2008 год 1 тур

Требование

Исходная база

Предприятие ведет бухгалтерский учет в программе «1С:Бухгалтерия 8 (7.7)».

Учет материалов ведется на счете 10 в разрезе субконто «Материалы» в стоимостном и натуральном выражении. Поступление материалов отражается в учете проводками д-т 10, кр-т 60. На счете 60 аналитический учет (условно) не ведется.

В информационной базе предприятия имеются проводки за 1 квартал 2008 г. по приходу материалов. Стоимость списания для разных групп материалов может определяться методами: «по средней скользящей» или ЛИФО.

Входные данные

Предусмотреть форму диалога, в которой пользователь выбирает наименование материала из справочника «Материалы», вводит для него величину расхода в натуральном выражении и методику списания: «Средняя скользящая» или «ЛИФО» путем выбора значения перечисления «МетодСписанияМЦ».  Введенная величина расхода материала не должна превышать величину остатка этого материала в натуральном выражении на конец периода (1 квартал 2008 г.).

Требуется

В каркасной конфигурации разработать отчет, в котором для выбранного материала и заданного количества расхода предлагается такой состав и порядок следования операций расхода (относительно операций прихода), при котором обеспечивается максимизация стоимости списанных материалов (по выбранной методике списания) минимальным количеством расходных операций. При разработке отчета следует исходить из следующего:

* в один день не может быть более одной операции прихода материала одного вида;
* поступивший материал может расходоваться в этот же день;
* если в один день имеются операции прихода и расхода материалов, то считается, что операции расхода в этот день всегда следуют за операцией прихода;
* количество материалов может быть выражено только целыми числами, количественный остаток материала на любую дату не может быть отрицательным.

Выходные данные

Результат должен быть представлен в виде отчета, содержащего операции прихода выбранного материала и операции расхода, обеспечивающие выполнение условий постановки задачи. Отчет должен быть представлен в следующем виде.

Списание 15 единиц материала М-01 по методу ЛИФО

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ДАТА | Количество  приход | Сумма приход | Количество  расход | Сумма  расход |
| 05.01.2008 | 8 | 8000,00 | 8 | 8 000,00 |
| 16.01.2008 | 12 | 11400.00 |  |  |
| 11.02.2008 | 5 | 5500,00 | 7 | 7 400,00 |
| 27.03.2008 | 10 | 9000,00 |  |  |
| ИТОГО: | 35 |  | 15 | 15 400,00 |

Решение

Для решения нужно понимать методики списания, используемые в задаче. Методика "скользящая средняя" – при использовании данной методики цена товара будет равна средней арифметической цене, за период (сумма цен за период деленная на сумму количества). Методика "ЛИФО" – означает "последний пришел, первый ушел" – означает, что списываться будут первыми те товары, которые поступили последними. То есть, если сначала поступило 3 товара с ценой 30 и потом 2 товара с ценой 50, то при списании 3 товаров, будет взято 2 товара по цене 50 и 1 товар с ценой 30.

Реализация

Модуль объекта:

// Производит основной расчет данных для отчета

//

Процедура РассчитатьОтчет() Экспорт

Если МетодикаСписания = Перечисления.МетодСписанияМЦ.СредняяСкользящая Тогда // средняя

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ПриходыМатериалов.Период КАК Дата,

| ПриходыМатериалов.КоличествоОборотДт КАК КоличествоПриход,

| ПриходыМатериалов.СуммаОборотДт КАК СуммаПриход

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(&ДатаНач, &ДатаКон, День, Счет.Код = ""10"", , Субконто1 = &Материал, , ) КАК ПриходыМатериалов

|ГДЕ

| (ПриходыМатериалов.СуммаОборотДт > 0

| ИЛИ ПриходыМатериалов.КоличествоОборотДт > 0)

|

|УПОРЯДОЧИТЬ ПО

| Дата";

Запрос.УстановитьПараметр("ДатаНач" , НачалоКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("ДатаКон" , КонецКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("Материал", Материал);

Результат = Запрос.Выполнить();

РезультатРасчетов.Загрузить(Результат.Выгрузить());

Если КоличествоРасхода = РезультатРасчетов.Итог("КоличествоПриход") Тогда // списываем все (обработка специфического случая)

ПоследняяСтрока = РезультатРасчетов.Получить(РезультатРасчетов.Количество() - 1);

ПоследняяСтрока.КоличествоРасход = КоличествоРасхода;

ПоследняяСтрока.СуммаРасход = РезультатРасчетов.Итог("СуммаПриход");

Возврат;

КонецЕсли;

ВременныйРезультат.Загрузить(Результат.Выгрузить());

ОсталосьСписать = КоличествоРасхода;

Пока ОсталосьСписать > 0 Цикл

ВременнаяТаблица = Новый Запрос;

МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;

ВременнаяТаблица.МенеджерВременныхТаблиц = МенеджерВременныхТаблиц;

ВременнаяТаблица.Текст = "ВЫБРАТЬ \*

|ПОМЕСТИТЬ врТаблицаВременныхДанных

|ИЗ &ТаблицаВременныхДанных КАК ТД";

ВременнаяТаблица.УстановитьПараметр("ТаблицаВременныхДанных", ВременныйРезультат);

ВременнаяТаблица.Выполнить();

Запрос = Новый Запрос;

Запрос.МенеджерВременныхТаблиц = МенеджерВременныхТаблиц;

Запрос.Текст = "ВЫБРАТЬ

| ВложенныйЗапрос.Дата КАК Дата,

| ВложенныйЗапрос.КоличествоПриход КАК КоличествоПриход,

| ВложенныйЗапрос.СуммаПриход,

| ВложенныйЗапрос.ОстатокКоличество,

| ВложенныйЗапрос.ОстатокСумма,

| ВложенныйЗапрос.ОстатокСумма / ВложенныйЗапрос.ОстатокКоличество КАК СредняяЦена

|ИЗ

| (ВЫБРАТЬ

| Основной.Дата КАК Дата,

| СРЕДНЕЕ(Основной.КоличествоПриход) КАК КоличествоПриход,

| СРЕДНЕЕ(Основной.СуммаПриход) КАК СуммаПриход,

| СУММА(Остатки.КоличествоПриход) КАК ОстатокКоличество,

| СУММА(Остатки.СуммаПриход) КАК ОстатокСумма

| ИЗ

| (ВЫБРАТЬ

| врТаблицаВременныхДанных.Дата КАК Дата,

| врТаблицаВременныхДанных.КоличествоПриход КАК КоличествоПриход,

| врТаблицаВременныхДанных.СуммаПриход КАК СуммаПриход

| ИЗ

| врТаблицаВременныхДанных КАК врТаблицаВременныхДанных) КАК Основной

| ЛЕВОЕ СОЕДИНЕНИЕ (ВЫБРАТЬ

| врТаблицаВременныхДанных.Дата КАК Дата,

| врТаблицаВременныхДанных.КоличествоПриход КАК КоличествоПриход,

| врТаблицаВременныхДанных.СуммаПриход КАК СуммаПриход

| ИЗ

| врТаблицаВременныхДанных КАК врТаблицаВременныхДанных) КАК Остатки

| ПО Основной.Дата >= Остатки.Дата

|

| СГРУППИРОВАТЬ ПО

| Основной.Дата) КАК ВложенныйЗапрос

|

|УПОРЯДОЧИТЬ ПО

| СредняяЦена УБЫВ,

| ОстатокКоличество УБЫВ,

| Дата";

Результат = Запрос.Выполнить();

ВременныйРезультат.Загрузить(Результат.Выгрузить());

//Предупреждение("ОсталосьСписать = " + ОсталосьСписать);

Если ВременныйРезультат.Количество() > 0 Тогда

ОстатокКоличество = ВременныйРезультат[0].ОстатокКоличество;

ОстатокСумма = ВременныйРезультат[0].ОстатокСумма;

СредняяЦенаЕдиницы = ВременныйРезультат[0].СредняяЦена;

ДатаПрихода = ВременныйРезультат[0].Дата;

Если ОстатокКоличество >= ОсталосьСписать Тогда // весь остаток спишем (часть текущей партии)

ОстатокКоличество = ОсталосьСписать;

ОстатокСумма = ОсталосьСписать \* СредняяЦенаЕдиницы;

Иначе // часть остатка спишем (всю текущую партию и все предыдущие)

ВременныйРезультат.Сортировать("Дата");

Пока ВременныйРезультат.Количество() > 0 Цикл

ДатаСтроки = ВременныйРезультат[0].Дата;

ВременныйРезультат.Удалить(0);

Если ДатаСтроки = ДатаПрихода Тогда

Прервать;

КонецЕсли;

ОбработкаПрерыванияПользователя();

КонецЦикла;

КонецЕсли;

СтрокаРезультата = РезультатРасчетов.Найти(ДатаПрихода, "Дата");

Если СтрокаРезультата <> Неопределено Тогда

СтрокаРезультата.КоличествоРасход = ОстатокКоличество;

СтрокаРезультата.СуммаРасход = ОстатокСумма;

Иначе

Сообщить(" Не могу найти строку по дате!", СтатусСообщения.Важное);

КонецЕсли;

ОсталосьСписать = ОсталосьСписать - ОстатокКоличество;

Иначе

Сообщить(" Во временном результате количество " + ВременныйРезультат.Количество() + ", что не больше 0!", СтатусСообщения.Важное);

КонецЕсли;

ОбработкаПрерыванияПользователя();

КонецЦикла;

ИначеЕсли МетодикаСписания = Перечисления.МетодСписанияМЦ.ЛИФО Тогда // ЛИФО

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ПриходыМатериалов.Период КАК Дата,

| ПриходыМатериалов.КоличествоОборотДт КАК КоличествоПриход,

| ПриходыМатериалов.СуммаОборотДт КАК СуммаПриход,

| ПриходыМатериалов.СуммаОборотДт / ПриходыМатериалов.КоличествоОборотДт КАК ЦенаЕдиницы

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(&ДатаНач, &ДатаКон, День, Счет.Код = ""10"", , Субконто1 = &Материал, , ) КАК ПриходыМатериалов

|ГДЕ

| (ПриходыМатериалов.СуммаОборотДт > 0

| ИЛИ ПриходыМатериалов.КоличествоОборотДт > 0)

|

|УПОРЯДОЧИТЬ ПО

| ЦенаЕдиницы УБЫВ,

| Дата";

Запрос.УстановитьПараметр("ДатаНач" , НачалоКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("ДатаКон" , КонецКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("Материал", Материал);

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

РезультатРасчетов.Очистить();

ОсталосьСписать = КоличествоРасхода;

Пока Выборка.Следующий() Цикл

НоваяСтрока = РезультатРасчетов.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, Выборка);

Если ОсталосьСписать <> 0 Тогда

Если НоваяСтрока.КоличествоПриход >= ОсталосьСписать Тогда // весь остаток спишем

НоваяСтрока.КоличествоРасход = ОсталосьСписать;

НоваяСтрока.СуммаРасход = ОсталосьСписать \* Выборка.ЦенаЕдиницы;

ОсталосьСписать = 0;

Иначе // часть остатка спишем

НоваяСтрока.КоличествоРасход = Выборка.КоличествоПриход;

НоваяСтрока.СуммаРасход = Выборка.СуммаПриход;

ОсталосьСписать = ОсталосьСписать - Выборка.КоличествоПриход;

КонецЕсли;

КонецЕсли;

КонецЦикла;

РезультатРасчетов.Сортировать("Дата");

Итератор = РезультатРасчетов.Количество() - 1;

Пока Итератор >= 0 Цикл

Строка = РезультатРасчетов.Получить(Итератор);

//Сообщить("Дата = " + Строка.Дата);

Если Строка.КоличествоРасход <> 0 Или Строка.СуммаРасход <> 0 Тогда // обрабатываем только строки с заполненным расходом

СобранноеКоличество = Строка.КоличествоРасход;

СобраннаяСумма = Строка.СуммаРасход;

ОбратныйСчетчик = Итератор - 1;

Пока ОбратныйСчетчик >= 0 Цикл

Подстрока = РезультатРасчетов.Получить(ОбратныйСчетчик);

//Сообщить(" Подстрока дата = " + Подстрока.Дата);

СобранноеКоличество = СобранноеКоличество + Подстрока.КоличествоРасход;

СобраннаяСумма = СобраннаяСумма + Подстрока.СуммаРасход;

КоличествоПодстроки = Подстрока.КоличествоРасход;

Подстрока.КоличествоРасход = 0;

Подстрока.СуммаРасход = 0;

Если КоличествоПодстроки <> Подстрока.КоличествоПриход Тогда // дошли до той партии, которая списана только частично

Прервать;

КонецЕсли;

ОбработкаПрерыванияПользователя();

ОбратныйСчетчик = ОбратныйСчетчик - 1;

КонецЦикла;

Строка.КоличествоРасход = СобранноеКоличество;

Строка.СуммаРасход = СобраннаяСумма;

КонецЕсли;

ОбработкаПрерыванияПользователя();

Итератор = Итератор - 1;

КонецЦикла;

КонецЕсли;

КонецПроцедуры // РассчитатьОтчет()

// Производит вывод отчета в переданный параметром табличный документ

//

Процедура ВывестиОтчет(ТабДок) Экспорт

Макет = ЭтотОбъект.ПолучитьМакет("МакетОтчета");

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьШапкаТаблицы = Макет.ПолучитьОбласть("ШапкаТаблицы");

ОбластьПодвалТаблицы = Макет.ПолучитьОбласть("ПодвалТаблицы");

ОбластьДетальныхЗаписей = Макет.ПолучитьОбласть("Детали");

ТабДок.Очистить();

ЗаполнитьЗначенияСвойств(ОбластьЗаголовок.Параметры, ЭтотОбъект);

ТабДок.Вывести(ОбластьЗаголовок);

ТабДок.Вывести(ОбластьШапкаТаблицы);

ТабДок.ФиксацияСверху = 4;

Для Каждого Строка Из РезультатРасчетов Цикл

ОбластьДетальныхЗаписей.Параметры.Заполнить(Строка);

ТабДок.Вывести(ОбластьДетальныхЗаписей);

КонецЦикла;

ОбластьПодвалТаблицы.Параметры.ИтогоКоличествоПриход = РезультатРасчетов.Итог("КоличествоПриход");

ОбластьПодвалТаблицы.Параметры.ИтогоСуммаПриход = РезультатРасчетов.Итог("СуммаПриход");

ОбластьПодвалТаблицы.Параметры.ИтогоКоличествоРасход = РезультатРасчетов.Итог("КоличествоРасход");

ОбластьПодвалТаблицы.Параметры.ИтогоСуммаРасход = РезультатРасчетов.Итог("СуммаРасход");

ТабДок.Вывести(ОбластьПодвалТаблицы);

КонецПроцедуры // ВывестиОтчет()

Модуль формы:

// Проводит предварительные проверки входной информации введенной пользователем, запускает основной код расчета и формирования отчета

//

Процедура ДействияФормыСформировать(Кнопка)

// сначала всякие проверки на корректный ввод исходных данных

Если Не ЗначениеЗаполнено(Материал) Тогда

Предупреждение("Пожалуйста укажите материал по которому необходимо рассчитать и сформировать отчет!",,);

Возврат;

КонецЕсли;

Если Не ЗначениеЗаполнено(КоличествоРасхода) Тогда

Предупреждение("Пожалуйста укажите величину расхода в натуральном выражении (поле ""количество расхода"") по которой нужно рассчитать отчет!",,);

Возврат;

КонецЕсли;

Если Не ЗначениеЗаполнено(МетодикаСписания) Тогда

Предупреждение("Пожалуйста укажите методику списания по которой необходимо рассчитать и сформировать отчет!",,);

Возврат;

КонецЕсли;

// проверка остатка величины расхода материала введенной пользователем

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ОстаткиМатериала.КоличествоОстаток

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&ДатаКон, Счет.Код = ""10"", , Субконто1 = &Материал) КАК ОстаткиМатериала

|ГДЕ

| ОстаткиМатериала.КоличествоОстаток < &КоличествоРасхода";

Запрос.УстановитьПараметр("Материал" , Материал);

Запрос.УстановитьПараметр("ДатаКон" , КонецКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("КоличествоРасхода", КоличествоРасхода);

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

Если Выборка.Следующий() Тогда

Сообщить("Величина расхода материала """ + Материал + """ в натуральном выражении (" + КоличествоРасхода + " единиц) не может быть больше остатка (" + Выборка.КоличествоОстаток + " единиц) по данным на конец I квартала 2008 г.!", СтатусСообщения.Важное);

Предупреждение("Расчет отчета не выполнен!",,);

Возврат;

КонецЕсли;

// выполним расчеты

РассчитатьОтчет();

// выведем расчеты

ВывестиОтчет(ЭлементыФормы.ТабДокумент);

КонецПроцедуры // ДействияФормыСформировать()

Автор: Андрей Александрович

Пояснения к коду:

Модуль объекта:

Процедура РассчитатьОтчет() Экспорт

Если МетодикаСписания = Перечисления.МетодСписанияМЦ.СредняяСкользящая Тогда // средняя

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ПриходыМатериалов.Период КАК Дата,

| ПриходыМатериалов.КоличествоОборотДт КАК КоличествоПриход,

| ПриходыМатериалов.СуммаОборотДт КАК СуммаПриход

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(&ДатаНач, &ДатаКон, День, Счет.Код = ""10"", , Субконто1 = &Материал, , ) КАК ПриходыМатериалов

|ГДЕ

| (ПриходыМатериалов.СуммаОборотДт > 0

| ИЛИ ПриходыМатериалов.КоличествоОборотДт > 0)

|

|УПОРЯДОЧИТЬ ПО

| Дата";

Запрос.УстановитьПараметр("ДатаНач" , НачалоКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("ДатаКон" , КонецКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("Материал", Материал);

Результат = Запрос.Выполнить();

РезультатРасчетов.Загрузить(Результат.Выгрузить());

Если КоличествоРасхода = РезультатРасчетов.Итог("КоличествоПриход") Тогда // списываем все (обработка специфического случая)

ПоследняяСтрока = РезультатРасчетов.Получить(РезультатРасчетов.Количество() - 1);

ПоследняяСтрока.КоличествоРасход = КоличествоРасхода;

ПоследняяСтрока.СуммаРасход = РезультатРасчетов.Итог("СуммаПриход");

Возврат;

КонецЕсли;

ВременныйРезультат.Загрузить(Результат.Выгрузить());

ОсталосьСписать = КоличествоРасхода;

Пока ОсталосьСписать > 0 Цикл

ВременнаяТаблица = Новый Запрос;

МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;

ВременнаяТаблица.МенеджерВременныхТаблиц = МенеджерВременныхТаблиц;

ВременнаяТаблица.Текст = "ВЫБРАТЬ \*

|ПОМЕСТИТЬ врТаблицаВременныхДанных

|ИЗ &ТаблицаВременныхДанных КАК ТД";

ВременнаяТаблица.УстановитьПараметр("ТаблицаВременныхДанных", ВременныйРезультат);

ВременнаяТаблица.Выполнить();

Запрос = Новый Запрос;

Запрос.МенеджерВременныхТаблиц = МенеджерВременныхТаблиц;

Запрос.Текст = "ВЫБРАТЬ

| ВложенныйЗапрос.Дата КАК Дата,

| ВложенныйЗапрос.КоличествоПриход КАК КоличествоПриход,

| ВложенныйЗапрос.СуммаПриход,

| ВложенныйЗапрос.ОстатокКоличество,

| ВложенныйЗапрос.ОстатокСумма,

| ВложенныйЗапрос.ОстатокСумма / ВложенныйЗапрос.ОстатокКоличество КАК СредняяЦена

|ИЗ

| (ВЫБРАТЬ

| Основной.Дата КАК Дата,

| СРЕДНЕЕ(Основной.КоличествоПриход) КАК КоличествоПриход,

| СРЕДНЕЕ(Основной.СуммаПриход) КАК СуммаПриход,

| СУММА(Остатки.КоличествоПриход) КАК ОстатокКоличество,

| СУММА(Остатки.СуммаПриход) КАК ОстатокСумма

| ИЗ

| (ВЫБРАТЬ

| врТаблицаВременныхДанных.Дата КАК Дата,

| врТаблицаВременныхДанных.КоличествоПриход КАК КоличествоПриход,

| врТаблицаВременныхДанных.СуммаПриход КАК СуммаПриход

| ИЗ

| врТаблицаВременныхДанных КАК врТаблицаВременныхДанных) КАК Основной

| ЛЕВОЕ СОЕДИНЕНИЕ (ВЫБРАТЬ

| врТаблицаВременныхДанных.Дата КАК Дата,

| врТаблицаВременныхДанных.КоличествоПриход КАК КоличествоПриход,

| врТаблицаВременныхДанных.СуммаПриход КАК СуммаПриход

| ИЗ

| врТаблицаВременныхДанных КАК врТаблицаВременныхДанных) КАК Остатки

| ПО Основной.Дата >= Остатки.Дата

|

| СГРУППИРОВАТЬ ПО

| Основной.Дата) КАК ВложенныйЗапрос

|

|УПОРЯДОЧИТЬ ПО

| СредняяЦена УБЫВ,

| ОстатокКоличество УБЫВ,

| Дата";

Результат = Запрос.Выполнить();

ВременныйРезультат.Загрузить(Результат.Выгрузить());

//Предупреждение("ОсталосьСписать = " + ОсталосьСписать);

Если ВременныйРезультат.Количество() > 0 Тогда

ОстатокКоличество = ВременныйРезультат[0].ОстатокКоличество;

ОстатокСумма = ВременныйРезультат[0].ОстатокСумма;

СредняяЦенаЕдиницы = ВременныйРезультат[0].СредняяЦена;

ДатаПрихода = ВременныйРезультат[0].Дата;

Если ОстатокКоличество >= ОсталосьСписать Тогда // весь остаток спишем (часть текущей партии)

ОстатокКоличество = ОсталосьСписать;

ОстатокСумма = ОсталосьСписать \* СредняяЦенаЕдиницы;

Иначе // часть остатка спишем (всю текущую партию и все предыдущие)

ВременныйРезультат.Сортировать("Дата");

Пока ВременныйРезультат.Количество() > 0 Цикл

ДатаСтроки = ВременныйРезультат[0].Дата;

ВременныйРезультат.Удалить(0);

Если ДатаСтроки = ДатаПрихода Тогда

Прервать;

КонецЕсли;

ОбработкаПрерыванияПользователя();

КонецЦикла;

КонецЕсли;

СтрокаРезультата = РезультатРасчетов.Найти(ДатаПрихода, "Дата");

Если СтрокаРезультата <> Неопределено Тогда

СтрокаРезультата.КоличествоРасход = ОстатокКоличество;

СтрокаРезультата.СуммаРасход = ОстатокСумма;

Иначе

Сообщить(" Не могу найти строку по дате!", СтатусСообщения.Важное);

КонецЕсли;

ОсталосьСписать = ОсталосьСписать - ОстатокКоличество;

Иначе

Сообщить(" Во временном результате количество " + ВременныйРезультат.Количество() + ", что не больше 0!", СтатусСообщения.Важное);

КонецЕсли;

ОбработкаПрерыванияПользователя();

КонецЦикла;

ИначеЕсли МетодикаСписания = Перечисления.МетодСписанияМЦ.ЛИФО Тогда // ЛИФО

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ПриходыМатериалов.Период КАК Дата,

| ПриходыМатериалов.КоличествоОборотДт КАК КоличествоПриход,

| ПриходыМатериалов.СуммаОборотДт КАК СуммаПриход,

| ПриходыМатериалов.СуммаОборотДт / ПриходыМатериалов.КоличествоОборотДт КАК ЦенаЕдиницы

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(&ДатаНач, &ДатаКон, День, Счет.Код = ""10"", , Субконто1 = &Материал, , ) КАК ПриходыМатериалов

|ГДЕ

| (ПриходыМатериалов.СуммаОборотДт > 0

| ИЛИ ПриходыМатериалов.КоличествоОборотДт > 0)

|

|УПОРЯДОЧИТЬ ПО

| ЦенаЕдиницы УБЫВ,

| Дата";

Запрос.УстановитьПараметр("ДатаНач" , НачалоКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("ДатаКон" , КонецКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("Материал", Материал);

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

РезультатРасчетов.Очистить();

ОсталосьСписать = КоличествоРасхода;

Пока Выборка.Следующий() Цикл

НоваяСтрока = РезультатРасчетов.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, Выборка);

Если ОсталосьСписать <> 0 Тогда

Если НоваяСтрока.КоличествоПриход >= ОсталосьСписать Тогда // весь остаток спишем

НоваяСтрока.КоличествоРасход = ОсталосьСписать;

НоваяСтрока.СуммаРасход = ОсталосьСписать \* Выборка.ЦенаЕдиницы;

ОсталосьСписать = 0;

Иначе // часть остатка спишем

НоваяСтрока.КоличествоРасход = Выборка.КоличествоПриход;

НоваяСтрока.СуммаРасход = Выборка.СуммаПриход;

ОсталосьСписать = ОсталосьСписать - Выборка.КоличествоПриход;

КонецЕсли;

КонецЕсли;

КонецЦикла;

РезультатРасчетов.Сортировать("Дата");

Итератор = РезультатРасчетов.Количество() - 1;

Пока Итератор >= 0 Цикл

Строка = РезультатРасчетов.Получить(Итератор);

//Сообщить("Дата = " + Строка.Дата);

Если Строка.КоличествоРасход <> 0 Или Строка.СуммаРасход <> 0 Тогда // обрабатываем только строки с заполненным расходом

СобранноеКоличество = Строка.КоличествоРасход;

СобраннаяСумма = Строка.СуммаРасход;

ОбратныйСчетчик = Итератор - 1;

Пока ОбратныйСчетчик >= 0 Цикл

Подстрока = РезультатРасчетов.Получить(ОбратныйСчетчик);

//Сообщить(" Подстрока дата = " + Подстрока.Дата);

СобранноеКоличество = СобранноеКоличество + Подстрока.КоличествоРасход;

СобраннаяСумма = СобраннаяСумма + Подстрока.СуммаРасход;

КоличествоПодстроки = Подстрока.КоличествоРасход;

Подстрока.КоличествоРасход = 0;

Подстрока.СуммаРасход = 0;

Если КоличествоПодстроки <> Подстрока.КоличествоПриход Тогда // дошли до той партии, которая списана только частично

Прервать;

КонецЕсли;

ОбработкаПрерыванияПользователя();

ОбратныйСчетчик = ОбратныйСчетчик - 1;

КонецЦикла;

Строка.КоличествоРасход = СобранноеКоличество;

Строка.СуммаРасход = СобраннаяСумма;

КонецЕсли;

ОбработкаПрерыванияПользователя();

Итератор = Итератор - 1;

КонецЦикла;

КонецЕсли;

КонецПроцедуры // РассчитатьОтчет()

Данная процедура производит получение данных из базы и вычисление результата в зависимости от выбранной методики списания.

Методика списания «средняя скользящая»:

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ПриходыМатериалов.Период КАК Дата,

| ПриходыМатериалов.КоличествоОборотДт КАК КоличествоПриход,

| ПриходыМатериалов.СуммаОборотДт КАК СуммаПриход

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(&ДатаНач, &ДатаКон, День, Счет.Код = ""10"", , Субконто1 = &Материал, , ) КАК ПриходыМатериалов

|ГДЕ

| (ПриходыМатериалов.СуммаОборотДт > 0

| ИЛИ ПриходыМатериалов.КоличествоОборотДт > 0)

|

|УПОРЯДОЧИТЬ ПО

| Дата";

Запрос.УстановитьПараметр("ДатаНач" , НачалоКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("ДатаКон" , КонецКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("Материал", Материал);

Результат = Запрос.Выполнить();

РезультатРасчетов.Загрузить(Результат.Выгрузить());

В данном блоке происходит получение данных о оборотах материала и выгрузка результата в переменную РезультатРасчетов.

Если КоличествоРасхода = РезультатРасчетов.Итог("КоличествоПриход") Тогда // списываем все (обработка специфического случая)

ПоследняяСтрока = РезультатРасчетов.Получить(РезультатРасчетов.Количество() - 1);

ПоследняяСтрока.КоличествоРасход = КоличествоРасхода;

ПоследняяСтрока.СуммаРасход = РезультатРасчетов.Итог("СуммаПриход");

Возврат;

КонецЕсли;

В данном блоке проверяется частный случай – количество расходуемого товара равно общему количеству прихода, в таком случае расходуется всё. И расчет завершается.

ВременныйРезультат.Загрузить(Результат.Выгрузить());

В данной строке в переменную ВременныйРезультат выгружаются данные из базы (так же как и в РезультатРасчетов).

ОсталосьСписать = КоличествоРасхода;

Пока ОсталосьСписать > 0 Цикл

ВременнаяТаблица = Новый Запрос;

МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;

ВременнаяТаблица.МенеджерВременныхТаблиц = МенеджерВременныхТаблиц;

ВременнаяТаблица.Текст = "ВЫБРАТЬ \*

|ПОМЕСТИТЬ врТаблицаВременныхДанных

|ИЗ &ТаблицаВременныхДанных КАК ТД";

ВременнаяТаблица.УстановитьПараметр("ТаблицаВременныхДанных", ВременныйРезультат);

ВременнаяТаблица.Выполнить();

Запрос = Новый Запрос;

Запрос.МенеджерВременныхТаблиц = МенеджерВременныхТаблиц;

Запрос.Текст = "ВЫБРАТЬ

| ВложенныйЗапрос.Дата КАК Дата,

| ВложенныйЗапрос.КоличествоПриход КАК КоличествоПриход,

| ВложенныйЗапрос.СуммаПриход,

| ВложенныйЗапрос.ОстатокКоличество,

| ВложенныйЗапрос.ОстатокСумма,

| ВложенныйЗапрос.ОстатокСумма / ВложенныйЗапрос.ОстатокКоличество КАК СредняяЦена

|ИЗ

| (ВЫБРАТЬ

| Основной.Дата КАК Дата,

| СРЕДНЕЕ(Основной.КоличествоПриход) КАК КоличествоПриход,

| СРЕДНЕЕ(Основной.СуммаПриход) КАК СуммаПриход,

| СУММА(Остатки.КоличествоПриход) КАК ОстатокКоличество,

| СУММА(Остатки.СуммаПриход) КАК ОстатокСумма

| ИЗ

| (ВЫБРАТЬ

| врТаблицаВременныхДанных.Дата КАК Дата,

| врТаблицаВременныхДанных.КоличествоПриход КАК КоличествоПриход,

| врТаблицаВременныхДанных.СуммаПриход КАК СуммаПриход

| ИЗ

| врТаблицаВременныхДанных КАК врТаблицаВременныхДанных) КАК Основной

| ЛЕВОЕ СОЕДИНЕНИЕ (ВЫБРАТЬ

| врТаблицаВременныхДанных.Дата КАК Дата,

| врТаблицаВременныхДанных.КоличествоПриход КАК КоличествоПриход,

| врТаблицаВременныхДанных.СуммаПриход КАК СуммаПриход

| ИЗ

| врТаблицаВременныхДанных КАК врТаблицаВременныхДанных) КАК Остатки

| ПО Основной.Дата >= Остатки.Дата

|

| СГРУППИРОВАТЬ ПО

| Основной.Дата) КАК ВложенныйЗапрос

|

|УПОРЯДОЧИТЬ ПО

| СредняяЦена УБЫВ,

| ОстатокКоличество УБЫВ,

| Дата";

Результат = Запрос.Выполнить();

ВременныйРезультат.Загрузить(Результат.Выгрузить());

//Предупреждение("ОсталосьСписать = " + ОсталосьСписать);

Если ВременныйРезультат.Количество() > 0 Тогда

ОстатокКоличество = ВременныйРезультат[0].ОстатокКоличество;

ОстатокСумма = ВременныйРезультат[0].ОстатокСумма;

СредняяЦенаЕдиницы = ВременныйРезультат[0].СредняяЦена;

ДатаПрихода = ВременныйРезультат[0].Дата;

Если ОстатокКоличество >= ОсталосьСписать Тогда // весь остаток спишем (часть текущей партии)

ОстатокКоличество = ОсталосьСписать;

ОстатокСумма = ОсталосьСписать \* СредняяЦенаЕдиницы;

Иначе // часть остатка спишем (всю текущую партию и все предыдущие)

ВременныйРезультат.Сортировать("Дата");

Пока ВременныйРезультат.Количество() > 0 Цикл

ДатаСтроки = ВременныйРезультат[0].Дата;

ВременныйРезультат.Удалить(0);

Если ДатаСтроки = ДатаПрихода Тогда

Прервать;

КонецЕсли;

ОбработкаПрерыванияПользователя();

КонецЦикла;

КонецЕсли;

СтрокаРезультата = РезультатРасчетов.Найти(ДатаПрихода, "Дата");

Если СтрокаРезультата <> Неопределено Тогда

СтрокаРезультата.КоличествоРасход = ОстатокКоличество;

СтрокаРезультата.СуммаРасход = ОстатокСумма;

Иначе

Сообщить(" Не могу найти строку по дате!", СтатусСообщения.Важное);

КонецЕсли;

ОсталосьСписать = ОсталосьСписать - ОстатокКоличество;

Иначе

Сообщить(" Во временном результате количество " + ВременныйРезультат.Количество() + ", что не больше 0!", СтатусСообщения.Важное);

КонецЕсли;

ОбработкаПрерыванияПользователя();

КонецЦикла;

В данном блоке идет получение количества, которое осталось списать и пока остается, что списывать будет проводиться списание.

ВременнаяТаблица = Новый Запрос;

МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;

ВременнаяТаблица.МенеджерВременныхТаблиц = МенеджерВременныхТаблиц;

ВременнаяТаблица.Текст = "ВЫБРАТЬ \*

|ПОМЕСТИТЬ врТаблицаВременныхДанных

|ИЗ &ТаблицаВременныхДанных КАК ТД";

ВременнаяТаблица.УстановитьПараметр("ТаблицаВременныхДанных", ВременныйРезультат);

ВременнаяТаблица.Выполнить();

Запрос = Новый Запрос;

Запрос.МенеджерВременныхТаблиц = МенеджерВременныхТаблиц;

Запрос.Текст = "ВЫБРАТЬ

| ВложенныйЗапрос.Дата КАК Дата,

| ВложенныйЗапрос.КоличествоПриход КАК КоличествоПриход,

| ВложенныйЗапрос.СуммаПриход,

| ВложенныйЗапрос.ОстатокКоличество,

| ВложенныйЗапрос.ОстатокСумма,

| ВложенныйЗапрос.ОстатокСумма / ВложенныйЗапрос.ОстатокКоличество КАК СредняяЦена

|ИЗ

| (ВЫБРАТЬ

| Основной.Дата КАК Дата,

| СРЕДНЕЕ(Основной.КоличествоПриход) КАК КоличествоПриход,

| СРЕДНЕЕ(Основной.СуммаПриход) КАК СуммаПриход,

| СУММА(Остатки.КоличествоПриход) КАК ОстатокКоличество,

| СУММА(Остатки.СуммаПриход) КАК ОстатокСумма

| ИЗ

| (ВЫБРАТЬ

| врТаблицаВременныхДанных.Дата КАК Дата,

| врТаблицаВременныхДанных.КоличествоПриход КАК КоличествоПриход,

| врТаблицаВременныхДанных.СуммаПриход КАК СуммаПриход

| ИЗ

| врТаблицаВременныхДанных КАК врТаблицаВременныхДанных) КАК Основной

| ЛЕВОЕ СОЕДИНЕНИЕ (ВЫБРАТЬ

| врТаблицаВременныхДанных.Дата КАК Дата,

| врТаблицаВременныхДанных.КоличествоПриход КАК КоличествоПриход,

| врТаблицаВременныхДанных.СуммаПриход КАК СуммаПриход

| ИЗ

| врТаблицаВременныхДанных КАК врТаблицаВременныхДанных) КАК Остатки

| ПО Основной.Дата >= Остатки.Дата

|

| СГРУППИРОВАТЬ ПО

| Основной.Дата) КАК ВложенныйЗапрос

|

|УПОРЯДОЧИТЬ ПО

| СредняяЦена УБЫВ,

| ОстатокКоличество УБЫВ,

| Дата";

Результат = Запрос.Выполнить();

ВременныйРезультат.Загрузить(Результат.Выгрузить());

В данном блоке происходит получение данных о приходе, средней цены, количества, суммы, остатков основываясь на предыдущем временном результате. Новый результат фиксируется в переменной ВременныйРезультат.

Если ВременныйРезультат.Количество() > 0 Тогда

ОстатокКоличество = ВременныйРезультат[0].ОстатокКоличество;

ОстатокСумма = ВременныйРезультат[0].ОстатокСумма;

СредняяЦенаЕдиницы = ВременныйРезультат[0].СредняяЦена;

ДатаПрихода = ВременныйРезультат[0].Дата;

Если ОстатокКоличество >= ОсталосьСписать Тогда // весь остаток спишем (часть текущей партии)

ОстатокКоличество = ОсталосьСписать;

ОстатокСумма = ОсталосьСписать \* СредняяЦенаЕдиницы;

Иначе // часть остатка спишем (всю текущую партию и все предыдущие)

ВременныйРезультат.Сортировать("Дата");

Пока ВременныйРезультат.Количество() > 0 Цикл

ДатаСтроки = ВременныйРезультат[0].Дата;

ВременныйРезультат.Удалить(0);

Если ДатаСтроки = ДатаПрихода Тогда

Прервать;

КонецЕсли;

ОбработкаПрерыванияПользователя();

КонецЦикла;

КонецЕсли;

СтрокаРезультата = РезультатРасчетов.Найти(ДатаПрихода, "Дата");

Если СтрокаРезультата <> Неопределено Тогда

СтрокаРезультата.КоличествоРасход = ОстатокКоличество;

СтрокаРезультата.СуммаРасход = ОстатокСумма;

Иначе

Сообщить(" Не могу найти строку по дате!", СтатусСообщения.Важное);

КонецЕсли;

ОсталосьСписать = ОсталосьСписать - ОстатокКоличество;

Иначе

Сообщить(" Во временном результате количество " + ВременныйРезультат.Количество() + ", что не больше 0!", СтатусСообщения.Важное);

КонецЕсли;

В данном блоке происходит списание остатка (если остатки есть). Если оставшееся количество больше необходимого – списываем весь остаток. Иначе спишем часть остатка – текущую партию и все предыдущие до даты прихода. Далее на дату прихода указывается количество расхода и сумма.

ОбработкаПрерыванияПользователя();

Данная стока позволяет прерывать выполнение цикла сочетанием Ctrl+Break.

Методика ЛИФО:

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ПриходыМатериалов.Период КАК Дата,

| ПриходыМатериалов.КоличествоОборотДт КАК КоличествоПриход,

| ПриходыМатериалов.СуммаОборотДт КАК СуммаПриход,

| ПриходыМатериалов.СуммаОборотДт / ПриходыМатериалов.КоличествоОборотДт КАК ЦенаЕдиницы

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(&ДатаНач, &ДатаКон, День, Счет.Код = ""10"", , Субконто1 = &Материал, , ) КАК ПриходыМатериалов

|ГДЕ

| (ПриходыМатериалов.СуммаОборотДт > 0

| ИЛИ ПриходыМатериалов.КоличествоОборотДт > 0)

|

|УПОРЯДОЧИТЬ ПО

| ЦенаЕдиницы УБЫВ,

| Дата";

Запрос.УстановитьПараметр("ДатаНач" , НачалоКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("ДатаКон" , КонецКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("Материал", Материал);

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

РезультатРасчетов.Очистить();

ОсталосьСписать = КоличествоРасхода;

Пока Выборка.Следующий() Цикл

НоваяСтрока = РезультатРасчетов.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, Выборка);

Если ОсталосьСписать <> 0 Тогда

Если НоваяСтрока.КоличествоПриход >= ОсталосьСписать Тогда // весь остаток спишем

НоваяСтрока.КоличествоРасход = ОсталосьСписать;

НоваяСтрока.СуммаРасход = ОсталосьСписать \* Выборка.ЦенаЕдиницы;

ОсталосьСписать = 0;

Иначе // часть остатка спишем

НоваяСтрока.КоличествоРасход = Выборка.КоличествоПриход;

НоваяСтрока.СуммаРасход = Выборка.СуммаПриход;

ОсталосьСписать = ОсталосьСписать - Выборка.КоличествоПриход;

КонецЕсли;

КонецЕсли;

КонецЦикла;

РезультатРасчетов.Сортировать("Дата");

Итератор = РезультатРасчетов.Количество() - 1;

Пока Итератор >= 0 Цикл

Строка = РезультатРасчетов.Получить(Итератор);

//Сообщить("Дата = " + Строка.Дата);

Если Строка.КоличествоРасход <> 0 Или Строка.СуммаРасход <> 0 Тогда // обрабатываем только строки с заполненным расходом

СобранноеКоличество = Строка.КоличествоРасход;

СобраннаяСумма = Строка.СуммаРасход;

ОбратныйСчетчик = Итератор - 1;

Пока ОбратныйСчетчик >= 0 Цикл

Подстрока = РезультатРасчетов.Получить(ОбратныйСчетчик);

//Сообщить(" Подстрока дата = " + Подстрока.Дата);

СобранноеКоличество = СобранноеКоличество + Подстрока.КоличествоРасход;

СобраннаяСумма = СобраннаяСумма + Подстрока.СуммаРасход;

КоличествоПодстроки = Подстрока.КоличествоРасход;

Подстрока.КоличествоРасход = 0;

Подстрока.СуммаРасход = 0;

Если КоличествоПодстроки <> Подстрока.КоличествоПриход Тогда // дошли до той партии, которая списана только частично

Прервать;

КонецЕсли;

ОбработкаПрерыванияПользователя();

ОбратныйСчетчик = ОбратныйСчетчик - 1;

КонецЦикла;

Строка.КоличествоРасход = СобранноеКоличество;

Строка.СуммаРасход = СобраннаяСумма;

КонецЕсли;

ОбработкаПрерыванияПользователя();

Итератор = Итератор - 1;

КонецЦикла;

Сначала происходит получение данных из базы, с заполнением таблицы значений РезультатРасчетов. После заполнения таблицы, она сортируется по дате, для списания по методике (Последний пришел – первый ушел). Далее записи, с последней по первую обходятся циклом, списывая строки с заполненным расходом.

Процедура ВывестиОтчет(ТабДок) Экспорт

Макет = ЭтотОбъект.ПолучитьМакет("МакетОтчета");

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьШапкаТаблицы = Макет.ПолучитьОбласть("ШапкаТаблицы");

ОбластьПодвалТаблицы = Макет.ПолучитьОбласть("ПодвалТаблицы");

ОбластьДетальныхЗаписей = Макет.ПолучитьОбласть("Детали");

ТабДок.Очистить();

ЗаполнитьЗначенияСвойств(ОбластьЗаголовок.Параметры, ЭтотОбъект);

ТабДок.Вывести(ОбластьЗаголовок);

ТабДок.Вывести(ОбластьШапкаТаблицы);

ТабДок.ФиксацияСверху = 4;

Для Каждого Строка Из РезультатРасчетов Цикл

ОбластьДетальныхЗаписей.Параметры.Заполнить(Строка);

ТабДок.Вывести(ОбластьДетальныхЗаписей);

КонецЦикла;

ОбластьПодвалТаблицы.Параметры.ИтогоКоличествоПриход = РезультатРасчетов.Итог("КоличествоПриход");

ОбластьПодвалТаблицы.Параметры.ИтогоСуммаПриход = РезультатРасчетов.Итог("СуммаПриход");

ОбластьПодвалТаблицы.Параметры.ИтогоКоличествоРасход = РезультатРасчетов.Итог("КоличествоРасход");

ОбластьПодвалТаблицы.Параметры.ИтогоСуммаРасход = РезультатРасчетов.Итог("СуммаРасход");

ТабДок.Вывести(ОбластьПодвалТаблицы);

КонецПроцедуры // ВывестиОтчет()

Данная процедура выводит результат расчетов в табличный документ. Для этого сначала идет получение макета, далее получаются области макета, табличный документ очищается от старых данных, заполняется заголовок и выводится, а так же шапка. После этого выводятся все строки результата расчетов. И в конце выводятся итоги.

Модуль формы:

Процедура ДействияФормыСформировать(Кнопка)

// сначала всякие проверки на корректный ввод исходных данных

Если Не ЗначениеЗаполнено(Материал) Тогда

Предупреждение("Пожалуйста укажите материал по которому необходимо рассчитать и сформировать отчет!",,);

Возврат;

КонецЕсли;

Если Не ЗначениеЗаполнено(КоличествоРасхода) Тогда

Предупреждение("Пожалуйста укажите величину расхода в натуральном выражении (поле ""количество расхода"") по которой нужно рассчитать отчет!",,);

Возврат;

КонецЕсли;

Если Не ЗначениеЗаполнено(МетодикаСписания) Тогда

Предупреждение("Пожалуйста укажите методику списания по которой необходимо рассчитать и сформировать отчет!",,);

Возврат;

КонецЕсли;

// проверка остатка величины расхода материала введенной пользователем

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ОстаткиМатериала.КоличествоОстаток

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&ДатаКон, Счет.Код = ""10"", , Субконто1 = &Материал) КАК ОстаткиМатериала

|ГДЕ

| ОстаткиМатериала.КоличествоОстаток < &КоличествоРасхода";

Запрос.УстановитьПараметр("Материал" , Материал);

Запрос.УстановитьПараметр("ДатаКон" , КонецКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("КоличествоРасхода", КоличествоРасхода);

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

Если Выборка.Следующий() Тогда

Сообщить("Величина расхода материала """ + Материал + """ в натуральном выражении (" + КоличествоРасхода + " единиц) не может быть больше остатка (" + Выборка.КоличествоОстаток + " единиц) по данным на конец I квартала 2008 г.!", СтатусСообщения.Важное);

Предупреждение("Расчет отчета не выполнен!",,);

Возврат;

КонецЕсли;

// выполним расчеты

РассчитатьОтчет();

// выведем расчеты

ВывестиОтчет(ЭлементыФормы.ТабДокумент);

КонецПроцедуры // ДействияФормыСформировать()

Данная процедура обрабатывает нажатие кнопки Сформировать. Она проверяет корректность данных, начинает расчет и выводит результат.

Если Не ЗначениеЗаполнено(Материал) Тогда

Предупреждение("Пожалуйста укажите материал по которому необходимо рассчитать и сформировать отчет!",,);

Возврат;

КонецЕсли;

Если Не ЗначениеЗаполнено(КоличествоРасхода) Тогда

Предупреждение("Пожалуйста укажите величину расхода в натуральном выражении (поле ""количество расхода"") по которой нужно рассчитать отчет!",,);

Возврат;

КонецЕсли;

Если Не ЗначениеЗаполнено(МетодикаСписания) Тогда

Предупреждение("Пожалуйста укажите методику списания по которой необходимо рассчитать и сформировать отчет!",,);

Возврат;

КонецЕсли;

В данном блоке проверяется заполнение необходимых полей.

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ОстаткиМатериала.КоличествоОстаток

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&ДатаКон, Счет.Код = ""10"", , Субконто1 = &Материал) КАК ОстаткиМатериала

|ГДЕ

| ОстаткиМатериала.КоличествоОстаток < &КоличествоРасхода";

Запрос.УстановитьПараметр("Материал" , Материал);

Запрос.УстановитьПараметр("ДатаКон" , КонецКвартала(Дата("20080101")));

Запрос.УстановитьПараметр("КоличествоРасхода", КоличествоРасхода);

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

Если Выборка.Следующий() Тогда

Сообщить("Величина расхода материала """ + Материал + """ в натуральном выражении (" + КоличествоРасхода + " единиц) не может быть больше остатка (" + Выборка.КоличествоОстаток + " единиц) по данным на конец I квартала 2008 г.!", СтатусСообщения.Важное);

Предупреждение("Расчет отчета не выполнен!",,);

Возврат;

КонецЕсли;

В данном блоке идет получение данных из базы и проверяется количество материала.

// выполним расчеты

РассчитатьОтчет();

// выведем расчеты

ВывестиОтчет(ЭлементыФормы.ТабДокумент);

В данном блоке вызывается расчет отчета и вывод в табличный документ формы.

Сформированный отчет



