

Процедура ОбработкаПроведения(Отказ, Режим)

```
// регистр ОстаткиТоваров Расход
Движения.ОстаткиТоваров.Записывать = Истина;
Для Каждого ТекСтрокаТовары Из Товары Цикл
    Движение = Движения.ОстаткиТоваров.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.НаименованиеТовара = ТекСтрокаТовары.НаименованиеТовара;
    Движение.Количество = ТекСтрокаТовары.Количество;
КонецЦикла;

// проведем контроль отрицательных остатков (чтобы можно было выяснить, не ушли ли мы в минус)
// сначала запишем все движения, которые мы провели, чтобы получить предварительный итог наших
действий

Движения.Записать();
// контролировать будем только при оперативном режиме проведения
Если Режим = РежимПроведенияДокумента.Оперативный Тогда
    // выбираем ОТРИЦАТЕЛЬНЫЕ остатки по товарам, находящимся в проводимом документе

    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |     ОстаткиТоваровОстатки.НаименованиеТовара КАК НаименованиеТовара,
        |     ОстаткиТоваровОстатки.КоличествоОстаток КАК Количество
        |ИЗ
        |     РегистрНакопления.ОстаткиТоваров.Остатки(
        |         ,
        |         НаименованиеТовара В
        |             (ВЫБРАТЬ
        |                 РеализацияТоваровТовары.НаименованиеТовара КАК
        |                 НаименованиеТовара
        |             ИЗ
        |                 Документ.РеализацияТоваров.Товары КАК
        |                 РеализацияТоваровТовары
        |             ГДЕ
        |                 РеализацияТоваровТовары.Ссылка = &Ссылка)) КАК
        |                 ОстаткиТоваровОстатки
        |ГДЕ
        |     ОстаткиТоваровОстатки.КоличествоОстаток < 0";

    Запрос.УстановитьПараметр("Ссылка", Ссылка);

    РезультатЗапроса = Запрос.Выполнить();

    // Если наш результат запроса не пустой (есть отрицательные остатки)
    Если РезультатЗапроса.Пустой()= Ложь Тогда
        // запрещаем проведение документа
        Отказ = Истина;
        // объясняем пользователю, почему не проводится
        ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
        Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
            Сообщить("Для реализации не хватает товара " +
                ВыборкаДетальныеЗаписи.НаименованиеТовара + " в количестве " +
                (-ВыборкаДетальныеЗаписи.Количество) + " единиц");
        КонецЦикла;
```

КонецЕсли;
КонецЕсли;

// Проведем проверку проведения документа
Если Не Отказ Тогда

 // пересчитаем себестоимость по цене поступления (по среднему)

 //Запишем движение по регистру Себестоимость (расход)
 Движения.Себестоимость.Записывать = Истина;
 //Запишем движение по регистру Продажи
 Движения.Продажи.Записывать = Истина;

 // 1) в подзапросе выбираем товары из нашего документа РеализацияТоваров
 // 2) по этому списку отбираем товары в регистре СебестоимостьОстатки (остатки перед
 продажей товара), их кол-во и стоимость

 // 3) на момент времени документа Реализация Товаров

 Запрос = Новый Запрос;

 Запрос.Текст =

 "ВЫБРАТЬ

 | СебестоимостьОстатки.Номенклатура КАК Номенклатура,
 | СебестоимостьОстатки.КоличествоОстаток КАК Количество,
 | СебестоимостьОстатки.СуммаОстаток КАК Сумма

 |ИЗ

 | РегистрНакопления.Себестоимость.Остатки(
 | &МоментВремени,
 | Номенклатура В
 | (ВЫБРАТЬ

 | РеализацияТоваровТовары.НаименованиеТовара КАК Товар

 | ИЗ

 | Документ.РеализацияТоваров.Товары КАК
 | РеализацияТоваровТовары

 | ГДЕ

 | РеализацияТоваровТовары.Ссылка = &Ссылка)) КАК
 | СебестоимостьОстатки";

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

 Запрос.УстановитьПараметр("Ссылка", Ссылка);

 РезультатЗапроса = Запрос.Выполнить();

 ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

 СписаннаяСебестоимость = 0; // задаем переменную для бухгалтерских проводок

 // 4) считаем себестоимость единицы товара по среднему по остаткам на момент реализации

 Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

 Если ВыборкаДетальныеЗаписи.Количество <> 0 Тогда

 СебестоимостьЕдиницы = ВыборкаДетальныеЗаписи.Сумма /
 ВыборкаДетальныеЗаписи.Количество;

 Иначе

 СебестоимостьЕдиницы = 0;

 КонецЕсли;

 // 5) записываем движение в наш регистр Себестоимость по цене себестоимости
 //(поступления), но по количеству товара из табличной части РеализацииТовара

```

Движение = Движения.Себестоимость.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
// берем строку табличной части документа
СтрокаТабЧасти = Товары.Найти(ВыборкаДетальныеЗаписи.Номенклатура,
"НаименованиеТовара");
Движение.Номенклатура = СтрокаТабЧасти.НаименованиеТовара;
Движение.Количество = СтрокаТабЧасти.Количество;
Движение.Сумма = СтрокаТабЧасти.Количество * СебестоимостьЕдиницы;

// 6) записываем движение в регистр Продажи
Движение = Движения.Продажи.Добавить();
Движение.Период = Дата;
Движение.Номенклатура = СтрокаТабЧасти.НаименованиеТовара;
Движение.Клиент = Покупатель;
Движение.Количество = СтрокаТабЧасти.Количество;
Движение.Сумма = СтрокаТабЧасти.Сумма;
Движение.Себестоимость = СтрокаТабЧасти.Количество * СебестоимостьЕдиницы;

// 7) считаем Списанную себестоимость по документу для бухгалтерских проводок
СписаннаяСебестоимость = СписаннаяСебестоимость + Движение.Себестоимость;

КонецЦикла;

КонецЕсли;

// регистр БухгалтерскиеПроводки
Движения.БухгалтерскиеПроводки.Записывать = Истина;

// Первая проводка - Получаем выручку от покупателя
Движение = Движения.БухгалтерскиеПроводки.Добавить();
Движение.СчетДт = ПланыСчетов.Основной.РасчетыСПокупателями;
Движение.СчетКт = ПланыСчетов.Основной.Выручка;
Движение.Период = Дата;
Движение.Сумма = СуммаДокумента;

// Вторая проводка - Корректируем выручку на сумму себестоимость
Движение = Движения.БухгалтерскиеПроводки.Добавить();
Движение.СчетДт = ПланыСчетов.Основной.СебестоимостьПродаж;
Движение.СчетКт = ПланыСчетов.Основной.Товары;
Движение.Период = Дата;
Движение.Сумма = СписаннаяСебестоимость;

КонецПроцедуры

```