приход | Истина | Пульт PW | Фили-2 | 1 | 17 |
| 01.06.2010 00:00:00 | Поступление товаров № 5 01.06.2010 00:00:00 | 1 | Приход | Истина | Пульт VH | Фили-2 | 1 | 18 |

Рисунок 2.6. – Таблица движений регистра.

Структура **таблицы итогов** регистра накопления **остатков** следующая:

*Период* – дата, на которую актуально состояние хранимого в таблице итога;

*<Измерение>* – значение измерения – разреза учета хранимых итогов. Количество таких полей равно количеству измерений, определенных в данных регистра как объекта конфигурации;

*<Ресурс>* – значение итога ресурса. Количество таких полей равно количеству ресурсов, определенных в данных регистра как объекта конфигурации;

*Разделитель* – поле, позволяющее распараллелить обновление записей итогов. Добавляется в структуру таблицы итогов для регистров накопления, у которых установлено свойство *Разрешить разделение итогов*.

Состав колонок **таблицы движений оборотного** регистра накопления следующий:

*Период* – дата записи. Совместно с полями *Регистратор* и *НомерСтроки* определяет положение данной записи на временной оси;

*Регистратор* – ссылка на документ, которому подчинена данная запись;

*НомерСтроки* – уникальный номер данной записи в наборе записей регистра, подчиненных документу, указанному в поле Регистратор;

*Активность* – тип Булево. Содержит признак влияния записи на итоги регистра;

*<Измерение>* – значение измерения. Количество таких полей равно количеству измерений, определенных в данных регистра как объекта конфигурации;

*<Ресурс>* – значение ресурса. Количество таких полей равно количеству ресурсов, определенных в данных регистра как объекта конфигурации;

*<Реквизит>* – значение реквизита. Количество таких полей равно количеству реквизитов, определенных в данных регистра как объекта конфигурации.

Структура **таблицы итогов оборотного** регистра накопления тоже схожа:

*Период* – период (месяц), за который накоплен оборот итогов ресурсов;

*<Измерение>* – значение измерения – разреза учета хранимых итогов.

Количество таких полей равно количеству измерений, определенных в данных регистра как объекта конфигурации, у которых установлено свойство Использование в итогах;

*<Ресурс>* – значение итога оборота ресурса. Количество таких полей равно количеству ресурсов, определенных в данных регистра как объекта конфигурации;

*Разделитель* – поле, позволяющее распараллелить обновление записей итогов. Добавляется в структуру таблицы итогов для регистров накопления, у которых установлено свойство *Разрешить разделение итогов*.

Пример создания:

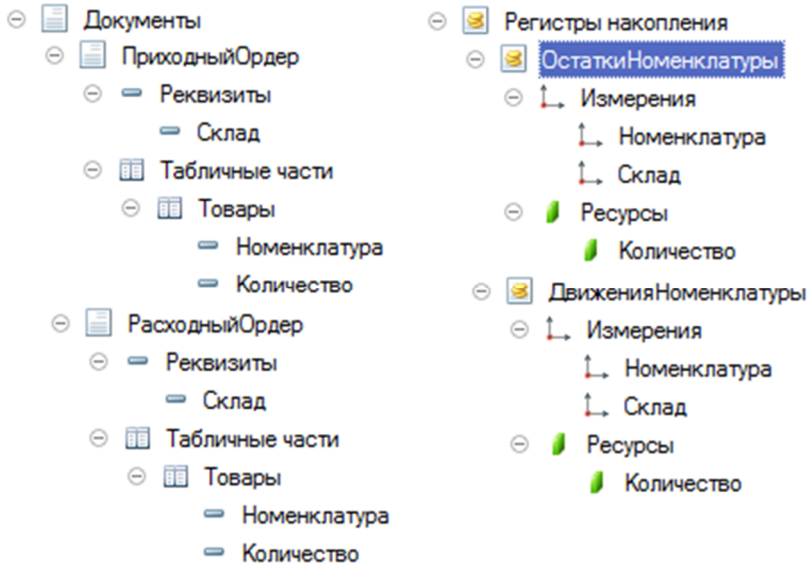



Рисунок 2.7. – Пример создания таблицы итогов оборотного регистра.

У регистра накопления могут быть лишь 3 вида форм (*Формы* – это объекты конфигурации, предназначенные для отображения и редактирования информации, содержащейся в базе данных. Формы могут как принадлежать конкретным объектам конфигурации (справочникам, документам, отчетам и т.д.), так и существовать отдельно от них и использоваться всей конфигурацией в целом):

- **форма списка** - в данной форме нельзя вносить изменения в регистр, но можно делать различные отборы и сортировки;
- **форма набора записей** - в ней можно редактировать записи регистра: добавлять, удалять и изменять их;

## Примеры программного кода для работы с регистром накопления

В программных модулях для общих действий над регистром накопления (получение остатков и оборотов по данным регистра накопления, поиска нужных

записей, формирования выборок, создания форм и макетов) служит объект *РегистрНакопленияМенеджер.<Имя регистра накопления>*.

*РегистрНакопленияЗапись.<имя>*. Используется для доступа к записи регистра накопления. Объект не создается непосредственно, а предоставляется другими объектами, отвечающими за регистр накопления. Например, данный объект представляет записи регистра в наборе записей.

*РегистрНакопленияКлючЗаписи.<имя>*. Представляет собой набор значений, однозначно идентифицирующих запись регистра. Объект используется в тех случаях, когда необходимо сослаться на определенную запись. Например, он выступает в качестве значения свойства ТекущаяСтрока табличного поля, отображающего список записей регистра. Ниже приведены примеры использования объектов встроенного языка для работы с регистрами накопления.

```
// 1. Глобальный контекст
//    РегистрыНакопления

// Пример: выполнить полный пересчет итогов регистра "ОстаткиМатериалов".
РегистрыНакопления.ОстаткиМатериалов.ПересчитатьИтоги();

// 2. объект РегистрыНакопленияМенеджер
//    .<имя регистра накопления>
//    [<имя регистра накопления>]
//    Для Каждого ... Из ... Цикл ... КонецЦикла;

// Пример: рассчитать итоги регистра "ОстаткиМатериалов" на указанную
дату.
ИмяРегистра = ОстаткиМатериалов;
РегистрыНакопления[ИмяРегистра].УстановитьПериодРассчитанныхИтогов(
УказаннаяДата);

// 3. объект РегистрНакопленияМенеджер.<имя>
//    СоздатьКлючЗаписи()

// Пример: активизировать требуемую строку списка регистра накопления.
СтруктураКлючевыхПолей = Новый Структура;
СтруктураКлючевыхПолей.Вставить("Регистратор", Документы.ПриходнаяНа
кладная.НайтиПоНомеру("0000002"));
СтруктураКлючевыхПолей.Вставить("НомерСтроки", 2);
Элементы.Материалы.ТекущаяСтрока = РегистрыНакопления.ОстаткиМатер
иалов.СоздатьКлючЗаписи(СтруктураКлючевыхПолей);

// 4. объект РегистрНакопленияМенеджер.<имя>
//    СоздатьНаборЗаписей()
```

```
// Пример: получить движения документа.
НужныйДокумент = Документы.ПриходнаяНакладная.НайтиПоНомеру(4);
Движения = РегистрыНакопления.ОстаткиМатериалов.СоздатьНаборЗаписей()
;
Движения.Отбор.Регистратор.Значение = НужныйДокумент;
Движения.Прочитать();

// 5. объект РегистрНакопленияМенеджер.<имя>
//   Выбрать()
//   ВыбратьПоРегистратору()

// Пример: выбрать все записи регистра "ОстаткиМатериалов" за текущий
месяц.
Выборка = РегистрыНакопления.ОстаткиМатериалов.Выбрать(НачалоМесяца(
ТекущаяДата()), КонецМесяца(ТекущаяДата()));

// 6. объект РегистрНакопленияНаборЗаписей.<имя>
//   [<индекс элемента коллекции>]
//   Для Каждого ... Из ... Цикл ... КонецЦикла;

// Пример: получить движения документа.
НужныйДокумент = Документы.ПриходнаяНакладная.НайтиПоНомеру(4);
Движения = РегистрыНакопления.ОстаткиМатериалов.СоздатьНаборЗаписей()
;
Движения.Отбор.Регистратор.Значение = НужныйДокумент;
Движения.Прочитать();
Для Каждого ОчередноеДвижение Из Движения Цикл
// Алгоритм обработки движений
 ...
КонецЦикла;
```

### 2.2.3. Регистр сведений

Удобным средством для реализации задач хранения информации необъектных данных, развернутой в некоторых разрезах, является использование регистров сведений.

Объект конфигурации **Регистр сведений** предназначен для описания структуры хранения данных в разрезе нескольких измерений. На основе объекта конфигурации Регистр сведений платформа создает в базе данных таблицу, в которой может храниться произвольная информация, «привязанная» к набору измерений.

Принципиальное **отличие** регистра сведений от регистра накопления заключается в том, что каждое движение регистра сведений устанавливает новое

значение ресурса, в то время как движение регистра накопления изменяет существующее значение ресурса. По этой причине регистр сведений может хранить любые данные.

*Регистры сведений* бывают различных видов. В первом приближении они делятся на **периодические** и **непериодические** регистры. Непериодические регистры проще всего сравнить с некими таблицами, в которых хранятся определенные сведения, развернутые *по* заданным в регистре измерениям. Примерами таких регистров могут быть *регистр*, фиксирующий содержание драгоценных металлов в каких-либо деталях, *регистр*, хранящий данные адресного *классификатора*, *регистр* с информацией о том, какой день недели является рабочим, а какой – выходным, и многие другие. Даже если одним из измерений непериодического регистра сведений является дата – он не становится от этого периодическим.

В противоположность непериодическому регистру, который, в сущности, хранит одни и те же показатели, периодический *регистр* может хранить историю изменения этих показателей. Такие регистры могут иметь различную *периодичность*. Например, для регистра, призванного фиксировать курсы валют, разумно будет установить *периодичность* в пределах дня, так как курсы валют обычно фиксируются в пределах одного дня.

Другая классификация регистров сведений заключается в том, является ли *регистр* **независимым** или **подчиненным** регистратору. Независимый *регистр* сведений можно заполнять самостоятельно, вручную вводя в него данные непосредственно в таблице регистра. Когда же *регистр* подчинен регистратору, записи в него производятся при *проведении документа*-регистратора.

***Периодический регистр сведений.***

Периодичность регистра сведений можно определить одним из следующих значений:

- в пределах секунды,
- в пределах дня,
- в пределах месяца,
- в пределах квартала,
- в пределах года;

В Регистре сведений можно хранить не просто статические данные, но и историю изменений этих данных. Например, периодический регистр сведений Цены сможет не только хранить информацию о том, какова цена на определенную номенклатуру сейчас (или была в прошлом), но и о том, как она будет изменяться в будущем (если эти изменения будут заранее планироваться).




Рисунок 2.8. – Периодический регистр сведений по записям.

Периодический регистр сведений всегда содержит служебное поле **Период**, добавляемое системой автоматически. Оно имеет тип **Дата** и служит для указания факта принадлежности записи к какому-либо периоду. При записи данных в регистр платформа всегда приводит значение этого поля к началу того периода, в который он попадает.

Например, если в регистр сведений с периодичностью в пределах месяца записать данные, в которых период указан как 08.04.2016, то регистр сохранит эти данные со значением периода, равным 01.04.2016.

Как и для других регистров, система контролирует уникальность записей для регистра сведений. Однако если для прочих регистров уникальным идентификатором записи является *регистратор* и *номер строки*, то для регистра сведений применяется другой принцип формирования ключевого значения.

Ключом записи, однозначно идентифицирующим запись, является в данном случае совокупность значений *измерений* регистра и *периода* (в случае если регистр сведений периодический). Регистр сведений не может содержать несколько записей с одинаковыми ключами. Если запись не уникальна – система 1С выдаст сообщение «Запись с такими ключевыми полями существует!» и не даст произвести запись в базу данных.

Например, для периодического регистра сведений с измерением Товар и ресурсом Цена ключом записи будет набор значений полей Период и Товар. Регистр сведений не может содержать несколько записей с одинаковыми ключами.

Еще один пример использования регистра сведений – хранение информации о курсе валют в разрезе валюты и периода. Данный пример будет подробно описан в п.3.

Одна из главных особенностей периодического регистра сведений – возможность получать готовые значения «Среза первых» и «Среза последних». Эта информация позволяет очень быстро получить из базы данных информацию о последнем (первом) установленном значении на определенную дату.

***Подчиненный регистр сведений***

Использование режима записи **Подчинение регистратору** может потребоваться в случае, когда логика работы прикладного решения требует того,

чтобы изменения, выполняемые в регистре сведений, были жестко связаны с документами, фиксирующими факты хозяйственной деятельности.

Например, изменение отпускных цен компании может производиться только определенным кругом лиц, и каждое такое изменение должно сопровождаться оформлением соответствующего документа. То есть данные регистра сведений должны четко «помнить», в связи с каким именно документом они там оказались. Для этого достаточно установить значение Подчинение регистратору свойству Режим записи регистра сведений как объекта конфигурации.

С точки зрения хранения данных в регистре установка значения *Подчинение регистратору* свойства Режим записи приводит к включению в состав регистра полей Регистратор, НомерСтроки и Активность.

Далее требуется указать, какие именно документы могут быть регистраторами записей регистра. Это можно сделать как при редактировании самого регистра, как объекта конфигурации, так и при редактировании документа-регистратора как объекта конфигурации. Благодаря этому облегчается труд разработчика, которому необходимо сформировать или модифицировать набор (наборы) записей регистра (регистров), подчиненных некоему документу.

Поле Регистратор обеспечивает привязку к документу, но не всегда входит в разряд ключевых полей. Для того чтобы поле Регистратор стало ключевым, необходимо для такого регистра сведений (с режимом записи Подчинение регистратору) установить для свойства Периодичность значение *По позиции регистратора*.

Если продолжать сравнение с регистром накопления, то можно сказать, что регистр сведений предоставляет больше свободы в редактировании хранимых данных. Наряду с возможностью использования в режиме подчинения регистратору (когда записи регистра сведений «привязаны» к документу-регистратору) регистр сведений может применяться и в независимом режиме, в котором пользователю предоставляется полная свобода интерактивной работы с данными регистра. Регистр сведений, не использующий подчинение регистратору, называют *независимым регистром сведений*.

### Структура регистра сведений

В состав регистра сведений как объекта конфигурации включаются:

- измерения,
- ресурсы,
- реквизиты.

**Ресурсы** хранят данные регистра, то есть ту информацию, ради хранения которой регистр, собственно, создавался и с которой работает функциональность регистра.

**Измерения** используются для обеспечения разрезов хранения этой информации.

Для ситуаций, когда необходимо кроме данных хранить дополнительную информацию собственно о записях регистра, возможно использование **реквизитов**.

Как было рассмотрено выше, регистры сведений могут иметь режим записи *Подчинение регистратору*, вследствие чего в состав таблиц регистра добавляется поле Регистратор. И регистр сведений может быть *периодическим*, вследствие чего в состав его таблиц добавляется поле Период.

Данные каждого регистра сведений хранятся в отдельной таблице базы данных.

Каждая такая таблица имеет следующий состав колонок:

*Период* – дата записи. Определяет положение данной записи на временной оси. Это поле существует только для периодических регистров;

*Регистратор* – содержит ссылку на документ, которому подчинена данная запись. Это поле существует только для регистров с режимом записи Подчинение регистратору;

*НомерСтроки* – уникальный номер данной записи в наборе записей регистра, подчиненных документу, указанному в поле Регистратор. Это поле существует только для регистров с режимом записи Подчинение регистратору;

*Активность* – имеет тип Булево. Содержит признак влияния записи на получение информации из регистра. Записи, для которых значение данного свойства установлено в Ложь, не будут учитываться при получении «первых» или «последних» записей регистра, а также при получении сведений на определенный момент времени. Это поле существует только для регистров с режимом записи Подчинение регистратору;

*Ключ* – короткий ключ записи регистра. Поле присутствует у непериодических регистров, имеющих хотя бы одно измерение;

<*Измерение*> – содержит значение измерения. Количество таких полей равно количеству измерений, определенных для регистра как объекта конфигурации;

<*Ресурс*> – значение ресурса. Количество таких полей равно количеству ресурсов, определенных для регистра как объекта конфигурации;

<*Реквизи*т> – значение реквизита. Количество таких полей равно количеству реквизитов, определенных для регистра как объекта конфигурации.

Для регистра ЦеныНоменклатуры с режимом записи *Подчинение регистратору* и периодичностью *По позиции регистратора* пример заполнения таблицы базы данных будет следующим (темным фоном выделены ключевые поля записей):

| Период | Регистратор | Номер строки | Активность | Номенклатура | ТипЦены | Цена | Ответственный |
|---|---|---|---|---|---|---|---|
| 16.04.2010 21:54:41 | Изменение цен компании 000000001 от 16.04.2010 21:54:41 | 1 | истина | Пульт РХ | Оптовая | 150,00 | Семенов |
| 16.04.2010 21:54:41 | Изменение цен компании 000000001 от 16.04.2010 21:54:41 | 2 | истина | Пульт РХ | Мелкооптовая | 170,00 | Семенов |
| 16.04.2010 21:54:41 | Изменение цен компании 000000001 от 16.04.2010 21:54:41 | 3 | истина | Пульт РХ | Розничная | 200,00 | Семенов |
| 16.04.2010 21:54:41 | Изменение цен компании000000001 от 16.04.2010 21:54:41 | 4 | истина | Пульт VH | Оптовая | 100,00 | Семенов |
| 16.04.2010 21:54:41 | Изменение цен компании 000000001 от 16.04.2010 21:54:41 | 5 | истина | Пульт VH | Мелкооптовая | 90,00 | Семенов |
| 18.04.2010 20:49:26 | Изменение цен компании 000000002 от 18.04.2010 20:49:26 | 1 | истина | Пульт РХ | Оптовая | 170,00 | Иванов |
| 18.04.2010 20:49:26 | Изменение цен компании 000000002 от 18.04.2010 20:49:26 | 2 | истина | Пульт РХ | Мелкооптовая | 190,00 | Иванов |
| 18.04.2010 20:49:26 | Изменение цен компании 000000002 от 18.04.2010 20:49:26 | 3 | истина | Пульт РХ | Розничная | 220,00 | Иванов |

Рисунок 2.9. – Обобщение данных.

Рассмотрим **непериодический** регистр сведений Сотрудники с ***независимым*** режимом записи. Таблица регистра Сотрудники, хранящая информацию в базе данных, может быть заполнена следующим образом:

| Физическое лицо | Подразделение | Должность |
|---|---|---|
| Иванов Иван Сергеевич | Отдел продаж | Руководитель |
| Семенов Сергей Петрович | Отдел продаж | Менеджер |
| Семенов Сергей Петрович | Отдел закупок | Руководитель |
| Петровский Семен Иванович | Руководство компании | Руководитель |
| Петровская Ирина Ивановна | Бухгалтерия | Бухгалтер |

Рисунок 2.10. – Непериодический регистр сведений.

Регистры, записываемые **независимо**, могут свободно редактироваться вручную или средствами встроенного языка. При этом если измерение такого регистра назначено как «ведущее» и значением измерения является ссылка на объект базы данных, то будет считаться, что запись регистра имеет смысл, только пока существует этот объект. Например, если назначить ведущим измерение «Конкурент», то считается, что запись имеет смысл только как информация по данному конкуренту. Соответственно, при удалении конкурента записи по нему будут удалены автоматически.

Если регистр записывается **регистратором**, то это значит, что записи будут жестко подчинены регистраторам – документам. Обычно это значит, что записи будут порождаться при проведении документов. Соответственно, при удалении

документа записи будут удаляться автоматически. В отличие от ведущих измерений, регистратор может быть только один.

**Формы**

Для того чтобы пользователь мог просматривать и изменять данные, содержащиеся в регистре сведений, система поддерживает несколько форм представления регистра. Система может автоматически генерировать все нужные формы регистра. Наряду с этим разработчик имеет возможность создать собственные формы, которые система будет использовать вместо форм по умолчанию:

Форма списка

Для просмотра данных, содержащихся в регистре сведений, используется форма списка. Она позволяет выполнять навигацию по регистру, добавлять, помечать на удаление и удалять записи регистра. Форма списка позволяет выполнять сортировку и отбор отображаемой информации по нескольким критериям.

Форма записи

Для просмотра и изменения отдельных записей регистра сведений используется форма записи. Как правило, она представляет данные в удобном для восприятия и редактирования виде.

## Примеры программного кода для работы с регистром сведений

В программных модулях для общих действий над регистром сведений (поиск, выбор и создание записей регистра) служит объект *РегистрСведенийМенеджер.<Имя регистра сведений>*.

*РегистрСведенийЗапись.<имя>*. Предоставляет доступ к записи регистра сведений. Объект не создается непосредственно, а предоставляется другими объектами, связанными с регистром сведений. Например, данный объект предоставляет записи регистра в наборе записей.

*РегистрСведенийКлючЗаписи.<имя>*. Представляет собой набор значений, однозначно идентифицирующих запись регистра. Объект используется в тех случаях, когда необходимо сослаться на определенную запись. Например, он выступает в качестве значения свойства ТекущаяСтрока табличного поля, отображающего список записей регистра.

Ниже приведены примеры использования объектов встроенного языка для работы с регистрами сведений.

```
// 1. Глобальный контекст
//    РегистрыСведений

// Пример: получить текущую цену из периодического регистра сведений "Цены".
Элемент = Справочники.Номенклатура.НайтиПоКоду(4);
```

```
Отбор = Новый Структура("Номенклатура", Элемент);
ЗначенияРесурсов = РегистрыСведений.Цены.ПолучитьПоследнее(ТекущаяДата(
), Отбор);
Цена = ЗначенияРесурсов.Цена;

// 2. объект РегистрыСведенийМенеджер
//    .<имя регистра сведений>
//    [<имя регистра сведений>]
//    Для Каждого ... Из ... Цикл ... КонецЦикла;

// Пример: получить начальную цену из периодического регистра сведений
"Цены".
ИмяРегистра = "Цены";
Услуга = Справочники.Номенклатура.НайтиПоНаименованию("Диагностика");
Отбор = Новый Структура;
Отбор.Вставить("Номенклатура", Услуга);
Цена = РегистрыСведений[ИмяРегистра].ПолучитьПервое(ТекущаяДата(), Отб
ор).Цена;

// 3. объект РегистрСведенийМенеджер.<имя>
//    СоздатьКлючЗаписи()

// Пример: активизировать требуемую строку списка регистра сведений.
СтруктураКлючевыхПолей = Новый Структура;
СтруктураКлючевыхПолей.Вставить("Период", Дата("20040331000000"));
СтруктураКлючевыхПолей.Вставить("Номенклатура", Справочники.Номенклат
ура.НайтиПоКоду("0000006"));
Элементы.Материалы.ТекущаяСтрока = РегистрыСведений.Цены.СоздатьКлюч
Записи(СтруктураКлючевыхПолей);

// 4. объект РегистрСведенийМенеджер.<имя>
//    СоздатьНаборЗаписей()

// Пример: показать номенклатуру, цена на которую была установлена в
заданную дату и время.
Набор = РегистрыСведений.Цены.СоздатьНаборЗаписей();
Набор.Отбор.Период.Установить(ЗаданнаяДата, Истина);
Набор.Прочитать();
Для Каждого ОчереднаяЗапись Из Набор Цикл
    Сообщить("Номенклатура = "+ОчереднаяЗапись.Номенклатура + ", цена =
" + ОчереднаяЗапись.Цена);
КонецЦикла;


// 5. объект РегистрСведенийМенеджер.<имя>
```

```
//    СоздатьМенеджерЗаписи()


// Пример: добавить новое значение в регистр "Цены".
Запись = РегистрСведений.Цены.СоздатьМенеджерЗаписи();
Запись.Период = ТекущаяДата();
Запись.Номенклатура = Справочники.Номенклатура.НайтиПоКоду("0000005");
Запись.Цена = 568;
Запись.Записать();


// 6. объект РегистрСведенийНаборЗаписей.<имя>
//    [<индекс элемента коллекции>]4
//    Для  Каждого ... Из ... Цикл ... КонецЦикла;


// Пример: показать номенклатуру, цена на которую была установлена в
заданную дату и время.
Набор = РегистрыСведений.Цены.СоздатьНаборЗаписей();
Набор.Отбор.Период.Установить(ЗаданнаяДата, Истина);
Набор.Прочитать();
Для Каждого ОчереднаяЗапись Из Набор Цикл
    Сообщить("Номенклатура = "+ОчереднаяЗапись.Номенклатура + ", цена =
" + ОчереднаяЗапись.Цена);
КонецЦикла;


// 7. объект РегистрСведенийВыборка.<имя>
//    Выбрать()
//    ВыбратьПоРегистратору()


// Пример: удалить все записи регистра сведений за текущий месяц.
Выборка = РегистрыСведений.Цены.Выбрать(НачалоМесяца(ТекущаяДата()), К
онецМесяца(ТекущаяДата()));
Пока Выборка.Следующий() Цикл
    Выборка.ПолучитьМенеджерЗаписи().Удалить();
КонецЦикла;

// 8. объект РегистрСведенийМенеджер.<имя>
//    Выбрать()
//    ВыбратьПоРегистратору()


// Пример: показать изменение цен на элемент в течение года.
```

```
Отбор =
Новый Структура("Номенклатура", Справочники.Номенклатура.НайтиПоКоду(
"0000005"));
Выборка = РегистрыСведений.Цены.Выбрать(НачалоГода(ТекущаяДата()), Теку
щаяДата(), Отбор);
Пока Выборка.Следующий() Цикл
   Сообщить("Дата = " + Выборка.Период + ", цена = " + Выборка.Цена);
КонецЦикла;
```

## 2.3. ПРИМЕРЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

### 2.3.1. Регистры сведений

**Пример 1**. Рассмотрим ситуацию, когда в организации имеется несколько филиалов, каждый из которых включает ряд подразделений. В каждом филиале определено ответственное лицо.

Таблица 2.3. - Табличное представление данных.

| *Филиал* | *Подразделение* | *Ответственное лицо* |
|---|---|---|
| Волжский | 1 | Иванов И.П. |
| Волжский | 2 | Петров Н.Е. |
| Ростовский | 1 | Сидоров О.К. |
| Ростовский | 2 | Попов П.А. |

Для хранения информации о филиалах сформируем **справочник** – *Филиалы*.

Таблица 2.4. - Справочник Филиалы.

| *Наименование филиала* |
|---|
| Волжский |
| Ростовский |
| Центральный |

После этого внесем справочники Сотрудники и Номенклатура.

Перейдите в режим 1С:Предприятие и внесите информацию в созданные справочники.

Далее необходимо создать **непериодический** регистр сведений Ответственные лица.

На вкладке данные укажите измерения и ресурсы регистратора.

Первое измерение регистра сведений – Филиал, для которого определим тип данных – СправочникСсылка.Филиалы. Для второго измерения (Подразделение)

выберем числовой тип данных, а имя ресурса регистра определим – Ответственный (тип данных – СправочникСсылка.Сотрудники). В результате вкладка Данные окна редактирования рассматриваемого объекта конфигурации будет выглядеть так, как показано на рис.2.11

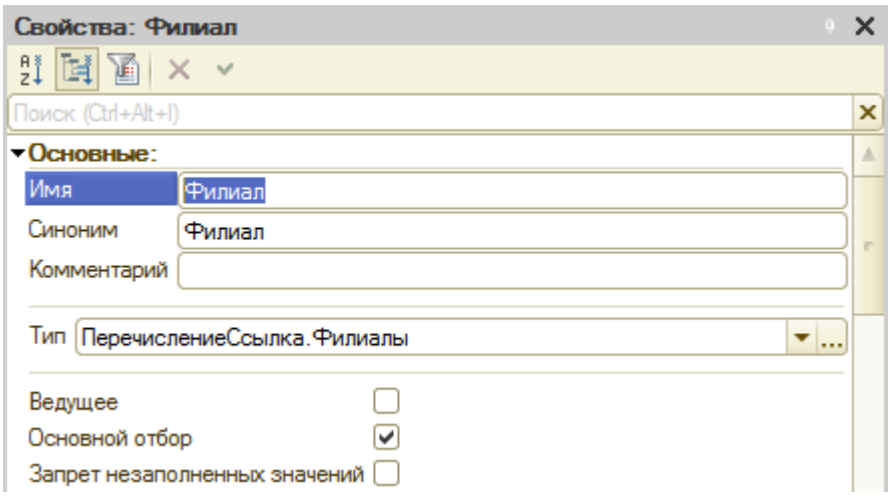



Рисунок 2.11. – Свойство Филиал.

Перейдите в режим 1С:Предприятие. Заполните регистр сведений.

Кроме регистра сведений в конфигурации, потребуется документ «Поступление». На рис. показана вкладка Данные окна редактирования рассматриваемого объекта конфигурации.

Филиал (тип данных – СправочникСсылка.Филиалы);

Подразделение (тип данных – Число);

Ответственный (тип данных – СправочникСсылка.Сотрудники).

В табличной части Перечень данного документа:

- Название (тип данных – СправочникСсылка.Номенклатура);
- Цена (тип данных – Число);
- Количество (тип данных – Число);
- Сумма (тип данных – Число).

Далее необходимо создать пользовательскую форму документа. Для этого следует перейти на вкладку Формы и воспользоваться конструктором форм.

Следующим пунктом будет настройка автоматического заполнения поля **Ответственные лица** (после выбора названия филиала и номера подразделения) в документе Поступление.

После двойного клика по полю Подразделение откроется палитра свойств. Обратимся к разделу События.

Здесь необходимо соответствующим образом оформить обработку события ПриИзменении. Для этого следует щелкнуть по изображению лупы напротив надписи данного события. В программном коде должен производиться запрос к регистру сведений ОтветственныеЛица для извлечения необходимой

информации. Эти действия могут быть выполнены только на сервере, поэтому программный код будет состоять из двух компонентов:

– процедуры ПодразделениеПриИзменении, которая выполняется в контексте клиента;

– функции ПолучитьОтветственного, которая выполняется в контексте сервера и вызывается из клиентской процедуры ПодразделениеПриИзменении.

Документ Поступление: ФормаДокумента

```
&НаКлиенте
Процедура ПодразделениеПриИзменении(Элемент)
    Объект.Ответственный=ПолучитьОтветственного();
  КонецПроцедуры

&НаСервере
Функция ПолучитьОтветственного()
  Запрос = Новый Запрос;
  Запрос.Текст = "ВЫБРАТЬ *
  | ИЗ  РегистрСведений.ОтветственныеЛица
  | ГДЕ Филиал=&Филиал И
  | Подразделение =&Подразделение";
  Запрос.УстановитьПараметр("Филиал",
          Объект.Филиал);
  Запрос.УстановитьПараметр("Подразделение",
         Объект.Подразделение);
  Результат = Запрос.Выполнить().Выбрать();
  Если Результат.Следующий() > 0 Тогда
     Возврат Результат.Ответственный;
  КонецЕсли;
КонецФункции
```

Рисунок 2.12. – Документ “Поступление – Форма Документа”.

Текст процедуры ПодразделениеПриИзменении приведен в листинге выше. В ней идет обращение только к серверной функции ПолучитьОтветственного. При этом в данной функции используются значения двух реквизитов документа:

– Филиал;
– Подразделение.

Текст функции ПолучитьОтветственного приведен в листинге выше. Там производится запрос к регистру сведений ОтветственныеЛица:

```
Запрос.Текст = "ВЫБРАТЬ *
| ИЗ РегистрСведений.ОтветственныеЛица
| ГДЕ Филиал=&Филиал И
| Подразделение =&Подразделение";
```

**Запросы** представляют собой очень важные объекты, им отведена **ЛР №.6** В ней синтаксические конструкции запросов будут рассмотрены подробнее.

Параметрами запроса являются Филиал и Подразделение. При определении параметров в тексте запроса используется специальный символ &. Для установки значений параметров запроса во внутреннем языке 1С имеется метод УстановитьПараметр. В результате выполнения рассматриваемого запроса возможны две ситуации:

- имеется единственная запись об интересующем человеке, который является ответственным в определенном отделе конкретного филиала;
- записи с указанными измерениями нет.

В первом случае возвращается извлеченная информация об ответственном в исходную процедуру ПодразделениеПриИзменении. Далее фамилия ответственного лица подставляется в поле Ответственный. Во втором же случае никакого заполнения поля Ответственный не производится.

Перейдите в режим 1С:Предприятие, убедитесь в работоспособности программного кода.

После заполнения филиала и подразделения, при нажатии на Enter, автоматически произойдет заполнения поля Ответственный.

**Пример 2.** Рассмотрим работу периодических данных на примере хранения курса валют. Курс валют изменяется каждый день, бывает необходимо получить текущий курс, либо курс за какой-то день в прошлом. Создадим необходимую структуру.

Для начала создадим новый справочник **Валюты**, т.к. для валюты необходимо знать международное название, национальное название, аббревиатуру, код валюты. Включим новый справочник в подсистему.

Создадим новый *регистр* сведений **КурсыВалют**, установим его *периодичность* – В пределах дня.

Периодичность – указывает какой минимальный интервал может быть между разными записями. При этом уникальной записью считается комбинация из измерений и времени записи (если таковое используется).

*Режим записи* – независимый, т.е. специального документа для изменения курса валют не требуется, т.к. курсы валют обычно загружаются из интернета. Для данного примера они будут внесены вручную.

На закладке **Данные** добавим следующее:

**Измерения**:

**Имя**: Валюта, тип СправочникСсылка.Валюты.

Свойство *Ведущее* имеет смысл использовать лишь тогда, когда измерение имеет тип ***ссылки на объект базы данных***. Установка свойства Ведущее будет говорить о том, что запись регистра сведений представляет интерес, пока существует тот объект, ссылка на который выбрана в качестве значения этого измерения в этой записи. При удалении объекта все записи регистра сведений по этому объекту тоже будут автоматически удалены.

Также в результате того, что это измерение регистра мы сделали ведущим, в форме элемента справочника Валюты, в панели навигации в группе Перейти появится ссылка. По ней возможен переход к записям этого регистра, которые содержат в измерении Валюта ссылку на этот элемент справочника.

Запустите приложение в режиме «1С:Предприятие».

Заполните регистр сведений. Справочник Валюты можно заполнять параллельно.

Курс валют привязан к определенной дате. Если создать ещё одно значение курса за текущую дату, например, для EUR, то система выдаст ошибку.

Это означает, что запись с таким значением периода и измерением уже есть.

### 2.3.2. Регистры накоплений

**Пример 1.** Регистр накопления остатков.

Рассмотрим пример конфигурации, которая позволяет контролировать остатки товаров на складе некоторой организации. Для этого будет создан регистр накопления остатков «ИнформацияОТоварах». Добавление ресурса регистра накопления будет обеспечивать документ «Поступление», уменьшение ресурса регистра – документ «Отгрузка».

Создайте две подсистемы:

Создайте справочники «Фирмы» и «Номенклатура», которые не будут иметь дополнительных реквизитов

Справочник Фирмы необходимо отнести только к подсистеме ЗаведующийФилиалом, а Номенклатура - к двум подсистемам.

Далее создайте документ «Поступление»

области шапки документа разместите один дополнительный реквизит – Поставщик (тип данных – СправочникСсылка.Фирмы).

Для табличной части документа выберите имя Перечень и внесите в нее два реквизита:

- Товар (тип данных – СправочникСсылка.Номенклатура);
- Количество (тип данных – Число) для внесения количества поступивших товаров.

В режиме «1С:Предприятие»:

Создайте регистр накопления остатков **ИнформацияОТоварах.**

На вкладке Основные установите имя объекта, а также вид регистра накопления – Остатки.

Вкладка Данные:

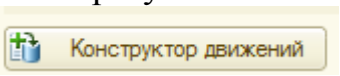
Измерения: Организации (тип данных – СправочникСсылка.Фирмы);

Измерения: Товар (тип данных – СправочникСсылка.Номенклатура);

Ресурсы: Количество (в качестве типа данных выберем Число).

Далее необходимо открыть окно редактирования созданного ранее документа «Поступление», перейти на вкладку **Движения** и определить, в каких регистрах будут происходить изменения при проведении документа «Поступление».

*Движения документа* – это записи в регистрах, которые создаются в процессе проведения документа и отражают изменения, производимые документом.

Обратите внимание, что сразу после отметки выбранного регистра становится доступной кнопка 

.

Конструктор устроен просто. В списке Регистры перечислены регистры, в которых документ может создавать движения.

В списке Реквизиты документа должны находиться исходные данные для создания движений – реквизиты документа.

А в таблице Поле – Выражение должны быть заданы формулы, по которым будут вычисляться значения измерений и ресурсов регистра при записи движений.

Обратите внимание, что по умолчанию конструктор предлагает создавать движения прихода (Тип движения регистра – Приход, символ + рядом с названием регистра) по регистру.

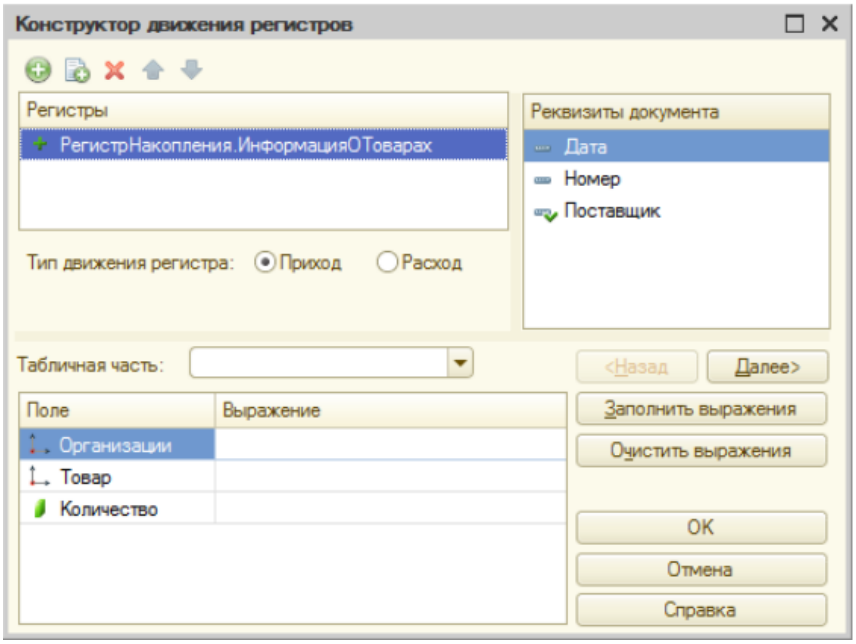



Рисунок 2.13. – Конструктор движения регистров.

В правое верхнее поле уже автоматически внесены реквизиты шапки документа. Это связано с тем, что в указанное поле включаются параметры документа, которые могут участвовать в движениях.

Но кроме шапки, в документе «Поступление» имеется еще и табличная часть, и для добавления реквизитов табличной части в поле Реквизиты документа следует выполнить несложную операцию. В центральной части присутствует поле со списком (с соответствующей подписью – Табличная часть), где перечислены имеющиеся табличные части в документе (в созданном документе табличная часть одна – Перечень). Для перенесения в поле Реквизиты документа реквизитов табличной части необходимо просто выбрать ее имя в списке табличных частей.

Далее перейдите к нижней левой части окна конструктора движений. Здесь необходимо указать соответствие полей регистра накопления и реквизитов документа. Технически это можно сделать с помощью двойного щелчка мышью по необходимому реквизиту документа в правом верхнем разделе. Также можно воспользоваться кнопкой **Заполнить выражения**. В результате напротив названий полей регистра появятся соответствующие выражения.

Теперь все действия в окне конструктора движений регистра выполнены и осталось щелкнуть по кнопке ОК. Это приведет к генерации конструктором программной процедуры, которая формирует движения по регистру при проведении документа «Поступление». В дальнейшем если потребуется эту процедуру найти (скажем, для коррекции или просто для просмотра), то в **окне редактирования** объекта конфигурации следует перейти на вкладку **Прочее**.

Документ Поступление: Модуль объекта

```
Процедура ОбработкаПроведения(Отказ, Режим)
    //{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
    // Данный фрагмент построен конструктором.
    // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

    // регистр ИнформацияОТоварах Приход
    Движения.ИнформацияОТоварах.Записывать = Истина;
    Для Каждого ТекСтрокаПеречень Из Перечень Цикл
        Движение = Движения.ИнформацияОТоварах.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
        Движение.Период = Дата;
        Движение.Организация = Поставщик;
        Движение.Товар = ТекСтрокаПеречень.Товар;
        Движение.Количество = ТекСтрокаПеречень.Количество;
    КонецЦикла;

    //}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

Рисунок 2.14. – Документ “Поступление”: Модули объекта.

Основное содержание процедуры ОбработкаПроведения составляет цикл, в котором перебираются строки табличной части Перечень документа ИнформацияОТоварах: Для Каждого ТекСтрокаПеречень Из Перечень Цикл При каждом проходе цикла с помощью метода Добавить создается очередная запись в регистре накопления. В последующих строках процедуры устанавливаются значения полей регистра в соответствии с данными из очередной строки табличной части документа.

Перейдите в режим 1С:Предприятие и проверьте результат выполненных изменений.

Чтобы просмотреть содержание регистра накопления следует обратиться к разделу **Все функции** главного меню приложения. В результате на экране откроется окно, в котором требуется обратиться к регистру накопления Информация о товарах.

Создайте документ «Отгрузка», который будет уменьшать ресурс регистра ИнформацияОТоварах.

Данный документ будет похож по своей структуре на уже разработанный документ «Поступление».

После формирования структуры документа «Отгрузка» перейдите на вкладку **Движения** и с помощью конструктора сформируйте соответствие реквизитов документа и полей регистра. Важно отметить, что в данном случае в качестве типа движения регистра указан Расход.

Документ Отгрузка: Модуль объекта

```
Процедура ОбработкаПроведения(Отказ, Режим)
    //{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
    // Данный фрагмент построен конструктором.
    // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

    // регистр ИнформацияОТоварах Расход
    Движения.ИнформацияОТоварах.Записывать = Истина;
    Для Каждого ТекСтрокаПеречень Из Перечень Цикл
        Движение = Движения.ИнформацияОТоварах.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Организация = Покупатель;
        Движение.Товар = ТекСтрокаПеречень.Товар;
        Движение.Количество = ТекСтрокаПеречень.Количество;
    КонецЦикла;

    //}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

Рисунок 2.15. – Модуль объекта.

Перейдите в режим 1С:Предприятие и проверьте результат выполненных изменений.

Регистр накоплений.

Информация о товарах

Найти... | Отменить поиск | Еще

| Период | Регистратор | Номер стр... | Орган... | Товар | Количество |
|---|---|---|---|---|---|
| − 25.03.2016 0:00:00 | Отгрузка 000000001 от 25.03.2016 0:00:00 | 3 | ООО "... | Майка... | 1 |
| − 25.03.2016 0:00:00 | Отгрузка 000000001 от 25.03.2016 0:00:00 | 2 | ООО "... | Джинс... | 25 |
| − 25.03.2016 0:00:00 | Отгрузка 000000001 от 25.03.2016 0:00:00 | 1 | ООО "... | Джинс... | 20 |
| + 25.03.2016 0:00:00 | Поступление 000000001 от 25.03.2016 0:00:00 | 3 | ЗАО "... | Джинс... | 50 |
| + 25.03.2016 0:00:00 | Поступление 000000001 от 25.03.2016 0:00:00 | 2 | ЗАО "... | Джинс... | 100 |
| + 25.03.2016 0:00:00 | Поступление 000000001 от 25.03.2016 0:00:00 | 1 | ЗАО "... | Майка... | 50 |

Рисунок 2.16. – Отгрузка: Регистр накоплений.

**Пример 2.** Регистр накопления остатков.

В ряде случаев нет необходимости учитывать остатки. Например, в случае подачи заявок на продукцию представляет интерес подсчет заявок за определенный интервал. Для подобных ситуаций предназначены оборотные регистры накопления.

Создайте в продолжение примера №1 ещё один документ Заявка.

Вкладка **Данные.** В шапке документа добавьте один дополнительный реквизит Поставщик (тип данных – СправочникСсылка.Фирмы).

Для табличной части документа выберите имя ПеречеьТоваров и внесите в нее два реквизита:

– Товар (тип данных – СправочникСсылка.Номенклатура);
– Количество (тип данных – Число).

Создайте новый регистр накопления. После установки имени регистра ИнформацияОЗаявках на вкладке Основные необходимо выбрать значение Обороты в поле Вид регистра.

После этого в окне редактирования данного объекта конфигурации необходимо перейти на вкладку Данные и сформировать следующие параметры регистра накопления:

Измерения: Фирма (тип данных – СправочникСсылка.Фирмы);

Измерения: Товар (тип данных – СправочникСсылка.Номенклатура);

Ресурсы: Количество (в качестве типа данных выберем Число).

Откройте окно редактирования документа Заявка и перейдите на вкладку Движения. Здесь требуется с помощью флажка отметить регистр ИнформацияОЗаявках, по которому будут формироваться движения.

Как и ранее, для разработки процедуры ОбработкаПроведения воспользуйтесь конструктором движений.

Перейдите в режим 1С:Предприятие и проверьте результат выполненных изменений.

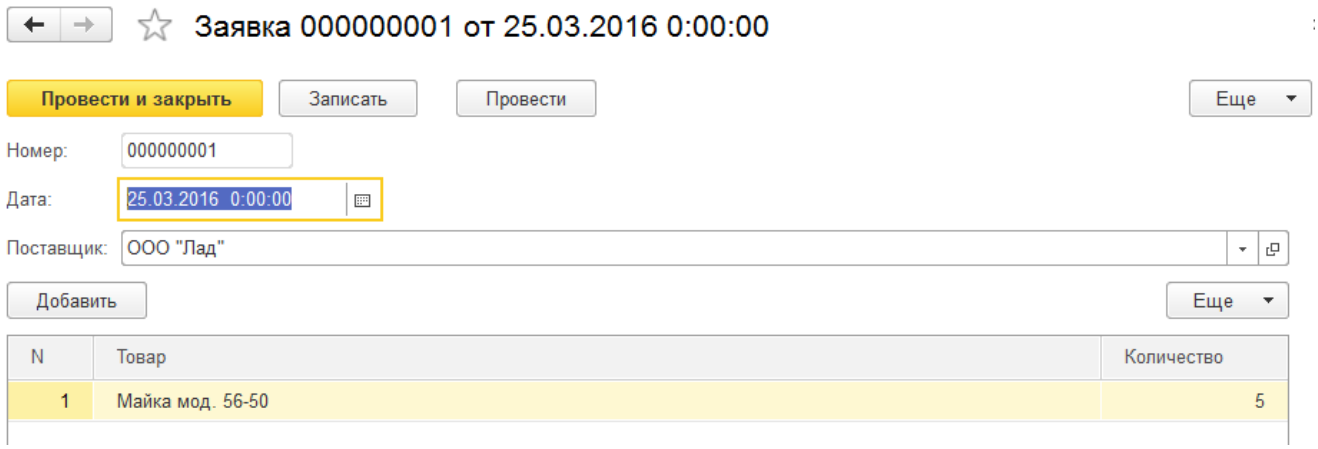



Рисунок 2.17. – Заявка в 1С системе.

При проведении документов производится формирование движений регистров накопления. Это и является основным механизмом регистрации и учета информации о событиях, привязанных ко времени.

## 2.4. ПРОГРАММА ВЫПОЛНЕНИЯ РАБОТЫ

1. Ознакомится с теоретическим материалом в методических указаниях.
2. Создать регистр сведений согласно заданной предметной области.
3. Запустить «1С:Предприятие» в режиме отладки и протестировать внесенные изменения.
4. Создать регистр накоплений согласно заданной предметной области.
5. Запустить «1С:Предприятие» в режиме отладки и протестировать внесенные изменения.
6. Сформулировать выводы.
7. Оформить отчет.

Выполнение задания подразумевает продолжение **ЛР №1, 3** (см.Приложение А).

## 2.5. СОДЕРЖАНИЕ ОТЧЕТА

5. Цель работы.
6. Описание процесса создания регистров.
7. Демонстрация результатов работы в режиме «1С:Предприятие».
8. Выводы.

## 2.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите объекты конфигурации, которые относятся к табличным типам данных.
2. Что представляют собой табличные данные с точки зрения «1С:Предприятие»?
3. В чем отличие объектных данных от табличных?
4. Что такое движения регистра, и что такое регистратор?
5. Все ли табличные записи относятся к какому-либо регистратору?
6. Для чего используются наборы записей?
7. Перечислите поля, по которым устанавливается отбор для объектов конфигурации, относящихся к табличным типам данных.
8. Опишите процесс добавления новой записи табличных данных к существующим.
9. Опишите процесс замещения существующей записи табличных данных.
10. Для чего предназначен объект конфигурации Регистр накопления?
11. Почему следует использовать регистры, хотя необходимая информация содержится в других объектах?
12. Для чего нужны измерения регистра, ресурсы и реквизиты?
13. Как создать новый регистр накопления и описать его структуру?
14. Как создать движения документа с помощью конструктора движений?
15. Как средствами встроенного языка обойти табличную часть документа и обратиться к ее данным?
16. Как показать команды открытия списка регистра в интерфейсе конфигурации и в интерфейсе формы?
17. Для чего предназначен объект конфигурации «Регистр сведений»?
18. Какими особенностями обладает объект конфигурации «Регистр сведений»?
19. В чем главные отличия регистра сведений от регистра накопления?
20. Какие поля определяют ключ уникальности регистра сведений?
21. Что такое периодический регистр сведений, и что такое независимый регистр сведений?
22. Как создать периодический регистр сведений?
23. Что такое ведущее измерение регистра?

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. М.Г. Радченко, Е.Ю. Хрусталева «1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы» (ISBN 978-5-9677-2041-3), М.: ООО «1С-Паблишинг», 2013. Электронный аналог печатного издания.

2. Ажеронок В. А., Габец А. П., Гончаров Д. И., Козырев Д. В., Кухлевский Д. С., Островерх А. В., Радченко М. Г., Хрусталева Е. Ю. Под редакцией М. Г. Радченко. "Профессиональная разработка в системе 1С:Предприятие 8" (второе издание). Том 1. М.: ООО «1С-Паблишинг», 2012. Электронная версия книги.

3. Работа с базой данных. Режим доступа: http://v8.1c.ru/overview/Term_000000641.htm?printversion=1

4. Понятие класса и объекта. Режим доступа: http://курсы-по-1с.рф/articles/понятие-класса-и-объекта/

5. Справочники в 1С. Режим доступа: http://programmist1s.ru/spravochniki-1s-8/

6. Основные объекты конфигурации в 1С: Предприятие 8.2. Режим доступа: http://1c-uroki.ru/articles/node/368

7. Язык 1С: Справочники. Режим доступа: http://1c-uroki.ru/articles/lang_1C_spravochniki

8. Документ. Режим доступа: http://v8.1c.ru/overview/Term_000000169.htm

9. Справочник. Режим доступа: http://v8.1c.ru/overview/Term_000000168.htm

10. Документы 1С - Общее описание. Режим доступа: http://1c-programist.ru/stati/dokumenty-1s-obschee-opisanie.html

11. Документы в 1С 8.3 (8.2). Режим доступа: http://programmist1s.ru/dokumentyi-v-1s/

12. Пошаговый пример разработки конфигурации в 1С 8.3. Режим доступа: http://1c-uroki.ru/articles/pochagoviy_primer_1C_8.3.3

13. Пошаговый пример разработки конфигурации в 1С 8.2 (1 часть). Режим доступа: http://1c-uroki.ru/articles/pochagoviy_primer_1C_8.2

14. Пошаговый пример разработки конфигурации в 1С 8.2 (2 часть). Режим доступа: http://1c-uroki.ru/articles/pochagoviy_primer_1C_8.2_2

15. Изучаем 1С Предприятие 8.3 на примере построения простой конфигурации. Режим доступа: http://infostart.ru/public/276629/

16. М.Г. Радченко, Е.Ю. Хрусталева «1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы» (ISBN 978-5-9677-2041-3), М.: ООО «1С-Паблишинг», 2013. Электронный аналог печатного издания.

17. Ажеронок В. А., Габец А. П., Гончаров Д. И., Козырев Д. В., Кухлевский Д. С., Островерх А. В., Радченко М. Г., Хрусталева Е. Ю. Под редакцией М. Г. Радченко. "Профессиональная разработка в системе 1С:Предприятие 8" (второе издание). Том 1. М.: ООО «1С-Паблишинг», 2012. Электронная версия книги.

18. Регистры 1С. Режим доступа: http://howknow1c.ru/programmirovanie-1c/registry-1s.html

19. Регистры. Режим доступа: http://www.mista.ru/tutor_1c/regs.htm

20. Регистры накопления в 1С. Режим доступа: http://life1c.ru/post/323

21. Регистры накопления 1С и их использование. Режим доступа: http://programmist1s.ru/registryi-nakopleniya-v-1s/

22. Регистры накопления в языке 1С 8.3, 8.2 (в примерах). Режим доступа: http://helpme1c.ru/registry-nakopleniya-v-yazyke-1s-8-v-primerax

23. Регистры сведений в языке 1С 8.3, 8.2 (в примерах). Режим доступа: http://helpme1c.ru/registry-svedenij-v-yazyke-1s-8-v-primerax

24. Объект конфигурации «Регистр накопления». Режим доступа: http://1c-ucheba.ru/obekt-konfiguracii-registr-nakopleniya/

25. Урок 6: Работа с регистрами сведений. Режим доступа: http://itpath.ru/lesson1cpart6/

26. Регистры сведений в языке 1С 8.3, 8.2 (в примерах). Режим доступа: http://helpme1c.ru/registry-svedenij-v-yazyke-1s-8-v-primerax

27. Регистр сведений. Режим доступа: http://v8.1c.ru/overview/Term_000000175.htm

28. Объект конфигурации «Регистр сведений». Режим доступа: http://1c-ucheba.ru/obekt-konfiguracii-registr-svedenij/

# ПРИЛОЖЕНИЕ А

Таблица 1 - Варианты заданий

| № варианта | Наименование предметной области | Подсистема справочников | | Подсистема документов | | Пользователи | |
|---|---|---|---|---|---|---|---|
| | | Наименование подсистем | Наименование справочников | Наименование подсистем | Наименование документов | Доступные роли | Наименование пользователя |
| 1 | Управление прайс-листами | Товары, Ценообразование | Товары<br>Типы цен<br>Поставщики | Ценовая политика | Установка цен | Администратор | Администратор |
| | | | | | | Менеджер по закупкам | ФИО студента |
| 2 | Взаиморасчеты | Номенклатура и контрагенты | Клиенты<br>Услуги | Покупки и продажи | Договор<br>Оказание услуг | Администратор | Администратор |
| | | | | | | Менеджер торгового направления | ФИО студента |
| 3 | Товарооборот | Номенклатура | Товары<br>Характеристики | Продажи | Реализация товаров | Администратор | Администратор |
| | | | | | | Менеджер по продажам | ФИО студента |
| 4 | Управление проектами | Проекты, Ресурсы | Проекты<br>Этапы<br>Сотрудники | Учет рабочего времени | Отчет о работе | Администратор | Администратор |
| | | | | | | Руководитель подразделения | ФИО студента |
| 5 | Управление персоналом | Структура предприятия, Кадры | Организации<br>Подразделения<br>Сотрудники | Учет кадровой информации | Прием на работу<br>Штатное расписание | Администратор | Администратор |
| | | | | | | Инспектор по кадрам | ФИО студента |
| 6 | Транспортные услуги | Ресурсы предприятия | Водители<br>Грузчики<br>Транспорт | Доставка товаров | Заказ клиента<br>Наряд | Администратор | Администратор |
| | | | | | | Диспетчер | ФИО студента |

| № варианта | Наименование предметной области | Подсистема справочников | | Подсистема документов | | Пользователи | |
|---|---|---|---|---|---|---|---|
| | | Наименование подсистем | Наименование справочников | Наименование подсистем | Наименование документов | Доступные роли | Наименование пользователя |
| | | | | | на доставку | службы доставки | |
| 7 | Активы предприятия | Внеоборотные активы | Виды оборудования Цеха | Учет оборудования | Ввод в эксплуатацию Перемещение оборудования | Администратор | Администратор |
| | | | | | | Бухгалтер | ФИО студента |
| 8 | Управление производством | Номенклатура | Продукция Материалы Спецификации | Производство | Требование-накладная Отчет за смену | Администратор | Администратор |
| | | | | | | Контроллер ОТК | ФИО студента |
| 9 | Управление торговлей | Номенклатура и склад | Товары на складе Единицы измерения | Покупки и продажи | Поступление товаров Реализация товаров | Администратор | Администратор |
| | | | | | | Заведующий складом | ФИО студента |
| 10 | Корпоративная мобильная связь | Контактная информация | Телефоны Сотрудники | Учет мобильной связи | Отчет о мобильной связи | Администратор | Администратор |
| | | | | | | Главный бухгалтер | ФИО студента |
| 11 | Интернет услуги | Нормативно-справочная информация | Абоненты Сотрудники | Подключения | Акт подключения услуг Отчет о предоставлении услуг | Администратор | Администратор |
| | | | | | | Мастер технической поддержки и подключений | ФИО студента |

| № варианта | Наименование предметной области | Подсистема справочников | | Подсистема документов | | Пользователи | |
|---|---|---|---|---|---|---|---|
| | | Наименование подсистем | Наименование справочников | Наименование подсистем | Наименование документов | Доступные роли | Наименование пользователя |
| 12 | Предоставление рекламных услуг | Номенклатура и контрагенты | Рекламные агентства<br>Продукты<br>Клиенты | Отчеты | Отчет о предоставлении услуг | Администратор | Администратор |
| | | | | | | Менеджер по рекламе | ФИО студента |
| 13 | Оплата труда на предприятии | Зарплата и персонал | Сотрудники<br>Тарифные ставки | Расчет заработной платы | Начисление заработной платы<br>Выплата заработной платы | Администратор | Администратор |
| | | | | | | Бухгалтер расчетного отдела | ФИО студента |
| 14 | Использование коммунальных услуг | Нормативно-справочная информация | Жилой фонд<br>Услуги | Учет услуг | Оплата услуг<br>Отчет об использовании | Администратор | Администратор |
| | | | | | | Бухгалтер-кассир | ФИО студента |

## ПРИЛОЖЕНИЕ В

Таблица 1 - Варианты заданий

| № | Наименование предметной области | Подсистема справочников | | Подсистема документов | | Пользователи | |
|---|---|---|---|---|---|---|---|
| | | Наименование подсистем | Наименование справочников | Наименование подсистем | Наименование документов | Доступные роли | Наименование пользователя |
| 1 | Управление прайс-листами | Товары, Ценообразовани | Товары<br>Типы цен | Ценовая политика | Установка цен | Администратор | Администратор |

| № | Наименование предметной области | Подсистема справочников | | Подсистема документов | | Пользователи | |
|---|---|---|---|---|---|---|---|
| | | Наименование подсистем | Наименование справочников | Наименование подсистем | Наименование документов | Доступные роли | Наименование пользователя |
| | | е | Поставщики | | | Менеджер по закупкам | ФИО студента |
| 2 | Взаиморасчеты | Номенклатура и контрагенты | Клиенты<br>Услуги | Покупки и продажи | Договор<br>Оказание услуг | Администратор | Администратор |
| | | | | | | Менеджер торгового направления | ФИО студента |
| 3 | Товарооборот | Номенклатура | Товары<br>Характеристики | Продажи | Реализация товаров | Администратор | Администратор |
| | | | | | | Менеджер по продажам | ФИО студента |
| 4 | Управление проектами | Проекты, Ресурсы | Проекты<br>Этапы<br>Сотрудники | Учет рабочего времени | Отчет о работе | Администратор | Администратор |
| | | | | | | Руководитель подразделения | ФИО студента |
| 5 | Управление персоналом | Структура предприятия, Кадры | Организации<br>Подразделения<br>Сотрудники | Учет кадровой информации | Прием на работу<br>Штатное расписание | Администратор | Администратор |
| | | | | | | Инспектор по кадрам | ФИО студента |
| 6 | Транспортные услуги | Ресурсы предприятия | Водители<br>Грузчики<br>Транспорт | Доставка товаров | Заказ клиента<br>Наряд на доставку | Администратор | Администратор |
| | | | | | | Диспетчер службы доставки | ФИО студента |
| 7 | Активы предприятия | Внеоборотные активы | Виды оборудования<br>Цеха | Учет оборудован | Ввод в эксплуатацию<br>Перемещение | Администратор | Администратор |
| | | | | | | Бухгалтер | ФИО студента |

| № | Наименование предметной области | Подсистема справочников | | Подсистема документов | | Пользователи | |
|---|---|---|---|---|---|---|---|
| | | Наименование подсистем | Наименование справочников | Наименование подсистем | Наименование документов | Доступные роли | Наименование пользователя |
| | | | | ия | оборудования | | |
| 8 | Управление производством | Номенклатура | Продукция<br>Материалы<br>Спецификации | Производство | Требование-накладная<br>Отчет за смену | Администратор | Администратор |
| | | | | | | Контроллер ОТК | ФИО студента |
| 9 | Управление торговлей | Номенклатура и склад | Товары на складе<br>Единицы измерения | Покупки и продажи | Поступление товаров<br>Реализация товаров | Администратор | Администратор |
| | | | | | | Заведующий складом | ФИО студента |
| 10 | Корпоративная мобильная связь | Контактная информация | Телефоны<br>Сотрудники | Учет мобильной связи | Отчет о мобильной связи | Администратор | Администратор |
| | | | | | | Главный бухгалтер | ФИО студента |
| 11 | Интернет услуги | Нормативно-справочная информация | Абоненты<br>Сотрудники | Подключения | Акт подключения услуг<br>Отчет о предоставлении услуг | Администратор | Администратор |
| | | | | | | Мастер технической поддержки и подключений | ФИО студента |
| 12 | Предоставление рекламных услуг | Номенклатура и контрагенты | Рекламные агентства<br>Продукты<br>Клиенты | Отчеты | Отчет о предоставлении услуг | Администратор | Администратор |
| | | | | | | Менеджер по рекламе | ФИО студента |
| 13 | Оплата труда на предприятии | Зарплата и персонал | Сотрудники<br>Тарифные ставки | Расчет заработной платы | Начисление заработной платы<br>Выплата заработной платы | Администратор | Администратор |
| | | | | | | Бухгалтер расчетного отдела | ФИО студента |

| № | Наименование предметной области | Подсистема справочников | | Подсистема документов | | Пользователи | |
|---|---|---|---|---|---|---|---|
| | | Наименование подсистем | Наименование справочников | Наименование подсистем | Наименование документов | Доступные роли | Наименование пользователя |
| 14 | Использование коммунальных услуг | Нормативно-справочная информация | Жилой фонд<br>Услуги | Учет услуг | Оплата услуг<br>Отчет об использовании | Администратор | Администратор |
| | | | | | | Бухгалтер-кассир | ФИО студента |