МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

А. А. Филиппов

РАЗРАБОТКА ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Практический курс

Построение информационных систем
на платформе 1С:Предприятие 8.3
в режиме обычного приложения

Учебное пособие

Ульяновск
УлГТУ
2021

УДК 004.451 (075.8)
ББК 32.973-018.2я73
Ф53

Рецензенты:

Руководитель проектов ООО «АгроСервис-ИТ» А. В. Романова

Ведущий программист ООО «ПРОФ-ИТ Эксперт» О. В. Ровенский

Начальник отдела развития и поддержания ИАСУП
ФНПЦ АО «НПО «Марс», канд. техн. наук, А. А. Перцев

Утверждено редакционно-издательским советом университета в качестве учебного пособия

Филиппов, Алексей Александрович
Ф53 Разработка предметно-ориентированных информационных систем. Практический курс. Построение информационных систем на платформе 1С:Предприятие 8.3 в режиме обычного приложения : учебное пособие / А.А. Филиппов. — Ульяновск : УлГТУ, 2021. — 220 с.

ISBN 978-5-9795-2137-4

Учебное пособие по курсу «Разработка предметно-ориентированных информационных систем» содержит основные сведения, необходимые для построения информационных систем на платформе 1С:Предприятие 8 в режиме обычного приложения.

Для студентов вузов, обучающихся по направлениям «Прикладная информатика(в экономике)» и другим, применяющим ЭВМ в задачах построения информационных систем.

Подготовлено на кафедре «Информационные системы».

УДК 004.451 (075.8)
ББК 32.973-018.2я73

ISBN 978-5-9795-2137-4

© Филиппов А.А., 2021
© Оформление. УлГТУ, 2021

ОГЛАВЛЕНИЕ

Введение **6**

1 Введение в платформу 1С:Предприятие 8 **7**
1.1 Объекты метаданных Платформы 7
1.2 Встроенный язык Платформы 8
1.2.1 Типы модулей конфигурации 8
1.2.2 Контекст модулей конфигурации 10
1.2.3 Примитивные типы данных 10
1.2.4 Конструкции встроенного языка 11
1.2.5 Универсальные коллекции значений 13

2 Создание информационной базы **22**

3 Универсальные объекты метаданных Платформы **28**
3.1 Константы 28
3.2 Перечисления 29
3.3 Справочники 32
3.4 Отчеты 32
3.5 Документы 32
3.6 Регистры накопления 32
3.7 Регистры сведений 33

4 Справочники **34**
4.1 Создание и настройка справочника 34
4.2 Методы работы со справочниками 40
4.3 Пример выполнения задания 44
4.4 Контрольные вопросы 55
4.5 Варианты заданий 55

5 Отчеты **73**
5.1 Создание и настройка отчета 73
5.2 Методы работы с табличным документом 77
5.3 Пример выполнения задания 82
5.4 Контрольные вопросы 88
5.5 Варианты заданий 89

6 Документы **90**
6.1 Создание и настройка документа 90
6.2 Методы работы с документами 93
6.3 Пример выполнения задания 94
6.4 Контрольные вопросы 103
6.5 Варианты заданий 103

7 Регистры накопления **113**
7.1 Создание и настройка регистра накопления 113
7.2 Методы работы с регистрами накопления 118
7.3 Пример выполнения задания 121
7.3.1 Автоматическое заполнение цены товара . . . 130
7.3.2 Контроль остатков 132
7.4 Контрольные вопросы 133
7.5 Варианты заданий 134

8 Язык запросов Платформы **143**
8.1 Методы работы с запросами 144
8.1.1 Таблицы-источники 145
8.1.2 Псевдонимы 146
8.1.3 Ограничения количества и уникальности . . . 146
8.1.4 Обращение к табличным частям 146
8.1.5 Условия запроса 147
8.1.6 Вложенные запросы 148
8.1.7 Параметры запроса 148
8.1.8 Дополнительные операторы условий запроса . 149
8.1.9 Упорядочивание результатов запроса 150
8.1.10 Группировка результатов запроса 151
8.1.11 Расчет итогов запроса 153
8.1.12 Соединение таблиц-источников 155
8.1.13 Проверка значений на равенство NULL 156
8.1.14 Объединение результатов запросов 157
8.1.15 Обращение к виртуальным таблицам регистров 158
8.1.16 Временные таблицы 159
8.1.17 Пакет запросов 161
8.1.18 Функции языка запросов 162
8.2 Виртуальные таблицы регистра накопления 164
8.2.1 Виртуальные таблицы оборотного регистра . . 164

8.2.2 Виртуальные таблицы регистра остатков 165
8.3 Конструктор запросов 166
8.4 Пример выполнения задания 175
8.4.1 Автоматическое заполнение цены товара . . . 180
8.4.2 Контроль остатков 181
8.5 Контрольные вопросы 183
8.6 Варианты заданий 184

9 Регистры сведений 193
9.1 Создание и настройка регистра сведений 193
9.2 Методы работы с регистрами сведений 195
9.3 Пример выполнения задания 200
9.4 Контрольные вопросы 209
9.5 Варианты заданий 209

Заключение 219

Библиографический список 220

ВВЕДЕНИЕ

В процессе деятельности любой современной крупной организации возникает необходимость в своевременном принятии управленческих решений. Желание максимально автоматизировать и ускорить процесс получения необходимой информации об объекте управления вызывает потребность в информационных системах различного назначения, позволяющих субъекту управления получать такую информацию своевременно и в различных разрезах.

В данном учебном пособии содержится материал, используя который можно приступить к реализации собственных предметно-ориентированных информационных систем, ядром которых будет являться платформа 1С:Предприятие 8.3. Конечно, материал учебного пособия не раскрывает все тонкости практической работы с данной платформой. В частности, не рассмотрены вопросы администрирования платформы, механизмы полнотекстового поиска, заданий и криптографии, групповая разработка решений, создание дистрибутивов и распределенной информационной базы, бизнес-процессы, анализ данных и прогнозирование. Для получение дополнительной информации следует обращаться к литературе из предложенного списка [1, 2, 3].

Задания для лабораторных работ и варианты заданий частично заимствованы из учебного пособия Меркуловой Тамары Алексеевны [2].

1 ВВЕДЕНИЕ В ПЛАТФОРМУ 1С:ПРЕДПРИЯТИЕ 8

Платформа 1С:Предприятие 8 (далее Платформа) позволяет разрабатывать информационные системы различного назначения и относится к предметно-ориентированным средствам разработки. Платформа включает в себя:

- объекты метаданных и встроенный язык Платформы для формирования абстракций над механизмами работы с базами данных, процессами разработки графического интерфейса пользователя, управления пользователями и т. д. Таким образом, Платформа позволяет разработчику сосредоточить свои усилия на реализации бизнес-логики приложения без необходимости разработки механизмов преобразования, хранения и выдачи информации в удобном для восприятия виде;
- средства для масштабирования решений, разработанных в рамках Платформы. В большинстве случаев разработчику не требуется адаптировать решение к возросшим нагрузкам;
- средства для интеграции с другими информационными системами: работа с HTML, XML, JSON, DBF, XLS файлами, создание и взаимодействие с Web и HTTP сервисами;
- средства для создания ролей и групп пользователей, позволяющие настроить варианты использования решения для различных должностей и видов деятельности;
- средства для организации обмена данными с филиалами, позволяющие формировать территориально распределенные решения.

Более подробно про архитектуру Платформы можно узнать на официальном сайте 1С [1].

1.1 Объекты метаданных Платформы

Решения, построенные на основе Платформы принято называть конфигурациями. Конфигурации строятся из базовых объектов платформы — объектов метаданных. Рассмотрим основные объекты Платформы подробнее.

Справочник, Перечисление, Константа, План видов расчета, План видов характеристик, План счетов — объекты, сохраняемые в базе данных и содержащие данные, меняющиеся сравнительно *редко*. Данные объекты метаданных содержат сведения

о сущностях некоторой проблемной области, в рамках которой производится автоматизация.

Документ, Журнал документов, Последовательность, Нумератор — объекты, предназначенные для регистрации событий и операций. Данные объекты метаданных представляют собой бизнес-процессы проблемной области.

Регистр сведений, Регистр накоплений, Регистр бухгалтерии, Регистр расчета — объекты, предназначенные для хранения информации о состояниях и количествах объектов базы данных. Данные объекты метаданных агрегируют историю изменения состояния сущностей проблемной области при совершении бизнес-процессов.

Обработка, Отчет — объекты, которые используют уже введенные в базу данные для их обработки и представления пользователю. Отчеты позволяют представить сведения о состоянии сущностей проблемной области в различных разрезах с применением различных методов группировки.

Объекты метаданных будут рассмотрены более подробно в следующих главах.

1.2 Встроенный язык Платформы

Для реализации алгоритмов конфигурации, разрабатываемой в рамках Платформы, применяется встроенный язык. Встроенный язык Платформы обычно используется в модулях различных типов.

1.2.1 Типы модулей конфигурации Платформы

Общие модули

Общие модули используются для описания часто используемых процедур и функций. Область видимости для методов общих модулей — глобальный контекст и контекст общего модуля. Общие модули не могут содержать объявления переменных, в них нет раздела основной программы.

Модуль приложения

Модуль приложения может быть только один. Область видимости для методов модуля приложения — глобальный контекст и

контекст модуля приложения. В модуле приложения располагаются обработчики событий, которые обрабатываются при старте и окончании работы решения. В модуле приложения возможно объявление переменных, описание процедур и функций, которые будут доступны в различных модулях конфигурации, кроме общих модулей и модуля внешнего соединения.

Модуль внешнего соединения

Модуль внешнего соединения может быть только один. Область видимости для методов модуля внешнего соединения — глобальный контекст и контекст модуля внешнего соединения. Модуль внешнего соединения используется вместо модуля приложения если конфигурация запускается не в режиме клиентской сессии, а через COM-соединение.

Модуль сеанса

Модуль сеанса автоматически выполняется при старте решения. Область видимости для методов модуля сеанса — контекст модуля сеанса. Модуль сеанса предназначен для инициализации параметров сеанса. Выполнение методов модуля происходит после начала исполнения модуля приложения.

Модули объектов метаданных

В модулях объектов метаданных содержатся часто используемые в рамках объекта процедуры, функции и переменные. Область видимости для методов модулей объектов метаданных — глобальный контекст и контекст модуля объекта метаданных.

Модули менеджера объектов метаданных

В модулях менеджера объектов метаданных содержатся часто используемые статические процедуры и функции, не требующие наличия ссылки на конкретный объект метаданных, и позволяющие работать со всей коллекцией экземпляров объектов определенного объекта метаданных. Область видимости для методов модулей менеджера объектов метаданных — глобальный контекст и контекст модуля менеджера объекта метаданных.

Модули форм

Модуль формы определяет методы для обработки событий элементов управления данной формы и их поведение. Область видимости для методов модулей форм — контекст модуля формы.

1.2.2 Контекст модулей конфигурации Платформы

Контекст модулей Платформы определяет доступные переменные, объекты, свойства, методы и события.

Глобальный контекст доступен во всех модулях. В глобальный контекст попадают статические методы встроенного языка Платформы, а также экспортируемые переменные, процедуры и функции других контекстов.

В контексте *модуля прикладного объекта* есть доступ к реквизитам и табличным частям объекта метаданных, а также его методам и событиям.

В контексте *модуля формы* доступны реквизиты формы и элементы управления, а также их свойства, методы и события.

1.2.3 Примитивные типы данных встроенного языка Платформы

Встроенный язык Платформы поддерживает следующие примитивные типы данных:

1. Число.
2. Строка.
3. Дата (дата, время, дата и время).
4. Булево (`Истина` или `Ложь`).
5. Неопределено (для определения отсутствующего значения).
6. Null (для определения отсутствующего значения в таблицах базы данных).
7. Тип (необходимо для представления и сравнения типов данных).

Также существует возможность создания составного типа данных, позволяющего совмещать различные типы данных в рамках одной переменной с возможностью указания конкретного типа, в том числе и в интерактивном режиме.

1.2.4 Конструкции встроенного языка Платформы

Условный оператор

```
Если <логич. выражение 1>
    ИЛИ <логич. выражение 2>
    И НЕ <логич. выражение 3> Тогда
    ...
ИначеЕсли <логич. выражение 4> Тогда
    ...
ИначеЕсли <логич. выражение 5> Тогда
    ...
Иначе
    ...
КонецЕсли;

// сокращенная форма записи
?(<логич. выражение 1>, <действие истина>, <действие ложь>)

?(<логич. выражение 1>,
    ?(<логич. выражение 2>,
        <действие истина 2>,
        <действие ложь 2>
    ),
    <действие ложь 1>
)
```

Оператор присваивания

Во встроенном языке Платформы применяется динамическая типизация — переменная связывается с типом данных в момент присваивания значения, а не в момент объявления переменной. Таким образом, в различных участках программы одна и та же переменная может принимать значения разных типов. Встроенный язык Платформы не требует обязательного предварительного объявления переменной, но переменную можно объявить заранее, используя оператор Перем.

```
<Переменная> = <Значение>;
Переменная1 = Переменная2;
ПерСтрока = "2s8dgaaффф";
ПерЧисло = 123;
ПерБулево = НЕ Истина;
ПерДата = "20080717";
ПерДата = Дата(2008, 07, 17);
ПерОбъект = Справочники.Товары.НайтиПоНаименованию("Тов1");
ПерНеопределено = Неопределено;
```

Оператор цикла

```
Для <переменная счетчик цикла> = <начальное значение>
    По <конечное значение> Цикл
    ...
    Если <переменная счетчик цикла> = 10 Тогда
        Прервать; // Прерывание цикла
    КонецЕсли;
    Если <переменная счетчик цикла> = 0 Тогда
        Продолжить; // Переход к следующей итерации
    КонецЕсли;
КонецЦикла;

Для Каждого <элемент коллекции> Из <коллекция> Цикл
    ...
КонецЦикла;

Пока <логич. выражение> Цикл
    ...
КонецЦикла;
```

Обработка исключений

```
Попытка
    ...
Исключение
    ...
КонецПопытки;
```

Процедура

```
// Переменная1 передается по ссылке
// Переменная2 передается по значению
// Если Переменная3 не определена, используется значение
// по умолчанию (Ложь)
Процедура Тест(Переменная1, Знач Переменная2,
    Переменная3 = Ложь)
    ...
    Если НЕ Переменная3 Тогда
        Возврат; // Выход из процедуры
    КонецЕсли;
    ...
КонецПроцедуры;
```

Функция

```
Функция ФТест(Переменная1, Знач Переменная2,
    Переменная3 = Ложь)
    ...
    // Предварительное объявление локальной переменной
    Перем Количество;
    ...
    Возврат Количество;
КонецФункции;
```

1.2.5 Универсальные коллекции значений встроенного языка Платформы

Универсальные коллекции значений встроенного языка Платформы предназначены для хранения временных наборов данных в течение сеанса работы пользователя.

Коллекции *не* являются объектами информационной базы и служат для агрегации, группировки, анализа и обработки информации.

В рамках встроенного языка Платформы можно выделить следующие основные универсальные коллекции:

1. Массив.
2. Фиксированный массив.
3. Список значений.
4. Таблица значений.
5. Структура и соответствие.

Массивы

Массив — структура данных, хранящая набор значений (элементов массива), идентифицируемых по индексу или набору индексов, принимающих целые (или приводимые к целым) значения из некоторого заданного непрерывного диапазона. К элементу массива можно обращаться по индексу через оператор `[...]`, причем индекс первого элемента равен 0.

Создание нового массива производится следующим образом:

```
// Одномерный массив без элементов
Мас1 = Новый Массив();
// Одномерный массив из 10 элементов
// При создании массива было создано 10 элементов
// со значение Неопределено
Мас2 = Новый Массив(10);
```

```
// Двумерный массив (матрица 4 на 5)
Мас3 = Новый Массив(4,5);
```

Для добавления и удаления элементов массива используются методы Добавить, Вставить и Удалить объекта Массив:

```
Мас1 = Новый Массив();
Мас1.Добавить("Иванов");  // Индекс = 0
Мас1.Добавить("Петров");  // Индекс = 1
Мас1.Добавить("Сидоров"); // Индекс = 2
// После вставки элемента Сидоров
// элементы Иванов и Петров будут сдвинуты ниже
Мас1.Вставить(0,"Сидоров");
// После удаления элемента Сидоров
// элементы Иванов и Петров сдвинутся обратно
Мас1.Удалить(0);

Мас2 = Новый Массив(4,5);
Мас2[0][0] = "Иванов";
Мас2[0][1] = "Петров";
// После вставки элемента Сидоров
// элементы Иванов и Петров будут сдвинуты ниже
Мас2[0].Вставить(0,"Сидоров");
// Следующий элемент будет относиться ко 2-у
// измерению и будет являться 6-м (Индекс = 5)
Мас2[1].Добавить("Петров");
```

Перебрать элементы массива можно следующим образом:

```
//1-й вариант
Для Каждого Элемент Из Массив Цикл
    Сообщить(Элемент);
КонецЦикла;

//2-й вариант
Для Ит = 0 По Массив.ВГраница() Цикл
    Сообщить(Массив[Ит]);
КонецЦикла;

//3-й вариант
Для Ит = 0 По Массив.Количество() - 1 Цикл
    Сообщить(Массив.Получить(Ит));
КонецЦикла;
```

В массив можно загружать данные из других коллекций или выгружать данные из него в другие коллекции.

Фиксированный массив

Фиксированные массивы создаются на основе «обычных» массивов:

```
ФиксированныйМассив = Новый ФиксированныйМассив(Массив);
```

При использовании фиксированного массива:

- нельзя менять значения элементов, находящихся в массиве;
- нельзя добавлять новые элементы в массив;
- нельзя удалять имеющиеся элементы из массива.

Список значений

Список значений — это аналог одномерного массива, который предназначен в основном для решения задачи выбора значения из предопределенного списка при разработке графического интерфейса пользователя.

Список значений можно представить в виде множества элементов с типом `ЭлементСпискаЗначений`, каждый элемент содержит следующие свойства:

1. Значение — хранимое значение.
2. Представление — пользовательское представление значения, которое будет показано при выводе значения на экран. Если представление не указано, то оно формируется системой автоматически на основе значения.
3. Пометка — используется для отметки некоторых значений из списка.
4. Картинка — графическое изображение, связанное с данным значением.

Создание списка значений производится следующим образом:

```
Список = Новый СписокЗначений();
```

Для добавления и удаления элементов списка значений используются методы `Добавить`, `Вставить`, `Удалить`, `Очистить` и `Сдвинуть` объекта `СписокЗначений`:

```
// Удаление всех элементов
Список.Очистить();
// Добавление элемента Иванов (Индекс = 0)
// с указанием представления Иванов Иван
Список.Добавить("Иванов", "Иванов Иван");
```

```
// Добавление элемента Петров (Индекс = 1)
// с указанием представления Петров Петр
// и установкой пометки в значение Истина
Список.Добавить("Петров", "Петров Петр", Истина);
// Удаление элемента Петров (Индекс = 1)
Список.Удалить(1);
// После вставки элемента Сидоров в начало списка
// элемент Иванов сдвигается вниз (Индекс = 1)
Список.Вставить(0, "Сидоров");
// Сдвиг элемента Иванов (Индекс = 1)
// на одну позицию вверх
Список.Сдвинуть(1, -1);
```

Перебрать элементы списка значений можно следующим образом:

```
// 1-й вариант
Для Каждого Элемент Из Список Цикл
    Сообщить(Элемент.Значение);
КонецЦикла;

// 2-й вариант
Для Ит = 0 По Список.Количество() - 1 Цикл
    Сообщить(Список[Ит].Значение);
КонецЦикла;

// 3-й вариант
Для Ит = 0 По Список.Количество() - 1 Цикл
    Сообщить(Список.Получить(Ит).Значение);
КонецЦикла;

// Поиск элемента в списке значений
Элемент = Список.НайтиПоЗначению("Иванов");
Если Элемент <> Неопределено Тогда
    Сообщить("Элемент найден!");
КонецЕсли;
```

Список значений также позволяет программно считывать и устанавливать значение свойства `Пометка` своих элементов:

```
Список = Новый СписокЗначений();
Список.Добавить("Иванов", , Истина); // Пометка установлена
Список.Добавить("Петров");  // Пометка не установлена
Список.Добавить("Сидоров"); // Пометка не установлена
ВыборПроизведен = Список
    .ОтметитьЭлементы("Укажите сотрудников");
Если ВыборПроизведен Тогда
    Для Каждого Элемент Из Список Цикл
        Если Элемент.Пометка Тогда
```

```
            Сообщить("Отмечен сотрудник" +
                Элемент.Значение);
        КонецЕсли;
    КонецЦикла;
КонецЕсли;
```

Методы списка значений ВыбратьЭлемент и ОтметитьЭлементы при вызове отображают диалоговое окно, в котором пользователь может выбрать один элемент или расставить пометки, тем самым выбрав несколько элементов.

Таблица значений

Таблица значений состоит из строк и колонок. В отличие от списка значений, таблица значений имеет структуру, которую определяет разработчик. При работе со строками таблицы значений необходимо понимать, что строка — это объект типа Строка- ТаблицыЗначений.

При создании новой ТЗ в нее обычно добавляются колонки:

```
Таблица = Новый ТаблицаЗначений();
Таблица.Колонки.Добавить("Ссылка");
Таблица.Колонки.Добавить("Наименование");
Таблица.Колонки.Добавить("Цена");
Таблица.Колонки.Добавить("Количество");
```

Для добавления и удаления строк таблицы значений используются методы Добавить, Вставить, Сдвинуть, Удалить и Очистить объекта ТаблицаЗначений:

```
// Добавление строки
НовСтрока = Таблица.Добавить();
НовСтрока.Ссылка = Ссылка;
НовСтрока.Наименование = "Товар 1";
НовСтрока.Цена = 14.1;
НовСтрока.Количество = 100;
// Добавление строки в начало таблицы значений
НовСтрока = Таблица.Вставить(0);
НовСтрока.Ссылка = Ссылка;
НовСтрока.Наименование = "Товар 2";
НовСтрока.Цена = 29.3;
// Сдвиг 2-й строки на одну позицию вверх
Таблица.Сдвинуть(1, -1);
// Удаление 1-й строки
Таблица.Удалить(0);
// Удаление всех строк
Таблица.Очистить();
```

Перебрать строки таблицы значений можно следующим образом:

```
// 1-й способ
Для Каждого Строка Из Таблица Цикл
    Сообщить(Строка.Наименование);
    Сообщить(Строка.Цена);
КонецЦикла;

// 2-й способ
Для Ит = 0 По Таблица.Количество() - 1 Цикл
    Сообщить(Таблица[Ит].Наименование);
    Сообщить(Таблица[Ит].Цена);
КонецЦикла;

// 3-й способ
Для Ит = 0 По Таблица.Количество() - 1 Цикл
    Строка = Таблица.Получить(Ит);
    Сообщить(Строка.Наименование);
    Сообщить(Строка.Цена);
КонецЦикла;
```

Поиск строк в таблице значений производится с помощью методов `Найти` и `НайтиСтроки` объекта `ТаблицаЗначений`:

```
// Поиск строки
// Метод Найти возвращает ссылку
// на строку таблицы значений
Строка = Таблица.Найти(14.1, "Цена");
Если Строка <> Неопределено Тогда
    Сообщить("Товар с такой ценой найден!"
        + Строка.Наименование);
КонецЕсли;

// Поиск нескольких строк
// Метод НайтиСтроки возвращает массив ссылок
// на строки таблицы значений
Отбор = Новый Структура("Количество", 100);
МассивСтрок = Таблица.НайтиСтроки(Отбор);
Сообщить("Найдено " + МассивСтрок.Количество()
    + " товаров с количеством = 100");
```

Так как методы `Найти` и `НайтиСтроки` возвращают ссылки на строки таблицы значений, изменение какой-либо строки в таблице значений приведет к изменениям этих же строк в переменных `Строки` и `МассивСтрок` и наоборот.

Таблица значений позволяет упорядочить содержащиеся в ней данные, для чего используется метод `Сортировать`:

```
// Сортировка таблицы по колонке Наименование
// в порядке возрастания
Таблица.Сортировать("Наименование");
// Сортировка таблицы по колонке Цена
// в порядке убывания
Таблица.Сортировать("Цена Убыв");
// Сортировка таблицы по колонке Цена в порядке убывания
// и колонке Наименование в порядке возрастания
Таблица.Сортировать("Цена Убыв, Наименование Возр");
```

Группировка строк таблицы значений и подсчет итогов по группе реализуется с помощью метода Свернуть. Метод Свернуть таблицы значений принимает два параметра: КолонкиГруппировок и КолонкиСуммирования. Данный метод осуществляет свертку таблицы значений по указанным колонкам группировки. Строки, у которых совпадают значения в колонках, указанных в первом параметре, сворачиваются в одну строку. Значения этих строк, хранящиеся в колонках, указанных во втором параметре, суммируются. Списки колонок не должны пересекаться. Колонки, не вошедшие ни в один из списков колонок, после выполнения метода удаляются из таблицы значений.

Если в колонке установлен тип и он единственный, то при суммировании будет предприниматься попытка преобразования значения к типу Число.

Если колонке не присвоены типы, то в процессе суммирования будут принимать участие только значения, имеющие тип Число, значения других типов будут игнорироваться.

Если в колонке несколько типов и среди них есть тип Число, то в процессе суммирования будут принимать участие только значения, имеющие тип Число, значения других типов будут игнорироваться.

Если в колонке несколько типов и среди них нет типа Число, то результат суммирования будет 0, который будет присвоен в соответствующую колонку, где будет преобразован к значению по умолчанию для типа, установленного в колонке.

```
Таблица.Свернуть("Наименование", "Цена, Количество");
```

Для создания копии таблицы значений используется метод Скопировать:

```
КопияТаблицы = Таблица.Скопировать();
Таблица.Свернуть("Наименование", "Цена, Количество");
```

Переменная КопияТаблицы содержит версию таблицы значений Таблица до ее группировки.

Таблица значений позволяет подсчитать итог (сумма всех значений в строках данной колонки) по колонке числового типа:

```
ВсегоКоличество = Таблица.Итог("Количество");
ВсегоЦена = Таблица.Итог("Цена");
```

Из таблицы значений можно выгрузить колонку в массив и загрузить ее из массива:

```
МассивЦен = Таблица.ВыгрузитьКолонку("Цена");
Таблица.ЗагрузитьКолонку("Цена", МассивНовыхЦен);
```

Структура и соответствие

Структура и соответствие являются динамическими коллекциями значений. Каждый элемент такой коллекции представляет собой пару «ключ» и «значение». Ключи структуры и соответствия уникальны, поэтому они однозначно идентифицируют элемент коллекции.

В структуре ключи могут иметь только строковый тип данных и должны подчиняться правилам именования переменных. В соответствиях ключи могут иметь почти любой тип данных.

Структуры часто используются для организации отборов:

```
ОтборЦена = Новый Структура("Цена", 100);
Выборка = Справочники.Номенклатура.Выбрать(, ОтборЦена);
```

Структуры также можно использовать для передачи списка параметров:

```
// Создание новой структуры с ключами ДатаНач и ДатаКон
Параметры = Новый Структура("ДатаНач, ДатаКон");
// Установка значения для ключа ДатаНач
Параметры.ДатаНач = Дата(1900, 01, 01);
// Установка значения для ключа ДатаКон
Параметры.ДатаКон = ДобавитьМесяц(Параметры.ДатаНач, 1);
// Добавление элемента с указанием ключа и значения
Параметры.Вставить("СворачиватьИтоги", Истина);
СформироватьОтчет(Параметры);
```

Для получения значения элемента структуры необходимо использовать ключ в качестве свойства, поскольку ключи являются свойствами структуры:

```
ДатаНач = Параметры.ДатаНач;
ДатаКон = Параметры.ДатаКон;
```

Перебрать элементы структуры можно следующим образом:

```
Для Каждого Элемент Из Параметры Цикл
    Ключ = Элемент.Ключ;
    Значение = Элемент.Значение;
КонецЦикла;
```

Создание соответствия происходит следующим образом:

```
ВидыРасчета = Новый Соответствие();
БазовыеВидаРасчета = Новый Соответствие();
```

Для вставки и удаления элементов соответствия используются методы `Вставить`, `Удалить` и `Очистить` объекта `Соответствие`:

```
// Вставка элемента соответствия
// Ключ --- Ссылка, Значение --- Соответствие
ВидыРасчета.Вставить(ВыборкаВидовРасчета.Ссылка,
    БазовыеВидаРасчета);
// Удаление элемента соответствия по ключу
ВидыРасчета.Удалить(ВыборкаВидовРасчета.Ссылка);
// Удаление всех элементов
ВидыРасчета.Очистить();
```

Для получения значения элемента соответствия необходимо использовать оператор `[...]` или метод `Получить`, так как ключи соответствия могут иметь почти любой тип данных:

```
ТелефонСотрудника = ТелефоныСотрудников
    .Получить(Сотрудник);
ТелефонСотрудника = ТелефоныСотрудников[Сотрудник];
```

Перебрать элементы соответствия можно следующим образом:

```
Для Каждого Элемент Из Соответствие Цикл
    Сообщить(Элемент.Ключ + " - " + Элемент.Значение);
КонецЦикла;
```

2 СОЗДАНИЕ ИНФОРМАЦИОННОЙ БАЗЫ

Для выполнения лабораторных работ необходимо установить и настроить Платформу. Лучше всего использовать учебную версию Платформы. Учебная версия Платформы является полностью бесплатной, но обладает некоторым набором ограничений, делающим ее непригодной для использования в реальных условиях.

Все примеры в рамках данного учебного пособия выполнены в учебной версии платформы 1С: Предприятие 8.3.16.

После установки Платформы необходимо запустить программу 1С: Предприятие 8.3. После запуска программы откроется окно, представленное на рисунке 2.1.

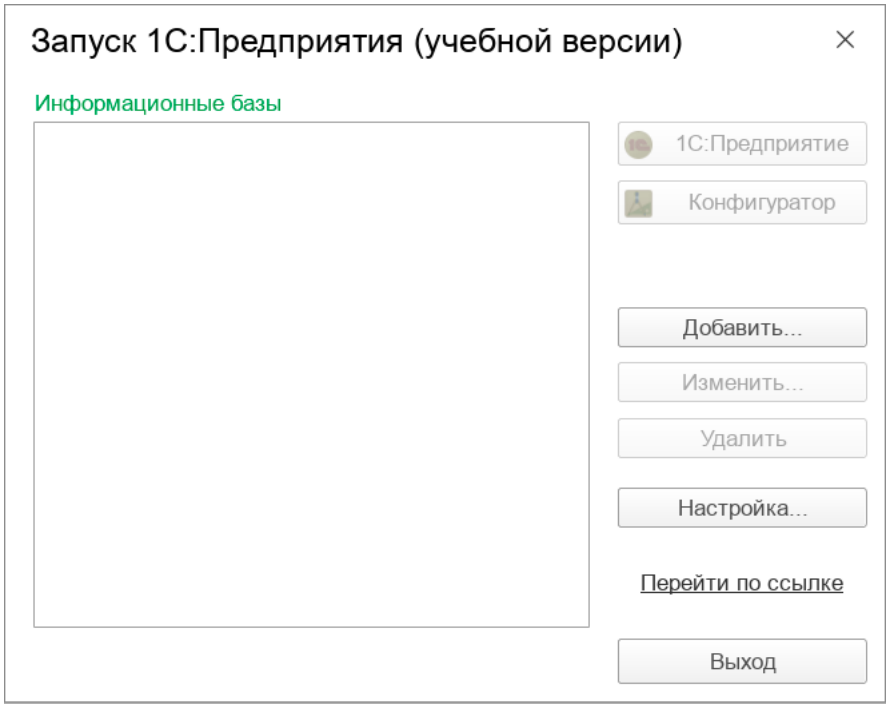


Рис. 2.1. Окно выбора информационной базы

Все лабораторные работы должны быть выполнены в рамках одной информационной базы. Для создания новой информационной базы необходимо нажать на кнопку «Добавить» (рис. 2.1). В появившемся окне мастера добавления информационной базы необходимо выбрать пункт «Создание новой информационной базы» (рис. 2.2) и нажать на кнопку «Далее».

На втором шаге создания новой информационной базы необходимо выбрать пункт «Создание информационной базы без конфигурации для разработки новой конфигурации...» (рис. 2.3) и нажать на кнопку «Далее».

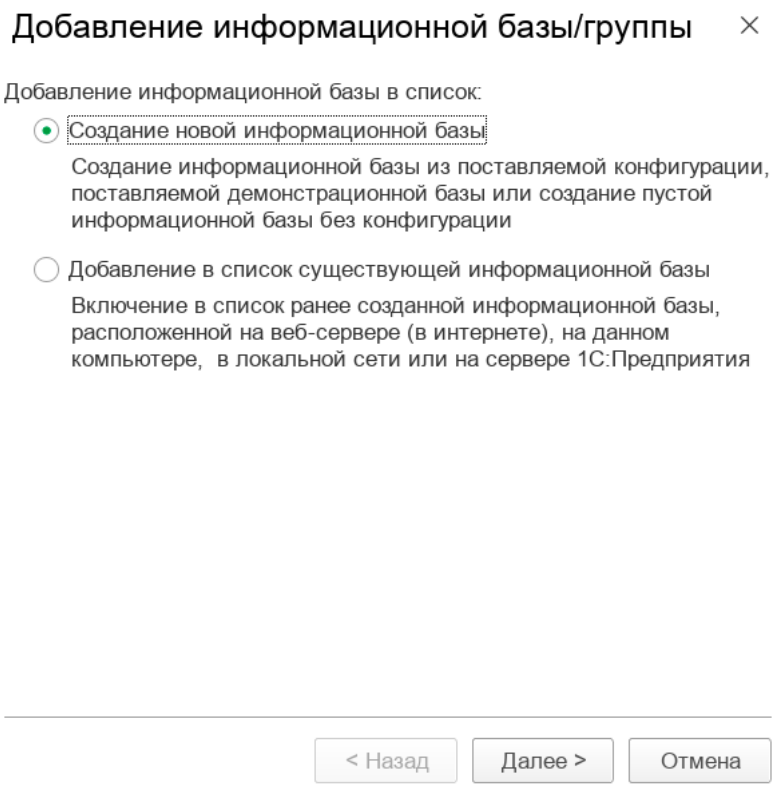


Рис. 2.2. Создание информационной базы, шаг 1

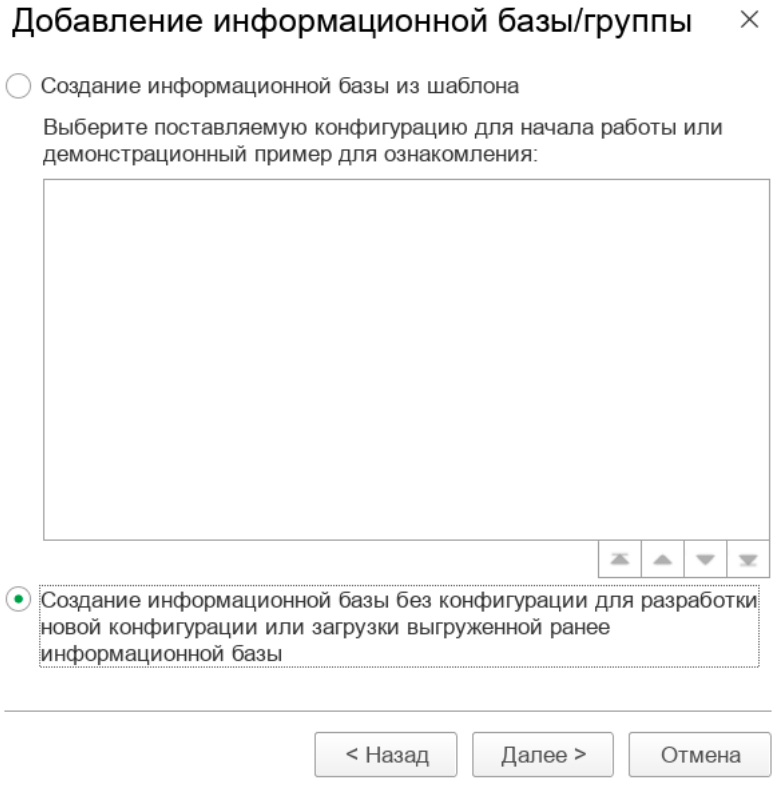


Рис. 2.3. Создание информационной базы, шаг 2

На третьем шаге создания новой информационной базы необходимо указать свою фамилию и номер группы в качестве наименования информационной базы, а в качестве типа расположения информационной базы выбрать пункт «На данном компьютере или на компьютере в локальной сети» (рис. 2.4). Для перехода к следующему шагу необходимо нажать на кнопку «Далее».

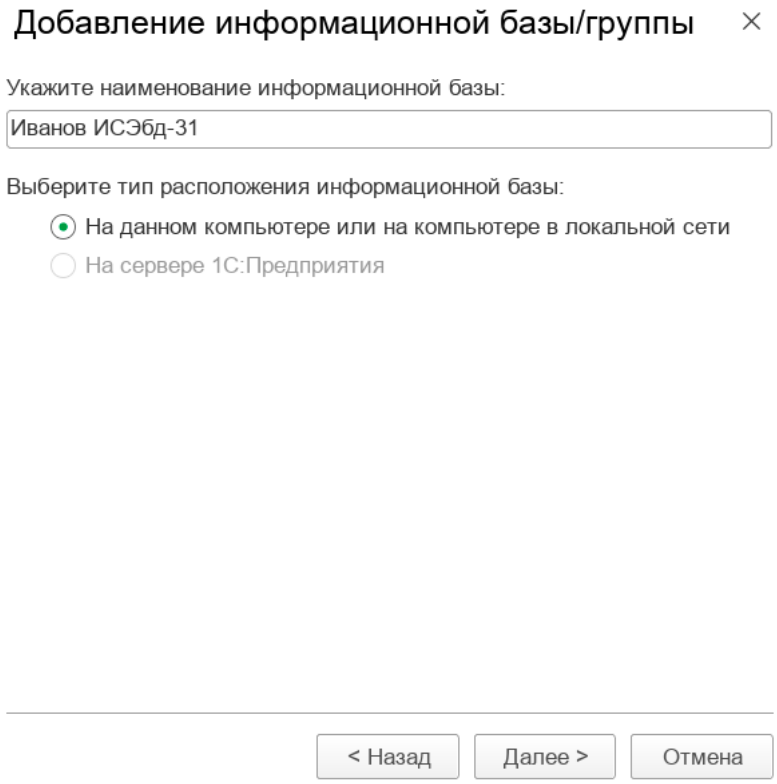


Рис. 2.4. Создание информационной базы, шаг 3

На четвертом шаге создания новой информационной базы необходимо указать путь до каталога, в котором будут храниться файлы информационной базы (рис. 2.5), а затем нажать на кнопку «Далее», а затем на кнопку «Готово».

В случае успеха созданная информационная база появится в списке (рис. 2.1) и ее можно будет использовать.

Информационная база Платформы состоит из конфигурации и данных. Конфигурация описывает структуру прикладного решения на уровне объектов метаданных Платформы, алгоритмов, экранных форм и т. д. С помощью разработанной конфигурации прикладное решение реализует функциональные возможности, необходимые для автоматизации бизнес-процессов и ведения учета: ввод данных, анализ данных, накопление данных, планирова-

ние, прогнозирование, ведение учета, формирование отчетности и т. д.

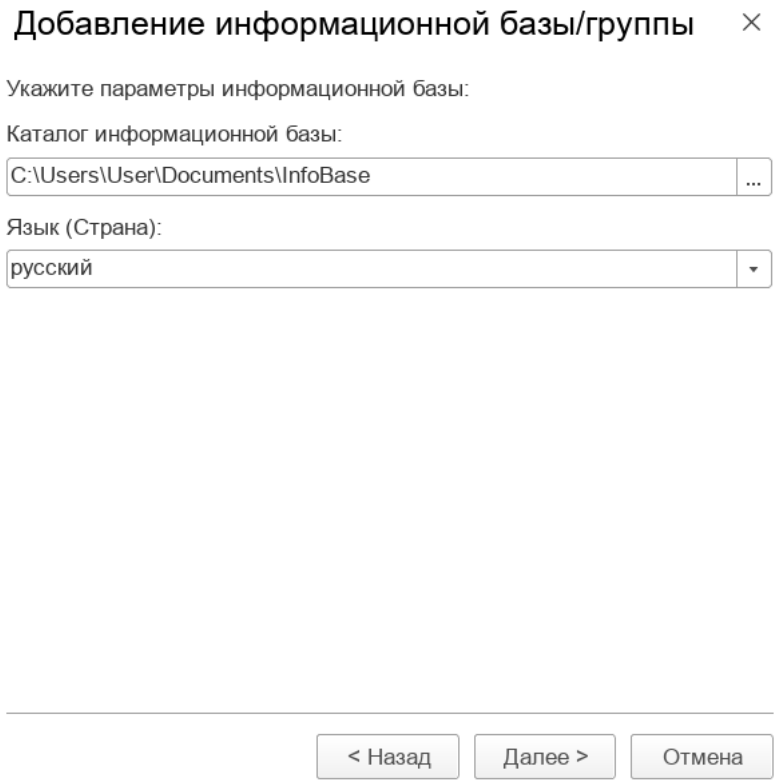


Рис. 2.5. Создание информационной базы, шаг 4

Как видно из рисунка 2.1 информационную базу можно запустить в двух режимах:

1. 1С: Предприятие — интерактивный пользовательский режим без возможности внесения изменений в конфигурацию информационной базы. В рамках данного режима производится ввод информации в информационную базу.
2. Конфигуратор — режим разработки прикладного решения. В рамках данного режима существует возможность изменения конфигурации прикладного решения.

Таким образом, для выполнения лабораторных работ необходимо использовать режим «Конфигуратор». Для тестирования и отладки разработанной конфигурации необходимо запустить режим «1С: Предприятие» нажав горячую клавишу *F5* или выбрав пункт меню «Отладка → Начать отладку».

Платформа позволяет разрабатывать прикладные решения для работы в двух режимах запуска:

1. Обычное приложение. В режиме обычного приложения используются «обычные» формы, характерные для решений, разработанных с применением Платформы версии 8.2. Обычное приложение работает только в режиме толстого клиента.
2. Управляемое приложение. В режиме управляемого приложения используются управляемый командный интерфейс и управляемые формы. При запуске управляемого приложения интерфейс пользователя генерируется на основе декларативного представления для работы в определенном окружении, что дает возможность запускать решение в следующих режимах:
   - тонкий клиент,
   - веб-клиент,
   - мобильный клиент,
   - толстый клиент,
   - приложение мобильной платформы.

   При этом, управляемое приложение выдвигает определенные требования к организации программного кода модулей. С помощью директив встроенного языка Платформы необходимо четко разграничить где выполняется определенный метод: на клиенте или на сервере.

Для выбора режима запуска необходимо в свойствах конфигурации установить значение параметра «Основной режим запуска» в одно из двух значений (рис. 2.6): «Управляемое приложение» или «Обычное приложение».

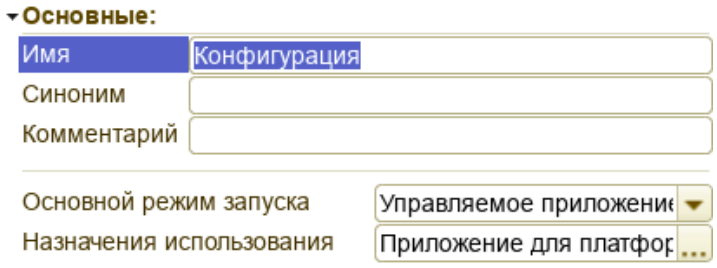


Рис. 2.6. Настройка свойств конфигурации

Для открытия свойств конфигурации необходимо выполнить двойное нажатие левой кнопкой мыши на объекте «Конфигурация» дерева объектов метаданных, либо нажать правой кнопкой мыши на объекте «Конфигурация», а затем выбрать пункт «Свойства» в появившемся меню.

В режиме конфигуратора существует возможность архивации информационной базы и сохранения конфигурации. В первом случае формируется файл с расширением *.dt*, содержащий данные и конфигурацию, во втором случае формируется файл с расширением *.cf*, содержащий только конфигурацию.

Для выгрузки информационной базы необходимо выбрать пункт меню «Администрирование → Выгрузить информационную базу» и указать путь и название целевого файла в диалоге сохранения.

Существует возможность загрузки информационной базы из файла в текущую информационную базу с полным замещением как данных так и конфигурации. Для этого необходимо выбрать пункт меню «Администрирование → Загрузить информационную базу» и открыть необходимый файл.

3 УНИВЕРСАЛЬНЫЕ ОБЪЕКТЫ МЕТАДАННЫХ ПЛАТФОРМЫ

3.1 Константы

Константы в конфигурации на базе Платформы используются для хранения отдельных значений, которые изменяются достаточно редко или не изменяются вообще.

Для создания константы необходимо выбрать узел «Константы» в дереве объектов метаданных, а затем нажать на него правой кнопкой мыши и выбрать пункт «Добавить» в появившемся меню (рис. 3.1).

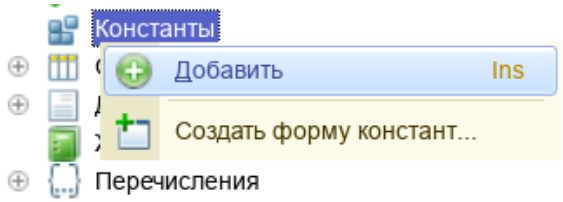


Рис. 3.1. Создание константы

В окне свойств константы можно указать ее имя, тип данных, а также значения других свойств (рис. 3.2).

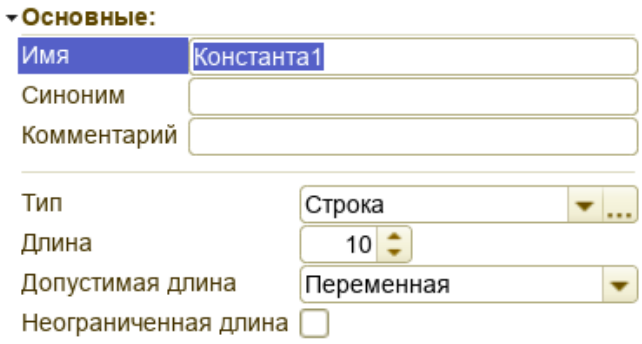


Рис. 3.2. Окно свойств константы

Рассмотрим основные примеры работы с константами средствами встроенного языка Платформы:

```
// Чтение значения из константы
Название = Константы.НазваниеОрганизации.Получить();
// Запись нового значения в константу
Константы.НазваниеОрганизации.Установить("Организация");
// Создание набора из двух констант
Набор = Константы.СоздатьНабор("Руководитель, ИНН");
// Чтение из базы значений сразу двух констант
Набор.Прочитать();
```

```
// Изменение значений констант в наборе (не в базе)
Набор.Руководитель = "Иванов И.И.";
Набор.ИНН = "1234567890";
// Запись в базу значений сразу двух констант
Набор.Записать();
```

При использовании набора констант все входящие в него константы считываются и записываются одновременно в рамках одной транзакции, т. е. все операция над несколькими константами, входящими в набор, производятся одновременно.

3.2 Перечисления

Перечисления в рамках Платформы используются для форсирования перечня значений, который может быть использован в качестве типа данных.

Список значений перечисления создается в конфигураторе во время разработки и его нельзя редактировать в режиме пользователя. Пользователь получает возможность выбрать конкретное значение перечисления из выпадающего списка в интерактивном режиме.

Каждый элемент перечисления содержит имя и синоним. Обычно имя элемента перечисления используется в алгоритмах конфигурации, а синоним — для представления элемента перечисления в режиме пользователя.

Для создания перечисления необходимо выбрать узел «Перечисления» в дереве объектов метаданных, а затем нажать на него правой кнопкой мыши и выбрать пункт «Добавить» в появившемся меню (рис. 3.3).

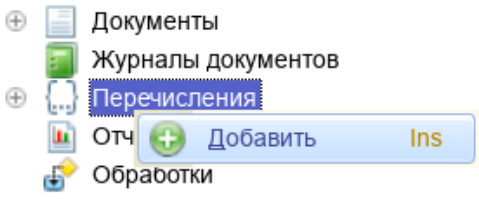


Рис. 3.3. Создание перечисления

На рисунке 3.4 представлено окно свойств вновь созданного перечисления. Как видно из рисунка 3.4, окно свойств перечисления позволяет указать имя и синоним перечисления на вкладке

«Основные», и перечень значений перечисления на вкладке «Данные» (рис. 3.5).

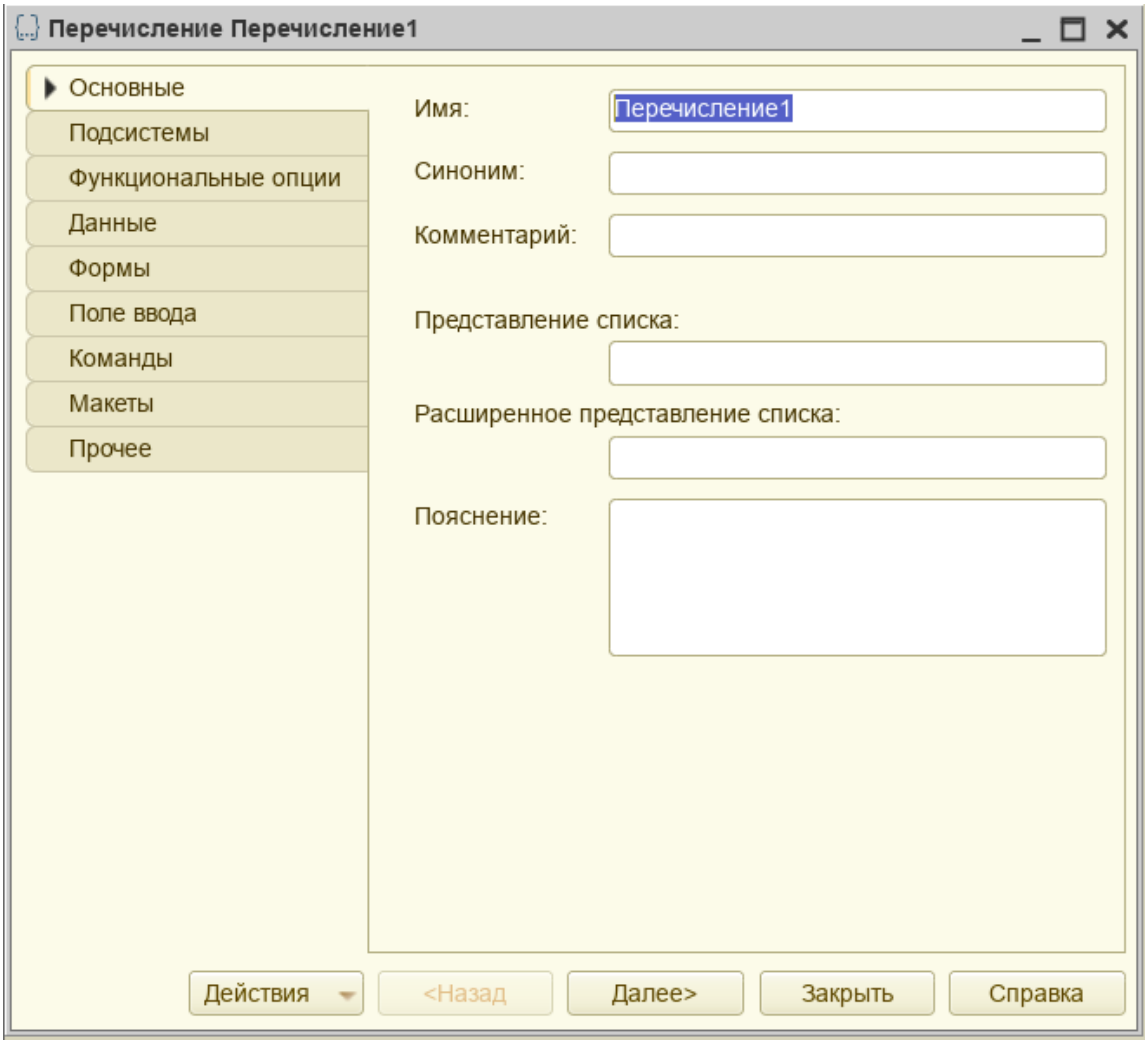


Рис. 3.4. Вкладка «Основные» окна свойств перечисления

В окне свойств (рис. 3.6) элемента перечисления можно задать значения его имени и синонима.

Рассмотрим основные примеры работы с перечислениями средствами встроенного языка Платформы:

```
Если ТипКлиента = Перечисления.ТипКлиента.Обычный Тогда
    //расчет скидки для обычных клиентов
    ...
ИначеЕсли ТипКлиента = Перечисления.ТипКлиента.Важный Тогда
    //расчет скидки для важных клиентов
    ...
Иначе
    Сообщить("Не указан тип клиента!");
КонецЕсли;
```

Помимо пользовательских перечислений, которые создаются в момент разработки, в Платформе присутствуют системные перечисления, позволяющие оперировать свойствами системных объектов платформы: режим диалоговых окон, ориентация страницы, тип элементов управления формы и т. д.

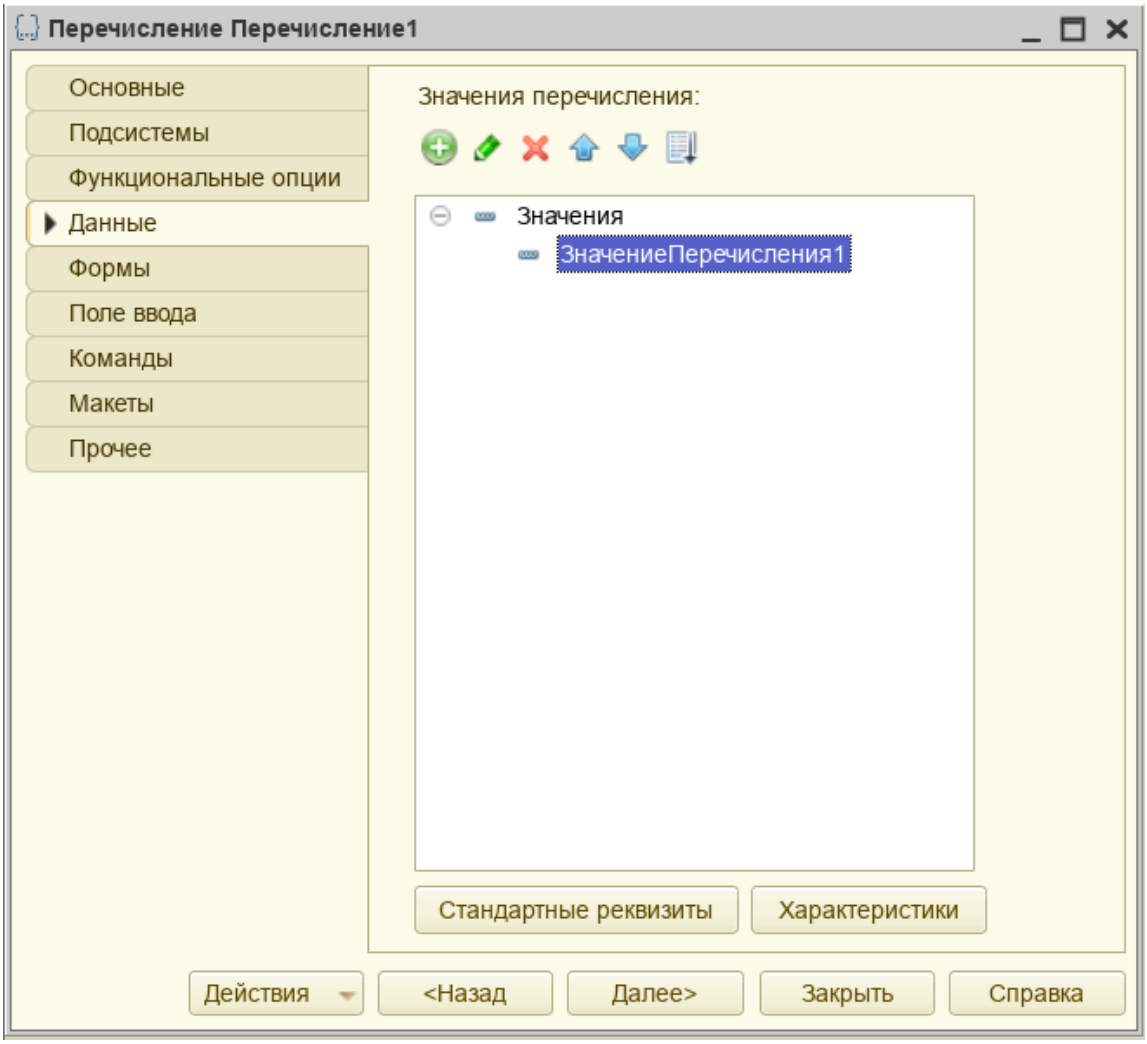


Рис. 3.5. Вкладка «Данные» окна свойств перечисления

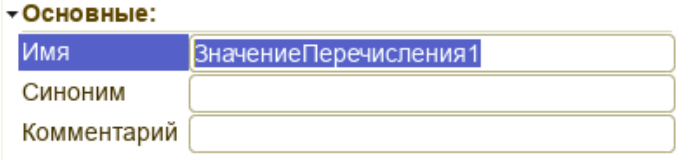


Рис. 3.6. Окно свойств элемента перечисления

3.3 Справочники

Справочники в рамках Платформы используются для хранения различной информации о сущностях (объектах) предметной области. Например, справочник «Товары» может содержать информацию об артикуле, наименовании, единице измерения и т. д. некоторого товара.

Более подробно справочники будут рассмотрены в главе 4.

3.4 Отчеты

Отчеты в рамках Платформы используются для агрегирования информации о состоянии сущностей предметной области с возможностью фильтрации и за определенный период времени. Например, оборотно-сальдовая ведомость, ведомость поступления товаров, расчетный лист сотрудника и т. д.

Более подробно отчеты будут рассмотрены в главе 5.

3.5 Документы

Документы в рамках Платформы используются для формирования и изменения состояния сущностей предметной области через реализацию определенных бизнес-процессов. Например, поступление товара, возврат товара поставщику, реализация товара и т. д.

Более подробно документы будут рассмотрены в главе 6.

3.6 Регистры накопления

Регистры накопления в рамках Платформы используются для учета движения сущностей, представляющих различные виды материальных, товарных, финансовых и других видов средств. Фактически, регистр накопления позволяет «накапливать» числовые данные в разрезе нескольких измерений, например, для автоматизации складского учета можно учитывать поступление и выбытие товаров в разрезе товаров и складов.

Более подробно регистры накопления будут рассмотрены в главе 7.

3.7 Регистры сведений

Регистры сведений в рамках Платформы используются для хранения произвольных данных в разрезе нескольких измерений. Также регистры сведений позволяют хранить историю изменения свойств сущностей предметной области. Например, в регистре сведений можно хранить курсы валют в разрезе валют, или цены в разрезе номенклатуры и типа цен.

Более подробно регистры сведений будут рассмотрены в главе 9.

4 СПРАВОЧНИКИ

Справочники в рамках Платформы используются для хранения различной информации о сущностях (объектах) предметной области. Например, справочник «Товары» может содержать информацию об артикуле, наименовании, единице измерения и т. д. некоторого товара.

4.1 Создание и настройка справочника

Для создания справочника необходимо выбрать узел «Справочники» в дереве объектов метаданных, а затем нажать на него правой кнопкой мыши и выбрать пункт «Добавить» в появившемся меню (рис. 4.1).

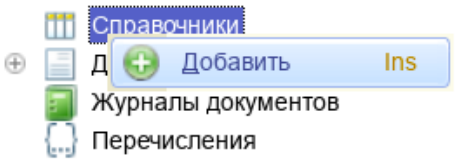


Рис. 4.1. Создание справочника

На рисунке 4.2 представлена вкладка «Основные» окна свойств вновь созданного справочника. Данная вкладка позволяет указать имя и синоним справочника. Имя определяет название объекта справочника для использования в алгоритмах решения, а синоним — название справочника в пользовательском режиме.

Вкладка «Иерархия» окна свойств справочника представлена на рисунке 4.3 и позволяет включать или выключать режим организации элементов справочника в виде дерева с помощью флага «Иерархический справочник». Параметр «Вид иерархии» может принимать два значения:

1. «Иерархия групп и элементов». При использовании данного значения параметра появляется возможность помимо элементов справочника создавать группы элементов. Группы элементов справочника предназначены для интерактивной группировки элементов по определенным критериям. Например, в группу с названием «Поставщики» справочника «Контрагенты» могут быть добавлены все элементы данного справочника, представляющие поставщиков. При этом, в качестве родительских элементов иерархического справочника могут использоваться только группы.

2. «Иерархия элементов». При использовании данного параметра появляется возможность формирования иерархии элементов справочника без использования групп элементов. В данном случае в качестве родительских элементов иерархического справочника могут выступать только элементы. Например, с помощью данного вида иерархии можно представить древовидную структуру подразделений организации.

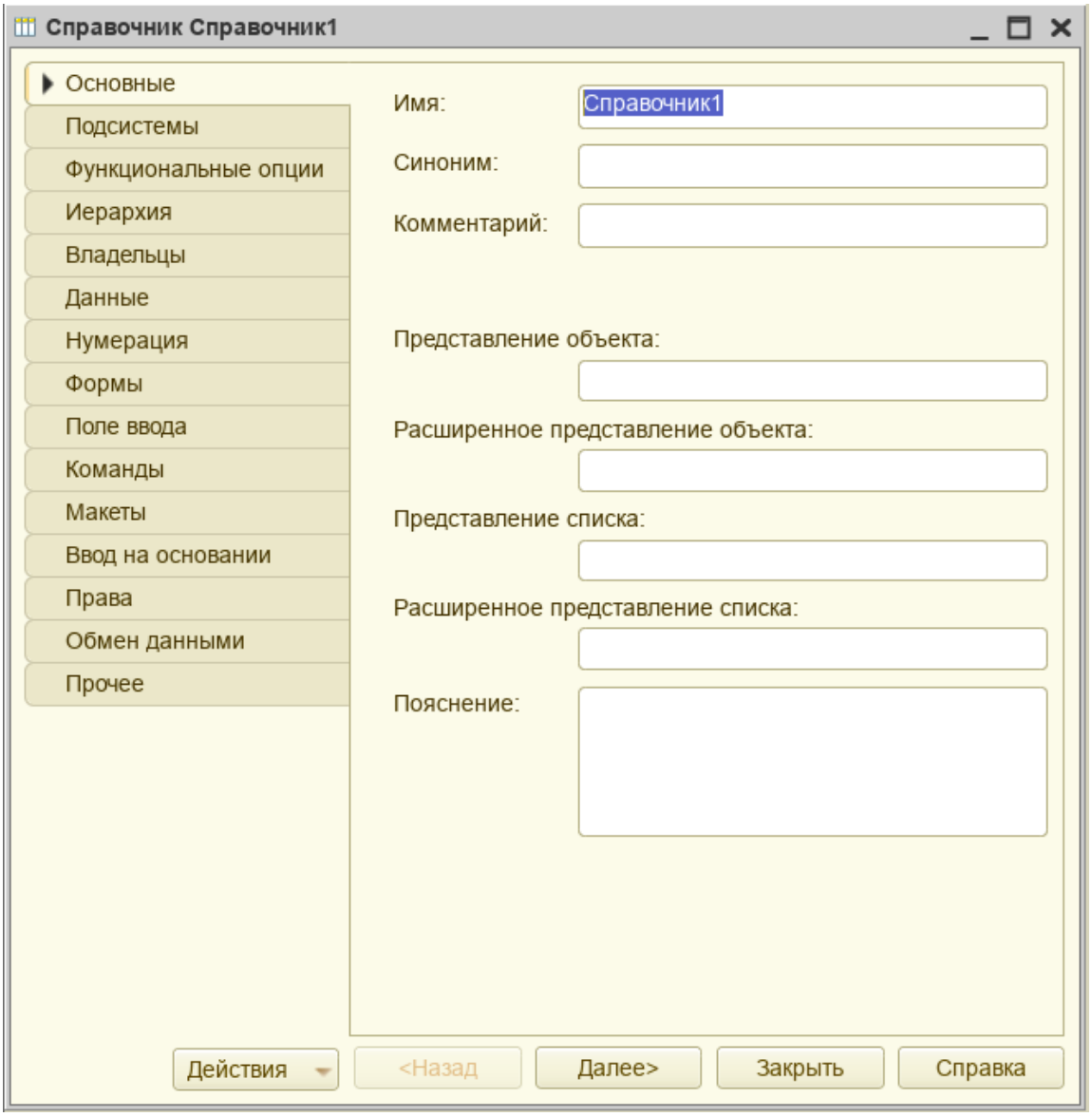


Рис. 4.2. Вкладка «Основные» окна свойств справочника

На рисунке 4.4 представлена вкладка «Владельцы» окна свойств справочника, с помощью которой возможно установить отношение подчиненности между справочниками. У справочника

может быть несколько владельцев, но каждый элемент подчиненного справочника может иметь только одного владельца. Существует возможность определять тип подчинения:

- элементам — в качестве владельца могут выступать только элементы справочника-владельца;
- группам — в качестве владельца могут выступать только группы элементов справочника-владельца;
- группам и элементам — в качестве владельца могут выступать как элементы, так и группы элементов справочника-владельца.

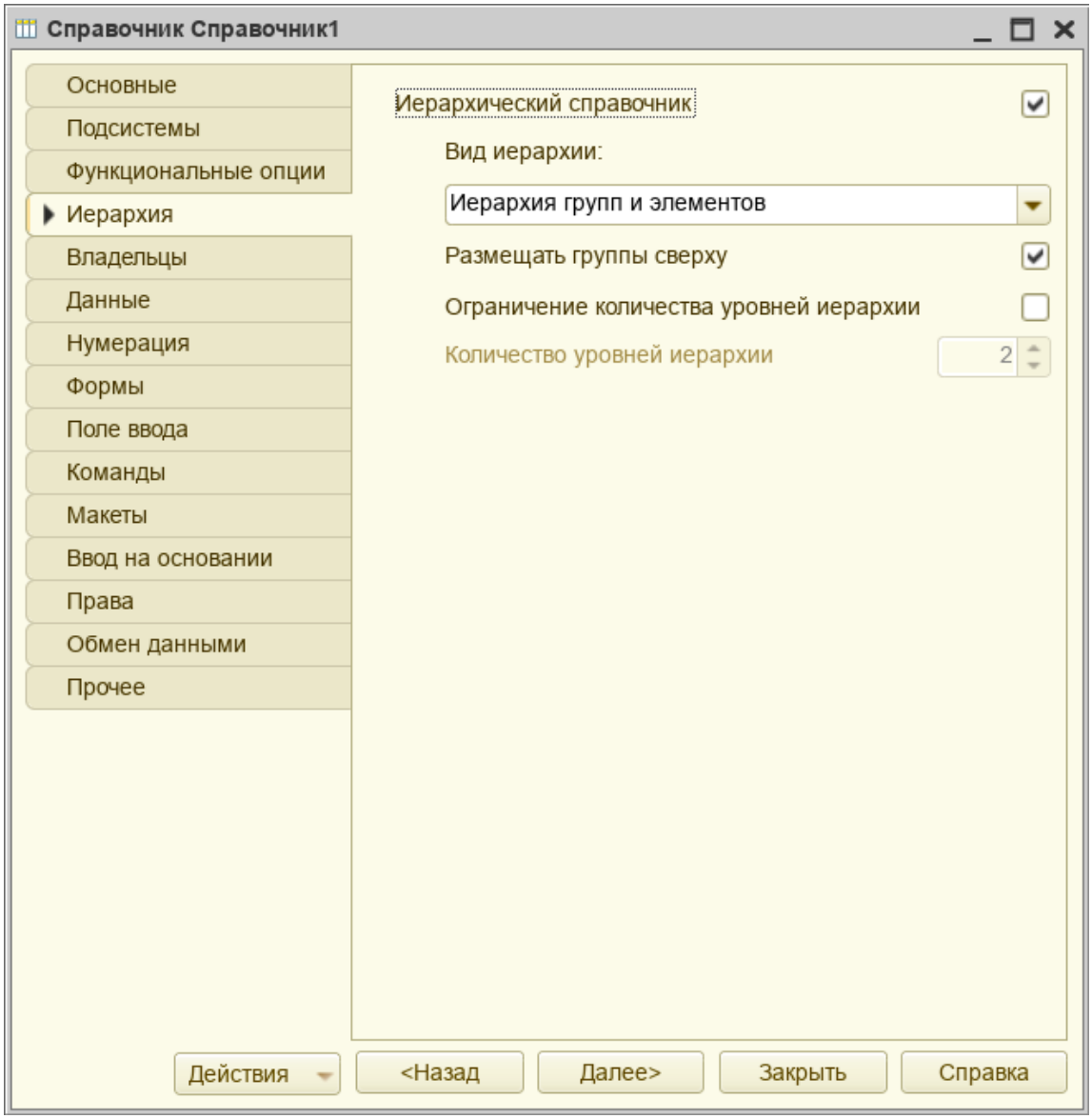


Рис. 4.3. Вкладка «Иерархия» окна свойств справочника

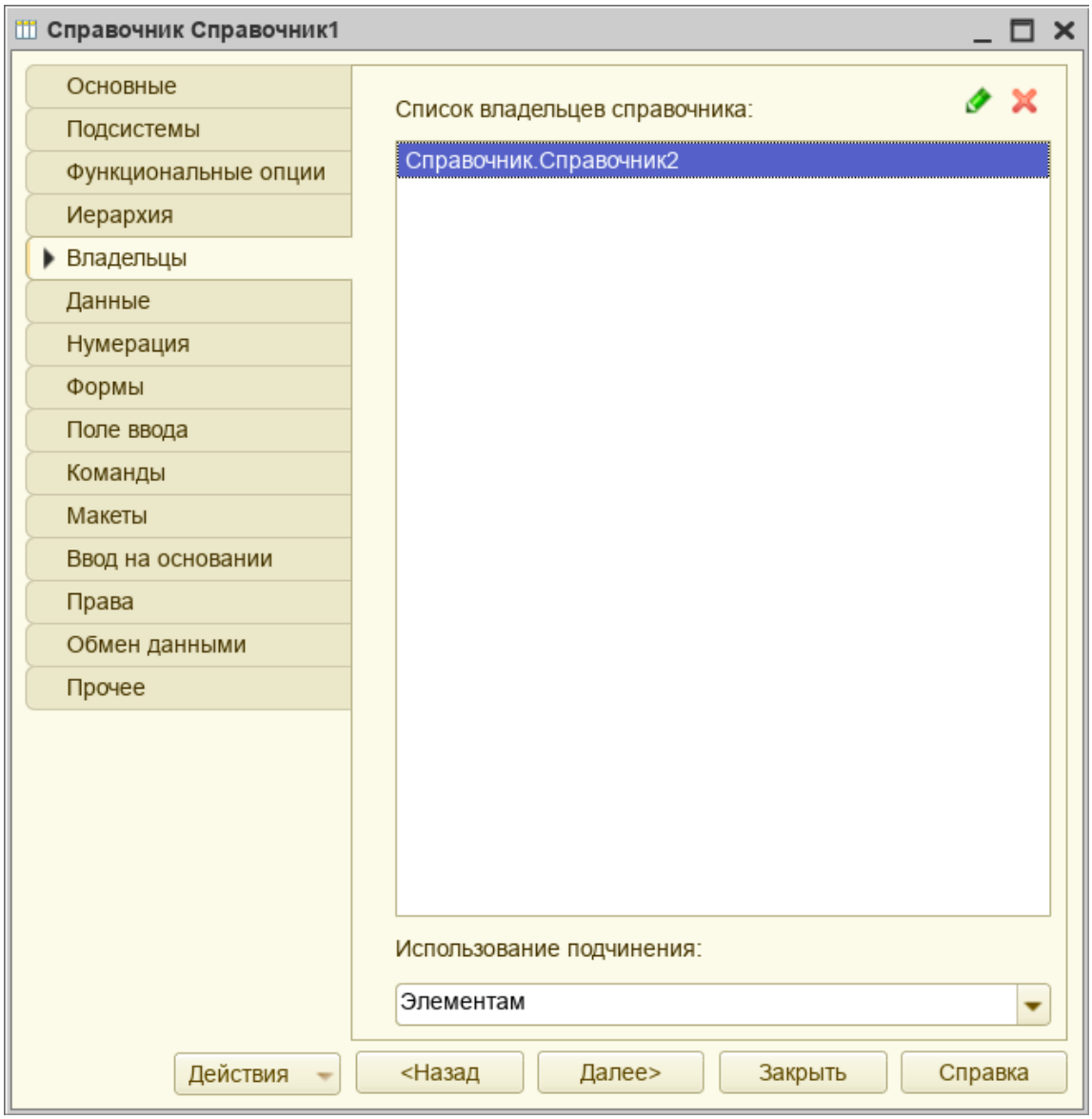


Рис. 4.4. Вкладка «Владельцы» окна свойств справочника

Вкладка «Данные» окна свойств справочника представлена на рисунке 4.5. Как видно из рисунка 4.5, данная вкладка позволяет настроить состав реквизитов и табличные части справочника, длину стандартных реквизитов «Код» и «Наименование», тип данных стандартного реквизита «Код» (число или строка), а также основное представление элемента справочника.

Основное представление элемента справочника позволяет определить, какие сведения необходимо показывать пользователю в интерактивном режиме при использовании в качестве значения элемента управления ссылки на элемент справочника.

Реквизиты справочника позволяют описывать свойства и атрибуты сущностей предметной области, например, у элемента

справочника «Товары» могут быть следующие реквизиты: единица измерения, артикул, стоимость и т. д. Некоторые реквизиты уже предопределены — стандартные реквизиты, а некоторые создаются программистами в процессе разработки решения.

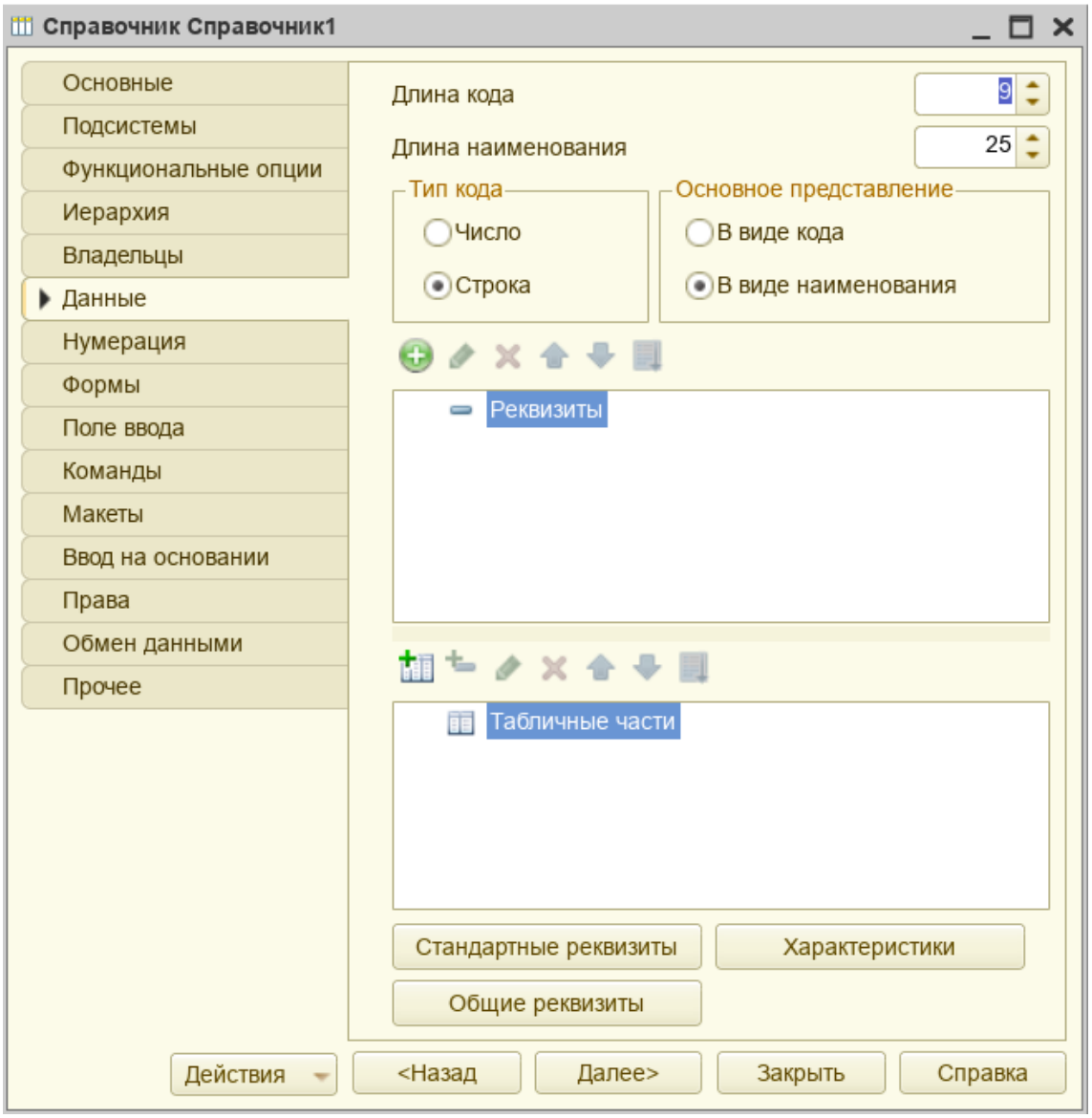


Рис. 4.5. Вкладка «Данные» окна свойств справочника

Табличные части справочника используются для организации связи с типом «один-ко-многим» в рамках одной сущности предметной области. Например, состав семьи сотрудника может храниться в табличной части формы элемента справочника сотрудников. Разработчик должен сделать выбор между формированием подчиненного справочника и разработкой табличной части для хранения подчиненной информации.

Если может возникнуть потребность ссылаться на подчиненные объекты, например, создавать ссылающиеся на них реквизиты документов или других справочников, то лучше завести подчиненный справочник. Если же ссылка на такие сведения не имеет смысла и никогда не может быть типом какого-либо реквизита, тогда можно завести табличную часть. Например, список расчетных счетов контрагента лучше хранить в подчиненном справочнике, т. к. в платежных документах необходимо выбирать расчетный счет из списка (реквизит документа).

Перечень стандартных реквизитов справочника представлен на рисунке 4.6.

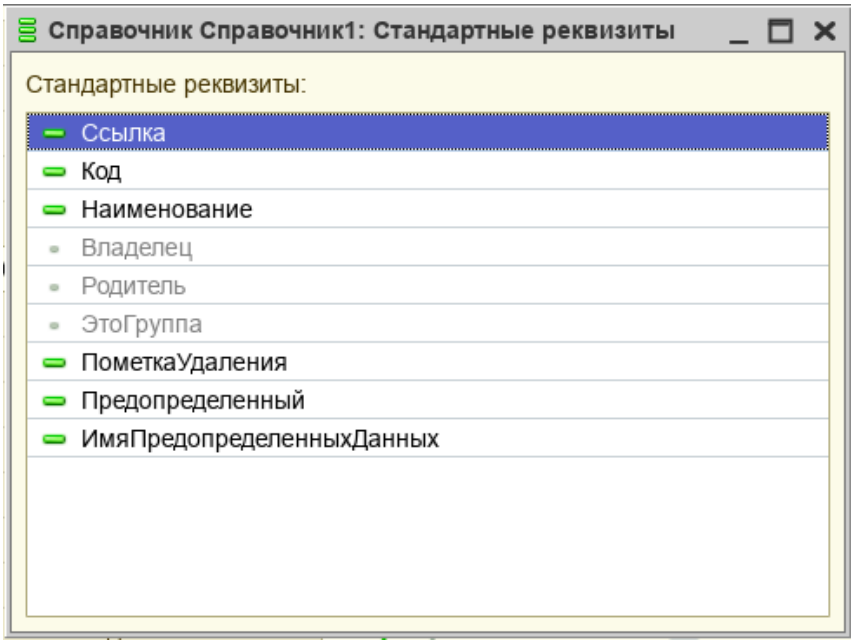


Рис. 4.6. Стандартные реквизиты справочника

Как видно из рисунка 4.6, к стандартным реквизитам справочника относятся:

- Ссылка — уникальный идентификатор элемента справочника, используется для организации связей и доступа к данным элемента;
- Код — обычно используется в рамках предметной области для идентификации элемента справочника со стороны пользователя;
- Наименование — текстовая информация, характеризующая элемент справочника, например, наименование товара или ФИО сотрудника;

- Владелец — ссылка на элемент справочника-владельца, которому подчинен данный элемент. Используется для организации подчиненности. Доступен только при использовании механизма подчиненности;
- Родитель — ссылка на родительский узел. Используется для организации иерархических справочников. Доступен только в иерархических справочниках;
- ЭтоГруппа — флаг, позволяющий определить чем является элемент справочника — группой или элементом. Доступен только в иерархических справочниках с видом иерархии «Иерархия групп и элементов»;
- ПометкаУдаления — флаг, определяющий помечен ли данный элемент справочника для удаления. Используется для фильтрации помеченных на удаление объектов в алгоритмах решения;
- Предопределенный — флаг, определяющий является ли данный элемент справочника предопределенным элементом;
- ИмяПредопределенныхДанных — используется для связи предопределенных элементов с метаданными.

Стандартные реквизиты «Код» и «Наименование» доступны в справочнике только в том случае, если их длина больше 0. Редактировать свойства данных реквизитов можно через диалог стандартных реквизитов (рис. 4.6).

На этапе разработки конфигурации в справочник могут быть введены предопределенные элементы и группы справочника. Эти элементы имеют имена, по которым к ним производится обращение в алгоритмах решения. Для создания предопределенных элементов на вкладке «Прочее» окна свойств справочника необходимо нажать на кнопку «Предопределенные» и ввести необходимые значения в появившемся окне (рис. 4.7).

Рассмотрим пример обращения к предопределенному элементу справочника:

```
Группа = Справочники.Сотрудники.Работающие;
```

4.2 Методы работы со справочниками

Рассмотрим пример программного создания элемента или группы справочника средствами встроенного языка Платформы:

```
// Создание группы справочника
НоваяГруппа = Справочники.Товары.СоздатьГруппу();
НоваяГруппа.Код = "000000001";
НоваяГруппа.Наименование = "Группа товаров";
НоваяГруппа.Записать();

// Создание элемента справочника
НовыйОбъект = Справочники.Товары.СоздатьЭлемент();
НовыйОбъект.Код = "000000002";
НовыйОбъект.Наименование = "Товар1";
НовыйОбъект.Цена = 100.12;
НовыйОбъект.Родитель = НоваяГруппа;
НовыйОбъект.Записать();
```

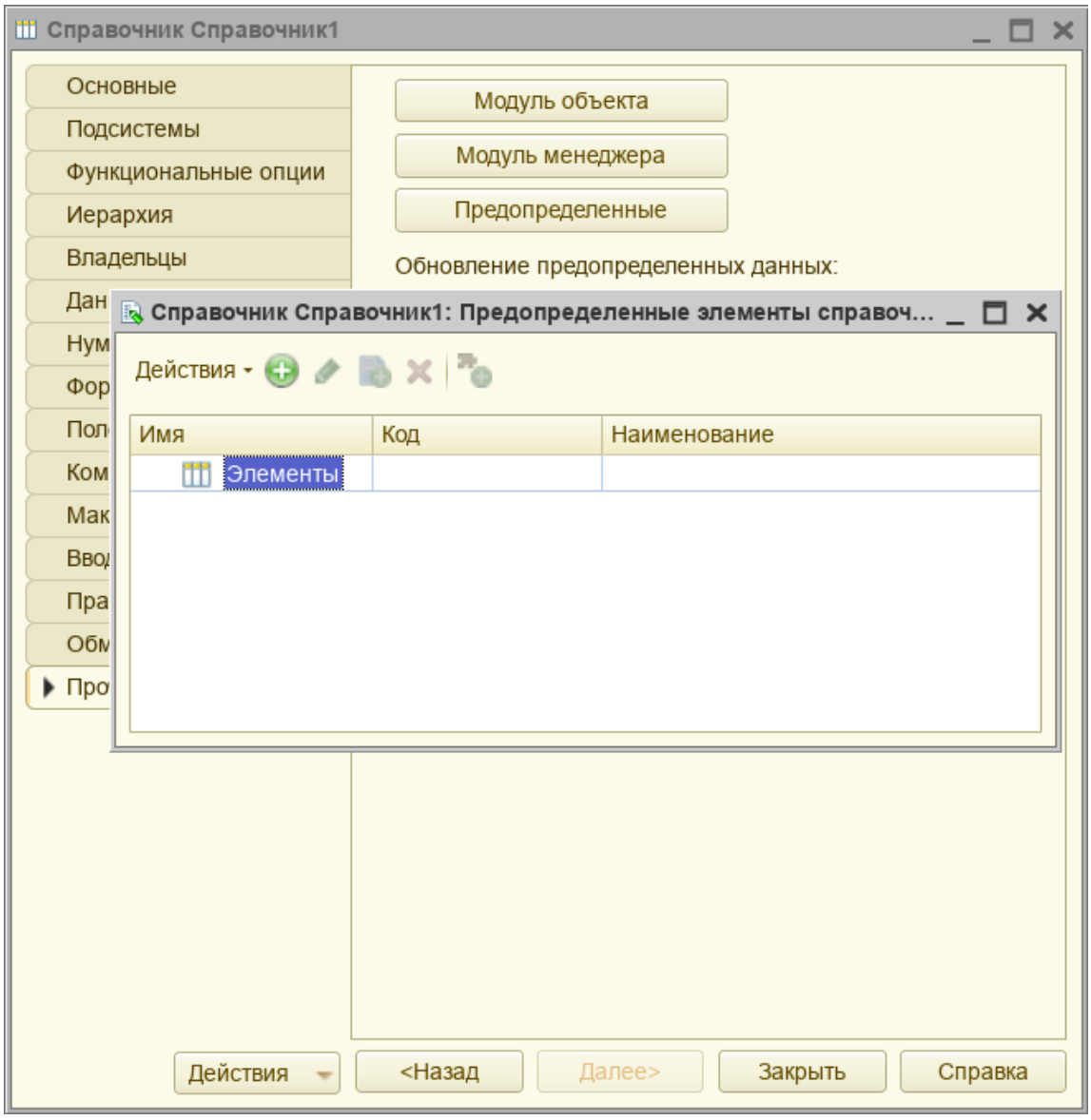


Рис. 4.7. Окно настройки перечня предопределенных элементов справочника

Перебрать элементы определенного справочника, в том числе с отбором, можно следующим образом:

```
// 1-й способ
// Перебор всех элементов
Выборка = Справочники.Товары.Выбрать();
Пока Выборка.Следующий() Цикл
    ...
КонецЦикла;

// 2-й способ
// Перебор всех элементов родителя
// иерархического справочника
Группа = Справочники.Товары.Ходовые;
Выборка = Справочники.Товары.Выбрать(Группа);
Пока Выборка.Следующий() Цикл
    ...
КонецЦикла;

// 3-й способ
// Перебор всех элементов владельца
// при использовании подчиненности
Владелец = Справочники.Товары.Товар1;
Выборка = Справочники.ЦеныНоменклатуры.Выбрать(, Владелец);
Пока Выборка.Следующий() Цикл
    ...
КонецЦикла;

// 4-й способ
// Перебор всех элементов
// с отбором по реквизиту
Отбор = Новый Структура("ТипЦены");
Отбор.ТипЦены = Перечисления.ТипЦены.Закупочная;
Выборка = Справочники.ЦеныНоменклатуры.Выбрать(, , Отбор);
```

Обратиться к реквизитам элемента справочника можно только при наличии ссылки на него:

```
Выборка = Справочники.Товары.Выбрать();
Пока Выборка.Следующий()  Цикл
    Сообщить(Выборка.Наименование);
КонецЦикла;
```

Для обращения к реквизитам табличной части элемента справочника необходимо выполнить перебор строк табличной части:

```
Выборка = Справочники.Товары.Выбрать();
Пока Выборка.Следующий() Цикл
```

```
        // перебор строк табличной части элемента справочника
        Для Каждого Строка Из Выборка.Поставки Цикл
            Сообщить(Строка.ДатаПоставки);
        КонецЦикла
    КонецЦикла;
```

Для поиска элемента справочника по значению реквизитов используются методы НайтиПоКоду, НайтиПоНаименованию и НайтиПоРеквизиту объекта СправочникСсылка:

```
// Поиск элемента справочника по коду
Ссылка = Справочники.Товары.НайтиПоКоду("000000002");
// Проверка результатов поиска на наличие ссылки
Если НЕ Ссылка.Пустая() Тогда
    Сообщить("Элемент найден!");
КонецЕсли;
```

Удалить элемент справочника можно тремя разными способами:

1. Установка пометки на удаление с помощью установки значения соответствующего реквизита.
2. Установка пометки на удаления с помощью вызова метода УстановитьПометкуУдаления объекта СправочникОбъект. Метод УстановитьПометкуУдаления позволяет установить (снять) пометку на удаление для всех подчиненных элементов, если справочник является иерархическим.
3. Непосредственное удаление элемента с помощью вызова метода Удалить объекта СправочникОбъект.

```
Ссылка = Справочники.Товары.НайтиПоКоду("000000002");
Если НЕ Ссылка.Пустая() Тогда
    // Для редактирования элемента необходимо
    // получить объект по ссылке
    Объект = Ссылка.ПолучитьОбъект();

    // 1-й способ
    // Установка пометки на удаление
    Объект.ПометкаУдаления = Истина;
    Объект.Записать();
    // 2-й способ
    // Установка пометки на удаление
    // с помощью метода
    Объект.УстановитьПометкуУдаления(Истина);
    // 3-й способ
    // Непосредственное удаление объекта
    Объект.Удалить();
КонецЕсли;
```

4.3 Пример выполнения задания

В рамках первой лабораторной работы необходимо научиться работать с объектом метаданных «Справочник». Для это требуется выполнить задание, состоящее из следующих пунктов:

1. Необходимо разработать основной справочник, содержащий одну табличную часть и набор реквизитов. Справочник должен быть многоуровневым. Количество уровней — произвольно. Название групп в справочнике — произвольно.
2. Для основного справочника сформировать подчиненный справочник.
3. В случае необходимости разработать дополнительные объекты метаданных: перечисления, справочники.
4. Разработать форму элемента, форму группы и форму списка справочника.
5. В интерактивном режиме ввести несколько групп, содержащих не менее 2-3 элементов.
6. В форму списка основного справочника вставить кнопку «Справка». При нажатии на данную кнопку необходимо вывести в окно сообщений сведения согласно варианту задания. Для вывода необходимо использовать метод глобального контекста `Сообщить`.

Данная лабораторная работа должна быть выполнена в режиме *обычного приложения*.

Рассмотрим процесс выполнения следующего варианта задания:

1. Основной справочник — Товары:
    - код товара;
    - наименование товара;
    - базовая единица измерения (справочник);
    - минимальный остаток;
    - ставка НДС (перечисление);
    - табличная часть:
        - поставщик (справочник);
        - адрес поставщика;
        - закупочная цена;
        - дата последней поставки.
2. Подчиненный справочник — Цены номенклатуры:

- тип цены (перечисление: закупочная, оптовая, розничная);
- цена.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «базовая единица измерения» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Ходовые товары» в основном справочнике.

Получить справку: вывести наименования всех товаров, у которых закупочная цена последней поставки больше средней закупочной цены всех поставок этого товара.

В задании сказано, что необходимо разработать два справочника — «Товары» и «Цены номенклатуры». Также существует необходимость создания дополнительных объектов метаданных:

- справочник «Единицы измерения»,
- справочник «Контрагенты»,
- перечисление «Ставка НДС»,
- перечисление «Тип цены».

После создания перечисленных объектов метаданных конфигурация будет иметь вид, представленный на рисунке 4.8.

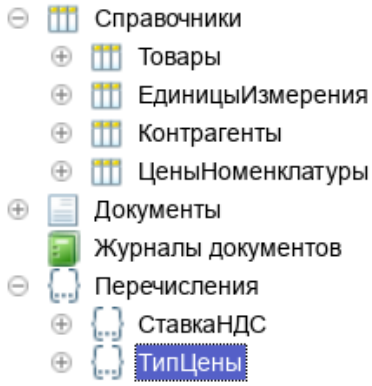


Рис. 4.8. Созданные объекты метаданных

Рассмотрим состав реквизитов вновь созданных объектов метаданных, представленных на рисунке 4.8 более подробно.

Справочник «ЕдиницыИзмерения» содержит следующие реквизиты:

- Код — Строка(9);
- Наименование — Строка(25).

Справочник «Контрагенты» содержит следующие реквизиты:

- Код — Строка(9);
- Наименование — Строка(25);
- Адрес — Строка(200).

Справочник «ЦеныНоменклатуры» содержит следующие реквизиты:

- ТипЦены — ПеречислениеСсылка.ТипЦены;
- ЗначениеЦены — Число(12, 2, Неотрицательное).

Реквизиты «Код» и «Наименование» в справочнике «ЦеныНоменклатуры» отключены путем установки значения их длины равным нулю.

Справочник «ЦеныНоменклатуры» является подчиненным справочнику «Товары» (рис. 4.9).

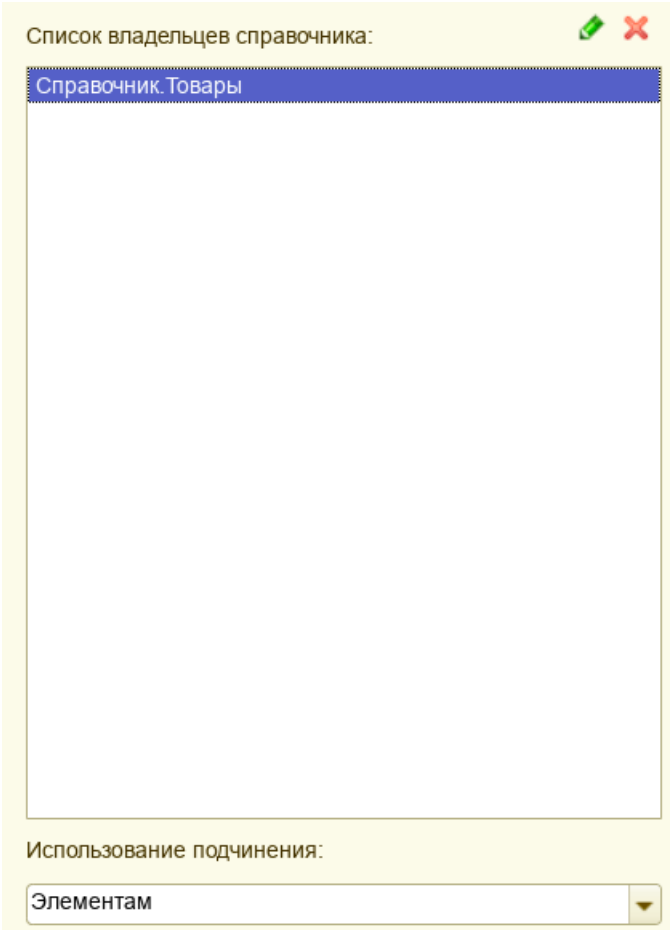


Рис. 4.9. Окно свойств справочника «ЦеныНоменклатуры»

Перечисление «СтавкаНДС» содержит следующие значения:

- НДС0 — 0%;
- НДС10 — 10%;

- НДС20 — 20%.

Перечисление «ТипЦены» содержит следующие значения:

- Оптовая;
- Закупочная;
- Розничная.

Справочник «Товары» содержит следующие реквизиты:

- Код — Строка(9);
- Наименование — Строка(50);
- ЕдиницаИзмерения — СправочникСсылка.Единицы-Измерения;
- МинимальныйОстаток — Число(10, 0, Неотрицательное);
- СтавкаНДС — ПеречислениеСсылка.СтавкаНДС.

Также справочник «Товары» содержит табличную часть «Поставки», состоящую из следующих реквизитов:

- Поставщик — СправочникСсылка.Контрагенты;
- Адрес — Строка(200);
- ЗакупочнаяЦена — Число(12, 2, Неотрицательное);
- ДатаПоставки — Дата(Дата).

Справочник «Товары» является иерархическим, настройки иерархического справочника представлены на рисунке 4.10.

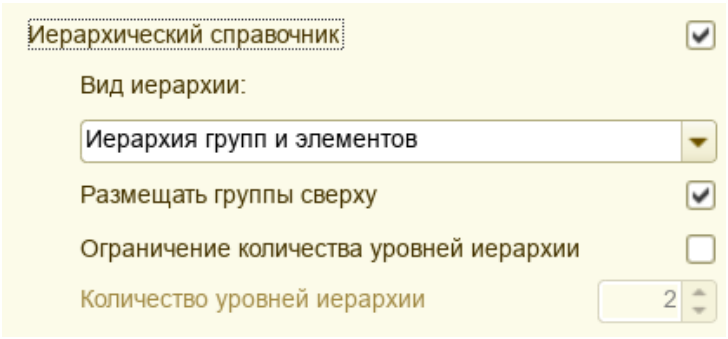


Рис. 4.10. Настройки иерархического справочника «Товары»

Для автоматического заполнения реквизита «ЕдиницаИзмерения» справочника «Товары» предопределенным значением справочника «ЕдиницыИзмерения» необходимо добавить в справочник «ЕдиницыИзмерения» предопределенный элемент «Штука». Для этого в окне свойств справочника «ЕдиницыИзмерения» необходимо перейти на вкладку «Прочее», нажать на кнопку «Предопределенные» и добавить предопределенный элемент в появившемся окне (рис. 4.11).

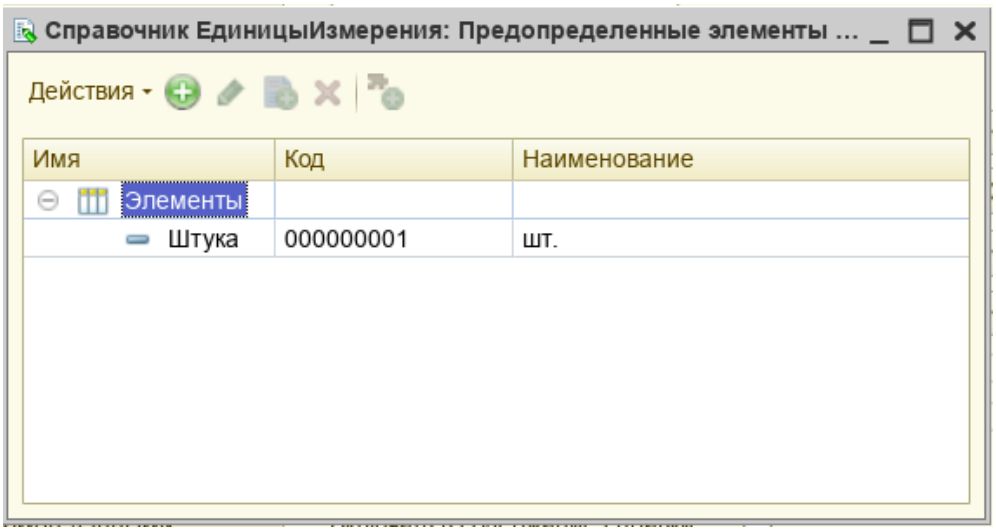


Рис. 4.11. Создание предопределенного элемента справочника «ЕдиницыИзмерения»

Далее в свойствах реквизита «ЕдиницаИзмерения» справочника «Товары» необходимо выбрать в качестве параметра «Значение заполнения» предопределенный элемент «Штука» справочника «ЕдиницыИзмерения» (рис. 4.12).

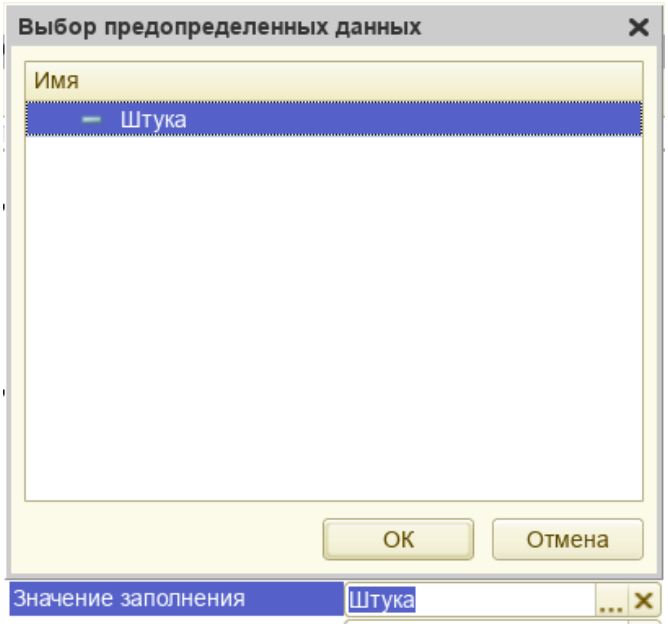


Рис. 4.12. Выбор значения для заполнения

Следующим шагом необходимо добавить предопределенную группу «Ходовые товары» в справочник «Товары». Для этого необходимо перейти на вкладку «Прочее» окна свойств справочника,

нажать на кнопку «Предопределенные» и добавить предопределенную группу в появившемся окне (рис. 4.13).

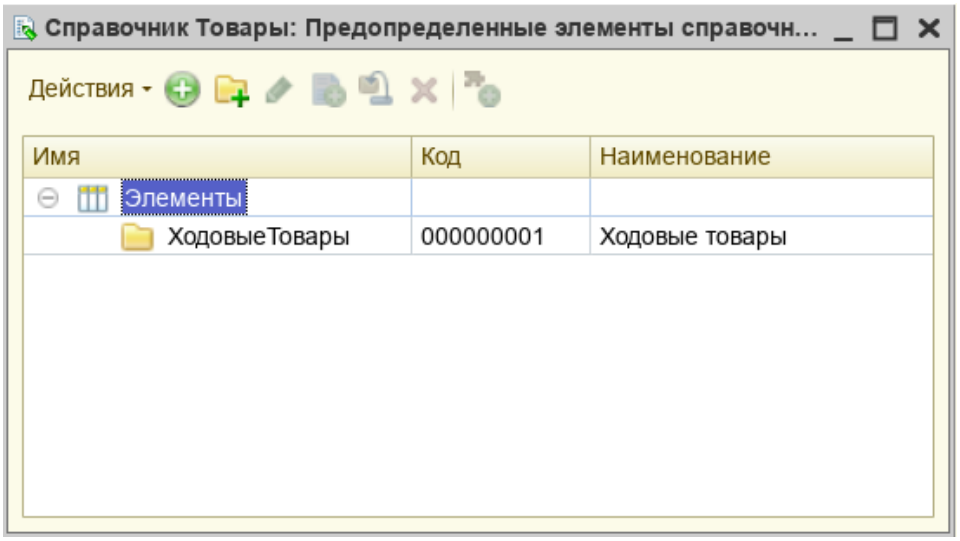


Рис. 4.13. Создание предопределенной группы справочника «Товары»

Разработаем формы основного справочника «Товары». Для управления формами справочника необходимо перейти на вкладку «Формы» окна свойств справочника и нажать на кнопку «Добавить» в панели команд. Для добавления форм справочника используется конструктор форм, представленный на рисунке 4.14. Как видно из рисунка 4.14, форма справочника может иметь несколько типов:

- форма элемента — используется для создания и редактирования элементов справочника;
- форма группы — используется для создания и редактирования групп справочника;
- форма списка — используется для отображения содержимого всего справочника;
- форма выбора — используется для выбора элементов справочника для поля ввода с типом данных — ссылкой на данный справочник;
- форма выбора группы — используется для выбора групп справочника для поля ввода с типом данных — ссылкой на данный справочник;
- произвольная форма.

После выбора типа формы справочника, например «Форма элемента справочника», необходимо нажать на кнопку «Далее».

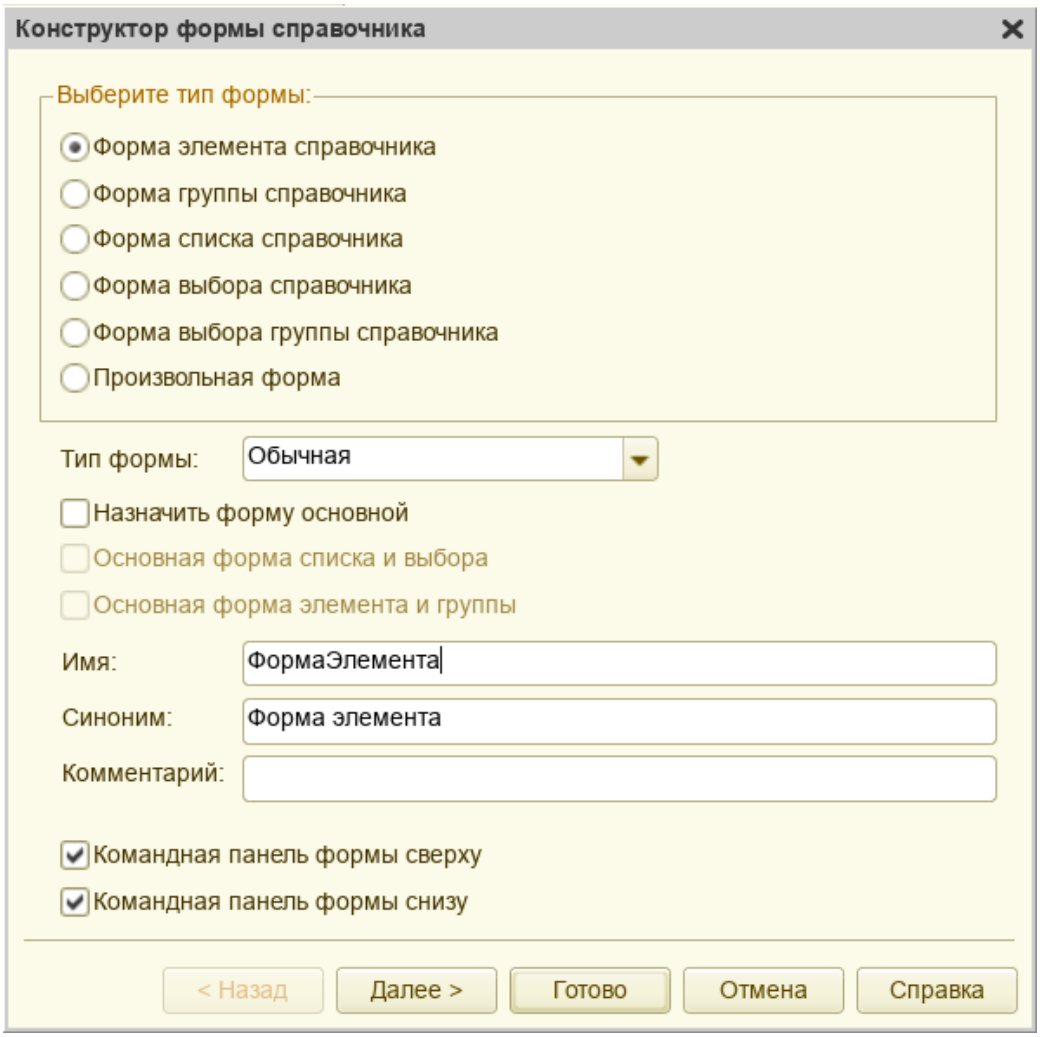


Рис. 4.14. Создание формы справочника, шаг 1

На следующем шаге работы конструктора формы существует возможность определить какие реквизиты будут отображаться на форме (рис. 4.15), например, можно убрать с формы поле ввода для реквизита «Родитель». Также на данном шаге возможно изменить тип элементов управления для реквизитов. После окончания настройки состава элементов формы необходимо нажать на кнопку «Готово». Созданная форма будет автоматически установлена в качестве формы по умолчанию соответствующего типа, если такой формы еще не существует.

Если в процессе разработки решения не было создано ни одной формы, или отсутствует форма определенного типа, в таком случае форма генерируется автоматически на основе состава реквизитов справочника с настройками по умолчанию. Также существует возможность создания нескольких форм одного типа,

выбор формы для отображения в таком случае может осуществляться программно.

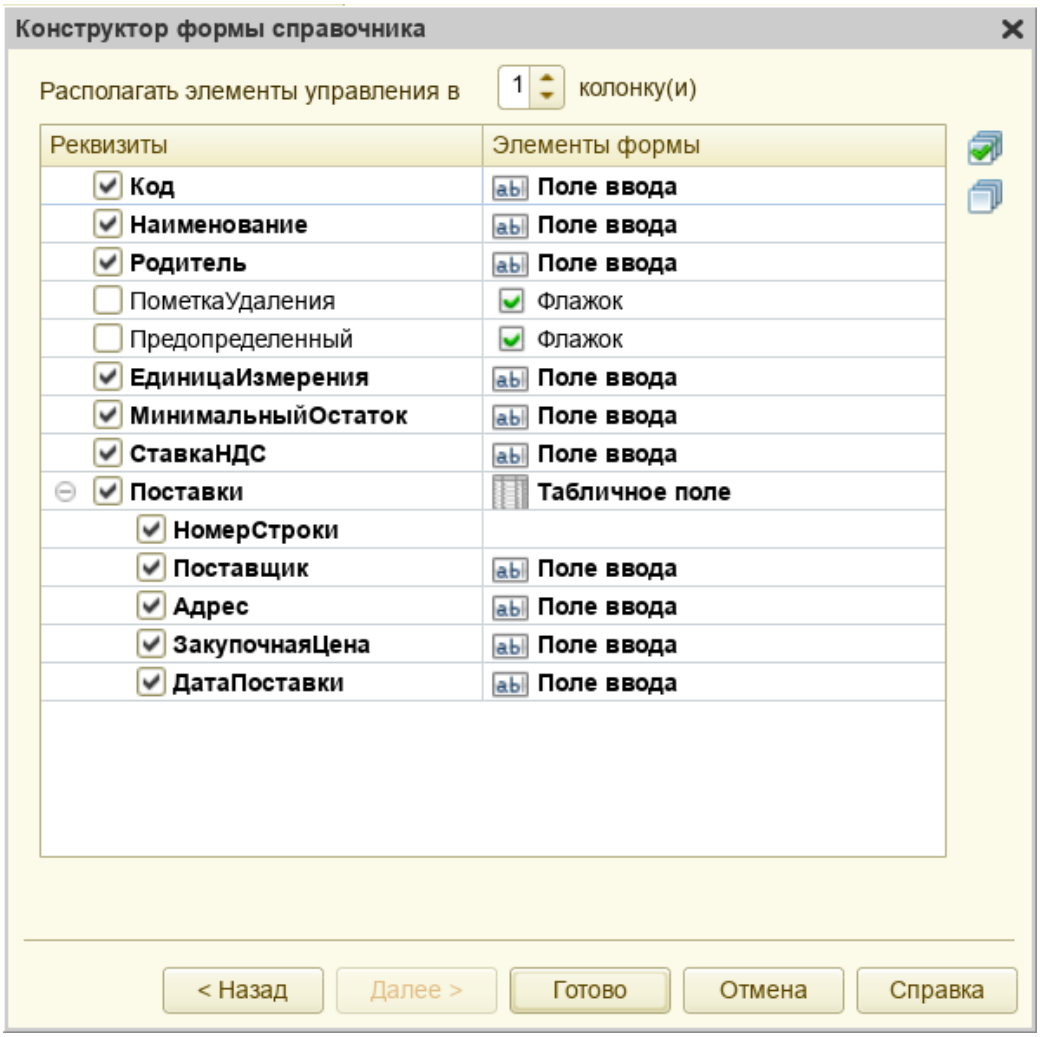


Рис. 4.15. Создание формы справочника, шаг 2

Таким образом, для справочника «Товары» были созданы следующие формы:

1. Форма элемента, созданная с использованием настроек по умолчанию и отсутствующим элементом формы для реквизита «Родитель».
2. Форма группы, созданная с использованием настроек по умолчанию и отсутствующим элементом формы для реквизита «Родитель».
3. Форма списка, созданная с использованием настроек по умолчанию.

Рассмотрим пример изменения поведения формы с помощью описания алгоритма работы обработчиков событий элементов

формы. Например, для автоматического заполнения реквизита «Адрес» при изменении значения реквизита «Поставщик» табличной части «Поставки» справочника «Товары» на основе значения реквизита «Адрес» справочника «Контрагенты» необходимо:

1. Открыть форму элемента справочника «Товары».
2. Для поля ввода реквизита «Поставщик» создать обработчик события «При изменении» (рис. 4.16). Для автоматического создания обработчика необходимо нажать на кнопку с изображением увеличительного стекла.

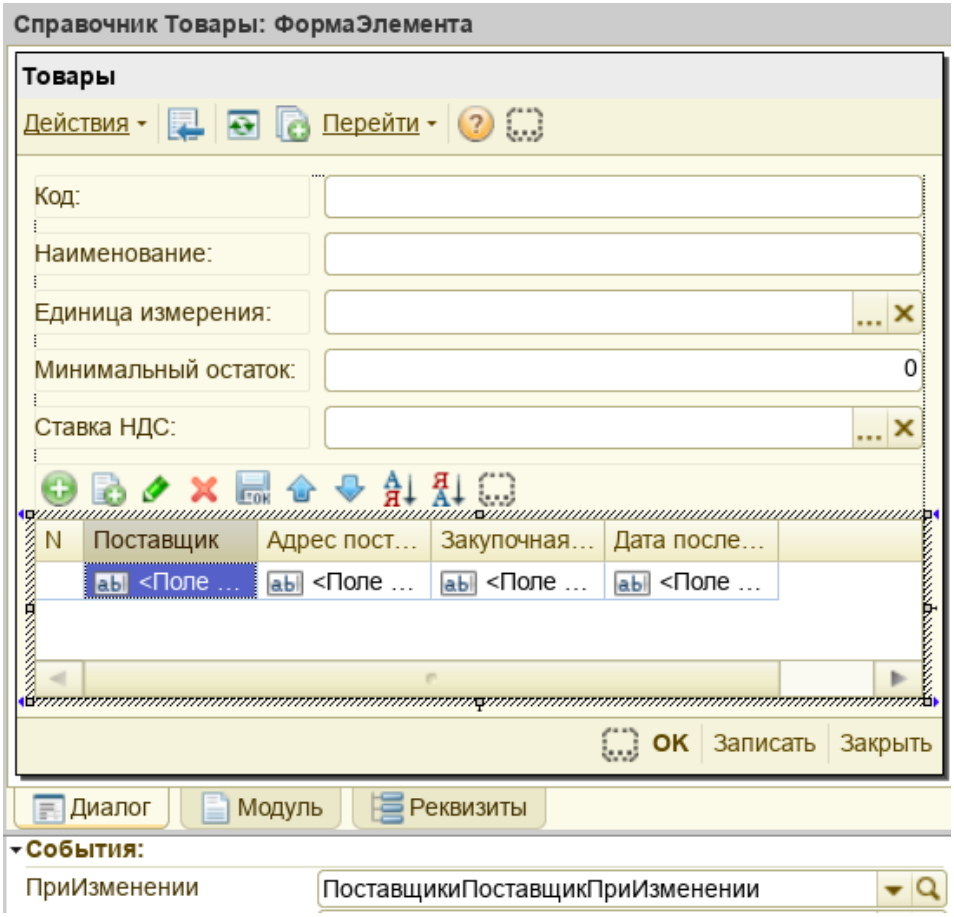


Рис. 4.16. Создание обработчика «При изменении» для поля ввода реквизита «Поставщик» табличной части «Поставки»

Обработчик события «При изменении» должен иметь вид:

```
Процедура ПоставщикиПоставщикПриИзменении(Элемент)
    ТС = ЭлементыФормы.Поставщики.ТекущаяСтрока;
    ТС.Адрес = ТС.Поставщик.Адрес;
КонецПроцедуры
```

Переменная ТС содержит ссылку на строку табличной части «Поставки», с которой в данный момент взаимодействует пользователь.

Для запрета редактирования значения реквизита «Адрес» формы элемента справочника «Товары» необходимо в свойствах элемента управления «Адрес» установить значение параметра «Только просмотр» в Истина установкой соответствующего флага.

Для добавления кнопки для вывода справки в форму списка справочника «Товары» необходимо:

1. Открыть форму списка справочника «Товары».
2. Перейти к концу командной панели «ДействияФормы» с помощью специальной кнопки (рис. 4.17, выделена красным).



Рис. 4.17. Кнопка для перехода к правой части командной панели

3. Выбрать пустой элемент панели, обведенный точками (рис. 4.18), и установить для него следующие свойства:
   - Имя — Справка;
   - Текст — Справка;
   - Картинка — Отчет;
   - Отображение — Надпись и картинка.



Рис. 4.18. Кнопка «Справка»

Следующим шагом необходимо создать обработчик нажатия на кнопку «Справка» следующего вида:

```
Процедура ДействияФормыСправка(Кнопка)
    Спр = Справочники.Товары.Выбрать();

    Пока Спр.Следующий() Цикл

        // Группы справочника пропускаются
        Если Спр.ЭтоГруппа Тогда
            Продолжить;
        КонецЕсли;
```

```
        // Если табличная часть не заполнена,
        // то такой элемент пропускается
        Если Спр.Поставки.Количество() = 0 Тогда
            Продолжить;
        КонецЕсли;

        // Выгрузка табличной части в таблицу значений
        Таблица = Спр.Поставки.Выгрузить();
        // Сортировка таб. значений по убыванию даты поставки
        // Самая последня поставка будет иметь индекс 0
        Таблица.Сортировать("ДатаПоставки Убыв");
        // Получение закупочной цены самой последней поставке
        ЦенаПоследнейПоставки = Таблица.Получить(0)
            .ЗакупочнаяЦена;
        // Расчет средней закупочной цены всех поставок
        СредняяЦена = Таблица.Итог("ЗакупочнаяЦена")
            / Таблица.Количество();
        // Вывод наименования товара, у которого
        // закупочная цена последней поставки больше
        // средней закупочной цены всех поставок этого товара
        Если ЦенаПоследнейПоставки > СредняяЦена Тогда
            Сообщить(Спр.Наименование);
        КонецЕсли;
    КонецЦикла;
КонецПроцедуры
```

Рассмотрим пример выполнения данного алгоритма. Для тестирования метода формирования справки согласно заданию в режиме пользователя было добавлено 2 товара:

- Товар1. Содержимое табличной части «Поставки» следующее:
    1. Закупочная цена — 100, Дата поставки — 01.01.2000
    2. Закупочная цена — 110, Дата поставки — 01.02.2000
    3. Закупочная цена — 150, Дата поставки — 01.03.2000
- Товар2. Содержимое табличной части «Поставки» следующее:
    1. Закупочная цена — 200, Дата поставки — 01.01.2000
    2. Закупочная цена — 220, Дата поставки — 01.02.2000
    3. Закупочная цена — 210, Дата поставки — 01.03.2000

Таким образом, после нажатия на кнопку «Справка» получен результат, представленный на рисунке 4.19.



Рис. 4.19. Результат формирования справки

4.4 Контрольные вопросы

1. Для чего используются предопределенные элементы и группы справочника?
2. Каким образом можно создать предопределенные элементы и группы справочника?
3. Для чего используются иерархические справочники?
4. Какие типы иерархических справочников бывают и в чем разница между ними?
5. Для чего используются табличные части справочников?
6. Для чего используются подчиненные справочники?
7. Какие типы подчиненных справочников бывают и в чем разница между ними?
8. В каких случаях лучше использовать табличную часть, а в каких подчиненные справочники?

4.5 Варианты заданий

Лабораторная работа должна быть выполнена в режиме *обычного приложения*.

В представленных вариантах задания необходимо:

1. Разработать основной справочник, содержащий одну табличную часть и набор реквизитов (см. варианты заданий). Справочник должен быть многоуровневым. Количество уровней — произвольно. Название групп в справочнике — произвольно.
2. Для основного справочника сформировать подчиненный справочник (см. варианты заданий).
3. В случае необходимости разработать дополнительные объекты метаданных: перечисления, справочники.
4. Разработать форму элемента, форму группы и форму списка справочника.
5. В интерактивном режиме ввести несколько групп, содержащих не менее 2-3 элементов.
6. В форму списка основного справочника вставить кнопку «Справка». При нажатии на данную кнопку необходимо вы-

вести в окно сообщений сведения согласно варианту задания. Для вывода необходимо использовать метод глобального контекста `Сообщить`.

Вариант 1

1. Основной справочник — Основные средства (ОС):
    - инвентарный номер ОС;
    - наименование товара ОС;
    - основное материально-ответственное лицо (справочник «Сотрудники»);
    - сумма первоначального износа;
    - дата ввода в эксплуатацию;
    - год выпуска;
    - срок эксплуатации (в мес.);
    - табличная часть (список подразделений, которым принадлежало ОС в течение времени эксплуатации):
        - подразделение (справочник «Подразделения»),
        - дата появления в подразделении.
2. Подчиненный справочник — Материально-ответственные лица:
    - материально-ответственное лицо (справочник «Сотрудники»);
    - домашний адрес.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «основное материально-ответственное лицо» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «ОС на консервации» в основном справочнике.

Получить справку: вывести наименование ОС, дата ввода в эксплуатацию которых позднее 1980 года.

Вариант 2

1. Основной справочник — Товары:
    - код товара;
    - наименование товара;
    - основная единица измерения (справочник «Единицы измерения»);

- тип товара (перечисление «Тип товара»: весовой, невесовой);
- табличная часть (список возможных единиц измерения каждого товара):
  - единица измерения (справочник «Единицы измерения»)
  - коэффициент пересчета относительно основной единицы измерения.

2. Подчиненный справочник — Цены товаров:
   - тип цены (перечисление: закупочная, оптовая, розничная);
   - цена.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «основная единица измерения» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Неликвидные товары» в основном справочнике.

Получить справку: вывести наименование всех товаров, у которых розничная цена меньше 1.5 от закупочной цены.

Вариант 3

1. Основной справочник — Сотрудники:
   - табельный номер сотрудника;
   - ФИО сотрудника;
   - тип сотрудника (перечисление «ТипСотрудника»: штатный, внутренний совместитель, внешний совместитель);
   - дата приема на работу;
   - оклад;
   - является членом профсоюза;
   - табличная часть (список детей сотрудника):
     - ФИО ребенка;
     - пол (перечисление «Пол»);
     - дата рождения.
2. Подчиненный справочник — Стандартные вычеты сотрудника:
   - тип вычета (перечисление «ТипВычета»: собственный, на ребенка, герой россии);

- сумма вычета;
- дата начала действия;
- дата окончания действия.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «тип сотрудника» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Уволенные» в основном справочнике.

Получить справку: вывести наименование всех сотрудников, которые не являются членами профсоюза.

Вариант 4

1. Основной справочник — Контрагенты:
   - код контрагента;
   - наименование контрагента;
   - адрес контрагента;
   - основной расчетный счет (справочник «Расчетные счета»);
   - банк (справочник «Банки»);
   - телефон;
   - размер кредита;
   - табличная часть (список договоров контрагента):
     - код договора;
     - наименование договора;
     - дата заключения договора;
     - условия договора.
2. Подчиненный справочник — Расчетные счета:
   - код;
   - наименование;
   - банк (справочник «Банки»).

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «размер кредита» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Зарубежные контрагенты» в основном справочнике.

Получить справку: вывести наименование всех контрагентов с нулевым значением кредита.

Вариант 5

1. Основной справочник — Транспортные средства:
    - тип транспортного средства (перечисление «ТипТранспортногоСредства»: легковой, грузовой, специальный);
    - наименование транспортного средства;
    - объем двигателя;
    - мощность двигателя;
    - дата выпуска;
    - водитель (справочник «Сотрудники»);
    - табличная часть (график прохождения технического осмотра транспортного средства):
        - номер по порядку;
        - дата прохождения;
        - примечание.
2. Подчиненный справочник — Товарно-транспортные накладные:
    - номер накладной;
    - дата выписки накладной;
    - время пребывания в пути;
    - стоимость израсходованного горючего.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «тип транспортного средства» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Транспортные средства по лизингу» в основном справочнике.

Получить справку: вывести наименование всех транспортных средств с объемом двигателя меньше 40 л.с.

Вариант 6

1. Основной справочник — Регистрационные карточки больного:
    - код больного;
    - ФИО больного;
    - место работы;
    - дата рождения;
    - пол (перечисление «Пол»);
    - табличная часть (история болезни больного):
        - болезнь (справочник «Болезни»);

  - дата начала заболевания;
  - дата выздоровления.

2. Подчиненный справочник — Прививки больного:
   - болезнь (справочник «Болезни»);
   - дата прививки.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «пол» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Амбулаторные» в основном справочнике.

Получить справку: вывести наименование всех больных мужского пола.

Вариант 7

1. Основной справочник — Фракции в государственной думе:
   - код фракции;
   - наименование фракции;
   - лидер фракции (справочник «Физические лица»);
   - табличная часть (список депутатов фракции):
     - депутат (справочник «Физические лица»);
     - год рождения;
     - регион (справочник «Регионы»);
     - оклад.
2. Подчиненный справочник — Комитеты депутатов фракции:
   - комитет (справочник «Комитеты»);
   - депутат (справочник «Физические лица»);
   - дата вхождения в комитет.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «лидер фракции» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Малочисленные фракции» в основном справочнике.

Получить справку: вывести наименование всех фракций, содержащих менее 10 депутатов.

Вариант 8

1. Основной справочник — Персональные компьютеры предприятия:

- процессор (справочник «Процессоры»);
- частота процессора;
- объем оперативной памяти;
- наличие звуковой карты;
- размер диагонали монитора;
- табличная часть (список цен компьютера):
  - тип цены (перечисление «Тип цены»: оптовая, розничная, закупочная);
  - сумма;
  - валюта (справочник «Валюта»).

2. Подчиненный справочник — Сотрудники, работающие на данном компьютере:
   - сотрудник (справочник «Сотрудники»);
   - время начала работы;
   - время окончания работы.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «наличие звуковой карты» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Списанные компьютеры» в основном справочнике.

Получить справку: вывести наименование всех компьютеров, у которых розничная цена больше 60 000 руб.

Вариант 9

1. Основной справочник — Библиографические единицы:
   - тип библиографической единицы (перечисление «тип библиографической единицы»: книга, журнал, пособие, электронный);
   - автор (справочник «Физические лица»);
   - наименование библиографической единицы;
   - год издания;
   - издательство (справочник «Издательства»);
   - тираж;
   - табличная часть (список библиографических ссылок на библиографическую единицу):
     - автор ссылки (справочник «Физические лица»);
     - наименование работы автора ссылки;
     - год издания работы.

2. Подчиненный справочник — Сотрудники, бравшие данную библиографическую единицу с абонемента:
    - сотрудник (справочник «Сотрудники»);
    - дата получения;
    - дата возврата.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «тип библиографической единицы» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Списанные» в основном справочнике.

Получить справку: вывести наименование всех библиографических единиц с типом книга.

Вариант 10

1. Основной справочник — Факультеты университета:
    - наименование факультета;
    - декан (справочник «Сотрудники»);
    - количество студентов на факультете;
    - год основания;
    - табличная часть (список кафедр факультета):
        - наименование кафедры;
        - зав. кафедрой (справочник «Сотрудники»);
        - количество штатных единиц.
2. Подчиненный справочник — Заместители декана факультета:
    - вид деятельности (перечисление «ВидДеятельности: учебная, научная, организационная);
    - заместитель декана (справочник «Сотрудники»);
    - дата вступления в должность.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «год основания» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Новые факультеты» в основном справочнике.

Получить справку: вывести наименование всех факультетов, у которых число штатных единиц больше 10.

Вариант 11

1. Основной справочник — Кафедры университета:
    - наименование кафедры;
    - заведующий кафедрой (справочник «Сотрудники»);
    - количество студентов на кафедре;
    - год основания;
    - табличная часть (список дисциплин, изучаемых на кафедре):
        - дисциплина (справочник «Дисциплины»);
        - преподаватель (справочник «Сотрудники»);
        - количество часов;
        - наличие экзамена;
        - наличие курсовой работы.
2. Подчиненный справочник — Лаборатории кафедры:
    - номер аудитории;
    - занимаемая площадь;
    - число компьютеров.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «год основания» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Новые кафедры» в основном справочнике.

Получить справку: вывести наименование всех кафедр, у которых есть дисциплина с наименованием «Базы данных».

Вариант 12

1. Основной справочник — Трудовые книжки:
    - код сотрудника;
    - ФИО сотрудника;
    - дата рождения;
    - образование (перечисление «ВидОбразования»: высшее, неполное высшее и т. д.);
    - основное место работы (справочник «Подразделения»);
    - табличная часть (список мест работы сотрудника):
        - дополнительное место работы (справочник «Подразделения»);
        - дата приема;
        - дата увольнения;

  - оклад.
2. Подчиненный справочник — Дети сотрудника:
   - имя ребенка;
   - пол ребенка (перечисление «Пол»);
   - дата рождения.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «образование» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Ветераны» в основном справочнике.

Получить справку: вывести наименование всех сотрудников, имеющих высшее образование.

Вариант 13

1. Основной справочник — Виды оплат и удержаний:
   - код вида оплаты/удержания;
   - наименование вида оплаты/удержания;
   - тип вида оплаты/удержания (перечисление «ТипВидаОплатыУдержания»: разовый, периодический);
   - счет (корреспондирующий для счета 70) (справочник «БухгалтерскиеСчета»);
   - табличная часть (список правил удержания налогов и взносов):
     - налог/взнос (перечисление «НалогиВзносы»: НДФЛ, профсоюзный взнос, алименты);
     - признак удержания налога/взноса.
2. Подчиненный справочник — Сотрудники, которым назначен данный вид расчета:
   - сотрудник (справочник «Сотрудники»);
   - дата начала действия;
   - дата окончания действия.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «счет» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Основные начисления» в основном справочнике.

Получить справку: вывести наименование всех периодических оплат, с которых не удерживается НДФЛ.

Вариант 14

1. Основной справочник — Автомобили:
    - инвентарный номер;
    - наименование;
    - марка (перечисление «МаркаАвтомобиля»);
    - грузоподъемность;
    - табличная часть (список водителей, работающих на данном автомобиле):
        - водитель (справочник «Сотрудники»);
        - дата начала работы;
        - дата окончания работы.
2. Подчиненный справочник — Ремонт и техническое обслуживание:
    - специалист (справочник «Сотрудники»);
    - тип работы (перечисление «ТипРаботы»: ремонт, обслуживание);
    - дата проведения;
    - описание.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «грузоподъемность» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «На ремонте» в основном справочнике.

Получить справку: вывести наименование всех автомобилей, имеющих только одного водителя.

Вариант 15

1. Основной справочник — Сотрудники:
    - табельный номер сотрудника;
    - ФИО сотрудника;
    - тип сотрудника (перечисление «ТипСотрудника»: штатный, внутренний совместитель, внешний совместитель);
    - дата приема на работу;
    - оклад;
    - является членом профсоюза;
    - табличная часть (список долговременных доплат и удержаний сотрудника):

  - вид расчета (справочник «ВидыРасчетов»);
  - дата начала действия;
  - дата окончания действия;
  - величина (сумма, если расчет вводится фиксированной суммой, процент, если результат расчета формируется как процент от расчетной базы);
2. Подчиненный справочник — Трудовая книжка:
   - место работы (справочник «Подразделения»);
   - должность (справочник «Должности»);
   - дата приема на работу;
   - дата увольнения.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «оклад» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Пенсионеры» в основном справочнике.

Получить справку: вывести ФИО всех сотрудников, имеющих более двух долговременных доплат.

Вариант 16

1. Основной справочник — Виды работ:
   - код вида работы;
   - название вида работ;
   - объект, для которого предназначен данный вид работы (справочник «Объекты»);
   - табличная часть (список норм и расценок для каждого вида работ):
     - единица измерения (справочник «ЕдиницыИзмерения»);
     - норма;
     - расценка.
2. Подчиненный справочник — Сотрудники, выполняющие данный вид работ:
   - сотрудник (справочник «Сотрудники»);
   - дата начала выполнения работы;
   - дата окончания работы.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «название вида работ» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Земляные работы» в основном справочнике.

Получить справку: вывести наименование всех видов работ, которые содержат хотя бы одну расценку более 1000 руб.

Вариант 17

1. Основной справочник — Склады:
    - номер склада;
    - название склада;
    - занимаемая площадь (m^2);
    - адрес склада;
    - тип склада (перечисление «ТипСклада»: оптовый, розничный, неавтоматизированная рабочая точка);
    - табличная часть (список материально-ответственных лиц склада):
        - сотрудник (справочник «Сотрудники»);
        - домашний адрес.
2. Подчиненный справочник — Товарно-материальные ценности склада:
    - ТМЦ (справочник «ТоварноМатериальныеЦенности»);
    - количество остатков;
    - Стоимость остатков.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «адрес склада» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Удаленные склады» в основном справочнике.

Получить справку: вывести наименование всех складов, имеющих более одного материально-ответственного лица.

Вариант 18

1. Основной справочник — Нормы расхода ГСМ:
    - код модели автомобиля;
    - название модели автомобиля;
    - грузоподъемность;

- табличная часть (список норм расхода ГСМ):
  - вид ГСМ (перечисление «ВидГСМ»: моторное масло, нитрол, солидол, бензин, керосин);
  - норма расхода на 10 литров;
  - норма расхода в %.

2. Подчиненный справочник — Водители:
   - водитель (справочник «Сотрудники»);
   - дата начала эксплуатации;
   - дата окончания эксплуатации.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «название модели автомобиля» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Списанные автомобили» в основном справочнике.

Получить справку: вывести наименование всех моделей автомобиля, имеющих норму расхода нитрола более 10 л.

Вариант 19

1. Основной справочник — Профессии:
   - код профессии;
   - название профессии;
   - минимальная ставка должностного оклада по данной профессии;
   - максимальная ставка должностного оклада по данной профессии;
   - табличная часть (список сотрудников, владеющих данной профессией на предприятии):
     - сотрудник (справочник «Сотрудники»);
     - образование (перечисление «ВидОбразования»: высшее, среднее и т. д.);
     - предыдущее место работы (справочник «Организации»);
2. Подчиненный справочник — Должностные ставки профессии:
   - должность (справочник «Должности»);
   - размер ставки.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «название профессии» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Рабочие профессии» в основном справочнике.

Получить справку: вывести наименование всех профессий, которой владеют более 3 сотрудников.

Вариант 20

1. Основной справочник — Штатное расписание:
   - кафедра (справочник «Кафедры»);
   - зав. кафедрой (справочник «Сотрудники»);
   - табличная часть (список должностей сотрудников подразделения и число ставок):
     - должность (справочник «Должности»);
     - общее число ставок по данной должности;
     - из них занято;
     - из них свободно.
2. Подчиненный справочник — Сотрудники кафедры:
   - сотрудник (справочник «Сотрудники»);
   - должность (справочник «Должности»);
   - размер занимаемой ставки.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «кафедра» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Удаленные подразделения» в основном справочнике.

Получить справку: вывести наименование всех подразделений, у которых нет свободных ставок.

Вариант 21

1. Основной справочник — Виды расчетов подразделений предприятия:
   - код подразделения;
   - наименование подразделения;
   - руководитель (справочник «Сотрудники»);

- табличная часть (список видов расчетов, используемых при расчете заработной платы сотрудников данного подразделения):
  - вид расчета (справочник «ВидыРасчета»);
  - дата начала действия;
  - дата окончания действия.

2. Подчиненный справочник — Сотрудники подразделения:
   - сотрудник (справочник «Сотрудники»);
   - дата поступления;
   - дата увольнения.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «наименование подразделения» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Обслуживающие подразделения» в основном справочнике.

Получить справку: вывести наименования всех подразделений, имеющих в качестве вида расчета расчет с названием «Доплата за вредность».

Вариант 22

1. Основной справочник — Сотрудники:
   - табельный номер сотрудника;
   - ФИО сотрудника;
   - пол сотрудника (перечисление «Пол»);
   - дата приема на работу;
   - оклад;
   - ставка;
   - табличная часть (список членов семьи сотрудника):
     - ФИО;
     - степень родства (перечисление);
     - дата рождения.
2. Подчиненный справочник — Кадровые приказы по сотруднику:
   - номер приказа;
   - дата приказа;
   - содержание приказа.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «пол сотрудника» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Молодые специалисты» в основном справочнике.

Получить справку: вывести ФИО всех сотрудников, имеющих более одного сына.

Вариант 23

1. Основной справочник — Пользователи:
   - код;
   - ФИО;
   - пароль;
   - табличная часть (список объектов, к которым пользователь имеет доступ):
     - объект (справочники «Фирмы», «Контрагенты», «Склады»);
     - тип объекта (перечисление «ТипОбъекта»: фирма, контрагент, склад).
2. Подчиненный справочник — Журнал доступа:
   - объект (справочники «Фирмы», «Контрагенты», «Склады»);
   - дата доступа.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «пароль» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Администраторы» в основном справочнике.

Получить справку: вывести ФИО всех пользователей, имеющих доступ к складам.

Вариант 24

1. Основной справочник — Контрагенты:
   - код контрагента;
   - наименование контрагента;
   - тип контрагента (перечисление «ТипКонтрагента»: юридическое лицо, физическое лицо);
   - адрес контрагента;

- основной расчетный счет (справочник «Расчетные счета»);
- табличная часть (список расчетных счетов контрагента):
  - расчетный счет (справочник «Расчетные счета»);
  - банк (справочник «Банки»);
  - адрес банка;
  - валюта (справочник «Валюты»).

2. Подчиненный справочник — Филиалы контрагентов:
   - название филиала;
   - адрес;
   - телефон.

Замечание 1. При вводе нового элемента справочника обеспечить автоматическое заполнение реквизита «тип контрагента» предопределенным значением.

Замечание 2. Сформировать предопределенную группу «Постоянные покупатели» в основном справочнике.

Получить справку: вывести наименование всех контрагентов, имеющих более одного расчетного счета.

5 ОТЧЕТЫ

Отчеты в рамках Платформы используются для агрегирования информации о состоянии сущностей предметной области с возможностью фильтрации и за определенный период времени. Например, оборотно-сальдовая ведомость, ведомость поступления товаров, расчетный лист сотрудника и т. д.

Отчеты отображают сформированные печатные формы. Печатные формы можно формировать отдельно от отчетов, например, для представления содержимого справочников или документов в виде, пригодном для вывода на печать. Примером такого рода печатной формы может служить накладная о получении товара.

5.1 Создание и настройка отчета

Для создания отчета необходимо выбрать узел «Отчеты» в дереве объектов метаданных, а затем нажать на него правой кнопкой мыши и выбрать пункт «Добавить» в появившемся меню (рис. 5.1).

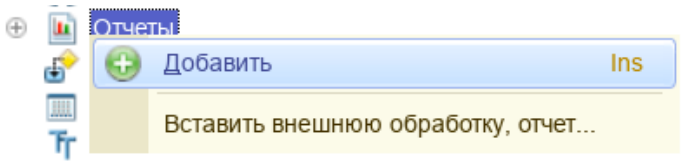


Рис. 5.1. Создание отчета

На рисунке 5.2 представлено окно свойств вновь созданного отчета. Как видно из рисунка 5.2, свойства отчета во много схожи со свойствами справочника.

Вкладка «Формы» окна свойств отчета позволяет создавать формы отчета. Как видно из рисунка 5.3 отчет содержит несколько типов форм:

- форма отчета — используется для настройки параметров отчета и отображения печатной формы отчета;
- форма настроек — используется для выбора настроек схемы компоновки данных;
- форма варианта — используется для выбора вариантов отчета схемы компоновки данных.

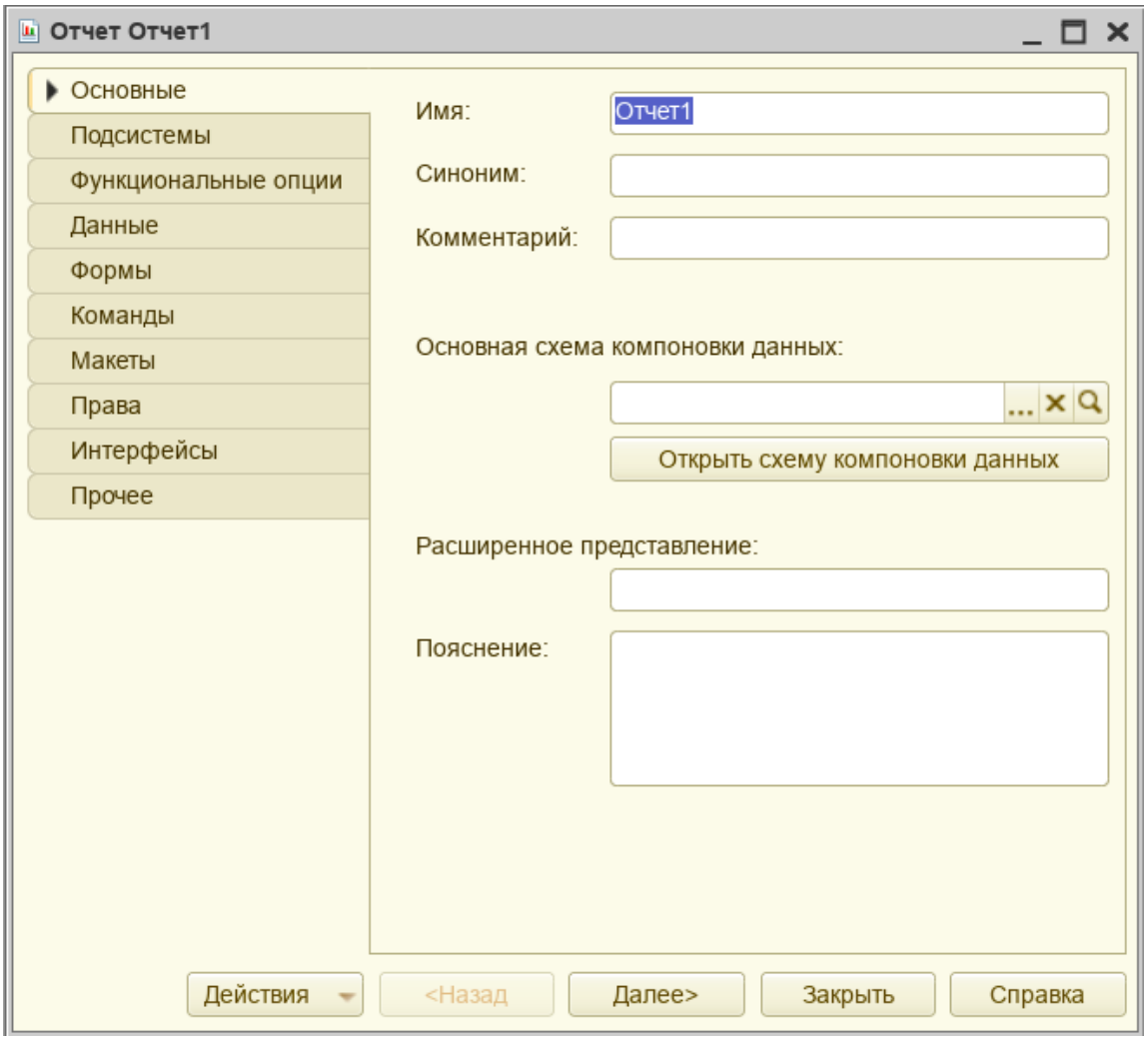


Рис. 5.2. Вкладка «Основные» окна свойств отчета

Схема компоновки данных используется в системе компоновки данных. Система компоновки данных представляет собой механизм, основанный на декларативном описании отчетов.

В текущей главе описан традиционный способ создания отчета с необходимостью создания макета и написанием алгоритма формирования печатной формы.

Макет — объект метаданных Платформы, предназначенный для настройки внешнего вида печатной формы. Макеты могут быть общими или принадлежать определенному объекту метаданных, например, справочнику, документу, отчету и т. д.

Для создания макета необходимо перейти на вкладку «Макеты» окна свойств отчета и нажать на кнопку «Добавить» панели команд. В появившемся окне конструктора макета (рис. 5.4) можно указать имя макета и его тип, а затем нажать на кнопку

«Готово». Обычно при разработке печатных форм используется тип макета «Табличный документ».

В основном табличный документ используется для вывода информации, но иногда его можно использовать для ввода данных в систему путем редактирования значений ячеек табличного документа. С помощью элемента управления «ПолеТабличногоДокумента» табличный документ может быть отображен на экранной форме отчета.

Особенности табличного документа:

1. Поддерживается механизм расшифровок.
2. В табличном документе могут быть использованы группировки для формирования промежуточных итогов, например, объем продаж по каждому отделу.

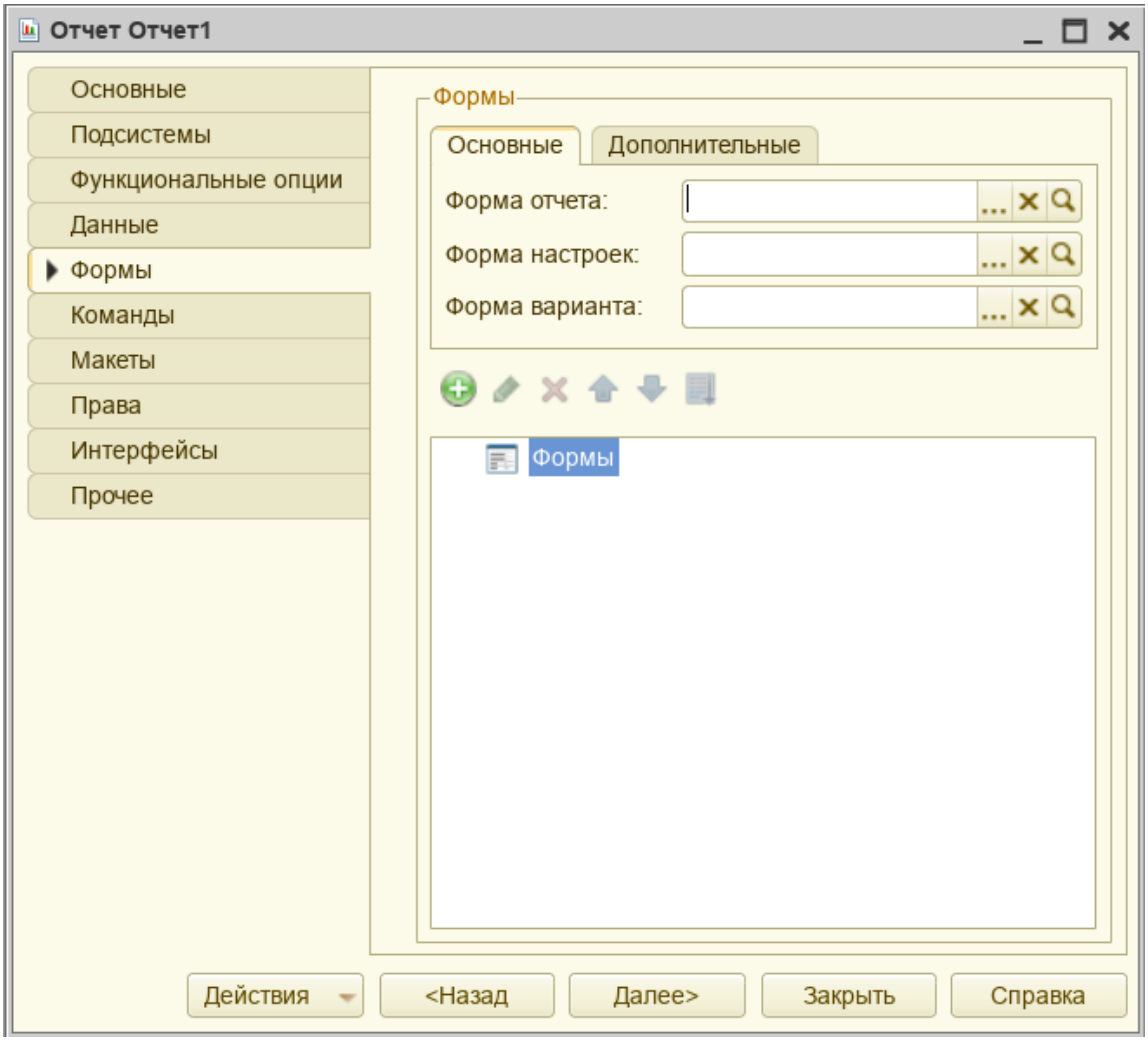


Рис. 5.3. Вкладка «Формы» окна свойств отчета

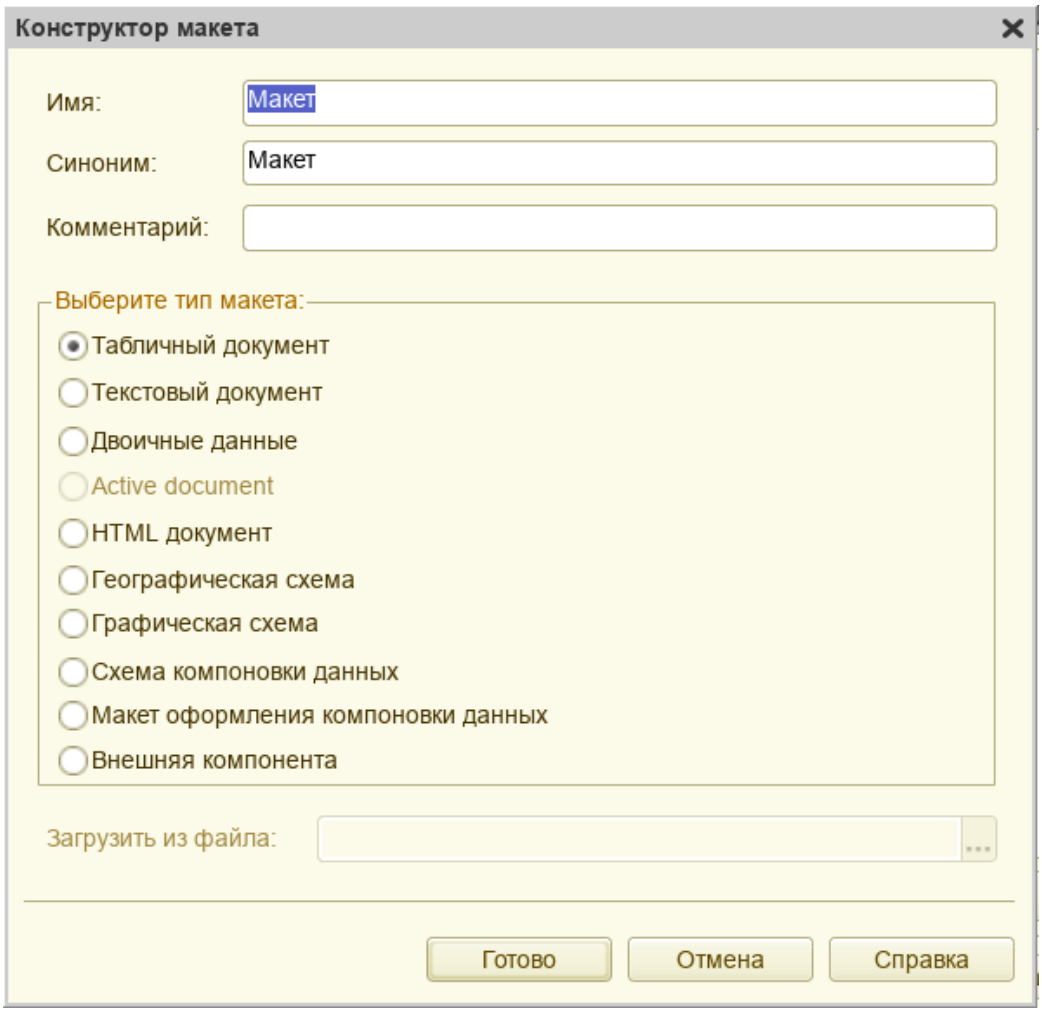


Рис. 5.4. Конструктор макета

3. Ячейки табличного документа могут иметь примечания, содержащие дополнительную информацию.
4. Табличный документ может быть сохранен интерактивно или программно во внешние файлы различного формата.
5. Табличный документ может быть загружен из внешнего файла.

Табличный документ представляет собой множество ячеек, объединенных в строки и столбцы.

Ячейки табличного документа могут быть заполнены данными следующих типов (рис. 5.5):

- «Текст»;
- «Параметр» — если ячейка табличного документа содержит параметр, то зная имя данного параметра с ним можно работать как с переменной: записывать и считывать значение;

- «Шаблон». Шаблон позволяет объединить текст и параметры. Параметры встраиваются в текст путем включения имени параметра в квадратные скобки, например, Ведомость поступления товара от поставщика `[ Поставщик ]` с `[ ДатаНачалаПериода ]` по `[ ДатаОкончанияПериода ]`.



Рис. 5.5. Параметр «Заполнение» свойств ячейки

Для удобства работы ячейки табличного документа можно объединять в области и назначать им имена (рис. 5.6). Область может включать в себя одну или несколько строк, колонок или любой прямоугольный диапазон ячеек. К области можно обратиться по координатам или имени.

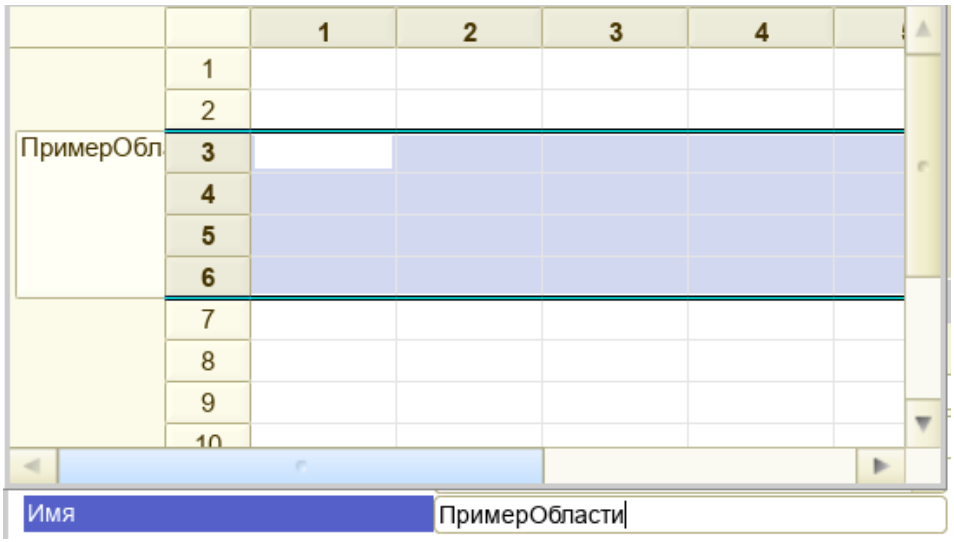


Рис. 5.6. Объединение ячеек в область с именем «ПримерОбласти»

5.2 Методы работы с табличным документом

В рамках Платформы макеты и области также имеют тип `ТабличныйДокумент`.

Рассмотрим пример создания нового табличного документа средствами встроенного языка Платформы;

```
// 1-й способ
// Создание динамического табличного документа
ТабДок = Новый ТабличныйДокумент();
```

```
// 2-й способ
// Использование элемента управления
// ПолеТабличногоДокумента
ТабДок = ЭлементыФормы.ПолеТабличногоДокумента1;
```

Получить макет можно следующим образом:

```
// 1-й способ
// Получить макет объекта метаданных
Макет = Справочники.ПримерСправочника
    .ПолучитьМакет("ПримерМакета");

// 2-й способ
// Получить общий макет
Макет = ПолучитьОбщийМакет("ПримерОбщегоМакета");
```

Для работы с областями необходимо получить ссылку на область с помощью метода ПолучитьОбласть объекта ТабличныйДокумент:

```
Область = Макет.ПолучитьОбласть("ПримерОбласти");
```

Для работы с параметрами и параметрами шаблона области используется свойство Параметры объекта ТабличныйДокумент:

```
Область.Параметры.Параметр1 = "Некоторое значение";
Область.Параметры.ПараметрШаблона1 = 100;
```

Если область содержит большое число параметров, для их быстрого заполнения используется метод Заполнить объекта ТабличныйДокумент:

```
Выборка = Справочники.ПримерСправочника.Выбрать();
Пока Выборка.Следующий() Цикл
    Область.Параметры.Заполнить(Выборка);
    Область.Параметры.Имя = Выборка.Наименование;
КонецЦикла;
```

При использовании метода Заполнить объекта ТабличныйДокумент необходимо помнить, что автоматически заполнятся только те параметры области, имена которых соответствуют именам свойств, переданного в метод Заполнить аргумента. Например, справочник «ПримерСправочника» содержит реквизиты «Код», «Наименование» и «Реквизит1», а область «Область» содержит параметры «Код», «Имя» и «Реквизит1», тогда при вызове метода Заполнить заполнены будут только параметры области «Код» и «Реквизит1». Параметр «Имя» области можно заполнить вручную.

Для вывода области в табличный документ необходимо использовать метод Вывести целевого объекта ТабличныйДокумент:

```
ТабДок.Вывести(Область);
```

Объект ТабличныйДокумент также содержит набор параметров, с помощью которых можно настраивать его внешний вид, например:

```
ТабДок.ТолькоПросмотр = Истина;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.ОтображатьСетку = Ложь;
```

При использовании динамического табличного документа необходимо вызвать метод Показать объекта ТабличныйДокумент:

```
ТабДок.Показать("Пример табличного документа");
```

Параметр метода Показать объекта ТабличныйДокумент позволяет задать заголовок окна, в котором будет отображаться динамический табличный документ.

Если для отображения табличного документа используется элемент управления «ПолеТабличногоДокумента», может возникнуть необходимость очистки табличного документа перед выводом в него новых сведений. Для очистки табличного документа используется метод Очистить объекта ТабличныйДокумент:

```
ТабДок.Очистить();
```

При двойном щелчке (или нажатии клавиши *Enter*) на ячейке табличного документа возникает событие «Выбор».

Табличный документ позволяет использовать механизм расшифровки. Расшифровки делятся на стандартные и нестандартные. Стандартные расшифровки обрабатываются системой автоматически:

- если в параметре расшифровки указана ссылка на объект, тогда открывается форма объекта по ссылке, например, форма элемента справочника, форма документа и т. д.;
- если в параметре расшифровки указан простой тип данных, то открывается диалоговое окно, содержащее значение параметра в виде текстового сообщения.

Для обеспечения работы механизма расшифровки необходимо установить имя параметра расшифровки. В свойствах ячейки необходимо указать имя параметра расшифровки в свойстве «ПараметрРасшифровки» (рис. 5.7):

```
Область.Параметры.ПримерПараметраРасшифровки = <Значение>;
```

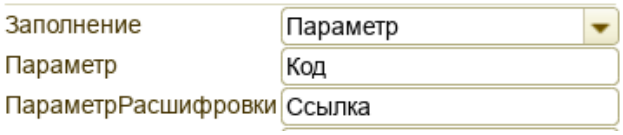


Рис. 5.7. Определение значения свойства «ПараметрРасшифровки» ячейки

Для определения алгоритма работы нестандартной расшифровки необходимо создать обработчик события «ОбработкаРасшифровки» для элемента управления «ПолеТабличногоДокумента» (рис. 5.8).



Рис. 5.8. Создание обработчика события «ОбработкаРасшифровки» для элемента управления «ПолеТабличногоДокумента»

Следует различать события «Выбор» и «ОбработкаРасшифровки. Событие «Выбор» возникает даже если у ячейки не задано свойство «ПараметрРасшифровки». Событие «ОбработкаРасшифровки» возникает только у ячеек с заданным свойством «ПараметрРасшифровки». Сначала возникает событие «Выбор», а затем «ОбработкаРасшифровки».

Событие «ОбработкаРасшифровки» можно обработать только при отображении табличного документа на форме в элементе управления «ПолеТабличногоДокумента». Обработать событие «ОбработкаРасшифровки» в динамическом табличном документе невозможно.

Обработать расшифровку можно следующим образом:

```
Процедура ПолеТабличногоДокумента1ОбработкаРасшифровки(
        Элемент,
        Расшифровка,
        СтандартнаяОбработка,
        ДополнительныеПараметры)
    // Отключение стандартной расшифровки
    СтандартнаяОбработка = Ложь;

    // Сохранение значения параметра расшифровки
    // в локальную переменную
    ТекТовар = Расшифровка;

    ...
КонецПроцедуры
```

Если параметр СтандартнаяОбработка установлен в значение Истина, то помимо нестандартной расшифровки будет выполняться и стандартная расшифровка.

Изменить поведение обработчика расшифровки у ячейки можно с помощью свойства «ИспользованиеРасшифровки» (рис. 5.9):

- «Ячейка» — вызов обработки расшифровки только для текущей ячейки;
- «Строка» — вызов обработки расшифровки для текущей ячейки, а также при попытке расшифровать любую ячейку, в которой не установлено значение параметра «ОбработкаРасшифровки», текущей строки;
- «Без обработки» — отключение обработки расшифровки для текущей ячейки.



Рис. 5.9. Определение типа использования расшифровки

Группировать сведения табличного документа можно по строкам и по столбцам (рис. 5.10).



Рис. 5.10. Пример группировки строк

В большинстве случаев для организации группировки используются методы НачатьАвтоГруппировкуСтрок и ЗакончитьАвтоГруппировкуСтрок объекта ТабличныйДокумент. Данные методы включают режим автоматической группировки строк:

```
ТабДок.НачатьАвтогруппировкуСтрок();
Выборка = Справочники.ПримерСправочника
    .ВыбратьИерархически();

Пока Выборка.Следующий() Цикл
    ...
    ТабДок.Вывести(Область,
        // Уровень группы
        Выборка.УровеньВВыборке(),
```

```
            // Наименование группы
            Выборка.Наименование,
            // Отключение автоматического раскрытия группы
            Ложь);
    КонецЦикла;
    ТабДок.ЗакончитьАвтогруппировкуСтрок();
```

Ячейки табличного документа могут иметь примечания для отображения дополнительной информации во всплывающем окне при наведении курсора мыши на ячейку (рис. 5.11). Программная установка примечаний реализуется следующим образом:

```
Выборка = Справочники.ПримерСправочника.Выбрать();
Пока Выборка.Следующий() Цикл
    // R2C2 - имя ячейки
    ОбластьПримечание = Макет.Область("R2C2");
    ОбластьПримечание.Примечание.Текст = Выборка
        .Примечание;
    Область = Макет.ПолучитьОбласть("ПримерОбласти");
    ...
    ТабДок.Вывести(Область);
КонецЦикла;
```



Рис. 5.11. Пример примечания ячейки

5.3 Пример выполнения задания

В рамках четвертой лабораторной работы необходимо научиться работать с объектом метаданных «Отчет» и объектом «ТабличныйДокумент». Для этого требуется выполнить следующее задание:

1. Разработать печатную форму для вывода содержимого основного справочника. Печатная форма должна иметь группировку и стандартную расшифровку для открытия формы элемента основного справочника. Для формирования печатной формы необходимо добавить кнопку «Печать» на командную панель формы списка основного справочника.
2. Разработать отчет для вывода печатной формы с содержимым основного справочника. Печатная форма должна иметь группировку и нестандартную расшифровку. В каче-

стве нестандартной расшифровки необходимо вывести печатные формы для отображения содержимого:

- табличной части элемента основного справочника;
- элементов подчиненного справочника для выбранного элемента основного справочника.

3. Предусмотреть возможность вывода примечания для всех печатных форм.

Лабораторная работа должна быть выполнена в режиме *обычного приложения*.

Для выполнения первого пункта задания необходимо добавить кнопку «Печать» на командную панель формы списка справочника «Товары» (рис. 5.12) и установить следующие свойства кнопки:

- Имя — Печать;
- Текст — Печать;
- Картинка — Печать;
- Отображение — Надпись и картинка.

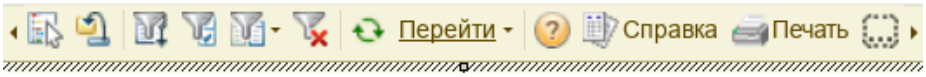


Рис. 5.12. Кнопка «Печать»

Далее необходимо создать макет справочника «Товары». Структура макета представлена на рисунке 5.13.

		1	2	3	4	5	6
Шапка	1						
	2			**<Список товаров на [Дата]>**			
	3						
	4		Код	Наименование	Единица измерения	Минимальный остаток	Ставка НДС
	5						
Группа	6		***<Код>***	***<Наименование>***			
	7						
Элемент	8		<Код>	<Наименование>	<ЕдиницаИзмерения	<МинимальныйОстато	<СтавкаНДС>

Рис. 5.13. Макет печатной формы

Как видно из рисунка 5.13, макет печатной формы содержит шаблон в области «Шапка» для вывода заголовка печатной формы, и несколько параметров («Код», «Наименование», «Единица измерения» и т. д.) в областях «Группа» и «Элемент» для вывода информации о группах и элементах справочника «Товары» соответственно. Параметры расшифровки «СсылкаГруппа» и «Ссылка» заданы для ячеек с параметром «Код», с установкой

свойства «ИспользованиеРасшифровки» в значение «Строка», для областей «Группа» и «Элемент» соответственно.

Формирование печатной формы при нажатии на кнопку «Печать» состоит из следующих шагов:

1. Создание динамического табличного документа.

```
ТабДок = Новый ТабличныйДокумент();
```

2. Получение ссылки на макет.

```
Макет = Справочники.Товары.ПолучитьМакет("Макет");
```

3. Заполнение и вывод шапки печатной формы.

```
ОбластьШапка = Макет.ПолучитьОбласть("Шапка");
ОбластьШапка.Параметры.Дата =
    // Форматирование даты
    Формат(ТекущаяДата(), "ДФ=dd.MM.уууу");
ТабДок.Вывести(ОбластьШапка);
```

4. Включение автогруппировки строк.

```
ТабДок.НачатьАвтогруппировкуСтрок();
```

5. Заполнение и вывод сведений о группах и элементах иерархического справочника «Товары».

```
Спр = Справочники.Товары.ВыбратьИерархически();
Пока Спр.Следующий() Цикл
    Если Спр.ЭтоГруппа Тогда
        ОбластьГруппа = Макет
            .ПолучитьОбласть("Группа");
        ОбластьГруппа.Параметры.Заполнить(Спр);
        // Заполнение параметра расшифровки
        ОбластьГруппа.Параметры.ССылкаГруппа =
            Спр.Ссылка;
        ТабДок.Вывести(ОбластьГруппа,
            Спр.УровеньВВыборке());
    Иначе
        ОбластьЭлемент = Макет
            .ПолучитьОбласть("Элемент");
        ОбластьЭлемент.Параметры.Заполнить(Спр);
        ТабДок.Вывести(ОбластьЭлемент,
            Спр.УровеньВВыборке());
    КонецЕсли;
КонецЦикла;
```

6. Отключение автогруппировки строк.

```
ТабДок.ЗакончитьАвтогруппировкуСтрок();
```

7. Настройка внешнего вида табличного документа.

```
ТабДок.ТолькоПросмотр = Истина;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.ОтображатьСетку = Ложь;
```

8. Отображение табличного документа.

```
ТабДок.Показать("Список товаров");
```

После нажатия на кнопку «Печать» будет сформирована печатная форма, изображенная на рисунке 5.14.

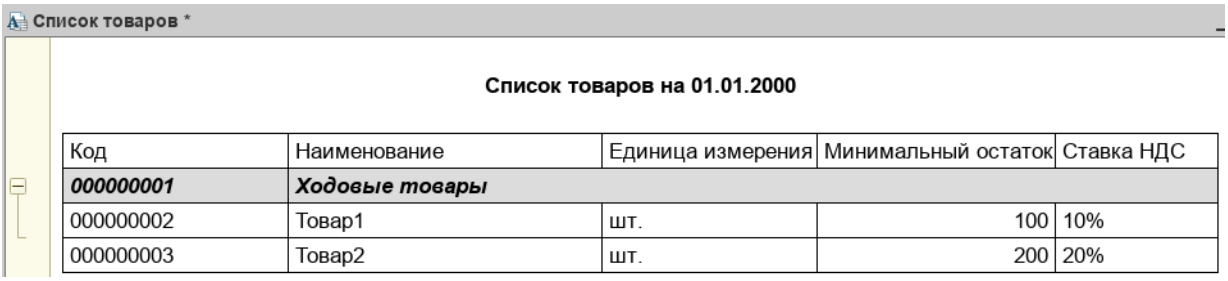


Рис. 5.14. Печатная форма справочника «Товары»

Для реализации второго пункта задания необходимо создать отчет с именем «ОтчетПоТоварам». Форма отчета «ОтчетПоТоварам» представлена на рисунке 5.15 и содержит элемент управления «ПолеТабличногоДокумента» с именем «ПолеТабличногоДокумента1».

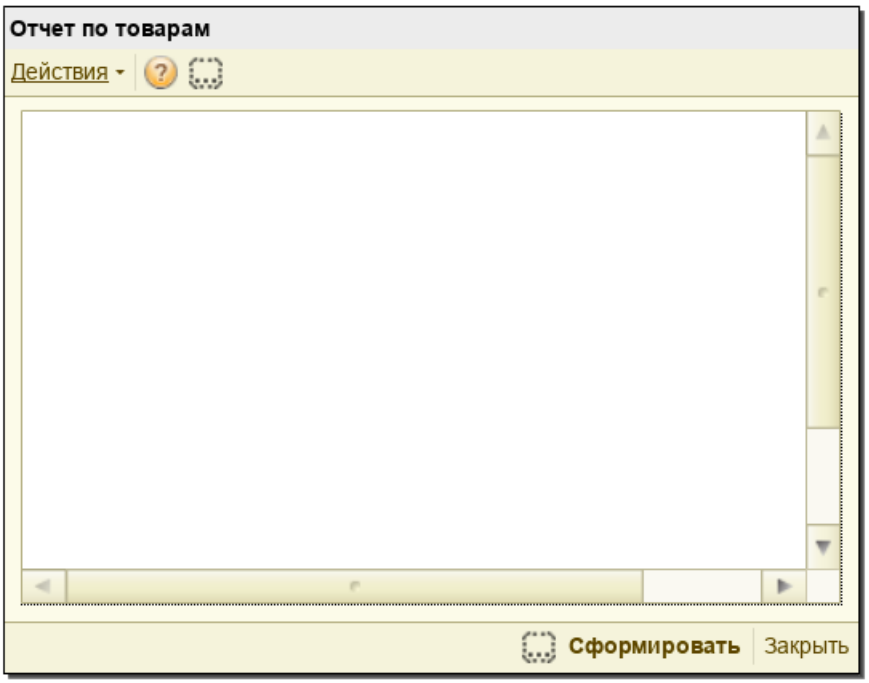


Рис. 5.15. Форма отчета «ОтчетПоТоварам»

Обработчик кнопки «Сформировать» отчета «ОтчетПоТоварам» будет идентичен обработчику кнопки «Печать» справочника «Товары» за исключением нескольких строк:

1. Вместо создания динамического табличного документа используется элемент управления «ПолеТабличногоДокумента1».

```
// Замените строку
// ТабДок = Новый ТабличныйДокумент();
// на
ТабДок = ЭлементыФормы.ПолеТабличногоДокумента1;
ТабДок.Очистить();
```

2. Строку для отображения динамического табличного документа можно удалить.

```
// Эту строку можно удалить
ТабДок.Показать("Список товаров");
```

Также необходимо создать два макета:

- макет для печатной формы с содержимым табличной части элемента справочника «Товары» (рис. 5.16);

		1	2	3	4	5
Шапка	1					
	2		<Информация о поставщиках товара [Товар]>			
	3					
	4		**Поставщик**	**Адрес**	**Закупочная цена**	**Дата поставки**
	5					
Строка	6		<Поставщик>	<Адрес>	<ЗакупочнаяЦена>	<ДатаПоставки>

Рис. 5.16. Макет для табличной части

- макет для печатной формы с содержимым подчиненного справочника для выбранного элемента справочника «Товары» (рис. 5.17).

		1	2	3
Шапка	1			
	2		<Список цен товара [Товар]>	
	3			
	4		**Тип цены**	**Значение цены**
	5			
Строка	6		<ТипЦены>	<ЗначениеЦены>

Рис. 5.17. Макет для подчиненного справочника

Обработка нестандартной расшифровки ячеек отчета «ОтчетПоТоварам» будет состоять из следующих шагов:

1. Отключение стандартной расшифровки.

```
СтандартнаяОбработка = Ложь;
```

2. Сохранение значения параметра расшифровки в локальную переменную.

```
ТекТовар = Расшифровка;
```

3. Добавление проверки для расшифровки только элементов справочника «Товары».

```
Если ТекТовар.ЭтоГруппа Тогда
    Возврат;
КонецЕсли;
```

4. Формирование печатной формы с содержимым табличной части элемента справочника «Товары».

```
ТабДокТЧ = Новый ТабличныйДокумент();
МакетТЧ = Отчеты.ОтчетПоТоварам
    .ПолучитьМакет("МакетТЧ");
ОбластьШапкаТЧ = МакетТЧ.ПолучитьОбласть("Шапка");
ОбластьШапкаТЧ.Параметры.Товар = ТекТовар;
ТабДокТЧ.Вывести(ОбластьШапкаТЧ);

Для Каждого Стр Из ТекТовар.Поставки Цикл
    ОбластьСтрокаТЧ = МакетТЧ
        .ПолучитьОбласть("Строка");
    ОбластьСтрокаТЧ.Параметры.Заполнить(Стр);
    ТабДокТЧ.Вывести(ОбластьСтрокаТЧ);
КонецЦикла;

ТабДокТЧ.ТолькоПросмотр = Истина;
ТабДокТЧ.ОтображатьЗаголовки = Ложь;
ТабДокТЧ.ОтображатьСетку = Ложь;
ТабДокТЧ.Показать("Список поставок");
```

5. Формирование печатной формы, содержащей элементы подчиненного справочника для выбранного элемента справочника «Товары».

```
ТабДокП = Новый ТабличныйДокумент();
МакетП = Отчеты.ОтчетПоТоварам
    .ПолучитьМакет("МакетП");
ОбластьШапкаП = МакетП.ПолучитьОбласть("Шапка");
ОбластьШапкаП.Параметры.Товар = ТекТовар;
ТабДокП.Вывести(ОбластьШапкаП);

СпрП = Справочники.ЦеныНоменклатуры
    .Выбрать(, ТекТовар);
```

```
Пока СпрП.Следующий() Цикл
    ОбластьСтрокаП = МакетП.ПолучитьОбласть("Строка");
    ОбластьСтрокаП.Параметры.Заполнить(СпрП);
    ТабДокП.Вывести(ОбластьСтрокаП);
КонецЦикла;

ТабДокП.ТолькоПросмотр = Истина;
ТабДокП.ОтображатьЗаголовки = Ложь;
ТабДокП.ОтображатьСетку = Ложь;
ТабДокП.Показать("Список цен товара");
```

Результаты нестандартной расшифровки представлены на рисунках 5.18 и 5.19.

Список поставок

Информация о поставщиках товара Товар1

Поставщик	Адрес	Закупочная цена	Дата поставки
Поставщик1	Адрес поставщика 1	100	01.01.2000
Поставщик1	Адрес поставщика 1	110	02.01.2000
Поставщик1	Адрес поставщика 1	150	03.01.2000

Рис. 5.18. Печатная форма с содержимым табличной части

Список цен товара

Список цен товара Товар1

Тип цены	Значение цены
Закупочная	200
Оптовая	210
Розничная	250

Рис. 5.19. Печатная форма с подчиненными элементами

Вывод примечаний необходимо разработать самостоятельно.

5.4 Контрольные вопросы

1. Каким образом работает механизм расшифровки?
2. В чем разница между стандартной и нестандартной расшифровкой?

3. Объясните, каким образом можно получить элементы табличной части элемента справочника?
4. Объясните, каким образом можно получить элементы подчиненного справочника элемента справочника?

5.5 Варианты заданий

Лабораторная работа должна быть выполнена в режиме *обычного приложения*.

В рамках данной лабораторной работы необходимо выполнить следующее задание:

1. Разработать печатную форму для вывода содержимого основного справочника. Печатная форма должна иметь группировку и стандартную расшифровку для открытия формы элемента основного справочника. Для формирования печатной формы необходимо добавить кнопку «Печать» на командную панель формы списка основного справочника.
2. Разработать отчет для вывода печатной формы с содержимым основного справочника. Печатная форма должна иметь группировку и нестандартную расшифровку. В качестве нестандартной расшифровки необходимо вывести печатные формы для отображения содержимого:
    - табличной части элемента основного справочника;
    - элементов подчиненного справочника для выбранного элемента основного справочника.
3. Предусмотреть возможность вывода примечания для всех печатных форм.

   Варианты заданий представлены в главе 4.

6 ДОКУМЕНТЫ

Документы в рамках Платформы используются для формирования и изменения состояния сущностей проблемной области через реализацию определенных бизнес-процессов. Примерами бизнес-процессов могут быть: поступление товара, возврат товара поставщику, реализация товара и т. д.

6.1 Создание и настройка документа

Для создания документа необходимо выбрать узел «Документы» в дереве объектов метаданных, а затем нажать на него правой кнопкой мыши и выбрать пункт «Добавить» в появившемся меню (рис. 6.1).

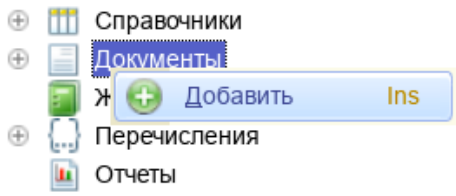


Рис. 6.1. Создание документа

На рисунке 6.2 представлено окно свойств вновь созданного документа.

На рисунке 6.3 представлено содержимое вкладки «Данные» документа. Как видно из рисунка 6.3, данная вкладка свойств документа аналогична справочнику. Однако варианты использования реквизитов и табличной части в документе отличаются от справочника.

Реквизиты создаются в документе для указания сведений, связанных непосредственно с бизнес-процессом, например, «Поставщик», «Склад» и т. д. Реквизиты описывают так называемую шапку документа.

Табличные части используются для перечисления сведений, представленных во множественном числе, но привязанных к одному документу, например, список товаров для поступления.

На рисунке 6.4 представлены стандартные реквизиты документа.

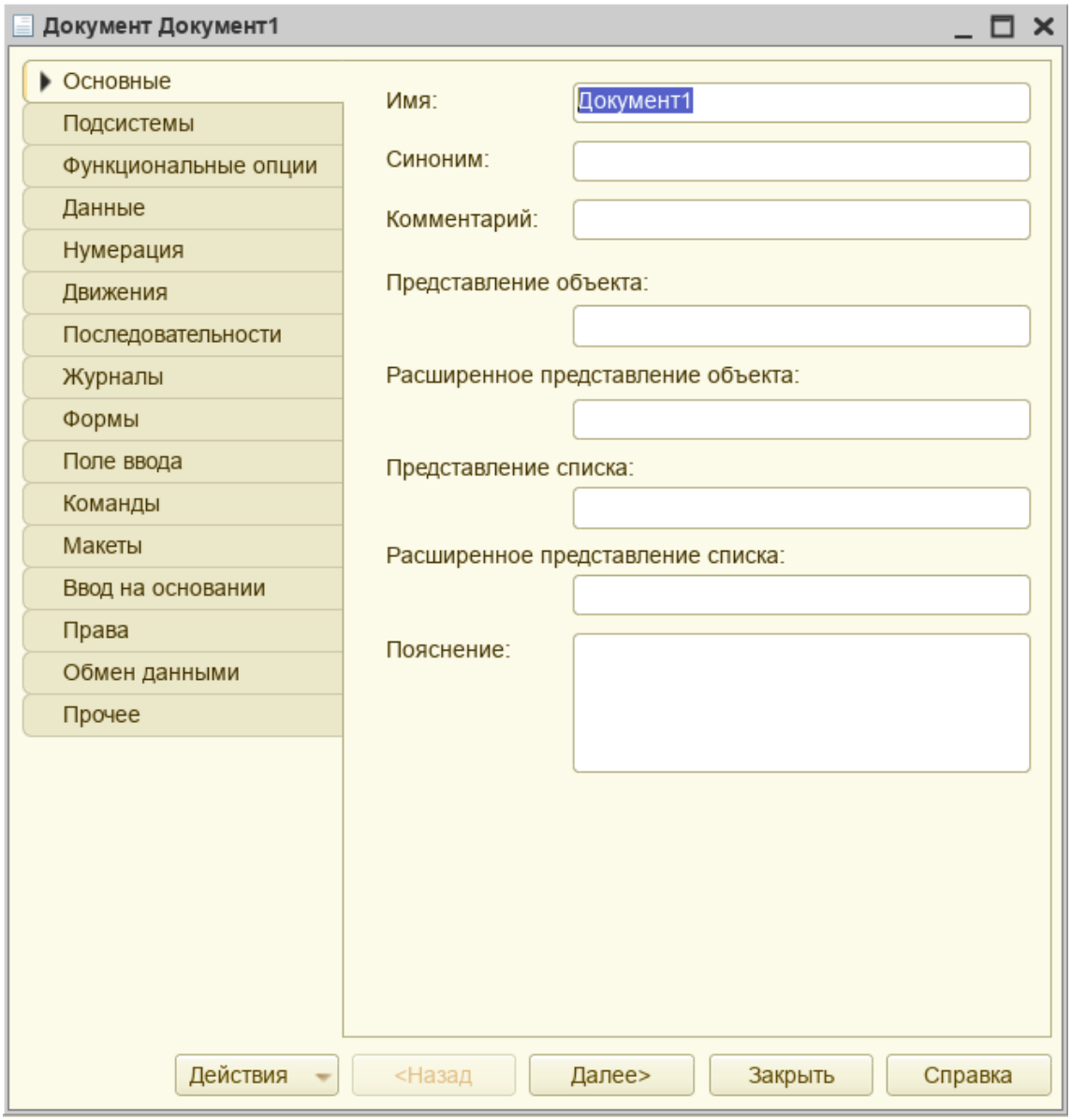


Рис. 6.2. Вкладка «Основные» окна свойств документа

Как видно из рисунка 6.4, к стандартным реквизитам документа относятся:

- Ссылка — уникальный идентификатор документа, используется для организации связей и доступа к данным документа;
- Номер — номер документа, обычно используется вместе с датой для идентификации документа со стороны пользователя в рамках предметной области, например, «Реализация товара» № 00000001 от 01.01.1990 года;
- Дата — дата и время создания документа (см. номер);
- Проведен — флаг, определяющий проведен ли документ;

- ПометкаУдаления — флаг, определяющий помечен ли данный документ для удаления (аналогично справочнику).

Если два документа имеют одинаковую дату и время, то они выстраиваются в хронологическую последовательность с учетом ссылки документа (момент времени).

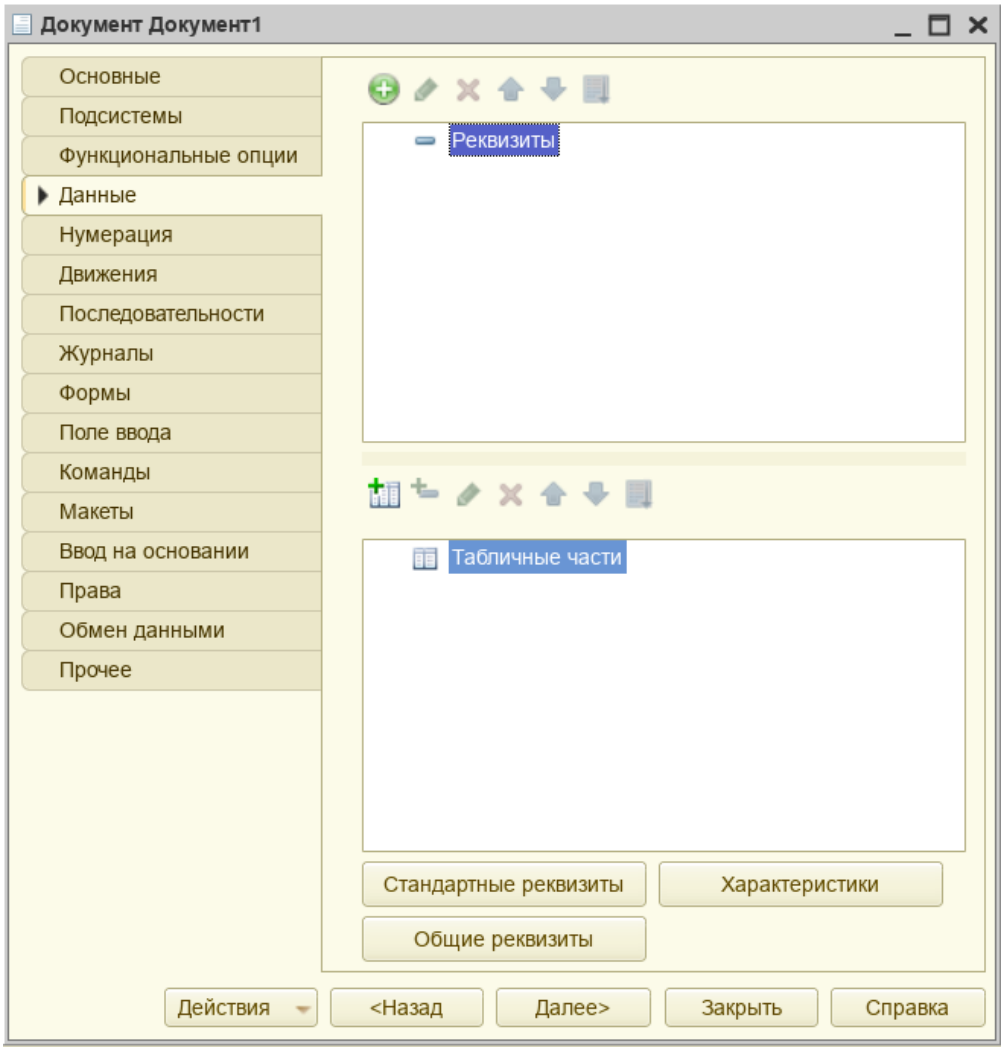


Рис. 6.3. Вкладка «Данные» окна свойств документа

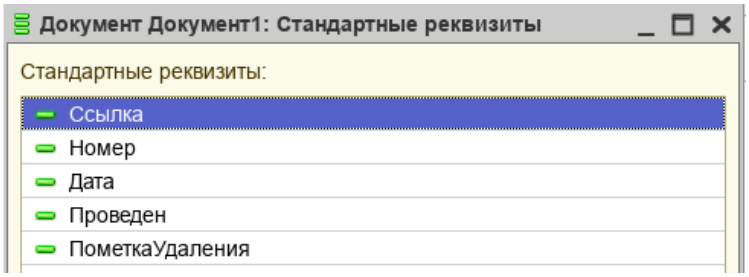


Рис. 6.4. Стандартные реквизиты документа

6.2 Методы работы с документами

Рассмотрим пример программного создания нового документа средствами встроенного языка Платформы:

```
НовыйДокумент = Документы.ПримерДокумента
    .СоздатьДокумент();
// При программном создании документа
// дата документа не заполняется автоматически,
// необходимо установить дату вручную
НовыйДокумент.Дата = Дата;
НовыйДокумент.Записать();
```

Перебрать документы определенного типа за некоторый период, в том числе с отбором, можно следующим образом:

```
// Перебор всех документов
Выборка = Документы.ПримерДокумента.Выбрать();
Пока Выборка.Следующий() Цикл
    Сообщить("Документ " + Выборка.Номер
        + " от " + Выборка.Дата);
    // Перебор строк табличной части
    Для Каждого Строка Из Выборка.ТабличнаяЧасть Цикл
        Сообщить(Строка.РеквизитТабличнойЧасти);
    КонецЦикла;
КонецЦикла;

// Перебор всех документов с отбором по периоду
ДатаНач = Дата(1990, 1, 1);
ДатаКон = Дата(1991, 1, 1);
Выборка= Документы.ПримерДокумента
    .Выбрать(ДатаНач, ДатаКон);
Пока Выборка.Следующий() Цикл
    ...
КонецЦикла;

// Перебор документов с отбором по реквизиту
// и сортировкой
Отбор = Новый Структура("РеквизитСсылка", Значение);
Выборка = Документы.ПримерДокумента.Выбрать( , , Отбор,
    "РеквизитЧисло Убыв")
Пока Выборка.Следующий() Цикл
    ...
КонецЦикла;
```

Найти документ по номеру или значению реквизита можно с помощью методов НайтиПоНомеру и НайтиПоРеквизиту соответственно:

```
ДокументСсылка = Документы.ПримерДокумента
    .НайтиПоНомеру("000000001");
Если ДокументСсылка.Пустая() Тогда
    Сообщить("Документ не найден!");
Иначе
    Сообщить("Документ найден!");
КонецЕсли;
```

Удалить документы можно с помощью метода `Удалить` объекта `Документ`:

```
ДокументСсылка = Документы.ПримерДокумента
    .НайтиПоНомеру("000000001");
Если НЕ ДокументСсылка.Пустая() Тогда
    // Для редактирования документа необходимо
    // получить объект документа по ссылке
    ДокументОбъект = ДокументСсылка.ПолучитьОбъект();
    // 1-й способ
    // Установка пометки удаления
    // Если документ был проведен,
    // то при удалении флаг Проведен
    // устанавливается в значение Ложь
    ДокументОбъект.УстановитьПометкуУдаления(Истина);
    // 2-й способ
    // Непосредственное удаление объекта
    ДокументОбъект.Удалить();
КонецЕсли;
```

6.3 Пример выполнения задания

В рамках третьей лабораторной работы необходимо научиться работать с объектом метаданных «Документ». Для это требуется выполнить задание, состоящее из следующих пунктов:

1. Разработать и заполнить справочники, необходимые для заполнения документов.
2. Разработать документы, обеспечивающие накопление информации, необходимой для формирования отчета.
3. С помощью разработанных документов подготовить исходные данные для формирования отчета.
4. Разработать отчет путем обращения к перечню документов:
    - сформировать форму диалога отчета;
    - разработать макет отчета;
    - разработать модуль формирования отчета;
    - обеспечить расшифровку ячеек отчета.

5. Выполнить проверку функциональности разработанного отчета.

Замечание: Для построения отчета, возможно, необходима предварительная сортировка и свертка данных, извлеченных из документов. Используйте для этих целей таблицу значений.

Рассмотрим процесс выполнения следующего варианта задания, в рамках которого необходимо разработать отчет:

Ведомость поступления товара от поставщиков с учетов возвратов за период с <Дата> по <Дата>

Товар	Поступило		Возвращено		% брака
	Количество	Сумма	Количество	Сумма	
Итого	<>		<>		<>

Замечание: Разработать документы «Поступление товара» и «Возврат поставщику».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Лабораторная работа должна быть выполнена в режиме *обычного приложения*.

Для выполнения задания нет необходимости создавать дополнительные справочники, так как предметная область не изменилась относительно задания прошлых лабораторных работ.

В задании сказано, что необходимо разработать два документа — «Поступление товара» и «Возврат поставщику». Таким образом, данные документы позволяют фиксировать факт поступления товара и выбытия товара (возврат), изменяя значение остатков товара на складе организации.

Содержимое вкладки «Данные» документа «Поступление товара» будет иметь вид, представленный на рисунке 6.5.

Как видно из рисунка 6.5, документ «Поступление товара» содержит один реквизит «Поставщик», имеющий тип «СправочникСсылка.Контрагенты», и одну табличную часть «Товары», состоящую из следующих реквизитов:

- Товар — СправочникСсылка.Товары;
- Цена — Число(10, 2, Неотрицательное);
- Количество — Число(10, 0, Неотрицательное);
- Сумма — Число(10, 2, Неотрицательное).

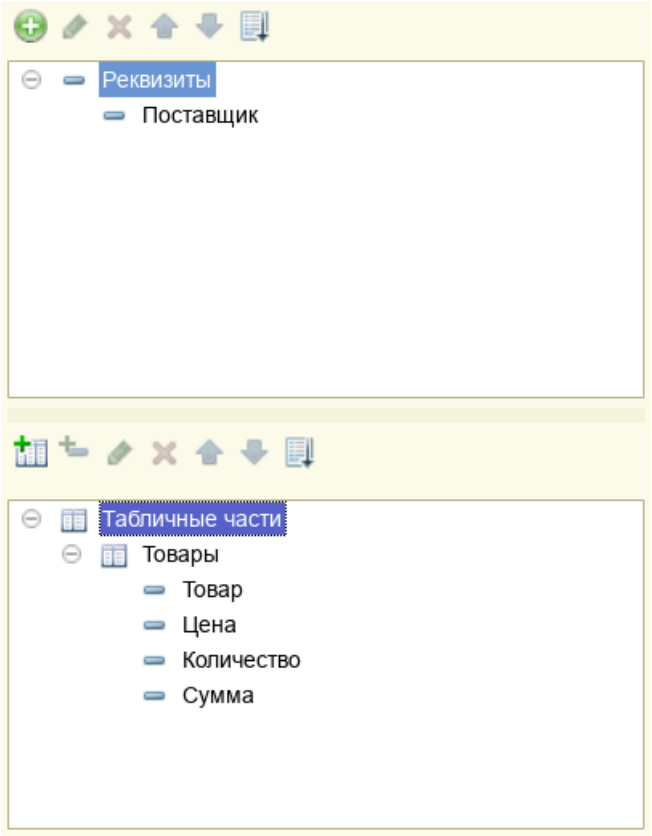


Рис. 6.5. Вкладка «Данные» документа «Поступление товара»

Структура данных документа «Возврат поставщику» аналогична документу «Поступление товара».

Для автоматического расчета реквизита «Сумма» табличной части «Товары» на основе значений реквизитов «Цена» и «Количество» необходимо:

1. Создать форму документа.
2. Для поля ввода реквизита «Цена» создать обработчик события «При изменении» (рис. 6.6).

   Обработчик события «При изменении» должен иметь вид:

```
Процедура ТоварыЦенаПриИзменении(Элемент)
    ТС = ЭлементыФормы.Товары.ТекущаяСтрока;
    ТС.Сумма = ТС.Цена * ТС.Количество;
КонецПроцедуры
```

Переменная `ТС` содержит ссылку на строку табличной части «Товары», с которой в данный момент взаимодействует пользователь.

Для поля ввода реквизита «Количество» необходимо выбрать созданный метод `ТоварыЦенаПриИзменении` в качестве обработчика события «При изменении».

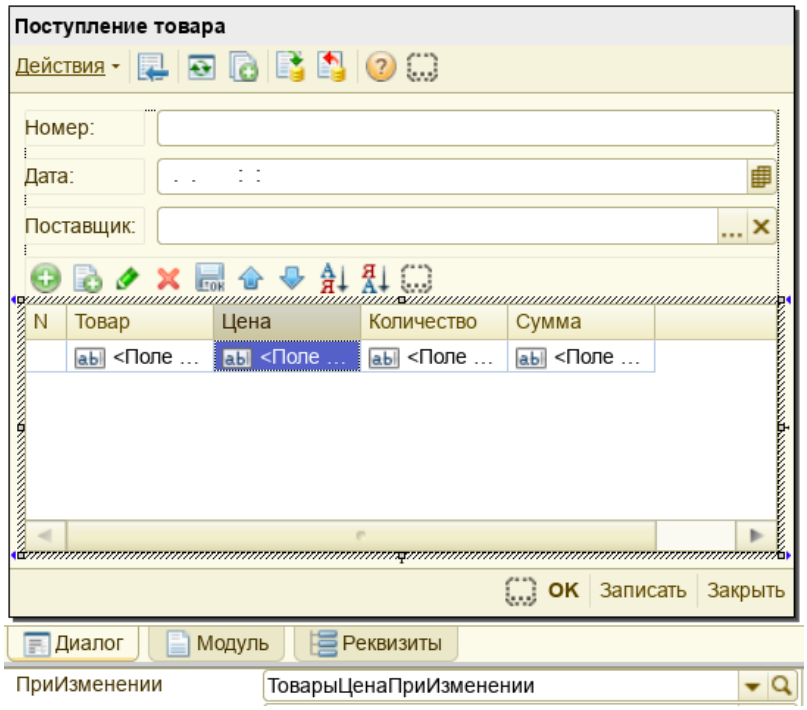


Рис. 6.6. Создание обработчика «При изменении» для поля ввода реквизита «Цена» табличной части «Товары»

А для поля ввода реквизита «Сумма» установить свойство «Только просмотр» в значение `Истина`. Аналогичные манипуляции необходимо произвести для документа «Возврат поставщику».

В интерактивном режиме в конфигурацию были добавлены следующие сведения:

1. Документ «Поступление товара» 000000001 от 01.01.2000 0:00:00:
   - Поставщик — Поставщик1;
   - Товар — Товар1, Цена — 100, Количество — 20.
2. Документ «Поступление товара» 000000002 от 01.01.2000 0:30:00:
   - Поставщик — Поставщик2;
   - Товар — Товар2, Цена — 340, Количество — 13.
3. Документ «Поступление товара» 000000003 от 01.01.2000 0:40:00:
   - Поставщик — Поставщик3;
   - Товар — Товар1, Цена — 110, Количество — 30;
   - Товар — Товар2, Цена — 400, Количество — 40.
4. Документ «Возврат товара» 000000001 от 01.01.2000 12:00:00:

- Поставщик — Поставщик2;
- Товар — Товар2, Цена — 340, Количество — 2.

5. Документ «Возврат товара» 000000001 от 01.02.2000 12:00:01:
   - Поставщик — Поставщик3;
   - Товар — Товар1, Цена — 110, Количество — 10.

Далее необходимо создать новый отчет с именем «ВедомостьПоступления», добавить в него макет с именем «Макет» и придать ему вид, определенный в варианте задания. Для текущего варианта задания макет будет иметь вид, представленный на рисунке 6.7.

Шапка	1						
	2	омость поступления товара от поставщиков с учетов возвратов за период с [ДатаНач] по [Дата					
	3						
	4		Поступило		Возвращено		
	5	Товар	Количество	Сумма	Количество	Сумма	% брака
	6						
Группа	7	<Поставщик>					
	8						
Строка	9	<Товар>	<ПКол>	<ПСум>	<ВКол>	<ВСум>	<Брак>
	10						
Итог	11	Итого	<ПКол>		<ВКол>		<Брак>

Рис. 6.7. Макет отчета «ВедомостьПоступления»

После создания макета необходимо создать форму отчета и добавить на нее компоненты (рис. 6.8): «Поле табличного документа» (см. панель компонентов внизу) и «Выбор периода» (пункт меню Форма -> Вставить элемент управления).

Содержимое обработчика `КнопкаСформироватьНажатие` события «Действие» кнопки «Сформировать» состоит из следующих блоков:

1. Создание таблицы значений со структурой для агрегации сведений, необходимых для формирования отчета. Названия колонок таблицы значений соответствуют названиям параметров макета для облегчения заполнения макета.

```
ТЗ = Новый ТаблицаЗначений();
ТЗ.Колонки.Добавить("Товар");
ТЗ.Колонки.Добавить("ПКол");
ТЗ.Колонки.Добавить("ПСум");
ТЗ.Колонки.Добавить("ВКол");
ТЗ.Колонки.Добавить("ВСум");
```

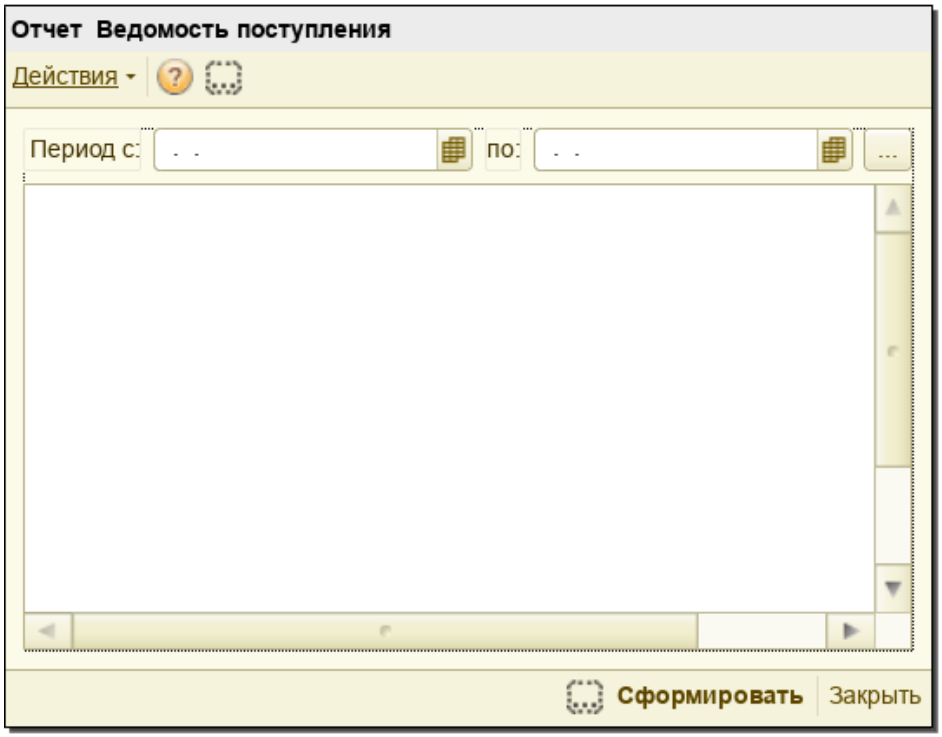


Рис. 6.8. Форма отчета «ВедомостьПоступления»

2. Перебор документов «Поступление товара» за указанный период и заполнение таблицы значений на основе содержимого табличной части «Товары» текущего документа. Значение периода берется из полей компонента «Выбор периода»: НачПериода и КонПериода соответственно.

```
ДокПост = Документы.ПоступлениеТовара
    .Выбрать(НачПериода, КонПериода);
Пока ДокПост.Следующий() Цикл
    // Пропускать непроведенные документы
    Если НЕ ДокПост.Проведен Тогда
        Продолжить;
    КонецЕсли;
    Для Каждого Товар Из ДокПост.Товары Цикл
        НовСтрока = ТЗ.Добавить();
        НовСтрока.Товар = Товар.Товар;
        НовСтрока.ПКол = Товар.Количество;
        НовСтрока.ПСум = Товар.Сумма;
    КонецЦикла;
КонецЦикла;
```

3. Перебор документов «Возврат товара» за указанный период и заполнение таблицы значений на основе содержимого табличной части «Товары» текущего документа. Значение

периода берется из полей компонента «Выбор периода»: НачПериода и КонПериода соответственно.

```
ДокВозврат = Документы.ВозвратТовара
    .Выбрать(НачПериода, КонПериода);
Пока ДокВозврат.Следующий() Цикл
    // Пропускать непроведенные документы
    Если НЕ ДокВозврат.Проведен Тогда
        Продолжить;
    КонецЕсли;
    Для Каждого Товар Из ДокВозврат.Товары Цикл
        НовСтрока = ТЗ.Добавить();
        НовСтрока.Товар = Товар.Товар;
        НовСтрока.ВКол = Товар.Количество;
        НовСтрока.ВСум = Товар.Сумма;
    КонецЦикла;
КонецЦикла;
```

После выполнения представленных выше команд содержимое таблицы значений ТЗ будет иметь вид, изображенный на рисунке 6.9.

Индекс	Значени…	Тип эле…	ВСум	ВКол	ПСум	ПКол	Товар
0	СтрокаТ…	СтрокаТ…			2,000	20	Товар1
1	СтрокаТ…	СтрокаТ…			3,300	30	Товар1
2	СтрокаТ…	СтрокаТ…			16,000	40	Товар2
3	СтрокаТ…	СтрокаТ…			4,420	13	Товар2
4	СтрокаТ…	СтрокаТ…	680	2			Товар2
5	СтрокаТ…	СтрокаТ…	1,100	10			Товар1

Рис. 6.9. Содержимое таблицы значений ТЗ

4. Свертка таблицы значений ТЗ.

```
ТЗ.Свернуть("Товар", "ПКол, ПСум, ВКол, ВСум");
```

После выполнения команды Свернуть содержимое таблицы значений ТЗ будет иметь вид, представленный на рисунке 6.10.

Индекс	Значени…	Тип эле…	ВСум	ВКол	ПСум	ПКол	Товар
0	СтрокаТ…	СтрокаТ…	1,100	10	5,300	50	Товар1
1	СтрокаТ…	СтрокаТ…	680	2	20,420	53	Товар2

Рис. 6.10. Содержимое таблицы значений ТЗ после свертки

5. Формирование отчета по содержимому таблицы значений.

```
ТабДок = ЭлементыФормы.ПолеТабличногоДокумента1;
ТабДок.Очистить();
Макет = Отчеты.ВедомостьПоступления
    .ПолучитьМакет("Макет");
ОбластьШапка = Макет.ПолучитьОбласть("Шапка");
ОбластьШапка.Параметры.ДатаНач =
    Формат(НачПериода, "ДФ=dd.MM.yyyy");
ОбластьШапка.Параметры.ДатаКон =
    Формат(КонПериода, "ДФ=dd.MM.yyyy");

ТабДок.Вывести(ОбластьШапка);
Если НЕ ТЗ.Количество() = 0 Тогда
    Для Каждого Стр Из ТЗ Цикл
        ОбластьСтрока = Макет
            .ПолучитьОбласть("Строка");
        ОбластьСтрока.Параметры.Заполнить(Стр);
        ОбластьСтрока.Параметры.Брак = Стр.ВКол * 100
            / Стр.ПКол;
        ТабДок.Вывести(ОбластьСтрока);
    КонецЦикла;
    ОбластьИтог = Макет.ПолучитьОбласть("Итог");
    ОбластьИтог.Параметры.ПКол = ТЗ.Итог("ПКол");
    ОбластьИтог.Параметры.ВКол = ТЗ.Итог("ВКол");
    ОбластьИтог.Параметры.Брак = ОбластьИтог
        .Параметры.ВКол * 100
        / ОбластьИтог.Параметры.ПКол;
    ТабДок.Вывести(ОбластьИтог);
КонецЕсли;

ТабДок.ТолькоПросмотр = Истина;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.ОтображатьСетку = Ложь;
```

После выполнения представленных выше команд сформированный отчет будет иметь вид, изображенный на рисунке 6.11.

Период с: 01.01.2000 по: 31.12.2000 ...

Ведомость поступления товара от поставщиков с учетов возвратов за период с 01.01.2000 по 31.12.2000

Товар	Поступило		Возвращено		% брака
	Количество	Сумма	Количество	Сумма	
Товар1	50	5 300	10	1 100	20,00
Товар2	53	20 420	2	680	3,77
Итого	103		12		11,65

Рис. 6.11. Сформированный отчет

Для группировки строк отчета по поставщикам необходимо внести следующие модификации в обработчик КнопкаСформироватьНажатие:

1. Добавить дополнительную колонку «Поставщик» в таблицу значений ТЗ.

```
ТЗ.Колонки.Добавить("Поставщик");
```

2. Заполнять колонку «Поставщик» таблицы значений ТЗ значениями при обходе документов.

```
...
НовСтрока.Поставщик = ДокПост.Поставщик;
...
НовСтрока.Поставщик = ДокВозврат.Поставщик;
...
```

3. Сворачивать таблицу значений ТЗ с учетом колонки «Поставщик».

```
ТЗ.Свернуть("Поставщик, Товар",
    "ПКол, ПСум, ВКол, ВСум");
```

4. Сортировать таблицу значений ТЗ по колонкам «Поставщик» и «Товар».

```
ТЗ.Сортировать("Поставщик, Товар");
```

5. Ввести переменную для хранения текущего значения сведений о поставщике перед циклом формирования отчета.

```
ТекПоставщик = Неопределено;
```

6. Добавить в цикл формирования отчета команды для вывода информации о текущей группировке.

```
Если НЕ ТекПоставщик = Стр.Поставщик Тогда
    ТекПоставщик = Стр.Поставщик;
    ОбластьГруппа = Макет.ПолучитьОбласть("Группа");
    ОбластьГруппа.Параметры.Поставщик = ТекПоставщик;
    ТабДок.Вывести(ОбластьГруппа);
КонецЕсли;
```

Таким образом, отчет с группировкой по поставщикам будет иметь вид, представленный на рисунке 6.12.

Нестандартную расшифровку отчета необходимо реализовать самостоятельно.



Ведомость поступления товара от поставщиков с учетов возвратов за период с 01.01.2000 по 31.12.2000

Товар	Поступило		Возвращено		% брака
	Количество	Сумма	Количество	Сумма	
Поставщик1					
Товар1	20	2 000			
Поставщик2					
Товар2	13	4 420	2	680	15,38
Поставщик3					
Товар1	30	3 300	10	1 100	33,33
Товар2	40	16 000			
Итого	103		12		11,65

Рис. 6.12. Отчет с группировкой по поставщикам

6.4 Контрольные вопросы

1. В чем разница между реквизитами шапки документа и реквизитами табличной части документа?
2. В каком случае лучше использовать реквизиты шапки документа?
3. В каком случае лучше использовать реквизиты табличной части документа?
4. Каким образом можно определить интервал выборки документов?
5. Каким образом можно определить фильтр выборки документов?

6.5 Варианты заданий

Лабораторная работа должна быть выполнена в режиме *обычного приложения*.

В представленных вариантах задания необходимо:

1. Разработать и заполнить справочники, необходимые для заполнения документов.
2. Разработать документы, обеспечивающие накопление информации, необходимой для формирования отчета.
3. С помощью разработанных документов подготовить исходные данные для формирования отчета.
4. Разработать отчет путем обращения к перечню документов:

- сформировать форму диалога отчета;
- разработать макет отчета;
- разработать модуль формирования отчета;
- обеспечить расшифровку ячеек отчета.

5. Выполнить проверку функциональности разработанного отчета.

Замечание: Для построения отчета, возможно, необходима предварительная сортировка и свертка данных, извлеченных из документов. Используйте для этих целей таблицу значений.

Вариант 1

Ведомость поступления товаров от поставщиков за <Месяц> <Год>

Поставщик	Товаров на сумму	НДС на сумму	Всего от поставщика на сумму	Адрес поставщика
Итого	<>	<>	<>	

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 2

Ведомость поступления товаров от поставщика <Поставщик> за период с <Дата> по <Дата>

Дата	Товар	Склад	Ед. изм.	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого				<>		<>	<>	<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 3

Ведомость поступления товаров на склад <Склад> от поставщиков за <Месяц> <Год>

Товар	Поставщик	Документ	Дата	Цена	Сумма
Итого					<>

Замечание: Строки отчета группируются по поставщикам. В конце группы следует включать строку «Итого по поставщику».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 4

Ведомость поступления товаров на склад <Склад> от поставщика <Поставщик> за период с <Дата> по <Дата>

Дата	Товар	Ед. изм.	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого			<>		<>	<>	<>

Замечание: Строки отчета группируются по товарам. В конце группы следует включать строку «Итого по товару».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 5

Ведомость поступления товара <Товар> на склады от поставщиков за период с <Дата> по <Дата>

Склад	Документ	Дата	Поставщик	Кол-во	Ед. изм.	Цена	Сумма
Итого				<>			<>

Замечание: Строки отчета группируются по поставщикам. В конце группы следует включать строку «Итого по поставщику».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 6

Ведомость поступления товара <Товар> на склады от поставщика <Поставщик> за период с <Дата> по <Дата>

Склад	Документ	Дата	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого			<>		<>	<>	<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 7

Ведомость продажи товаров покупателям за <Месяц> <Год>

Покупатель	Сумма	Сумма НДС	Всего с НДС
Итого	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 8

Ведомость продажи товаров покупателю <Покупатель> за период с <Дата> по <Дата>

Дата	Товар	Склад	Ед. изм.	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого				<>		<>	<>	<>

Замечание: Строки отчета группируются по товарам. В конце группы следует включать строку «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 9

Ведомость продажи товаров со склада <Склад> покупателям за <Месяц> <Год>

Товар	Покупатель	Валюта	Курс	Документ	Дата	Цена	Сумма
Итого							<>

Замечание: Строки отчета группируются по покупателям. В конце группы следует включать строку «Итого по покупателю».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 10

Ведомость продажи товаров со склада <Склад> покупателю <Покупатель> за период с <Дата> по <Дата>

Дата	Товар	Валюта	Курс	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого				<>		<>	<>	<>

Замечание: Строки отчета группируются по товарам. В конце группы следует включать строку «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 11

Ведомость продажи товара <Товар> со складов покупателям за период с <Дата> по <Дата>

Склад	Документ	Дата	Покупатель	Кол-во	Ед. изм.	Цена	Сумма
Итого				<>			<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 12

Ведомость продажи товара <Товара> со складов покупателю <Покупатель> за период с <Дата> по <Дата>

Склад	Документ	Дата	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого			<>		<>	<>	<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 13

Ведомость движения товаров за <Месяц> <Год>

Товар	Нач. остаток		Приход		Расход		Кон. остаток	
	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма
Итого	<>	<>	<>	<>	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 14

Ведомость движения товара <Товар> за <Месяц> <Год>

Склад	Начальный остаток	Приход	Расход	Конечный остаток
	Сумма	Сумма	Сумма	Сумма
Итого	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 15

Ведомость движения товаров на складе <Склад> за <Месяц> <Год>

Товар	Нач. остаток		Приход		Расход		Кон. остаток	
	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма
Итого	<>	<>	<>	<>	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 16

Ведомость движения товара <Товар> на складе <Склад> за <Месяц> <Год>

Дата	Документ	Номер	Приход		Расход	
			Кол-во	Сумма	Кол-во	Сумма
Итого			<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 17

Ведомость взаиморасчетов с поставщиками за <Месяц> <Год>

Клиент	Сумма долга на начало периода	Сумма прихода	Сумма расхода	Сумма долга на конец периода
Итого	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Оплата товаров поставщику».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 18

Справка о возвратах товаров от покупателей за период с <Дата> по <Дата>

Дата возврата	Документ	Покупатель	Сумма возврата	Причина возврата
Итого			<>	

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Возврат товаров поставщику».

Замечание: Использовать стандартную расшифровку.

Вариант 19

Ведомость взаиморасчетов с покупателем <Покупатель> за <Месяц> <Год>

Дата	Номер	Документ	Приход	Расход	Остаток
Итого			<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков», «Реализация продукции покупателям» и «Поступление оплаты от покупателей».

Замечание: Использовать стандартную расшифровку.

Вариант 20

Ведомость передачи на реализацию товаров реализаторам за <Месяц> <Год>

Реализатор	Документ	Дата	Товар	Кол-во	Сумма
Итого				<>	<>

Замечание: Данные группируются по реализаторам. В конце группы выводится строка «Итого по реализатору».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Передача товаров реализаторам».

Замечание: Использовать стандартную расшифровку.

Вариант 21

Ведомость продажи товаров реализатором <Реализатор> за период с <Дата> по <Дата>

Товар	Дата	Документ	Покупатель	Кол-во	Сумма
Итого				<>	<>

Замечание: Данные группируются по товарам. В конце группы выводится строка «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков», «Передача товаров реализаторам» и «Продажа товаров реализатором».

Замечание: Использовать стандартную расшифровку.

Вариант 22

Ведомость продажи товаров реализаторами за <Месяц> <Год>

Реализатор	Товар	Дата	Документ	Покупатель	Сумма
Итого					<>

Замечание: Данные группируются по реализаторам. В конце группы выводится строка «Итого по реализатору».

Замечание: Разработать документы «Поступление товаров от поставщиков», «Передача товаров реализаторам» и «Продажа товаров реализатором».

Замечание: Использовать стандартную расшифровку.

Вариант 23

Ведомость продажи товаров по заявкам клиентов за период с <Дата> по <Дата>

Клиент	Всего товаров в заявках на сумму	Продано по заявкам клиентов на сумму	Осталось продать на сумму
Итого	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков», «Заявка покупателю» и «Реализация по заявке».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 24

Ведомость продажи товаров по заявкам клиента <Клиент> за период с <Дата> по <Дата>

Товар	Заказано			Продано			Остаток	
	Док.	Кол-во	Сумма	Док.	Кол-во	Сумма	Кол-во	Сумма
Итого		<>	<>		<>	<>	<>	<>

Замечание: Данные группируются по товарам. В конце группы выводится строка «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков», «Заявка покупателю» и «Реализация по заявке».

Замечание: Использовать стандартную расшифровку.

7 РЕГИСТРЫ НАКОПЛЕНИЯ

Регистры накопления в рамках Платформы используются для учета движения сущностей, представляющих различные виды материальных, товарных, финансовых и других видов средств. Фактически, регистр накопления позволяет «накапливать» числовые данные в разрезе нескольких измерений, например, для автоматизации складского учета можно учитывать поступление и выбытие товаров в разрезе товаров и складов.

7.1 Создание и настройка регистра накопления

Для создания регистра накопления необходимо выбрать узел «Регистры накопления» в дереве объектов метаданных, а затем нажать на него правой кнопкой мыши и выбрать пункт «Добавить» в появившемся меню (рис. 7.1).

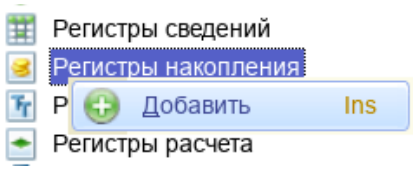


Рис. 7.1. Создание регистра накопления

На рисунке 7.2 представлено окно свойств вновь созданного регистра накопления. На вкладке «Основные» окна свойств регистра накопления можно выбрать вид регистра:

- остатки позволяет накапливать движения сущностей проблемной области и формировать итоги. Например, в качестве сущности будем использовать товары, в качестве движений — бизнес процессы поступления и реализации товаров, итогом в данном случае будет остаток конкретного товара на предприятии на определенную дату;
- обороты позволяет только накапливать движения сущностей проблемной области. Например, в качестве сущности будем использовать услуги, в качестве движений — бизнес процесс оказания услуги. В данном случе значение имеет только объем оказанных услуг.

На рисунке 7.3 представлено содержимое вкладки «Данные» регистра накопления.

Как видно из рисунка 7.3, данная вкладка позволяет создавать специфичные для регистров Платформы типы данных:

- измерения используются для формирования разрезов, в рамках которых происходит учет движений. Измерения можно рассматривать в виде осей координат многомерного пространства, на пересечении координат которых располагаются данные;
- ресурсы используются для хранения и накопления числовых данных движений. Ресурсы регистра накопления могут принимать только числовые значения;
- реквизиты используются для хранения дополнительной информации по движениям. Реквизиты не учитываются при расчете остатков и оборотов.

На рисунке 7.4 представлены стандартные реквизиты регистра накопления.

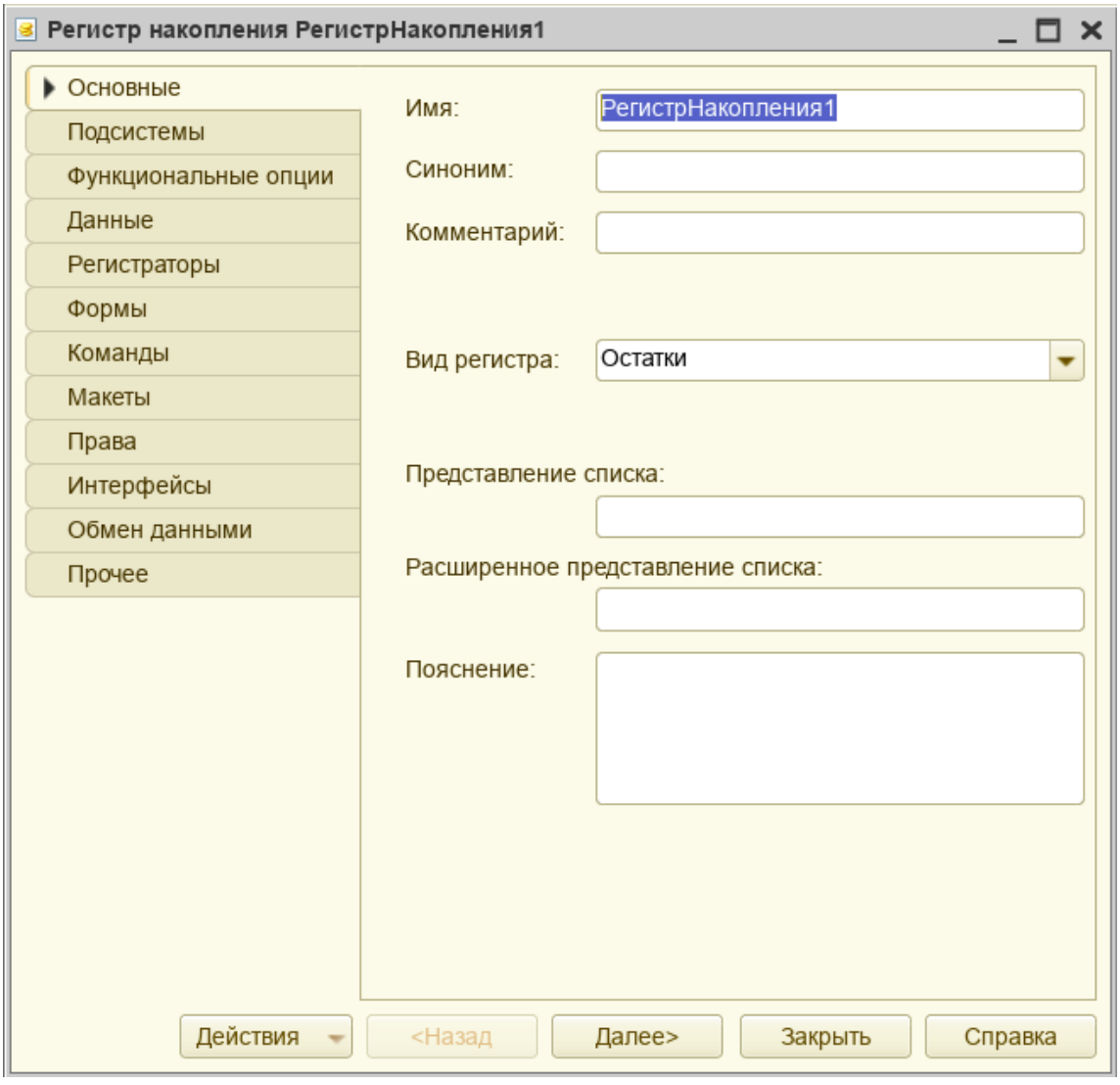


Рис. 7.2. Вкладка «Основные» окна свойств регистра накопления

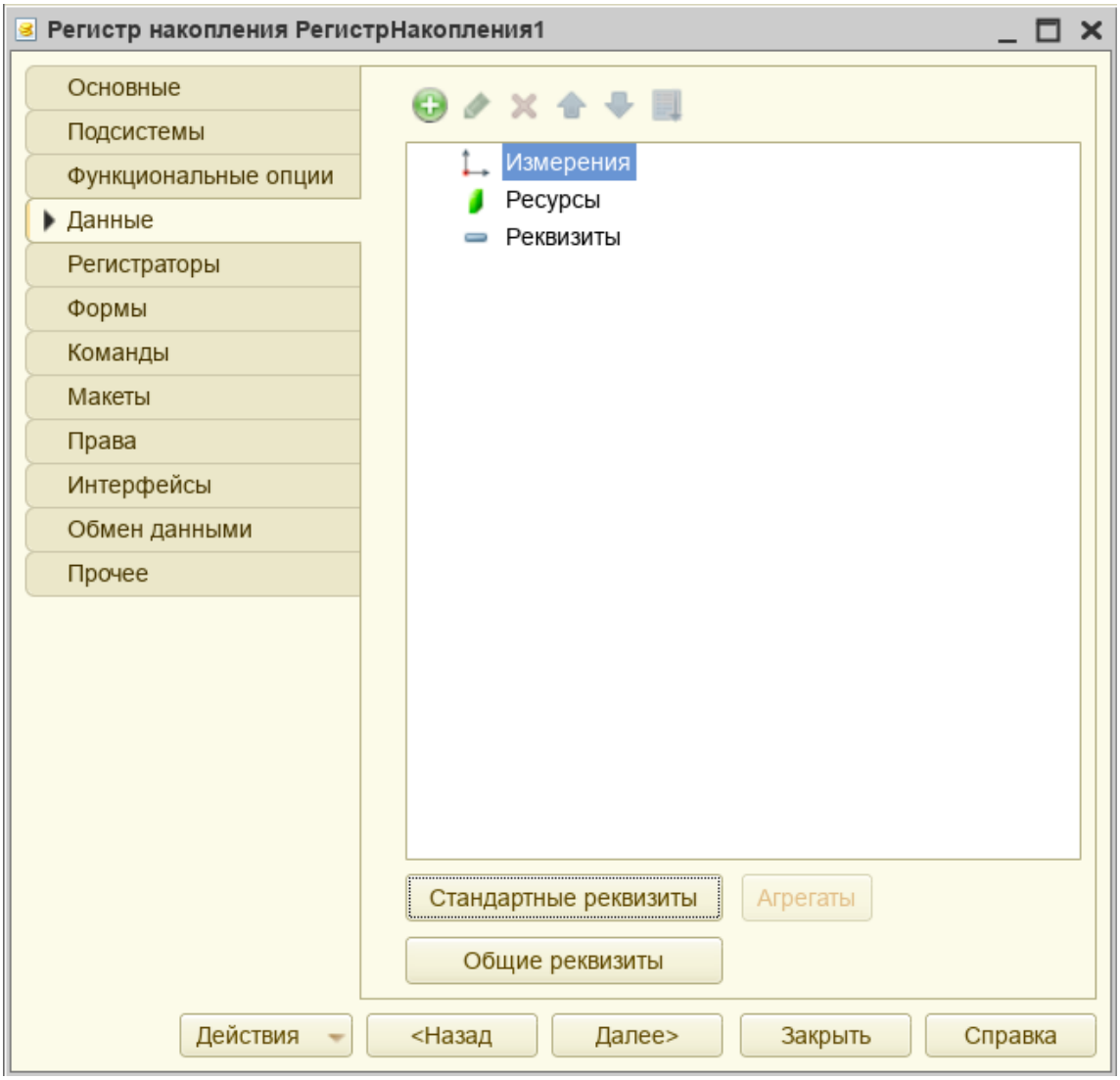


Рис. 7.3. Вкладка «Данные» окна свойств регистра накопления

Как видно из рисунка 7.4, к стандартным реквизитам регистра накопления относятся:

- Период — дата, связанная с движением. Обычно стандартный реквизит «Период» принимает значение даты регистратора, но иногда требуется установить другое значение периода;
- Регистратор — документ, при проведении которого было сформировано соответствующее движение. Обычно изменение состояния регистра накопления происходит при проведении некоторого документа;
- НомерСтроки — номер строки регистратора, на основании которой было сформировано движение;

- Активность позволяет «отключить» некоторые движения без необходимости отмены проведения регистратора;
- ВидДвижения — определяет вид движения: приход (+) или расход (-). Доступен только для регистров накопления с видом «Остатки». Стандартный реквизит «ВидДвижения» принимает значение системного перечисления «ВидДвиженияНакопления».

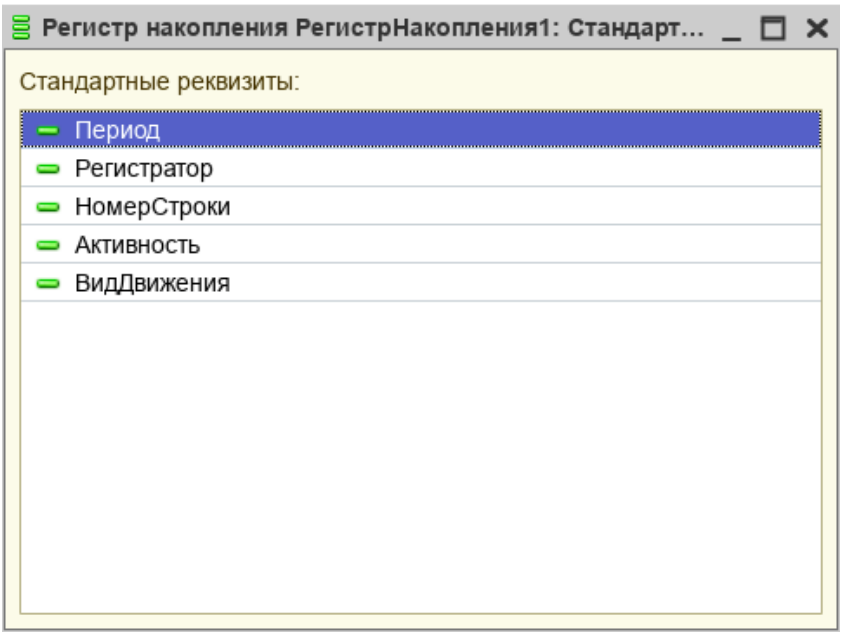


Рис. 7.4. Стандартные реквизиты регистра накопления

Платформа обеспечивает контроль уникальности записей, хранящихся в регистре накопления, на уровне измерений и стандартных реквизитов.

Для определения допустимых регистраторов регистра накопления используется вкладка «Регистраторы» окна свойств регистра накопления (рис. 7.5).

Осуществить «привязку» регистра накопления к регистратору можно на вкладе «Движения» соответствующего документа. Как видно из рисунка 7.6, существуют следующие параметры, определяющие поведение документа при его проведении или отмене проведения:

- Проведение — определяет возможность проведения документа при записи. Если параметр установлен в значение «Разрешить», то документ осуществляет формирование движений регистров;

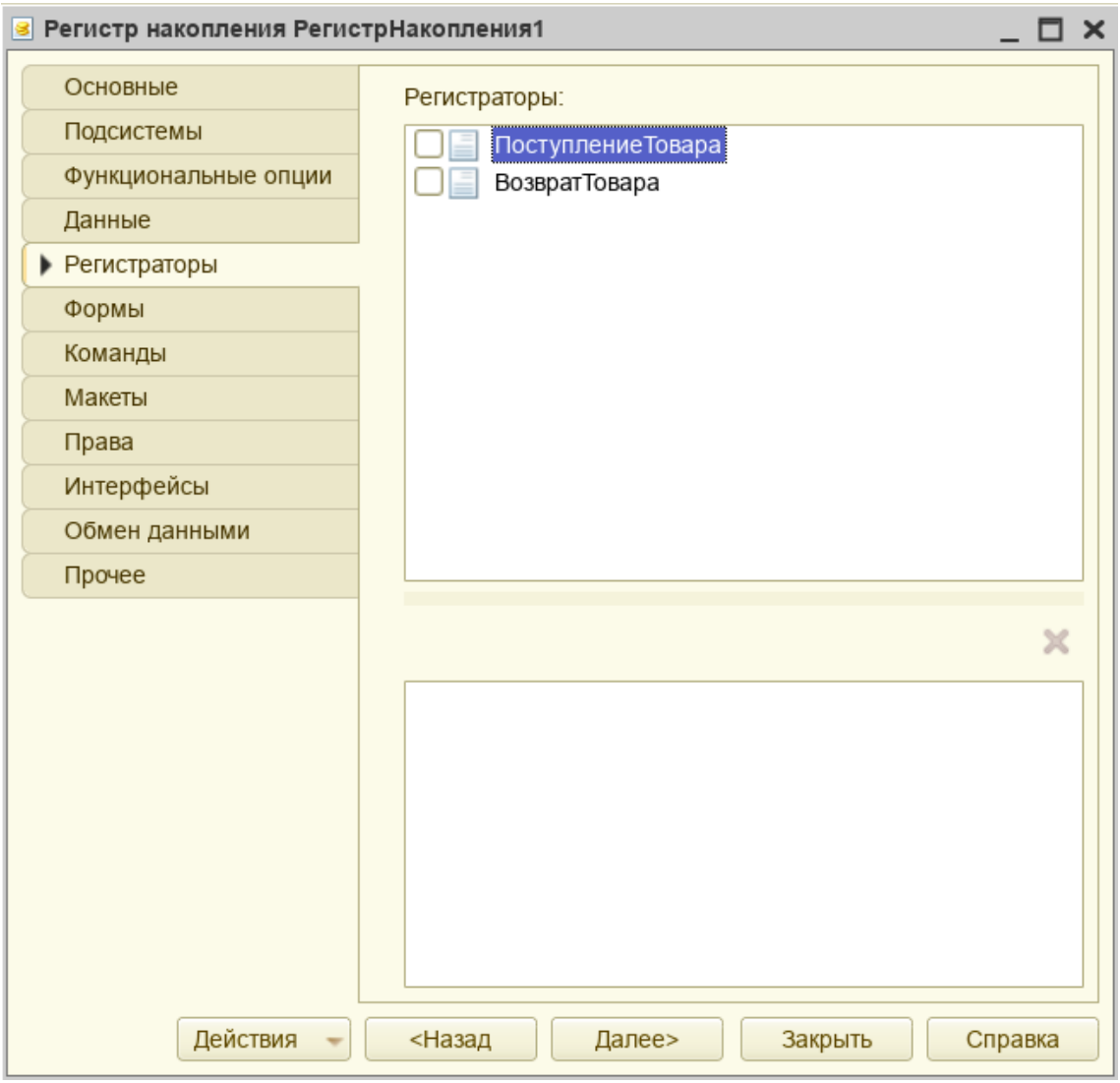


Рис. 7.5. Вкладка «Регистраторы» окна свойств регистра накопления

- Оперативное проведение — определяет возможность оперативного проведения документа. Если параметр установлен в значение «Разрешить» и дата документа не соответствует текущей дате, то проведение документа осуществляется в неоперативном режиме. При оперативном проведении система всегда автоматически устанавливает дату документа равной текущей дате. При оперативном проведении документа существует возможность контролировать текущие остатки регистров, например, при проведении расходной накладной нужно проверить, достаточно ли товара на складе. При неоперативном проведении документа в конфигурации не должно быть обращений к текущим итогам регистров;

- Удаление движений — определяет режим автоматического удаления движений документа при его повторном проведении или отмене проведения. Данное свойство доступно, если разрешено проведение документа.

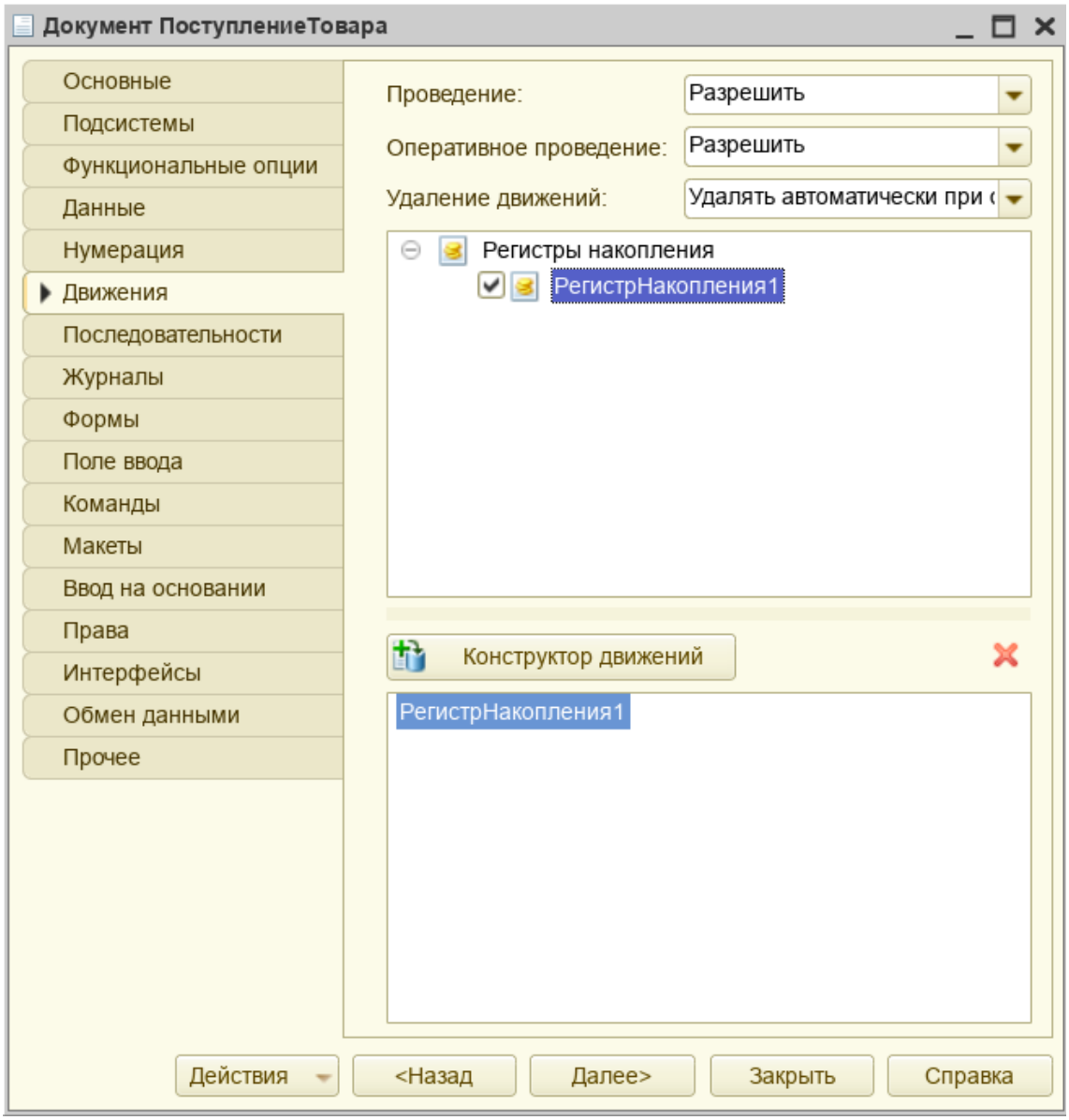


Рис. 7.6. Вкладка «Движения» окна свойств документа

7.2 Методы работы с регистрами накопления

Объект метаданных «Документ» используется для изменения состояния регистров. При проведении документа вызывается обработчик события `ОбработкаПроведения`, в котором на встроенном языке Платформы описывается алгоритм формирования

движений документа по определенному регистру. Обработчик события ОбработкаПроведения располагается в модуле объекта документа.

Алгоритм проведения документа, содержащийся в обработчике события ОбработкаПроведения, может быть сформирован с использованием конструктора движений (см. пример выполнения задания).

Программное проведение документа выполняется с помощью метода Записать объекта ДокументОбъект:

```
Документ = Документы.ПримерДокумента.СоздатьДокумент();
Документ.Дата = ТекущаяДата();
...
// Запись документа
Документ.Записать();
// Запись и проведение документа в неоперативном режиме
Документ.Записать(РежимЗаписиДокумента.Проведение,
    РежимПроведенияДокумента.Неоперативный);
```

При непосредственном удалении документа контроль за удалением движений документа возлагается на разработчика.

Для формирования движений регистра накопления при проведении документа (обработчик события ОбработкаПроведения), являющегося регистратором для данного регистра, используется объект РегистрНакопленияНаборЗаписей. Получить доступ к объекту РегистрНакопленияНаборЗаписей можно через свойство Движения с типом КоллекцияДвижений объекта ДокументОбъект:

```
// Включение автоматической записи набора записей
Движения.Регистр.Записывать = Истина;
// Удаление всех записей из набора
Движения.Регистр.Очистить();
Для Каждого Строка Из ТабличнаяЧастьДокумента Цикл
    // Добавление новой записи о движении регистра
    Движение = Движения.Регистр.Добавить();
    // Определение вида движения: Приход или Расход
    // Доступно только в регистрах накопления
    // с видом Остатки
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    // Дата документа
    Движение.Период = Дата;
    Движение.Измерение1 = Строка.Реквизит1;
    Движение.Ресурс1 = Строка.Реквизит2;
    Движение.Реквизит1 = Строка.Реквизит3;
КонецЦикла;
```

Для выбора всех записей регистра, у которых реквизит равен определенному значению, используется метод Выбрать объекта РегистрНакопленияМенеджер:

```
Выборка = РегистрыНакопления.Регистр.Выбрать();
Пока Выборка.Следующий() Цикл
    Если Выборка.Реквизит1 = ЗначениеФильтра Тогда
        Сообщить(Выборка.Реквизит2);
    КонецЕсли;
КонецЦикла;
```

В данном случае под реквизитом понимаются стандартные реквизиты, измерения, ресурсы и реквизиты регистра накопления.

Выборку записей регистра накопления также можно ограничить, используя параметры метода Выбрать объекта РегистрНакопленияМенеджер:

```
ДатаНач = НачалоГода(Дата(2000, 1, 1));
ДатаКон = КонецГода(Дата(2000, 1, 1));
Отбор = Новый Структура("Измерение1", ЗначениеФильтра);
Выборка = РегистрыНакопления.Регистр
    .Выбрать(ДатаНач, ДатаКон, Отбор);
Пока Выборка.Следующий() Цикл
    Сообщить(Выборка.Реквизит2);
КонецЦикла;
```

Для корректной работы отбора по измерению регистра накопления при использовании метода Выбрать объекта РегистрНакопленияМенеджер необходимо установить параметр «Индексировать» данного измерения в значение Индексировать.

Получение остатков регистра накопления на заданную дату осуществляется с помощью метода Остатки объекта РегистрНакопленияМенеджер:

```
Дата = ТекущаяДата();
Отбор = Новый Структура("Измерение1", ЗначениеФильтра);
Остатки = РегистрыНакопления.Регистр
    .Остатки(Дата, Отбор, "Измерение2", "Ресурс1");
Для Каждого Строка Из Остатки Цикл
    Сообщить(Строка.Измерение2.Наименование
        + ": " + Строка.Ресурс1);
КонецЦикла;
```

Функция Остатки позволяет формировать остатки с учетом отбора по значениям измерений. Результатом функции Остатки

является таблица значений, колонки которой соответствуют измерениям, указанным в параметре Измерения функции, а также ресурсам, указанным в параметре Ресурсы. Функция Остатки работает только для регистров накопления с видом «Остатки».

Для получения оборотов регистра накопления за заданный период используется метод Обороты объекта РегистрНакопленияМенеджер:

```
ДатаНач = НачалоГода(Дата(2000, 1, 1));
ДатаКон = КонецГода(Дата(2000, 1, 1));
Отбор = Новый Структура("Измерение1", ЗначениеФильтра);
Обороты = РегистрыНакопления.Регистр
    .Обороты(ДатаНач, ДатаКон, Отбор,
                "Измерение2", "Ресурс1");
Для Каждого Строка Из Обороты Цикл
    Сообщить(Строка.Измерение2.Наименование
        + ": " + Строка.Ресурс1Приход
        + ", " + Строка.Ресурс1Расход);
КонецЦикла;
```

Функция Обороты позволяет формировать обороты с учетом отбора по значениям измерений. Результатом функции Обороты является таблица значений, колонки которой соответствуют измерениям, указанным в параметре Измерения функции, а также ресурсам, указанным в параметре Ресурсы. Для регистров накопления с видом «Остатки» для каждого ресурса, указанного в параметре Ресурсы, создается по две колонки: <ИмяРесурса>Приход и <ИмяРесурса>Расход, а для регистров накопления с видом «Обороты» одна колонка — <ИмяРесурса>.

7.3 Пример выполнения задания

В рамках четвертой лабораторной работы необходимо научиться работать с объектом метаданных «РегистрНакопления». Для это требуется выполнить задание, состоящее из следующих пунктов:

1. Разработать структуру регистра накопления для хранения движений, формируемых при проведении документов.
2. Разработать алгоритм формирования движений регистра накопления при проведении документов.
3. Разработать отчет путем обращения к движениям регистра накопления.

4. Выполнить проверку функциональности разработанного отчета.

Замечание: Для построения отчета, возможно, необходима предварительная сортировка и свертка данных, извлеченных из регистра накопления. Используйте для этих целей таблицу значений.

Рассмотрим процесс выполнения следующего варианта задания, в рамках которого необходимо разработать отчет:

Ведомость поступления товара от поставщиков с учетов возвратов за период с <Дата> по <Дата>

Товар	Поступило		Возвращено		% брака
	Количество	Сумма	Количество	Сумма	
Итого	<>		<>		<>

Замечание: Разработать документы «Поступление товара» и «Возврат поставщику».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Лабораторная работа должна быть выполнена в режиме *обычного приложения*.

Для выполнения задания нет необходимости создавать документы «ПоступлениеТовара» и «ВозвратПоставщику», так как они были созданы в рамках третьей лабораторной работы.

В задании сказано, что необходимо разработать структуру регистра накопления для хранения движений, формируемых при проведении документов «ПоступлениеТовара» и «ВозвратПоставщику».

Таким образом, содержимое вкладки «Данные» регистра накопления «Товары» с видом регистра «Остатки» будет иметь вид, представленный на рисунке 7.7.

Как видно из рисунка 7.7, регистр накопления «Товары» содержит:

1. Измерения:
    - Товар — СправочникСсылка.Товары;
    - Поставщик — СправочникСсылка.Контрагенты.
2. Ресурсы:
    - Количество — Число(10, 0, Неотрицательное);
    - Сумма — Число(10, 2, Неотрицательное).

3. Реквизиты:
    - ПричинаВозврата — Строка(100).

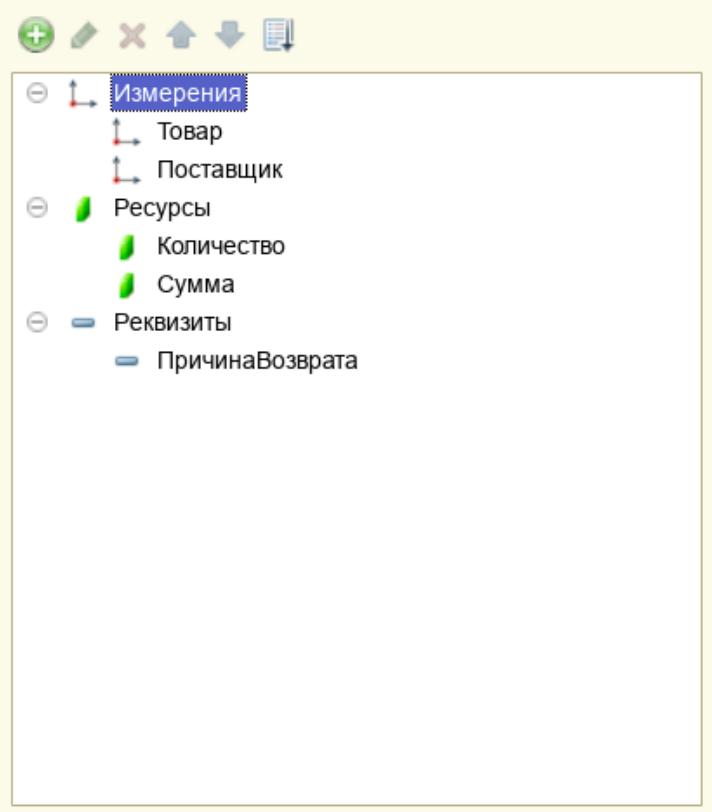


Рис. 7.7. Вкладка «Данные» регистра накопления «Товары»

В качестве регистраторов регистра накопления «Товары» указаны документы «ПоступлениеТовара» и «ВозвратПоставщику» (рис. 7.8).

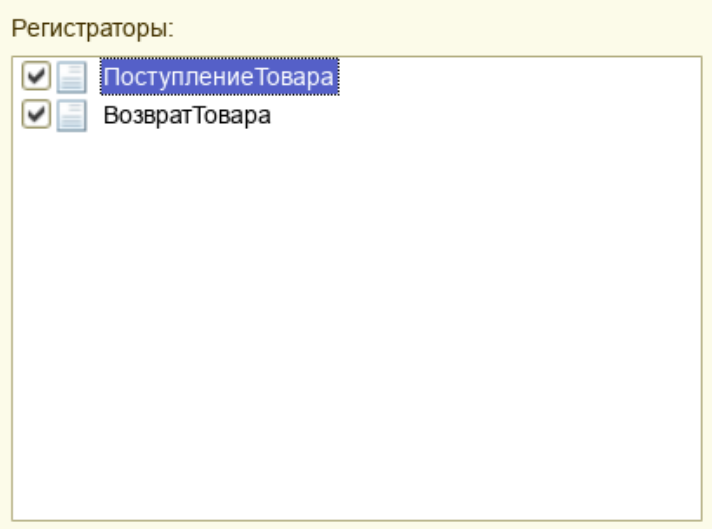


Рис. 7.8. Вкладка «Регистраторы» регистра накопления «Товары»

Для заполнения реквизита «ПричинаВозврата» регистра накопления «Товары» необходимо добавить в документ «ВозвратПоставщику» реквизит документа «ПричинаВозврата» с типом `Строка(100)`. Затем необходимо добавить поле ввода для реквизита «ПричинаВозврата» на форму документа «ВозвратПоставщику» (рис. 7.9).

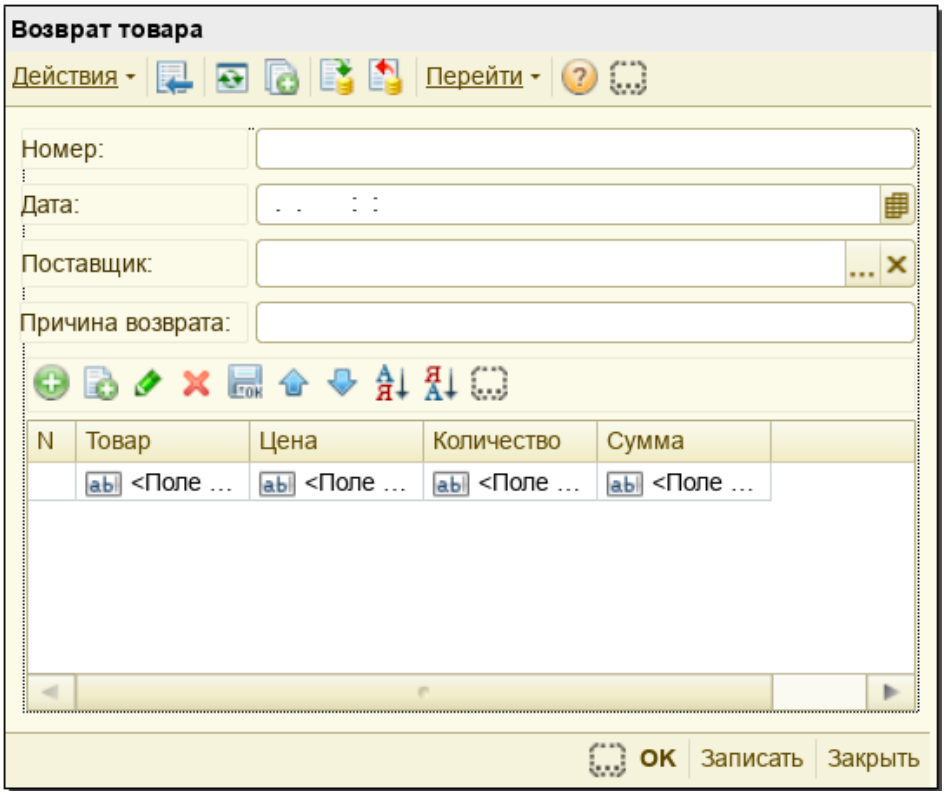


Рис. 7.9. Форма документа «ВозвратПоставщику»

Алгоритмы формирования движений регистра накопления при проведении документов «ПоступлениеТовара» и «ВозвратПоставщику» можно получить с помощью конструктора движений. Для открытия конструктора движений необходимо перейти на вкладку «Движения» окна свойств соответствующего документа и нажать на кнопку «Конструктор движений».

Конструктор движений позволяет в графическом режиме сопоставить данные документа с данными регистра. На рисунках 7.10 и 7.11 представлены конструкторы движений документов «ПоступлениеТовара» и «ВозвратПоставщику» соответственно.

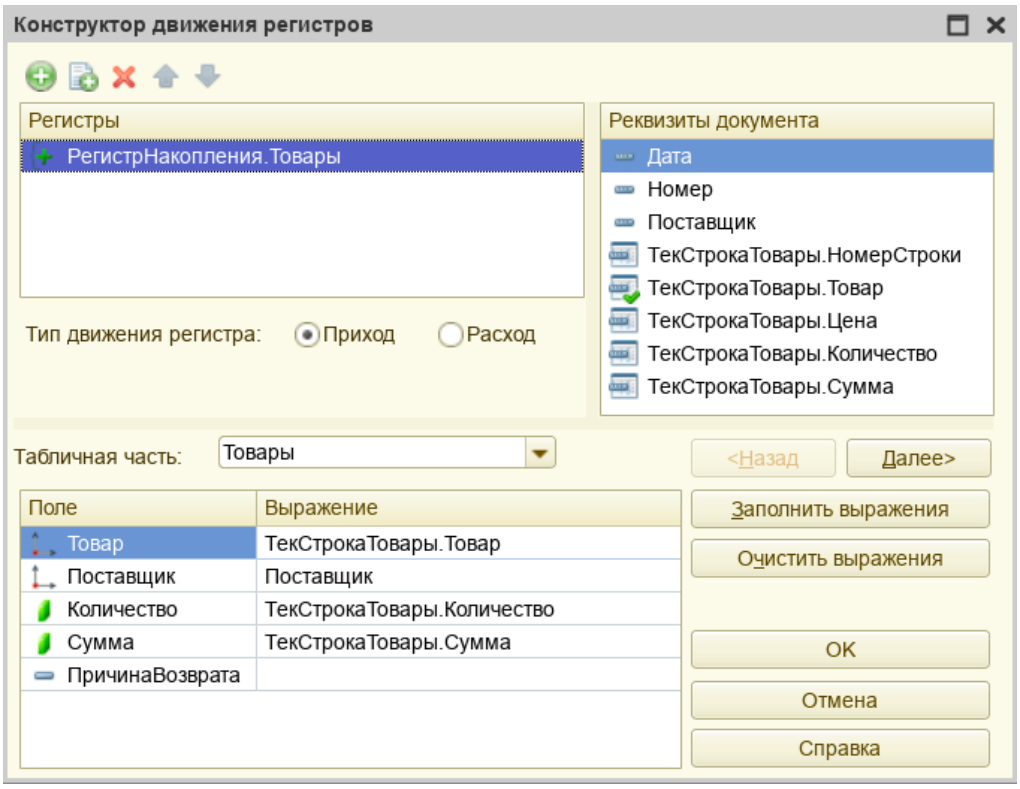


Рис. 7.10. Конструктор движений документа «ПоступлениеТовара»

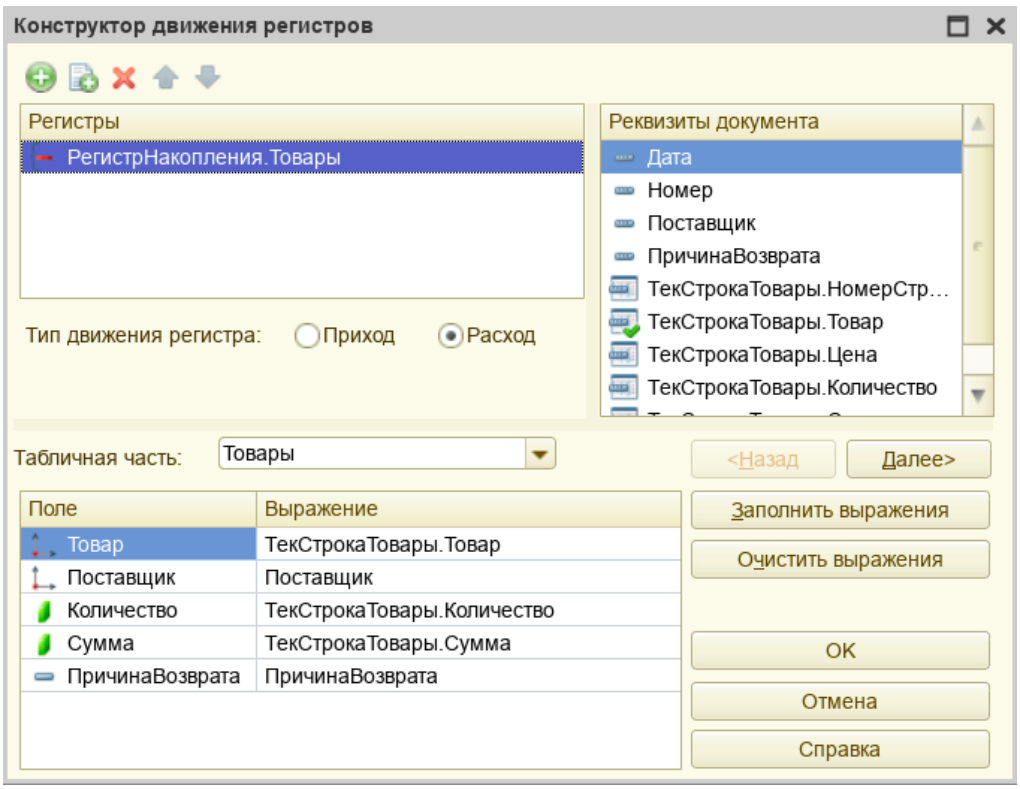


Рис. 7.11. Конструктор движений документа «ВозвратПоставщику»

Для документа «ПоступлениеТовара» обработчик события ОбработкаПроведения будет иметь вид:

```
Процедура ОбработкаПроведения(Отказ, Режим)
    Движения.Товары.Записывать = Истина;
    Движения.Товары.Очистить();
    Для Каждого ТекСтрокаТовары Из Товары Цикл
        Движение = Движения.Товары.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления
            .Приход;
        Движение.Период = Дата;
        Движение.Товар = ТекСтрокаТовары.Товар;
        Движение.Поставщик = Поставщик;
        Движение.Количество = ТекСтрокаТовары.Количество;
        Движение.Сумма = ТекСтрокаТовары.Сумма;
    КонецЦикла;
КонецПроцедуры
```

Обработчик события ОбработкаПроведения документа «ВозвратПоставщику» будет следующим:

```
Процедура ОбработкаПроведения(Отказ, Режим)
    Движения.Товары.Записывать = Истина;
    Движения.Товары.Очистить();
    Для Каждого ТекСтрокаТовары Из Товары Цикл
        Движение = Движения.Товары.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления
            .Расход;
        Движение.Период = Дата;
        Движение.Товар = ТекСтрокаТовары.Товар;
        Движение.Поставщик = Поставщик;
        Движение.Количество = ТекСтрокаТовары.Количество;
        Движение.Сумма = ТекСтрокаТовары.Сумма;
        Движение.ПричинаВозврата = ПричинаВозврата;
    КонецЦикла;
КонецПроцедуры
```

После добавления в документы «ПоступлениеТовара» и «ВозвратПоставщику» обработчиков события ОбработкаПроведения для формирования движений регистра накопления «Товары», данные документы необходимо перепровести в режиме «1С: Предприятие». Для перепроведения документов «ПоступлениеТовара» и «ВозвратПоставщику» необходимо выбрать пункт меню «Операции → Проведение документов». В открывшемся окне (рис. 7.12) необходимо нажать на кнопку «Выполнить».

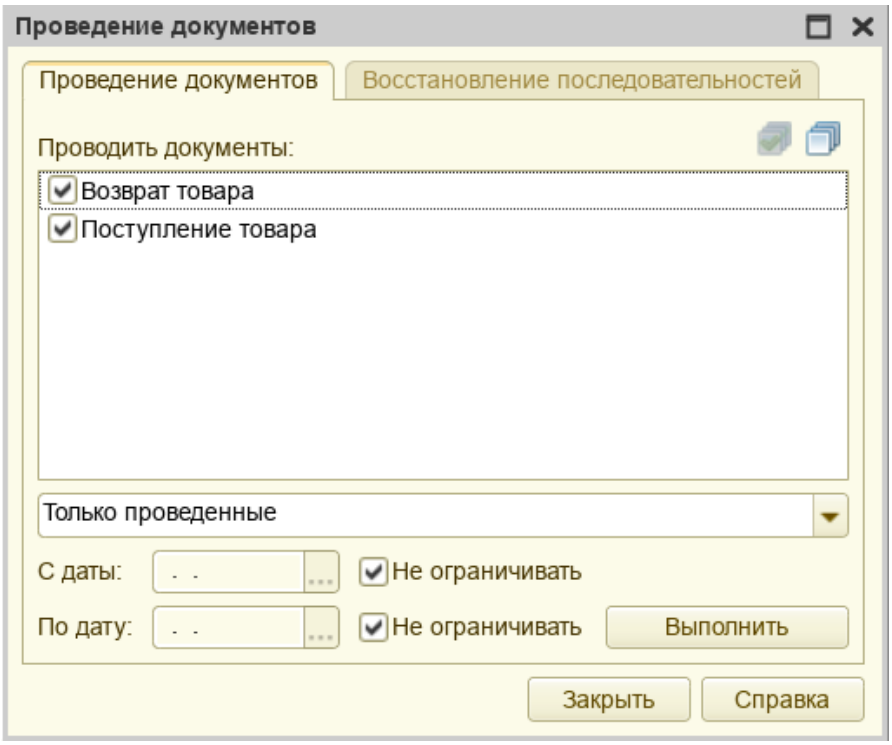


Рис. 7.12. Перепроведение документов

После перепровдения документов в регистре накопления будут сформированы движения. Содержимое регистра накопления «Товары» представлено на рисунке 7.13.

	Период	Регистратор	Ном…	Актив…	Товар	Поставщик	Количес…	Сумма	Причина возврата
+	01.01.2000 0:00:00	Поступление…	1	✔	Товар1	Поставщик1	20	2 000,00	
+	01.01.2000 0:30:00	Поступление…	1	✔	Товар2	Поставщик2	13	4 420,00	
+	01.01.2000 0:40:00	Поступление…	1	✔	Товар1	Поставщик3	30	3 300,00	
+	01.01.2000 0:40:00	Поступление…	2	✔	Товар2	Поставщик3	40	16 000,00	
−	01.02.2000 12:00:00	Возврат това…	1	✔	Товар2	Поставщик2	2	680,00	Брак
−	01.02.2000 12:00:01	Возврат това…	1	✔	Товар1	Поставщик3	10	1 100,00	Брак

Рис. 7.13. Содержимое регистра накопления «Товары»

Отчет, формирование которого будет осуществляться на основе движений регистра накопления «Товары», можно создать путем копирования отчета «ВедомостьПоступления». Отчет «ВедомостьПоступления» получен в результате выполнения третьей лабораторной работы.

Для копирования отчета «ВедомостьПоступления» необходимо выбрать соответствующий узел дерева объектов метаданных, нажать него правой кнопкой мыши, а затем выбрать пункт меню «Скопировать».

Далее необходимо изменить свойства «Имя» и «Синоним», указав для них значения «ВедомостьПоступленияРег» и «Ведомость поступления (регистр)» соответственно.

Содержимое обработчика КнопкаСформироватьНажатие события «Действие» кнопки «Сформировать» формы отчета «ВедомостьПоступленияРег» состоит из следующих блоков:

1. Создание таблицы значений со структурой для агрегации сведений, необходимых для формирования отчета. Названия колонок таблицы значений соответствуют названиям параметров макета для облегчения заполнения макета.

```
ТЗ = Новый ТаблицаЗначений();
ТЗ.Колонки.Добавить("Товар");
ТЗ.Колонки.Добавить("ПКол");
ТЗ.Колонки.Добавить("ПСум");
ТЗ.Колонки.Добавить("ВКол");
ТЗ.Колонки.Добавить("ВСум");
```

2. Перебор движений регистра «Товары» за указанный период и заполнение таблицы значений с учетом значения стандартного реквизита «ВидДвижения». Значение периода берется из полей компонента «Выбор периода»: НачПериода и КонПериода соответственно.

```
Регистр = РегистрыНакопления
    .Товары.Выбрать(НачПериода, КонПериода);
Пока Регистр.Следующий() Цикл
    НовСтрока = ТЗ.Добавить();
    НовСтрока.Поставщик = Регистр.Поставщик;
    НовСтрока.Товар = Регистр.Товар;
    Если Регистр.ВидДвижения =
        ВидДвиженияНакопления.Приход Тогда
        НовСтрока.ПКол = Регистр.Количество;
        НовСтрока.ПСум = Регистр.Сумма;
    Иначе
        НовСтрока.ВКол = Регистр.Количество;
        НовСтрока.ВСум = Регистр.Сумма;
    КонецЕсли;
КонецЦикла;
```

3. Свертка таблицы значений ТЗ.

```
ТЗ.Свернуть("Товар", "ПКол, ПСум, ВКол, ВСум");
```

4. Сортировка таблицы значений ТЗ по колонкам «Поставщик» и «Товар».

```
ТЗ.Сортировать("Поставщик, Товар");
```

5. Формирование отчета по содержимому таблицы значений.

```
ТабДок = ЭлементыФормы.ПолеТабличногоДокумента1;
ТабДок.Очистить();

Макет = Отчеты.ВедомостьПоступления
    .ПолучитьМакет("Макет");

ОбластьШапка = Макет.ПолучитьОбласть("Шапка");
ОбластьШапка.Параметры.ДатаНач =
    Формат(НачПериода, "ДФ=dd.ММ.уууу");
ОбластьШапка.Параметры.ДатаКон =
    Формат(КонПериода, "ДФ=dd.ММ.уууу");
ТабДок.Вывести(ОбластьШапка);

Если НЕ ТЗ.Количество() = 0 Тогда
    ТекПоставщик = Неопределено;
    Для Каждого Стр Из ТЗ Цикл
        Если НЕ ТекПоставщик = Стр.Поставщик Тогда
            ТекПоставщик = Стр.Поставщик;
            ОбластьГруппа = Макет
                .ПолучитьОбласть("Группа");
            ОбластьГруппа.Параметры.Поставщик =
                ТекПоставщик;
            ТабДок.Вывести(ОбластьГруппа);
        КонецЕсли;
        ОбластьСтрока = Макет
            .ПолучитьОбласть("Строка");
        ОбластьСтрока.Параметры.Заполнить(Стр);
        ОбластьСтрока.Параметры.Брак = Стр.ВКол * 100
            / Стр.ПКол;
        ТабДок.Вывести(ОбластьСтрока);
    КонецЦикла;

    ОбластьИтог = Макет.ПолучитьОбласть("Итог");
    ОбластьИтог.Параметры.ПКол = ТЗ.Итог("ПКол");
    ОбластьИтог.Параметры.ВКол = ТЗ.Итог("ВКол");
    ОбластьИтог.Параметры.Брак = ОбластьИтог
        .Параметры.ВКол * 100
        / ОбластьИтог.Параметры.ПКол;
    ТабДок.Вывести(ОбластьИтог);
КонецЕсли;

ТабДок.ТолькоПросмотр = Истина;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.ОтображатьСетку = Ложь;
```

После выполнения представленных выше команд сформированный отчет будет иметь вид, изображенный на рисунке 7.14.

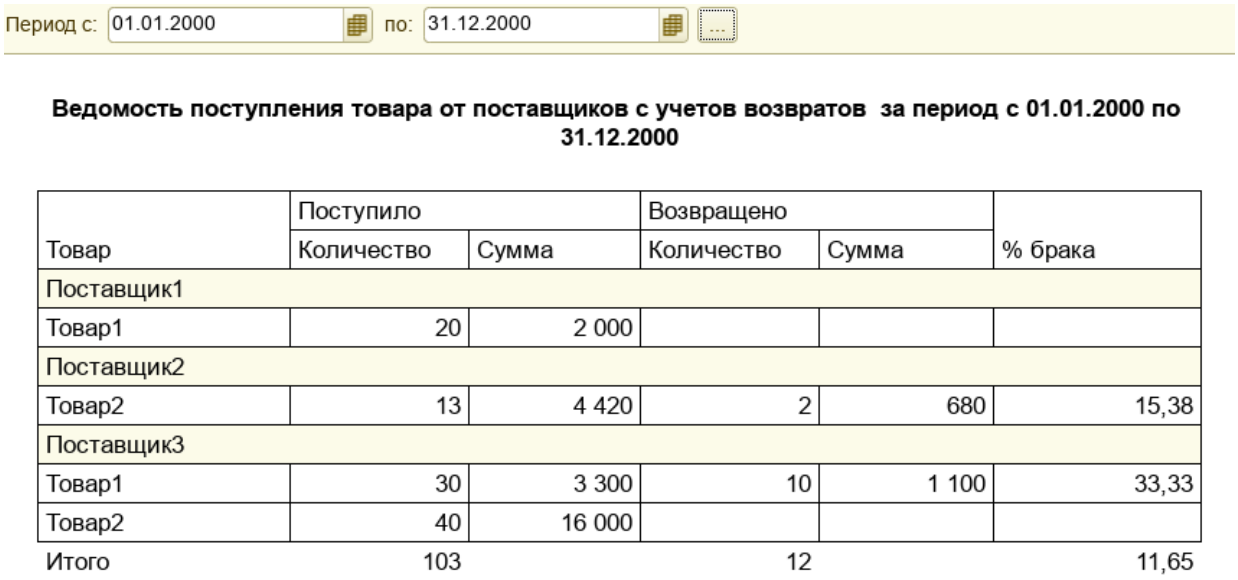


Рис. 7.14. Сформированный отчет

Как видно из рисунка 7.14, содержимое печатной формы отчета «ВедомостьПоступленияРег» идентично содержимому печатной формы отчета «ВедомостьПоступления», разработанного в рамках третьей лабораторной работы. При этом, для формирования отчета требуется обращение только к одному объекту — регистру накопления «Товары», вместо обращения к документам «ПоступлениеТовара» и «ВозвратПоставщику».

7.3.1 Автоматическое заполнение цены товара при возврате поставщику

В данном разделе будет рассмотрено применение регистра накопления для автоматизации заполнения цены товара при его возврате поставщику. Будем рассчитывать цену товара по среднему, так как это самый простой способ.

Для расчета цены товара по среднему необходимо найти среднюю стоимость имеющегося товара, разделив общее количество данного товара в остатке на его общую стоимость. Например, было введено два документа «ПоступлениеТовара», в которых указан товар «Товар1». В первом документе поступило 10 единиц товара «Товар1» по 10 рублей за единицу, а во втором 20 единиц по 15 рублей. Тогда средняя стоимость единицы товара «Товар1» будет (10 * 10 + 20 * 15) / (10 + 20) = 13.3.

Для автоматического заполнения цены товара при его выборе необходимо добавить в форму документа «ВозвратПоставщику» обработчик события «При изменении» для поля ввода реквизита «Товар» табличной части «Товары». Содержимое обработчика имеет вид:

```
Процедура ТоварыТоварПриИзменении(Элемент)
    ТС = ЭлементыФормы.Товары.ТекущаяСтрока;
    ТС.Цена = ЭтотОбъект
        .ПолучитьСреднююЦенуТовара(ТС.Товар);
КонецПроцедуры
```

Функция ПолучитьСреднююЦенуТовара реализует метод расчета цены товара по среднему и располагается в модуле объекта документа «ВозвратПоставщику»:

```
Функция ПолучитьСреднююЦенуТовара(Товар) Экспорт
    // Проверка на заполнение реквизита Поставщик
    Если НЕ ЗначениеЗаполнено(Поставщик) Тогда
        Сообщить("Поле 'Поставщик' не заполнено");
        Возврат 0;
    КонецЕсли;

    // Получение таблицы остатков регистра
    // накопления Товары с отбором
    // по товару и поставщику
    Отбор = Новый Структура("Поставщик, Товар");
    Отбор.Поставщик = Поставщик;
    Отбор.Товар = Товар;
    Остатки = РегистрыНакопления.Товары
        .Остатки(МоментВремени(), Отбор, ,
            "Количество, Сумма");

    Если Остатки.Количество() = 0 Тогда
        Возврат 0;
    КонецЕсли;

    // Получение первой строки таблицы остатков
    // Либо строк нет, либо всегда одна строка
    Остаток = Остатки.Получить(0);
    // Расчет цены товара методом по среднему
    Возврат ?(Остаток.Количество = 0,
        0,
        Остаток.Сумма / Остаток.Количество);
КонецФункции
```

Для возможности вызова функции ПолучитьСреднююЦенуТовара вне контекста модуля документа к ее определению добавлен оператор Экспорт.

7.3.2 Контроль остатков при возврате товара поставщику

Возможности регистра накопления также можно использовать для контроля остатков. Обычно под контролем остатка понимают предотвращение ситуации, когда, например, можно продать товара больше, чем его есть в остатках.

Для контроля остатков при проведении документа «ВозвратПоставщику» используется функция ПроверитьОстаткиТоваров:

```
Функция ПроверитьОстаткиТоваров()
    // Выгрузка табличной части Товары
    // в таблицу значений
    // Выгружаются только колонки Товар и Количество
    ТЗТовары = ЭтотОбъект.Товары
        .Выгрузить(, "Товар, Количество");
    // Свертка таблицы значений по колонке Товар,
    // колонка Количество суммируется
    ТЗТовары.Свернуть("Товар", "Количество");
    // Выгрузка колонки Товар таблицы значений
    // в массив
    ТоварыМассив = ТЗТовары.ВыгрузитьКолонку("Товар");

    // Формирование отбора по поставщику
    // и массиву товаров
    Отбор = Новый Структура("Поставщик, Товар");
    Отбор.Поставщик = Поставщик;
    Отбор.Товар = ТоварыМассив;
    // Формирование таблицы остатков
    Остатки = РегистрыНакопления.Товары.Остатки(
        // Остатки берутся на момент времени
        // точно перед текущим документом
        Новый Граница(МоментВремени(),
            ВидГраницы.Исключая),
        Отбор, , "Количество");
    // Очистка массива товаров
    ТоварыМассив.Очистить();

    // Контроль остатков
    Для Каждого Строка Из ТЗТовары Цикл
        // Поиск товара в таблице остатков
        Остаток = Остатки.Найти(Строка.Товар, "Товар");
```

```
            // Если такого товара в таблице остатков нет,
            // выводится сообщение, проверка провалена
            Если Остаток = Неопределено Тогда
                Сообщить("Товар ’" + Строка.Товар.Наименование
                    + "’ отсутствует");
                Возврат Ложь;
            КонецЕсли;
            // Если остатки по товару меньше, чем необходимо,
            // выводится сообщение, проверка провалена
            Если Остаток.Количество < Строка.Количество Тогда
                Сообщить("Недостаточно товара ’"
                    + Строка.Товар.Наименование
                    + "’, в наличии " + Остаток.Количество
                    + ", необходимо " + Строка.Количество);
                Возврат Ложь;
            КонецЕсли;
        КонецЦикла;

        // Контроль остатков успешно пройден
        Возврат Истина;
    КонецФункции
```

Для вызова функции контроля остатков `ПроверитьОстаткиТоваров` необходимо в обработчик проведения `ОбработкаПроведения` документа «ВозвратПоставщику» добавить следующие строки кода:

```
// Контроль остатков проводится только при
// оперативном проведении документа
Если Режим = РежимПроведенияДокумента.Оперативный
    // Если контроль остатков не пройден,
    // проведение прерывается
    И НЕ ПроверитьОстаткиТоваров() Тогда
    Отказ = Истина;
    Возврат;
КонецЕсли;
```

Для корректной работы функции `ПроверитьОстаткиТоваров` необходимо установить на вкладке «Движения» окна свойств документа «ВозвратПоставщику» значение свойства «Удаление движений» в значение «Удалять автоматически».

7.4 Контрольные вопросы

1. Какие виды регистров накопления бывают и в чем разница между ними?

2. Для чего используются измерения регистров накопления?
3. Для чего используются ресурсы регистров накопления?
4. Для чего используются реквизиты регистров накопления?
5. Какие функции для работы с содержимым регистра накопления существуют и в чем разница между ними?

7.5 Варианты заданий

Лабораторная работа должна быть выполнена в режиме *обычного приложения*.

В представленных вариантах задания необходимо:

1. Разработать структуру регистра накопления для хранения движений, формируемых при проведении документов (см. варианты заданий).
2. Разработать алгоритм формирования движений регистра накопления при проведении документов.
3. Разработать отчет путем обращения к движениям регистра накопления (см. варианты заданий).
4. Выполнить проверку функциональности разработанного отчета.

Замечание: Для построения отчета, возможно, необходима предварительная сортировка и свертка данных, извлеченных из регистра накопления. Используйте для этих целей таблицу значений.

Вариант 1

Ведомость поступления товаров от поставщиков за <Месяц> <Год>

Поставщик	Товаров на сумму	НДС на сумму	Всего от поставщика на сумму	Адрес поставщика
Итого	<>	<>	<>	

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 2

Ведомость поступления товаров от поставщика <Поставщик> за период с <Дата> по <Дата>

Дата	Товар	Склад	Ед. изм.	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого				<>		<>	<>	<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 3

Ведомость поступления товаров на склад <Склад> от поставщиков за <Месяц> <Год>

Товар	Поставщик	Документ	Дата	Цена	Сумма
Итого					<>

Замечание: Строки отчета группируются по поставщикам. В конце группы следует включать строку «Итого по поставщику».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 4

Ведомость поступления товаров на склад <Склад> от поставщика <Поставщик> за период с <Дата> по <Дата>

Дата	Товар	Ед. изм.	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого			<>		<>	<>	<>

Замечание: Строки отчета группируются по товарам. В конце группы следует включать строку «Итого по товару».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 5

Ведомость поступления товара <Товар> на склады от поставщиков за период с <Дата> по <Дата>

Склад	Документ	Дата	Поставщик	Кол-во	Ед. изм.	Цена	Сумма
Итого				<>			<>

Замечание: Строки отчета группируются по поставщикам. В конце группы следует включать строку «Итого по поставщику».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 6

Ведомость поступления товара <Товар> на склады от поставщика <Поставщик> за период с <Дата> по <Дата>

Склад	Документ	Дата	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого			<>		<>	<>	<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 7

Ведомость продажи товаров покупателям за <Месяц> <Год>

Покупатель	Сумма	Сумма НДС	Всего с НДС
Итого	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 8

Ведомость продажи товаров покупателю <Покупатель> за период с <Дата> по <Дата>

Дата	Товар	Склад	Ед. изм.	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого				<>		<>	<>	<>

Замечание: Строки отчета группируются по товарам. В конце группы следует включать строку «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 9

Ведомость продажи товаров со склада <Склад> покупателям за <Месяц> <Год>

Товар	Покупатель	Валюта	Курс	Документ	Дата	Цена	Сумма
Итого							<>

Замечание: Строки отчета группируются по покупателям. В конце группы следует включать строку «Итого по покупателю».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 10

Ведомость продажи товаров со склада <Склад> покупателю <Покупатель> за период с <Дата> по <Дата>

Дата	Товар	Валюта	Курс	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого				<>		<>	<>	<>

Замечание: Строки отчета группируются по товарам. В конце группы следует включать строку «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 11

Ведомость продажи товара <Товар> со складов покупателям за период с <Дата> по <Дата>

Склад	Документ	Дата	Покупатель	Кол-во	Ед. изм.	Цена	Сумма
Итого				<>			<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 12

Ведомость продажи товара <Товара> со складов покупателю <Покупатель> за период с <Дата> по <Дата>

Склад	Документ	Дата	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого			<>		<>	<>	<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 13

Ведомость движения товаров за <Месяц> <Год>

Товар	Нач. остаток		Приход		Расход		Кон. остаток	
	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма
Итого	<>	<>	<>	<>	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 14

Ведомость движения товара <Товар> за <Месяц> <Год>

Склад	Начальный остаток	Приход	Расход	Конечный остаток
	Сумма	Сумма	Сумма	Сумма
Итого	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 15

Ведомость движения товаров на складе <Склад> за <Месяц> <Год>

Товар	Нач. остаток		Приход		Расход		Кон. остаток	
	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма
Итого	<>	<>	<>	<>	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 16

Ведомость движения товара <Товар> на складе <Склад> за <Месяц> <Год>

Дата	Документ	Номер	Приход		Расход	
			Кол-во	Сумма	Кол-во	Сумма
Итого			<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 17

Ведомость взаиморасчетов с поставщиками за <Месяц> <Год>

Клиент	Сумма долга на начало периода	Сумма прихода	Сумма расхода	Сумма долга на конец периода
Итого	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Оплата товаров поставщику».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 18

Справка о возвратах товаров от покупателей за период с <Дата> по <Дата>

Дата возврата	Документ	Покупатель	Сумма возврата	Причина возврата
Итого			<>	

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Возврат товаров поставщику».

Замечание: Использовать стандартную расшифровку.

Вариант 19

Ведомость взаиморасчетов с покупателем <Покупатель> за <Месяц> <Год>

Дата	Номер	Документ	Приход	Расход	Остаток
Итого			<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков», «Реализация продукции покупателям» и «Поступление оплаты от покупателей».

Замечание: Использовать стандартную расшифровку.

Вариант 20

Ведомость передачи на реализацию товаров реализаторам за <Месяц> <Год>

Реализатор	Документ	Дата	Товар	Кол-во	Сумма
Итого				<>	<>

Замечание: Данные группируются по реализаторам. В конце группы выводится строка «Итого по реализатору».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Передача товаров реализаторам».

Замечание: Использовать стандартную расшифровку.

Вариант 21

Ведомость продажи товаров реализатором <Реализатор> за период с <Дата> по <Дата>

Товар	Дата	Документ	Покупатель	Кол-во	Сумма
Итого				<>	<>

Замечание: Данные группируются по товарам. В конце группы выводится строка «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков», «Передача товаров реализаторам» и «Продажа товаров реализатором».

Замечание: Использовать стандартную расшифровку.

Вариант 22

Ведомость продажи товаров реализаторами за <Месяц> <Год>

Реализатор	Товар	Дата	Документ	Покупатель	Сумма
Итого					<>

Замечание: Данные группируются по реализаторам. В конце группы выводится строка «Итого по реализатору».

Замечание: Разработать документы «Поступление товаров от поставщиков», «Передача товаров реализаторам» и «Продажа товаров реализатором».

Замечание: Использовать стандартную расшифровку.

Вариант 23

Ведомость продажи товаров по заявкам клиентов за период с <Дата> по <Дата>

Клиент	Всего товаров в заявках на сумму	Продано по заявкам клиентов на сумму	Осталось продать на сумму
Итого	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков», «Заявка покупателю» и «Реализация по заявке».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 24

Ведомость продажи товаров по заявкам клиента <Клиент> за период с <Дата> по <Дата>

Товар	Заказано			Продано			Остаток	
	Док.	Кол-во	Сумма	Док.	Кол-во	Сумма	Кол-во	Сумма
Итого		<>	<>		<>	<>	<>	<>

Замечание: Данные группируются по товарам. В конце группы выводится строка «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков», «Заявка покупателю» и «Реализация по заявке».

Замечание: Использовать стандартную расшифровку.

8 ЯЗЫК ЗАПРОСОВ ПЛАТФОРМЫ

Язык запросов Платформы используется для выборки данных из информационной базы (базы данных) решения. Язык запросов позволяет осуществлять только чтение данных. Синтаксис языка запросов Платформы основан на языке SQL, но имеет различные расширения для учета специфики Платформы и решаемых задач.

При использовании запроса вся считанная информация передается на клиентский компьютер и размещается в оперативной памяти. Таким образом, если требуется обработать большие объемы данных, лучше использовать для этого метод `Выбрать` менеджера соответствующего объекта метаданных. При использовании метода `Выбрать` происходит считывание всех полей и табличных частей объектов, но данные считываются порциями, не занимая большие объемы оперативной памяти. При получении объектов из выборки их повторное считывание не производится.

Язык запросов Платформы ориентирован на работу с данными, организованными в виде связанных между собой таблиц, представленных объектами метаданных решения. Запросы позволяют осуществлять выборку и группировку данных с учетом условий и ограничений, производить сортировку и подсчет итогов.

В большинстве случаев язык запросов позволяет более эффективно получать необходимые данные, так как в запросе определяются необходимые поля и условия. В отличие от использования прямого обращения, при выполнении запроса запрашиваются только необходимые данные.

Также запрос позволяет за один вызов получить данные из нескольких таблиц, что снижает общее количество обращений к базе данных. Но при многократном вызове одного и того же запроса, обращение к базе данных будет происходить каждый раз.

При использовании запросов для заполнения печатных форм или вывода сведений на экран, рекомендуется получать не ссылку на объект, а его представление. Автоматическое преобразование ссылки на объект в его представление может существенно повысить время выполнения алгоритма при увеличении количества объектов.

Запросы также рекомендуется использовать при обработке состояния проведения документа, например, для контроля остатков.

Для поиска объектов по коду или наименованию можно использовать методы НайтиПоКоду и НайтиПоНаименованию менеджеров соответствующих объектов метаданных, так как скорость их выполнения не отличается от скорости выполнения аналогичных запросов.

8.1 Методы работы с запросами

Для выполнения запроса используется объект Платформы Запрос:

```
// Выборка данных
Выборка = Справочники.ПримерСправочника.Выбрать();
Пока Выборка.Следующий() Цикл
    Если Выборка.Реквизит1 > 10000 Тогда
        Сообщить(Выборка.Наименование
            + " имеет значение реквизита "
            + Выборка.Реквизит1);
    КонецЕсли;
КонецЦикла;

// Запрос данных
Запрос = Новый Запрос(
    "ВЫБРАТЬ * ИЗ Справочник.ПримерСправочника
    |ГДЕ Реквизит1 > 10000");
Выборка = Запрос.Выполнить().Выбрать();
Пока Выборка.Следующий() Цикл
    Сообщить(Выборка.Наименование
        + " имеет значение реквизита "
        + Выборка.Реквизит1);
КонецЦикла;
```

Общая схема выполнения запроса:

1. Создание объекта типа Запрос.

```
Запрос = Новый Запрос();
```

2. Формирование текста запроса.

```
Запрос.Текст =
"ВЫБРАТЬ
|   *
|ИЗ
|   Справочник.ПримерСправочника
```

```
|ГДЕ
|   Реквизит1 > &Параметр1";
```

3. Установка параметров запроса с помощью метода `УстановитьПараметр` объекта `Запрос`.

```
Запрос.УстановитьПараметр("Параметр1", 0);
```

4. Выполнение запроса, получение результата.

```
Результат = Запрос.Выполнить();
```

5. Получение выборки из результата запроса или выгрузка результата в таблицу значений.

```
// Получение выборки
Выборка = Результат.Выбрать();

// Выгрузка таблицы значений
ТЗ = Результат.Выгрузить();
```

6. Обработка выборки или таблицы значений.

```
// Обработка выборки
Пока Выборка.Следующий() Цикл
    ...
КонецЦикла;

// Обработка таблицы значений
Для Каждого Строка Из ТЗ Цикл
    ...
КонецЦикла;
```

В общем виде текст запроса строится по следующей схеме:

```
ВЫБРАТЬ <Список полей | *>
ИЗ <Список таблиц-источников>
[ГДЕ <Список условий>]
[УПОРЯДОЧИТЬ ПО <Список полей>]
[АВТОУПОРЯДОЧИВАНИЕ]
[СГРУППИРОВАТЬ ПО <Список полей>]
[ИТОГИ [<Агрегатные функции>]
    ПО <Список полей> [ОБЩИЕ]
]
```

8.1.1 Таблицы-источники

С помощью оператора `ИЗ` в качестве источника данных для запроса можно использовать справочники, документы, журналы документов, регистры:

```
ВЫБРАТЬ * ИЗ Справочник.ПримерСправочника

ВЫБРАТЬ * ИЗ Документ.ПримерДокумента

ВЫБРАТЬ * ИЗ РегистрНакопления.ПримерРегистра
```

Если в запросе используется ссылочный тип данных, существует возможность обращаться к данным объекта по ссылке:

```
ВЫБРАТЬ
    Родитель.Реквизит1.Наименование
ИЗ
    Справочник.ПримерСправочника
```

8.1.2 Псевдонимы

Оператор КАК языка запросов позволяет задавать псевдонимы для полей запроса и для таблиц-источников:

```
ВЫБРАТЬ
    Источник.Родитель.ЕдиницаИзмерения.Наименование
        КАК Поле
ИЗ
    Справочник.ПримерСправочника КАК Источник
```

8.1.3 Ограничения количества и уникальности записей

Оператор РАЗЛИЧНЫЕ позволяет выбрать строки из таблицы-источника без повторений:

```
ВЫБРАТЬ РАЗЛИЧНЫЕ
    Родитель.Реквизит1.Наименование
ИЗ
    Справочник.ПримерСправочника
```

Оператор ПЕРВЫЕ n позволяет выбрать первые n записей из таблицы-источника:

```
ВЫБРАТЬ ПЕРВЫЕ 4
    Родитель.Реквизит1.Наименование
ИЗ
    Справочник.ПримерСправочника
```

8.1.4 Обращение к табличным частям

В качестве таблиц-источников запроса могут выступать табличные части справочников и документов:

```
ВЫБРАТЬ
    НомерСтроки,
    Реквизит1ТабличнойЧасти,
    Реквизит2ТабличнойЧасти,
    Ссылка.Наименование
ИЗ
    Справочник.ПримерСправочника.ТабличнаяЧасть
```

При обращении к табличной части как к ссылочному типу данных, результат запроса будет содержать данные из табличной части в поле с типом ТаблицаЗначений:

```
ВЫБРАТЬ
    ТабличнаяЧасть.(
        НомерСтроки,
        Реквизит1ТабличнойЧасти,
        Реквизит2ТабличнойЧасти
    ) КАК Поле1,
    Наименование
ИЗ
    Справочник.ПримерСправочника
```

8.1.5 Условия запроса

Оператор ГДЕ позволяет определить условия выборки данных из таблицы-источника при выполнении запроса:

```
ВЫБРАТЬ
    НомерСтроки,
    Реквизит1ТабличнойЧасти,
    Реквизит2ТабличнойЧасти,
    Ссылка.Наименование
ИЗ
    Справочник.ПримерСправочника.ТабличнаяЧасть
ГДЕ
    Реквизит2ТабличнойЧасти > 0
```

При формировании условий запроса позволяется использовать логические операторы И, ИЛИ, НЕ:

```
ВЫБРАТЬ
    Наименование,
    РеквизитСтрока,
    РеквизитЧисло
ИЗ Справочник.ПримерСправочника
ГДЕ
    РеквизитСтрока = "Строка"
    И РеквизитЧисло < 1000
```

Приоритет логических операторов следующий: НЕ, И, ИЛИ. Для группировки условий необходимо использовать скобки (и).

8.1.6 Вложенные запросы

Язык запросов Платформы также позволяет указывать в качестве таблицы-источника результаты выполнения другого запроса (вложенный запрос):

```
ВЫБРАТЬ
    ВложенныйЗапрос.НомерСтроки,
    ВложенныйЗапрос.Поле1ВложенногоЗапроса,
    ВложенныйЗапрос.Поле2ВложенногоЗапроса
ИЗ
    (ВЫБРАТЬ
        НомерСтроки,
        Реквизит1ТабличнойЧасти
            КАК Поле1ВложенногоЗапроса,
        Реквизит2ТабличнойЧасти
            КАК Поле2ВложенногоЗапроса
    ИЗ
        Справочник.ПримерСправочника.ТабличнаяЧасть)
            КАК ВложенныйЗапрос
ГДЕ
    ВложенныйЗапрос.Поле2ВложенногоЗапроса > 0
```

8.1.7 Параметры запроса

Язык запросов Платформы позволяет использовать в запросах параметры. Параметры — внешние по отношению к запросу переменные, значения которых передаются в запрос. Для указания параметра в запросе применяется оператор &:

```
Запрос = Новый Запрос();
Запрос.Текст =
"ВЫБРАТЬ
|   Наименование,
|   РеквизитЧисло
| ИЗ
|   Справочник.ПримерСправочника
| ГДЕ
|   РеквизитЧисло >= &ПараметрЧисло";
// Передача значения параметра в запрос
Запрос.УстановитьПараметр("ПараметрЧисло", 1000);
// Выполнение запроса с учетом значения параметра
Результат = Запрос.Выполнить();
```

Параметры и поля запроса могут иметь одинаковое имя.

8.1.8 Дополнительные операторы для определения условий запроса

Оператор МЕЖДУ используется для задания допустимого интервала для значения при формировании условия запроса:

```
ВЫБРАТЬ
    Наименование,
    РеквизитЧисло
ИЗ
    Справочник.ПримерСправочника
ГДЕ
    РеквизитЧисло МЕЖДУ &Число1 И &Число2
```

Оператор МЕЖДУ работает аналогично условию РеквизитЧисло >= &Число1 И РеквизитЧисло <= &Число2.

Оператор В позволяет передавать коллекцию значений в качестве значения параметра условия:

```
ВЫБРАТЬ
    Наименование,
    РеквизитЧисло
ИЗ
    Справочник.ПримерСправочника
ГДЕ
    Ссылка В (&КоллекцияСсылок)
```

Оператор В ИЕРАРХИИ позволяет определять элементы справочника, являющиеся потомками определенной группы справочника независимо от уровня иерархии (только для иерархических справочников):

```
ВЫБРАТЬ
    Наименование,
    РеквизитЧисло
ИЗ
    Справочник.ПримерСправочника
ГДЕ
    Ссылка В ИЕРАРХИИ (&ГруппаСправочника)
```

Оператор ПОДОБНО позволяет организовать поиск текстовых данных, совпадающих с определенным шаблоном:

```
ВЫБРАТЬ
    Наименование
ИЗ
    Справочник.ПримерСправочника
ГДЕ
    Наименование ПОДОБНО "М”%
```

Оператор % обозначает любую последовательность символов. Для задания шаблонов используются также и другие специальные операторы.

8.1.9 Упорядочивание результатов запроса

Оператор УПОРЯДОЧИТЬ ПО используется для сортировки результата запроса:

```
// Сортировка по значению поля Наименование
// в порядке возрастания
ВЫБРАТЬ
    Код,
    Наименование
ИЗ
    Справочник.ПримерСправочника
УПОРЯДОЧИТЬ ПО
    Наименование ВОЗР

// Сортировка по значению поля РеквизитЧисло
// в порядке убывания
ВЫБРАТЬ ПЕРВЫЕ 5
    Код,
    Наименование,
    РеквизитЧисло
ИЗ
    Справочник.ПримерСправочника
УПОРЯДОЧИТЬ ПО
    РеквизитЧисло УБЫВ
```

При сортировке результатов запроса в порядке возрастания оператор ВОЗР можно опустить.

Оператор ИЕРАРХИЯ при упорядочивании результатов запроса используется для сортировки групп и/или элементов иерархических справочников с учетом иерархии:

```
ВЫБРАТЬ
    Ссылка
ИЗ
    Справочник.ПримерСправочника
УПОРЯДОЧИТЬ ПО
    Наименование ИЕРАРХИЯ
```

Оператор АВТОУПОРЯДОЧИВАНИЕ позволяет включить режим автоматического выбора «естественного» порядка упорядочивания результатов запроса:

```
ВЫБРАТЬ
    Код,
    Наименование
ИЗ
    Справочник.ПримерСправочника
АВТОУПОРЯДОЧИВАНИЕ
```

Если в запросе отсутствуют операторы УПОРЯДОЧИТЬ ПО, ИТОГИ и СГРУППИРОВАТЬ ПО, результат будет упорядочен по полям сортировки по умолчанию для таблиц-источников в порядке их появления в запросе.

8.1.10 Группировка результатов запроса

Для получения сводной информации в запросе используются агрегатные функции, группировки и оператор ИМЕЮЩИЕ.

Язык запросов Платформы позволяет использовать следующие агрегатные функции:

1. СУММА — вычисляет сумму всех значений в столбце.
2. МАКСИМУМ — определяет максимальное значение в столбце.
3. МИНИМУМ — определяет минимальное значение в столбце.
4. СРЕДНЕЕ — вычисляет среднее арифметическое значений в столбце.
5. КОЛИЧЕСТВО — определяет количество записей в столбце.

Если в качестве параметра функции КОЛИЧЕСТВО передать звездочку («*»), то функция подсчитает количество строк в таблице результата запроса.

Рассмотрим пример использования агрегатных функций:

```
ВЫБРАТЬ
    СУММА(РеквизитЧисло) КАК Сумма,
    МИНИМУМ(РеквизитЧисло) КАК Минимум,
    МАКСИМУМ(РеквизитЧисло) КАК Максимум,
    СРЕДНЕЕ(РеквизитЧисло) КАК Среднее,
    КОЛИЧЕСТВО(*) КАК Количество
ИЗ
    Справочник.ПримерСправочника

ВЫБРАТЬ
    КОЛИЧЕСТВО(*) КАК Количество
ИЗ
    Справочник.ПримерСправочника
ГДЕ
    РеквизитЧисло > 1000
```

```
ВЫБРАТЬ
    КОЛИЧЕСТВО(РАЗЛИЧНЫЕ Наименование) КАК Количество
ИЗ
    Справочники.ПримерСправочника
ГДЕ
    РеквизитЧисло МЕЖДУ
        &ПараметрЧисло1 И &ПараметрЧисло2
```

Оператор СГРУППИРОВАТЬ ПО используется для свертки по определенным полям с вычислением агрегатных функций по каждой группе:

```
ВЫБРАТЬ
    Номенклатура,
    СУММА(РеквизитЧисло) КАК Сумма
ИЗ
    Справочник.ПримерСправочника
ГДЕ
    РеквизитДата МЕЖДУ
        &ДатаНачала И &ДатаОкончания
СГРУППИРОВАТЬ ПО
    Номенклатура

ВЫБРАТЬ
    Наименование,
    СУММА(РеквизитЧисло) КАК Сумма
    МИНИМУМ(РеквизитЧисло) КАК Минимум,
    МАКСИМУМ(РеквизитЧисло) КАК Максимум,
    СРЕДНЕЕ(РеквизитЧисло) КАК Среднее,
    КОЛИЧЕСТВО(*) КАК Количество
ИЗ
    Справочник.ПримерСправочника
СГРУППИРОВАТЬ ПО
    Наименование
```

При использовании оператора СГРУППИРОВАТЬ ПО можно группировать данные по нескольким полям, при этом агрегатные функции будут подсчитаны для каждой комбинации группировок.

Оператор ИМЕЮЩИЕ позволяет накладывать условия на значения агрегатных функций.

```
ВЫБРАТЬ
    Наименование,
    СУММА(РеквизитЧисло) КАК Сумма
ИЗ
    Справочник.ПримерСправочника
СГРУППИРОВАТЬ ПО
    Наименование
```

```
ИМЕЮЩИЕ
    СУММА(РеквизитЧисло) >= 1000
```

При использовании оператора ИМЕЮЩИЕ условие отбора применяется не к исходным записям таблицы-источника, а к получившимся группам.

8.1.11 Расчет итогов запроса

В отличие от языка SQL, язык запросов Платформы имеет возможность расчета итогов. Данный механизм дает возможность включить в результат запроса дополнительные строки, содержащие общие и промежуточные итоги по заданным полям и группировкам. Для формирования итогов используется оператор ИТОГИ ... ПО.

Общий итог содержит сумму всех значений определенного столбца (столбцов) в рамках всей таблицы результатов запроса. При формировании общих итогов с применением типа итогов ОБЩИЕ в таблице результатов запроса формируется дополнительная строка с типом записи запроса ОбщийИтог системного перечисления ТипЗаписиЗапроса:

```
// Запрос
ВЫБРАТЬ
    Наименование,
    РеквизитЧисло
ИЗ
    Справочник.ПримерСправочника
ИТОГИ
    СУММА(РеквизитЧисло)
ПО
    ОБЩИЕ

// Обработка результатов запроса
Выборка = Запрос.Выполнить().Выбрать();
Пока Выборка.Следующий() Цикл
    Если Выборка.ТипЗаписи() =
        ТипЗаписиЗапроса.ОбщийИтог Тогда
        // Обработка общих итого
        ...
    Иначе
        // Обработка результатов запроса
        ...
    КонецЕсли;
КонецЦикла;
```

«Обычным» строкам результата запроса соответствует тип записи ДетальнаяЗапись системного перечисления ТипЗаписи-Запроса.

Если группировочное поле является ссылкой на справочник, то для расчета итогов по родительским элементам/группам справочника используется тип итогов ИЕРАРХИЯ:

```
ВЫБРАТЬ
    Наименование,
    РеквизитЧисло
ИЗ
    Справочники.ПримерСправочника
ИТОГИ
    СУММА(РеквизитЧисло)
ПО
    Наименование ИЕРАРХИЯ
```

В этом случае в результат будут добавлены записи с итогами для уровней иерархии справочника с типом записи запроса ИтогПоИерархии системного перечисления ТипЗаписиЗапроса.

Для получения итогов про группировке необходимо указать имя поля группировки в качестве типа итогов:

```
ВЫБРАТЬ
    Наименование,
    СУММА(РеквизитЧисло) КАК Сумма
ИЗ
    Справочники.ПримерСправочника
СГРУППИРОВАТЬ ПО
    Наименование
ИТОГИ
    СУММА(РеквизитЧисло)
ПО
    Наименование
```

Строкам результата запроса с итогами по группировке будет соответствовать тип записи запроса ИтогПоГруппировке системного перечисления ТипЗаписиЗапроса.

В языке запросов Платформы допускается совместное использование различных типов итогов в одном запросе, например, общих, иерархических и итогов по группировкам.

```
ВЫБРАТЬ
    Ссылка.РеквизитСсылка КАК Ссылка,
    Номенклатура,
    Ссылка.Код КАК Код,
    РеквизитЧисло
```

```
ИЗ
    Справочники.ПримерСправочника
ИТОГИ
    СУММА(РеквизитЧисло)
ПО
    ОБЩИЕ, Ссылка, Номенклатура ИЕРАРХИЯ
```

8.1.12 Соединение таблиц-источников

Оператор СОЕДИНЕНИЕ ... ПО позволяет обращаться в одном запросе к нескольким таблицам-источникам:

```
ВЫБРАТЬ
    Док.РеквизитСсылка,
    Спр.РеквизитЧисло,
    Спр.Родитель КАК Группа
ИЗ
    Документ.ПримерДокумента.ТабличнаяЧасть КАК Док
    СОЕДИНЕНИЕ
        Справочник.ПримерСправочника КАК Спр
    ПО
        Док.РеквизитСсылка = Спр.Ссылка
```

Оператор СОЕДИНЕНИЕ ... ПО работает аналогично оператору JOIN языка SQL. Существует четыре вида операции соединения:

1. Левое соединение (оператор ЛЕВОЕ СОЕДИНЕНИЕ). Выбирает все записи левой таблицы-источника и соединяет их с записями правой таблицы-источника при выполнении некоторого условия (совпадение значений полей). Если в правой таблице-источнике отсутствует запись, удовлетворяющая условию соединения, в качестве правого значения используется NULL.
2. Правое соединение (оператор ПРАВОЕ СОЕДИНЕНИЕ). Выбирает все записи правой таблицы-источника и соединяет их с записями левой таблицы-источника при выполнении некоторого условия. Если в левой таблице-источнике отсутствует запись, удовлетворяющая условию соединения, в качестве левого значения используется NULL.
3. Внутреннее соединение (оператор ВНУТРЕННЕЕ СОЕДИНЕНИЕ или СОЕДИНЕНИЕ). Выбирает из левой и правой таблиц-источников только те записи, которые выполняют некоторое условие, и соединяет их. Если условия не выполняются, результат запроса может не содержать записей.

4. Полное соединение (оператор ПОЛНОЕ СОЕДИНЕНИЕ). Выбирает все записи из левой и правой таблиц-источников и соединяет только те записи, которые выполняют некоторое условие. Если в левой таблице-источнике отсутствует запись, удовлетворяющая условию соединения, в качестве левого значения используется NULL. Если в правой таблице-источнике отсутствует запись, удовлетворяющая условию соединения, в качестве правого значения используется NULL.

В некоторых случаях вместо оператора СОЕДИНЕНИЕ ... ПО можно использовать простое обращение к свойству объекта по ссылке (при использовании ссылочного типа данных). При этом соединение таблиц производится неявно:

```
ВЫБРАТЬ
    РеквизитСсылка,
    РеквизитСсылка.РеквизитЧисло,
    РеквизитСсылка.Родитель КАК Группа
ИЗ
    Документ.ПримерДокумента.ТабличнаяЧасть
```

8.1.13 Проверка значений на равенство NULL

NULL-значения — это неуказанные, отсутствующие или неизвестные значения. Ноль, пустая строка и пробел не являются NULL-значением.

Оператор ЕСТЬ NULL позволяет проверить значение заданного выражения на равенство NULL:

```
ВЫБРАТЬ
    Наименование,
    РеквизитСтрока
ИЗ
    Справочник.ПримерСправочника
ГДЕ
    РеквизитСтрока ЕСТЬ NULL
```

Таблица результатов данного запроса будет содержать только группы справочника, так как группы справочника не содержат реквизита «РеквизитСтрока». Таким образом, для групп справочника «ПримерСправочника» поле РеквизитСсылка при выполнении запроса будет иметь значение NULL (значение не указано).

Также для работы с NULL-значениями язык запросов Платформы содержит функцию ЕСТЬNULL(<Выражение>, <Выражение>),

которая возвращает предустановленное значение вместо NULL-значения, содержащегося в определенном поле выборки:

```
ВЫБРАТЬ
    Наименование,
    ЕСТЬNULL(РеквизитСтрока, """")
ИЗ
    Справочник.ПримерСправочника
ГДЕ
    ЭтоГруппа = ИСТИНА
```

8.1.14 Объединение результатов запросов

Язык запросов Платформы также позволяет объединять результаты нескольких запросов с помощью оператора `ОБЪЕДИНИТЬ [ВСЕ]`. Оператор `ОБЪЕДИНИТЬ [ВСЕ]` работает аналогично оператору `UNION [ALL]` языка SQL.

При использовании оператора `ОБЪЕДИНИТЬ [ВСЕ]` записи, полученные с помощью каждого из объединяемых запросов, будут объединены в одну таблицу результатов запроса. При объединении каждый запрос собирает данные независимо, а такие операции, как упорядочивание результатов и расчет итогов, выполняются уже над результатом объединения запросов.

Поля результата запроса будут называться также, как поля выборки первого из объединяемых запросов. Поля выборки остальных запросов сопоставляются с полями результата в соответствии с порядком их следования в списке полей выборки.

Объединяемые запросы должны иметь одинаковое количество полей в списке полей выборки:

```
ВЫБРАТЬ
    Док.Номенклатура КАК Товар,
    Док.Цена КАК ЦенаТовара
ИЗ
    Документ.РасходнаяНакладная.Товары КАК Док
ГДЕ
    Док.Ссылка.Дата МЕЖДУ
        &ДатаНачала И &ДатаОкончания

ОБЪЕДИНИТЬ

ВЫБРАТЬ
    Док.Номенклатура,
    Док.Цена
```

```
ИЗ
    Документ.ПриходнаяНакладная.Товары КАК Док
ГДЕ
    Док.Ссылка.Дата МЕЖДУ
        &ДатаНачала И &ДатаОкончания
```

Если поля выборки объединяемых запросов имеют разный тип, то поля результата запроса будут иметь составной тип данных.

При использовании оператора ОБЪЕДИНИТЬ полностью одинаковые строки, полученные в результате выполнения разных запросов, заменяются одной. Если необходимо, чтобы все одинаковые строки вошли в таблицу результатов, необходимо использовать оператор ОБЪЕДИНИТЬ ВСЕ:

```
ВЫБРАТЬ
    Док.Номенклатура КАК Товар,
    Док.Цена КАК ЦенаТовара
ИЗ
    Документ.РасходнаяНакладная.Товары КАК Док
ГДЕ
    Док.Ссылка.Дата МЕЖДУ
        &ДатаНачала И &ДатаОкончания

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ
    Док.Номенклатура,
    Док.Цена
ИЗ
    Документ.ПриходнаяНакладная.Товары КАК Док
ГДЕ
    Док.Ссылка.Дата МЕЖДУ
        &ДатаНачала И &ДатаОкончания
```

8.1.15 Обращение к виртуальным таблицам регистров

Виртуальные таблицы регистров доступны в рамках языка запросов Платформы в качестве таблиц-источников и позволяют получить данные регистров без необходимости составления сложных запросов. Например, виртуальная таблица «Остатки» регистра накопления может предоставить данные по остаткам товаров в разрезе измерений и периодов на какой-то момент времени:

```
ВЫБРАТЬ
    Измерение1,
    Ресурс1,
    Ресурс2
ИЗ
    РегистрНакопления.ПримерРегистра
        .Остатки(&Дата, Измерение2 = &Параметр)
```

Как видно из данного примера, работа с виртуальными таблицами похожа на вызов функций. У каждого типа регистров Платформы существует свой набор виртуальных таблиц со своим набором параметров.

Для корректной работы с виртуальными таблицами необходимо передавать условия отбора в качестве параметров виртуальной таблицы. Если установить отбор с помощью оператора ГДЕ, сначала будет сформирована виртуальная таблица на основе всех данных регистра, а затем к этим результатам будет применен отбор:

```
ВЫБРАТЬ
    Измерение1,
    Ресурс1,
    Ресурс2
ИЗ
    РегистрНакопления.ПримерРегистра
ГДЕ
    Измерение2 = &Параметр
```

При увеличении количества записей в регистре использование данного запроса приведет к резкому снижению производительности.

Если приведенный выше запрос применять в обработчике события проведения документа, например, для контроля остатков, таблица результатов запроса будет содержать некорректные данные, так как не задан момент времени, на который необходимо получить остатки — остатки будут получены на текущую дату.

8.1.16 Временные таблицы

Язык запросов Платформы позволяет использовать в запросах временные таблицы. Временные таблицы могут повысить производительность запросов, снизить количество блокировок в некоторых случаях и сделать текст запроса более легким для восприятия.

Временные таблицы можно создавать и уничтожать с помощью операторов языка запросов ПОМЕСТИТЬ и УНИЧТОЖИТЬ соответственно в рамках одного пакета запросов:

```
Запрос = Новый Запрос();
Запрос.Текст =
"ВЫБРАТЬ
|   Ссылка
|ПОМЕСТИТЬ ВТСпр
|ИЗ
|   Справочник.ПримерСправочника
|ГДЕ
|   РеквизитСсылка = &ПараметрСсылка
|;
|
|ВЫБРАТЬ
|   Измерение1,
|   Ресурс1
|ИЗ
|   РегистрНакопления.ПримерРегистра.Остатки(
|           &Дата,
|           Измерение2 В
|               (ВЫБРАТЬ
|                   ВТСпр.Ссылка
|               ИЗ
|                   ВТСпр))
|;
|
|УНИЧТОЖИТЬ ВТСпр";
Запрос.УстановитьПараметр("ПараметрСсылка",
    Перечисления.ПримерПеречисления.Значение1);
Запрос.УстановитьПараметр("Дата", ТекущаяДата());
Запрос.Выполнить();
```

Для передачи временных таблиц между методами конфигурации используется объект МенеджерВременныхТаблиц:

```
Процедура ОсновнаяПроцедура()
    МенеджерВТ = Новый МенеджерВременныхТаблиц;
    ПолучитьЭлементыСправочника(МенеджерВТ);
    ПолучитьОстатки(МенеджерВТ);
    МенеджерВТ.Закрыть();
КонецПроцедуры

Процедура ПолучитьСписокТовар(МенеджерВТ)
    ЗапросСпр = Новый Запрос();
    ЗапросСпр.МенеджерВременныхТаблиц = МенеджерВТ;
    ЗапросСпр.Текст =
```

```
    "ВЫБРАТЬ
    |   Ссылка
    |ПОМЕСТИТЬ ВТСпр
    |ИЗ
    |   Справочник.ПримерСправочника
    |ГДЕ
    |   РеквизитСсылка = &ПараметрСсылка";
    ЗапросСпр.УстановитьПараметр("ПараметрСсылка",
        Перечисления.ПримерПеречисления.Значение1);
    ЗапросСпр.Выполнить();
КонецПроцедуры

Процедура ПолучитьОстатки(МенеджерВТ)
    ЗапросРег = Новый Запрос();
    ЗапросРег.МенеджерВременныхТаблиц = МенеджерВТ;
    ЗапросРег.Текст =
    "ВЫБРАТЬ
    |   Измерение1,
    |   Ресурс1
    |ИЗ
    |   РегистрНакопления.ПримерРегистра.Остатки(
    |           &Дата,
    |           Измерение2 В
    |               (ВЫБРАТЬ
    |                   ВТСпр.Ссылка
    |               ИЗ
    |                   ВТСпр)),
    |   ВТСпр КАК ВТСпр";
    ЗапросРег.УстановитьПараметр("Дата", ТекущаяДата());
    ЗапросРег.Выполнить();

    // Обработка результата запроса
    ...
КонецПроцедуры
```

8.1.17 Пакет запросов

Пакет запрос представляет собой последовательность запросов, разделенных символом «;», при этом запросы выполняются по порядку. Результатом выполнения пакета запросов будет являться либо результат, возвращаемый последним запросом пакета, либо массив результатов всех запросов пакета в той последовательности, в которой запросы представлены в пакете.

Обычно в языке запросов Платформы пакеты запросов используются для более удобной работы с временными таблицами.

Пакеты запросов позволяют создавать, использовать и уничтожать временные таблицы в рамках одного запроса.

Пример пакета запросов представлен выше.

8.1.18 Функции языка запросов

Язык запросов Платформы содержит ряд функций для работы с полями выборки:

1. Для работы со строками. Данная группа содержит функцию ПОДСТРОКА(<Строка>, <НачальнаяПозиция>, <Длина>).
2. Для работы с датами:
   - функции для получения составной части даты (даты и времени): ГОД(<Дата>), КВАРТАЛ(<Дата>), МЕСЯЦ(<Дата>), ДЕНЬГОДА(<Дата>), ДЕНЬ(<Дата>), НЕДЕЛЯ(<Дата>), ДЕНЬНЕДЕЛИ(<Дата>), ЧАС(<Дата>), МИНУТА(<Дата>), СЕКУНДА(<Дата>);
   - функции для получения начала и конца периода указанного типа для даты (даты и времени): НАЧАЛОПЕРИОДА(<Дата>, <Период>) и КОНЕЦПЕРИОДА(<Дата>, <Период>) соответственно;
3. Агрегатные функции.
4. Прочие функции:
   - функция ЕСТЬNULL(<Выражение>, <Выражение>);
   - функция для получения текстового представления значения ПРЕДСТАВЛЕНИЕ(<Выражение>). Данная функция всегда возвращает значение текстового типа;
   - функция для получения текстового представления объекта ссылочного типа ПРЕДСТАВЛЕНИЕССЫЛКИ(<Выражение>). Данная функция возвращает текстовое представление только для объекта ссылочного типа, для остальных типов значений тип не изменится;
   - функция для получения типа значения ТИПЗНАЧЕНИЯ(<Выражение>);
   - функция для создания последовательности уникальных возрастающих последовательностей в рамках временной таблицы АВТОНОМЕРЗАПИСИ().

Язык запросов Платформы также содержит множество операторов:

1. Арифметические операторы: «+», «-», «*», «/».

2. Логические операторы:
   - операторы сравнения: «=», «<>», «<», «<=», «>», «>=»;
   - логические операторы: И, ИЛИ, НЕ;
   - оператор сравнения строки с шаблоном ПОДОБНО;
   - операторы определения вхождения значения в коллекцию: В и В ИЕРАРХИИ;
   - оператор определения вхождения значения в интервал значений МЕЖДУ;
   - оператор для сравнения значения на равенство NULL-значению ЕСТЬ NULL;
   - оператор ССЫЛКА используется для определения является ли значение ссылкой на указанный объект метаданных (справочник, документ и т. д.).
3. Прочие операторы:
   - оператор ВЫБОР используется для формирования условных операторов и операторов выбора в рамках языка запросов;

```
ВЫБОР
    КОГДА <Выражение> ТОГДА <Выражение>
    [КОГДА <Выражение> ТОГДА <Выражение>]
    [ ... ]
    [ИНАЧЕ <Выражение>]
КОНЕЦ
```

   - функция ВЫРАЗИТЬ позволяет привести тип составного значения к определенному типу значения;
   - функция ВЫРАЗИТЬ КАК СТРОКА используется для усечения длины строки, а также для приведения строки неограниченной длины к строке с фиксированной длиной;
   - функция ВЫРАЗИТЬ КАК ЧИСЛО используется для округления чисел. При использовании функции ВЫРАЗИТЬ КАК ЧИСЛО 0.5 округляется до 1.0. Если требуется получить целое значение, то следует перед округлением вычесть из числового значения 0.5. Для округления до десятых в меньшую сторону необходимо вычесть из числового значения 0.5, а для округления до сотых — 0.05.

Для создания внутри запроса значения с типом дата и время используется функция ДАТАВРЕМЯ(<Год>, <Месяц>, <День>, <Час>, <Минута>, <Секунда>).

Функция ЗНАЧЕНИЕ(<Имя>) используется для получения значения предопределенного элемента справочника или перечисления внутри запроса по имени, например, Перечисление.ПримерПеречисления.Значение1.

Для получения типа значения по его имени используется функция ТИП(<Имя типа>).

8.2 Виртуальные таблицы регистра накопления

Доступные виртуальные таблицы регистра накопления зависят от вида регистра накопления.

8.2.1 Виртуальные таблицы оборотного регистра накопления

У регистра накполения с видом регистра «Обороты» в рамках запроса доступна одна виртуальная таблица — «Обороты». Виртуальная таблица «Обороты» позволяет получить обороты ресурсов в разрезе измерений и периода.

Параметрами виртуальной таблицы «Обороты» являются:

- НачалоПериода. Данный параметр определяет дату начала периода получения оборотов — значение стандартного реквизита «Период» записи регистра накопления должно быть больше или равным значению параметра НачалоПериода;
- КонецПериода. Данный параметр определяет дату окончания периода получения оборотов — значение стандартного реквизита «Период» записи регистра накопления должно быть меньше или равным значению параметра НачалоПериода;
- Периодичность. Периодичность используется для выбора типа группировки записей регистра накопления. При изменении значения параметра «Периодичность» появляется возможность использовать дополнительные поля даной виртуальной таблицы. Например, при указании значения «Регистратор» параметра «Периодичность» появляется возможность обращаться к данным документа-регистратора;

- Условие. Условие позволяет установить отборы для записей регистра накопления.

8.2.2 Виртуальные таблицы регистра накопления остатков

Для регистра накопления с видом «Остатки» в рамках запроса доступны три виртуальные таблицы: «Обороты», «Остатки», «ОстаткиИОбороты».

Виртуальная таблица «Обороты»

В отличие от виртуальной таблицы «Обороты» оборотного регистра накопления, виртуальная таблица «Обороты» регистра накопления остатков позволяет получать не только обороты, но и приход и расход по каждому ресурсу в разрезе периода и измерений.

Параметры виртуальной таблицы «Обороты» регистра накопления остатков соответствуют параметрам аналогичной таблицы оборотного регистра накопления.

Виртуальная таблица «Остатки»

Виртуальная таблица «Остатки» регистра накопления остатков используется для получения остатков ресурсов в разрезе измерений.

Параметрами виртуальной таблицы «Остатки» являются:

- Период. Данный параметр используется для определения даты получения остатков;
- Условие. Условие позволяет установить отборы для записей регистра накопления.

Виртуальная таблица «ОстаткиИОбороты»

Виртуальная таблица «ОстаткиИОбороты» регистра накопления остатков используется для получения обороты за выбранный период времени, а также для получения остатков на начало и конец выбранного периода.

Виртуальная таблица «ОстаткиИОбороты» объединяет в себе виртуальны таблицы «Обороты» и «Остатки».

Параметры виртуальной таблицы «ОстаткиИОбороты» практически полностью соответствуют параметрам таблицы «Обороты».

8.3 Конструктор запросов

Для автоматизации процесса формирования текста запроса Платформа содержит конструктор запросов двух видов:

1. Конструктор запросов. Конструктор запросов позволяет формировать текст запроса и контролировать его корректность. Для формирования текста запроса используются визуальные средства.
2. Конструктор запросов с обработкой результата. Данный вид конструктора запросов помимо текста запроса также формирует фрагмент программного кода, который исполняет запрос и выводит результаты в табличный документ или диаграмму.

Результатом работы конструктора запросов является синтаксически правильный текст запроса, необходимые синтаксические конструкции формируются автоматически. Разработчик может составить работоспособный запрос, даже не владея синтаксисом языка запросов. Готовый текст запроса может быть добавлен в текст модуля или скопирован в буфер обмена.

Также конструктор запросов позволяет редактировать существующие запросы. Имеющийся текст запроса будет проанализирован и представлен в конструкторе запросов в виде соответствующих настроек.

Для вызова конструктора запроса необходимо установить курсор на необходимую строку модуля, а затем нажать правую кнопу мыши и в появившемся меню выбрать пункт «Конструктор запроса» или «Конструктор запроса с обработкой результата». Для создания нового запроса курсор должен быть установлен на пустую строку, а для редактирования запроса — на какую-либо строку текста запроса.

На рисунке 8.1 представлено окно конструктора запроса с активной вкладкой «Таблицы и поля».

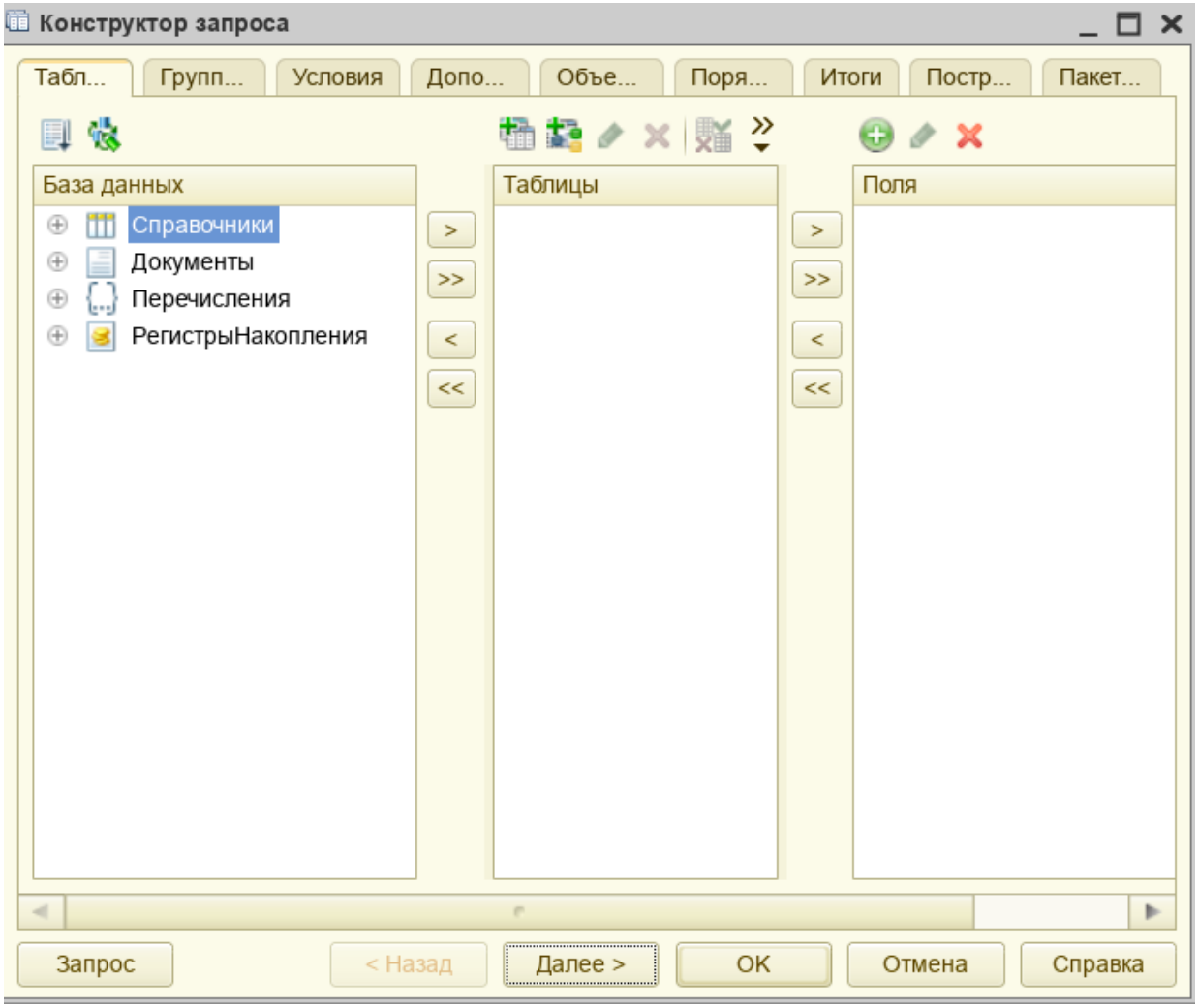


Рис. 8.1. Вкладка «Таблицы и поля» конструктора запроса

На вкладке «Таблицы и поля» можно осуществлять выбор таблиц-источников из доступных объектов метаданных, а также указать необходимые поля таблиц-источников для выборки.

Для создания вложенного запроса необходимо нажать на кнопку «Создать вложенный запрос» (рис. 8.2, кнопка 1), для установки параметров виртуальной таблицы необходимо использовать кнопку «Параметры виртуальной таблицы» (рис. 8.2, кнопка 2). Кнопка «Добавить» (рис. 8.2, кнопка 3) позволяет создавать собственные поля выборки запроса, используя для этого различные функции и операторы языка запросов Платформы.

При указанных на рисунке 8.2 настройках конструктора запроса результирующий запрос будет иметь вид:

```
ВЫБРАТЬ
    ТоварыОстатки.Товар КАК Товар,
```

```
        ТоварыОстатки.КоличествоОстаток
            КАК КоличествоОстаток
    ИЗ
        РегистрНакопления.Товары.Остатки(
            &Дата, Поставщик = &Поставщик)
            КАК ТоварыОстатки
```

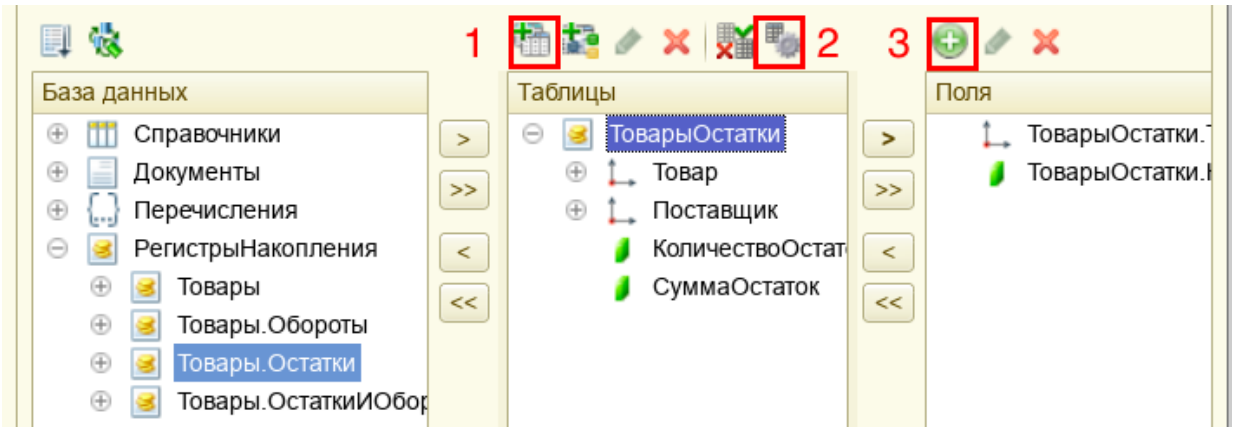


Рис. 8.2. Основные функции вкладки «Таблицы и поля» конструктора запроса

Вкладка «Группировка» конструктора запроса (рис. 8.3) позволяет определять поле группировки и суммируемое поле с указанием используемой агрегатной функции.

При настройках группировки указанных на рисунке 8.3 результирующий запрос будет иметь вид:

```
ВЫБРАТЬ
    ТоварыОстатки.Товар КАК Товар,
    СУММА(ТоварыОстатки.КоличествоОстаток)
        КАК КоличествоОстаток
ИЗ
    РегистрНакопления.Товары.Остатки(
        &Дата, Поставщик = &Поставщик)
    КАК ТоварыОстатки
СГРУППИРОВАТЬ ПО
    ТоварыОстатки.Товар
```

На вкладке «Условия» конструктора запроса задаются условия запроса. При установке условий, представленных на рисунке 8.4,запрос будет иметь вид:

```
ВЫБРАТЬ
    ТоварыОстатки.Товар КАК Товар,
    СУММА(ТоварыОстатки.КоличествоОстаток)
        КАК КоличествоОстаток
```

```
ИЗ
    РегистрНакопления.Товары.Остатки(
        &Дата, Поставщик = &Поставщик)
        КАК ТоварыОстатки
ГДЕ
    ТоварыОстатки.КоличествоОстаток = &КоличествоОстаток
СГРУППИРОВАТЬ ПО
    ТоварыОстатки.Товар
```

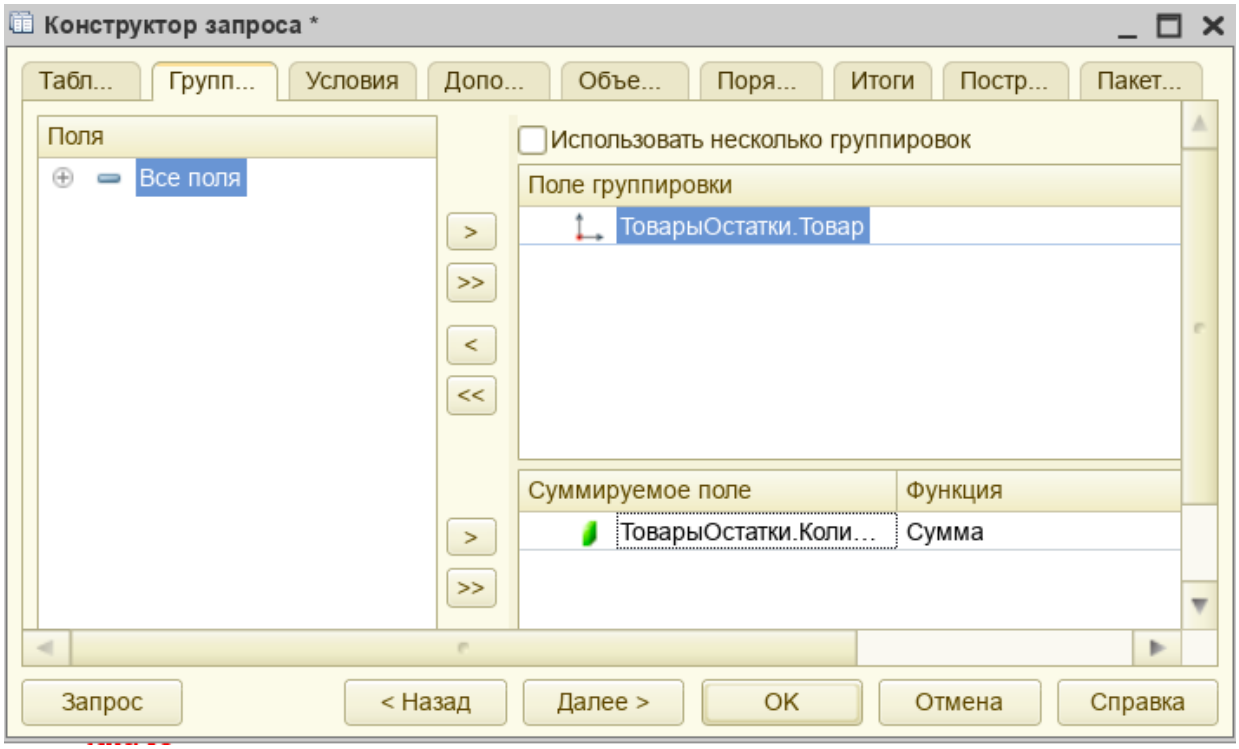


Рис. 8.3. Вкладка «Группировка» конструктора запроса

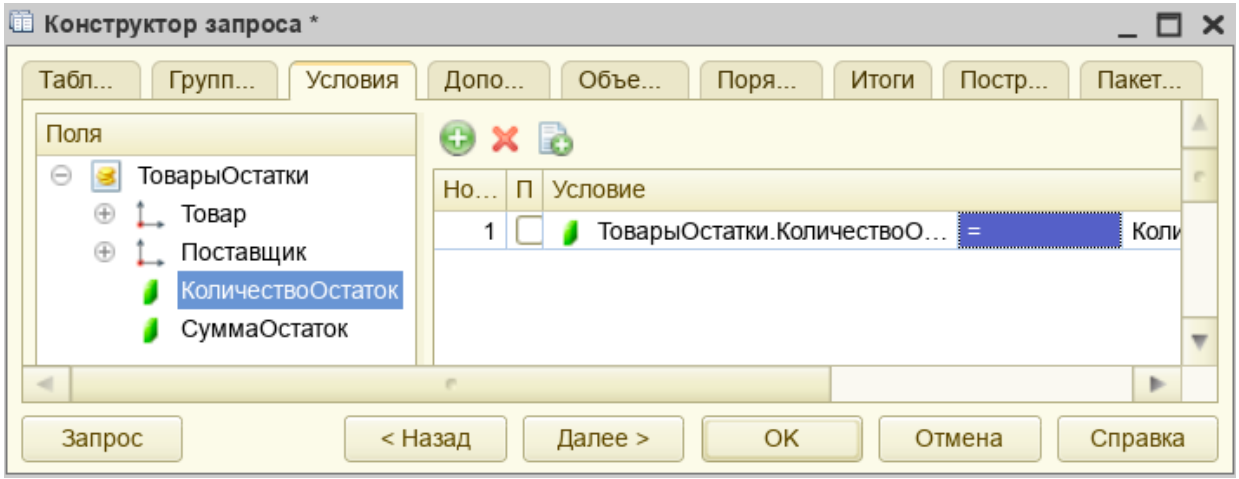


Рис. 8.4. Вкладка «Условия» конструктора запроса

Вкладка «Дополнительно» конструктора запроса используется для:

- установки ограничения уникальности срок таблицы результатов запроса через параметр «Без повторяющихся»;
- установки ограничения на максимальное количество строк таблицы результатов запроса через параметр «Первые»;
- определения типа запроса: «Выборка данных», «Создание временной таблицы» и «Уничтожение временной таблицы».

Настройки вкладки «Дополнительно», представленной на рисунке 8.5, позволяют получить следующий запрос:

```
ВЫБРАТЬ РАЗЛИЧНЫЕ ПЕРВЫЕ 3
    ТоварыОстатки.Товар КАК Товар,
    СУММА(ТоварыОстатки.КоличествоОстаток)
        КАК КоличествоОстаток
ПОМЕСТИТЬ ВТ
ИЗ
    РегистрНакопления.Товары.Остатки(
        &Дата, Поставщик = &Поставщик)
        КАК ТоварыОстатки
ГДЕ
    ТоварыОстатки.КоличествоОстаток = &КоличествоОстаток
СГРУППИРОВАТЬ ПО
    ТоварыОстатки.Товар
```

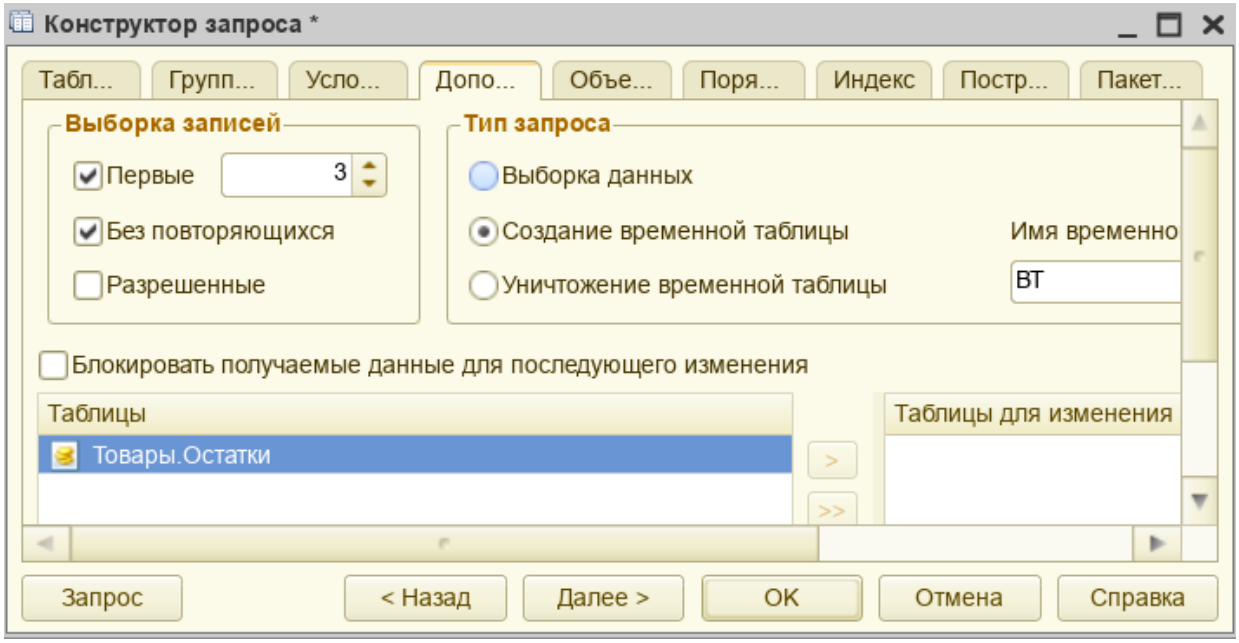


Рис. 8.5. Вкладка «Дополнительно» конструктора запроса

Вкладка «Объединения/Псевдонимы» конструктора запроса используется для задания псевдонимов полей и объединения результатов запросов. Для добавления запроса с целью объединения результатов необходимо нажать на кнопку «Добавить». Новый запрос будет доступен для редактирования в боковой вкладке вкладки «Таблицы и поля».

На рисунке 8.6 представлены настройки объединения результатов двух запросов, а также задан псевдоним для поля. Полученный запрос имеет вид:

```
ВЫБРАТЬ РАЗЛИЧНЫЕ ПЕРВЫЕ 3
    ТоварыОстатки.Товар КАК Товар,
    СУММА(ТоварыОстатки.КоличествоОстаток)
        КАК Количество
ПОМЕСТИТЬ ВТ
ИЗ
    РегистрНакопления.Товары.Остатки(
        &Дата, Поставщик = &Поставщик)
        КАК ТоварыОстатки
ГДЕ
    ТоварыОстатки.КоличествоОстаток = &КоличествоОстаток
СГРУППИРОВАТЬ ПО
    ТоварыОстатки.Товар

ОБЪЕДИНИТЬ

ВЫБРАТЬ
    ТоварыОстатки.Товар,
    ТоварыОстатки.КоличествоОстаток
ИЗ
    РегистрНакопления.Товары.Остатки(
        &Дата2, Поставщик = &Поставщик2)
        КАК ТоварыОстатки
```

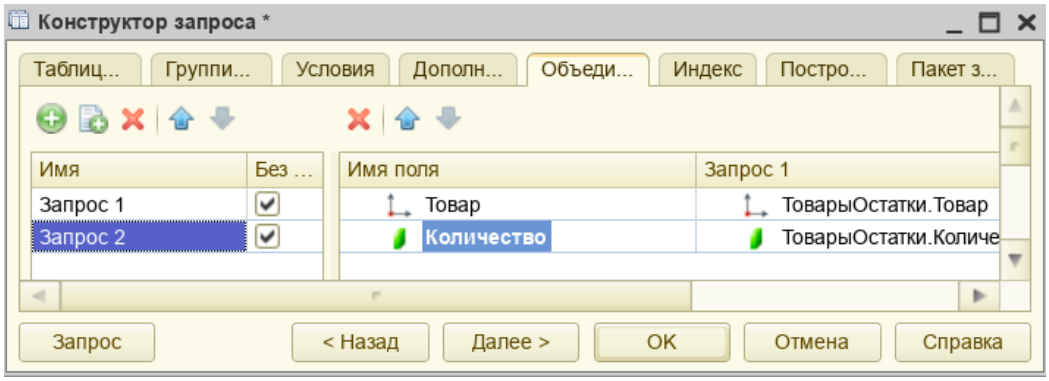


Рис. 8.6. Вкладка «Объединения/Псевдонимы» конструктора запроса

Для сортировки записей таблицы результатов запроса используется вкладка «Порядок». На рисунке 8.7 заданы настройки упорядочивания, которые позволяют получить следующий запрос:

```
ВЫБРАТЬ РАЗЛИЧНЫЕ ПЕРВЫЕ 3
    ТоварыОстатки.Товар КАК Товар,
    СУММА(ТоварыОстатки.КоличествоОстаток)
    КАК Количество
ПОМЕСТИТЬ ВТ
ИЗ
    РегистрНакопления.Товары.Остатки(
        &Дата, Поставщик = &Поставщик)
        КАК ТоварыОстатки
ГДЕ
    ТоварыОстатки.КоличествоОстаток = &КоличествоОстаток

СГРУППИРОВАТЬ ПО
    ТоварыОстатки.Товар

УПОРЯДОЧИТЬ ПО
    Товар
АВТОУПОРЯДОЧИВАНИЕ
```

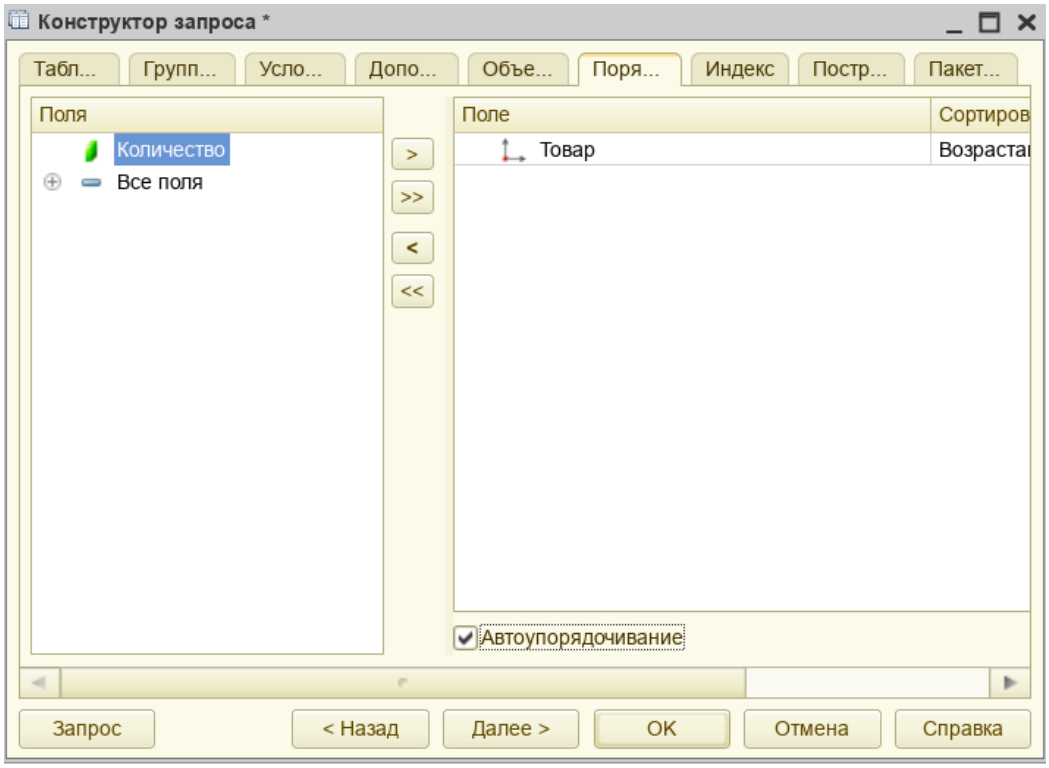


Рис. 8.7. Вкладка «Порядок» конструктора запроса

Для формирования пакета запросов используется вкладка «Пакет запросов» (рис. 8.8). Данная вкладка позволяет добавлять запрос в пакет, добавлять запрос в пакет копированием, удалять запрос из пакета, а также менять порядок запросов в пакете. Новый запрос будет доступен для редактирования в боковой вкладке вкладки «Таблицы и поля».

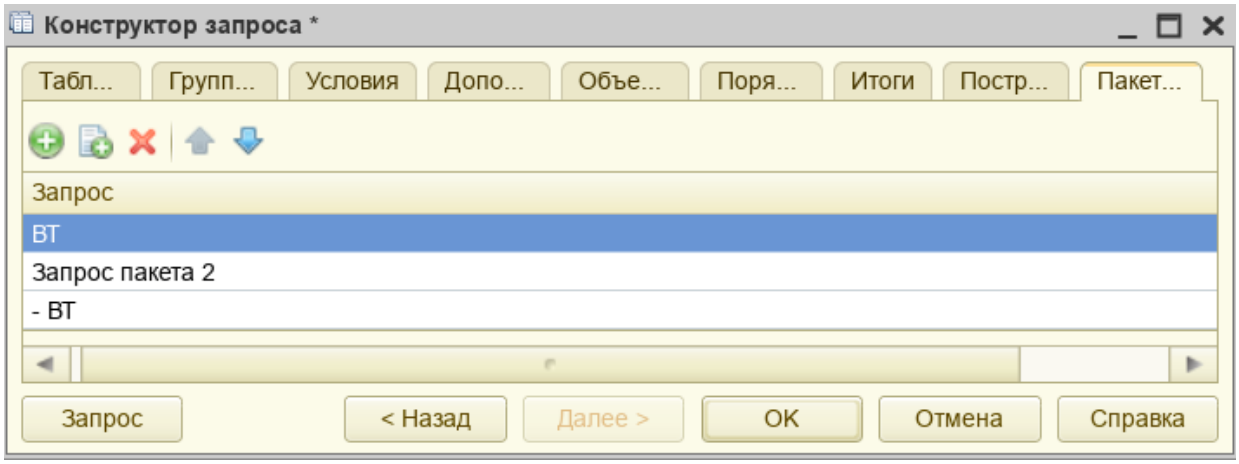


Рис. 8.8. Вкладка «Пакет запросов» конструктора запроса

Настройки пакета запросов, представленные на рисунке 8.8, формируют следующий запрос:

```
ВЫБРАТЬ РАЗЛИЧНЫЕ ПЕРВЫЕ 3
    ТоварыОстатки.Товар КАК Товар,
    СУММА(ТоварыОстатки.КоличествоОстаток)
    КАК Количество
ПОМЕСТИТЬ ВТ
ИЗ
    РегистрНакопления.Товары.Остатки(
        &Дата, Поставщик = &Поставщик)
        КАК ТоварыОстатки
ГДЕ
    ТоварыОстатки.КоличествоОстаток = &КоличествоОстаток
СГРУППИРОВАТЬ ПО
    ТоварыОстатки.Товар

УПОРЯДОЧИТЬ ПО
    Товар
АВТОУПОРЯДОЧИВАНИЕ
;

ВЫБРАТЬ
    ВТ.Товар КАК Товар,
    ВТ.Количество КАК Количество
```

```
ИЗ
    ВТ КАК ВТ
;

УНИЧТОЖИТЬ ВТ
```

Для настройки вариантов соединения таблиц-источников используется вкладка «Связи» конструктора запроса. Данная вкладка появляется при указании нескольких таблиц-источников на вкладке «Таблицы и поля» (рис. 8.9).

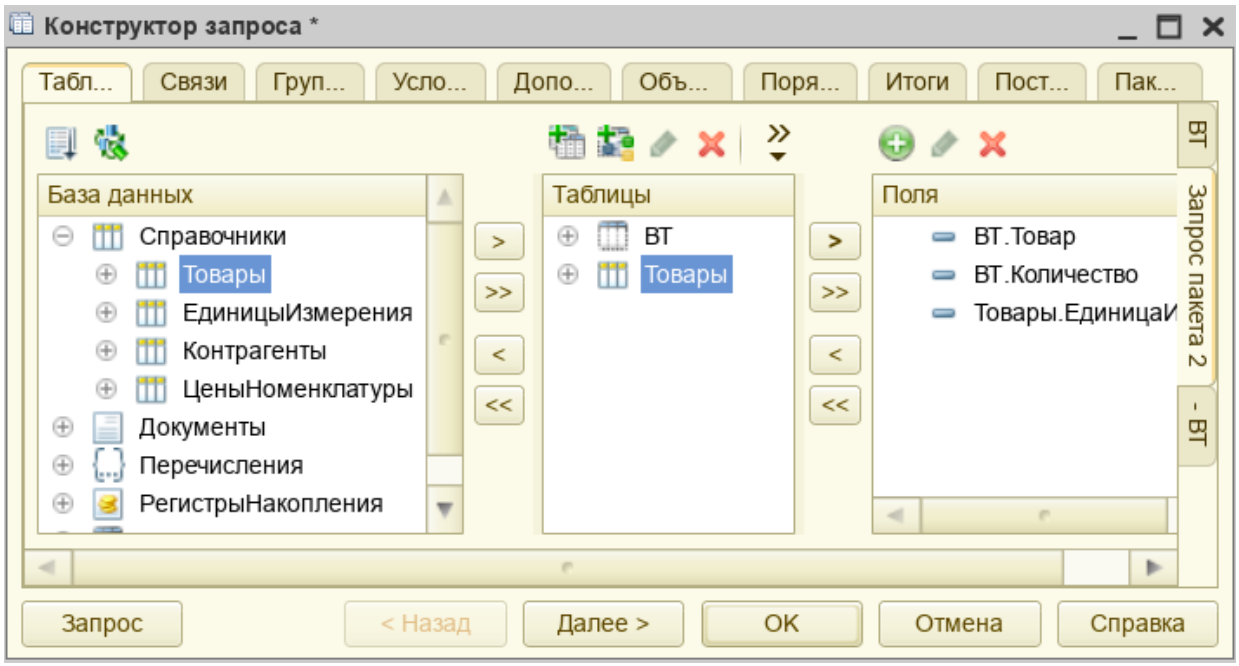


Рис. 8.9. Вкладка «Таблицы и поля» конструктора запроса с несколькими таблицами-источниками

Настройки вкладки «Связи» позволяют определить тип и условие связи (рис. 8.10).

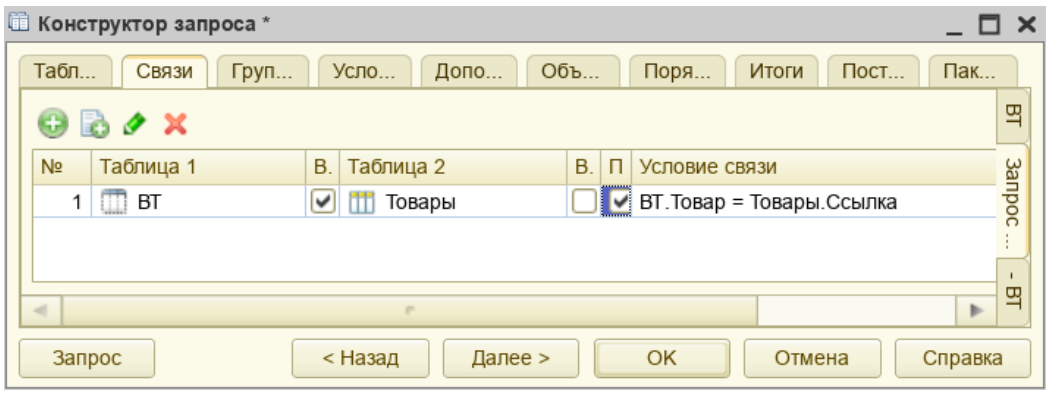


Рис. 8.10. Вкладка «Связи» конструктора запроса

В результате настройки связи между таблицами-источниками будет получен следующий запрос:

```
ВЫБРАТЬ РАЗЛИЧНЫЕ ПЕРВЫЕ 3
    ТоварыОстатки.Товар КАК Товар,
    СУММА(ТоварыОстатки.КоличествоОстаток)
    КАК Количество
ПОМЕСТИТЬ ВТ
ИЗ
    РегистрНакопления.Товары.Остатки(
        &Дата, Поставщик = &Поставщик)
        КАК ТоварыОстатки
ГДЕ
    ТоварыОстатки.КоличествоОстаток = &КоличествоОстаток

СГРУППИРОВАТЬ ПО
    ТоварыОстатки.Товар

УПОРЯДОЧИТЬ ПО
    Товар
АВТОУПОРЯДОЧИВАНИЕ
;

ВЫБРАТЬ
    ВТ.Товар КАК Товар,
    ВТ.Количество КАК Количество,
    Товары.ЕдиницаИзмерения КАК ЕдиницаИзмерения
ИЗ
    ВТ КАК ВТ
        ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Товары КАК Товары
        ПО (ВТ.Товар = Товары.Ссылка)
;

УНИЧТОЖИТЬ ВТ
```

8.4 Пример выполнения задания

В рамках пятой лабораторной работы необходимо научиться работать с языком запросов Платформы. Для это требуется выполнить задание, состоящее из следующих пунктов:

1. Разработать отчет путем обращения к движениям регистра накопления с применением языка запросов Платформы.
2. Выполнить проверку функциональности разработанного отчета.

Замечание: Для построения отчета, возможно, необходима сортировка и свертка данных, извлеченных из регистра накопления. Используйте для этих целей средства языка запросов Платформы.

Замечание: В некоторых случаях для получения необходимых данных требуется обращение к виртуальной таблице регистра накопления.

Рассмотрим процесс выполнения следующего варианта задания, в рамках которого необходимо разработать отчет:

Ведомость поступления товара от поставщиков с учетов возвратов за период с <Дата> по <Дата>

Товар	Поступило		Возвращено		% брака
	Количество	Сумма	Количество	Сумма	
Итого	<>		<>		<>

Замечание: Разработать документы «Поступление товара» и «Возврат поставщику».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Лабораторная работа должна быть выполнена в режиме *обычного приложения*.

Для выполнения задания нет необходимости создавать документы «ПоступлениеТовара», «ВозвратПоставщику» и регистр накопления «Товары», так как они были созданы в рамках третьей и четвертой лабораторных работ соответственно.

Отчет, формирование которого будет осуществляться на основе движений регистра накопления «Товары» с применением языка запросов Платформы, можно создать путем копирования отчета «ВедомостьПоступленияРег». Отчет «ВедомостьПоступленияРег» получен в результате выполнения четвертой лабораторной работы.

Для копирования отчета «ВедомостьПоступленияРег» необходимо выбрать соответствующий узел дерева объектов метаданных, нажать на него правой кнопкой мыши, а затем выбрать пункт меню «Скопировать».

Далее необходимо изменить свойства «Имя» и «Синоним», указав для них значения «ВедомостьПоступленияЗапрос» и «Ведомость поступления (запрос)» соответственно.

Содержимое обработчика КнопкаСформироватьНажатие события «Действие» кнопки «Сформировать» формы отчета «ВедомостьПоступленияЗапрос» состоит из следующих блоков:

1. Создание объекта с типом Запрос.

```
Запрос = Новый Запрос();
```

2. Формирование текста запроса.

```
Запрос.Текст =
"ВЫБРАТЬ
|   ТоварыОбороты.Поставщик КАК Поставщик,
|   ТоварыОбороты.Товар КАК Товар,
|   СУММА(ТоварыОбороты.КоличествоПриход) КАК ПКол,
|   СУММА(ТоварыОбороты.СуммаПриход) КАК ПСум,
|   СУММА(ТоварыОбороты.КоличествоРасход) КАК ВКол,
|   СУММА(ТоварыОбороты.СуммаРасход) КАК ВСум,
|   СУММА(ТоварыОбороты.КоличествоРасход
|       * 100 / ТоварыОбороты.КоличествоПриход)
|       КАК Брак
|ИЗ
|   РегистрНакопления.Товары.Обороты(
|       &НачПериода, &КонПериода, Период, )
|       КАК ТоварыОбороты
|
|СГРУППИРОВАТЬ ПО
|   ТоварыОбороты.Поставщик,
|   ТоварыОбороты.Товар
|
|УПОРЯДОЧИТЬ ПО
|   Поставщик, Товар
|ИТОГИ
|   СУММА(ПКол),
|   СУММА(ПСум),
|   СУММА(ВКол),
|   СУММА(ВСум)
|ПО
|   ОБЩИЕ,
|   Поставщик
|АВТОУПОРЯДОЧИВАНИЕ";
```

Как видно из текста запрос, поля выборки запроса повторяют структуру таблицы значений, используемой в отчете «ВедомостьПоступленияРег». Для получения данных регистра накопления «Товары» происходит обращение к виртуальной таблице «Обороты» данного регистра. Свертка и сортировка таблицы значений заменена на использование операторов

СГРУППИРОВАТЬ ПО и УПОРЯДОЧИТЬ ПО соответственно. Также при группировке строк таблицы результатов запроса используется агрегатная функция СУММА. Процент брака в разрезе товара и поставщика также рассчитывается в рамках запроса. Итоги по отчету формируются с помощью оператора ИТОГИ ... ПО с видом итогов ОБЩИЕ.

3. Установка значений параметров запроса &НачПериода и &КонПериода.

```
Запрос.УстановитьПараметр("НачПериода", НачПериода);
Запрос.УстановитьПараметр("КонПериода", КонПериода);
```

4. Выполнение запроса и вызов функции Выбрать объекта Запрос.

```
Выборка = Запрос.Выполнить().Выбрать();
```

5. Формирование отчета по содержимому таблицы значений.

```
ТабДок = ЭлементыФормы.ПолеТабличногоДокумента1;
ТабДок.Очистить();

Макет = Отчеты.ВедомостьПоступления
    .ПолучитьМакет("Макет");

ОбластьШапка = Макет.ПолучитьОбласть("Шапка");
ОбластьШапка.Параметры.ДатаНач =
    Формат(НачПериода, "ДФ=dd.ММ.уууу");
ОбластьШапка.Параметры.ДатаКон =
    Формат(КонПериода, "ДФ=dd.ММ.уууу");
ТабДок.Вывести(ОбластьШапка);

Если НЕ Выборка.Количество() = 0 Тогда
    // Переменная для хранения ссылки на
    // область с итогами по отчету
    ОбластьИтог = Неопределено;
    Пока Выборка.Следующий() Цикл
        // Переменная для хранения ссылки на
        // текущую область
        Область = Неопределено;
        // Заполнение области с итогами по отчету
        Если Выборка.ТипЗаписи() =
            ТипЗаписиЗапроса.ОбщийИтог Тогда
            ОбластьИтог = Макет
                .ПолучитьОбласть("Итог");
            ОбластьИтог.Параметры.Заполнить(Выборка);
            ОбластьИтог.Параметры.Брак =
                ОбластьИтог.Параметры.ВКол * 100
```

```
                / ОбластьИтог.Параметры.ПКол;
            // Установка области с группировкой
            // по поставщику в качестве текущей области
            ИначеЕсли Выборка.ТипЗаписи() =
                ТипЗаписиЗапроса.ИтогПоГруппировке
                И Выборка.Группировка() = "Поставщик"
            Тогда
                Область = Макет.ПолучитьОбласть("Группа");
            // Установка области с информацией
            // о товаре в качестве текущей области
            Иначе
                Область = Макет.ПолучитьОбласть("Строка");
            КонецЕсли;
            // Заполнение и вывод текущей области
            Если НЕ Область = Неопределено Тогда
                Область.Параметры.Заполнить(Выборка);
                ТабДок.Вывести(Область);
            КонецЕсли;
        КонецЦикла;

        // Вывод итогов по отчету
        Если НЕ ОбластьИтог = Неопределено Тогда
            ТабДок.Вывести(ОбластьИтог);
        КонецЕсли;
    КонецЕсли;

    ТабДок.ТолькоПросмотр = Истина;
    ТабДок.ОтображатьЗаголовки = Ложь;
    ТабДок.ОтображатьСетку = Ложь;
```

После выполнения представленных выше команд сформированный отчет будет иметь вид, изображенный на рисунке 8.11.

Период с: 01.01.2000 по: 31.12.2000

Ведомость поступления товара от поставщиков с учетов возвратов за период с 01.01.2000 по 31.12.2000

	Поступило		Возвращено		
Товар	Количество	Сумма	Количество	Сумма	% брака
Поставщик1					
Товар1	20	2 000			
Поставщик2					
Товар2	13	4 420	2	680	15,38
Поставщик3					
Товар1	30	3 300	10	1 100	33,33
Товар2	40	16 000			
Итого	103		12		11,65

Рис. 8.11. Сформированный отчет

Как видно из рисунка 8.11, содержимое печатной формы отчета «ВедомостьПоступленияЗапрос» идентично содержимому печатной формы отчетов «ВедомостьПоступления» и «ВедомостьПоступленияРег». При этом для формирования большей части отчета используется запрос.

8.4.1 Автоматическое заполнение цены товара при возврате поставщику

В данном разделе будет рассмотрено применение регистра накопления для автоматизации заполнения цены товара при его возврате поставщику с применением запроса для получения данных.

Таким образом, функция ПолучитьСреднююЦенуТовара в модуле объекта документа «ВозвратПоставщику» будет иметь вид:

```
Функция ПолучитьСреднююЦенуТовара(Товар) Экспорт
    // Проверка на заполнение реквизита Поставщик
    Если НЕ ЗначениеЗаполнено(Поставщик) Тогда
        Сообщить("Поле 'Поставщик' не заполнено");
        Возврат 0;
    КонецЕсли;

    // Формирование запроса для получения данных
    // и расчета цены товара по среднему
    Запрос = Новый Запрос();
    Запрос.Текст =
    "ВЫБРАТЬ
    |   ВЫБОР
    |       КОГДА ТоварыОстатки.КоличествоОстаток = 0
    |           ТОГДА 0
    |       ИНАЧЕ ТоварыОстатки.СуммаОстаток
    |           / ТоварыОстатки.КоличествоОстаток
    |   КОНЕЦ КАК Цена
    |ИЗ
    |   РегистрНакопления.Товары.Остатки(
    |           &Дата,
    |           Поставщик = &Поставщик
    |               И Товар = &Товар) КАК ТоварыОстатки";
    Запрос.УстановитьПараметр("Дата", МоментВремени());
    Запрос.УстановитьПараметр("Поставщик", Поставщик);
    Запрос.УстановитьПараметр("Товар", Товар);
    Выборка = Запрос.Выполнить().Выбрать();
```

```
    Если Выборка.Следующий() Тогда
        Возврат Выборка.Цена;
    КонецЕсли;
    Возврат 0;
КонецФункции
```

Как видно из приведенного выше программного кода, алгоритм работы функции стал более простым, все расчеты были перенесены в запрос.

8.4.2 Контроль остатков при возврате товара поставщику

Контроль остатков также можно реализовать с применением языка запросов Платформы. В данном разделе будет рассмотрен пример применения языка запросов Платформы для контроля остатков при проведении документа «ВозвратПоставщику».

Функция ПроверитьОстаткиТоваров будет иметь вид:

```
Функция ПроверитьОстаткиТоваров()
    Запрос = Новый Запрос();
    Запрос.Текст =
    "ВЫБРАТЬ
    |   ВозвратТовараТовары.Товар КАК Товар,
    |   СУММА(ВозвратТовараТовары.Количество)
    |       КАК Количество
    |ПОМЕСТИТЬ ВТТовары
    |ИЗ
    |   Документ.ВозвратТовара.Товары
    |       КАК ВозвратТовараТовары
    |ГДЕ
    |   ВозвратТовараТовары.Ссылка = &ДокументВозврат
    |
    |СГРУППИРОВАТЬ ПО
    |   ВозвратТовараТовары.Товар
    |;
    |ВЫБРАТЬ
    |   ПРЕДСТАВЛЕНИЕ(ВТТовары.Товар) КАК Товар,
    |   ЕСТЬNULL(ТоварыОстатки.КоличествоОстаток, 0)
    |       КАК КоличествоОст,
    |   ВТТовары.Количество КАК Количество
    |ПОМЕСТИТЬ ВТОстатки
    |ИЗ
    |   ВТТовары КАК ВТТовары
    |       ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Товары
    |           .Остатки(
    |               &Дата,
```

```
        |                Поставщик = &Поставщик
        |                И Товар В
        |                    (ВЫБРАТЬ
        |                        ВТ.Товар
        |                    ИЗ
        |                        ВТТовары КАК ВТ))
        |            КАК ТоварыОстатки
        |        ПО ВТТовары.Товар = ТоварыОстатки.Товар
        |;
        |
        |УНИЧТОЖИТЬ ВТТовары
        |;
        |
        |ВЫБРАТЬ
        |   ВТОстатки.Товар КАК Товар,
        |   ВТОстатки.КоличествоОст КАК КоличествоОст,
        |   ВТОстатки.Количество КАК Количество
        |ИЗ
        |   ВТОстатки КАК ВТОстатки
        |ГДЕ
        |   ВТОстатки.КоличествоОст < ВТОстатки.Количество
        |;
        |
        |УНИЧТОЖИТЬ ВТОстатки";
        Запрос.УстановитьПараметр("ДокументВозврат", Ссылка);
        Запрос.УстановитьПараметр("Дата",
            Новый Граница(МоментВремени(),
                ВидГраницы.Исключая));
        Запрос.УстановитьПараметр("Поставщик", Поставщик);
        ТЗОстатки = Запрос.Выполнить().Выгрузить();

        Для Каждого Строка Из ТЗОстатки Цикл
            Если Строка.КоличествоОст = 0 Тогда
                Сообщить("Товар '" + Строка.Товар
                    + "' отсутствует");
                Возврат Ложь;
            КонецЕсли;
            Сообщить("Недостаточно товара '" + Строка.Товар
                + "', в наличии " + Строка.КоличествоОст
                + ", необходимо " + Строка.Количество);
            Возврат Ложь;
        КонецЦикла;
        Возврат Истина;
    КонецФункции
```

Как видно из представленного выше программного кода, основная логика работы по контролю остаток перенесена в запрос.

Представленный запрос является пакетом запросов и состоит из пяти запросов:

1. В рамках первого запроса происходит выборка данных из табличной части «Товары» текущего документа «ВозвратТовара». Полученные данные группируются по товару с использованием агрегатной функции `СУММА` для реквизита «Количество». Результаты отчета помещаются во временную таблицу «ВТТовары».
2. В рамках второго запроса происходит получение остатков из регистра накопления «Товары» для всех товаров из временной таблицы «ВТТовары». Сведения об остатках выбираются из виртуальной таблицы «Остатки» регистра накопления «Товары» с отбором по поставщику. Для соединения данных используется оператор `ЛЕВОЕ СОЕДИНЕНИЕ`. В качестве левой таблицы выступает временная таблица «ВТТовары», а в качестве правой — виртуальная таблица «Остатки» регистра накопления «Товары». Результаты запроса помещаются во временную таблицу «ВТОстатки».
3. В третьем запросе пакета происходит уничтожение временной таблицы «ВТТовары».
4. В рамках четвертого запроса происходит выборка только тех строк таблицы ВТОстатки, в которых нарушается условие контроля остатков — количество остатков товара меньше требуемого количества.
5. В пятом запросе пакета происходит уничтожение временной таблицы «ВТОстатки».

8.5 Контрольные вопросы

1. Для чего в запросе используются псевдонимы?
2. Каким образом можно ограничить количество записей в таблице результатов запроса?
3. Каким образом можно добиться уникальности строк таблицы результатов запроса?
4. В чем особенность обращения к табличным частям из запроса?
5. Что такое вложенные запроса?
6. Для чего используются параметры запроса?

7. Каким образом происходит упорядочивание записей таблицы результатов запроса?
8. Каким образом происходит группировка записей таблицы результатов запроса?
9. Каким образом происходит расчет итого запроса?
10. Какие виды соединения поддерживает язык запросов Платформы?
11. Каким образом происходит объединение результатов запросов?
12. Для чего используется конструктор запросов?
13. Какие виртуальные таблицы регистра накопления существуют и в чем разница между ними?

8.6 Варианты заданий

Лабораторная работа должна быть выполнена в режиме *обычного приложения*.

В представленных вариантах задания необходимо:

1. Разработать отчет путем обращения к движениям регистра накопления с применением языка запросов Платформы.
2. Выполнить проверку функциональности разработанного отчета.

Замечание: Для построения отчета, возможно, необходима сортировка и свертка данных, извлеченных из регистра накопления. Используйте для этих целей средства языка запросов Платформы.

Замечание: В некоторых случаях для получения необходимых данных требуется обращение к виртуальной таблице регистра накопления.

Вариант 1

Ведомость поступления товаров от поставщиков за <Месяц> <Год>

Поставщик	Товаров на сумму	НДС на сумму	Всего от поставщика на сумму	Адрес поставщика
Итого	<>	<>	<>	

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 2

Ведомость поступления товаров от поставщика <Поставщик> за период с <Дата> по <Дата>

Дата	Товар	Склад	Ед. изм.	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого				<>		<>	<>	<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 3

Ведомость поступления товаров на склад <Склад> от поставщиков за <Месяц> <Год>

Товар	Поставщик	Документ	Дата	Цена	Сумма
Итого					<>

Замечание: Строки отчета группируются по поставщикам. В конце группы следует включать строку «Итого по поставщику».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 4

Ведомость поступления товаров на склад <Склад> от поставщика <Поставщик> за период с <Дата> по <Дата>

Дата	Товар	Ед. изм.	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого			<>		<>	<>	<>

Замечание: Строки отчета группируются по товарам. В конце группы следует включать строку «Итого по товару».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 5

Ведомость поступления товара <Товар> на склады от поставщиков за период с <Дата> по <Дата>

Склад	Документ	Дата	Поставщик	Кол-во	Ед. изм.	Цена	Сумма
Итого				<>			<>

Замечание: Строки отчета группируются по поставщикам. В конце группы следует включать строку «Итого по поставщику».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 6

Ведомость поступления товара <Товар> на склады от поставщика <Поставщик> за период с <Дата> по <Дата>

Склад	Документ	Дата	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого			<>		<>	<>	<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документ «Поступление товаров от поставщиков».

Замечание: Использовать стандартную расшифровку.

Вариант 7

Ведомость продажи товаров покупателям за <Месяц> <Год>

Покупатель	Сумма	Сумма НДС	Всего с НДС
Итого	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 8

Ведомость продажи товаров покупателю <Покупатель> за период с <Дата> по <Дата>

Дата	Товар	Склад	Ед. изм.	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого				<>		<>	<>	<>

Замечание: Строки отчета группируются по товарам. В конце группы следует включать строку «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 9

Ведомость продажи товаров со склада <Склад> покупателям за <Месяц> <Год>

Товар	Покупатель	Валюта	Курс	Документ	Дата	Цена	Сумма
Итого							<>

Замечание: Строки отчета группируются по покупателям. В конце группы следует включать строку «Итого по покупателю».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 10

Ведомость продажи товаров со склада <Склад> покупателю <Покупатель> за период с <Дата> по <Дата>

Дата	Товар	Валюта	Курс	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого				<>		<>	<>	<>

Замечание: Строки отчета группируются по товарам. В конце группы следует включать строку «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 11

Ведомость продажи товара <Товар> со складов покупателям за период с <Дата> по <Дата>

Склад	Документ	Дата	Покупатель	Кол-во	Ед. изм.	Цена	Сумма
Итого				<>			<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 12

Ведомость продажи товара <Товара> со складов покупателю <Покупатель> за период с <Дата> по <Дата>

Склад	Документ	Дата	Кол-во	Цена	Сумма	Сумма НДС	Всего с НДС
Итого			<>		<>	<>	<>

Замечание: Строки отчета группируются по складам. В конце группы следует включать строку «Итого по складу».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 13

Ведомость движения товаров за <Месяц> <Год>

Товар	Нач. остаток		Приход		Расход		Кон. остаток	
	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма
Итого	<>	<>	<>	<>	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 14

Ведомость движения товара <Товар> за <Месяц> <Год>

Склад	Начальный остаток	Приход	Расход	Конечный остаток
	Сумма	Сумма	Сумма	Сумма
Итого	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 15

Ведомость движения товаров на складе <Склад> за <Месяц> <Год>

Товар	Нач. остаток		Приход		Расход		Кон. остаток	
	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма	Кол-во	Сумма
Итого	<>	<>	<>	<>	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 16

Ведомость движения товара <Товар> на складе <Склад> за <Месяц> <Год>

Дата	Документ	Номер	Приход		Расход	
			Кол-во	Сумма	Кол-во	Сумма
Итого			<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Реализация товаров покупателям».

Замечание: Использовать стандартную расшифровку.

Вариант 17

Ведомость взаиморасчетов с поставщиками за <Месяц> <Год>

Клиент	Сумма долга на начало периода	Сумма прихода	Сумма расхода	Сумма долга на конец периода
Итого	<>	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Оплата товаров поставщику».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 18

Справка о возвратах товаров от покупателей за период с <Дата> по <Дата>

Дата возврата	Документ	Покупатель	Сумма возврата	Причина возврата
Итого			<>	

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Возврат товаров поставщику».

Замечание: Использовать стандартную расшифровку.

Вариант 19

Ведомость взаиморасчетов с покупателем <Покупатель> за <Месяц> <Год>

Дата	Номер	Документ	Приход	Расход	Остаток
Итого			<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков», «Реализация продукции покупателям» и «Поступление оплаты от покупателей».

Замечание: Использовать стандартную расшифровку.

Вариант 20

Ведомость передачи на реализацию товаров реализаторам за <Месяц> <Год>

Реализатор	Документ	Дата	Товар	Кол-во	Сумма
Итого				<>	<>

Замечание: Данные группируются по реализаторам. В конце группы выводится строка «Итого по реализатору».

Замечание: Разработать документы «Поступление товаров от поставщиков» и «Передача товаров реализаторам».

Замечание: Использовать стандартную расшифровку.

Вариант 21

Ведомость продажи товаров реализатором <Реализатор> за период с <Дата> по <Дата>

Товар	Дата	Документ	Покупатель	Кол-во	Сумма
Итого				<>	<>

Замечание: Данные группируются по товарам. В конце группы выводится строка «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков», «Передача товаров реализаторам» и «Продажа товаров реализатором».

Замечание: Использовать стандартную расшифровку.

Вариант 22

Ведомость продажи товаров реализаторами за <Месяц> <Год>

Реализатор	Товар	Дата	Документ	Покупатель	Сумма
Итого					<>

Замечание: Данные группируются по реализаторам. В конце группы выводится строка «Итого по реализатору».

Замечание: Разработать документы «Поступление товаров от поставщиков», «Передача товаров реализаторам» и «Продажа товаров реализатором».

Замечание: Использовать стандартную расшифровку.

Вариант 23

Ведомость продажи товаров по заявкам клиентов за период с <Дата> по <Дата>

Клиент	Всего товаров в заявках на сумму	Продано по заявкам клиентов на сумму	Осталось продать на сумму
Итого	<>	<>	<>

Замечание: Разработать документы «Поступление товаров от поставщиков», «Заявка покупателю» и «Реализация по заявке».

Замечание: Форму конкретизирующего отчета при расшифровке разработать самостоятельно.

Вариант 24

Ведомость продажи товаров по заявкам клиента <Клиент> за период с <Дата> по <Дата>

Товар	Заказано			Продано			Остаток	
	Док.	Кол-во	Сумма	Док.	Кол-во	Сумма	Кол-во	Сумма
Итого		<>	<>		<>	<>	<>	<>

Замечание: Данные группируются по товарам. В конце группы выводится строка «Итого по товару».

Замечание: Разработать документы «Поступление товаров от поставщиков», «Заявка покупателю» и «Реализация по заявке».

Замечание: Использовать стандартную расшифровку.

9 РЕГИСТРЫ СВЕДЕНИЙ

Регистры сведений в рамках Платформы используются для хранения информации различного типа в разрезе нескольких измерений. Регистр сведений позволяет хранить историю изменения значений своих ресурсов. Также регистр сведений позволяет вносить в него данные не только при проведении документа-регистратора, но и через экранную форму регистра в интерактивном режиме.

В отличие от регистра накоплений, в регистрах сведений ресурсы могут иметь тип, соответствующий как одному из примитивных типов данных (число, строка, дата, булево), так и ссылаться на любой объект метаданных Платформы (элемент справочника, документ и др.).

В регистре сведений не может существовать две записи с одинаковым набором измерений. Таким образом, регистр сведений можно рассматривать как коллекцию элементов «ключ-значение», в которой ключ состоит из значений всех измерений регистра, а значение — совокупность ресурсов и реквизитов.

9.1 Создание и настройка регистра сведений

Для создания регистра сведения необходимо выбрать узел «Регистры сведений» в дереве объектов метаданных, а затем нажать на него правой кнопкой мыши и выбрать пункт «Добавить» в появившемся меню (рис. 9.1).

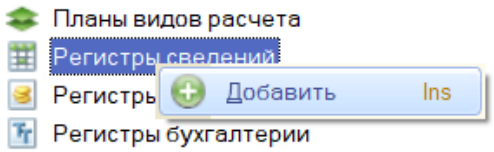


Рис. 9.1. Создание регистра сведений

На рисунке 9.2 представлено окно свойств вновь созданного регистра сведений. На вкладке «Основные» окна свойств регистра сведений можно выбрать настройки периодичности:

- непереодический. Регистр сведений с периодичностью «Непереодический» не позволяет хранить историю изменения значений ресурсов регистра;
- в пределах секунды. Регистр сведений с периодичностью «В пределах секунды» позволяет хранить историю значений ре-

сурсов регистра за счет добавления дополнительного измерения — «Период». При использовании периодичности «В пределах секунды» минимальной единицей различия значений измерения «Период» является секунда;

- в пределах дня. При использовании периодичности «В пределах дня» минимальной единицей различия значений измерения «Период» является день;
- в пределах месяца. При использовании периодичности «В пределах месяца» минимальной единицей различия значений измерения «Период» является месяц;
- в пределах квартала. При использовании периодичности «В пределах квартала» минимальной единицей различия значений измерения «Период» является квартал;

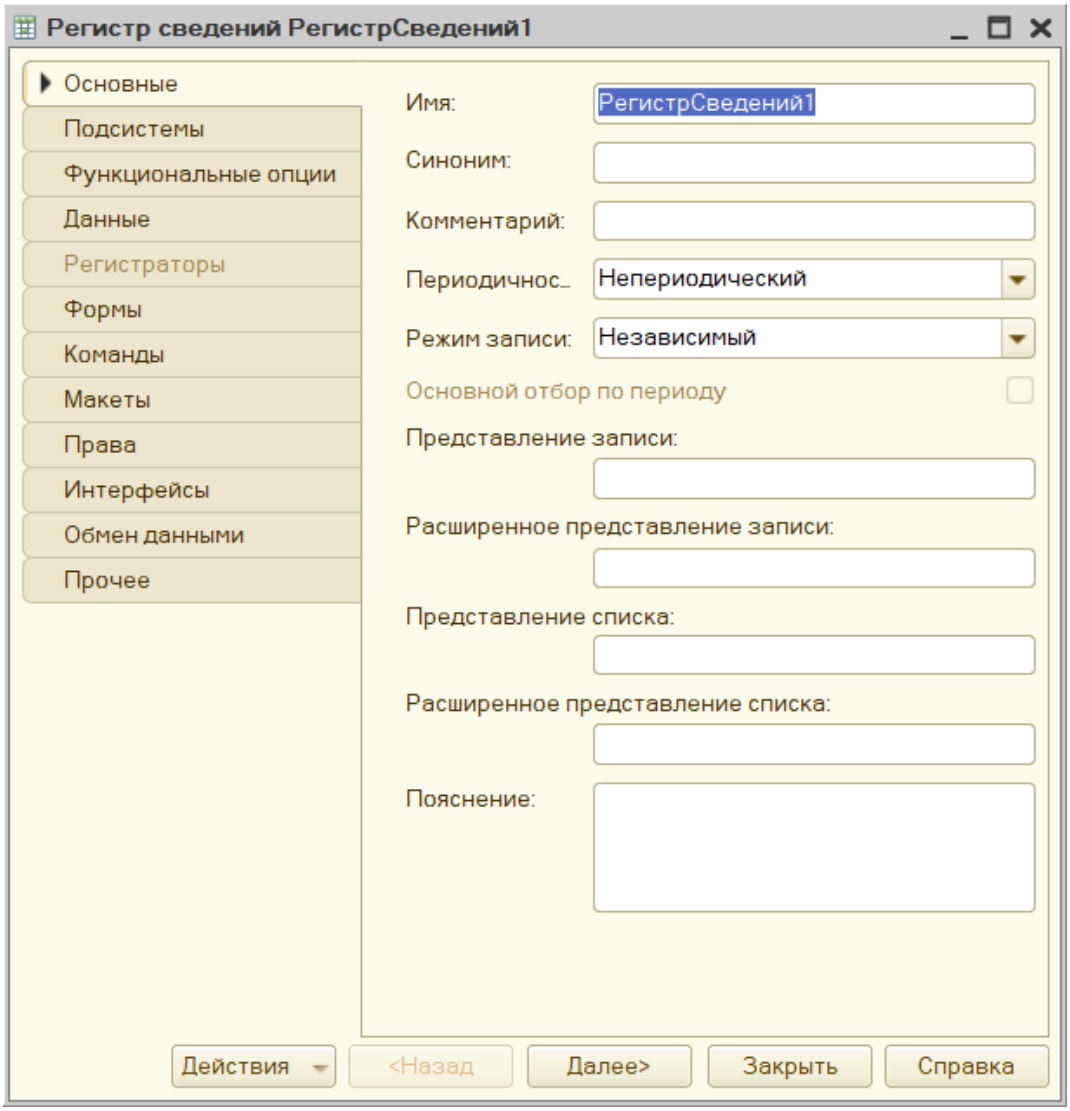


Рис. 9.2. Вкладка «Основные» окна свойств регистра сведений

- в пределах года. При использовании периодичности «В пределах года» минимальной единицей различия значений измерения «Период» является год,

Также на вкладке «Основные» определяется режим записи регистра:

- независимый. Регистр сведений с режимом записи «Независимый» позволяет вносить данные только в интерактивном режиме через экранную форму регистра;
- подчинение регистратору. Регистр сведений с режимом записи «Подчинение регистратору» имеет дополнительное измерение «Регистратор» и позволяет вносить данные при проведении документа-регистратора или с привязкой к такому документу.

Остальные свойства регистра сведений за некоторым исключением аналогичны свойствам регистра накоплений.

Измерения регистра сведений с ссылочным типом данных можно назначить ведущими (рис. 9.3), тогда удаление объекта, на который указывает ссылка, приведет к автоматическому удалению записей регистра сведений, ссылающихся на этот объект.

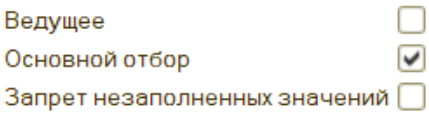


Рис. 9.3. Настройка измерения регистра сведений в качестве ведущего

9.2 Методы работы с регистрами сведений

Для выбора всех записей регистра, у которых реквизит равен определенному значению, используется метод `Выбрать` объекта `РегистрСведенийМенеджер`:

```
Выборка = РегистрыСведений.Регистр.Выбрать();
Пока Выборка.Следующий() Цикл
    Если Выборка.Реквизит1 = Значение Тогда
        ...
    КонецЕсли;
КонецЦикла;
```

В данном случае под реквизитом понимаются стандартные реквизиты, измерения, ресурсы и реквизиты регистра сведений.

Выборку записей регистра сведений также можно ограничить, используя параметры метода Выбрать объекта РегистрСведенийМенеджер:

```
ДатаНач = НачалоГода(Дата(2000, 1, 1));
ДатаКон = КонецГода(Дата(2000, 1, 1));
Отбор = Новый Структура("Измерение1", ЗначениеФильтра);
Выборка = РегистрыСведений.Регистр
    .Выбрать(ДатаНач, ДатаКон, Отбор);
Пока Выборка.Следующий() Цикл
    Сообщить(Выборка.Реквизит2);
КонецЦикла;
```

Для корректной работы отбора по измерению регистра сведений при использовании метода Выбрать объекта РегистрСведенийМенеджер необходимо установить параметр «Индексировать» в значение Индексировать.

Отбор данных регистра сведений осуществляется с помощью методов Получить, ПолучитьПервое, ПолучитьПоследнее, СрезПервых и СрезПоследних объекта РегистрСведенийМенеджер, а также с помощью запросов к основной таблице регистра и его виртуальным таблицам СрезПервых и СрезПоследних.

Методы ПолучитьПервое, ПолучитьПоследнее, СрезПервых и СрезПоследних объекта РегистрСведенийМенеджер, а также виртуальные таблицы СрезПервых и СрезПоследних доступны только для периодических реквизитов (параметр «Периодичность» не равен значению «Непереодический»).

Получение значений ресурсов записи регистра сведений осуществляется с помощью метода Получить объекта РегистрСведенийМенеджер.

```
// Для непериодического регистра
Регистр = РегистрыСведений.Регистр;
Отбор = Новый Структура("Измерение1", Значение);
Ресурсы = Регистр.Получить(Отбор);

// Для периодического регистра
Регистр = РегистрыСведений.Регистр;
Период = Дата(2000, 1, 1);
Отбор = Новый Структура("Измерение1", Значение);
Ресурсы = Регистр.Получить(Период, Отбор);
```

Метод Получить объекта РегистрСведенийМенеджер возвращает структуру, ключи которой совпадают с именами ресурсов. Для использования метода Получить необходимо указать значение для

всех измерений регистра сведений. При использовании метода Получить результат будет пустым, если записей с такой комбинацией измерений и периода (для периодических регистров) не существует.

Получение значений ресурсов записи регистра сведений, соответствующей указанным значениям отбора, на текущую или более раннюю дату осуществляется с помощью метода ПолучитьПервое объекта РегистрСведенийМенеджер.

```
Регистр = РегистрыСведений.Регистр;
Период = Дата(2000, 1, 1);
Отбор = Новый Структура("Измерение1", Значение);
Ресурсы = Регистр.ПолучитьПервое(Период, Отбор);
```

Если значение параметра «Период» не указано, то будут возвращены значения ресурсов самой первой записи регистра.

Для получения значений ресурсов записи регистра сведений на текущую или более позднюю дату с учетом отбора используется метод ПолучитьПоследнее объекта РегистрСведенийМенеджер.

```
Регистр = РегистрыСведений.Регистр;
Период = Дата(2000, 1, 1);
Отбор = Новый Структура("Измерение1", Значение);
Ресурсы = Регистр.ПолучитьПоследнее(Период, Отбор);
```

Если значение параметра «Период» не указано, то будут возвращены значения ресурсов самой последней записи регистра.

Получить наиболее ранние записи регистра, соответствующие установленным в параметрах метода значениям ключевых полей, можно с помощью метода СрезПервых объекта РегистрСведенийМенеджер.

```
Регистр = РегистрыСведений.Регистр;
Период = Дата(2000, 1, 1);
Отбор = Новый Структура("Измерение1", Значение);
ЗаписиРегистра = Регистр.СрезПервых(Период, Отбор);
```

Для получения наиболее поздних записей регистра с учетом отбора можно использовать метод СрезПоследних объекта РегистрСведенийМенеджер.

```
Регистр = РегистрыСведений.Регистр;
Период = Дата(2000, 1, 1);
Отбор = Новый Структура("Измерение1", Значение);
ЗаписиРегистра = Регистр.СрезПоследних(Период, Отбор);
```

Для создания новой или удаления существующей записи регистра сведений используются методы Записать и Удалить соответственно объекта РегистрСведенийМенеджер.

```
// Формирование новой записи
Регистр = РегистрыСведений.Регистр
    .СоздатьМенеджерЗаписи();
Регистр.Период = ТекущаяДата();
Регистр.Измерение1 = Значение1;
Регистр.Измерение2 = Значение2;
Регистр.Ресурс1 = 1;
Регистр.Ресурс2 = "Строка";
Регистр.Ресурс3 = Спр.Ссылка;
Регистр.Записать();

// Удаление всех записей регистра сведений
Выборка = РегистрыСведений.Регистр.Выбрать();
Пока Выборка.Следующий() Цикл
    МенеджерЗаписи = Выборка.ПолучитьМенеджерЗаписи();
    МенеджерЗаписи.Удалить();
КонецЦикла
```

Данные методы доступны только для регистра сведений с режимом записи «Независимый».

Для работы с коллекцией записей регистра сведений используется тип данных РегистрСведенийНаборЗаписей. Набор записей позволяет обрабатывать все множество данных в рамках одной транзакции: создание, чтение, изменение, удаление.

```
// Создание пустого набора записей
Набор = РегистрыСведений.Регистр.СоздатьНаборЗаписей();
// Установка периода выборки и отбора по значению измерения
Набор.Отбор.Период.Значение = ТекущаяДата();
Набор.Отбор.Измерение1.Значение = Значение;
// Включение выборки за период
Набор.Отбор.Период.Использование = Истина;
// Включение отбора
Набор.Отбор.Измерение1.Использование = Истина;
// Добавление записи в набор (не в информационную базу)
НовЗапись = Набор.Добавить();
НовЗапись.Период = Дата(2000, 1, 1);
НовЗапись.Измерение1 = НовоеЗначение;
НовЗапись.Курс = НовыйКурс;
// Запись набора в информационную базу
Набор.Записать(Истина);
```

Если отбор в наборе записей не установлен, то такой набор перепишет весь регистр. Чтобы этого не произошло, необходимо

установить отбор по измерениям регистра, а для периодических регистров дополнительно — по полю «Период», если же регистр имеет периодичность «По позиции регистратора», то еще и по полю «Регистратор».

Параметр метода Записать объекта РегистрСведенийНаборЗаписей определяет режим замещения существующей записи в соответствии с текущими установками отбора:

- Истина — перед записью существующие записи будут удалены,
- Ложь — записи будут дописаны к уже существующим в информационной базе записям.

Набор записей также позволяет обращаться к уже существующим записям регистра сведений.

```
// Создание пустого набора записей
Набор = РегистрыСведений.Регистр.СоздатьНаборЗаписей();
// Установка периода выборки
Набор.Отбор.Период.Значение = ТекущаяДата();
Набор.Отбор.Период.Использование = Истина;
//Чтение записей из информационной базы
Набор.Прочитать();
Если НЕ Набор.Выбран() Тогда
    Сообщить("Набор не был прочитан");
КонецЕсли;
Для Каждого Запись Из Набор Цикл
    ...
КонецЦикла;
```

Записи набора можно выгружать в таблицу значений или загружать записи из таблицы значений в набор с помощью методов Выгрузить и Загрузить соответственно объекта РегистрСведенийНаборЗаписей.

```
Набор.Прочитать ();
ТаблицаЗначений = Набор.Выгрузить();
...
Набор.Загрузить(ТаблицаЗначений);
Набор.Записать();
```

При использовании метода Выгрузить структура полученной таблицы значений полностью соответствует структуре записи регистра сведений. При использовании метода Загрузить в поля записи регистра сведений заносятся значения только тех колонок таблицы значений, имена которых совпадают с именами полей записи.

Аналогично работе с таблицами значений набор записей регистра сведений позволяет работать с массивами с помощью методов `ВыгрузитьКолонку` и `ЗагрузитьКолонку` объекта `РегистрСведенийНаборЗаписей`.

```
МассивКурсов = Набор.ВыгрузитьКолонку("Измерение1");
...
Набор.ЗагрузитьКолонку(Массив, "Измерение1");
```

9.3 Пример выполнения задания

В рамках шестой лабораторной работы необходимо научиться работать с объектом метаданных «РегистрыСведений». Для это требуется выполнить задание, состоящее из следующих пунктов:

1. В основном справочнике один из реквизитов объявлен периодическим (помечен символом «*»). Система должна иметь возможность хранить историю изменения значений периодического реквизита в регистре сведений. Структуру регистра сведений разработать самостоятельно.
2. Из структуры основного справочника (вкладка «Данные» свойств справочника) периодический реквизит необходимо удалить. В форме элемента основного справочника требуется добавить реквизит формы (вкладка «Реквизиты») с именем, соответствующим названию периодического реквизита, а затем связать его с оставшимся от удаленного реквизита справочника элементом управления (поле ввода).
3. При открытии формы элемента основного справочника в поле ввода созданного реквизита формы должно отображаться последнее значение периодического реквизита, полученное из регистра сведений.
4. При изменении значения периодического реквизита его необходимо сохранять в регистре сведений.
5. Сформировать отчет, вызываемый из формы списка основного справочника, для отображения истории изменения периодического реквизита выбранного элемента справочника. Отчет должен иметь следующую структуру:

Дата	Значение реквизита

6. Модифицировать отчеты, разработанные в рамках второй лабораторной работы, для работы с периодическим реквизитом.

Рассмотрим процесс выполнения следующего варианта задания:

Основной справочник — Товары:

- код товара;
- наименование товара;
- базовая единица измерения (справочник);
- **минимальный остаток***;
- ставка НДС (перечисление);
- табличная часть:
  - поставщик (справочник);
  - адрес поставщика;
  - закупочная цена;
  - дата последней поставки.

Лабораторная работа должна быть выполнена в режиме *обычного приложения*.

В задании сказано, что необходимо разработать структуру регистра сведений для хранения истории изменения значений периодического реквизита.

Таким образом, содержимое вкладки «Основные» регистра сведений «ИсторияЗначений» будет иметь вид, представленный на рисунке 9.4.

Как видно из рисунка 9.4, регистр сведений «ИсторияЗначений» является независимым и периодическим в пределах секунды. Структура данных регистра представлена на рисунке 9.5.

Регистр сведений «ИсторияЗначений» имеет следующую структуру:

1. Измерение «Товар» — СправочникСсылка.Товары.
2. Ресурс «Значение» — Число(10, 0, Неотрицательное).

Далее необходимо удалить реквизит «МинимальныйОстаток» справочника «Товары». После удаления реквизита «МинимальныйОстаток» вкладка «Данные» справочника «Товары» будет иметь вид, представленный на рисунке 9.6.

Также в форме элемента справочника «Товары» необходимо создать реквизит формы с именем «МинимальныйОстаток» с типом данных Число(10, 0, Неотрицательное) (рис. 9.7).

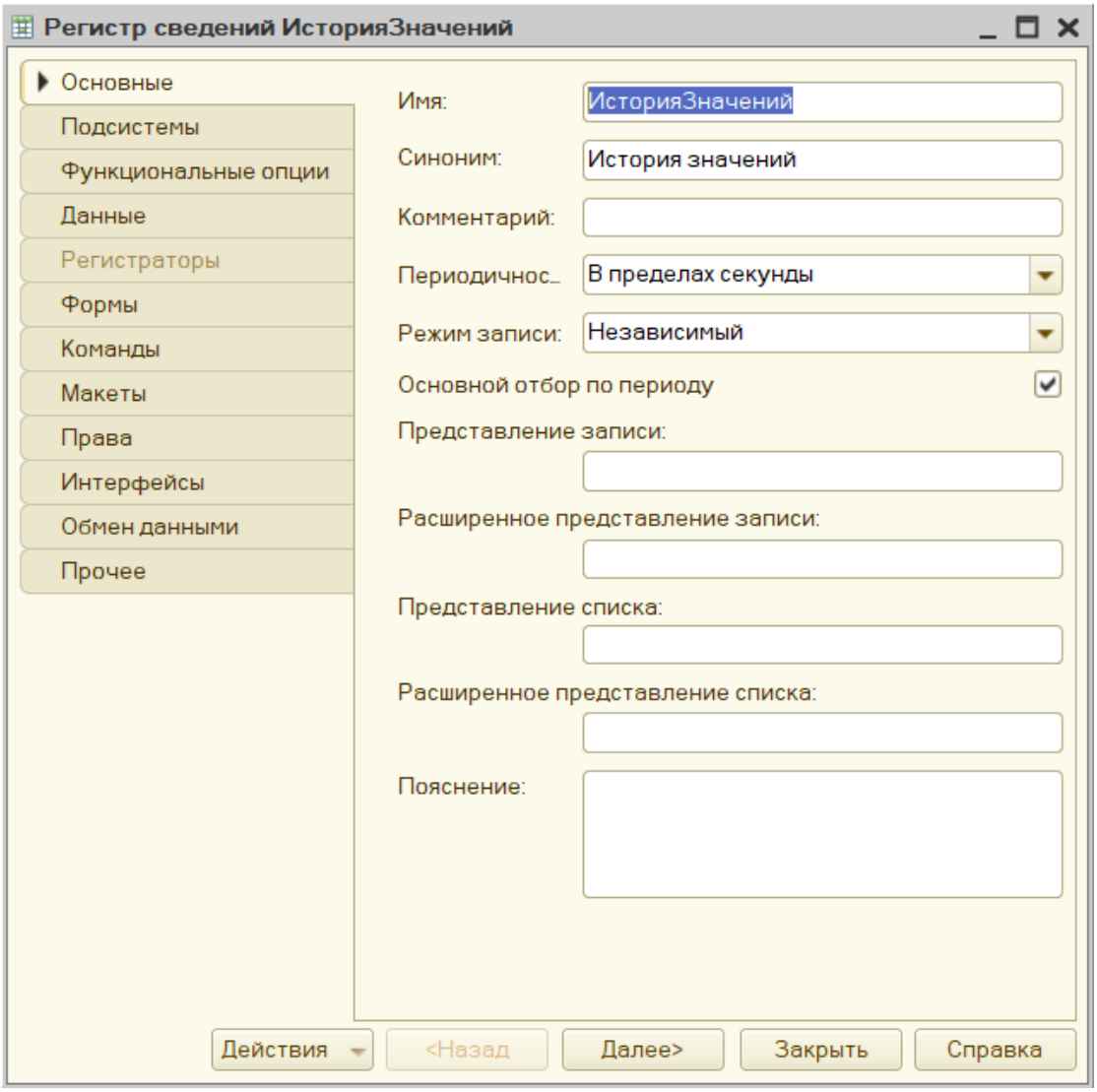


Рис. 9.4. Вкладка «Основные» регистра сведений «ИсторияЗначений»

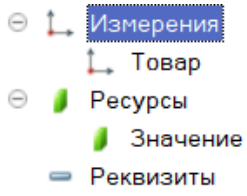


Рис. 9.5. Вкладка «Данные» регистра сведений «ИсторияЗначений»

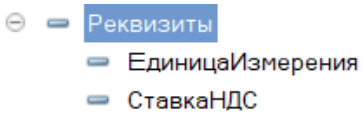


Рис. 9.6. Вкладка «Данные» справочника «Товары»

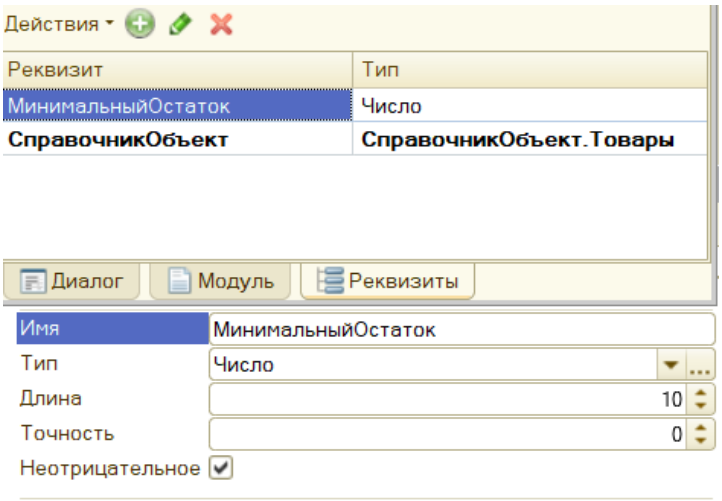


Рис. 9.7. Вкладка «Реквизиты» элемента формы справочника «Товары»

Затем необходимо связать элемент управления (поле ввода), оставшееся от удаленного реквизита справочника «МинимальныйОстаток», с вновь созданным реквизитом формы «МинимальныйОстаток». Для этого необходимо на вкладке «Диалог» формы элемента справочника «Товары» выбрать соответствующий элемент управления и установить в качестве значения свойства «Данные» значение «МинимальныйОстаток» (рис. 9.8).

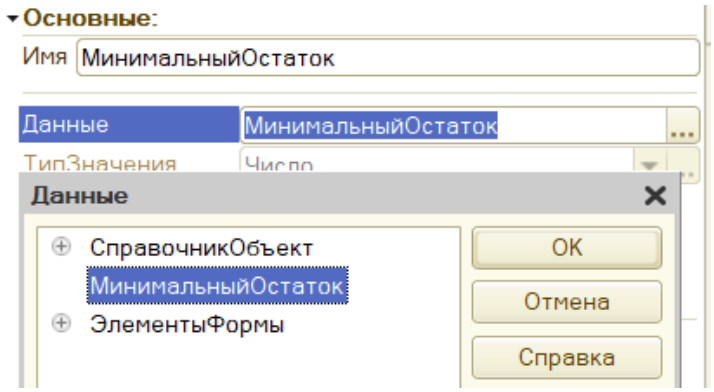


Рис. 9.8. Настройка элемента управления для реквизита формы «МинимальныйОстаток»

Для чтения значения периодического реквизита «МинимальныйОстаток» из регистра сведений при открытии формы элемента справочника «Товары» следует создать обработчик события `ПриОткрытии`. Тело обработчика будет иметь вид:

```
Процедура ПриОткрытии()
    // Не использовать для новых объектов
    Если ЭтоНовый() Тогда
```

```
            Возврат;
        КонецЕсли;
        // Запрос к виртуальной таблице СрезПоследних
        // регистра сведений ИсторияЗначений
        Запрос = Новый Запрос();
        Запрос.Текст =
        "ВЫБРАТЬ
        |   ИсторияЗначенийСрезПоследних.Значение
        |       КАК МинимальныйОстаток
        |ИЗ
        |   РегистрСведений.ИсторияЗначений
        |       .СрезПоследних(&Дата, Товар = &Товар)
        |               КАК ИсторияЗначенийСрезПоследних";
        Запрос.УстановитьПараметр("Дата", ТекущаяДата());
        Запрос.УстановитьПараметр("Товар", Ссылка);
        Результат = Запрос.Выполнить().Выбрать();
        // Получение последнего значения
        Если Результат.Следующий() Тогда
            МинимальныйОстаток = Результат.МинимальныйОстаток;
        КонецЕсли;
    КонецПроцедуры
```

Сохранять значение периодического реквизита в регистр сведений следует в обработчике события ПриЗаписи. Таким образом, тело обработчика будет выглядеть следующим образом:

```
    Процедура ПриЗаписи(Отказ)
        Рег = РегистрыСведений.ИсторияЗначений
            .СоздатьМенеджерЗаписи();
        Рег.Период = ТекущаяДата();
        Рег.Товар = Ссылка;
        Рег.Значение = МинимальныйОстаток;
        Рег.Записать();
    КонецПроцедуры
```

Не следует записывать значение периодического реквизита в обработчике события ПриИзменении элемента управления «МинимальныйОстаток», так как пользователь может передумать вносить изменения в элемент справочника, а данные в регистр сведения будут записаны. Также может возникнуть проблема с новыми элементами справочника «Товары», так как для них значение реквизита «Ссылка» будет пустым, что приведет к созданию ошибочных записей в регистре сведений «ИсторияИзменений».

Для унификации поведения формы элемента справочника «Товары» при работе с реквизитами справочника и периодическим реквизитом внесем следующие изменения:

1. Форма элемента (добавить первой строкой):

```
Перем МодификацияРеквизита;
```

2. Обработчик ПриОткрытии:

```
Процедура ПриОткрытии()
    МодификацияРеквизита = Ложь;
    Если ЭтоНовый() Тогда
    ...
КонецПроцедуры
```

3. Обработчик ПриЗаписи:

```
Процедура ПриЗаписи(Отказ)
    Если НЕ МодификацияРеквизита Тогда
        Возврат;
    КонецЕсли;
    Рег = РегистрыСведений.ИсторияЗначений
        .СоздатьМенеджерЗаписи();
    ...
    Рег.Записать();
    МодификацияРеквизита = Ложь;
КонецПроцедуры
```

Далее необходимо добавить обработчик события ПриИзменении для элемента управления «МинимальныйОстаток», имеющий следующий вид:

```
Процедура МинимальныйОстатокПриИзменении(Элемент)
    МодификацияРеквизита = Истина;
    ЭтаФорма.Модифицированность = Истина;
КонецПроцедуры
```

Для вывода отчета для отображения истории изменения периодического реквизита выбранного элемента справочника используется макет, представленный на рисунке 9.9.

		1	2	3
Шапка	1			
	2		<История периодического реквизита товара [Товар]>	
	3			
	4		Период	Минимальный остаток
	5			
Строка	6		<Период>	<МинимальныйОстаток>

Рис. 9.9. Макет отчета для отображения истории изменения периодического реквизита

Также на форму списка справочника «Товары» была добавлена кнопка «История» для формирования и вывода отчета для

отображения истории изменения периодического реквизита. Обработчик нажатия на кнопку «История» будет иметь вид:

```
Процедура ДействияФормыИстория(Кнопка)
    // Получение выбранного элемента формы списка
    ТС = ЭлементыФормы.СправочникСписок.ТекущаяСтрока;
    // Игнорировать группы справочника
    Если ТС.ЭтоГруппа Тогда
        Возврат;
    КонецЕсли;

    // Запрос к регистру сведений на получение
    // всех значений ресурса Значение в разрезе периода
    // для выбранного товара
    Запрос = Новый Запрос;
    Запрос.Текст =
    "ВЫБРАТЬ
    |   ИсторияЗначений.Период КАК Период,
    |   ИсторияЗначений.Значение КАК МинимальныйОстаток
    |ИЗ
    |   РегистрСведений.ИсторияЗначений КАК ИсторияЗначений
    |ГДЕ
    |   ИсторияЗначений.Товар = &Товар";
    Запрос.УстановитьПараметр("Товар", ТС.Ссылка);
    Результат = Запрос.Выполнить().Выбрать();

    // Создание табличного документа
    ТабДок = Новый ТабличныйДокумент();
    Макет = Справочники.Товары
        .ПолучитьМакет("МакетИстория");
    // Формирование шапки отчета
    ОбластьШапка = Макет.ПолучитьОбласть("Шапка");
    ОбластьШапка.Параметры.Товар = ТС.Наименование;
    ТабДок.Вывести(ОбластьШапка);

    // Формирование строк отчета
    Пока Результат.Следующий() Цикл
        ОбластьСтрока = Макет.ПолучитьОбласть("Строка");
        ОбластьСтрока.Параметры.Заполнить(Результат);
        ТабДок.Вывести(ОбластьСтрока);
    КонецЦикла;

    // Настройка и показ отчета
    ТабДок.ТолькоПросмотр = Истина;
    ТабДок.ОтображатьЗаголовки = Ложь;
    ТабДок.ОтображатьСетку = Ложь;
    ТабДок.Показать("История периодического реквизита");
КонецПроцедуры
```

Для модификации отчета, формирующего печатную форму справочника «Товары», необходимо внести следующие изменения в форму списка справочника «Товары»:

1. Добавление функции для получения самого позднего (последнего) значения периодического реквизита «Минимальный-Остаток» из регистра сведений для указанного товара:

```
Функция ПолучитьЗначениеПериодическогоРеквизита(Товар)
    ОтборТовар = Новый Структура("Товар");
    ОтборТовар.Товар = Товар;
    Рег = РегистрыСведений.ИсторияЗначений
        .ПолучитьПоследнее(ТекущаяДата(), ОтборТовар);
    Возврат Рег.Значение;
КонецФункции
```

2. Модификация метода ДействияФормыПечать:

```
Процедура ДействияФормыПечать(Кнопка)
    ТабДок = Новый ТабличныйДокумент();
    ...
    Пока Спр.Следующий() Цикл
        Если Спр.ЭтоГруппа Тогда
            ...
        Иначе
            ОбластьЭлемент = Макет
                .ПолучитьОбласть("Элемент");
            ОбластьЭлемент.Параметры.Заполнить(Спр);
            ОбластьЭлемент.Параметры
                .МинимальныйОстаток =
            ПолучитьЗначениеПериодическогоРеквизита(
                Спр.Ссылка);
            ТабДок.Вывести(ОбластьЭлемент,
                Спр.УровеньВВыборке());
        КонецЕсли;
    КонецЦикла;
    ...
    ТабДок.Показать("Список товаров");
КонецПроцедуры
```

Для модификации отчета «ОтчетПоТоварам» для работы с регистром сведений следует заменить строку `Спр = Справочники.Товары.ВыбратьИерархически();` в обработчике кнопки «Сформировать» отчета «ОтчетПоТоварам» на строки:

```
Запрос = Новый Запрос();
Запрос.Текст =
```

```
"ВЫБРАТЬ
|   Товары.Ссылка КАК Ссылка,
|   Товары.ЭтоГруппа КАК ЭтоГруппа,
|   Товары.Код КАК Код,
|   Товары.Наименование КАК Наименование,
|   Товары.ЕдиницаИзмерения КАК ЕдиницаИзмерения,
|   Товары.СтавкаНДС КАК СтавкаНДС,
|   ИсторияЗначенийСрезПоследних.Значение
|       КАК МинимальныйОстаток
|ИЗ
|   Справочник.Товары КАК Товары
|       ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ИсторияЗначений
|           .СрезПоследних(&Дата, )
|               КАК ИсторияЗначенийСрезПоследних
|       ПО ИсторияЗначенийСрезПоследних.Товар =
|           Товары.Ссылка
|
|УПОРЯДОЧИТЬ ПО
|   ЭтоГруппа ИЕРАРХИЯ,
|   Наименование
|АВТОУПОРЯДОЧИВАНИЕ";
Запрос.УстановитьПараметр("Дата", ТекущаяДата());
Спр = Запрос.Выполнить().Выбрать();
```

Представленный выше запрос позволяет сформировать отчет нужной структуры с учетом хранения значения периодического реквизита «МинимальныйОстаток» в регистре сведений «ИсторияЗначений».

Содержимое регистра сведений «ИсторияЗначений» представлено на рисунке 9.10.

Период	Товар	Значе…
01.01.2000 7:58:45	Товар2	11
01.01.2000 7:58:55	Товар3	33
01.01.2000 7:58:59	Товар2	22
01.01.2000 7:59:03	Товар1	100
01.01.2000 7:59:46	Товар4	200

Рис. 9.10. Содержимое регистра сведений «ИсторияЗначений»

Внешний вид отчета для отображения истории изменения периодического реквизита выбранного элемента справочника представлен на рисунке 9.11.

Печатная форма модифицированного отчета «ОтчетПоТоварам» представлена на рисунке 9.12.

История периодического реквизита товара Товар2

Период	Минимальный остаток
01.01.2000 7:58:45	11
01.01.2000 7:58:59	22

Рис. 9.11. Внешний вид отчета для отображения истории изменения периодического реквизита

Список товаров на

Код	Наименование	Единица измерения	Минимальный остаток	Ставка НДС
000000001	***Ходовые товары***			
000000002	Товар1	шт.	100	10%
000000003	Товар2	шт.	22	20%
000000004	Товар3	шт.	33	20%
000000005	Товар4	шт.	200	20%
000000006	Товар5	шт.		20%

Рис. 9.12. Печатная форма модифицированного отчета «ОтчетПоТоварам»

9.4 Контрольные вопросы

1. Объясните принцип работы регистра сведений.
2. Какие режимы записи регистра сведений существуют и в чем разница между ними?
3. Какие виды периодичности регистра сведений существуют и в чем разница между ними?

9.5 Варианты заданий

Лабораторная работа должна быть выполнена в режиме *обычного приложения*.

В представленных вариантах задания необходимо:

1. В основном справочнике один из реквизитов объявлен периодическим (помечен символом «*»). Система должна иметь возможность хранить историю изменения значений периодического реквизита в регистре сведений. Структуру регистра сведений разработать самостоятельно.
2. Из структуры основного справочника (вкладка «Данные» свойств справочника) периодический реквизит необходимо удалить. В форме элемента основного справочника требуется добавить реквизит формы (вкладка «Реквизиты») с именем, соответствующим названию периодического реквизита, а затем связать его с оставшимся от удаленного реквизита справочника элементом управления (поле ввода).

3. При открытии формы элемента основного справочника в поле ввода созданного реквизита формы должно отображаться последнее значение периодического реквизита, полученное из регистра сведений.
4. При изменении значения периодического реквизита его необходимо сохранять в регистре сведений.
5. Сформировать отчет, вызываемый из формы списка основного справочника, для отображения истории изменения периодического реквизита выбранного элемента справочника. Отчет должен иметь следующую структуру:

Дата	Значение реквизита

6. Модифицировать отчеты, разработанные в рамках второй лабораторной работы, для работы с периодическим реквизитом.

Вариант 1

Основной справочник — Основные средства (ОС):

- инвентарный номер ОС;
- наименование товара ОС;
- **основное материально-ответственное лицо*** (справочник «Сотрудники»);
- сумма первоначального износа;
- дата ввода в эксплуатацию;
- год выпуска;
- срок эксплуатации (в мес.);
- табличная часть (список подразделений, которым принадлежало ОС в течение времени эксплуатации):
  - подразделение (справочник «Подразделения»),
  - дата появления в подразделении.

Вариант 2

Основной справочник — Товары:

- код товара;
- наименование товара;
- **основная единица измерения*** (справочник «Единицы измерения»);

- тип товара (перечисление «Тип товара»: весовой, невесовой);
- табличная часть (список возможных единиц измерения каждого товара):
  - ед. измерения (справочник «Единицы измерения»)
  - коэффициент пересчета относительно основной единицы измерения.

Вариант 3

Основной справочник — Сотрудники:

- табельный номер сотрудника;
- ФИО сотрудника;
- тип сотрудника (перечисление «ТипСотрудника»: штатный, внутренний совместитель, внешний совместитель);
- дата приема на работу;
- **оклад***;
- является членом профсоюза;
- табличная часть (список детей сотрудника):
  - ФИО ребенка;
  - пол (перечисление «Пол»);
  - дата рождения.

Вариант 4

Основной справочник — Контрагенты:

- код контрагента;
- наименование контрагента;
- адрес контрагента;
- **основной расчетный счет*** (справочник «Расчетные счета»);
- банк (справочник «Банки»);
- телефон;
- размер кредита;
- табличная часть (список договоров контрагента):
  - код договора;
  - наименование договора;
  - дата заключения договора;
  - условия договора.

Вариант 5

Основной справочник — Транспортные средства:

- тип транспортного средства (перечисление «ТипТранспортногоСредства»: легковой, грузовой, специальный);
- наименование транспортного средства;
- объем двигателя;
- мощность двигателя;
- дата выпуска;
- **водитель*** (справочник «Сотрудники»);
- табличная часть (график прохождения технического осмотра транспортного средства):
  - номер по порядку;
  - дата прохождения;
  - примечание.

Вариант 6

Основной справочник — Регистрационные карточки больного:

- код больного;
- ФИО больного;
- **место работы***;
- дата рождения;
- пол (перечисление «Пол»);
- табличная часть (история болезни больного):
  - болезнь (справочник «Болезни»);
  - дата начала заболевания;
  - дата выздоровления.

Вариант 7

Основной справочник — Фракции в государственной думе:

- код фракции;
- наименование фракции;
- **лидер фракции*** (справочник «Физические лица»);
- табличная часть (список депутатов фракции):
  - депутат (справочник «Физические лица»);
  - год рождения;
  - регион (справочник «Регионы»);
  - оклад.

Вариант 8

Основной справочник — Персональные компьютеры предприятия:

- процессор (справочник «Процессоры»);
- частота процессора;
- **объем оперативной памяти***;
- наличие звуковой карты;
- размер диагонали монитора;
- табличная часть (список цен компьютера):
  - тип цены (перечисление «Тип цены»: оптовая, розничная, закупочная);
  - сумма;
  - валюта (справочник «Валюта»).

Вариант 9

Основной справочник — Библиографические единицы:

- тип библиографической единицы (перечисление «тип библиографической единицы»: книга, журнал, пособие, электронный);
- автор (справочник «Физические лица»);
- наименование библиографической единицы;
- год издания;
- издательство (справочник «Издательства»);
- **тираж***;
- табличная часть (список библиографических ссылок на библиографическую единицу):
  - автор ссылки (справочник «Физические лица»);
  - наименование работы автора ссылки;
  - год издания работы.

Вариант 10

Основной справочник — Факультеты университета:

- наименование факультета;
- **декан*** (справочник «Сотрудники»);
- количество студентов на факультете;
- год основания;
- табличная часть (список кафедр факультета):
  - наименование кафедры;

  - заведующий кафедрой (справочник «Сотрудники»);
  - количество штатных единиц.

Вариант 11

Основной справочник — Кафедры университета:

- наименование кафедры;
- **заведующий кафедрой*** (справочник «Сотрудники»);
- количество студентов на кафедре;
- год основания;
- табличная часть (список дисциплин, изучаемых на кафедре):
  - дисциплина (справочник «Дисциплины»);
  - преподаватель (справочник «Сотрудники»);
  - количество часов;
  - наличие экзамена;
  - наличие курсовой работы.

Вариант 12

Основной справочник — Трудовые книжки:

- код сотрудника;
- ФИО сотрудника;
- дата рождения;
- образование (перечисление «ВидОбразования»: высшее, неполное высшее и т. д.);
- **основное место работы*** (справочник «Подразделения»);
- табличная часть (список мест работы сотрудника):
  - дополнительное место работы (справочник «Подразделения»);
  - дата приема;
  - дата увольнения;
  - оклад.

Вариант 13

Основной справочник — Виды оплат и удержаний:

- код вида оплаты/удержания;
- наименование вида оплаты/удержания;
- **тип вида оплаты/удержания*** (перечисление «ТипВидаОплатыУдержания»: разовый, периодический);

- счет (корреспондирующий для счета 70) (справочник «БухгалтерскиеСчета»);
- табличная часть (список правил удержания налогов и взносов):
  - налог/взнос (перечисление «НалогиВзносы»: НДФЛ, профсоюзный взнос, алименты);
  - признак удержания налога/взноса.

Вариант 14

Основной справочник — Автомобили:

- инвентарный номер;
- наименование;
- **марка*** (перечисление «МаркаАвтомобиля»);
- грузоподъемность;
- табличная часть (список водителей, работающих на данном автомобиле):
  - водитель (справочник «Сотрудники»);
  - дата начала работы;
  - дата окончания работы.

Вариант 15

Основной справочник — Сотрудники:

- табельный номер сотрудника;
- ФИО сотрудника;
- тип сотрудника (перечисление «ТипСотрудника»: штатный, внутренний совместитель, внешний совместитель);
- дата приема на работу;
- **оклад***;
- является членом профсоюза;
- табличная часть (список долговременных доплат и удержаний сотрудника):
  - вид расчета (справочник «ВидыРасчетов»);
  - дата начала действия;
  - дата окончания действия;
  - величина (сумма, если расчет вводится фиксированной суммой, процент, если результат расчета формируется как процент от расчетной базы);

Вариант 16

Основной справочник — Виды работ:

- код вида работы;
- название вида работ;
- **объект, для которого предназначен данный вид работы*** (справочник «Объекты»);
- табличная часть (список норм и расценок для каждого вида работ):
  - ед. измерения (справочник «Единицы измерения»);
  - норма;
  - расценка.

Вариант 17

Основной справочник — Склады:

- номер склада;
- название склада;
- занимаемая площадь (m^2);
- адрес склада;
- **тип склада*** (перечисление «ТипСклада»: оптовый, розничный, неавтоматизированная рабочая точка);
- табличная часть (список материально-ответственных лиц склада):
  - сотрудник (справочник «Сотрудники»);
  - домашний адрес.

Вариант 18

Основной справочник — Нормы расхода ГСМ:

- код модели автомобиля;
- название модели автомобиля;
- **грузоподъемность***;
- табличная часть (список норм расхода ГСМ):
  - вид ГСМ (перечисление «ВидГСМ»: моторное масло, нитрол, солидол, бензин, керосин);
  - норма расхода на 10 литров;
  - норма расхода в %.

Вариант 19

Основной справочник — Профессии:

- код профессии;
- название профессии;
- минимальная ставка должностного оклада по данной профессии;
- **максимальная ставка должностного оклада по данной профессии***;
- табличная часть (список сотрудников, владеющих данной профессией на предприятии):
  - сотрудник (справочник «Сотрудники»);
  - образование (перечисление «ВидОбразования»: высшее, среднее и т. д.);
  - предыдущее место работы (справочник «Организации»);

Вариант 20

Основной справочник — Штатное расписание:

- кафедра (справочник «Кафедры»);
- **руководитель*** (справочник «Сотрудники»);
- табличная часть (список должностей сотрудников подразделения и число ставок):
  - должность (справочник «Должности»);
  - общее число ставок по данной должности;
  - из них занято;
  - из них свободно.

Вариант 21

Основной справочник — Виды расчетов подразделений предприятия:

- код подразделения;
- наименование подразделения;
- **руководитель*** (справочник «Сотрудники»);
- табличная часть (список видов расчетов, используемых при расчете заработной платы сотрудников данного подразделения):
  - вид расчета (справочник «ВидыРасчета»);
  - дата начала действия;
  - дата окончания действия.

Вариант 22

Основной справочник — Сотрудники:

- табельный номер сотрудника;
- ФИО сотрудника;
- пол сотрудника (перечисление «Пол»);
- дата приема на работу;
- оклад;
- **ставка***;
- табличная часть (список членов семьи сотрудника):
  - ФИО;
  - степень родства (перечисление «СтепеньРодства»);
  - дата рождения.

Вариант 23

Основной справочник — Пользователи:

- код;
- ФИО;
- **пароль***;
- табличная часть (список объектов, к которым пользователь имеет доступ):
  - объект (справ. «Фирмы», «Контрагенты», «Склады»);
  - тип объекта (перечисление «ТипОбъекта»: фирма, контрагент, склад).

Вариант 24

Основной справочник — Контрагенты:

- код контрагента;
- наименование контрагента;
- тип контрагента (перечисление «ТипКонтрагента»: юридическое лицо, физическое лицо);
- адрес контрагента;
- **основной расчетный счет*** (справочник «Расчетные счета»);
- табличная часть (список расчетных счетов контрагента):
  - расчетный счет (справочник «Расчетные счета»);
  - банк (справочник «Банки»);
  - адрес банка;
  - валюта (справочник «Валюты»).

ЗАКЛЮЧЕНИЕ

Данное учебное пособие направлено на формирование у студентов профессиональных компетенций, связанных с использованием теоретических знаний в области изучения принципов и средств автоматизации бизнес-процессов предприятии. Особое внимание уделяется изучению общей методологии создания предметно-ориентированных информационных систем, технологиям обработки информации, приобретению практических навыков решения задач по автоматизации оперативного учета.

В данном учебном пособии рассмотрены:

- основы построения информационных систем для решения задач автоматизации бизнес-процессов с применением платформы 1С:Предприятие 8.3 в режиме обычного приложения;
- приемы работы с технологической платформой 1С:Предприятие 8.3;
- применение объектов метаданных платформы 1С:Предприятие 8.3 для автоматизации бизнес-процессов оперативного учета;
- различные алгоритмы обработки информации в задачах автоматизации бизнес-процессов.

Получение студентами теоретических знаний и практических навыков работы с платформой 1С:Предприятие 8.3 необходимо для построения промышленных предметно-ориентированных информационных систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Обзор системы 1С:Предприятие 8 — Официальный сайт фирмы 1С. URL: http://v8.1c.ru/overview/ (дата обращения: 27.10.2020).
2. Меркулова, Т.А. Разработка управляемого приложения на платформе 1С: Предприятие 8 : учебное пособие / Т.А. Меркулова. — Ульяновск, 2012. URL: http://venec.ulstu.ru/lib/disk/2013/Merkulova_up.pdf (дата обращения: 27.10.2020).
3. Ажеронок, В.А. Профессиональная разработка в системе 1С: Предприятие 8 / В.А. Ажеронок, А.П. Габец, Д.И. Гончаров, Д.В. Козырев, Д.С.Кухлевский, А.В. Островерх, М.Г. Радченко, Е.Ю. Хрусталева; под редакцией М.Г. Радченко. — изд-е 2. — Москва: 1С-Паблишинг, 2012. — 2800 с.

Учебное издание

ФИЛИППОВ Алексей Александрович

РАЗРАБОТКА ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ.
Практический курс.
Построение информационных систем на платформе 1С:Предприятие 8.3 в режиме обычного приложения

Учебное пособие

Редактор М.В. Теленкова
Корректоры: К.Я. Выборнова, Е.В. Долгушева, Е.Д. Зиновьева

Оригинал-макет изготовлен в системе LaTeX2.

ЛР № 020640 от 22.10.97.
Подписано в печать 25.08.2021.
Формат 60×84/16. Усл. печ. л. 12,79.
Тираж 100 экз. Заказ 431. ЭИ № 1609.

Ульяновский государственный технический университет,
432027, Ульяновск, Сев. Венец, 32.

ИПК «Венец», УлГТУ, 432027, Ульяновск, Сев. Венец, 32.