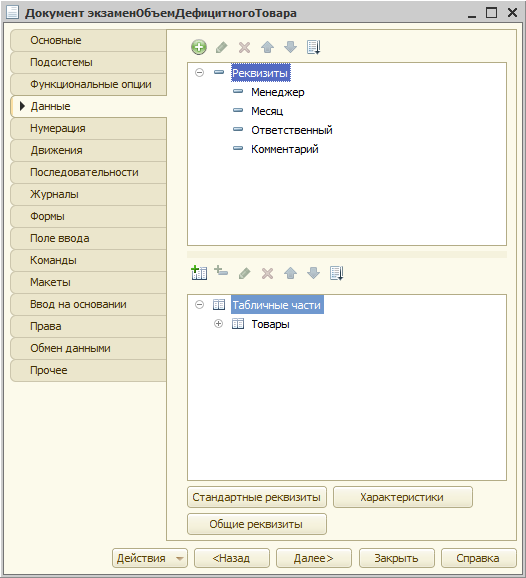
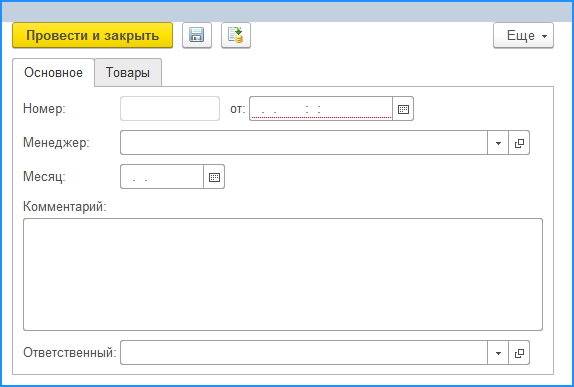
Создаем документ экзаменОбъемДефицитногоТовара, включаем в подсистему Планирование, запрещаем оперативное проведение, выключаем автоматическое удаление движений при отмене проведения.



Создаем форму



Подключим документ к подсистеме БСП отчеты, а также реализуем пересчет упаковок. Модуль формы

&НаКлиенте

Перем КэшированныеЗначения; //используется механизмом обработки изменения реквизитов ТЧ

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

// МенюОтчеты

МенюОтчеты.ПриСозданииНаСервере(ЭтотОбъект, Элементы.ПодменюОтчеты);

// Конец МенюОтчеты

НоменклатураСервер.УстановитьУсловноеОформлениеЕдиницИзмерения(ЭтаФорма);

КонецПроцедуры

// МенюОтчеты

&НаКлиенте

Процедура Подключаемый\_ВыполнитьКомандуОтчет(Команда)

МенюОтчетыКлиент.ВыполнитьПодключаемуюКомандуОтчет(Команда, ЭтотОбъект, Объект);

КонецПроцедуры

// Конец МенюОтчеты

&НаКлиенте

Процедура ТоварыКоличествоУпаковокПриИзменении(Элемент)

ДействияПриИзмененииУпаковкиКоличества();

КонецПроцедуры

&НаКлиенте

Процедура ТоварыУпаковкаПриИзменении(Элемент)

ДействияПриИзмененииУпаковкиКоличества();

КонецПроцедуры

&НаКлиенте

Процедура ДействияПриИзмененииУпаковкиКоличества()

ТекущаяСтрока = Элементы.Товары.ТекущиеДанные;

СтруктураДействий = Новый Структура;

СтруктураДействий.Вставить("ПересчитатьКоличествоЕдиниц");

ОбработкаТабличнойЧастиКлиент.ОбработатьСтрокуТЧ(ТекущаяСтрока, СтруктураДействий, КэшированныеЗначения);

КонецПроцедуры // ДействияПриИзмененииУпаковкиКоличества()

&НаКлиенте

Процедура ТоварыНоменклатураПриИзменении(Элемент)

ТекущаяСтрока = Элементы.Товары.ТекущиеДанные;

СтруктураДействий = Новый Структура;

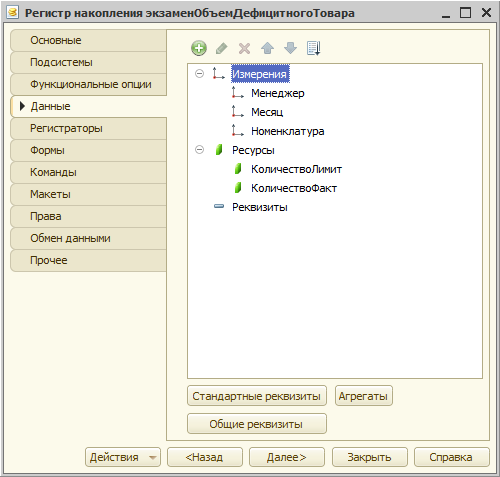
СтруктураДействий.Вставить("ПроверитьЗаполнитьУпаковкуПоВладельцу", ТекущаяСтрока.Упаковка);

СтруктураДействий.Вставить("ПересчитатьКоличествоЕдиниц");

ОбработкаТабличнойЧастиКлиент.ОбработатьСтрокуТЧ(ТекущаяСтрока, СтруктураДействий, КэшированныеЗначения);

КонецПроцедуры

Создадим оборотный регистр накопления экзаменОбъемДефицитногоТовара. Если использовать остаточный регистр, то нужно будет предусмотреть алгоритм сворачивания итогов регистра в 0, так как движения прихода будут каждый месяц, а расхода может и не быть.



## Алгоритм проведения документа экзаменОбъемДефицитногоТовара по РН экзаменОбъемДефицитногоТовара

Модуль объекта документа

Процедура ОбработкаЗаполнения(ДанныеЗаполнения, ТекстЗаполнения, СтандартнаяОбработка)

Ответственный = Пользователи.ТекущийПользователь();

Месяц = НачалоМесяца(ТекущаяДатаСеанса());

КонецПроцедуры

Процедура ПриКопировании(ОбъектКопирования)

Ответственный = Пользователи.ТекущийПользователь();

Месяц = НачалоМесяца(ТекущаяДатаСеанса());

КонецПроцедуры

Процедура ПриУстановкеНовогоНомера(СтандартнаяОбработка, Префикс)

ПрефиксацияОбъектовСобытия.УстановитьПрефиксИнформационнойБазы(ЭтотОбъект, СтандартнаяОбработка, Префикс);

КонецПроцедуры

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

Если ОбменДанными.Загрузка Тогда

Возврат;

КонецЕсли;

ОбновлениеИнформационнойБазы.ПроверитьОбъектОбработан(ЭтотОбъект);

ПроведениеСервер.УстановитьРежимПроведения(ЭтотОбъект, РежимЗаписи, РежимПроведения);

ДополнительныеСвойства.Вставить("ЭтоНовый", ЭтоНовый());

ДополнительныеСвойства.Вставить("РежимЗаписи", РежимЗаписи);

КонецПроцедуры

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

ПроведениеСервер.ИнициализироватьДополнительныеСвойстваДляПроведения(Ссылка, ДополнительныеСвойства, РежимПроведения);

Документы.экзаменОбъемДефицитногоТовара.ИнициализироватьДанныеДокумента(Ссылка, ДополнительныеСвойства);

ПроведениеСервер.ПодготовитьНаборыЗаписейКРегистрацииДвижений(ЭтотОбъект);

ОтразитьОбъемДефицитногоТовара();

ПроведениеСервер.ЗаписатьНаборыЗаписей(ЭтотОбъект);

ПроведениеСервер.ОчиститьДополнительныеСвойстваДляПроведения(ДополнительныеСвойства);

КонецПроцедуры

Процедура ОбработкаУдаленияПроведения(Отказ)

ПроведениеСервер.ИнициализироватьДополнительныеСвойстваДляПроведения(Ссылка, ДополнительныеСвойства);

ПроведениеСервер.ПодготовитьНаборыЗаписейКРегистрацииДвижений(ЭтотОбъект);

ПроведениеСервер.ЗаписатьНаборыЗаписей(ЭтотОбъект);

ПроведениеСервер.ОчиститьДополнительныеСвойстваДляПроведения(ДополнительныеСвойства);

КонецПроцедуры

Процедура ОтразитьОбъемДефицитногоТовара()

Таблица = ДополнительныеСвойства.ТаблицыДляДвижений.ТаблицаэкзаменОбъемДефицитногоТовара;

Если Таблица.Количество() = 0 Тогда

Возврат;

КонецЕсли;

ДвиженияОбъемДефицитногоТовара = Движения.экзаменОбъемДефицитногоТовара;

ДвиженияОбъемДефицитногоТовара.Записывать = Истина;

ДвиженияОбъемДефицитногоТовара.Загрузить(Таблица);

КонецПроцедуры // ОтразитьЦеныНоменклатуры()

Модуль менеджера документа

#Область Проведение

Процедура ЗаполнитьПараметрыИнициализации(Запрос, ДокументСсылка)

Запрос.МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;

Запрос.УстановитьПараметр("Ссылка", ДокументСсылка);

КонецПроцедуры

Процедура ИнициализироватьДанныеДокумента(ДокументСсылка, ДополнительныеСвойства, Регистры = Неопределено) Экспорт

Запрос = Новый Запрос;

ЗаполнитьПараметрыИнициализации(Запрос, ДокументСсылка);

ТекстыЗапроса = Новый СписокЗначений;

ТекстЗапросаТаблицаОбъемДефицитногоТовара(Запрос, ТекстыЗапроса, Регистры);

ПроведениеСервер.ИницализироватьТаблицыДляДвижений(Запрос, ТекстыЗапроса, ДополнительныеСвойства.ТаблицыДляДвижений, Истина);

КонецПроцедуры

Функция ТекстЗапросаТаблицаОбъемДефицитногоТовара(Запрос, ТекстыЗапроса, Регистры)

ИмяРегистра = "экзаменОбъемДефицитногоТовара";

Если НЕ ПроведениеСервер.ТребуетсяТаблицаДляДвижений(ИмяРегистра, Регистры) Тогда

Возврат "";

КонецЕсли;

ТекстЗапроса =

"ВЫБРАТЬ

| ОбъемДефицитногоТовараТовары.НомерСтроки КАК НСтр,

| ОбъемДефицитногоТовараТовары.Ссылка.Дата КАК Период,

| ОбъемДефицитногоТовараТовары.Ссылка.Менеджер,

| ОбъемДефицитногоТовараТовары.Ссылка.Месяц,

| ОбъемДефицитногоТовараТовары.Номенклатура,

| ОбъемДефицитногоТовараТовары.Количество КАК КоличествоЛимит

|ИЗ

| Документ.экзаменОбъемДефицитногоТовара.Товары КАК ОбъемДефицитногоТовараТовары

|ГДЕ

| ОбъемДефицитногоТовараТовары.Ссылка = &Ссылка

| И ОбъемДефицитногоТовараТовары.Номенклатура.ТипНоменклатуры = ЗНАЧЕНИЕ(Перечисление.ТипыНоменклатуры.Товар)

|

|УПОРЯДОЧИТЬ ПО

| НСтр";

ТекстыЗапроса.Добавить(ТекстЗапроса, ИмяРегистра);

Возврат ТекстЗапроса;

КонецФункции

#КонецОбласти

#Область Отчеты

Процедура ДобавитьКомандыОтчетов(КомандыОтчетов) Экспорт

ВариантыОтчетовУТПереопределяемый.ДобавитьКомандуДвиженияДокумента(КомандыОтчетов);

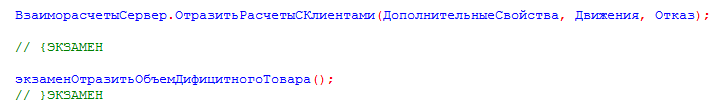
КонецПроцедуры

#КонецОбласти

Создадим роль экзаменПревышениеОбъемаДефицитногоТовара, у которой снимем все флажки.

## Алгоритм проведения документа ЗаказКлиента по РН экзаменОбъемДефицитногоТовара

Модуль объекта, процедура ОбработкаПроведения



#Область Экзамен

Процедура экзаменОтразитьОбъемДифицитногоТовара()

Таблица = ДополнительныеСвойства.ТаблицыДляДвижений.ТаблицаэкзаменОбъемДефицитногоТовара;

Если Таблица.Количество() = 0 Тогда

Возврат;

КонецЕсли;

НаборЗаписей = Движения.экзаменОбъемДефицитногоТовара;

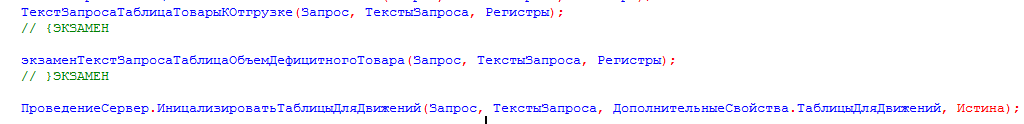
НаборЗаписей.Записывать = Истина;

НаборЗаписей.Загрузить(Таблица);

КонецПроцедуры

#КонецОбласти

Модуль менеджера, процедура ИнициализироватьДанныеДокумента

**

#Область Экзамен

Функция экзаменТекстЗапросаТаблицаОбъемДефицитногоТовара(Запрос, ТекстыЗапроса, Регистры)

ИмяРегистра = "экзаменОбъемДефицитногоТовара";

Если НЕ ПроведениеСервер.ТребуетсяТаблицаДляДвижений(ИмяРегистра, Регистры) Тогда

Возврат "";

КонецЕсли;

ТекстЗапроса =

"ВЫБРАТЬ

| ЗаказКлиентаТовары.НомерСтроки КАК НСтр,

| &ПЕРИОД КАК Период,

| НАЧАЛОПЕРИОДА(&Период, МЕСЯЦ) КАК Месяц,

| &Менеджер КАК Менеджер,

| ЗаказКлиентаТовары.Номенклатура КАК Номенклатура,

| ЗаказКлиентаТовары.Количество КАК КоличествоФакт

|ИЗ

| Документ.ЗаказКлиента.Товары КАК ЗаказКлиентаТовары

| ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрНакопления.экзаменОбъемДефицитногоТовара.Обороты(

| ,

| ,

| ,

| Менеджер = &Менеджер

| И Месяц = НАЧАЛОПЕРИОДА(&Период, МЕСЯЦ)) КАК ОбъемДефицитногоТовараОбороты

| ПО ЗаказКлиентаТовары.Номенклатура = ОбъемДефицитногоТовараОбороты.Номенклатура

|ГДЕ

| ЗаказКлиентаТовары.Ссылка = &Ссылка

|

|УПОРЯДОЧИТЬ ПО

| НСтр";

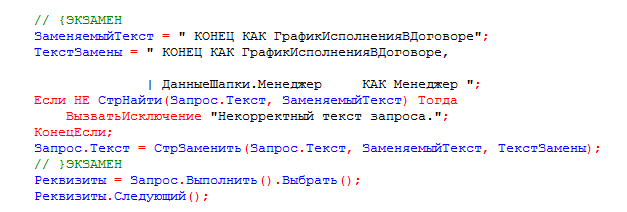
ТекстыЗапроса.Добавить(ТекстЗапроса, ИмяРегистра);

Возврат ТекстЗапроса;

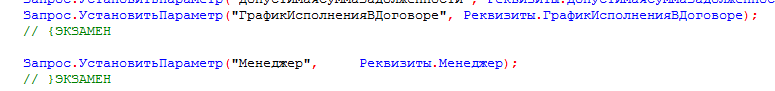
КонецФункции

#КонецОбласти

Процедура ЗаполнитьПараметрыИнициализации

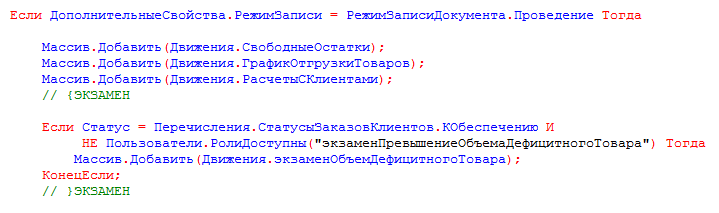


Далее в этой же процедуре



Контроль объема дефицитных товаров

В процедуре СформироватьСписокРегистровДляКонтроля



ОМ ПроведениеСервер, процедура ВыполнитьКонтрольРезультатовПроведения

// {ЭКЗАМЕН

// Контроль отрицательных остатков экзаменОбъемДефицитногоТовара.

Если ЕстьИзмененияВТаблице(ДанныеТаблиц,"ОбъемДефицитногоТовараИзменение") Тогда

МассивКонтролей.Добавить(Врег("экзаменОбъемДефицитногоТовара"));

ТекстЗапроса = ТекстЗапроса +

"

|ВЫБРАТЬ

| ТаблицаОстатков.Менеджер КАК Менеджер,

| ТаблицаОстатков.Месяц КАК Месяц,

| ТаблицаОстатков.Номенклатура КАК Номенклатура,

| ТаблицаОстатков.Номенклатура.ЕдиницаИзмерения КАК ЕдиницаИзмерения,

| ТаблицаОстатков.КоличествоЛимитОборот - ТаблицаОстатков.КоличествоФактОборот КАК Количество

|ИЗ

| РегистрНакопления.экзаменОбъемДефицитногоТовара.Обороты(

| ,,,

| (Менеджер, Месяц, Номенклатура) В

| (ВЫБРАТЬ

| Таблица.Менеджер,

| Таблица.Месяц,

| Таблица.Номенклатура

|

| ИЗ

| ОбъемДефицитногоТовараИзменение КАК Таблица)

| ) КАК ТаблицаОстатков

|

|ГДЕ

| ТаблицаОстатков.КоличествоЛимитОборот - ТаблицаОстатков.КоличествоФактОборот < 0;

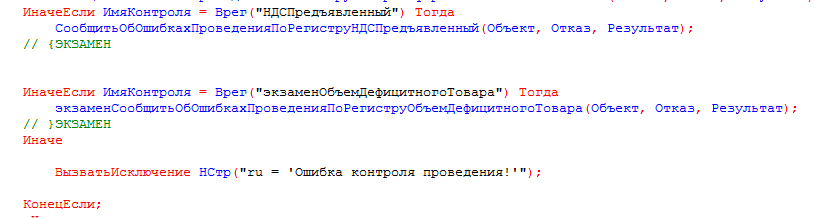
|///////////////////////////////////////////////////////////////////

|";

КонецЕсли;

// }ЭКЗАМЕН

Далее в этой же процедуре

**

Модуль набора записей РН экзаменОбъемДефицитногоТовара

ПроцедураПередЗаписью(Отказ, Замещение)

Если ОбменДанными.Загрузка ИЛИ НЕ ПроведениеСервер.РассчитыватьИзменения(ДополнительныеСвойства)Тогда

## Возврат;

## КонецЕсли;

## ОбновлениеИнформационнойБазы.ПроверитьОбъектОбработан(ЭтотОбъект);

## СтруктураВременныеТаблицы = ДополнительныеСвойства.ДляПроведения.СтруктураВременныеТаблицы;

## БлокироватьДляИзменения = Истина;

## // Текущее состояние набора помещается во временную таблицу, чтобы при записи получить

## // изменение нового набора относительно текущего.

## Запрос = Новый Запрос(

## "ВЫБРАТЬ

## | Таблица.Менеджер КАК Менеджер,

## | Таблица.Месяц КАК Месяц,

## | Таблица.Номенклатура КАК Номенклатура,

## | Таблица.КоличествоЛимит КАК КоличествоЛимитПередЗаписью,

## | Таблица.КоличествоФакт КАК КоличествоФактПередЗаписью

## |ПОМЕСТИТЬ ОбъемДефицитногоТовараПередЗаписью

## |ИЗ

## | РегистрНакопления.экзаменОбъемДефицитногоТовара КАК Таблица

## |

## |ГДЕ

## | Таблица.Регистратор = &Регистратор

## | И НЕ &ЭтоНовый");

## 

## Запрос.УстановитьПараметр("Регистратор", Отбор.Регистратор.Значение);

## Запрос.УстановитьПараметр("ЭтоНовый", ДополнительныеСвойства.ЭтоНовый);

## Запрос.МенеджерВременныхТаблиц = СтруктураВременныеТаблицы.МенеджерВременныхТаблиц;

## Запрос.Выполнить();

## КонецПроцедуры

## Процедура ПриЗаписи(Отказ, Замещение)

## Если ОбменДанными.Загрузка ИЛИ НЕ ПроведениеСервер.РассчитыватьИзменения(ДополнительныеСвойства) Тогда

## Возврат;

## КонецЕсли;

## СтруктураВременныеТаблицы = ДополнительныеСвойства.ДляПроведения.СтруктураВременныеТаблицы;

## // Рассчитывается изменение нового набора относительно текущего с учетом накопленных изменений

## // и помещается во временную таблицу.

## Запрос = Новый Запрос(

## "ВЫБРАТЬ

## | ТаблицаИзменений.Менеджер КАК Менеджер,

## | ТаблицаИзменений.Месяц КАК Месяц,

## | ТаблицаИзменений.Номенклатура КАК Номенклатура,

## | СУММА(ТаблицаИзменений.КоличествоЛимитИзменение) КАК КоличествоЛимитИзменение,

## | СУММА(ТаблицаИзменений.КоличествоФактИзменение) КАК КоличествоФактИзменение

## |ПОМЕСТИТЬ ОбъемДефицитногоТовараИзменение

## |ИЗ

## | (ВЫБРАТЬ

## | Таблица.Менеджер КАК Менеджер,

## | Таблица.Месяц КАК Месяц,

## | Таблица.Номенклатура КАК Номенклатура,

## | Таблица.КоличествоЛимитПередЗаписью КАК КоличествоЛимитИзменение,

## | Таблица.КоличествоФактПередЗаписью КАК КоличествоФактИзменение

## | ИЗ

## | ОбъемДефицитногоТовараПередЗаписью КАК Таблица

## |

## | ОБЪЕДИНИТЬ ВСЕ

## |

## | ВЫБРАТЬ

## | Таблица.Менеджер КАК Менеджер,

## | Таблица.Месяц КАК Месяц,

## | Таблица.Номенклатура КАК Номенклатура,

## | Таблица.КоличествоЛимит КАК КоличествоЛимитИзменение,

## | Таблица.КоличествоФакт КАК КоличествоФактИзменение

## | ИЗ

## | РегистрНакопления.экзаменОбъемДефицитногоТовара КАК Таблица

## |

## | ГДЕ

## | Таблица.Регистратор = &Регистратор) КАК ТаблицаИзменений

## |

## |СГРУППИРОВАТЬ ПО

## | ТаблицаИзменений.Менеджер,

## | ТаблицаИзменений.Месяц,

## | ТаблицаИзменений.Номенклатура

## |

## |ИМЕЮЩИЕ

## | (СУММА(ТаблицаИзменений.КоличествоЛимитИзменение) <> 0 ИЛИ

## | СУММА(ТаблицаИзменений.КоличествоФактИзменение) <> 0)

## |;

## |

## |////////////////////////////////////////////////////////////////////////////////

## |УНИЧТОЖИТЬ ОбъемДефицитногоТовараПередЗаписью");

## Запрос.УстановитьПараметр("Регистратор", Отбор.Регистратор.Значение);

## Запрос.МенеджерВременныхТаблиц = СтруктураВременныеТаблицы.МенеджерВременныхТаблиц;

## Выборка = Запрос.ВыполнитьПакет()[0].Выбрать();

## Выборка.Следующий();

## // Новые изменения были помещены во временную таблицу.

## // Добавляется информация о ее существовании и наличии в ней записей об изменении.

## СтруктураВременныеТаблицы.Вставить("ОбъемДефицитногоТовараИзменение", Выборка.Количество > 0);

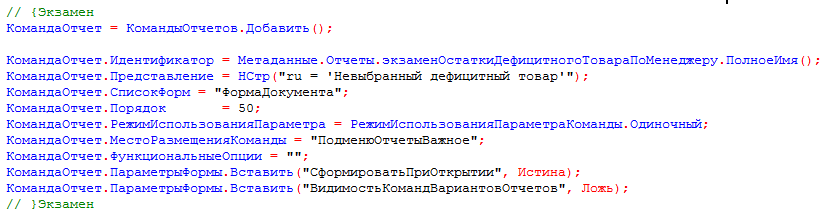
## КонецПроцедуры

## #КонецОбласти

## #КонецЕсли

## Отчет

Далее в модуле менеджера ЗаказКлиента, в процедуре ДобавитьКомандыОтчетов

**

Вставить отчет в кофигурацию