

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 29.07.2022 18:05:32

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d9a44eb3e175b0e1a2bb1c1e

Министерство науки и высшего образования Российской Федерации

Южно-Российский государственный политехнический  
университет (НПИ) имени М.И. Платова

**С.Н. Широбокова**

# **БИЗНЕС-АНАЛИТИКА В ПРИЛОЖЕНИЯХ НА ПЛАТФОРМЕ «1С:ПРЕДПРИЯТИЕ»**

**Учебное пособие к практическим занятиям**

**Новочеркасск  
ЮРГПУ(НПИ)  
2021**

УДК 004.42 (075.8)  
ББК 32.973-018.1я73  
Ш64

**Широбокова С.Н.**

Ш64      Бизнес-аналитика в приложениях на платформе «1С:Предприятие»:  
учебное пособие к практическим занятиям / С.Н. Широбокова; Южно-  
Российский государственный политехнический университет (НПИ) имени  
М.И. Платова. – Новочеркасск: ЮРГПУ(НПИ), 2021.– 96с.

Учебное пособие содержит материалы к практическим занятиям по дисциплине  
"Бизнес-аналитика в приложениях на платформе «1С:Предприятие»".

Предназначено для студентов вузов, обучающихся по направлению подготовки  
«Прикладная информатика» (магистратура). Пособие может быть полезно студентам  
других компьютерных направлений подготовки, изучающим основы разработки  
систем бизнес-аналитики на платформе «1С:Предприятие».

УДК 004.42 (075.8)  
ББК 32.973-018.1я73

## Содержание

Введение	4
Тема 1. Алгоритмы машинного обучения. Проведение анализа «Поиск ассоциаций» в системе «1С:Предприятие»	5
Тема 2. Алгоритмы машинного обучения. Построение дерева решений в системе «1С:Предприятие».	17
Тема 3. Алгоритмы машинного обучения. Кластерный анализ в системе «1С:Предприятие»	29
Тема 4. Практическое использование языка запросов «1С:Предприятие»	41
Тема 5. Методы и инструментальные средства бизнес-аналитики. Разработка сложных аналитических отчетов с помощью СКД	53
Тема 6. Web-интерфейсы. Предоставление информации через web-интерфейсы	65
Тема 7. Использование HTTP-сервисов в 1С	77
Тема 8. WS-ссылки и получение данных с Web-сервисов сторонних поставщиков	89
Заключение	101
Библиографический список	102

## ВВЕДЕНИЕ

Учебное пособие предназначено для методического обеспечения практических занятий по дисциплине "Бизнес-аналитика в приложениях на платформе «1С:Предприятие»".

Практические занятия ориентированы на получение практических навыков конфигурирования и программирования задач анализа учетных данных и прогнозирования с использованием механизма анализа и прогнозирования данных платформы «1С:Предприятие», в частности поиска ассоциативных групп объектов и событий, построения дерева решений, проведения кластерного анализа. Практически задания по разработке различных запросов и построения отчетов формируют навыки использования языка запросов системы «1С:Предприятие» и системы компоновки данных для разработки сложных аналитических отчетов.

Практические навыки по интеграции приложений с помощью Web-интерфейсов, HTTP-сервисов и WS-ссылок сформируют компетенции по интеграции различных информационных систем и приложений бизнес-аналитики в разнородном IT-ландшафте предприятия.

Для проведения практических занятий обязательно требуется использование лабораторий с компьютерной техникой (рабочими местами обучающихся, оснащенными компьютерами с установленной учебной версией платформы «1С:Предприятие»).

## Тема 1. Алгоритмы машинного обучения. Проведение анализа «Поиск ассоциаций» в системе «1С:Предприятие»

За большими объемами данных часто скрываются различные закономерности, которые можно эффективно использовать в управленческой деятельности [1-2]. Для выявления таких закономерностей в платформе «1С:Предприятие» предусмотрен механизм анализа данных и прогнозирования. Механизм представляется совокупностью объектов встроенного языка 1С:Предприятие.

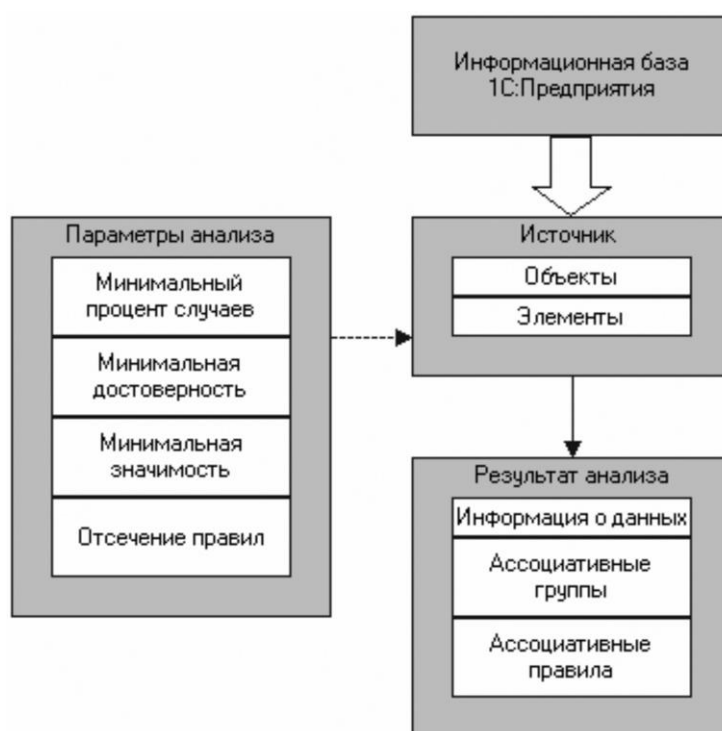
Одним из видов анализа, который он реализует, является анализ «Поиск ассоциаций». Как видно из самого названия, этот тип анализа направлен на выявление часто встречающихся вместе групп объектов или значений характеристик, а также поиска правил ассоциаций.

Клиентоориентированный подход в торговле требует все более эффективных методов анализа данных в области планирования продаж и закупок. Например, такой анализ может быть полезен для торговой организации: выявить группы покупаемых одновременно товаров (комплементарные товары) для того, чтобы учесть этот факт при раскладке товаров в торговом зале.

На рис. 1.1 приведена общая схема выполнения анализа «Поиск ассоциаций» [3].

Источником данных для данного типа анализа может выступать результат запроса, таблица значений или область ячеек табличного документа. Любой из этих источников состоит из определенных колонок. Каждая колонка для анализа имеет определенный тип. Она или НеИспользуемая, или Объект, или Элемент.

Колонки с настройкой «НеИспользуемая» игнорируются. Если колонка имеет тип «Объект», то для анализа данные этой колонки используются как объекты или события. Т.е. значения из колонки с типом Элемент будут отнесены к одной ассоциативной группе исходя из значений колонки с типом «Объект». Тип «Элемент» имеют колонки, данные которых используются для получения устойчивых значений и построения ассоциативные правила. Например, если в одном документе номенклатура считается одновременно покупаемой, то Регистратор – Объект, а



**Рис. 1.1. Общая схема выполнения анализа «Поиск ассоциаций»**

Как видно из схемы, кроме настройки типов колонок для проведения анализа надо задать ряд дополнительных параметров анализа. Рассмотрим их.

**Минимальный процент случаев** – данным параметром задают числовое значение (%), «порог срабатывания», в соответствии с которым ведется сравнение числа случаев, по которому можно отнести комбинацию к ассоциативной группе. Если у группы число случаев, когда они совместно встречаются в комбинации, меньше этого параметра – группа не включается в результат анализа. Это дает возможность не учитывать слабо распространенные ассоциации [3].

**Минимальная достоверность** – параметр задает минимальное значение процента случаев, когда правило соблюдается. Определяет требуемую устойчивость искомым ассоциаций.

**Минимальная Значимость** – группы, у которых данное значение меньше указанного, не попадут в результат анализа. Параметр позволяет выявить наиболее приоритетные из них.

**Тип отсечения правил** – этот параметр имеет тип системного перечисления ТипОтсеченияПравилАссоциации со значениями Избыточные и Покрытие. ТипОтсеченияПравилАссоциации.Избыточные – обозначает, что будут отсекаются избыточные правила. ТипОтсеченияПравилАссоциации.Покрытые – означает, что будут отсекаются правила, которые покрыты другими правилами.

Проведем анализ «Поиск ассоциаций» на практическом примере по поиску закономерностей в документах продажи товаров.

Реализуем минимально необходимые прикладные объекты.

Справочник «Номенклатура». Минимально необходимой структурой будут являться стандартные реквизиты «Код», «Наименование» и введем пользовательский реквизит «ЦенаПродажи»,

Таблица 1.1

Исходные данные о продажах товаров

Код	Наименование	ЦенаПродажи, руб.
000000001	Бумага для принтера А4	250,00
000000002	Файлы для документов А4 формата	272,00
000000003	Степлер	293,00
000000004	Маркер перманентный черный	344,00
000000005	Корректирующая жидкость	74,00
000000006	Скрепки канцелярские	97,00
000000007	Дырокол	696,00
000000008	Ножницы	114,00
000000009	Скоросшиватель	115,00
000000010	Клей-карандаш	30,00
000000011	Антистеплер	154,00
000000012	Ручка шариковая	35,00

Создадим документ «ПродажаТовара». В нем создадим табличную часть «Товары» с реквизитами табличной части:

«Номенклатура», тип СправочникСсылка.Номенклатура;

«Цена», тип Число, длина 15, точность 2;

«Количество», тип Число, длина 15, точность 3;

«Сумма», тип Число, длина 15, точность 2.

Создадим форму документа. Реализуем в ней обработчики изменения полей для расчета по строке поля Сумма при изменении полей Количество и Цена. А также при изменении поля Номенклатура из справочника должна подставляться Цена.

В модуле формы реализуем следующие обработчики:

&НаКлиенте

Процедура КоличествоПриИзменении(Элемент)

Стр = Элементы.Товары.ТекущиеДанные;

Стр.Сумма = Стр.Количество \* Стр.Цена;

КонецПроцедуры

&НаСервереБезКонтекста

Функция ПолучитьЦенуНоменклатуры(Номенклатура)

Возврат Номенклатура.ЦенаПродажи;

КонецФункции

&НаКлиенте

Процедура НоменклатураПриИзменении(Элемент)

Стр = Элементы.Товары.ТекущиеДанные;

Стр.Цена=ПолучитьЦенуНоменклатуры(Стр.Номенклатура);

КоличествоПриИзменении(Элемент);

КонецПроцедуры

Создадим регистр накопления «Продажи», имеющий следующую структуру:

Измерение:

«Номенклатура», тип СправочникСсылка.Номенклатура.

Ресурсы:

«Количество», тип Число, длина 15, точность 3;

«Сумма», тип Число, длина 15, точность 2.

Вид регистра – Обороты. Регистратором созданному регистру назначим документ «ПродажаТоваров».

Создадим регистр сведений «ЦеныПоставщков»: измерения: Контрагент, тип СправочникСсылка.Контрагенты, Номенклатура, тип СправочникСсылка.Номенклатура; Ресурс – Цена, тип Число, длина 15, точность 2. Режим записи – подчинение регистратору. Периодичность – По позиции регистратора. Регистратор – документ ПоступлениеТоваров, его структура аналогична документу «ПродажаТоваров».

Для документов реализуем их проведение по регистрам.

Введем данные из табл 1.1 в режиме исполнения в документы «ПродажаТовара». Регистратор – это объект анализа, Номенклатура – элемент. Введем следующие исходные данные, приведенные в табл 1.1.

Таблица 1.1

Исходные данные о продажах товаров

Регистратор	Номенклатура	Количество
1	2	3
Расходная накладная № 000000001 от 01.09.2021	Бумага для принтера А4	5
	Клей-карандаш	3
Расходная накладная № 000000002 от 04.09.2021	Корректирующая жидкость	4
Расходная накладная № 000000003 от 04.10.2021	Степлер	1
	Скрепки канцелярские	5
Расходная накладная № 000000004 от 05.10.2021	Ножницы	2
	Маркер перманентный черный	5
Расходная накладная № 000000005 от 05.10.2021	Ножницы	1
	Ручка шариковая	3
	Корректирующая жидкость	2
Расходная накладная № 000000006 от 25.10.2021	Бумага для принтера А4	5
	Файлы для документов А4 формата	2
Расходная накладная № 000000007 от 27.10.2021	Дырокол	4
Расходная накладная № 000000008 от 30.10.2021	Дырокол	1
Расходная накладная № 000000009 от 30.10.2021	Антистеплер	2
Расходная накладная № 000000010 от 01.11.2021г.	Бумага для принтера А4	6
	Файлы для документов А4 формата	3
	Скоросшиватели	5
Расходная накладная № 000000011 от 10.11.2021	Ручка шариковая	10
	Клей-карандаш	2

Создадим обработку «АнализДанных» и ее форму (рис. 1.2).

Создадим команду «ПоискАссоциаций», вынесем ее на форму. Создадим реквизит формы «Результат» типа Табличный документ. Вынесем его на форму.

В модуле формы реализуем следующий код обработчика команды:

```
&НаКлиенте
```

```
Процедура ПоискАссоциаций(Команда)
```

```
    Результат = АнализПоискАссоциаций();
```

```
КонецПроцедуры
```



&НаСервереБезКонтекста

Функция АнализПоискАссоциаций()

Анализ = Новый АнализДанных;

Анализ.ТипАнализа = Тип("АнализДанныхПоискАссоциаций");

Запрос = Новый Запрос;

Запрос.Текст = "

|ВЫБРАТЬ

|Продажи.Регистратор,

|Продажи.Номенклатура

|ИЗ

|РегистрНакопления.Продажи КАК Продажи";

Анализ.ИсточникДанных = Запрос.Выполнить();

Анализ.НастройкаКолонок.Номенклатура.ТипКолонки =

ТипКолонкиАнализаДанныхПоискАссоциаций.Элемент;

Анализ.Параметры.ТипОтсеченияПравил.Значение =

ТипОтсеченияПравилАссоциации.Избыточные;

РезультатАнализа = Анализ.Выполнить();

Построитель = Новый ПостроительОтчетаАнализаДанных();

Построитель.Макет = Неопределено;

Построитель.ТипАнализа = Тип("АнализДанныхПоискАссоциаций");

ТабДок = Новый ТабличныйДокумент;

Построитель.Вывести(РезультатАнализа, ТабДок);

Возврат ТабДок;

КонецФункции

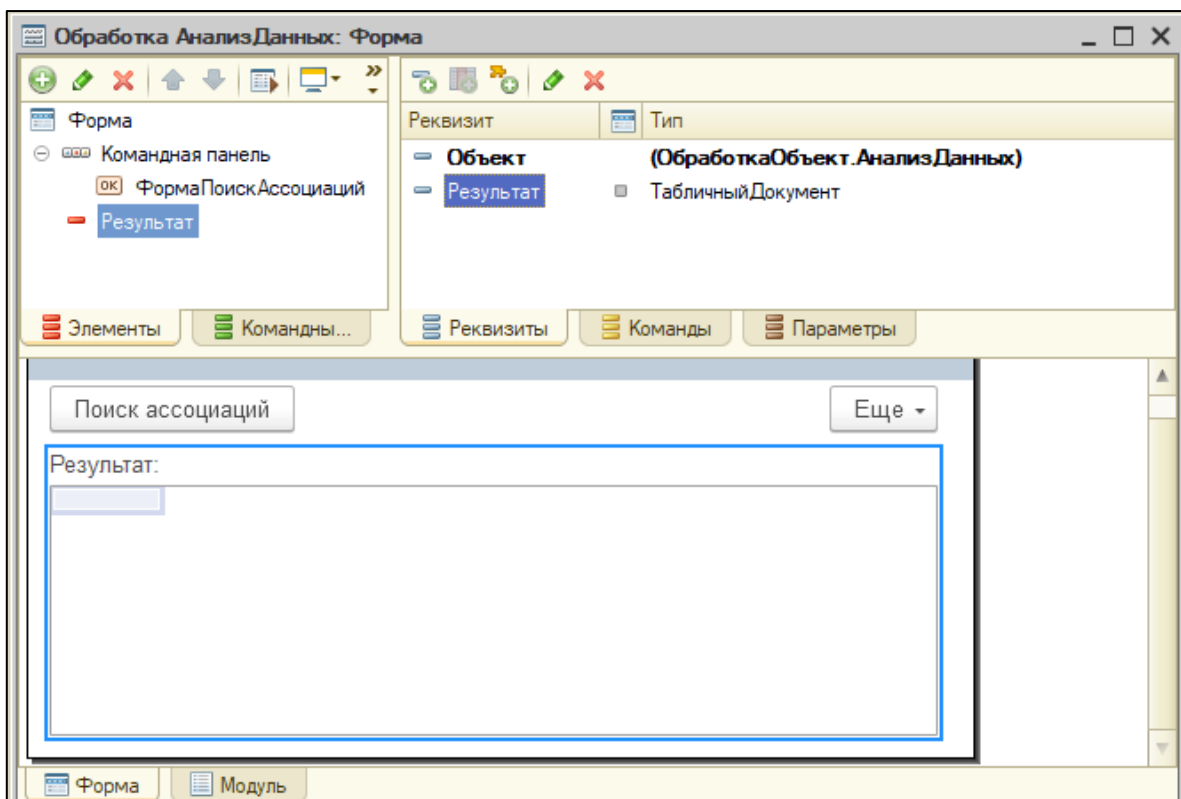


Рис. 1.2. Форма обработки «АнализДанных»

В режиме исполнения откроем обработку и нажмем команду «АнализДанных». Получим в табличном документе следующий результат анализа (рис. 1.3).

Результат:

### Поиск ассоциативных правил

**Параметры анализа**

Минимальный процент случаев:	10
Минимальная достоверность:	60
Минимальная значимость:	0
Отсечение правил:	Избыточные

**Колонки источника данных**

**Входные колонки**

Имя колонки	Тип данных
Номенклатура	Дискретный

**Информация о данных**

Количество элементов:	12
Количество объектов:	11
Среднее количество элементов в объекте:	1,82

**Результат анализа**

Найдено часто встречаемых групп:	1
Найдено ассоциативных правил:	2

**Часто встречаемые группы**

№	Количество случаев	Процент случаев
<b>Состав</b>		
1	2	18,18
Номенклатура = Бумага для принтера А4		
Номенклатура = Файлы для документов А4 формата		

**Рис. 1.3. Результаты анализа «Поиск ассоциаций»**

Результат анализа включает:

– информацию о данных (количество объектов, количество элементов, среднее количество элементов в объекте, количество найденных групп, количество найденных правил ассоциаций);

– найденные группы элементов – указывается состав группы, количество случаев, процент случаев, в которых эта группа встречается;

– найденные ассоциативные правила – указывается исходный состав элементов, следствие (состав элементов), процент случаев, достоверность, значимость правила.

Как видно из рис. 1.3 найдена следующая группа товаров:

- Табурет прямоугольный;
- Стол кухонный раскладной.

В исходной выборке 11 документов продажи товаров. Из них в 2 встречается вышеуказанная группа.

Результат анализа также содержит найденные ассоциативные правила (рис. 1.4).

<b>Ассоциативные правила</b>				
Исходный набор	Следствие	Процент случаев	Достоверность	Значимость
1		18,18	100,00	3,67
Номенклатура = Файлы для документов А4 формата	Номенклатура = Бумага для принтера А4			
2		18,18	66,67	3,67
Номенклатура = Бумага для принтера А4	Номенклатура = Файлы для документов А4 формата			

**Рис. 1.4. Найденные ассоциативные правила**

Рассмотрим более подробно второе из найденных ассоциативных правил. В двух случаях из одиннадцати в документе вместе с позицией Стол кухонный раскладной встречалась позиция Табурет прямоугольный. Исходя из этого, был рассчитан процент случаев:  $(2 / 11 * 100 = 18,18 \%)$ .

Достоверность была рассчитана следующим образом: обе номенклатурные позиции закупались в двух случаях, товарная позиция Стол кухонный раскладной встречалась в покупках 3 раза. Исходя из этого, достоверность равна:  $2 / 3 * 100 = 66,67 \%$ .

Значимость определяется как отношение достоверности правила к проценту нахождения Табурет прямоугольный в закупаемых товарах. Эта позиция встречается в двух документах из одиннадцати (18,18 %). Значимость равна:  $66,67 \% / 18,18 \% = 3,67$ .

Дальнейшее использование результата анализа заключается в том, что на его основе может быть создана модель прогноза, позволяющая прогнозировать поведение новых данных в соответствии с имеющейся моделью.

Например, можно если проанализировать, какие товары приобретаются вместе (в одной накладной) и сохранить этот результат анализа в базе данных, то в дальнейшем, при создании очередной накладной на основании сохраненного результата анализа можно построить модель прогноза, подать ей «на вход» новые данные, содержащиеся в этой накладной, и «на выходе» получить прогноз, — список товаров, которые контрагент Сидоров К.А. тоже, скорее всего, приобретет, если их ему предложить:

**Типы отсечения правил.** Одним из важных параметров анализа «Поиск ассоциаций» является параметр ТипОтсеченияПравил, который принимает значение из системного перечисления ТипОтсеченияПравилАссоциации.

Перед тем как перейти к рассмотрению особенностей вариантов отсечения, рассмотрим несколько общих моментов, применимых к правилам ассоциации [1]

Любое правило состоит из **предпосылки** и **следствия**. Например:

- Предпосылка: **Если** купили Товар № 1.
- Следствие: **То** купят Товар № 2.

При этом не нужно забывать, что следствие наступает с определенной достоверностью. При отсечении правил вероятностные характеристики могут рассматриваться, а могут и игнорироваться (важно только содержание правила).

#### *Отсечение покрытых правил*

Рассмотрим вариант отсечения Покрытые.

Правило может быть покрыто как по предпосылке, так и по следствию.

Например:

- Правило № 1: **Если** купили товар № 1 и № 3, **То** купят товар № 2.
- Правило № 2: **Если** купили товар № 1, **То** купят товар № 2.

В этом случае правило № 1 будет считаться покрытым, т. к. предпосылка первого правила получается «избыточной» по отношению к предпосылке второго правила.

Пример покрытия по следствию:

- Правило № 1: **Если** купили товар № 1, **То** купят товар № 2, № 3.
- Правило № 2: **Если** купили товар № 1, **То** купят товар № 3.

Правило № 2 будет покрыто по следствию, так как следствие правила № 1 более полное.

#### *Отсечение избыточных правил*

Покрытие не учитывает вероятностных характеристик правил, они учитываются в случае, если используется вариант отсечения Избыточные.

Правило будет считаться избыточным по предпосылке, если оно покрыто по предпосылке и его достоверность равна достоверности покрывающего правила. Например:

- Правило № 1: **Если** купили товар № 1 и № 3, **То** купят товар № 2 с достоверностью 75 %.
- Правило № 2 **Если** купили товар № 1, **То** купят товар № 2 с достоверностью 75 %.

Правило № 1 избыточно по отношению к правилу № 2 (оно содержит дополнительное условие, не вносящее «возмущения» в вероятностные характеристики правила).

Правило № 1 будет считаться избыточным по следствию, если количество случаев данного правила равно количеству случаев покрывающего правила.

- **Правило № 1: Если** купили товар № 1, **То** купят товар № 2, № 3 в трех случаях.
- **Правило № 2: Если** купили товар № 1, **То** купят товар № 3 в трех случаях.

Правило № 2 будет считаться избыточным по отношению к правилу № 1, так как оно содержит более простое следствие с теми же вероятностными характеристиками.

Первоначально метод был разработан для поиска типичных сочетаний товаров в покупках, поэтому иногда его еще называют анализом покупательской корзины. Применительно к этому сценарию в качестве ассоциируемых элементов, как правило, выступают товарные группы или отдельные товары. А группирующим объектом, объединяющим элементы выборки, может быть любой объект информационной системы, идентифицирующий сделку, например заказ покупателя, акт об оказании услуг или кассовый чек [1].

Информация о закономерностях в товарных предпочтениях покупателей позволяет повысить эффективность управления отношениями с клиентами (в части рекламных кампаний и маркетинговых акций), ценообразования (формирование комплексных предложений и системы скидок), управления запасами и мерчендайзинга (распределение товаров в торговых залах). Прогноз полезен также при планировании закупок для того, чтобы товары из одной ассоциативной группы всегда были в наличии на складе. Другой пример использования этого метода — определение предпочитаемых клиентами комбинаций рекламных каналов для исключения их дублирования при проведении целевых рекламных кампаний. Это позволяет существенно снизить издержки на подобные мероприятия.

Этот тип анализа может работать с иерархическими данными, что позволяет, например, находить правила не только для конкретных товаров, но и для их групп. Важной особенностью этого типа анализа является возможность работать как с объектным источником данных, в котором каждая колонка содержит некоторую характеристику объекта, так и с событийным источником, где характеристики объекта располагаются в одной колонке.

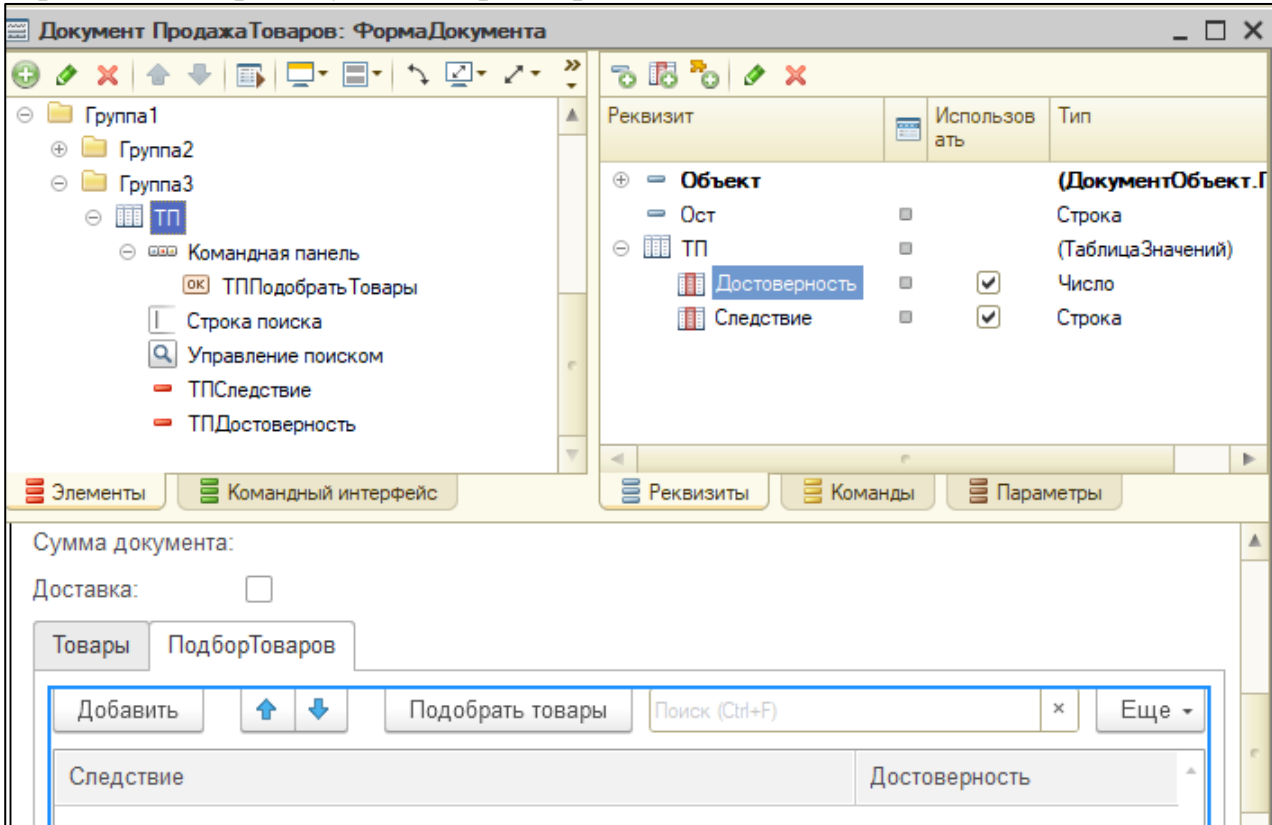
Реализуем в нашей конфигурации простейшую подсказку в подборе товара на основе анализа «Поиск ассоциаций».

Для этого добавим на форму документа «ПродажаТоваров» реквизит формы ТП типа ТаблицаЗначений. В этот реквизит добавим две новые колонки: «Достоверность», тип Число, точность 3, «Следствие», тип Строка.

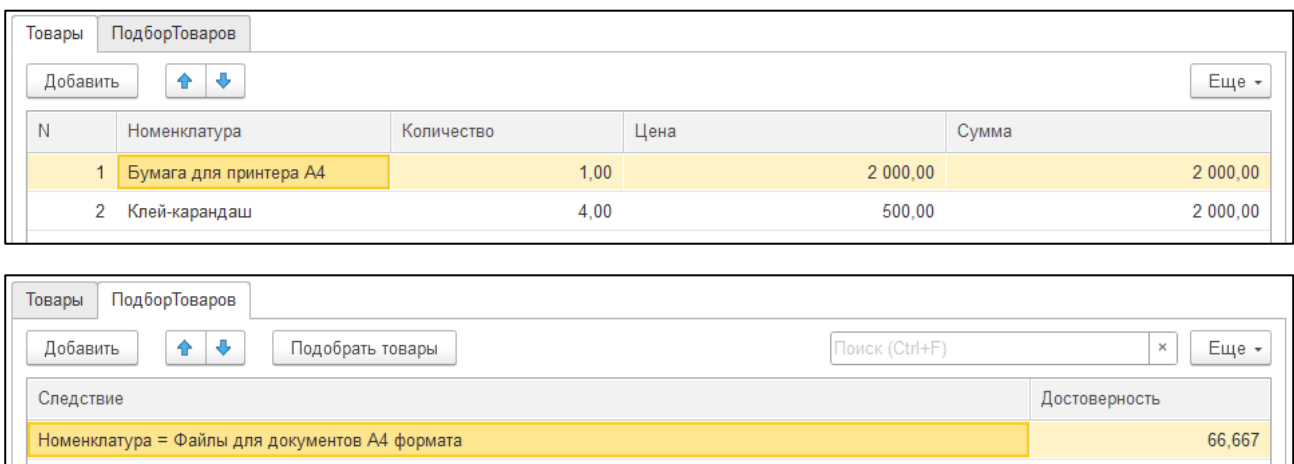
Создадим на форме в группе «Группа1» типа «Страницы» еще одну группу типа «Страница» с заголовком «ПодборТоваров» и на нее вытащим реквизит формы ТП, который будет отражать результаты прогноза (рис. 1.5-1.6). При перетаскивании в дерево элементов система задаст вопрос создавать ил колонки, ответим утвердительно.

Добавим команду формы «ПодобратьТовары», вытащим ее на форму на страницу «ПодборТоваров».

Реализуем в модуле формы код обработчика команды «ПодобратьТовары». А также реализует возможность автоматического заполнения таблицы значений с результатами прогноза при смете страницы и переходе на страницу «ПодборТоваров».



**Рис. 1.5. Новая группа и новый реквизит формы для функциональности подбора товаров на основе прогноза**



**Рис. 1.6. Подбор товаров по результатам прогноза**

```

&НаКлиенте
Процедура ПодобратьТовары(Команда)
    ПодобратьТоварыНаСервере();
КонецПроцедуры
    
```

&НаСервере

Процедура ПодобратьТоварыНаСервере()

Анализ = Новый АнализДанных;

Анализ.ТипАнализа = Тип("АнализДанныхПоискАссоциаций");

Запрос = Новый Запрос;

Запрос.Текст = "

|ВЫБРАТЬ

|Продажи.Регистратор,

|Продажи.Номенклатура

|ИЗ

|РегистрНакопления.Продажи КАК Продажи";

Анализ.ИсточникДанных = Запрос.Выполнить();

Анализ.НастройкаКолонок.Номенклатура.ТипКолонки =

ТипКолонкиАнализаДанныхПоискАссоциаций.Элемент;

Анализ.Параметры.ТипОтсеченияПравил.Значение =

ТипОтсеченияПравилАссоциации.Избыточные;

РезультатАнализа = Анализ.Выполнить();

// Исходя из полученного анализа создать прогноз.

МодельПрогноза = РезультатАнализа.СоздатьМодельПрогноза();

МодельПрогноза.ИсточникДанных = Объект.Товары.Выгрузить();

ТП.Очистить();

ТП.Загрузить(МодельПрогноза.Выполнить());

КонецПроцедуры

&НаКлиенте

Процедура Группа1ПриСменеСтраницы(Элемент, ТекущаяСтраница)

Если ТекущаяСтраница.Заголовок="ПодборТоваров" Тогда

ПодобратьТоварыНаСервере();

КонецЕсли;

КонецПроцедуры

Протестируем созданную функциональность в режиме исполнения. Создадим новый документ «ПродажаТоваров» и добавим в табличную часть «Товары» два товара, например, следующие:

«Бумага для принтера А4»;

«Клей-карандаш».

По результатам анализа ассоциаций было найдено ассоциативное правило, согласно которому если при текущем наборе покупаемых товаров предложить товар «Файлы для документов А4 формата», то клиент согласится его купить (с достоверностью 66,7).

Как видно из рис. 1.4 для текущей исходной выборке данных по осуществленным продажам было найдено соответствующее ассоциативное правило, связывающее в ассоциативную группу товары «Бумага для принтера А4» и «Файлы для документов А4 формата». На его основе и был получен результат прогноза, поскольку в уже закупаемых товарах, добавленных в

табличную часть «Товары» присутствует один из товаров, входящих в ассоциативную группу.

Выполнение подбора производится как при переходе на страницу «ПодборТоваров» (событие «ПриСменеСтраницы»), так и, если пользователь нажмет соответствующую кнопку «ПодобратьТовары».

Протестируем, не будет ли в таблице подбора дублироваться товары, которые уже выбраны в табличной части «Товары». Добавим товар «Файлы для документов А4 формата» в табличную часть «Товары», т.е. допустим клиент сам выбрал и этот товар тоже, а значит его не надо еще раз «Подсказывать» в подборе товаров.

Перейдем на страницу «ПодборТоваров» и убедимся, что сформированная ранее подсказка исчезнет из таблицы рекомендуемых сопутствующих товаров, поскольку товар и так уже присутствует в выбранных товарах.

Таким образом, в задаче прогнозирования можно выделить две основные компоненты:

- обучение модели на какой-либо выборке;
- использование обученной модели для работы с фактическими данными для прогноза.

Используя результаты анализа «Поиск ассоциаций» автоматически строится модель прогноза ассоциаций, из которой следует, что клиент, по аналогии с другими случаями, может приобрести предложенный ему товар с определенной степенью вероятности вместе с теми товарами, что он только что купил.

В качестве еще одного из сценариев использования можно привести сценарий «Управление цепочками поставок» [1].

Цель – оптимизация выбора поставщиков по товарной группе.

Выбор доминирующих поставщиков «первого ряда» для ключевых товарных групп чрезвычайно важен для стабилизации системы логистики в частности и общей системы управления цепочками поставок в целом, уменьшения средней продолжительности цепочек поставок. Вместе с тем более тесная интеграция с основными поставщиками позволяет, как правило, существенно снизить себестоимость товаров. В связи с этим представляет интерес анализ устойчивых комбинаций поставщиков в различных товарных группах в сравнении с аналитикой по ассоциированным в рамках групп поставщикам. Это дает возможность выявить «пересечения» поставщиков в различных товарных группах и оптимизировать взаимоотношения с ними.

Алгоритм — «Поиск ассоциаций».

Прогнозные атрибуты — устойчивые комбинации поставщиков

Основные факторы — товарные группы.

Расшифровка — аналитика по поставщикам (объем закупок, выручка, условия поставки, оплаты, пессимистичный, оптимистичный, средний сроки выполнения заказа).

Пример закономерности. Устойчивая ассоциация крупного и непредсказуемого поставщика А и предсказуемого среднего поставщика Б в



большом количестве товарных групп. Возможно при формировании заказов по конкурентным товарным группам в качестве основного позиционировать среднего поставщика, если объем заказа крупному не превышает некоторого (дающего существенный выигрыш на масштабах) порога.

## Тема 2. Алгоритмы машинного обучения. Построение дерева решений в системе «1С:Предприятие»

Метод дерева решений — это прекрасный способ выбрать стратегию последовательных действий в условиях риска. Именно риск здесь выступает ключевым словом, поскольку при опасности принять рациональное решение очень сложно, а продуманный план помогает проанализировать сложившуюся ситуацию [3].

С помощью этого метода решается целый ряд задач, когда имеются два или более последовательных множества решений, причем, последующие решения основываются на результатах предыдущих состояний среды, т.е. появляется цепочка решений, вытекающих одно из другого. Подобные задачи проще решать с использованием дерева решений, которое представляет собой графическое изображение последовательности решений и состояний среды с указанием соответствующих вероятностей и выигрышей для всевозможных комбинаций.

Дерево принятия решений подобно настоящему: у него есть ствол, ветви и листья. «Ствол» — основа всего — это главный вопрос, на который нужно ответить. Ветви — это стрелочки с несколькими вариантами ответов. А листья — это ситуации, к которым приведет нас выбранный ответ.

Пример: фрагмент дерева решений для банковской сферы – принятие решения о кредитовании заемщика банка на основе фактов, представлен на рис. 2.1.

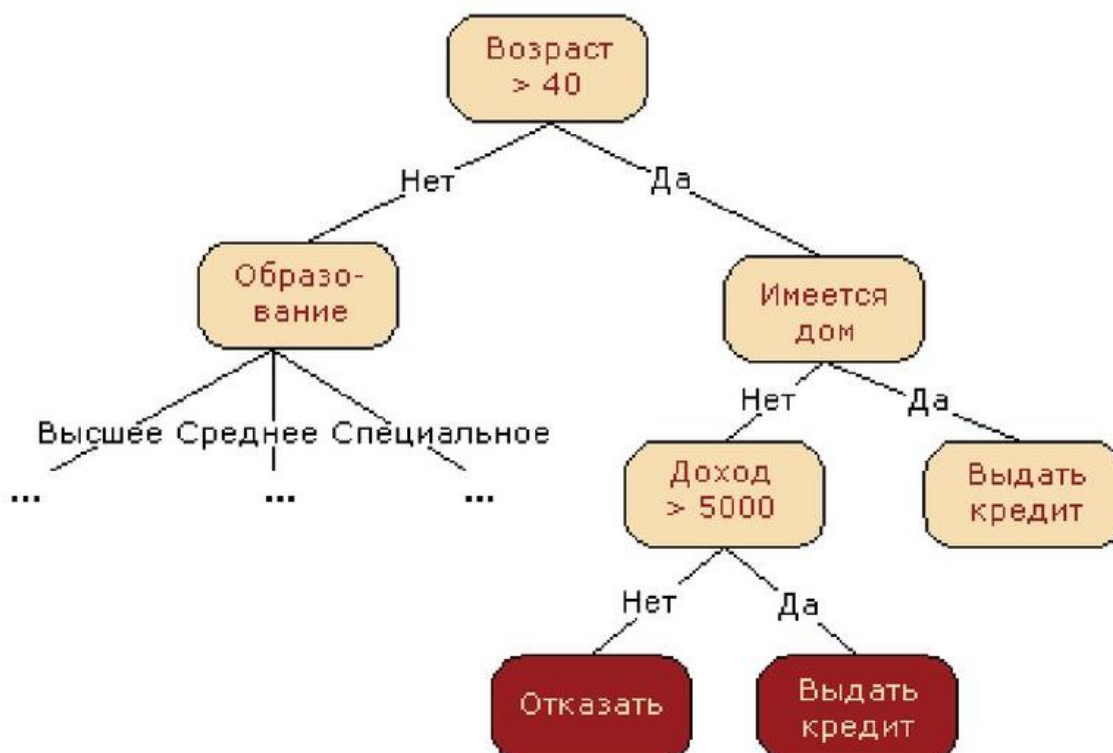


Рис. 2.1. Дерево решений для сегментации заемщиков банка

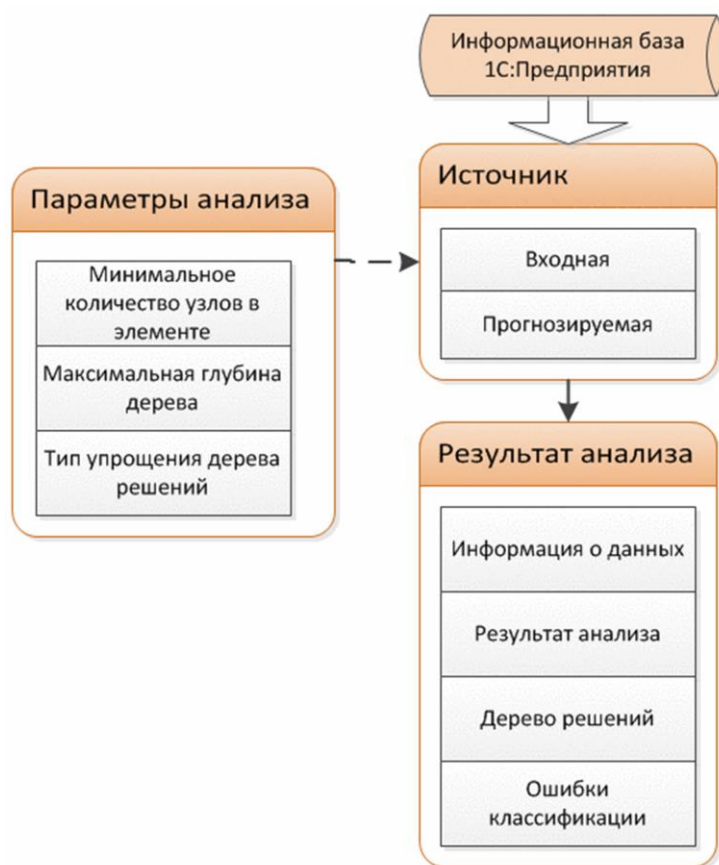
Средствами механизма анализа данных и прогнозирования в системе «1С:Предприятие» тип анализа «Дерево решений» для анализа больших данных, накапливаемых в корпоративных информационных системах также можно реализовать.

Тип анализа дерево решений в системе «1С:Предприятие» позволяет построить иерархическую структуру классифицирующих правил, представленную в виде дерева. Для построения дерева решений необходимо выбрать целевой атрибут, по которому будет строиться классификатор и ряд входных атрибутов, которые будут использоваться для создания правил. Общая схема процесса проведения анализа «Дерево решений» представлена на рис 2.2 [1].

Результат работы анализа представляется в виде дерева, каждый узел которого содержит некоторое условие. Для принятия решения, к какому классу следует отнести некий новый объект, необходимо, отвечая на вопросы в узлах, пройти цепочку от корня до листа дерева, переходя к дочерним узлам в случае утвердительного ответа и к соседнему узлу в случае отрицательного. Набор параметров анализа позволяет регулировать точность полученного дерева.

С помощью данного типа анализа можно получить причинно-следственную иерархию условий, приводящую к определенному решению, например, выигрышу в сделке. Еще одним пример использования можно привести следующий: получить дерево условий, по которому (с определенной долей вероятности) можно понять причину расторжения договоров с клиентами компании, определения условий, влияющих на вариант заключаемого договора. Можно проводить «профилирование» менеджеров компании по различным видам ее клиентов и т. п.

Для построения дерева решений необходимо выбрать целевой атрибут, по которому будет строиться классификатор и ряд входных атрибутов, которые будут использоваться для создания правил. Целевой атрибут может содержать, например, информацию о



**Рис. 2.2. Общая схема выполнения анализа «Дерево решений»**

том, перешел ли клиент к другому поставщику услуг, удачна ли была сделка, качественно ли была выполнена работа и т. д. Входными атрибутами, для примера, могут выступать возраст сотрудника, стаж его работы, материальное состояние клиента, количество сотрудников в компании и т. п.

Результат работы анализа представляется в виде дерева, каждый узел которого содержит некоторое условие. Для принятия решения, к какому классу следует отнести некий новый объект, необходимо, отвечая на вопросы в узлах, пройти цепочку от корня до листа дерева, переходя к дочерним узлам в случае утвердительного ответа и к соседнему узлу в случае отрицательного [3].

Колонки источника для данного типа анализа можно разделить на 3 типа:

- «Неиспользуемая»;
- «Входная»;
- «Прогнозируемая».

Как уже упоминалось ранее регулировать точность получаемого дерева позволяет набор используемых параметров анализа:

- «МинимальноеКоличествоСлучаев» – это параметр задает минимальное количество элементов в узле;
- «МаксимальнаяГлубина» – параметр задает максимальную глубину дерева;
- «ТипУпрощения» – с помощью параметра задается тип упрощения дерева решений.

В результате проведенного анализа можно получить дерево решений и ошибки классификации.

Реализуем в нашей конфигурации пример использования этого типа анализа на данных о контрагентах.

Вначале создадим дополнительные вспомогательные объекты.

Создадим перечисления:

«ВидыДоговоров» со значениями «Дилер», «Дистрибьютор», «ПостоянныйПартнер»;

«СостоянияВзаимоотношений» со значениями «НесоблюдениеДоговора», «ПрекращениеКонтрагентом», «НеПрекращены»;

«Длительность» со значениями «МеньшеГода», «ОтГодаДоТрех», «ОтТрехДоДесятиЛет», «СвышеДесятиЛет».

Далее дополним справочник «Контрагенты» новыми реквизитами.

«КоличествоРозничныхТочек», тип Число, длина 4, точность 0;

«КоличествоАвтомобилей», тип Число, длина 4, точность 0;

«ВремяРаботыОрганизации», тип ПеречислениеСсылка.Длительность»;

«ВремяЗаключенияДоговора», тип ПеречислениеСсылка.Длительность»;

«ВидДоговора», тип ПеречислениеСсылка.ВидыДоговоров;

«ПрекращениеОтношений», тип ПеречислениеСсылка.Состояния Взаимоотношений.

Введем в режиме исполнения в справочник «Контрагенты» данные о нескольких новых контрагентах, представленные в табл.2.1

Таблица 2.1

## Выборка данных о контрагентах

Контрагент	Количество розничных точек	Количество автомобилей	Время работы организации	Время заключения договора	Вид договора	Состояние взаимоотношений
ЗАО МДК	1	0	МеньшеГода	МеньшеГода	Дилер	Несоблюдение Договора
ООО «Лидер»	15	4	ОтТрех-ДоДесяти Лет	МеньшеГода	Дистрибьютер	Прекращение-Контрагентом
ООО «Дуб-Мебель»	1	10	ОтТрех-ДоДесяти Лет	ОтГода-ДоТрех	Дистрибьютер	Прекращение-Контрагентом
ИП Огарьков С.К.	1	1	ОтГода-ДоТрех	МеньшеГода	Дилер	Прекращение-Контрагентом
Магазин №23	1	1	Свыше-Десяти-Лет	ОтТрех-ДоДесяти Лет	Постоянный-Партнер	НеПрекращены
ООО «Быстрая-Пицца»	3	2	МеньшеГода	МеньшеГода	Постоянный-Партнер	НеПрекращены
ООО «Василиса»	7	3	ОтТрех-ДоДесяти Лет	ОтГода-ДоТрех	Постоянный-Партнер	Прекращение-Контрагентом
ООО «Силай»	2	2	Свыше-Десяти-Лет	ОтТрех-ДоДесяти Лет	Постоянный-Партнер	НеПрекращены
ИП Добров К.Л.	0	1	МеньшеГода	МеньшеГода	Дилер	НеПрекращены

Для проведения анализа создадим в обработке «АнализДанных» новую команду «ПостроитьДеревоРешений». Вытащим ее на командную панель формы.

В модуле формы реализуем следующий код обработчика команды, выбрав при создании вариант «Создать на клиенте и процедуру на сервере»:

&НаКлиенте

Процедура ПостроитьДеревоРешений(Команда)  
ПостроитьДеревоРешенийНаСервере();

```

    КонецПроцедуры
&НаСервере
Функция ПостроитьДеревоРешенийНаСервере()
    Анализ = Новый АнализДанных;
    Анализ.ТипАнализа = Тип("АнализДанныхДеревоРешений");
    Запрос = Новый Запрос;
    Запрос.Текст = "
|ВЫБРАТЬ
| Контрагенты.Ссылка,
| Контрагенты.КоличествоРозничныхТочек,
| Контрагенты.КоличествоАвтомобилей,
| Контрагенты.ВремяРаботыОрганизации,
| Контрагенты.ВремяЗаключенияДоговора,
| Контрагенты.ВидДоговора,
| Контрагенты.ПрекращениеОтношений
|ИЗ
| Справочник.Контрагенты КАК Контрагенты";
    Анализ.ИсточникДанных = Запрос.Выполнить();
    Анализ.Параметры.ТипУпрощения.Значение =
ТипУпрощенияДереваРешений.НеУпрощать;
    РезультатАнализа = Анализ.Выполнить();
    Построитель = Новый ПостроительОтчетаАнализаДанных();
    Построитель.Макет = Неопределено;
    Построитель.ТипАнализа = Тип("АнализДанныхДеревоРешений");
    ТабДок = Новый ТабличныйДокумент;
    Построитель.Вывести(РезультатАнализа, ТабДок);
    Возврат ТабДок;
КонецФункции

```

В серверной процедуре с помощью конструктора Новый создаем объект встроенного языка типа АнализДанных, присваиваем переменной Анализ. В свойстве «ТипАнализа» указываем значение нужного нам сейчас типа анализа «АнализДанныхДеревоРешений».

Далее с помощью конструктора запроса формируем запрос к справочнику «Контрагенты», результат выполнения которого передаем в качестве источника данных в объект Анализ.

Перед выполнением анализа можно настроить параметры:

```

Анализ.Параметры.МинимальноеКоличествоСлучаев;
Анализ.Параметры.МаксимальнаяГлубина;
Анализ.Параметры.ТипУпрощения.

```

Как видно в вышеприведенном коде был задан один параметр Анализ.Параметры.ТипУпрощения, значения остальных – использовались по умолчанию.

Далее с помощью метода Выполнить() реализуется поиск прогнозируемых решений. Результат выполнения в виде таблицы значений, в

которой для каждого объекта указано прогнозируемое решение, записываем в переменную РезультатАнализа.

Для визуализации результатов анализа используется объект встроенного языка ПостоительОтчетаАнализаДанных. Указываем тип анализа и выводим результаты в табличный документ.

В режиме исполнения проведем анализ «Дерево решений» над имеющимися исходными данными (табл. 2.1). Для этого нажмем кнопку «ПостроитьДеревоРешений» в форме обработке и получим результаты, представленные в табл. 2.2.

Как видно из таблицы, результаты, формируемые системой, состоят из нескольких секций. В секции «Параметры анализа» приведены параметры, определяющие точность получаемого дерева. В том числе показаны параметры, которые не были заданы в коде и для них устанавливаются некоторые значения по умолчанию. В нашем примере мы явно задавали в коде только один параметр «Тип упрощения дерева решений», равным значению «Не упрощать». Параметры «Минимальное количество элементов в узле» и «Максимальная глубина дерева» показаны использованные значения по умолчанию.

Секция «Колонки источника данных» разбиты на две подсекции, отражающие перечень входных и прогнозируемых параметров.

Секция «Информация о данных» содержит статистику о количестве объектов и количестве классов.

Секция «Результат анализа» содержит сводные данные о результате анализа: глубина дерева решений, количестве внутренних узлов, количестве листьев, проценте ошибок.

Таблица 2.2

### Результаты построения дерева решений (тип упрощения = «НеУпрощать»)

	<b>Параметры анализа</b>	
	Минимальное количество элементов в узле:	0
	Максимальная глубина дерева:	1 000
	Тип упрощения дерева решений:	Не упрощать
	<b>Колонки источника данных</b>	
	<b>Входные колонки</b>	
	<b>Имя колонки</b>	<b>Тип данных</b>
	Ссылка	Дискретный
	КоличествоРозничныхТочек	Непрерывный
	КоличествоАвтомобилей	Непрерывный
	ВремяРаботыОрганизации	Дискретный
	ВремяЗаключенияДоговора	Дискретный
	ВидДоговора	Дискретный

Продолжение табл. 2.2.

**Прогнозируемые колонки**

Имя колонки	Тип данных
ПрекращениеОтношений	Дискретный

**Информация о данных**

Количество объектов: 9  
 Количество классов: 3

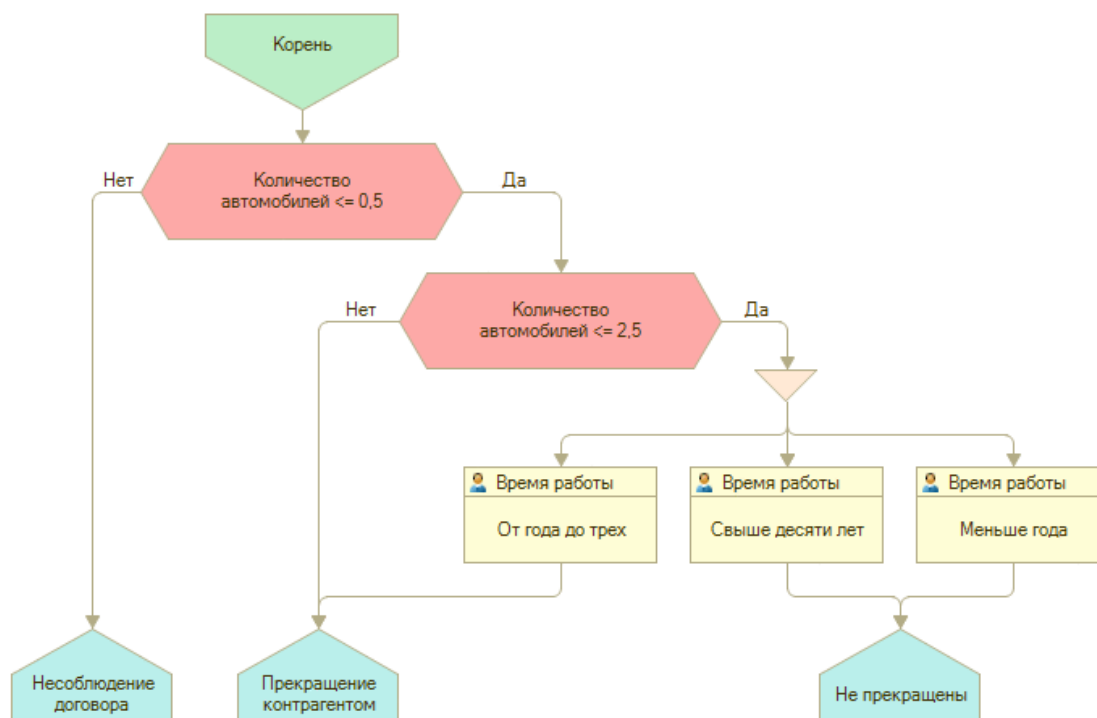
**Результат анализа**

Глубина дерева решений: 4  
 Количество внутренних узлов: 3  
 Количество листьев: 5  
 Ошибка, %: 0

**Дерево решений**

Условие	Решение	Вероятность	Количество
Корень	ПрекращениеОтношений = Прекращение контрагентом	44,44	9
КоличествоАвтомобилей <= 0,5	ПрекращениеОтношений = Несоблюдение договора	100,00	1
КоличествоАвтомобилей > 0,5	ПрекращениеОтношений = Прекращение контрагентом	50,00	8
КоличествоАвтомобилей <= 2,5	ПрекращениеОтношений = Не прекращены	80,00	5
ВремяРаботыОрганизации = Меньше года	ПрекращениеОтношений = Не прекращены	100,00	2
ВремяРаботыОрганизации = От года до трех	ПрекращениеОтношений = Прекращение контрагентом	100,00	1
ВремяРаботыОрганизации = Свыше десяти лет	ПрекращениеОтношений = Не прекращены	100,00	2
КоличествоАвтомобилей > 2,5	ПрекращениеОтношений = Прекращение контрагентом	100,00	3

Если представить дерево решений в графическом виде, то оно будет иметь следующий вид (рис. 2.3.).



**Рис. 2.3. Графическое представление дерева решения**



В табл. 2.3 приведены ошибки классификации.

Таблица 2.3

### Ошибки классификации

Факт\Классифицировано	ПрекращениеОтношений = Несоблюдение договора	ПрекращениеОтношений = Прекращение контрагентом	ПрекращениеОтношений = Не прекращены	Итого	% Ошибки
ПрекращениеОтношений = Несобл	1			1	
ПрекращениеОтношений = Прекра		4		4	
ПрекращениеОтношений = Не прек			4	4	
<b>Итого</b>	1	4	4	9	

Они показывают, в каких случаях полученные данные расходятся с действительностью (исходной выборкой данных). Как видно по данным табл. 2.3, в полученной классификации нет ошибок. Это означает, что в фактической выборке данные совпадают с данными классификации.

Точность получаемого дерева зависит от набора используемых параметров анализа. Реализуем возможность задания пользователем параметра ТипУпрощения.

Добавим в реквизиты обработки новый реквизит «ТипУпрощения». Установим ему тип системного перечисления ТипУпрощенияДереваРешений (рис. 2.4).

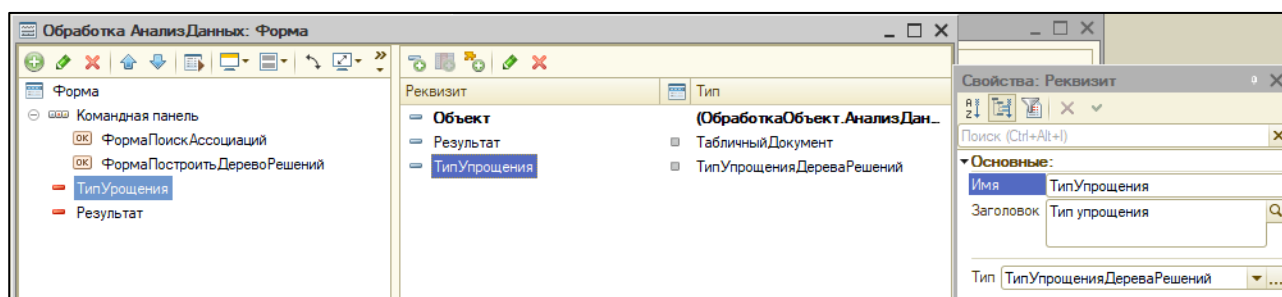
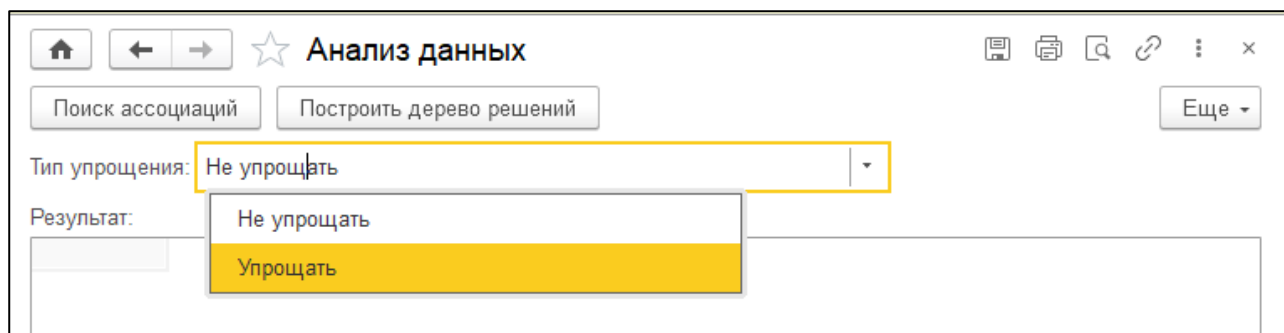


Рис. 2.4. Реквизит обработки «ТипУпрощения»

Вынесем этот новый реквизит на форму и разместим его над поле Результат (рис. 2.5).



**Рис. 2.5. Задание параметра «ТипУпрощения» в режиме исполнения**

Для подстановки выбранного пользователем значения параметра откорректируем немного код в модуле формы.

Вместо строки:

Анализ.Параметры.ТипУпрощения.Значение =  
ТипУпрощенияДереваРешений.НеУпрощать;

вставим следующую строку кода:

Анализ.Параметры.ТипУпрощения.Значение = ТипУпрощения;

Протестируем в режиме исполнения. Выберем на форме обработки перед формированием дерева значение типа упрощения равное «Упрощать» и затем, нажав кнопку «ПостроитьДеревоРешений» на командной панели, сформируем дерево решений с учетом этого параметра.

Полученный результат представлен в табл. 2.4.

Таблица 2.4

### Результаты построения дерева решений (тип упрощения = «Упрощать»)

#### Дерево решений

Условие	Решение	Вероятность	Количество
Корень	ПрекращениеОтношений = Прекращение контрагентом	44,44	9
КоличествоАвтомобилей <= 0,5	ПрекращениеОтношений = Несоблюдение договора	100,00	1
КоличествоАвтомобилей > 0,5	ПрекращениеОтношений = Прекращение контрагентом	50,00	8
КоличествоАвтомобилей <= 2,5	ПрекращениеОтношений = Не прекращены	80,00	5
КоличествоАвтомобилей > 2,5	ПрекращениеОтношений = Прекращение контрагентом	100,00	3

В табл. 2.5 приведены ошибки классификации, полученные при установленном типе упрощения в значение «Упрощать». Они показывают, в каких случаях полученные данные расходятся с действительностью (исходной выборкой данных).

Как видно из табл. 2.5, в связи с упрощением в классификации появились ошибки. Существует один случай, когда в реальной выборке данных было значение «ПрекращениеКонтрагентом», а по полученной классификации должно было быть значение «НеПрекращены»

Таблица 2.5

Ошибки классификации при анализе с параметром  
Тип Упрощения равным значению «Упрощать»

**Ошибки классификации**

Факт\Классифицировано	Прекращение Отношений = Несоблюдение договора	Прекращение Отношений = Прекращение контрагентом	Прекращение Отношений = Не прекращены	Итого	% Ошибки
Прекращение Отношений = Несобл	1			1	
Прекращение Отношений = Прекра		3	1	4	25,00
Прекращение Отношений = Не прек			4	4	
Итого	1	3	5	9	11,11

Упрощение заключается в том, что по формулам, которые рассмотрены ниже, узлы дерева преобразовываются в листья, т.е. отсекаются лишние ветвления.

При принятии решения о том, будет ли произведено преобразование узла в лист, учитываются следующие показатели:

- «Ошибок» – количество ошибок в узле;
- «ОшибокДочерних» – количество ошибок в дочерних узлах;
- «Листов» – количество листов в узле;
- «Случаев» – количество случаев.

Производится расчет по следующей формуле:

$$СКОшибки = \sqrt{\frac{(ОшибокДочерних + \frac{Листов}{2}) * (Случаев - (ОшибокДочерних + \frac{Листов}{2}))}{Случаев}}$$

В случае выполнения условия:

$$Ошибок + 0,5 < ОшибокДочерних + \frac{Листов}{2} + СКОшибки$$

принимается решение о превращении узла в лист.

В нашем примере для узлов Время работы организации условие выполняется ( $0,5 < 1$ ).

Таким образом, данный алгоритм получил наибольшее распространение при выявлении причинно-следственных связей в данных и описании поведенческих моделей. Типичная зона применения деревьев решений – оценка различных рисков, например, еще одной задачей, где может быть использован этот вид анализа, является оценка риска закрытия заказа клиентом или его перехода к конкуренту, несвоевременной поставки товара поставщиком или просрочки оплаты товарного кредита. В качестве типичных входных факторов модели выступают сумма и состав заказа, текущее сальдо взаиморасчетов, кредитный лимит, процент предоплаты, условия поставки и иные параметры, характеризующие объект прогноза. Адекватная оценка рисков обеспечивает принятие информированных решений для оптимизации отношения доходность/риск в деятельности компании, а также полезна для увеличения реалистичности различных бюджетов. Таким образом, можно решить задачу оптимизации работы отдела продаж. Для ее решения в качестве прогнозируемой величины выберем показатель эффективности менеджеров по продажам, например, удельную доходность на клиента, а в качестве факторов – совокупность данных, потенциально влияющих на результат. Алгоритм определит факторы, оказывающие наибольшее влияние на результат, а также типичные комбинации условий, приводящих к тому или иному результату [3].

Анализ данных по типу «Дерево решений» позволит оценить (спрогнозировать) ожидаемые значения целевого показателя на основании актуальных данных, а также составить прогноз "что, если...", изменяя показатели, подаваемые на вход модели. Результаты анализа и прогноза с помощью деревьев решений позволяют существенно снизить влияние неопределенности бизнес-окружения на состояние компании, а также решать широкий спектр задач, связанных с выявлением сложных и неочевидных причинно-следственных связей. Алгоритм "Дерево решений" формирует причинно-следственную иерархию условий, приводящую к определенным решениям. В результате применения этого метода к обучающей выборке создается иерархическая (древовидная) структура правил расщепления вида "если... то...". Алгоритм анализа (обучения модели) сводится к итеративному процессу вычленения наиболее значимых условий и переходов между ними. Условия могут иметь как количественный, так и качественный характер и формируют "ветви" этого абстрактного дерева. Его "листву" образуют значения прогнозируемого атрибута (решения), которые, как и условия переходов, допускают и качественную, и количественную трактовку. Совокупность этих условий, налагаемых на факторы, и структура переходов между ними до конечного решения и образуют модель прогноза.

Приведем еще один сценарий использования анализа «Дерево решений» в сфере «Управление персоналом»: сценарий "Профилирование менеджеров отдела продаж по ключевым показателям эффективности".

Определение эффективности менеджеров (удержание, поиск клиентов, эффективность коммуникаций, инкассация условной и безусловной дебиторской задолженности, удельные показатели эффективности на клиента и т. д.) представляет интерес не только с точки зрения формирования системы

материального стимулирования менеджеров, но и с точки зрения эффективного нормирования параметров их деятельности. Алгоритм - "Деревья решений". Прогнозные атрибуты - ключевые показатели эффективности отдела продаж (количество ключевых клиентов, коэффициенты оттока и привлечения, упущенный доход в месяц, привлеченный доход в месяц, доход в месяц с клиента, суммарные поступления от клиентов и т. д.). Основные факторы - количество активных клиентов, выручка, доход, удельные показатели на клиента, эффективность коммуникации. В зависимости от прогнозных атрибутов состав факторов может существенно варьироваться. Пример закономерности: менеджеры, обеспечивающие лучшие показатели инкассации дебиторской задолженности (отношение поступлений ДС к выручке), имеют коэффициент удержания  $> 0,8$ ; коэффициент привлечения  $> 0,25$ ; число одновременно открытых сделок не более 15, но не менее 10; интенсивность событий в день не более 10, но не менее 3; число активных клиентов в периоде не менее 50, но не более 100.

Итак, деревья решений являются одним из наиболее эффективных инструментов интеллектуального анализа данных и предсказательной аналитики. Они представляют собой иерархические древовидные структуры, состоящие из решающих правил вида «Если ..., то ...». Правила автоматически генерируются в процессе обучения на обучающем множестве. Они могут служить алгоритмическими правилами для построения вероятности исхода какого-то события (покупка товара неким потребителем, выход из строя оборудования в ближайшем будущем и т.д.) [1].

Деревья решений успешно применяются на практике в следующих областях:

*Банковское дело.* Оценка кредитоспособности клиентов банка при выдаче кредитов в зависимости от таких факторов как возраст, уровень дохода, наличие иждивенцев, семейного положения, наличия движимого и недвижимого имущества, наличия других кредитов и т.п.

*Промышленность.* Контроль за качеством продукции (выявление дефектов), испытания без разрушений (например, проверка качества сварки) и т.д.

*Медицина.* Диагностика заболеваний (есть ли температура, как давно кашель, заложен ли нос, сердечный ритм, возраст, наличие флюорографии, как прослушиваются легкие, бронхи, наличие храпов и т.л.).

*Молекулярная биология.* Анализ строения аминокислот.

*Торговля.* Классификация клиентов и товаров.

*Автострахование.* Выбор параметров страховки ОСАГО в зависимости от значений характеристик автовладельца (стар вождения, возраст и т.п.).

Это далеко не полный список областей где можно использовать деревья решений. Вместе с анализом данных деревья решений постоянно расширяют круг своего использования, становясь важным инструментом управления бизнес-процессами и поддержки принятия решений. Метод эффективно применяется для тех проектов, которые имеют обозримое количество вариантов развития.

### Тема 3. Алгоритмы машинного обучения. Кластерный анализ в системе «1С:Предприятие»

Третий типа анализа, реализованный в «1С:Предприятие» в составе механизма анализа данных и прогнозирования, который мы рассмотрим, это Кластеризация.

Кластерный анализ – это математическая процедура многомерного анализа, позволяющая на основе множества показателей, характеризующих ряд объектов, сгруппировать их в кластеры таким образом, чтобы объекты, входящие в один кластер, были более однородными, сходными, по сравнению с объектами, входящими в другие кластеры.

Общая схема выполнения анализа «Кластеризация» в системе «1С:Предприятие» приведена на рис. 3.1 [3]. Этот тип анализа тесно связан с понятием расстояние между объектами, на основании которого и производится группировка по кластерам. Вычисление расстояния может производиться по разным метрикам (разными способами).

В типе анализа «Кластеризация» в системе «1С:Предприятие» поддерживаются следующие метрики:

- Евклидова метрика,
- Евклидова метрика в квадрате,
- Метрика города,
- Метрика доминирования.

После определения расстояний между объектами может использоваться один из нескольких алгоритмов распределения объектов по кластерам. Поддерживаются следующие методы кластеризации [3]:

- Ближняя связь,
- Дальняя связь,
- k-средних,



**Рис. 3.1. Общая схема выполнения анализа «Кластеризация»**

– Центр тяжести.

Источником данных для данного типа анализа может выступать результат запроса, таблица значений или область ячеек табличного документа. Любой из этих источников состоит из определенных колонок. Каждая колонка для анализа имеет определенный тип. Она определяется или как ТипКолонкиАнализаДанныхКластеризация.НеИспользуемая, или как ТипКолонкиАнализаДанныхКластеризация.Входная. Системное перечисление ТипКолонкиАнализаДанныхКластеризация вообще говоря имеет больше значений:

- ТипКолонкиАнализаДанныхКластеризация.ВходнаяИПрогнозируемая;
- ТипКолонкиАнализаДанныхКластеризация.Ключ;
- ТипКолонкиАнализаДанныхКластеризация.Прогнозируемая;
- ТипКолонкиАнализаДанныхКластеризация.Входная;
- ТипКолонкиАнализаДанныхКластеризация.НеИспользуемая.

Остальные значения системного перечисления используются при прогнозировании.

В качестве параметров, которые можно задать до проведения анализа и которые влияют на его расчеты, выступают следующие [3]:

- КоличествоКластеров – количество искомых кластеров (Число);
- ТипЗаполненияТаблицы – параметр задает, какие поля выводить в таблицу. (ТипЗаполненияТаблицыРезультатаАнализаДанных);
- Стандартизация – с помощью параметра указывается необходимость стандартизации данных (ТипСтандартизацииДанных);
- МераРасстояния – параметр указывает, каким образом вычислять расстояние между объектами (ТипМерыРасстояния);
- МетодКластеризации – параметр задает, каким методом выполнять кластеризацию (МетодКластеризации).

На введенных ранее данных о контрагентах реализуем анализ типа «Кластеризация». Справочник «Контрагенты» должен быть предварительно заполнен данными, ранее приведенными в табл. 2.1.

В обработке «АнализДанных» в форме добавим новую команду формы «ПровестиКластерныйАнализ». В модуле формы реализуем следующий код обработчика команды, выбрав при создании вариант «Создать на клиенте и процедуру на сервере без контекста» (созданную заготовку под серверную процедуру надо изменить на функцию):

```
&НаКлиенте
```

```
Процедура ПровестиКластерныйАнализ(Команда)
```

```
ПровестиКластерныйАнализНаСервере();
```

```
КонецПроцедуры
```

```
&НаСервере
```

```
Функция ПровестиКластерныйАнализНаСервере()
```

```
Анализ = Новый АнализДанных;
```

```
Анализ.ТипАнализа = Тип("АнализДанныхКластеризация");
```

```

Группа =
Справочники.Контрагенты.НайтиПоНаименованию("Юридические лица");
Запрос = Новый Запрос;
Запрос.Текст = "
|ВЫБРАТЬ
|Контрагенты.Ссылка,
|Контрагенты.КоличествоРозничныхТочек,
|Контрагенты.КоличествоАвтомобилей,
|Контрагенты.ВремяРаботыОрганизации,
|Контрагенты.ВремяЗаключенияДоговора,
|Контрагенты.ВидДоговора,
|Контрагенты.ПрекращениеОтношений
|ИЗ
|Справочник.Контрагенты КАК Контрагенты";
Запрос.УстановитьПараметр("Родитель", Группа);
Анализ.ИсточникДанных = Запрос.Выполнить();
// Выбор метрики.
Анализ.Параметры.МераРасстояния.Значение =
ТипМерыРасстоянияАнализаДанных.ЕвклидоваМетрикаВКвадрате;
// Выбор метода кластеризации.
Анализ.Параметры.МетодКластеризации.Значение =
МетодКластеризации.КСредних;
РезультатАнализа = Анализ.Выполнить();
Построитель = Новый ПостроительОтчетаАнализаДанных();
Построитель.Макет = Неопределено;
Построитель.ТипАнализа = Тип("АнализДанныхКластеризация");
ТабДок = Новый ТабличныйДокумент;
Построитель.Вывести(РезультатАнализа, ТабДок);
Возврат ТабДок;
КонецФункции

```

Вынесем кнопку команды «ПровестиКластерныйАнализ» на командную панель формы.

В режиме исполнения протестируем созданный тип анализа.

В табл. 3.1-3.2 приведены параметры анализа и полученные результаты анализа данных контрагентов по справочнику «Контрагенты».

Таблица 3.1

Параметры анализа



## Кластерный анализ

### Параметры анализа

Количество искомых кластеров:	3
Стандартизация:	Стандартизировать
Мера расстояния:	Евклидова метрика в квадрате
Метод кластеризации:	Метод k-средних

Продолжение табл. 3.1.

### Колонки источника данных

#### **Входные колонки**

Имя колонки	Тип данных	Вес
Ссылка	Дискретный	1
КоличествоРозничныхТочек	Непрерывный	1
КоличествоАвтомобилей	Непрерывный	1
ВремяРаботыОрганизации	Дискретный	1
ВремяЗаключенияДоговора	Дискретный	1
ВидДоговора	Дискретный	1
ПрекращениеОтношений	Дискретный	1

Таблица 3.2

### Полученные результаты кластерного анализа

#### Информация о данных

Количество объектов: 9

#### Результат анализа

Найдено кластеров: 3

#### Кластеры

№ кластера	Количество объектов	Процент
1	6	66,67
2	2	22,22
3	1	11,11

**Центры кластеров**

Поле \ Кластер	1	2	3
Ссылка = ЗАО МДК	0,1667		
Ссылка = ИП Огарьков С.К.	0,1667		
Ссылка = Магазин №23	0,1667		
Ссылка = ООО "БыстраяПицца"	0,1667		
Ссылка = ООО "Силай"	0,1667		
Ссылка = ИП Добров К.Л.	0,1667		
Ссылка = ООО "Лидер"		0,5000	
Ссылка = ООО "Василиса"		0,5000	
Ссылка = ООО "ДубМебель"			1,0000
КоличествоРозничныхТочек	1,3333	11,0000	1,0000
КоличествоАвтомобилей	1,1667	3,5000	10,0000
ВремяРаботыОрганизации = Меньше года	0,5000		
ВремяРаботыОрганизации = От года до трех	0,1667		
ВремяРаботыОрганизации = Свыше десяти лет	0,3333		
ВремяРаботыОрганизации = От трех до десяти лет		1,0000	1,0000
ВремяЗаключенияДоговора = Меньше года	0,6667	0,5000	
ВремяЗаключенияДоговора = От трех до десяти лет	0,1667		
ВремяЗаключенияДоговора = Свыше десяти лет	0,1667		
ВремяЗаключенияДоговора = От года до трех		0,5000	1,0000
ВидДоговора = Дилер	0,5000		
ВидДоговора = Постоянный партнер	0,5000	0,5000	
ВидДоговора = Дистрибьютор		0,5000	1,0000
ПрекращениеОтношений = Несоблюдение договора	0,1667		
ПрекращениеОтношений = Прекращение контрагентом	0,1667	1,0000	1,0000
ПрекращениеОтношений = Не прекращены	0,6667		

Продолжение табл. 3.2.

**Расстояния между кластерами**

Кластеры	1	2	3
1		8,7184	15,3968
2	8,7184		11,5370
3	15,3968	11,5370	

Представленные выше результаты содержат данные о найденных кластерах (их количество, центры, расстояния между ними). Чтобы получить подробную информацию о том, какие объекты входят в какие кластеры, надо в настройках параметров проводимого анализа явно указать:

Анализ.Параметры.ТипЗаполненияТаблицы.Значение =

ТипЗаполненияТаблицыРезультатаАнализаДанных.ИспользуемыеПоля;

Добавим эту строку кода в наш код обработчик команды. Ее надо добавить перед выполнением анализа, но после определения типа анализа.

Протестируем в режиме исполнения. Получаемые результаты дополнились таблицей классификации (табл. 3.3).

Таблица 3.3

**Распределение объектов по кластерам**

Таблица классификации

Ссылка	Количество	Количество	ВремяРаботыОрганизации	ВремяЗаключенияДоговора	ВидДоговора	ПрекращениеОтношений	Кластер
ЗАО МДК	1		Меньше года	Меньше года	Дилер	Несоблюдение договора	1
ООО "Лидер"	15	4	От трех до десяти лет	Меньше года	Дистрибьютор	Прекращение контрагентом	2
ООО "ДубМебель"	1	10	От трех до десяти лет	От года до трех	Дистрибьютор	Прекращение контрагентом	3
ИП Огарьков С.К.	1	1	От года до трех	Меньше года	Дилер	Прекращение контрагентом	1
Магазин №23	1	1	Свыше десяти лет	От трех до десяти лет	Постоянный партнер	Не прекращены	1
ООО "БыстраяПицца"	3	2	Меньше года	Меньше года	Постоянный партнер	Не прекращены	1
ООО "Василиса"	7	3	От трех до десяти лет	От года до трех	Постоянный партнер	Прекращение контрагентом	2
ООО "Силай"	2	2	Свыше десяти лет	Свыше десяти лет	Постоянный партнер	Не прекращены	1
ИП Добров К.Л.		1	Меньше года	Меньше года	Дилер	Не прекращены	1

Как видно из табл. 3.1 в нашем примере входные колонки имели как непрерывный тип (для этого типа очевидно понятие «расстояния»), так и были колонки дискретных типов (ссылки на справочники, значения перечислений).

Состав колонок определился, исходя из состава полей выборки запроса. По умолчанию они определены с равным весом. Для типов Число и Дата определен вид данных Непрерывные, для остальных типов - Дискретные. При необходимости изменить параметры колонок это можно сделать по аналогии с приведенным фрагментом:

Анализ.НастройкаКолонок.КоличествоАвтомобилей.ДополнительныеПараметры.Вес = 2;

В данной строке кода для колонки КоличествоАвтомобилей увеличен вес.

Рассмотрим подробнее метрики, которые могут быть использованы в анализе «Кластеризация» [3].

### ***Евклидова метрика.***

Расстояние между двумя объектами вычисляется по формуле:

$$\text{Расстояние} = \sqrt{\sum (W_i * (X_i - Y_i)^2)}$$

где:

$X_i, Y_i$  – значения атрибутов двух объектов (между которыми определяется расстояние);

$W_i$  – весовой коэффициент атрибута (устанавливается в колонке анализа);

$i$  – номер атрибута, от 1 до  $n$ ;

$n$  – число атрибутов.

*Расчетный пример:* рассчитайте расстояние между объектами с использованием Евклидовой метрики, если объекты характеризуются одним свойством, которое у одного объекта равно 7, а у другого равно 4. Весовой коэффициент данного атрибута равен 1.

### ***Евклидова метрика в квадрате.***

Расстояние между двумя объектами вычисляется по формуле:

$$\text{Расстояние} = \sum (W_i * (X_i - Y_i)^2),$$

где:

$X_i, Y_i$  – значения атрибутов двух объектов (между которыми определяется расстояние);

$W_i$  – весовой коэффициент атрибута (устанавливается в колонке анализа);

$i$  – номер атрибута, от 1 до  $n$ ;

$n$  – число атрибутов.

*Расчетный пример:* рассчитайте расстояние между объектами с использованием Евклидовой метрики в квадрате, если объекты характеризуются одним свойством, которое у одного объекта равно 8, а у другого равно 3. Весовой коэффициент данного атрибута равен 3.

**Метрика города.**

Расстояние между двумя объектами вычисляется по формуле:

$$\text{Расстояние} = \sum (W_i * |X_i - Y_i|)$$

где:

$X_i, Y_i$  – значения атрибутов двух объектов (между которыми определяется расстояние);

$W_i$  – весовой коэффициент атрибута (устанавливается в колонке анализа);

$i$  – номер атрибута, от 1 до  $n$ ;

$n$  – число атрибутов.

*Расчетный пример:* рассчитайте расстояние между объектами с использованием Метрики города, если объекты характеризуются двумя атрибутам:

- первый атрибут равен у объектов соответственно 4 и 6. Вес атрибута 1;
- второй атрибут равен у объектов соответственно 3 и 7. Вес атрибута 2.

**Метрика доминирования.**

Расстояние между двумя объектами вычисляется по формуле:

$$\text{MAX}(W_i * |X_i - Y_i|)$$

где:

$X_i, Y_i$  – значения атрибутов двух объектов (между которыми определяется расстояние);

$W_i$  – весовой коэффициент атрибута (устанавливается в колонке анализа);

$i$  – номер атрибута, от 1 до  $n$ ;

$n$  – число атрибутов.

*Расчетный пример:* рассчитайте расстояние между объектами с использованием Метрики доминирования, если объекты характеризуются двумя атрибутам:

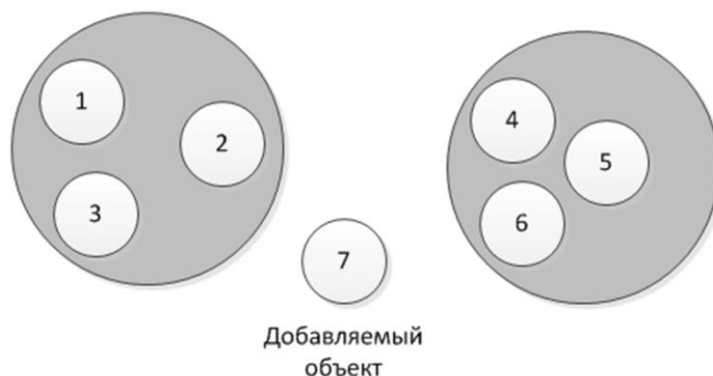
- первый атрибут равен у объектов соответственно 4 и 6. Вес атрибута 1;
- второй атрибут равен у объектов соответственно 3 и 7. Вес атрибута 2.

Рассмотрим подробнее **методы кластеризации**, которые определяют, исходя из каких принципов объект соотносится к той или иной группе, по какому алгоритму производится формирование кластеров [3].

Пусть имеющиеся объекты образуют 2 группы (рис. 3.2).

Можно сказать, что целью любого алгоритма кластеризации является:

- минимизация изменчивости внутри кластеров,



**Рис. 3.2. Группы объектов**

– максимизация изменчивости между кластерами.

**Ближняя связь.** Метод кластеризации, в котором объект присоединяется к той группе, для которой расстояние до ближайшего объекта минимально.

В примере на рис 3.2 объект 7 будет включен в группу, в которой находится объект 4. Самыми близкими объектами двух групп являются объекты 4 и 3. Расстояние до объекта 4 минимально.

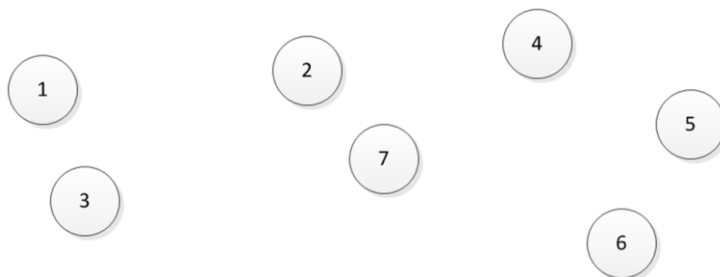
**Дальняя связь.** Метод кластеризации, в котором объект присоединяется к той группе, для которой расстояние до наиболее дальнего объекта минимально.

В примере на рис 3.2 объект 7 будет включен в группу, в которой находится объект 5. Самыми дальними объектами двух групп являются объекты 1 и 5. Расстояние до объекта 5 меньше.

**Центр тяжести.** Метод кластеризации, в котором объект присоединяется к той группе, для которой расстояние до центра тяжести минимально.

В примере на рис 3.2 объект 7 добавится в группу, содержащую объекты 4, 5 и 6. Расстояние до центра тяжести (некоего мифического объекта со средними значениями атрибутов) минимально.

**К-средних.** В данном методе выбираются объекты, находящиеся первыми в выборке. Они считаются центрами кластеров. Далее выбирается следующий объект и, в соответствии с расстоянием до центров кластеров, относится к тому или иному кластеру. Центр кластера, к которому был добавлен объект, пересчитывается.



**Рис. 3.3. Пример расположения объектов**

Процедура повторяется до полного перебора всех объектов. Далее опять производится новая выборка объектов (начиная с первого). Процедура повторяется до тех пор, пока изменяются центры кластеров (рис. 3.3).

Предположим, что произвольно выбраны в качестве центров кластеров объекты 1 и 2. Объект 3 добавляется к кластеру, центром которого является объект 1. Центр первого кластера перерасчитывается (он находится между объектом 1 и 3). Объект 4 добавляется ко второму кластеру (его центр также перерасчитывается).

После перебора всех анализируемых объектов к первому кластеру относятся объекты 1 и 3, ко второму – остальные объекты (его центр предположительно находится в центре треугольника из объектов 4, 7, 6).

Далее опять производится выборка объектов и распределение их по кластерам (относительно постоянно рассчитываемых центров кластеров).

Где-то на третьей выборке объектов, скорее всего, объект 2, который изначально был центром второго кластера, станет относиться к первому кластеру.

В конце алгоритма к первому кластеру будут относиться объекты 1, 2, 3.  
Ко второму – объекты 4, 5, 6, 7.

Реализуем в нашей обработке возможность выбора метода кластеризации пользователем на форме и затем подстановку в параметр анализа выбранного метода.

Добавим реквизит формы «ВыборМетодаКластеризации», укажем для него тип системного перечисления «МетодКластеризации». Вынесем его на форму, расположив под реквизитом формы «ТипУпрощения», используемым нами ранее для типа анализа «Дерево решений» (рис. 3.4).

**Рис. 3.4. Выбор метода кластеризации на форме обработки «АнализДанных».**

В код обработчика на сервере внесем коррективы. Вместо строки

```
Анализ.Параметры.МетодКластеризации.Значение =
МетодКластеризации.КСредних;
```

которая жестко задавала метод кластеризации равным «КСредних», добавим следующий блок:

```
Анализ.Параметры.МетодКластеризации.Значение =
ВыборМетодаКластеризации;
```

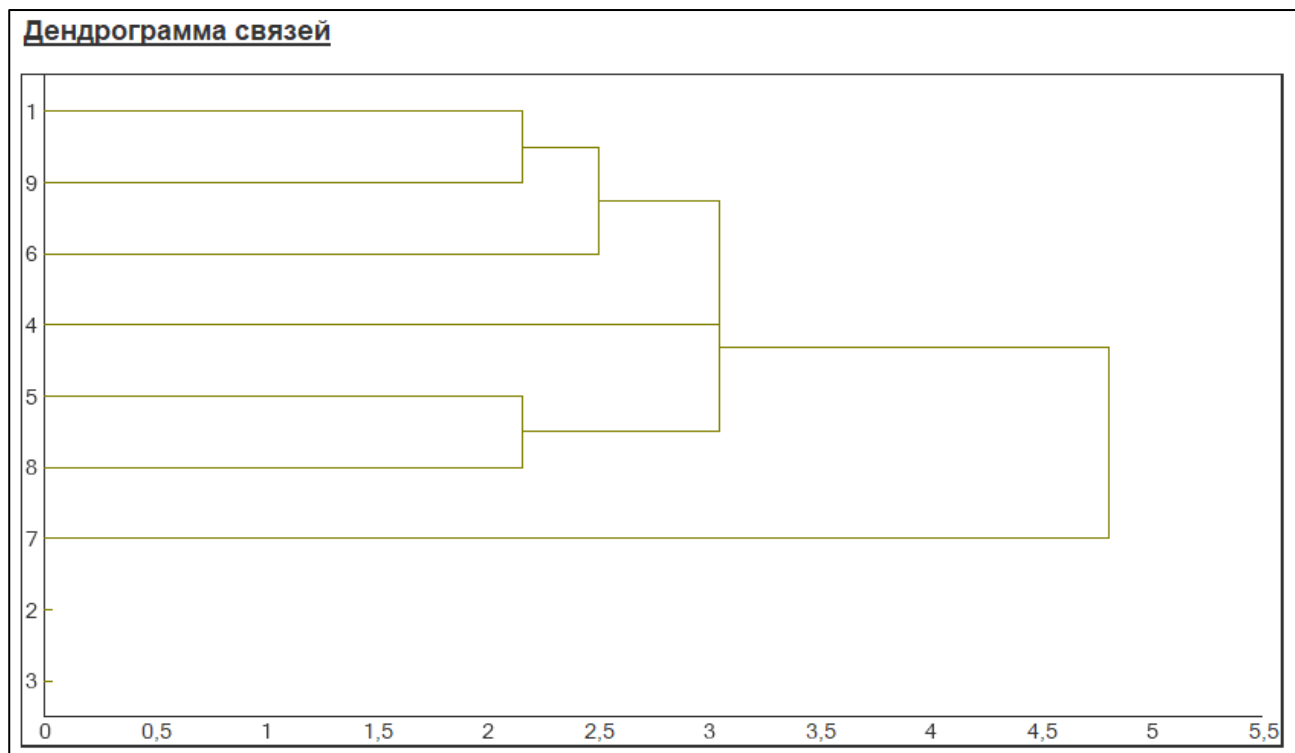
Как видно, в приведенном фрагменте в качестве метода кластеризации в соответствующий параметр подставляется метод, выбранный пользователем.

Протестируем в режиме исполнения. Перед проведением анализа выберем метод, отличный от «КСредних» (если пользователь не осуществляет сам выбор, то этот метод подставляется по умолчанию). Выберем, например, «Метод ближайшей связи» или «Метод центра тяжести».

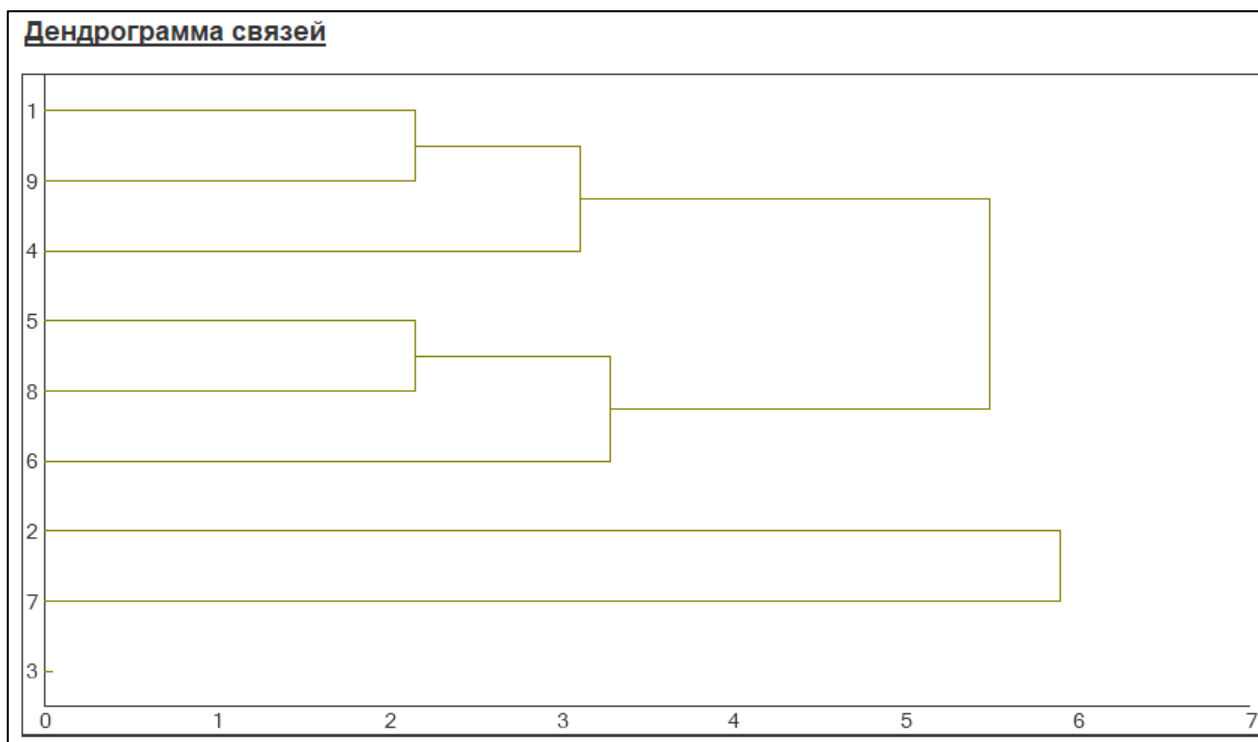
При выводе данных кластерного анализа, если используется алгоритм, отличный от алгоритма k-средних, результаты кластерного анализа выводятся еще и в виде дендрограммы (алгоритм анализа должен предусматривать вывод распределения анализируемых объектов по кластерам).

Дендрограмма представляет собой специализированный вид диаграмм, используемый, чаще всего, для визуализации результатов кластерного анализа. В общем случае дендрограмма показывает, насколько одни объекты похожи на другие объекты, или на группы объектов. Сравнимые объекты располагаются на одной оси дендрограммы, а другая ось содержит значения расстояния между объектами. Чем меньше расстояние — тем больше объекты похожи друг на друга (рис. 3.5 – 3.7).

**Интерпретация дендрограммы.** В левой части дендрограммы каждый лист дендрограммы представляет одно из исходных наблюдений. Но по мере того, как мы продвигаемся вверх к корню, некоторые листья начинают сливаться в ветви. Они соответствуют наблюдениям, которые похожи друг на друга. При продвижении вверх по дереву ветви сливаются с листьями или с другими ветвями. Чем раньше (ниже на дереве) происходит слияние, тем больше похожи группы наблюдений друг на друга.



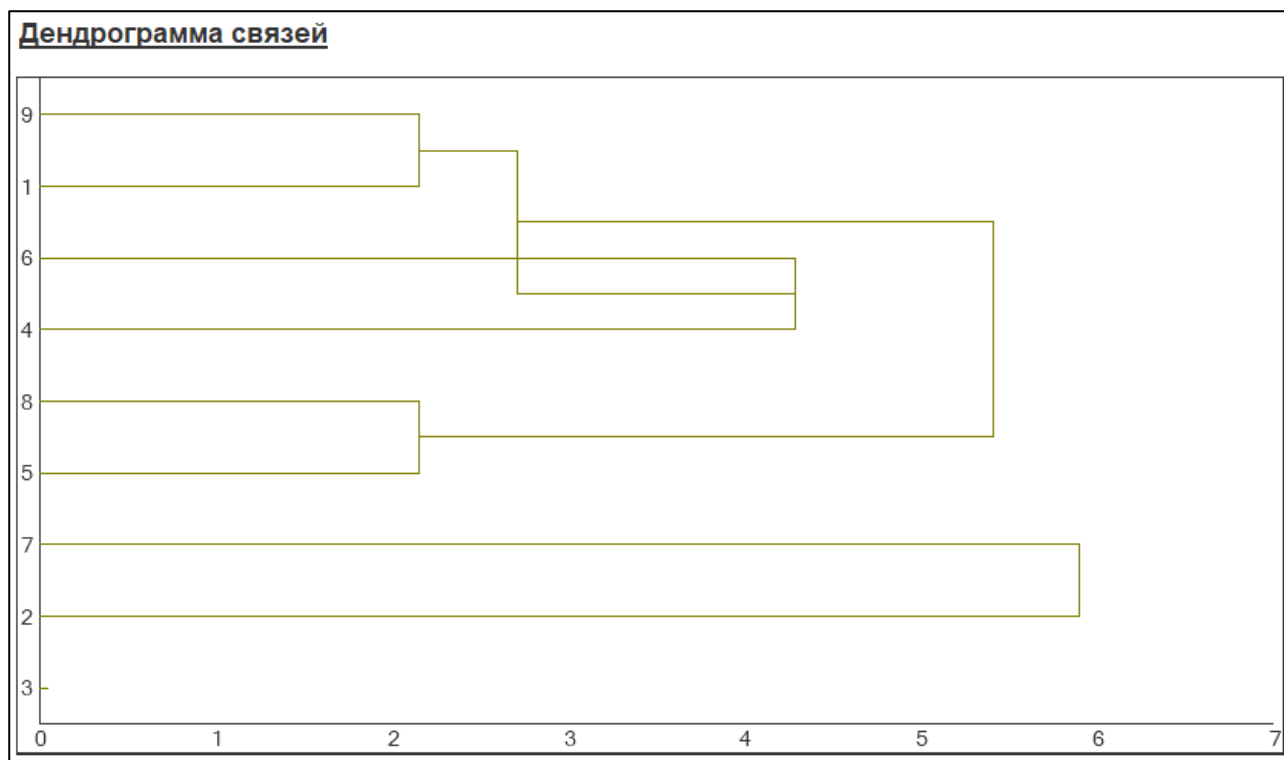
**Рис. 3.5. Дендрограмма связей при выбранном методе кластеризации «Метод ближней связи»**



**Рис. 3.6. Дендрограмма связей при выбранном методе кластеризации «Метод дальней связи»**

С другой стороны, наблюдения, которые сливаются позже (возле верхушки дерева), могут сильно отличаться. Это утверждение можно сформулировать точно: для любых двух наблюдений можно найти точку на дереве, где ветви, содержащие эти наблюдения, впервые сливаются. Высота этой точки слияния, измеренная по вертикальной оси, указывает, насколько различаются эти наблюдения. Таким образом, наблюдения, которые сливаются в самом низу дерева, очень похожи друг на друга, в то время как наблюдения, которые сливаются близко к верху дерева, будут скорее всего сильно различаться.





**Рис. 3.7. Дендрограмма связей при выбранном методе кластеризации «Метод центра тяжести»**

Таким образом, кластерный анализ – объединение объектов в группы (кластеры) по некоторым неочевидным общим признакам.

**Анализ. ТипАнализа = Тип("АнализДанныхКластеризация");**

Цель кластеризации – выделение из множества объектов одной природы некоторого количества относительно однородных групп — сегментов или кластеров. Объекты распределяются по группам так, чтобы внутригрупповые отличия были минимальными, а межгрупповые — максимальными.

Любой из реализованных в платформе методов кластеризации предполагает явное указание количества искомым кластеров. Для атрибутов объектов можно вводить веса, что позволяет расставлять приоритеты между ними.

Методы кластеризации позволяют перейти от пообъектного к групповому представлению совокупности произвольных объектов, что существенно упрощает оперирование ими.

Алгоритмы кластеризации позволяют не только провести кластерный анализ объектов на множестве заданных атрибутов, но и спрогнозировать значение одного или нескольких из них для актуальной выборки на основании отнесения объектов этой выборки к тому или иному кластеру [3].

Возможные сценарии применения кластеризации на практике.

- Сегментация клиентов по определенной совокупности параметров позволяет выделить среди них устойчивые группы, имеющие сходные покупательские предпочтения, уровни продаж и

платежеспособности, что существенно упрощает управление взаимоотношениями с клиентами.

- При классификации товаров часто используются достаточно условные принципы классификации. Выделение сегментов на основе группы формальных критериев позволяет определить действительно однородные группы товаров. В условиях широкой и довольно разнородной номенклатуры товаров управление ассортиментом на уровне сегментов, по сравнению с управлением на уровне номенклатуры, существенно повышает эффективность продвижения, ценообразования, мерчендайзинга, управления цепочками поставок.
- Сегментация менеджеров позволяет эффективнее спланировать организационные изменения, улучшить мотивационные схемы, скорректировать требования к нанимаемому персоналу, что в конечном счете позволяет повысить управляемость компании и стабильность бизнеса в целом.

Примером сценария в сфере управления взаимоотношениями с клиентами может быть сценарий — «Планирование рекламной кампании».

Планирование предстоящей рекламной кампании рассматривается с точки зрения оптимизации распределения выделенного бюджета по рекламным каналам исходя из регионального, продуктового, клиентского и иных показателей целевого сегмента, а также эффективности рекламных каналов в указанных разрезах в некотором, предшествующем планируемому периоде.

Алгоритм — «Кластерный анализ».

Прогнозные атрибуты — доли откликов на рекламный канал условно однородных сегментов, выделенных алгоритмом.

Вычисляемые колонки — доли рекламных каналов в бюджете рекламной кампании с учетом вероятной доли откликов и эффективности (в смысле результирующей выручки) каждого рекламного канала.

Пример закономерности. Клиенты класса А региона П, предпочитающие товарную группу Р, привлечены тем же рекламным каналом, что и клиенты региона Н, предпочитающие товарную группу У.

Кластерный анализ играет важную роль на этапе редукции экономико-математической модели товарной конъюнктуры, способствуя облегчению и упрощению вычислительных процедур, обеспечению большей компактности получаемых результатов при одновременном сохранении необходимой точности. Применение кластерного анализа дает возможность разбить всю исходную совокупность показателей конъюнктуры на группы (кластеры) по соответствующим критериям, облегчая тем самым выбор наиболее репрезентативных показателей. Практически основное большинство задач прогнозирования опирается на использование кластерного анализа.

## Тема 4. Практическое использование языка запросов «1С:Предприятие»

Практическое освоение языка запросов будет производиться с использованием внешней обработки Консоль запросов, которая предназначена для составления и исполнения запросов. Скачать ее можно с сайта 1С:ИТС по адресу:

<https://its.1c.ru/db/files/1CITS/EXE/ExtReps/Unireps82/RequestConsole/RequestConsole.zip>

Обработка «КонсольЗапросов.epf» находится в этом архиве в каталоге \1CITS\EXE\ExtReps\Unireps82\RequestConsole\ Чтобы полностью использовать все функциональные возможности консоли запросов, ее нужно запускать в режиме Толстый клиент.

Запустим нашу конфигурацию в режиме исполнения Толстый клиент. В меню выберем Файл->Открыть и откроем нашу обработку «Консоль запросов». Для проверки работоспособности создадим в ней в помощью конструктора запроса простой запрос к справочнику Номенклатура (рис. 4.1). Сформируем текст запроса в поле «Текст запроса» с помощью конструктора запроса, вызванного из контекстного меню по правой кнопке мыши. Затем нажмем кнопку «Выполнить». В поле «Результат запроса» будет отражен результат запрос к информационной базе.

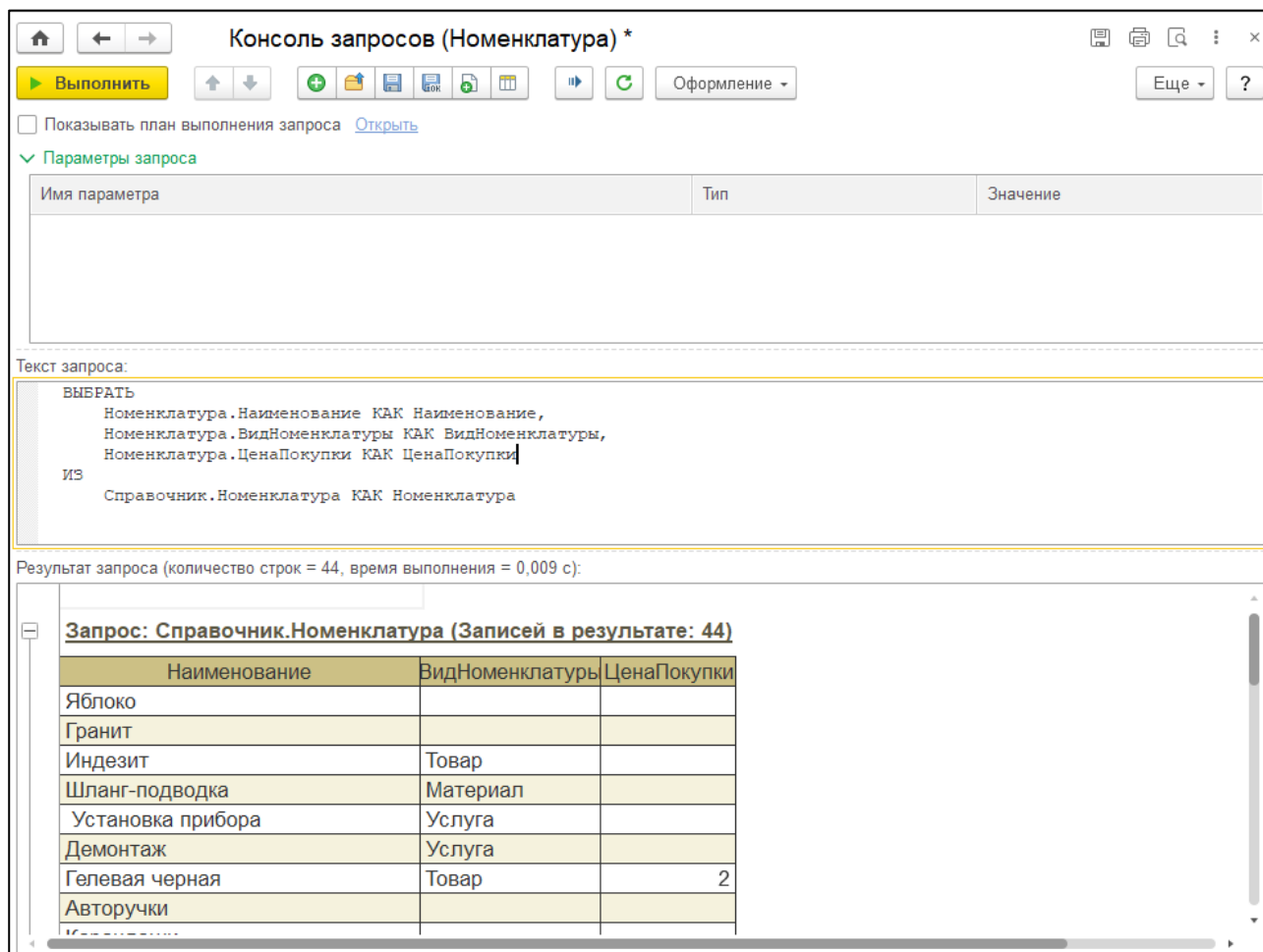


Рис. 4.1. Внешняя обработка «Консоль запросов»

Текст любого запроса всегда содержит секцию описания запроса, в которой определяются источники данных для запроса, список полей выборки и т. д. Все секции запроса приведены на рис. 4.2 [4].

Секции запроса

<p>-----</p> <p>ВЫБРАТЬ</p> <p>    Приход.Ссылка КАК Документ,            Приход.Товар КАК Товар,            Сумма(Приход.Количество) КАК КоличествоВсего,            Сумма(Приход.Сумма) КАК СуммаВсего</p> <p>ИЗ</p> <p>    Документ.ПриходнаяНакладная.Состав КАК Приход</p> <p>СГРУППИРОВАТЬ ПО</p> <p>    Приход.Ссылка,            Приход.Товар</p> <p>-----</p>	<p>Описание запроса</p>
<p>ОБЪЕДИНИТЬ ВСЕ</p> <p>-----</p>	<p>Объединение запросов</p>
<p>ВЫБРАТЬ</p> <p>    Расход.Ссылка,            Расход.Товар,            Сумма(Расход.Количество),            Сумма(Расход.Сумма)</p> <p>ИЗ</p> <p>    Документ.РасходнаяНакладная.Состав КАК Расход</p> <p>СГРУППИРОВАТЬ ПО</p> <p>    Расход.Ссылка,            Расход.Товар</p> <p>-----</p>	<p>Описание запроса</p>
<p>УПОРЯДОЧИТЬ ПО</p> <p>    Документ,            Товар</p> <p>-----</p>	<p>Упорядочивание результатов</p>
<p>АВТОУПОРЯДОЧИВАНИЕ</p> <p>-----</p>	<p>Автоупорядочивание</p>
<p>ИТОГИ</p> <p>    Сумма(КоличествоВсего),            Сумма(СуммаВсего)</p> <p>ПО</p> <p>    Документ</p> <p>-----</p>	<p>Описание итогов</p>

**Рис. 4.2. Пример запроса со всеми секциями в тексте запроса**

Язык запросов «1С:Предприятия» основан на стандартном SQL, но при этом содержит значительное количество расширений, ориентированных на финансово-экономические задачи, и значительно облегчает разработку бизнес-приложений. Язык запросов – мощный инструмент, предоставляющий разнообразные возможности получения данных. Рассмотрим некоторые из этих возможностей на практических примерах [4].

***Пример №4.1. Получение всех данных из таблицы***

Чтобы получить все данные, например, справочника «Контрагенты» можно использовать следующий текст запроса:

ВЫБРАТЬ

Справочник.Контрагенты.\*

Когда нужны все поля таблицы вместо перечисления имен полей можно использовать звездочку «\*».

Второй вариант этого запроса:

ВЫБРАТЬ

\*

ИЗ

Справочник.Контрагенты

***Пример №4.2. Упорядочивание записей в результате запроса.***

Возможность упорядочивания записей в результате запроса – важная возможность языка запросов, поскольку если его не использовать, то порядок записей в запросе будет не определен. Один и тот же запрос, выполненный на разных СУБД, может дать разный порядок записей. Поэтому в запросе очень важно всегда явно указывать, как должны быть упорядочены записи [4].

Порядок расположения записей в результате запроса определяется в секции Упорядочивание результатов текста запроса.

ВЫБРАТЬ

Цены.Период КАК Период,

Цены.Товар КАК Товар,

Цены.Цена КАК Цена

ИЗ

РегистрСведений.Цены КАК Цены

УПОРЯДОЧИТЬ ПО

Период ВОЗР

Поле, по которому производится упорядочивание, необязательно должно входить в список выборки запроса. Чтобы расположить записи результата запроса по убыванию какого-либо поля, нужно в предложении упорядочивания указать после имени поля ключевое слово УБЫВ. В предложении УПОРЯДОЧИТЬ ПО могут участвовать несколько полей, каждое из этих полей может использовать разный способ упорядочивания. В этом случае записи результата запроса будут упорядочены сначала по первому полю, затем – по второму полю (в случае, если существует несколько записей с одинаковым значением первого поля) и т. д.

Если используется упорядочивание по ссылочному полю, то полезно использовать дополнительно конструкцию АВТОУПОРЯДОЧИВАНИЕ [4].

ВЫБРАТЬ

ПродажаТовара.Дата,  
ПродажаТовара.Номер,  
ПродажаТовара.Клиент,  
ПродажаТовара.Сумма

ИЗ

Документ. ПродажаТовара КАК ПродажаТовара

УПОРЯДОЧИТЬ ПО

ПродажаТовара.Ссылка  
АВТОУПОРЯДОЧИВАНИЕ

***Пример №4.3. Получение только первых несколько записей с наибольшими значениями некоторого поля.***

Часто требуется получить из большой таблицы только определенное количество записей с наибольшим или наименьшим значением какого-то поля. Например, нам нужно получить десять документов продаж с наибольшей суммой продажи.

ВЫБРАТЬ ПЕРВЫЕ 10

ПродажаТовара.Номер,  
ПродажаТовара.Клиент,  
ПродажаТовара.Сумма КАК Сумма

ИЗ

Документ.ПродажаТовара КАК ПродажаТовара

УПОРЯДОЧИТЬ ПО

Сумма УБЫВ

Конструкция ВЫБРАТЬ ПЕРВЫЕ <Количество> позволяет задать предельное количество строк в результате запроса. Чтобы выбрать десять продаж с наименьшей суммой заказа, нужно в этом же примере поменять порядок сортировки по полю Сумма.

***Пример №4.4. Получение записей, в которых определенные поля не содержат одинаковых значений.***

Если нужно получить только записи с неповторяющимися значениями определенного поля, то следует использовать конструкцию ВЫБРАТЬ РАЗЛИЧНЫЕ [4].

ВЫБРАТЬ РАЗЛИЧНЫЕ

ЗаказТовара.Клиент

ИЗ

Документ.ЗаказТовара КАК ЗаказТовара

Если в списке выборки указано несколько полей, то при использовании конструкции ВЫБРАТЬ РАЗЛИЧНЫЕ в результат запроса отбираются записи, содержащие неповторяющиеся комбинации значений сразу по нескольким полям.

**Пример №4.5. Получение общего количества записей в таблице и количества записей с различным значением некоторого поля**

Если требуется получить количество записей в результате запроса. Это делается с помощью агрегатной функции КОЛИЧЕСТВО().

```

ВЫБРАТЬ
    КОЛИЧЕСТВО(*) КАК Всего,
    КОЛИЧЕСТВО(РАЗЛИЧНЫЕ ПоступлениеТоваров.Контрагент) КАК
РазныеКонтрагенты
ИЗ
    Документ.ПоступлениеТоваров КАК ПоступлениеТоваров

```

Как видно из примера, в списке полей выборки, после ключевого слова ВЫБРАТЬ, можно использовать не только поля таблиц запроса, но и различные выражения с их использованием, то есть агрегатные функции, функции языка запросов и т. п.

**Пример №4.6. Получение записей таблицы, содержащих строки, соответствующие заданному шаблону**

Используется оператор ПОДОБНО. Результатом выполнения оператора ПОДОБНО будет Истина или Ложь в зависимости от того, удовлетворяет шаблону значение выражения (в данном случае поля Наименование) или нет [4].

```

ВЫБРАТЬ
    Контрагенты.Наименование КАК Наименование,
    Контрагенты.Адрес,
    Контрагенты.Телефон
ИЗ

```

```

    Справочник.Контрагенты КАК Контрагенты
ГДЕ

```

```

    Наименование ПОДОБНО "%Александр%"

```

Символ «%» (процент) заменяет в шаблоне строки любую последовательность символов. Символ «\_» (подчеркивание) заменяет в шаблоне строки один произвольный символ. Например, чтобы узнать, телефоны каких контрагентов не соответствуют шаблону «\_ -\_\_ -\_\_ -\_\_ -\_\_», можно изменить условие и подставить в него такой шаблон. С помощью оператора НЕ ... ПОДОБНО можно найти те строки, которые не соответствуют некоторому шаблону, например, узнать, какие телефоны были введены в базу данных неправильно.

**Пример №4.7. Получение данных из табличной части некоторого документа**

Если необходимо вывести все данные из табличной части определенного документа или элемента справочника или другого объекта конфигурации, имеющего табличную часть, для этого нужно обращаться к табличной части как к отдельной таблице [4]. При этом чтобы получить данные из основной таблицы документа, нужно обращаться к полям таблицы через точку от поля

табличной части Ссылка. Например, поле Контрагент находится в основной таблице.

**ВЫБРАТЬ**

Товары.Ссылка.Контрагент,  
Товары.Номенклатура,  
Товары.Количество,  
Товары.Сумма

**ИЗ**

Документ.ПродажаТоваров.Товары КАК Товары

**Пример №4.8. Вычисление минимального, максимального, среднего значения**

**ВЫБРАТЬ**

ПродажаТоваровТовары.Номенклатура КАК Номенклатура,  
МИНИМУМ(ПродажаТоваровТовары.Цена) КАК Минимум,  
МАКСИМУМ(ПродажаТоваровТовары.Цена) КАК Максимум,  
СРЕДНЕЕ(ПродажаТоваровТовары.Цена) КАК Среднее

**ИЗ**

Документ.ПродажаТоваров.Товары КАК ПродажаТоваровТовары  
**СГРУППИРОВАТЬ ПО**

ПродажаТоваровТовары.Номенклатура

Для решения задачи использованы агрегатные функции МИНИМУМ(), МАКСИМУМ(), СРЕДНЕЕ() и ключевое слово СГРУППИРОВАТЬ ПО, после которого следует список полей, по которым нужно сгруппировать записи исходной таблицы. Следует отметить, что в списке полей выборки запроса, помимо полей, по которым группируются записи исходной таблицы, могут присутствовать только агрегатные функции СУММА(), МИНИМУМ(), МАКСИМУМ(), СРЕДНЕЕ(), КОЛИЧЕСТВО(), применяемые к исходным записям с одинаковым значением поля группировки.

В предложении СГРУППИРОВАТЬ ПО должны указываться именно имена полей, а не их псевдонимы, определенные в запросе.

**Пример №4.9. Расчет итогов по нескольким полям.**

Часто требуется рассчитать итоги результата запроса по значению нескольких полей сразу. Для этого после ключевого слова ПО нужно перечислить через запятую список полей – контрольных точек [4].

**ВЫБРАТЬ**

ПоступлениеТоваровТовары.Ссылка.Контрагент КАК Поставщик,  
ПоступлениеТоваровТовары.Номенклатура КАК Товар,  
ПоступлениеТоваровТовары.Количество КАК Количество,  
ПоступлениеТоваровТовары.Цена КАК Цена,  
ПоступлениеТоваровТовары.Сумма КАК Сумма

**ИЗ**

Документ.ПоступлениеТоваров.Товары КАК ПоступлениеТоваровТовары  
**ИТОГИ**



СУММА(Количество),  
СУММА(Сумма)

ПО

Поставщик,  
Товар

В результате итоговые поля рассчитываются сначала для каждого товара по каждому поставщику и затем отдельно для каждого поставщика по всем товарам в целом.

#### ***Пример №4.10. Расчет общих итогов***

Помимо итогов по контрольным точкам, можно задать расчет общих итогов результата запроса. Для этого после ключевого слова ПО необходимо указать ключевое слово ОБЩИЕ.

ВЫБРАТЬ

ПоступлениеТоваровТовары.Ссылка.Контрагент КАК Поставщик,  
ПоступлениеТоваровТовары.Номенклатура КАК Товар,  
ПоступлениеТоваровТовары.Количество КАК Количество,  
ПоступлениеТоваровТовары.Сумма КАК Сумма

ИЗ

Документ.ПоступлениеТоваров.Товары КАК ПоступлениеТоваровТовары

ИТОГИ

СУММА(Количество),  
СУММА(Сумма)

ПО

ОБЩИЕ,  
Поставщик

В результате итоговые поля рассчитываются для каждого поставщика, и, кроме того, в результат запроса добавляется самая верхняя итоговая строка, содержащая итоги для всех поступлений в целом.

***Пример №4.12. Использование выражений в списке полей выборки запроса.*** В тексте запросов можно использовать разнообразные выражения: литералы, агрегатные функции, функции языка запросов, операции выбора и т. п.

Выражение языка запросов может содержать [4]:

– Литералы типов: число, строка (в кавычках), булево (значения Истина и Ложь), Null, Неопределено. Чтобы указать литерал типа Дата, можно воспользоваться ключевым словом языка запросов ДАТАВРЕМЯ или передать дату через параметр запроса.

– Арифметические операции (+, -, /, \*). Операция получения остатка % в языке запросов не поддерживается.

– Операцию конкатенации строк (+). Операцию конкатенации нельзя использовать для виртуальных полей.

– Встроенные функции языка запросов (ДЕНЬ, МЕСЯЦ, ГОД и т. д.).

– Агрегатные функции (СУММА, МИНИМУМ, МАКСИМУМ, СРЕДНЕЕ, КОЛИЧЕСТВО).

– Операцию выбора ВЫБОР – позволяет получить одно из возможных значений в соответствии с указанными условиями.

– Операцию приведения типов ВЫРАЗИТЬ – позволяет привести значение составного типа к одному из составляющих это значение типов. А также функцию ВЫРАЗИТЬ() используют для получения результатов нужной длины и точности.

Например, попробуем вывести записи из иерархического справочника Номенклатура. При этом после поля Наименование в скобках мы хотим видеть Код товара. Также для каждой записи справочника мы хотим видеть текстовое обозначение, является ли эта запись группой или нет («Это группа»/«Это элемент»).

ВЫБРАТЬ

Товары.Наименование + " (" + Товары.Код + ")" КАК Товар,

ВЫБОР

КОГДА Товары.ЭтоГруппа = ИСТИНА ТОГДА "Это группа"

ИНАЧЕ "Это элемент"

КОНЕЦ КАК ПризнакГруппы

ИЗ

Справочник.Номенклатура КАК Товары

Для формирования поля выборки Товар в данном запросе используется операция конкатенации (сложения) строк, в результате наименование и код товара выводятся в одном текстовом поле выборки результата запроса.

**Пример №4.13. Использование данных одного запроса внутри другого запроса.** Иногда внутри одного запроса необходимо использовать данные другого запроса. Например, требуется ограничить выборку значений в условии отбора одного запроса данными другого запроса. Второй запрос по отношению к первому является вложенным, а первый запрос по отношению ко второму является основным или внешним. Причем уровней вложенности запросов друг в друга в общем случае может быть несколько. Например, мы хотим узнать, в каких расходных накладных и каким покупателям продавались товары, перечисленные в составе конкретной приходной накладной.

ВЫБРАТЬ

Расход.Ссылка КАК Документ,

Расход.Покупатель КАК Покупатель

ИЗ

Документ.ПродажаТоваров КАК Расход

ГДЕ

Расход.Товары.Номенклатура В

(

ВЫБРАТЬ

ПриходСостав.Номенклатура КАК Номенклатура

ИЗ

Документ.ПоступлениеТоваров.Товары КАК ПриходСостав  
 ГДЕ  
 ПриходСостав.Ссылка = &Документ  
 )

#### **Пример №4.14. Объединение**

При выборке данных с помощью запросов бывает необходимо дополнить данные, получившиеся в результате выполнения одного запроса, данными другого запроса. Например, можно дополнить информацию из документов ПоступлениеТоваров со структурой (Контрагент, Товар, Количество, Цена) данными из ПродажаТоваров с такой же структурой.

Для этого в языке запросов существует возможность объединения нескольких запросов. При этом записи, полученные с помощью каждого из объединяемых запросов, собираются в один результат запроса. При объединении каждый запрос получает данные независимо, то есть у каждого из запросов – свое описание выбираемых полей (ВЫБРАТЬ), источников запроса (ИЗ), условий отбора (ГДЕ), полей группировки (СГРУППИРОВАТЬ ПО). Затем данные, получаемые в результате каждого запроса, объединяются, и уже над этим объединением выполняются такие операции, как упорядочивание результатов (УПОРЯДОЧИТЬ ПО) и расчет итогов (ИТОГИ ПО).

Для объединения запросов используется предложение ОБЪЕДИНИТЬ, которое располагается в секции Объединение запросов текста запроса.

**ВЫБРАТЬ**

ПродажаТоваровТовары.Ссылка.Контрагент КАК Контрагент,  
 ПродажаТоваровТовары.Номенклатура КАК Номенклатура,  
 СУММА(ПродажаТоваровТовары.Количество) КАК Заказано,  
 СУММА(0) КАК Продано

**ИЗ**

Документ.ПродажаТоваров.Товары КАК ПродажаТоваровТовары  
 СГРУППИРОВАТЬ ПО

ПродажаТоваровТовары.Ссылка.Контрагент,  
 ПродажаТоваровТовары.Номенклатура

**ОБЪЕДИНИТЬ ВСЕ**

**ВЫБРАТЬ**

ПродажаТоваровТовары.Ссылка.Контрагент,  
 ПродажаТоваровТовары.Номенклатура,  
 СУММА(0),  
 СУММА(ПродажаТоваровТовары.Количество)

**ИЗ**

Документ.ПродажаТоваров.Товары КАК ПродажаТоваровТовары  
 СГРУППИРОВАТЬ ПО

ПродажаТоваровТовары.Ссылка.Контрагент,  
 ПродажаТоваровТовары.Номенклатура

## ИТОГИ ПО ОБЩИЕ, Клиент

Названия полей результата запроса описываются в списке полей выборки первого из объединяемых запросов. Поля выборки второго запроса сопоставляются с полями результата в соответствии с порядком их следования в списке полей выборки.

Поэтому объединяемые запросы должны иметь одинаковое количество полей в списке полей выборки. В случае, если поля выборки объединяемых запросов имеют разный тип, поля результата запроса будут иметь составной тип.

По умолчанию при объединении запросов полностью одинаковые строки в результате запроса, сформированные разными запросами, заменяются одной. Если требуется, чтобы были оставлены разные строки, необходимо указать ключевое слово **ОБЪЕДИНИТЬ ВСЕ**.

### ***Пример №4.15. Временные таблицы и пакетные запросы***

Бывают ситуации, когда в качестве источника запроса необходимо использовать не таблицы базы данных, а данные, полученные в результате выполнения другого запроса. Например, источником запроса могут служить данные документов ПоступлениеТоваров, сгруппированные по номенклатуре или отобранные по дате документа.

**ВЫБРАТЬ**

Товары.Наименование,  
ПоступлениеТоваров.Дата,  
ПоступлениеТоваров.Контрагент

**ИЗ**

Справочник.Номенклатура КАК Товары  
**ЛЕВОЕ СОЕДИНЕНИЕ**

**(ВЫБРАТЬ**

Поступление.Товар,  
Поступление.Ссылка.Дата,  
Поступление.Ссылка.Контрагент

**ИЗ**

Документ.ПоступлениеТоваров.Товары КАК Поступление

**ГДЕ**

МЕСЯЦ(Поступление.Ссылка.Дата) = 11

) КАК ПоступлениеТоваров

ПО Товары.Ссылка = ПоступлениеТоваров.Номенклатура

**УПОРЯДОЧИТЬ ПО**

Товары.Наименование ИЕРАРХИЯ

Разобьем запрос, содержащий вложенный запрос, на два запроса и объединим их в один пакетный запрос.

**ВЫБРАТЬ**

Поступление.Номенклатура,  
 Поступление.Ссылка.Дата КАК Дата,  
 Поступление.Ссылка.Контрагент

**ПОМЕСТИТЬ** ПоступлениеТоваров  
**ИЗ**

Документ.ПоступлениеТоваров.Товары КАК Поступление  
**ГДЕ**

МЕСЯЦ(Поступление.Ссылка.Дата) = 11

;

**ВЫБРАТЬ**

Товары.Наименование,  
 ПоступлениеТоваров.Дата,  
 ПоступлениеТоваров.Контрагент

**ИЗ**

Справочник.Номенклатура КАК Товары

**ЛЕВОЕ СОЕДИНЕНИЕ** ПоступлениеТоваров КАК ПоступлениеТоваров

**ПО** Товары.Ссылка = ПоступлениеТоваров.Номенклатура

**УПОРЯДОЧИТЬ ПО**

Товары.Наименование **ИЕРАРХИЯ**

В первом запросе мы помещаем данные о поступлении товаров за ноябрь во временную таблицу с именем ПоступлениеТоваров. Затем следует символ «;» (точка с запятой), который указывает на то, что это – пакетный запрос. В следующем запросе справочник товаров связывается левым соединением с временной таблицей по ссылкам товаров, и в итоге мы получаем результат, аналогичный результату при выполнении запроса, содержащего вложенный запрос. Использование временных таблиц вместо вложенных запросов почти всегда делает запрос более оптимальным.

***Пример №4.15. Виды соединений.***

Часто в результате запроса требуется вывести данные из разных таблиц, связав их по значению некоторого поля, то есть какие-то поля вывести из одной таблицы, какие-то – из другой для одного и того же значения поля из этих таблиц.

В этом случае используются несколько источников данных, которые перечисляются после ключевого слова **ИЗ**. В качестве источников запроса могут выступать реальные и виртуальные таблицы, а также вложенные запросы.

Исходные таблицы запроса обычно связываются (соединяются) между собой по некоторому условию – условию связи. Поле, по которому производится связь, обычно имеет ссылочный тип. При соединении данных из исходных таблиц запроса для каждой записи из этих таблиц проверяется условие равенства значений ссылочных полей этого типа. Условие связи источников запроса задается в предложении **ИЗ**, после ключевого слова **ПО**. Например, **ПО** Номенклатура.Ссылка = ЦеныПоставщиков.Номенклатура.

*Внутреннее соединение.* При внутреннем соединении таблиц в результате запроса попадут только те записи из таблиц-источников, которые удовлетворяют заданному условию связи (после ключевого слова ПО).

ВЫБРАТЬ

Товары.Код,  
Товары.Наименование,  
Цены.Цена

ИЗ

Справочник.Номенклатура КАК Товары  
ВНУТРЕННЕЕ СОЕДИНЕНИЕ  
РегистрСведений.ЦеныПоставщиков.СрезПоследних КАК Цены  
ПО Товары.Ссылка = Цены.Номенклатура

*Левое внешнее соединение.* Если связать таблицы регистров накопления левым соединением, то в результате запроса попадут записи из обеих таблиц, удовлетворяющие условию связи, и, кроме того, записи из первой таблицы, расположенной слева от ключевого слова СОЕДИНЕНИЕ, для которых не найдено соответствия во второй таблице [4].

*Правое внешнее соединение.* При правом соединении таблиц ситуация зеркально противоположная. То есть из правой таблицы в результате запроса попадут все записи, а из левой – только те, которые удовлетворяют условию связи.

*Полное соединение.* При полном соединении таблиц в результате запроса будут включены записи из обеих исходных таблиц, которые соответствуют указанному условию. Кроме того, в результате запроса будут включены также еще и те записи из обоих источников, для которых не найдено соответствий.

### ***Работа с запросами во встроенном языке***

Для формирования и выполнения запроса, а также для получения и обработки его результатов во встроенном языке предназначены следующие программные объекты [4]:

- Запрос,
- РезультатЗапроса,
- ВыборкаИзРезультатаЗапроса.

Вся работа с запросами выполняется только на сервере.

*Создание запроса.* Заготовка для создания запроса:

Запрос = Новый Запрос;

Запрос.Текст = "";

*Установка параметров:*

Запрос.УстановитьПараметр("ИмяПараметра", Значение);

*Получение выборки из результата.*

РезультатЗапроса = Запрос.Выполнить();

*Обход выборки из результата запроса*

Для обхода выборки из результата запроса нужно организовать цикл, в котором перебираются элементы коллекции данных, содержащихся в объекте ВыборкаИзРезультатаЗапроса. Для этого используется метод выборки

Следующий(), который позволяет перейти к следующей записи результата запроса в соответствии с порядком обхода выборки (об этом будет рассказано ниже). Этот метод вызывается в цикле Пока Выборка.Следующий() ... Цикл до тех пор, пока не будет получено значение Ложь. При первом проходе цикла метод Следующий() позиционирует выборку на первую запись.

Существует три вида обхода выборки из этого результата запроса: линейный, иерархический, обход по группировкам.

## Тема 5. Методы и инструментальные средства бизнес-аналитики. Разработка сложных аналитических отчетов с помощью СКД

Изучение некоторых возможностей СКД и нюансов разработки аналитических отчетов будем выполнять на нескольких практических примерах [5].

**Пример №5.1.** Создание отчета, отражающего объем продаж по контрагентам с детализацией за период.

Для выполнения задания в конфигурации должен присутствовать регистр накопления оборотов Продажи.

Создадим новый отчет «ОтчетПоПродажам». Откроем конструктор схемы компоновки данных и добавим новый набор данных – запрос. Откроем конструктор запроса. В качестве источника данных для запроса выберем виртуальную таблицу регистра накопления Продажи.Обороты. Вызовем диалог параметров виртуальной таблицы (кнопка Параметры виртуальной таблицы). В параметрах установим Периодичность равной значению Авто.

В список полей перенесем поля из таблицы Продажи.Обороты: «ПериодДень», «ПериодМесяц», «Контрагент», «КоличествоОборот», «СуммаОборот».

На закладке Объединения/Псевдонимы изменим имя у полей КоличествоОборот на Количество, СтоимостьОборот на Стоимость.

Выйдем из конструктора запросов, нажав ОК, вернемся в конструктор схемы компоновки данных. Оставим включенным свойство Атозаполнение. на основании запроса платформа заполнила доступные поля набора данных и для полей, являющихся измерениями регистра, задала вид роли Измерение, а для полей, связанных с детализацией по периодам, – вид роли Период.

Для поля ПериодДень зададим формат поля, чтобы дата отображалась без времени. Выполним это в колонке Оформление. В параметре Формат зададим (вручную или воспользовавшись конструктором форматной строки) следующий формат: ДФ=dd.ММ.уууу. Аналогичные действия выполним для поля ПериодМесяц, но для него введем другой формат: ДФ=ММММ.уууу.

Выражения для представления полей. Система компоновки данных позволяет определять выражения для представления полей в отчете (Переопределение представления полей). Например, мы хотим отображать код контрагента в скобках слева от его наименования. Для поля Контрагент в колонку Выражение представления введем следующий текст:

"("+Контрагент.Код+") "+Контрагент.Наименование

На закладке Ресурсы определим доступные поля ресурсов отчета: Количество и Стоимость.

Для указания периода отчета дадим возможность пользователю использовать стандартный период. На закладке Параметры добавим параметр с именем Период типа СтандартныйПериод, а для параметров НачалоПериода и КонецПериода укажем Выражение для расчета и запретим их редактирование пользователем (рис. 5.1).



Определим структуру отчета. На закладке «Настройки» выделим корневой элемент Отчет и добавим в него группировку по полю Контрагент. В ней добавим вложенную в нее группировку ПериодМесяц, в которую добавим еще одну вложенную группировку ПериодДень.

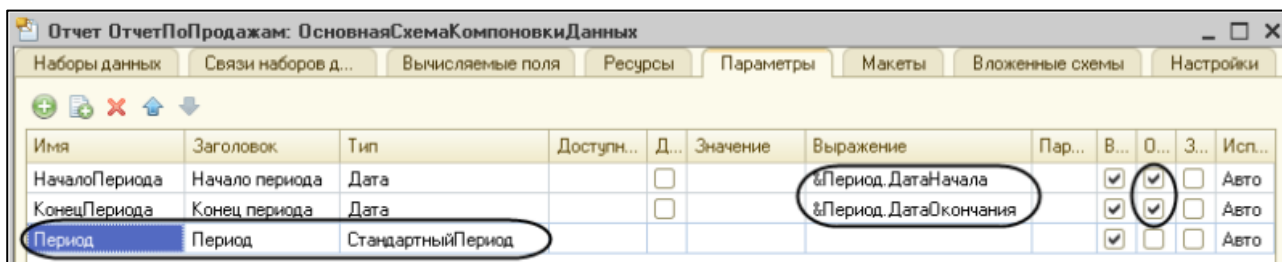


Рис. 5.1. Задание параметров отчета

Стоя на корне Отчет, на закладке Выбранные поля перенесем из списка доступных полей следующие поля: «Сумма», «Сумма.ПроцентВГруппе», «Сумма.ПроцентОбщий».

На закладке Параметры установим значение параметра Период – Прошлый месяц и включим этот параметр в состав быстрых пользовательских настроек.

На закладке Другие настройки установим макет оформления отчета (Макеты оформления), например, «Море».

Отчет готов. Запустим режим исполнения для проверки разработанного отчета. Внешний вид сформированного отчета представлен на рис. 5.2.

Контрагент	Сумма Оборот	Сумма Оборот.% в группировке	Сумма Оборот.% общий
(000000010) ООО "БыстраяПицца"	2 000,00	15,38	15,38
01.11.2021 0:00:00	2 000,00	100,00	15,38
10.11.2021	2 000,00	100,00	15,38
(000000011) ООО "Василиса"	11 000,00	84,62	84,62
01.11.2021 0:00:00	11 000,00	100,00	84,62
01.11.2021	11 000,00	100,00	84,62
<b>Итого</b>	<b>13 000,00</b>	<b>100,00</b>	<b>100,00</b>

Рис. 5.2. Сформированный отчет «ОтчетПоПродажам»

Как видно из рис. 5.2, представление поля «Контрагент» и формат полей периодов соответствуют заданным. В отчет выводятся системные поля, отражающие для каждой группировки процент от общей стоимости продаж и

процент от стоимости продаж в группе. Есть возможность задания периода отчета с помощью стандартного периода.

**Пример №5.2.** Создадим отчет, отражающий список номенклатуры с остатками на складах, и реализуем его программный запуск из списка номенклатуры.

Создадим новый отчет «СписокНоменклатуры». Откроем конструктор схемы компоновки данных и добавим новый набор данных – запрос. Откроем конструктор запроса. В качестве источника данных для запроса выберем два источника: объектную ссылочную таблицу Номенклатура и виртуальную таблицу регистра накопления ОстаткиНоменклатуры.Остатки. Чтобы исключить неоднозначность имен в запросе, переименуем таблицу Номенклатура в спрНоменклатура. Для этого выделим ее в списке Таблицы и вызовем ее контекстное меню, выберем пункт «Переименовать таблицу...». В список полей перенесем поля из этих таблиц: «спрНоменклатура.Ссылка», «спрНоменклатура.Родитель», «спрНоменклатура.ВидНоменклатуры», «ОстаткиНоменклатурыОстатки.Склад», «ОстаткиНоменклатурыОстатки.КоличествоОстаток».

Поскольку в запросе более одного источника, то свяжем их на закладке Связи. По умолчанию платформой уже будет создана связь по полю Номенклатура. Но нам нужно снять флажок Все у таблицы ОстаткиНоменклатуры.Остатки и установить его у таблицы спрНоменклатура. Тем самым мы задаем тип связи как Левое соединение, то есть в результат запроса будут включены записи обеих таблиц, удовлетворяющих условию связи по полю Номенклатура. Кроме того, у таблицы спрНоменклатура, которая будет слева в тексте запроса, будут отражаться в отчете все записи, независимо от того, найдено ли им соответствие в таблице справа ОстаткиНоменклатуры.Остатки или нет. Это нужно потому, что в регистре ОстаткиНоменклатуры отражаются записи только по материалам, а отчет выводится по всем видам номенклатуры.

На закладке Условия установим отбор следующего вида:

спрНоменклатура.ЭтоГруппа = ЛОЖЬ

На закладке Объединения/Псевдонимы изменим имена у полей запроса на более понятные: Ссылка на Номенклатура; Родитель на Группа; ВидНоменклатуры на Вид; КоличествоОстаток на Остаток.

Выйдем из конструктора запросов, нажав кнопку ОК, вернемся в конструктор схемы компоновки данных. Будет сформирован текст запроса:

**ВЫБРАТЬ**

спрНоменклатура.Ссылка КАК Номенклатура,  
спрНоменклатура.Родитель КАК Группа,  
спрНоменклатура.ВидНоменклатуры КАК Вид,  
ОстаткиНоменклатурыОстатки.Склад КАК Склад,  
ОстаткиНоменклатурыОстатки.КоличествоОстаток КАК Остаток

**ИЗ**

Справочник.Номенклатура КАК спрНоменклатура

## ЛЕВОЕ СОЕДИНЕНИЕ

РегистрНакопления.ОстаткиНоменклатуры.Остатки КАК

ОстаткиНоменклатурыОстатки

ПО ОстаткиНоменклатурыОстатки.Номенклатура =

спрНоменклатура.Ссылка

ГДЕ

спрНоменклатура.ЭтоГруппа = ЛОЖЬ

На закладке Настройки выделим корневой элемент Отчет и добавим в него группировку по полю Группа. Выделим группировку Группа и добавим подчиненную ей группировку, не указывая группировочное поле, – Детальные записи. Затем на закладке Выбранные поля перенесем из списка доступных полей поля: «Номенклатура», «Вид», «Склад», «Остаток». На закладке Другие настройки зададим макет оформления Яркий.

Запустим режим исполнения, сформируем отчет (рис. 5.3).

Список номенклатуры				
Сформировать				
Выбрать вариант...				
Настройки...				
Группа	Номенклатура	Вид	Склад	Остаток
<b>Канцелярские Товары</b>				
	Бумага для принтера А4	Товар	Основной	120
	Ножницы	Товар	Основной	56
	Маркер перманентный черный	Товар	Основной	124
	Скрепки канцелярские	Товар	Основной	100
	Степлер	Товар	Основной	320
	Корректирующая жидкость	Товар	Основной	98
	Клей-карандаш	Товар	Основной	100
	Файлы для документов А4 формата	Товар	Основной	155
	Антистеплер	Товар	Основной	140
	Скоросшиватель	Товар	Основной	200
	Дырокол	Товар	Основной	100
<b>Материалы</b>				
	Подводка электрическая	Материал	Основной	200
	Вентиль водоснабжения	Материал	Основной	200

Рис. 5.3. Сформированный отчет «СписокНоменклатуры»

Реализуем вызов нашего отчета из формы списка справочника Номенклатура. Если еще не была создана ранее, то создадим для справочника Номенклатура форму списка. Создадим команду Печать и соответствующую ей кнопку формы, по нажатию которой содержимое отчета «СписокНоменклатуры» будет выводиться в табличный документ и показываться пользователю. В обработчике команды напишем следующий код:

```
&НаКлиенте
```

```
Процедура Печать(Команда)
```

```
    Результат = ВывестиСписокНоменклатуры();
```

```
    Результат.ОтображатьСетку = Ложь;
```

Результат.ОтображатьЗаголовки = Ложь;

Результат.Показать());

КонецПроцедуры

Отчет формируется в серверной внеконтекстной функции ВывестиСписокНоменклатуры(). Затем заполненный данными табличный документ возвращается на клиента и показывается пользователю.

&НаСервереБезКонтекста

Функция ВывестиСписокНоменклатуры()

// В качестве схемы компоновки данных используем

// макет отчета "Список Номенклатуры".

СхемаКомпоновкиДанных =

Отчеты.СписокНоменклатуры.ПолучитьМакет("ОсновнаяСхемаКомпоновкиДанных");

// Получим настройки по умолчанию.

Настройки = СхемаКомпоновкиДанных.НастройкиПоУмолчанию;

// Сформируем макет компоновки данных.

КомпоновщикМакета = Новый КомпоновщикМакетаКомпоновкиДанных;

МакетКомпоновки =

КомпоновщикМакета.Выполнить(СхемаКомпоновкиДанных,Настройки);

// Выполним компоновку данных.

ПроцессорКомпоновкиДанных = Новый ПроцессорКомпоновкиДанных;

ПроцессорКомпоновкиДанных.Инициализировать(МакетКомпоновки);

// Выведем результат компоновки в табличный документ.

ДокументРезультат = Новый ТабличныйДокумент;

ПроцессорВывода = Новый

ПроцессорВыводаРезультатаКомпоновкиДанныхВТабличныйДокумент;

ПроцессорВывода.УстановитьДокумент(ДокументРезультат);

ПроцессорВывода.Вывести(ПроцессорКомпоновкиДанных);

Возврат ДокументРезультат;

КонецФункции

В серверной функции программно реализуются все этапы компоновки данных.

Запустим режим исполнения. Откроем форму списка справочника Номенклатура. На панели присутствует кнопка «Печать», нажатие которой программно открывает наш отчет «СписокНоменклатуры».

**Пример №5.3.** Разработаем отчет, отражающий движение и остатки номенклатуры на складах. Будем использовать вывод в диаграмму.

Создадим новый отчет «ОстаткиНоменклатурыПоПериодам».

Откроем конструктор схемы компоновки данных и добавим новый набор данных – запрос. Источником данных для запроса будет служить виртуальная таблица регистра накопления ОстаткиНоменклатуры.ОстаткиИОбороты.

Сформируем конструктором запроса следующий запрос:

**ВЫБРАТЬ**

ОстаткиНоменклатурыОстаткиИОбороты.Номенклатура,

ОстаткиНоменклатурыОстаткиИОбороты.Склад,  
ОстаткиНоменклатурыОстаткиИОбороты.КоличествоНачальныйОстаток  
КАК  
НачальныйОстаток,  
ОстаткиНоменклатурыОстаткиИОбороты.КоличествоКонечныйОстаток КАК  
КонечныйОстаток,  
ОстаткиНоменклатурыОстаткиИОбороты.КоличествоПриход КАК Приход,  
ОстаткиНоменклатурыОстаткиИОбороты.КоличествоРасход КАК Расход,  
ОстаткиНоменклатурыОстаткиИОбороты.Регистратор,  
ОстаткиНоменклатурыОстаткиИОбороты.ПериодСекунда  
ИЗ  
РегистрНакопления.ОстаткиНоменклатуры.ОстаткиИОбороты(, , Авто, , )  
КАК  
ОстаткиНоменклатурыОстаткиИОбороты

На закладке Ресурсы конструктора схемы компоновки данных зададим поля ресурсов: «НачальныйОстаток», «КонечныйОстаток», «Приход», «Расход».

На закладке Параметры зададим настройки параметров, как и в предыдущих примерах. Для параметров НачалоПериода и КонецПериода в колонке Тип нажмем кнопку выбора и в нижней части окна редактирования типа данных установим Состав даты в значение Дата. Для параметра КонецПериода зададим значение колонки Выражение, чтобы данные за последний день отчета были включены в расчет.

Определим структуру отчета. На закладке Настройки выделим корневой элемент Отчет и добавим в него группировку по полю Склад. Затем добавим вложенную в нее группировку по полю Номенклатура. В группировку Номенклатура вложим группировку Детальные записи.

На закладке Выбранные поля перенесем из списка доступных полей поля: «Регистратор», «НачальныйОстаток», «Приход», «Расход», «КонечныйОстаток».

На закладке Другие настройки установим макет оформления (Макеты оформления) отчета – Зеленый.

На закладке Параметры включим параметры НачалоПериода и КонецПериода в состав быстрых пользовательских настроек.

Сформируем отчет в пользовательском режиме «1С:Предприятие» и убедимся в его работоспособности.

Создадим еще один вариант отчета с выводом в диаграмму, в котором попробуем возможность платформы – вывод в диаграмму нескольких ресурсов и возможность изменения их положения.

На закладке Настройки создадим новый вариант отчета «ОстаткиНоменклатурыПоПериодам» с именем «ДвижениеНоменклатуры».

Выделим корневой элемент Отчет, нажмем кнопку Добавить в командной панели и добавим диаграмму. Затем выделим ветку Серии и добавим в нее группировку по полю Склад, а в ветку Точки – группировку Номенклатура. На

закладке Выбранные поля перенесем из списка доступных полей поля ресурсов Приход и Расход.

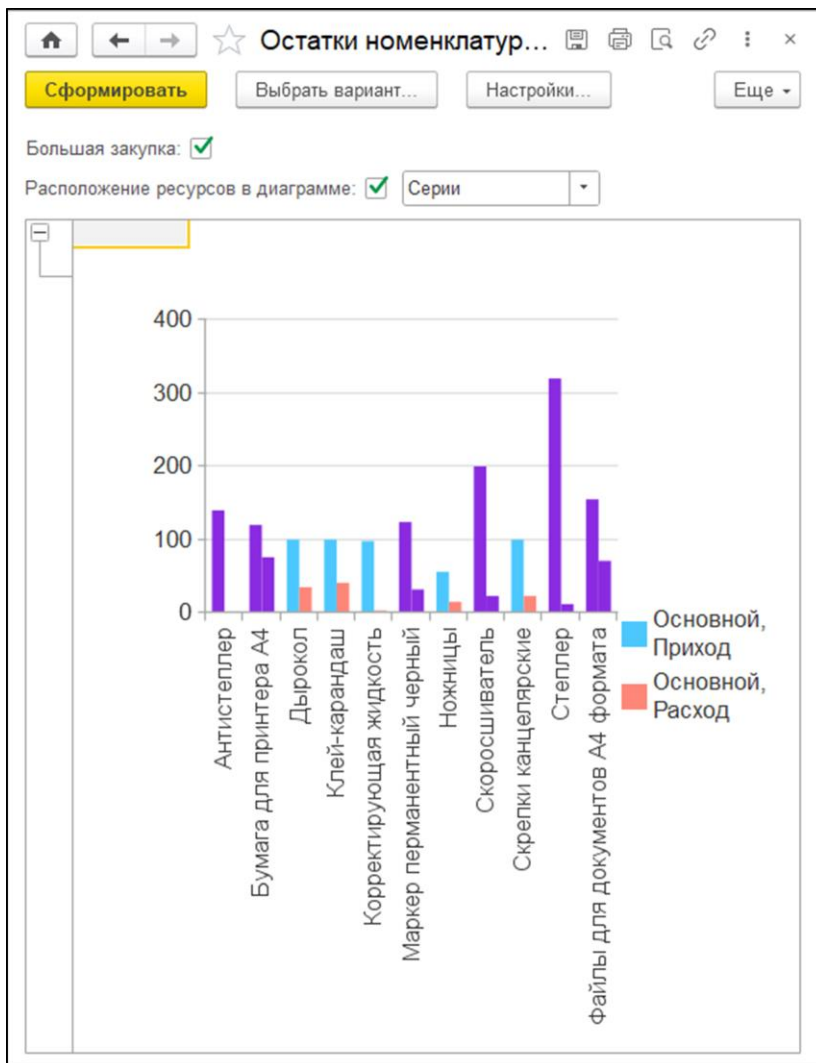
На закладке Другие настройки установим параметр Расположение ресурсов в диаграмме в значение Серии и включим эту настройку в состав быстрых пользовательских настроек.

Настроим условное оформление диаграммы. На закладке Условное оформление установим вертикальную ориентацию (90) текста для точек диаграммы (поле Номенклатура). Зададим представление этой настройки как Вертикально.

Создадим еще один элемент условного оформления диаграммы. Установим фиолетовый цвет для точек диаграммы (поле Номенклатура), в которых значение поля Приход больше 8. Зададим представление этой настройки как Большая закупка.

Включим обе эти настройки в состав быстрых пользовательских настроек.

Запустим отчет на исполнение в режиме 1С:Предприятие. В открывшемся окне отчета нажмем кнопку «Выбрать вариант...». Выберем вариант «ДвижениеНоменклатуры» и сформируем отчет. Мы видим, что данные о приходе/расходе номенклатуры по складам выводятся в диаграмму (тип диаграммы по умолчанию – Гистограмма объемная), в точках которой данные группируются по номенклатуре, а в сериях – по складам. Текст в точках диаграммы (наименования номенклатуры) располагается вертикально, столбцы номенклатуры с большим приходом выделяются фиолетовым цветом. Ресурсы диаграммы расположены в сериях, в легенде две серии, два ресурса (Приход, Расход). Все перечисленные настройки расположены



**Рис. 5.4. Вариант отчета «ДвиженияНоменклатуры»**

непосредственно в отчетной форме, поэтому удобно ими манипулировать и сразу видеть результат отчета.

**Задача №5.4.** Создадим отчет, который будет показывать пользователю информацию с промежуточными итогами по группам номенклатуры, созданным в иерархическом справочнике.

Создадим новый объект конфигурации Отчет и назовем его ПродажиНоменклатуры. Откроем конструктор схемы компоновки данных и добавим новый набор данных – запрос. Откроем конструктор запроса.

В качестве источника данных для запроса выберем виртуальную таблицу регистра накопления Продажи.Обороты. В параметрах виртуальной таблицы зададим Периодичность равной День.

В список полей перенесем поля из таблицы Продажи.Обороты: «Период», «Номенклатура», «Контрагент», «КоличествоОборот», «СуммаОборот».

На закладке Объединения/Псевдонимы изменим имя у поля КоличествоОборот на Остаток и СуммаОборот на Стоимость.

Выйдем из конструктора запроса, нажав ОК. В конструкторе схемы компоновки данных на закладке Ресурсы в качестве полей ресурсов определим поля: «Количество» и «Сумма». Для ресурса Количество в колонке Рассчитывать по нажмем кнопку выбора и укажем поле Номенклатура и НоменклатураИерархия. Это сделано для того, чтобы итоги по количеству выводились для конкретной номенклатуры и ее иерархических группировок.

На закладке Параметры Для параметров НачалоПериода и КонецПериода в колонке Тип нажмем кнопку выбора и в нижней части окна редактирования типа данных установим Состав даты в значение Дата. Для параметра КонецПериода зададим значение колонки Выражение, чтобы данные за последний день отчета были включены в расчет:

КонецПериода(&КонецПериода,"День")

На закладке Настройки создадим три вложенные группировки соответственно по полям: «Контрагент», «Номенклатура», «Период». На закладке Выбранные поля перенесем из списка доступных полей поля Количество, Сумма.

На закладке Наборы данных в списке полей набора данных в колонке Оформление зададим формат поля Количество ЧЦ=15. Аналогично зададим формат поля Период: ДЛФ=DD.

Предоставим пользователю возможность задавать отчетный период перед формированием отчета. Включим параметры Начало периода и Конец периода в состав быстрых пользовательских настроек. То есть эти настройки будут находиться непосредственно в отчетной форме. На закладке Параметры выделим каждый из параметров и нажмем кнопку Свойства элемента пользовательских настроек, расположенную в правом нижнем углу окна настроек. Установим флажок «Включать в пользовательские настройки» и оставим предложенное по умолчанию для свойства Режим редактирования значение «Быстрый доступ».



Предоставим пользователю возможность задавать отбор и условное оформление отчета. Откроем закладку Настройки. На закладке Отбор установим отбор записей отчета по определенному виду номенклатуры. Для этого в списке доступных полей раскроем поле Номенклатура и выберем поле ВидНоменклатуры. Остальные параметры отбора оставим по умолчанию – Вид сравнения – Равно, а Правое значение – Перечисление.ВидыНоменклатуры.ПустаяСсылка. Включим настройку отбора в состав быстрых пользовательских настроек. Для этого нажмем кнопку «Свойства элемента пользовательских настроек», расположенную в правом нижнем углу окна настроек, и установим флажок «Включать в пользовательские настройки».

На закладке Условное оформление выделим цветом записи отчета со стоимостью продаж меньше 3000 руб. Нажмем кнопку Добавить справа на командной панели окна. Выберем Оформление для выделения полей. В нашем случае – это красный цвет текста. Зададим Отбор – условие для выделения полей. Зададим список оформляемых полей, в нашем случае – поле Стоимость, то есть укажем, какие поля требуется выделить. Обозначим Представление условного оформления как Мелкий заказ. Включим настройку условного оформления в состав быстрых пользовательских настроек.

Запустим отчет на исполнение в режиме «1С:Предприятие».

В форме отчета, под настройками отчетного периода, добавились настройки отбора и условного оформления, которые пользователь может включить или выключить. В настройке отбора пользователь может указать критерий отбора записей в отчет, в нашем случае выбрать вид номенклатуры, или выключить использование отбора (рис. 5.5). если пользователь не считает их жизненно важными и не собирается задавать их перед каждым формированием отчета, то он может изменить состав настроек, выполнив команду «Все действия»→»Изменить состав настроек».

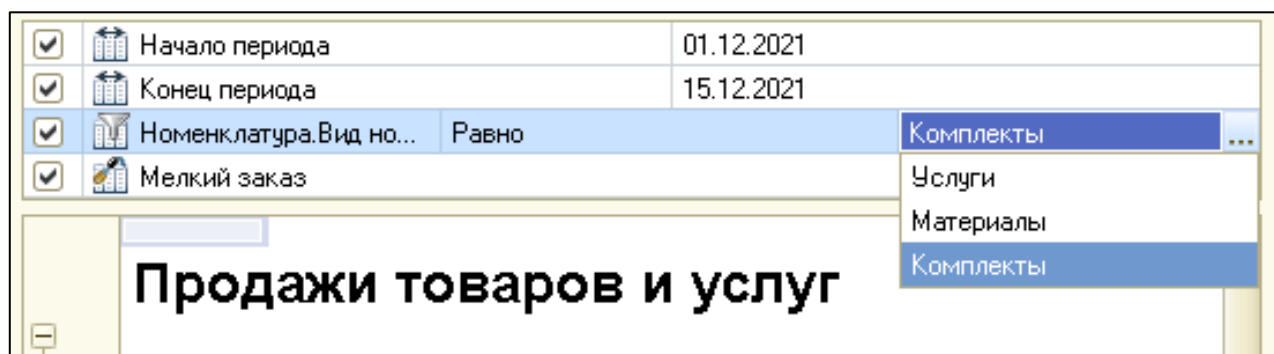


Рис. 5.5. Настройки отбора

Вернемся в конфигуратор. На закладке Настройки схемы компоновки данных содержатся полные настройки отчета, которые задает разработчик. Часть из них может быть предоставлена пользователю для создания произвольного отбора, условного оформления отчета и пр. Для этого нажмем кнопку Свойства элемента пользовательских настроек, расположенную сверху командной панели окна настроек. В появившемся окне мы можем



редактировать состав пользовательских настроек отчета. Установим признак использования для настроек Отбор и Условное оформление и установим для них свойство Режим редактирования в значение Обычный. Таким образом, мы включили настройки отбора и условного оформления в состав пользовательских настроек и предоставили пользователю возможность задавать их в отдельном окне, вызываемом кнопкой Настройка.

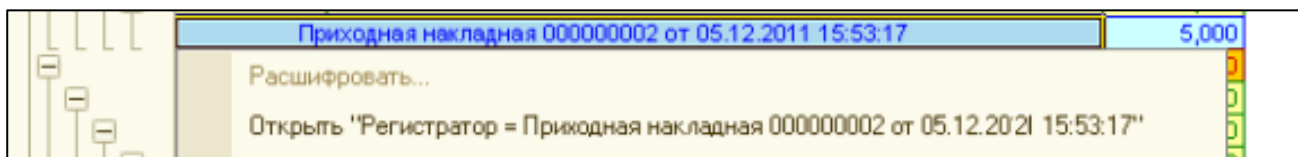
В режиме 1С:Предприятие. Выполним команду «Все действия»→»Изменить вариант...» и выделим в структуре отчета группировку «Номенклатура». На закладке «Группируемые поля» для группировки «Номенклатура» укажем «Тип группировки» – «Иерархия». Нажмем Завершить редактирование и сформируем отчет. Информация в отчете будет представлена с промежуточными итогами по группам справочника Номенклатура.

**Стандартная расшифровка.** Допустим, что пользователь хочет иметь возможность открывать из отчета первичные документы по поступлению и продаже товаров.

Для этого нам нужно детализировать записи в отчете по полю Регистратор, то есть документу, который произвел движения о поступлении товаров в регистре накопления «ОстаткиНоменклатуры». Откроем схему компоновки данных на закладке Наборы данных и вызовем конструктор запроса набора данных Приход. Вызовем диалог параметров виртуальной таблицы «ОстаткиНоменклатуры.Обороты». Для этого в списке таблицы нужно нажать кнопку Параметры виртуальной таблицы. В открывшемся окне параметров зададим значение параметра «Периодичность» равным Регистратор.

В список полей запроса добавим поле Регистратор. Нажмем ОК и вернемся в конструктор схемы компоновки данных. Поле Регистратор появится в списке доступных полей набора данных Приход. Теперь добавим его в структуру отчета. На закладке Настройки выделим группировку Номенклатура (Иерархия) и добавим вложенную в нее группировку Детальные записи без указания группировочного поля. На закладке Выбранные поля перенесем из списка доступных полей поле Регистратор.

Выполним отчет в пользовательском режиме. Записи в нем будут детализированы по документу-регистратору. Двойным щелчком мыши на этом поле мы можем открыть соответствующий документ, например, о поступлении товаров, а правой кнопкой мыши вызвать выпадающее меню расшифровки. Двойным щелчком на поле ресурса (колонок Приход) можно вызвать диалог выбора поля для расшифровки (рис. 5.6).



**Рис. 5.6. Меню расшифровки**

Таким образом, если мы хотим расшифровывать информацию по какому-либо детальному полю, мы должны добавить его в список доступных полей отчета.

**Пример №5.5.** Реализуем отчет, который будет максимально универсальным и будет позволять пользователю в режиме «1С:Предприятие», не перебирая вариант отчета, изменять структуру отчета и его внешний вид, например, поменять строки и колонки таблицы или изменить данные, выводимые в ячейках таблицы.

Создадим отчет по продажам номенклатуры контрагентам по регистру Продажи.

Создадим новый объект конфигурации Отчет и назовем его «ПродажиПоКонтрагентам». Откроем конструктор схемы компоновки данных и добавим новый набор данных – запрос. Откроем конструктор запроса. В качестве источника данных для запроса выберем виртуальную таблицу регистра накопления оборотов Продажи. Запрос отчета:


ВЫБРАТЬ

ПродажиОбороты.Номенклатура КАК Номенклатура,  
ПродажиОбороты.Контрагент КАК Контрагент,  
ПродажиОбороты.КоличествоОборот КАК КоличествоОборот,  
ПродажиОбороты.СуммаОборот КАК СуммаОборот

ИЗ

РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты

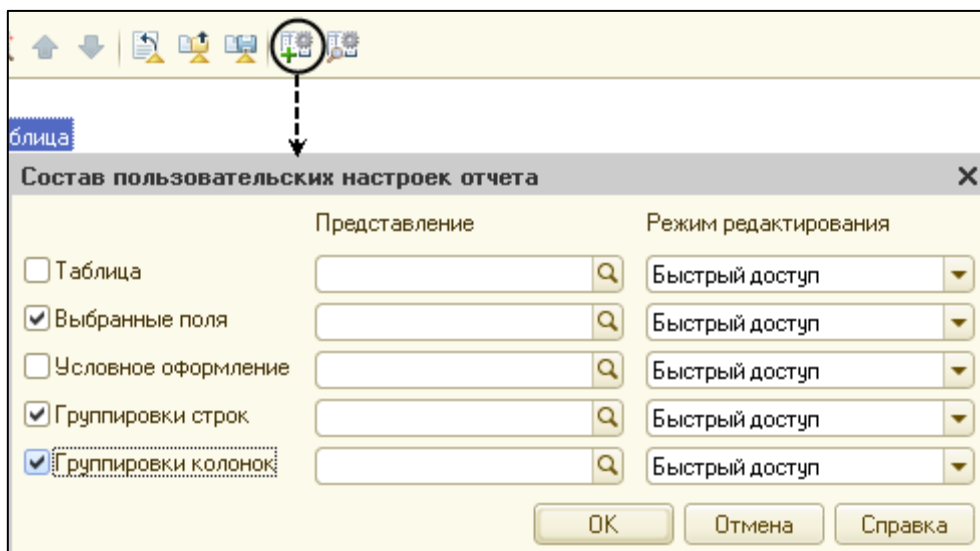
Нажмем ОК и вернемся в конструктор схемы компоновки данных. На закладке «Ресурсы» в качестве полей ресурсов выберем: «Количество» и «Сумма».

Для быстрого создания стандартных настроек воспользуемся конструктором настроек: нажмем кнопку  в командной панели схемы компоновки на закладке Настройки. На первом шаге выберем тип отчета Таблица. На следующем шаге определим поля, выводящиеся в отчет. Перенесем из списка доступных полей все поля. На следующем шаге определим поля для группировки строк и столбцов таблицы. Мышью перетащим из списка доступных полей в поля «Контрагент», а в поля Колонки – поле «Номенклатура». Остальные шаги пропустим и нажмем ОК.

В результате наших действий в окне настроек появилась таблица, в строках которой находится группировка Контрагент, в колонках – группировка Номенклатура, а также определен список выбранных полей. Сделаем наш отчет более гибким, чтобы он выводился в таблицу со свободной структурой.

Для этого в схеме компоновки данных на закладке Настройки снимем признак использования у всех полей отчета. Затем выделим в структуре элементов отчета элемент Таблица и нажмем кнопку Свойства элемента пользовательских настроек, расположенную вверху, в командной панели окна настроек. В появившемся окне мы можем редактировать состав

пользовательских настроек таблицы. Установим признак использования для настроек Выбранные поля, Группировки строк и Группировки колонок и оставим для них по умолчанию свойство Режим редактирования в значении Быстрый доступ (рис. 5.7). Этим мы предоставили пользователю возможность самостоятельно определять состав выбранных полей, группировок строк и колонок таблицы непосредственно в отчетной форме перед формированием отчета.



**Рис. 5.7. Настройка состава пользовательских настроек**

Откроем отчет в пользовательском режиме 1С:Предприятие. Допустим пользователь режим поменять местами колонки и строки. А также вывести в качестве выбранных полей только сумму (рис. 5.8).

Номенклатура	Сумма Оборот	Контрагент
Бумага для принтера А4		2 000,00
Дырокол		2 000,00
Клей-карандаш		2 000,00
Корректирующая жидкость	2 000,00	
Маркер перманентный черный	4 000,00	
Ножницы	10 000,00	
Ручка шариковая	999,00	1 000,00
Скоросшиватель		6 000,00
Скрепки канцелярские		
Степлер		
Файлы для документов А4 формата		4 000,00
Итого	14 000,00	5 998,00 4 000,00 13 000,00 4 000,00

**Рис. 5.8. Изменение структуры отчета в пользовательском режиме**

С помощью конструктора настроек мы вывели данные отчета в виде таблицы и предоставили пользователю возможность самостоятельно определять структуру отчета в соответствии со стоящими перед ним задачами.

Итак, система компоновки данных (СКД) — мощный механизм платформы «1С:Предприятие». Предназначена она для того, чтобы декларативно создавать отчеты. Отчет создается без необходимости писать программный код. С помощью определенного конструктора и настроек, произведенных в нем, программист задает желаемый результат, система компоновки данных это понимает и выводит этот результат пользователю.

## Тема 6. Web-интерфейсы. Предоставление информации через web-интерфейсы

Механизм Web-сервисов позволяет использовать систему «1С:Предприятие» как набор сервисов в сложных распределенных и гетерогенных системах, а также позволяет интегрировать ее с другими промышленными системами с использованием сервисно-ориентированной архитектуры.

Конфигурация системы «1С:Предприятие» может экспортировать свою функциональность через Web-сервисы. Определения Web-сервисов задаются в дереве конфигурации и становятся доступны произвольным информационным системам благодаря публикации их на веб-сервере [1,3].

Для того чтобы функциональность системы «1С:Предприятие» стала доступна внешним потребителям Web-сервисов, нужно выполнить следующие действия:

- создать в конфигурации необходимое количество Web-сервисов,
- опубликовать Web-сервисы с помощью специального инструмента конфигууратора.

**Web-сервис** – это общий объект конфигурации. Он предназначен для того, чтобы описать веб-сервис, поставщиком которого является прикладное решение 1С:Предприятия 8. Создание Web-сервиса заключается:

- в добавлении в дерево метаданных объекта конфигурации Web-сервис,
- описании операций, которые может выполнять создаваемый Web-сервис,
- описании параметров операций Web-сервиса.

Объект конфигурации Web-сервис содержит модуль, в котором создаются процедуры на встроенном языке, выполняемые при вызове тех или иных операций Web-сервиса. Web-сервис описывается набором операций. Каждая операция может иметь некоторое количество параметров для передачи данных Web-сервису и получения возвращаемых значений. Типы параметров операций Web-сервиса описываются с помощью типов XDTO и могут представлять собой либо значения XDTO, либо объекты XDTO.

**XDTO-пакеты** в системе 1С:Предприятие 8 – объект метаданных конфигурации, который помогает получить описание структуры хранения данных в системе. Они реализуют механизм, носящий название XML Data Transfer Objects или, если коротко - XDTO. Аббревиатура XDTO родилась в недрах фирмы 1С и представляет собой ее собственный механизм. Для чего нужен механизм? XDTO - это механизм, разработанный фирмой “1С” для обмена данными с другими программными системами посредством XML, позволяющий на уровне языка 1С оперировать не узлами XML, а прикладными понятиями “Сотрудник”, “Счет” и привычными встроенными типами (“ТаблицаЗначений”, “СправочникСсылка” и т.п.).

**URL-пространство имен.** На свете существует огромное количество программистов, которые создают те или иные XML-документы. При этом, очень часто они оперируют одинаковыми понятиями, например, “Дата”, “Цена” и “Сотрудник”. Если вдруг две системы имен (созданных разными программистами) встретятся в рамках одной информационной системы, то произойдет конфликт имен. Например, Иван создал тип данных “Сотрудник” со свойством “ФИО” и Петя создал объект "Сотрудник" со свойствами “Фамилия”, “Имя”, “Отчество” и “ИНН”. Объекты разные, а имя типа одно, возникает путаница. Чтобы этого избежать, используются пространства имен. Все имена должны быть уникальны в рамках одного пространства имен. Имена в разных пространствах запросто могут повторять друг друга. По традиции (и по многим другим соображениям), пространства именуются в виде URL-подобных строк. Например, “<http://vasya.org/xml/sotrudniki>”. Причем, это не ссылка в сети, это просто строка-идентификатор. Документы пестрят этими “ссылками” и, сталкиваясь с ними, начинающий специалист впадает в ступор - “что это за адреса, что по ним расположено, а что, если нет интернета...” Так вот, это не адреса, это уникальные идентификаторы пространств имен. Строка может быть любой.

Типы данных. Тип данных обязан принадлежать пространству имен. Пространство это задается в схеме XML и однозначно определяет все типы, которые в него входят. Тип данных всегда дополняется пространством имен, в котором этот тип нужно искать [3].

В конфигураторе, каждая схема XML может быть представлена в виде ПакетаXDTO, а вся совокупность типов, имеющихся в конфигурации составляет Модель данных конфигурации. Простые типы XML представляются в виде объектов языка 1С с типом “ЗначениеXDTO”. Составные типы XML представляются в виде объектов языка 1С с типом “ОбъектXDTO”.

Подобно тому, как в конфигурации есть метаданные, предоставляющие информацию о типах, в XDTO тоже есть подобные объекты - ТипЗначенияXDTO и ТипОбъектаXDTO. Эти объекты позволяют узнать информацию о типе данных - размер, число знаков после запятой, неотрицательность и т.п.

Пример №1. Создадим в нашей конфигурации Web-сервис системы «1С:Предприятие», который должен возвращать список контрагентов.

В ветке «Общие» создадим новый web-сервис с именем «WebСервис1».

За закладке Прочие зададим URI пространства имен, например, <http://www.npi.ru/Customer>.

Во многих случаях, можно не создавать свой пакет XDTO, а использовать по умолчанию – выбрав из списка пакет [www.sample-package.org](http://www.sample-package.org).

Мы создадим пакет явно. Новый пакет XDTO.

В свойстве URI пространства имен укажем свое – <http://www.npi.ru/Customer> (рис. 6.1).

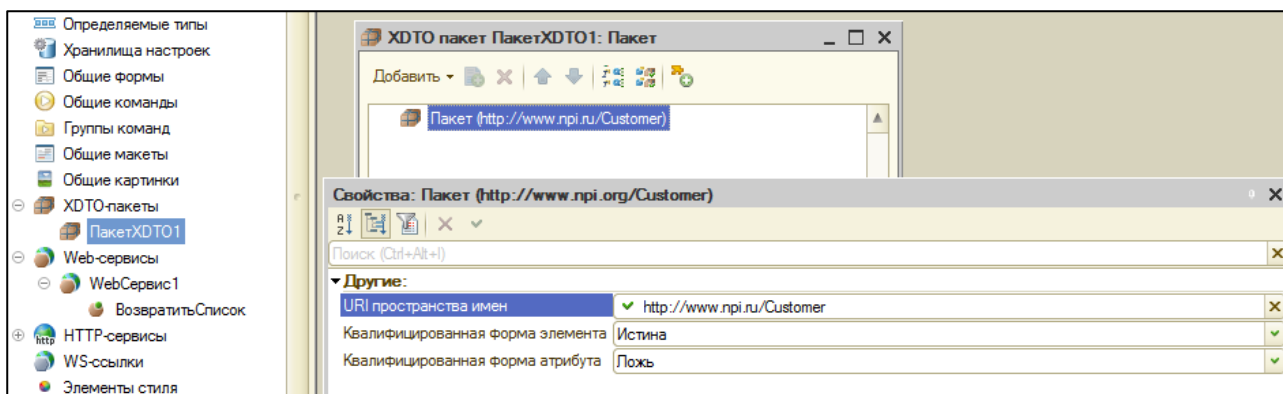


Рис. 6.1. Создание нового XDTO пакета

Поскольку нам потребуется использовать типы объектов, определенные в текущей конфигурации (справочник Контрагенты) из соответствующего стандартного пакета XTDO, то добавим директиву импорта (рис. 6.2).

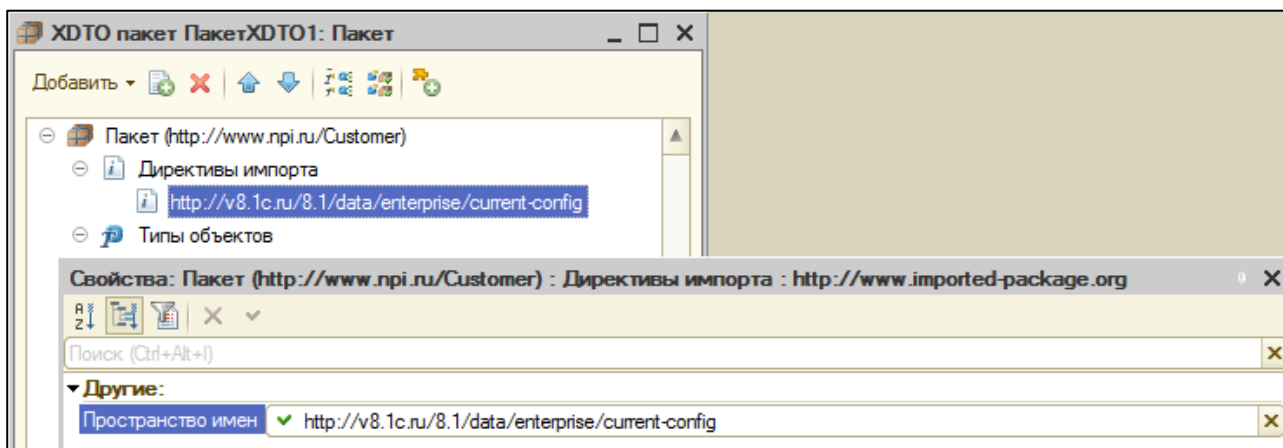


Рис. 6.2. Добавление директивы импорта

Далее добавьте в пакет XDTO Тип объекта. Дадим наименование типу объекта «Список». Добавим новое свойство типа объекта. Назовем его Контрагент.

У свойства укажите "Максимальное количество" равное -1. Это свойство и указывает что Элемент объекта "Список" может быть повторен любое количество раз. Не забудьте указать тип свойства – CatalogObject.Контрагенты.

Проверим созданный пакет на ошибки – для этого щёлкнем правой кнопкой мыши на созданном пакете и выберем из контекстного меню команду «Проверить пакет».

Укажем у web-сервиса используемые пакеты XTDO на закладке «Прочие»

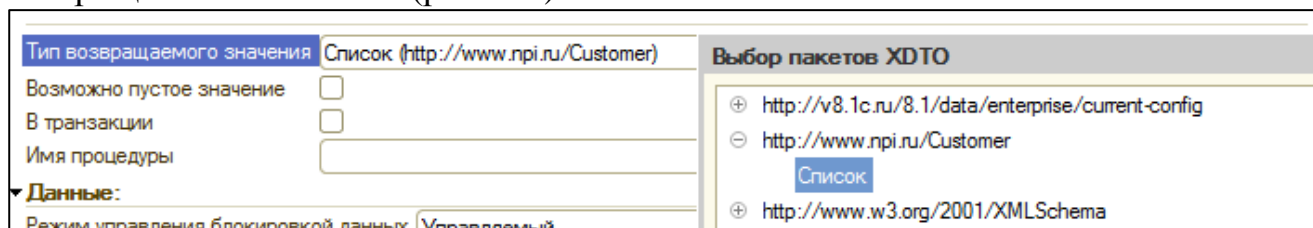
Выберем из списка стандартный пакет:



И наш созданный пакет:



После этого можно у ws-операции "ВозвратитьСписок" указать Тип возвращаемого значения (рис. 6.3).



**Рис. 6.3. Указание у WS-операции типа возвращаемого значения**

Далее следует описать обработчик. Нажимаем в свойстве «Имя процедуры» операции «ВозвратитьСписок» значок лупы справа.

В качестве обработчика укажем следующий код:

Функция ЗагрузитьСписок()

```
КонтрагентыВыборка = Справочники.Контрагента.Выбрать();
```

```
// получить тпы объекта XDTO
```

```
КонтрагентыТип =ФабрикаXDTO.Тип("http://v8.1c.ru/8.1/data/enterprise/
current-config", "CatalogObject.Контрагенты");
```

```
КонтрагентыСписокТип =
```

```
ФабрикаXDTO.Тип("http://www.npi.ru/Customer1", "Список");
```

```
КонтрагентыРезультат =
```

```
ФабрикаXDTO.Создать(КонтрагентыСписокТип);
```

```
Пока Контрагенты.Выборка.Следующий() Цикл
```

```
КонтрагентXDTO = ФабрикаXDTO.Создать(КонтрагентыТип);
```

```
КонтрагентXDTO.Description = КонтрагентыВыборка.Наименование;
```

```
КонтрагентXDTO.Code = КонтрагентыВыборка.Код;
```

```
КонтрагентXDTO.Телефон = КонтрагентыВыборка.Телефон;
```

```
КонтрагентXDTO.ЭлектроннаяПочта =
```

```
КонтрагентыВыборка.ЭлектроннаяПочта;
```

```
КонтрагентXDTO.Ref = КонтрагентыВыборка.Ссылка;
```

```
КонтрагентXDTO.JivoID = КонтрагентыВыборка.JivoID;
```

```
КонтрагентXDTO.DeletionMark = Контрагенты.ПометкаУдаления;
```

```
КонтрагентРезультат.Контрагент.Добавить(КонтрагентXDTO);
```

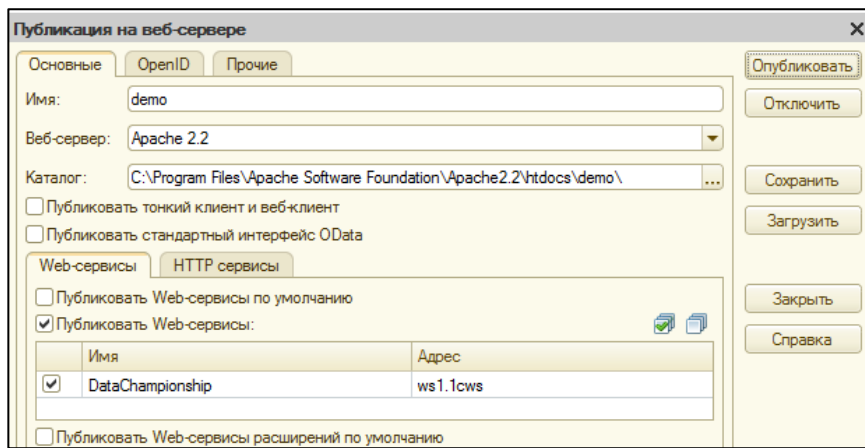
```
КонецЦикла;
```

```
Возврат КонтрагентыРезультат;
```

КонецФункции

Выполним публикацию web-сервиса (рис. 6.4).





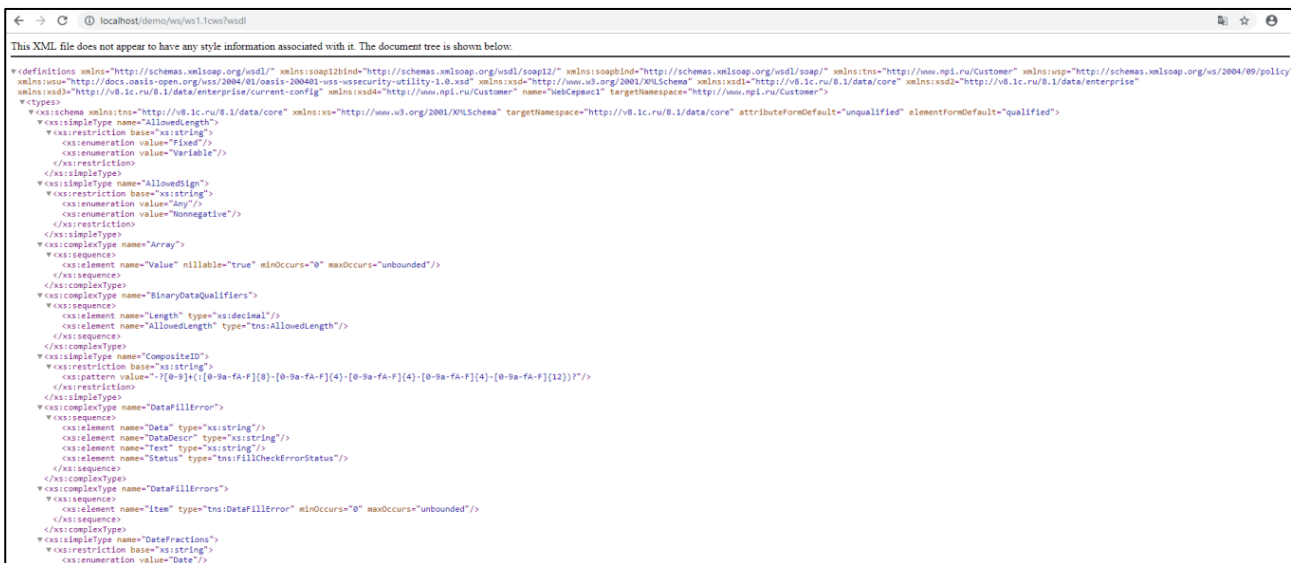
**Рис. 6.4. Публикация web-сервиса**

Если система не предложила перезапустить веб-сервер, то перезапускаем его сами.

Проверим корректность публикации. Для этого в окне браузера наберем адрес:

`http://localhost/demo/ws/ws1.1cws?wsdl`

Если в окне браузера отразится wsdl-описание опубликованного web-сервиса, то публикация произведена (рис. 6.5).



**Рис. 6.5. Wsdl-описание опубликованного web-сервиса**

Теперь в новой второй конфигурации разрабатываем обработку, которая будет, используя опубликованный web-сервис, получать список контрагентов.

Создадим новую обработку. Создадим для нее форму. В ней реализуем новую команду «Загрузить». Вынесем ее на форму.

У новой конфигурации в свойстве «Назначения использования» укажем «Приложение для мобильной платформы».

Примерный код обработчика следующий:

&НаСервере

Процедура ЗагрузитьНаСервере()

Определение =

```

Новый WSOпределения("http://localhost/demo/ws/ws1.1cws?wsdl");
Прокси = Новый WSПрокси(Определение, "http://www.npi.ru/Customer1",
                          "WebСервис1", "WebСервис1Soap");
Отклик = Прокси.ЗагрузитьСписок();
Для Каждого Элемент Из Отклик.Контрагент Цикл
    НовКонтрагентСсылка =
        Справочники.Контрагенты.НайтиПоКоду(Элемент.Code, Истина);
    Если НовКонтрагентСсылка.Пустая() Тогда
        НовКонтрагентОбъект =
            Справочники.Контрагенты.СоздатьЭлемент();
    Иначе
        НовКонтрагентОбъект =
            НовКонтрагентСсылка.ПолучитьОбъект();
    КонецЕсли;
    НовКонтрагентОбъект.Код = Элемент.Code;
    НовКонтрагентОбъект.Наименование = Элемент.Discription;
    НовКонтрагентОбъект.Телефон = Элемент.Телефон;
    НовКонтрагентОбъект.ЭлектроннаяПочта =
        Элемент.ЭлектроннаяПочта;
    НовКонтрагентОбъект.JivoID = Элемент.JivoID;
    НовКонтрагентОбъект.ПометкаУдаления=Элемент.DeletionMark;
    НовКонтрагентОбъект.Записать();
КонецЦикла;
КонецПроцедуры

```

&НаКлиенте

```

Процедура Загрузить(Команда)
    ЗагрузитьНаСервере();
КонецПроцедуры

```

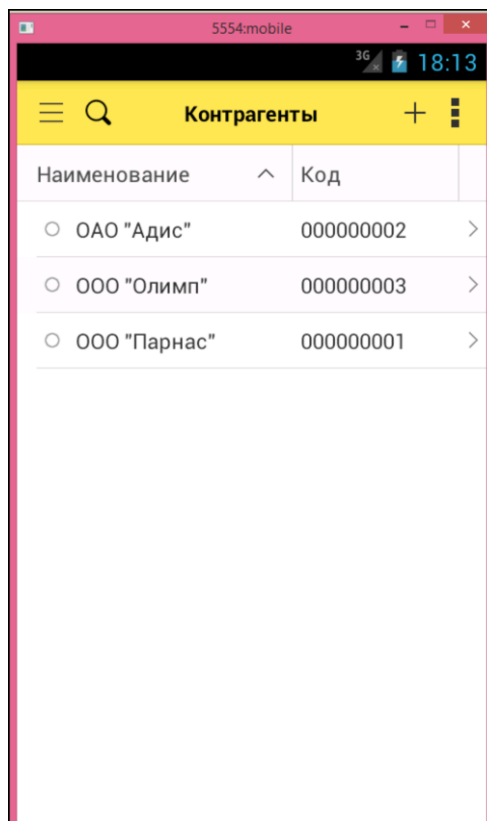
Целесообразно сначала отладить код в режиме запуска как стационарное приложение.

После того как приложение корректно заработало в этом режиме, следует попробовать его запустить как мобильное приложение.

В этом случае в WSOпределения изменить с localhost на 10.0.2.2.

Обязательно следует перед публикацией сохранить информационную базу (обновить конфигурацию БД).

Если при обновлении выдает ошибку блокировки, то следует



остановить веб-сервер, сохранить БД и снова запустить веб-сервер.

Опубликовать мобильное приложение (Конфигурация ->Мобильное приложение-> Опубликовать...).

В качестве каталога следует указать путь к каталогу \htdocs в каталоге с установленным веб-сервером. Например, C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\... + подкаталог публикации, например, Web2MA.

После окончания выполнения загрузки в обработке следует открыть справочник Контрагенты и убедиться, что благодаря обращению к Web-Сервису, опубликованному из первой конфигурации (стационарной), в мобильное приложение перенесутся записи о контрагентах (рис. 6.6).

Для того, чтобы заработала отладка функций веб-сервиса, нужно в файле:

C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\demo\default.vrd

Добавить следующую строку:

```
<debug enable="true" url="tcp://localhost" />
```

Должно получиться примерно так:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<point xmlns="http://v8.1c.ru/8.2/virtual-resource-system"
```

```
xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" base="/Test">
```

```
<ws>
```

```
<point name="test" alias="test.1cws" enable="true"/>
```

```
</ws>
```

```
<debug enable="true" url="tcp://localhost" />
```

```
</point>
```

Далее в каталоге, где установлена система 1С найти подкаталог conf, например:

C:\Program Files (x86)\1cv8t\conf\

Нужно создать файл с названием *comcntrcfg.xml* со следующим содержанием:

```
<config xmlns="http://v8.1c.ru/v8/comcntrcfg">
```

```
<debugconfig debug="true" debuggerURL="tcp://localhost:1560" />
```

```
</config>
```

**Обязательно перезагружаем Apache.**

И в заключении нужно включить отладку веб-сервисов в конфигураторе: пункт меню Отладка -> Подключение ... В открывшемся окне «Предметы отладки» нажать кнопку «Автоматическое подключение...». Откроется окно «Автоматическое подключение», в котором надо отметить флагом WEB-сервисы.

Теперь, если установить в функции обработчика метода web-сервиса точку останова, то можно пошагово отследить в отладчике как выполняется метод.

Пример №2. Создадим в нашей конфигурации Web-сервис системы «1С:Предприятие», который должен возвращать по коду контрагента информацию о нем.

В ветви Общие создаем новый Web-сервис с именем «*WebСервис2*».

Заполним свойства Web-сервиса следующим образом:

Пакеты XDTO: <http://www.npi.ru/Customer1>, <http://v8.1c.ru/8.1/data-composition-system/core>, <http://v8.1c.ru/8.1/data/enterprise/current-config>.

URL пространство имен: <http://www.npi.ru/Customer1>

Имя файла публикации: ws2.2cws.

Повторное использование сеансов: Использовать автоматически.

Создадим у web-сервиса метод: *ЗагрузитьКонтрагента*. Укажем у него в свойстве «Тип возвращаемого значения»: Catalog.Object.Контрагенты (<http://v8.1c.ru/8.1/data/enterprise/current-config>). Поставим флаг «Возможно пустое значение». Имя процедуры: *ЗагрузитьКонтрагента*. Данные управления блокировкой данных: Управляемый.

Создадим у метода web-сервиса параметр: имя – *КодКонтрагента*; БТП значения – string (<http://www.w3.org/2001/XMLSchema>).

Примерный код обработчика метода следующий:

Функция *ЗагрузитьКонтрагента*(КодКонтрагента)

КонтрагентТип=ФабрикаXDTO.Тип("http://v8.1c.ru/8.1/data/enterprise/current-config", CatalogObject.Контрагенты");

КонтрагентРезультат = ФабрикаXDTO.Создать(КонтрагентТип);

КонтрагентСсылка =

Справочники.Контрагенты.НайтиПоКоду(КодКонтрагента);

Если Не КонтрагентСсылка.Пустая() Тогда

КонтрагентРезультат.Description = КонтрагентСсылка.Наименование;

КонтрагентРезультат.Code = КонтрагентСсылка.Код;

КонтрагентРезультат.Телефон = КонтрагентСсылка.Телефон;

КонтрагентРезультат.ЭлектроннаяПочта =

КонтрагентСсылка.ЭлектроннаяПочта;

КонтрагентРезультат.JivoID = КонтрагентСсылка.JivoID;

КонтрагентРезультат.Ref = КонтрагентСсылка.Ссылка;

КонтрагентРезультат.DeletionMark =

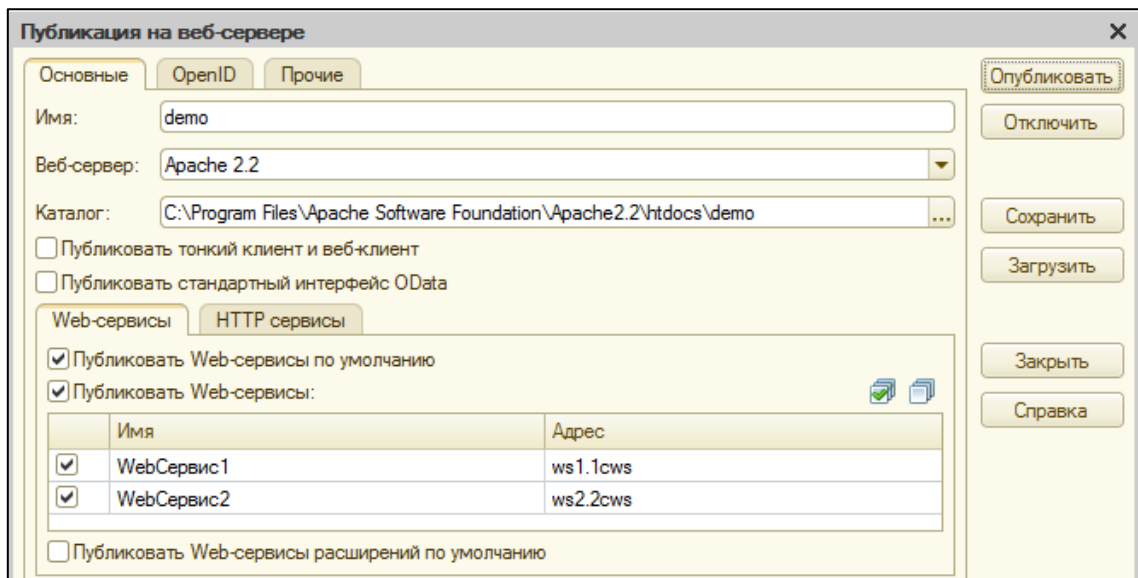
КонтрагентСсылка.ПометкаУдаления;

КонецЕсли;

Возврат КонтрагентРезультат;

КонецФункции

Публикуем web-сервисы (рис. 6.7).



**Рис. 6.7. Публикация web-сервисов**

Создаем во второй конфигурации (использующей опубликованный сервис и у которой в свойстве «Назначения использования» указано «Приложение для мобильной платформы»), обработку с именем «Обработка2».

Создаем форму этой обработки. В ней создадим два реквизита формы: ВыборКонтрагент и Ответ, оба строковые.

Создаем команду формы «Загрузить». Выносим кнопку с командой на форму.

Примерный код обработчика команды в модуле формы следующий:

&НаСервере

Процедура ЗагрузитьНаСервере()

Определение =

Новый WSOпределения("http://10.0.2.2/demo/ws/ws2.2cws?wsdl");

Прокси = Новый WSПрокси(Определение, "http://www.npi.ru/Client1",  
"WebСервис2", "WebСервис2Soap");

Отклик = Прокси.ЗагрузитьКонтрагента(ВыборКонтрагент);

Ответ = "Код="+Отклик.Code+"; Наименование="+Отклик.Description+  
"; Телефон="+Отклик.Телефон+"; ЭлектроннаяПочта="+  
Отклик.ЭлектроннаяПочта+"; JivoID="+Отклик.JivoID;

КонецПроцедуры

&На Клиенте

Процедура Загрузить(Команда)

ЗагрузитьНаСервере();

КонецПроцедуры

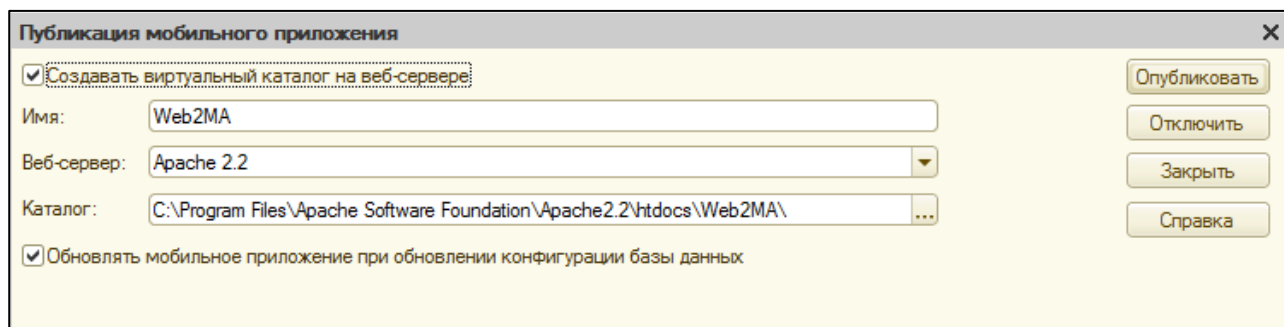
Целесообразно сначала отладить код в режиме запуска как стационарное приложение. В этом случае в WSOпределения изменить 10.0.2.2 на localhost.

Когда приложение корректно работает в стационарном режиме, меняем назад на 10.0.2.2.

Обязательно следует перед публикацией сохранить информационную базу (обновить конфигурацию БД).

Если при обновлении выдает ошибку блокировки, то следует остановить веб-сервер, сохранить БД и снова запустить веб-сервер.

Публикуем мобильное приложение (рис. 6.8). Чтобы это сделать выберем в меню: Конфигурация -> Мобильное приложение-> Опубликовать...



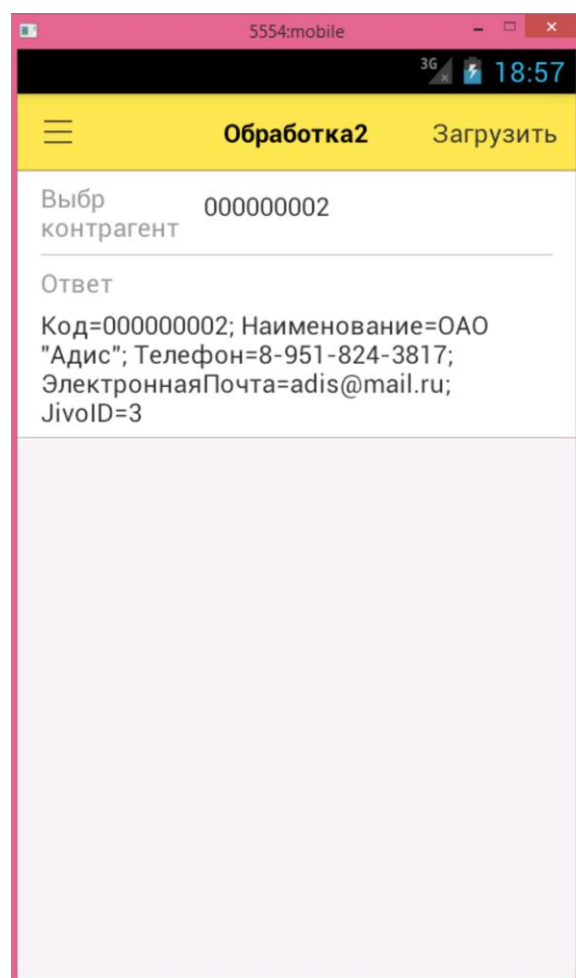
**Рис. 6.8. Публикация мобильного приложения**

В режиме исполнения в эмуляторе мобильного устройства проверяем работоспособность (рис. 6.9).

В поле «ВыбрКонтрагент» вводим код контрагента, сведения о котором следует получить. Например, мы хотим узнать данные контрагента с кодом 000000002.

Нажав «Загрузить» получаем сформированный ответ от web-сервиса с данными о контрагенте (код, наименование, телефон, электронная почта, JivoID).

Итак, Web-сервисы – это один из механизмов платформы, используемых для интеграции с другими информационными системами. Он является средством поддержки SOA (Service-Oriented Architecture) — сервис-ориентированной архитектуры, которая является современным стандартом интеграции приложений и информационных систем. Это означает, что программы для обмена данными друг с другом используют «сервисы» [1].



**Рис. 6.9. Результат выполнения в мобильном приложении**

Веб сервис – это набор функций (процедур), которые могут быть вызваны извне программы и базы данных.

Для того, чтобы другая программа знала, что такой «сервис» есть у этой программы, первая на веб сайте выкладывает сведения о своих сервисах. Формат такого описания – файл WSDL.

Значительным преимуществом сервис — ориентированной архитектуры является то, что она позволяет развивать инфраструктуру предприятия однородным образом, без разрушения уже существующих решений. Ее использование позволяет минимизировать издержки за счет интеграции разнородных и унаследованных систем в современный ландшафт предприятия. Она позволяет реализовывать слабо связанные программные компоненты с тем, чтобы максимально повысить их повторную используемость.

Сервис-ориентированная архитектура интенсивно развивается и поддерживается крупными вендорами. Она строится на базе сервисов, автономных или управляемых извне. Предпочтительным способом их реализации являются веб-сервисы. Они независимы от платформы, автономны и поддерживаются повсеместно.

Веб-сервисы независимы от платформы, автономны и поддерживаются повсеместно, что позволяет минимизировать издержки за счет интеграции разнородных и унаследованных систем (программных компонентов) в современную структуру предприятия.

Web-сервис в 1С описывается набором операций, каждая из которых может иметь некоторое количество параметров для передачи данных Web-сервису и получения возвращаемых значений.

Типы параметров и возвращаемых значений Web-сервисов описываются с помощью механизма XDTO. XDTO (XML Data Transfer Objects) — механизм объектного моделирования данных, описываемых с помощью схемы XML. Большинство примитивных типов сериализованных объектов разработана компанией W3 и их описание находится в пространстве имен «<http://www.w3.org/2001/XMLSchema>».

Например, в случае если функция, реализующая операцию WEB-сервиса, принимает какое-либо значение в качестве параметра (или возвращает какое-либо значение), то такое значение определяется (при настройке соответствующего объекта конфигурации) как имеющее тип объект или значение XDTO.

Общий объект конфигурации 1С "Web-сервис" содержит модуль, в котором располагаются процедуры, выполняемые при вызове тех или иных операций Web-сервиса.

Прикладное решение 1С:Предприятия 8 может являться как поставщиком веб-сервисов, так и потребителем веб-сервисов, опубликованных другими поставщиками.

*1С:Предприятие — поставщик веб-сервисов:* в конфигурацию можно добавить специальный объект, — Web-сервис, — с помощью которого описать некоторую функциональность прикладного решения, например, получение списка имеющихся на некотором складе товаров, их количества и цен. После



публикации на веб-сервере такой сервис будет доступен сторонним потребителям. В качестве потребителей могут выступать системы, использующие произвольные аппаратные и программные платформы. Технология веб-сервисов является платформенно независимой [1].

Если прикладное решение является поставщиком веб-сервиса то и в файловом, и в клиент-серверном варианте работы взаимодействие между прикладным решением и потребителями веб-сервиса осуществляется через веб-сервер, с помощью модуля расширения веб-сервера. При этом, когда потребитель обращается к веб-сервису прикладного решения, выполняется модуль веб-сервиса. Этот модуль содержится в конфигурации и в нем располагаются процедуры, выполняемые при вызове тех или иных операций веб-сервиса. В случае клиент-серверного варианта работы этот модуль будет исполняться в кластере. В случае файлового варианта работы — в модуле расширения веб-сервера.

*1С:Предприятие — потребитель веб-сервисов:* в прикладном решении можно описать ссылку на веб-сервис, опубликованный сторонним поставщиком. После этого прикладное решение сможет использовать данные, получаемые с помощью такого веб-сервиса, в своих внутренних прикладных алгоритмах [1].

Если прикладное решение является потребителем веб-сервиса стороннего поставщика, то в этом случае взаимодействие между прикладным решением и поставщиком веб-сервиса осуществляет клиентское приложение. Оно вызывает те или иные операции веб-сервиса и обрабатывает полученные данные. Платформа 1С:Предприятия, выступающая в роли потребителя веб-сервиса, поддерживает оптимизированную передачу двоичных данных по протоколу МТОМ. Включение режима МТОМ в системе «1С:Предприятие» происходит автоматически.

WEB и HTTP сервисы очень сходны между собой, но все же в них есть кардинальные отличия.

Сходны они тем, что: предназначены для доступа к базе данных снаружи, работают посредством веб-технологий, поверх протокола TCP, оба поддерживают технологии XML и JSON.

Теперь расскажем о различиях (табл. 6.1).

Таблица 6.1

Различия технологий WEB и HTTP



<b>WEB</b>	<b>HTTP</b>
Сервисно-ориентированный	Ресурсно-ориентированный
Есть описание процедур (WSDL)	Нет описания (только документация)
Есть проверка типов данных (XDTO)	Нет проверки типов (есть заголовки, параметры и тело запроса)
Преимущественно XML	Преимущественно JSON

WEB технология — это сервисно-ориентированная технология, она по сути является удаленным вызовом процедур. Мы проектируем описание процедур, описание передаваемых параметров, и с помощью WEB сервисов мы эти процедуры можем вызывать. 1С со своей стороны также предоставляет технологию XDTO, которая позволяет валидировать входящие и исходящие данные, передаваемые в формате XML.

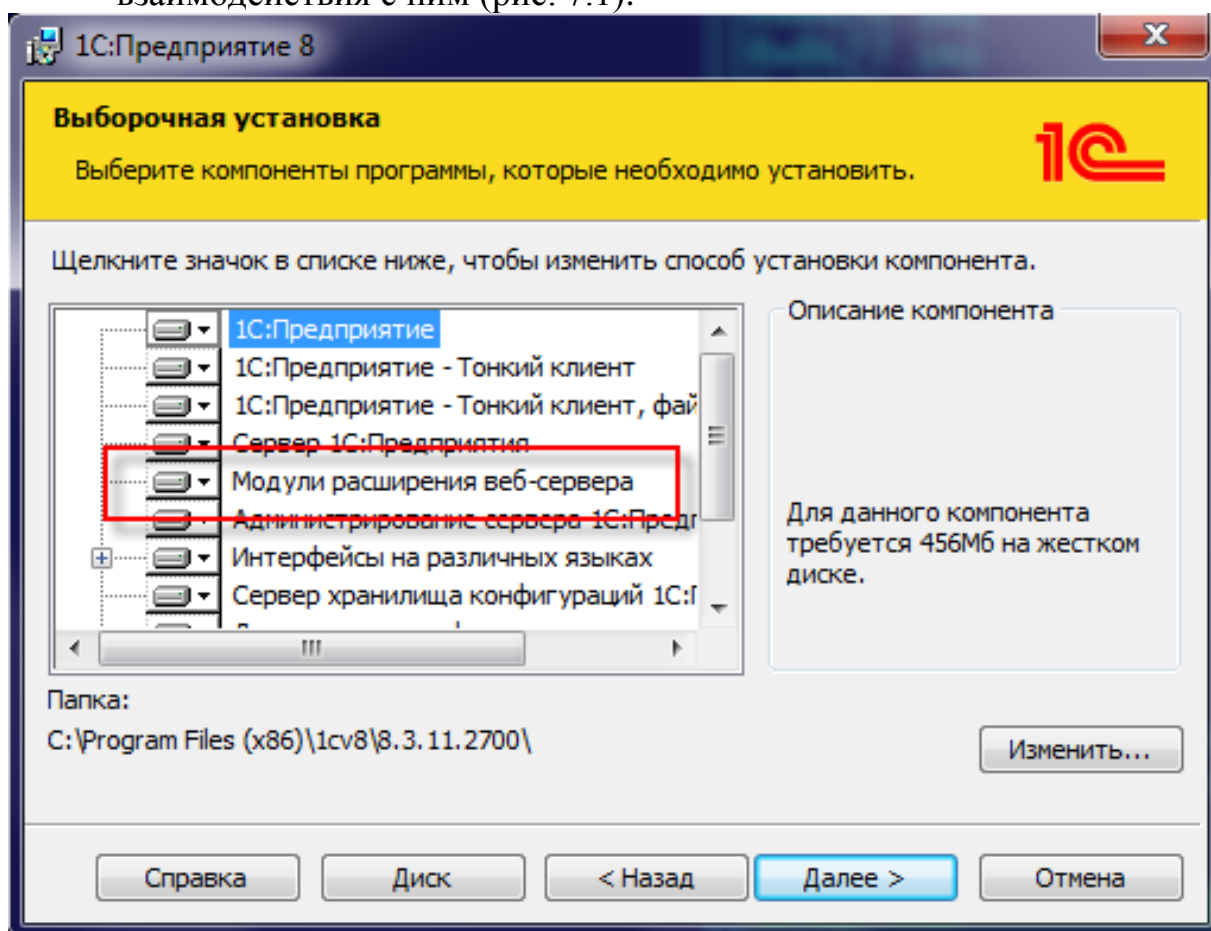
HTTP сервисы же основаны практически на голом HTTP, и эта технология ресурсно-ориентированная. Нет описания, нет проверки типов, нет проверки входящих и исходящих данных — есть только заголовки, параметры и тело запроса. И исторически используется формат данных JSON.

Логично, что WEB-сервисы потенциально сложнее в реализации, потенциально используют больший объем передаваемых данных и дают потенциально большую вычислительную нагрузку.

## Тема 7. Использование HTTP-сервисов в 1С

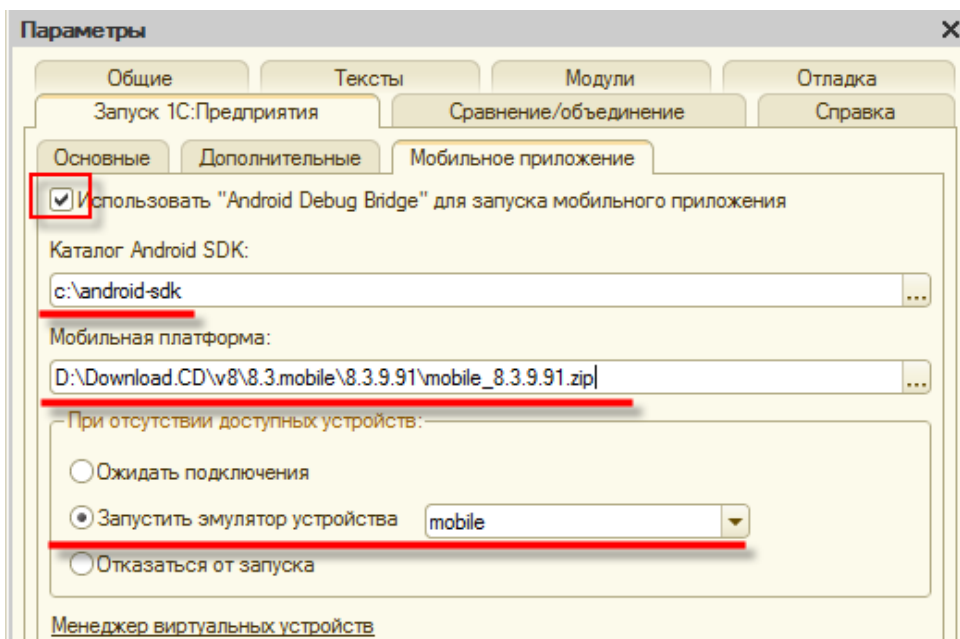
Для выполнения практически заданий по этой теме необходимо убедиться, что на компьютере установлено все необходимое программное обеспечение.

- 1) Убедитесь, что JAVA в ОС установлен и работает корректно. Проверить корректность установки можно набрав "java -version" в консоли Windows и получив в ответ номер версии.
- 2) Должен быть установлен Web-сервер (например, Apache) и у системы «1С:Предприятие» должны быть установлены компоненты для взаимодействия с ним (рис. 7.1).



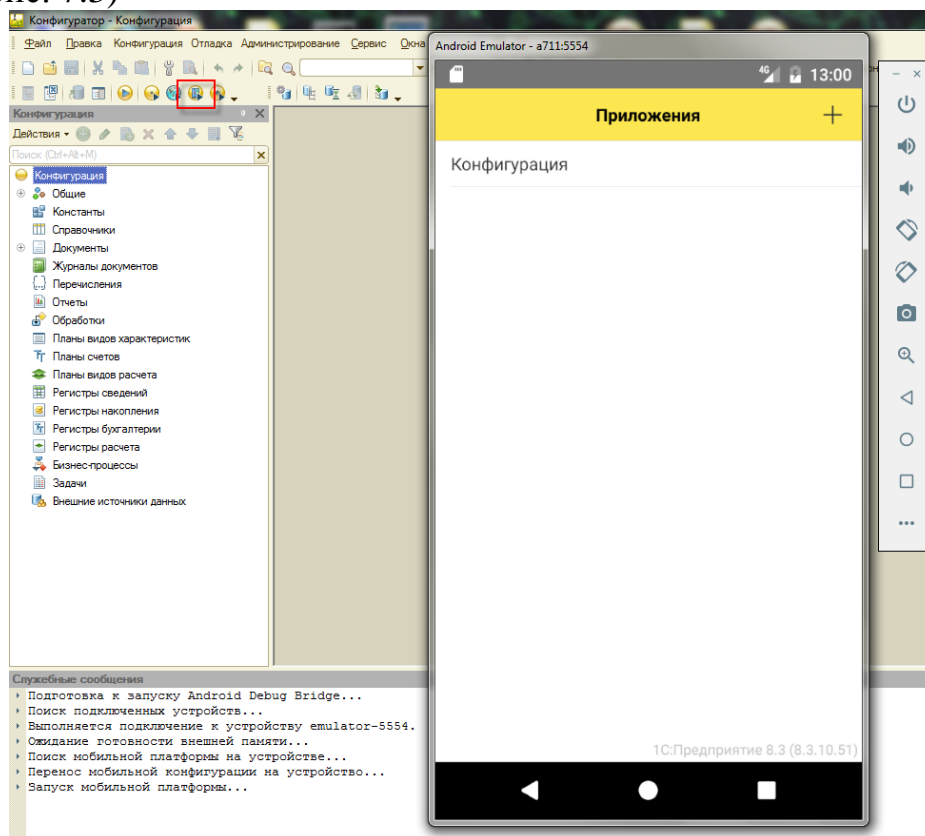
**Рис. 7.1. Компоненты платформы «1С:Предприятие» для работы с web-сервером**

- 3) Должен присутствовать эмулятор мобильного устройства.
- 4) Запускаем Конфигуратор 1С, идем в настройки взаимодействия с мобильной платформой ("Сервис - Параметры") и указываем параметры доступа к эмулятору и мобильной платформе (в виде поставки в архиве zip, распаковывать его не нужно. Версия может отличаться от той, что указана на скрине (рис. 7.2). Путь, куда сбросить архив с мобильной платформой, тоже можно выбрать произвольно).



**Рис. 7.2. Настройка параметров для работы с мобильной платформой**

- 5) Далее крайне рекомендуется запустить мобильную отладку из конфигуратора 1С, чтобы в эмулятор установилась мобильная платформа 1С, если она там отсутствует (это занимает ощутимое время в первый раз). Для запуска отладки достаточно опубликовать конфигурацию на веб-сервере (можно даже пустую), отвечая утвердительно на все вопросы платформы, и запустить отладку (рис. 7.3)



**Рис. 7.3. Запуск отладки в эмуляторе мобильного устройства**

Возможность разработки HTTP-сервисов в составе конфигурации появилась начиная с версии платформы 8.3.5.

Система «1С:Предприятие» позволяет прикладному разработчику реализовать собственные HTTP-сервисы. Создание HTTP-сервисов позволяет создать некий прикладной интерфейс, который будет доступен с помощью HTTP-запросов.

Обращение к HTTP-сервису выполняется по некоторому URL. В качестве примера можно привести HTTP-сервис, который при обращении по одному URL возвращает список каких-либо документов (например, расходных накладных), а при обращении по другому URL будет возвращена конкретная накладная. Действие, которое следует выполнить серверу, определяется тем, какой HTTP-метод адресован серверу. Например, если на сервер отправляется GET-запрос, то выполняется получение каких-либо данных, DELETE-запрос приводит к удалению каких-либо данных и т. д.

При обращении к HTTP-сервису формируется URL, который выглядит следующим образом: `http://localhost/hs/корневойURL/относительныйURL`. Более подробно рассмотрим составные части адреса:

- `http://localhost` – обычный URL, по которому выполняется доступ, например, к информационной базе с помощью веб-клиента.
- `hs` – признак того, что выполняется обращение к HTTP-сервису (в отличие от `ws`, который определяет доступ к Web-сервису).
- `корневойURL` – имя ресурса, которое определяет группу ресурсов, связанных общим смыслом. Задается в свойстве объекта HTTP-сервис.
- `относительныйURL` – определяет ресурс, к которому будет выполняться обращение. Относительный URL, указанный в запросе, будет использован для определения конкретного ресурса, к которому выполнялось обращение. Правило сопоставления задается в объекте Шаблон URL.

Разработаем в рамках практических занятий сервис, который должен уметь принимать GET запрос и отдавать ответ в формате JSON. Создадим в рамках нашей конфигурации http-сервис, передающий в ответ на запрос список контрагентов.

Доработаем справочник «Контрагенты». Добавим в него несколько простых реквизитов, на примере передачи которых покажем работу с http-сервисами:

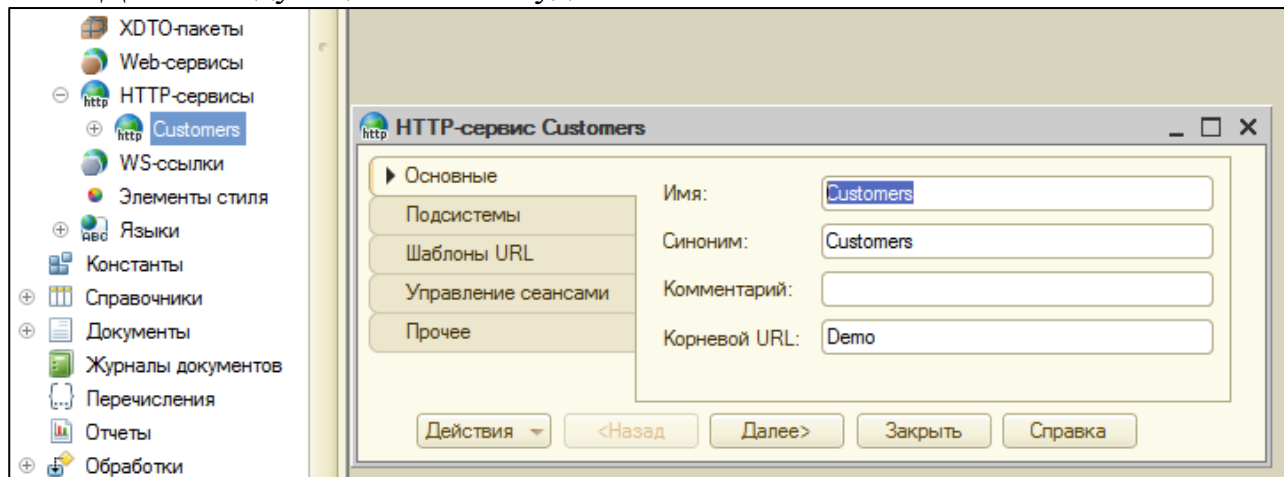
- «Телефон», тип Строка, длина 25;
- «ЭлектроннаяПочта», тип Строка, длина 50;
- «JivoID», тип Строка, длина 25;

Откроем в дереве конфигурации раздел «Общие», найдем в нем группу «HTTP-сервисы» и добавим новый веб сервис. В окне создания нового сервиса заполним поля:

- Имя – Customers;
- Синоним – Customers;
- Корневой URL – Demo.

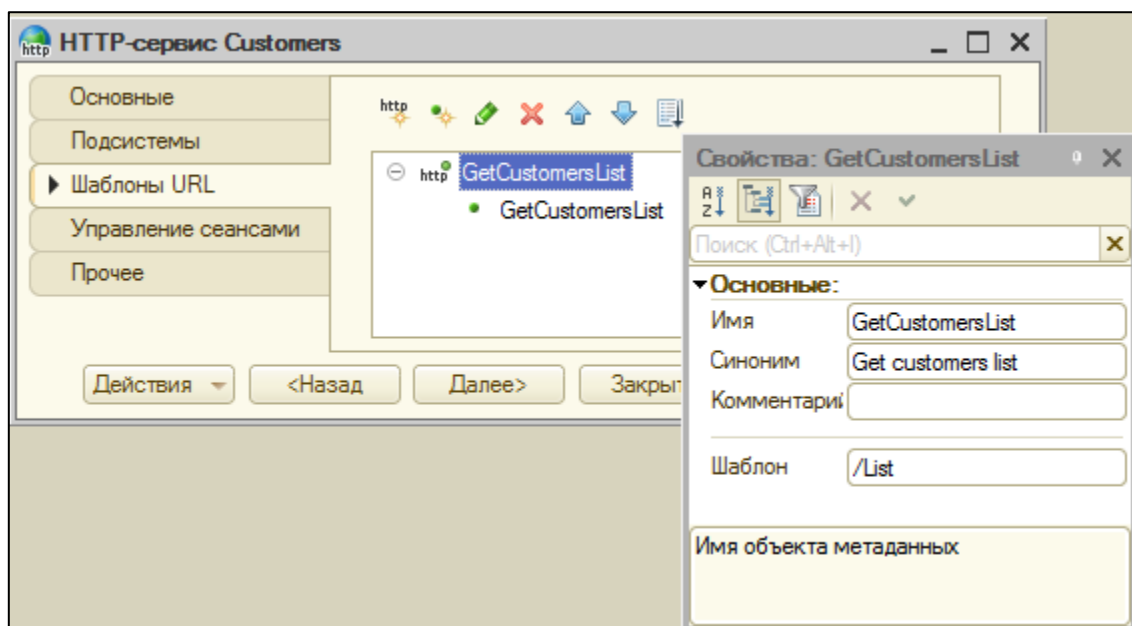
Указанное в поле «Корневой URL» наименование будет использоваться при обращении к сервису. Здесь не должно быть пробелов и желательно не использовать кириллицу (рис. 7.4).

Далее следующим шагом будет описание шаблона URL.



**Рис. 7.4. Создание http-сервиса**

Перейдем на соответствующую закладку и добавим шаблон с именем «GetCustomersList». В инспекторе свойств найдем свойство «Шаблон» и напишем в нем «/List» (рис. 7.5). Этим самым мы зададим путь, по которому будет происходить взаимодействие с http-сервисом. Шаблон может содержать специальные символы, позволяющие определить передаваемые параметры (как обязательные, так и нет), но для первого примера мы ограничимся простым «/List». При переходе по этому пути будет срабатывать наш единственный метод, и отдавать клиенту набор данных.



**Рис. 7.5. Добавление шаблона с именем «GetCustomersList»**

Таким образом, шаблон есть, теперь дело за методом. Добавим для нашего шаблона, который назовем метод GetCustomersList (рис. 7.6). В инспекторе свойств нам необходимо указать HTTP-метод. Указанное значение

определяет, на какие типы запросов будет реагировать наш метод. Текущая задача вполне позволяет обойтись методом “GET”.

Сервис почти готов, остается только прописать обработчик события для сконструированного метода GetCustomersList. Создать обработчик события можно через инспектор свойств.

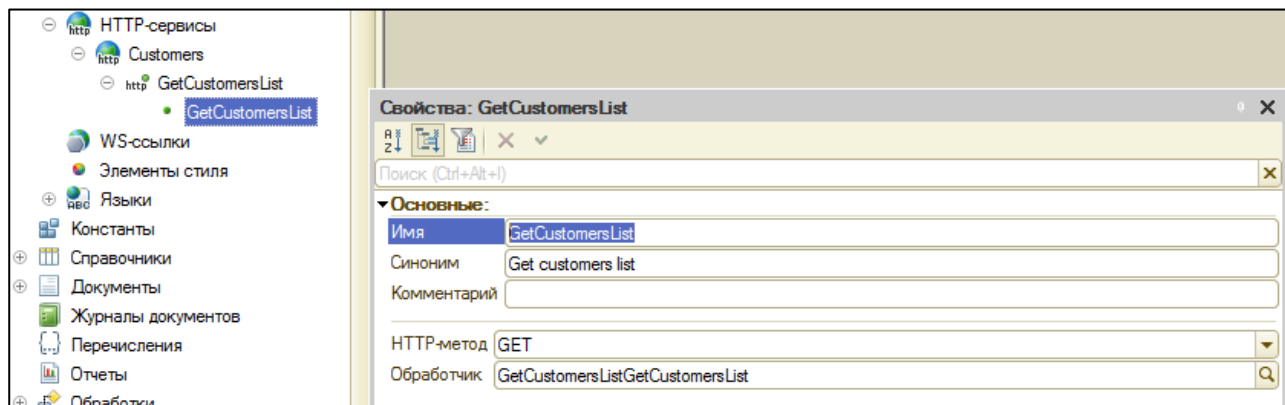


Рис. 7.6. Добавление метода GetCustomersList

В теле обработчика напишем следующий код:

Функция GetCustomersListGetCustomersList(Запрос)

// формируем с помощью объектной техники

// массив клиентов для отправки

Клиенты = Справочники.Контагенты.Выбрать();

МассивКлиентов = Новый Массив;

Пока Клиенты.Следующий()

Цикл

ДанныеКлиента = Новый Структура;

ДанныеКлиента.Вставить("Код", Клиенты.Код);

ДанныеКлиента.Вставить("Наименование", Клиенты.Наименование);

ДанныеКлиента.Вставить("Телефон", Клиенты.Код);

ДанныеКлиента.Вставить("ЭлектроннаяПочта",  
Клиенты.Электронная почта);

ДанныеКлиента.Вставить("JivoID", Клиенты.JivoID);

МассивКлиентов.Добавить(ДанныеКлиента);

КонецЦикла;

// сериализуем массив клиентов в JSON

ЗаписьJSON = Новый ЗаписьJSON;

ЗаписьJSON.УстановитьСтроку();

ЗаписатьJSON(ЗаписьJSON, МассивКлиентов);

СтрокаДляОтвета = ЗаписьJSON.Закрыть();

// формируем ответ

Ответ = Новый HTTPСервисОтвет(200);

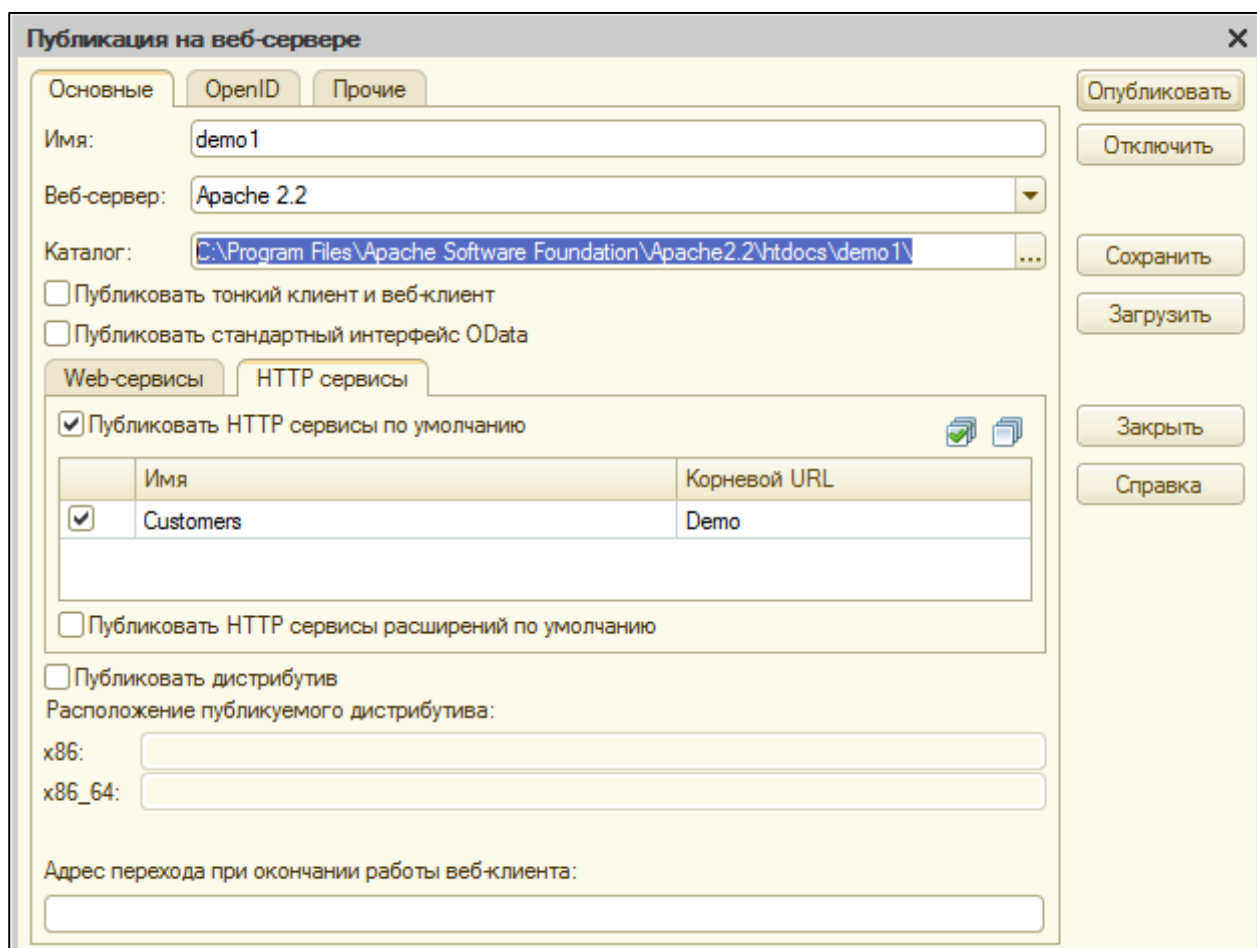
Ответ.Заголовки.Вставить("Content-type", "application/JSON;  
charset=utf-8");

Ответ.УстановитьТелоИзСтроки(СтрокаДляОтвета,

КодировкаТекста.UTF8,  
ИспользованиеByteOrderMark.Неиспользовать);  
Возврат Ответ;  
КонецФункции

Разберем код. В самом начале описываем массив, состоящий из структур. Все его содержимое мы переведем в JSON и отдадим как ответ. Инициализируем объект типа ЗаписьJSON. Обязательно вызываем метод УстановитьСтроку(), т.к. нам необходимо получить текст JSON в строковую переменную. Далее вызываем глобальный метод ЗаписатьJSON(), которому передаем объект типа ЗаписьJSON и массив, который необходимо сконвертировать. Дальше получаем результат в переменную «Ответ» и готовим HTTP-ответ. В ответ мы добавляем (сметод «УстановитьТелоИзСтроки») получившийся JSON.

Опубликуем созданный HTTP-сервис для тестирования результатов проделанной работы (рис. 7.7). Для простейшего теста в вашей системе должен быть установлен веб-сервер, например Apache.



**Рис. 7.7. Публикация HTTP-сервиса для тестирования результатов**

Для публикации необходимо перейти в меню «Администрирование» и выбрать пункт «Публикация на веб-сервере». В появившемся окне заполняем:

Имя – имя нашего решения. Оно будет использоваться в URL при доступе к опубликованной ИБ. Например, если указать здесь demo, то ваша ИБ будет доступна по адресу <http://localhost/demo>.

Веб-сервер - Если в качестве веб-сервера используется Apache 2.2, по путь может быть таким: C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\;

Каталог – путь к директории, в которую будет помещен конфигурационный файл публикуемой ИБ;

Снимаем флажки «Публиковать тонкий клиент и веб-клиент», «Публиковать стандартный интерфейс OData», «Публиковать WEB-сервисы по умолчанию»;

На вкладке HTTP-сервисы отмечаем флаг «Публиковать HTTP-сервисы по умолчанию» и в табличной части отмечаем созданный сервис.

Нажимаем кнопку «Опубликовать», платформа предложит перезапустить веб-сервер – соглашаемся.

Для тестирования созданного http-сервиса запустим какой-нибудь браузер и попробуем обратиться к нему. Если вы повторяли все мои действия, то путь должен получиться таким:

**http://localhost:8080/demo1/hs/demo/list**

Первая часть– localhost. Это имя хоста, на котором установлен веб-сервер. Дальше указывается ссылка (services) на опубликованную ИБ. Следом идет hs, этот элемент пути указывает, что нас интересует взаимодействие с http-сервисом. Последние два элемента – корневой URL нашего сервиса и шаблон URL. Стоит обратить внимание на порт 8080, который указан через двоеточие после имени хоста. Если вы ставили Apache с настройками по умолчанию, то у вас он будет слушать 80-й порт, следовательно, ничего указывать не нужно. В итоге URL будет таким:

**http://localhost/demo1/hs/demo/list**

Попробуйте перейти по указанному выше адресу, и если все работает корректно, вы получите страницу с некими данными о контрагентах в JSON формате, подобно той, что представлена на рис. 7.8:





```

[
  {
    "Код": "000000002",
    "Наименование": "ОАО \"Адис\"",
    "Телефон": "8-951-824-3817",
    "ЭлектроннаяПочта": "adis@mail.ru",
    "JivoID": 3
  },
  {
    "Код": "000000003",
    "Наименование": "ООО \"Олимп\"",
    "Телефон": "8-938-456-3491",
    "ЭлектроннаяПочта": "olimp@gmail.ru",
    "JivoID": 3
  },
  {
    "Код": "000000001",
    "Наименование": "ООО \"Парнас\"",
    "Телефон": "8-951-367-3512",
    "ЭлектроннаяПочта": "parnas@mail.ru",
    "JivoID": 2
  }
]

```

**Рис. 7.8. Страница с данными в JSON-формате**

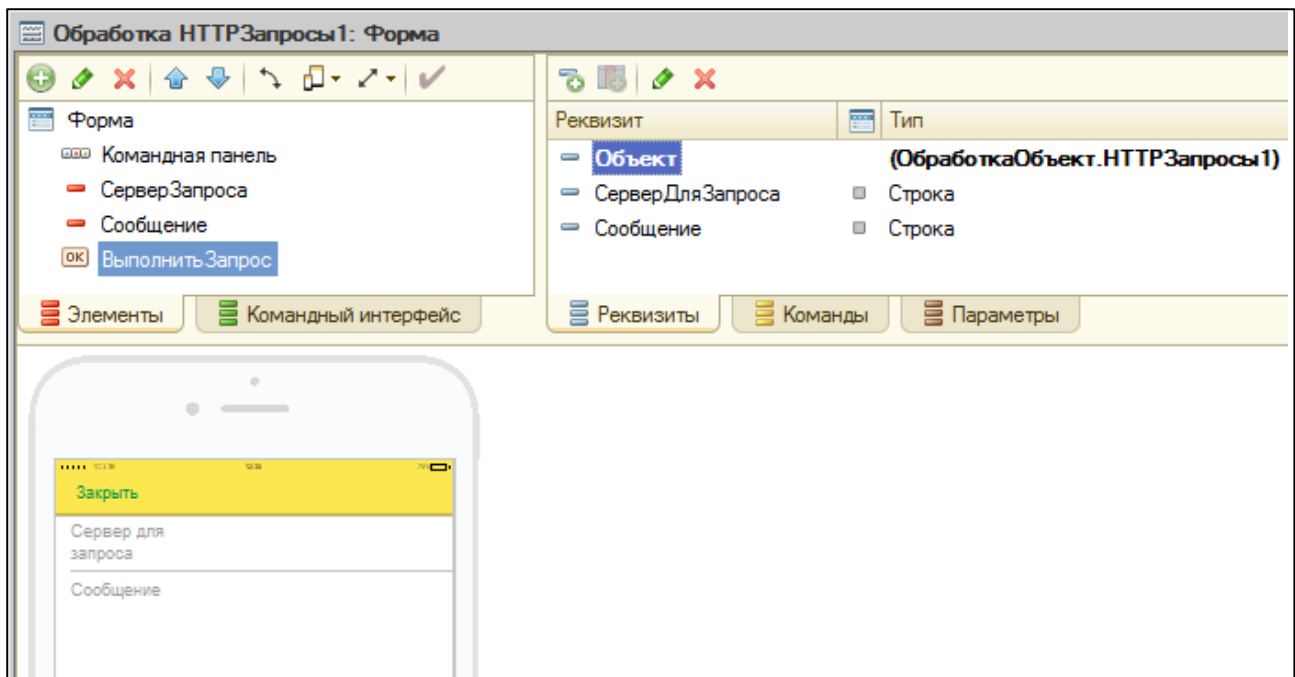
Решим обратную задачу – в другой конфигурации необходимо реализовать Импорт списка контрагентов из опубликованного http-сервиса.

Создадим новую конфигурацию, в которой создадим справочник «Контрагенты» со структурой, аналогичной справочнику из первой конфигурации. У этой конфигурации в свойстве «Назначения использования» укажем «Приложение для мобильной платформы».

Создадим в конфигурации №2 обработку «НТТРЗапросы1». На ее форме создадим реквизиты формы «СерверДляЗапроса» и «Сообщение», оба реквизита имеют тип Строка. Первый будет содержать путь к сервису, а второе поле – служить для отображения ответа сервиса. У реквизита формы «Сообщение» надо в свойствах отметить МногострочныйРежим. Оба реквизита выносим на форму обработки.

Создаем команду формы «ВыполнитьЗапрос». Выносим кнопку на форму обработки (рис. 7.9).

Далее через свойство Действие команды «ВыполнитьЗапрос» создаем заготовки процедур обработчика, выбрав вариант «Создать на клиенте и процедуру на сервере».



**Рис. 7.9. Форма обработки «HTTPЗапросы1»**

Код обработчика команды в модуле формы следующий:

&НаКлиенте

Процедура ВыполнитьЗапрос(Команда)

```
// Создаем HTTPСоединение, параметры оставляем по умолчанию
Соединение = Новый HTTPСоединение(СерверДляЗапроса, );
HTTPЗапрос = Новый HTTPЗапрос("/");
// выполнить GET-запрос
Q=Соединение.Получить(HTTPЗапрос);
Ответ=Q.ПолучитьТелоКакСтроку("UTF-8");
Сообщение=Ответ;
ВыполнитьЗапросНаСервере(Ответ);
```

КонецПроцедуры

&НаСервере

Процедура ВыполнитьЗапросНаСервере(Ответ)

```
// получаем тело запроса и десериализуем данные о клиенте из JSON
ЧтениеJSON = Новый ЧтениеJSON;
ЧтениеJSON.УстановитьСтроку(Ответ);
ДанныеКлиента = ПрочитатьJSON(ЧтениеJSON);
ЧтениеJSON.Закреть();
// ищем клиента, если нет в ИБ – создаем новую запись
Для Каждого Клиент Из ДанныеКлиента Цикл
    Если Клиент.Свойство("Код"), Тогда
        КлиентСсылка =
            Справочники.Контрагенты.найтиПоКоду(Клиент.Код);
        Если НЕ ЗначениеЗаполнено(КлиентСсылка) Тогда
```

```

КлиентОбъект =
    Справочники.Контрагенты.СоздатьЭлемент();
Иначе
    КлиентОбъект = КлиентСсылка.ПОлучитьОбъект();
КонецЕсли;
КлиентОбъект.Код = Клиент.Код;
КлиентОбъект.Наименование=Клиент.Наименование;
КлиентОбъект.Телефон=Клиент.Телефон;
КлиентОбъект.ЭлектроннаПочта=Клиент.ЭлектроннаяПочта;
КлиентОбъект.JivoID=Клиент.JivoID;
КлиентОбъект.Записать();
КонецЕсли;
КонецЦикла;
КонецПроцедуры

&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтрандартнаяОбработка)
    СерверДляЗапроса = "10.0.2.2/Demo/hs/Demo/List";
    Сообщение = "";
КонецПроцедуры

```

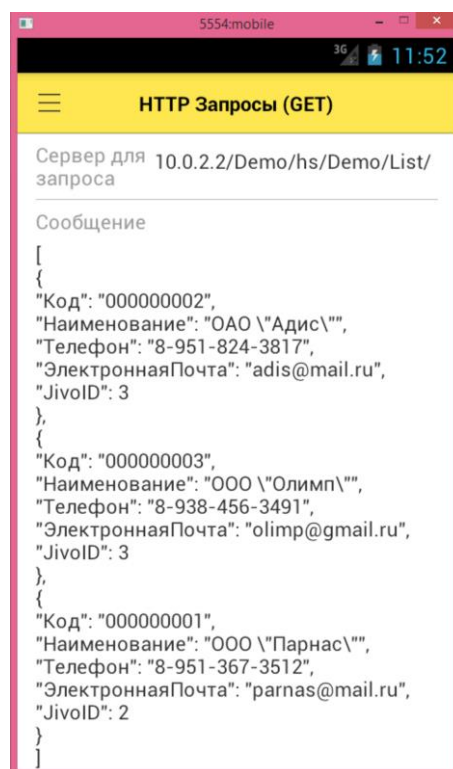
Результатом выполнения запроса к серверу будет список контрагентов в формате JSON. На основе списка обновляются (если они существуют) или создаются новые записи о контрагентах в справочнике.

В случае запуска приложения в режиме стационарного приложения адрес для сервиса начинается с localhost, а если в режиме мобильного приложения – 10.0.2.2.

Запустим конфигурацию №2 в эмуляторе мобильного приложения. Результат представлен на рис. 7.10. Поскольку запуск в режиме мобильного приложения, то сервер для запроса установлен в значение 10.0.2.2/Demo/hs/Demo/List.

Таким образом, в конфигурации №1 были введены данные о нескольких контрагентах и опубликованы через HTTP-сервис.

В конфигурации №2 в случае корректного соединения в многострочное поле Сообщение будут отражены сведения о контрагентах в формате JSON, полученные через HTTP-сервис из конфигурации №1 и записаны в



**Рис. 7.10. Результат работы в эмуляторе мобильного приложения**

справочник «Контрагент» конфигурации №2.

При обработке полученных данных будет проверяться наличие в информационной базе контрагента с таким Кодом. Если запись существует, то данные о контрагенте будут перезаписаны, если не существует, то будет создана новая запись.

Реализуем еще одну функциональность: **Запрос контрагента по коду**. Создадим еще одну новую обработку «НТТРЗапросы», создадим форму для нее (рис. 7.11).

На форме данной обработки помимо реквизитов формы СерверДляЗапроса и Сообщение типа Строка, назначение которых аналогичное обработке, разработанной ранее, создадим еще один реквизит формы «КодКонтрагента», тип Строка, длина 9. Вынесем все реквизиты формы на форму обработки (рис. 7.11).

Код обработчика команды в модуле формы следующий:

&НаКлиенте

Процедура ВыполнитьЗапрос(Команда)

СерверДляЗапроса = СерверДляЗапроса+КодКонтрагента;

// Создаем НТТРСоединение, параметры оставляем по умолчанию

Соединение = Новый НТТРСоединение(СерверДляЗапроса, );

НТТРЗапрос = Новый НТТРЗапрос("/");

// выполнить GET-запрос

Q=Соединение.Получить(НТТРЗапрос);

Сообщение= Q.ПолучитьТелоКакСтроку("UTF-8");

ВыполнитьЗапросНаСервере(Сообщение);

КонецПроцедуры

&НаСервере

Процедура ВыполнитьЗапросНаСервере(Сообщение)

// получаем тело запроса и десериализуем данные о клиенте из JSON

ЧтениеJSON = Новый ЧтениеJSON;

ЧтениеJSON.УстановитьСтроку(Сообщение);

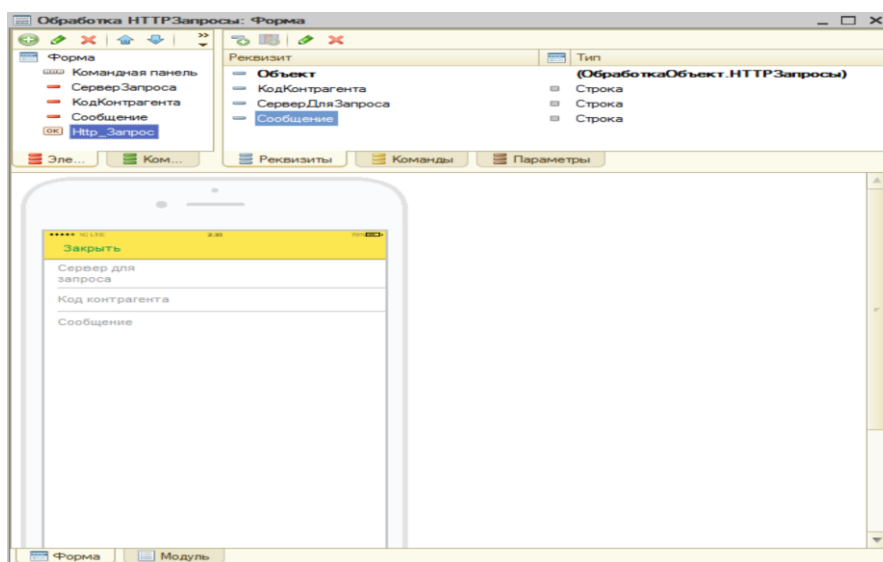


Рис. 7.11. Обработка «НТТРЗапросы»

```

ДанныеКлиента = ПрочитатьJSON(ЧтениеJSON);
ЧтениеJSON.Закрыть();
// ищем клиента, если нет в ИБ – создаем новую запись
Для Каждого Клиент Из ДанныеКлиента Цикл
    Если Клиент.Свойство("Код",) Тогда
        КодКлиента = ДанныеКлиента.Код;
        КлиентСсылка =
            Справочники.Контрагенты.найтиПоКоду(КодКлиента);
    Если НЕ ЗначениеЗаполнено(КлиентСсылка) Тогда
        КлиентОбъект =
            Справочники.Контрагенты.СоздатьЭлемент();
    Иначе
        КлиентОбъект = КлиентСсылка.ПОЛУЧИТЬОбъект();
    КонецЕсли;
    КлиентОбъект.Код = Клиент.Код;
    КлиентОбъект.Наименование=Клиент.Наименование;
    КлиентОбъект.Телефон=Клиент.Телефон;
    КлиентОбъект.ЭлектроннаяПочта=Клиент.ЭлектроннаяПочта;
    КлиентОбъект.JivoID=Клиент.JivoID;
    КлиентОбъект.Записать();

```

КонецЕсли;

КонецЦикла;

КонецПроцедуры

&НаСервере

Процедура

ПриСозданииНаСервере(Отказ,  
СтандартнаяОбработка)

СерверДляЗапроса =

"10.0.2.2/Demo/hs/Demo/List";

Сообщение = "";

КонецПроцедуры

Опубликуем мобильное приложение (в меню пункт «Конфигурация | Мобильное приложение | Опубликовать ...»).

Запустим в режиме мобильного приложения и протестируем http-сервис, опубликованный в первой конфигурации.

Откроем обработку HTTPЗапросы. Введем в качестве кода контрагента, о котором хотим

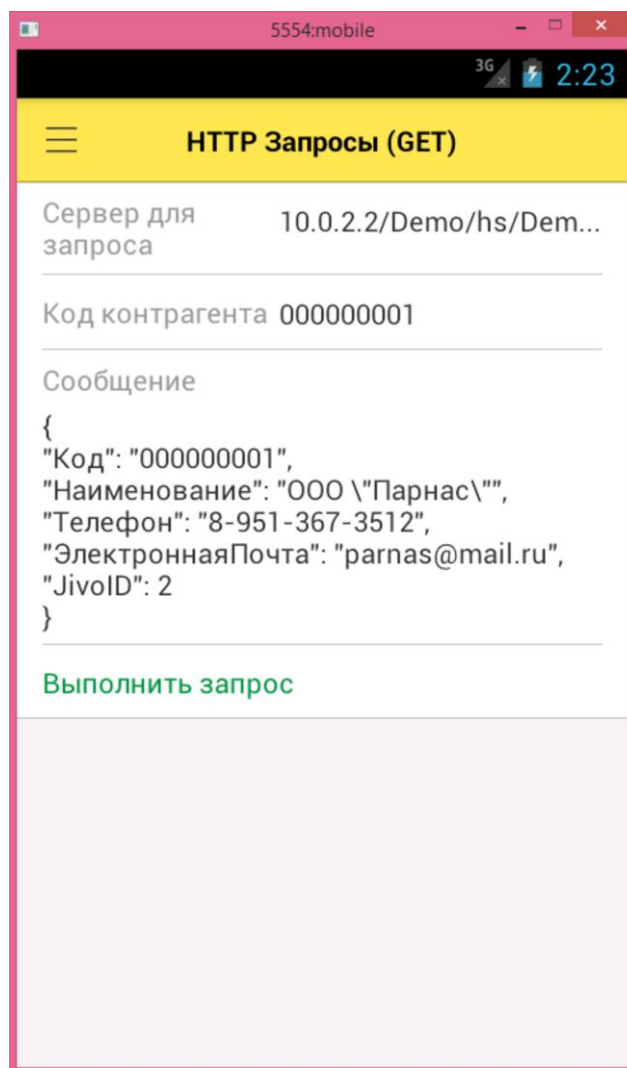


Рис. 7.12. Результат работы

узнать информацию с помощью http-сервиса – 000000001.

Результат с некоторыми данными (сами данные могут отличаться) будут отражены в поле Сообщение (рис. 7.12).

Итак, Бизнесы все активнее используют интернет-сервисы: интернет-магазины, торговые площадки, облачные CRM, социальные сети, системы рассылок...

Естественно, возникает задача интеграции с корпоративной системой – это сокращает рутинные работы и повышает скорость принятия решений. Для интеграции с конфигурациями на платформе “1С:Предприятие 8” могут быть в качестве одно из вариантов использованы Http-сервисы.

HTTP-сервис позволяет получать/изменять данные, но при этом, HTTP-сервисы потенциально позволяют упростить создание клиентских приложений, уменьшить объем передаваемых данных и вычислительную нагрузку, все это особенно важно для мобильных устройств.

В рамках практических занятий по теме №7 мы познакомились с объектами встроенного языка 1С, которые реализуют возможности обмена по протоколу HTTP.

## Тема 8. WS-ссылки и получение данных с Web-сервисов сторонних поставщиков

Платформа 1С позволяет разработчику вызывать Web-сервисы, предоставляемые другими поставщиками, двумя способами [1]:

- с помощью *статических ссылок*, создаваемых в дереве объектов конфигурации. При использовании статической ссылки платформа 1С получает описание Web-сервиса поставщика только один раз, при создании ссылки. За счет этого достигается большая скорость работы;
- с помощью *динамических ссылок*, создаваемых средствами встроенного языка. При использовании динамической ссылки, описание Web-сервиса получается каждый раз при вызове Web-сервиса. Скорость работы при этом уменьшается, но зато такой подход обеспечивает актуальность описания Web-сервиса поставщика, то есть используется последняя версия WSDL-документа.

WS-ссылка – это общий объект конфигурации. Она предназначена для описания в прикладном решении "статической" ссылки на некоторый внешний веб-сервис стороннего поставщика. WS-ссылка в системе «1С:Предприятие» – это описание стороннего веб-сервиса по его WSDL описанию.

Это может быть веб-сервис другой базы 1С. Это может быть веб-сервис другой программы или веб-сайта.

Для нас это совершенно не важно – важно, что есть набор функций, которые мы можем вызвать через интернет.

WS-ссылка представляет собой WSDL-описание веб-сервиса, импортированное из указанного источника.

WS-ссылка недоступна для редактирования, однако можно просмотреть ее структуру и структуру типов данных, которые используются для описания параметров и возвращаемых значений [1].

Дальнейшая работа с такой ссылкой выполняется средствами встроенного языка. Описание ссылки на веб-сервис в дереве конфигурации, с помощью объекта WS-ссылка, удобно для редко меняющихся сервисов. Описание веб-сервиса получается один раз, при создании объекта конфигурации, и хранится в конфигурации. Поэтому все обращения к такой ссылке из встроенного языка будут выполняться быстро. Однако в случае изменения описания веб-сервиса, потребуется повторный импорт его описания в WS-ссылку.

Для работы с часто меняющимися веб-сервисами во встроенном языке предусмотрена возможность динамического считывания описания веб-сервиса и построение его прокси. Также эта возможность позволяет вызывать веб-сервисы, расположение которых станет известно только уже в процессе выполнения программы. Однако, несмотря на все преимущества, такой способ работы медленнее, т.к. каждый раз при создании прокси веб-сервера будет тратиться время на получение описания веб-сервиса.



Реализуем пример №1 работы с объектом WS-ссылка.

У Центробанка России существует веб-сервис для получения ежедневных данных: курсы валют, новости, динамика курсов и т.д. Описание сервиса можно найти здесь <http://www.cbr.ru/development/dws/>

Нас интересует один из методов этого сервиса: `GetCursOnDate(On_date)` – получение курсов валют на заданную дату.

Изучим описание на вышеуказанном сайте этого метода:

***GetCursOnDate(On\_date)*** получение курсов валют на определенную дату (ежедневные курсы валют).

Аргументы:

*On\_date* — Дата запроса для курсов, формат — `System.DateTime`

Результат:

XML документ в формате `System.Data.Dataset`, содержащий таблицу `[ValuteCursOnDate]` XSD.

Таблица содержит поля:

*Vname* — Название валюты

*Vnom* — Номинал

*Vcurs* — Курс

*Vcode* — ISO Цифровой код валюты

*VchCode* — ISO Символьный код валюты.

Результирующий `DataSet` содержит `ExtendedProperty` поле «`OnDate`» показывающее дату для которой присланы данные в таблице.

Таким образом, в метод передается один аргумент `On_date` – это дата, на которую требуется получить курсы. В результате возвращается XML, содержащий таблицу `ValuteCursOnDate` (сами курсы и сопутствующая информация).

Создаем обработку «ЗагрузкаКурсовВалютЦБР». Для нее создаем форму. На форме создаем реквизиты формы и размещаем элементы управления так, как показано на рис 8.1. Таблица значений «таблицаКурсовВалют» включает следующие колонки:

«НазваниеВалюты», тип Строка,

«Номинал», тип Строка,

«ЦифровойКодВалюты», тип Строка,

«СимвольныйКодВалюты», тип Строка,

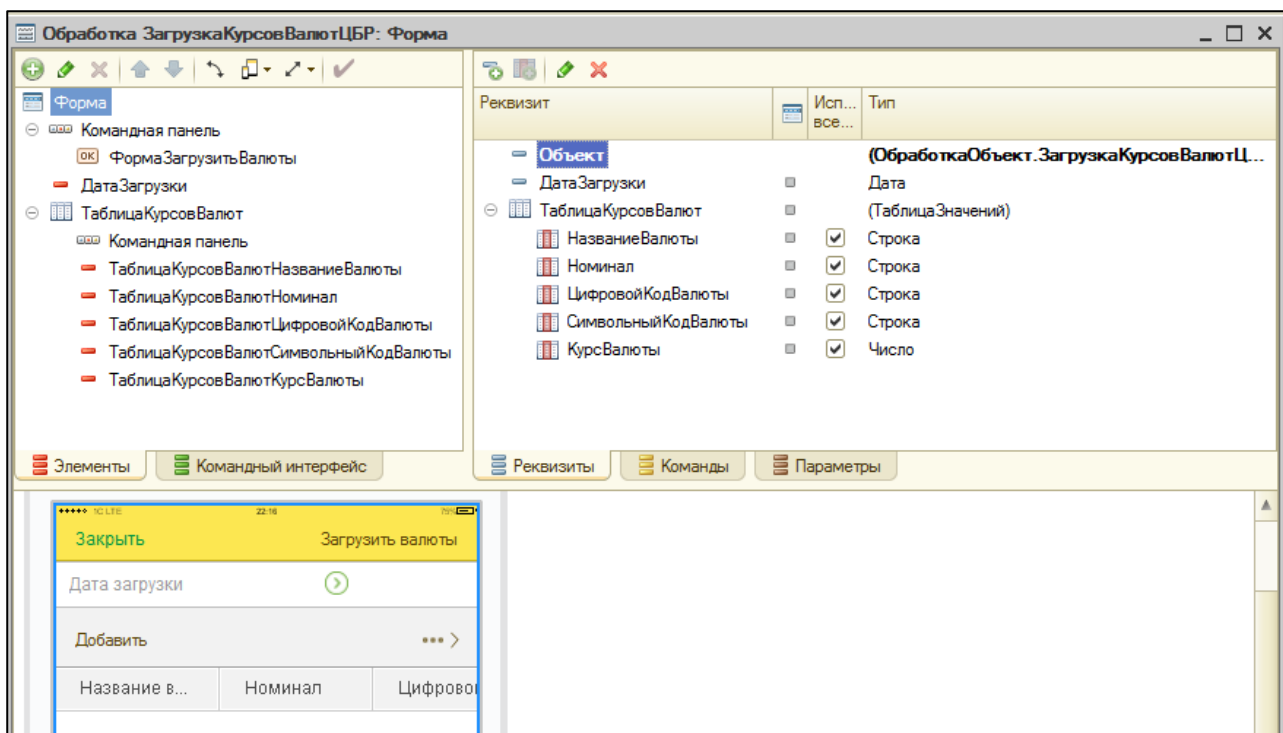
«КурсВалюты», тип Число, длина 15, точность 2.

Создаем ссылку на описание веб-сервиса. В конфигурации добавляем новый объект типа WS-ссылка. В появившемся окне указываем ссылку на WSDL.

**<http://www.cbr.ru/DailyInfoWebServ/DailyInfo.asmx?WSDL>**

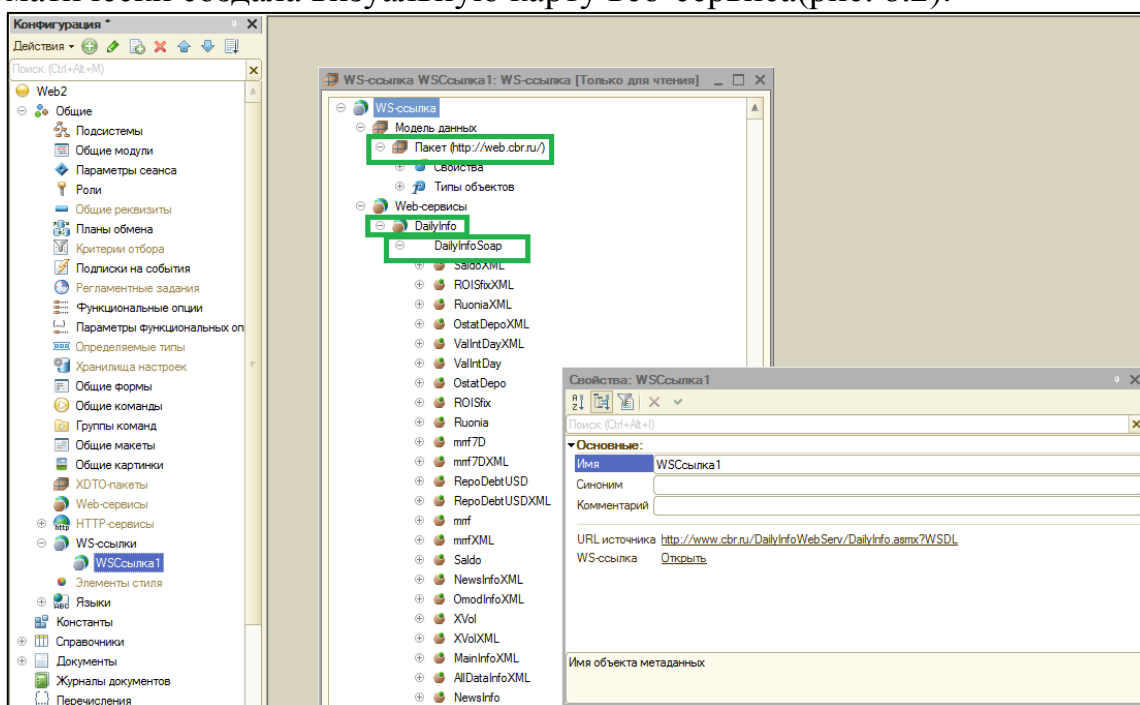
WSDL (англ. Web Services Description Language) — язык описания веб-сервисов и доступа к ним, основанный на языке XML. Для получения WSDL описания сервиса к ссылке на него надо прибавить `?wsdl`.





**Рис. 8.1. Форма обработки «ЗагрузкаКурсовВалютЦБР»**

Система «1С:Предприятие» на основании полученного описания автоматически создала визуальную карту веб-сервиса(рис. 8.2).



**Рис. 8.3. Визуальная карта веб-сервиса по его wsdl-описанию**

Можно увидеть название веб-сервиса, посмотреть какие у него доступны операции, а также используемые типы данных. Зелеными квадратами обведены три параметра, которые будут в дальнейшем использованы для создания прокси.

Модель данных — описывает Пакет XDTO, используемый для работы со ссылкой. Всё дело в том, что разные системы используют разную структуру

хранения данных. Для унификации используются Пакеты XDTO, в которых описываются свойства, типы значений, типы объектов.

Ссылку на веб сервис в конфигурацию в принципе добавлять не обязательно. Возможно создавать ее динамически в момент выполнения кода на языке 1С (указав URL WSDL описания). В этом случае на импорт WSDL в момент создания объекта будет потрачено определенное время.

В режиме редактирования формы обработки необходимо добавить новую команду формы, назовем ее «ЗагрузитьВалюты». Эту команду необходимо связать с кнопкой, расположенной на форме.

Можно обратиться динамически к внешнему веб-сервису, то есть без добавления объекта WS-ссылка. Это бывает удобнее, если создается внешняя обработка и для ее «автономной» работы нежелательно использование добавленных в конфигурацию ws-ссылок. Таким образом, мы можем использовать такие веб-сервисы из внешних обработок без привязки к конфигурации.

В процедуре ЗагрузитьКурсыВалют() строку

*Прокси =*

*WSCсылки.CBR\_DailyInfoWebServ.СоздатьWSПрокси("http://web.cbr.ru/", "DailyInfo", "DailyInfoSoap");*

необходимо заменить двумя следующими строками

*Определения = Новый*

*WSОпределения("http://www.cbr.ru/DailyInfoWebServ/DailyInfo.asmx?WSDL");*

*Прокси = Новый WSПрокси(Определения, "http://web.cbr.ru/", "DailyInfo", "DailyInfoSoap");*

Действие для команды заполним следующим кодом:

&НаСервере

Процедура ЗагрузитьВалютыНаСервере(фДатаЗагрузки)

*//создаем так называемые определения для веб-сервиса из его WSDL.*

*Определения = Новый*

*WSОпределения("http://www.cbr.ru/DailyInfoWebServ/DailyInfo.asmx?WSDL");*

*//Создаем прокси для обращения к нему.*

*//Создает прокси на основе определения веб-сервиса и связывает его с точкой подключения веб-сервиса.*

*Прокси = Новый WSПрокси(Определения, "http://web.cbr.ru/", "DailyInfo", "DailyInfoSoap");*

*//Получаем тип параметра, который передается в метод GetCursOnDate.*

*ТипWSПараметра =*

*Прокси.ФабрикаXDTO.Пакеты.Получить("http://web.cbr.ru/").Получить("GetCursOnDate");*

*//Создаем параметр на основе типа и заполняем значение параметра On\_Date.*

*WSПараметр = Прокси.ФабрикаXDTO.Создать(ТипWSПараметра);*

*WSПараметр.On\_Date= ДатаЗагрузки;*

```
//Вызываем метод веб-сервиса, записываем результат в переменную
КурсыВалют.
```

```
КурсыВалют = Прокси.GetCursOnDate(WSПараметр);
```

```
//Перебираем таблицу ValuteCursOnDate, каждое значение таблицы
```

```
// добавляем в таблицу на форме (колонки заполняем соответствующими значениями).
```

```
Для Каждого Элемент Из
КурсыВалют.GetCursOnDateResult.diffgram.ValuteData.ValuteCursOnDate Цикл
```

```
НоваяСтрокаТЗ = ТаблицаКурсовВалют.Добавить();
```

```
НоваяСтрокаТЗ.НазваниеВалюты = Элемент.Vname;
```

```
НоваяСтрокаТЗ.Номинал = Элемент.Vnom;
```

```
НоваяСтрокаТЗ.ЦифровойКодВалюты = Элемент.Vcode;
```

```
НоваяСтрокаТЗ.СимвольныйКодВалюты = Элемент.VChCode;
```

```
НоваяСтрокаТЗ.КурсВалюты = Элемент.Vcurs;
```

```
КонецЦикла;
```

```
КонецПроцедуры
```

&НаКлиенте

Процедура ЗагрузитьВалюты(Команда)

```
Если НЕ ЗначениеЗаполнено(ДатаЗагрузки) Тогда
```

```
Сообщить("Не выбрана дата загрузки!", СтатусСообщения.Важное);
```

```
Возврат;
```

```
КонецЕсли;
```

```
ТаблицаКурсовВалют.Очистить();
```

```
ЗагрузитьВалютыНаСервере(ДатаЗагрузки);
```

```
КонецПроцедуры
```

Для большинства методов необходимы параметры (например, дата, на которую необходимо узнать курс валют).

Чтобы узнать какие нужны параметры можно найти соответствующий тип объекта в дереве описания сервиса.

Какую структуру вернет переменная WSПараметр можно посмотреть в отладчике. В рассматриваемом примере метод имеет один параметр. Поэтому структура WSПараметр содержит одно поле (рис. 8.4).

В отладчике можно посмотреть структуру вложенности возвращаемого в переменную КурсыВалют объекта XTDO, для обращения к нему в коде.

Сначала проверяем работоспособность приложения, запустив в обычном режиме (стационарного приложения для ПК). Получение данных в режиме исполнения (в стационарном приложении) – рис. 8.5.

Введя в реквизит «ДатаЗагрузки» дату получения данных, например, текущую дату, и нажав кнопку «Загрузить валюты», получаем в таблице значений курсы валют с сайта ЦБР.

Чтобы проверить работоспособность функциональности в мобильном приложении, в свойствах конфигурации укажем в «Назначении использования» значение «Приложение для мобильной платформы».

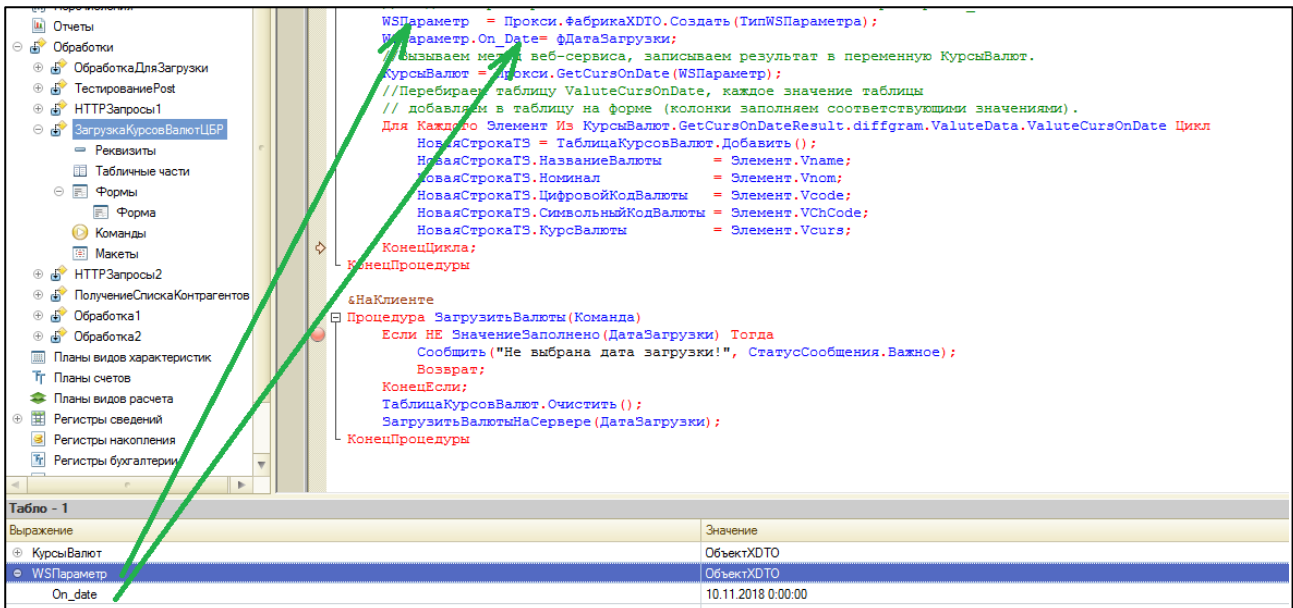


Рис. 8.4. Структура, которую возвращает переменная WSПараметр

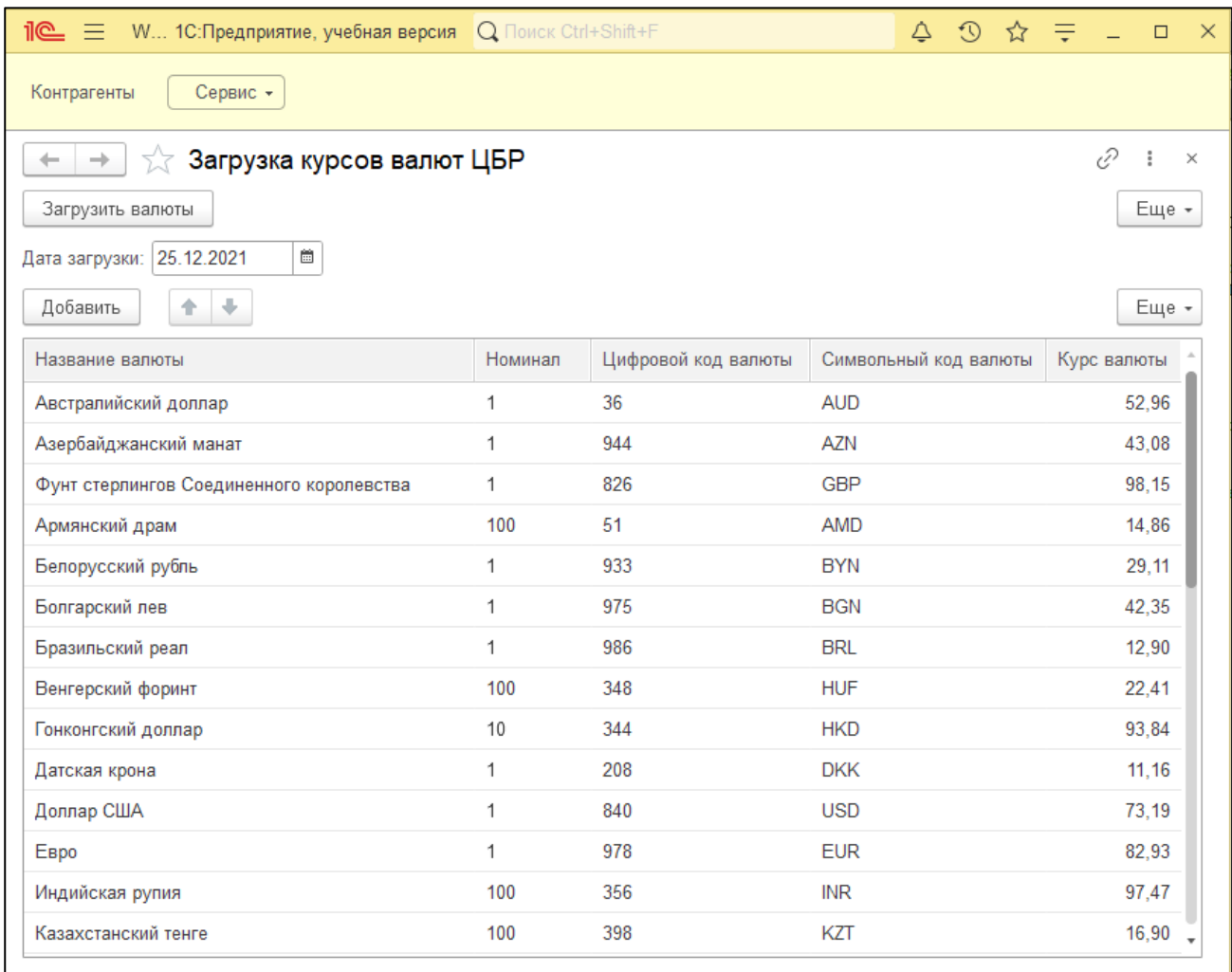


Рис. 8.5. Результат загрузки курсов валют с сайта ЦБР  
Сохраним изменения в информационной базе.

Далее надо опубликовать мобильное приложение. (в меню пункт «Конфигурация | Мобильное приложение | Опубликовать ...»).

Запустив мобильное приложение в эмуляторе получим следующий результат (рис. 8.6).

Реализуем пример №2. Используя тот же сервис <http://www.cbr.ru/development/dws/>, получим новости с сайта Центробанка России. Для этого будем использовать метод `NewsInfo`. Его описание на сайте ЦРБ:

***NewsInfo(FromDate, ToDate)***

Получение новостей сервера

Аргументы:

*FromDate* Дата начала, тип `System.DateTime`

*ToDate* Дата окончания, тип `System.DateTime`

Результат:

XML документ в формате `System.Data.Dataset`, содержащий таблицу [News], таблица содержит поля:

*Doc\_id* – ID документа;

*DocDate* – Дата документа;

*Title* – Заголовок;

*Url* – URL документа.

Создадим новую обработку «ЗагрузкаНовостейСайтаЦБР». Создадим форму обработки. В ней создадим реквизит формы типа таблица значений «ТаблицаНовостей». В ней создадим колонки:

«ID\_Документа», тип Строка;

«ДатаДокумента», тип Строка;

«Заголовок», тип Строка;

«URL\_документа», тип Строка.

Создадим два реквизита формы типа Дата – «ДатаНачала» и «ДатаОкончания».

Все созданные реквизиты, включая таблицу значений, вынесем на форму.

Создадим команду формы «ЗагрузитьНовости».

Вытащим команду на командную панель формы.

Реализуем в модуле формы обработчик команды:

Назва...	Номинал	Цифро...	Симво...
Австр...	1	36	AUD >
Азерб...	1	944	AZN >
Фунт с...	1	826	GBP >
Армян...	100	51	AMD >
Белор...	1	933	BYN >
Болга...	1	975	BGN >
Брази...	1	986	BRL >
Венге...	100	348	HUF >
Гонко...	10	344	HKD >

Рис. 8.6. Результат работы

&НаСервере

Процедура ЗагрузитьНовостиНаСервере(фДатаЗагрузки)

//создаем так называемые определения для веб-сервиса из его

WSDL.

Определения = Новый

WSОпределения("http://www.cbr.ru/DailyInfoWebServ/DailyInfo.asmx?WSDL");

//Создаем прокси для обращения к нему.

//Создает прокси на основе определения веб-сервиса и связывает его с точкой подключения веб-сервиса.

Прокси = Новый WSПрокси(Определения, "http://web.cbr.ru/", "DailyInfo", "DailyInfoSoap");

//Получаем тип параметра, который передается в метод NewsInfo.

ТипWSПараметра =

Прокси.ФабрикаXDTO.Пакеты.Получить("http://web.cbr.ru/").Получить("NewsInfo");

//Создаем параметр на основе типа и заполняем значение параметров FromDate и ToDate.

WSПараметр = Прокси.ФабрикаXDTO.Создать(ТипWSПараметра);

WSПараметр.FromDate= ДатаНачала;

WSПараметр.ToDate= ДатаОкончания;

//Вызываем метод веб-сервиса, записываем результат в переменную КурсыВалют.

Новости = Прокси.NewsInfo(WSПараметр);

//Перебираем таблицу, каждое значение таблицы

// добавляем в таблицу на форме (колонки заполняем соответствующими значениями).

Для Каждого Элемент Из

Новости.NewsInfoResult.diffgram.NewsInfo.News Цикл

НоваяСтрокаТЗ = ТаблицаНовостей.Добавить();

НоваяСтрокаТЗ.ID\_Документа = Элемент.Doc\_id;

НоваяСтрокаТЗ.ДатаДокумента = Элемент.DocDate;

НоваяСтрокаТЗ.Заголовок = Элемент.Title;

НоваяСтрокаТЗ.URL\_документа = Элемент.Url;

КонецЦикла;

КонецПроцедуры

&НаКлиенте

Процедура ЗагрузитьНовости(Команда)

Если НЕ ЗначениеЗаполнено(ДатаНачала) или НЕ ЗначениеЗаполнено(ДатаОкончания) Тогда

Сообщить("Не верно выбран диапазон дат!",

СтатусСообщения.Важное);

Возврат;

КонецЕсли;

```

ТаблицаНовостей.Очистить();
ЗагрузитьНовостиНаСервере(ДатаНачала);
КонецПроцедуры

```

Обработчик команды использует ту же WSDL-описание сервиса, только другой метод сервиса.

Для большинства методов необходимы параметры. Чтобы узнать, какие нужны параметры можно найти соответствующий тип объекта в дереве описания сервиса.

Какую структуру вернет переменная WSПараметр можно также посмотреть в отладчике. В рассматриваемом примере метод имеет два параметра FromDate и ToDate.

В отладчике можно посмотреть структуру вложенности возвращаемого в переменную Новости объекта XDTO, для обращения к нему в коде (рис. 8.6):

Сначала проверяем работоспособность приложения, запустив в обычном режиме (стационарного приложения для ПК). Получение данных в режиме исполнения (в стационарном приложении) представлено рис. 8.7.

```

//Вызываем метод веб-сервиса, записываем результат в переменную КурсыВалют.
Новости = Прокси.NewsInfo(WSПараметр);
//Перебираем таблицу, каждое значение таблицы
// добавляем в таблицу на форме (колонки заполняем соответствующими значениями).
Для Каждого Элемент Из Новости.NewsInfoResult.diffgram.NewsInfo.Цикл
    НоваяСтрокаТЗ = ТаблицаНовостей.Добавить();
    НоваяСтрокаТЗ.ID_документа = Элемент.Doc_id;
    НоваяСтрокаТЗ.ДатаДокумента = Элемент.DocDate;
    НоваяСтрокаТЗ.Заголовок = Элемент.Title;
    НоваяСтрокаТЗ.URL_документа = Элемент.Url;
КонецЦикла;
КонецПроцедуры

```

Выражение	Значение	Тип
Новости	ОбъектXDTO	ОбъектXDTO
NewsInfoResult	ОбъектXDTO	ОбъектXDTO
diffgram	ОбъектXDTO	ОбъектXDTO
NewsInfo	ОбъектXDTO	ОбъектXDTO
News	СписокXDTO	СписокXDTO
Владелец	ОбъектXDTO	ОбъектXDTO
ВладеющееСвойство	News	СвойствоXDTO
schema	ОбъектXDTO	ОбъектXDTO

**Рис. 8.6. Структура вложенности объекта XDTO**

Введя диапазон дат, например, текущего месяца, и, нажав кнопку «Загрузить новости» получим таблицу значений, заполненную ссылками на документы новостей с сайта ЦБР (рис. 8.7).

Чтобы проверить работоспособность функциональности в мобильном приложении, в свойствах конфигурации укажем в «Назначении использования» значение «Приложение для мобильной платформы». Сохраним изменения в информационной базе.

Далее надо опубликовать мобильное приложение. (в меню пункт «Конфигурация | Мобильное приложение | Опубликовать ...»).

Запустив мобильное приложение в эмуляторе мобильного устройства и введя диапазон дат, получим аналогичный результат.



ID документа	Дата документа	Заголовок	URL документа
28599	2021-12-24T20:45:00.127+03:00	Информационно-аналитический материал «О возможных значениях макропруденциальных лимитов по необеспеченным потр...	/analytics/finstab/#_a_131954file
10072	2021-12-24T20:10:50.37+03:00	Информация о ценных бумагах, принимаемых в обеспечение по кредитам Банка России на 27.12.2021	/hd_base/BankPapers/
9973	2021-12-24T20:00:01.763+03:00	Решения Банка России в отношении участников финансового рынка	/rbr/rbr_fr/
28598	2021-12-24T17:50:00.073+03:00	Основные направления развития финансового рынка Российской Федерации на 2022 год и период 2023 и 2024 годов	/about_br/publ/onfinmarket/#_a_131935file
9301	2021-12-24T17:33:26.86+03:00	Реестр паевых инвестиционных фондов	/registries/RSCI/activity_uk_if#a_14190
9295	2021-12-24T17:19:24.15+03:00	Реестр приостановленных лицензий профессиональных участников рынка ценных бумаг	/registries/rcb/#_a_14177
9294	2021-12-24T16:58:13.37+03:00	Единый реестр саморегулируемых организаций в сфере финансового рынка	/registries/sro/#_a_14200
9302	2021-12-24T16:52:24.54+03:00	Реестр зарегистрированных правил доверительного управления ипотечным покрытием	/registries/RSCI/#_a_14187
9282	2021-12-24T16:50:02.577+03:00	Реестр жилищных накопительных кооперативов	/registries/microfinance/#_a_14198
9320	2021-12-24T16:50:02.153+03:00	Государственный реестр сельскохозяйственных кредитных потребительских кооперативов	/registries/microfinance/#_a_14197
10675	2021-12-24T16:50:01.45+03:00	Государственный реестр ломбардов	/registries/microfinance/#_a_33636
9362	2021-12-24T16:50:00.967+03:00	Государственный реестр кредитных потребительских кооперативов	/registries/microfinance/#_a_14196
28597	2021-12-24T16:40:45.767+03:00	О влиянии доминирования на рынке депозитов/платежей на кредитование	/ec_research/analitics/#_a_131934file
28596	2021-12-24T16:25:00.45+03:00	Информационно-аналитический материал «Оценка эффективности макропруденциальных мер Банка России в сегменте необе...	/analytics/finstab/#_a_131931file
28594	2021-12-24T16:25:00.083+03:00	Оценка эффективности макропруденциальной политики Банка России по ограничению необеспеченного потребительского кре...	/ec_research/ser/#_a_131927page
9975	2021-12-24T16:17:33.687+03:00	Ставки группы МИАСР	/hd_base/mkr/mkr_base/
15161	2021-12-24T16:05:56.583+03:00	Экспорт Российской Федерации энергетических товаров	/statistics/macro_itm/svs/#_a_71439

Рис. 8.7. Загрузка новостей в режиме исполнения

Пример №3 для выполнения.

Реализуйте из приложения 1С работу с сервисом проверки правописания Яндекс.Спеллер.

WSDL-документ для SOAP доступен по адресу:

**<http://speller.yandex.net/services/spellservice?WSDL>**.

Создайте новую обработку «ПроверкаНаОшибки» (рис. 8.8). Создайте форму обработки, в ней добавьте два новых строковых реквизита:

«Текст» – тип Строка;

«Ошибки» – типа Строка.

Добавьте новую команду формы «ПроверитьОрфографию».

Примерный код функции вызова вышеуказанного Web-сервиса может иметь следующий вид:

&НаКлиенте

Процедура ПроверитьОтфографию(Команда)

ЭтаФорма.Ошибки =

ПроверитьОрфографиюНаСервере(ЭтаФорма.Текст);

КонецПроцедуры

// Функция на сервере осуществляет проверку орфографии текста.

// Параметры

// ПроверяемыйТекст - Строка - текст для проверки правописания.

//



```

// Возвращаемое значение:
// Строка - слова, в которых допущены ошибки.
//
&НаСервере
Функция ПроверитьОрфографиюНаСервере(ПроверяемыйТекст)
    МестоположениеWSDL =
        "http://speller.yandex.net/services/spellservice?WSDL";
    WSOпределение = Новый
        WSOпределения(МестоположениеWSDL);
    ПространствоИмен = "http://speller.yandex.net/services/spellservice";
    SpellService = Новый WSPрокси(WSOпределение,
ПространствоИмен, "SpellService", "SpellServiceSoap");
    Ответ = SpellService.checkText(ПроверяемыйТекст);
    Ошибки = "";
    Для Каждого Ошибка Из Ответ.error Цикл
        Ошибки = Ошибки + "Ошибка в слове: " + Ошибка.word +
            Символы.ПС;
    КонецЦикла;
    Возврат Ошибки;
КонецФункции // ПроверитьОрфографиюНаСервере()
    
```

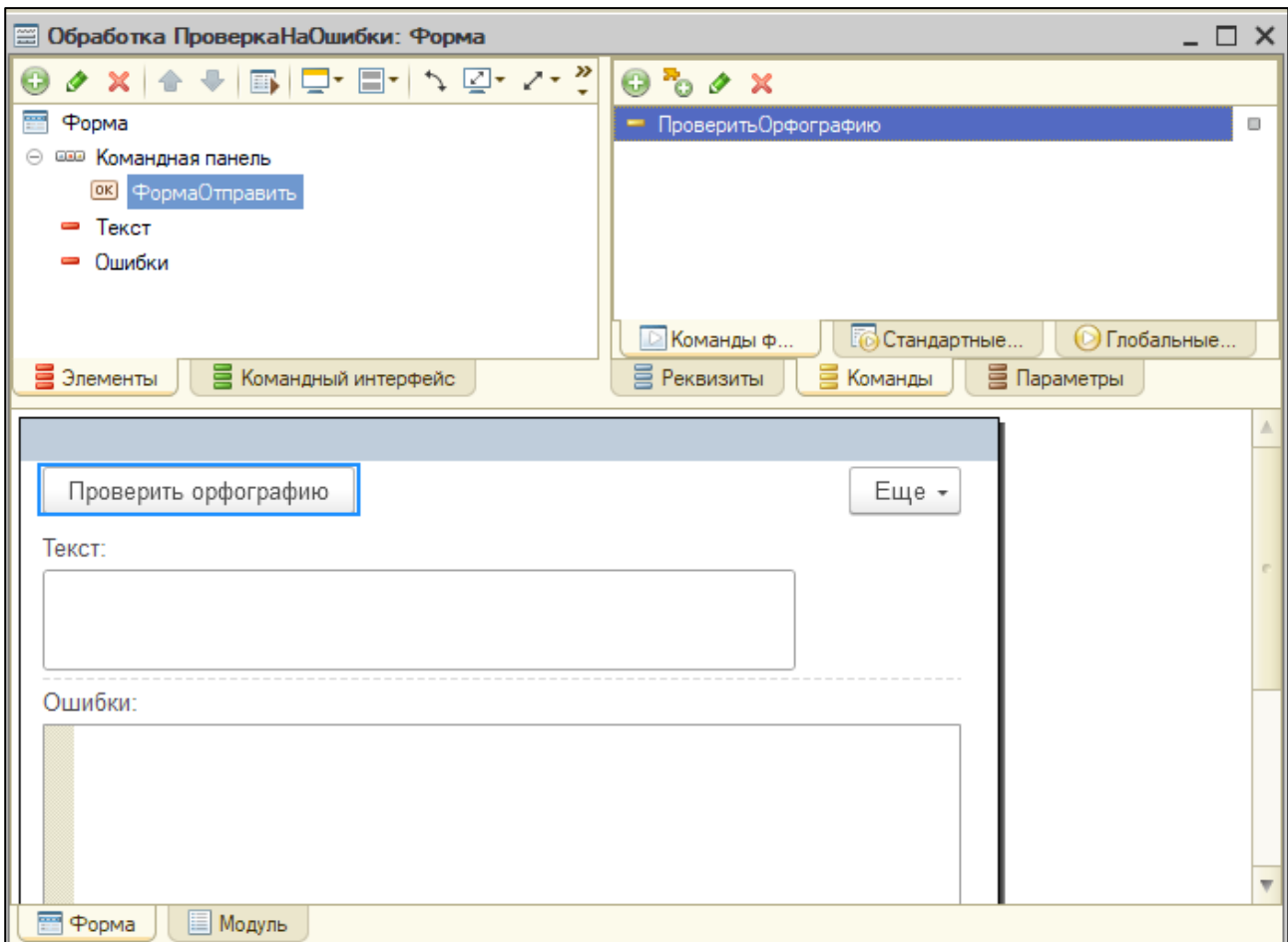
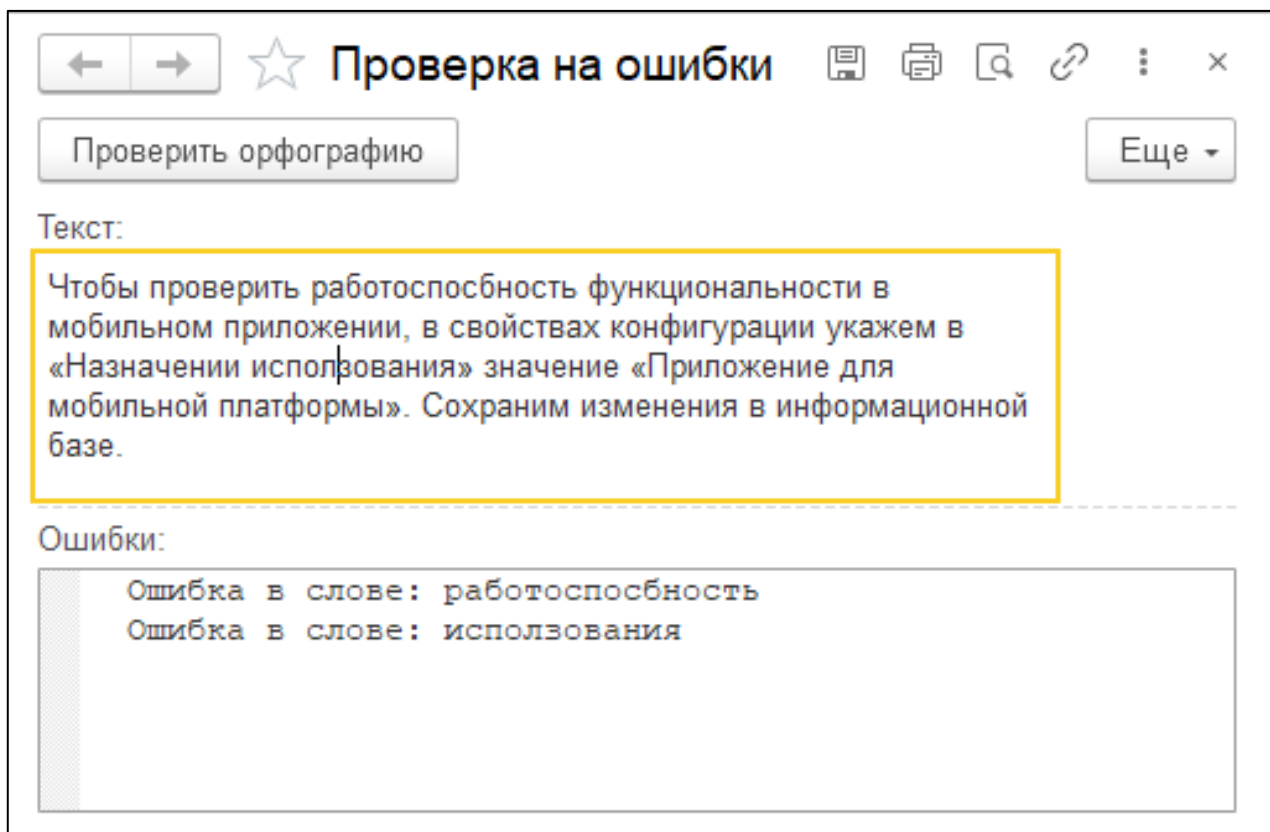


Рис. 8.8. Форма обработки «ПроверкаНаОшибки»

Проверим работоспособность созданного функционала в режиме исполнения.

В поле Текст введем абзац текста и намеренно сделаем в двух словах опечатки. Нажав на кнопку «Проверить орфографию» получим в поле Ошибки список слов, в которых допущены опечатки.

Результат проверки представлен на рис. 8.9.



**Рис. 8.9. Результат проверки текста**

Итак, для работы с веб-сервисами сторонних поставщиков может быть использован такой объект системы «1С:Предприятие» как WS-ссылка. Она представляет собой WSDL-описание веб-сервиса полученное путем импорта из источника на котором расположен веб-сервис. То есть в первую очередь нам надо знать адрес, по которому мы сможем получить WSDL-описание веб-сервиса.

Могут использоваться динамические WS-ссылки. В этом случае WS-ссылка создается программно.

Вместо программного создания объекта WSOпределения мы можем непосредственно в дереве конфигурации создать объект метаданных WS-ссылка. В процессе создания будет выведено окно с предложением указать адрес WSDL определения для его импорта. После этого мы можем в коде ссылаться непосредственно на эту WS-ссылку.

## Заключение

В рамках работы на практических занятиях обучающимися освоены навыки конфигурирования и программирования задач анализа учетных данных и прогнозирования с использованием механизма анализа и прогнозирования данных платформы «1С:Предприятие», в частности поиска ассоциативных групп объектов и событий, построения дерева решений, проведения кластерного анализа, разработке различных запросов и построения сложных аналитических отчетов с использованием языка запросов системы «1С:Предприятие» и системы компоновки данных, навыки по интеграции приложений с помощью Web-интерфейсов, HTTP-сервисов и WS-ссылок, сформированы компетенции по интеграции различных информационных систем и приложений бизнес-аналитики в разнородном IT-ландшафте предприятия.

## Библиографический список

1. <https://its.1c.ru> – сайт Информационно-технологического сопровождение пользователей системы «1С:Предприятие».
2. Профессиональная разработка в системе «1С:Предприятие»: в 2 т. / 2-е изд.– М.: 1С-Публишинг, 2012.–Т.1.–690с.
3. Профессиональная разработка в системе «1С:Предприятие»: в 2 т. / 2-е изд.– М.: 1С-Публишинг, 2012.–Т.2.–683с.
4. Хрусталева Е. Ю. Язык запросов «1С:Предприятия 8».– М.: 1С-Публишинг, 2013.– 369с.
5. Хрусталева Е. Ю. Разработка сложных отчетов в «1С:Предприятии 8». Система компоновки данных. Изд.2.– М.: 1С-Публишинг, 2016.– 485с.