2012 год 2 тур

Требования

Задание финального тура

Исходная ситуация

Для хранения спецодежды на предприятии используется три склада.

Предприятие ведет бухгалтерский учет в программе "1С:Бухгалтерия 8". Учет спецодежды ведется в составе материальных запасов на счете 10 в разрезе субконто "Материалы" и "Склады" в стоимостном и натуральном выражении. Оценка материальных запасов производится по средневзвешенной цене, которая определяется отдельно для каждого материала по предприятию в целом.

В информационной базе предприятия зафиксированы остатки спецодежды на 1 апреля 2012 г. по каждому наименованию спецодежды и по каждому складу.

Предприятие решает ликвидировать один из складов для хранения спецодежды, а остатки спецодежды с этого склада переместить на два оставшихся склада таким образом, чтобы стоимостная оценка запасов спецодежды на этих двух складах по возможности составляла бы одинаковую величину или, если это невозможно, различалась бы минимально. При этом следует исходить из того, что перемещать исходные запасы спецодежды между неликвидируемыми складами запрещено.

Требуется

В каркасной конфигурации разработать программу, которая анализирует текущие остатки спецодежды на складах и формирует отчет, содержащий две таблицы.

В первой таблице отображается текущее наличие запасов спецодежды на складах и определяется средняя цена для каждого наименования спецодежды.

***Запасы спецодежды на складах предприятия на 01.04.2012***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Спецодежда** | **Цена** | **1 склад** | | **2 склад** | | **3 склад** | |
| **Кол-во** | **Стоимость** | **Кол-во** | **Стоимость** | **Кол-во** | **Стоимость** |
| СО-01 | 450 | 30 | 13500 | 10 | 4200 | 20 | 9300 |
| СО-02 | 280 | 18 | 5040 | 50 | 14000 | 15 | 4200 |
| СО-03 | 320 | 85 | 27200 | 60 | 19200 | 18 | 5760 |
| СО-04 | 610 | 5 | 3050 | 20 | 12200 | 62 | 37820 |
| СО-05 | 550 | 64 | 35200 | 12 | 6600 | 5 | 2750 |
| ИТОГО: |  |  | 83990 |  | 56200 |  | 59830 |
| ИТОГО по предприятию: | | 200020 | | | | | |

Во второй таблице должен быть представлен вариант размещения всех имеющихся запасов спецодежды на двух сохраняемых складах с соблюдением вышеуказанного условия – обеспечение максимальной близости (или равенства) стоимостной оценки запасов на этих складах, рассчитанной по средневзвешенной цене.

***Предлагаемый вариант размещения спецодежды на двух складах***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Спецодежда** | **Цена** | **1 склад** | | **2 склад** | |
| **Кол-во** | **Стоимость** | **Кол-во** | **Стоимость** |
| СО-01 | 450 | 46 | 20700 | 14 | 6300 |
| СО-02 | 280 | 28 | 7840 | 55 | 154000 |
| СО-03 | 320 | 100 | 32000 | 63 | 20160 |
| СО-04 | 610 | 7 | 4270 | 80 | 48800 |
| СО-05 | 550 | 64 | 35200 | 17 | 9350 |
| ИТОГО: |  |  | 100010 |  | 100010 |

Исходные данные

Исходные данные представлены в ИБ каркасной конфигурации.

В каркасную конфигурацию включены:

План счетов бухгалтерского учета, содержащий счета 10 "Материалы" и   
00 "Вспомогательный"

Справочники "Материалы" и "Склады", используемые, соответственно, для ведения аналитического учета на счете 10 в разрезе материалов и складов

Документ для ввода данных о запасах материалов на складах, при помощи которого введено несколько проводок в дебет счета 10 в корреспонденции со вспомогательным счетом 00 для отражения запасов спецодежды на складах на начало дня 01.04.2012

Результаты и их представление

В результате выполнения задания должен быть разработан отчет "1С:Предприятия 8", формирующий две вышеуказанные таблицы. Необходимо предусмотреть форму диалога, в которой предоставить пользователю возможность определить дату, на которую производится формирование отчета, а также возможность выбора ликвидируемого склада. По умолчанию должна устанавливаться дата ‑ 01.04.2012, а в качестве ликвидируемого склада ‑ склад с максимальным значением кода.

Для отладки модуля отчета допускается добавлять в информационную базу каркасной конфигурации дополнительные элементы справочников и проводки. Жюри будет проводить тестирование на собственных наборах данных, которые могут отличаться от данных, содержащихся в исходной информационной базе каркасной конфигурации.

Разработанный отчет должен быть сохранен как внешний отчет "1С:Предприятия 8" и передан на проверку жюри.

Решение

Для получения средневзвешанных цен товаров требуется получить сумму товаров и сумму количества товаров по всему предприятию и после этого разделить сумму на количество. Для поиска оптимального варианта размещения спецодежды на оставшихся складах можно использовать следующий алгоритм: сначала оставшиеся склады сортируются по убыванию по сумме, которой не хватает до средней сумме по предприятию. Затем товары перемещаются на склады (сначала на склад у которого отклонение от средней суммы максимально, потом с меньшим), сначала дорогие, потом дешевые, при чем так, чтобы отклонение от средней суммы было минимальным. Рассмотрим пример:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Товар | Склад 1 | | Склад 2 | | Склад 3 (ликвид.) | |
| Количество | Сумма | Количество | Сумма | Количество | Сумма |
| Товар 1 | 3 | 21 | 2 | 14 | 4 | 28 |
| Товар 2 | 2 | 20 | 3 | 30 | 1 | 10 |

Сперва находятся средневзвешанные цены:

|  |  |  |  |
| --- | --- | --- | --- |
| Товар | Сумма по предприятию | Количество по предприятию | Средневзвешанная цена |
| Товар 1 | 63 | 9 | 7 |
| Товар 2 | 60 | 6 | 10 |

Далее расчитывается общая сумма товаров по предприятию и средняя сумма:

|  |  |
| --- | --- |
| Сумма цен по предприятию | 123 |
| Средняя сумма цен по предприятию | 61.5 |

Далее высчитывается нехватка суммы на остающихся складах до средней суммы:

|  |  |
| --- | --- |
| Склад 1 | Склад 2 |
| 61.5 – (21 + 20) = 20.5 | 61.5 – (14 + 30) = 17.5 |

Теперь склады сортируются по убыванию:

|  |  |
| --- | --- |
| Склад | Сумма |
| Склад 1 | 20.5 |
| Склад 2 | 17.5 |

Далее на склады, отсортированные по убыванию, размещаются товары (так же отсортированные по убыванию) перенося по одной единице товара с ликвидируемого склада пока сумма на складе не выше средней по предприятию.

Сначала обрабатывается Склад 1:

|  |  |  |  |
| --- | --- | --- | --- |
| Итерация | Добавленный товар | Остаток добавленного товара на ликвидируемом складе | Сумма товаров на складе |
| 1 | Товар 2 | 0 | 10.5 |
| 2 | Товар 1 | 3 | 3.5 |
| 3 | При добавлении Товара 1 на склад сумма товаров будет больше средней по предприятию, обрываем добавление на склад 1 | | |

Далее обрабатывается Склад 2:

|  |  |  |  |
| --- | --- | --- | --- |
| Итерация | Добавленный товар | Остаток добавленного товара на ликвидируемом складе | Сумма товаров на складе |
| 1 | Товар 1 | 2 | 10.5 |
| 2 | Товар 1 | 1 | 3.5 |
| 3 | При добавлении Товара 1 на склад сумма товаров будет больше средней по предприятию, обрываем добавление на склад 2 | | |

Остается один не распределенный товар, он отправляется на склад, у которого больше всего отклонение от средневзвешенной цены, но так как в примере отклонение после распределения у обоих складов будет 3.5 товар можно отправить на любой склад (к примеру на склад 1).

В результате получается следующее распределение товаров:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Товар | Склад 1 | | Склад 2 | |
| Количество | Сумма | Количество | Сумма |
| Товар 1 | 5 | 35 | 4 | 28 |
| Товар 2 | 1 | 30 | 3 | 30 |

Итог: на складе 1 сумма товаров – 65, на складе 2 сумма товаров - 58

Реализация

//

// Модуль объекта

//

// функция получения остатков товаров, учитывая/не учитывая разрез складов и товаров

Функция ОстаткиТоваров(Дата, Склад = Неопределено, СО = Неопределено, ЗнакСравненияСклада = "=") Экспорт

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ЖурналПроводокОстатки.Субконто1 КАК СО,

| ЖурналПроводокОстатки.Субконто2 КАК Склад,

| ЖурналПроводокОстатки.СуммаОстаток КАК Сумма,

| ЖурналПроводокОстатки.КоличествоОстаток КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналПроводок.Остатки(&Дата, Счет = &Счет, , " +

?(СО <> Неопределено, "Субконто1 = &СО", "") +

?(Склад <> Неопределено и СО <> Неопределено, " И ", "") +

?(Склад <> Неопределено, "Субконто2 " + ЗнакСравненияСклада + " &Склад", "") + ") КАК ЖурналПроводокОстатки

|

|УПОРЯДОЧИТЬ ПО

| ЖурналПроводокОстатки.Субконто1.Наименование,

| ЖурналПроводокОстатки.Субконто2.Наименование";

Запрос.УстановитьПараметр("Дата", Дата);

Запрос.УстановитьПараметр("СО", СО);

Запрос.УстановитьПараметр("Склад", Склад);

Запрос.УстановитьПараметр("Счет", ПланыСчетов.ПланСчетов.Материалы);

Возврат Запрос.Выполнить().Выгрузить();

КонецФункции

// функция посчета средневзвешанной цены исходя из остатков на указанную дату

Функция СредневзвешаннаяЦена(Дата, СО) Экспорт

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| СУММА(ЖурналПроводокОстатки.СуммаОстаток) КАК Сумма,

| СУММА(ЖурналПроводокОстатки.КоличествоОстаток) КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналПроводок.Остатки(&Дата, Счет = &Счет, , Субконто1 = &СО) КАК ЖурналПроводокОстатки";

Запрос.УстановитьПараметр("Дата", Дата);

Запрос.УстановитьПараметр("СО", СО);

Запрос.УстановитьПараметр("Счет", ПланыСчетов.ПланСчетов.Материалы);

Результат = Запрос.Выполнить().Выгрузить();

Если Результат.Количество() = 0 Тогда

Возврат 0;

ИначеЕсли Результат[0].Сумма = 0 или Результат[0].Количество = 0 Тогда

Возврат 0;

Иначе

Возврат Результат[0].Сумма / Результат[0].Количество;

КонецЕсли;

КонецФункции

// функция возвращает остатки по складам, с разрезом по количеству и сумме (по складам)

Функция ОстаткиНаСкладах(Дата, УказыватьСО = Ложь, ВычислятьСумму = Истина, ВычислятьКоличество = Ложь, ВычислятьИтого = Истина) Экспорт

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| " + ?(УказыватьСО, "ЖурналПроводокОстатки.Субконто1 КАК СО,", "") + "

| ЖурналПроводокОстатки.Субконто2 КАК Склад,

| " + ?(ВычислятьСумму, "СУММА(ЖурналПроводокОстатки.СуммаОстаток) КАК Сумма"

+ ?(ВычислятьКоличество,",",""), "") + "

| " + ?(ВычислятьКоличество, "ЖурналПроводокОстатки.КоличествоОстаток КАК Количество", "") + "

|ИЗ

| РегистрБухгалтерии.ЖурналПроводок.Остатки(&Дата, Счет = &Счет, , ) КАК ЖурналПроводокОстатки

|

|СГРУППИРОВАТЬ ПО

| " + ?(УказыватьСО, "ЖурналПроводокОстатки.Субконто1,", "") + "

| ЖурналПроводокОстатки.Субконто2,

| ЖурналПроводокОстатки.КоличествоОстаток" +

?(ВычислятьИтого и (ВычислятьКоличество или ВычислятьСумму), "

|ИТОГИ " +

?(ВычислятьСумму, "

| СУММА(Сумма)", "") +

?(ВычислятьКоличество, "

| СУММА(Количество)", "") + "

|ПО

| ОБЩИЕ", "");

Запрос.УстановитьПараметр("Дата", Дата);

Запрос.УстановитьПараметр("Счет", ПланыСчетов.ПланСчетов.Материалы);

Возврат Запрос.Выполнить();

КонецФункции

// функция возвращает количественные остатки на дату в разрезе складов и товаров

Функция КоличествоНаСкладе(Дата, Склад, СО) Экспорт

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ЖурналПроводокОстатки.КоличествоОстаток КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналПроводок.Остатки(

| &Дата,

| Счет = &Счет,

| ,

| Субконто1 = &СО

| И Субконто2 = &Склад) КАК ЖурналПроводокОстатки";

Запрос.УстановитьПараметр("Дата", Дата);

Запрос.УстановитьПараметр("Счет", ПланыСчетов.ПланСчетов.Материалы);

Запрос.УстановитьПараметр("СО", СО);

Запрос.УстановитьПараметр("Склад", Склад);

Результат = Запрос.Выполнить().Выгрузить();

Если Результат.Количество() = 0 Тогда

Возврат 0;

Иначе

Возврат Результат[0].Количество;

КонецЕсли;

КонецФункции

//

// Модуль формы

//

Процедура КнопкаСформироватьНажатие(Кнопка)

// таблица средневзвешанных цен

СрЦ = Новый ТаблицаЗначений;

СрЦ.Колонки.Добавить("СО");

СрЦ.Колонки.Добавить("Цена");

СрЦ.Колонки.Добавить("Количество");

ТабДок = ЭлементыФормы.ПолеВывода;

ТабДок.Очистить();

// получение областей макета

Макет = ПолучитьМакет("ТекущиеОстатки");

ОбластьЗаголовка = Макет.ПолучитьОбласть("Заголовок");

ОбластьШапкаОбщая = Макет.ПолучитьОбласть("Шапка|Общее");

ОбластьШапкаСклад = Макет.ПолучитьОбласть("Шапка|Склад");

ОбластьСтрокаОбщая = Макет.ПолучитьОбласть("Строка|Общее");

ОбластьСтрокаСклад = Макет.ПолучитьОбласть("Строка|Склад");

ОбластьИтогоОбщая = Макет.ПолучитьОбласть("Итого|Общее");

ОбластьИтогоСклад = Макет.ПолучитьОбласть("Итого|Склад");

ОбластьОбщееИтого = Макет.ПолучитьОбласть("ИтогоОбщее");

ОбластьРазрыв = Макет.ПолучитьОбласть("Разрыв");

// вывод заголовка первой таблицы

ОбластьЗаголовка.Параметры.Заголовок = "Запасы спецодежды на складах предприятия на " + Формат(Дата, "ДФ=dd.MM.yyyy");

ТабДок.Вывести(ОбластьЗаголовка);

// вывод всех складов

ТабДок.Вывести(ОбластьШапкаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

Пока ВыборкаСкладов.Следующий() Цикл

ОбластьШапкаСклад.Параметры.Склад = ВыборкаСкладов.Наименование;

ТабДок.Присоединить(ОбластьШапкаСклад);

КонецЦикла;

ВыборкаСО = Справочники.Материалы.Выбрать();

Пока ВыборкаСО.Следующий() Цикл

ОбластьСтрокаОбщая.Параметры.Спецодежда = ВыборкаСО.Ссылка;

// заполнения таблицы средневзвешанных цен

НовСтр = СрЦ.Добавить();

НовСтр.СО = ВыборкаСО.Ссылка;

НовСтр.Цена = СредневзвешаннаяЦена(Дата, ВыборкаСО.Ссылка);

ОбластьСтрокаОбщая.Параметры.Цена = НовСтр.Цена;

ТабДок.Вывести(ОбластьСтрокаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

Пока ВыборкаСкладов.Следующий() Цикл

// для каждой спецодежды каждого склада вычисляем остатки

Остатки = ОстаткиТоваров(Дата, ВыборкаСкладов.Ссылка, ВыборкаСО.Ссылка);

Если Остатки.Количество() = 0 Тогда

ОбластьСтрокаСклад.Параметры.Количество = 0;

ОбластьСтрокаСклад.Параметры.Стоимость = 0;

Иначе

ОбластьСтрокаСклад.Параметры.Количество = Остатки[0].Количество;

ОбластьСтрокаСклад.Параметры.Стоимость = Остатки[0].Сумма;

КонецЕсли;

// и выводим их в отчет

ТабДок.Присоединить(ОбластьСтрокаСклад);

КонецЦикла;

КонецЦикла;

ТабДок.Вывести(ОбластьИтогоОбщая);

// общая стоимостная оценка спецодежды на складах

СОСклада = Новый ТаблицаЗначений;

СОСклада.Колонки.Добавить("Склад");

СОСклада.Колонки.Добавить("Сумма");

// получаем остатки, не учитывая разрез по материалам и складам

ОстаткиСклады = ОстаткиНаСкладах(Дата).Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);

ОстаткиСклады.Следующий();

// остатки по всем складам и всей СО (сумма остатков по всем складам)

ВсегоНаПредприятии = ОстаткиСклады.Сумма;

ОбластьОбщееИтого.Параметры.Стоимость = ОстаткиСклады.Сумма;

ОстаткиСкладов = ОстаткиСклады.Выбрать();

Пока ОстаткиСкладов.Следующий() Цикл

// заполняем таблицу общей стоиомостной оценки складов

НовСтр = СОСклада.Добавить();

НовСтр.Склад = ОстаткиСкладов.Склад;

НовСтр.Сумма = ОстаткиСкладов.Сумма;

// выводим общую сумму в отчет

ОбластьИтогоСклад.Параметры.Стоимость = ОстаткиСкладов.Сумма;

ТабДок.Присоединить(ОбластьИтогоСклад);

КонецЦикла;

// вывод "итого по предприятию"

ТабДок.Вывести(ОбластьОбщееИтого);

// сортируем таблицу средневзвешанных цен по обыванию цен

СрЦ.Сортировать("Цена Убыв");

// получаем количество, которое необходимо распределить на оставшиеся склады

Для Каждого Стр Из СрЦ Цикл

Стр.Количество = КоличествоНаСкладе(Дата, ЛиквидируемыйСклад, Стр.СО);

КонецЦикла;

// остатки на складах по количеству, без учета ликвидированного склада

ОстаткиСкл = ОстаткиТоваров(Дата, ЛиквидируемыйСклад, , "<>");

// массив для хранения сумм спецодежды после переоценки

МассивСО = Новый ТаблицаЗначений;

МассивСО.Колонки.Добавить("Склад");

МассивСО.Колонки.Добавить("Сумма");

МассивСО.Колонки.Добавить("ДляРаспределения");

// переоценка по средневзвешанной цене

Для Каждого Стр Из ОстаткиСкл Цикл

// добавление склада в массив, если его еще там нет

Если МассивСО.Найти(Стр.Склад) = Неопределено Тогда

НовСтрСкл = МассивСО.Добавить();

НовСтрСкл.Склад = Стр.Склад;

НовСтрСкл.Сумма = 0;

НовСтрСкл.ДляРаспределения = 0;

КонецЕсли;

Для Каждого СтрСО Из СрЦ Цикл

Если Стр.СО = СтрСО.СО Тогда

Стр.Сумма = Стр.Количество \* СтрСО.Цена;

// получаем общую стоимостную оценку по складам после переоценки (с учетом средневзвешанных цен)

МассивСО.Найти(Стр.Склад, "Склад").Сумма = МассивСО.Найти(Стр.Склад, "Склад").Сумма + Стр.Сумма;

Прервать;

КонецЕсли;

КонецЦикла;

КонецЦикла;

// вычисляем сумму, которой не хватает до средней стоимостной оценки всех товаров на каждом складе после ликвидации

Для Каждого Стр Из МассивСО Цикл

Стр.ДляРаспределения = ВсегоНаПредприятии / МассивСО.Количество() - Стр.Сумма;

КонецЦикла;

МассивСО.Сортировать("ДляРаспределения Убыв");

Распределение = Новый ТаблицаЗначений;

Распределение.Колонки.Добавить("Склад");

Распределение.Колонки.Добавить("СО");

Распределение.Колонки.Добавить("Количество");

Распределение.Колонки.Добавить("Сумма");

//поиск оптимального перемещения товаров с ликвидируемого склада на оставшиеся два

//////////////////////////////////////////////////////////////////////////////////

//Для поиска оптимального варианта размещения спецодежды ликвидируемого склада и

//оставшихся двух складов использую следующий алогритм: сначала сортируем склады

//по убыванию по сумме, которой не хватает до средней стоимостной оценки (для этого

//общую сумму по предприятию делим на два и вычитаем из нее стоимостную оценку после

//переоценки находящуюся на каждом складе). Затем, чтобы отклонение от стоимостной

//оценки, которую необходимо набрать было минимальным, начинаем перемещение с самых

//дорогих (по переоценочным ценам) товарам, что дает возможность заполнить и маленькое

//отклонение, перемещая товары от дорогих к дешевым. При этом заполняем таблицу

//для посчета суммы и количества перемещенных товаров на каждый склад.

/////////////////////////////////////////////////////////////////////////////////

Для Каждого Стр Из МассивСО Цикл

Для Каждого СтрСО Из СрЦ Цикл

Пока СтрСО.Количество > 0 Цикл

Если Стр.ДляРаспределения - СтрСО.Цена < 0 Тогда

Прервать;

КонецЕсли;

СтрСО.Количество = СтрСО.Количество - 1;

Стр.ДляРаспределения = Стр.ДляРаспределения - СтрСО.Цена;

Стр.Сумма = Стр.Сумма + СтрСО.Цена;

НовСтр = Распределение.Добавить();

НовСтр.Склад = Стр.Склад;

НовСтр.СО = СтрСО.СО;

НовСтр.Количество = 1;

НовСтр.Сумма = СтрСО.Цена;

КонецЦикла;

КонецЦикла;

КонецЦикла;

// если не все до конца распределилось, то ищем куда лучше отнести последний товар, если он есть

ОстатокРаспределения = МассивСО.Итог("ДляРаспределения");

СрЦ.Сортировать("Количество Убыв, Цена Убыв");

МассивСО.Сортировать("Сумма Убыв");

Для Каждого Стр Из СрЦ Цикл

// выбираем товар, который остался (если остался)

Если Стр.Количество > 0 Тогда

Стр.Количество = Стр.Количество - 1;

МассивСО[1].ДляРаспределения = МассивСО[1].ДляРаспределения - Стр.Цена;

МассивСО[1].Сумма = МассивСО[1].Сумма + Стр.Цена;

// и выбираем для перемещения тот склад, добавив в который отклонение будет минимальным

НовСтр = Распределение.Добавить();

НовСтр.Склад = МассивСО[1].Склад;

НовСтр.СО = Стр.СО;

НовСтр.Количество = 1;

НовСтр.Сумма = Стр.Цена;

КонецЕсли;

КонецЦикла;

// сворачиваем таблицу с распределенными товарами, для посчета суммы и количества

Распределение.Свернуть("Склад, СО", "Количество, Сумма");

Распределение.Сортировать("Склад Возр, СО Возр");

// вывод разрыва между отчетами

ТабДок.Вывести(ОбластьРазрыв);

// вывод заголовка второй таблицы

ОбластьЗаголовка.Параметры.Заголовок = "Предлагаемый вариант размещения спецодежды на двух складах";

ТабДок.Вывести(ОбластьЗаголовка);

// выводим шапку таблицы для каждого склада

ТабДок.Вывести(ОбластьШапкаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

Пока ВыборкаСкладов.Следующий() Цикл

Если ВыборкаСкладов.Ссылка <> ЛиквидируемыйСклад Тогда

ОбластьШапкаСклад.Параметры.Склад = ВыборкаСкладов.Наименование;

ТабДок.Присоединить(ОбластьШапкаСклад);

КонецЕсли;

КонецЦикла;

// массив общей стоимостной оценки по складам после ликвидации вабранного склада

ОбщиеИтоговыеСуммы = Новый Массив;

ОбщиеИтоговыеСуммы.Добавить(0);

ОбщиеИтоговыеСуммы.Добавить(0);

// вывод спецодежды по складам (по количеству и сумме)

ВыборкаСО = Справочники.Материалы.Выбрать();

Пока ВыборкаСО.Следующий() Цикл

ОбластьСтрокаОбщая.Параметры.Спецодежда = ВыборкаСО.Ссылка;

ОбластьСтрокаОбщая.Параметры.Цена = СрЦ.Найти(ВыборкаСО.Ссылка, "СО").Цена;

ТабДок.Вывести(ОбластьСтрокаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

н = 0;

Пока ВыборкаСкладов.Следующий() Цикл

Если ВыборкаСкладов.Ссылка <> ЛиквидируемыйСклад Тогда

Остатки = ОстаткиТоваров(Дата, ВыборкаСкладов.Ссылка, ВыборкаСО.Ссылка);

Отбор = Новый Структура;

Отбор.Вставить("СО", ВыборкаСО.Ссылка);

Отбор.Вставить("Склад", ВыборкаСкладов.Ссылка);

РаспределеннаяСО = Распределение.НайтиСтроки(Отбор);

// ищем остатки на тех складах, которые не ликвидировали

// количество считаем как остатки + распределенные товары на данный склад

Если Остатки.Количество() = 0 Тогда

ОбщееКоличество = ?(РаспределеннаяСО.Количество() = 0, 0, РаспределеннаяСО[0].Количество);

Иначе

ОбщееКоличество = Остатки[0].Количество + ?(РаспределеннаяСО.Количество() = 0, 0, РаспределеннаяСО[0].Количество);

КонецЕсли;

ОбластьСтрокаСклад.Параметры.Количество = ОбщееКоличество;

ОбщиеИтоговыеСуммы[н] = ОбщиеИтоговыеСуммы[н] + ОбщееКоличество \* СрЦ.Найти(ВыборкаСО.Ссылка, "СО").Цена;

// сумму считаем, как количество (посчитанное выше) \* на цену (после переоценки по средневзвешанной цене)

ОбластьСтрокаСклад.Параметры.Стоимость = ОбщееКоличество \* СрЦ.Найти(ВыборкаСО.Ссылка, "СО").Цена;

ТабДок.Присоединить(ОбластьСтрокаСклад);

н = н + 1;

КонецЕсли;

КонецЦикла;

КонецЦикла;

ТабДок.Вывести(ОбластьИтогоОбщая);

// вывод общих итоговых сумм по складам

Для н = 0 по ОбщиеИтоговыеСуммы.Количество() - 1 Цикл

// выводим общую сумму в отчет

ОбластьИтогоСклад.Параметры.Стоимость = ОбщиеИтоговыеСуммы[н];

ТабДок.Присоединить(ОбластьИтогоСклад);

КонецЦикла;

КонецПроцедуры

Процедура ПриОткрытии()

// установка даты по умолчанию (1 апреля 2012 года)

Дата = '20120401';

// установка склада по умолчанию (склад, с максимальным значением кода)

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ ПЕРВЫЕ 1

| Склады.Ссылка КАК Склад

|ИЗ

| Справочник.Склады КАК Склады

|

|УПОРЯДОЧИТЬ ПО

| Склады.Код УБЫВ";

Результат = Запрос.Выполнить().Выгрузить();

Если Результат.Количество() <> 0 Тогда

ЛиквидируемыйСклад = Результат[0].Склад;

КонецЕсли;

КонецПроцедуры

Автор: Завладаев Алексей

Пояснения к коду:

// функция получения остатков товаров, учитывая/не учитывая разрез складов и товаров

Функция ОстаткиТоваров(Дата, Склад = Неопределено, СО = Неопределено, ЗнакСравненияСклада = "=") Экспорт

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ЖурналПроводокОстатки.Субконто1 КАК СО,

| ЖурналПроводокОстатки.Субконто2 КАК Склад,

| ЖурналПроводокОстатки.СуммаОстаток КАК Сумма,

| ЖурналПроводокОстатки.КоличествоОстаток КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналПроводок.Остатки(&Дата, Счет = &Счет, , " +

?(СО <> Неопределено, "Субконто1 = &СО", "") +

?(Склад <> Неопределено и СО <> Неопределено, " И ", "") +

?(Склад <> Неопределено, "Субконто2 " + ЗнакСравненияСклада + " &Склад", "") + ") КАК ЖурналПроводокОстатки

|

|УПОРЯДОЧИТЬ ПО

| ЖурналПроводокОстатки.Субконто1.Наименование,

| ЖурналПроводокОстатки.Субконто2.Наименование";

Запрос.УстановитьПараметр("Дата", Дата);

Запрос.УстановитьПараметр("СО", СО);

Запрос.УстановитьПараметр("Склад", Склад);

Запрос.УстановитьПараметр("Счет", ПланыСчетов.ПланСчетов.Материалы);

Возврат Запрос.Выполнить().Выгрузить();

КонецФункции

Данная функция расположена в модуле объекта, она требуется для получения остатков из счета Материалы всей спецодежды, либо определенной, по всем складам, либо по определенному, с возможностью выбора складов по разному условию (либо один из всех, либо все кроме одного "="/"<>"). Функция возвращает выгрузку результата запроса. При составлении текста запроса используются тернарные условные операции – конструкции вида (условие?при выполнении:при не выполнении). Они используются для уменьшения дублирующего кода и объема кода в целом. Разберем одно из них:

?(СО <> Неопределено, "Субконто1 = &СО", "")

В данной конструкции "СО <> Неопределено" – условие, если условие выполняется, т.е. СО не равно Неопределено, то конструкция вернет "Субконто1 = &СО", если условие не выполнится то будет возвращено "".

// функция посчета средневзвешанной цены исходя из остатков на указанную дату

Функция СредневзвешаннаяЦена(Дата, СО) Экспорт

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| СУММА(ЖурналПроводокОстатки.СуммаОстаток) КАК Сумма,

| СУММА(ЖурналПроводокОстатки.КоличествоОстаток) КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналПроводок.Остатки(&Дата, Счет = &Счет, , Субконто1 = &СО) КАК ЖурналПроводокОстатки";

Запрос.УстановитьПараметр("Дата", Дата);

Запрос.УстановитьПараметр("СО", СО);

Запрос.УстановитьПараметр("Счет", ПланыСчетов.ПланСчетов.Материалы);

Результат = Запрос.Выполнить().Выгрузить();

Если Результат.Количество() = 0 Тогда

Возврат 0;

ИначеЕсли Результат[0].Сумма = 0 или Результат[0].Количество = 0 Тогда

Возврат 0;

Иначе

Возврат Результат[0].Сумма / Результат[0].Количество;

КонецЕсли;

КонецФункции

Данная функция расположена в модуле объекта, она требуется для получения средневзвешанной цены спецодежды на определенную дату, для этого получается сумма остатков и количество остатков указанной спецодежды, на указанную дату из счета Материалы. Далее возвращается средневзвешанная цена (Сумма / Количество), с проверкой на Количество = 0 (чтобы предотвратить деление на ноль).

// функция возвращает остатки по складам, с разрезом по количеству и сумме (по складам)

Функция ОстаткиНаСкладах(Дата, УказыватьСО = Ложь, ВычислятьСумму = Истина, ВычислятьКоличество = Ложь, ВычислятьИтого = Истина) Экспорт

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| " + ?(УказыватьСО, "ЖурналПроводокОстатки.Субконто1 КАК СО,", "") + "

| ЖурналПроводокОстатки.Субконто2 КАК Склад,

| " + ?(ВычислятьСумму, "СУММА(ЖурналПроводокОстатки.СуммаОстаток) КАК Сумма"

+ ?(ВычислятьКоличество,",",""), "") + "

| " + ?(ВычислятьКоличество, "ЖурналПроводокОстатки.КоличествоОстаток КАК Количество", "") + "

|ИЗ

| РегистрБухгалтерии.ЖурналПроводок.Остатки(&Дата, Счет = &Счет, , ) КАК ЖурналПроводокОстатки

|

|СГРУППИРОВАТЬ ПО

| " + ?(УказыватьСО, "ЖурналПроводокОстатки.Субконто1,", "") + "

| ЖурналПроводокОстатки.Субконто2,

| ЖурналПроводокОстатки.КоличествоОстаток" +

?(ВычислятьИтого и (ВычислятьКоличество или ВычислятьСумму), "

|ИТОГИ " +

?(ВычислятьСумму, "

| СУММА(Сумма)", "") +

?(ВычислятьКоличество, "

| СУММА(Количество)", "") + "

|ПО

| ОБЩИЕ", "");

Запрос.УстановитьПараметр("Дата", Дата);

Запрос.УстановитьПараметр("Счет", ПланыСчетов.ПланСчетов.Материалы);

Возврат Запрос.Выполнить();

КонецФункции

Данная функция расположена в модуле объекта, она требуется для получения остатков на складах, на указанную дату, с настройкой параметров – включать в выборку спецодежду или нет, вычислять сумму по складам или нет, вычислять количество или нет, вычислять итоги или нет. По умолчанию спецодежда не указывается, сумма вычисляется, количество нет, итоги вычисляются. Для составления текста запроса используются тернарные условные операции. Возвращает функция результат запроса, готовый для обхода.

// функция возвращает количественные остатки на дату в разрезе складов и товаров

Функция КоличествоНаСкладе(Дата, Склад, СО) Экспорт

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ЖурналПроводокОстатки.КоличествоОстаток КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналПроводок.Остатки(

| &Дата,

| Счет = &Счет,

| ,

| Субконто1 = &СО

| И Субконто2 = &Склад) КАК ЖурналПроводокОстатки";

Запрос.УстановитьПараметр("Дата", Дата);

Запрос.УстановитьПараметр("Счет", ПланыСчетов.ПланСчетов.Материалы);

Запрос.УстановитьПараметр("СО", СО);

Запрос.УстановитьПараметр("Склад", Склад);

Результат = Запрос.Выполнить().Выгрузить();

Если Результат.Количество() = 0 Тогда

Возврат 0;

Иначе

Возврат Результат[0].Количество;

КонецЕсли;

КонецФункции

Данная функция расположена в модуле объекта, она требуется для получения количества спецодежды на определенном складе, на определенную дату.

Процедура КнопкаСформироватьНажатие(Кнопка)

// таблица средневзвешанных цен

СрЦ = Новый ТаблицаЗначений;

СрЦ.Колонки.Добавить("СО");

СрЦ.Колонки.Добавить("Цена");

СрЦ.Колонки.Добавить("Количество");

ТабДок = ЭлементыФормы.ПолеВывода;

ТабДок.Очистить();

// получение областей макета

Макет = ПолучитьМакет("ТекущиеОстатки");

ОбластьЗаголовка = Макет.ПолучитьОбласть("Заголовок");

ОбластьШапкаОбщая = Макет.ПолучитьОбласть("Шапка|Общее");

ОбластьШапкаСклад = Макет.ПолучитьОбласть("Шапка|Склад");

ОбластьСтрокаОбщая = Макет.ПолучитьОбласть("Строка|Общее");

ОбластьСтрокаСклад = Макет.ПолучитьОбласть("Строка|Склад");

ОбластьИтогоОбщая = Макет.ПолучитьОбласть("Итого|Общее");

ОбластьИтогоСклад = Макет.ПолучитьОбласть("Итого|Склад");

ОбластьОбщееИтого = Макет.ПолучитьОбласть("ИтогоОбщее");

ОбластьРазрыв = Макет.ПолучитьОбласть("Разрыв");

// вывод заголовка первой таблицы

ОбластьЗаголовка.Параметры.Заголовок = "Запасы спецодежды на складах предприятия на " + Формат(Дата, "ДФ=dd.MM.yyyy");

ТабДок.Вывести(ОбластьЗаголовка);

// вывод всех складов

ТабДок.Вывести(ОбластьШапкаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

Пока ВыборкаСкладов.Следующий() Цикл

ОбластьШапкаСклад.Параметры.Склад = ВыборкаСкладов.Наименование;

ТабДок.Присоединить(ОбластьШапкаСклад);

КонецЦикла;

ВыборкаСО = Справочники.Материалы.Выбрать();

Пока ВыборкаСО.Следующий() Цикл

ОбластьСтрокаОбщая.Параметры.Спецодежда = ВыборкаСО.Ссылка;

// заполнения таблицы средневзвешанных цен

НовСтр = СрЦ.Добавить();

НовСтр.СО = ВыборкаСО.Ссылка;

НовСтр.Цена = СредневзвешаннаяЦена(Дата, ВыборкаСО.Ссылка);

ОбластьСтрокаОбщая.Параметры.Цена = НовСтр.Цена;

ТабДок.Вывести(ОбластьСтрокаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

Пока ВыборкаСкладов.Следующий() Цикл

// для каждой спецодежды каждого склада вычисляем остатки

Остатки = ОстаткиТоваров(Дата, ВыборкаСкладов.Ссылка, ВыборкаСО.Ссылка);

Если Остатки.Количество() = 0 Тогда

ОбластьСтрокаСклад.Параметры.Количество = 0;

ОбластьСтрокаСклад.Параметры.Стоимость = 0;

Иначе

ОбластьСтрокаСклад.Параметры.Количество = Остатки[0].Количество;

ОбластьСтрокаСклад.Параметры.Стоимость = Остатки[0].Сумма;

КонецЕсли;

// и выводим их в отчет

ТабДок.Присоединить(ОбластьСтрокаСклад);

КонецЦикла;

КонецЦикла;

ТабДок.Вывести(ОбластьИтогоОбщая);

// общая стоимостная оценка спецодежды на складах

СОСклада = Новый ТаблицаЗначений;

СОСклада.Колонки.Добавить("Склад");

СОСклада.Колонки.Добавить("Сумма");

// получаем остатки, не учитывая разрез по материалам и складам

ОстаткиСклады = ОстаткиНаСкладах(Дата).Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);

ОстаткиСклады.Следующий();

// остатки по всем складам и всей СО (сумма остатков по всем складам)

ВсегоНаПредприятии = ОстаткиСклады.Сумма;

ОбластьОбщееИтого.Параметры.Стоимость = ОстаткиСклады.Сумма;

ОстаткиСкладов = ОстаткиСклады.Выбрать();

Пока ОстаткиСкладов.Следующий() Цикл

// заполняем таблицу общей стоиомостной оценки складов

НовСтр = СОСклада.Добавить();

НовСтр.Склад = ОстаткиСкладов.Склад;

НовСтр.Сумма = ОстаткиСкладов.Сумма;

// выводим общую сумму в отчет

ОбластьИтогоСклад.Параметры.Стоимость = ОстаткиСкладов.Сумма;

ТабДок.Присоединить(ОбластьИтогоСклад);

КонецЦикла;

// вывод "итого по предприятию"

ТабДок.Вывести(ОбластьОбщееИтого);

// сортируем таблицу средневзвешанных цен по обыванию цен

СрЦ.Сортировать("Цена Убыв");

// получаем количество, которое необходимо распределить на оставшиеся склады

Для Каждого Стр Из СрЦ Цикл

Стр.Количество = КоличествоНаСкладе(Дата, ЛиквидируемыйСклад, Стр.СО);

КонецЦикла;

// остатки на складах по количеству, без учета ликвидированного склада

ОстаткиСкл = ОстаткиТоваров(Дата, ЛиквидируемыйСклад, , "<>");

// массив для хранения сумм спецодежды после переоценки

МассивСО = Новый ТаблицаЗначений;

МассивСО.Колонки.Добавить("Склад");

МассивСО.Колонки.Добавить("Сумма");

МассивСО.Колонки.Добавить("ДляРаспределения");

// переоценка по средневзвешанной цене

Для Каждого Стр Из ОстаткиСкл Цикл

// добавление склада в массив, если его еще там нет

Если МассивСО.Найти(Стр.Склад) = Неопределено Тогда

НовСтрСкл = МассивСО.Добавить();

НовСтрСкл.Склад = Стр.Склад;

НовСтрСкл.Сумма = 0;

НовСтрСкл.ДляРаспределения = 0;

КонецЕсли;

Для Каждого СтрСО Из СрЦ Цикл

Если Стр.СО = СтрСО.СО Тогда

Стр.Сумма = Стр.Количество \* СтрСО.Цена;

// получаем общую стоимостную оценку по складам после переоценки (с учетом средневзвешанных цен)

МассивСО.Найти(Стр.Склад, "Склад").Сумма = МассивСО.Найти(Стр.Склад, "Склад").Сумма + Стр.Сумма;

Прервать;

КонецЕсли;

КонецЦикла;

КонецЦикла;

// вычисляем сумму, которой не хватает до средней стоимостной оценки всех товаров на каждом складе после ликвидации

Для Каждого Стр Из МассивСО Цикл

Стр.ДляРаспределения = ВсегоНаПредприятии / МассивСО.Количество() - Стр.Сумма;

КонецЦикла;

МассивСО.Сортировать("ДляРаспределения Убыв");

Распределение = Новый ТаблицаЗначений;

Распределение.Колонки.Добавить("Склад");

Распределение.Колонки.Добавить("СО");

Распределение.Колонки.Добавить("Количество");

Распределение.Колонки.Добавить("Сумма");

//поиск оптимального перемещения товаров с ликвидируемого склада на оставшиеся два

//////////////////////////////////////////////////////////////////////////////////

//Для поиска оптимального варианта размещения спецодежды ликвидируемого склада и

//оставшихся двух складов использую следующий алогритм: сначала сортируем склады

//по убыванию по сумме, которой не хватает до средней стоимостной оценки (для этого

//общую сумму по предприятию делим на два и вычитаем из нее стоимостную оценку после

//переоценки находящуюся на каждом складе). Затем, чтобы отклонение от стоимостной

//оценки, которую необходимо набрать было минимальным, начинаем перемещение с самых

//дорогих (по переоценочным ценам) товарам, что дает возможность заполнить и маленькое

//отклонение, перемещая товары от дорогих к дешевым. При этом заполняем таблицу

//для посчета суммы и количества перемещенных товаров на каждый склад.

/////////////////////////////////////////////////////////////////////////////////

Для Каждого Стр Из МассивСО Цикл

Для Каждого СтрСО Из СрЦ Цикл

Пока СтрСО.Количество > 0 Цикл

Если Стр.ДляРаспределения - СтрСО.Цена < 0 Тогда

Прервать;

КонецЕсли;

СтрСО.Количество = СтрСО.Количество - 1;

Стр.ДляРаспределения = Стр.ДляРаспределения - СтрСО.Цена;

Стр.Сумма = Стр.Сумма + СтрСО.Цена;

НовСтр = Распределение.Добавить();

НовСтр.Склад = Стр.Склад;

НовСтр.СО = СтрСО.СО;

НовСтр.Количество = 1;

НовСтр.Сумма = СтрСО.Цена;

КонецЦикла;

КонецЦикла;

КонецЦикла;

// если не все до конца распределилось, то ищем куда лучше отнести последний товар, если он есть

ОстатокРаспределения = МассивСО.Итог("ДляРаспределения");

СрЦ.Сортировать("Количество Убыв, Цена Убыв");

МассивСО.Сортировать("Сумма Убыв");

Для Каждого Стр Из СрЦ Цикл

// выбираем товар, который остался (если остался)

Если Стр.Количество > 0 Тогда

Стр.Количество = Стр.Количество - 1;

МассивСО[1].ДляРаспределения = МассивСО[1].ДляРаспределения - Стр.Цена;

МассивСО[1].Сумма = МассивСО[1].Сумма + Стр.Цена;

// и выбираем для перемещения тот склад, добавив в который отклонение будет минимальным

НовСтр = Распределение.Добавить();

НовСтр.Склад = МассивСО[1].Склад;

НовСтр.СО = Стр.СО;

НовСтр.Количество = 1;

НовСтр.Сумма = Стр.Цена;

КонецЕсли;

КонецЦикла;

// сворачиваем таблицу с распределенными товарами, для посчета суммы и количества

Распределение.Свернуть("Склад, СО", "Количество, Сумма");

Распределение.Сортировать("Склад Возр, СО Возр");

// вывод разрыва между отчетами

ТабДок.Вывести(ОбластьРазрыв);

// вывод заголовка второй таблицы

ОбластьЗаголовка.Параметры.Заголовок = "Предлагаемый вариант размещения спецодежды на двух складах";

ТабДок.Вывести(ОбластьЗаголовка);

// выводим шапку таблицы для каждого склада

ТабДок.Вывести(ОбластьШапкаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

Пока ВыборкаСкладов.Следующий() Цикл

Если ВыборкаСкладов.Ссылка <> ЛиквидируемыйСклад Тогда

ОбластьШапкаСклад.Параметры.Склад = ВыборкаСкладов.Наименование;

ТабДок.Присоединить(ОбластьШапкаСклад);

КонецЕсли;

КонецЦикла;

// массив общей стоимостной оценки по складам после ликвидации вабранного склада

ОбщиеИтоговыеСуммы = Новый Массив;

ОбщиеИтоговыеСуммы.Добавить(0);

ОбщиеИтоговыеСуммы.Добавить(0);

// вывод спецодежды по складам (по количеству и сумме)

ВыборкаСО = Справочники.Материалы.Выбрать();

Пока ВыборкаСО.Следующий() Цикл

ОбластьСтрокаОбщая.Параметры.Спецодежда = ВыборкаСО.Ссылка;

ОбластьСтрокаОбщая.Параметры.Цена = СрЦ.Найти(ВыборкаСО.Ссылка, "СО").Цена;

ТабДок.Вывести(ОбластьСтрокаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

н = 0;

Пока ВыборкаСкладов.Следующий() Цикл

Если ВыборкаСкладов.Ссылка <> ЛиквидируемыйСклад Тогда

Остатки = ОстаткиТоваров(Дата, ВыборкаСкладов.Ссылка, ВыборкаСО.Ссылка);

Отбор = Новый Структура;

Отбор.Вставить("СО", ВыборкаСО.Ссылка);

Отбор.Вставить("Склад", ВыборкаСкладов.Ссылка);

РаспределеннаяСО = Распределение.НайтиСтроки(Отбор);

// ищем остатки на тех складах, которые не ликвидировали

// количество считаем как остатки + распределенные товары на данный склад

Если Остатки.Количество() = 0 Тогда

ОбщееКоличество = ?(РаспределеннаяСО.Количество() = 0, 0, РаспределеннаяСО[0].Количество);

Иначе

ОбщееКоличество = Остатки[0].Количество + ?(РаспределеннаяСО.Количество() = 0, 0, РаспределеннаяСО[0].Количество);

КонецЕсли;

ОбластьСтрокаСклад.Параметры.Количество = ОбщееКоличество;

ОбщиеИтоговыеСуммы[н] = ОбщиеИтоговыеСуммы[н] + ОбщееКоличество \* СрЦ.Найти(ВыборкаСО.Ссылка, "СО").Цена;

// сумму считаем, как количество (посчитанное выше) \* на цену (после переоценки по средневзвешанной цене)

ОбластьСтрокаСклад.Параметры.Стоимость = ОбщееКоличество \* СрЦ.Найти(ВыборкаСО.Ссылка, "СО").Цена;

ТабДок.Присоединить(ОбластьСтрокаСклад);

н = н + 1;

КонецЕсли;

КонецЦикла;

КонецЦикла;

ТабДок.Вывести(ОбластьИтогоОбщая);

// вывод общих итоговых сумм по складам

Для н = 0 по ОбщиеИтоговыеСуммы.Количество() - 1 Цикл

// выводим общую сумму в отчет

ОбластьИтогоСклад.Параметры.Стоимость = ОбщиеИтоговыеСуммы[н];

ТабДок.Присоединить(ОбластьИтогоСклад);

КонецЦикла;

КонецПроцедуры

Основная процедура, вызываемая при нажатии на кнопку «Сформировать отчет», расположена в модуле формы. В ней реализовано формирование документа-результата и распределение товаров с ликвидируемого склада на остальные.

// таблица средневзвешанных цен

СрЦ = Новый ТаблицаЗначений;

СрЦ.Колонки.Добавить("СО");

СрЦ.Колонки.Добавить("Цена");

СрЦ.Колонки.Добавить("Количество");

ТабДок = ЭлементыФормы.ПолеВывода;

ТабДок.Очистить();

В данном блоке создается таблица значений для хранения средневзвешанных цен, с колонками СО (Спецодежда), Цена и Количество. Далее поле вывода для отображения результата записывается в переменную ТабДок и очищается.

// получение областей макета

Макет = ПолучитьМакет("ТекущиеОстатки");

ОбластьЗаголовка = Макет.ПолучитьОбласть("Заголовок");

ОбластьШапкаОбщая = Макет.ПолучитьОбласть("Шапка|Общее");

ОбластьШапкаСклад = Макет.ПолучитьОбласть("Шапка|Склад");

ОбластьСтрокаОбщая = Макет.ПолучитьОбласть("Строка|Общее");

ОбластьСтрокаСклад = Макет.ПолучитьОбласть("Строка|Склад");

ОбластьИтогоОбщая = Макет.ПолучитьОбласть("Итого|Общее");

ОбластьИтогоСклад = Макет.ПолучитьОбласть("Итого|Склад");

ОбластьОбщееИтого = Макет.ПолучитьОбласть("ИтогоОбщее");

ОбластьРазрыв = Макет.ПолучитьОбласть("Разрыв");

В данном блоке проводится получение областей макета, для вывода результата.

// вывод заголовка первой таблицы

ОбластьЗаголовка.Параметры.Заголовок = "Запасы спецодежды на складах предприятия на " + Формат(Дата, "ДФ=dd.MM.yyyy");

ТабДок.Вывести(ОбластьЗаголовка);

// вывод всех складов

ТабДок.Вывести(ОбластьШапкаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

Пока ВыборкаСкладов.Следующий() Цикл

ОбластьШапкаСклад.Параметры.Склад = ВыборкаСкладов.Наименование;

ТабДок.Присоединить(ОбластьШапкаСклад);

КонецЦикла;

В данном блоке в табличный документ результата выводится заголовок и шапка таблицы – "Спецодежда","Цена","Склад 1","Склад 2","Склад 3". При выводе складов просто обходим выборку из складов, получаемую через Справочники.Склады.Выбрать().

ВыборкаСО = Справочники.Материалы.Выбрать();

Пока ВыборкаСО.Следующий() Цикл

ОбластьСтрокаОбщая.Параметры.Спецодежда = ВыборкаСО.Ссылка;

// заполнения таблицы средневзвешанных цен

НовСтр = СрЦ.Добавить();

НовСтр.СО = ВыборкаСО.Ссылка;

НовСтр.Цена = СредневзвешаннаяЦена(Дата, ВыборкаСО.Ссылка);

ОбластьСтрокаОбщая.Параметры.Цена = НовСтр.Цена;

ТабДок.Вывести(ОбластьСтрокаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

Пока ВыборкаСкладов.Следующий() Цикл

// для каждой спецодежды каждого склада вычисляем остатки

Остатки = ОстаткиТоваров(Дата, ВыборкаСкладов.Ссылка, ВыборкаСО.Ссылка);

Если Остатки.Количество() = 0 Тогда

ОбластьСтрокаСклад.Параметры.Количество = 0;

ОбластьСтрокаСклад.Параметры.Стоимость = 0;

Иначе

ОбластьСтрокаСклад.Параметры.Количество = Остатки[0].Количество;

ОбластьСтрокаСклад.Параметры.Стоимость = Остатки[0].Сумма;

КонецЕсли;

// и выводим их в отчет

ТабДок.Присоединить(ОбластьСтрокаСклад);

КонецЦикла;

КонецЦикла;

В данном блоке заполняется таблица значений средневзвешанных цен. Для этого берется выборка всей спецодежды (через Справочники.Материалы.Выбрать). Происходит обход выборки, с выводом в табличный документ.

// заполнения таблицы средневзвешанных цен

НовСтр = СрЦ.Добавить();

НовСтр.СО = ВыборкаСО.Ссылка;

НовСтр.Цена = СредневзвешаннаяЦена(Дата, ВыборкаСО.Ссылка);

Данный блок используется для заполнения таблицы значений. СрЦ.Добавить – создает новую строку в таблице значений и возвращает её для заполнения. Далее данные строки заполняются: СО – ссылкой на материал, а цена – значением возвращенным функцией СредневзвешаннаяЦена на указанную дату (указанная в форме) и для текущего материала.

После добавления новой записи в таблицу значений делается вывод общей части таблицы (Спецодежда и Цена). После этого берется выборка складов (через Справочники.Склады.Выбрать) и обходится, с высчитыванием остатков для каждого склада, используя функцию ОстаткиТоваров на указанную дату, для текущего склада и текущего материала. После получения остатков Количество и Сумма выводятся в табличный документ результата.

ТабДок.Вывести(ОбластьИтогоОбщая);

// общая стоимостная оценка спецодежды на складах

СОСклада = Новый ТаблицаЗначений;

СОСклада.Колонки.Добавить("Склад");

СОСклада.Колонки.Добавить("Сумма");

// получаем остатки, не учитывая разрез по материалам и складам

ОстаткиСклады = ОстаткиНаСкладах(Дата).Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);

ОстаткиСклады.Следующий();

// остатки по всем складам и всей СО (сумма остатков по всем складам)

ВсегоНаПредприятии = ОстаткиСклады.Сумма;

ОбластьОбщееИтого.Параметры.Стоимость = ОстаткиСклады.Сумма;

ОстаткиСкладов = ОстаткиСклады.Выбрать();

Пока ОстаткиСкладов.Следующий() Цикл

// заполняем таблицу общей стоиомостной оценки складов

НовСтр = СОСклада.Добавить();

НовСтр.Склад = ОстаткиСкладов.Склад;

НовСтр.Сумма = ОстаткиСкладов.Сумма;

// выводим общую сумму в отчет

ОбластьИтогоСклад.Параметры.Стоимость = ОстаткиСкладов.Сумма;

ТабДок.Присоединить(ОбластьИтогоСклад);

КонецЦикла;

// вывод "итого по предприятию"

ТабДок.Вывести(ОбластьОбщееИтого);

В данном блоке выводятся итоги по каждому складу и общий итог по предприятию. Для хранения информации по остаткам на складах создается таблица значений СОСклада с колонками Склад и Сумма. Для подсчета итогов по каждому складу используется функция ОстаткиНаСкладах, с указанием даты указанной в форме, остальные параметры функции берутся по умолчанию (при вызове не указываются). Так как функция возвращает результат запроса, для обхода данных требуется получить выборку, используя Выбрать, и для получения результата по всему предприятию и каждому складу используется ОбходРезультатаЗапроса.ПоГруппировкам. Тогда первым элементом выборки будет общая сумма на предприятии, которая записывается в переменную ВсегоНаПредприятии. А далее получаются остатки по каждому складу, для этого берется выборка (через Выбрать) без указания обхода и при обходе выборки заполняется таблица значений СОСклада и выводится в табличный документ. По завершению в табличный документ выводится общий итог по предприятию. На этом завершается первая часть задачи – вывести текущие остатки спецодежды на предприятии.

// сортируем таблицу средневзвешанных цен по обыванию цен

СрЦ.Сортировать("Цена Убыв");

// получаем количество, которое необходимо распределить на оставшиеся склады

Для Каждого Стр Из СрЦ Цикл

Стр.Количество = КоличествоНаСкладе(Дата, ЛиквидируемыйСклад, Стр.СО);

КонецЦикла;

В данном блоке таблица значений средневзвешанных цен сортируется по убыванию цены, чтобы самая дорогая спецодежда шла в начале, а та что дешевле в конце (для распределения на склады). После сортировки заполняется последняя колонка таблицы значений – Количество. В данное поле будет записано количество товара на ликвидируемом складе, которое предстоит распределить на другие склады. Для получения этого количества используется функция КоличествоНаСкладе с указанием даты, ликвидируемого склада (и дата и склад указываются в форме) и спецодежды для которой считается количество.

// остатки на складах по количеству, без учета ликвидированного склада

ОстаткиСкл = ОстаткиТоваров(Дата, ЛиквидируемыйСклад, , "<>");

// массив для хранения сумм спецодежды после переоценки

МассивСО = Новый ТаблицаЗначений;

МассивСО.Колонки.Добавить("Склад");

МассивСО.Колонки.Добавить("Сумма");

МассивСО.Колонки.Добавить("ДляРаспределения");

// переоценка по средневзвешанной цене

Для Каждого Стр Из ОстаткиСкл Цикл

// добавление склада в массив, если его еще там нет

Если МассивСО.Найти(Стр.Склад) = Неопределено Тогда

НовСтрСкл = МассивСО.Добавить();

НовСтрСкл.Склад = Стр.Склад;

НовСтрСкл.Сумма = 0;

НовСтрСкл.ДляРаспределения = 0;

КонецЕсли;

Для Каждого СтрСО Из СрЦ Цикл

Если Стр.СО = СтрСО.СО Тогда

Стр.Сумма = Стр.Количество \* СтрСО.Цена;

// получаем общую стоимостную оценку по складам после переоценки (с учетом средневзвешанных цен)

МассивСО.Найти(Стр.Склад, "Склад").Сумма = МассивСО.Найти(Стр.Склад, "Склад").Сумма + Стр.Сумма;

Прервать;

КонецЕсли;

КонецЦикла;

КонецЦикла;

// вычисляем сумму, которой не хватает до средней стоимостной оценки всех товаров на каждом складе после ликвидации

Для Каждого Стр Из МассивСО Цикл

Стр.ДляРаспределения = ВсегоНаПредприятии / МассивСО.Количество() - Стр.Сумма;

КонецЦикла;

МассивСО.Сортировать("ДляРаспределения Убыв");

В данном блоке происходит переоценка сумм остатков на остающихся складах. Для хранения данных создается таблица значений МассивСО с колонками Склад, Сумма, ДляРаспределения. В колонке ДляРаспределения будет храниться сумма, которой не хватает до средней стоимостной оценки после ликвидации. Для получения остатков используется функция ОстаткиТоваров с указанием даты и ликвидируемого склада (указанные в форме), и с условием "<>" – что означает, что остатки берутся для всех складов кроме ликвидируемого. Полученные остатки обходятся, если склад на котором хранится текущий материал не найден в таблице значений МассивСО – то в таблицу значений добавляется новая строка с этим складом. Далее в таблице значений СрЦ (средневзвешанные цены) ищется строка хранящая данные о текущей спецодежде, для получения средневзвешанной цены. Когда запись найдена – вычисляется сумма остатков на складе (количество на складе \* средневзвешанную цену материала), добавлется сумма этого материала к общей сумме по складу в таблице значений МассивСО и цикл прерывается, т.к. запись уже найдена. После обхода остатков для всех записей таблицы значений МассивСО высчитывается сумма ДляРаспределения по формуле ВсегоНаПредприятии / количество складов - сумма на складе, после чего таблица значений сортируется по убыванию колонки ДляРаспределения (сначала будет заполнен склад, который требует большую сумму до средней цены, а дальше требующий меньшую).

Распределение = Новый ТаблицаЗначений;

Распределение.Колонки.Добавить("Склад");

Распределение.Колонки.Добавить("СО");

Распределение.Колонки.Добавить("Количество");

Распределение.Колонки.Добавить("Сумма");

В данном блоке создается таблица значений для хранения результатов распределения материалов по складам с колонками Склад, СО, Количество, Сумма.

Для Каждого Стр Из МассивСО Цикл

Для Каждого СтрСО Из СрЦ Цикл

Пока СтрСО.Количество > 0 Цикл

Если Стр.ДляРаспределения - СтрСО.Цена < 0 Тогда

Прервать;

КонецЕсли;

СтрСО.Количество = СтрСО.Количество - 1;

Стр.ДляРаспределения = Стр.ДляРаспределения - СтрСО.Цена;

Стр.Сумма = Стр.Сумма + СтрСО.Цена;

НовСтр = Распределение.Добавить();

НовСтр.Склад = Стр.Склад;

НовСтр.СО = СтрСО.СО;

НовСтр.Количество = 1;

НовСтр.Сумма = СтрСО.Цена;

КонецЦикла;

КонецЦикла;

КонецЦикла;

// если не все до конца распределилось, то ищем куда лучше отнести последний товар, если он есть

ОстатокРаспределения = МассивСО.Итог("ДляРаспределения");

СрЦ.Сортировать("Количество Убыв, Цена Убыв");

МассивСО.Сортировать("Сумма Убыв");

Для Каждого Стр Из СрЦ Цикл

// выбираем товар, который остался (если остался)

Если Стр.Количество > 0 Тогда

Стр.Количество = Стр.Количество - 1;

МассивСО[1].ДляРаспределения = МассивСО[1].ДляРаспределения - Стр.Цена;

МассивСО[1].Сумма = МассивСО[1].Сумма + Стр.Цена;

// и выбираем для перемещения тот склад, добавив в который отклонение будет минимальным

НовСтр = Распределение.Добавить();

НовСтр.Склад = МассивСО[1].Склад;

НовСтр.СО = Стр.СО;

НовСтр.Количество = 1;

НовСтр.Сумма = Стр.Цена;

КонецЕсли;

КонецЦикла;

Данный блок – главный, в нем происходит распределение товаров по складам. Сначала обходятся все склады в таблице значений МассивСО, для каждого склада идет обход всех материалов (таблица значений СрЦ) у которых количество (количество хранящееся на ликвидируемом складе) больше нуля, если при переносе материала на текущий склад сумма товаров не преувеличит среднюю по предприятию (ДляРаспределения – Цена < 0), то один материал переносится на текущий склад, если же при переносе будет преувеличение средней суммы, то цикл по Количеству элементов (в котором проходит перенос по одному материалу) прерывается, для скорейшего перехода к другому, более дешевому материалу. После обхода всех складов получается сумма нераспределенных товаров (через МассивСО.Итог), если сумма нераспределенных не равна нулю – значит требуется найти куда распределить оставшийся товар (Прим. Ред. – в коде пропущена данная проверка и ОстатокРаспределения не используется, хотя комментарий поставлен с описанием верного действия, т.к. реализация принадлежит победителю олимпиады 2 тура 2012 года исправление не внесено. Данное условие не необходимо, но позволяет пропустить ненужное действие с обходом всех товаров для поиска нераспределенных). Если не все товары распределены, то таблица значений средневзвешанных цен сортируется по убыванию по колонкам Количество и Цена, а таблице значений МассивСО по убыванию колонки Сумма (чтобы элемент 0 имел данные с наиболее заполненным складом, а элемент 1 с менее заполненным), далее для каждого товара, у которого нераспределенное количество больше 0 переносится по 1 элементу на склад отвечающий элементу 1 (менее заполненный).

// сворачиваем таблицу с распределенными товарами, для посчета суммы и количества

Распределение.Свернуть("Склад, СО", "Количество, Сумма");

Распределение.Сортировать("Склад Возр, СО Возр");

В данном блоке таблица значений с распределенными элементами сворачивается и сортируется по возрастанию Склада и спецодежды.

// вывод разрыва между отчетами

ТабДок.Вывести(ОбластьРазрыв);

// вывод заголовка второй таблицы

ОбластьЗаголовка.Параметры.Заголовок = "Предлагаемый вариант размещения спецодежды на двух складах";

ТабДок.Вывести(ОбластьЗаголовка);

// выводим шапку таблицы для каждого склада

ТабДок.Вывести(ОбластьШапкаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

Пока ВыборкаСкладов.Следующий() Цикл

Если ВыборкаСкладов.Ссылка <> ЛиквидируемыйСклад Тогда

ОбластьШапкаСклад.Параметры.Склад = ВыборкаСкладов.Наименование;

ТабДок.Присоединить(ОбластьШапкаСклад);

КонецЕсли;

КонецЦикла;

// массив общей стоимостной оценки по складам после ликвидации вабранного склада

ОбщиеИтоговыеСуммы = Новый Массив;

ОбщиеИтоговыеСуммы.Добавить(0);

ОбщиеИтоговыеСуммы.Добавить(0);

// вывод спецодежды по складам (по количеству и сумме)

ВыборкаСО = Справочники.Материалы.Выбрать();

Пока ВыборкаСО.Следующий() Цикл

ОбластьСтрокаОбщая.Параметры.Спецодежда = ВыборкаСО.Ссылка;

ОбластьСтрокаОбщая.Параметры.Цена = СрЦ.Найти(ВыборкаСО.Ссылка, "СО").Цена;

ТабДок.Вывести(ОбластьСтрокаОбщая);

ВыборкаСкладов = Справочники.Склады.Выбрать();

н = 0;

Пока ВыборкаСкладов.Следующий() Цикл

Если ВыборкаСкладов.Ссылка <> ЛиквидируемыйСклад Тогда

Остатки = ОстаткиТоваров(Дата, ВыборкаСкладов.Ссылка, ВыборкаСО.Ссылка);

Отбор = Новый Структура;

Отбор.Вставить("СО", ВыборкаСО.Ссылка);

Отбор.Вставить("Склад", ВыборкаСкладов.Ссылка);

РаспределеннаяСО = Распределение.НайтиСтроки(Отбор);

// ищем остатки на тех складах, которые не ликвидировали

// количество считаем как остатки + распределенные товары на данный склад

Если Остатки.Количество() = 0 Тогда

ОбщееКоличество = ?(РаспределеннаяСО.Количество() = 0, 0, РаспределеннаяСО[0].Количество);

Иначе

ОбщееКоличество = Остатки[0].Количество + ?(РаспределеннаяСО.Количество() = 0, 0, РаспределеннаяСО[0].Количество);

КонецЕсли;

ОбластьСтрокаСклад.Параметры.Количество = ОбщееКоличество;

ОбщиеИтоговыеСуммы[н] = ОбщиеИтоговыеСуммы[н] + ОбщееКоличество \* СрЦ.Найти(ВыборкаСО.Ссылка, "СО").Цена;

// сумму считаем, как количество (посчитанное выше) \* на цену (после переоценки по средневзвешанной цене)

ОбластьСтрокаСклад.Параметры.Стоимость = ОбщееКоличество \* СрЦ.Найти(ВыборкаСО.Ссылка, "СО").Цена;

ТабДок.Присоединить(ОбластьСтрокаСклад);

н = н + 1;

КонецЕсли;

КонецЦикла;

КонецЦикла;

ТабДок.Вывести(ОбластьИтогоОбщая);

// вывод общих итоговых сумм по складам

Для н = 0 по ОбщиеИтоговыеСуммы.Количество() - 1 Цикл

// выводим общую сумму в отчет

ОбластьИтогоСклад.Параметры.Стоимость = ОбщиеИтоговыеСуммы[н];

ТабДок.Присоединить(ОбластьИтогоСклад);

КонецЦикла;

В данном блоке данные распределения выводятся в табличный документ. Сначала выводится разрыв (чтобы две таблицы не слились друг с другом), далее шапка таблицы, без ликвидируемого склада. После создаётся массив для хранения общей суммы на обоих остающихся складах. Далее берется выборка всех материалов и идет обход всех материалов, выводится спецодежда, средняя цена (ищется в таблице значений СрЦ, после обходятся все склады, кроме ликвидируемого, для них получаются остатки на указанную дату, текущего склада и текущей спецодежды, далее находятся строки из таблицы значений Распределение, которые соответствуют текущей спецодежде и складу. Далее выводится общее количество материала на складе (как сумма остатков + распределенных с ликвидируемого склада), после подсчитываются итоговые суммы текущего склада и стоимость товара на текущем складе. После вывода таблицы, выводятся итоги (итоговая стоимость по каждому складу).

Процедура ПриОткрытии()

// установка даты по умолчанию (1 апреля 2012 года)

Дата = '20120401';

// установка склада по умолчанию (склад, с максимальным значением кода)

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ ПЕРВЫЕ 1

| Склады.Ссылка КАК Склад

|ИЗ

| Справочник.Склады КАК Склады

|

|УПОРЯДОЧИТЬ ПО

| Склады.Код УБЫВ";

Результат = Запрос.Выполнить().Выгрузить();

Если Результат.Количество() <> 0 Тогда

ЛиквидируемыйСклад = Результат[0].Склад;

КонецЕсли;

КонецПроцедуры

Данная процедура расположена в модуле формы и отвечает за установку даты и склада по умолчанию (дата – 1 апреля 2012, а склад – с максимальным значением кода).

Сформированный отчет

