2007 год 2 тур

Требование

Исходная ситуация

Предприятие ведет бухгалтерский учет в программе «1С:Бухгалтерия 8.0 (7.7)».

Учет материалов ведется на счете 10 в разрезе субконто «Материалы» в стоимостном и натуральном выражении. Поступление материалов от-ражается в учете проводками д-т 10, кр-т 60. Расход материалов отражается проводками д т 20, кр-т 10. На счетах 20 и 60 аналитический учет (условно) не ведется.

В информационной базе предприятия имеются проводки за 1 квартал 2007 г. по приходу и расходу материалов. Стоимость списания материалов в них определяется по «средней скользящей».

Требуется

В каркасной конфигурации разработать программу, которая анализирует все проводки информационной базы и формирует отчет, где для выбранного материала отображается такой порядок следования операций расхода (относительно операций прихода), при котором общая сумма списанных материалов (по «средней скользящей цене») будет с точностью до копейки равна заданной величине.

При этом 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, количество операций расхода в один день не ограничено;
* даты поступления материала изменять не допускается;
* количество списываемого материала в операциях расхода изменять не допускается;
* количественный остаток материала на каждую дату не может быть отрицательным;
* при вычислениях округление выполняется по правилу: 0,5 =1.

Искомый материал и предполагаемая сумма списания вводятся в диалоге.

Если требуемого порядка следования операций расхода не существует, то необходимо выдать соответствующее сообщение.

Требования по представлению результата

Разработанный модуль отчета следует сохранить его как внешний файл. Имя файла должно включать фамилию участника, его имя (без разделителя, русскими буквами) и расширение, например, ИвановОлег.ert. Форма отчета должна соответствовать представленной в примере.

Пример

Исходные данные

Пусть ИБ содержит следующие проводки по приходу и расходу материала М-01:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Дата | Счет Дт | Субк. Дт | Счет Кт | Субк. Кт | Кол-во | Сумма | Содержание |
| 05.03.2007 | 10 | М-01 | 60 |  | 8 | 8000,00 | Приход МЦ |
| 16.03.2007 | 10 | М-01 | 60 |  | 12 | 10800,00 | Приход МЦ |
| 21.03.2007 | 10 | М-01 | 60 |  | 5 | 5500,00 | Приход МЦ |
| 27.03.2007 | 10 | М-01 | 60 |  | 10 | 9500,00 | Приход МЦ |
| 31.03.2007 | 20 |  | 10 | М-01 | 7 | 6760,00 | Расход МЦ |
| 31.03.2007 | 20 |  | 10 | М-01 | 5 | 4828,57 | Расход МЦ |
| 31.03.2007 | 20 |  | 10 | М-01 | 3 | 2897,14 | Расход МЦ |
| 31.03.2007 | 20 |  | 10 | М-01 | 8 | 7725,72 | Расход МЦ |

Необходимо

найти вариант списания материала М-01 на общую сумму 22 300,00 руб.

Результат

При заданных исходных данных искомый отчет будет иметь вид:

Вариант расхода Материала: М-01 на сумму: 22300,00

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ДАТА | Кол-во приход | Цена приход | Сумма приход | Кол-во расход | Цена расход | Сумма расход |
| 05.03.2007 | 8 | 1000,00 | 8 000,00 | 3 | 1000,00 | 3 000,00 |
|  |  |  |  | 5 | 1000,00 | 5 000,00 |
| 16.03.2007 | 12 | 900,00 | 10 800,00 | 7 | 900,00 | 6 300,00 |
| 21.03.2007 | 5 | 1100,00 | 5 500,00 | 8 | 1000,00 | 8 000,00 |
| 27.03.2007 | 10 | 950,00 | 9 500,00 |  |  |  |
| ИТОГО: | 35 |  | 33 800,00 | 23 |  | 22 300,00 |

Решение

В задаче требуется определить порядок списания материала, не изменяя количество списания. Возможно множество различных комбинаций списания, и для того, чтобы найти нужный вариант – нужно обойти дерево комбинаций, для этого можно использовать рекурсивную функцию, перебирающую варианты. Более подробно расписано в разделе Реализация.

Реализация

Перем ТипБулево;

Перем ТипЧисло;

Перем ФлагНашли;

Перем ТаблицаСписания;

Функция ПолучитьТаблицуДопустимыхОпераций(ТаблицаРасход, КоличествоОстаток)

ОбщийСписокВариантов = ТаблицаРасход.Скопировать();

ОбщийСписокВариантов.Очистить();

Для Каждого СтрокаТЧ из ТаблицаРасход Цикл

Если СтрокаТЧ.Флаг Тогда

Продолжить;

КонецЕсли;

НоваяСтрока = ОбщийСписокВариантов.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, СтрокаТЧ);

КонецЦикла;

СписокДопустимыхОпераций = Новый СписокЗначений;

Для Каждого Строка1 из ОбщийСписокВариантов Цикл

БуфКоличество = КоличествоОстаток;

Если Строка1.Количество > БуфКоличество Тогда

Продолжить;

КонецЕсли;

НовыйСписокОпераций = Новый ТаблицаЗначений;

НовыйСписокОпераций.Колонки.Добавить("Регистратор");

НовыйСписокОпераций.Колонки.Добавить("Количество", ТипЧисло);

СписокДопустимыхОпераций.Добавить(НовыйСписокОпераций);

НоваяСтрока = НовыйСписокОпераций.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, Строка1);

БуфКоличество = БуфКоличество - Строка1.Количество;

Если БуфКоличество = 0 Тогда

Прервать;

КонецЕсли;

Для Каждого Строка2 Из ОбщийСписокВариантов Цикл

Если Строка1.Регистратор = Строка2.Регистратор Тогда

Продолжить;

КонецЕсли;

Если Строка2.Количество > БуфКоличество Тогда

Продолжить;

КонецЕсли;

НоваяСтрока = НовыйСписокОпераций.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, Строка2);

БуфКоличество = БуфКоличество - Строка2.Количество;

Если БуфКоличество = 0 Тогда

Прервать;

КонецЕсли;

КонецЦикла;

КонецЦикла;

НовыйСписокОпераций = Новый ТаблицаЗначений;

НовыйСписокОпераций.Колонки.Добавить("Регистратор");

НовыйСписокОпераций.Колонки.Добавить("Количество", ТипЧисло);

СписокДопустимыхОпераций.Добавить(НовыйСписокОпераций);

Возврат СписокДопустимыхОпераций;

КонецФункции

Процедура РассчитатьДеревоРешений(ТаблицаПриход, ТаблицаРасход, НомерСтрокиПриход = 0,

КоличествоСтрокПриход, Знач КоличествоОстаток = 0, Знач СуммаОстаток = 0, Знач НайденнаяСумма = 0)

Если НомерСтрокиПриход = КоличествоСтрокПриход Тогда

Возврат;

КонецЕсли;

ОбрабатываемаяСтрока = ТаблицаПриход[НомерСтрокиПриход];

КоличествоОстаток = ОбрабатываемаяСтрока.Количество + КоличествоОстаток;

СуммаОстаток = ОбрабатываемаяСтрока.Сумма + СуммаОстаток;

МассивНайденныхСтрок = Новый Массив;

СписокДопустимыхОпераций = ПолучитьТаблицуДопустимыхОпераций(ТаблицаРасход, КоличествоОстаток);

Для Каждого ЭлементСписка из СписокДопустимыхОпераций Цикл

ТаблицаДопустимыхОпераций = ЭлементСписка.Значение;

ТаблицаДопустимыхОпераций.Колонки.Добавить("Цена", ТипЧисло);

ТаблицаДопустимыхОпераций.Колонки.Добавить("Сумма", ТипЧисло);

МассивНайденныхСтрок.Очистить();

ВсегоСписать = 0;

СуммаТекущегоУровня = 0;

СуммаОстатокБуф = СуммаОстаток;

КоличествоОстатокБуф = КоличествоОстаток;

Для Каждого СтрокаТаблицы из ТаблицаДопустимыхОпераций Цикл

ВсегоСписать = ВсегоСписать + СтрокаТаблицы.Количество;

НайденнаяСтрока = ТаблицаРасход.Найти(СтрокаТаблицы.Регистратор, "Регистратор");

НайденнаяСтрока.Флаг = Истина;

МассивНайденныхСтрок.Добавить(НайденнаяСтрока);

СтрокаТаблицы.Цена = СуммаОстатокБуф / КоличествоОстатокБуф;

СтрокаТаблицы.Сумма = СтрокаТаблицы.Количество \* СтрокаТаблицы.Цена;

СуммаТекущегоУровня = СуммаТекущегоУровня + СтрокаТаблицы.Сумма;

СуммаОстатокБуф = СуммаОстатокБуф - СтрокаТаблицы.Сумма;

КоличествоОстатокБуф = КоличествоОстатокБуф - СтрокаТаблицы.Количество;

КонецЦикла;

БуфНайденнаяСумма = НайденнаяСумма + СуммаТекущегоУровня;

Если Окр(БуфНайденнаяСумма, 2) = СуммаСписания И ТаблицаРасход.Найти(Ложь, "Флаг") = Неопределено Тогда

ФлагНашли = Истина;

ИначеЕсли БуфНайденнаяСумма < СуммаСписания Тогда

РассчитатьДеревоРешений(ТаблицаПриход, ТаблицаРасход, НомерСтрокиПриход + 1,

КоличествоСтрокПриход, КоличествоОстатокБуф,

СуммаОстатокБуф, БуфНайденнаяСумма);

КонецЕсли;

Если ФлагНашли Тогда

НоваяСтрока = ТаблицаСписания.Добавить();

НоваяСтрока.Период = ОбрабатываемаяСтрока.Период;

НоваяСтрока.Материалы = ТаблицаДопустимыхОпераций;

Возврат;

КонецЕсли;

Для Каждого СтрокаТаблицы из МассивНайденныхСтрок Цикл

СтрокаТаблицы.Флаг = Ложь;

КонецЦикла;

ОбработкаПрерыванияПользователя();

КонецЦикла;

КонецПроцедуры

Процедура ОсновнаяПанельСформировать(Кнопка)

Если Материал = Справочники.Материалы.ПустаяСсылка() Тогда

Предупреждение("Укажите материал!");

Возврат;

КонецЕсли;

Запрос = Новый Запрос("ВЫБРАТЬ

| ЖурналОперацийОбороты.Период КАК Период,

| ЖурналОперацийОбороты.КоличествоОборотДт КАК Количество,

| ЖурналОперацийОбороты.СуммаОборотДт КАК Сумма,

| ЖурналОперацийОбороты.СуммаОборотДт / ЖурналОперацийОбороты.КоличествоОборотДт КАК Цена

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(, , День, Счет = &Счет, , Субконто1 = &Материал, , ) КАК ЖурналОперацийОбороты

|

|УПОРЯДОЧИТЬ ПО

| Период");

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

Запрос.УстановитьПараметр("Материал", Материал);

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

Запрос.Текст = "ВЫБРАТЬ

| ЖурналОперацийОбороты.Регистратор,

| ЖурналОперацийОбороты.КоличествоОборотКт КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(, , Регистратор, Счет = &Счет, , Субконто1 = &Материал, , ) КАК ЖурналОперацийОбороты";

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

ТаблицаРасход.Колонки.Добавить("Флаг", ТипБулево);

Если ТаблицаПриход.Итог("Количество") < ТаблицаРасход.Итог("Количество") Тогда

Предупреждение("Поступившее количество меньше списанного количества. Списание не возможно.");

Возврат;

КонецЕсли;

ТаблицаСписания = Новый ТаблицаЗначений;

ТаблицаСписания.Колонки.Добавить("Период", Новый ОписаниеТипов("Дата"));

ТаблицаСписания.Колонки.Добавить("Материалы", Новый ОписаниеТипов("ТаблицаЗначений"));

ФлагНашли = Ложь;

НайденнаяСумма = 0;

РассчитатьДеревоРешений(ТаблицаПриход, ТаблицаРасход,,ТаблицаПриход.Количество());

ТаблицаСписания.Сортировать("Период Возр");

Макет = ПолучитьМакет("Отчет");

ДокументРезультат = ЭлементыФормы.ДокументРезультат;

ДокументРезультат.Очистить();

Если Не ФлагНашли Тогда

СтрокаБезВариантов = Макет.ПолучитьОбласть("СтрокаБезВариантов");

ДокументРезультат.Вывести(СтрокаБезВариантов);

Возврат;

КонецЕсли;

ШапкаТаблицы = Макет.ПолучитьОбласть("ШапкаТаблицы");

СтрокаТаблицы = Макет.ПолучитьОбласть("СтрокаТаблицы");

СтрокаПодвал = Макет.ПолучитьОбласть("СтрокаПодвал");

ШапкаТаблицы.Параметры.Материал = Материал;

ШапкаТаблицы.Параметры.СуммаСписания = Формат(СуммаСписания, "ЧДЦ=2");

ДокументРезультат.Вывести(ШапкаТаблицы);

СуммаРасход = 0;

КоличествоРасход = 0;

Для Каждого СтрокаПриход Из ТаблицаПриход Цикл

Для Сч = 0 По СтрокаТаблицы.Параметры.Количество() - 1 Цикл

СтрокаТаблицы.Параметры[Сч] = "";

КонецЦикла;

СтрокаТаблицы.Параметры.Период = СтрокаПриход.Период;

СтрокаТаблицы.Параметры.КоличествоПриход = СтрокаПриход.Количество;

СтрокаТаблицы.Параметры.ЦенаПриход = СтрокаПриход.Цена;

СтрокаТаблицы.Параметры.СуммаПриход = СтрокаПриход.Сумма;

НайденнаяСтрока = ТаблицаСписания.Найти(СтрокаПриход.Период, "Период");

Если НайденнаяСтрока <> Неопределено И НайденнаяСтрока.Материалы.Количество() > 0 Тогда

Для Каждого СтрокаРасход из НайденнаяСтрока.Материалы Цикл

СтрокаТаблицы.Параметры.КоличествоРасход = СтрокаРасход.Количество;

СтрокаТаблицы.Параметры.ЦенаРасход = СтрокаРасход.Цена;

СтрокаТаблицы.Параметры.СуммаРасход = СтрокаРасход.Сумма;

ДокументРезультат.Вывести(СтрокаТаблицы);

Для Сч = 0 По СтрокаТаблицы.Параметры.Количество() - 1 Цикл

СтрокаТаблицы.Параметры[Сч] = "";

КонецЦикла;

КоличествоРасход = КоличествоРасход + СтрокаРасход.Количество;

СуммаРасход = СуммаРасход + СтрокаРасход.Сумма;

КонецЦикла;

Иначе

ДокументРезультат.Вывести(СтрокаТаблицы);

КонецЕсли;

КонецЦикла;

СтрокаПодвал.Параметры.ИтогоСуммаПриход = ТаблицаПриход.Итог("Сумма");

СтрокаПодвал.Параметры.ИтогоКоличествоПриход = ТаблицаПриход.Итог("Количество");

СтрокаПодвал.Параметры.ИтогоСуммаРасход = СуммаРасход;

СтрокаПодвал.Параметры.ИтогоКоличествоРасход = КоличествоРасход;

ДокументРезультат.Вывести(СтрокаПодвал);

КонецПроцедуры

ТипБулево = Новый ОписаниеТипов("Булево");

ТипЧисло = Новый ОписаниеТипов("Число");

Автор: Наумов Сергей

Пояснения к коду:

Перем ТипБулево;

Перем ТипЧисло;

Перем ФлагНашли;

Перем ТаблицаСписания;

В данном блоке объявляются глобальные переменные, ТипБулево и ТипЧисло – для упрощения работы с описаниями типов. ФлагНашли – для индикации нахождения решения. ТаблицаСписания – для сохранения решения в виде таблицы значений.

Функция ПолучитьТаблицуДопустимыхОпераций(ТаблицаРасход, КоличествоОстаток)

ОбщийСписокВариантов = ТаблицаРасход.Скопировать();

ОбщийСписокВариантов.Очистить();

Для Каждого СтрокаТЧ из ТаблицаРасход Цикл

Если СтрокаТЧ.Флаг Тогда

Продолжить;

КонецЕсли;

НоваяСтрока = ОбщийСписокВариантов.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, СтрокаТЧ);

КонецЦикла;

СписокДопустимыхОпераций = Новый СписокЗначений;

Для Каждого Строка1 из ОбщийСписокВариантов Цикл

БуфКоличество = КоличествоОстаток;

Если Строка1.Количество > БуфКоличество Тогда

Продолжить;

КонецЕсли;

НовыйСписокОпераций = Новый ТаблицаЗначений;

НовыйСписокОпераций.Колонки.Добавить("Регистратор");

НовыйСписокОпераций.Колонки.Добавить("Количество", ТипЧисло);

СписокДопустимыхОпераций.Добавить(НовыйСписокОпераций);

НоваяСтрока = НовыйСписокОпераций.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, Строка1);

БуфКоличество = БуфКоличество - Строка1.Количество;

Если БуфКоличество = 0 Тогда

Прервать;

КонецЕсли;

Для Каждого Строка2 Из ОбщийСписокВариантов Цикл

Если Строка1.Регистратор = Строка2.Регистратор Тогда

Продолжить;

КонецЕсли;

Если Строка2.Количество > БуфКоличество Тогда

Продолжить;

КонецЕсли;

НоваяСтрока = НовыйСписокОпераций.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, Строка2);

БуфКоличество = БуфКоличество - Строка2.Количество;

Если БуфКоличество = 0 Тогда

Прервать;

КонецЕсли;

КонецЦикла;

КонецЦикла;

НовыйСписокОпераций = Новый ТаблицаЗначений;

НовыйСписокОпераций.Колонки.Добавить("Регистратор");

НовыйСписокОпераций.Колонки.Добавить("Количество", ТипЧисло);

СписокДопустимыхОпераций.Добавить(НовыйСписокОпераций);

Возврат СписокДопустимыхОпераций;

КонецФункции

Данная функция получает список допустимых операций. Результат представлен как СписокЗначений, в котором каждая запись это ТаблицаЗначений, с колонками Регистратор и Количество. В таблицу значений попадают операции с одинаковым Регистратором и допустимым количеством, то есть сумма по колонке Количество не превышает КоличествоОстаток.

ОбщийСписокВариантов = ТаблицаРасход.Скопировать();

ОбщийСписокВариантов.Очистить();

Для Каждого СтрокаТЧ из ТаблицаРасход Цикл

Если СтрокаТЧ.Флаг Тогда

Продолжить;

КонецЕсли;

НоваяСтрока = ОбщийСписокВариантов.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, СтрокаТЧ);

КонецЦикла;

В данном блоке создается копия таблицы расхода, очищается (сделано для того, чтобы взять от таблицы расхода только колонки) и заполняется теми строками таблицы расхода, у которых Флаг не равен Истина.

СписокДопустимыхОпераций = Новый СписокЗначений;

В данной строке создается СписокЗначений для хранения результата выполнения функции.

Для Каждого Строка1 из ОбщийСписокВариантов Цикл

БуфКоличество = КоличествоОстаток;

Если Строка1.Количество > БуфКоличество Тогда

Продолжить;

КонецЕсли;

НовыйСписокОпераций = Новый ТаблицаЗначений;

НовыйСписокОпераций.Колонки.Добавить("Регистратор");

НовыйСписокОпераций.Колонки.Добавить("Количество", ТипЧисло);

СписокДопустимыхОпераций.Добавить(НовыйСписокОпераций);

НоваяСтрока = НовыйСписокОпераций.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, Строка1);

БуфКоличество = БуфКоличество - Строка1.Количество;

Если БуфКоличество = 0 Тогда

Прервать;

КонецЕсли;

Для Каждого Строка2 Из ОбщийСписокВариантов Цикл

Если Строка1.Регистратор = Строка2.Регистратор Тогда

Продолжить;

КонецЕсли;

Если Строка2.Количество > БуфКоличество Тогда

Продолжить;

КонецЕсли;

НоваяСтрока = НовыйСписокОпераций.Добавить();

ЗаполнитьЗначенияСвойств(НоваяСтрока, Строка2);

БуфКоличество = БуфКоличество - Строка2.Количество;

Если БуфКоличество = 0 Тогда

Прервать;

КонецЕсли;

КонецЦикла;

КонецЦикла;

В данном блоке идет обход всего списка вариантов (те строки из таблицы расхода, у которых флаг = ложь), с выбором тех, которые при добавлении не переполняют остаток количества. При нахождении подходящего варианта, создается новая таблица значений, которая добавляется в список результата и в данную таблицу значений добавляются все строки относящиеся к одному и тому же регистратору и не вызывающие переполнение остатка количества. Когда остаток количества становится равен нулю – составление списка допустимых операций прекращается.

НовыйСписокОпераций = Новый ТаблицаЗначений;

НовыйСписокОпераций.Колонки.Добавить("Регистратор");

НовыйСписокОпераций.Колонки.Добавить("Количество", ТипЧисло);

СписокДопустимыхОпераций.Добавить(НовыйСписокОпераций);

Возврат СписокДопустимыхОпераций;

В данном блоке создается и добавляется в список пустая таблица значений, для предотвращения ошибок, в случае отсутствия возможных операций. После этого выполняется возврат сгенерированного списка.

Процедура РассчитатьДеревоРешений(ТаблицаПриход, ТаблицаРасход, НомерСтрокиПриход = 0,

КоличествоСтрокПриход, Знач КоличествоОстаток = 0, Знач СуммаОстаток = 0, Знач НайденнаяСумма = 0)

Если НомерСтрокиПриход = КоличествоСтрокПриход Тогда

Возврат;

КонецЕсли;

ОбрабатываемаяСтрока = ТаблицаПриход[НомерСтрокиПриход];

КоличествоОстаток = ОбрабатываемаяСтрока.Количество + КоличествоОстаток;

СуммаОстаток = ОбрабатываемаяСтрока.Сумма + СуммаОстаток;

МассивНайденныхСтрок = Новый Массив;

СписокДопустимыхОпераций = ПолучитьТаблицуДопустимыхОпераций(ТаблицаРасход, КоличествоОстаток);

Для Каждого ЭлементСписка из СписокДопустимыхОпераций Цикл

ТаблицаДопустимыхОпераций = ЭлементСписка.Значение;

ТаблицаДопустимыхОпераций.Колонки.Добавить("Цена", ТипЧисло);

ТаблицаДопустимыхОпераций.Колонки.Добавить("Сумма", ТипЧисло);

МассивНайденныхСтрок.Очистить();

ВсегоСписать = 0;

СуммаТекущегоУровня = 0;

СуммаОстатокБуф = СуммаОстаток;

КоличествоОстатокБуф = КоличествоОстаток;

Для Каждого СтрокаТаблицы из ТаблицаДопустимыхОпераций Цикл

ВсегоСписать = ВсегоСписать + СтрокаТаблицы.Количество;

НайденнаяСтрока = ТаблицаРасход.Найти(СтрокаТаблицы.Регистратор, "Регистратор");

НайденнаяСтрока.Флаг = Истина;

МассивНайденныхСтрок.Добавить(НайденнаяСтрока);

СтрокаТаблицы.Цена = СуммаОстатокБуф / КоличествоОстатокБуф;

СтрокаТаблицы.Сумма = СтрокаТаблицы.Количество \* СтрокаТаблицы.Цена;

СуммаТекущегоУровня = СуммаТекущегоУровня + СтрокаТаблицы.Сумма;

СуммаОстатокБуф = СуммаОстатокБуф - СтрокаТаблицы.Сумма;

КоличествоОстатокБуф = КоличествоОстатокБуф - СтрокаТаблицы.Количество;

КонецЦикла;

БуфНайденнаяСумма = НайденнаяСумма + СуммаТекущегоУровня;

Если Окр(БуфНайденнаяСумма, 2) = СуммаСписания И ТаблицаРасход.Найти(Ложь, "Флаг") = Неопределено Тогда

ФлагНашли = Истина;

ИначеЕсли БуфНайденнаяСумма < СуммаСписания Тогда

РассчитатьДеревоРешений(ТаблицаПриход, ТаблицаРасход, НомерСтрокиПриход + 1,

КоличествоСтрокПриход, КоличествоОстатокБуф,

СуммаОстатокБуф, БуфНайденнаяСумма);

КонецЕсли;

Если ФлагНашли Тогда

НоваяСтрока = ТаблицаСписания.Добавить();

НоваяСтрока.Период = ОбрабатываемаяСтрока.Период;

НоваяСтрока.Материалы = ТаблицаДопустимыхОпераций;

Возврат;

КонецЕсли;

Для Каждого СтрокаТаблицы из МассивНайденныхСтрок Цикл

СтрокаТаблицы.Флаг = Ложь;

КонецЦикла;

ОбработкаПрерыванияПользователя();

КонецЦикла;

КонецПроцедуры

Данная процедура рекурсивно обходит дерево решений, по завершению работы этой процедуры, в глобальной переменной ТаблицаСписания будет находиться результат выполнения.

Если НомерСтрокиПриход = КоличествоСтрокПриход Тогда

Возврат;

КонецЕсли;

В данном блоке проверяется, был ли достигнут конец таблицы прихода (все записи прихода проверены).

ОбрабатываемаяСтрока = ТаблицаПриход[НомерСтрокиПриход];

КоличествоОстаток = ОбрабатываемаяСтрока.Количество + КоличествоОстаток;

СуммаОстаток = ОбрабатываемаяСтрока.Сумма + СуммаОстаток;

В данном блоке получаются остатки количества и суммы для текущей строки из таблицы прихода.

МассивНайденныхСтрок = Новый Массив;

СписокДопустимыхОпераций = ПолучитьТаблицуДопустимыхОпераций(ТаблицаРасход, КоличествоОстаток);

В данном блоке создается массив для хранения строк таблицы расхода, над которыми будем работать (для управления флагом). И получаем весь список допустимых операций.

Для Каждого ЭлементСписка из СписокДопустимыхОпераций Цикл

ТаблицаДопустимыхОпераций = ЭлементСписка.Значение;

ТаблицаДопустимыхОпераций.Колонки.Добавить("Цена", ТипЧисло);

ТаблицаДопустимыхОпераций.Колонки.Добавить("Сумма", ТипЧисло);

МассивНайденныхСтрок.Очистить();

ВсегоСписать = 0;

СуммаТекущегоУровня = 0;

СуммаОстатокБуф = СуммаОстаток;

КоличествоОстатокБуф = КоличествоОстаток;

Для Каждого СтрокаТаблицы из ТаблицаДопустимыхОпераций Цикл

ВсегоСписать = ВсегоСписать + СтрокаТаблицы.Количество;

НайденнаяСтрока = ТаблицаРасход.Найти(СтрокаТаблицы.Регистратор, "Регистратор");

НайденнаяСтрока.Флаг = Истина;

МассивНайденныхСтрок.Добавить(НайденнаяСтрока);

СтрокаТаблицы.Цена = СуммаОстатокБуф / КоличествоОстатокБуф;

СтрокаТаблицы.Сумма = СтрокаТаблицы.Количество \* СтрокаТаблицы.Цена;

СуммаТекущегоУровня = СуммаТекущегоУровня + СтрокаТаблицы.Сумма;

СуммаОстатокБуф = СуммаОстатокБуф - СтрокаТаблицы.Сумма;

КоличествоОстатокБуф = КоличествоОстатокБуф - СтрокаТаблицы.Количество;

КонецЦикла;

БуфНайденнаяСумма = НайденнаяСумма + СуммаТекущегоУровня;

Если Окр(БуфНайденнаяСумма, 2) = СуммаСписания И ТаблицаРасход.Найти(Ложь, "Флаг") = Неопределено Тогда

ФлагНашли = Истина;

ИначеЕсли БуфНайденнаяСумма < СуммаСписания Тогда

РассчитатьДеревоРешений(ТаблицаПриход, ТаблицаРасход, НомерСтрокиПриход + 1,

КоличествоСтрокПриход, КоличествоОстатокБуф,

СуммаОстатокБуф, БуфНайденнаяСумма);

КонецЕсли;

Если ФлагНашли Тогда

НоваяСтрока = ТаблицаСписания.Добавить();

НоваяСтрока.Период = ОбрабатываемаяСтрока.Период;

НоваяСтрока.Материалы = ТаблицаДопустимыхОпераций;

Возврат;

КонецЕсли;

Для Каждого СтрокаТаблицы из МассивНайденныхСтрок Цикл

СтрокаТаблицы.Флаг = Ложь;

КонецЦикла;

ОбработкаПрерыванияПользователя();

КонецЦикла;

В данном блоке происходит обход списка допустимых операций. Для начала в таблицу значений допустимых операций добавляются две колонки – Цена и Сумма, типа Число. Далее очищается массив найденных строк, для заполнения его новыми строками, которые будем использовать на данной итерации цикла. Каждая строка из таблицы допустимых операций находится по регистратору в таблице расхода, с записью результата в массив найденных строк и выставлением флага в значение истина, а также рассчитывается списанная сумма, цена, сумма и сумма по уровню обновляются остатки. Далее определяется значение найденной суммы. Если найденная сумма равна сумме списания (с округлением 0.5 = 1) и в таблице расхода все строки помечены флагом истина, то запоминаем, что решение найдено и рекурсивно возвращаясь к корню вызова функции таблица списания будет заполнена. Если найденная сумма меньше суммы списания, то функция рекурсивно вызывается, для следующей строки прихода. Далее если ФлагНашли истина, значит решение найдено и требуется добавлять обрабатываемую строку в ТаблицуСписания, с возвратом из процедуры. Если же решение не найдено, то используя массив найденных строк, таблица расхода будет восстановлена к первоначальному состоянию. Так же вызываем процедуру ОбработкаПрерыванияПользователя(), для возможности остановить работу процедуры (в случае зацикливания к примеру) клавишами Ctrl+Break.

Процедура ОсновнаяПанельСформировать(Кнопка)

Если Материал = Справочники.Материалы.ПустаяСсылка() Тогда

Предупреждение("Укажите материал!");

Возврат;

КонецЕсли;

Запрос = Новый Запрос("ВЫБРАТЬ

| ЖурналОперацийОбороты.Период КАК Период,

| ЖурналОперацийОбороты.КоличествоОборотДт КАК Количество,

| ЖурналОперацийОбороты.СуммаОборотДт КАК Сумма,

| ЖурналОперацийОбороты.СуммаОборотДт / ЖурналОперацийОбороты.КоличествоОборотДт КАК Цена

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(, , День, Счет = &Счет, , Субконто1 = &Материал, , ) КАК ЖурналОперацийОбороты

|

|УПОРЯДОЧИТЬ ПО

| Период");

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

Запрос.УстановитьПараметр("Материал", Материал);

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

Запрос.Текст = "ВЫБРАТЬ

| ЖурналОперацийОбороты.Регистратор,

| ЖурналОперацийОбороты.КоличествоОборотКт КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(, , Регистратор, Счет = &Счет, , Субконто1 = &Материал, , ) КАК ЖурналОперацийОбороты";

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

ТаблицаРасход.Колонки.Добавить("Флаг", ТипБулево);

Если ТаблицаПриход.Итог("Количество") < ТаблицаРасход.Итог("Количество") Тогда

Предупреждение("Поступившее количество меньше списанного количества. Списание не возможно.");

Возврат;

КонецЕсли;

ТаблицаСписания = Новый ТаблицаЗначений;

ТаблицаСписания.Колонки.Добавить("Период", Новый ОписаниеТипов("Дата"));

ТаблицаСписания.Колонки.Добавить("Материалы", Новый ОписаниеТипов("ТаблицаЗначений"));

ФлагНашли = Ложь;

НайденнаяСумма = 0;

РассчитатьДеревоРешений(ТаблицаПриход, ТаблицаРасход,,ТаблицаПриход.Количество());

ТаблицаСписания.Сортировать("Период Возр");

Макет = ПолучитьМакет("Отчет");

ДокументРезультат = ЭлементыФормы.ДокументРезультат;

ДокументРезультат.Очистить();

Если Не ФлагНашли Тогда

СтрокаБезВариантов = Макет.ПолучитьОбласть("СтрокаБезВариантов");

ДокументРезультат.Вывести(СтрокаБезВариантов);

Возврат;

КонецЕсли;

ШапкаТаблицы = Макет.ПолучитьОбласть("ШапкаТаблицы");

СтрокаТаблицы = Макет.ПолучитьОбласть("СтрокаТаблицы");

СтрокаПодвал = Макет.ПолучитьОбласть("СтрокаПодвал");

ШапкаТаблицы.Параметры.Материал = Материал;

ШапкаТаблицы.Параметры.СуммаСписания = Формат(СуммаСписания, "ЧДЦ=2");

ДокументРезультат.Вывести(ШапкаТаблицы);

СуммаРасход = 0;

КоличествоРасход = 0;

Для Каждого СтрокаПриход Из ТаблицаПриход Цикл

Для Сч = 0 По СтрокаТаблицы.Параметры.Количество() - 1 Цикл

СтрокаТаблицы.Параметры[Сч] = "";

КонецЦикла;

СтрокаТаблицы.Параметры.Период = СтрокаПриход.Период;

СтрокаТаблицы.Параметры.КоличествоПриход = СтрокаПриход.Количество;

СтрокаТаблицы.Параметры.ЦенаПриход = СтрокаПриход.Цена;

СтрокаТаблицы.Параметры.СуммаПриход = СтрокаПриход.Сумма;

НайденнаяСтрока = ТаблицаСписания.Найти(СтрокаПриход.Период, "Период");

Если НайденнаяСтрока <> Неопределено И НайденнаяСтрока.Материалы.Количество() > 0 Тогда

Для Каждого СтрокаРасход из НайденнаяСтрока.Материалы Цикл

СтрокаТаблицы.Параметры.КоличествоРасход = СтрокаРасход.Количество;

СтрокаТаблицы.Параметры.ЦенаРасход = СтрокаРасход.Цена;

СтрокаТаблицы.Параметры.СуммаРасход = СтрокаРасход.Сумма;

ДокументРезультат.Вывести(СтрокаТаблицы);

Для Сч = 0 По СтрокаТаблицы.Параметры.Количество() - 1 Цикл

СтрокаТаблицы.Параметры[Сч] = "";

КонецЦикла;

КоличествоРасход = КоличествоРасход + СтрокаРасход.Количество;

СуммаРасход = СуммаРасход + СтрокаРасход.Сумма;

КонецЦикла;

Иначе

ДокументРезультат.Вывести(СтрокаТаблицы);

КонецЕсли;

КонецЦикла;

СтрокаПодвал.Параметры.ИтогоСуммаПриход = ТаблицаПриход.Итог("Сумма");

СтрокаПодвал.Параметры.ИтогоКоличествоПриход = ТаблицаПриход.Итог("Количество");

СтрокаПодвал.Параметры.ИтогоСуммаРасход = СуммаРасход;

СтрокаПодвал.Параметры.ИтогоКоличествоРасход = КоличествоРасход;

ДокументРезультат.Вывести(СтрокаПодвал);

КонецПроцедуры

Данная процедура вызывается при нажатии кнопки формы Сформировать. В ней происходит получение данных из базы, вызов функции поиска решения и вывод результата в табличный документ.

Если Материал = Справочники.Материалы.ПустаяСсылка() Тогда

Предупреждение("Укажите материал!");

Возврат;

КонецЕсли;

В данном блоке проверяется указан ли материал.

Запрос = Новый Запрос("ВЫБРАТЬ

| ЖурналОперацийОбороты.Период КАК Период,

| ЖурналОперацийОбороты.КоличествоОборотДт КАК Количество,

| ЖурналОперацийОбороты.СуммаОборотДт КАК Сумма,

| ЖурналОперацийОбороты.СуммаОборотДт / ЖурналОперацийОбороты.КоличествоОборотДт КАК Цена

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(, , День, Счет = &Счет, , Субконто1 = &Материал, , ) КАК ЖурналОперацийОбороты

|

|УПОРЯДОЧИТЬ ПО

| Период");

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

Запрос.УстановитьПараметр("Материал", Материал);

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

В данном блоке происходит получение таблицы приходов из базы и выгрузка в переменную.

Запрос.Текст = "ВЫБРАТЬ

| ЖурналОперацийОбороты.Регистратор,

| ЖурналОперацийОбороты.КоличествоОборотКт КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Обороты(, , Регистратор, Счет = &Счет, , Субконто1 = &Материал, , ) КАК ЖурналОперацийОбороты";

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

ТаблицаРасход.Колонки.Добавить("Флаг", ТипБулево);

В данном блоке происходит получение таблицы расходов из базы и выгрузка в переменную, а также добавление колонки Флаг.

Если ТаблицаПриход.Итог("Количество") < ТаблицаРасход.Итог("Количество") Тогда

Предупреждение("Поступившее количество меньше списанного количества. Списание не возможно.");

Возврат;

КонецЕсли;

В данном блоке проверяется корректность данных.

ТаблицаСписания = Новый ТаблицаЗначений;

ТаблицаСписания.Колонки.Добавить("Период", Новый ОписаниеТипов("Дата"));

ТаблицаСписания.Колонки.Добавить("Материалы", Новый ОписаниеТипов("ТаблицаЗначений"));

ФлагНашли = Ложь;

НайденнаяСумма = 0;

РассчитатьДеревоРешений(ТаблицаПриход, ТаблицаРасход,,ТаблицаПриход.Количество());

ТаблицаСписания.Сортировать("Период Возр");

В данном блоке создается таблица значений для хранения результата, с колонками Период и Материалы, указываются начальные данные (флаг = ложь и найденная сумма = 0) и вызывается процедура нахождения решения. После нахождения решения таблица с результатом сортируется по колонке период.

Макет = ПолучитьМакет("Отчет");

ДокументРезультат = ЭлементыФормы.ДокументРезультат;

ДокументРезультат.Очистить();

В данном блоке получается макет для вывода результата и табличный документ, в который будет выводиться результат, с очисткой от старых данных.

Если Не ФлагНашли Тогда

СтрокаБезВариантов = Макет.ПолучитьОбласть("СтрокаБезВариантов");

ДокументРезультат.Вывести(СтрокаБезВариантов);

Возврат;

КонецЕсли;

Если поиск решений не дал результатов, то в табличный документ будет выведена соответствующая область.

ШапкаТаблицы = Макет.ПолучитьОбласть("ШапкаТаблицы");

СтрокаТаблицы = Макет.ПолучитьОбласть("СтрокаТаблицы");

СтрокаПодвал = Макет.ПолучитьОбласть("СтрокаПодвал");

В данном блоке идет получение областей макета.

ШапкаТаблицы.Параметры.Материал = Материал;

ШапкаТаблицы.Параметры.СуммаСписания = Формат(СуммаСписания, "ЧДЦ=2");

ДокументРезультат.Вывести(ШапкаТаблицы);

В данном блоке выводится шапка таблицы.

СуммаРасход = 0;

КоличествоРасход = 0;

Для Каждого СтрокаПриход Из ТаблицаПриход Цикл

Для Сч = 0 По СтрокаТаблицы.Параметры.Количество() - 1 Цикл

СтрокаТаблицы.Параметры[Сч] = "";

КонецЦикла;

СтрокаТаблицы.Параметры.Период = СтрокаПриход.Период;

СтрокаТаблицы.Параметры.КоличествоПриход = СтрокаПриход.Количество;

СтрокаТаблицы.Параметры.ЦенаПриход = СтрокаПриход.Цена;

СтрокаТаблицы.Параметры.СуммаПриход = СтрокаПриход.Сумма;

НайденнаяСтрока = ТаблицаСписания.Найти(СтрокаПриход.Период, "Период");

Если НайденнаяСтрока <> Неопределено И НайденнаяСтрока.Материалы.Количество() > 0 Тогда

Для Каждого СтрокаРасход из НайденнаяСтрока.Материалы Цикл

СтрокаТаблицы.Параметры.КоличествоРасход = СтрокаРасход.Количество;

СтрокаТаблицы.Параметры.ЦенаРасход = СтрокаРасход.Цена;

СтрокаТаблицы.Параметры.СуммаРасход = СтрокаРасход.Сумма;

ДокументРезультат.Вывести(СтрокаТаблицы);

Для Сч = 0 По СтрокаТаблицы.Параметры.Количество() - 1 Цикл

СтрокаТаблицы.Параметры[Сч] = "";

КонецЦикла;

КоличествоРасход = КоличествоРасход + СтрокаРасход.Количество;

СуммаРасход = СуммаРасход + СтрокаРасход.Сумма;

КонецЦикла;

Иначе

ДокументРезультат.Вывести(СтрокаТаблицы);

КонецЕсли;

КонецЦикла;

В данном блоке выводятся операции прихода и операции списания, полученные в результате решения. А также считается сумма расхода и количество расхода.

СтрокаПодвал.Параметры.ИтогоСуммаПриход = ТаблицаПриход.Итог("Сумма");

СтрокаПодвал.Параметры.ИтогоКоличествоПриход = ТаблицаПриход.Итог("Количество");

СтрокаПодвал.Параметры.ИтогоСуммаРасход = СуммаРасход;

СтрокаПодвал.Параметры.ИтогоКоличествоРасход = КоличествоРасход;

ДокументРезультат.Вывести(СтрокаПодвал);

В данном блоке выводятся итоговые сумма и количество по расходу и приходу.

ТипБулево = Новый ОписаниеТипов("Булево");

ТипЧисло = Новый ОписаниеТипов("Число");

В данном блоке происходит получение описаний типов для глобальных переменных.

Сформированный отчет

