

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное  
учреждение высшего образования  
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра корпоративных информационных систем

## **ПЛАТФОРМА 1С**

### **Методические указания**

к выполнению лабораторной работы № 4  
по дисциплине

для студентов очной формы обучения  
по направлениям подготовки

09.03.01 «Информатика и вычислительная техника»,

09.03.02 «Информационные системы и технологии»,

09.03.03 «Прикладная информатика»,

09.03.04 «Программная инженерия»

В четырех частях

Часть 4

Севастополь  
СевГУ  
2023

**УДК 004.45:65.011.56(076.5)**

**ББК 32.973-018.2я73**

**ПЗ76**

**Р е ц е н з е н т :**

**М. В. Заморёнов** - доцент кафедры «Информационные технологии  
и компьютерные системы»

*Составители:* В. И. Шевченко, Е. В. Виноградов, В. С. Кудашев,  
Н. О. Сергеев

**ПЗ76 Платформа 1С : В 4-х частях. Часть 4 : методические указания для**  
выполнения лабораторной работы № 4 по дисциплине для студентов  
очной формы обучения по направлениям подготовки 09.03.01 «Информатика  
и вычислительная техника», 09.03.02 «Информационные системы и  
технологии», 09.03.03 «Прикладная информатика», 09.03.04 «Программная  
инженерия» / Севастопольский государственный университет, Институт  
информационных технологий, кафедра корпоративных информационных  
систем ; сост.: В. И. Шевченко, Е. В. Виноградов, В. С. Кудашев,  
Н. О. Сергеев. – Севастополь : СевГУ, 2023. – 37 с. – Текст : электронный.

Целью методических указаний является оказание помощи студентам в  
выполнении лабораторных работ по дисциплине. Методические указания содержат  
общие положения и требования к отчету, основные теоретические сведения, описание  
вариантов индивидуальных заданий.

УДК 004.45:65.011.56(076.5)

ББК 32.973-018.2я73

Методические указания рассмотрены и утверждены на заседании базовой  
кафедры «Корпоративные информационные системы», протокол № 10 от  
26 мая 2022 г.

© Шевченко В. И., Виноградов Е. В.,  
Кудашев В. С., Сергеев Н. О., сост., 2023  
© ФГАОУ ВО «Севастопольский  
государственный университет», 2023

## СОДЕРЖАНИЕ

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| СОДЕРЖАНИЕ .....                                                           | 3  |
| ВВЕДЕНИЕ .....                                                             | 4  |
| ЛАБОРАТОРНАЯ РАБОТА № 4. ПРОСТАЯ УЧЕТНАЯ КОНФИГУРАЦИЯ                      | 5  |
| 1. ОСНОВНЫЕ ПОЛОЖЕНИЯ .....                                                | 5  |
| 2. РЕГИСТРЫ НАКОПЛЕНИЯ.....                                                | 5  |
| 3. ОСТАТОЧНЫЕ И ОБОРОТНЫЕ РЕГИСТРЫ НАКОПЛЕНИЯ И ИХ<br>ПРЕДНАЗНАЧЕНИЕ ..... | 7  |
| 4. ПРОГРАММНЫЙ КОД ДЛЯ РАБОТЫ С РЕГИСТРОМ НАКОПЛЕНИЯ                       | 15 |
| 5. ПРИМЕР ЗАДАЧИ ПРЕДПРИЯТИЯ ПО УЧЕТУ ОБОРОТНЫХ<br>ПОКАЗАТЕЛЕЙ .....       | 17 |
| 6. ПРИМЕР ЗАДАЧИ ПРЕДПРИЯТИЯ ПО УЧЕТУ ОСТАТОЧНЫХ<br>ПОКАЗАТЕЛЕЙ .....      | 24 |
| ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ .....                                       | 34 |

## **ВВЕДЕНИЕ**

Технологическая платформа «1С:Предприятие» обеспечивает поддержку работы прикладных решений с различными операционными системами и СУБД, в том числе в среде открытого программного обеспечения и на мобильных устройствах. Открытость решений, возможность их динамичного развития, высокая функциональность и гибкость, возможность применения программ единой системы как в небольших предприятиях, так и в крупных организациях федерального масштаба обеспечивают высокую популярность решений на платформе «1С:Предприятия»: их использование широко распространено в России, Казахстане, Беларуси, они успешно применяются организациями многих других стран. Это создаёт большой рынок для программистов, владеющих языком и объектной моделью «1С:Предприятия».

Платформа «1С:Предприятие» является не универсальным, а специализированным, предметно-ориентированным средством разработки, предназначенным на решение задач автоматизации бизнеса. Одно из основных преимуществ этой предметно-ориентированной среды разработки – построение системы на основе технологической модели работы приложения, метаданных и прикладной модели работы приложения, что позволяет существенно упростить и ускорить разработку.

## ЛАБОРАТОРНАЯ РАБОТА № 4. ПРОСТАЯ УЧЕТНАЯ КОНФИГУРАЦИЯ

**Цель работы:** научиться применять специальные механизмы платформы 1С для ведения учета по остаточным и оборотным показателям, а также получить опыт разработки небольшой учетной конфигурации.

**Используемое программное обеспечение:** учебная версия платформы «1С:Предприятие».

### Теоретические сведения

#### 1. Основные положения

Один из главных моментов разработки любой конфигурации - создание механизма учета накопления данных.

В системе 1С: Предприятие есть несколько объектов конфигурации, которые позволяют создавать в базе данных структуры, назначенные для накопления информации в удобном для последующего анализа виде. Использование таких «хранилищ» данных дает нам, с одной стороны, накапливать в них данные, поставляемые различными документами (или другими объектами базы данных), а с другой стороны, легко создавать нужные нам отчеты или использовать эти данные в алгоритмах работы конфигурации.

В конфигурации существует несколько объектов, называемых регистрами для описания подобных «хранилищ». Объект конфигурации Регистр накопления предназначен для описания структуры накопления данных. На основе объекта конфигурации Регистр накопления платформа создает в базе данных таблицы, в которых будут накапливаться данные, «поставляемые» различными объектами базы данных.

Эти данные будут храниться в таблицах в виде отдельных записей, каждая из которых имеет одинаковую, заданную в конфигураторе структуру.

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

Отличительной особенностью регистра накопления является то, что он не предназначен для интерактивного редактирования пользователем.

#### 2. Регистры накопления

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

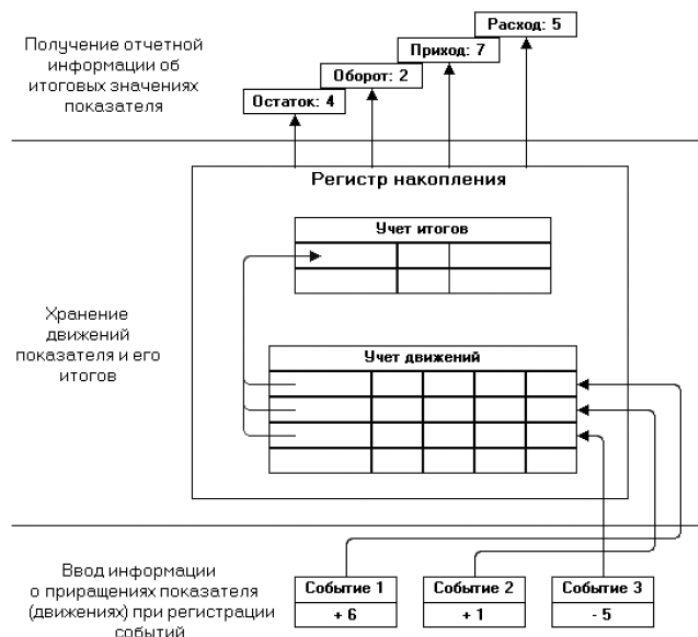


Рисунок 2.1 – Структура хранения данных регистра накопления

Данные каждого регистра накопления хранятся в двух таблицах базы данных:

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

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

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

Информация о движениях показателей (они могут быть как положительными, так и отрицательными) вносится в регистр накопления посредством наборов записей регистра, подчиненных документу-регистратору. Этим обеспечивается обоснованность регистрации изменений показателей – регистрацией событий, приводящих к этим изменениям.

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

Преимущества регистра накоплений:

- позволяет получить нужные итоги на любой момент времени или документ;
- все необходимые изменения в итогах, связанные с отменой проведения документа, изменением проведенного документа и удалением проведенного документа рассчитываются системой автоматически (если надо – можно прописать вручную);

- обеспечивает контроль уникальности записей, хранящихся в регистре накопления.

Благодаря этому в регистре накоплений не может находиться двух записей, относящихся к одной и той же строке одного и того же документа.

## Структура регистра накопления

В конфигураторе регистры расположены в соответствующих ветках.

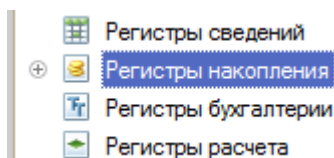


Рисунок 2.2 – Регистры накопления в дереве метаданных конфигурации

Закладка Основные:

- 1) Название регистра 1С в конфигураторе (Имя) и в режиме «1С:Предприятие» (Синоним).
- 2) Вид регистра 1С:
  - для регистра накопления 1С – остатки или обороты;
  - для регистра сведений 1С – периодичность.

## 3. Остаточные и оборотные регистры накопления и их предназначение

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

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

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

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

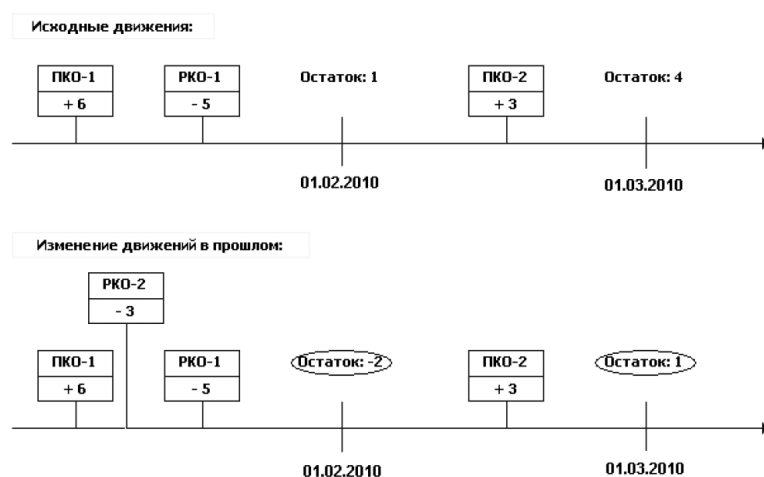


Рисунок 3.1 – Схема получения остатков

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

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

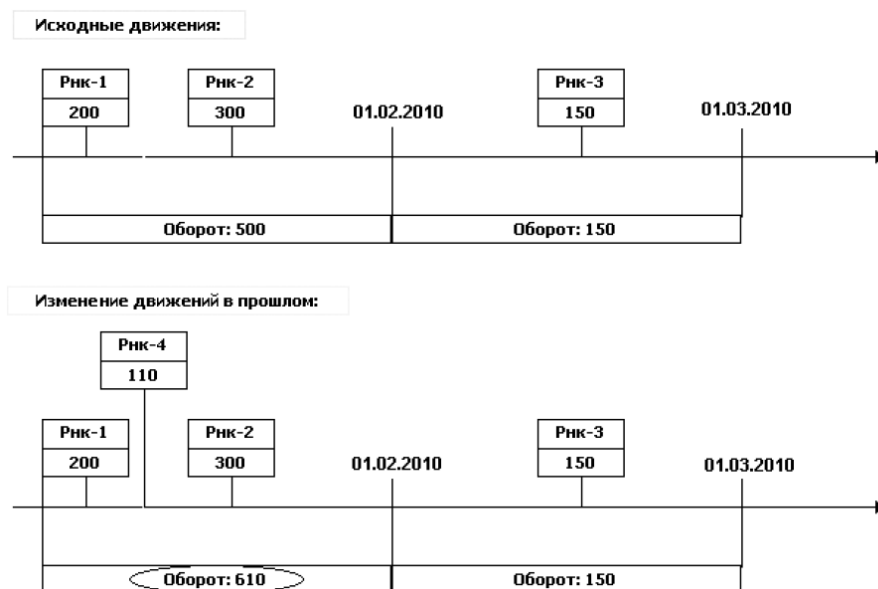


Рисунок 3.2 – Схема получения оборотов

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

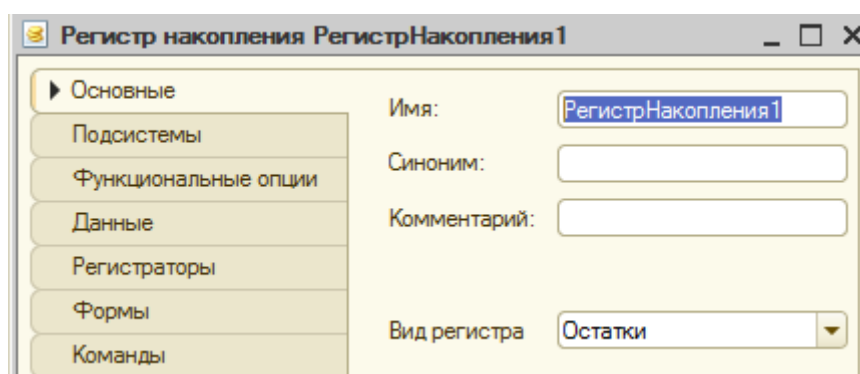


Рисунок 3.3 – Создание нового регистра накопления

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

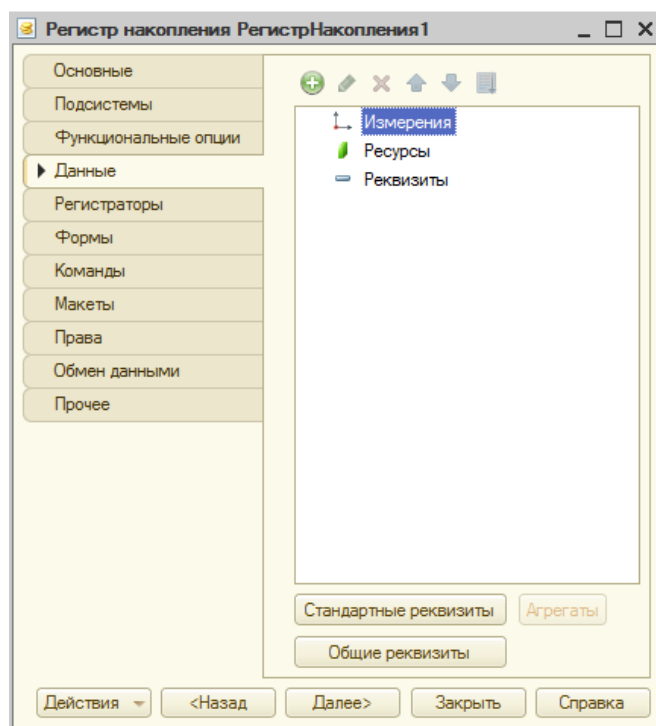


Рисунок 3.4 – Измерения, ресурсы и реквизиты регистра накопления

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

В регистрах накопления информационные разрезы данных реализуются с помощью измерений.

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

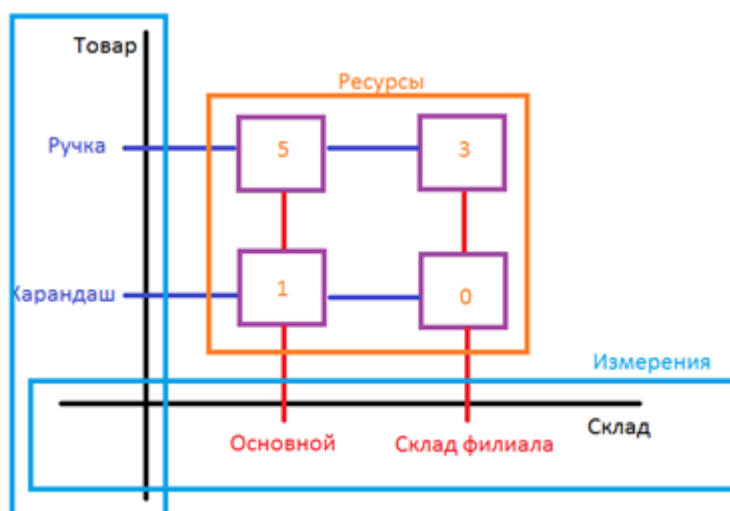


Рисунок 3.5 – Графическая интерпретация регистра накопления

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

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

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

Данные этой графической двумерной модели можно представить в виде табличной модели (табл. 3.1).

Таблица 3.1 – Табличное представление данных регистров

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

*Реквизиты* – дополнительные характеристики движений, то есть первичных записей регистра. Реквизиты не влияют на итоги, хранимые в регистре. На вышеприведенной схеме, отражающей итоги, нет места реквизитам. Однако реквизиты могут быть использованы при решении задач анализа выполненных движений. То есть таких задач, в которых работа идет только с приращениями показателей, безотносительно их итоговых значений.

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

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

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

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

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

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

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

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

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

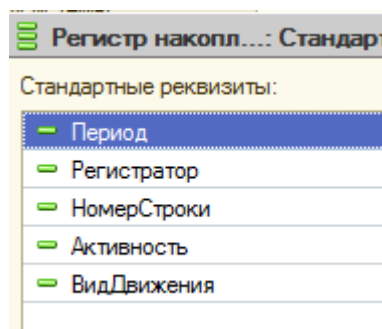


Рисунок 3.6 – Стандартные реквизиты регистра накопления

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

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

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

Таблица движений регистра хранит данные о выполненных по регистру движениях.

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

Рисунок 3.7 – Таблица движений регистра остатков

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

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

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

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

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

|                                                      |                        | Измерение    |         | Ресурс     |       |
|------------------------------------------------------|------------------------|--------------|---------|------------|-------|
|                                                      | Период                 | Номенклатура | Склад   | Количество | Сумма |
| Итоги апреля 2010 года                               | 01.05.2010<br>00:00:00 | Пульт VH     | Главный | 1          | 10    |
| Итоги мая 2010 года (граница<br>рассчитанных итогов) | 01.06.2010<br>00:00:00 | Пульт VH     | Главный | 2          | 20    |
|                                                      | 01.06.2010<br>00:00:00 | Пульт PW     | Главный | 7          | 490   |
|                                                      | 01.06.2010<br>00:00:00 | Пульт PW     | Фили-2  | 1          | 17    |
| Текущие итоги                                        | 01.11.3999<br>00:00:00 | Пульт VH     | Фили-2  | 1          | 18    |
|                                                      | 01.11.3999<br>00:00:00 | Пульт VH     | Главный | 2          | 20    |
|                                                      | 01.11.3999<br>00:00:00 | Пульт PW     | Главный | 7          | 490   |
|                                                      | 01.11.3999<br>00:00:00 | Пульт PW     | Фили-2  | 1          | 17    |

Рисунок 3.8 – Таблица движений регистра остатков с разделенными итогами

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

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

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

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

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

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

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

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

| Измерение              |                                       |              |            |              |                     | Ресурс     |       |
|------------------------|---------------------------------------|--------------|------------|--------------|---------------------|------------|-------|
| Период                 | Регистратор                           | Номер строки | Активность | Номенклатура | Контрагент          | Количество | Сумма |
| 27.05.2010<br>12:30:45 | Реализация № 1<br>27.05.2010 12:30:45 | 1            | Истина     | Пульт VH     | Ялта-Лтд            | 9          | 135   |
| 29.05.2010<br>15:00:01 | Реализация № 2<br>29.05.2010 15:00:01 | 1            | Истина     | Пульт VH     | Компания<br>"Риона" | 10         | 150   |
| 13.06.2010<br>15:00:00 | Реализация № 3<br>13.06.2010 15:00:00 | 1            | Истина     | Пульт VH     | Компания<br>"Риона" | 1          | 15    |
| 23.06.2010<br>10:00:00 | Реализация № 3<br>23.06.2010 10:00:00 | 1            | Истина     | Пульт VH     | Компания<br>"Риона" | 1          | 15    |
| 23.06.2010<br>15:30:45 | Реализация № 4<br>23.06.2010 15:30:45 | 1            | Ложь       | Пульт PW     | Крона               | 1          | 90    |
| 01.11.2010<br>08:59:59 | Реализация № 5<br>01.10.2010 08:59:59 | 1            | Истина     | Пульт PW     | Ялта-Лтд            | 1          | 95    |

Рисунок 3.9 – Таблица движений регистра оборотов

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

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

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

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

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

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

| Измерение              |                                       |              |            |              |                     | Ресурс     |       |
|------------------------|---------------------------------------|--------------|------------|--------------|---------------------|------------|-------|
| Период                 | Регистратор                           | Номер строки | Активность | Номенклатура | Контрагент          | Количество | Сумма |
| 27.05.2010<br>12:30:45 | Реализация № 1<br>27.05.2010 12:30:45 | 1            | Истина     | Пульт VH     | Ялта-Лтд            | 9          | 135   |
| 29.05.2010<br>15:00:01 | Реализация № 2<br>29.05.2010 15:00:01 | 1            | Истина     | Пульт VH     | Компания<br>"Риона" | 10         | 150   |
| 13.06.2010<br>15:00:00 | Реализация № 3<br>13.06.2010 15:00:00 | 1            | Истина     | Пульт VH     | Компания<br>"Риона" | 1          | 15    |
| 23.06.2010<br>10:00:00 | Реализация № 3<br>23.06.2010 10:00:00 | 1            | Истина     | Пульт VH     | Компания<br>"Риона" | 1          | 15    |
| 23.06.2010<br>15:30:45 | Реализация № 4<br>23.06.2010 15:30:45 | 1            | Ложь       | Пульт PW     | Крона               | 1          | 90    |
| 01.11.2010<br>08:59:59 | Реализация № 5<br>01.10.2010 08:59:59 | 1            | Истина     | Пульт PW     | Ялта-Лтд            | 1          | 95    |

Рисунок 3.10 – Таблица движений регистра оборотов с разделением итогов

Пример создания:

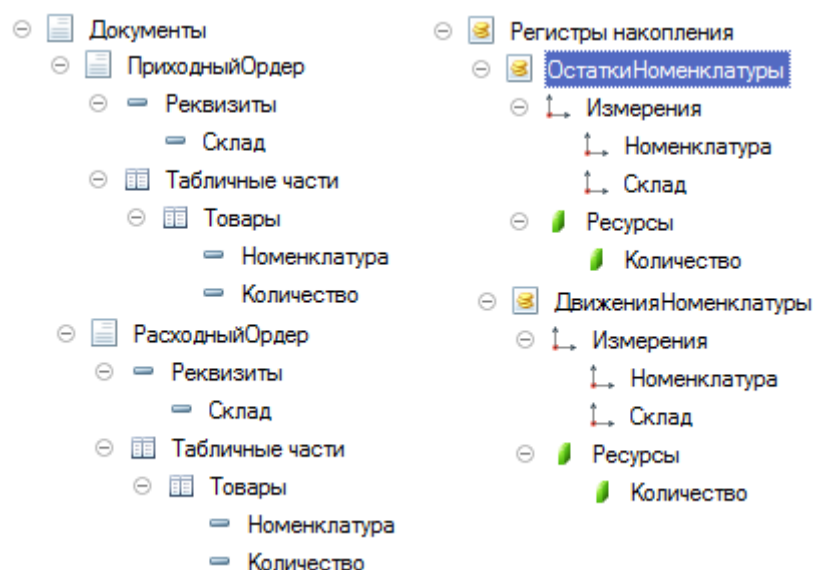


Рисунок 3.11 – Пример создания регистров накопления в дереве конфигурации

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

- **форма списка** - в данной форме нельзя вносить изменения в регистр, но можно делать различные отборы и сортировки;

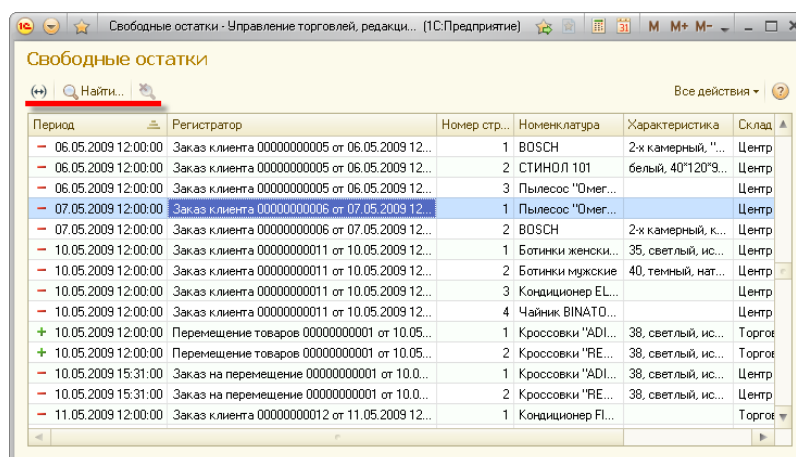


Рисунок 3.12 – Форма списка регистра накопления

- **форма набора записей** – в ней можно редактировать записи регистра: добавлять, удалять и изменять их;

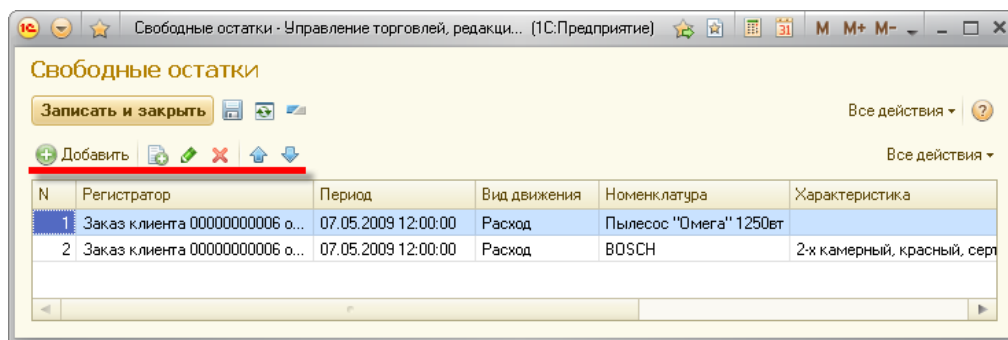


Рисунок 3.13 – Форма набора записей регистра накопления

#### 4. Программный код для работы с регистром накопления

На следующей схеме изображено взаимодействие объектов встроенного языка для работы с регистрами накопления.



Рисунок 4.1 – Взаимодействие объектов встроенного языка для работы с регистрами накопления

Примечание: Заливкой выделен объект манипулирования данными. Метод объекта, от которого идет стрелка, приводится в листинге под соответствующей цифрой, а объект, к которому идет стрелка, – это тип объекта, возвращаемого методом.

В программных модулях для общих действий над регистром накопления (получение остатков и оборотов по данным регистра накопления, поиска нужных записей, формирования выборок, создания форм и макетов) служит объект *РегистрНакопленияМенеджер.<Имя регистра накопления>*.

Для считывания и занесения набора записей в базу данных по определенному условию отбора служит объект *РегистрНакопленияНаборЗаписей.<Имя регистра накопления>*.

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

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

Для динамического обхода записей регистра служит объект *РегистрНакопленияВыборка.<Имя регистра накопления>*.

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

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

// Пример: выполнить полный пересчет итогов регистра "ОстаткиМатериалов".

РегистрыНакопления.ОстаткиМатериалов.ПересчитатьИтоги();

2.
// объект РегистрыНакопленияМенеджер
// .
// []
// Для Каждого ... Из ... Цикл ... КонецЦикла;

// Пример: рассчитать итоги регистра "ОстаткиМатериалов" на указанную дату.

ИмяРегистра = ОстаткиМатериалов;

РегистрыНакопления[ИмяРегистра].УстановитьПериодРассчитанныхИтогов(УказаннаяДата);

3.
// объект РегистрНакопленияМенеджер.
// СоздатьКлючЗаписи()

// Пример: активизировать требуемую строку списка регистра накопления.

СтруктураКлючевыхПолей = Новый Структура;

СтруктураКлючевыхПолей.Вставить("Регистратор", Документы.ПриходнаяНакладная.НайтиПоНомеру("0000002"));

СтруктураКлючевыхПолей.Вставить("НомерСтроки", 2);

Элементы.Материалы.ТекущаяСтрока = РегистрыНакопления.ОстаткиМатериалов.СоздатьКлючЗаписи(СтруктураКлючевыхПолей);

4.
// объект РегистрНакопленияМенеджер.
// СоздатьНаборЗаписей()

// Пример: получить движения документа.

НужныйДокумент = Документы.ПриходнаяНакладная.НайтиПоНомеру(4);
|
Движения = РегистрыНакопления.ОстаткиМатериалов.СоздатьНаборЗаписей();

Движения.Отбор.Регистратор.Значение = НужныйДокумент;

Движения.Прочитать();
```

Рисунок 4.2 – Пример работы с регистром накопления

```

5.
// объект РегистрНакопленияМенеджер.
// Выбрать ()
// ВыбратьПоРегистратору ()

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

6.
// объект РегистрНакопленияНаборЗаписей.
// []
// Для Каждого ... Из ... Цикл ... КонецЦикла;

// Пример: получить движения документа.
НужныйДокумент = Документы.ПриходнаяНакладная.НайтиПоНомеру (4);

Движения = РегистрыНакопления.ОстаткиМатериалов.СоздатьНаборЗаписей ();

Движения.Отбор.Регистратор.Значение = НужныйДокумент;

Движения.Прочитать ();

Для Каждого ОчередноеДвижение Из Движения Цикл
// Алгоритм обработки движений
...
КонецЦикла;

```

Рисунок 4.3 – Создание набора записей регистра накопления

## 5. Пример задачи предприятия по учету оборотных показателей

Заказчик просит создать небольшую информационную систему для регистрации продаж в студенческом киоске. В киоске продают канцелярские принадлежности, литературу и булочки. Нужно отобразить только факт продажи товаров в киоске.

Для решения необходимо создать справочник «Товары», в нём будет храниться перечень товаров, продаваемых в киоске:

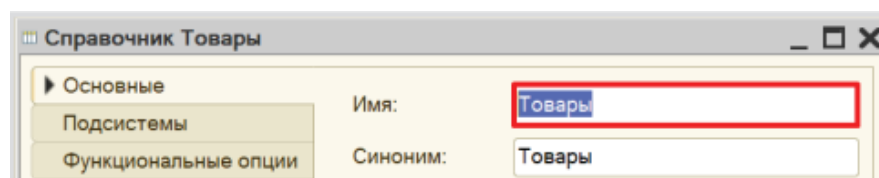


Рисунок 5.1 – Хранение товаров с помощью регистра накопления

Откроем справочник в режиме «1С: Предприятие» и добавим несколько товаров, например, «Ручка», «Линейка», «Пончик».

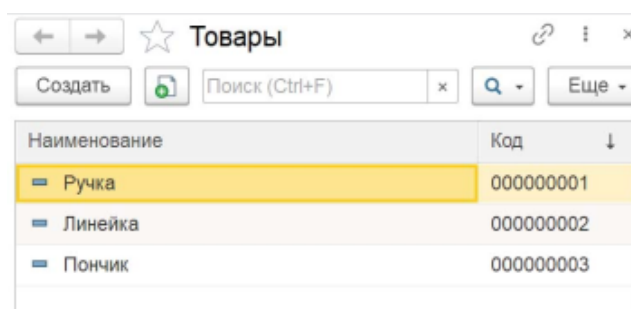


Рисунок 5.2 – Хранимые товары в пользовательском режиме

Создав справочник, мы организовали хранение объектов аналитики. Далее следует ответить на вопрос: «Как мы будем регистрировать продажи?».

Для данной цели подойдёт объект конфигурации «Документ». Добавим новый документ «Продажа», но следует определиться со структурой документа. Мы хотим, чтобы в одном документе можно было регистрировать продажу нескольких товаров. В этом нам поможет табличная часть документа.

Создадим табличную часть (ТЧ) «СписокТоваров». Далее нужно добавить реквизиты ТЧ – колонки таблицы. Для поставленной задачи создадим колонки «Товар» и «Количество»:

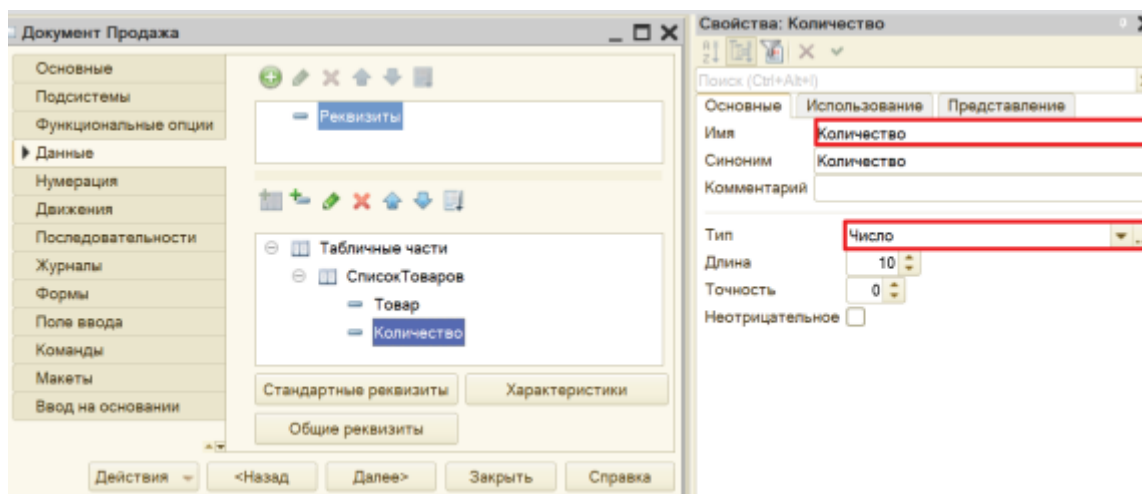


Рисунок 5.3 – Табличная часть «Список товаров»

Для каждого из реквизитов нужно определить имя и тип данных:

- Реквизит «Товар» должен иметь тип «СправочникСсылка.Товары», чтобы хранить ссылку на элемент справочника «Товары».
- Реквизит «Количество» должен быть числовым.

Проверим работу системы. Создадим документ и добавим в него несколько товаров, а также укажем их количество.

| N | Товар   | Количество |
|---|---------|------------|
| 1 | Пончик  | 4          |
| 2 | Ручка   | 5          |
| 3 | Линейка | 2          |

Рисунок 5.4 – Создание документа продажи товаров

Можно ли теперь на основе таких документов построить отчет по продажам? Можно, но для этого придется прибегнуть к грубому перебору всех существующих документов. Данный вариант является неправильным, потому что,

если таких документов окажется очень много, система будет требовать большого количества ресурсов и времени.

Для решения данной проблемы и ускорения процесса извлечения данных создадим ещё один объект – «Регистр накопления».

Данный регистр должен быть оборотным, чтобы накапливать данные о продаже товаров.

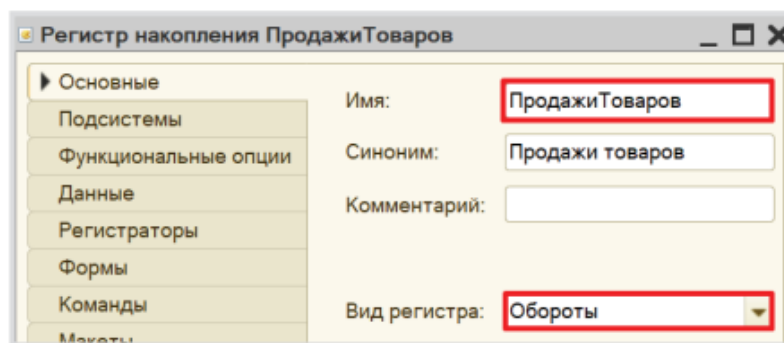


Рисунок 5.5 – Оборотный регистр «ПродажиТоваров»

Переходим к описанию структуры регистра накопления. Для этого переходим на вкладку «Данные».

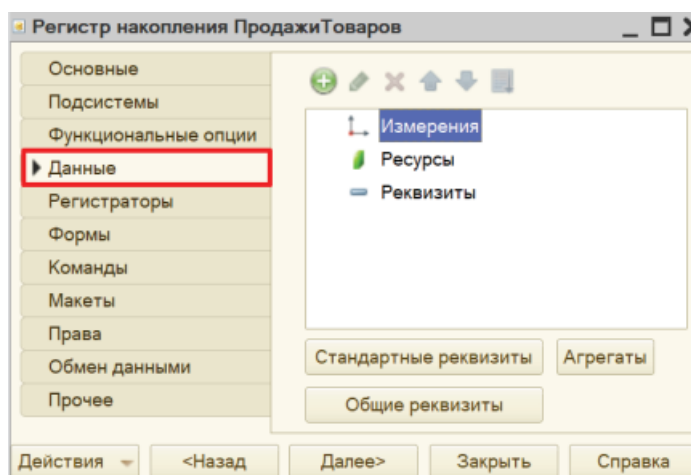


Рисунок 5.6 – Данные оборотного регистра

Заполнение данного окна всегда проще всего начинать с добавления ресурса. Чтобы понять, что использовать в качестве ресурса, необходимо задать вопрос: «Что нам нужно считать?». Нам нужно считать количество. Следовательно, количество и будет являться ресурсом. Тип. Данного реквизита – «Число».

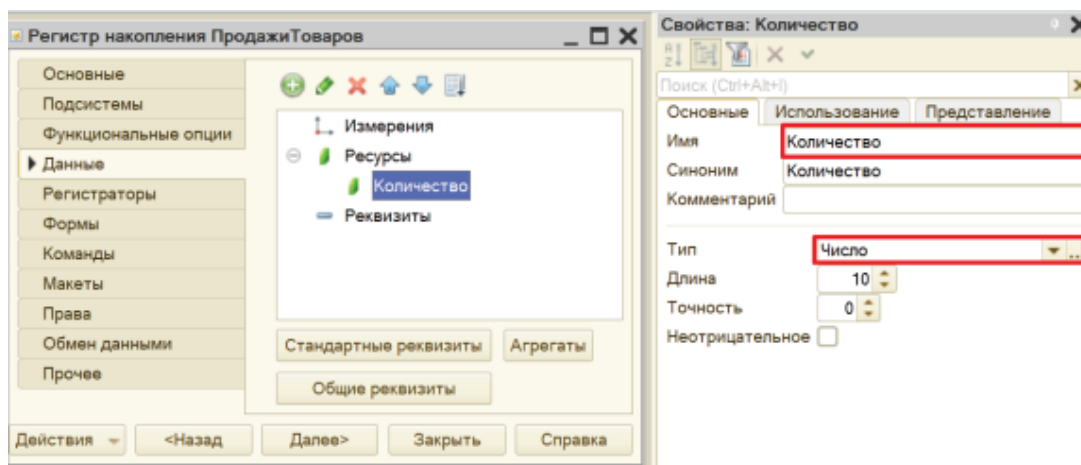


Рисунок 5.7 – Ресурсы оборотного регистра

Чтобы разобраться с измерением, нужно понять, в разрезе чего мы хотим считать количество. Мы хотим считать количество (Чего?) товаров. Значит, в качестве измерения следует добавить реквизит «Товар». Тип данного реквизита – «СправочникСсылка.Товары».

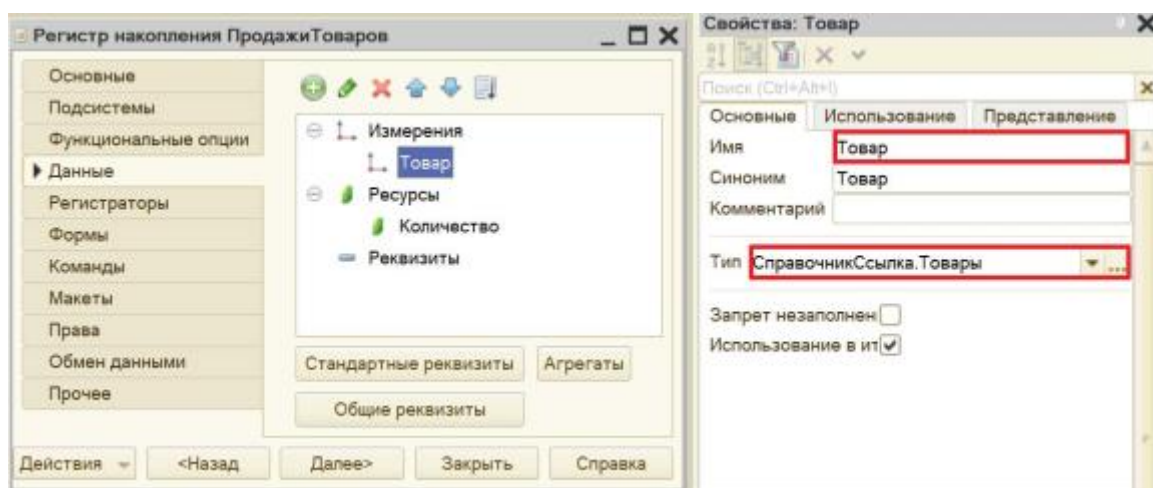


Рисунок 5.8 – Измерения оборотного регистра

Чтобы регистр накопления заработал, нужно сделать следующее:

- 1) Определить источники данных, которые должны попадать в регистр (определить документы регистраторы).
- 2) Описать, каким образом данные из документа-регистратора должны попадать в регистр.

В нашем случае, данные в регистр накопления должны попадать из документа «Продажа», следовательно, данный документ и будет являться регистратором.

Для определения регистратора регистра накопления необходимо выбрать нужный документ на вкладке «Регистраторы».

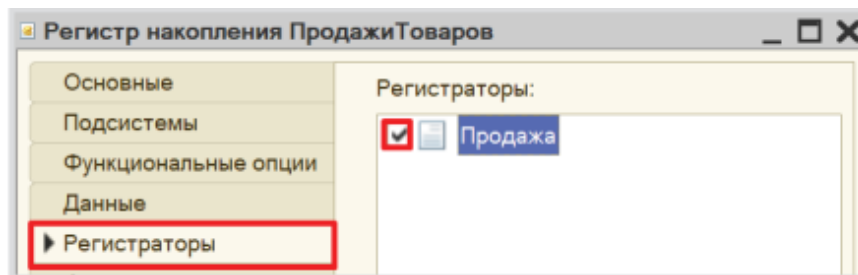


Рисунок 5.9 – Задание документа-регистратора

Чтобы определить правила передачи данных из регистратора в регистр накопления, откроем вкладку «Движения» окна редактирования документа «Продажа» и воспользуемся конструктором движений.

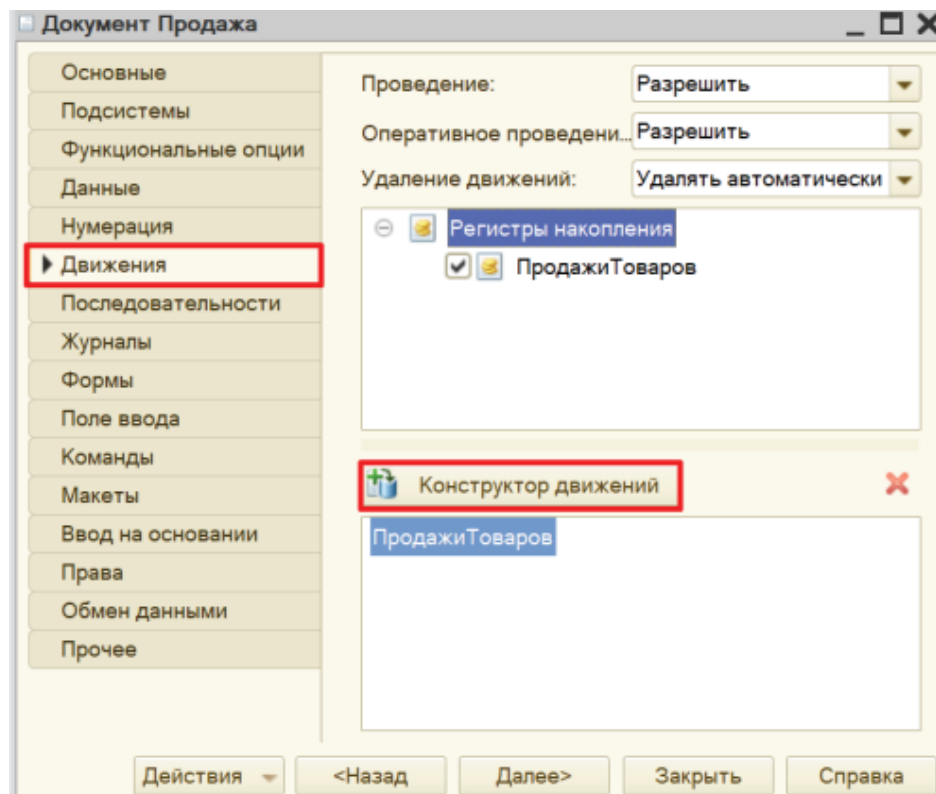


Рисунок 5.10 – Запуск конструктора движений

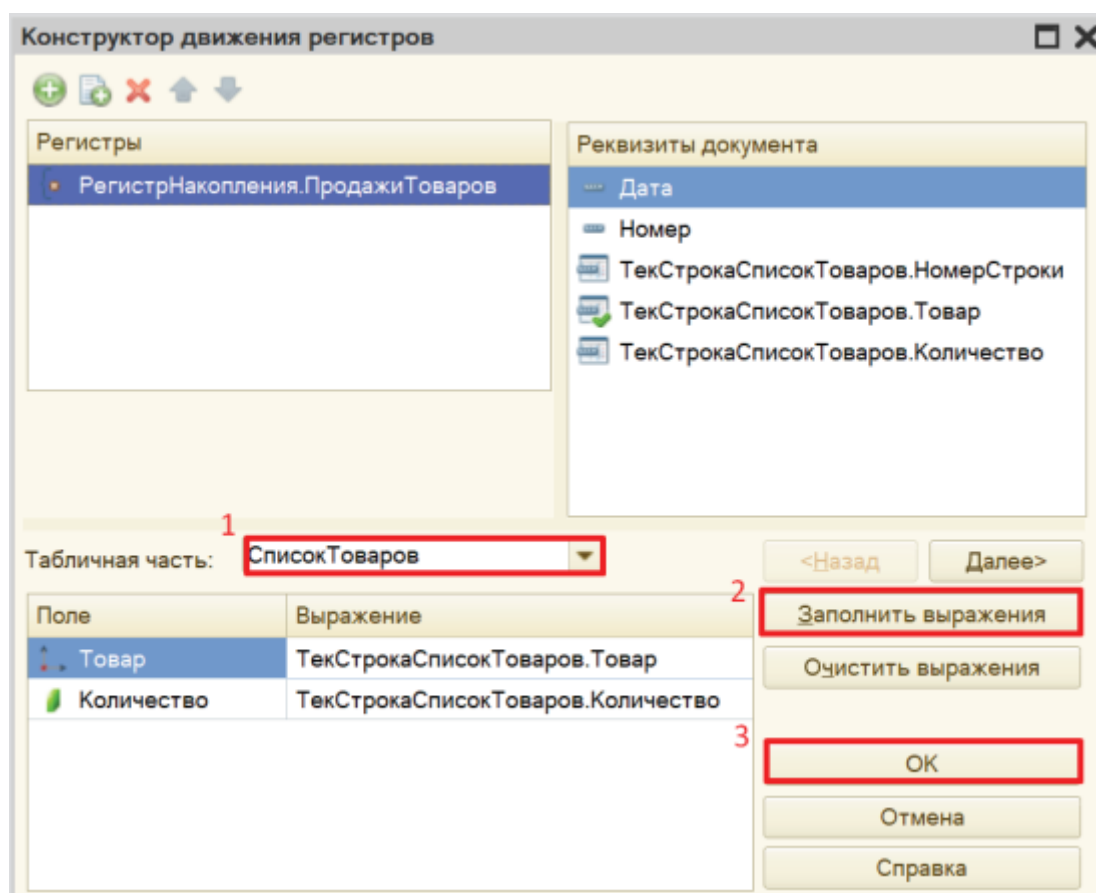


Рисунок 5.11 – Заполнение выражений

Окно конструктора движений состоит из нескольких областей:

- Левая верхняя область позволяет переключаться между регистрами (один документ может делать движения сразу в несколько разных регистров).
- Правая верхняя область описывает реквизиты документа-регистратора. Чтобы отобразить в данной области реквизиты табличной части нужно выбрать ее в соответствующем поле.
- В нижней части окна описаны реквизиты регистра накопления. Нужно заполнить поле «Выражение» реквизитами документа.

Если все было сделано правильно, имена и типы реквизитов совпадают, то при нажатии на кнопку «Заполнить выражения» реквизиты регистра заполнятся автоматически. Если этого не произошло, то заполните поле «Выражение» вручную, путем выбора соответствующих реквизитов документа.

При нажатии на кнопку «ОК» система сформирует программный код, который при успешном проведении документа произведет движения в регистр накопления, то есть скопирует данные из документа в регистр накопления.

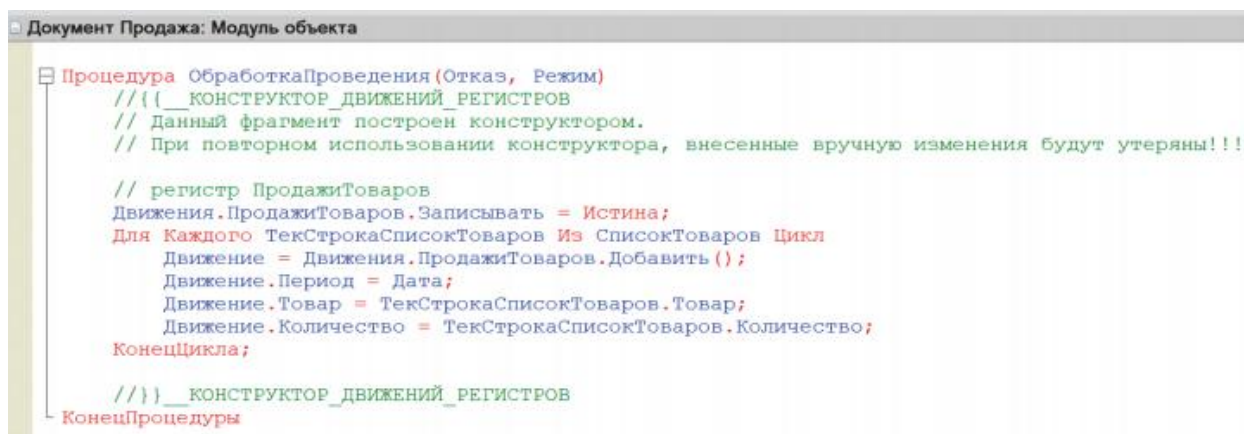


Рисунок 5.12 – Программный код конструктора движений

Проверим работу регистра накопления. В первую очередь, необходимо перепровести (провести заново) созданный нами ранее документ, чтобы он сформировал движение в регистр накопления.

Чтобы посмотреть на движения в регистре накопления воспользуемся функциями для технического специалиста:

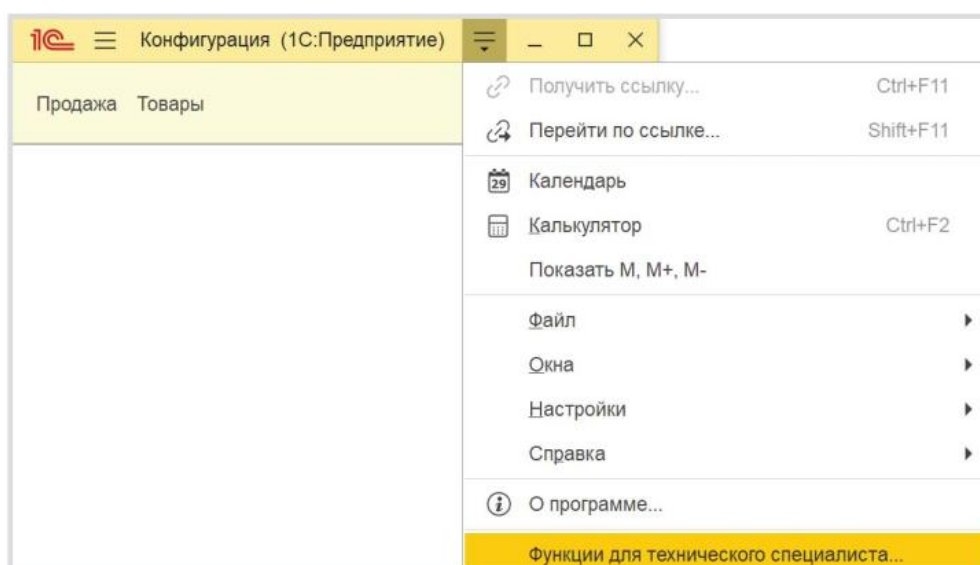


Рисунок 5.13 – Запуск функций для технического специалиста

Найдем наш регистр, откроем его и посмотрим на движения.

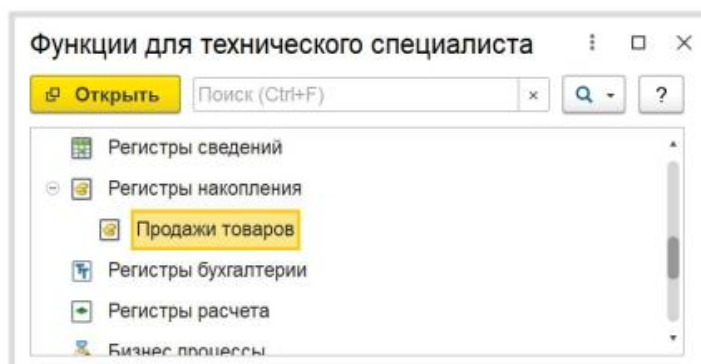


Рисунок 5.14 – Регистр накопления в списке

| Период              | Регистратор                         | Номер ... | Товар   | Количество |
|---------------------|-------------------------------------|-----------|---------|------------|
| 28.08.2020 18:27:52 | Продажа 000000001 от 28.08.2020 ... | 1         | Линейка | 3          |
| 28.08.2020 18:27:52 | Продажа 000000001 от 28.08.2020 ... | 2         | Ручка   | 6          |

Рисунок 5.15 – Движения оборотного регистра накопления

Таким образом, регистр накопления является своеобразной итоговой таблицей, куда заносятся данные из документа «Продажа».

## 6. Пример задачи предприятия по учету остаточных показателей

Заказчик просит разработать конфигурацию для учета товаров. В системе необходимо регистрировать закупку товара. При поступлении товара пользователь в табличной части указывает, какие товары и в каком количестве поступили. Нужно предусмотреть учет до граммов. В системе следует регистрировать продажу товара. При продаже товаров указывается, какие товары были проданы и в каком количестве.

Продать товар «в минус» нельзя, в момент продажи необходимо проверять остаток товара.

Для данной задачи ранее были созданы справочник «Товары» и документ «Продажи», но этого недостаточно.

Для регистрации поступления товара создадим документ «ЗакупкаТоваров».

Рисунок 6.1 – Документ «ЗакупкаТоваров»

По условию задачи, говорится, что необходимо использовать табличную часть. Добавим табличную часть «СписокТоваров». Далее добавим два реквизита табличной части (колонки таблицы): «Товар» (тип – СправочникСсылка.Товары) и «Количество» (тип – «Число»).

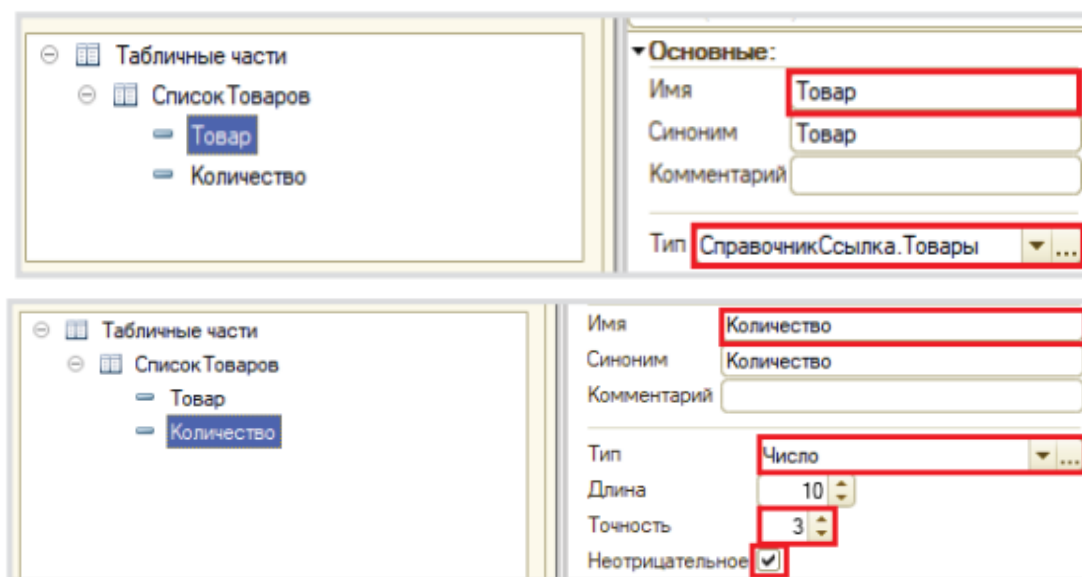


Рисунок 6.2 – Табличная часть «СписокТоваров»

Продажа товара осуществляется аналогично закупке, следовательно, документ по структуре будет точно таким же, как документ «ЗакупкаТовара». В нашей системе уже есть документ «Продажа», переименуем в «ПродажиТоваров», необходимо настроить реквизит «Количество», указав на палитре свойств «Длину» – 15, «Точность» – 3 и «Неотрицательное» число.

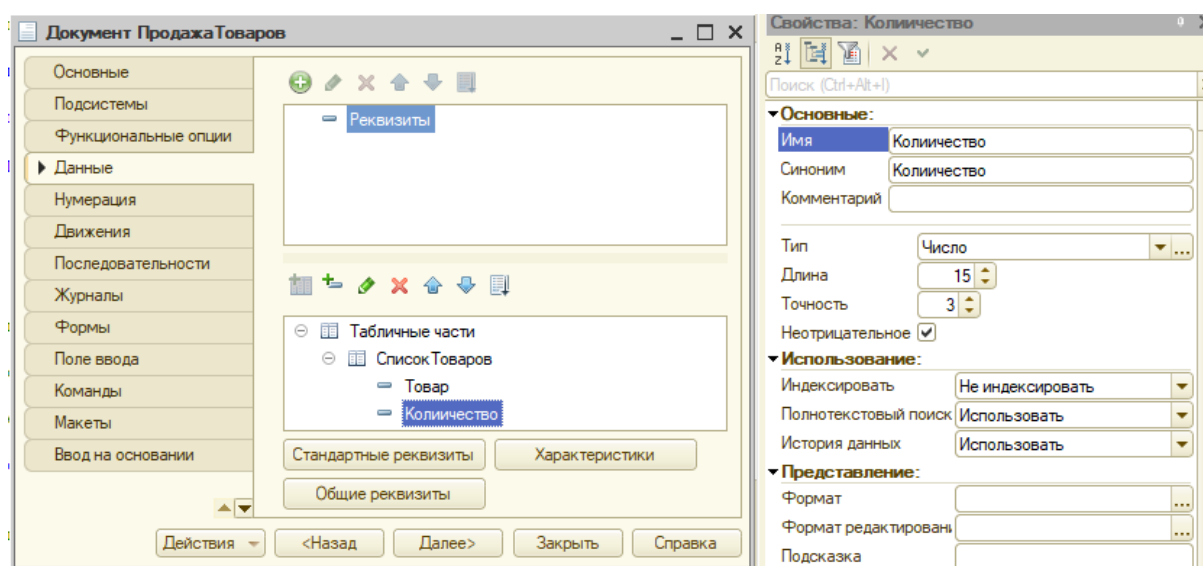


Рисунок 6.3 – Документ «ПродажаТоваров»

На вкладке «Данные» структура должна быть аналогична структуре документа «ЗакупкаТовара»: иметь табличную часть с реквизитами «Товар» и «Количество».

Можно ли теперь на основе таких документов построить отчет по остаткам товаров? Можно, но для этого придется прибегнуть к грубому перебору всех существующих документов. Данный вариант является неправильным, потому что, если таких документов окажется очень много, система будет требовать большого количества ресурсов и времени.

Для решения данной проблемы и ускорения процесса извлечения данных создадим еще один объект – регистр накопления.

Создадим регистр накопления «ОстаткиТоваров» вида «Остатки».

Вид регистра «Остатки» позволяет настроить данный регистр таким образом, что какие-то объекты будут вносить в него данные, а какие-то, наоборот, вычитать. Таким образом и получается хранение остатков.

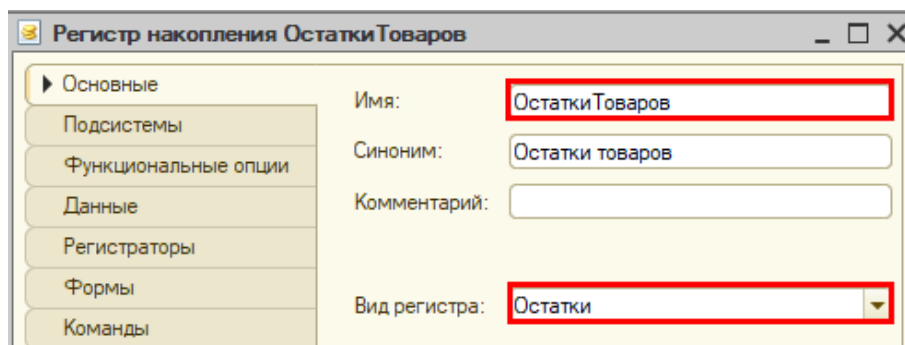


Рисунок 6.4 – Остаточный регистр накопления «ОстаткиТоваров»

Для формирования структуры регистра переходим на вкладку «Данные».

Структура регистра накопления отличается от структуры документа.

Заполнение данного окна проще всего начинать с добавления ресурса. Чтобы понять, что использовать в качестве ресурса, нужно задать вопрос: «Что мы хотим накапливать/считать в данном регистре?». Мы хотим считать количество. Следовательно, количество и будет являться ресурсом. Тип данного реквизита – «Число».

Чтобы разобраться с измерением, необходимо понять, в разрезе чего мы хотим считать количество. Мы хотим считать количество (чего?) товаров. Значит, в качестве измерения следует добавить реквизиты «Товар» (тип – «СправочникСсылка.Товары»).

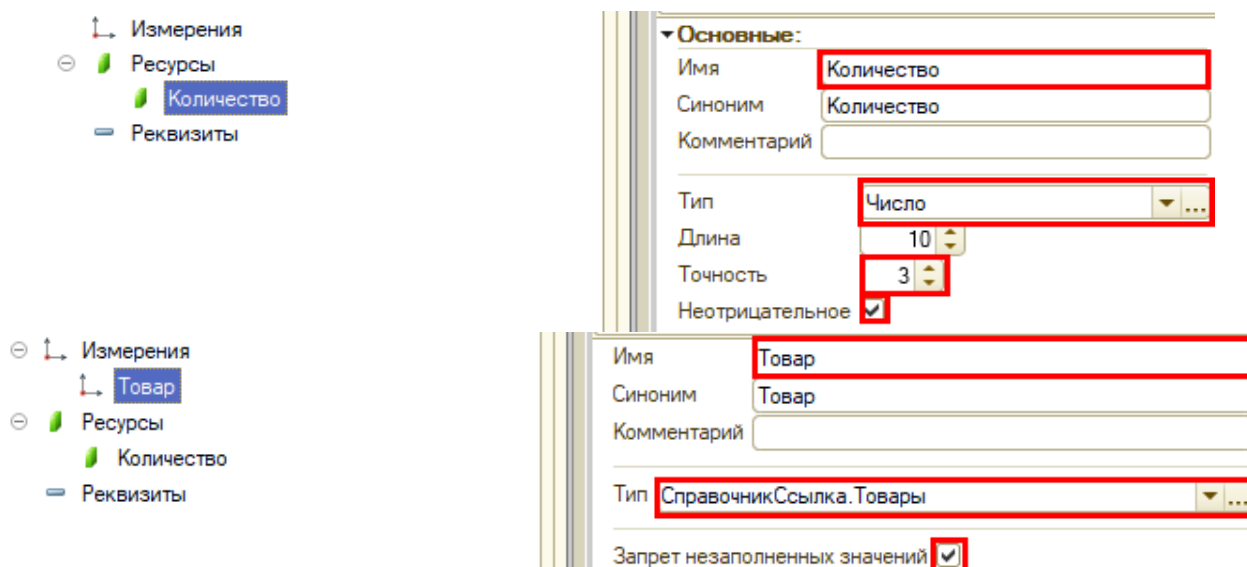


Рисунок 6.5 – Данные регистра накопления

Чтобы регистр накопления заработал, нужно сделать следующее:

- 1) Определить источники данных, которые должны попадать в регистр (определить документы-регистраторы).
- 2) Описать, каким образом данные из документа-регистратора должны попадать в регистр.

Укажем документы, из которых будут формироваться движения в созданный регистр накопления. Для этого перейдем на вкладку «Регистраторы» и отметим галочкой созданные нами документы.

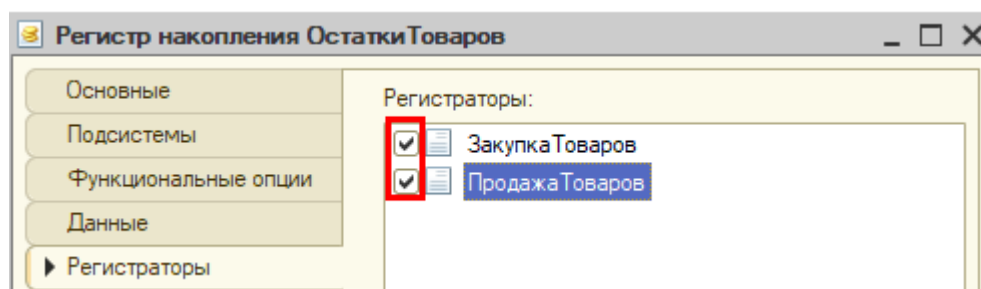


Рисунок 6.6 – Установка документов-регистраторов

Далее для каждого документа-регистратора требуется сформировать движения в созданный регистр накопления.

Откройте окно редактирования объекта «ЗакупкаТоваров» на вкладке «Движения». Убедитесь, что галочка у регистра активна и воспользуйтесь конструктором движений.

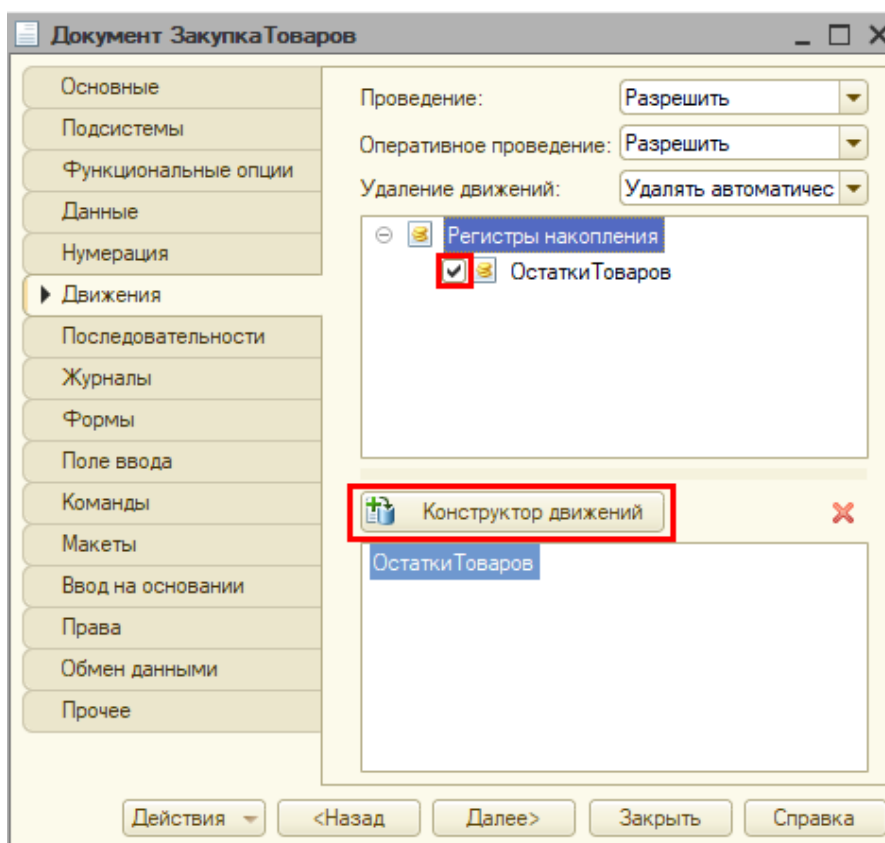


Рисунок 6.7 – Запуск конструктора движений

Окно конструктора движений состоит из нескольких областей:

- Левая верхняя область позволяет переключаться между регистрами (один документ может делать движения сразу в несколько разных регистров).
- Правая верхняя область описывает реквизиты документа-регистратора. Чтобы отобразить в данной области реквизиты табличной части, нужно выбрать ее в соответствующем поле.
- В нижней части окна описаны реквизиты регистра накопления. Нужно заполнить поле «Выражение» реквизитами документа.

Поскольку закупка товара должна увеличивать количество товаров на складе, то тип движения регистра необходимо выбрать «Приход». Регистр будет обозначаться знаком «+» (плюс).

Если все было сделано правильно, имена и типы реквизитов совпадают, то при нажатии на кнопку «Заполнить выражения» реквизиты регистра заполнятся автоматически. Если этого не произошло, то заполните поле «Выражение» вручную, путем выбора соответствующих реквизитов документа.

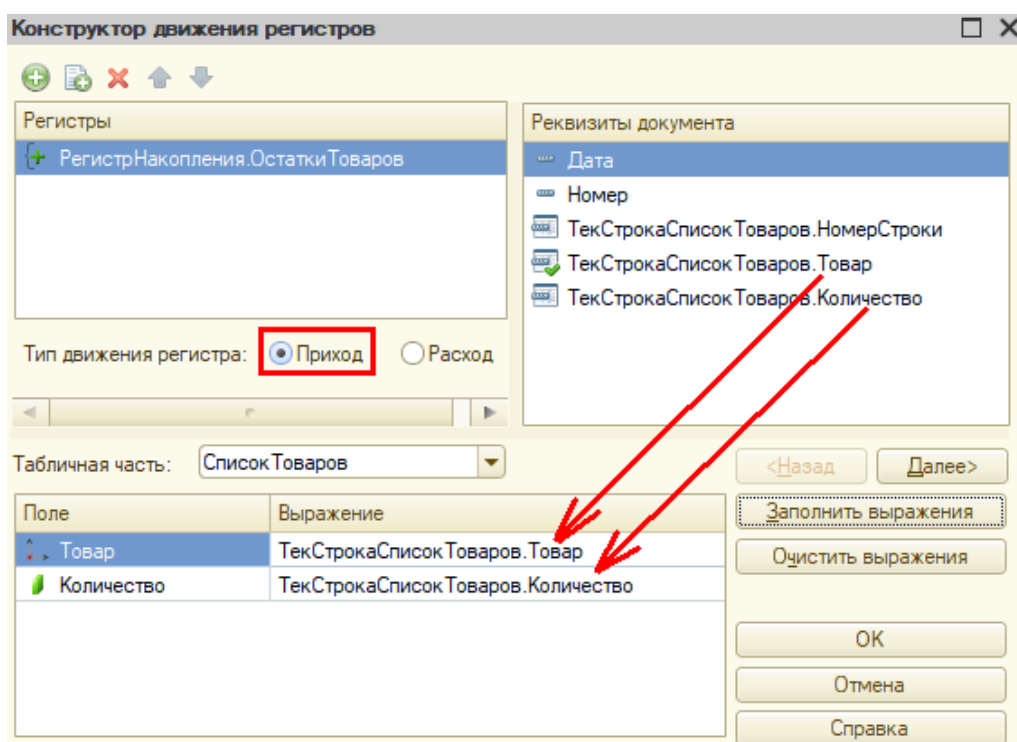


Рисунок 6.8 – Заполнение прихода товаров

При нажатии на кнопку «ОК» система формирует программный код, который при успешном проведении документа произведет движения в регистр накопления, то есть копирует данные из документа в регистр накопления.

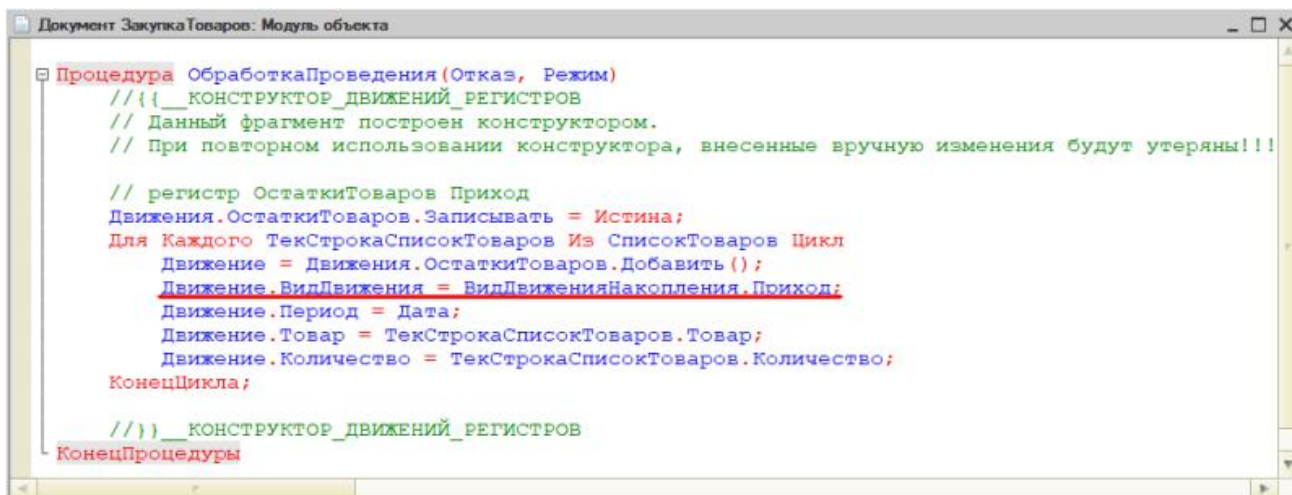


Рисунок 6.9 – Приход товаров

По аналогии сформируйте движения для второго документа «ПродажаТоваров», изменив вид движения на «Расход».

В результате в модуле объекта документа «ПродажаТоваров» вы получите аналогичный программный код, который отличается только видом движения.

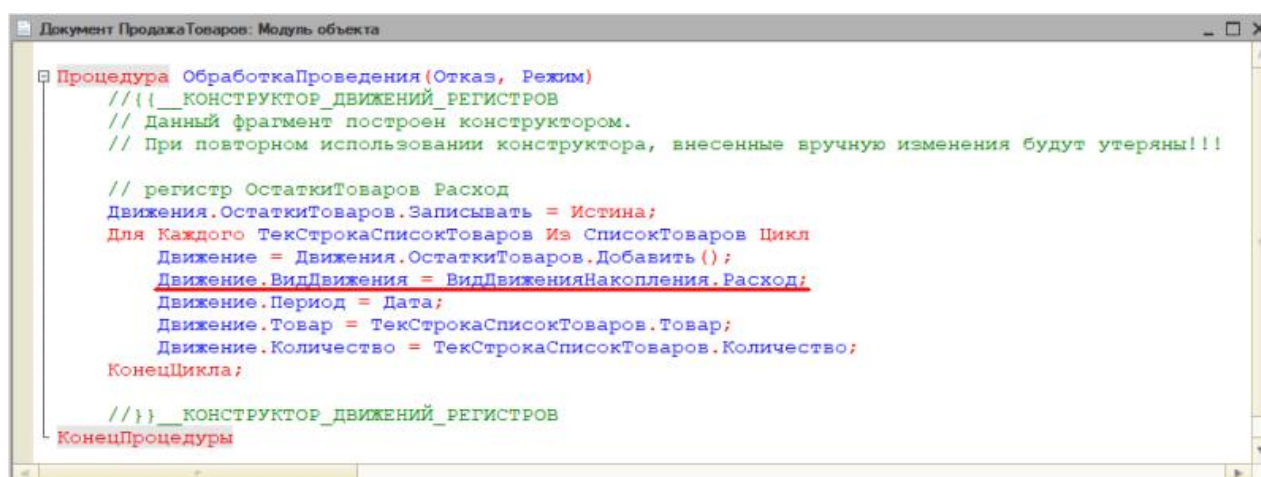


Рисунок 6.10 – Расход товаров

Проверим работоспособность системы, заполнив тестовыми данными справочник, документы «ЗакупкаТоваров» и «ПродажаТоваров».

Если в документе «Продажа товаров» указать количество больше, чем имеется, то будут получены отрицательные остатки.

Контроль отрицательных остатков является заключительным и самым важным этапом выполнения данной работы.

Проверять остатки товаров будем следующим образом:

- 1) Сделаем движение данных из документа в регистр накопления.
- 2) Проверим, появились ли в регистре остатки, значение которых меньше нуля (то есть отрицательные).
- 3) Если есть отрицательные остатки, то отменим сделанное движение в регистр накопления и выведем пользователю сообщение об ошибке.

Контроль отрицательных остатков должен происходить в момент проведения документа «ПродажаТоваров». Откроем модуль объекта данного документа.

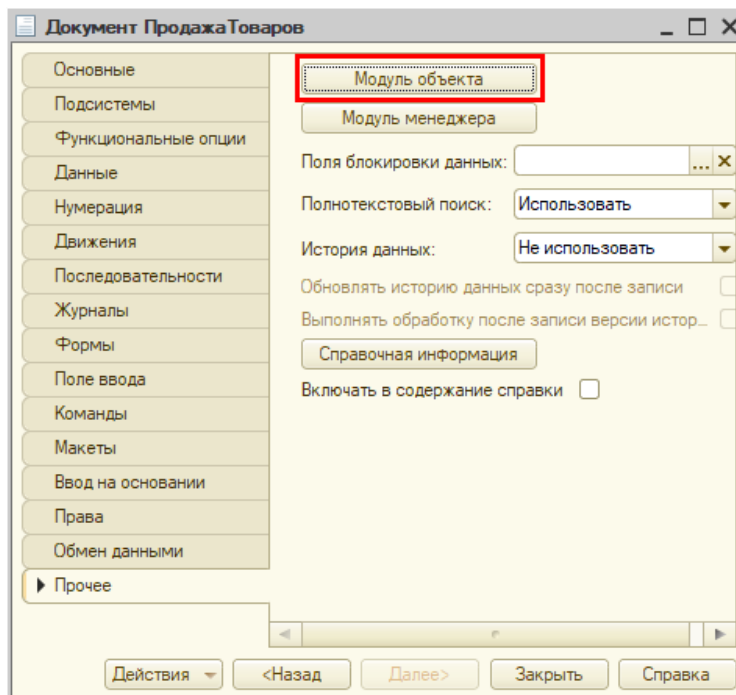


Рисунок 6.11 – Модуль объекта документа «ПродажаТоваров»

Чтобы сделать движение данных из документа в регистр накопления, допишем после окончания цикла строку «Движения.Записать();». Метод записывает только те движения документа, у которых установлен флаг «Записывать», при этом флаг в итоге снимается, что не приводит к повторной записи движений по окончании транзакции проведения.

И главное, «Движения.Записать();» всегда записывают движения в том порядке, в котором таблицы указаны в дереве метаданных, что на порядок уменьшает шансы взаимных блокировок, ведь все транзакции в одинаковом порядке блокируют таблицы.

```

□ Процедура ОбработкаПроведения(Отказ, Режим)

    // регистр ОстаткиТоваров Расход
    Движения.ОстаткиТоваров.Записывать = Истина;
    Для Каждого ТекСтрокаСписокТоваров Из СписокТоваров Цикл
        Движение = Движения.ОстаткиТоваров.Добавить ();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Товар = ТекСтрокаСписокТоваров.Товар;
        Движение.Количество = ТекСтрокаСписокТоваров.Количество;
    КонецЦикла;

    Движения.Записать ();

КонецПроцедуры

```

Рисунок 6.12 – Запись движений регистра

Теперь, когда движение было сделано, можно обратиться к данным регистра накопления.

Для получения данных из регистра существует два ключевых подхода: использование запросов к базе данных и использование объектной модели 1С.

Если в процессе получения данных необходимо выполнять сложный отбор или получать данные нескольких объектов лучше использовать запросы. Но если необходимо простое получение данных с простыми условиями отбора можно использовать объектную модель.

Для каждого объекта конфигурации существуют встроенные свои функции для работы с объектной моделью. Например для регистра накопления используются три главных объектных метода: Выбрать, Остатки и Обороты. Каждый из методов соответствует варианту получения данных с помощью виртуальных таблиц в запросах которые будут разобраны в следующей работе.

В данном примере воспользуемся объектной моделью получения остатков регистра накопления. Для этого воспользуемся методом менеджера регистра «ОстаткиТоваров» остатки.

```
СтруктураОтбора = Новый Структура;  
СтруктураОтбора.Вставить("Товар", СписокТоваров.ВыгрузитьКолонку("Товар"));  
  
ГраницаВыборки = Новый Граница(МоментВремени(), ВидГраницы.Включая);  
  
ВыборкаРегистра = РегистрыНакопления.ОстаткиТоваров.Остатки(ГраницаВыборки, СтруктураОтбора);
```

Рисунок 6.13 – Получение остатков товаров

Для получения только необходимых данных нам надо наложить отбор на получаемые данные, это позволит при дальнейшей обработке результатов работать только с теми товарами которые используются в документе.

В первую очередь необходимо ограничить выборку данных по времени, для этого создадим параметр момент граница времени тип данных граница. Важно отметить что для того что бы в выборку данных были включены данные записанные в регистр документом граница должна иметь вид «Включая».

Второй параметр процедуры предназначен для ограничения выборки по измерениям регистра. Создадим структуру в которой имя ключа структуры должно совпадать с именем измерения регистра, заданного в конфигураторе, а значение элемента структуры - задает отбираемое по данному измерению значение.

В результате выполнения функции будет получена таблица значений со столбцами Товар и Количество, и строками в которых товар равен товару из документа.

Воспользуемся циклом для обхода результатов выборки. Так как мы получаем данные регистра сразу после записи движения документа, а дополнительных условий в отборе мы не накладывали таблица никогда не будет пустой.

Нам необходимо найти те записи регистра количество товара в которых является отрицательным. Для этого добавим в цикл проверку значения.

```

Для Каждого ЗаписьРегистра Из ВыборкаРегистра Цикл
    Если ЗаписьРегистра.Количество < 0 Тогда

        КонецЕсли;
    КонецЦикла;

```

Рисунок 6.14 – Проверка наличия отрицательных остатков товара

Внутри условия можно попасть только в том случае, если количество товара отрицательное. В таком случае нужно отменить проведение документа и выдать пользователю сообщение:

```

Для Каждого ЗаписьРегистра Из ВыборкаРегистра Цикл
    Если ЗаписьРегистра.Количество < 0 Тогда
        Отказ = Истина;

        Сообщить ("Не хватает товара " + ЗаписьРегистра.Товар +
            ", в количестве " + ( - ЗаписьРегистра.Количество));
    КонецЕсли;
КонецЦикла;

```

Рисунок 6.15 – Отказ от проведения документа в случае наличия отрицательных остатков товара

Код процедуры полностью должен выглядеть следующим образом:

```

□ Процедура ОбработкаПроведения(Отказ, Режим)

    // регистр ОстаткиТоваров Расход
    Движения.ОстаткиТоваров.Записывать = Истина;
    Для Каждого ТекСтрокаСписокТоваров Из СписокТоваров Цикл
        Движение = Движения.ОстаткиТоваров.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Товар = ТекСтрокаСписокТоваров.Товар;
        Движение.Количество = ТекСтрокаСписокТоваров.Количество;
    КонецЦикла;

    Движения.Записать();

    СтруктураОтбора = Новый Структура;
    СтруктураОтбора.Вставить("Товар", СписокТоваров.ВыгрузитьКолонку("Товар"));

    ГраницаВыборки = Новый Граница(МоментВремени(), ВидГраницы.Включая);

    ВыборкаРегистра = РегистрыНакопления.ОстаткиТоваров.Остатки(ГраницаВыборки, СтруктураОтбора);

    Для Каждого ЗаписьРегистра Из ВыборкаРегистра Цикл
        Если ЗаписьРегистра.Количество < 0 Тогда
            Отказ = Истина;

            Сообщить ("Не хватает товара " + ЗаписьРегистра.Товар +
                ", в количестве " + ( - ЗаписьРегистра.Количество));
        КонецЕсли;
    КонецЦикла;

КонецПроцедуры

```

Рисунок 6.16 – Итоговый код процедуры

Теперь проверим корректность работы системы на предмет наличия ошибок при попытке продать товара больше, чем его имеется в наличии.

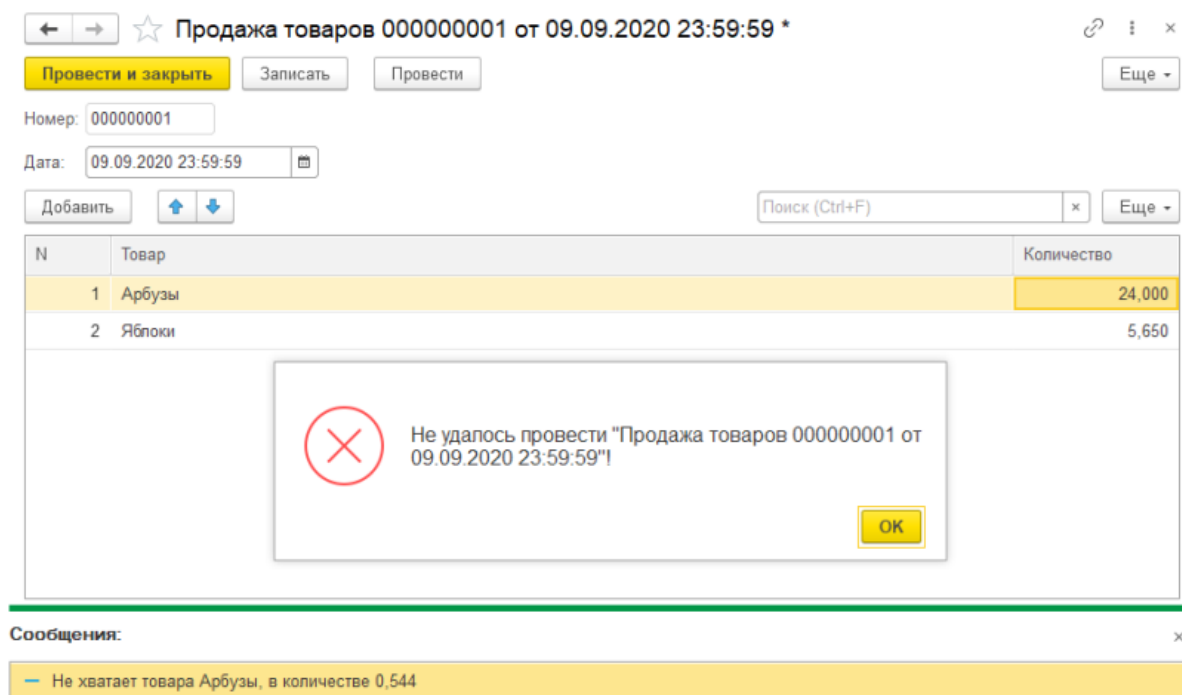


Рисунок 6.17 – Сообщение о наличии отрицательного остатка товара

Как видно из сообщения, товара не хватает в том количестве, которое мы указали, чтобы механизм работал необходимо указывать не больше количества по данному товару из документа «ЗакупкиТоваров».

Если изменить количество на меньшее, то механизм будет работать корректно.

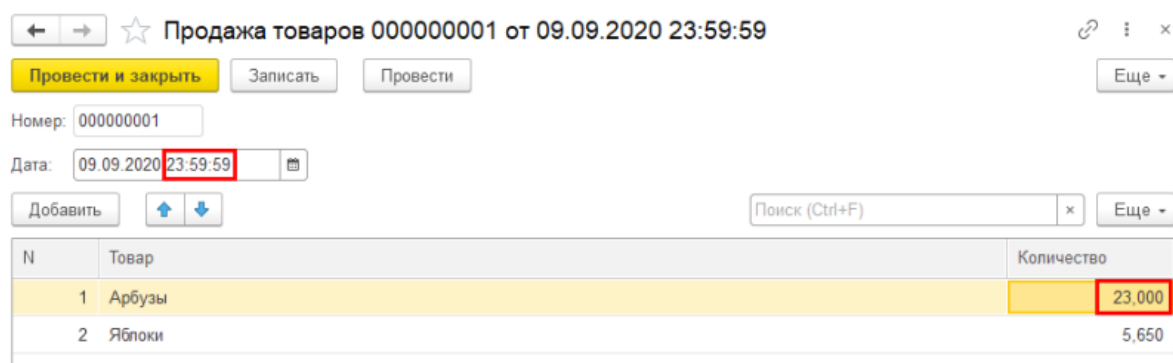


Рисунок 6.18 – Корректная работа механизма контроля остатков

Таким образом, был рассмотрен пример, позволяющий реализовать контроль остатков товаров с использованием объектной модели.

## **Задание на лабораторную работу**

Разработка конфигурации для хранения данных предметной области с использованием регистров накопления. Структура справочников, документов и регистров выбираются самостоятельно.

Необходимо выполнить две задачи с использованием объектной модели получения данных:

Задача 1 – организация регистра накопления остатки.

Задача 2 – организация регистра накопления обороты.

### **Вариант 1**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы автошколы. В частности, конфигурация должна позволять хранить:

- Хранение информации о неоплаченной сумме оплаты (долг) за обучение слушателю курса.
- Каждый слушатель должен пройти в течение курса определенное количество часов вождения.

### **Вариант 2**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы университета. В частности, конфигурация должна позволять хранить:

- Контроль зачисления студентов на бюджетные места.
- Информацию о количестве успешно окончивших бакалавр в разрезе специальностей.

### **Вариант 3**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы гостиницы. В частности, конфигурация должна позволять хранить:

- Контроль свободных мест (номеров) в гостинице.
- Информацию о доходах гостиницы.

### **Вариант 4**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы салона красоты. В частности, конфигурация должна позволять хранить:

- Контроль остатков по номенклатуре (Например, краска, кисти, перчатки и т.д.).
- Информацию о жалобах от клиентов.

### **Вариант 5**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы туристического агентства. В частности, конфигурация должна позволять хранить:

- Информацию о задолженностях перед туроператором.
- Информацию о продажах клиентам различных туров сотрудниками турагентств.

### **Вариант 6**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы школьной библиотеке. В частности, конфигурация должна позволять хранить:

- Контроль книг, отслеживание выдачи книг и возвратов читателем.
- Информацию о рейтинге книг.

### **Вариант 7**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы системы обеспечения дошкольного питания. В частности, конфигурация должна позволять хранить:

- Контроль остатков продуктов.
- Информацию о продаже блюд.

### **Вариант 8**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы торгового предприятия. В частности, конфигурация должна позволять хранить:

- Контроль остатков товаров.
- Информацию о продаже товаров и услуг.

### **Вариант 9**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы транспортного предприятия. В частности, конфигурация должна позволять хранить:

- Информацию о свободном грузчике и транспорте.
- Информацию о продажах услуг.

### **Вариант 10**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы Интернет-провайдера. В частности, конфигурация должна позволять хранить:

- Информацию о задолженностях абонентов.
- Информацию о жалобах от абонентов.

### **Вариант 11**

Постановка: Заказчик просит разработать информационную систему для автоматизации расчета оплаты труда на предприятии. В частности, конфигурация должна позволять хранить:

- Контроль выплат сотрудникам (начисление и выплаты).
- Информацию о доходах каждого сотрудника.

### **Вариант 12**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы жилищно-коммунального предприятия. В частности, конфигурация должна позволять хранить:

- Информацию о задолженностях каждого собственника дома.
- Информацию о количестве выполненных услуг за весь период.

### **Вариант 13**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы мобильного оператора. В частности, конфигурация должна позволять хранить:

- Информацию о задолженности клиента.
- Информацию об отзывах оставленные клиентом.

### **Вариант 14**

Постановка: Заказчик просит разработать информационную систему для автоматизации управления прайс-листами. В частности, конфигурация должна позволять хранить:

- Контроль остатков товаров.
- Информацию о продажах товаров.

### **Вариант 15**

Постановка: Заказчик просит разработать информационную систему для автоматизации работы подсистемы товарооборота предприятия. В частности, конфигурация должна позволять хранить:

- Информацию по остаткам товаров.
- Информацию о продажах товаров.

## **Содержание отчета**

1. Цель работы.
2. Описание варианта задания.
3. Пошаговое описание процесса выполнения варианта задания.
4. Выводы

## **Контрольные вопросы**

1. Для чего предназначен объект конфигурации Регистр накопления.
2. Почему следует использовать регистры, хотя необходимая информация содержится в других объектах.
3. Для чего нужны измерения регистра, ресурсы и реквизиты.
4. Что такое движения регистра и что такое регистратор.
5. Как создать новый регистр накопления и описать его структуру.
6. Как создать движения документа с помощью конструктора движений.
7. В чем главные отличия регистра сведений от регистра накопления?
8. Как средствами встроенного языка обойти табличную часть документа и обратиться к ее данным?
9. Какое максимальное количество измерений можно определять для регистра накопления с видом Остатки?
10. Что следует учитывать при определении вида регистра накопления?
11. Какой регистр накопления можно использовать для получения информации об остатках накопленных средств?
12. Какой регистр накопления можно использовать для получения информации об оборотах накопленных средств?
13. Как система определяет обороты для регистра накопления с видом остатки?
14. Зачем существует регистр накопления оборотный, если в регистре накопления остатков есть возможность получения оборотов?
15. Для каких задач может использоваться механизм получения оборотных данных в регистре накопления с видом остатки?



# **ПЛАТФОРМА 1С**

*Методические указания*

Часть 4

*Составители:* **Шевченко** Виктория Игоревна, **Виноградов** Евгений Викторович,  
**Кудашев** Вячеслав Сергеевич, **Сергеев** Никита Олегович

В авторской редакции

Изд. № 89/2023. Объем 2,5 п.л. Усл. печ. л. 2,32. Уч.-изд. л. 2,45.  
РИИЦМ ФГАОУ ВО «Севастопольский государственный университет»