

## Лабораторная работа №14

### «Отловить» первый запуск информационной системы

#### 1. Задание:

После установки программы пользователю нужно заполнить некоторые параметры для дальнейшей работы. Без этого корректная работа программы невозможна. Необходимо сделать так, чтобы при первом запуске системы открывалась форма с параметрами. Признаком первого запуска будет служить константа со значением «ЛОЖЬ». Для упрощения задачи форму следует открыть с простым текстом-декорацией «Обнаружен первый запуск программы!».

#### 2. Подготовка:

- создать новую информационную базу;
- открыть информационную базу в режиме «Конфигуратор»;
- открыть окно конфигурации.

#### 3 Выполнение:

3.1 Создать константу «ПервыйЗапускОсуществлен», тип: Булево.

Данная константа должна имитировать первый запуск системы. Если константа содержит значение «Ложь» при запуске программы в режиме «1С:Предприятие», то следует оповестить пользователя о том, что это первый запуск системы. Если константа содержит значение «Истина», то при запуске программы окно с оповещением открываться не должно. Для оповещения пользователя следует создать форму.

3.2 Создать общую форму «УведомлениеОПервомЗапуске», тип: произвольная. Обязательно поставьте галочку «Использовать стандартные команды». Нажать кнопку «Далее», затем кнопку «Готово».

3.3 В открывшемся окне добавить новый элемент формы «Декорация – Надпись», нажать кнопку «ОК».

Среди элементов формы появится элемент «Декорация1», а также откроется палитра свойств данного элемента. Добавим заголовок: «Обнаружен первый запуск программы!» и посмотрим на результат в области предпросмотра.

3.4 На вкладке «Оформление» нужно настроить размер и цвет новой декорации.

3.5 Данная форма будет выводиться пользователю, если константа содержит значение «Ложь». Но необходимо также добавить команду (кнопку формы), с помощью которой пользователь сможет указать системе, чтобы в дальнейшем эта форма не открывалась, как показано на рисунках 1, 2.

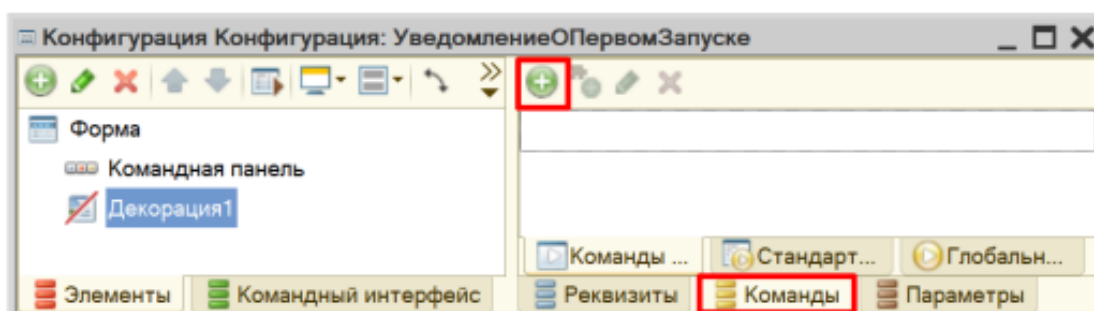


Рисунок 1 – Настройка уведомления о первом запуске

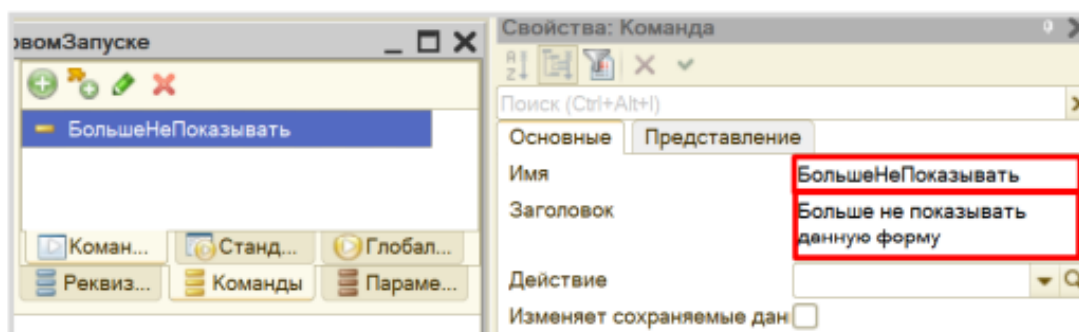


Рисунок 2 – Настройка уведомления о первом запуске

3.6 Перетащить команду на форму. Для этого следует «зажать» кнопку мыши на команде и перетащить ее на форму. Команда автоматически встанет вслед за декорацией. Причем команда предстала перед нами как элемент формы «Кнопка», как показано на рисунке 3.

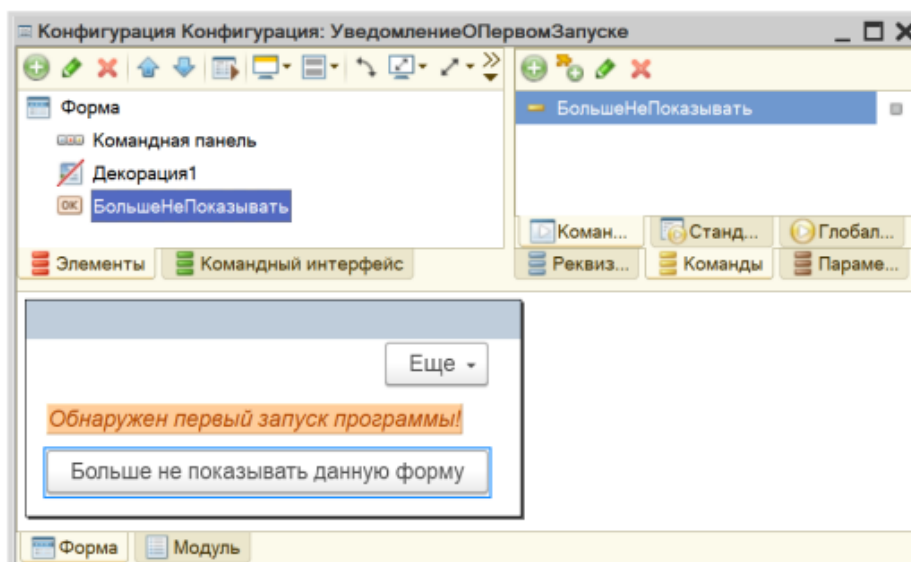


Рисунок 3 – Настройка уведомления о первом запуске

Если запустить систему в режиме «1С:Предприятие», то кнопка на форме будет, но она не будет совершать никаких действий, поскольку мы их еще не описали. Вернемся к этому чуть позже.

Вторая часть задачи заключается в том, чтобы «поймать» момент запуска системы, потому что именно в этот момент должна произойти проверка константы и открытие формы.

Алгоритмы, происходящие при запуске системы, описываются в модуле приложения. Найти этот модуль можно в контекстном меню корня конфигурации.

3.7 Открыть список процедур, как показано на рисунке 4 и выбрать процедуру «ПриНачалеРаботыСистемы».

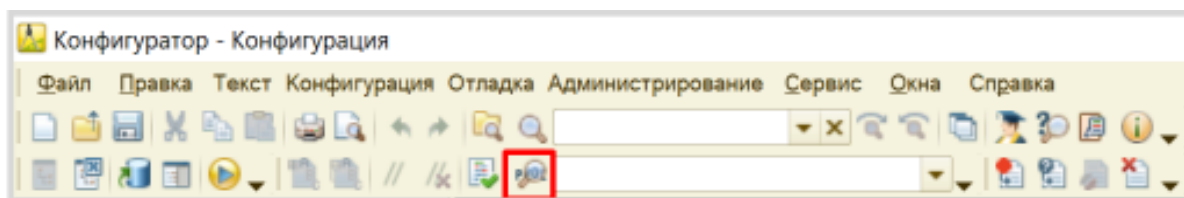


Рисунок 4 – Список процедур

Наша задача состоит в том, чтобы прочитать значение константы и, если значение константы «Ложь», открыть форму. Но получить данные константы, которая хранится в базе данных, напрямую из формы приложения мы никак не сможем. Это связано с тем, что модуль приложения выполняется на стороне клиента, а к базе данных получить доступ можно только со стороны сервера.

Чтобы разобраться, как устроено клиент-серверное взаимодействие, рассмотрим конкретный пример.

Предположим, пользователь хочет сформировать отчет.

1. Клиент нажимает на кнопку «Сформировать».
2. Запрос отправляется на Сервер 1С.
3. Сервер 1С отправляет запрос SQL-серверу.
4. SQL-сервер запрашивает данные из информационной базы и формирует таблицы.
5. Заполненные таблицы SQL-сервер отправляет Серверу 1С.
6. Сервер 1С рисует форму отчета и отправляет отчет клиенту в простом и понятном для него виде, например, таблицей или диаграммой.

Таким образом, для того, чтобы получить любые данные из базы данных, нужно обратиться к серверу. Как это сделать? Для получения контекста сервера необходимо создать общий модуль «Сервер». Обязательно нужно проставить галочки у свойств «Сервер» и «Вызов сервера», как показано на рисунке 5.

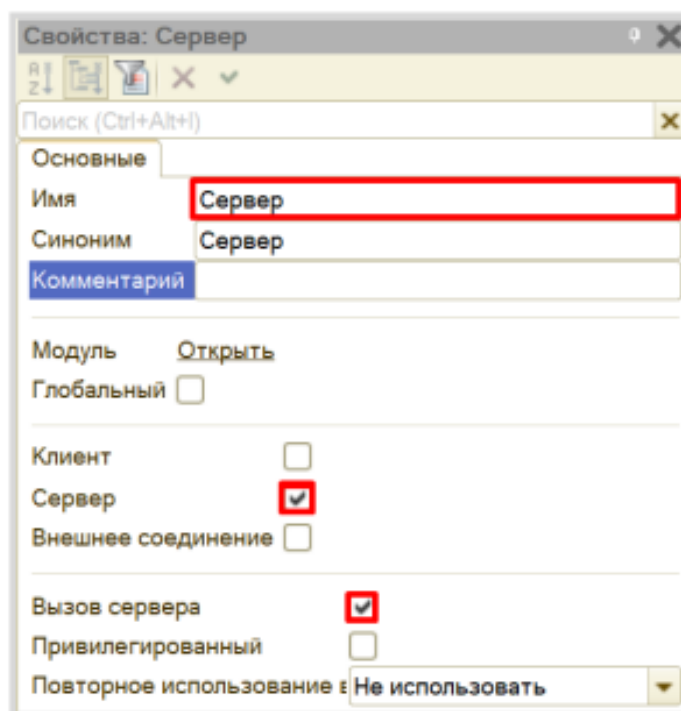


Рисунок 5 – Создание общего модуля «Сервер»

Включение свойства «Сервер» означает, что данный модуль будет выполняться на стороне сервера. Включение свойства «Вызов сервера» значит, что клиентские модули (например, созданная нами ранее форма) смогут обращаться к данному модулю и использовать его процедуры и функции.

Опишем в общем модуле функцию «ЭтоПервыйЗапуск», как показано на рисунке 6.

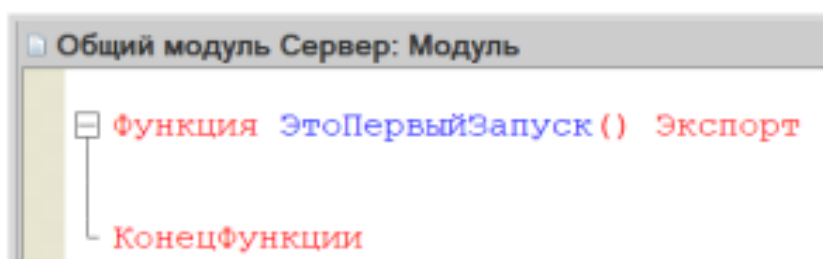
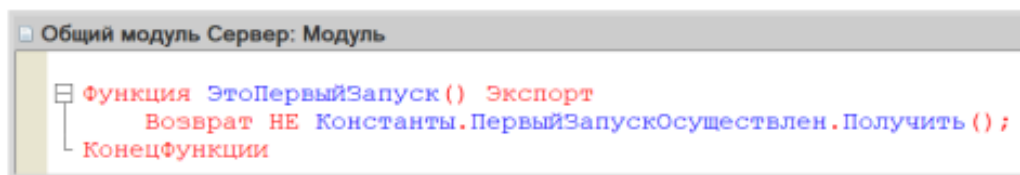


Рисунок 6 – Код функции

3.8 Служебное слово «Экспорт», указанное после названия функции, делает ее видимой для вызова из других модулей системы. В самой функции опишем получение константы и возврат ее перевернутого значения. Если в константе хранится значение «Истина», то есть первый запуск уже был осуществлен, то функция вернет значение «Ложь», и наоборот, как показано на рисунке 7.

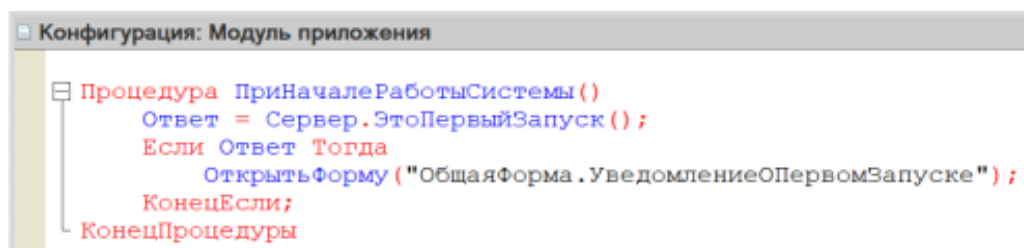


```
Общий модуль Сервер: Модуль

Функция ЭтоПервыйЗапуск() Экспорт
    Возврат НЕ Константы.ПервыйЗапускОсуществлен.Получить();
КонецФункции
```

Рисунок 7- Код функции

3.9 Поскольку общий модуль работает на стороне сервера, то он легко сможет получить значение константы. А служебное слово «Экспорт» позволит обратиться к этой функции откуда угодно. Значит, можно вызвать эту функцию из модуля приложения, где был подготовлен шаблон процедуры «ПриНачалеРаботыСистемы», как показано на рисунке 8.



```
Конфигурация: Модуль приложения

Процедура ПриНачалеРаботыСистемы()
    Ответ = Сервер.ЭтоПервыйЗапуск();
    Если Ответ Тогда
        ОткрытьФорму ("ОбщаяФорма.УведомлениеОПервомЗапуске");
    КонецЕсли;
КонецПроцедуры
```

Рисунок 8 – Код модуля приложения

Переменной «Ответ» будет присвоено значение функции «ЭтоПервыйЗапуск», осуществляемой на стороне сервера. Затем, если в

переменной «Ответ» хранится значение «Истина», будет открыта созданная форма.

Не забудьте проверить модуль на наличие синтаксических ошибок.

### 3.10 Запустить систему в режиме «1С:Предприятие».

При запуске системы открылась созданная форма, уведомляющая пользователя о том, что был произведен первый запуск. Данная форма будет открываться при каждом запуске системы до тех пор, пока значение константы «ПервыйЗапускОсуществлен» не изменится на «Истина» (по умолчанию всегда «Ложь»).

3.11 Для того, чтобы форма не открывалась повторно, описать алгоритм работы кнопки, которая была добавлена на форму. Открыть редактор общей формы и в палитре свойств команды создать новое действие, которое будет выполнять данная команда, как показано на рисунке 9.

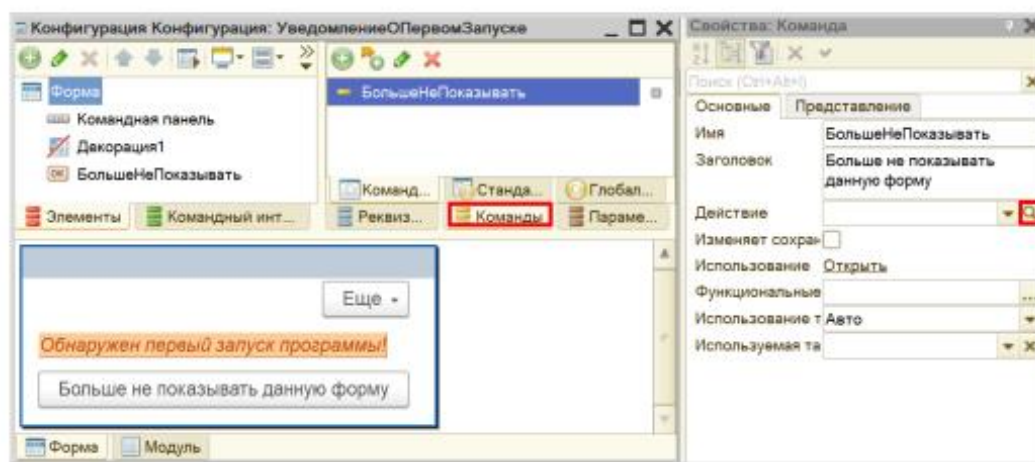


Рисунок 9 – Создание нового действия

Система задаст вопрос: «Где создать обработчик команды?». Для того чтобы сделать правильный выбор, нужно разобраться, что мы хотим сделать? Мы хотим установить значение константы в значение «Истина». Для этого нужен доступ к базе данных, следовательно, нам необходим контекст сервера. Выбрать «Создать на клиенте и процедуру на сервере».

Система создала в модуле формы две процедуры: нижняя – обработчик события нажатия кнопки, верхняя – серверная процедура, которая вызывается из нижнего обработчика.

3.12 Описать присвоение значения константе, как показано на рисунке 10.

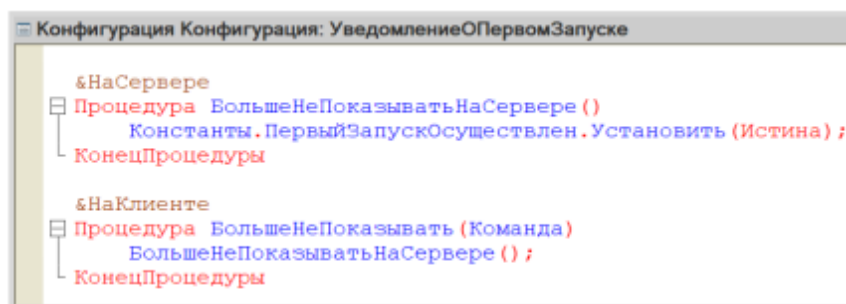


Рисунок 10 – Описание присвоения значения константе

3.13 Осталось лишь закрыть форму. Форма всегда существует на клиенте, закрывать форму будем в контексте клиента, как показано на рисунке 11.

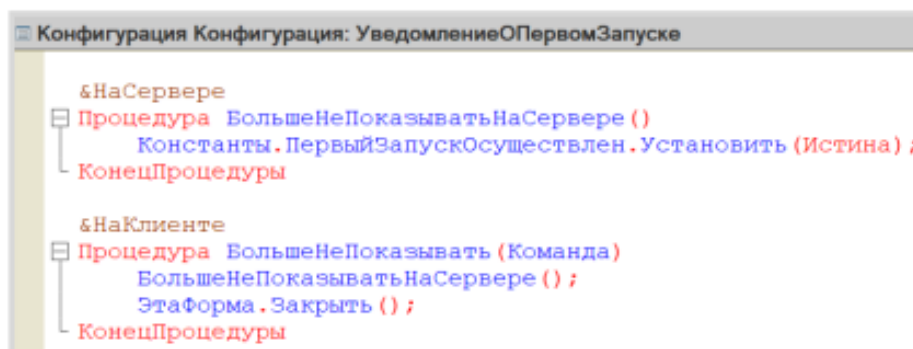


Рисунок 11 – Описание присвоения значения константе

3.14 Проверьте модуль на наличие синтаксических ошибок.

3.15 Открыть систему в режиме «1С:Предприятие».

При запуске отрывается окно с уведомлением.



3.16 Нажать на кнопку «Больше не показывать данную форму».

Форма должна закрыться. Проверим, изменилось ли значение константы, как показано на рисунке 12.

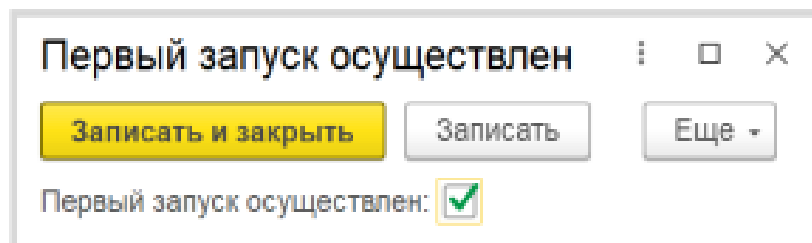


Рисунок 12 – Изменение значения константы

Значение константы изменилось на «Истина».

При повторном запуске системы форма с уведомлением открываться не должна.

## Лабораторная работа №15

### Разработка конфигурации для учета товаров

#### 1. Задание:

Заказчик просит разработать конфигурацию для учета товаров.

Многоскладской учет не ведется. Суммовой учет не ведется.

Взаиморасчеты с покупателями и поставщиками не ведутся.

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.

Нужно построить «Отчет» по остаткам товаров.

Форма отчета:

Остатки товаров на 31.01.2020

Товар	Остаток
Ложка	100.000
Вилка	148.000
Поварешка	2.000
Сахар	1.560

Отчет строится на конец дня, указанного пользователем. Особое внимание следует уделить последней секунде дня: документы, записанные на эту секунду, должны попадать в отчет.

#### 2. Подготовка:

- создать новую информационную базу;
- открыть информационную базу в режиме «Конфигуратор»;
- открыть окно конфигурации.

### **3 Выполнение:**

3.1 Создать справочник «Товары».

3.2 Создать документ «ЗакупкаТоваров».

3.3 Перейти на вкладку «Данные», создать табличную часть «СписокТоваров».

3.4 Добавить два реквизита табличной части (колонки таблицы): «Товар» (тип – СправочникСсылка.Товары) и «Количество» (тип – «Число», точность – 3, поставить галочку «неотрицательное»).

3.5 Создать копированием документа «ЗакупкаТоваров», новый документ и изменить имя документа на «ПродажаТоваров».

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

3.6 Создать регистр накопления «ОстаткиТоваров» вида «Остатки».

3.7 Перейти на вкладку «Данные».

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»). Поставить галочку «Запрет незаполненных значений».

3.8 Перейти на вкладку «Регистраторы» и отметить галочкой созданные нами документы.

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

3.9 Открыть окно редактирования объекта «ЗакупкаТоваров» на вкладке «Движения». Убедиться, что галочка у регистра активна и воспользоваться конструктором движений.

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.

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, как показано на рисунке 1.

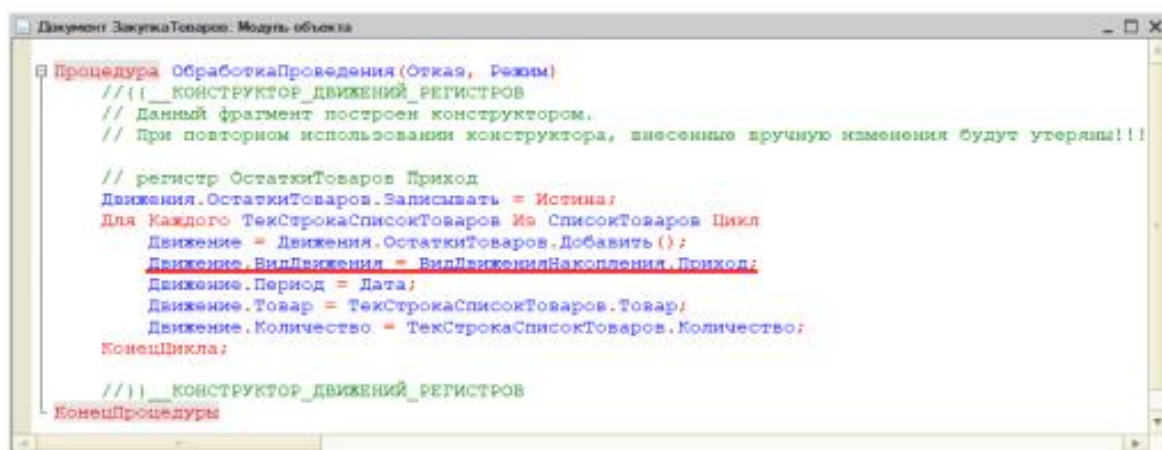


Рисунок 1 – Код модуля объекта

3.10 По аналогии сформировать 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, как показано на рисунке 2.

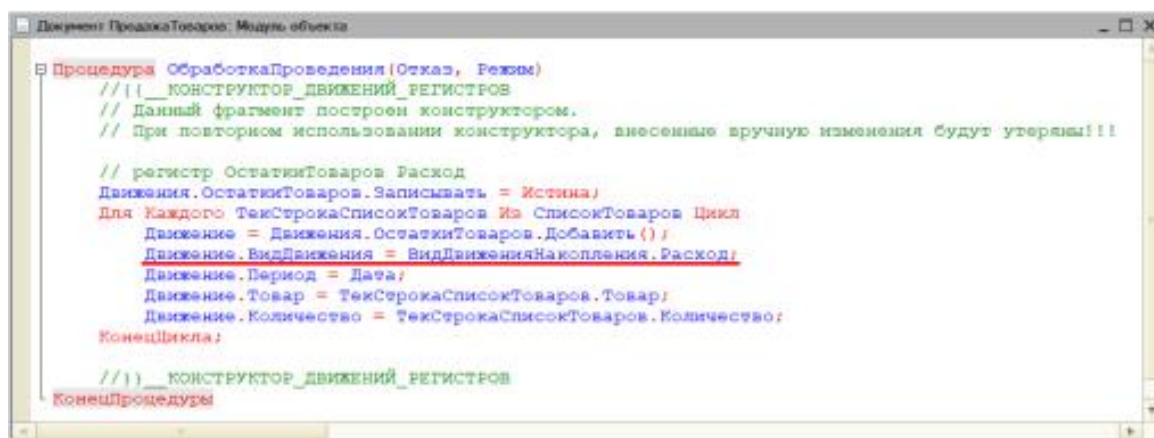


Рисунок 2 – Код модуля объекта

Для визуализации работы регистра накопления нужно создать отчет.

3.11 Создать новый отчет «ОстаткиТоваров». Для наполнения отчета воспользоваться конструктором схемы компоновки данных.

Все созданные нами объекты конфигурации представляют собой таблицы базы данных. В режиме «1С:Предприятие» мы заполняем эти таблицы данными. Чтобы получить эти данные для отображения в отчете, нужно сформировать запрос к базе данных.

При построении запросов используется собственный язык запросов 1С. Их можно писать вручную или воспользоваться конструктором запросов.

Чтобы перенести данные из одного окна в другое, просто перетащите нужные поля с помощью мыши либо воспользуйтесь стрелочками, расположенными между окнами, как показано на рисунке 3.

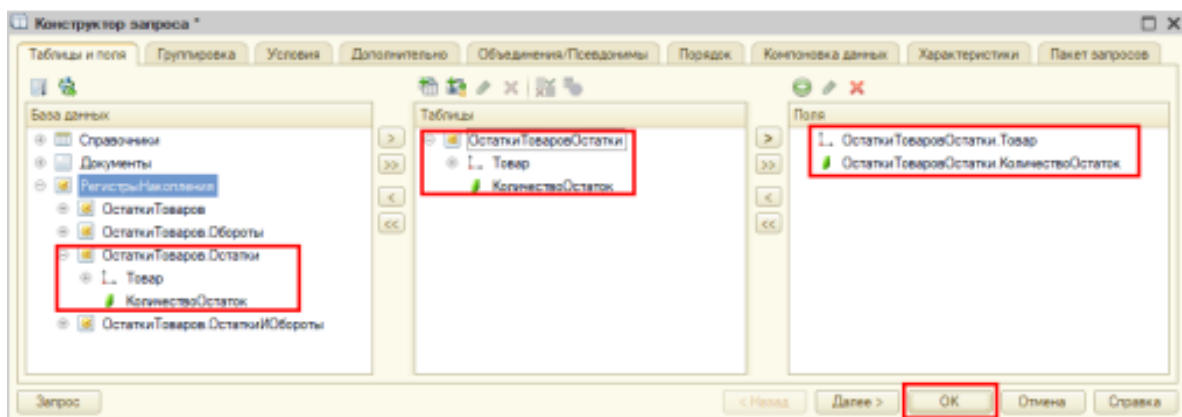


Рисунок 3 – Заполнение отчета данными

3.12 Изменить заголовок у поля «КоличествоОстаток» на «Количество», как показано на рисунке 4.

Поля:							
	Поле	Путь		Ограничение поля			
			Автозаголовок	По_	Ус_	Гр_	Уп_
				Ограничение реквиз...			
				По_	Ус_	Гр_	Уп_
⇒	Товар	Товар		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/> Товар		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
⇒	КоличествоОс	Количество		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/> Количество Остаток		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рисунок 4 – Изменение заголовка

3.13 Из условия следует, что отчет должен включать документы, записанные на последнюю секунду дня. При использовании стандартных методов такие документы в отчет попадать не будут. Поэтому нужно добавить новый параметр «ДатаОтчета» на соответствующей вкладке, как показано на рисунке 5.

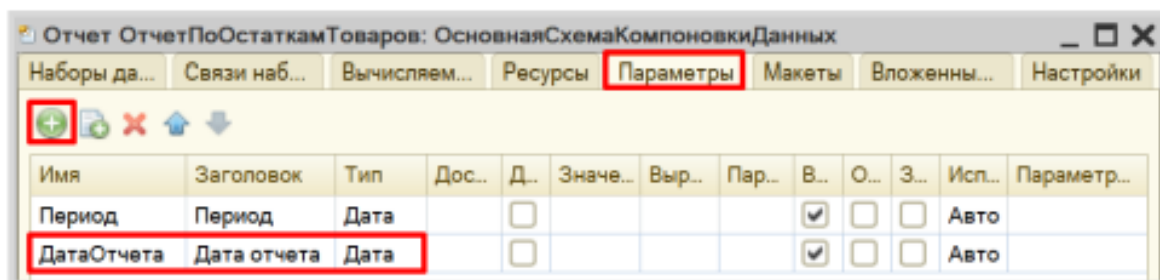


Рисунок 5 – Изменение параметров отчета

3.14 Чтобы у пользователя была возможность выбирать только даты, без указания секунд, нужно настроить формат редактирования параметра «ДатаОтчета», как показано на рисунках 6, 7, 8.

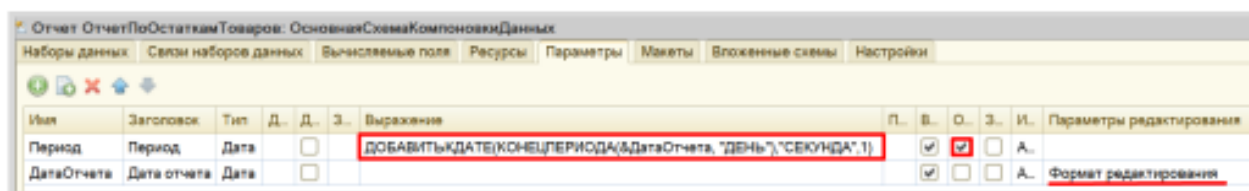


Рисунок 6 – Настройка формата редактирования параметра

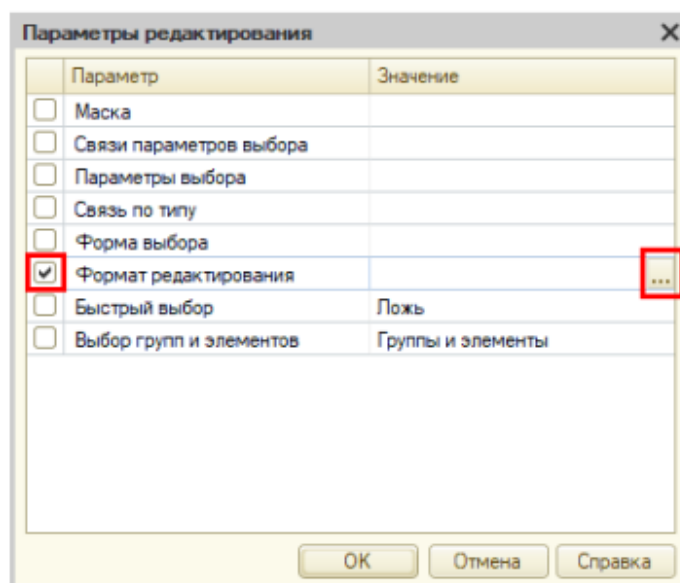


Рисунок 7 – Настройка формата редактирования параметра

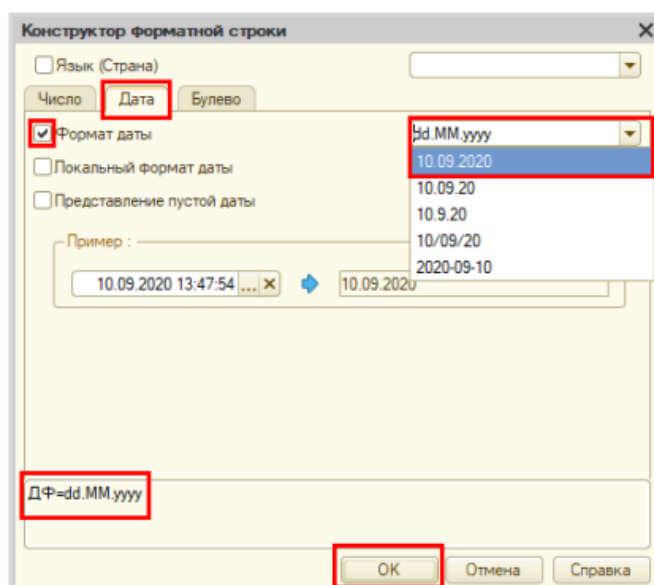


Рисунок 8 – Настройка формата редактирования параметра

3.15 После нажатия кнопки «ОК» нужно настроить стандартный параметр «Период» для корректного учета последней секунды дня:

- во-первых, этот параметр должен быть недоступен пользователю, так как носит вычислительный характер;
- во-вторых, для корректного расчета требуется написать выражение для стандартного параметра «Период»:

Сначала берется последний момент дня, указанного в параметре «ДатаОтчета», а затем прибавляется еще одна секунда, чтобы учитывались даже документы, проведенные за эту последнюю секунду, как показано на рисунке 9.

Отчет ОтчетПоОстаткамТоваров: ОсновнаяСхемаКомпоновкиДанных												
Наборы данных   Связи наборов данных   Вычисляемые поля   Ресурсы   Параметры   Макеты   Вложенные схемы   Настройки												
Имя	Заголовок	Тип	Д..	З..	З..	Выражение	П..	В..	О..	З..	И..	Параметры редактирования
Период	Период	Дата	<input type="checkbox"/>			ДОБАВИТЬКДАТЕ(КОНЕЦПЕРИОДА(&ДатаОтчета, "ДЕНЬ"), "СЕКУНДА", 1)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		A..	
ДатаОтчета	Дата отчета	Дата	<input type="checkbox"/>				<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		A..	Формат редактирования

Рисунок 9 - Настройка формата редактирования параметра

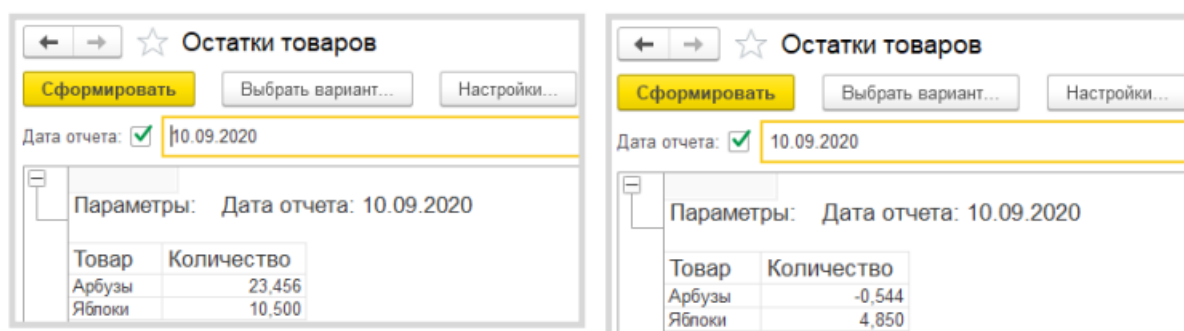


3.16 Перейти на вкладку «Настройки» и воспользоваться конструктором настроек отчета, выбрать «Список», нажать кнопку «Далее», затем выбрать поля «Товар», «Количество», нажать кнопку «ОК».

3.17 Чтобы у пользователя была возможность выбирать требуемый день, вам нужно включить параметр «ДатаОтчета» в пользовательские настройки. Это можно сделать в нижней части вкладки «Настройки».

3.18 Проверьте работоспособность системы, заполнив тестовыми данными справочник, документы «ЗакупкаТоваров» и «ПродажаТоваров», а также построив отчет.

Чтобы в дальнейшем убедиться в корректности вывода информации о документах, проведенных в конце дня, создавайте документы за предыдущие числа. Это необходимо, поскольку созданные вами документы имеют оперативный режим проведения. Для наглядности постройте отчет до проведения документа «ПродажаТоваров» и после, как показано на рисунке 10.



Остатки товаров	
Товар	Количество
Арбузы	23,456
Яблоки	10,500

Остатки товаров	
Товар	Количество
Арбузы	-0,544
Яблоки	4,850

Рисунок 10 – Отчет

Если в документе «Продажа товаров» указать количество больше, чем имеется на складе, 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.

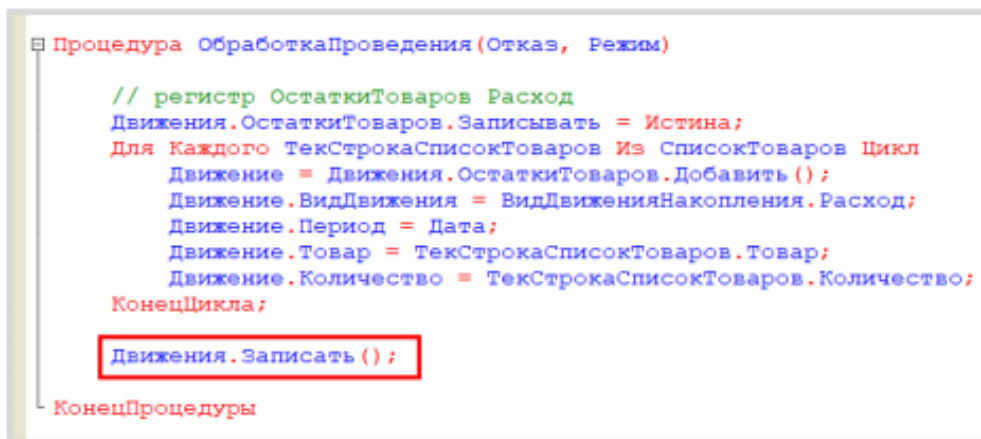
2. Проверим, появились ли в регистре остатки, значение которых меньше нуля (то есть отрицательные).

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.

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, как показано на рисунке 11.



```
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 11 – Код модуля объекта

3.19 Теперь, когда движение было сделано, можно обратиться к данным регистра накопления. Чтобы это сделать, воспользуемся конструктором запроса с обработкой результата. Этот конструктор можно открыть из

контекстного меню, открывающегося щелчком правой кнопки мыши по области модуля. Данный конструктор обязательно должен быть вызван внутри процедуры «ОбработкаПроведения», как показано на рисунке 12.

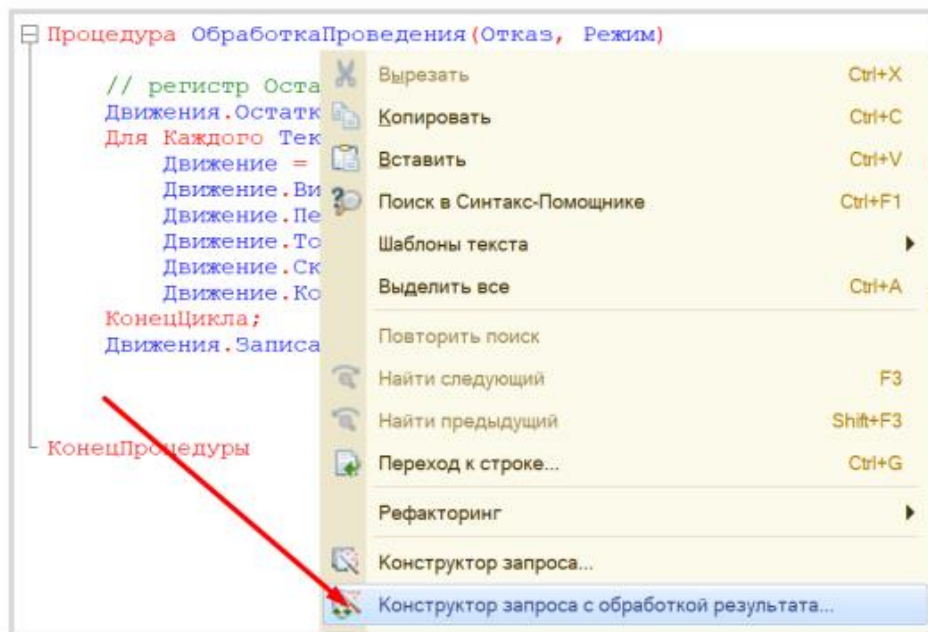


Рисунок 12 – Конструктор запроса с обработкой результата

3.20 Соглашаемся с созданием нового запроса. Открывается окно конструктора запроса с обработкой результата. Переходим на вкладку «Таблицы и поля».

В результате данное окно должно быть заполнено следующим образом, как показано на рисунке 13.

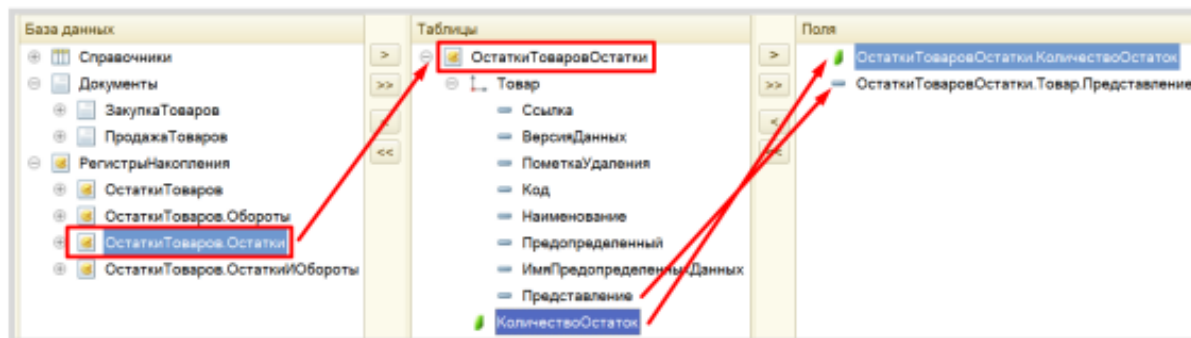


Рисунок 13 – Заполнение конструктора запроса

3.21 Перейти на вкладку «Условие» и добавим новое условие. Пусть в запрос попадут только данные с отрицательными остатками, как показано на рисунке 14.

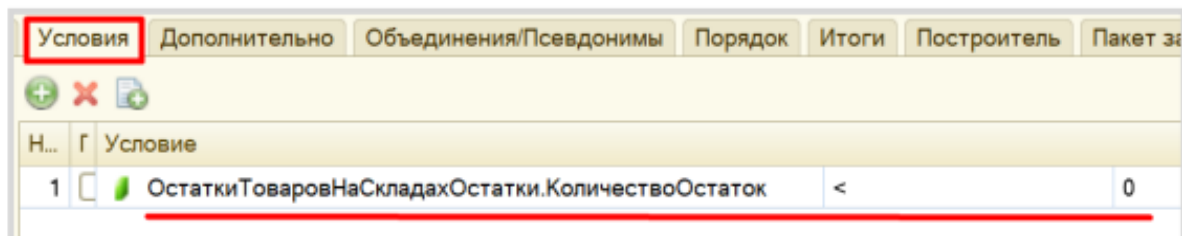


Рисунок 14 – Заполнение условия

3.22 По окончании работы с данным конструктором нажимаем на кнопку «ОК». Конструктор выдаст предупреждение об ошибке, которое нужно проигнорировать. Для корректной работы запроса следует удалить знак амперсанта (&) перед нулем в условии. Запрос должен выглядеть следующим образом, как показано на рисунке 15.

```
"ВЫБРАТЬ
| ОстаткиТоваровОстатки.КоличествоОстаток КАК КоличествоОстаток,
| ОстаткиТоваровОстатки.Товар.Представление КАК ТоварПредставление
| ИЗ
| РегистрНакопления.ОстаткиТоваров.Остатки КАК ОстаткиТоваровОстатки
| ГДЕ
| ОстаткиТоваровОстатки.КоличествоОстаток < 0"
```

Рисунок 15 – Код условия

С помощью данного запроса из базы данных можно получить отрицательные остатки по всем товарам. Но нам нет необходимости получать такую большую выборку, нужно сузить запрос до перечня товаров, перечисленных в табличной части.

3.23 Открыть конструктор запроса. Для этого следует щелкнуть в любом месте самого запроса (черный текст в двойных кавычках) правой кнопкой мыши и вызвать конструктор запроса.

3.24 Далее нужно наложить условия на виртуальную таблицу регистра накопления, как показано на рисунках 16, 17, 18, 19.

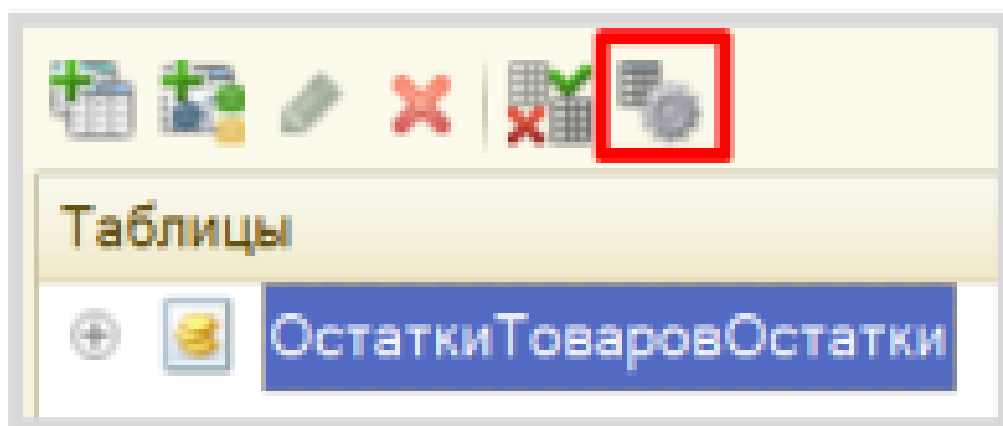


Рисунок 16 – Наложение условия на виртуальную таблицу

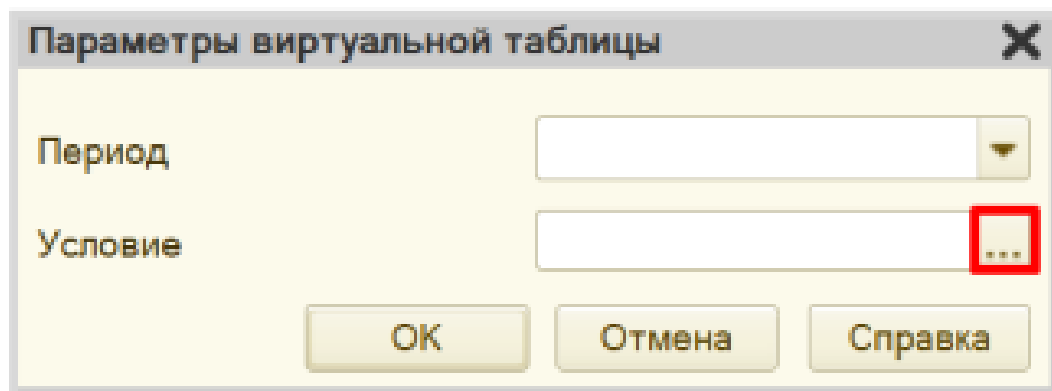


Рисунок 17 – Наложение условия на виртуальную таблицу

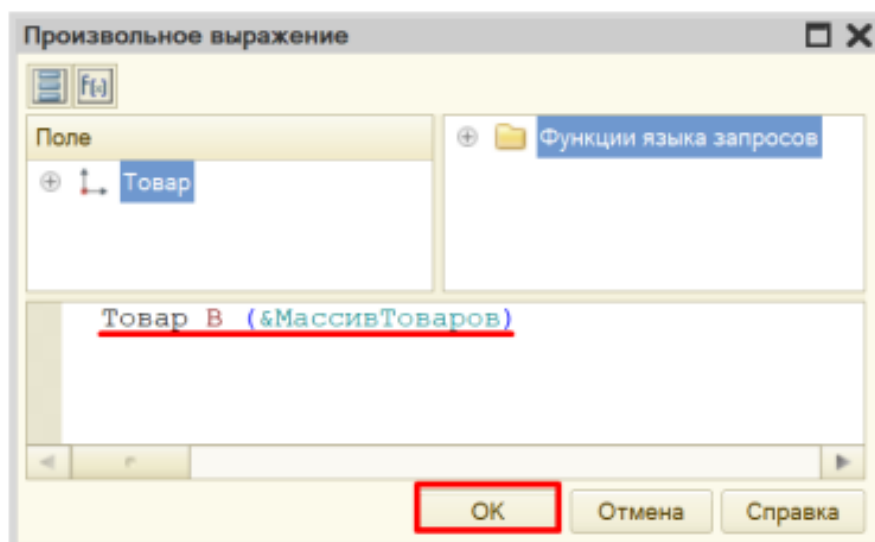


Рисунок 18 – Наложение условия на виртуальную таблицу

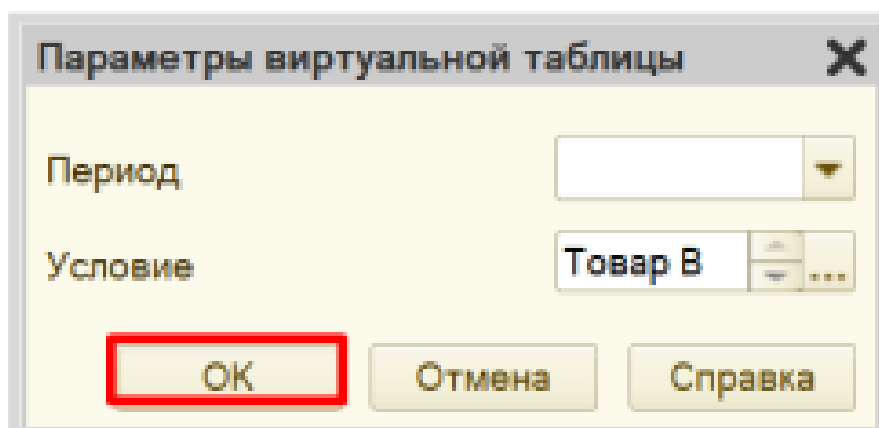


Рисунок 19 – Наложение условия на виртуальную таблицу

Данное условие поможет ограничить запрос по тем товарам, которые находятся в табличной части документа. Нажимаем на кнопку «ОК». Текст запроса изменился, как показано на рисунке 20.

```
ГДЕ
| ОстаткиТоваровОстатки.КоличествоОстаток < 0";
Запрос.УстановитьПараметр("Дата", Новый Граница( МоментВремени(), ВидГраницы.Включая));
Запрос.УстановитьПараметр("МассивТоваров", СписокТоваров.ВыгрузитьКолонку("Товар"));
```

Рисунок 20 – Текст запроса

3.25 Мы добавили параметры. Теперь запрос будет проводить поиск только по конкретному списку товаров. Осталось лишь указать эти товары сразу после текста запроса. Так же необходимо добавить параметр «Дата», как показано на рисунке 21.

```
Запрос.УстановитьПараметр("Дата",Новый Граница( МоментВремени(), ВидГраницы.Включая));  
Запрос.УстановитьПараметр("МассивТоваров", СписокТоваров.ВыгрузитьКолонку("Товар"));
```

Рисунок 21 – Добавление параметра «Дата»

3.26 Ну, и последний шаг – выдать сообщение пользователю, если запрос вернул отрицательные остатки. В первую очередь, добавить блок условия, как показано на рисунке 22.

```
РезультатЗапроса = Запрос.Выполнить();  
Если НЕ РезультатЗапроса.Пустой() Тогда  
КонецЕсли;
```

Рисунок 22 – Добавление блока условия

3.27 Внутри цикла можно попасть только в том случае, если запрос пришел не пустой, то есть если были найдены отрицательные остатки. В таком случае нужно отменить проведение документа и выдать пользователю сообщение, как показано на рисунке 23.

```

Если НЕ РезультатЗапроса.Пустой() Тогда

    Отказ = Истина;
    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

    Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
        Сообщить ("Не хватает товара " + ВыборкаДетальныеЗаписи.ТоварПредставление +
            ", в количестве " + ( - ВыборкаДетальныеЗаписи.КоличествоОстаток));
    КонечЦикла;

КонечЕсли;

```

Рисунок 23 – Добавление блока условия

3.27 Код процедуры полностью должен выглядеть следующим образом, как показано на рисунке 24.

```

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;
    КонечЦикла;
    Движения.Записать();

    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        | ОстаткиТоваровОстатки.Товар.Представление КАК ТоварПредставление,
        | ОстаткиТоваровОстатки.КоличествоОстаток КАК КоличествоОстаток
        | ИЗ
        | РегистрНакопления.ОстаткиТоваров.Остатки(&Дата, Товар В (&МассивТоваров) ) КАК ОстаткиТоваровОстатки
        | ГДЕ
        | ОстаткиТоваровОстатки.КоличествоОстаток < 0";

    Запрос.УстановитьПараметр("Дата",Новый Граница( МоментВремени(), ВидГраницы.Включая));
    Запрос.УстановитьПараметр("МассивТоваров", СписокТоваров.ВыгрузитьКолонку("Товар"));
    РезультатЗапроса = Запрос.Выполнить();

    Если НЕ РезультатЗапроса.Пустой() Тогда
        Отказ = Истина;
        ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
        Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
            Сообщить ("Не хватает товара " + ВыборкаДетальныеЗаписи.ТоварПредставление +
                ", в количестве " + ( - ВыборкаДетальныеЗаписи.КоличествоОстаток));
        КонечЦикла;
    КонечЕсли;
КонечПроцедуры

```

Рисунок 24 – Код процедуры

3.28 Теперь проверьте корректность работы системы на предмет наличия ошибок при попытке продать товара больше, чем его имеется в наличии, как показано на рисунке 25.



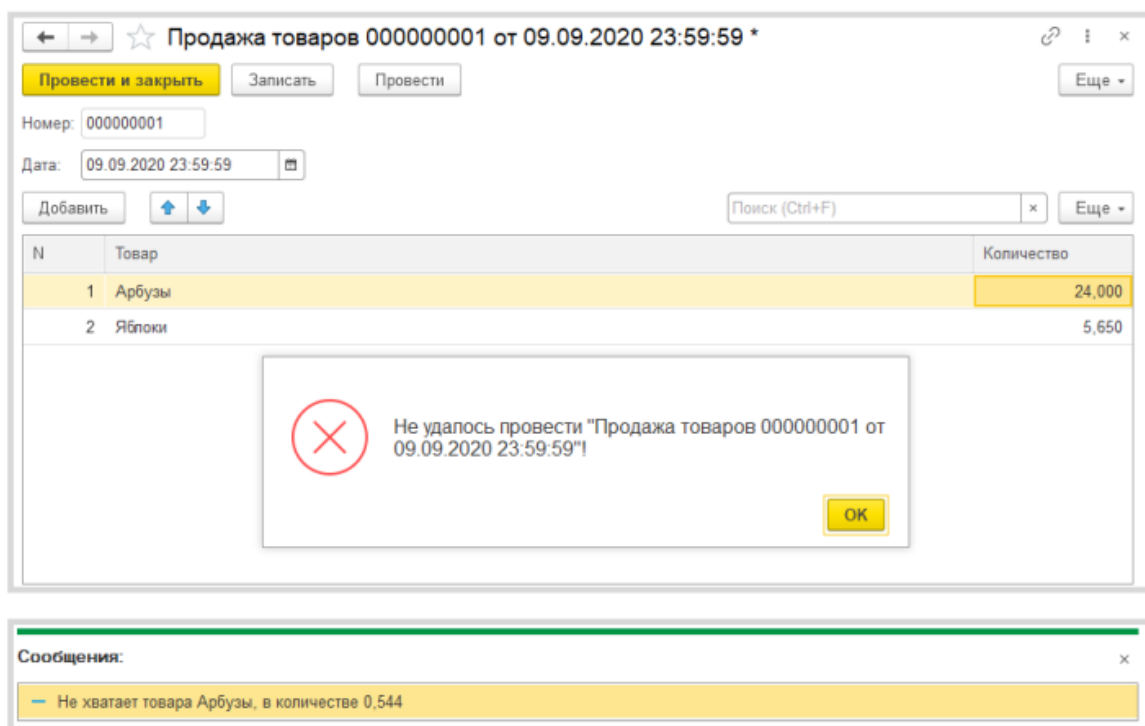


Рисунок 25 – Ошибка при проведении

3.29 Как видно из сообщения, товара не хватает ровно в том количестве, которое раньше выводилось в отчете со знаком «-» (минус). Если изменить количество на меньшее, то механизм будет работать корректно, как показано на рисунке 26.

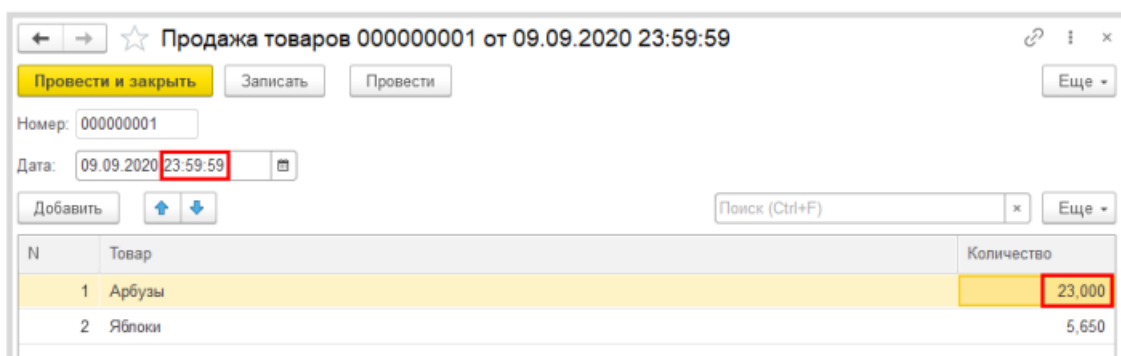


Рисунок 26 – Корректное отображение работы механизма

Также отчет будет учитывать документы, проведенные в конце дня, как показано на рисунке 27.

← → ☆ **Остатки товаров**

**Сформировать** **Выбрать вариант...** **Настройки...**

Дата отчета: ☒ **09.09.2020**

Параметры: Дата отчета: 09.09.2020

Товар	Количество
Арбузы	0,456
Яблоки	4,850

Рисунок 27 – Отчет «Остатки товаров»

## Лабораторная работа №16

### Разработка конфигурации для учета товаров. Продажа товаров с одного склада

#### 1. Задание:

Заказчик просит разработать конфигурацию для учета товаров.

Суммовой учет не ведется. Взаиморасчеты с покупателями и поставщиками не ведутся.

Учет товаров ведется в разрезе складов.

В системе необходимо регистрировать поступление товара. При поступлении товара пользователь в табличной части указывает, какие товары и в каком количестве поступили в организацию. Нужно предусмотреть учет до граммов. В шапке документа выбираем склад, куда поступают товары.

В системе следует регистрировать продажу товара. При продаже товаров указывается, какие товары были проданы и в каком количестве, с какого склада производится списание. Склад выбирается в шапке документа.

Продать товар «в минус» нельзя, то есть в момент продажи нужно проверять остаток товара.

Необходимо построить «Отчет» по остаткам товаров следующего вида:

Остатки товаров на 31.01.2020

Товар/Склад	Юг	Север	Запад	Итого
Ложка	100.000	40.000		140.000
Вилка	45.000		80.000	125.000
Поварешка		12.000	1.000	13.000

Отчет строится на конец дня, указанного пользователем. Особое внимание следует уделить последней секунде дня: документы, записанные на эту секунду, должны попадать в отчет.

## **2. Подготовка:**

- создать новую информационную базу;
- открыть информационную базу в режиме «Конфигуратор»;
- открыть окно конфигурации.

## **3 Выполнение:**

3.1 Создать справочники «Товары» и Склады».

3.2 Открыть программу в режиме «1С:Предприятие» и заполнить справочники элементами.

3.3 Добавить документ «ПоступлениеТоваров».

3.4 Перейти на вкладку «Данные», создать табличную часть «СписокТоваров», в табличной части добавить реквизиты «Товар» с ссылкой на соответствующий справочник, а также реквизит «Количество» с типом «Число» и точностью «3».

3.5 Добавить реквизит «Склад» с ссылкой на соответствующий справочник. Выбрать «Представление», найти «Проверка заполнения», выбрать «Выдавать ошибку».

3.6 В режиме «1С:Предприятие» добавить склады Северный, Южный, Западный. Провести несколько документов с разных складов.

Продажа товара осуществляется аналогично получению, следовательно, документ по структуре будет точно таким же, как документ «ПолучениеТоваров».

3.7 Создать копированием предыдущего документа документ «ПродажаТоваров».

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

Если создать новый документ «ПродажаТовара» и попытаться продать со склада больше товаров, чем на складе имеется в данный момент, то система не сможет этого предотвратить, поскольку учет остатков никак не ведется. Для

начала следует каким-то образом вести подсчет остатков товаров на складах. Для этого нам потребуется регистр накопления.

3.8 Добавить новый регистр накопления «ОстаткиТоваровНаСкладах». Вид данного регистра – «Остатки».

3.9 Перейти на вкладку «Данные».

Заполнение данного окна проще всего начинать с добавления ресурса. Чтобы понять, что использовать в качестве ресурса, следует задать вопрос: «Что мы хотим накапливать/считать в данном регистре?». Мы хотим считать количество. Следовательно, количество и будет являться ресурсом. Тип данного реквизита – «Число». Точность – «3», поскольку в реквизит должно попадать количество с точностью до грамм.

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

3.10 Определить оба документа в качестве документов-регистраторов на вкладке «Регистраторы».

3.11 Далее для каждого из этих документов нужно описать процедуру копирования данных в регистр накопления.

Начнем с документа «ПолучениеТоваров», откроем окно редактирования данного документа на вкладке «Движения». Воспользуемся конструктором движений.

Поскольку получение товара должно увеличивать количество товаров на складе, то тип движения регистра следует 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, как показано на рисунке 1.

Рисунок 1 – Заполнение конструктора движения регистров

3.12 Аналогичные действия нужно проделать с документом «ПродажаТоваров», только тип движения регистра: Расход.

В первую очередь, необходимо перепровести (провести заново) созданный документ «Поступление товаров», а также создать и провести хотя бы один документ «Продажа товаров». Без проведения документов данные не будут скопированы в регистр накопления.

3.13 Проверить результат отображения, как показано на рисунке 2.

Рисунок 2 – Результат движения товаров

К сожалению, регистра накопления недостаточно для того, чтобы вести учет отрицательных остатков. Нужно описать алгоритм работы документа «ПродажаТоваров».

3.14 Отроем модуль объекта документа «ПродажаТоваров» и дополним процедуру «ОбработкаПроведения».

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.

3.15 Чтобы сделать движение данных их 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, как показано на рисунке 3.

```
Процедура ОбработкаПроведения(Отказ, Режим)
// регистр ОстаткиТоваровНаСкладах Расход
Движения.ОстаткиТоваровНаСкладах.Записывать = Истина;
Для Каждого ТекСтрокаСписокТоваров Из СписокТоваров Цикл
    Движение = Движения.ОстаткиТоваровНаСкладах.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Товар = ТекСтрокаСписокТоваров.Товар;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаСписокТоваров.Количество;
КонецЦикла;
Движения.Записать();
КонецПроцедуры
```

Рисунок 3 – Код движения данных

3.16 Теперь, когда движение было сделано, можно обратиться к данным регистра накопления. Чтобы это сделать, воспользуемся конструктором запроса с обработкой результата. Этот конструктор можно открыть из контекстного меню щелчком правой кнопки мыши по области модуля. Данный конструктор обязательно должен быть вызван внутри процедуры «ОбработкаПроведения».

3.17 Соглашаемся с созданием нового запроса. Открывается окно конструктора запроса с обработкой результата. Переходим на вкладку «Таблицы и поля». Заполнить таблицу и поля, как показано на рисунке 4.

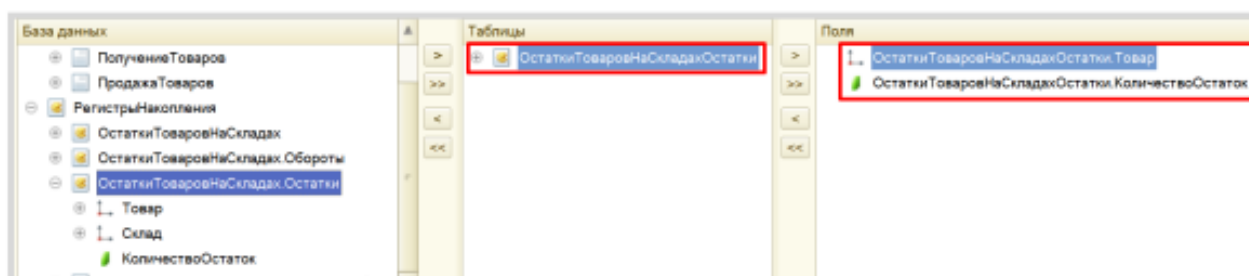


Рисунок 4 – Заполнение таблицы запроса

3.18 Переходим на вкладку «Условие» и добавим новое условие. Пусть в запрос попадут только данные с отрицательными остатками, как показано на рисунке 5.

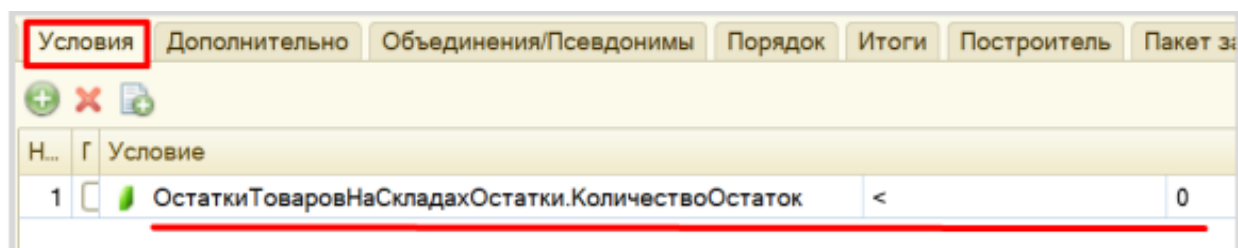


Рисунок 5 – Добавление нового условия



По окончании работы с данным конструктором нажимаем на кнопку «ОК».

3.19 Конструктор выдаст предупреждение об ошибке, которое следует проигнорировать. Для корректной работы запроса нужно удалить знак амперсанта (&) перед нулем в условии. Запрос должен выглядеть следующим образом, как показано на рисунке 6.

```
Запрос.Текст =  
"ВЫБРАТЬ  
|   ОстаткиТоваровНаСкладахОстатки.Товар КАК Товар,  
|   ОстаткиТоваровНаСкладахОстатки.КоличествоОстаток КАК КоличествоОстаток  
| ИЗ  
|   РегистрНакопления.ОстаткиТоваровНаСкладах.Остатки КАК ОстаткиТоваровНаСкладахОстатки  
| ГДЕ  
|   ОстаткиТоваровНаСкладахОстатки.КоличествоОстаток < 0";
```

Рисунок 6 – Запрос

С помощью данного запроса из базы данных можно получить отрицательные остатки по всем товарам со всех складов.

Но нам нет необходимости получать такую большую выборку, нужно сузить запрос до склада, который указан в шапке документа и товаров, перечисленных в табличной части.

3.20 Откроем конструктор запроса. Для этого нужно щелкнуть в любом месте самого запроса (черный текст в двойных кавычках) правой кнопкой мыши и вызвать конструктор запроса.

3.21 Далее следует наложить условия на виртуальную таблицу регистра накопления, как показано на рисунках 7, 8, 9, 10.



Рисунок 7 – Наложение условия

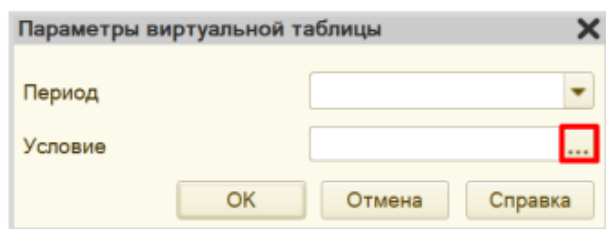


Рисунок 8 – Наложение условия

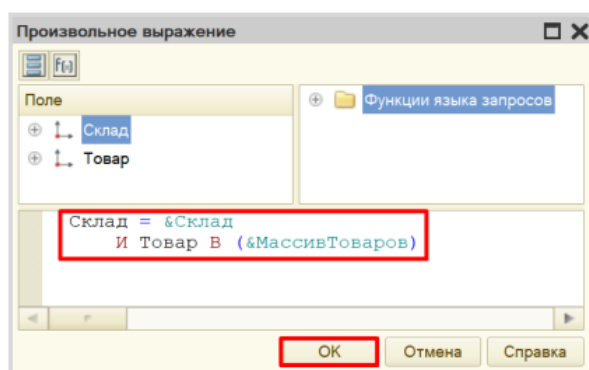


Рисунок 9 – Наложение условия

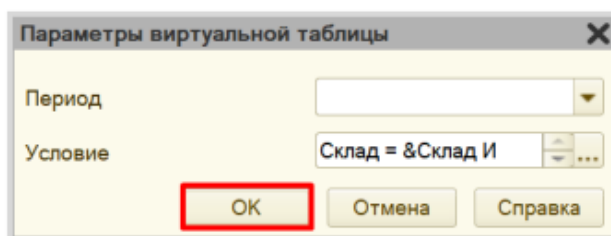


Рисунок 10 – Наложение условия

Данное условие поможет ограничить запрос по складу и по тем товарам, которые находятся в табличной части документа. Нажимаем на кнопку ОК.

3.22 Текст запроса изменился, как показано на рисунке 11.

```

Запрос.Текст =
"ВЫБРАТЬ
| ОстаткиТоваровНаСкладах.Остатки.Товар КАК Товар,
| ОстаткиТоваровНаСкладах.Остатки.КоличествоОстаток КАК КоличествоОстаток
| ИЗ
| РегистрНакопления.ОстаткиТоваровНаСкладах.Остатки (
|
|         Склад = &Склад
|         И Товар В (&МассивТоваров)) КАК ОстаткиТоваровНаСкладах.Остатки
| ГДЕ
| ОстаткиТоваровНаСкладах.Остатки.КоличествоОстаток < 0";

```

Рисунок 11 – Запрос

Мы добавили параметры. Теперь запрос будет проводить поиск только по конкретному складу и конкретному списку товаров.

3.23 Осталось лишь указать этот склад и товары сразу после текста запроса, как показано на рисунке 12.

```

Запрос.УстановитьПараметр("Склад", Склад);
Запрос.УстановитьПараметр("МассивТоваров", СписокТоваров.ВыгрузитьКолонку("Товар"));

```

Рисунок 12 – Добавление запроса

3.23 Ну, и последний шаг – выдать сообщение пользователю, если запрос вернул отрицательные остатки. В первую очередь, добавим блок условия, как показано на рисунке 13.

```

РезультатЗапроса = Запрос.Выполнить();
Если НЕ РезультатЗапроса.Пустой() Тогда
КонецЕсли;

```

Рисунок 13 – Добавление сообщения

Внутри цикла можно попасть только в том случае, если запрос пришел не пустой, то есть если были найдены отрицательные остатки. В таком случае нужно отменить проведение документа и выдать пользователю сообщение.

3.24 Переместим весь оставшийся код внутрь данного цикла, а также сократим название переменной «ВыборкаДетальныеЗаписи» на «Выборка» для удобства, как показано на рисунке 14.

```

РезультатЗапроса = Запрос.Выполнить();
Если НЕ РезультатЗапроса.Пустой() Тогда
    Отказ = Истина;
    Выборка = РезультатЗапроса.Выбрать();
    Пока Выборка.Следующий() Цикл
        Сообщить("На складе " + Склад + " не хватает " +
            Выборка.КоличествоОстаток*(-1) + " шт. товаров " + Выборка.Товар);
    КонецЦикла;
КонецЕсли;

```

Рисунок 14 – Изменение кода

3.25 Код процедуры полностью должен выглядеть следующим образом, как показано на рисунке 15.

```

Процедура ОбработкаПроведения(Отказ, Режим)
// регистр ОстаткиТоваровНаСкладах Расход
Движения.ОстаткиТоваровНаСкладах.Записывать = Истина;
Для Каждого ТекСтрокаСписокТоваров Из СписокТоваров Цикл
    Движение = Движения.ОстаткиТоваровНаСкладах.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Товар = ТекСтрокаСписокТоваров.Товар;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаСписокТоваров.Количество;
КонецЦикла;
Движения.Записать();

// контроль отрицательных остатков
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫВЕРАТЬ
| ОстаткиТоваровНаСкладахОстатки.Товар КАК Товар,
| ОстаткиТоваровНаСкладахОстатки.КоличествоОстаток КАК КоличествоОстаток
| ИЗ
| РегистрНакопления.ОстаткиТоваровНаСкладах.Остатки (
|
|         Склад = &Склад
|         И Товар В (&МассивТоваров)) КАК ОстаткиТоваровНаСкладахОстатки
| ГДЕ
| ОстаткиТоваровНаСкладахОстатки.КоличествоОстаток < 0";

Запрос.УстановитьПараметр("Склад", Склад);
Запрос.УстановитьПараметр("МассивТоваров", СписокТоваров.ВыгрузитьКолонку("Товар"));

РезультатЗапроса = Запрос.Выполнить();
Если НЕ РезультатЗапроса.Пустой() Тогда
    Отказ = Истина;
    Выборка = РезультатЗапроса.Выбрать();
    Пока Выборка.Следующий() Цикл
        Сообщить("На складе " + Склад + " не хватает " +
            Выборка.КоличествоОстаток*(-1) + " шт. товаров " + Выборка.Товар);
    КонецЦикла;
КонецЕсли;
КонецПроцедуры

```

Рисунок 15 – Код процедуры

3.26 Переходим в режим «1С:Предприятие». Добавим новый документ «Продажа товаров» так, чтобы хотя бы одного из товаров не хватало на складе.

Если все было сделано правильно, и вы пытаетесь продать товаров больше, чем имеется на складе, то система выдаст ошибку, как показано на рисунке 16.

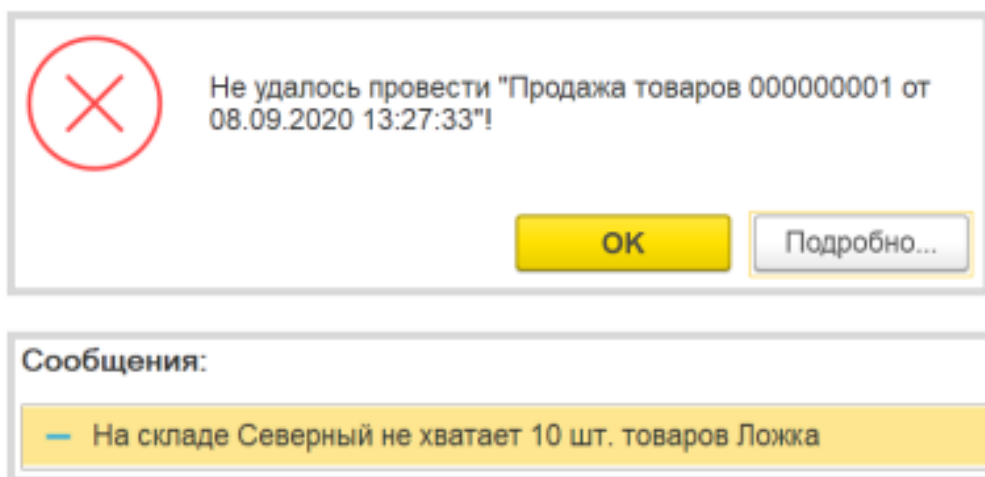


Рисунок 16 – Ошибка

3.27 Добавим отчет «ОтчетПоОстаткамТоваров». Воспользуемся схемой компоновки данных.

3.28 Заполнить таблицу и поля, как показано на рисунке 17.

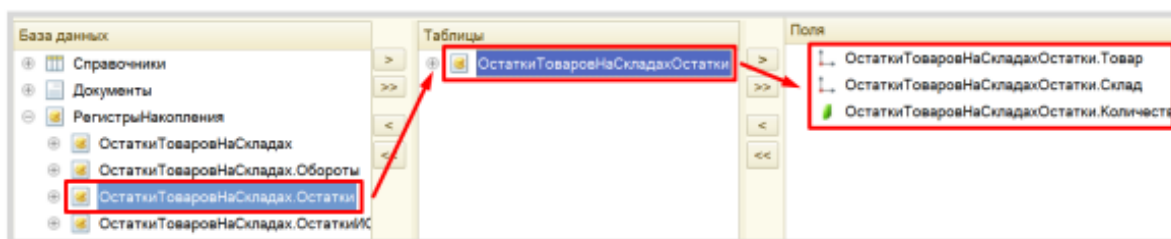


Рисунок 17 – Заполнение конструктора запроса

3.29 Переходим на вкладку «Ресурсы» и устанавливаем реквизит «КоличествоОстаток» в качестве ресурса: это позволит нам в отчете получать итоговые (просуммированные) значения.

3.30 Из условия следует, что отчет должен включать документы, записанные на последнюю секунду дня. При использовании стандартных методов такие документы в отчет попадать не будут. Поэтому нужно добавить новый параметр «ДатаОтчета» на соответствующей вкладке, как показано на рисунке 18.

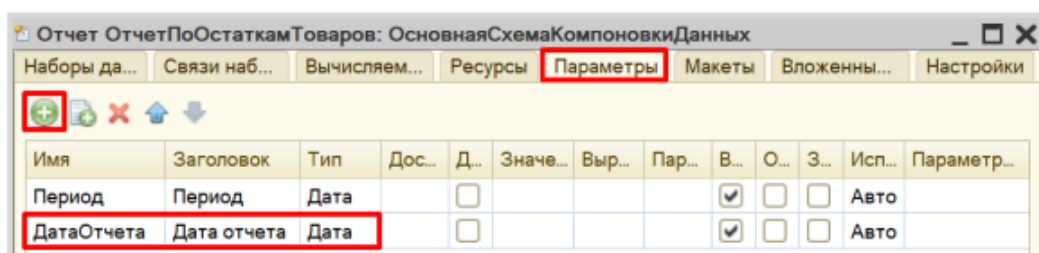


Рисунок 18 – Редактирование параметров

3.31 Чтобы у пользователя была возможность выбирать только даты, без указания секунд, нужно настроить формат редактирования параметра «ДатаОтчета», как показано на рисунках 19, 20, 21.

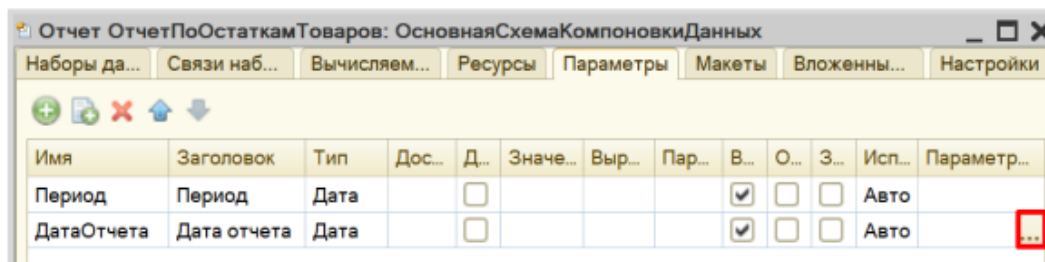


Рисунок 19 – Редактирование параметров

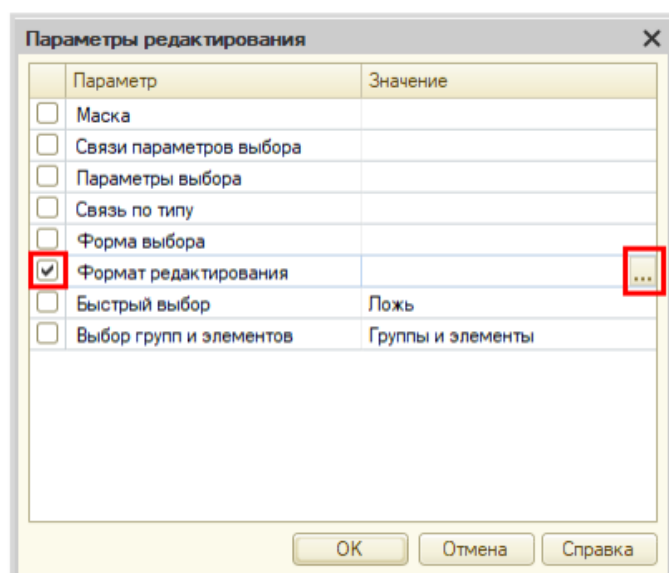


Рисунок 20 – Редактирование параметров

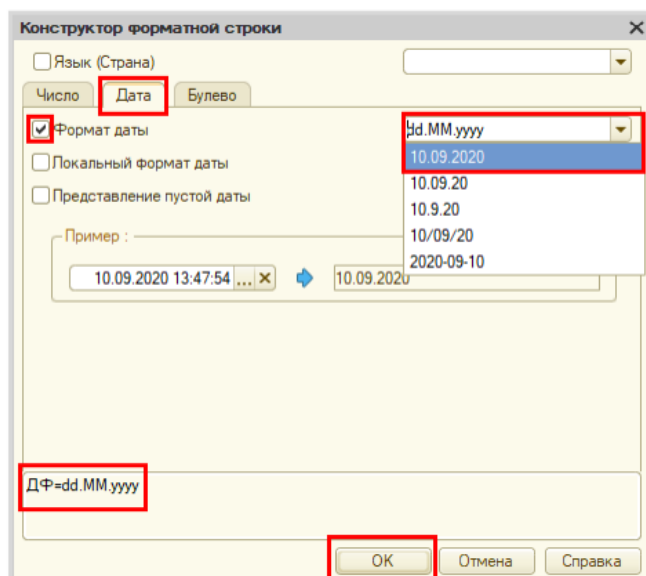


Рисунок 21 – Редактирование параметров

3.32 После нажатия кнопки «ОК» нужно настроить стандартный параметр «Период» для корректного учета последней секунды дня:

- во-первых, этот параметр должен быть недоступен пользователю, так как носит вычислительный характер;

- во-вторых, для корректного расчета требуется написать выражение для стандартного параметра «Период», как показано на рисунке 22.

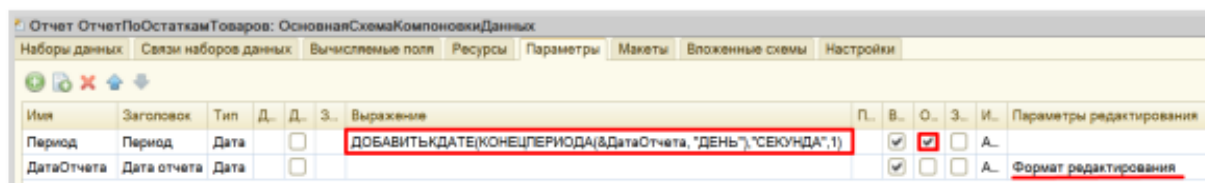


Рисунок 22 – Редактирование параметров

3.33 Теперь переходим на вкладку «Настройки» для оформления внешнего вида отчета. Воспользуемся конструктором настроек отчета.

3.34 Построить отчет в виде Таблицы.

3.35 Далее следует выбрать ресурс, который будет отображен в таблице. В данном случае реквизит у нас всего один, его и выбираем.

3.36 На следующем шаге необходимо определить те реквизиты, которыми будут заполняться колонки и строки таблицы: Строки – Товар, Колонки – Склад.

3.37 Ну, и на последнем шаге нужно установить сортировку. Установим сортировку по товарам по возрастанию (то есть по алфавиту).

3.38 Чтобы у пользователя была возможность выбирать день, на который он хочет построить отчет, нужно включить параметр «ДатаОтчета» в пользовательские настройки.

3.39 Запустим систему в режиме «1С:Предприятие». Добавьте еще несколько документов по получению и продаже товаров, чтобы убедиться, что отчет работает корректно, как показано на рисунке 23.



←

→

☆

Отчет по остаткам товаров

🖨

🔄

Сформировать

Выбрать вариант...

Настройки...

Дата отчета:

✓

08.09.2020

☰

Параметры:   Дата отчета: 08.09.2020

Товар	Западный Количество Остаток	Северный Количество Остаток	Южный Количество Остаток	Итого Количество Остаток
Вилка		8,000	15,000	23,000
Ложка	13,000	10,000	10,000	33,000
Поварешка		8,000		8,000
Итого	13,000	26,000	25,000	64,000

Рисунок 23 – Отчет

## **Самостоятельная работа (обязательно для выполнения)**

### **1 Задание:**

При первом использовании нового кассового аппарата в торговой точке требуется провести его инициализацию. Без ввода базовых настроек работа кассы невозможна. Необходимо сделать так, чтобы при первом запуске кассовой программы автоматически открывалась форма с обязательными параметрами (название магазина, номер кассы). Признаком первого запуска будет служить константа со значением «Ложь». Для упрощения задачи форму следует открыть с информационным сообщением «Первый запуск кассовой программы. Требуется настройка».

### **2 Задание:**

Заказчик просит разработать конфигурацию для учета ингредиентов на кухне ресторана.

Многоскладской учет не ведется. Суммовой учет не ведется. Взаиморасчеты с поставщиками не ведутся.

В системе необходимо регистрировать «Поступление ингредиентов». При поступлении пользователь в табличной части указывает, какие ингредиенты и в каком количестве поступили на кухню.

Необходимо предусмотреть учет с точностью до грамма для сыпучих продуктов и до миллилитра для жидкостей.

В системе следует регистрировать «Списание ингредиентов» (на приготовление блюд). При списании указывается, какие ингредиенты и в каком количестве были израсходованы.

Израсходовать ингредиент «в минус» нельзя, в момент списания необходимо проверять остаток.

Нужно построить «Отчет по остаткам ингредиентов».

Форма отчета:

### Остатки ингредиентов на 15.03.2024

Ингредиент	Остаток
Мука пшеничная	25.450 кг
Сахар	12.800 кг
Соль	5.100 кг
Масло подсолнечное	18.750 л
Яйца куриные	120 шт

### 3 Задание:

Заказчик просит разработать конфигурацию для учета лекарственных средств в аптечной сети.

Суммовой учет не ведется. Взаиморасчеты с поставщиками и покупателями не ведутся.

Учет препаратов ведется в разрезе аптек-филиалов.

В системе необходимо регистрировать «Поступление препаратов». При поступлении в табличной части указывается наименование лекарства, количество и форма выпуска (таблетки, флаконы, граммы для мазей). В шапке документа выбирается аптека, в которую поступили препараты. Учет количества таблеток ведется поштучно, мазей и жидкостей — в граммах и миллилитрах.

В системе следует регистрировать «Реализацию препаратов». При продаже указывается, какие препараты, в каком количестве и с какой аптеки были проданы (аптека выбирается в шапке). Продать препарат «в минус» нельзя — в момент продажи система проверяет остаток на выбранной аптеке.

Необходимо построить «Отчет по остаткам лекарственных средств» следующего вида:

### Остатки препаратов на 15.03.2024

Препарат / Аптека	Центральная	Северная	Западная	Итого
Парацетамол	150 таб.	80 таб.		230 таб.
Йод 10мл	45 фл.		30 фл.	75 фл.
Мазь цинковая		120 гр.	50 гр.	170 гр.

Отчет формируется на конец указанной пользователем даты. Особое внимание следует уделить последней секунде выбранной даты: документы, зарегистрированные в эту секунду, должны быть учтены в отчете.