

Учебный курс

Подготовка к аттестации 1С:Специалист по платформе 1С:Предприятие 8.3

Общие, универсальные механизмы,
которые используются для решения
задач всех разделов

Главы 22 – 34

Версия 1.1.1

Оглавление

22. Алгоритм автоматического зачета оплат и/или погашения задолженности при выполнении взаиморасчетов	5
Подведение итогов	8
23. Как выполнить погашение задолженности и зачет аванса по методу FIFO	9
Решение практической задачи	9
Используемые объекты каркасной конфигурации	9
Создание документа «Поступление денег» и регистра накопления «Взаиморасчеты»	10
Обработка проведения для документа «Расходная накладная»	11
Обработка проведения для документа «Поступление денег»	12
Подготовка наборов записей регистра «Взаиморасчеты» (п. 1, 2)	14
Управляемая блокировка данных регистра (п. 3)	14
Запрос (п. 4)	14
Погашение задолженности и создание аванса (п. 5, 6, 7)	15
Проверка промежуточных результатов в режиме «1С:Предприятие»	15
Дополнение процедуры «Обработка проведения» для документа «Расходная накладная»	18
Запрос (п. 4)	20
Зачет аванса и создание задолженности (п. 5, 6, 7)	20
Проверка итоговых результатов в режиме «1С:Предприятие»	21
Подведение итогов	23
24. Задачи с валютами – как выбрать верную методику решения на экзамене в зависимости от условий и как решать задачи с фиксированным перечнем валют	24
Вопрос о точности курса валюты и кратности	25
Задачи с фиксированным перечнем валют	26
Практический пример задачи с фиксированным списком валют	26
Подготовка объектов метаданных	28
Проведение документа «Приходная накладная»	31
Проведение документа «Расходная накладная»	33
Подведем итог	38
25. Как решать задачи с переменным перечнем валют	39
Где и как хранить список валют и курсы валют?	40
Изменения в документах и регистрах для отражения операций в валюте	43
Нужно ли хранить в шапке документа курс валюты?	43

Практический пример задачи с переменным перечнем валют	43
Подготовка объектов метаданных	44
Отражение возникновения задолженности. Проведение документа «Расходная накладная»	46
Подведем итоги	48
26. Как погасить задолженность в разных валютах оплатой в рублях	49
Проведение документа «Поступление денег»	49
Разберем ключевые точки алгоритма	55
Запрос для получения данных по задолженностям в разрезе накладных (п. 3)	55
Алгоритм погашения задолженности (п. 5, 6, 7, 8, 9)	56
Алгоритм формирования движения по авансу (п. 10)	56
Проверка погашения задолженности в режиме «1С:Предприятие»	57
Тестовый пример № 1	58
Тестовый пример № 2	60
Подведем итоги	63
27. Как корректно зачесть аванс в рублях на накладную в валюте	64
Проведение документа «Расходная накладная»	64
Запрос получения данных по авансу (п. 3)	70
Формирование движений по РН «Взаиморасчеты»	70
Проверка зачета аванса в режиме «1С:Предприятие»	71
Тестовый пример № 1	71
Тестовый пример № 2	73
Подведем итоги	76
28. Как выполнить распределение дополнительных затрат	77
Подготовка объектов метаданных	78
Проведение документа «Расходная накладная»	80
Проведение документа «Дополнительные затраты»	82
Разберем ключевые точки алгоритма	84
Подготовка регистра к записи (п. 1)	84
Запрос для получения данных для распределения суммы (п. 2)	84
Заполнение параметров (п. 3)	85
Выполнение запроса по общему итогу, фиксация в переменную общего количества по накладной параметров (п. 4, 5, 6, 7)	86
Обход выборки по номенклатуре и формирование движений (п. 8-12)	86
Проверка распределения суммы дополнительных затрат	86
Подведем итог	90

29. Какие объекты конфигурации следует использовать для хранения произвольных свойств	92
Использование плана видов характеристик	93
Пошаговая инструкция по настройке плана видов характеристик в конфигураторе	100
Заключение	101
30. Как следует организовать хранение значений произвольных свойств объектов в базе	102
Хранение значений характеристик в отдельном регистре сведений	102
Указание принадлежности характеристик	108
Итоги	111
31. Как реализовать использование дополнительных свойств объектов в отчетах, чтобы ускорить разработку на экзамене	112
Использование механизма характеристик в отчетах на СКД	112
Описание характеристик с помощью расширения языка запросов	114
Указание характеристик на уровне объектов метаданных	118
Настройка характеристик для различных вариантов их хранения	122
32. Как программно разместить на форме поля для редактирования характеристик	124
33. Как на экзамене решить практическую задачу – создать отчет на СКД, в котором свойства номенклатуры выводятся отдельными колонками	130
Настройки ПВХ и справочника значений дополнительных свойств	132
Создание и настройка регистра сведений	134
Настройка характеристик справочника «Номенклатура»	139
Разработка отчета с использованием СКД	141
34. Как на экзамене настроить командный интерфейс конфигурации	149
Как при помощи подсистем организовать командный интерфейс?	150
Как в интерфейсе сгруппировать объекты по видам?	154

22. Алгоритм автоматического зачета оплат и/или погашения задолженности при выполнении взаиморасчетов

Взаиморасчеты – это расчеты между контрагентами за поставленные товары, услуги. Взаиморасчеты всегда ведутся в разрезе контрагентов, между которыми произошли расчеты. Очень часто к этой аналитике добавляется дополнительная детализация расчетов: договоры, расчетные документы, сделки, проекты и пр.

В учете движения взаиморасчетов называют уменьшением или увеличением долга (задолженности), а остатки расчетов называют задолженностью или авансом. Рассмотрим виды остатков взаиморасчетов с точки зрения нашей фирмы «Наша организация»:

- **Покупатели должны** фирме «Наша организация» – такой остаток называется задолженностью или долгом
- Фирма «Наша организация» **должна покупателям** – такой остаток называется авансом (от покупателей)
- **Поставщики должны** фирме «Наша организация» – такой остаток называется авансом (поставщикам)
- Фирма «Наша организация» **должна поставщикам** – такой остаток называется задолженностью.

В ряде задач экзамена требуется реализовать автоматический зачет авансов и/или погашение задолженности покупателей или поставщиков (пример задач сборника: 1.9, 1.18, 2.2, 2.5). Вместо покупателей и поставщиков во взаиморасчетах могут участвовать сотрудники предприятия (примеры задач сборника: 2.25, 2.26). Алгоритм решения такой задачи одинаков как для оперативного, так и для бухгалтерского учетов, поэтому приведем его в общем разделе.

Перед тем как приступить к решению практической задачи, разберем алгоритм работы с взаиморасчетами с помощью схем. Для примера возьмем вариант ведения взаиморасчетов с покупателями в разрезе документов расчетов. При отгрузке и оплате от покупателя необходимо автоматически погашать задолженность и зачитывать авансы по методу FIFO (документы, оформленные первыми, погашаются в первую очередь). Метод FIFO будем рассматривать как наиболее часто используемый при автоматическом зачете авансов и/или погашении задолженности.

Итак, оформим несколько сделок с покупателем:

- Накладная 1 от 15.08.18 на сумму 1 000,00 руб
- Накладная 2 от 30.08.18 на сумму 5 000,00 руб
- Накладная 3 от 05.09.18 на сумму 2 500,00 руб.

Эти отгрузки увеличивают долг покупателя перед нашей фирмой на 8 500,00 руб. Движения взаиморасчетов, обозначающие отгрузки, и остатки взаиморасчетов в разрезе документов можно изобразить следующим образом:

	Документ расчетов	Начальный остаток	Увеличение долга	Уменьшение долга	Конечный остаток	Итого долг
Накладная 1 от 15.08.18 1000,00 руб	Накладная 1 от 15.08.18		1000,00		1000,00	1000,00
Накладная 2 от 30.08.18 5000,00 руб	Накладная 2 от 30.08.18		5000,00		5000,00	6000,00
Накладная 3 от 05.09.18 2500,00 руб	Накладная 3 от 05.09.18		2500,00		2500,00	8500,00

Рисунок 1 – Движения и остатки взаиморасчетов, обозначающие отгрузки по трем накладным

Теперь от покупателя поступает оплата в размере 5 000,00 руб. Оплату необходимо распределить по накладным в порядке даты их оформления (погасить задолженность):

	Документ расчетов	Начальный остаток	Увеличение долга	Уменьшение долга	Конечный остаток	Итого долг
Можно погасить 1000,00 руб	Накладная 1 от 15.08.18	1000,00		1000,00	0,00	7500,00
Можно погасить 4000,00 руб	Накладная 2 от 30.08.18	5000,00		4000,00	1000,00	3500,00
Поступление денег 1 от 08.09.18 5000,00 руб	Накладная 3 от 05.09.18	2500,00			2500,00	3500,00

Рисунок 2 – Погашение задолженности покупателя при поступлении оплаты

После первой оплаты долг покупателя уменьшается до 3 500,00 руб. по двум накладным. Остатки взаиморасчетов будут выглядеть так, как указано на Рисунке 3 (пунктирная линия).

Далее от покупателя поступает вторая оплата в размере 5 000,00 руб.. Это поступление денег полностью погасит долг по второй и третьей накладной и еще останется 1 500,00 руб., для которых нет задолженности. Эти деньги станут авансом от покупателя:

Документ расчетов	Начальный остаток	Увеличение долга	Уменьшение долга	Конечный остаток	Итого долг
Накладная 1 от 15.08.18	0,00			0,00	3500,00
Накладная 2 от 30.08.18	1000,00		1000,00	0,00	2500,00
Накладная 3 от 05.09.18	2500,00		2500,00	0,00	0,00
Поступление денег 2 от 15.09.18			1500,00	-1500,00	-1500,00

Можно погасить 1000,00 руб —

Можно погасить 2500,00 руб —

Поступление денег 2 от 15.09.18 5000,00 руб

Нечего погасить (аванс) на 1500,00 руб

Рисунок 3 – Погашение задолженности и возникновение аванса при поступлении оплаты

И снова покупателю выписывается накладная на сумму 3 000,00 руб. Но, так как существует аванс от покупателя, сумма долга по последней накладной автоматически уменьшается на величину остатка аванса (происходит зачет аванса):

Документ расчетов	Начальный остаток	Увеличение долга	Уменьшение долга	Конечный остаток	Итого долг
Накладная 1 от 15.08.18	0,00			0,00	-1500,00
Накладная 2 от 30.08.18	0,00			0,00	-1500,00
Накладная 3 от 05.09.18	0,00			0,00	-1500,00
Поступление денег 2 от 15.09.18	-1500,00	1500,00		0,00	0,00
Накладная 4 от 20.09.18		1500,00		1500,00	1500,00

Можно зачесть 1500,00 руб —

Накладная 4 от 20.09.18 3000,00 руб

Нечего зачесть (задолженность) на 1500,00 руб

Рисунок 4 – Зачет аванса и возникновение задолженности после отгрузки по накладной

По итогам всех отгрузок и поступлений денег долг покупателя составил 1 500,00 руб.

Таким образом, движения взаиморасчетов будут происходить и далее, пока предприятие ведет деятельность с этим покупателем.

Подведение итогов

В пройденном разделе мы получили определение взаиморасчетов и узнали, с какой детализацией они могут вестись. Схематически разобрали алгоритм автоматического погашения задолженности и зачета авансов.

Основные моменты алгоритма:

- В момент проведения отгрузки ищем остаток авансов от покупателя:
 - Если авансы есть – зачитываем их. Расчетным документом при этом указывается документ поступления денег
 - Если авансов нет или недостаточно – формируем новую задолженность. Расчетным документом при этом становится накладная
- В момент проведения оплаты ищем остатки задолженности покупателя:
 - Если задолженность есть – погашаем ее. Расчетным документом при этом указывается накладная
 - Если задолженности нет или недостаточно – формируем новый аванс. Расчетным документом при этом становится документ поступления денег.

В следующем разделе рассмотрим, как реализовать алгоритм автоматического зачета авансов и погашения задолженности на практике.

23. Как выполнить погашение задолженности и зачет аванса по методу FIFO

Решение практической задачи

Постановка задачи:

Компания оказывает услуги своим контрагентам. Взаиморасчеты с контрагентами ведутся в разрезе документов расчетов. Оказание услуг оформляется документом «Расходная накладная», поступление денег – документом «Поступление денег». Погашение задолженности и зачет аванса происходит автоматически, по методу ФИФО.

Для решения поставленной задачи потребуются следующие объекты конфигурации:

- Справочник «Контрагенты»
- Документ «РасходнаяНакладная»
- Документ «ПоступлениеДенег»
- Регистр накопления «Взаиморасчеты».

Настройку будем выполнять по следующему алгоритму:

- Настроить существующие и добавить недостающие объекты метаданных каркасной конфигурации
- Разработать процедуру *ОбработкаПроведения* для документа «ПоступлениеДенег»
- Разработать процедуру *ОбработкаПроведения* для документа «РасходнаяНакладная»
- Проверить результаты в режиме «1С:Предприятие».

Используемые объекты каркасной конфигурации

Справочник «Контрагенты» в каркасной конфигурации уже создан, оставляем его без изменений.

В документ «РасходнаяНакладная» добавим реквизит «Контрагент» (СправочникСсылка.Контрагенты):

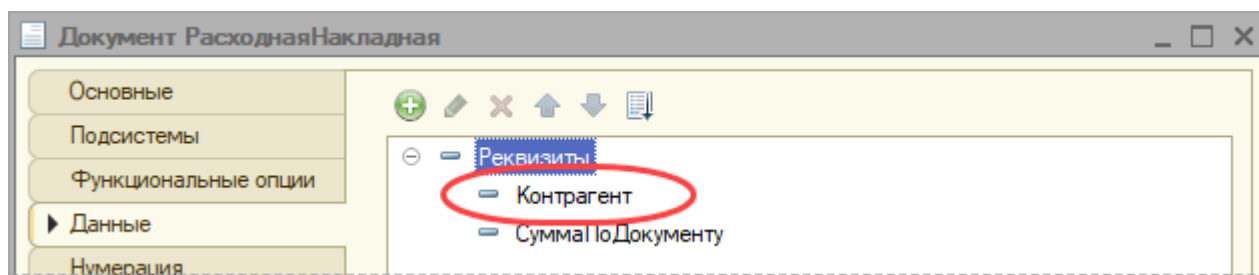


Рисунок 1 – Закладка *Данные* документа «РасходнаяНакладная»

Создание документа «Поступление денег» и регистра накопления «Взаиморасчеты»

Добавим в конфигурацию новый документ «ПоступлениеДенег». В окне редактирования объекта конфигурации на закладке *Данные* создадим структуру документа.

Реквизиты:

- «Контрагент» (СправочникСсылка.Контрагенты)
- «Сумма» (Число 12, 2).

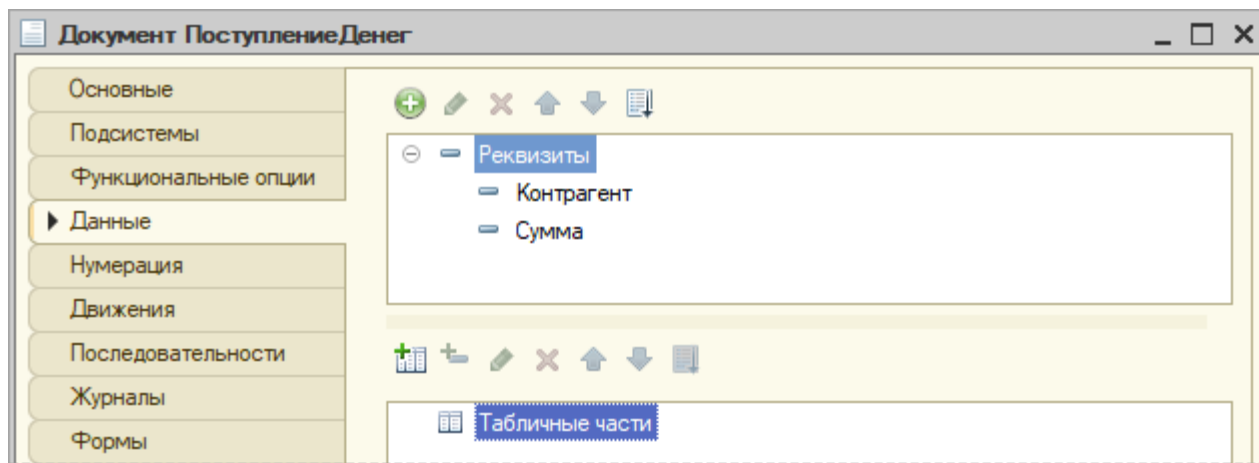


Рисунок 2 – Реквизиты документа «ПоступлениеДенег»

Для хранения движений и итогов по взаиморасчетам с покупателями создадим регистр накопления «Взаиморасчеты» вида *Остатки*. Выбор вида регистра обусловлен тем, что для решения задачи нужно получать информацию об остатках задолженности или авансов от покупателя для их автоматического погашения:

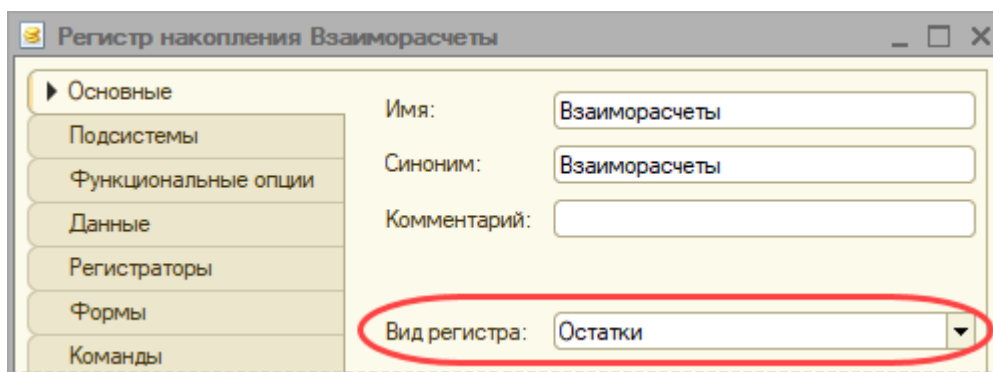


Рисунок 3 – Закладка Основные регистра накопления «Взаиморасчеты»

В окне редактирования объекта конфигурации на закладке *Данные* создадим структуру регистра накопления.

Измерения:

- «Контрагент» (СправочникСсылка.Контрагенты)
- «ДокументРасчетов» (ДокументСсылка.РасходнаяНакладная.ДокументСсылка.ПоступлениеДенег).

Ресурсы:

- «Сумма» (Число 12, 2):

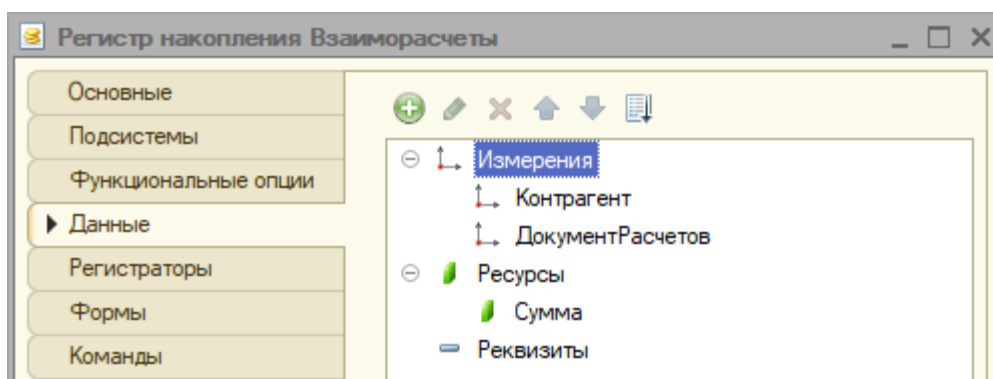


Рисунок 4 – Закладка Данные регистра накопления «Взаиморасчеты»

Обработка проведения для документа «Расходная накладная»

С помощью конструктора движений сформируем движения документа «Расходная накладная» по регистру накопления «Взаиморасчеты» с видом *Приход*. Подробное описание работы с конструктором приведено в блоке «8. Как реализовать контроль остатков по новой методике».

Процедура *ОбработкаПроведения* должна выглядеть следующим образом:

Процедура *ОбработкаПроведения*(Отказ, Режим)

```
Движения.Взаиморасчеты.Записывать = Истина;
Движение = Движения.Взаиморасчеты.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Контрагент = Контрагент;
Движение.ДокументРасчетов = Ссылка;
Движение.Сумма = СуммаПоДокументу;
```

КонецПроцедуры

Текст этой процедуры в дальнейшем будет изменен. На данном этапе решения задачи документ «Расходная накладная» только фиксирует увеличение задолженности покупателя перед компанией, но не выполняет автоматического зачета авансов. Формирование задолженности нужно для промежуточной проверки правильности настройки кода проведения документа «Поступление денег», к которому приступим далее.

Обработка проведения для документа «Поступление денег»

В модуле документа «ПоступлениеДенег» создадим процедуру *ОбработкаПроведения* со следующим текстом:

Процедура *ОбработкаПроведения*(Отказ, РежимПроведения)

```
// 1. Подготовка наборов записей регистра
Движения.Взаиморасчеты.Записывать = Истина;
Движения.Записать();

// 2. Восстановление для свойства набора движений Записывать значения Истина
Движения.Взаиморасчеты.Записывать = Истина;

// 3. Управляемая блокировка данных регистра
Блокировка = Новый БлокировкаДанных;
ЭлементБлокировки = Блокировка.Добавить("РегистрНакопления.Взаиморасчеты");
ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;
ЭлементБлокировки.УстановитьЗначение("Контрагент", Контрагент);
Блокировка.Заблокировать();

// 4. Запрос к остаткам задолженности покупателя
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|     ВзаиморасчетыОстатки.ДокументРасчетов,
|     ВзаиморасчетыОстатки.СуммаОстаток
```

```

|ИЗ
|      РегистрНакопления.Взаиморасчеты.Остатки(&МоментВремени, Контрагент = &Контрагент) КАК
ВзаиморасчетыОстатки
|ГДЕ
|      ВзаиморасчетыОстатки.СуммаОстаток > 0
|
|УПОРЯДОЧИТЬ ПО
|      ВзаиморасчетыОстатки.ДокументРасчетов.МоментВремени";

Запрос.УстановитьПараметр("Контрагент", Контрагент);
Запрос.УстановитьПараметр("МоментВремени", МоментВремени());

РезультатЗапроса = Запрос.Выполнить();

// 5. Определение переменной для отслеживания распределения денег по остаткам задолженности
СуммаЗачетуИтого = Сумма;

// 6. Обход выборки из результата запроса по остаткам задолженности
//      в каждой строке выборки хранится документ с остатком задолженности
//      всю сумму поступивших денег нужно распределить по этим документам
ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
Пока ВыборкаДетальныеЗаписи.Следующий() И СуммаЗачетуИтого > 0 Цикл

    //определение суммы конкретной записи
    //она должна быть минимальной из непогашенной части поступивших денег и остатка долга по
документу
    СуммаЗачетуДок = Мин(СуммаЗачетуИтого, ВыборкаДетальныеЗаписи.СуммаОстаток);

    //добавление движения в регистр накопления
    Движение = Движения.Взаиморасчеты.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Контрагент = Контрагент;
    Движение.ДокументРасчетов = ВыборкаДетальныеЗаписи.ДокументРасчетов;
    Движение.Сумма = СуммаЗачетуДок;

    //уменьшение суммы поступивших денег на ту часть, которая погасила долг текущей строки
выборки
    СуммаЗачетуИтого = СуммаЗачетуИтого - СуммаЗачетуДок;

КонецЦикла;

// 7. В случае, когда долгов не хватило для погашения всей суммы поступивших денег,
//      формирование записи аванса
Если СуммаЗачетуИтого > 0 Тогда

    Движение = Движения.Взаиморасчеты.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Контрагент = Контрагент;
    Движение.ДокументРасчетов = Ссылка;
    Движение.Сумма = СуммаЗачетуИтого;

```

КонецЕсли;

КонецПроцедуры

Прокомментируем основные моменты кода процедуры *ОбработкаПроведения*.

Подготовка наборов записей регистра «Взаиморасчеты» (п. 1, 2)

Выполним очистку существующих движений документа, принудительно записав пустой набор в регистр накопления «Взаиморасчеты». После очистки установим для свойства *Записывать* набора движений по регистру «Взаиморасчеты» значение *Истина*.

Управляемая блокировка данных регистра (п. 3)

Установим признак необходимости блокировки данных регистра накопления «Взаиморасчеты». Сделаем это до выполнения запроса, чтобы гарантированно получить неизменные остатки задолженности. Подробное объяснение темы управляемых блокировок приведено в блоке «**11. Какие навыки в использовании управляемых блокировок потребуются на экзамене**».

Запрос (п. 4)

Запрос выбирает из виртуальной таблицы регистра накопления «Взаиморасчеты.Остатки» информацию о задолженности конкретного покупателя. Для виртуальной таблицы устанавливаем параметры:

- «МоментВремени» – значение параметра = *МоментВремени*
- «Контрагент» – значение должно быть равным реквизиту документа «Контрагент».

Так как документ «РасходнаяНакладная» формирует приход в регистр накопления «Взаиморасчеты», то остаток задолженности будет положительным. Установим в запросе условие по полю «СуммаОстаток». Значение этого поля должно быть больше 0.

Упорядочим результат запроса по полю «ДокументРасчетов.МоментВремени», чтобы реализовать требование задачи выполнять погашение задолженности по методу ФИФО.

Погашение задолженности и создание аванса (п. 5, 6, 7)

Результатом выполнения запроса будет выборка документов, по которым у контрагента есть остаток задолженности. Нужно распределить всю сумму поступивших денег на эти документы.

При этом общий остаток долга может быть как больше, так и меньше суммы денег, указанных в документе «ПоступлениеДенег». Для того чтобы распределить поступившую оплату правильно, создадим переменную «СуммаКЗачетуИтого» и запишем в нее сумму документа. По мере погашения задолженности будем уменьшать ее на сумму закрытого остатка долга, пока она не обнулится.

Погашение задолженности происходит в цикле по строкам выборки из результата запроса. Для каждой строки определяем сумму, которую можно зачесть. Для этого выбираем минимальное из двух чисел: значение переменной «СуммаКЗачетуИтого» и остатка долга из регистра накопления «Взаиморасчеты», это «СуммаОстаток». Таким образом, проконтролируем:

- 1) что по конкретному документу не погашено больше, чем должны
- 2) что общая сумма погашения задолженности не больше, чем нераспределенная сумма документа.

После добавления записи с видом *Расход* в регистр накопления «Взаиморасчеты» уменьшаем переменную «СуммаКЗачетуИтого» на сумму этой записи.

Перебор строк выборки продолжаем до тех пор, пока переменная «СуммаКЗачетуИтого» больше нуля.

Если строки выборки закончились, а переменная «СуммаКЗачетуИтого» не равна нулю, значит, оплаты поступило больше, чем существовало задолженности покупателя. Остаток денег отнесем на аванс. Для этого сформируем запись с видом *Расход* в регистр накопления «Взаиморасчеты» и в качестве документа расчетов укажем ссылку на текущий документ «ПоступлениеДенег».

Погашение задолженности и создание аванса выполнено. Проверим решение этой части задачи в режиме «1С:Предприятие».

Проверка промежуточных результатов в режиме «1С:Предприятие»

В пользовательском режиме создадим документы «Расходная накладная» так, как приведено в первой таблице раздела:

← →

☆

Расходная накладная 000000001 от 15.08.2018 12:00:00

Провести и закрыть

Записать

Провести

Номер:

000000001

Дата:

15.08.2018 12:00:00

📅

Контрагент:

Купим всё

Сумма по документу:

1 000,00

📊

Добавить

N	Номенклатура
---	--------------

Рисунок 5 – Пример документа «Расходная накладная» в режиме «1С:Предприятие»

← →

☆

Расходная накладная 000000002 от 30.08.2018 12:00:00

Провести и закрыть

Записать

Провести

Номер:

000000002

Дата:

30.08.2018 12:00:00

📅

Контрагент:

Купим всё

Сумма по документу:

5 000,00

📊

Добавить

N	Номенклатура
---	--------------

Рисунок 6 – Пример документа «Расходная накладная» в режиме «1С:Предприятие»

← → ☆ **Расходная накладная 000000003 от 05.09.2018 12:00:00**

Провести и закрыть Записать Провести

Номер: 000000003

Дата: 05.09.2018 12:00:00

Контрагент: Купим всё

Сумма по документу: 2 500,00

Добавить

N	Номенклатура
---	--------------

Рисунок 7 – Пример документа «Расходная накладная» в режиме «1С:Предприятие»

Создадим один документ «Поступление денег»:

← → ☆ **Поступление денег 000000001 от 08.09.2018 12:00:00**

Провести и закрыть Записать Провести

Номер: 000000001

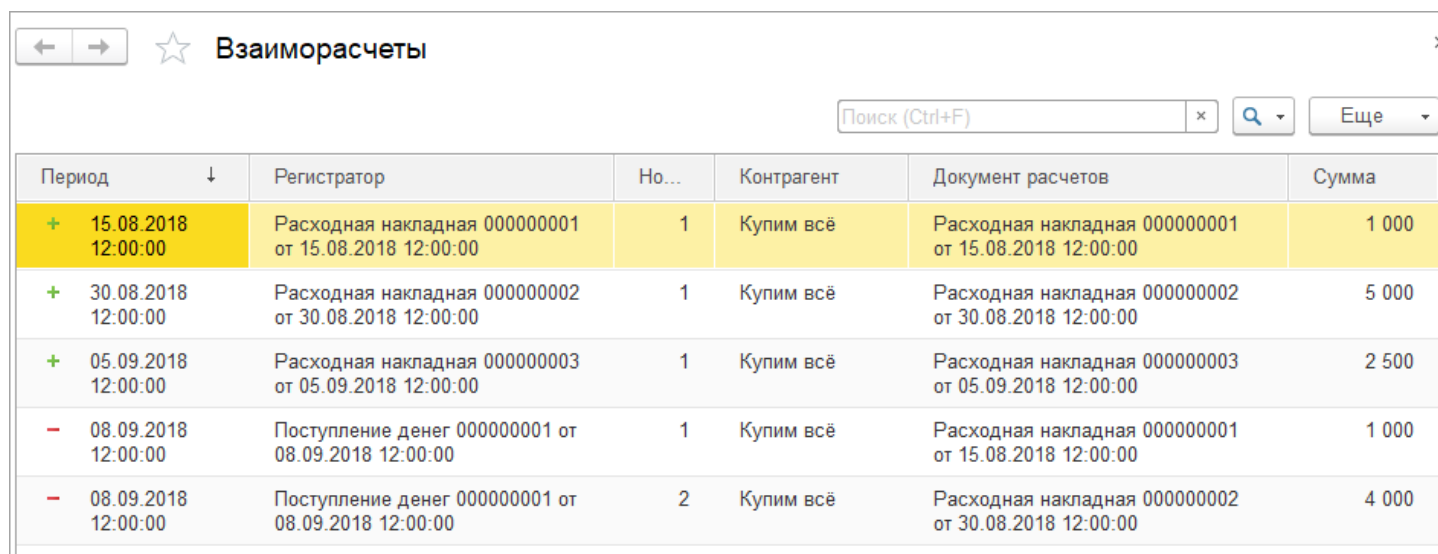
Дата: 08.09.2018 12:00:00

Контрагент: Купим всё

Сумма: 5 000

Рисунок 8 – Пример документа «Поступление денег» в режиме «1С:Предприятие»

Если в конфигурации все сделано правильно, то в регистре накопления «Взаиморасчеты» должны появиться следующие записи:



Период ↓	Регистратор	Но...	Контрагент	Документ расчетов	Сумма
+ 15.08.2018 12:00:00	Расходная накладная 000000001 от 15.08.2018 12:00:00	1	Купим всё	Расходная накладная 000000001 от 15.08.2018 12:00:00	1 000
+ 30.08.2018 12:00:00	Расходная накладная 000000002 от 30.08.2018 12:00:00	1	Купим всё	Расходная накладная 000000002 от 30.08.2018 12:00:00	5 000
+ 05.09.2018 12:00:00	Расходная накладная 000000003 от 05.09.2018 12:00:00	1	Купим всё	Расходная накладная 000000003 от 05.09.2018 12:00:00	2 500
- 08.09.2018 12:00:00	Поступление денег 000000001 от 08.09.2018 12:00:00	1	Купим всё	Расходная накладная 000000001 от 15.08.2018 12:00:00	1 000
- 08.09.2018 12:00:00	Поступление денег 000000001 от 08.09.2018 12:00:00	2	Купим всё	Расходная накладная 000000002 от 30.08.2018 12:00:00	4 000

Рисунок 9 – Результаты проведения документов в режиме «1С:Предприятие»

Дополнение процедуры «Обработка проведения» для документа «Расходная накладная»

Откорректируем процедуру *ОбработкаПроведения* документа «Расходная накладная» по регистру накопления «Взаиморасчеты». Ранее в ней был создан код для отражения увеличения задолженности покупателя перед предприятием. Теперь сделаем, чтобы при проведении «Расходной накладной» автоматически зачитывались ранее полученные авансы:

Процедура *ОбработкаПроведения*(Отказ, РежимПроведения)

```
// 1. Подготовка наборов записей регистра
Движения.Взаиморасчеты.Записывать = Истина;
Движения.Записать();

// 2. Восстановление для свойства набора движений Записывать значения Истина
Движения.Взаиморасчеты.Записывать = Истина;

// 3. Управляемая блокировка данных регистра
Блокировка = Новый БлокировкаДанных;
ЭлементБлокировки = Блокировка.Добавить("РегистрНакопления.Взаиморасчеты");
ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;
ЭлементБлокировки.УстановитьЗначение("Контрагент", Контрагент);
Блокировка.Заблокировать();

// 4. Запрос к остаткам авансов от покупателя
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
```

```

|      ВзаиморасчетыОстатки.ДокументРасчетов,
|      ВзаиморасчетыОстатки.СуммаОстаток * -1 КАК СуммаОстаток
|ИЗ
|      РегистрНакопления.Взаиморасчеты.Остатки(&МоментВремени, Контрагент = &Контрагент) КАК
ВзаиморасчетыОстатки
|ГДЕ
|      ВзаиморасчетыОстатки.СуммаОстаток < 0
|
|УПОРЯДОЧИТЬ ПО
|      ВзаиморасчетыОстатки.ДокументРасчетов.МоментВремени";

Запрос.УстановитьПараметр("Контрагент", Контрагент);
Запрос.УстановитьПараметр("МоментВремени", МоментВремени());

РезультатЗапроса = Запрос.Выполнить();

// 5. Определение переменной для отслеживания распределения денег по остаткам задолженности
СуммаКЗачетуИтого = СуммаПоДокументу;

// 6. Обход выборки из результата запроса по остаткам авансов
//      в каждой строке выборки хранится документ с авансом
//      всю сумму отгрузки нужно распределить по этим документам
ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
Пока ВыборкаДетальныеЗаписи.Следующий() И СуммаКЗачетуИтого > 0 Цикл

    //определение суммы конкретной записи
    //она должна быть минимальной из непогашенной части отгрузки и остатка аванса по документу
    СуммаКЗачетуДок = Мин(СуммаКЗачетуИтого, ВыборкаДетальныеЗаписи.СуммаОстаток);

    //добавление движения в регистр накопления
    Движение = Движения.Взаиморасчеты.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Контрагент = Контрагент;
    Движение.ДокументРасчетов = ВыборкаДетальныеЗаписи.ДокументРасчетов;
    Движение.Сумма = СуммаКЗачетуДок;

    //уменьшение суммы отгрузки на сумму зачета аванса текущей строки выборки
    СуммаКЗачетуИтого = СуммаКЗачетуИтого - СуммаКЗачетуДок;

КонецЦикла;

// 7. В случае, когда авансов не хватило для погашения всей суммы отгрузки,
//      формирование записи долга
Если СуммаКЗачетуИтого > 0 Тогда

    Движение = Движения.Взаиморасчеты.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Контрагент = Контрагент;
    Движение.ДокументРасчетов = Ссылка;
    Движение.Сумма = СуммаКЗачетуИтого;

```

КонецЕсли;

КонецПроцедуры

Код процедуры *ОбработкаПроведения* документа «РасходнаяНакладная» полностью идентичен процедуре *ОбработкаПроведения* документа «ПоступлениеДенег», только в запросе выбираются отрицательные остатки регистра накопления «Взаиморасчеты». Прокомментируем только отличия кода.

Запрос (п. 4)

Запрос выбирает из виртуальной таблицы регистра накопления «Взаиморасчеты.Остатки» информацию об авансах от конкретного покупателя. Для виртуальной таблицы устанавливаем параметры:

- «МоментВремени» – значение параметра = *МоментВремени*
- «Контрагент» – значение должно быть равным реквизиту документа «Контрагент».

Так как документ «ПоступлениеДенег» формирует расход в регистр накопления «Взаиморасчеты», то остаток аванса будет отрицательным. Установим в запросе условие по полю «СуммаОстаток». Значение этого поля должно быть меньше нуля.

Для того, чтобы в коде процедуры не работать с отрицательными числами, умножим значение поля «СуммаОстаток» на -1 .

Упорядочим результат запроса по полю «ДокументРасчетов.МоментВремени», чтобы реализовать требование задачи выполнять погашение задолженности по методу FIFO.

Зачет аванса и создание задолженности (п. 5, 6, 7)

Результатом выполнения запроса будет выборка документов, по которым у контрагента есть остаток аванса. Нужно распределить всю сумму отгрузки на эти документы. Действуем по тому же алгоритму, который использовали при погашении задолженности.

Создадим переменную «СуммаКЗачетуИтого» и запишем в нее сумму документа. По мере зачета авансов будем уменьшать ее на сумму закрытого аванса, пока она не обнулится.

Зачет аванса происходит в цикле по строкам выборки из результата запроса. Для каждой строки определяем сумму, которую можно зачесть. Для этого выбираем минимальное из двух чисел: значения переменной «СуммаКЗачетуИтого» и остатка долга из регистра накопления «Взаиморасчеты» – это «СуммаОстаток».

После добавления записи с видом *Приход* в регистр накопления «Взаиморасчеты» уменьшаем переменную «СуммаКЗачетуИтого» на сумму этой записи.

Перебор строк выборки продолжаем до тех пор, пока переменная «СуммаКЗачетуИтого» больше нуля.

Если строки выборки закончились, а переменная «СуммаКЗачетуИтого» не равна нулю, значит, отгрузки больше, чем существовало авансов от покупателя. Остаток суммы отгрузки отнесем на задолженность. Для этого сформируем запись с видом *Приход* в регистр накопления «Взаиморасчеты» и в качестве документа расчетов укажем ссылку на текущий документ «РасходнаяНакладная».

Зачет аванса и создание задолженности выполнено. Проверим решение этой части задачи в режиме «1С:Предприятие».

Проверка итоговых результатов в режиме «1С:Предприятие»

В пользовательском режиме создадим документ «Поступление денег» так, как приведено в третьей таблице раздела:

← → ☆ Поступление денег 000000002 от 15.09.2018 12:00:00

Провести и закрыть Записать Провести

Номер: 000000002

Дата: 15.09.2018 12:00:00

Контрагент: Купим всё

Сумма: 5 000

Рисунок 10 – Пример документа «Поступление денег» в режиме «1С:Предприятие»

Создадим последний документ «Расходная накладная»:

← →
☆
Расходная накладная 000000004 от 20.09.2018 12:00:00

Провести и закрыть
Записать
Провести

Номер:

Дата:

Контрагент:

Сумма по документу:

Добавить

N	Номенклатура

Рисунок 11 – Пример документа «Расходная накладная» в режиме «1С:Предприятие»

Результатом добавления этих двух документов должны стать следующие записи в регистр накопления «Взаиморасчеты»:

←

→

☆

Взаиморасчеты

×

Поиск (Ctrl+F)

×

Q

Еще

▼

Период	↓	Регистратор	Но...	Контрагент	Документ расчетов	Сумма	
1.	+	15.08.2018 12:00:00	Расходная накладная 000000001 от 15.08.2018 12:00:00	1	Купим всё	Расходная накладная 000000001 от 15.08.2018 12:00:00	1 000
	+	30.08.2018 12:00:00	Расходная накладная 000000002 от 30.08.2018 12:00:00	1	Купим всё	Расходная накладная 000000002 от 30.08.2018 12:00:00	5 000
	+	05.09.2018 12:00:00	Расходная накладная 000000003 от 05.09.2018 12:00:00	1	Купим всё	Расходная накладная 000000003 от 05.09.2018 12:00:00	2 500
2.	-	08.09.2018 12:00:00	Поступление денег 000000001 от 08.09.2018 12:00:00	1	Купим всё	Расходная накладная 000000001 от 15.08.2018 12:00:00	1 000
	-	08.09.2018 12:00:00	Поступление денег 000000001 от 08.09.2018 12:00:00	2	Купим всё	Расходная накладная 000000002 от 30.08.2018 12:00:00	4 000
3.	-	15.09.2018 12:00:00	Поступление денег 000000002 от 15.09.2018 12:00:00	1	Купим всё	Расходная накладная 000000002 от 30.08.2018 12:00:00	1 000
	-	15.09.2018 12:00:00	Поступление денег 000000002 от 15.09.2018 12:00:00	2	Купим всё	Расходная накладная 000000003 от 05.09.2018 12:00:00	2 500
	-	15.09.2018 12:00:00	Поступление денег 000000002 от 15.09.2018 12:00:00	3	Купим всё	Поступление денег 000000002 от 15.09.2018 12:00:00	1 500
4.	+	20.09.2018 12:00:00	Расходная накладная 000000004 от 20.09.2018 12:00:00	1	Купим всё	Поступление денег 000000002 от 15.09.2018 12:00:00	1 500
	+	20.09.2018 12:00:00	Расходная накладная 000000004 от 20.09.2018 12:00:00	2	Купим всё	Расходная накладная 000000004 от 20.09.2018 12:00:00	1 500

Рисунок 12 – Результаты проведения документов в режиме «1С:Предприятие»

1. Движения документов «Расходная накладная» № 1, № 2 и № 3 – возникновение задолженности.
2. Движения документа «Поступление денег» № 1 – погашение задолженности. «Расходная накладная» № 1 погашена полностью, «Расходная накладная» № 2 погашена частично.
3. Движения документа «Поступление денег» № 2. Погашение остатка задолженности по «Расходная накладная» № 2, полное погашение «Расходная накладная» № 3 и возникновение аванса.
4. Движения документа «Расходная накладная» № 4 – зачет аванса по «Поступление денег» № 2 и возникновение новой задолженности покупателя.

Задача выполнена.

Подведение итогов

В этом разделе мы реализовали алгоритм автоматического зачета оплаты и/или погашения задолженности покупателя на практике. Для этого:

- Определили, какие объекты метаданных нужны для решения практической задачи и выполнили их настройку
- Реализовали алгоритмы проведения документов, формирующих движение взаиморасчетов:
 - «Расходная накладная» – возникновение задолженности и зачет авансов от покупателя
 - «Поступление денег» – погашение задолженности по накладным и возникновение аванса.

В следующем разделе приступим к изучению операций в валюте.

24. Задачи с валютами – как выбрать верную методику решения на экзамене в зависимости от условий и как решать задачи с фиксированным перечнем валют

В сборнике задачи с валютой встречаются в разделах оперативного учета (см. 1.27, 1.41, 1.42), бухгалтерского учета (2.6) и даже в разделе по сложным периодическим расчетам (3.46, 3.47, 3.48, 3.49). Согласно информации от сдававших экзамен, в билетах такие задачи также присутствуют.

Формулировки могут быть следующие:

- Документ «Расходная накладная» **выставляется в своей валюте**. Оплаты производятся в рублях. Одна оплата может относиться к нескольким накладным. **Курс валюты** для закрытия задолженности берется **на дату оплаты...**
- Весь учет ведется одновременно в **2 валютах: рубли и доллары**. При проведении документов **курс указывается непосредственно в самом документе...**
- Взаиморасчеты ведутся в разрезе контрагентов и договоров в **валюте взаиморасчетов...** В каждом договоре может быть указана **своя валюта взаиморасчетов...**
- Начисление зарплаты сотрудникам ведется одновременно в **двух валютах (в рублях и долларах)...**

Если на экзамене досталась задача с валютой, то в первую очередь нужно ответить на вопрос: является ли по условию задачи перечень используемых валют фиксированным или он заранее неизвестен? Исходя из ответа на этот вопрос все задачи с валютой подразделяются на два типа. Для каждого типа применяется своя методика решения:



Рисунок 1 – Типы задач с валютой

Далее будем разбирать методику решения задач каждого типа, но сначала определимся с одним вопросом, общим для обоих типов задач, т.к. в обоих типах задач нужно будет выполнять пересчет сумм из одной валюты в другую по курсу валюты.

Вопрос о точности курса валюты и кратности

В типовых конфигурациях 1С значения курсов валют задаются с точностью до 4 знаков после запятой. Если курс валюты по отношению к рублю очень низкий (точность составляет меньше четырех знаков после запятой), то кроме курса еще указывается кратность.

Кратность обозначает за сколько единиц иностранной валюты указан курс. Например, кратность для доллара равна единице, т.к. 1 доллар стоит, например, 65 рублей. Для японской иены кратность равна 100, т.е. 100 японских иен стоят, например, 58,7379 рубля. 1 иена стоит 0,587379 рубля. Еще пример: кратность для узбекского сума равна 10 000.

В Каркасной конфигурации для хранения курсов валют уже создан регистр сведения «Курсы валют» с ресурсом *Курс* (Число 5, 2). Ресурс для кратности не предусмотрен. На экзамене так и оставим: не будем увеличивать точность у ресурса *Курса* до 4 знаков и заводить ресурс для кратности.

В реальной практике будем делать иначе: точность курса валюты будем устанавливать 4 знака после запятой. Обязательно будем предусматривать ресурс для хранения кратности валюты (например, в типовой конфигурации «1С:Бухгалтерия» Кратность – это Число (10,0)).

Задачи с фиксированным перечнем валют

В таких задачах в условии говорится, что учет ведется, например, в двух валютах: рубли и доллары. Или в 3 валютах: рубли, доллары и евро. Это задачи с так называемым двухвалютным или трехвалютным учетом.

Можно провести аналогию с типовыми конфигурациями 1С:УПП или 1С:ERP, в которых регламентированный учет ведется в валюте регламентированного учета, а управленческий учет – в валюте управленческого учета. Валюты регламентированного и управленческого учета устанавливаются в константах.

В экзаменационных задачах нужно реализовать упрощенный вариант такого учета: не нужно предоставлять пользователю возможность выбора валют учета. Валюты явно указаны в условии задачи, поэтому константы заводить не будем. По части курса валюты в экзаменационных задачах, как правило, сказано, что он указывается в документе.

Цель учета в дополнительной валюте – получение отчетности не только в национальной валюте, но и в дополнительной. Причем суммы в валюте интересны не по курсу на дату отчета, а, как правило, по курсу на дату совершения хозяйственной операции.

Какие тут могут быть варианты реализации? При получении данных для отчета делать «на лету» пересчет сумм всех нужных хозяйственных операций в дополнительную валюту по курсу на дату совершения операции? Естественно, такое решение не является эффективным, потому что нужно будет обрабатывать таблицы движений регистров.

Поэтому применяется другое решение – в регистры добавляются ресурсы для хранения сумм в дополнительных валютах учета: *Сумма в долларах*, *Сумма в евро*. При таком подходе для получения отчетов можно будет использовать остатки и обороты регистров по этим ресурсам.

Практический пример задачи с фиксированным списком валют

Постановка задачи:

Компания занимается оптовой торговлей. Приход товаров оформляется документом «Приходная накладная». Расход товаров оформляется документом «Расходная накладная».

Весь учет в компании ведется одновременно в 2 валютах: рублях и долларах. При проведении документов курс указывается непосредственно в самом документе. Возникновение курсовых разниц на себестоимость при продаже не предполагается.

Складской учет товаров не ведется. Себестоимость списания рассчитывается как средняя по номенклатурной позиции.

Необходимо создать отчет по продажам товаров за период. Продажи с 01.09.2018 по 31.10.2018.

Номенклатура	Кол-во	Себест-ть руб./дол.	Продажа руб./дол.	Прибыль руб./дол.
Йогурт «Малина»	20	800 12,31	1400 20	600 7,69
Крупа манная	10	300 4,62	400 5,71	100 1,09

Прибыль по каждой валюте рассчитывается как «Сумма продаж» – «Себестоимость».

Подход к решению задачи:

При поступлении товара себестоимость в рублях – это сумма закупки, указанная в документе «Приходная накладная». Себестоимость в долларах будем рассчитывать исходя из себестоимости в рублях по курсу доллара, указанному в документе.

При продаже сумма продажи в рублях для каждого товара указывается в документе «Расходная накладная». Сумму продажи в долларах будем рассчитывать из суммы продажи в рублях по курсу из документа.

Себестоимость списания в рублях при продаже будем рассчитывать как среднюю по номенклатурной позиции. Примеры такого расчета уже разбирали в главах «Проблема копеек» и «9. Как выполнить контроль остатков по старой методике».

Как рассчитывать себестоимость списания в долларах? Что означает загадочная фраза в условии: «Возникновение курсовых разниц на себестоимость при продаже не предполагается»?

Разберем на примере.

Поступил товар в количестве 3 шт на сумму 200 рублей. Предположим, на момент оформления товара курс был 60 рублей. Сумма прихода в долларах составила $200 / 60 = 3,33$ долларов.

На следующий день товар решили продать. Курс доллара изменился и составлял 50 рублей. Себестоимость товаров в рублях как была 200 рублей, так и осталась. Но при продаже себестоимость списания в валюте, если пересчитать по новому курсу, составит $200 / 50 = 4$ доллара. Т.е. в валюте себестоимость товара увеличилась на 0,67 доллара из-за падения курса. Это и есть курсовая разница. Она может быть как положительной, так и отрицательной.

Фраза «Возникновение курсовых разниц не предполагается» говорит о том, что не нужно при продаже рассчитывать курсовые разницы. Чтобы при списании себестоимости в долларах не возникали

курсовые разницы, ее не нужно рассчитывать по курсу от себестоимости в рублях. Будем выполнять расчет по формуле, аналогичный как для расчета себестоимости в рублях, но только остаток суммы будем брать в долларах.

Переходим к практическому решению задачи.

Подготовка объектов метаданных

Документы «Приходная накладная», «Расходная накладная» в Каркасной конфигурации есть. В них нам понадобится добавить реквизит шапки *Курс доллара*. Тип – Число (5,2):

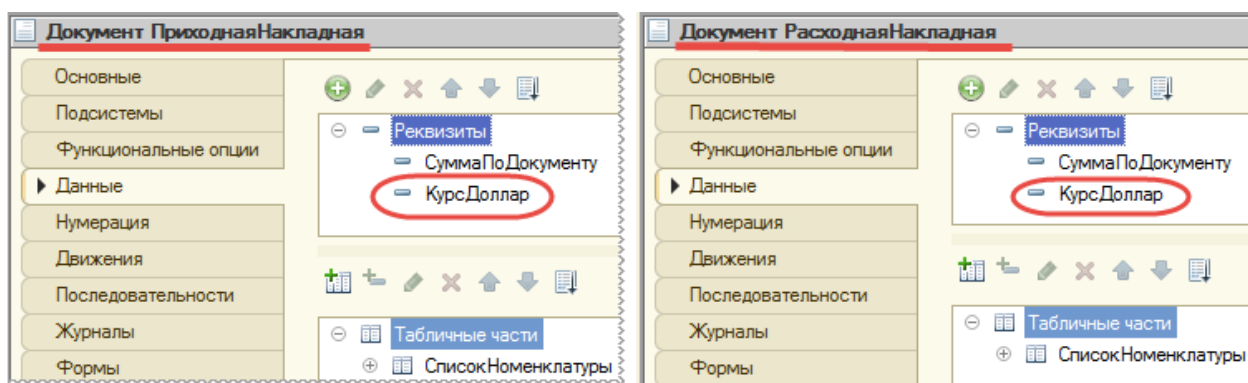


Рисунок 2 – Реквизит *Курс доллара* в документах

Для того, чтобы нельзя было провести документы с незаполненным курсом, в каждом документе для реквизита *КурсДоллар* свойство *Проверка заполнения* установим в значение «Выдавать ошибку»:

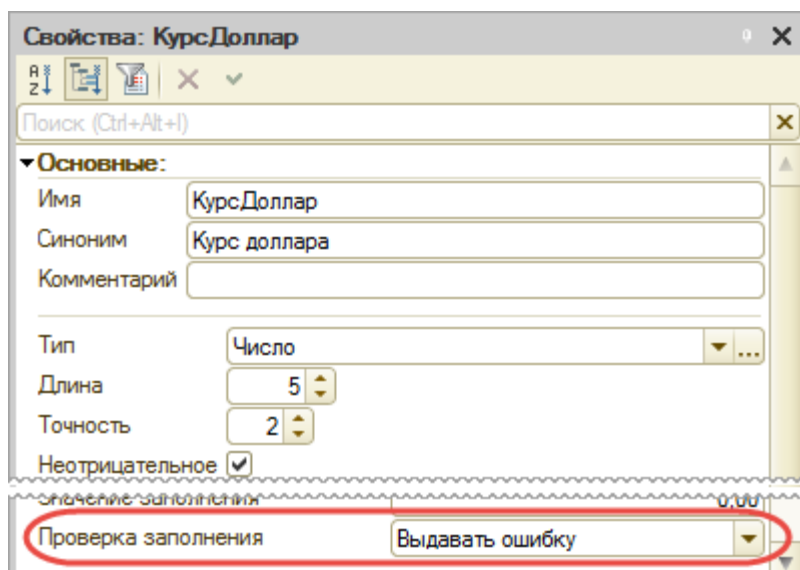
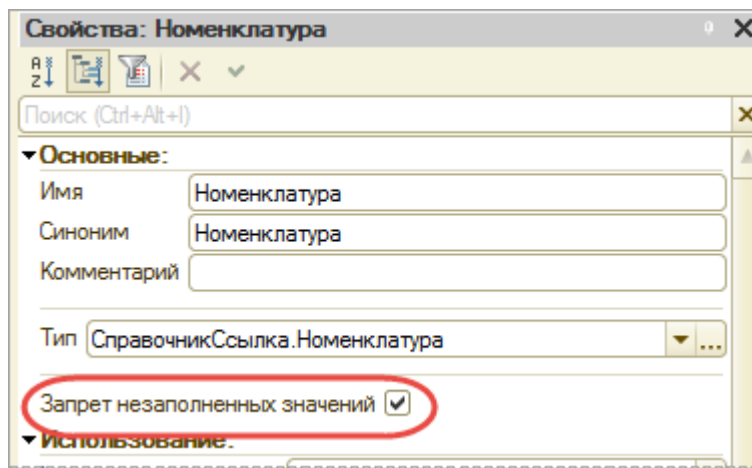


Рисунок 3 – Установка свойства «Проверка заполнения»

По условию задачи складской учет не ведется – это значит, что количественный учет будем вести только в разрезе номенклатуры. Себестоимость списания определяется как средняя по номенклатуре. Видим, что по условию задачи разрезы ведения количественного учета и стоимостного учета совпадают, поэтому построим решение на базе одного регистра с видом «Остатки». Вид регистра «Остатки», т.к. интересуют именно остатки стоимости и количества.

В каркасной конфигурации есть РН остатков «ОстаткиНоменклатуры». Измерение «Номенклатура» в нем есть, установим для него свойство *Запрет незаполненных значений* в истину, т.к. движения по регистру с пустой номенклатурой не имеют смысла:



**Рисунок 4 – Запрет незаполненных значений
для измерения *Номенклатура* в РН «Остатки номенклатуры»**

Ресурс «Количество» есть, добавим ресурсы:

- *Сумма* (Число 12, 2) – будем хранить себестоимость в рублях. Изменим синоним на значение «Сумма (руб.)»
- *СуммаДоллар* (Число 12, 2) – будем хранить себестоимость в долларах. Изменим синоним на значение «Сумма (дол.)».

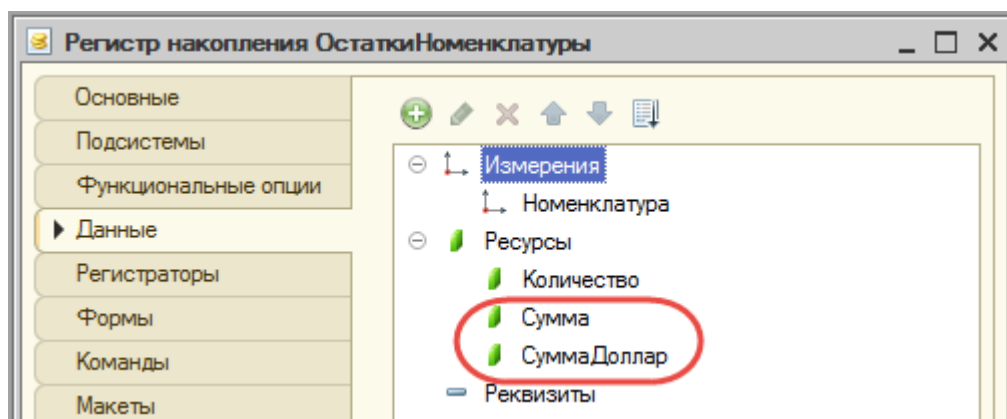


Рисунок 5 – Структура РН «Остатки номенклатуры»

Регистраторы:

- «Приходная накладная»
- «Расходная накладная»:

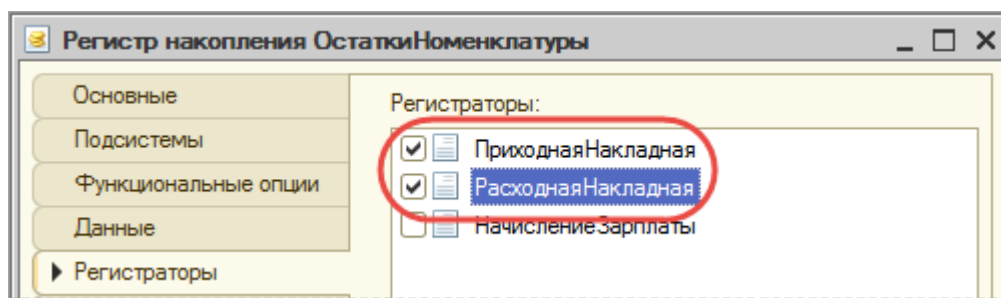


Рисунок 6 – Регистраторы РН «Остатки номенклатуры»

Для построения отчета о продажах нам понадобится оборотный регистр «Продажи номенклатуры». Данные в отчете нужны в разрезе товаров, поэтому создаем единственное измерение регистра:

- *Номенклатура* (СправочникСсылка.Номенклатура) – устанавливаем в истину флаг *Запрет незаполненных значений*.

Ресурсы:

- *Количество* (Число 10, 0)
- *Себестоимость* (Число 12, 2)
- *СебестоимостьДоллар*(Число 12, 2)
- *СуммаПродажи* (Число 12, 2).
- *СуммаПродажиДоллар* (Число 12, 2):

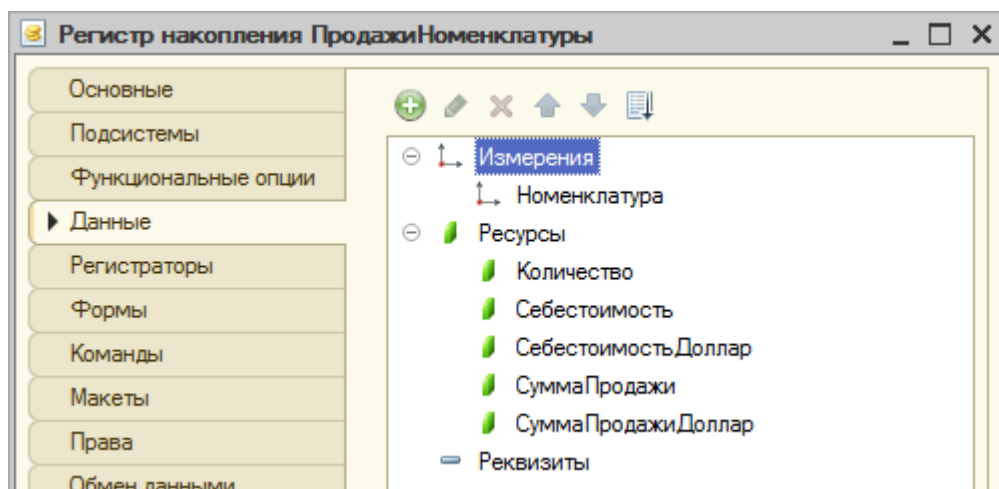


Рисунок 7 – Структура РН «Продажи номенклатуры»

Здесь может возникнуть вопрос: зачем создавать ресурсы для себестоимости, если они есть в РН «Остатки номенклатуры»? Дело тут в следующем:

- Для получения данных о продажах и себестоимости для отчета не придется выполнять в запросе соединение таблиц, т.к. вся необходимая информация есть в РН «Продажи номенклатуры». Кроме того, для отчета нужны данные оборотов, а не остатков, а оборотный регистр с точки зрения быстродействия более приспособлен для решения задачи получения оборотов, чем регистр остатков
- В экзаменационной задаче условие по списанию себестоимости может быть иное, чем в учебной задаче. Например, может быть указано, что списание должно быть организовано по партиям. А в отчете по продажам партии не интересуют – только в разрезе номенклатуры. Решение задачи будет гораздо проще, если данные о себестоимости будут в РН «Продажи номенклатуры».

Регистраторы:

- «Расходная накладная»:

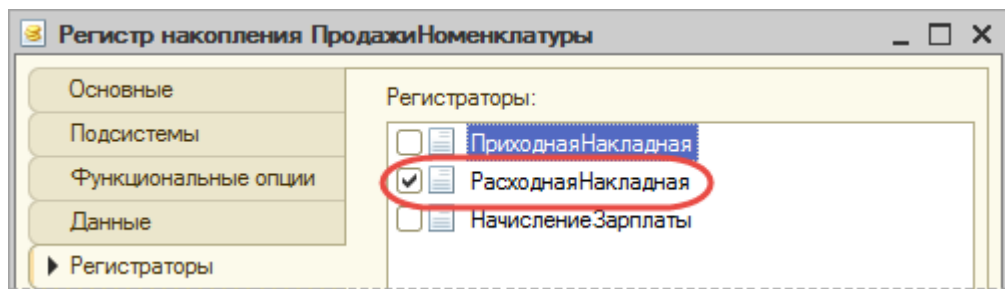


Рисунок 8 – Регистраторы РН «Продажи номенклатуры»

Все объекты метаданных готовы. Переходим к проведению документов.

Проведение документа «Приходная накладная»

Построение движений документа «Приходная накладная» с помощью конструктора разбиралось в главе «6. Как реализовать поступление товаров в компанию». Значение ресурса *СуммаДоллар* будем рассчитывать как *Сумма*, деленная на *Курс доллара*, указанный в документе.

Листинг обработки проведения:

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
```

// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

// регистр ОстаткиНоменклатуры Приход

Движения.ОстаткиНоменклатуры.Записывать = Истина;

Для Каждого ТекСтрокаСписокНоменклатуры Из СписокНоменклатуры Цикл

Движение = Движения.ОстаткиНоменклатуры.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение.Период = Дата;

Движение.Номенклатура = ТекСтрокаСписокНоменклатуры.Номенклатура;

Движение.Количество = ТекСтрокаСписокНоменклатуры.Количество;

Движение.Сумма = ТекСтрокаСписокНоменклатуры.Сумма;

Движение.СуммаДоллар =

? (КурсДоллар=0,0,ТекСтрокаСписокНоменклатуры.Сумма/КурсДоллар);

КонецЦикла;

//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ

КонецПроцедуры

Проверим работу документа в режиме «1С:Предприятие»:

← → ☆ Приходная накладная 000000001 от 01.10.2018 14:26:18

Основное Остатки номенклатуры

Провести и закрыть Записать Провести

Номер: 000000001

Дата: 01.10.2018 14:26:18

Курс доллара: 65,00

Добавить

N	Номенклатура	Количество	Сумма
1	Йогурт "Малина"	30	1 200,00
2	Крупа манная	10	300,00

Рисунок 9 – Документ «Приходная накладная №1»

Движения:

←

→

☆

Приходная накладная 000000001 от 01.10.2018 14:26:18

Основное

Остатки номенклатуры

Остатки номенклатуры

Период	↓	Регистратор	Ном...	Номенклатура	Количество	Сумма (руб.)	Сумма (дол.)
+ 01.10.2018 1...		Приходная ...	1	Йогурт "Малина"	30	1 200,00	18,46
+ 01.10.2018 1...		Приходная ...	2	Крупа манная	10	300,00	4,62

Рисунок 10 – Движения документа «Приходная накладная №1»

Видим, что в движениях все ресурсы заполнились верно.

Проведение документа «Расходная накладная»

Как рассчитать себестоимость по средней и при этом решить «проблему копеек», уже разбирали ранее в практическом примере в главе «12. Что такое “Проблема копеек” и как ее решить». Отличие текущей задачи в том, что еще нужно в движениях по регистру точно таким же способом рассчитывать значение ресурса СуммаДоллар. Еще будем формировать движения по РН «Продажи номенклатуры»

Примечание: Контроль отрицательных остатков и установку блокировки в рамках данного решения рассматривать и выполнять не будем. Про блокировки можно узнать в главе «11. Какие навыки в использовании управляемых блокировок потребуются на экзамене», про контроль отрицательных остатков – см. главу «9. Как выполнить контроль остатков по старой методике».

Листинг обработки проведения:

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

```
// 1. Подготовка наборов записей регистра
Движения.ОстаткиНоменклатуры.Записывать = Истина;
Движения.Записать();
// 2. Установка маркера Записи у регистра
Движения.ОстаткиНоменклатуры.Записывать = Истина;
Движения.ПродажиНоменклатуры.Записывать = Истина;

// 3. Запрос для получения данных для расчета себестоимости
Запрос = Новый Запрос("ВЫБРАТЬ
| РасходнаяНакладнаяСписокНоменклатуры.Номенклатура КАК Номенклатура,
| СУММА(РасходнаяНакладнаяСписокНоменклатуры.Количество) КАК Количество,
| СУММА(РасходнаяНакладнаяСписокНоменклатуры.Сумма) КАК Сумма
| ПОМЕСТИТЬ ТЧСписокНоменклатуры
| ИЗ
```

```

|      Документ.РасходнаяНакладная.СписокНоменклатуры КАК
РасходнаяНакладнаяСписокНоменклатуры
|ГДЕ
|      РасходнаяНакладнаяСписокНоменклатуры.Ссылка = &Ссылка
|
|СГРУППИРОВАТЬ ПО
|      РасходнаяНакладнаяСписокНоменклатуры.Номенклатура
|
|ИНДЕКСИРОВАТЬ ПО
|      Номенклатура
|;
|
|////////////////////////////////////
|ВЫБРАТЬ
|      ТЧСписокНоменклатуры.Номенклатура,
|      ТЧСписокНоменклатуры.Количество,
|      ТЧСписокНоменклатуры.Сумма КАК СуммаПродажи,
|      ЕСТЬNULL(ОстаткиНоменклатурыОстатки.СуммаОстаток, 0) КАК СуммаОстаток,
|      ЕСТЬNULL(ОстаткиНоменклатурыОстатки.КоличествоОстаток, 0) КАК КоличествоОстаток,
//3.1 получаем остаток суммы себестоимости в долларах
|      ЕСТЬNULL(ОстаткиНоменклатурыОстатки.СуммаДолларОстаток, 0) КАК СуммаДолларОстаток
|ИЗ
|      ТЧСписокНоменклатуры КАК ТЧСписокНоменклатуры
|          ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОстаткиНоменклатуры.Остатки(
|              &МоментВремени,
|              Номенклатура В
|                  (ВЫБРАТЬ
|                      ТЧСписокНоменклатуры.Номенклатура
|                  ИЗ
|                      ТЧСписокНоменклатуры КАК ТЧСписокНоменклатуры)) КАК
ОстаткиНоменклатурыОстатки
|          ПО ТЧСписокНоменклатуры.Номенклатура =
ОстаткиНоменклатурыОстатки.Номенклатура");

//4. Заполнение параметров запроса
Запрос.УстановитьПараметр("МоментВремени", МоментВремени());
Запрос.УстановитьПараметр("Ссылка", Ссылка);

//5. выполняем запрос, обходим выборку,
//рассчитываем себестоимость, формируем движения по списанию товара
//и по продаже товара
Результат = Запрос.Выполнить();
ВыборкаНоменклатура = Результат.Выбрать();
Пока ВыборкаНоменклатура.Следующий() Цикл

    КоличествоСписания = ВыборкаНоменклатура.Количество;
    КоличествоОстаток = ВыборкаНоменклатура.КоличествоОстаток;

//5.1 расчет себестоимости списания в рублях
//и в долларах по формуле с учетом последнего списания
Если КоличествоСписания = КоличествоОстаток Тогда
    Себестоимость = ВыборкаНоменклатура.СуммаОстаток;

```

```

СебестоимостьДоллар = ВыборкаНоменклатура.СуммаДолларОстаток;
Иначе
    Себестоимость =
ВыборкаНоменклатура.СуммаОстаток*КоличествоСписания/КоличествоОстаток;
    СебестоимостьДоллар =
ВыборкаНоменклатура.СуммаДолларОстаток*КоличествоСписания/КоличествоОстаток;
КонецЕсли;

//5.2 расход по РН ОстаткиНоменклатуры
Движение = Движения.ОстаткиНоменклатуры.ДобавитьРасход();
Движение.Период = Дата;
Движение.Номенклатура = ВыборкаНоменклатура.Номенклатура;
Движение.Количество = КоличествоСписания;
Движение.Сумма = Себестоимость;
Движение.СуммаДоллар = СебестоимостьДоллар;

//5.2 движение по РН ПродажиНоменклатуры
Движение = Движения.ПродажиНоменклатуры.Добавить();
Движение.Период = Дата;
Движение.Номенклатура = ВыборкаНоменклатура.Номенклатура;
Движение.Количество = КоличествоСписания;
Движение.Себестоимость = Себестоимость;
Движение.СебестоимостьДоллар = СебестоимостьДоллар;
Движение.СуммаПродажи = ВыборкаНоменклатура.СуммаПродажи;
Движение.СуммаПродажиДоллар = ?(КурсДоллар
=0,0,ВыборкаНоменклатура.СуммаПродажи/КурсДоллар);

КонецЦикла;

КонецПроцедуры

```

Приведенный алгоритм во многом совпадает с алгоритмом, который разбирался в разделе «**12. Что такое “Проблема копеек” и как ее решить**». Поэтому подробный разбор приводить не будем. Остановимся только на изменениях алгоритма:

1. Запрос получения данных (п. № 3 в листинге кода): во втором запросе пакета получаем не только остаток суммы в рублях, но еще и остаток суммы в долларах (п. № 3.1 в листинге кода)
2. Формулы для расчета себестоимости списания в рублях и в долларах (п. № 5.1 в листинге кода): и для суммы в рублях, и для суммы в долларах используем одинаковый подход.

Проверим решение в режиме «1С:Предприятие».

Введем документ «Расходная накладная»:

← →
★ **Расходная накладная 000000001 от 03.10.2018 15:37:59**

Основное
Остатки номенклатуры
Продажи номенклатуры

Провести и закрыть
Записать
Провести

Номер:

Дата:

Курс доллара: 70,00

Добавить

N	Номенклатура	Количество	Сумма
1	Йогурт "Малина"	20	1 400,00
2	Крупа манная	10	400,00

Рисунок 11 – Документ «Расходная накладная №1»

Посмотрим таблицу движений:

← →	★	Остатки номенклатуры					
Период	↓	Регистратор	Н..	Номенклатура	Количест...	Сумма (руб.)	Сумма (дол.)
+ 01.10.2018...		Приходная накладная 000000001 от 01.10....	1	Йогурт "Малина"	30	1 200,00	18,46
+ 01.10.2018...		Приходная накладная 000000001 от 01.10....	2	Крупа манная	10	300,00	4,62
- 03.10.2018...		Расходная накладная 000000001 от 03.10....	1	Йогурт "Малина"	20	800,00	12,31
- 03.10.2018...		Расходная накладная 000000001 от 03.10....	2	Крупа манная	10	300,00	4,62

Рисунок 12 – Движения документов по РН «Остатки номенклатуры»

Если бы для номенклатуры «Йогурт "Малина"» Сумма (дол) рассчиталась по курсу, то она была бы равна $800 / 75 = 11,43$ дол. Но т.к. по условию курсовые разницы не предполагаются, то расчет произошел по формуле расчета себестоимости по средней: $18,46 * 20 / 30 = 12,31$. Списано 20 шт из 30, имеющихся на остатке.

Для «Крупы манной» это было последнее списание, поэтому сумма списания в долларах равна остатку Сумма (дол).

Движения документе по РН «Продажи номенклатуры»:

Период	Регист...	Н.	Номенклатура	Кол...	Себестоимость (руб.)	Себестоимость (дол.)	Сумма продажи (руб.)	Сумма продажи (дол.)
03.10.201...	Расхо...	1	Йогурт "Малина"	20	800,00	12,31	1 400,00	20,00
03.10.201...	Расхо...	2	Крупа манная	10	300,00	4,62	400,00	5,71

Рисунок 13 – Движения документа «Расходная накладная №2» по РН «Продажи номенклатуры»

Видим, что себестоимость такая же, как в РН «Остатки номенклатуры», а сумма продажи в долларах рассчиталась по курсу из документа. Что и требовалось.

Проверим, как обстоят дела с остатками по РН «Остатки номенклатуры» с помощью обработки «Консоль запросов».

Как запустить «Консоль запросов» в режиме управляемого приложения разбирали в главе «5. Как обеспечить возможность запуска обычных форм в режиме управляемого приложения (используем Консоль запросов из каркасной конфигурации)»:

Консоль запросов : E:\Запросы\ДвухвалютныйУчет.sel

Действия ▾

Запрос

⊕ Запросы

ВЫБРАТЬ

ОстаткиНоменклатурыОстатки.Номенклатура,
ОстаткиНоменклатурыОстатки.КоличествоОстаток,
ОстаткиНоменклатурыОстатки.СуммаОстаток,
ОстаткиНоменклатурыОстатки.СуммаДолларОстаток

ИЗ

РегистрНакопления.ОстаткиНоменклатуры.Остатки

КАК ОстаткиНоменклатурыОстатки

Способ выгрузки:

Номенклатура	КоличествоОстаток	СуммаОстаток	СуммаДолларОстаток
Йогурт "Малина"	10	400,00	6,15

Рисунок 14 – Состояние остатков по РН «Остатки номенклатуры»

Видим, что все в порядке. В таблице остатков отсутствует номенклатура «Крупа манная», есть только остаток в количестве 10 шт по «Йогурт "Малина"».

Отчет разрабатывать не будем, но построим запрос, аналогичный тому, который требуется для отчета в «Консоли запросов»:

Консоль запросов : E:\Запросы\Двухвалютный

Действия ▾

Выполнить & Параметры

Имя параметра В Значение параметра

НачПериод		01.09.2018 0:00:00
КонПериод		31.10.2018 23:59:59

Запрос

Запросы

Продажи

ВЫБРАТЬ

ПродажиНоменклатурыОбороты.Номенклатура,
 ПродажиНоменклатурыОбороты.КоличествоОборот КАК Кол_во,
 ПродажиНоменклатурыОбороты.СебестоимостьОборот КАК Себест_сть,
 ПродажиНоменклатурыОбороты.СебестоимостьДолларОборот КАК Себест_стьДоллар,
 ПродажиНоменклатурыОбороты.СуммаПродажиОборот КАК Продажа,
 ПродажиНоменклатурыОбороты.СуммаПродажиДолларОборот КАК ПродажаДоллар,
 ПродажиНоменклатурыОбороты.СуммаПродажиОборот -
 ПродажиНоменклатурыОбороты.СебестоимостьОборот КАК Прибыль,
 ПродажиНоменклатурыОбороты.СуммаПродажиДолларОборот -
 ПродажиНоменклатурыОбороты.СебестоимостьДолларОборот КАК ПрибыльДоллар

ИЗ

РегистрНакопления.ПродажиНоменклатуры.Обороты(&НачПериод, &КонПериод, ,)
 КАК ПродажиНоменклатурыОбороты

Способ выгрузки: Список

Номенклатура	Кол_во	Себест_сть	Себест_стьДоллар	Продажа	ПродажаДоллар	Прибыль	ПрибыльДоллар
Йогурт "Малина"	20	800,00	12,31	1 400,00	20,00	600	7,69
Крупа манная	10	300,00	4,62	400,00	5,71	100	1,09

Рисунок 15 – Результат запроса для анализа продаж в двух валютах

Подведем итог

Мы разобрали, что в задачах с валютой в первую очередь нужно определить, является ли список валют фиксированным или переменным, т.к. это повлияет на выбор методики решения задачи.

Мы рассмотрели, что из себя представляют **задачи с фиксированным списком валют** и как их решать.

В следующем блоке материалов разберем, как решать **задачи с переменным перечнем валют**.

25. Как решать задачи с переменным перечнем валют

Примеры формулировок условий таких задач:

- Закупка и продажа товаров может происходить в **любой валюте**. Валюта указывается для каждого документа поступления и продажи...
- Взаиморасчеты ведутся в разрезе контрагентов и договоров в **валюте взаиморасчетов...** В каждом договоре может быть указана **своя валюта взаиморасчетов...**
- Учет ведется в рублях, а с агентами дополнительно **ведется в валюте**, причем для каждого агента установлена **своя собственная валюта, не подлежащая пересмотру...** При начислении вознаграждения агенту валютная сумма определяется исходя из рублевой суммы, валюты агента и **курса валюты на дату документа...**

В задачах этого типа перечень валют заранее неизвестен и может расширяться в процессе деятельности предприятия. Кроме того, встречаются несколько вариантов дополнительных условий, касающихся особенностей хранения валюты. В зависимости от этих условий подготовку объектов метаданных можно представить в виде схемы:

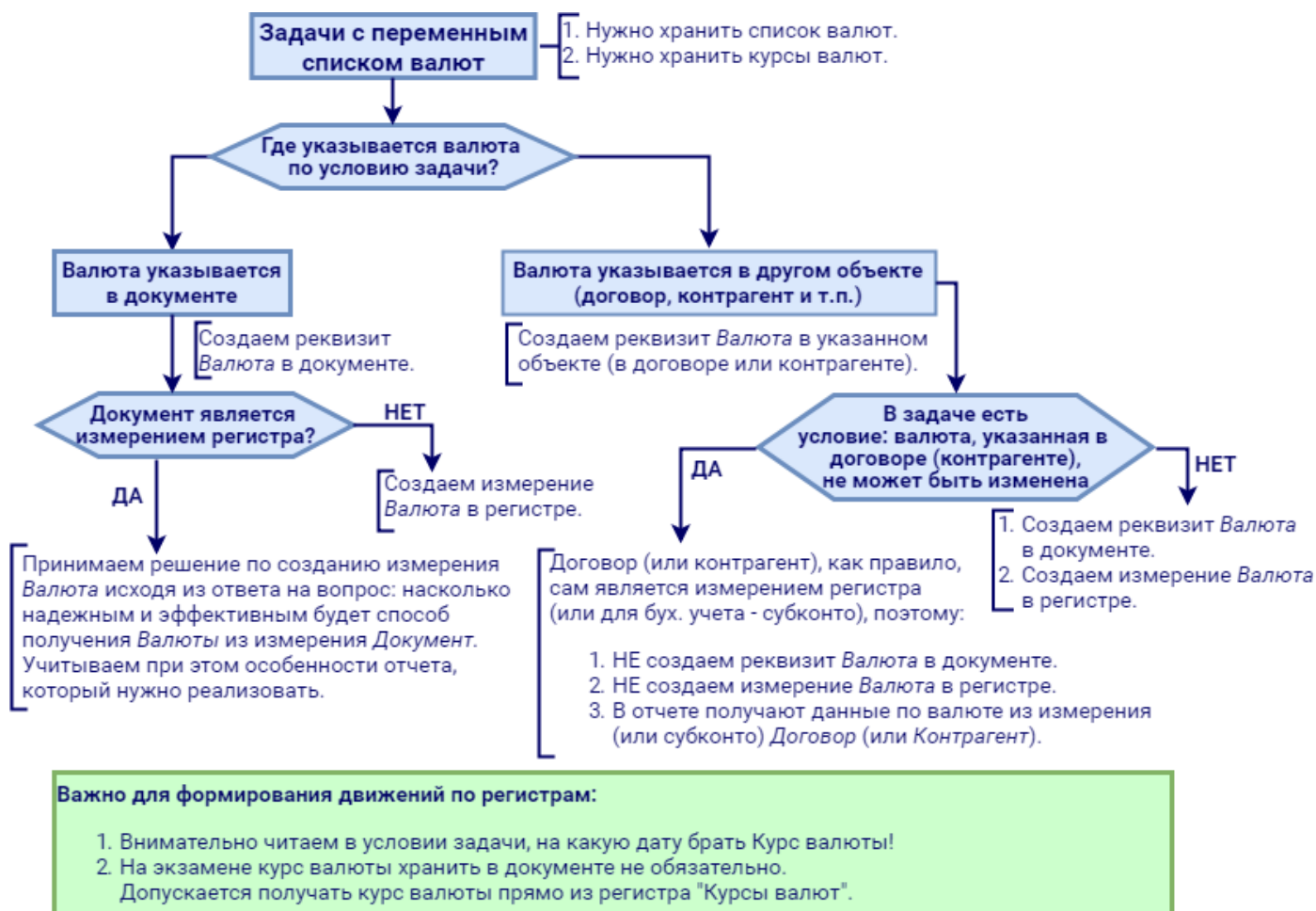


Рисунок 1 – Схема подготовки объектов метаданных в задачах с переменным списком валют

Разберем подробнее некоторые элементы схемы.

Где и как хранить список валют и курсы валют?

В каркасной конфигурации есть справочник «Валюты». Обратим внимание, что в нем уже создан предопределенный элемент «Российский рубль».

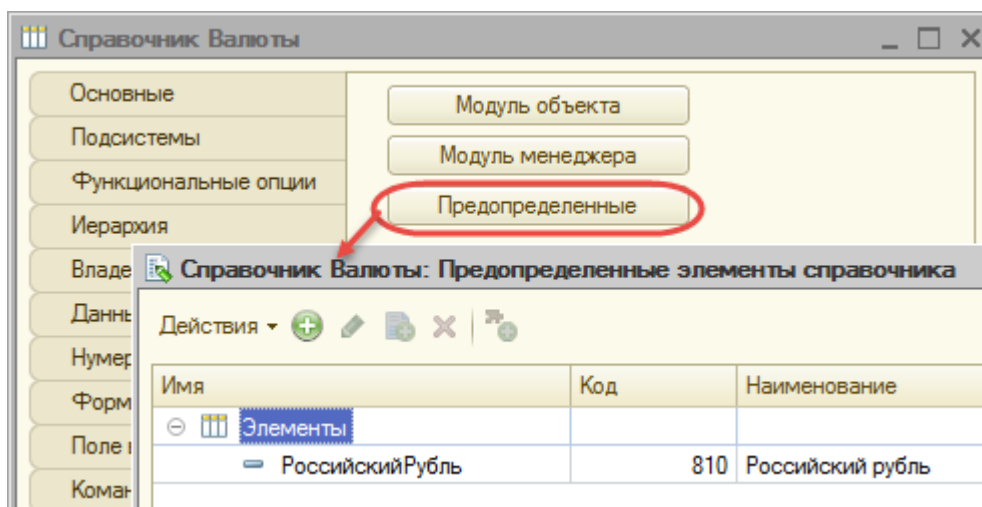


Рисунок 2 – Предопределенный элемент справочника «Валюты»

Для хранения курсов валют понадобится регистр сведений с измерением *Валюта* и ресурсом *Курс*. Значения курса пользователь будет вносить вручную, поэтому регистр сведений должен быть независимым. С течением времени курс меняется, поэтому регистр сведений нужен периодический. Период – день. Будем считать, что курс меняется один раз в день: так же, как в типовых конфигурациях 1С.

В каркасной конфигурации РС «Курсы валют» есть.

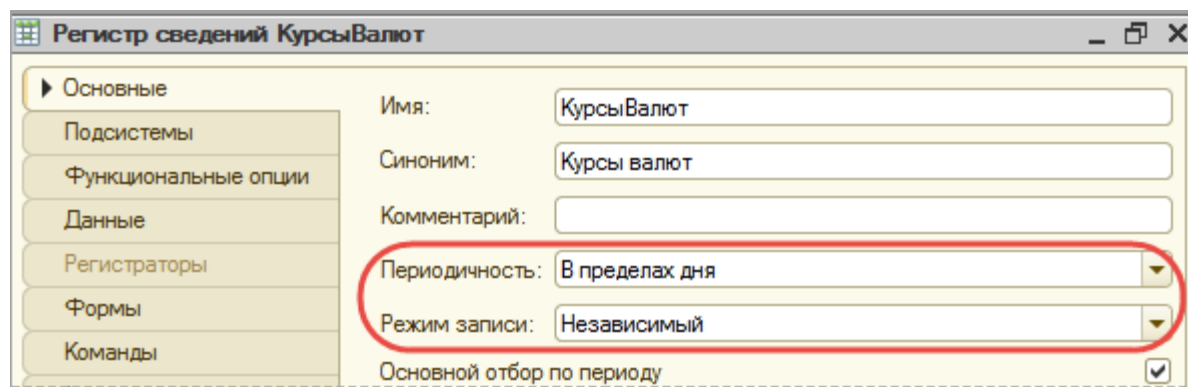


Рисунок 3 – Периодичность и режим записи РС «Курсы валют»

Измерения регистра:

- *Валюта (СправочникСсылка.Валюты).*

Ресурсы:

- *Курс (Число 5, 2), Неотрицательное = истина:*

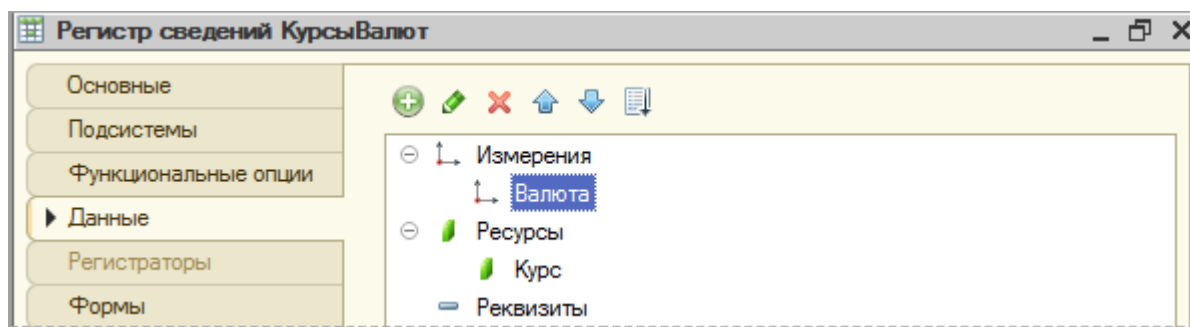


Рисунок 4 – Измерения и ресурсы РС «Курсы валют»

Чтобы нельзя было установить курс валюты равным нулю, для ресурса *Курс* свойство *Проверка заполнения* установим в значение «Выдавать ошибку». Так же сделано в типовых конфигурациях 1С:

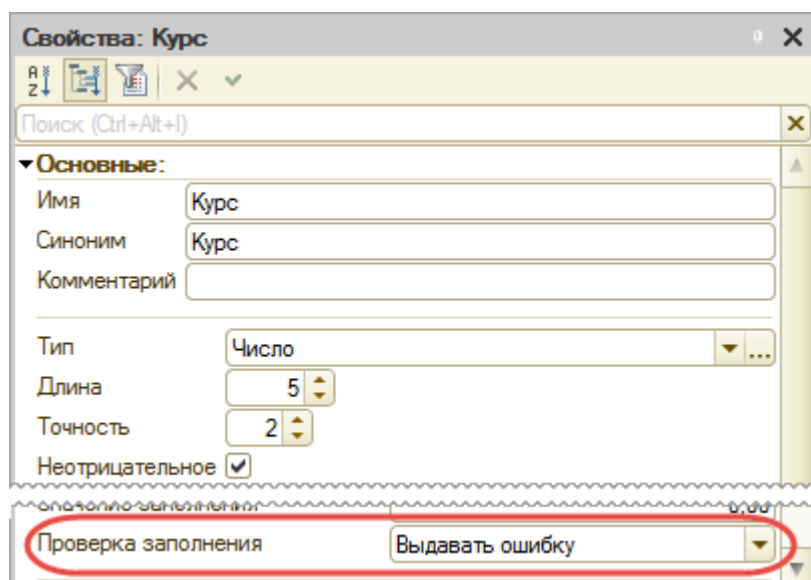


Рисунок 5 – Установка свойства «Проверка заполнения» для ресурса Курс

Установка проверки заполнения для ресурса *Курс* не является обязательным требованием на экзамене. Но если этого не сделать, то следует помнить, что у пользователя есть возможность установить курс валюты равный нулю и, соответственно, при делении на курс нужно учитывать возможность возникновения ошибки деления на нуль.

Изменения в документах и регистрах для отражения операций в валюте

Читаем внимательно условие, где именно указывается валюта: непосредственно в документе, в договоре или в контрагенте. Создаем в соответствующем объекте реквизит *Валюта*. Он будет иметь тип СправочникСсылка.Валюты.

В задаче может быть оговорка, что валюта, установленная в договоре (или в контрагенте), не подлежит пересмотру. Как правило, по условиям этих задач договор сам является измерением (или субконто – для задачи бух. учета) регистра. Поэтому везде в отчетах валюту можно получить из договора. В этом случае валюту не нужно заводить как измерение в регистре.

Из опыта сдававших экзамен: если при наличии такого условия в задаче бухгалтерского учета завести валюту как измерение на регистре бухгалтерии, то со стороны экзаменатора это может быть оценено как некорректное создание аналитики по валюте.

Нужно ли хранить в шапке документа курс валюты?

На экзамене не требуется создавать реквизит *Курс* в документе. В момент проведения документа можно просто получать курс валюты из регистра запросом или используя объектную модель. Главное – не делать это в цикле.

В реальной практике для обеспечения более устойчивого поведения системы лучше хранить курс валюты в документе. Например, после проведения расходной накладной пользователь изменил курс валюты, возможно, ошибочно. При повторном проведении того же документа в движениях документа будет уже другая сумма.

Для обеспечения более стабильного поведения системы в шапку добавляют реквизиты «Курс» и «Кратность» и используют их значения при проведении. Таким образом, при проведении будут использоваться только данные документа, тем самым обеспечивая стабильное поведение системы.

Практический пример задачи с переменным перечнем валют

Начинаем разбор практического примера задачи с переменным перечнем валют.

Постановка задачи:

Компания оказывает услуги контрагентам. Оказание услуг оформляется документом «Расходная накладная», причем **каждая накладная может быть выставлена в своей валюте**. Оплата от контрагента приходит в рублях и отражается документом «Поступление денег». Погашение задолженности оплатой происходит автоматически по методу FIFO по курсу валюты на дату оплаты. В случае, если сумма оплат превышает долг по накладным, то эта переплата засчитывается как аванс. При оформлении услуг необходимо проверять наличие аванса от покупателя. Если есть аванс, то он засчитывается как оплата накладной по курсу на дату накладной.

Требуется сформировать отчет по взаиморасчетам (отгружено, оплачено) за период в разрезе накладных с группировкой по валюте.

Подготовка объектов метаданных

Разберем условие задачи согласно схеме на рис. № 1.

Перечень валют заранее неизвестен, значит будем использовать справочник «Валюты» и регистр сведений «Курсы валют».

Валюта по условию задачи указывается в документе, значит, в расходную накладную добавим реквизит *Валюта* (СправочникСсылка.Валюты). Также нам понадобится реквизит *Контрагент* (СправочникСсылка.Контрагенты):

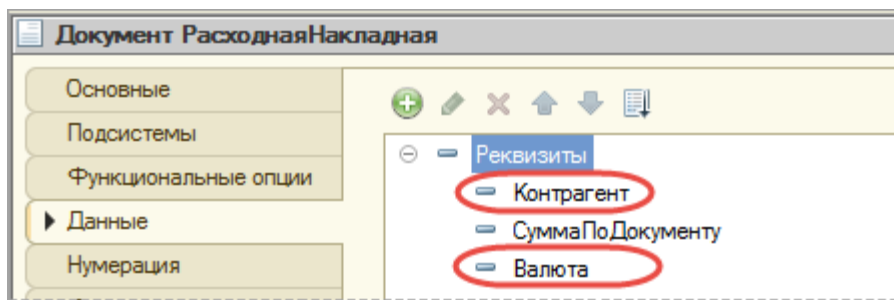


Рисунок 6 – Реквизиты документа «РасходнаяНакладная»

Для отражения факта поступления оплаты от клиента добавим в конфигурацию новый документ «Поступление денег».

Реквизиты:

- *Контрагент* (СправочникСсылка.Контрагенты)
- *Сумма* (Число 12, 2).

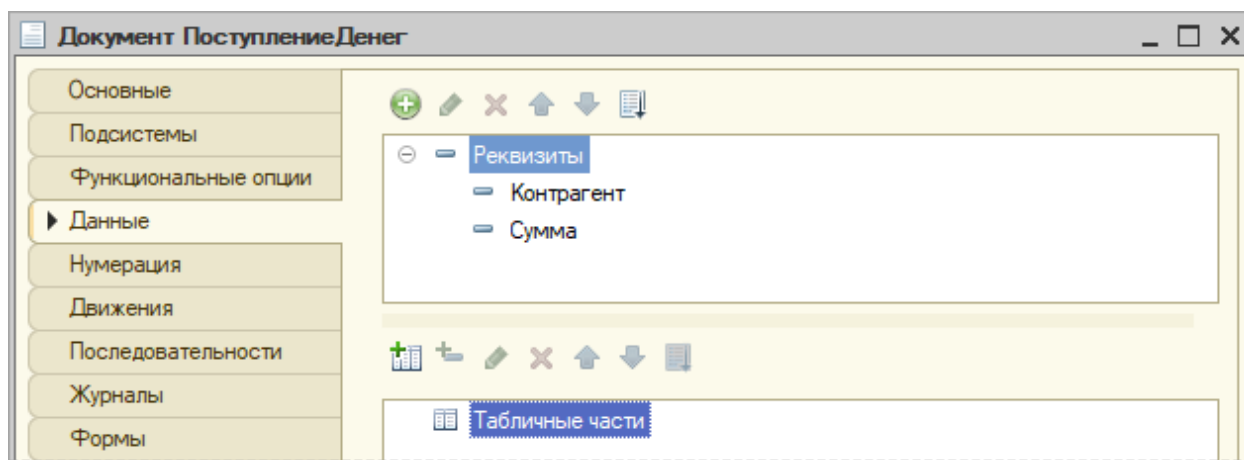


Рисунок 7 – Реквизиты документа «Поступление денег»

По условию задачи оплата всегда поступает в рублях. Поэтому реквизит *Валюта* не создаем. Если его создать, то в пользовательском режиме можно будет ввести оплату в другой валюте, а это противоречит условию задачи.

Для отслеживания состояния взаиморасчетов с клиентами создадим РН «Взаиморасчеты». Вид регистра – *Остатки*, т.к. необходимо будет получать данные по остаткам задолженностей и авансов. Задолженности интересуют в разрезе валют, контрагентов и накладных. Авансы – только в разрезе контрагентов.

Измерения:

- *Контрагент* (СправочникСсылка.Контрагенты): устанавливаем в истину флаг *Запрет незаполненных значений*
- *Документ расчетов* (ДокументСсылка.РасходнаяНакладная): ссылку на документ «Поступление денег» в состав типа не включаем, т.к. по условию авансы будем учитывать в целом по контрагенту без детализации по документам оплаты.
В движениях по авансам (при проведении документа «Поступление денег») измерение *Документ расчетов* заполнять не будем. Поэтому флаг *Запрет незаполненных значений* не устанавливаем.
- *Валюта* (СправочникСсылка.Валюты): устанавливаем в истину флаг *Запрет незаполненных значений*.

Казалось бы, зачем заводить измерение *Валюта*? Ведь есть измерение *Документ расчетов*, из которого для отчета можно было бы взять валюту. Но кроме задолженностей, есть еще авансы. А для авансов *Документ расчетов* будет не заполнен. Тут можно заметить, что авансы всегда в рублях, но чтобы не усложнять задачу различными условиями для интерпретации данных, полученных из регистра, заведем в регистре измерение *Валюта*.

Ресурсы:

- Сумма (Число 12, 2):

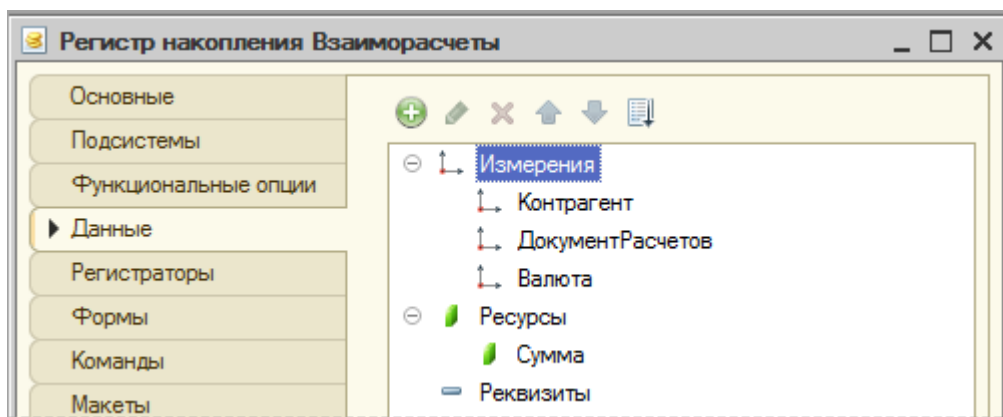


Рисунок 8 – Структура РН «Взаиморасчеты»

Регистраторы:

- «Приходная накладная»
- «Поступление денег»:

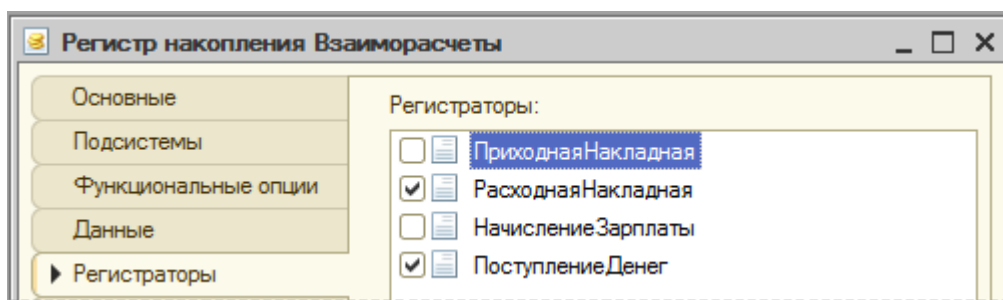


Рисунок 9 – Регистраторы РН «Взаиморасчеты»

Отражение возникновения задолженности. Проведение документа «Расходная накладная»

При оказании услуг или отгрузке товаров клиентам происходит увеличение задолженности клиента на стоимость оказанных услуг или отгруженных товаров.

Увеличение задолженности будем отражать как приход по РН «Взаиморасчеты».

Сформируем обработку проведения при помощи конструктора движений. Пока в обработке проведения отразим только увеличение задолженности контрагента. По части зачета авансов пока ничего не делаем, вернемся к этой части условия позже.

Процедура ОбработкаПроведения(Отказ, Режим)

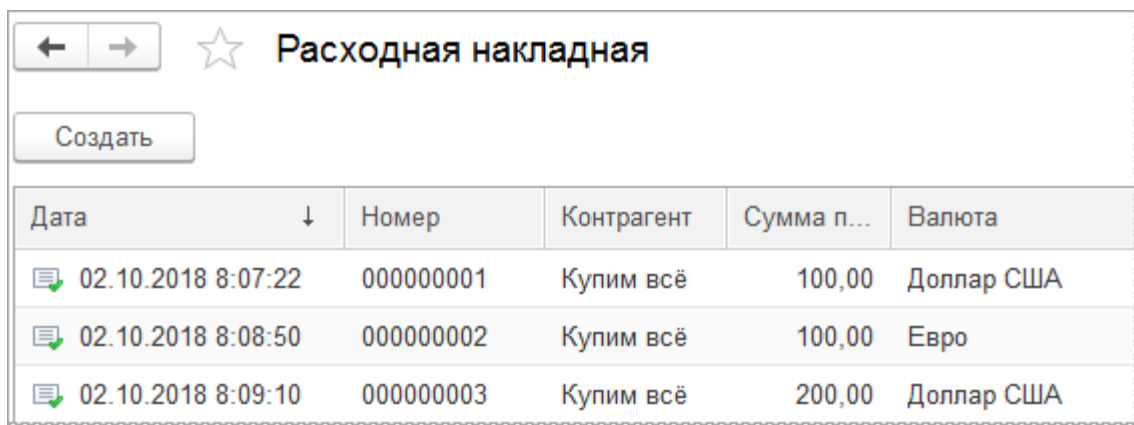
```
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

// регистр Взаиморасчеты Приход
Движения.Взаиморасчеты.Записывать = Истина;
Движение = Движения.Взаиморасчеты.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Контрагент = Контрагент;
Движение.ДокументРасчетов = Ссылка;
Движение.Валюта = Валюта;
Движение.Сумма = СуммаПоДокументу;

//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

КонецПроцедуры

В режиме «1С:Предприятие» сформируем несколько документов «Расходная накладная»:



Дата	Номер	Контрагент	Сумма п...	Валюта
02.10.2018 8:07:22	000000001	Купим всё	100,00	Доллар США
02.10.2018 8:08:50	000000002	Купим всё	100,00	Евро
02.10.2018 8:09:10	000000003	Купим всё	200,00	Доллар США

Рисунок 10 – Пример списка документов «Расходная накладная»

Посмотрим движения по одному из документов:

Как добавить для документа возможность просмотра движений по регистру, разбирали в блоке «4. Как быстро получить движения по регистрам из управляемой формы – без единой строчки кода».

←

→

☆

Расходная накладная 000000001 от 02.10.2018 8:07:22

Основное

Взаиморасчеты

Взаиморасчеты

Период	↓	Регистратор	Номер	Контрагент	Документ расчетов	Валюта	Сумма
+	02.10.2018 8:07:22	Расходная накладная 000000001...	1	Купим всё	Расходная накладная 000000001...	Доллар США	100,00

Рисунок 11 – Движения документа «Расходная накладная № 1»

Видим, что все хорошо: сформировалось единственное движение по увеличению задолженности контрагента за оказанную услугу на сумму документа «Расходная накладная».

Подведем итоги

Мы рассмотрели, что собой представляют **задачи с переменным списком валют**, какие в них встречаются варианты дополнительных условий. Разобрали схему проектирования структуры объектов метаданных для задач с разными вариантами условий.

Также мы начали решение практического примера задачи с переменным списком валют.

В следующих блоках продолжим разбор практического примера и решим следующие подзадачи:

- Как реализовать погашение задолженностей в разных валютах оплатой в рублях

Как реализовать зачет аванса в рублях как оплату накладной в валюте.

26. Как погасить задолженность в разных валютах оплатой в рублях

Продолжаем разбор практического примера задачи с переменным списком валют.

Реализуем требование задачи по погашению задолженности оплатой по методу FIFO. Переплату будем отражать как аванс по контрагенту. Сделаем это в обработке проведения документа «Поступление денег».

Проведение документа «Поступление денег»

Алгоритм погашения задолженности в разрезе накладных по FIFO уже разбирали ранее в главе .

Особенность данной задачи состоит в том, что отличаются валюты оплаты и задолженности. Все оплаты оформляются в рублях. Задолженности могут быть в любых валютах. Погашать их нужно по курсу на дату оплаты.

Чтобы понимать, какие суммы долга по накладной можно погасить, будем сравнивать задолженности по накладным с суммой оплаты. Для этого сначала нужно сделать суммы соизмеримыми, т.е. выразить в одной валюте.

Для всех имеющихся задолженностей будем определять курс валюты на дату оплаты и рассчитывать рублевый эквивалент. Для этого при получении данных по остаткам задолженностей в запросе будем делать соединение таблицы остатков РН «Взаиморасчеты» с данными виртуальной таблицы срез последних РС «Курсы валют»:



Рисунок 1 – Схема соединения таблиц для получения данных о задолженностях

При поступлении оплаты возможны три варианта:

- Оплата **меньше** рублевого эквивалента долга по накладной
- Оплата **равна** рублевому эквиваленту долга по накладной
- Оплата **больше** рублевого эквивалента долга по накладной.

Разберем, как определить сумму погашения долга в валюте для каждого из вариантов.

Вариант №1	Вариант №2	Вариант №3
Оплата меньше рублевого эквивалента долга	Оплата равна рублевому эквиваленту долга	Оплата больше рублевого эквивалента долга
<p>Сумму погашения долга в валюте нужно рассчитать по курсу исходя из суммы оплаты:</p> <p>Сумма погашения = СуммаОплаты/Курс.</p> <p>Алгоритм на этом закончен, т.к. вся оплата была израсходована.</p>	<p>Сумма погашения долга будет равна остатку долга:</p> <p>Сумма погашения = Сумма долга (вал.)</p> <p>В данном случае не нужно рассчитывать сумму по курсу, чтобы избежать возможных ошибок округления при пересчете.</p> <p>На этом алгоритм обхода задолженностей закончится,</p>	<p>Сумма погашения долга будет равна остатку долга:</p> <p>Сумма погашения = Сумма долга (вал.)</p> <p>Также пересчет по курсу не выполняем.</p> <p>Алгоритм на этом не заканчивается. Рассчитываем остаток оплаты после погашения долга: для этого от суммы оплаты отнимаем</p>

	<p>т.к. остатка оплаты для погашения следующей задолженности уже не будет.</p>	<p>рублевый эквивалент погашенного долга.</p> <p>Далее применяем ту же логику для анализа остатка оплаты.</p> <p>Если окажется, что задолженностей для погашения больше нет, а остаток оплаты есть, то остаток оплаты станет авансом.</p>
<p>Пример №1</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Документ "Поступление денег №1" Контрагент = "Купим все" Сумма = 1 000,00 Руб</p> </div> <p>Сумма погашения долга по Накладной № 1 составит $1\,000 / 20 = 50,00$ Дол.</p>	<p>Пример № 2</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Документ "Поступление денег №1" Контрагент = "Купим все" Сумма = 2 000,00 Руб</p> </div> <p>Сумма погашения долга по Накладной № 1 равна Сумме долга (вал.) и составит 100,00 Дол.</p>	<p>Пример №3</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Документ "Поступление денег №2" Контрагент = "Купим все" Сумма = 3 000,00 Руб</p> </div> <p>Сумма погашения долга по Накладной № 1 равна Сумме долга (вал.) и составит 100,00 Дол. Сумма погашения долга по Накладной № 2 составит $1\,000 / 30 = 33,33$ Евро.</p>
	<p>Обращаем внимание, что с точки зрения определения суммы погашения долга варианты 2 и 3 при программной реализации можно объединить в одно условие: если сумма оплаты больше или равна рублевому эквиваленту долга, то сумма погашения равна остатку долга.</p>	

Из схемы на Рисунке № 1 видим, что для того, чтобы полностью погасить задолженность по Накладной № 1, покупатель должен перечислить 2 000,00 руб.

А что будет, если он перечислит на несколько копеек больше или меньше? Какое поведение программы будет правильным? Рассмотрим на примерах.

Оплата на несколько копеек меньше рублевого эквивалента долга	Оплата на несколько копеек больше рублевого эквивалента долга
<p>Пример №4</p> <div data-bbox="221 495 651 730" style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p>Документ "Поступление денег №1" Контрагент = "Купим все" Сумма = 1 999,90 Руб</p> </div> <p>Сумма оплаты меньше суммы рублевого эквивалента долга.</p> <p>Согласно алгоритму сумму погашения рассчитаем по курсу: $1\,999,90 / 20 = 99,995$ Дол.</p> <p>Но, как мы знаем, при записи в ресурс регистра произойдет округление и будет записано значение 100 долларов, т.е. произойдет полное погашение задолженности.</p> <p>Это будет правильно, т.к. недостающая сумма оплаты в несколько копеек в валютном эквиваленте меньше одного цента, т.е. нельзя сказать, что остался хотя бы один цент задолженности.</p>	<p>Пример №5</p> <div data-bbox="943 495 1372 730" style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p>Документ "Поступление денег №2" Контрагент = "Купим все" Сумма = 2 000,08 Руб</p> </div> <p>Сумма оплаты больше суммы рублевого эквивалента долга.</p> <p>Согласно алгоритму сумма погашения долга будет равна остатку долга и составит 100 Дол. Остаток оплаты после погашения долга составит 8 копеек.</p> <p>Если Накладная была единственная, то на 8 копеек должно быть сделано движение по авансу.</p> <p>А что, если следующая накладная есть и для этой накладной валютный эквивалент для 8 копеек будет равен нулю?</p>

Итак, обнаружена проблема: оплата или остаток оплаты в рублях могут быть настолько маленькие, что при расчете суммы погашения долга (вал.) получаем сумму, меньшую, чем самая маленькая единица валюты. Как в примере №5: 8 копеек меньше одного цента.

Проанализируем возможные варианты решения этой проблемы.

Вариант №1: «присоединять» их к предыдущему погашению задолженности (в примере №5 списать 2 000,08 рублей, а не 2 000 рублей). Будет неверно, т.к. следующая накладная может быть в рублях и аванс в 8 копеек без проблем на нее зачтется. И даже если бы «присоединили», то для оплаты в 2 000,10 рублей (100,01 доллар по курсу 20) совершенно правильно остался бы аванс в 10 копеек. Если бы следующая накладная была в евро, например, 10 копеек по курсу 30 рублей составили бы меньше одного евро-цента – имеем все ту же проблему.

Вариант №2: просто списать 8 копеек на следующую накладную и считать, что произошел зачет с суммой погашения 0 валюте. Ноль так ноль, все честно рассчитано по курсу. А теперь гипотетически представим, что покупатель сделал 10 таких оплат по 8 копеек, а это уже математически вполне значимая сумма для погашения долга $0,80/20 = 4$ цента. Но при таком подходе, если между оплатами были накладные, окажется, что каждая накладная спишет такую оплату, и задолженность при этом по накладным не уменьшится, т.е. как будто никаких оплат и не было. Такое поведение программы кажется сомнительным.

Вариант №3: оставлять такие «хвостики» оплат в виде авансов. Они будут накапливаться до тех пор, пока не накопится значимая сумма для какой-либо валюты из очередной накладной. По условию задачи авансы нужно учитывать в целом по контрагенту, а не в разрезе поступивших оплат. Как только сумма станет значимой, произойдет погашение аванса одной суммой. В итоге все хорошо: не будет никаких исчезнувших копеек. Это и есть правильный вариант решения проблемы слишком маленькой оплаты.

Листинг кода обработки проведения документа «Поступление оплаты»:

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

```
// 1. Подготовка наборов записей регистра
Движения.Взаиморасчеты.Записывать = Истина;
Движения.Записать();

// 2. Восстановление для свойства набора движений Записывать значения Истина
Движения.Взаиморасчеты.Записывать = Истина;

// 3. Запрос по остаткам задолженности по накладным
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|      ВзаиморасчетыОстатки.ДокументРасчетов,
|      ВзаиморасчетыОстатки.СуммаОстаток КАК СуммаДолгаВал,
|      ВзаиморасчетыОстатки.Валюта,
|      ЕСТЬNULL(КурсыВалютСрезПоследних.Курс, 1) КАК Курс,
|      ЕСТЬNULL(КурсыВалютСрезПоследних.Курс, 1) * ВзаиморасчетыОстатки.СуммаОстаток КАК
СуммаДолгаРуб
|ИЗ
|      РегистрНакопления.Взаиморасчеты.Остатки(&МоментВремени, Контрагент = &Контрагент) КАК
ВзаиморасчетыОстатки
|      ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.КурсыВалют.СрезПоследних(&МоментВремени, )
КАК КурсыВалютСрезПоследних
|      ПО ВзаиморасчетыОстатки.Валюта = КурсыВалютСрезПоследних.Валюта
|ГДЕ
|      ВзаиморасчетыОстатки.СуммаОстаток > 0
|
|УПОРЯДОЧИТЬ ПО
|      ВзаиморасчетыОстатки.ДокументРасчетов.МоментВремени";
```

//4. Установка параметров запроса

Запрос.УстановитьПараметр("Контрагент", Контрагент);

Запрос.УстановитьПараметр("МоментВремени", МоментВремени());

// 5. Переменная для отслеживания нераспределенного остатка оплаты

ОплатаКРаспределению = Сумма;

// 6. Выполняем запрос, обходим выборку по накладным и

//формируем движения по погашению задолженности

РезультатЗапроса = Запрос.Выполнить();

Выборка = РезультатЗапроса.Выбрать();

Пока Выборка.Следующий() И ОплатаКРаспределению > 0 Цикл

СуммаПогашенияДолгаРуб = МИН(Выборка.СуммаДолгаРуб, ОплатаКРаспределению);

//7.1 полное погашение задолженности по накладной

Если СуммаПогашенияДолгаРуб=Выборка.СуммаДолгаРуб Тогда

СуммаПогашенияДолгаВал = Выборка.СуммаДолгаВал;

Иначе //7.2 частичное погашение задолженности по накладной

//сразу выполним округления до 2 знаков, чтобы выяснить

//значимая ли в валюте сумма погашения получилась

СуммаПогашенияДолгаВал = Окр(СуммаПогашенияДолгаРуб/Выборка.Курс,2);

КонецЕсли;

//8. добавление движения в регистр накопления, только если

//валютная сумма погашения долга не равна нулю

Если СуммаПогашенияДолгаВал > 0 Тогда

Движение = Движения.Взаиморасчеты.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Контрагент = Контрагент;

Движение.ДокументРасчетов = Выборка.ДокументРасчетов;

Движение.Сумма = СуммаПогашенияДолгаВал;

Движение.Валюта = Выборка.Валюта;

//9. уменьшаем сумму нераспределенной оплаты на сумму погашения долга

ОплатаКРаспределению = ОплатаКРаспределению - СуммаПогашенияДолгаРуб;

КонецЕсли;

КонецЦикла;

//10. если после погашения всех имеющихся задолженностей

//остался нераспределенный остаток, сформируем движение по авансу

Если ОплатаКРаспределению > 0 Тогда

Движение = Движения.Взаиморасчеты.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Контрагент = Контрагент;

```

Движение.Сумма = ОплатаКРаспределению;
//11. Т.к. оплаты всегда поступают в рублях, то заполним валюту
//предопределенным значением
Движение.Валюта = Справочники.Валюты.РоссийскийРубль;
Движение.ДокументРасчетов = Документы.РасходнаяНакладная.ПустаяСсылка();
КонецЕсли;

```

КонецПроцедуры

Разберем ключевые точки алгоритма

Запрос для получения данных по задолженностям в разрезе накладных (п. 3)

Источником данных для получения информации по остаткам задолженностей будет виртуальная таблица остатков РН «Взаиморасчеты». Будем использовать следующие параметры виртуальной таблицы остатков:

- В качестве периода задаем параметр *МоментВремени*
- Используем отбор по *Контрагенту*.

Делаем левое соединение с виртуальной таблицей срез последних РС «Курсы валют» по полю *Валюта*. В качестве периода для виртуальной таблицы *КурсыВалютСрезПоследних* указываем параметр *МоментВремени*. Используем левое соединение на случай, если соответствующих записей в таблице *КурсыВалютСрезПоследних* не будет. По этой же причине используем функцию *ЕстьNULL()* применительно к полю *Курс*.

Будем считать, что если курс для валюты в РС «Курсы валют» не установлен, то значит курс равен 1. Например, так будет для валюты «Российский рубль».

Сразу же в запросе рассчитываем рублевый эквивалент задолженности: *ЕстьNULL(КурсыВалютСрезПоследних.Курс, 1) * ВзаиморасчетыОстатки.СуммаОстаток*.

Тут нужно помнить, что если в РС «Курсы валют» не установить проверку заполнения для ресурса *Курс*, то даже приняв, что если курс не установлен, то он равен единице, возможность ошибки деления на ноль не исключена, т.к. пользователь сможет в режиме 1С:Предприятие задать курс, равный нулю. Поэтому такое допущение выгоднее использовать совместно с проверкой заполнения курса в РС «Курсы валют».

*Возможно использовать другое допущение: считать, что если курс валюты не задан, то он равен нулю, т.е. используем выражение *ЕстьNULL(КурсыВалютСрезПоследних.Курс, 0)*. В этом случае рублевый эквивалент любой суммы задолженности в валюте с незадаанным курсом будет равен нулю. Соответственно, необходимо позаботится в алгоритме о следующих моментах:*

1. Если рублевый эквивалент задолженности равен нулю, то ее не нужно погашать оплатой, т.к. невозможно определить сумму погашения
2. Не допустить ошибку деления на нуль.

В режиме «1С:Предприятие» для predetermined валюты «Российский рубль» нужно ввести курс, равный 1.

Упорядочим результат запроса по полю *ДокументРасчетов.МоментВремени*, чтобы реализовать требование задачи выполнять погашение задолженности по методу FIFO.

Используем именно *МоментВремени*, а не *Дату* документа, т.к. может быть внесено несколько документов с одной датой и временем, отличаться документы будут только моментом времени.

Алгоритм погашения задолженности (п. 5, 6, 7, 8, 9)

Алгоритм аналогичен тому, который разбирали в главе «**22. Алгоритм автоматического зачета оплат и/или погашения задолженности при выполнении взаиморасчетов**». Поэтому подробный разбор кода алгоритма делать не будем. Отличается алгоритм только тем, что сравниваем остаток нераспределенной оплаты не с долгом по накладной, а с рублевым эквивалентом долга. Также при частичном погашении задолженности рассчитываем сумму погашения в валюте по курсу валюты.

Обратим внимание, что проверка курса валюты на равенство нулю не требуется, т.к. в РС «Курсы валют» нулевой курс ввести нельзя, поскольку установлена проверка заполнения. А при отсутствии записи в регистре для валюты в запросе устанавливаем, что курс равен 1.

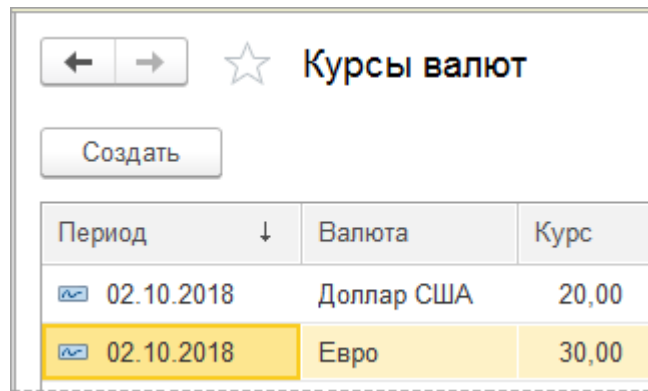
Кроме того, *СуммаПогашенияДолгаРуб* может быть настолько маленькой, что при расчете по курсу *СуммаПогашенияДолгаВал* будет равна нулю после округления до 2 знаков после запятой. Поэтому выполняем проверку и не делаем в этом случае движение по погашению задолженности. Остаток нераспределенной оплаты в этом случае также не уменьшаем, т.к. не было погашения долга.

Алгоритм формирования движения по авансу (п. 10)

Алгоритм аналогичен разобранному в разделе «**22. Алгоритм автоматического зачета оплат и/или погашения задолженности при выполнении взаиморасчетов**» за исключением одного момента: по условию задачи аванс не нужно формировать в разрезе накладных, поэтому измерение *ДокументРасчетов* заполняем пустой ссылкой на документ «Расходная накладная». По условию оплата поступает всегда в рублях, поэтому измерение *Валюта* заполняем predetermined значением *Справочник.Валюты.РоссийскийРубль*.

Проверка погашения задолженности в режиме «1С:Предприятие»

Для тестовых примеров заполним курсы валют следующим образом:



Период	Валюта	Курс
02.10.2018	Доллар США	20,00
02.10.2018	Евро	30,00

Рисунок 2 – Курсы валют для тестового примера

Для того, чтобы было удобно себя проверять в «Консоли запросов», создадим запрос, который сформирует нам такую же таблицу, как на рисунке № 1:

Консоль запросов : E:\Запросы\СостояниеЗадолженности.sel

Действия ▾ | | Выполнить **& Параметры** ?

Запрос

Запросы

ВЫБРАТЬ

ВзаиморасчетыОстатки.Контрагент,
 ВзаиморасчетыОстатки.ДокументРасчетов,
 ВзаиморасчетыОстатки.СуммаОстаток КАК СуммаДолгаВал,
 ВзаиморасчетыОстатки.Валюта,
 ЕСТЬNULL (КурсыВалютСрезПоследних.Курс, 1) КАК Курс,
 ЕСТЬNULL (КурсыВалютСрезПоследних.Курс, 1)
 * ВзаиморасчетыОстатки.СуммаОстаток КАК СуммаДолгаРуб

ИЗ

РегистрНакопления.Взаиморасчеты.Остатки (&Период,) КАК ВзаиморасчетыОстатки
 ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.КурсыВалют.СрезПоследних (&Период,)
 КАК КурсыВалютСрезПоследних
 ПО ВзаиморасчетыОстатки.Валюта = КурсыВалютСрезПоследних.Валюта

УПОРЯДОЧИТЬ ПО

ВзаиморасчетыОстатки.ДокументРасчетов.МоментВремени

Способ выгрузки:

Контрагент	ДокументРасчетов	СуммаДолгаВал	Валюта	Курс	СуммаДолгаРуб
Купим всё	Расходная накладная 000000001 от 02.10.2018 8:07:22	100,00	Доллар США	20	2 000
Купим всё	Расходная накладная 000000002 от 02.10.2018 8:08:50	100,00	Евро	30	3 000
Купим всё	Расходная накладная 000000003 от 02.10.2018 8:09:10	200,00	Доллар США	20	4 000

Рисунок 3 – Запрос для получения данных о состоянии задолженности в разрезе накладных (состояние остатков по РН «Взаиморасчеты»)

Тестовый пример № 1

Внесем документ оплаты на 3 000,00 руб. Ситуация аналогична Примеру № 3 в таблице вариантов оплаты.

← → ☆ Поступление денег 000000001 от 02.10.2018 10:19:52

Основное Взаиморасчеты

Провести и закрыть Записать Провести

Номер: 000000001

Дата: 02.10.2018 10:19:52

Контрагент: Купим всё

Сумма: 3 000,00

Рисунок 4 – Документ «Поступление денег № 1»

Движения документа после проведения:

Период	Регистратор	Номер строки	Контрагент	Документ расчетов	Валюта	Сумма
- 02.10.2018 1...	Поступление денег 000000001 о...	1	Купим всё	Расходная накладная 000000001 от 02.10.201...	Доллар США	100,00
- 02.10.2018 1...	Поступление денег 000000001 о...	2	Купим всё	Расходная накладная 000000002 от 02.10.201...	Евро	33,33

Рисунок 5 – Движения документа «Поступление денег № 1»

Видим, что суммы погашения долга рассчитались верно.

В «Консоли запросов» посмотрим состояние задолженности, выполнив запрос, приведенный на рис. № 3:

Контрагент	Документ	Сумма	Валюта	Курс	Сумма
Купим всё	Расходная накладная 000000002 от 02.10.2018 ...	66,67	Евро	30	2 000,1
Купим всё	Расходная накладная 000000003 от 02.10.2018 ...	200,00	Доллар США	20	4 000

Рисунок 6 – Состояние остатков по РН «Взаиморасчеты»

Создадим еще один документ оплаты на 10 000 рублей, чтобы проверить, как сформируются движения по авансу:

← → ☆ Поступление денег 000000002 от 02.10.2018 10:20:31

Основное Взаиморасчеты

Провести и закрыть Записать Провести

Номер: 000000002

Дата: 02.10.2018 10:20:31

Контрагент: Купим всё

Сумма: 10 000,00

Рисунок 7 – Документ «Поступление денег № 2»

Движения:

← → ☆ Поступление денег 000000002 от 02.10.2018 10:20:31

Основное Взаиморасчеты

Взаиморасчеты

Период	↓	Регистратор	Номер строки	Контрагент	Документ расчетов	Валюта	Сумма
- 02.10.2018 1...		Поступление денег 000000002 о...	1	Купим всё	Расходная накладная 000000002 от 02.10.201...	Евро	66,67
- 02.10.2018 1...		Поступление денег 000000002 о...	2	Купим всё	Расходная накладная 000000003 от 02.10.201...	Доллар США	200,00
- 02.10.2018 1...		Поступление денег 000000002 о...	3	Купим всё		Российский рубль	3 999,90

Рисунок 8 – Движения документа «Поступление денег № 2»

В строке № 1 движений видим, что оплата полностью погасила долг по «Расходной накладной № 2».

В строке № 2 движений – полное погашение долга по «Расходной накладной № 2».

Строка № 3 движений отражает формирование аванса. Документ расчетов не заполнен, валюта – рубль. Все как требуется по условию задачи.

Тестовый пример № 2

Промоделируем ситуацию для примера № 5 из таблицы № 2.

Внесем расходную накладную, но теперь по контрагенту «Купим всегда»:

← → ☆ Расходная накладная 000000005 от 02.10.2018 8:09:11

Основное Взаиморасчеты

Провести и закрыть Записать Провести

Номер: 000000005

Дата: 02.10.2018 8:09:11

Контрагент: Купим всегда

Сумма по документу: 100,00

Валюта: Доллар США

Рисунок 9 – Документ «Расходная накладная № 5»

Проводим документ и смотрим его движения:

← → ☆ Расходная накладная 000000005 от 02.10.2018 8:09:11

Основное Взаиморасчеты

Взаиморасчеты

Период	Регистратор	Номер	Контрагент	Документ расчетов	Валюта	Сумма
+ 02.10.2018 8:09:11	Расходная накладная 0...	1	Купим все...	Расходная накладная 000000005 от 0...	Доллар США	100,00

Рисунок 10 – Движения документа «Расходная накладная № 5»

В движениях видим, что сформировалась задолженность по накладной в размере 100 долларов.

Теперь внесем оплату на сумму 2 000,08 рублей:

← → ☆ Поступление денег 000000004 от 02.10.2018 11:20:32

Основное Взаиморасчеты

Провести и закрыть Записать Провести

Номер: 000000004

Дата: 02.10.2018 11:20:32

Контрагент: Купим всегда

Сумма: 2 000,08

Рисунок 11 – Документ «Поступление денег № 4»

Проводим оплату и смотрим движения документа:

Период	Регистратор	Номер...	Контрагент	Документ расчетов	Валюта	Сумма
- 02.10.2018 11:20:32	Поступление денег 000000004 ...	1	Купим всегда	Расходная накладная 000000005 от 1...	Доллар США	100,00
- 02.10.2018 11:20:32	Поступление денег 000000004 ...	2	Купим всегда		Российский рубль	0,08

Рисунок 12 – Движения документа «Поступление денег № 4»

Строка № 1 движений отражает полное погашение оплатой долга по «Расходной накладной № 5».

Строка № 3 движений отражает формирование аванса на 8 копеек. Документ расчетов не заполнен, валюта – рубль.

Алгоритм проведения отработал верно.

Смотрим остатки задолженности в «Консоли запросов»:

Контрагент	ДокументРасчетов	СуммаДолгаВал	Валюта	Курс	СуммаДолгаРуб
Купим всё		-3 999,90	Российский рубль		
Купим всегда		-0,08	Российский рубль		

Рисунок 13 – Состояние остатков по РН «Взаиморасчеты»

Видим, что авансы представляют собой отрицательные суммы в таблице остатков.

Подведем итоги

Мы разобрали технологию погашения задолженностей в разных валютах при поступлении оплаты в рублях.

В следующей главе продолжим решение практического примера. Разберем, как реализовать зачет аванса в рублях на накладную в валюте.

27. Как корректно зачесть аванс в рублях на накладную в валюте

Продолжаем разбор практического примера задачи с переменным списком валют.

Осталось доработать обработку проведения «Расходной накладной» для выполнения следующего требования задачи: если есть аванс, то он засчитывается как оплата накладной по курсу на дату накладной.

Проведение документа «Расходная накладная»

Движение по отражению задолженности оставим без изменений. Оно должно формироваться всегда на полную сумму накладной, независимо ни от каких условий задачи по части авансов. Это важно. Сейчас разберем почему.

Есть соблазн формировать движение по задолженности только на ту часть суммы, которая осталась неоплаченной после зачета аванса. Но так делать нельзя – потеряем часть информации по взаиморасчетам в разрезе накладной. А если аванс будет больше суммы накладной? Тогда накладная сделает только движение по списанию аванса. В качестве *Документа расчетов* накладная не будет присутствовать ни в одном движении: ни по возникновению задолженности, ни по оплате. Событие было, документ был – следов в РН «Взаиморасчеты» в качестве *Документа расчетов* нет. Только в качестве регистратора по списанию аванса. Как в такой ситуации получить информацию для отчета, сколько было оплачено по этой накладной, не обращаясь к таблице движений?

Если сформировать в «Консоли запросов» запрос по остаткам и оборотам по РН «Взаиморасчеты» без указания периода, то приход по регистру в разрезе расчетных документов должен совпадать с суммой накладной, расход будет показывать оплаты:

Консоль запросов : E:\Запросы\Взаиморасчеты.sel

Действия ▾ | | Выполнить & Параметры ?

Запрос

Запросы

ВЫБРАТЬ

ВзаиморасчетыОстаткиИОбороты.Контрагент,
ВзаиморасчетыОстаткиИОбороты.ДокументРасчетов,
ВзаиморасчетыОстаткиИОбороты.ДокументРасчетов.СуммаПоДокументу
КАК СуммаДокумента,
ВзаиморасчетыОстаткиИОбороты.Валюта,
ВзаиморасчетыОстаткиИОбороты.СуммаНачальныйОстаток **КАК** НачОст,
ВзаиморасчетыОстаткиИОбороты.СуммаПриход **КАК** Приход,
ВзаиморасчетыОстаткиИОбороты.СуммаРасход **КАК** Расход,
ВзаиморасчетыОстаткиИОбороты.СуммаКонечныйОстаток **КАК** КонОст

ИЗ

РегистрНакопления.Взаиморасчеты.ОстаткиИОбороты(, , , ,)
КАК ВзаиморасчетыОстаткиИОбороты

Способ выгрузки:

Контрагент	ДокументРасчетов	СуммаДокумента	Валюта	НачОст	Приход	Расход	КонОст
Купим всё			Российский рубль			3 999,90	-3 999,90
Купим всё	Расходная накладная 000000001...	100,00	Доллар США		100,00	100,00	
Купим всё	Расходная накладная 000000002...	100,00	Евро		100,00	100,00	
Купим всё	Расходная накладная 000000003...	200,00	Доллар США		200,00	200,00	
Купим всегда			Российский рубль			0,08	-0,08
Купим всегда	Расходная накладная 000000005...	100,00	Доллар США		100,00	100,00	

Рисунок 1 – Пример результата запроса по остаткам и оборотам РН «Взаиморасчеты»

В обработке проведения «Расходной накладной» добавим формирование следующих движений:

- По списанию аванса. Аванс в рублях, поэтому движение по списанию аванса – это приход в рублях. *Документ расчетов* заполняем пустой ссылкой, чтобы регистр закрылся в ноль в разрезе измерений
- По зачету аванса. Взаиморасчеты по накладным ведутся в валюте, поэтому движение по зачету аванса будем делать в валюте накладной. Аванс – это вариант оплаты, поэтому вид движения – расход. В качестве документа расчетов указываем ссылку на «Расходную накладную», которую проводим.

Осталось выяснить, как правильно определить рублевую сумму списания аванса и валютную сумму зачета аванса.

Для начала определимся, в какой валюте будем сравнивать суммы, чтобы понять, хватает ли аванса для оплаты накладной. Рассмотрим на примере.

Накладная №4
Контрагент = "Купим все"
Сумма = 129,03 Евро

Курс валюты на дату накладной

Валюта	Курс
Евро	31

Рублевый эквивалент суммы накладной
равен $129,03 \times 31 = 3\,999,93$ руб.

Ситуация с остатком аванса №1

Таблица остатков РН "Взаиморасчеты"

Контрагент	Расчетный документ	Сумма долга (вал.)	Валюта
Купим все		- 3 999,90	Рубль

Сумма аванса в валюте накладной равна
 $3\,999,90 / 31 = 129,02903...$ Евро

Ситуация с остатком аванса №2

Таблица остатков РН "Взаиморасчеты"

Контрагент	Расчетный документ	Сумма долга (вал.)	Валюта
Купим все		- 4 000,08	Рубль

Сумма аванса в валюте накладной равна
 $4\,000,08 / 31 = 129,03483...$ Евро

Рисунок 2 – Примеры ситуаций для зачета аванса в накладной

Из рисунка видим, что, независимо от того, будем мы сравнивать суммы в валюте или суммы в рублях, выводы получаются одинаковые: в ситуации № 1 сумма аванса меньше суммы накладной, в ситуации № 2 сумма аванса больше суммы накладной. Выберем вариант сравнения сумм в рублях – он более симпатичен отсутствием операции деления.

При зачете аванса оплаты возможны три варианта:

- Аванс **меньше** рублевого эквивалента суммы накладной
- Аванс **равен** рублевому эквиваленту суммы накладной
- Аванс **больше** рублевого эквивалента суммы накладной.

Разберем, как определить сумму погашения долга в валюте для каждого из вариантов.

Вариант №1	Вариант №2	Вариант №3
Аванс меньше рублевого эквивалента суммы накладной	Аванс равен рублевому эквиваленту суммы накладной	Аванс больше рублевого эквивалента суммы накладной
<p>Вся сумма аванса пойдет в зачет оплаты по накладной.</p> <p>Накладная будет оплачена частично.</p> <p>Сумма списания аванса (руб.) будет равна остатку аванса. Сумму зачета (вал.) рассчитываем по курсу исходя из рублевой суммы аванса.</p>	<p>Вся сумма аванса пойдет в зачет оплаты по накладной.</p> <p>Накладная будет оплачена полностью.</p> <p>Сумма списания аванса (руб.) будет равна остатку аванса. Обратим внимание, что в данном варианте остаток аванса равен рублевому эквиваленту суммы накладной.</p> <p>Сумма зачета (вал.) будет равна сумме накладной.</p>	<p>Часть аванса пойдет в зачет оплаты по накладной. Останется остаток аванса.</p> <p>Накладная будет оплачена полностью.</p> <p>Сумма зачета (вал.) будет равна сумме накладной.</p> <p>Сумма списания аванса (руб.) равна рублевому эквиваленту суммы накладной .</p>
	<p>Заметим, что с точки зрения определения суммы зачета (вал.) и суммы списания аванса (руб.) варианты 2 и 3 при программной реализации можно объединить в одно условие: если сумма аванса больше или равна рублевому эквиваленту суммы накладной, то сумма зачета (вал.) равна сумме накладной, а сумма списания аванса (руб.) равна рублевому эквиваленту накладной.</p>	
Состояние взаиморасчетов после проведения накладной по РН «Взаиморасчеты» должно быть следующим		
<p>Есть остаток неоплаченной задолженности по накладной.</p> <p>Остатка аванса быть не должно.</p>	<p>Остатка задолженности по накладной быть не должно.</p> <p>Остатка аванса быть не должно.</p>	<p>Остатка задолженности по накладной быть не должно.</p> <p>Есть остаток аванса.</p>

Ситуация № 1 на рисунке № 2 – сумма аванса меньше суммы накладной. Сумма списания аванса (руб.) будет равна остатку аванса, а сумма зачета аванса (вал.) после расчета по курсу $3\,999,90 / 31 = 129,0293...$ и записи в ресурс регистра окажется равной сумме накладной. Накладная окажется полностью оплачена, несмотря на то, что сумма аванса на 3 копейки меньше рублевого эквивалента суммы накладной, а это меньше одного цента.

В ситуации № 2 на рисунке № 2 сумма аванса больше суммы накладной. Остаток аванса равен разнице между суммой аванса и рублевым эквивалентом суммы накладной и должен составить 15 копеек ($4\,000,08 - 3\,999,93 = 0,15$).

При рассмотрении тестового примера № 2 в прошлой главе по контрагенту «Купим всегда» остался аванс в 8 копеек. Также в прошлой главе определились, что списание и зачет такого аванса на накладную нужно производить только в том случае, если при пересчете в валюту сумма зачета будет больше нуля. Если нет, то такой аванс должен остаться.

Листинг кода обработки проведения документа «Расходная накладная»:

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

```
//1. Подготовка наборов записей регистра
Движения.Взаиморасчеты.Записывать = Истина;
Движения.Записать();

// 2. Восстановление для свойства набора движений Записывать значения Истина
Движения.Взаиморасчеты.Записывать = Истина;

// 3. Запрос по остатку аванса покупателя
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|      ВзаиморасчетыОстатки.СуммаОстаток*-1 КАК СуммаАвансаРуб
|ИЗ
|      РегистрНакопления.Взаиморасчеты.Остатки(&МоментВремени, Контрагент = &Контрагент) КАК
ВзаиморасчетыОстатки
|ГДЕ
|      ВзаиморасчетыОстатки.СуммаОстаток < 0
|
|УПОРЯДОЧИТЬ ПО
|      ВзаиморасчетыОстатки.ДокументРасчетов.МоментВремени";

//4. Установка параметров запроса и выполнение запроса
Запрос.УстановитьПараметр("Контрагент", Контрагент);

Запрос.УстановитьПараметр("МоментВремени", МоментВремени());
РезультатЗапроса = Запрос.Выполнить();

//5.Сформируем движение по задолженности
Движение = Движения.Взаиморасчеты.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Валюта = Валюта;
Движение.Контрагент = Контрагент;
Движение.ДокументРасчетов = Ссылка;
Движение.Сумма = СуммаПоДокументу;
```

```
//6.Зачет аванса
//Аванс есть, если результат запроса аванса не пустой
Если НЕ РезультатЗапроса.Пустой() Тогда
    //6.1 по авансу запись может быть только одна в регистре остатков
    //поэтому цикл не нужен
    Выборка = РезультатЗапроса.Выбрать();
    Выборка.Следующий();

    //6.2 получаем курс валюты на дату документа и рассчитываем
    // рублевый эквивалент суммы документа
    Курс = РегистрыСведений.КурсыВалют.ПолучитьПоследнее(Дата,Новый
Структура("Валюта",Валюта)).Курс;
    Курс = ?(Курс=0,1,Курс);
    СуммаДокументаРуб = СуммаПоДокументу*Курс;

    //6.3 Определяем сумму списания аванса в рублях и
    //сумму зачета аванса по накладной в валюте
    //в зависимости от условия превышает ли сумма аванса
    //рублевый эквивалент суммы накладной
    Если Выборка.СуммаАвансаРуб >= СуммаДокументаРуб Тогда
        СуммаЗачетаАвансаВал = СуммаПоДокументу;
        //сумма списания аванса равна рублевому эквиваленту
        //суммы накладной
        СуммаСписанияАвансаРуб = СуммаДокументаРуб;
    Иначе //сумма аванса меньше суммы рублевого эквивалента по накладной
        //сумма списания аванса равна остатку аванса
        СуммаСписанияАвансаРуб = Выборка.СуммаАвансаРуб;
        //сумму зачета рассчитаем по курсу из суммы аванса
        //сразу выполним округление с точность 2 знака
        СуммаЗачетаАвансаВал = Окр(СуммаСписанияАвансаРуб/Курс,2);
    КонецЕсли;

    //6.4 формируем движения по списанию аванса и зачету аванса
    //только если округленная сумма зачета аванса в валюте больше нуля
    Если СуммаЗачетаАвансаВал > 0 Тогда
        //формируем движение по списанию аванса
        Движение = Движения.Взаиморасчеты.ДобавитьПриход();
        Движение.Период = Дата;
        Движение.Контрагент = Контрагент;
        Движение.Сумма = СуммаСписанияАвансаРуб;
        Движение.ДокументРасчетов = Документы.РасходнаяНакладная.ПустаяСсылка();
        Движение.Валюта = Справочники.Валюты.РоссийскийРубль;

        //формируем движение по зачету аванса в качестве оплаты по накладной
        Движение = Движения.Взаиморасчеты.ДобавитьРасход();
        Движение.Период = Дата;
        Движение.Контрагент = Контрагент;
        Движение.ДокументРасчетов = Ссылка;
        Движение.Сумма = СуммаЗачетаАвансаВал;
        Движение.Валюта = Валюта;
    КонецЕсли;
КонецЕсли;
```

КонецПроцедуры

Разберем ключевые точки алгоритма.

Запрос получения данных по авансу (п. 3)

Источником данных для получения информации по остатку аванса будет виртуальная таблица остатков РН «Взаиморасчеты». Будем использовать следующие параметры виртуальной таблицы остатков: в качестве периода задаем параметр *МоментВремени* и используем отбор по *Контрагенту*.

Так как документ «ПоступлениеДенег» формирует расход в РН «Взаиморасчеты», то остаток аванса будет отрицательным. Установим в запросе условие *СуммаОстаток<0*.

Для того, чтобы в коде процедуры не работать с отрицательными числами, умножим значение поля «СуммаОстаток» на -1 .

Формирование движений по РН «Взаиморасчеты»

п.5 – Формирование прихода на сумму документа – это отразили задолженность по накладной.

п. 6 – Алгоритм зачета аванса.

п.6.1 – Аванс по условию формируется сводно по контрагенту без детализации по документам расчетов. Поэтому если аванс есть, то в таблице результата запроса будет единственная запись.

п. 6.2 – Получаем курс валюты на дату документа и рассчитываем рублевый эквивалент суммы документа. Для получения курса валюты используем объектную модель. При этом курс, равный нулю, можем получить только в случае, если для валюты отсутствуют записи в РС «Курсы валют». Установить в регистре нулевой курс нельзя, т.к. установлена проверка заполнения. Поэтому если курс равен нулю, то примем, что курс валюты равен 1. Соответственно, проверка деления на ноль при делении на курс будет не нужна.

Можно было использовать допущение, что если курс валюты не задан, то он равен нулю. В этом случае необходимо позаботиться в алгоритме о следующих моментах:

1. На задолженность в валюте с неустановленным курсом не нужно зачитывать аванс
2. Не допустить ошибку деления на ноль.

В режиме «1С:Предприятие» для predetermined валюты «Российский рубль» нужно ввести курс, равный 1.

п.6.3, 6.4, 6.5 – Определяем сумму списания аванса в рублях и сумму зачета аванса в валюте согласно логике, описанной выше. Формируем движения по списанию и зачету аванса только в том случае, если сумма зачета аванса в валюте, округленная до 2 знаков после запятой, больше нуля.

Проверка зачета аванса в режиме «1С:Предприятие»

Решение задачи продолжаем на базе с данными, которую использовали в предыдущем блоке *«Погашение задолженности в валюте оплатой в рублях»*.

Ситуация в базе данных после предыдущего тестового примера сформировалась следующая:

Контрагент	ДокументРасчетов	СуммаДолгаВал	Валюта	Курс	СуммаДолгаРуб
Купим всё		-3 999,90	Российский рубль	1	-3 999,9
Купим всегда		-0,08	Российский рубль	1	-0,08

Рисунок 3 – Остатки авансов

Тестовый пример № 1

Курс Евро на 03.10.2018 установим 31 рубль:

← → ☆ Курсы валют		
Создать		
Период ↓	Валюта	Курс
02.10.2018	Доллар США	20,00
02.10.2018	Евро	30,00
03.10.2018	Евро	31,00

Рисунок 4 – Курс валюты для тестового примера

Внесем расходную накладную по контрагенту «Купим все» на сумму 129,03 рублей:

← → ☆ Расходная накладная 000000004 от 03.10.2018 16:08:28

Основное Взаиморасчеты

Провести и закрыть Записать Провести

Номер: 000000004

Дата: 03.10.2018 16:08:28

Контрагент: Купим всё

Сумма по документу: 129.03

Валюта: Евро

Рисунок 5 – Документ «Расходная накладная № 4»

Проведем документ и посмотрим, какие он сформировал движения:

Период	Регистратор	Номер	Контрагент	Документ расчетов	Валюта	Сумма
+ 03.10.2018 16:08:28	Расходная накладная 000000004...	1	Купим всё	Расходная накладная 000000004...	Евро	129,03
+ 03.10.2018 16:08:28	Расходная накладная 000000004...	2	Купим всё		Российский рубль	3 999,90
- 03.10.2018 16:08:28	Расходная накладная 000000004...	3	Купим всё	Расходная накладная 000000004...	Евро	129,03

Рисунок 6 – Движения документа «Расходная накладная № 4»

Видим, что движения сформировались так как нужно.

Строка № 1 в движениях отражает возникновение долга контрагента за отгруженный товар на сумму «Расходной накладной».

Строка № 2 отражает списание аванса в размере рублевого эквивалента суммы, которую зачили в качестве аванса.

Строка № 3 отражает сумму аванса, которую зачислили как оплату по накладной.

Тестовый пример № 2

Теперь внесем накладную по контрагенту «Купим всегда», чтобы проверить, что произойдет с суммой аванса в 8 копеек:

Рисунок 7 – Документ «Расходная накладная № 6»

Проводим документ и смотрим его движения:

Период	Регистратор	Н...	Контрагент	Документ расчетов	Валюта	Сумма
+ 03.10.2018...	Расходная накладная 000000006 от ...	1	Купим всегда	Расходная накладная 000000006 от...	Доллар США	10,00

Рисунок 8 – Движения документа «Расходная накладная № 6»

Видим, что несмотря на наличие аванса в 8 копеек, движения по списанию и зачету не сформировались, т.к. валютный эквивалент этой суммы равен нулю.

Проверим состояние остатков по РН «Взаиморасчеты» с помощью запроса в «Консоли запросов» (тест запроса приведен на рисунке 1):

Контрагент	Документ	Расчетов	Сумма	Документа	Валюта	НачОст	Приход	Расход	КонОст
Купим всё					Российский рубль			3 999,90	-3 999,90
Купим всё	Расходная накладная 000000001 ...		100,00		Доллар США		100,00	100,00	
Купим всё	Расходная накладная 000000002 ...		100,00		Евро		100,00	100,00	
Купим всё	Расходная накладная 000000003 ...		200,00		Доллар США		200,00	200,00	
Купим всегда					Российский рубль			0,08	-0,08
Купим всегда	Расходная накладная 000000005 ...		100,00		Доллар США		100,00	100,00	
Купим всегда	Расходная накладная 000000006 ...		10,00		Доллар США		10,00		10,00

Рисунок 9 – Остатки и обороты по РН «Взаиморасчеты»

Видим, что по контрагенту «Купим всегда» остался аванс на 8 копеек и сформировалась задолженность на сумму документа «Расходная накладная № 6». Так и должно быть.

Теперь внесем еще одну расходную накладную по контрагенту «Купим всегда», но уже в рублях:

←

→

★

Расходная накладная 000000007 от 04.10.2018 9:41:09

Основное

Взаиморасчеты

Провести и закрыть

Записать

Провести

Номер:

000000007

Дата:

04.10.2018 9:41:09

📅

Контрагент:

Купим всегда

Сумма по документу:

1 000,00

📊

Валюта:

Российский рубль

Рисунок 10 – Документ «Расходная накладная № 7»

Проводим документ и смотрим его движения:

←

→

☆

Расходная накладная 000000007 от 04.10.2018 9:41:09

Основное

Взаиморасчеты

Взаиморасчеты

Период	Регистратор	Ном...	Контрагент	Документ расчетов	Валюта	Сумма
+ 04.10.2018 9:41:09	Расходная накладная 000000007 от 04.10.2018...	1	Купим всегда	Расходная накладная 000000007 от 04.10.2018...	Российский рубль	1 000,00
+ 04.10.2018 9:41:09	Расходная накладная 000000007 от 04.10.2018...	2	Купим всегда		Российский рубль	0,08
- 04.10.2018 9:41:09	Расходная накладная 000000007 от 04.10.2018...	3	Купим всегда	Расходная накладная 000000007 от 04.10.2018...	Российский рубль	0,08

Рисунок 11 – Движения документа «Расходная накладная №7»

Строка № 1 в движениях отражает возникновение долга контрагента на сумму «Расходной накладной №7».

Строка № 2 отражает списание аванса в размере 8 копеек.

Строка № 3 отражает зачет аванса в качестве оплаты по накладной.

Проверим теперь состояние взаиморасчетов с помощью запроса:

Контрагент	ДокументРасчетов	СуммаДокумента	Валюта	НачОст	Приход	Расход	КонОст
Купим всё			Российский рубль			3 999,90	-3 999,90
Купим всё	Расходная накладная 000000001...	100,00	Доллар США		100,00	100,00	
Купим всё	Расходная накладная 000000002...	100,00	Евро		100,00	100,00	
Купим всё	Расходная накладная 000000003...	200,00	Доллар США		200,00	200,00	
Купим всегда			Российский рубль		0,08	0,08	
Купим всегда	Расходная накладная 000000005...	100,00	Доллар США		100,00	100,00	
Купим всегда	Расходная накладная 000000006...	10,00	Доллар США		10,00		10,00
Купим всегда	Расходная накладная 000000007...	1 000,00	Российский рубль		1 000,00	0,08	999,92

Рисунок 12 – Остатки и обороты по РН «Взаиморасчеты»

Видим, что по «Расходной накладной №7» осталась задолженность за вычетом аванса. Как и требовалось.

Задача решена.

Подведем итоги

Мы разобрали технологию зачета рублевого аванса на накладную в валюте и полностью завершили разбор практического примера задачи с переменным списком валют.

Этой главой закрываем раздел «Задачи с валютами». Мы разобрали, на какие типы подразделяются такие задачи. Для каждого типа задачи рассмотрели по шагам методику решения и опробовали ее на практике. Теперь, если в экзаменационном билете достанется задача с валютой, мы знаем, как с ней справиться.

28. Как выполнить распределение дополнительных затрат

Подзадачи, в которых требуется распределить некоторые дополнительные затраты, встречаются в задачах из разделов оперативного (см. 1.15, 1.28, 1.29) и бухгалтерского учета (см. 2.7, 2.8).

Как правило, по условию задачи есть некоторый документ («Дополнительные затраты», «Распределение затрат», «Реклама» и т.п.), в котором фиксируется некоторая сумма расходов. Далее в условии задачи указывается на что нужно распределить эту сумму и каким образом (например, пропорционально количеству, себестоимости и т.п.).

Классический пример подобной задачи из реальной практики – это документ «Поступление доп. расходов», присутствующий в типовых конфигурациях 1С.

Разберем решение задачи распределения дополнительных затрат на практическом примере из оперативного учета.

Постановка задачи:

Компания занимается оптовой торговлей. Поступление товаров отражается документом «Приходная накладная», продажа – «Расходная накладная»...

(особенности учета товаров и расчета себестоимости опустим – могут быть различными, на решение задачи распределения расходов не повлияют)

*... После продажи в систему вносят дополнительную информацию о сумме затрат на доставку товара с помощью документа «Дополнительные затраты». В документе указывается сумма доставки и накладная. Сумму доставки нужно **распределить на товары накладной пропорционально проданному количеству**. Необходимо создать отчет по продажам за период, в котором будет указано проданное количество, **себестоимость (с учётом доставки вне зависимости от того, в каком периоде были внесены данные об увеличении себестоимости)**, сумма продаж, прибыль.*

Продажи с 01.01.2018 по 31.10.2018

Номенклатура	Накладная	Кол-во	Себест-ть	Сумма продажи	Прибыль
Паркер «Golg»		7	2 266,11	4 000	1 733,89
	Расх. накладная № 1	2	1 200	2 000	800

	Расх. накладная № 2	5	1 066,11	2 000	933,89
--	---------------------	---	----------	-------	--------

Подготовка объектов метаданных

Сумма доставки распределяется на товары «Расходной накладной». В каркасной конфигурации этот документ есть.

Понадобится новый документ «Дополнительные затраты». Создаем его.

Реквизиты:

- Накладная (ДокументСсылка.РасходнаяНакладная)
- Сумма (Число 12, 2).

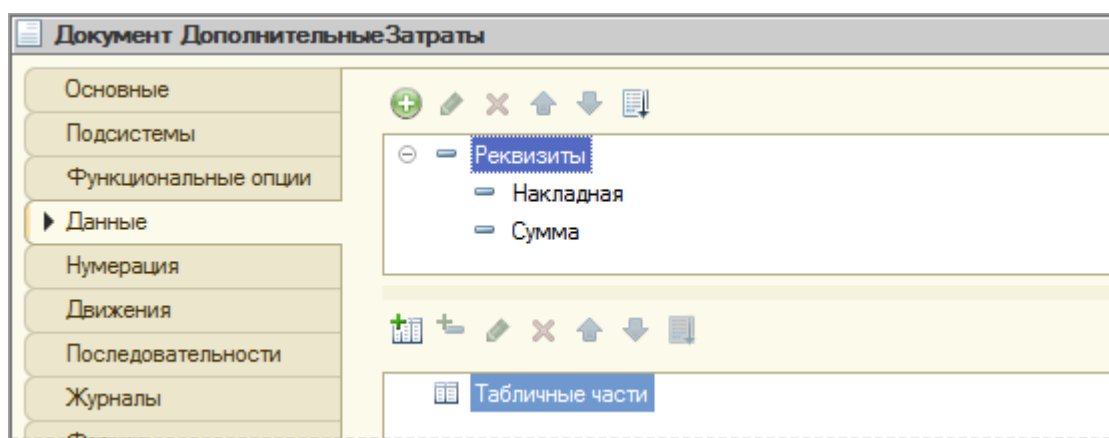


Рисунок 1 – Реквизиты документа «Дополнительные затраты»

Сумму доставки по условию нужно распределять пропорционально проданному количеству. В отчете нам необходимо выводить информацию о сумме продажи. Соответственно, нужен регистр для хранения этой информации.

Создадим новый РН «Продажи номенклатуры». Вид регистра *Обороты*, т.к. нужно просто фиксировать значения показателей по продажам – количество, сумма продажи. Остатки по этим показателям не имеют смысла.

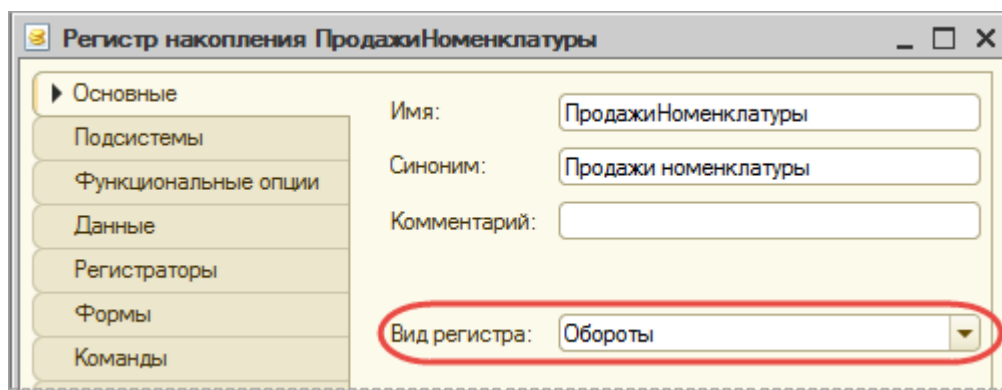


Рисунок 2 – Вид регистра РН «Продажи номенклатуры»

Информацию о продажах необходимо в отчете получать в разрезе *Номенклатуры* и *Накладной*. Для распределения дополнительных расходов данные о количестве проданных товаров нужно получать с отбором по накладной. Себестоимость номенклатуры нужно увеличивать только в разрезе указанной в документе накладной. Исходя из перечисленных соображений создаем следующие измерения регистра:

- *Номенклатура* (СправочникСсылка.Номенклатура): устанавливаем в истину флаг *Запрет незаполненных значений*
- *Накладная* (ДокументСсылка.РасходнаяНакладная): флаг *Запрет незаполненных значений* = истина.

Ресурсы:

- *Количество* (Число 10, 0)
- *Себестоимость* (Число 12, 2)
- *Сумма продажи* (Число 12, 2).

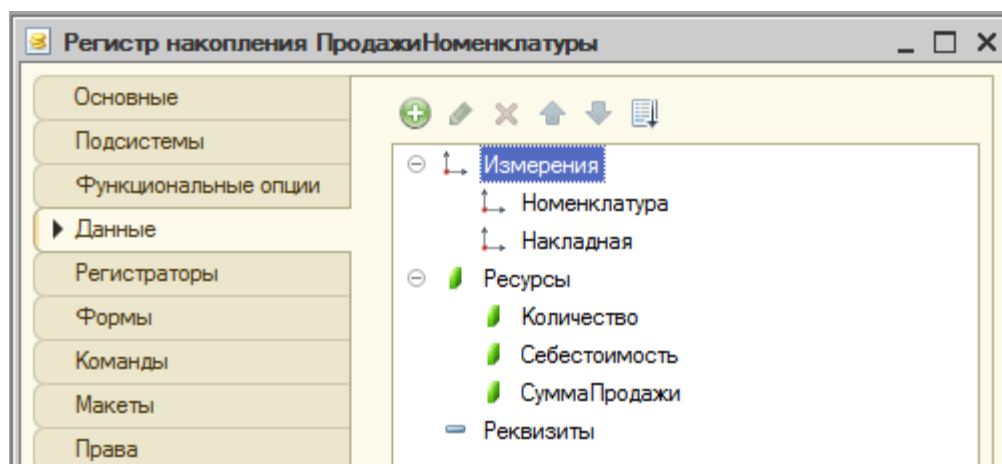


Рисунок 3 – Структура РН «Продажи номенклатуры»

Как правильно: создать ресурс Себестоимость в РН «Продажи номенклатуры» или работать с регистром остатков, на котором организован учет себестоимости?

Т.к. по условию нам нужно увеличивать себестоимость продаж, то правильный вариант решения – создать ресурс Себестоимость. Аргументы следующие:

- Для получения данных о продажах и себестоимости для отчета не придется выполнять в запросе соединение таблиц, т.к. вся необходимая информация есть в РН «Продажи номенклатуры»
- По условию задачи списание себестоимости может быть организовано по партиям, то есть в регистре учета себестоимости будет измерение «Партия». Как в этом случае увеличить себестоимость продаж на сумму доставки? Заниматься еще распределением суммы доставки по партиям? Эта проблема решается введением ресурса Себестоимость в РН «Продажи номенклатуры». При проведении документа «Расходная накладная» записываем в него информацию без учета разделения по партиям. При распределении суммы доставки на товары расходной накладной работаем только с РН «Продажи номенклатуры», поэтому никакие особенности расчета себестоимости списания на задачу распределения больше не влияют.

Регистраторы:

- «Расходная накладная»
- «Дополнительные затраты»:

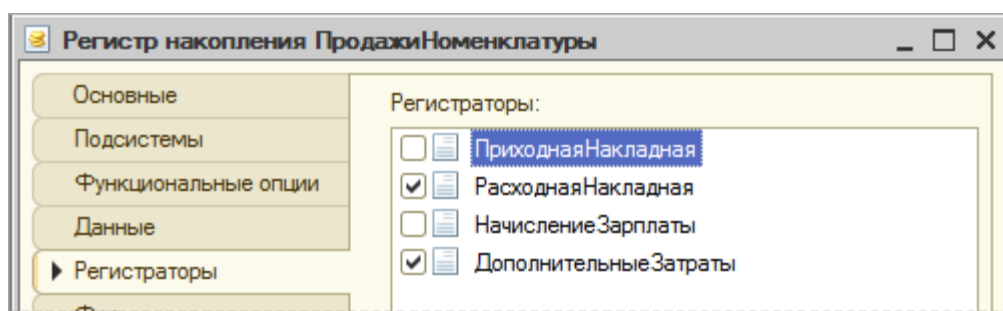


Рисунок 4 – Регистраторы РН «Продажи номенклатуры»

Проведение документа «Расходная накладная»

При проведении «Расходной накладной» сформируем движениями по РН «Продажи номенклатуры». В рамках данного примера расчет себестоимости списания реализовывать не будем, поэтому при заполнении ресурса Себестоимость позволим себе упрощение: будем считать, что себестоимость номенклатуры равна половине суммы продаж.

Движения сформируем при помощи конструктора движений:

Процедура ОбработкаПроведения(Отказ, Режим)

```
//{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
```

```
// регистр ПродажиНоменклатуры
```

```
Движения.ПродажиНоменклатуры.Записывать = Истина;
```

```
Для Каждого ТекСтрокаСписокНоменклатуры Из СписокНоменклатуры Цикл
```

```
    Движение = Движения.ПродажиНоменклатуры.Добавить();
```

```
    Движение.Период = Дата;
```

```
    Движение.Номенклатура = ТекСтрокаСписокНоменклатуры.Номенклатура;
```

```
    Движение.Накладная = Ссылка;
```

```
    Движение.Количество = ТекСтрокаСписокНоменклатуры.Количество;
```

```
    Движение.Себестоимость = ТекСтрокаСписокНоменклатуры.Сумма/2;
```

```
    Движение.СуммаПродажи = ТекСтрокаСписокНоменклатуры.Сумма;
```

```
КонецЦикла;
```

```
//}_{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

КонецПроцедуры

В режиме «1С:Предприятие» сформируем документ «Расходная накладная». Дату документа установим специально в предыдущем месяце.

N	Номенклатура	Количество	Сумма
1	Виг (капиллярная)	3	2 000,00
2	Паркер "Golg"	2	2 000,00
3	Шариковая обычная	5	2 000,00

Рисунок 5 – Документ «Расходная накладная № 1»

Проверяем движения:

<div> <div>← →</div> <div>☆ Расходная накладная 000000001 от 10.09.2018 10:43:02</div> </div>								
<div> <div>Основное</div> <div>Продажи номенклатуры</div> </div>								
Продажи номенклатуры								
Период	↓	Регистратор	Но...	Номенклатура	Накладная	Коли...	Себестоимость	Сумма продажи
• 10.09.2018 ...		Расходная нак...	1	Big (капиллярная)	Расходная наклад...	3	1 000,00	2 000,00
• 10.09.2018 ...		Расходная нак...	2	Паркер "Golg"	Расходная наклад...	2	1 000,00	2 000,00
• 10.09.2018 ...		Расходная нак...	3	Шариковая обычная	Расходная наклад...	5	1 000,00	2 000,00

Рисунок 6 – Движения документа «Расходная накладная № 1»

Все как и должно быть.

Проведение документа «Дополнительные затраты»

В обработке проведения необходимо:

1. Получить данные для распределения дополнительных затрат: номенклатуру и проданное количество
2. Реализовать алгоритм распределения. Рассчитанная сумма дополнительных затрат должна увеличивать себестоимость продаж, поэтому записывать ее будем в РН «Продажи номенклатуры» в ресурс Себестоимость. Количество и Сумма продажи при этом изменяться не должны – заполняем их нулевыми значениями
3. В алгоритме распределения ошибку округления отнесем на последнее движение.

Алгоритм распределения суммы пропорционально коэффициентам можно найти в типовых конфигурациях, построенных на базе «Библиотеки стандартных подсистем» – см. модуль «ОбщегоНазначенияКлиентСервер», функция `РаспределитьСуммуПропорциональноКоэффициентам`. Там ошибка округления относится на сумму с максимальным коэффициентом, то есть на строку с максимальной суммой, получившейся в результате распределения. Причина в том, что на самой большой сумме удельный вес корректировки суммы на ошибку округления будет наименьшим.

В условии есть указание, что, независимо от периода внесения данных о дополнительных затратах, себестоимость в отчете должна указываться с учётом доставки. Для выполнения этого условия движения по увеличению себестоимости будем делать не датой документа «Дополнительные затраты», а датой выбранной в этом документе накладной.

Листинг кода обработки проведения документа «Дополнительные затраты»:

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

//1.Подготовка РН к записи

Движения.ПродажиНоменклатуры.Записывать = истина;

//2. запрос для получения данных о количестве проданных товаров

//по указанной в документе накладной

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ПродажиНоменклатурыОбороты.Номенклатура,

| ПродажиНоменклатурыОбороты.КоличествоОборот КАК Количество

|ИЗ

| РегистрНакопления.ПродажиНоменклатуры.Обороты(&НДата, &КДата, , Накладная =

&Накладная) КАК ПродажиНоменклатурыОбороты

|ИТОГИ

| СУММА(Количество)

|ПО

| ОБЩИЕ";

//3. заполнение параметров запроса

Запрос.УстановитьПараметр("Накладная", Накладная);

ДатаНакладной = Накладная.Дата;

Запрос.УстановитьПараметр("НДата", НачалоМесяца(ДатаНакладной));

Запрос.УстановитьПараметр("КДата", КонецДня(ДобавитьМесяц(НачалоМесяца(ДатаНакладной),1)));

//4. выполнение запроса и получение выборки по группировке - по накладной

РезультатЗапроса = Запрос.Выполнить();

ВыборкаНакладная = РезультатЗапроса.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);

//5. т.к. в запросе установлен отбор по накладной, то в выборке может быть

//только одна запись, поэтому цикл не нужен

Если ВыборкаНакладная.Следующий() Тогда

//6. запоминаем суммарное количество по всей номенклатуре накладной

КоличествоПоНакладной = ВыборкаНакладная.Количество;

//7.Заводим переменную для хранения значения суммы, которую

//мы распределили

РаспределеннаяСумма = 0;

//8.получаем выборку по товарам накладной и обходим в цикле

//на каждую строку выборки будем добавлять движение по регистру

ВыборкаНоменклатура = ВыборкаНакладная.Выбрать();

Пока ВыборкаНоменклатура.Следующий() Цикл

Движение = Движения.ПродажиНоменклатуры.Добавить();

//9. в период пишем дату накладной, а не дату документа

Движение.Период = ДатаНакладной;

Движение.Накладная = Накладная;

Движение.Номенклатура = ВыборкаНоменклатура.Номенклатура;

```
Движение.Количество = 0;
Движение.СуммаПродажи = 0;
//10. рассчитываем долю суммы доставки пропорционально количеству
//помним, что при записи произойдет округление до точности ресурса
Движение.Себестоимость =
Сумма*ВыборкаНоменклатура.Количество/КоличествоПоНакладной;
//11. увеличиваем значение распределенной суммы на сумму, записанную в движение
РаспределеннаяСумма = РаспределеннаяСумма + Движение.Себестоимость;
КонецЦикла;

// 12. Погрешности округления отнесем на последнее движение
Движение.Себестоимость = Движение.Себестоимость + Сумма - РаспределеннаяСумма;
КонецЕсли

КонецПроцедуры
```

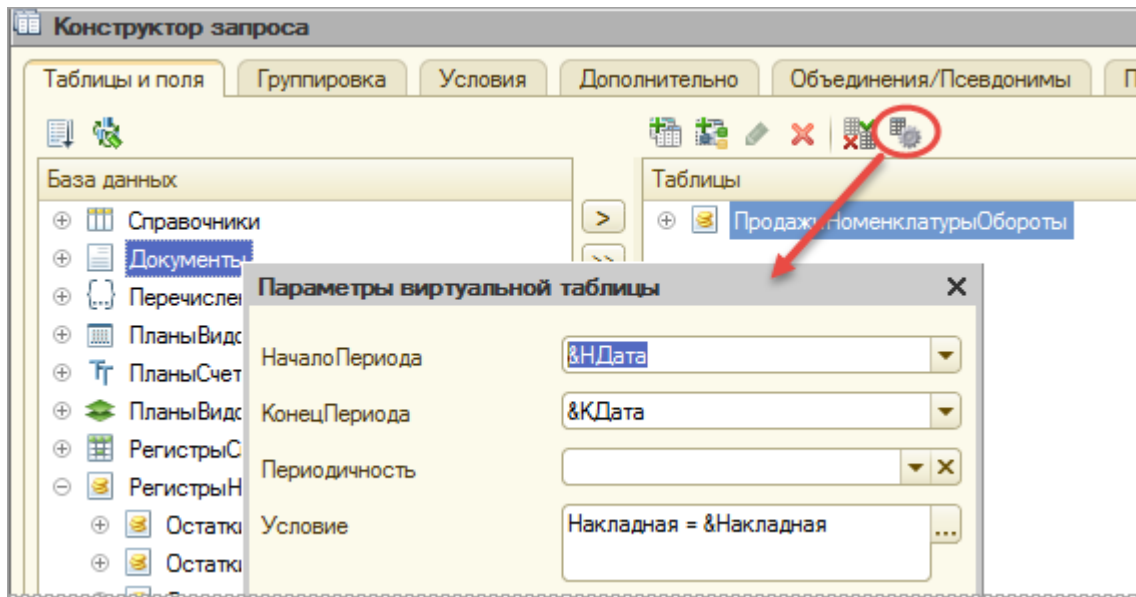
Разберем ключевые точки алгоритма

Подготовка регистра к записи (п. 1)

В данном случае достаточно установить маркер необходимости записи движений в истину. Выполнять предварительную очистку движений документа не нужно, т.к. они никакого воздействия на результат запроса не оказывают: получаем данные о количестве проданного товара, а документ пишет только в ресурс *Себестоимость*.

Запрос для получения данных для распределения суммы (п. 2)

Строим запрос к данным виртуальной таблицы оборотов РН «Продажи номенклатуры», устанавливаем параметры виртуальной таблицы:



**Рисунок 7 – Параметры виртуальной таблицы оборотов
РН «Продажи номенклатуры»**

Ограничиваем период и устанавливаем отбор по *Накладной*. Параметр *Периодичность* не заполняем, будет использоваться значение по умолчанию – *Период*.

Подробнее про параметры виртуальной таблицы оборотов оборотного регистра накопления, а также про то, как заполнение параметров влияет на обращение к реальным таблицам регистра, рассказывается в курсе «[Разработка и оптимизация запросов в 1С:Предприятие 8.3](#)».

В запросе сразу получаем итог по количеству. Он нам будет нужен для определения доли дополнительных затрат, которую нужно списать на номенклатуру.

Заполнение параметров (п. 3)

Чтобы установить ограничение по периоду, мы могли бы взять дату накладной.

Но! Т.к. в таблице оборотов накопленные показатели хранятся с периодичностью месяца, то, чтобы не происходило обращение к таблице движений, нужно передавать период, кратный месяцу. Это вполне подходит – будем передавать месяц накладной.

В переменную *ДатаНакладной* запишем значение даты документа, она нам еще пригодится при формировании движений.

Начало периода берем на начало месяца от *ДатаНакладной*. Конец периода – прибавляем один месяц и берем значение на конец дня.

Если для ограничения периода будет использоваться дата накладной, на аттестации это не будет считаться ошибкой, несмотря на то, что данные оборотов в этом случае будут получаться по таблице движений.

Выполнение запроса по общему итогу, фиксация в переменную общего количества по накладной параметров (п. 4, 5, 6, 7)

Цикл не нужен, т.к. в запросе установлен отбор по накладной. Фиксируем общее количество проданных товаров по накладной в переменную *КоличествоПоНакладной*.

Заводим переменную *РаспределеннаяСумма* для накопления суммы распределения. Она пригодится для вычисления ошибки округления. Изначально устанавливаем ее в ноль.

Обход выборки по номенклатуре и формирование движений (п. 8-12)

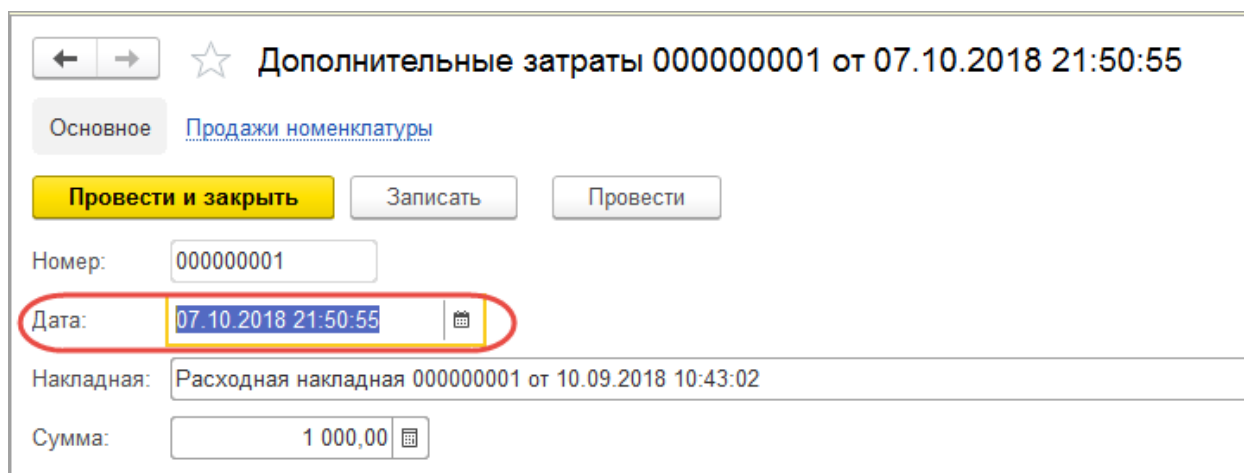
Движения формируем датой накладной, а не датой документа.

Рассчитываем сумму дополнительных затрат пропорционально проданному количеству товара. При этом помним, что у нас при записи в ресурс регистра происходит округление. Сумму с учетом накопленных ошибок округления накапливаем в переменной *РаспределеннаяСумма*.

В п.12 относим накопленную ошибку округления на последнее движение.

Проверка распределения суммы дополнительных затрат

В режиме «1С:Предприятие» создаем документ «Дополнительные затраты». Внесем его датой следующего месяца по отношению к указанной в нем накладной:



← → ☆ **Дополнительные затраты 000000001 от 07.10.2018 21:50:55**

Основное [Продажи номенклатуры](#)

Провести и закрыть Записать Провести

Номер: 000000001

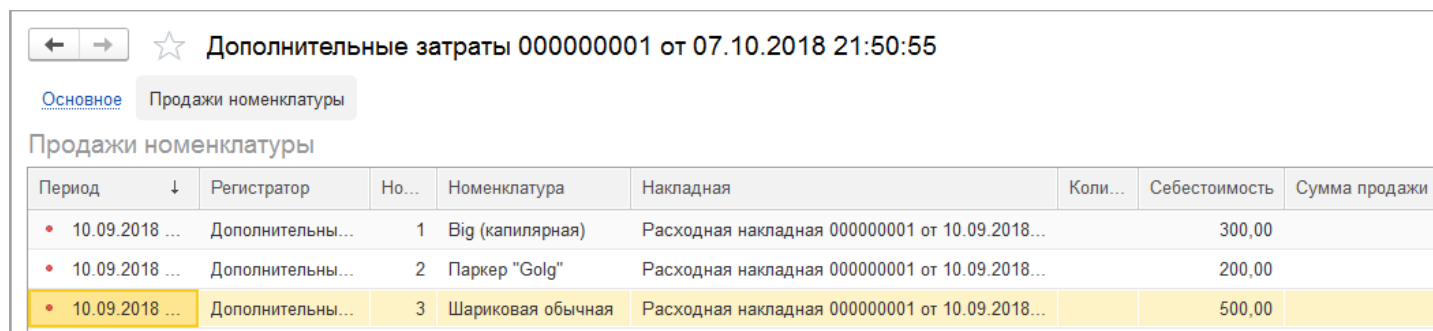
Дата: 07.10.2018 21:50:55

Накладная: Расходная накладная 000000001 от 10.09.2018 10:43:02

Сумма: 1 000,00

Рисунок 8 – Документ «Дополнительные затраты № 1»

Проверим движения:



← → ☆ **Дополнительные затраты 000000001 от 07.10.2018 21:50:55**

[Основное](#) [Продажи номенклатуры](#)

Продажи номенклатуры

Период	↓	Регистратор	Но...	Номенклатура	Накладная	Коли...	Себестоимость	Сумма продажи
• 10.09.2018 ...		Дополнительны...	1	Big (капиллярная)	Расходная накладная 000000001 от 10.09.2018...		300,00	
• 10.09.2018 ...		Дополнительны...	2	Паркер "Golg"	Расходная накладная 000000001 от 10.09.2018...		200,00	
• 10.09.2018 ...		Дополнительны...	3	Шариковая обычная	Расходная накладная 000000001 от 10.09.2018...		500,00	

Рисунок 9 – Движения документа «Дополнительные затраты № 1»

Видим, что распределение суммы дополнительных затрат выполнилось верно.

Создадим еще один документ «Расходная накладная»:

← →
☆
Расходная накладная 000000002 от 11.09.2018 12:50:56

Основное
Продажи номенклатуры

Провести и закрыть
Записать
Провести

Номер:

Дата:

Добавить

N	Номенклатура	Количество	Сумма
1	Виг (капиллярная)	12	3 000,00
2	Паркер "Golg"	5	2 000,00
3	Шариковая обычная	10	1 000,00

Рисунок 10 – Документ «Расходная накладная № 2»

Проверим движения:

← →
☆
Расходная накладная 000000002 от 11.09.2018 12:50:56

Основное
Продажи номенклатуры

Продажи номенклатуры

Период	↓	Регистратор	Но...	Номенклатура	Накладная	Коли...	Себестоимость	Сумма продажи
• 11.09.2018 ...		Расходная нак...	1	Виг (капиллярная)	Расходная наклад...	12	1 500,00	3 000,00
• 11.09.2018 ...		Расходная нак...	2	Паркер "Golg"	Расходная наклад...	5	1 000,00	2 000,00
• 11.09.2018 ...		Расходная нак...	3	Шариковая обычная	Расходная наклад...	10	500,00	1 000,00

Рисунок 11 – Движения документа «Расходная накладная № 2»

Чтобы удостовериться, что не остается ошибка округления, внесем документ «Дополнительные затраты» с «некрасивой» суммой и проверим его движения:

← → ☆ **Дополнительные затраты 000000002 от 07.10.2018 15:15:15**

Основное Продажи номенклатуры

Провести и закрыть Записать Провести

Номер: 000000002

Дата: 07.10.2018 15:15:15

Накладная: Расходная накладная 000000002 от 11.09.2018 12:50:56

Сумма: 357,00

Рисунок 12 – Документ «Дополнительные затраты № 2»

Проводим документ и проверяем движения документа:

← → ☆ **Дополнительные затраты 000000002 от 07.10.2018 15:15:15**

Основное Продажи номенклатуры

Продажи номенклатуры

Период	↓	Регистратор	Но...	Номенклатура	Накладная	Коли...	Себестоимость	Сумма продажи
• 11.09.2018 ...		Дополнительны...	1	Виг (капиллярная)	Расходная накладная 000000002 от 11.09.2018...		158,67	
• 11.09.2018 ...		Дополнительны...	2	Паркер "Golg"	Расходная накладная 000000002 от 11.09.2018...		66,11	
• 11.09.2018 ...		Дополнительны...	3	Шариковая обычная	Расходная накладная 000000002 от 11.09.2018...		132,22	

Сумма 357,00 руб

Рисунок 13 – Движения документа «Дополнительные затраты № 2»

Все отлично. Сумма распределилась полностью.

Отчет разрабатывать не будем, но построим запрос, аналогичный тому, который требуется для отчета. Воспользуемся «Консолью запросов»:

Консоль запросов : E:\Запросы\ПродажиСебестоимостьДопРасх

Параметры запроса

Имя параметра В Значение параметра

ЧДата 01.09.2018 0:00:00

КДата 30.09.2018 0:00:00

ВЫБРАТЬ

ПродажиНоменклатурыОбороты.Номенклатура КАК Номенклатура,
ПродажиНоменклатурыОбороты.Накладная КАК Накладная,
ПродажиНоменклатурыОбороты.КоличествоОборот КАК Количество,
ПродажиНоменклатурыОбороты.СебестоимостьОборот КАК СебестоимостьОборот,
ПродажиНоменклатурыОбороты.СуммаПродажиОборот КАК СуммаПродажиОборот,
ПродажиНоменклатурыОбороты.СуммаПродажиОборот - ПродажиНоменклатурыОбороты.СебестоимостьОборот КАК Прибыль

ИЗ

РегистрНакопления.ПродажиНоменклатуры.Обороты (&НДата, &КДата, ,) КАК ПродажиНоменклатурыОбороты

ИТОГИ

СУММА (Количество),
СУММА (СебестоимостьОборот),
СУММА (СуммаПродажиОборот),
СУММА (Прибыль)

Способ выгрузки: Список

Номенклатура	Накладная	Количество	СебестоимостьОборот	СуммаПродажиОборот	Прибыль
Big (капиллярная)		15	2 958,67	5 000	2 041,33
Big (капиллярная)	Расходная накладная 000000001 ...	3	1 300	2 000	700
Big (капиллярная)	Расходная накладная 000000002 ...	12	1 658,67	3 000	1 341,33
Паркер "Golg"		7	2 266,11	4 000	1 733,89
Паркер "Golg"	Расходная накладная 000000001 ...	2	1 200	2 000	800
Паркер "Golg"	Расходная накладная 000000002 ...	5	1 066,11	2 000	933,89
Шариковая обычная		15	2 132,22	3 000	867,78
Шариковая обычная	Расходная накладная 000000001 ...	5	1 500	2 000	500
Шариковая обычная	Расходная накладная 000000002 ...	10	632,22	1 000	367,78

Рисунок 14 – Результат запроса для анализа себестоимости продаж с учетом дополнительных затрат

Видим, что дополнительные затраты, внесенные в октябре, увеличили себестоимость продаж в сентябре. Что и требовалось по условию.

Подведем итог

Мы рассмотрели подзадачу распределения дополнительных затрат. В этих задачах важно из условия внимательно выявить, на что нужно распределять дополнительные затраты и пропорционально какому показателю.

После получения данных для распределения алгоритм распределения будет одинаков для всех задач данного типа. Важно не забыть учесть ошибку округления.

При записи данных в регистр нужно внимательно подходить к выбору периода движений исходя из условия задачи.

29. Какие объекты конфигурации следует использовать для хранения произвольных свойств

Для ведения полноценного учета в информационной базе «1С:Предприятие» часто требуется реализовать возможность хранения дополнительных свойств используемых объектов. Причем эти свойства могут отличаться для разных типов объектов.

Например, для клиентов организации могут потребоваться для анализа следующие свойства:

- Вид клиента (разовый, постоянный, VIP-клиент)
- Источник информации о нашей фирме (контекстная реклама в интернете, реклама на радио, статьи в журнале)
- Бизнес-регион, к которому относится покупатель.

Используемые свойства могут отличаться и для разных групп товаров. Так, для смартфонов можно использовать следующие характеристики:

- Диагональ экрана
- Разрешение экрана
- Операционная система (Android, iOS)
- Объем оперативной памяти
- Емкость аккумулятора
- Камера
- Возможность использования карты памяти
- Тип SIM-карты.

Для автомобильных шин используемые свойства будут другими:

- Сезонность (летние/зимние)
- Диаметр
- Ширина профиля
- Высота профиля
- Наличие шипов.

Чтобы организовать учет подобных свойств в базе, в конфигураторе для каждого свойства можно создать отдельный реквизит справочника.

Однако на этапе конфигурирования заранее неизвестно, какими именно товарами будет торговать организация, использующая разработанное прикладное решение, и какие дополнительные сведения по товарам потребуются пользователю при работе. Также заранее неизвестно, какие типы данных

необходимо использовать для хранения значений этих свойств. При этом если для определенной категории товаров потребуется добавить новое свойство, то придется обращаться к программисту, чтобы он в конфигураторе реализовал новый реквизит.

К тому же такой способ может оказаться неэкономным с точки зрения объемов хранимых данных. Предположим, что большую часть ассортимента будут составлять мобильные телефоны, тогда свойство «Наличие шипов» для таких товаров будет лишним. В данном реквизите для большинства товаров будет храниться пустое значение.

Поэтому создавать реквизиты для хранения подобных свойств в конфигураторе – не самая лучшая идея. Требуется реализовать такой вариант учета, при котором пользователь в режиме «1С:Предприятие» сможет самостоятельно добавлять дополнительные свойства для объектов, указывать, какие типы данных должны использоваться для хранения значений этих свойств.

Для решения подобных задач в платформе «1С:Предприятие» существуют специальные объекты метаданных – планы видов характеристик, которые используются в решении задач оперативного учета (1.10, 1.36, 1.46), задач по управляемым формам (5.16).

В теоретической части по вопросам хранения и использования произвольных свойств (характеристик) подробно рассмотрены достаточно сложные механизмы реализации. На практике такие глубокие знания могут не потребоваться. После изучения вопросов теории будет рассмотрена практическая задача.

Использование плана видов характеристик

Постановка задачи:

Организация торгует различными группами товаров. Каждая группа товаров имеет свои собственные свойства, которые служат для описания отдельных товаров. Для хранения значений свойств могут использоваться различные типы данных. Необходимо обеспечить возможность хранения таких данных в базе.

Следует понимать, что в реальной экзаменационной задаче может идти речь не только о дополнительных свойствах товаров, а также о дополнительных свойствах контрагентов, организаций и т. д.

Рассмотрим возможности использования для достижения этих целей регистра сведений. Создадим регистр сведений «ЗначенияСвойствНоменклатуры».

По сути необходимо организовать учет значений свойств для однозначно идентифицируемых пар значений *Объект – Свойство объекта*. В конкретном случае в роли объектов выступают элементы справочника «Номенклатура». Таким образом, измерениями регистра будут:

- Объект (*СправочникСсылка.Номенклатура*)
- Свойство (*СправочникСсылка.СвойстваНоменклатуры*).

Предварительно необходимо создать справочник «СвойстваНоменклатуры».

Создадим ресурс регистра сведений – *ЗначениеСвойства*.

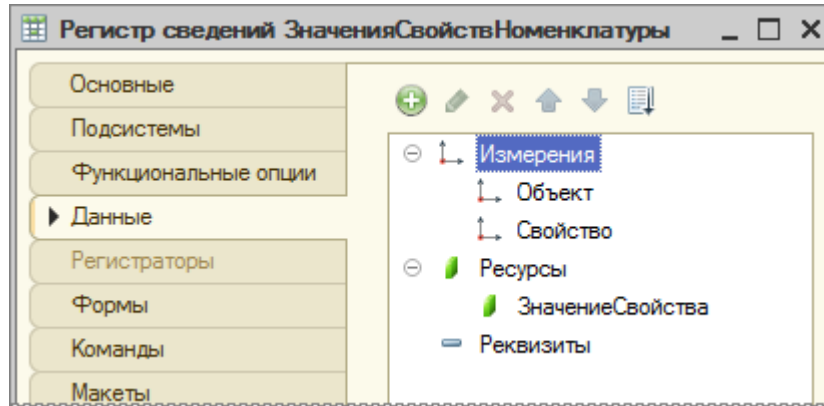


Рисунок 1 – Структура регистра сведений для хранения значений свойств

Возникает вопрос: какой тип данных следует использовать для ресурса *ЗначениеСвойства*? В данном случае это может быть:

- Число (величина диагонали экрана в дюймах – 4, 4,5, 5, 5,2, 5,5 и т.д.)
- Булево (возможность использования карты памяти – *Истина/Ложь*)
- Собственный классификатор (*разрешение экрана* – 1280x720, 1920x1080, 2960x1440 и т.д.).

Следовательно, нужно будет использовать составной тип данных.

Однако для конкретного свойства требуется указание значений только определенного типа. Так, нет смысла выбирать разрешение экрана из списка *Истина/Ложь*. То есть требуется реализовать возможность хранения дополнительных свойств объектов с указанием типа данных для каждого свойства.

Подобный механизм можно попробовать реализовать самостоятельно, но проще воспользоваться готовым, специально предназначенным для решения подобных задач объектом, который называется *План видов характеристик (ПВХ)*. В плане видов характеристик хранится информация о доступных свойствах (видах характеристик) и о типах значений этих свойств. В каркасной конфигурации уже присутствует план видов характеристик *СвойстваОбъектов*.

По доступным в конфигураторе настройкам объект метаданных *ПВХ* похож на объект метаданных *Справочник*. Сходство заключается в возможности:

- Использования иерархии (создание элементов и групп).

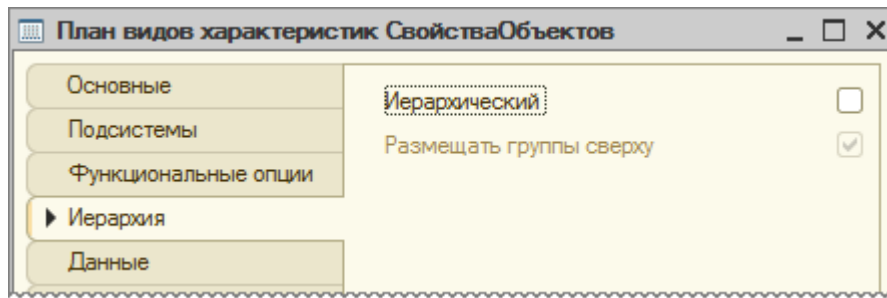


Рисунок 2 – Настройки иерархии для ПВХ

- Создания предопределенных элементов.

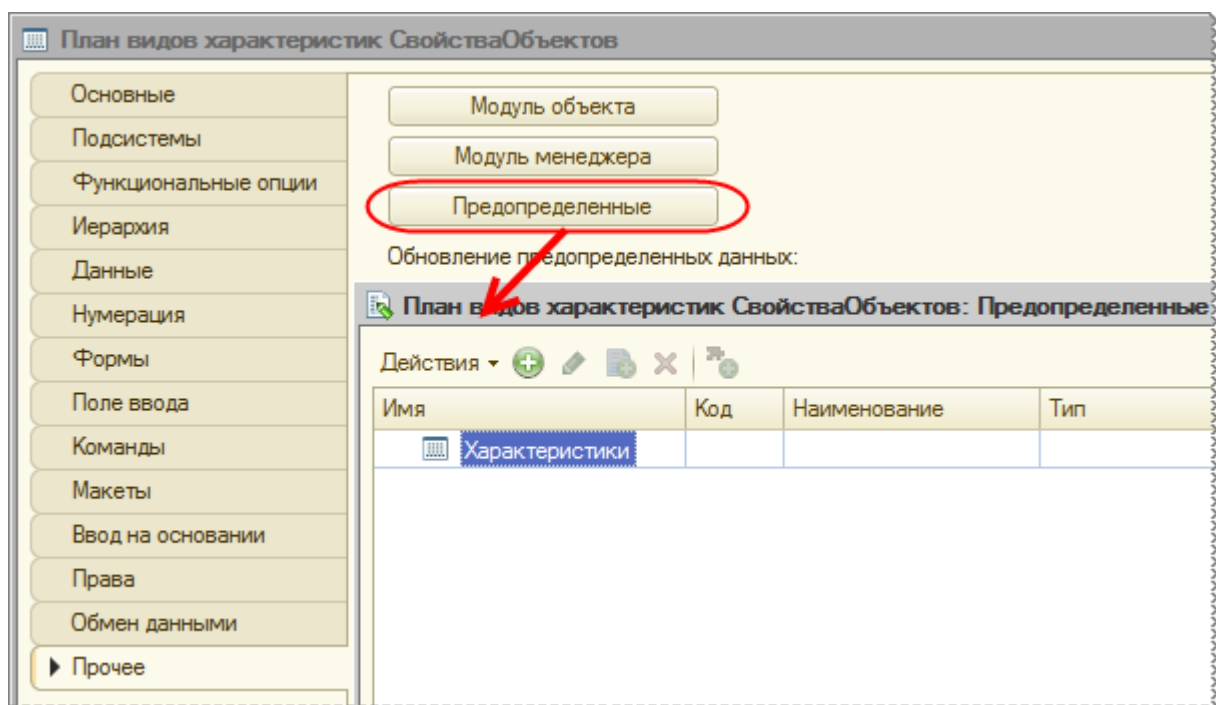


Рисунок 3 – Настройка предопределенных элементов для ПВХ

- Создания реквизитов и табличных частей.

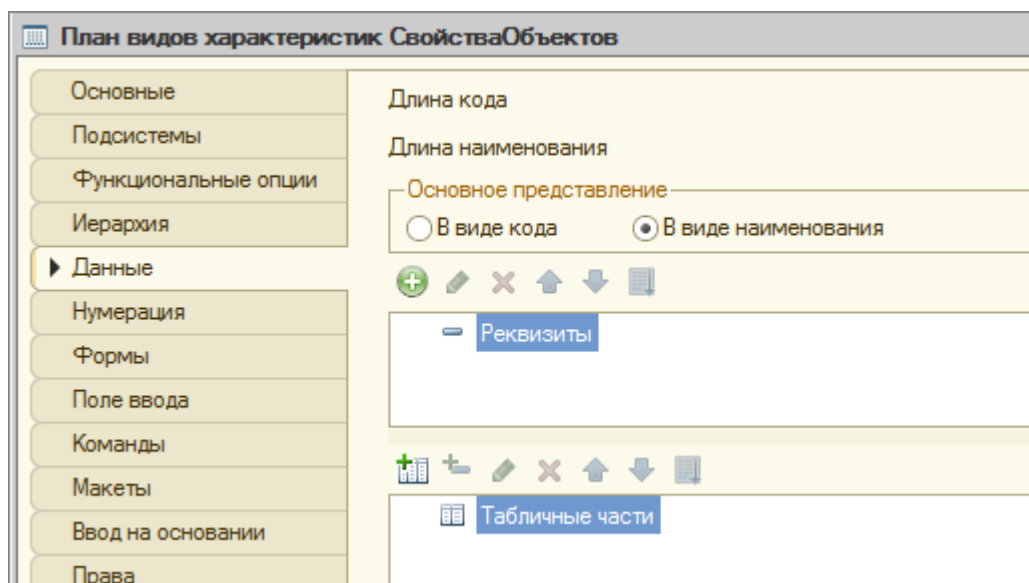


Рисунок 4 – Добавление реквизитов и табличных частей в ПВХ

Принципиально важным отличием объекта метаданных *ПВХ* от *Справочника* и от других объектов метаданных является наличие свойства *Тип значения характеристик*.

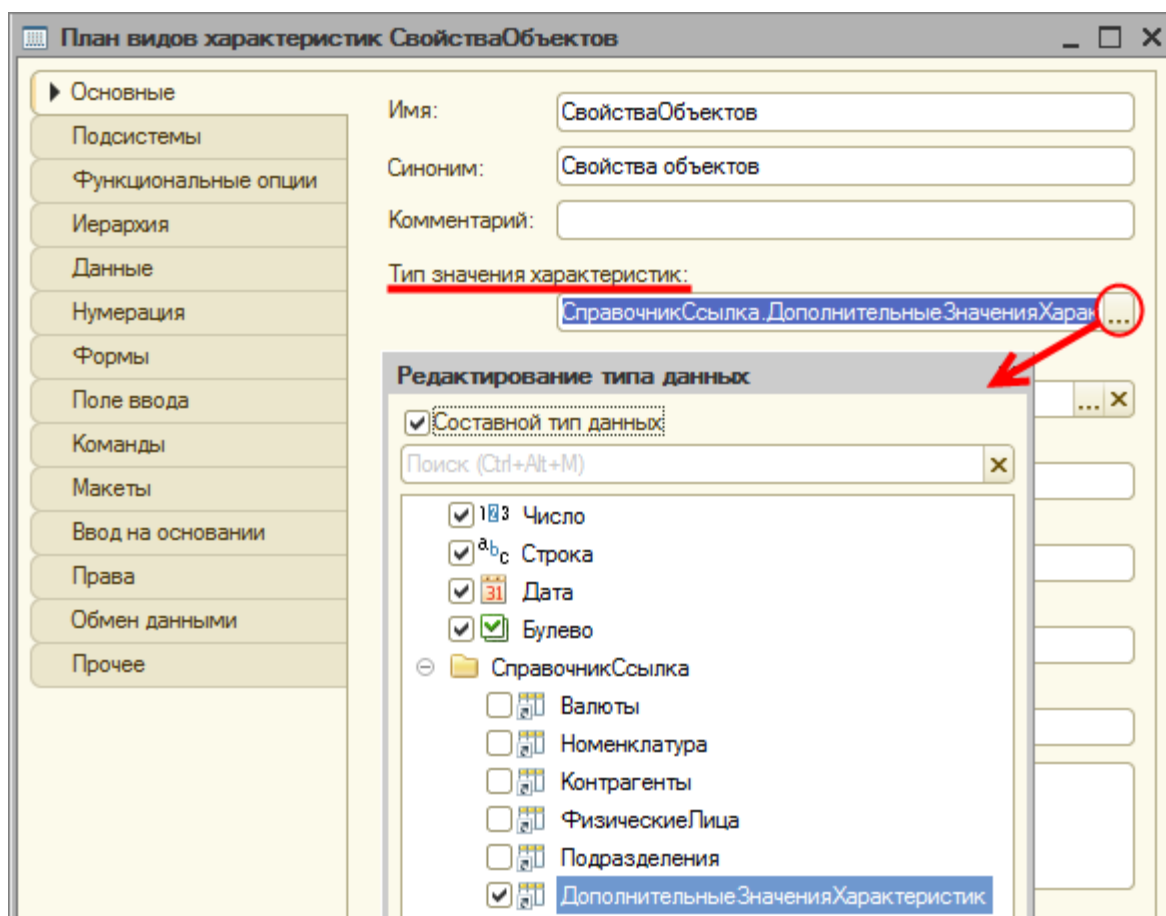


Рисунок 5 – Свойство ПВХ «Тип значения характеристик»

Тип значения этого поля – *ОписаниеТипов*. У других объектов метаданных нельзя создать реквизит такого типа – это отличительная особенность плана видов характеристик.

Вернемся к примеру с мобильными телефонами. Для описания произвольных свойств можно использовать объект метаданных *План видов характеристик*. При этом каждое свойство (диагональ экрана, разрешение экрана, возможность использования карты памяти и т.д.) – это отдельный элемент ПВХ в базе. Для каждого такого элемента можно указать его тип. Для диагонали экрана следует указать тип *Число*, для возможности использования карты памяти – *Булево*.

The image contains two screenshots of the 1C software interface, specifically the 'Значения характеристик' (Characteristic Values) tab. Both windows have a title bar with the 1C logo and the text '(1С:Предприятие)'. The top window is titled 'Диагональ экрана (Свойства объектов)' and shows a 'Код' (Code) of '000000003', a 'Наименование' (Name) of 'Диагональ экрана', and a 'Тип значения' (Value Type) of 'Число' (Number), which is circled in red. The bottom window is titled 'Возможность использования карты памяти (...)' and shows a 'Код' (Code) of '000000004', a 'Наименование' (Name) of 'Возможность использования карты памяти', and a 'Тип значения' (Value Type) of 'Булево' (Boolean), which is also circled in red. Both windows have buttons for 'Записать и закрыть' (Save and Close), 'Записать' (Save), and 'Еще' (More).

Рисунок 6 – Тип значения элементов ПВХ

Таким образом, в ПВХ хранятся отдельные элементы (конкретные виды характеристик, свойства объектов), для каждого из которых можно указать тип значения этого свойства. Элементы ПВХ – это по сути описание типа данных для других элементов системы, которые будут строиться на базе элементов ПВХ. В этом заключается принципиальное отличие от элементов справочников, которые могут использоваться самостоятельно.

Для свойства «Разрешение экрана» используется собственный список значений (1280x720, 1920x1080, 2960x1440 и т.д.). Для хранения такого списка в конфигураторе можно создать специальное перечисление. Но в таком случае при появлении нового значения для разрешения экрана необходимо будет вносить изменения в конфигурацию, добавлять еще одно значение перечисления. Пользователь будет вынужден обращаться к программисту. Это нерационально, логичнее использовать для хранения таких значений отдельный справочник, который пользователь может заполнять сам в режиме «1С:Предприятие».

В настройках плана видов характеристик можно указать такой справочник в свойстве *Дополнительные значения характеристик*.

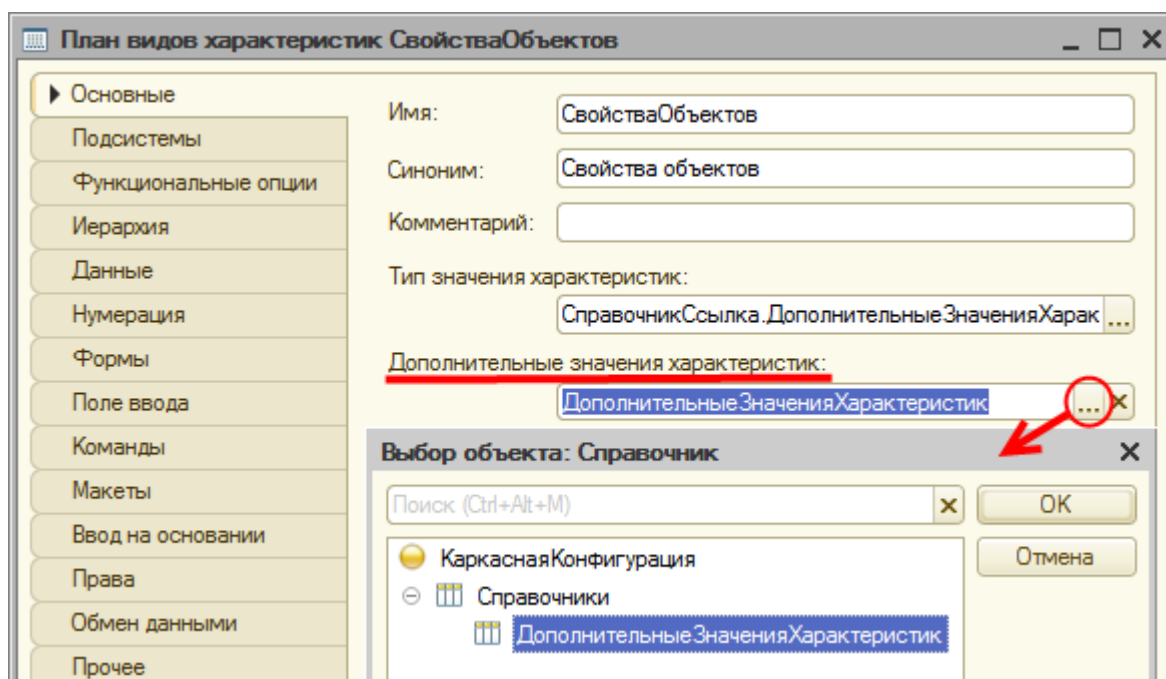


Рисунок 7 – Свойство ПВХ «Дополнительные значения характеристик»

Чтобы в настройках *ПВХ* определить значение в поле *Дополнительные значения характеристик*, предварительно создадим справочник «ДополнительныеЗначенияХарактеристик» и произведем его настройку. На закладке *Владельцы* окна редактирования справочника в качестве владельца выберем план видов характеристик «Свойства объектов».

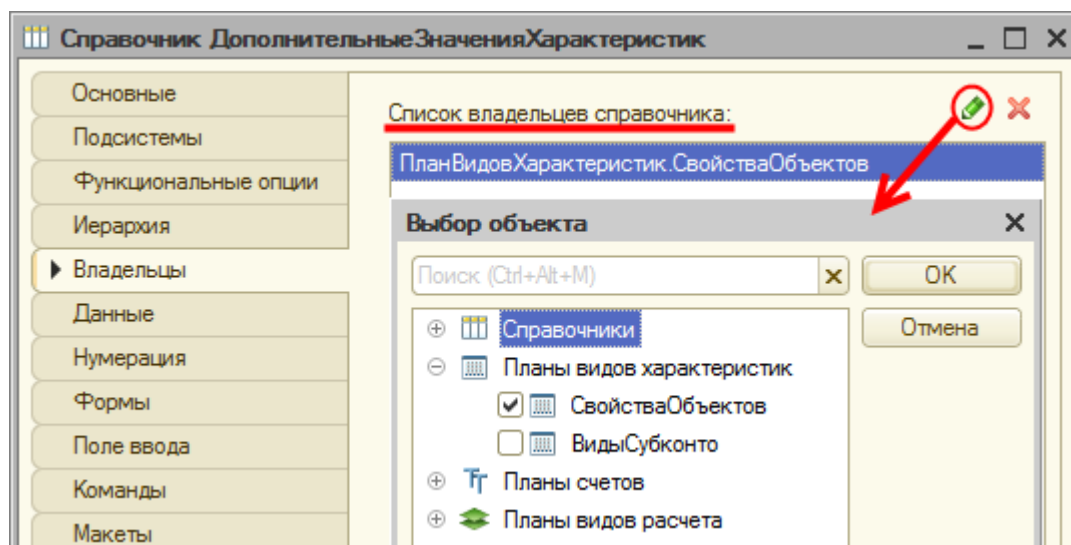


Рисунок 8 – Назначение владельца для справочника «Дополнительные значения характеристик»

Именно в справочнике «ДополнительныеЗначенияХарактеристик» пользователь сможет создавать новые значения для произвольных свойств. Например, для свойств «Тип SIM-карты» и «Разрешение экрана».

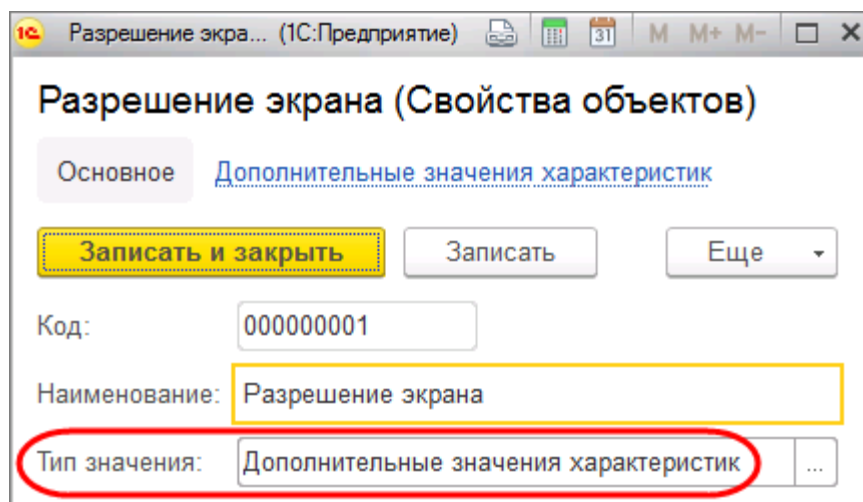


Рисунок 9 – Использование дополнительных значений характеристик

Для хранения различных свойств используется общий справочник. При работе с таким справочником пользователь должен понимать, какие значения относятся к свойству «Тип SIM-карты», а какие к свойству «Разрешение экрана», чтобы не возникало путаницы, поскольку все значения хранятся в общем справочнике.

При работе с конкретным свойством (видом характеристик) пользователю должны быть доступны для выбора только значения, которые относятся к этому свойству. Данное требование как раз

обеспечивается настройкой подчиненности справочника «ДополнительныеЗначенияХарактеристик» владельцу – плану видов характеристик «СвойстваОбъектов».

При такой настройке в пользовательском режиме будет видно, какие значения предусмотрены для выбранного свойства (отдельного элемента ПВХ).

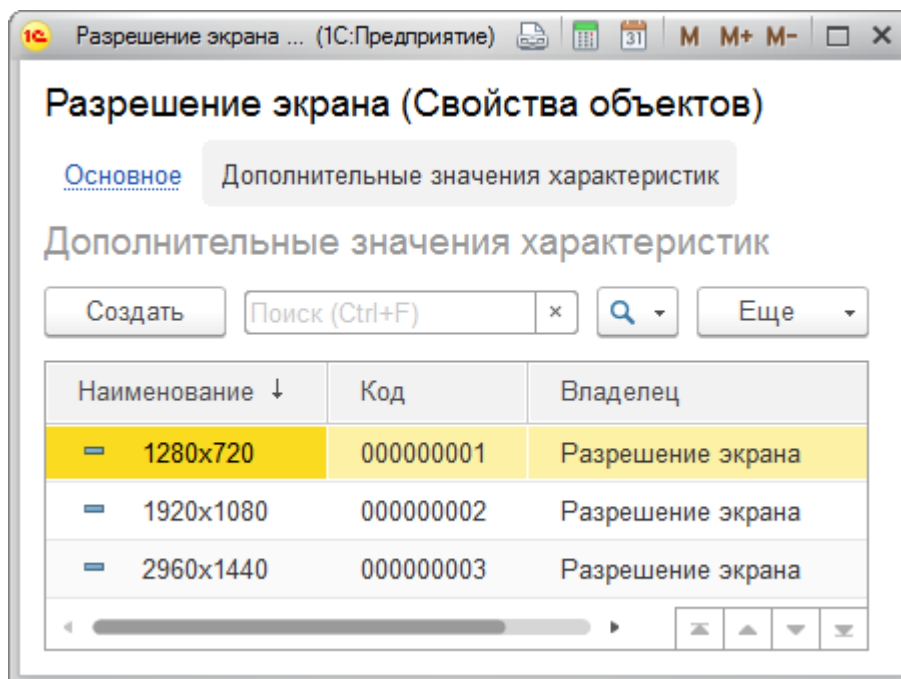


Рисунок 10 – Переход к значениям характеристик из формы элемента ПВХ

Пошаговая инструкция по настройке плана видов характеристик в конфигураторе

Подытожим разобранный материал и составим пошаговую инструкцию по настройке плана видов характеристик:

1. Создать ПВХ
2. Создать справочник для хранения дополнительных значений свойств
3. Для справочника дополнительных значений свойств определить в качестве владельца ранее созданный ПВХ
4. В настройках ПВХ для поля *Дополнительные значения характеристик* выбрать ранее созданный справочник дополнительных значений свойств

5. В настройках ПВХ определить значение для поля *Тип значения характеристик*. Как правило, это составной тип данных, который в обязательном порядке включает в себя ссылку на элемент ранее созданного справочника дополнительных значений свойств.

Заключение

Итак, в данном разделе рассмотрены вопросы, связанные с использованием дополнительных свойств объектов и их значений. Для хранения информация о доступных свойствах (видах характеристик) и о типах значений этих свойств используется специальный объект *ПланВидовХарактеристик*. В следующем разделе рассмотрим, как организовать хранение значений свойств объектов в базе.

Подробнее работа с дополнительными свойствами объектов рассматривается в курсе [«Профессиональная разработка интерфейсов и форм в 1С:Предприятие 8.3»](#) (Занятие 14. Свойства объектов).

30. Как следует организовать хранение значений произвольных свойств объектов в базе

Можно организовать несколько вариантов хранения значений дополнительных характеристик объектов метаданных. Это зависит от специфики решаемой задачи.

Может потребоваться организовать хранение характеристик только для одного объекта (например, справочника номенклатуры). Также можно организовать использование свойств для разных объектов метаданных (справочник номенклатуры, контрагентов, складов, валют, видов цен, организаций и т.д.), но в каждом свойстве можно указывать привязку к объекту метаданных.

Рассмотрим подробнее несколько вариантов, которые могут использоваться на экзамене. Следует отметить, что в типовых конфигурациях могут применяться и другие, более сложные способы хранения дополнительных свойств объектов. Но они выходят за рамки экзамена, поэтому в данном курсе не рассматриваются.

Хранение значений характеристик в отдельном регистре сведений

В простейшем случае ПВХ содержит перечень доступных свойств, а значения характеристик хранятся в регистре сведений.

Например, необходимо реализовать возможность назначения характеристик для справочника Номенклатура. Структура такого регистра может выглядеть следующим образом:

- Измерения:
 - Объект (*СправочникСсылка.Номенклатура*)
 - Свойство (*ПланВидовХарактеристикСсылка.СвойстваОбъектов*)
- Ресурс:
 - *ЗначениеСвойства*

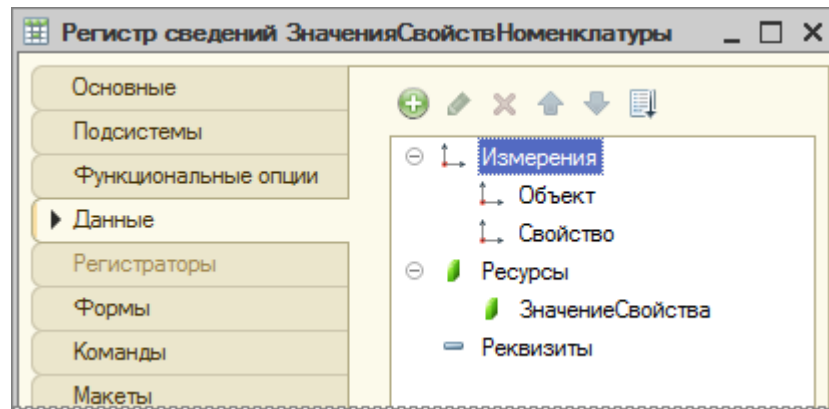


Рисунок 1 – Структура регистра сведений для хранения характеристик справочника Номенклатура

Для ресурса регистра сведений можно использовать составной тип данных и при этом заново вручную перечислить все типы из свойства *Тип значения характеристик* используемого плана видов характеристик.

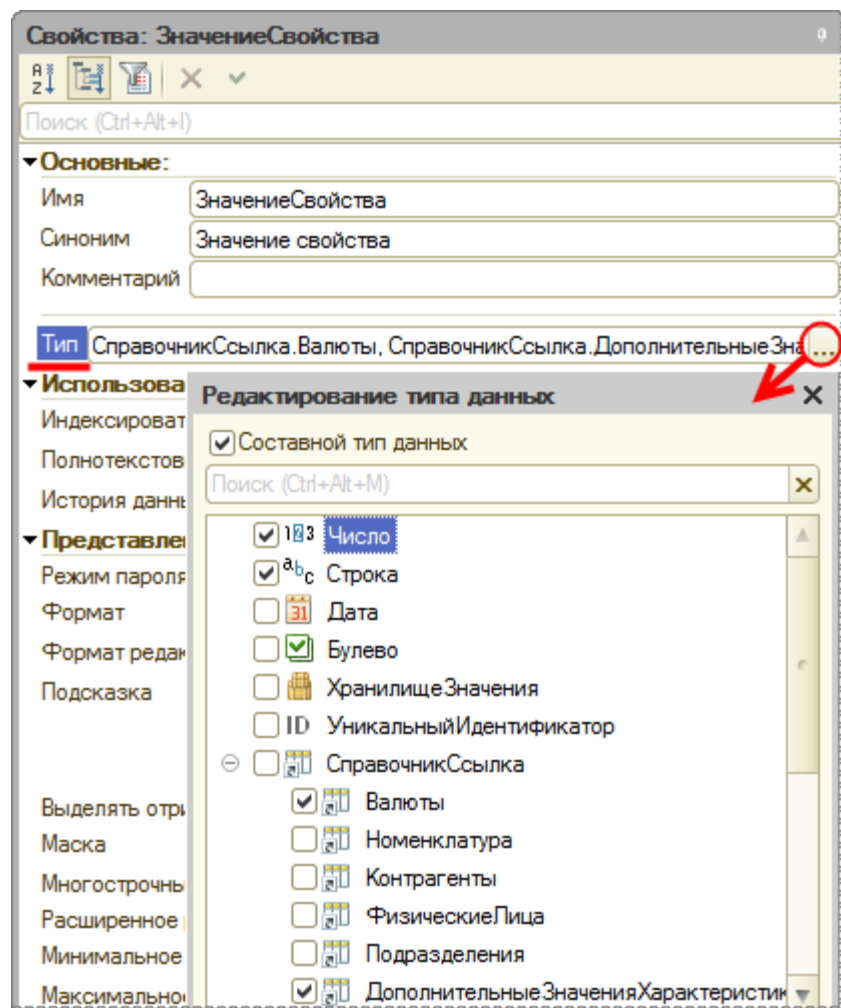


Рисунок 2 – Ручная настройка составного типа данных

Недостатки данного способа разработки:

- В случае появления нового типа данных для дополнительных свойств необходимо вносить соответствующие изменения как для ПВХ, так и для ресурса регистра сведений
- При добавлении нового типа данных в ПВХ платформа не будет подсказывать, что необходимо изменить составной тип данных ресурса регистра сведений.

Однако имеется возможность упростить разработку. Правильно для ресурса использовать специальный тип данных *Характеристика.СвойстваОбъектов*. Этот тип данных появился в системе и стал доступен для выбора после того, как был создан план видов характеристик «Свойства объектов».

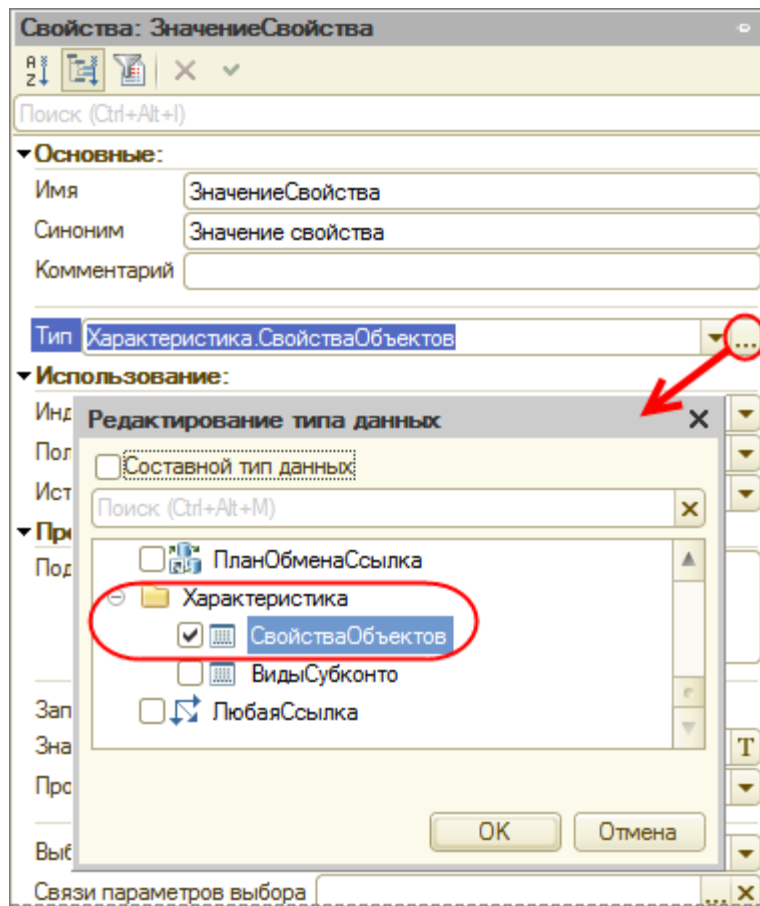


Рисунок 3 – Использование типа данных *Характеристика.СвойстваОбъектов*

Это позволит использовать в качестве типа данных для ресурса регистра сведений составной тип данных, который указан в свойстве *Тип значения характеристик* соответствующего ПВХ. При этом если в составном типе данных ПВХ добавится новый тип, то система автоматически добавит его и для ресурса регистра сведений.

Все необходимые настройки для хранения данных в базе выполнены. Теперь для удобства работы пользователей нужно выполнить настройки, которые обеспечат выбор нужных значений в пользовательском режиме.

При заполнении свойств номенклатуры в режиме «1С:Предприятие» поле для значения характеристики будет составного типа, а это значит, что пользователю сначала придется выбрать тип данных.

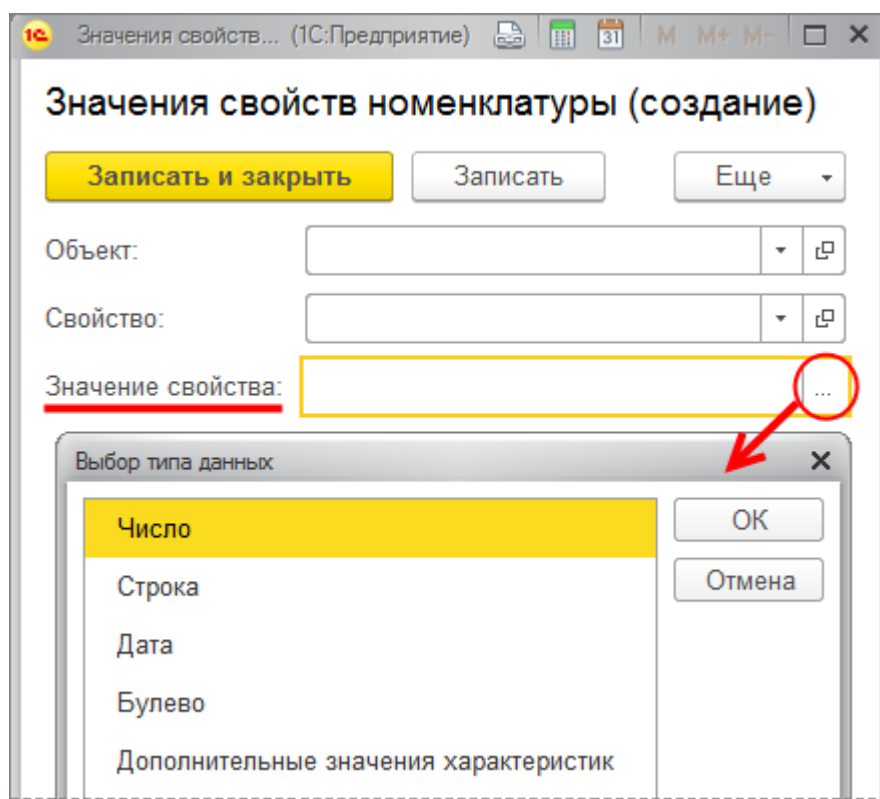


Рисунок 4 – Выбор типа данных

Однако выбор типа данных избыточен, ведь в конкретном виде характеристики (элементе плана видов характеристик) уже указан тип данных.

Поэтому для ресурса регистра сведений предусмотрено свойство *Связь по типу*, которое необходимо заполнить. *Связь по типу* позволяет ограничить тип поля ввода в зависимости от значения другого поля. В свойстве *Связь по типу* для ресурса нужно указать измерение *Свойство*, которое содержит ссылку на элемент плана видов характеристик.

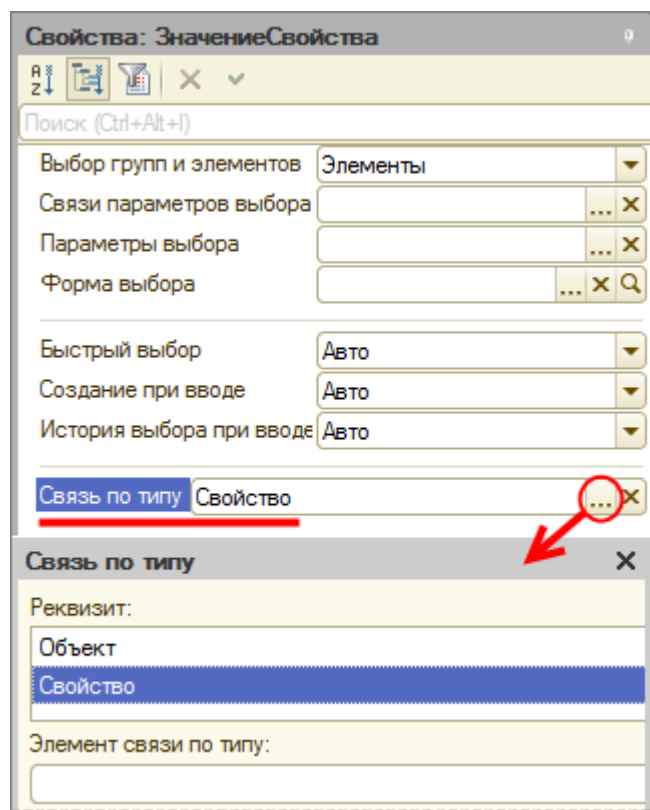


Рисунок 5 – Настройка свойства «Связь по типу»

Когда пользователь выбирает конкретный вид характеристики в поле «Свойство», система автоматически приведет поле «Значение свойства» к типу этой характеристики. То есть вручную выбирать тип поля не придется.

В конфигураторе для ресурса *ЗначениеСвойства* необходимо также заполнить свойство *Связи параметров выбора*. Для этого в окне выбора значения следует также выбрать доступный реквизит *Свойство* с параметром *Отбор.Владелец*.

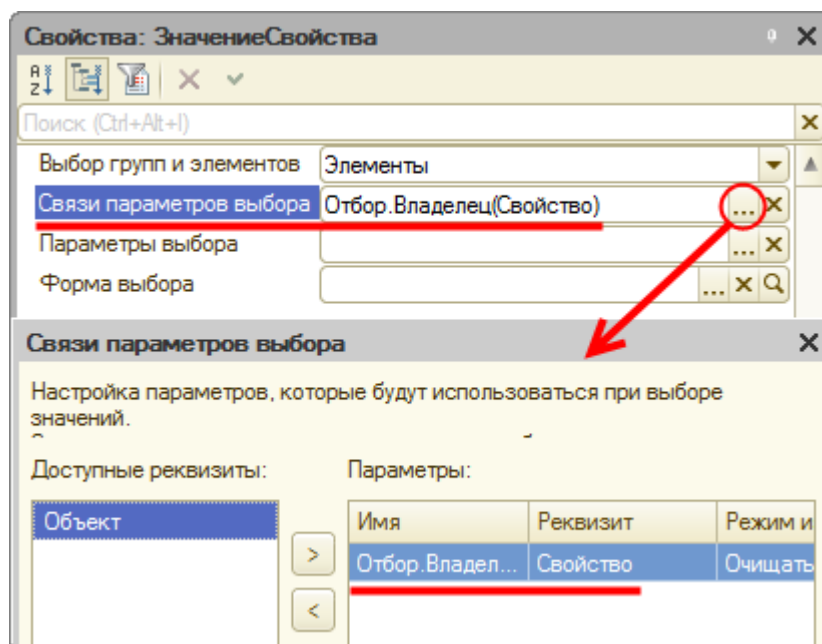
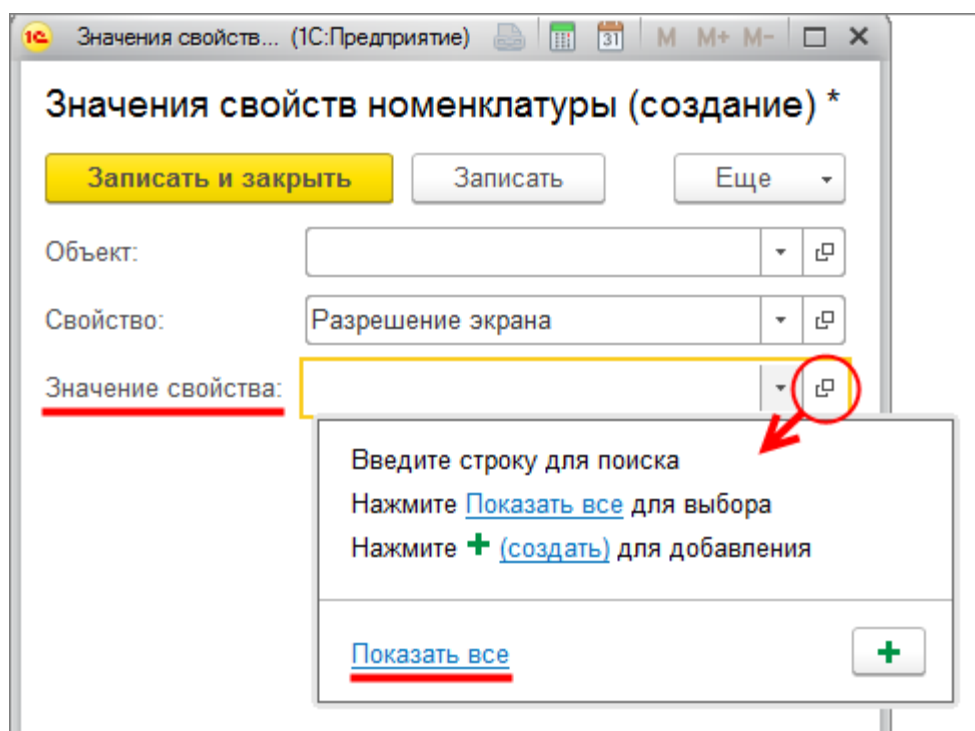


Рисунок 6 – Настройка свойства «Связи параметров выбора»

После этого в пользовательском режиме можно будет выбрать только те значения, владельцем которых является вид характеристики, определяемый по ссылке, содержащейся в реквизите *Свойство*. То есть для каждого свойства (характеристики) будет доступен соответствующий набор его значений из подчиненного справочника.



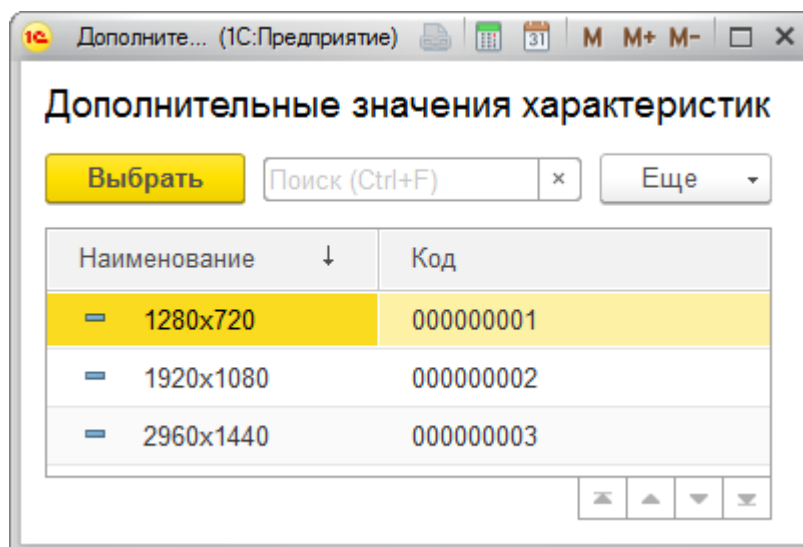


Рисунок 7 – Выбор значений характеристик с отбором по владельцу

Свойства *Связь по типу* и *Связи параметров выбора* для ресурса регистра сведений устанавливаются в конфигураторе на уровне объекта метаданных, а не в конкретной форме. Поэтому типизация поля со значением свойства (характеристики) в пользовательском режиме будет происходить в любой форме. При этом для выбора будут предлагаться только элементы, подчиненные данному виду характеристики.

В разобранный пример в регистре сведений хранятся значения свойств только для объектов из одного справочника (элементов справочника «Номенклатура»). Если потребуется обеспечить возможность назначения свойств для элементов другого справочника, то для измерения *Объект* нужно будет использовать составной тип данных. При этом дополнительно нужно будет разделить характеристики объектов из разных справочников, чтобы свойства номенклатуры не предлагались для выбора при работе с элементами других справочников, которым определены свойства в том же ПВХ. При этом для видов характеристик необходимо будет указывать, к элементам какого справочника они относятся.

Указание принадлежности характеристик

Рассмотрим ситуацию, когда ПВХ содержит перечень доступных свойств для различных объектов метаданных (справочников номенклатуры, контрагентов, складов, валют, видов цен, организаций и т.д.). Значения характеристик хранятся в регистре сведений. Структура такого регистра была определена ранее.

При рассмотрении примера для наглядности поменяем название ранее созданного регистра сведений «ЗначенияСвойствНоменклатуры» на «ЗначенияСвойствОбъектов»:

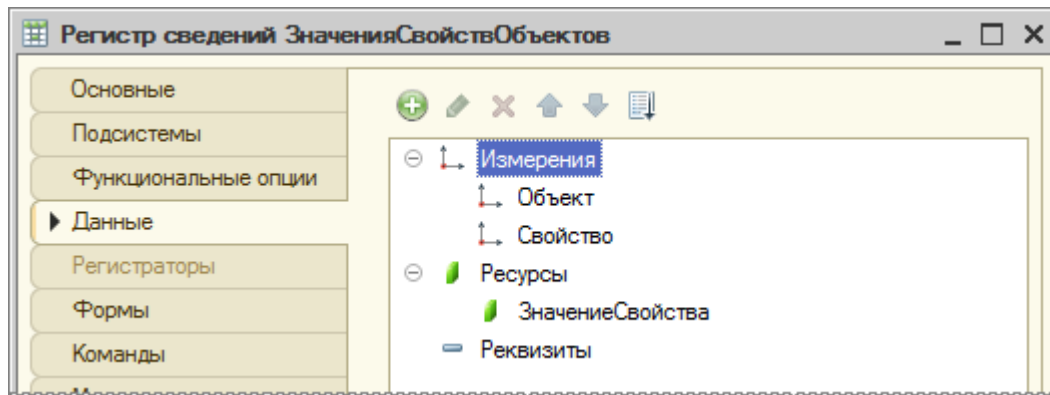


Рисунок 8 – Структура регистра сведений для хранения характеристик объектов

Для измерения Объект используем составной тип данных с добавлением типа *СправочникСсылка.Контрагенты*.

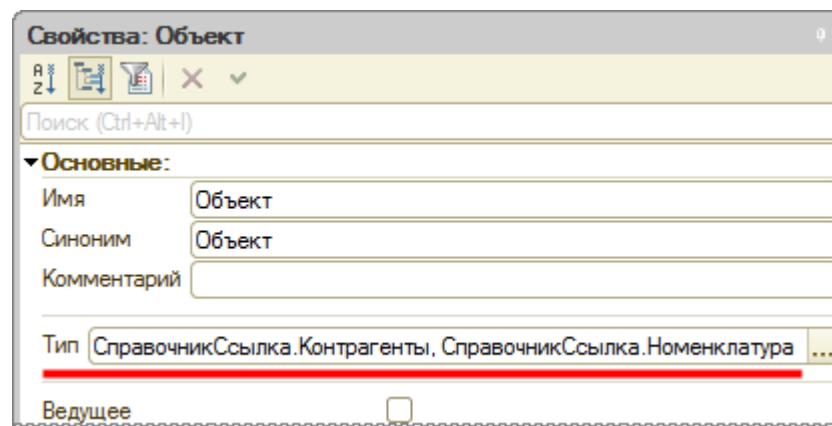
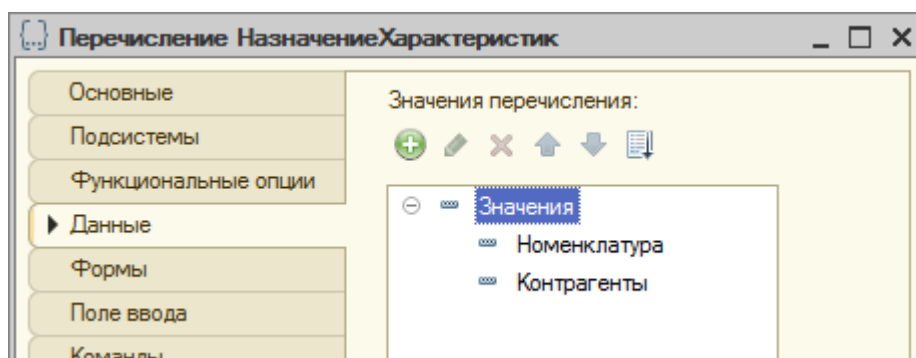


Рисунок 9 – Использование составного типа для измерения регистра

Чтобы для каждого вида характеристик указать, к какому объекту метаданных он относится, в ПВХ добавим реквизит *Принадлежность* с типом данных *ПеречислениеСсылка.НазначениеХарактеристик*. Для этого предварительно создадим перечисление *НазначениеХарактеристик* со значениями «Номенклатура» и «Контрагенты».



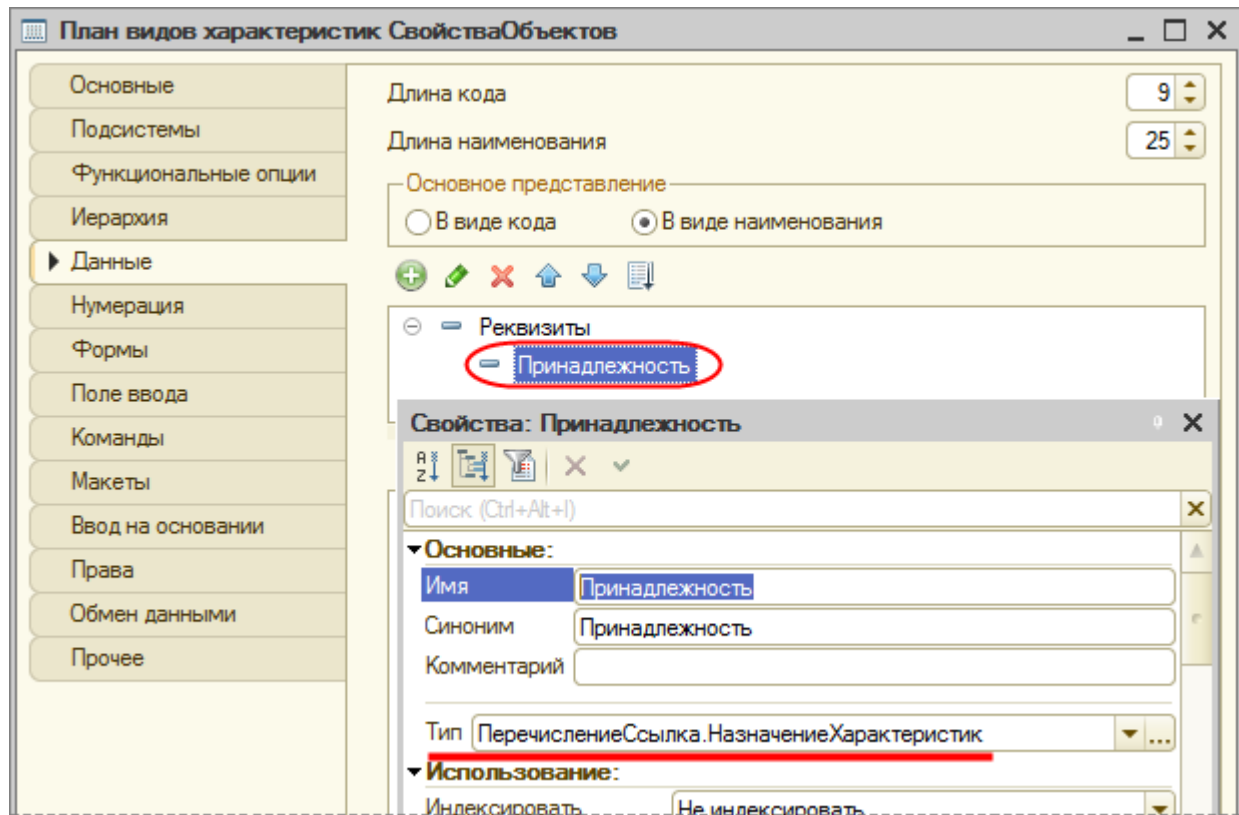


Рисунок 10 – Добавление реквизита Принадлежность в ПВХ

В пользовательском режиме для каждого элемента плана видов характеристик необходимо заполнить реквизит *Принадлежность*:

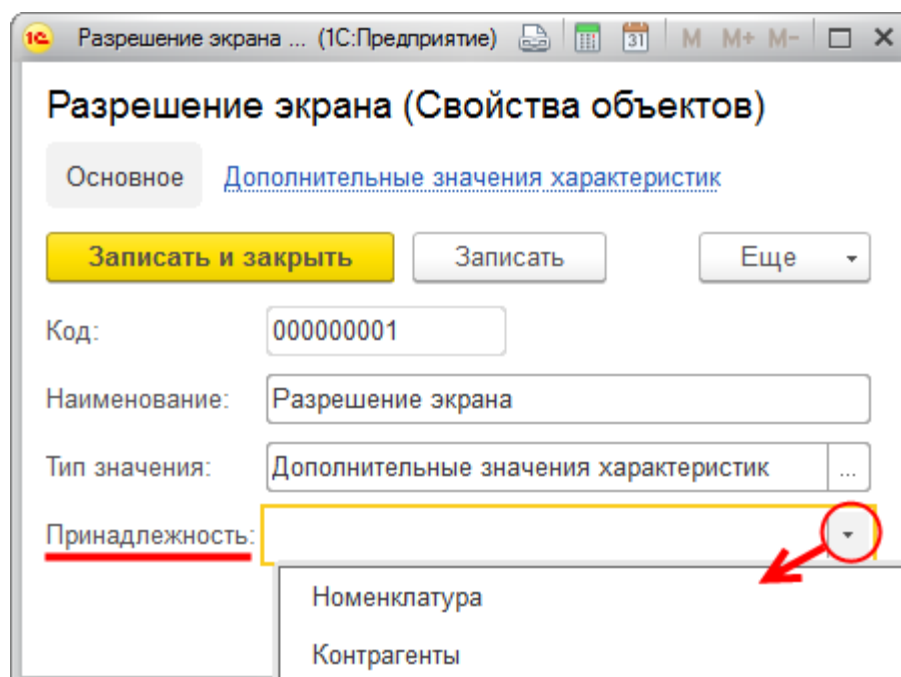


Рисунок 11 – Заполнение реквизита Принадлежность

Таким образом можно однозначно разделить характеристики для разных типов объектов.

Итоги

Итак, в данном разделе рассмотрено, как организовать хранение значений характеристик в базе.

Далее перейдем непосредственно к настройкам объектов метаданных, использующих дополнительные свойства, чтобы эти свойства были доступны в отчетах и пользовательских формах.

31. Как реализовать использование дополнительных свойств объектов в отчетах, чтобы ускорить разработку на экзамене

Постановка задачи:

Необходимо реализовать использование дополнительных свойств объектов в отчетах. Произвести в конфигураторе настройки объектов метаданных для обеспечения использования дополнительных свойств экземпляров этих объектов в пользовательском режиме (в отчетах и формах).

При выполнении поставленной задачи используются объекты конфигурации, созданные в ходе изучения вариантов решения предыдущей задачи.

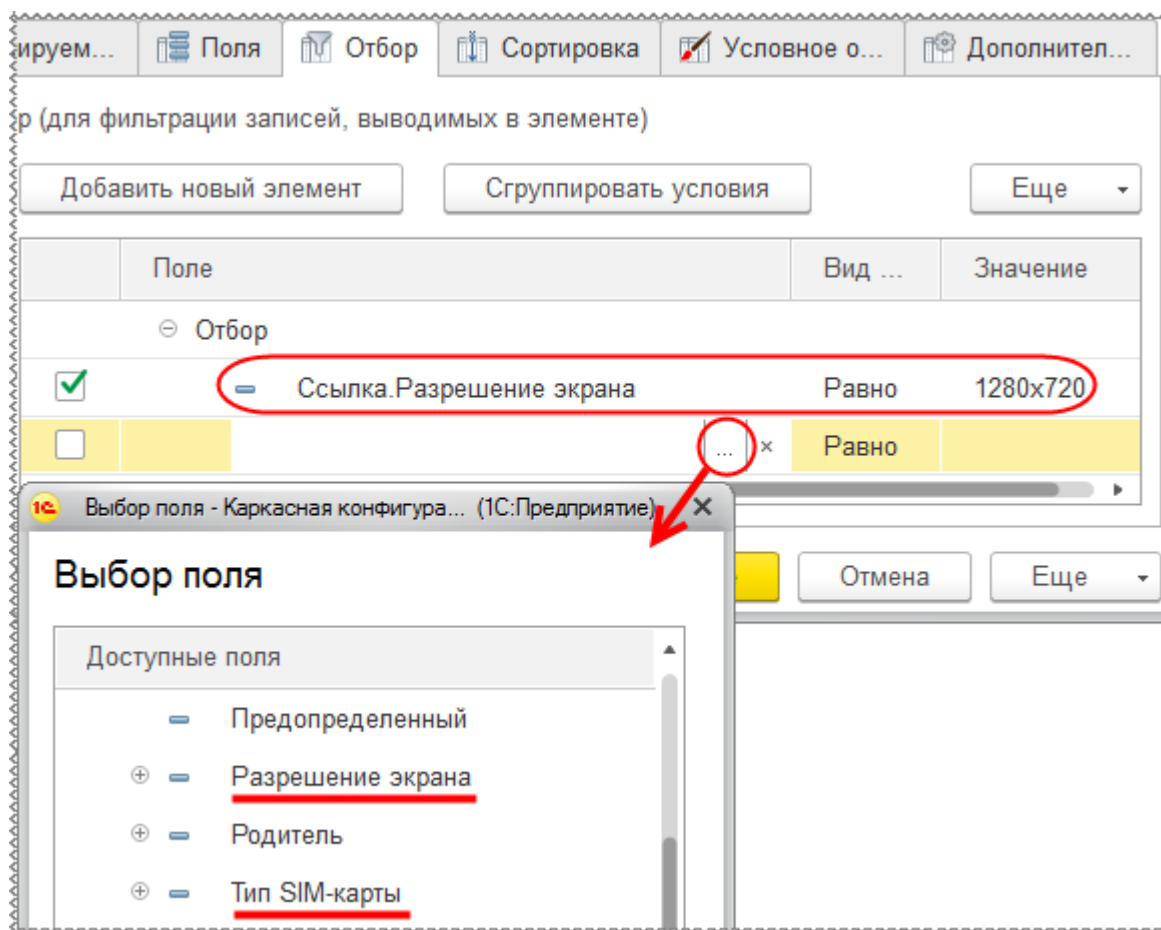
Пользователь может использовать характеристики, назначенные на уровне объекта метаданных, в отборах отчета, построенного на СКД, в настройках динамических списков и т.д.

Использование механизма характеристик в отчетах на СКД

При использовании ПВХ пользователь в режиме «1С:Предприятие» может настроить произвольные характеристики для различных объектов базы. Специальные механизмы для работы с характеристиками предоставляет Система компоновки данных.

В конфигураторе для конкретного отчета можно описать характеристики, которые в настройках отчета в пользовательском режиме будут использоваться как реквизиты (вложенные поля объектов). Такие «реквизиты» можно использовать для отборов, выводить в отчет, использовать для условного оформления.

Рассмотрим пример. При настройке отчета в пользовательском режиме в отборах можно использовать поля «Разрешение экрана» или «Тип SIM-карты».



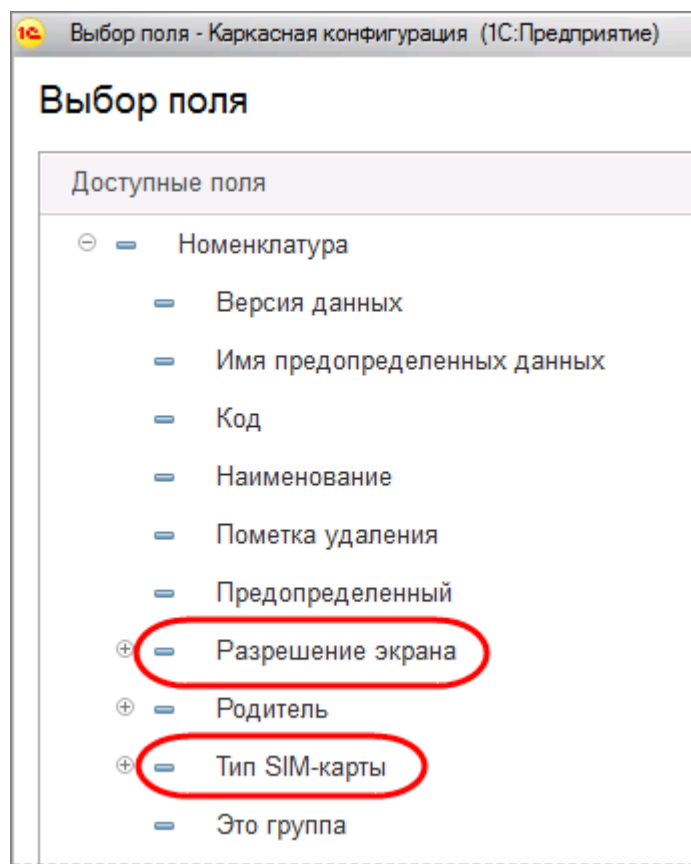


Рисунок 1 – Доступность характеристик в виде вложенных полей

Однако в справочнике «Номенклатура» такие реквизиты в конфигураторе не создавались. В предыдущем разделе демонстрировалось, как организовать назначение таких характеристик при помощи ПВХ. Система компоновки данных позволяет выводить дополнительные характеристики вместе с реквизитами объектов. Причем пользователю даже не нужно знать, является ли конкретное поле реквизитом объекта или характеристикой. В настройках отчета они будут отображаться одинаково – как реквизиты объекта (в виде вложенных полей).

Подобное поведение можно реализовать при помощи расширения языка запросов для системы компоновки данных. Таким образом в отчетах на СКД можно указывать, какие виды характеристик могут быть использованы в отчете, и как значения этих характеристик получать из базы. Рассмотрим этот механизм подробнее.

Описание характеристик с помощью расширения языка запросов

В конструкторе схемы компоновки данных для набора данных-запроса можно обратиться к конструктору запроса. В открываемом таким образом конструкторе запроса доступна закладка *Характеристики*.

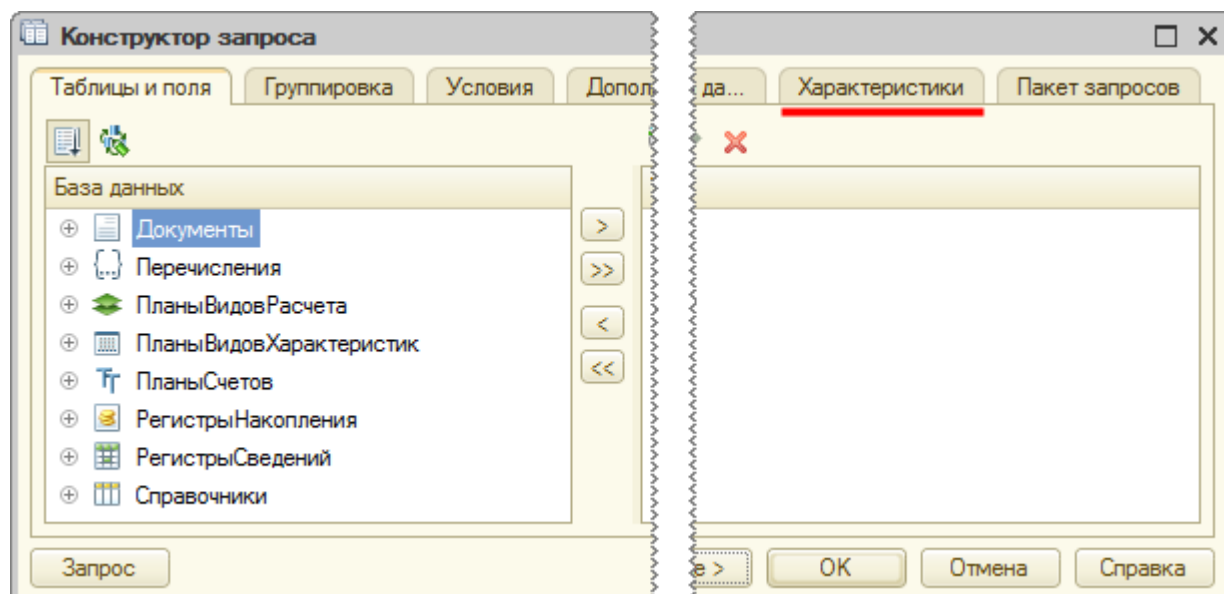


Рисунок 2 – Конструктор запроса

На закладке *Характеристики* описывается, какие виды характеристик должны использоваться в отчете, а также как получить из базы данных значения этих характеристик:

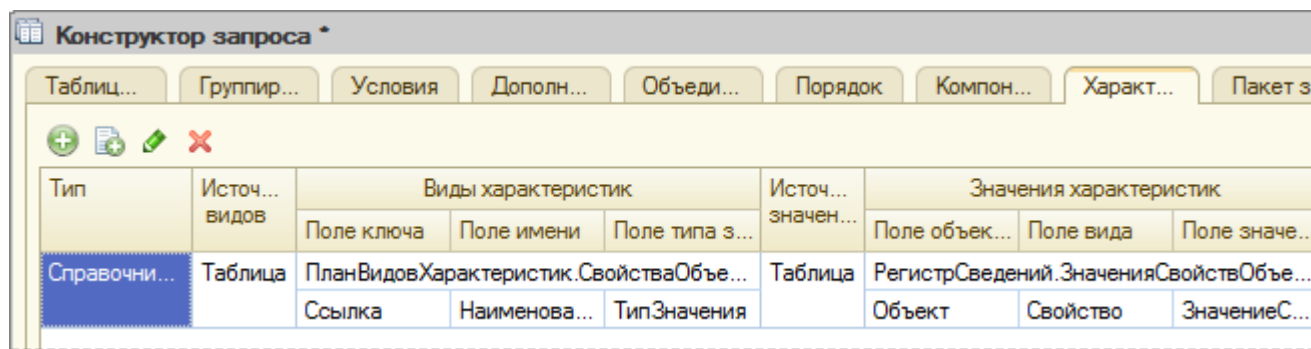


Рисунок 3 – Настройка характеристик в конструкторе запроса

Данную таблицу можно разделить на три основные части:

1. **Тип данных** – конкретный справочник или документ, для которого в отчете нужно использовать характеристики
2. **Виды характеристик** – описание того, как системе получить виды характеристик для конкретного типа данных из п. 1
3. **Значения характеристик** – описание того, как системе получить значения характеристик из п. 2.

В колонке *Тип* нужно указать, для какого типа данных будут использоваться характеристики. Например, можно указать тип *СправочникСсылка.Номенклатура*. Это означает, что для полей

указанного типа в настройках отчета характеристики будут доступны как вложенные поля (как «реквизиты» этого объекта).

Рассмотрим подробнее, как следует заполнить отдельные колонки на закладке *Характеристики*.

Колонка *Источник видов* может принимать два значения:

- *Запрос* – если список видов характеристик нужно получать из базы запросом. В этом случае сам запрос указывается в поле *Виды характеристик*
- *Таблица* – если виды характеристик хранятся в таблице базы. Например, в таблице плана видов характеристик – в этом случае в поле *Виды характеристик* указывается данная таблица.

Для вида характеристики необходимо заполнить поля:

- *Поле ключа* – идентификатор вида характеристики, например, ссылка на план видов характеристик
- *Поле имени* – указывается наименование характеристики
- *Поле типа значения* – указывается тип значения характеристики.

Поле типа значения может не заполняться, тогда тип характеристики будет *Булево*.

Если *Источник видов* принимает значение *Таблица*, то система формирует запрос на основании указанной таблицы и списка полей, указанных при настройке характеристик.

Вторая часть настроек характеристик представляет собой описание способа получения значений характеристик.

Как и колонка *Источник видов*, колонка *Источник значений* может принимать два значения:

- *Запрос* – если список значений характеристик нужно получать из базы запросом. В этом случае сам запрос указывается в поле *Значения характеристик*
- *Таблица* – если значения характеристик хранятся в таблице базы. Например, в таблице регистра сведений – в этом случае в поле *Значения характеристик* указывается данная таблица.

Необходимо заполнить поля:

- *Поле объекта* – имя поля с идентификатором объекта, для которого получается значение характеристики, например, измерение *Объект* регистра сведений, содержащее ссылку на номенклатуру, контрагента и т.д.
- *Поле вида* – имя поля с идентификатором вида характеристики, например, измерение *Свойство* регистра сведений, содержащее ссылку на элемент плана видов характеристик
- *Поле значения* – имя поля со значением характеристики, например, ресурс *ЗначениеСвойства* регистра сведений.

Поле значения может не заполняться. В этом случае считается, что значение имеет тип Булево. Если у объекта есть характеристика, значение характеристики будет равно Истина. В противном случае значение равно Ложь.

При сохранении запроса в конструкторе в полученный текст запроса будут добавлены конструкции в фигурных скобках:

{ХАРАКТЕРИСТИКИ

ТИП(Справочник.Номенклатура)

ВИДЫХАРАКТЕРИСТИК ПланВидовХарактеристик.СвойстваОбъектов

ПОЛЕКЛЮЧА Ссылка

ПОЛЕИМЕНИ Наименование

ПОЛЕТИПАЗНАЧЕНИЯ ТипЗначения

ЗНАЧЕНИЯХАРАКТЕРИСТИК РегистрСведений.ЗначенияСвойствОбъектов

ПОЛЕОБЪЕКТА Объект

ПОЛЕВИДА Свойство

ПОЛЕЗНАЧЕНИЯ ЗначениеСвойства }

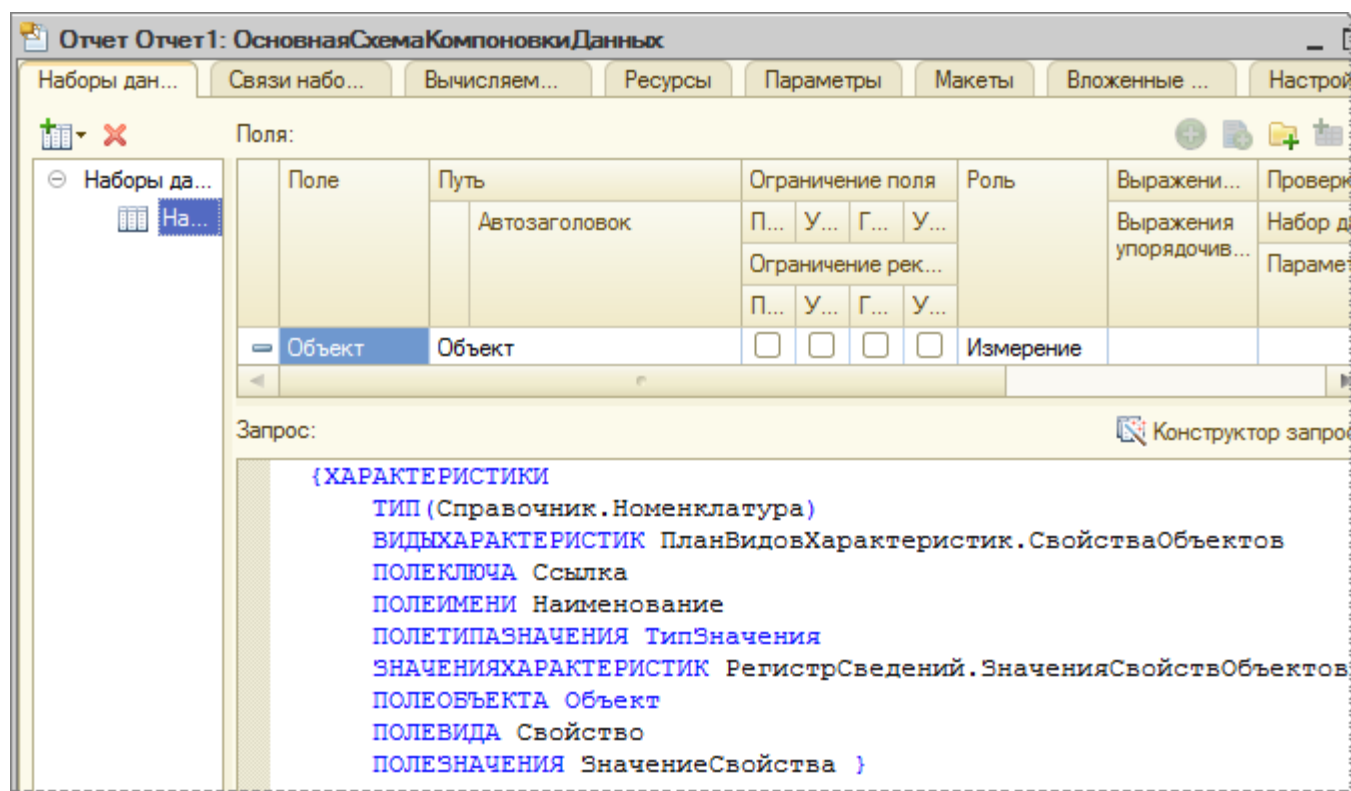


Рисунок 4 – Конструкции в фигурных скобках, добавленные в текст запроса

Итак, при помощи подобных конструкций описываются характеристики, которые будут использоваться в конкретном отчете. Характеристики можно настроить интерактивно в конструкторе

запроса, как было рассмотрено выше, либо секцию в фигурных скобках можно прописать вручную в тексте запроса.

Указание характеристик на уровне объектов метаданных

Чтобы в конкретном отчете можно было использовать характеристики того или иного объекта, требуется в схеме компоновки данных использовать расширение языка запросов для системы компоновки данных (предложение **ХАРАКТЕРИСТИКИ**).

Если необходимо, чтобы характеристики могли использоваться во всех отчетах конфигурации, придется вручную во всех схемах компоновки добавить конструкцию в фигурных скобках. Если в конфигурации присутствует большое количество отчетов, указание такого предложения в каждом отчете может занять много времени. Сама конструкция может быть простой, но ее необходимо будет скопировать во все отчеты, которых в современных типовых конфигурациях достаточно много.

Намного удобнее было бы один раз настроить характеристики на уровне самого объекта метаданных, а при формировании отчетов система использовала бы эти настройки. Такая возможность в платформе «1С:Предприятие» предусмотрена.

Для настройки характеристик на уровне объекта метаданных нужно в дереве конфигурации выбрать соответствующий объект, например, справочник «Номенклатура», открыть контекстное меню и выбрать пункт *Характеристики*.

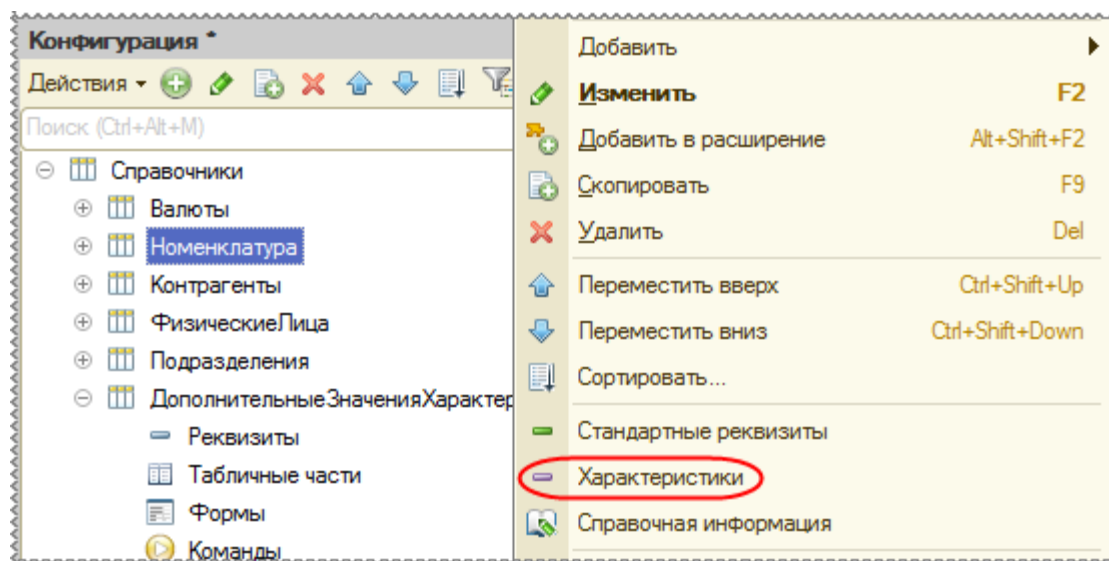


Рисунок 5 – Контекстное меню для объекта метаданных

В открывшемся окне необходимо описать, где хранятся виды характеристик и как получить значения характеристик из базы. Поэтому таблица разделена на две части (описания двух источников данных): *Виды характеристик* и *Значения характеристик*.

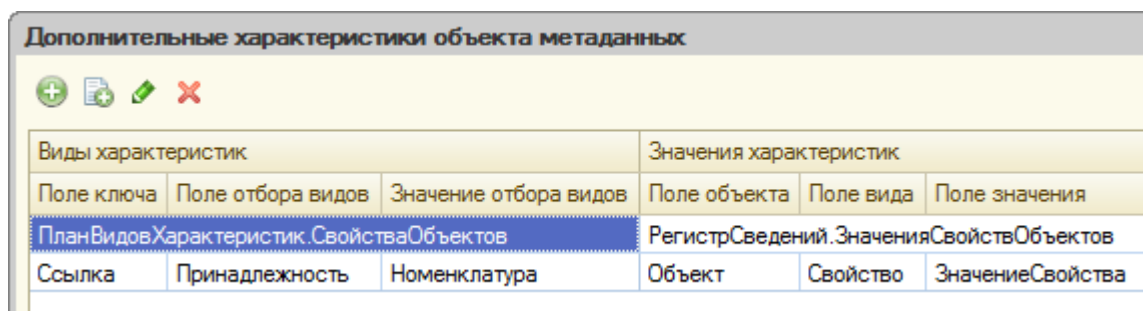


Рисунок 6 – Настройка характеристик на уровне объекта метаданных

Поля описания используемых видов характеристик:

- *Виды характеристик:* указывается объект метаданных – таблица, в которой хранятся виды характеристик. Чаще используется план видов характеристик, но можно указать и табличную часть справочника с перечнем используемых свойств:

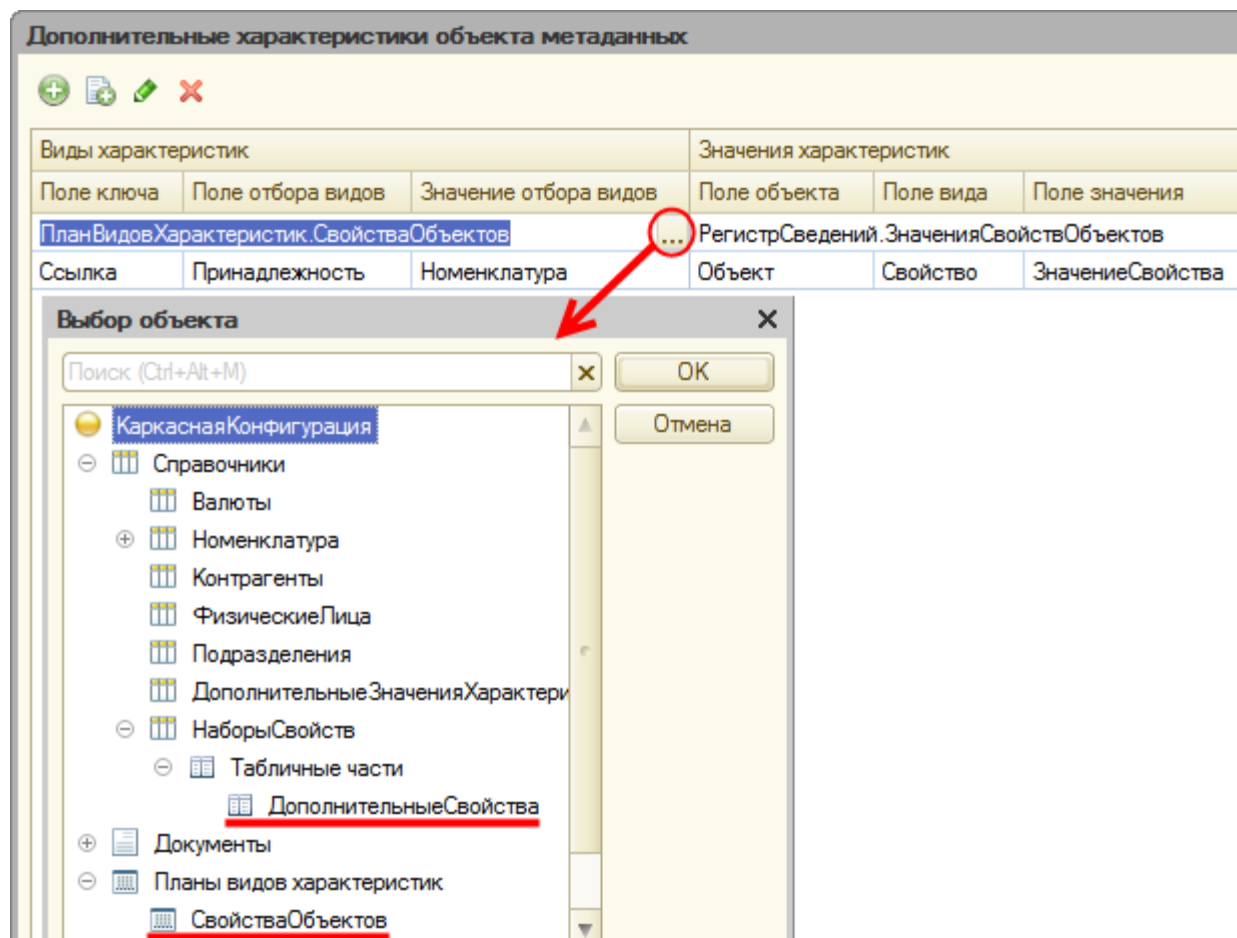


Рисунок 7 – Указание объекта метаданных, в котором хранятся виды характеристик

- *Поле ключа* – ссылка на вид характеристики

- *Поле отбора видов* используется, если в таблице-источнике видов характеристик (ПВХ или ТЧ справочника с наборами свойств) хранятся свойства, относящиеся к различным объектам метаданных. При помощи отбора можно выбрать характеристики только текущего объекта метаданных. При отборе из ПВХ указывается поле со значением принадлежности к настраиваемому объекту метаданных
- *Значение отбора видов:* При использовании отбора характеристик настраиваемого объекта метаданных указывается, какое значение должно быть для поля, по которому осуществляется отбор характеристик.

Поля описания получения значений характеристик:

- *Значения характеристик:* (указывается объект метаданных – таблица, в которой хранятся значения характеристик. Используется регистр сведений или табличная часть объекта метаданных):

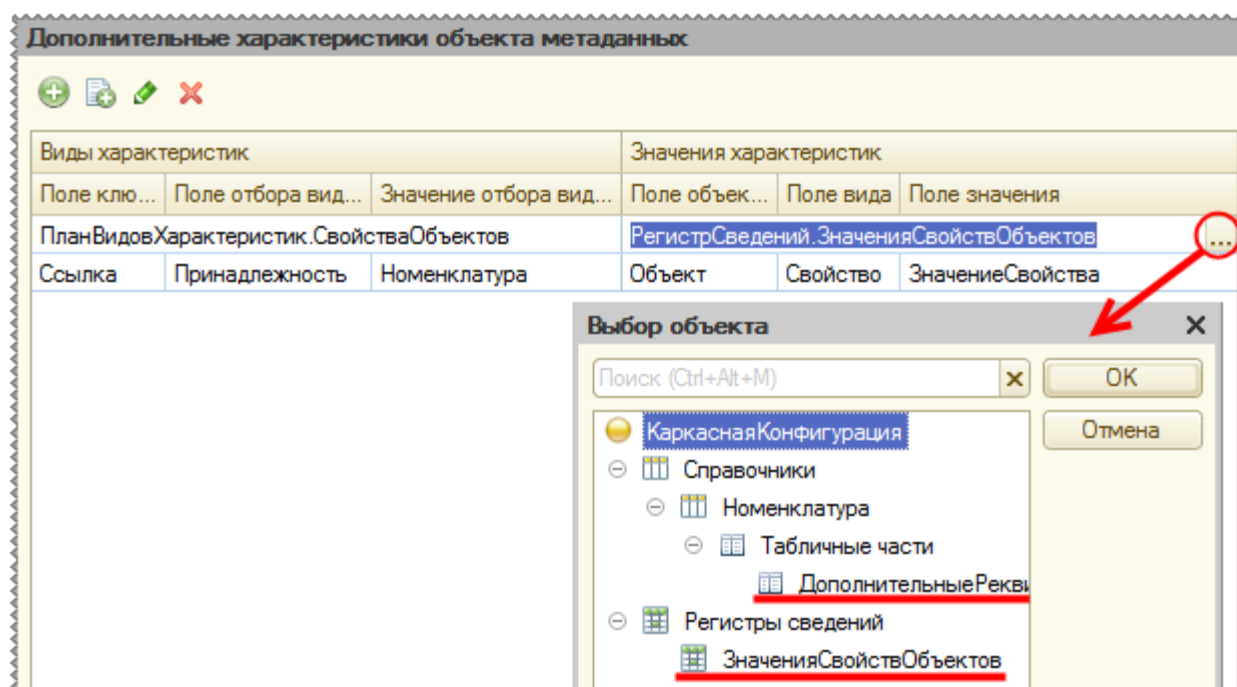


Рисунок 8 – Указание объекта метаданных, в котором хранятся значения характеристик

- *Поле объекта* – имя поля с идентификатором объекта, для которого получается значение характеристики, например, измерение *Объект* регистра сведений, содержащее ссылку на номенклатуру, контрагента и т.д.
- *Поле вида* – имя поля с идентификатором вида характеристики, например, измерение *Свойство* регистра сведений, содержащее ссылку на элемент плана видов характеристик
- *Поле значения* – имя поля со значением характеристики, например, ресурс *ЗначениеСвойства* регистра сведений.

После настройки характеристик на уровне объекта метаданных дополнительных действий в отчете выполнять не нужно, указывать в тексте запроса предложение **ХАРАКТЕРИСТИКИ** не потребуется, система автоматически увидит в отчете такие характеристики и представит их как «вложенные» поля, что продемонстрировано на скриншоте:

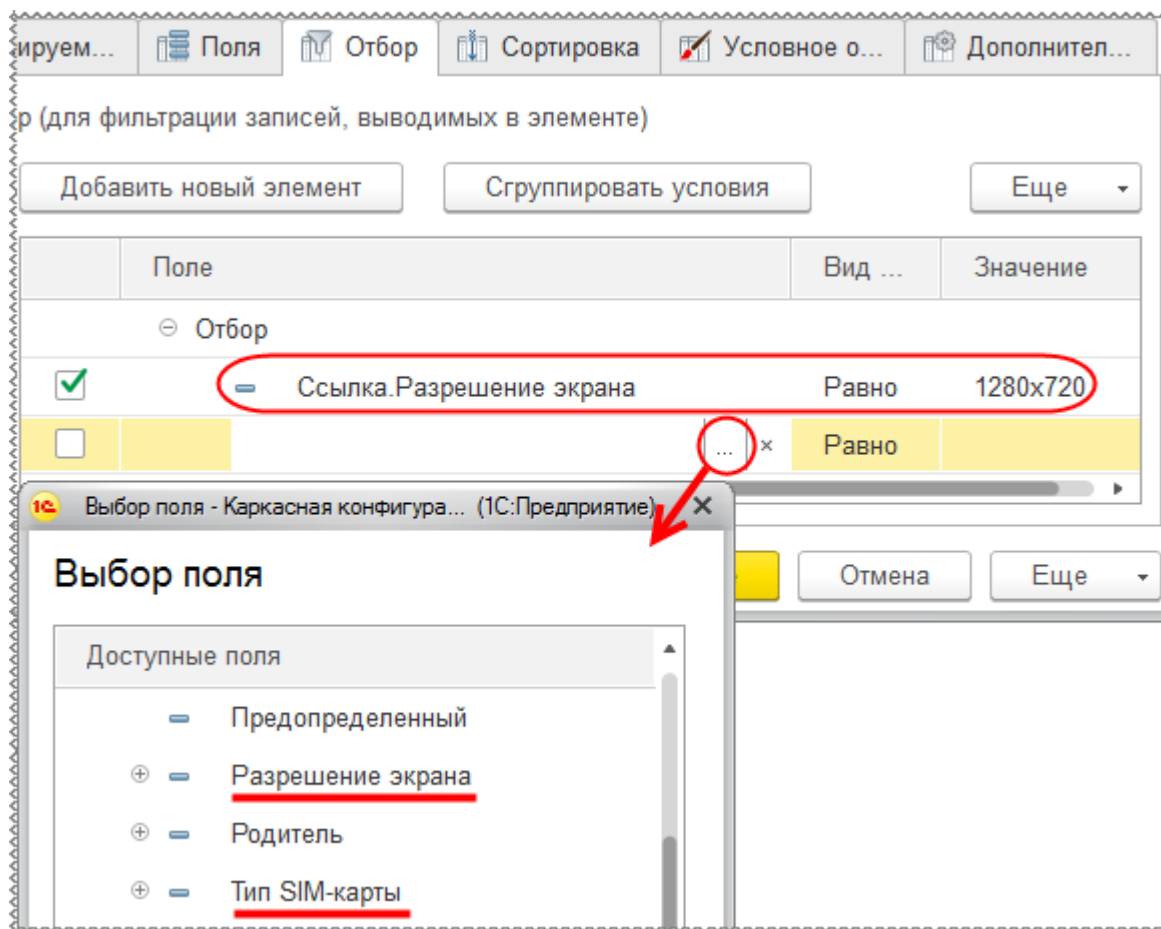


Рисунок 9 – Доступность характеристик в виде вложенных полей

Важно отметить, что динамические списки тоже используют настройки характеристик на уровне объектов метаданных. Например, в динамическом списке можно настроить отбор по характеристике товаров.

Важно. Использовать характеристики (для отборов, выходных полей, условного оформления и т.д.) в отчете или в динамическом списке возможно только в пользовательском режиме, поскольку сначала должен быть выполнен запрос, извлекающий виды характеристик из базы данных. В конфигураторе эти данные недоступны, т.к. работа производится не с самими данными, а только с их описанием.

Настройка характеристик для различных вариантов их хранения

Рассмотрим примеры настройки характеристик объектов метаданных при различных вариантах их хранения в базе.

1. Характеристики назначены для справочника *Номенклатура*. Перечень свойств содержится в плане видов характеристик, а значения характеристик хранятся в регистре сведений.

Поскольку характеристики назначены только для номенклатуры, *Поле отбора видов* и *Значение отбора* не заполняются. Настройка выглядит следующим образом:

Виды характеристик			Значения характеристик		
Поле ключа	Поле отбора видов	Значение отбора видов	Поле объекта	Поле вида	Поле значения
ПланВидовХарактеристик.СвойстваОбъектов			РегистрСведений.ЗначенияСвойствОбъектов		
Ссылка			Объект	Свойство	ЗначениеСвойства

Рисунок 10 – Настройка характеристик для справочника «Номенклатура»

2. В плане видов характеристик существует реквизит *Принадлежность*, который разделяет характеристики для разных объектов метаданных (справочники номенклатуры, контрагентов, складов, валют, видов цен, организаций и т.д.). Значения характеристик хранятся в регистре сведений.

Чтобы осуществить отбор характеристик данного объекта метаданных (справочника), необходимо заполнить *Поле отбора видов* и *Значение отбора*. Аналогичный пример был рассмотрен ранее.

Виды характеристик			Значения характеристик		
Поле ключа	Поле отбора видов	Значение отбора видов	Поле объекта	Поле вида	Поле значения
ПланВидовХарактеристик.СвойстваОбъектов			РегистрСведений.ЗначенияСвойствОбъектов		
Ссылка	Принадлежность	Номенклатура	Объект	Свойство	ЗначениеСвойства

Рисунок 11 – Настройка характеристик при использовании поля *Принадлежность*

Итак, в данном блоке рассмотрена настройка характеристик на уровне объектов метаданных. В следующем блоке рассмотрим, как выполнить отображение полей для редактирования свойств прямо на форме объекта. Таким образом, пользователю не нужно будет обращаться к регистру сведений для редактирования значений характеристик. Такой подход позволит упростить работу со свойствами в конфигурации.

32. Как программно разместить на форме поля для редактирования характеристик

Постановка задачи:

Значения характеристик хранятся в регистре сведений. Пользователь непосредственно не работает с этим регистром, все значения он редактирует на форме объекта.

На форме элемента справочника «Контрагенты» должны отображаться дополнительные поля ввода, предназначенные для редактирования значений характеристик текущего контрагента. При открытии формы элемента справочника «Контрагенты» — значения свойств считываются из регистра сведений и отображаются на форме. При закрытии формы элемента справочника «Контрагенты» происходит обратное действие — отредактированные значения свойств сохраняются в регистр сведений.

Подобная задача (5.16) присутствует в сборнике для подготовки к экзамену «1С:Специалист по платформе «1С:Предприятие 8.3».

Решение:

Первым делом настроим дополнительные характеристики справочника «Контрагенты»:

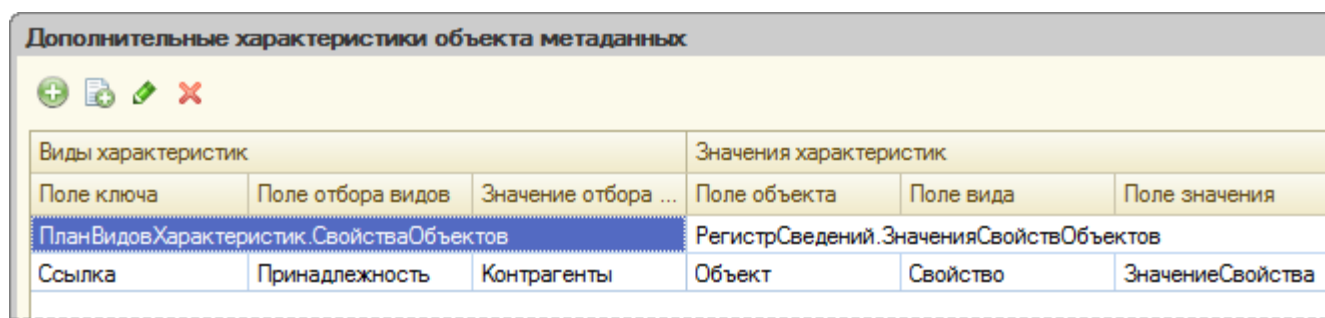


Рисунок 19 – Настройка характеристик для справочника «Контрагенты»

Рассмотрим общую логику решения задачи.

Свойства справочника «Контрагенты» хранятся в ПВХ, значения свойств – в регистре сведений. Это не реквизиты, не табличные части справочника, это внешние по отношению к справочнику «Контрагенты» объекты метаданных. Поэтому поля ввода для редактирования значений свойств будем создавать на форме программно. Для каждого свойства из ПВХ будем добавлять на форму один реквизит – одно поле ввода.

При открытии формы элемента справочника считываем значения свойств из регистра сведений и отображаем их в полях на форме. При записи элемента справочника создаем в регистре сведений записи со значениями свойств.

Теперь перейдем к реализации данного функционала.

Создадим форму элемента справочника «Контрагенты». На форме добавим группу «ГруппаСвойства»:

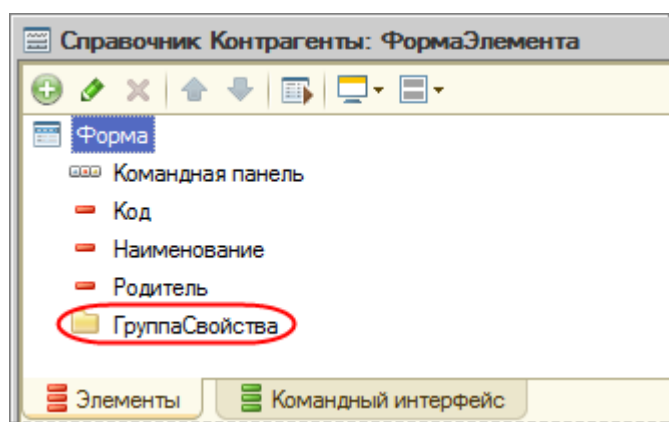


Рисунок 20 – Размещение группы на форме

В данной группе будут располагаться все поля ввода, создаваемые программно. Для группы свойство *Группировка* установим в значение *Вертикальная*, чтобы новые поля для отображения значений свойств располагались друг под другом:

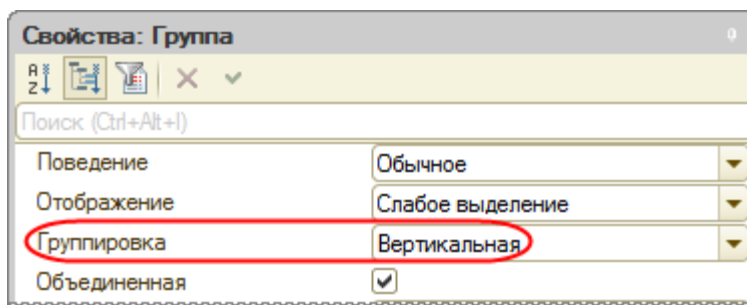


Рисунок 21 – Настройка свойств группы

Создадим реквизит формы *ТаблицаСвойств* с типом *ТаблицаЗначений*. В таблице будет храниться список реквизитов, которые программно добавляются на форму.

Для созданного реквизита добавим колонку *ИмяПоля* (тип *Строка 100*), предназначенную для хранения наименований характеристик, и колонку *Свойство* (тип *ПланВидовХарактеристикСсылка.СвойстваОбъектов*) для хранения значений характеристик:

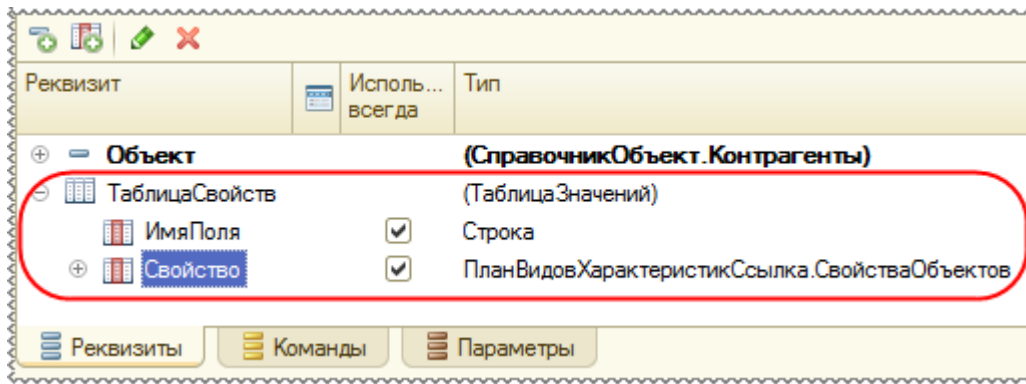


Рисунок 22 – Создание таблицы значений на форме

Для отображения на форме полей со значениями свойств необходимо получить из базы перечень дополнительных характеристик, которые назначены справочнику «Контрагенты». Эти данные сохраним в реквизите формы *ТаблицаСвойств*. В созданных полях ввода нужно отобразить значения свойств, которые хранятся в регистре сведений. Для этого также потребуется обращение к базе данных. Следовательно, программный код должен выполняться на сервере. Для программного создания полей формы подойдет обработчик события *ПриСозданииНаСервере*.

&НаСервере

Процедура *ПриСозданииНаСервере*(Отказ, СтандартнаяОбработка)

```
// 1. Из плана видов характеристик получим список свойств, которые назначены справочнику
«Контрагенты», и типы их значений
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| СвойстваОбъектов.Ссылка КАК Ссылка,
| СвойстваОбъектов.Наименование КАК Наименование,
| СвойстваОбъектов.ТипЗначения КАК ТипЗначения
| ИЗ
| ПланВидовХарактеристик.СвойстваОбъектов КАК СвойстваОбъектов
| ГДЕ
| СвойстваОбъектов.Принадлежность = &Принадлежность";
Запрос.УстановитьПараметр("Принадлежность", Перечисления.НазначениеХарактеристик.Контрагенты);

Выборка = Запрос.Выполнить().Выбрать();

// 2. Для каждого свойства создадим новый реквизит формы соответствующего типа и новое поле,
заполним таблицу свойств
Сч = 1;
Пока Выборка.Следующий() Цикл

    ИмяПоляСвойства = "ПолеСвойство" + Формат(Сч, "ЧГ=0");

// 3. Добавление реквизита формы соответствующего типа
ДобавляемыеРеквизиты = Новый Массив;
Реквизит = Новый РеквизитФормы(ИмяПоляСвойства, Выборка.ТипЗначения, ,
```

```

Выборка.Наименование);
    ДобавляемыеРеквизиты.Добавить(Реквизит);
    ИзменитьРеквизиты(ДобавляемыеРеквизиты);

// 4. Добавление на форму нового поля
НовыйЭлементФормы = Элементы.Добавить(ИмяПоляСвойства, Тип("ПолеФормы"),
Элементы.ГруппаСвойства);
НовыйЭлементФормы.Вид = ВидПоляФормы.ПолеВвода;
НовыйЭлементФормы.ПутьКДанным = ИмяПоляСвойства;
НовыйЭлементФормы.Заголовок = Выборка.Наименование;
НовыйЭлементФормы.Видимость = Истина;
НовыйЭлементФормы.Доступность = Истина;

// 5. Заполнение таблицы свойств
НоваяСтрокаТаблицыСвойств = ТаблицаСвойств.Добавить();
НоваяСтрокаТаблицыСвойств.ИмяПоля = ИмяПоляСвойства;
НоваяСтрокаТаблицыСвойств.Свойство = Выборка.Ссылка;

Сч = Сч + 1;

КонецЦикла;

Если НЕ Объект.Ссылка.Пустая() Тогда

// 6. Считаем значения характеристик для текущего элемента справочника «Контрагенты»
Запрос = Новый Запрос;
Запрос.УстановитьПараметр("Объект", Объект.Ссылка);

Запрос.Текст =
"ВЫБРАТЬ
|   ЗначенияСвойствОбъектов.Свойство КАК Свойство,
|   ЗначенияСвойствОбъектов.ЗначениеСвойства КАК ЗначениеСвойства
|ИЗ
|   РегистрСведений.ЗначенияСвойствОбъектов КАК ЗначенияСвойствОбъектов
|ГДЕ
|   ЗначенияСвойствОбъектов.Объект = &Объект";

Выборка = Запрос.Выполнить().Выбрать();

// 7. Заполнение добавленных полей формы найденными значениями
Пока Выборка.Следующий() Цикл

    НайденныеСтроки = ТаблицаСвойств.НайтиСтроки(Новый Структура("Свойство",
Выборка.Свойство));
    Если НайденныеСтроки.Количество() > 0 Тогда
        ЭтотОбъект[НайденныеСтроки[0].ИмяПоля] = Выборка.ЗначениеСвойства;
    КонецЕсли;

КонецЦикла;

КонецЕсли;

```

КонецПроцедуры

Рассмотрим основные моменты приведенного программного кода:

1. **П. 1 – 3.** Первым делом из ПВХ нужно получить список свойств, которые назначены справочнику «Контрагенты». Для каждого свойства создаем новый реквизит формы. Для создания реквизитов используется метод управляемой формы ИзменитьРеквизиты. Тип данных получаем из поля *ТипЗначения* плана видов характеристик.
2. **П. 4 – 5.** Для каждого созданного реквизита на форму добавляем поле ввода, которое располагается в группе «ГруппаСвойства». В качестве заголовка поля ввода используем наименование элемента плана видов характеристик. В реквизите формы ТаблицаСвойств сопоставляем элемент ПВХ в базе и имя созданного на форме реквизита. В дальнейшем эти данные будут использоваться для сохранения введенных пользователем значений в базу.
3. **П. 6 – 7.** Если текущий элемент справочника «Контрагенты» не вновь созданный, то необходимо из регистра сведений получить значения характеристик для этого элемента справочника. Считанные из регистра значения присваиваем реквизитам формы, добавленным программно.

В пользовательском режиме открытая форма элемента справочника «Контрагенты» выглядит следующим образом:

Магазин техники (... (1С:Предприятие)

Магазин техники (Контрагенты)

Записать и закрыть Записать Еще ▾

Код: 000000007

Наименование: Магазин техники

Родитель: Покупатели ▾ □

Группа свойства

Регион: Москва ▾ □

Директор: Иванов ▾ □

Рисунок 23 – Расположение значений свойств на форме

На форме отображаются поля со значениями характеристик, пользователь прямо на этой форме может их редактировать, обращаться непосредственно к регистру сведений не требуется.

После редактирования необходимо значения свойств с формы сохранить в регистр сведений. Подберем событие, обработчик которого нужно создать. Поскольку будет производиться запись в базу данных, требуется реализовать серверную процедуру. Кроме этого важно, чтобы запись самого элемента справочника уже была завершена, так как в регистр будет записываться ссылка на элемент справочника.

Этим требованиям удовлетворяет событие модуля формы *ПослеЗаписиНаСервере*. Поэтому создадим обработчик именно этого события. В обработчике перебираем все строки таблицы значений *ТаблицаСвойств*, запишем в регистр новые значения характеристик.

Использованный программный код:

&НаСервере

Процедура ПослеЗаписиНаСервере(ТекущийОбъект, ПараметрыЗаписи)

Для каждого СтрокаТаблицыСвойств Из ТаблицаСвойств Цикл

```
Запись = РегистрыСведений.ЗначенияСвойствОбъектов.СоздатьМенеджерЗаписи();
Запись.Объект = ТекущийОбъект.Ссылка;
Запись.Свойство = СтрокаТаблицыСвойств.Свойство;
Запись.ЗначениеСвойства = ЭтотОбъект[СтрокаТаблицыСвойств.ИмяПоля];
Запись.Записать();
```

КонецЦикла;

КонецПроцедуры

Таким образом, задача по размещению на форме полей для редактирования характеристик выполнена. Вместе с тем разбор методик применения планов видов характеристик для обеспечения возможности хранения и использования дополнительных свойств объектов завершен.

В следующем разделе будет рассмотрено решение практической задачи с использованием ПВХ, условия которой максимально приближены к условиям экзаменационных задач и задач из сборника для подготовки к экзамену «1С:Специалист по платформе 1С:Предприятие 8.3» (1.10, 1.36, 1.46).

33. Как на экзамене решить практическую задачу – создать отчет на СКД, в котором свойства номенклатуры выводятся отдельными колонками

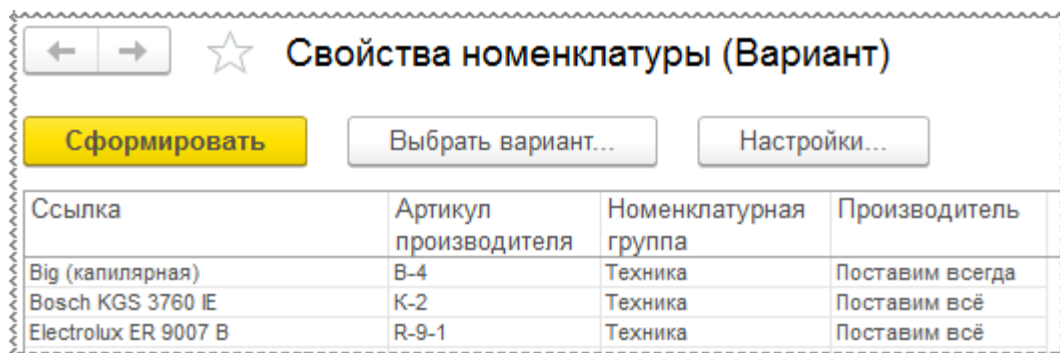
Постановка задачи:

Для справочника Номенклатура необходимо реализовать возможность назначения произвольных свойств при помощи плана видов характеристик.

В пользовательском режиме создать следующие характеристики номенклатуры:

- Номенклатурная группа (ссылка на элемент классификатора, который может дополняться пользователем: Техника, Мебель, Продукты и т.д.)
- Производитель (тип СправочникСсылка.Контрагенты)
- Артикул производителя (тип Строка)

Реализовать отчет на СКД, в котором необходимо выводить созданные свойства номенклатуры отдельными колонками:



Ссылка	Артикул производителя	Номенклатурная группа	Производитель
Big (капиллярная)	B-4	Техника	Поставим всегда
Bosch KGS 3760 IE	K-2	Техника	Поставим всё
Electrolux ER 9007 B	R-9-1	Техника	Поставим всё

Рисунок 1 – Требуемый внешний вид отчета

Для решения поставленной задачи потребуются следующие объекты конфигурации:

- Справочник «Номенклатура»
- Справочник «Контрагенты» (для хранения значений характеристики *Производитель*)
- План видов характеристик
- Справочник для хранения возможных значений дополнительных свойств
- Регистр сведений для хранения конкретных значений свойств элементов справочника «Номенклатура»

- Отчет на СКД.

Для реализации решения необходимо выполнить следующие действия:

- Произвести соответствующие настройки ПВХ и справочника значений дополнительных свойств
- Сформировать структуру и произвести соответствующие настройки регистра сведений, в котором будут храниться значения свойств элементов справочника *Номенклатура*
- Произвести настройку характеристик для объекта метаданных справочник *Номенклатура* с целью обеспечения доступности дополнительных свойств элементов справочника в отчете
- Разработать отчет с использованием СКД.

Справочники «Номенклатура» и «Контрагенты» в каркасной конфигурации уже имеются:

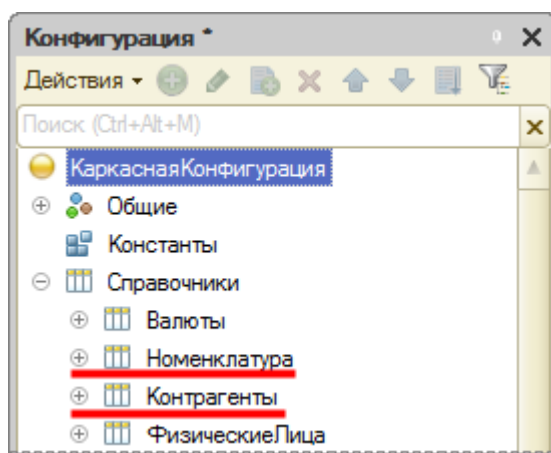


Рисунок 2 – Справочники, присутствующие в каркасной конфигурации

План видов характеристик «СвойстваОбъектов» в конфигурации также присутствует. Требуется соответствующим образом его настроить:

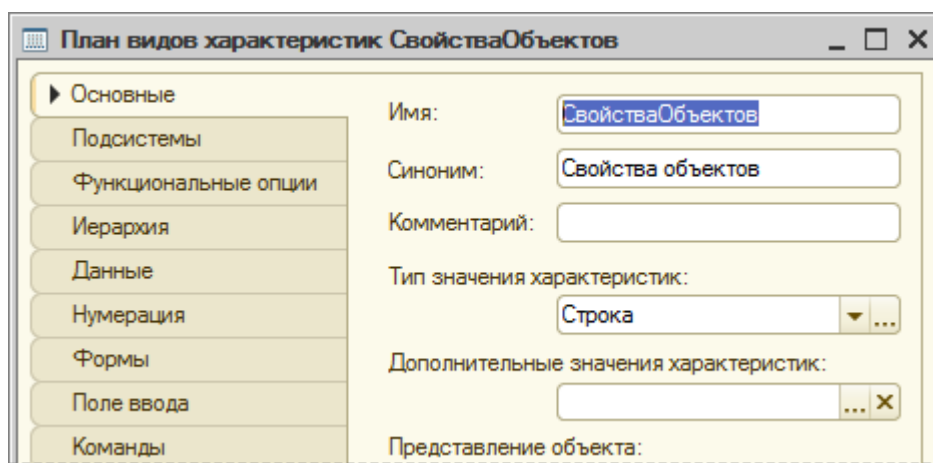


Рисунок 3 – ПВХ, который необходимо настроить при решении задачи

Сразу создадим справочник «ЗначенияХарактеристик» для хранения возможных значений дополнительных свойств.

Настройки ПВХ и справочника значений дополнительных свойств

В окне редактирования объекта конфигурации (справочник «Значения характеристик») на закладке *Владельцы* укажем в качестве владельца ПВХ «СвойстваОбъектов»:

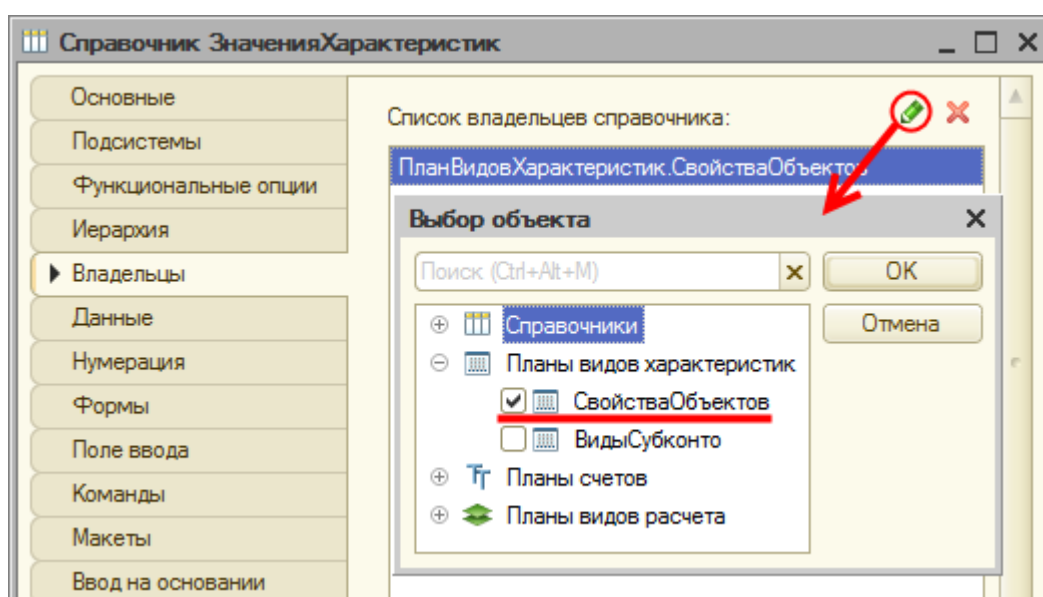


Рисунок 4 – Настройка владельца для справочника, хранящего значения дополнительных свойств

Перейдем к настройкам плана видов характеристик «СвойстваОбъектов».

В окне редактирования объекта конфигурации (ПВХ «СвойстваОбъектов») на закладке *Основные* в поле *Дополнительные значения характеристик* выберем ранее созданный справочник «Значения характеристик»:

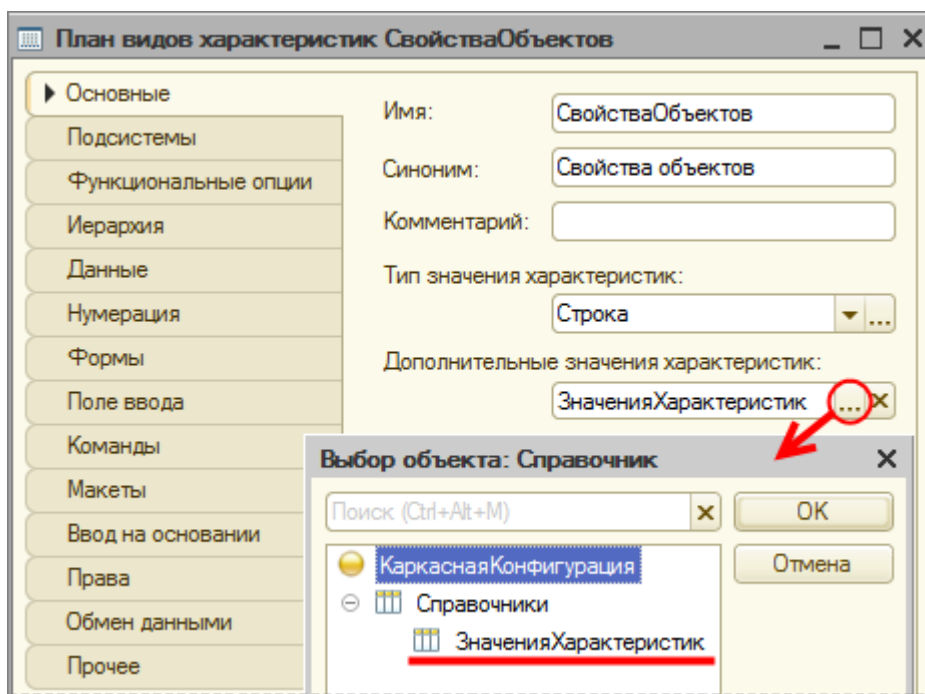


Рисунок 5 – Заполнение свойства ПВХ «Дополнительные значения характеристик»

После этого необходимо определить типы значений для характеристик. В условии задачи в явном виде перечислены следующие типы: *Строка*, *СправочникСсылка.Контрагенты* и ссылка на элемент классификатора, который может дополняться пользователем. Таким классификатором является ранее созданный справочник «ЗначенияХарактеристик».

Необходимо использовать несколько типов данных, поэтому для поля Тип значения характеристик выберем составной тип и отметим все ранее перечисленные типы:

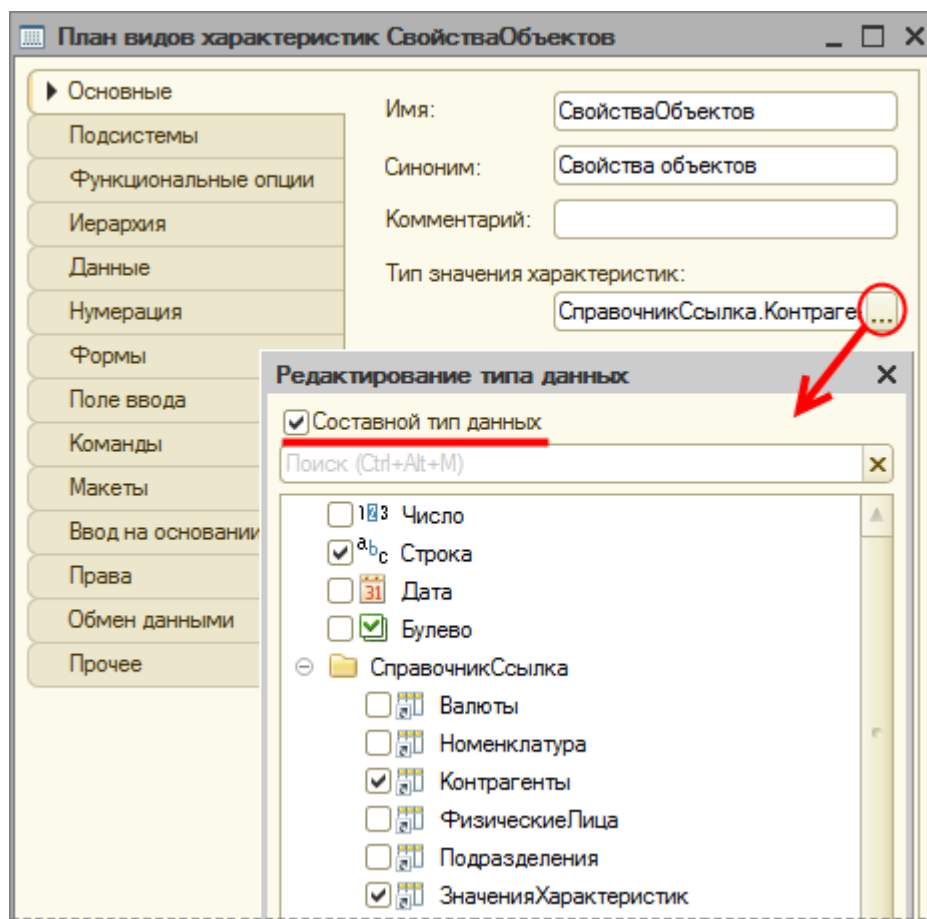


Рисунок 6 – Настройка типа значений характеристик для ПВХ

Важно! Чтобы пользователь мог использовать собственный классификатор «ЗначенияХарактеристик», в настройках ПВХ *Тип значения характеристик* в обязательном порядке должен включать справочник «ЗначенияХарактеристик». Если это не сделать, то при обновлении конфигурации базы данных система выдаст ошибку: «ПланВидовХарактеристик.СвойстваОбъектов: Ссылка на справочник значений не указана в типе значений плана видов характеристик».

Создание и настройка регистра сведений

По условию задачи назначение и хранение произвольных свойств требуется реализовать только для элементов справочника «Номенклатура». Поэтому реализация механизма определения принадлежности характеристик элементам того или иного справочника не требуется. Для хранения конкретных значений дополнительных свойств позиций номенклатуры будем использовать регистр сведений.

Создадим неперiodический регистр сведений «ЗначенияСвойствОбъектов». Непериодический, так как хранимые в нем значения с течением времени не меняются. Регистр будет иметь следующую структуру:

- Измерения:
 - Объект (тип СправочникСсылка.Номенклатура)
 - Свойство (тип ПланВидовХарактеристикСсылка.СвойстваОбъектов)
- Ресурсы:
 - ЗначениеСвойства (тип Характеристика.СвойстваОбъектов):

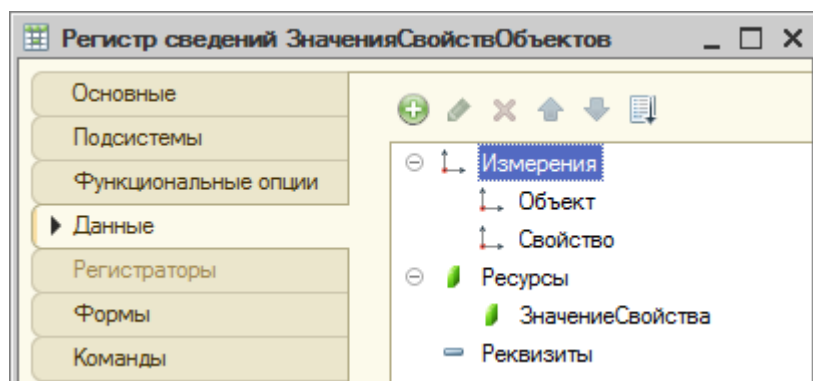


Рисунок 8 – Структура регистра сведений для хранения значений дополнительных свойств

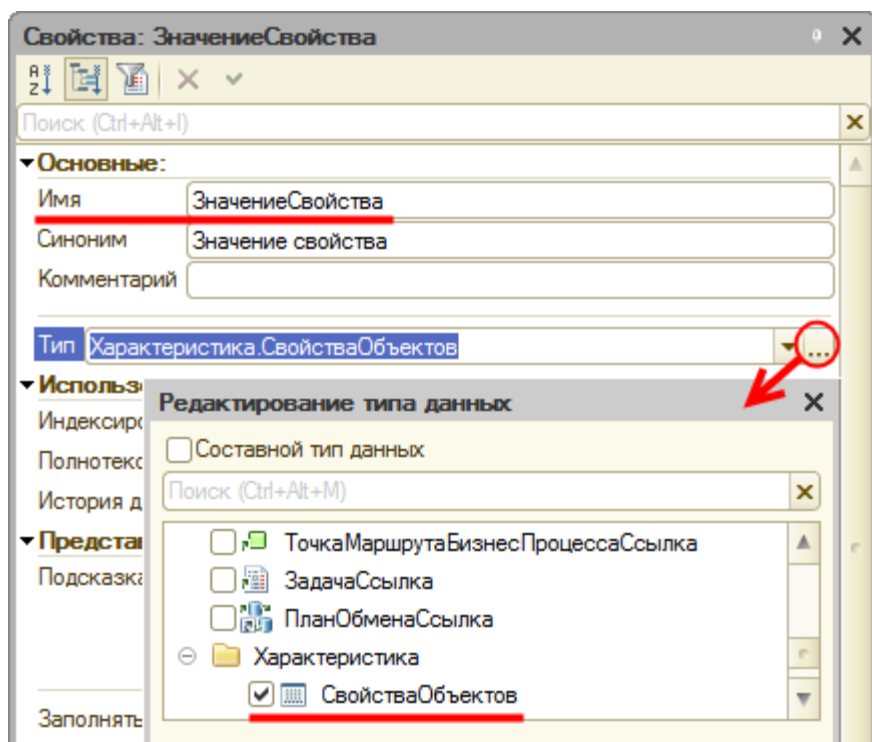


Рисунок 9 – Настройка типа данных для ресурса регистра сведений

Выбранный для ресурса тип данных *Характеристика.СвойстваОбъектов* означает, что в качестве типа данных для ресурса можно использовать все типы, которые указаны в свойстве *Тип значения характеристик* используемого ПВХ.

Для измерений регистра сведений установим свойство *Ведущее*, чтобы при удалении элементов справочника «Номенклатура», а также элементов ПВХ «СвойстваОбъектов» из базы автоматически удалялись соответствующие записи регистра сведений, которые содержат ссылку на удаляемые элементы:

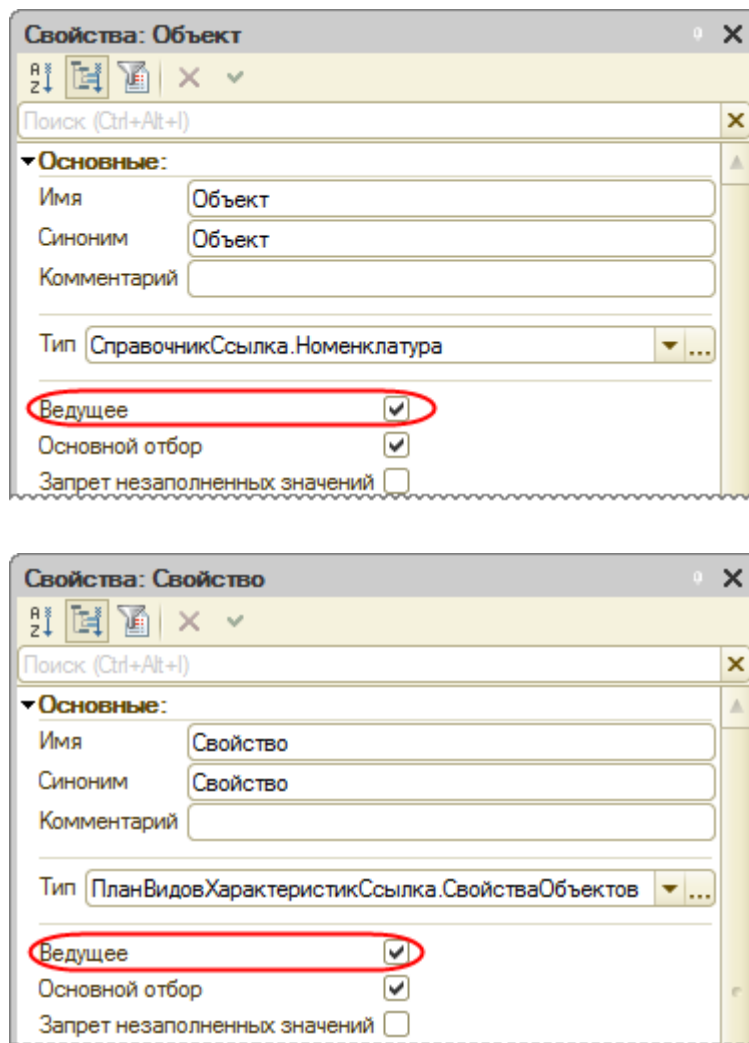


Рисунок 10 – Установка свойства «Ведущее» для измерений регистра сведений

Также платформа учитывает свойство *Ведущее* при построении командного интерфейса. Так, в панели навигации формы элемента справочника *Номенклатура* автоматически добавляется ссылка для просмотра записей регистра сведений *ЗначенияСвойствОбъектов* с отбором по текущей номенклатуре:

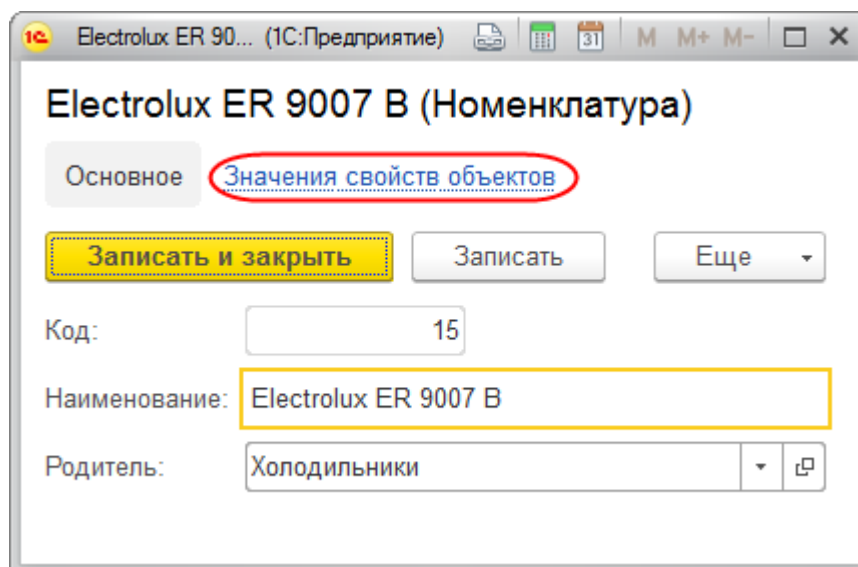


Рисунок 11 – Ссылка в панели навигации для перехода к значениям свойств текущей номенклатуры

Для ресурса *ЗначениеСвойства* необходимо выполнить настройку свойств *Связь по типу* и *Связи параметров выбора*.

В поле *Связь по типу* необходимо указать измерение *Свойство* регистра сведений. Это значит, что тип поля ввода для ресурса *ЗначениеСвойства* будет изменяться в зависимости от выбранного значения измерения *Свойство* регистра сведений:

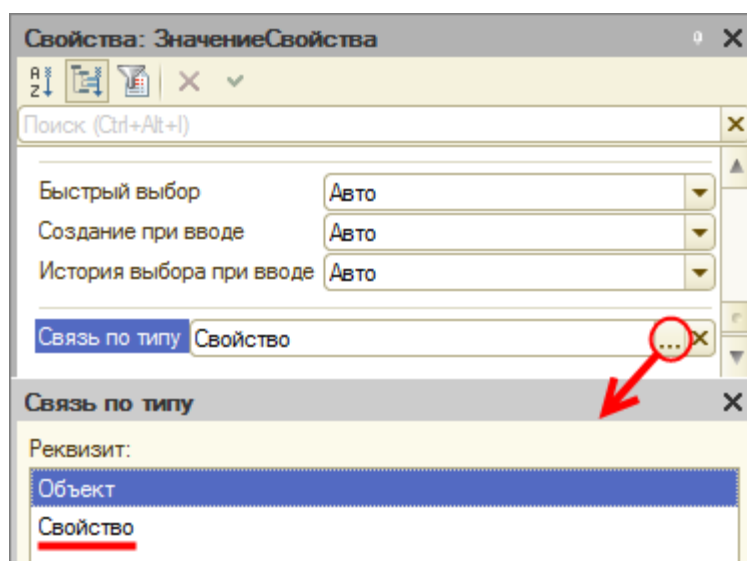


Рисунок 12 – Настройка свойства «Связь по типу»

Для ресурса *ЗначениеСвойства* в поле *Связи параметров выбора* требуется выбрать значение *Отбор.Владелец(Свойство)*. В этом случае в пользовательском режиме в качестве значения ресурса можно будет выбрать только те значения, владелец которых указан в измерении *Свойство* регистра сведений.

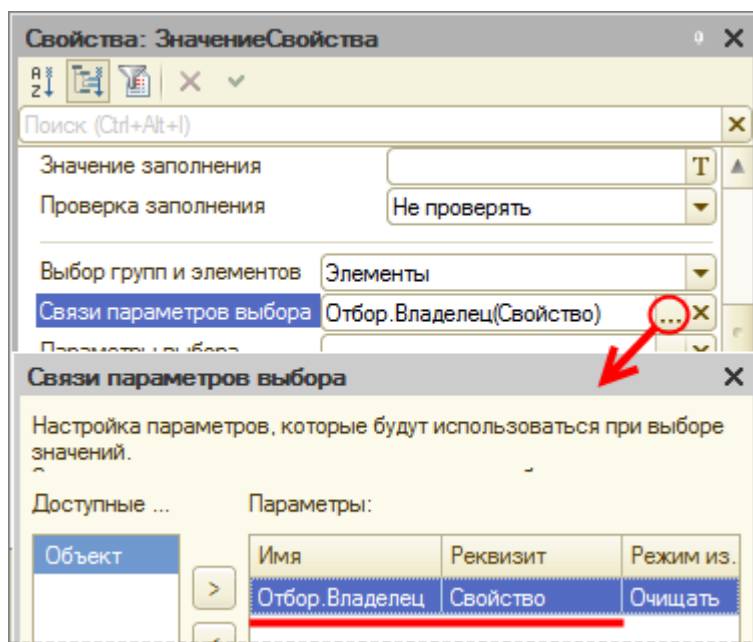


Рисунок 13 – Настройка свойства «Связи параметров выбора»

После обновления конфигурации базы данных в пользовательском режиме создадим необходимые свойства номенклатуры (элементы справочника «Свойства объектов»): *Номенклатурная группа, Производитель, Артикул производителя*:

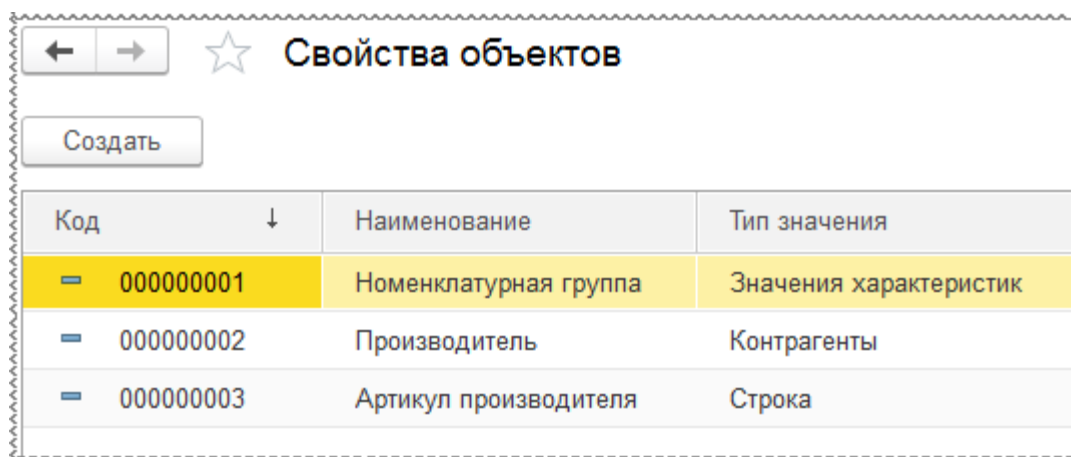


Рисунок 14 – Свойства номенклатуры, созданные в пользовательском режиме

После этого элементам справочника «Номенклатура» можно назначить значения ранее созданных свойств:

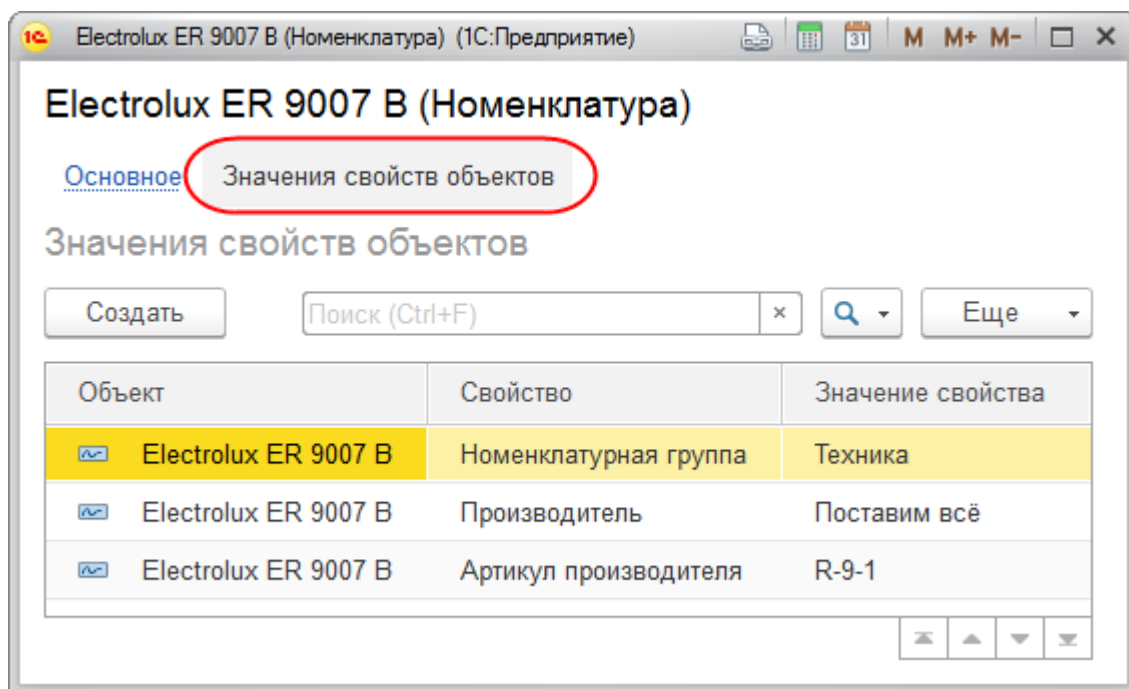


Рисунок 15 – Значения свойств номенклатуры

Настройка характеристик справочника «Номенклатура»

Чтобы свойства номенклатуры были доступны в отчетах на СКД как реквизиты, можно или воспользоваться расширением языка запросов для системы компоновки данных, или настроить характеристики справочника «Номенклатура» на уровне объекта метаданных. Подробнее эти способы рассматривались ранее в разделе **«30. Как следует организовать хранение значений произвольных свойств объектов в базе»**.

Второй способ более универсальный, поскольку настройки необходимо выполнить только один раз для справочника «Номенклатура», во всех отчетах на СКД эти настройки будут использоваться. Дорабатывать каждый отчет по-отдельности не потребуется. Поэтому для решения задачи настроим характеристики справочника «Номенклатура» на уровне объекта метаданных.

Для этого в конфигураторе в контекстном меню для объекта метаданных справочника «Номенклатура» выберем пункт Характеристики:

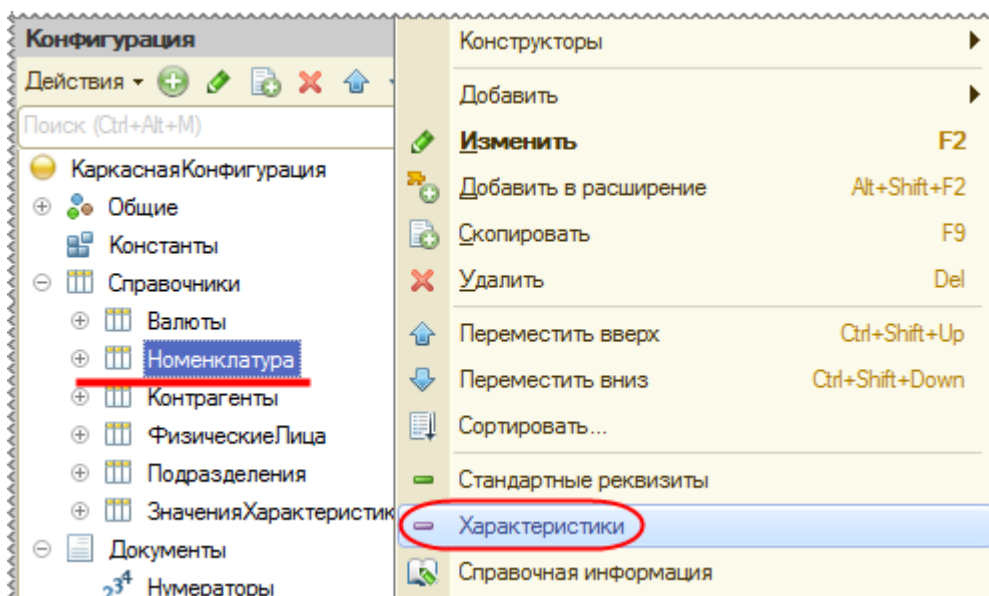


Рисунок 16 – Настройка характеристик на уровне объекта метаданных

В окне с настройками дополнительных характеристик для поля *Виды характеристик* выберем значение *ПланВидовХарактеристик.СвойстваОбъектов*. Для поля *Значения характеристик* выберем значение *РегистрСведений.ЗначенияСвойствОбъектов*. При этом часть полей платформа заполнит автоматически:

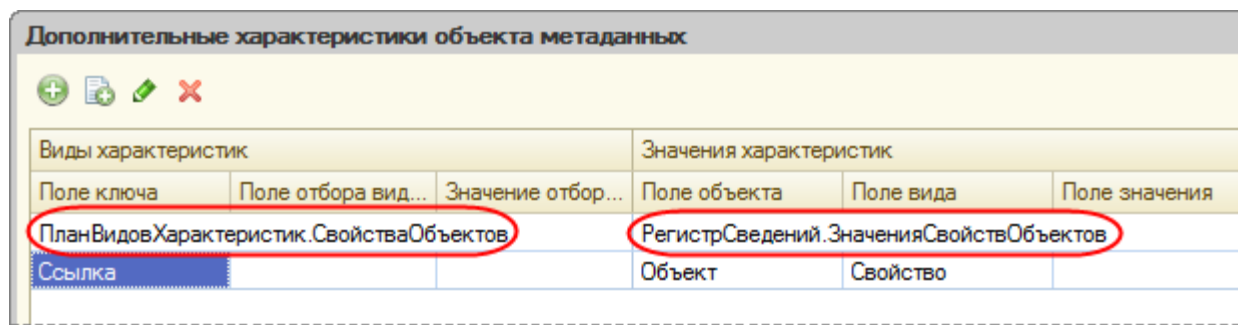


Рисунок 17 – Выбор объектов, где хранятся виды и значения характеристик

Остается только заполнить *Поле значения*, в котором следует выбрать ресурс *ЗначениеСвойства* регистра сведений:

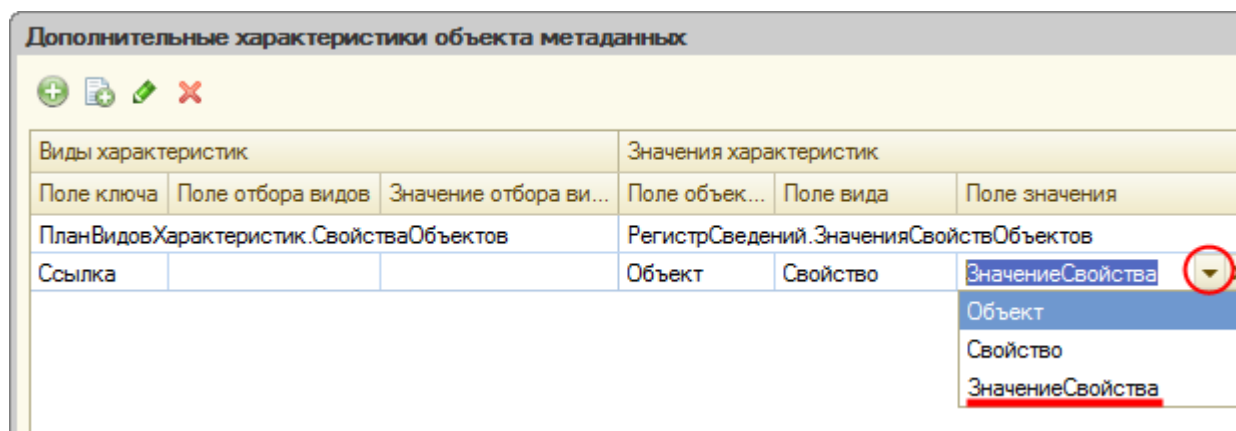


Рисунок 18 – Выбор поля, в котором хранятся значения характеристик

Левая часть таблицы используется системой для получения перечня характеристик, которые в отчетах на СКД будут доступны в качестве вложенных полей («реквизитов») номенклатуры. В нашем случае эти данные необходимо получать из созданного плана вида характеристик *СвойстваОбъектов*. Каждый элемент плана видов характеристик – это отдельное дополнительное свойство, «реквизит» номенклатуры.

Правая часть таблицы используется системой для получения из базы данных значений характеристик. В нашем случае для этого требуется обратиться к регистру сведений *ЗначенияСвойствОбъектов*. Ресурс *ЗначениеСвойства* этого регистра будет использован в качестве значения «реквизита» номенклатуры.

Разработка отчета с использованием СКД

Добавим в конфигурацию новый отчет *СвойстваНоменклатуры*. Создадим схему компоновки для отчета, для этого нажмем кнопку *Открыть схему компоновки данных*:

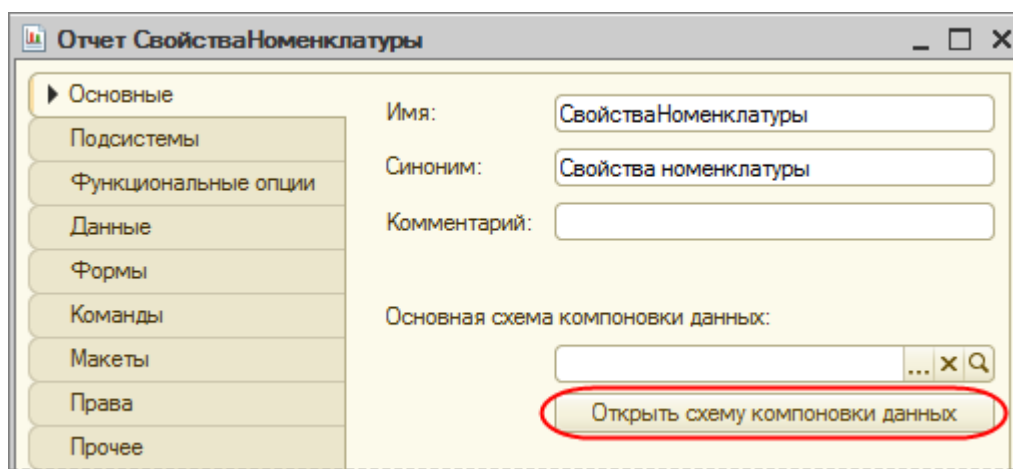


Рисунок 19 – Создание схемы компоновки для отчета

Данные для отчета будем получать из базы запросом, поэтому в схему компоновки добавим набор данных – запрос. Для построения запроса воспользуемся *Конструктором запроса*. Выберем несколько полей из справочника «Номенклатура». Главное, чтобы присутствовало поле *Ссылка*, поскольку через ссылку можно получить любые другие данные по номенклатуре:

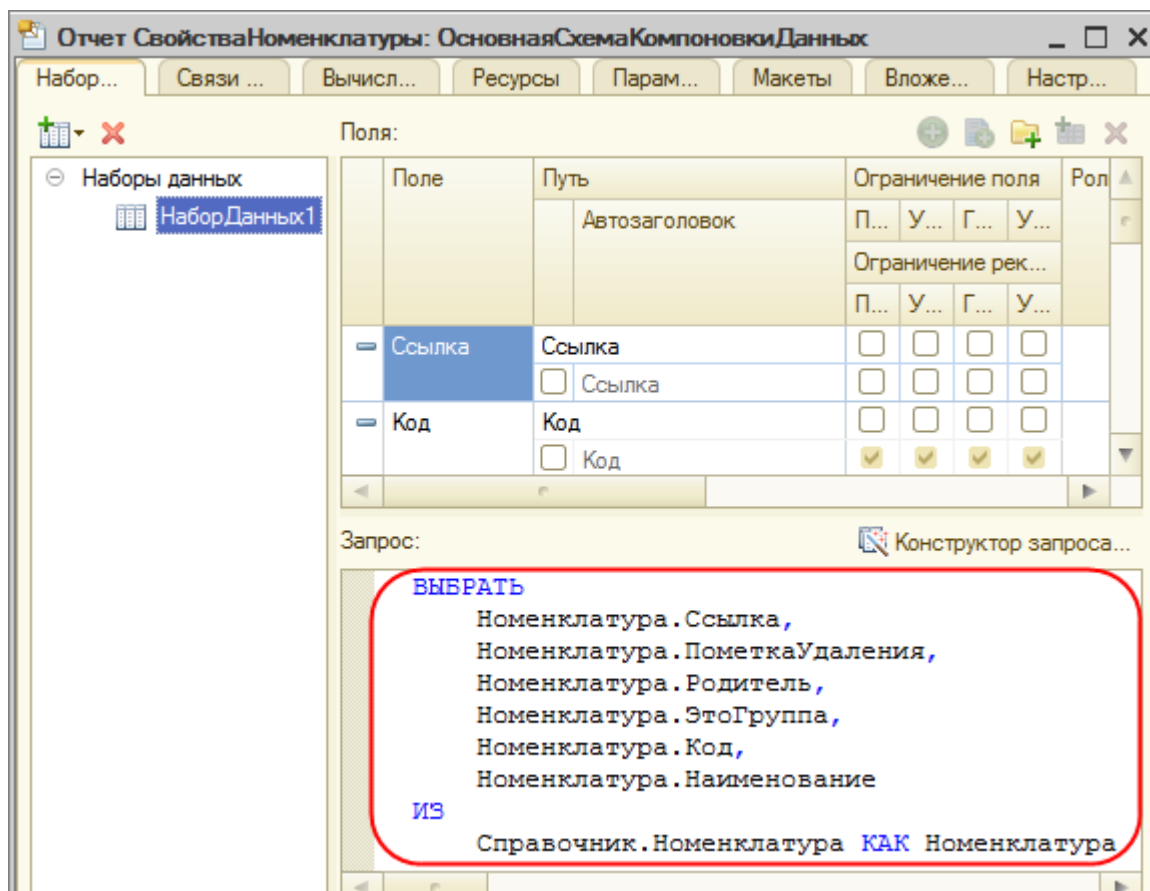


Рисунок 20 – Создание текста запроса набора данных

В схеме компоновки данных на закладке *Настройки* добавим в отчет группировку по полю *Ссылка*.

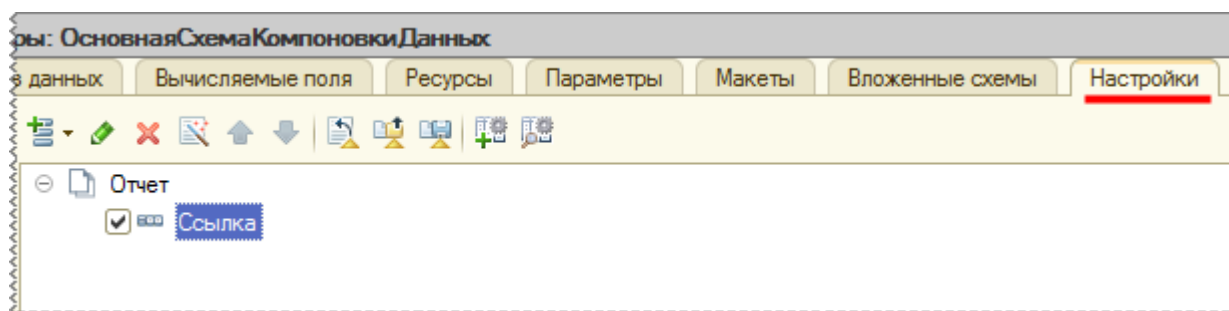


Рисунок 21 – Создание настроек компоновки

Важно. Дополнительные характеристики, созданные в режиме «1С:Предприятие», нельзя использовать при настройке схемы компоновки в конфигураторе, поскольку для получения списка характеристик необходимо выполнить запрос к базе данных, а в конфигураторе работа происходит не с конкретными данными, а только с их описанием.

Окончательную настройку варианта отчета выполним в пользовательском режиме. Для этого откроем созданный отчет и в меню *Еще* выберем команду *Изменить вариант*:

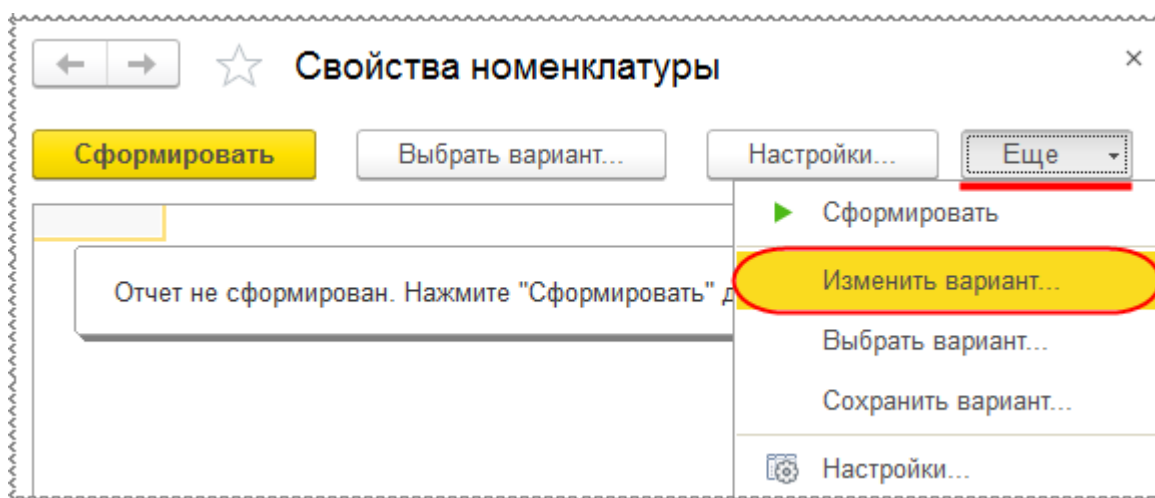


Рисунок 22 – Изменение варианта отчета в пользовательском режиме

Обратимся к свойствам группировки *Ссылка*. Для поля *Ссылка* доступны вложенные поля («реквизиты»):

- *Артикул производителя*
- *Номенклатурная группа*
- *Производитель.*

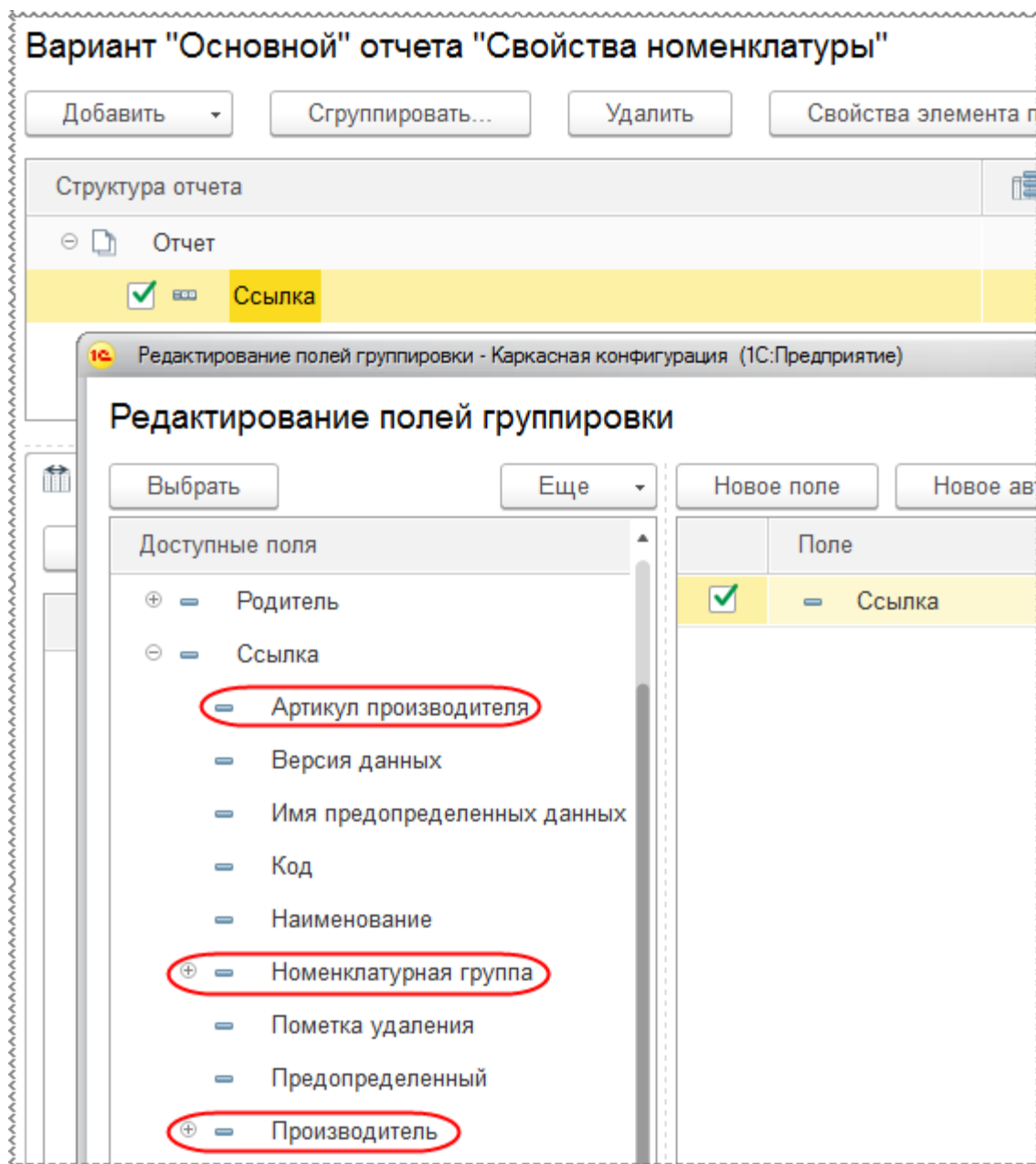


Рисунок 23 – Работа с характеристиками как с реквизитами номенклатуры

Это дополнительные свойства, добавленные в пользовательском режиме. Их можно использовать при настройке отчета так же, как и обычные реквизиты справочника.

Добавим указанные «реквизиты» в качестве полей группировки:

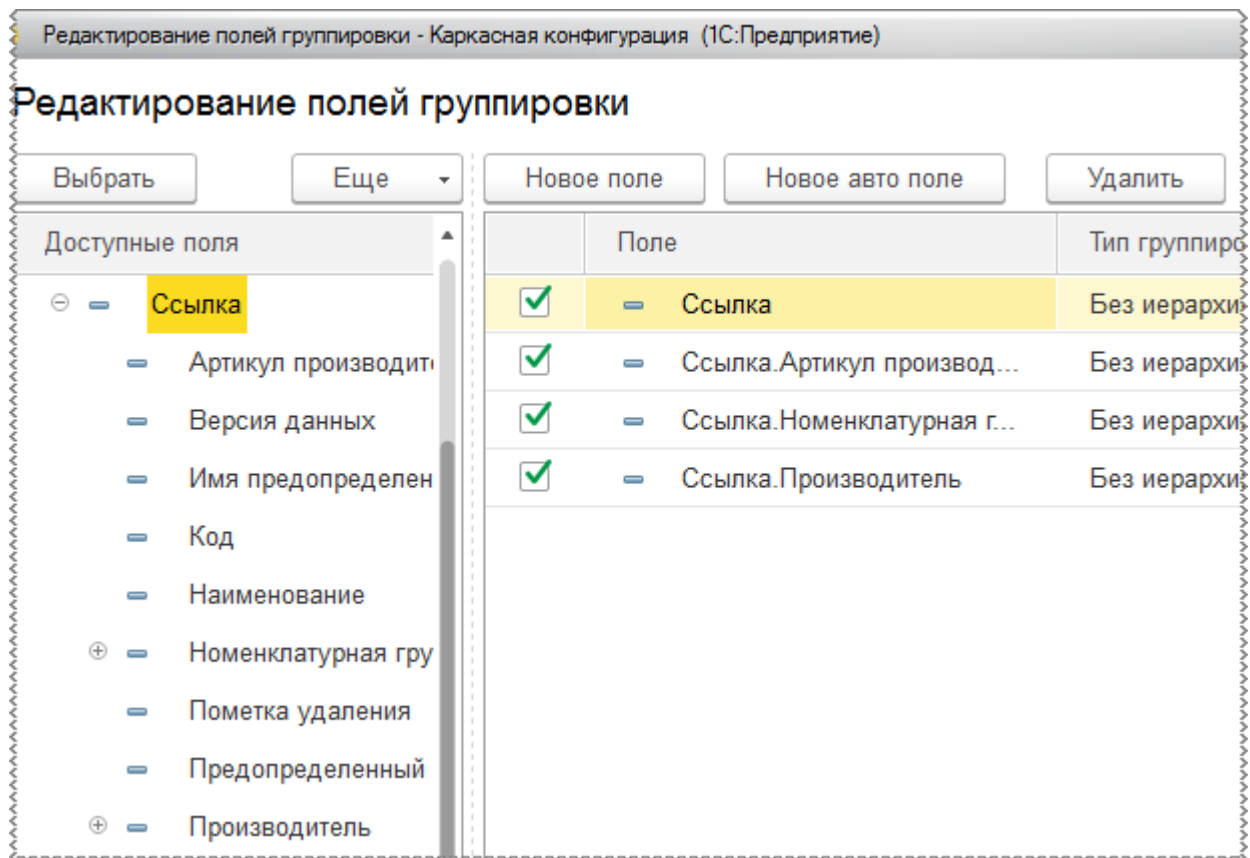


Рисунок 24 – Настройка группировки

Сохраним настройки и сформируем отчет:

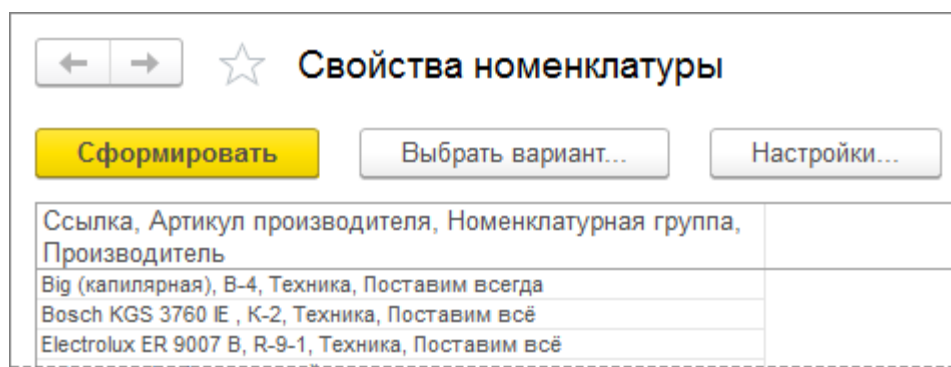


Рисунок 25 – Внешний вид отчета с характеристиками номенклатуры

В отчет выводятся все данные, требуемые по условию задачи. Необходимо поработать над внешним видом отчета, так как по условию требуется выводить поля отдельными колонками. Для этого еще раз обратимся к настройкам варианта отчета.

Добавленные характеристики в отчете представляются как вложенные поля («реквизиты») для поля Ссылка, поэтому на закладке *Дополнительные настройки* укажем для параметра *Расположение реквизитов* значение *Отдельно*.

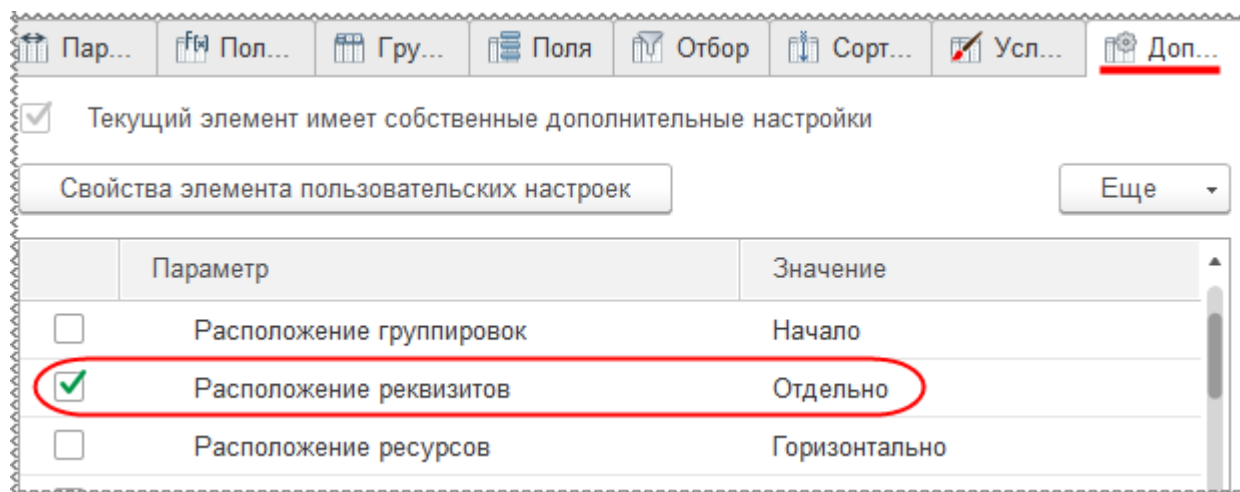


Рисунок 26 – Настройка расположения реквизитов в отчете

Данная настройка обеспечивает вывод дополнительных свойств отдельными колонками:

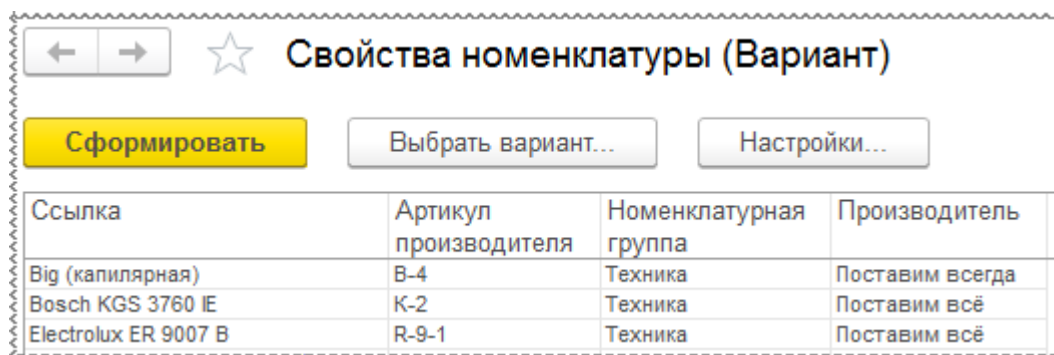


Рисунок 27 – Результирующий вид отчета

Чтобы в дальнейшем не приходилось перенастраивать отчет, сохраним полученный вариант отчета. Для этого в меню *Еще* выберем команду *Сохранить вариант*.

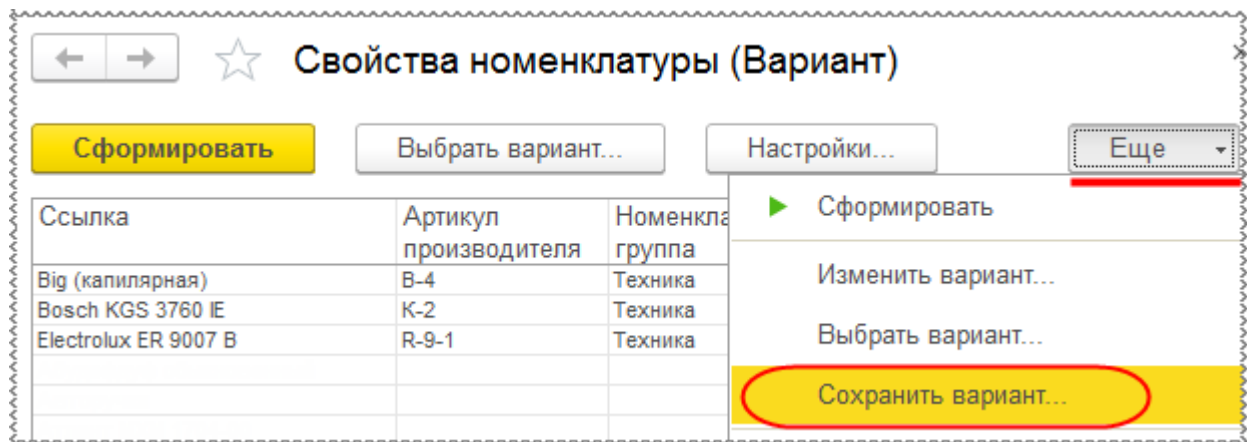


Рисунок 28 – Сохранение варианта отчета

Укажем имя для варианта отчета и нажмем кнопку *Сохранить*.

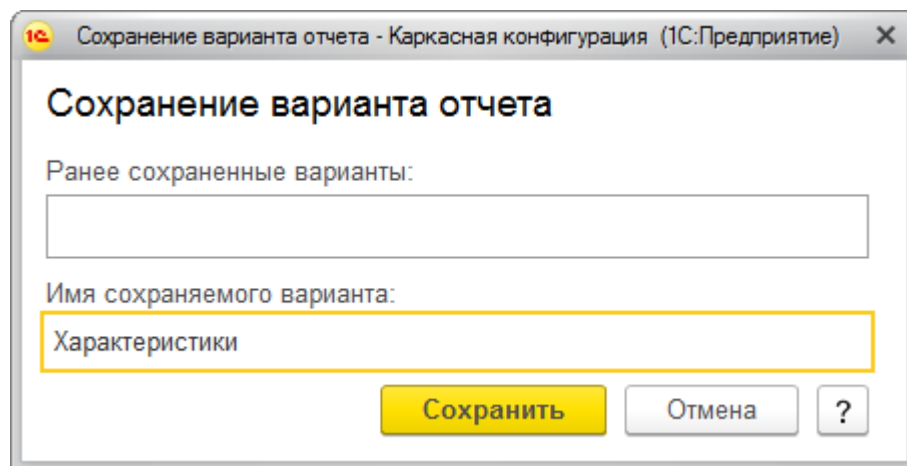


Рисунок 29 – Редактирование имени варианта отчета

Чтобы открыть сохраненный вариант, нужно нажать на кнопку *Выбрать вариант* на форме отчета.

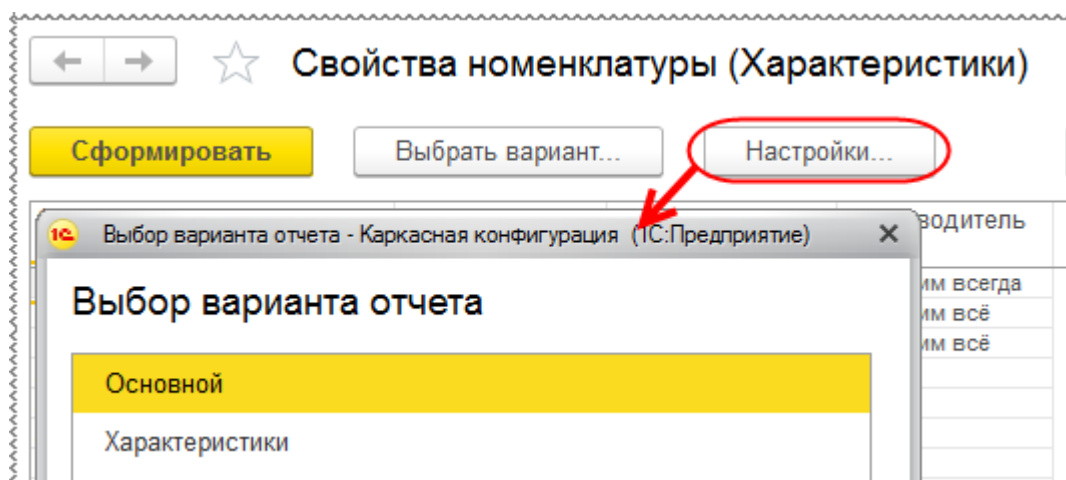


Рисунок 30 – Выбор сохраненного варианта отчета

Поставленная задача решена. При помощи плана видов характеристик обеспечено хранение произвольных свойств для номенклатуры. Создаваемые характеристики можно использовать в отчетах на СКД.

Подробно работа с системой компоновки данных рассмотрена в курсе «[Профессиональная разработка отчетов в 1С 8.3 на Системе Компоновки Данных \(СКД\)](#)».

34. Как на экзамене настроить командный интерфейс конфигурации

В аттестационной задаче есть требование – необходимо реализовать командный интерфейс конфигурации. В зависимости от того, задачи каких разделов присутствуют в экзаменационном билете, в панели разделов должны присутствовать следующие пункты:

- Оперативный учет
- Бухгалтерский учет
- Периодические расчеты
- Бизнес-процессы
- Сервисные механизмы.

При помощи панели навигации необходимо иметь доступ ко всем объектам данного раздела. Регистры также должны отображаться в панели навигации. Все объекты должны быть сгруппированы по видам: справочники отдельно, документы отдельно и т.д.

Примерный внешний вид конфигурации приведен на рисунке:

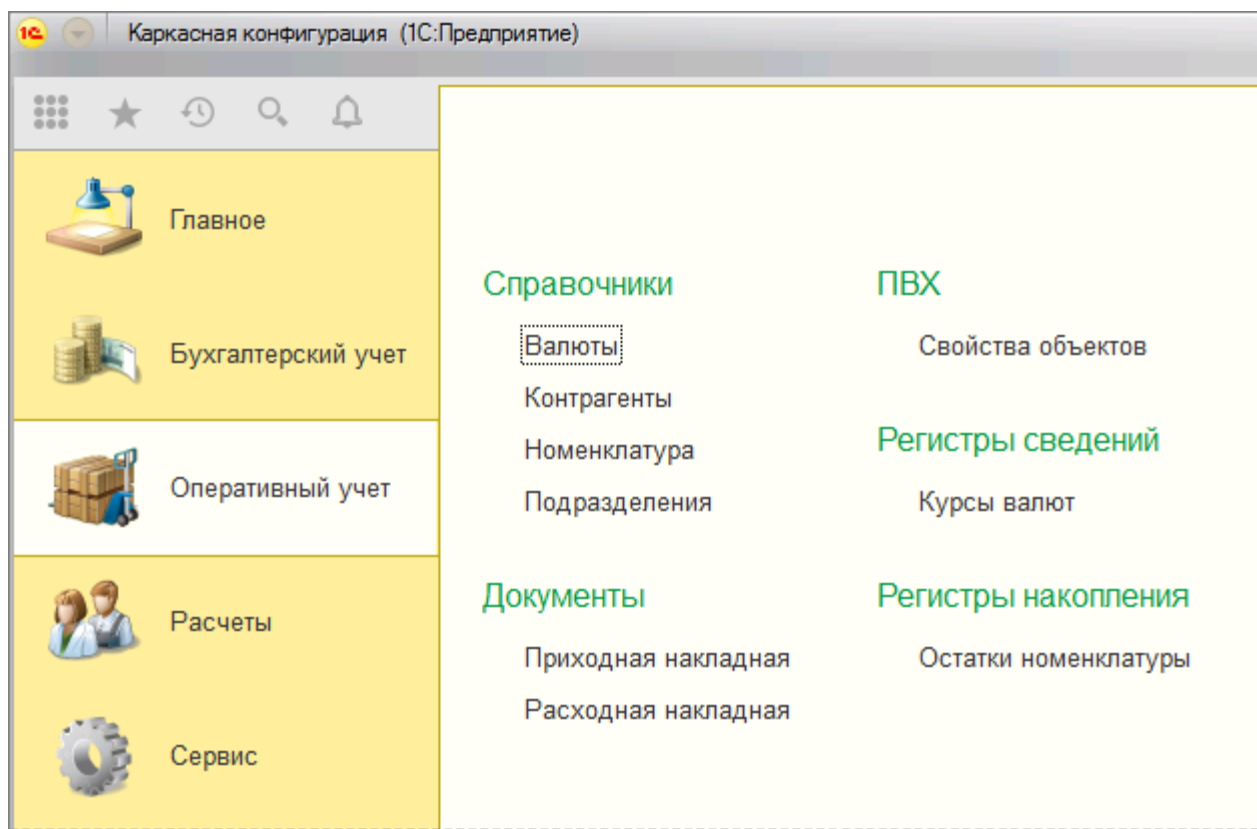


Рисунок 1 – Командный интерфейс конфигурации

Рассмотрим подробнее, как решить поставленную задачу.

Как при помощи подсистем организовать командный интерфейс?

Важно, что командный интерфейс конфигурации строится системой автоматически на базе объектов метаданных *Подсистема*. Подсистемы первого уровня отображаются в панели разделов. Поэтому первым делом необходимо создать в конфигурации следующие подсистемы:

- Оперативный учет
- Бухгалтерский учет
- Расчеты
- Бизнес-процессы
- Сервис.

Если в экзаменационной задаче нет какого-либо раздела (например, бизнес-процессов), соответствующую подсистему не создаем.

Для примера возьмем каркасную конфигурацию, в которой еще не было выполнено доработок. Создадим необходимые подсистемы:

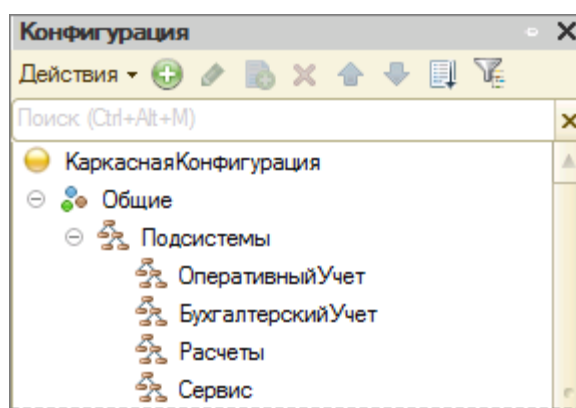


Рисунок 2 – Созданные подсистемы

Далее переходим к настройкам каждой отдельной подсистемы. Чтобы подсистема отображалась в командном интерфейсе, необходимо установить галочку *Включать в командный интерфейс*:

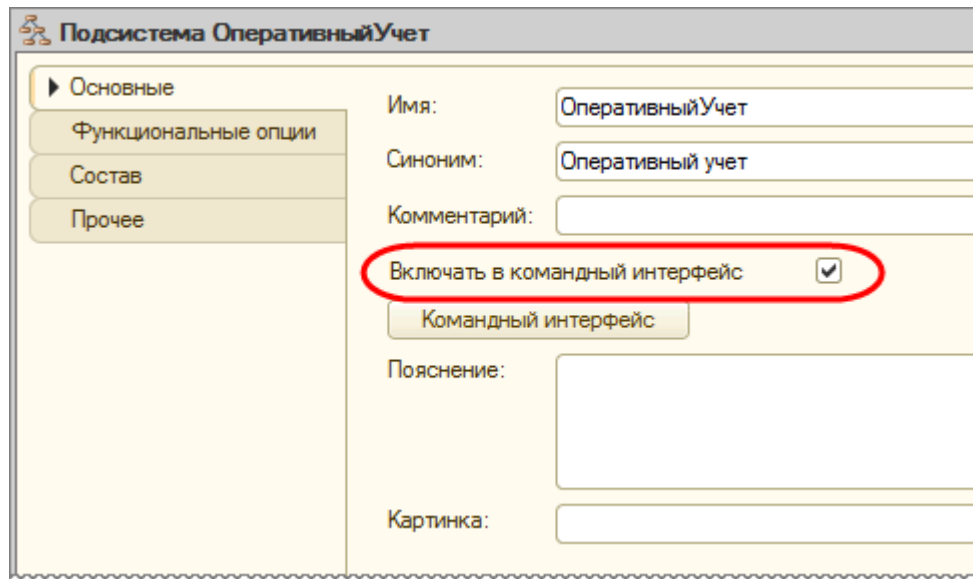


Рисунок 3 – Свойства подсистемы

В каркасной конфигурации присутствуют уже готовые общие картинки, рисовать их вручную на экзамене не требуется:

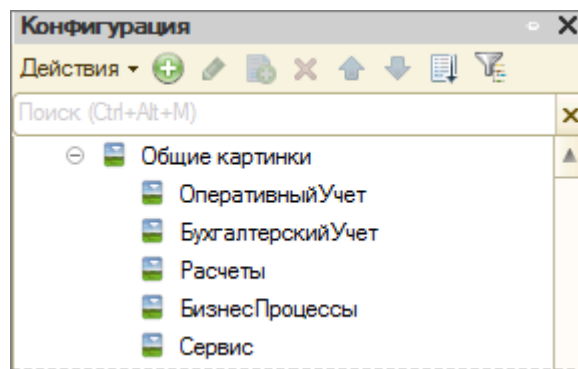


Рисунок 4 – Общие картинки в каркасной конфигурации

Эти общие картинки можно указать в свойствах подсистемы:

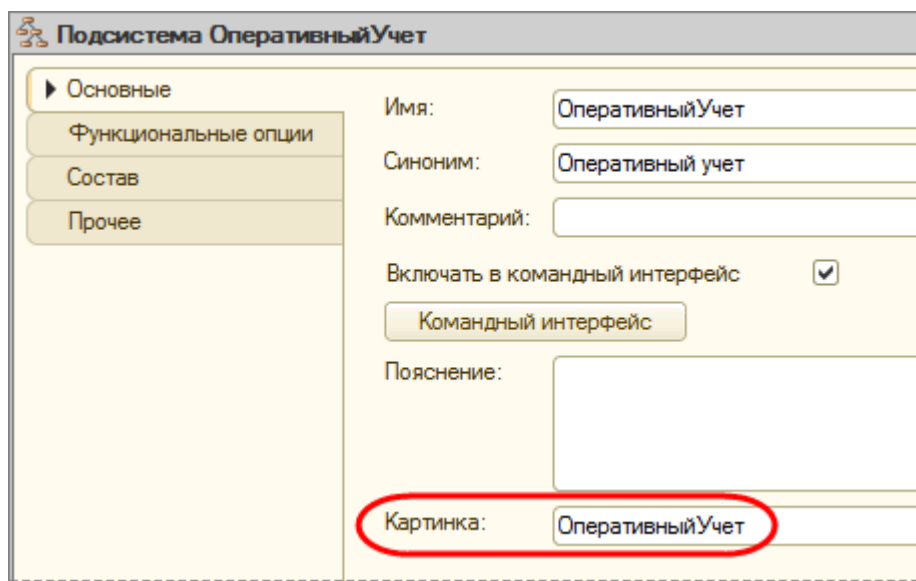


Рисунок 5 – Картинка подсистемы

При таких настройках в пользовательском режиме будет отображаться пиктограмма для подсистемы.

Далее необходимо указать, какие объекты метаданных будут входить в конкретную подсистему. Для этого на закладке *Состав* отмечаем галочками нужные объекты:

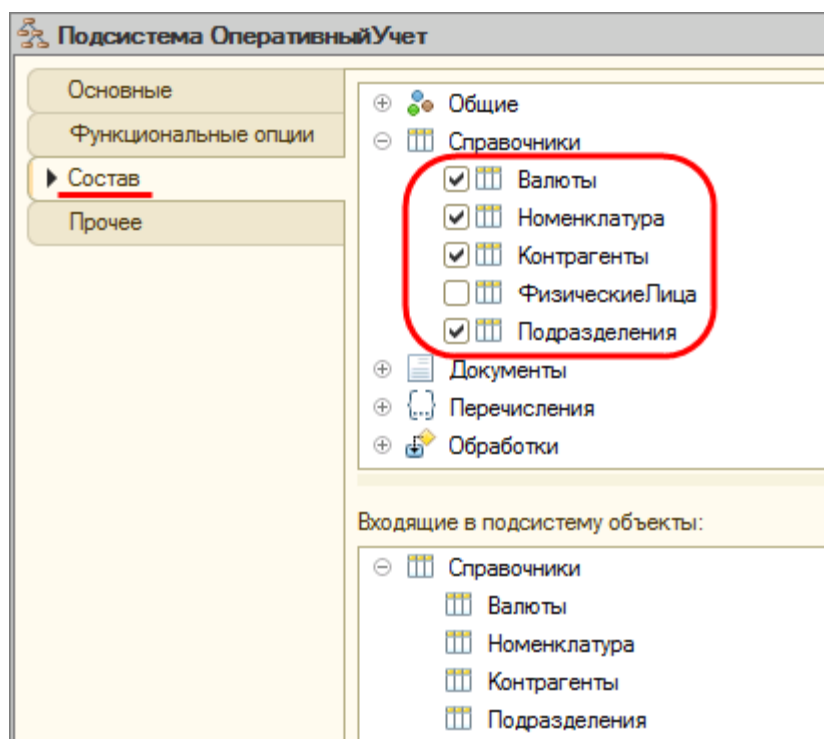


Рисунок 6 – Состав подсистемы

Настроим состав и расположение панелей, которые по умолчанию будут использоваться в клиентском приложении. Для этого в дереве объектов метаданных в контекстном меню выберем пункт *Открыть интерфейс клиентского приложения*:

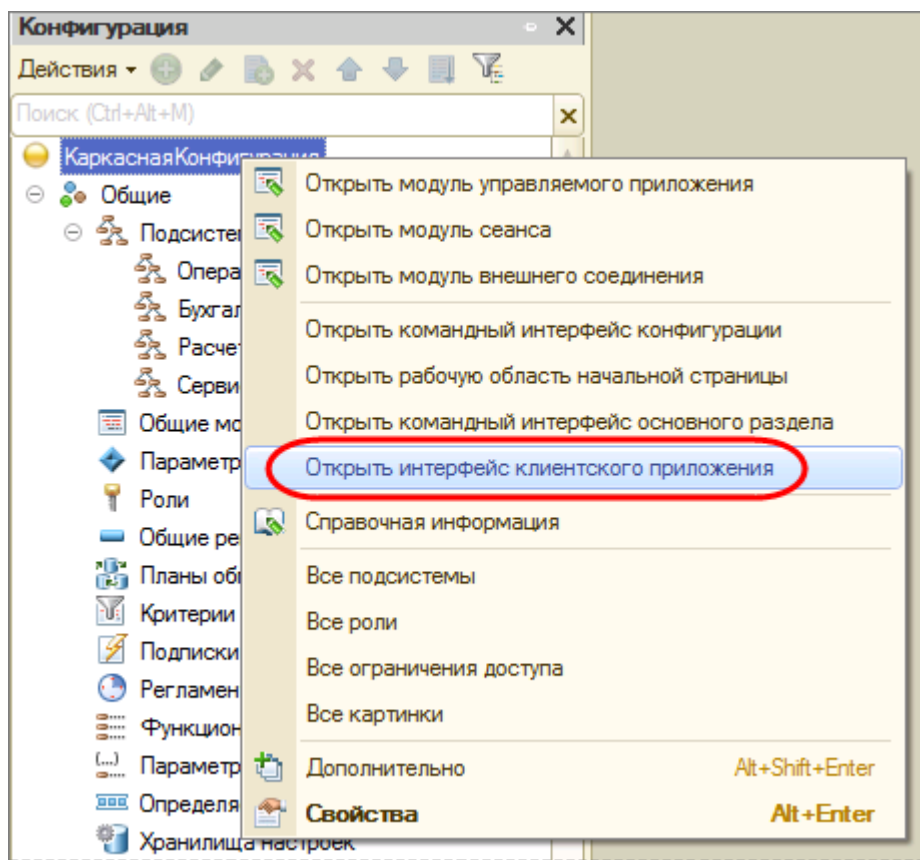


Рисунок 7 – Контекстное меню

В открывшемся окне настроим, какие панели необходимо отображать, а также в каких частях экрана они будут выводиться. Чтобы внешний вид командного интерфейса был похож на приведенный в постановке задачи, отобразим панель разделов в левой части экрана, а панель инструментов – в верхней части, остальные панели не используем:

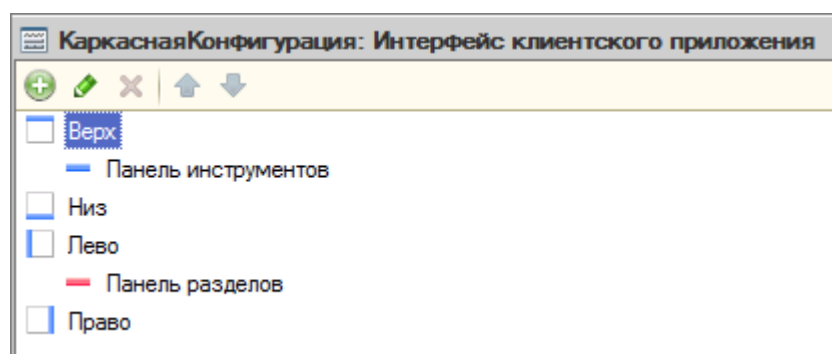


Рисунок 8 – Настройка интерфейса клиентского приложения

Важно, что указанные настройки – это значения по умолчанию, каждый пользователь в режиме «1С:Предприятие» может настроить собственное расположение панелей.

Теперь в пользовательском режиме конфигурация выглядит следующим образом:

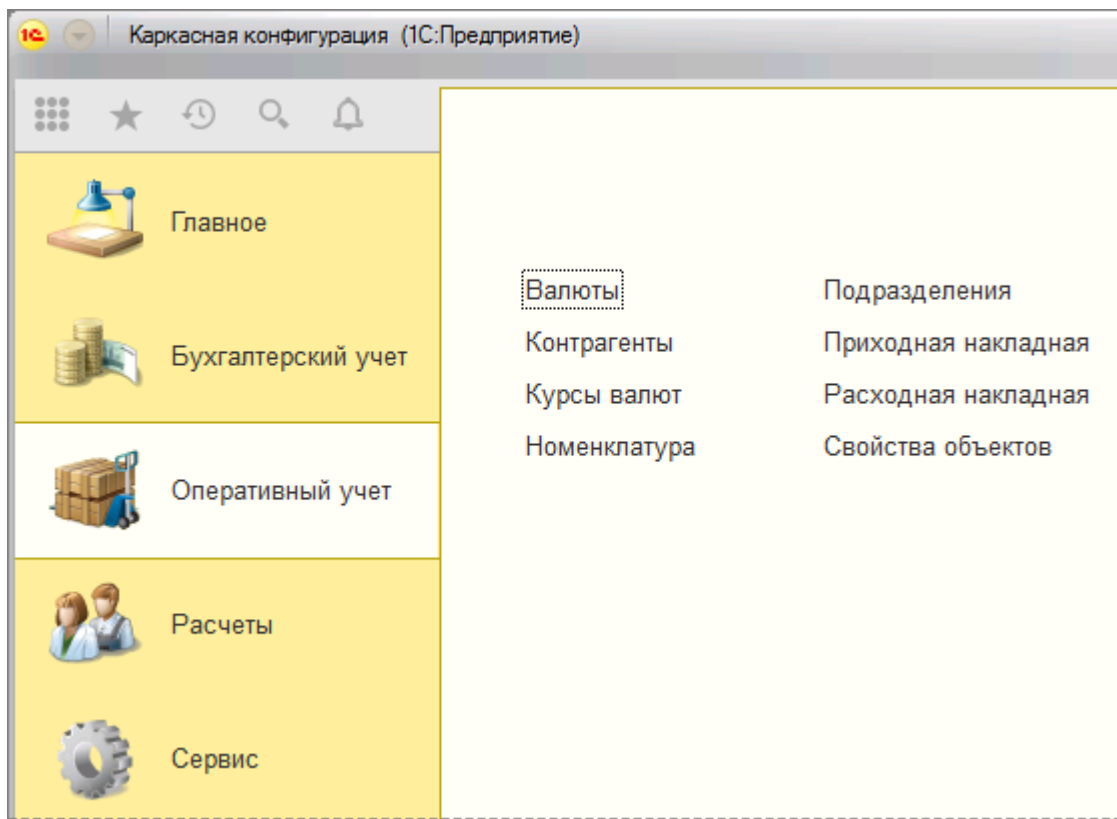


Рисунок 9 – Командный интерфейс конфигурации

В командном интерфейсе все объекты выводятся общим списком. Однако по условию задачи объекты необходимо группировать по видам. То есть отдельно должны выводиться справочники, отдельно документы, регистры и остальные объекты метаданных. Рассмотрим, как обеспечить такое поведение системы.

Как в интерфейсе сгруппировать объекты по видам?

В дереве объектов метаданных есть специальный объект, который называется *Группы команд*. Этот объект позволяет вывести команды в интерфейс в отдельных группах.

Создадим группы команд по видам объектов, которые требуется отобразить согласно постановке задачи (Справочники, Документы, Регистры сведений и т.д.):

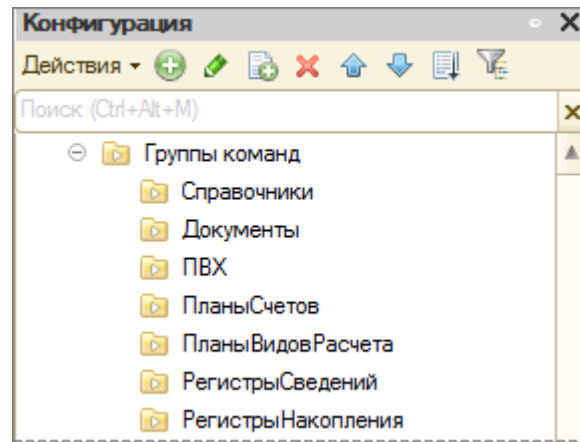


Рисунок 10 – Группы команд

Обратимся к окну свойств для группы команд. Чтобы группы размещались в панели навигации, необходимо установить свойство *Категория* в значение *Панель навигации*:

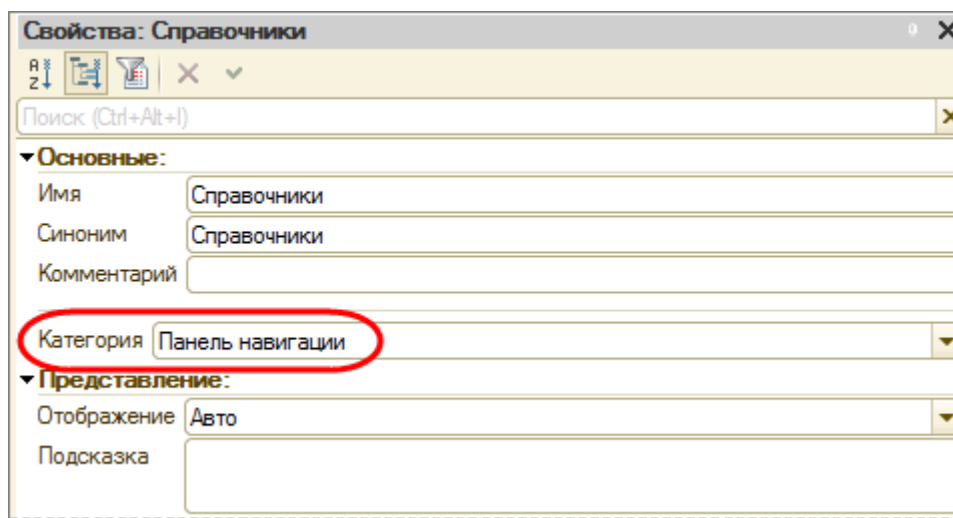


Рисунок 11 – Свойства группы команд

Далее необходимо настроить, какие команды будут отображаться в той или иной группе. Для этого в карточке подсистемы необходимо нажать кнопку *Командный интерфейс*:

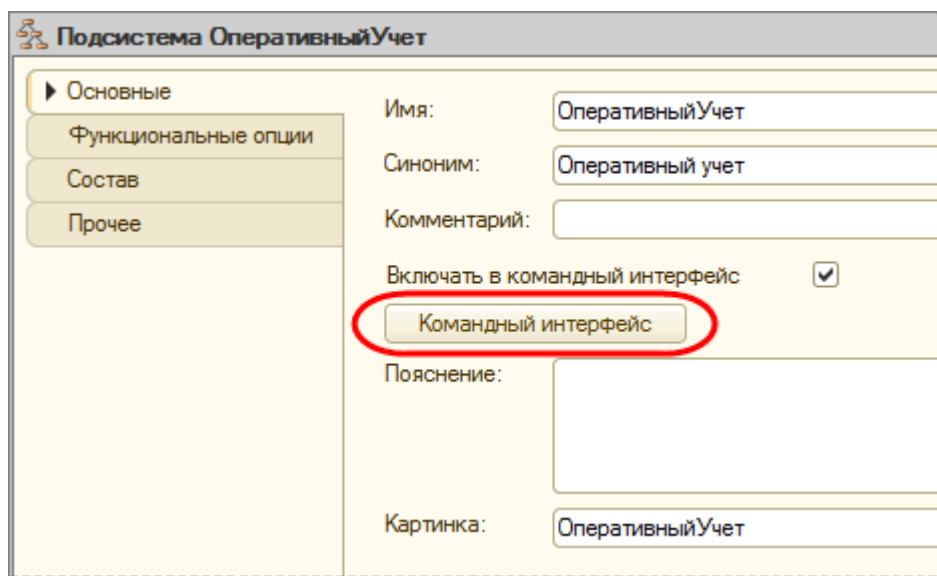


Рисунок 12 – Настройка командного интерфейса

В открывшемся окне необходимо отнести объекты, входящие в подсистему, в соответствующую группу команд. Например, справочник *Валюты* перенести в группу команд *Справочники*, а документ *Приходная накладная* – в группу *Документы* и т.д. Для этого удобно воспользоваться перетаскиванием при помощи мыши:

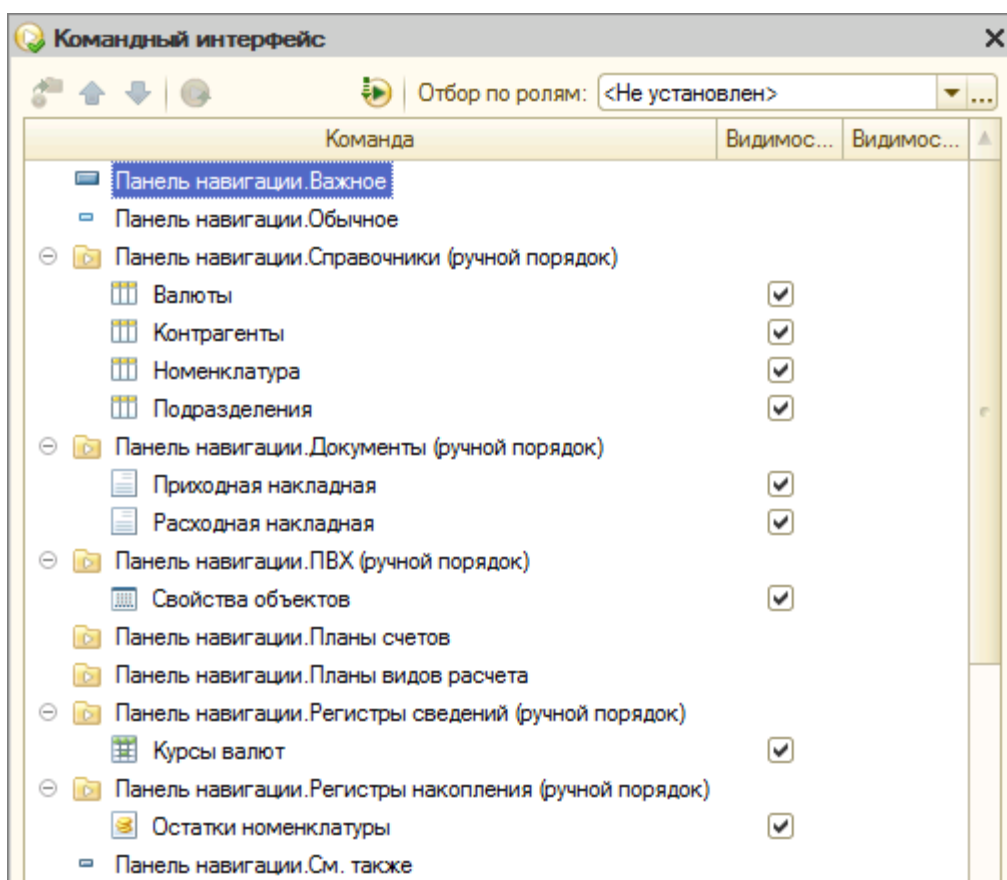


Рисунок 13 – Редактор командного интерфейса

Также необходимо для всех объектов установить галочку в колонке *Видимость*, чтобы эти объекты отображались в командном интерфейсе.

При запуске в пользовательском режиме приложение выглядит так, как требуется в постановке задачи:

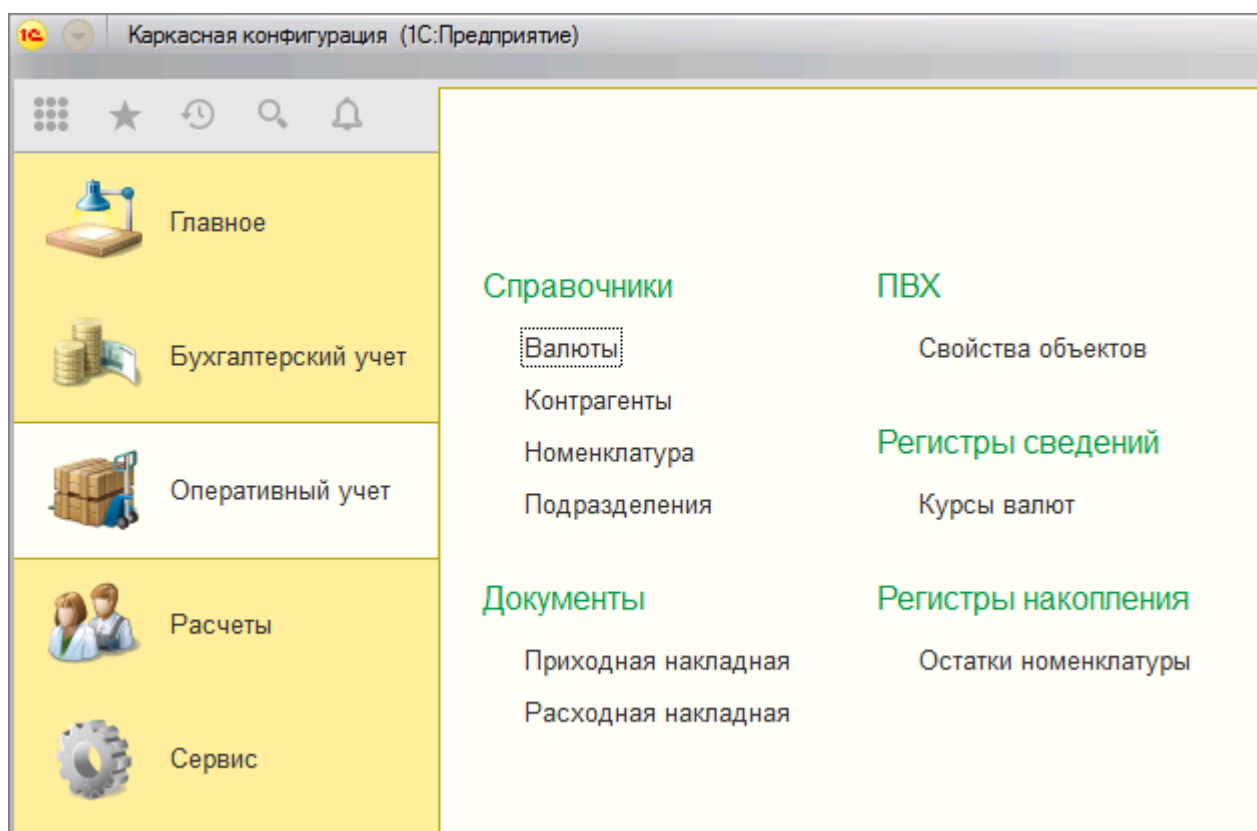


Рисунок 14 – Командный интерфейс конфигурации

Альтернативный вариант разделения объектов командного интерфейса на группы – использование подчиненных подсистем. Для этого в подсистеме *ОперативныйУчет* потребуется создать подчиненные подсистемы *ОперативныйУчетСправочники*, *ОперативныйУчетДокументы* и т.д.

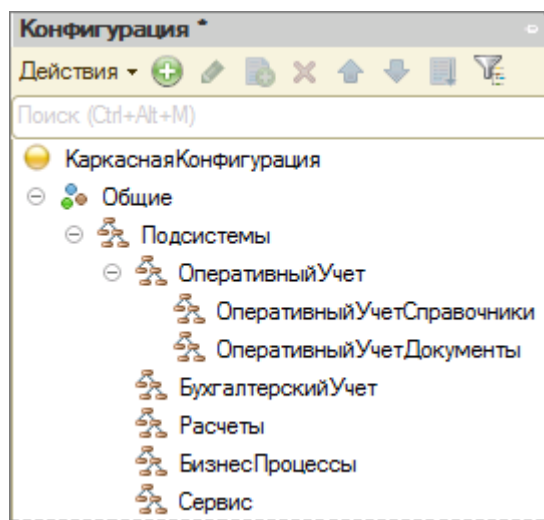


Рисунок 15 – Подчиненные подсистемы

Аналогичные подчиненные подсистемы нужно создать и для бухгалтерского учета, расчетов и т.д. Получается дублирование информации, повторное выполнение одних и тех же действий. При использовании групп команд эти объекты создаются только один раз, поэтому вместо подчиненных подсистем при решении задачи целесообразно использовать именно группы команд.

При решении экзаменационной задачи важно для всех объектов метаданных указывать принадлежность к подсистемам – это необходимо для формирования командного интерфейса:

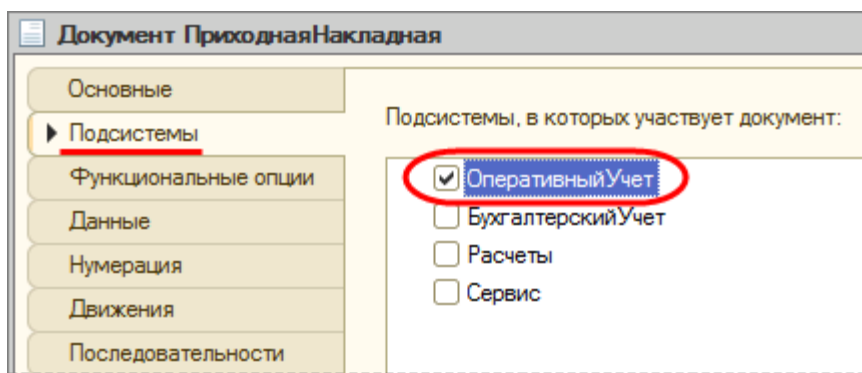


Рисунок 16 – Настройка подсистем для объектов метаданных

На экзамене при добавлении в конфигурацию нового объекта рекомендуется сразу указывать, к какой подсистеме он будет относиться. Таким образом можно сэкономить немного времени, поскольку не придется отдельно настраивать состав подсистем, останется только настроить принадлежность объектов к группам команд.

К подсистеме «Сервис» можно отнести служебные обработки *КонсольЗапросов* и *ЗаполнениеГрафика*, которые уже присутствуют в каркасной конфигурации. Также в эту подсистему можно включить обработки, которые требуется разработать при решении экзаменационной задачи:

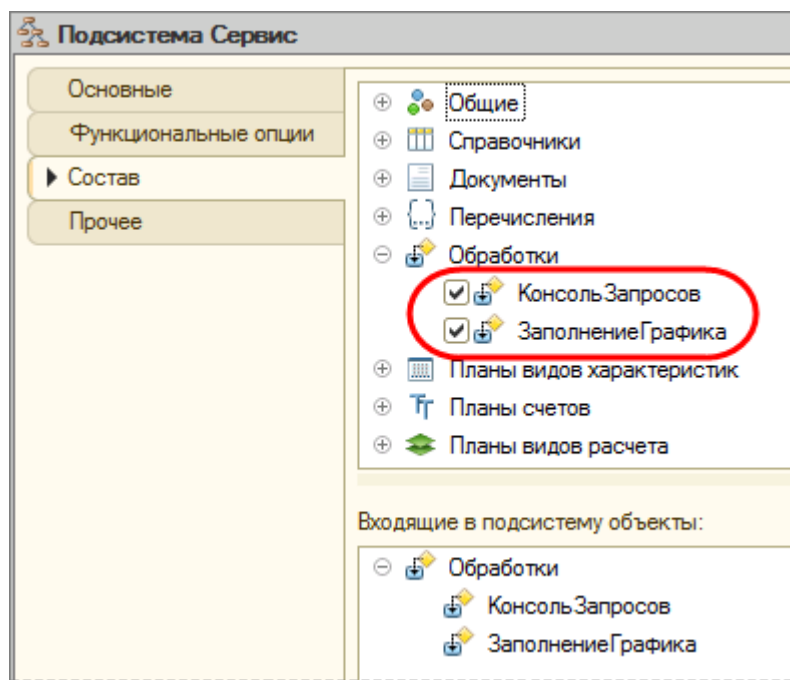


Рисунок 17 – Состав подсистемы «Сервис»

Итак, в данном разделе рассмотрено, как следует выполнять настройку командного интерфейса конфигурации на аттестации.

Настройки командного интерфейса, приведенные в данном курсе, являются достаточно простыми по сравнению с интерфейсами в реальных конфигурациях. При сдаче экзамена такой организации командного интерфейса вполне достаточно. А если нужно более детально разобраться с темой командных интерфейсов в управляемых приложениях, можно обратиться к курсам [«1С:Программист - Быстрый старт в профессию!»](#) и [«Профессиональная разработка интерфейсов и форм в 1С:Предприятие 8.3»](#).