2013 год 2 тур

Требования

Пятая задача Кипукамайока

Исходная ситуация

В информационной базе «1С:Бухгалтерии» имеются проводки за некоторый период ведения учета. Каждая проводка содержит корреспонденцию двух счетов и сумму. Счета могут иметь субсчета первого порядка в виде: <код синтетического счета>[.<код субсчета>] . Проводки с отрицательными суммами («сторно») отсутствуют.

Требуется

Разработать внешний отчет, в котором предлагается вариант разбиения всех счетов, представленных в проводках, на минимальное количество групп, каждая из которых содержит синтетические счета, которые не корреспондируют друг с другом ни в одной из проводок, имеющихся в информационной  базе. При этом проводки, в которых корреспондируют одинаковые синтетические счета, а также проводки с нулевыми суммами учитывать не надо. Результат представить в виде отчета, форма которого приведена в примере.

Пример

**Исходные данные (проводки в ИБ)**

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Счет дебета | Счет кредита | Сумма |
| 03.03.13 | 62.1 | 90.1 | 31 000.00 |
| 10.03.13 | 76.9 | 51 | 17 000.00 |
| 12.03.13 | 26 | 51 | 150.00 |
| 14.03.13 | 60.2 | 51 | 22 000.00 |
| 17.03.13 | 26 | 60.1 | 1 200.00 |
| 19.03.13 | 76.5 | 90.1 | 41 000.00 |
| 19.03.13 | 62.1 | 76.5 | 41 000.00 |
| 23.03.13 | 51 | 62.1 | 72 000.00 |
| 24.03.13 | 60.1 | 60.2 | 22 000.00 |
| 24.03.13 | 20 | 60.1 | 22 000.00 |
| 31.03.13 | 20 | 26 | 1 350.00 |
| 31.03.13 | 90.2 | 20 | 23 350.00 |

**Результат (отчет)**

|  |  |
| --- | --- |
| Номер группы | Счета |
| 1 | 51, 20 |
| 2 | 60, 76 |
| 3 | 26, 62 |
| 4 | 90 |

 Исходные данные

Исходные данные представлены в ИБ каркасной конфигурации.

В каркасную конфигурацию включены:

План счетов бухгалтерского учета, содержащий несколько счетов.

Регистр бухгалтерии, с проводками, соответствующими примеру.

Документ для ввода проводок.

 Результаты и их представление

В результате выполнения задания должен быть разработан отчет «1С:Предприятия 8», формирующий результатную таблицу.

Для отладки модуля отчета допускается добавлять в информационную базу каркасной конфигурации дополнительные счета и проводки. Жюри будет проводить тестирование на собственных наборах данных, отличных  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 разбиения счетов на группы можно использовать следующий алгоритм: обходим все синтетические счета, участвующие в корреспонденции, для каждого, ещё не сгруппированного, собирается группа других счетов (так же не сгруппированных), проверяя на каждой итерации есть ли корреспонденция между каждым из счетов. При добавлении счета в группу (при условии, что он не корреспондирует ни с одним из группы) он помечается сгруппированным и обход продолжается, пока не будут сгруппированы все счета. Для наглядности рассмотрим пример:

Проводки:

|  |  |
| --- | --- |
| Счет дебет | Счет кредит |
| 10 | 11 |
| 20 | 11 |
| 10 | 21 |
| 21 | 20 |
| 11 | 30 |

Процесс работы алгоритма:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация | Начальные данные | | Составление группы | |
| Счет | Сгруппирован? | Итерация | Группа |
| 1 | 10 | Нет | 1 | 10 |
| 11 | Нет | 2 | 10,20 |
| 20 | Нет | 3 | 10,20,30 |
| 21 | Нет |  | |
| 30 | Нет |
| 2 | 10 | Да | 1 | 11 |
| 11 | Нет | 2 | 11,21 |
| 20 | Да |  | |
| 21 | Нет |
| 30 | Да |
| 3 | 10 | Да | Разбиение закончено | |
| 11 | Да |
| 20 | Да |
| 21 | Да |
| 30 | Да |

Для реализации алгоритма требуются следующие данные: список счетов участвующих в корреспонденции и списки корреспондирующих между собой счетов.

Реализация

Процедура КнопкаСформироватьНажатие(Кнопка)

табДок = Новый ТабличныйДокумент;

СформироватьОтчет(табДок);

табДок.Показать("Результат");

КонецПроцедуры

&НаСервере

Процедура СформироватьОтчет(табДок)

// отладка

ВремяНачала = ТекущаяУниверсальнаяДата();

Макет = ОтчетОбъект.ПолучитьМакет("Макет");

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьДанные = Макет.ПолучитьОбласть("Данные");

табДок.Вывести(ОбластьЗаголовок);

// массив счетов, участвующих в проводках

СчетаПроводок = Новый Массив();

// массив соответствий корреспонденции

ПроводкиСоответствие = Новый Массив();

// массив "сгруппирован или нет"

СчетСгруппирован = Новый Массив();

// получаем счета имеющиеся в проводках

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ РАЗЛИЧНЫЕ

| ПланСчетов.Код КАК Код,

| ПланСчетов.Родитель.Код

|ИЗ

| ПланСчетов.ПланСчетов КАК ПланСчетов,

| (ВЫБРАТЬ

| РегистрБухгалтерии.СчетДт.Код КАК СчетДтКод,

| РегистрБухгалтерии.СчетКт.Код КАК СчетКтКод

| ИЗ

| РегистрБухгалтерии.РегистрБухгалтерии КАК РегистрБухгалтерии) КАК ВложенныйЗапрос

|ГДЕ

| ПланСчетов.Код В (ВложенныйЗапрос.СчетДтКод, ВложенныйЗапрос.СчетКтКод)

|

|УПОРЯДОЧИТЬ ПО

| Код";

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

Пока Выборка.Следующий() Цикл

// получаем код синтетического счета

СинтетическийСчет = Выборка.Код;

Если Выборка.РодительКод <> null Тогда

СинтетическийСчет = Выборка.РодительКод;

КонецЕсли;

КодСинтетическогоСчета = Число(СтрЗаменить(СинтетическийСчет,".",""));

// проверяем не добавлен ли уже счет в массив

Если СчетаПроводок.Найти(КодСинтетическогоСчета) = Неопределено Тогда

СчетаПроводок.Добавить(КодСинтетическогоСчета);

// создаем массив для корреспонденции

МассивКорреспонденции = Новый Массив();

ПроводкиСоответствие.Добавить(МассивКорреспонденции);

// отладка

//Сообщить("Добавлен счет: " + КодСинтетическогоСчета);

СчетСгруппирован.Добавить(Ложь);

КонецЕсли;

КонецЦикла;

// считываем корреспонденцию

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| РегистрБухгалтерии.СчетДт.Код,

| РегистрБухгалтерии.СчетДт.Родитель.Код,

| РегистрБухгалтерии.СчетКт.Код,

| РегистрБухгалтерии.СчетКт.Родитель.Код,

| РегистрБухгалтерии.Сумма

|ИЗ

| РегистрБухгалтерии.РегистрБухгалтерии КАК РегистрБухгалтерии

|ГДЕ

| РегистрБухгалтерии.Сумма <> 0";

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

// заполняем массив счетов, учавствующих в проводках

Пока Выборка.Следующий() Цикл

// СчетДТ

СинтетическийСчетДТ = Выборка.СчетДтКод;

Если Выборка.СчетДтРодительКод <> null Тогда

СинтетическийСчетДТ = Выборка.СчетДтРодительКод;

КонецЕсли;

КодСинтетическогоСчетаДТ = Число(СтрЗаменить(СинтетическийСчетДТ,".",""));

// СчетКТ

СинтетическийСчетКТ = Выборка.СчетКтКод;

Если Выборка.СчетКтРодительКод <> null Тогда

СинтетическийСчетКТ = Выборка.СчетКтРодительКод;

КонецЕсли;

КодСинтетическогоСчетаКТ = Число(СтрЗаменить(СинтетическийСчетКТ,".",""));

// отладка

//Сообщить("синтетические счета: " + КодСинтетическогоСчетаДТ + "," + КодСинтетическогоСчетаКТ);

// проверяем разные ли счета

Если КодСинтетическогоСчетаДТ <> КодСинтетическогоСчетаКТ Тогда

// ищем счета

СчетДТ = СчетаПроводок.Найти(КодСинтетическогоСчетаДТ);

СчетКТ = СчетаПроводок.Найти(КодСинтетическогоСчетаКТ);

// получаем массивы корреспонденции

МассивКорреспонденцииДТ = ПроводкиСоответствие.Получить(СчетДТ);

МассивКорреспонденцииКТ = ПроводкиСоответствие.Получить(СчетКТ);

// проверяем не было ли между ними корреспонденции

Если МассивКорреспонденцииДТ.Найти(КодСинтетическогоСчетаКТ) = Неопределено Тогда

// добавляем в массивы данные

МассивКорреспонденцииДТ.Добавить(КодСинтетическогоСчетаКТ);

МассивКорреспонденцииКТ.Добавить(КодСинтетическогоСчетаДТ);

//Сообщить("Добавлена корреспонденция: " + КодСинтетическогоСчетаДТ + "," + КодСинтетическогоСчетаКТ);

КонецЕсли;

КонецЕсли;

КонецЦикла;

НомерГруппы = 1;

КолСчетов = СчетаПроводок.Количество();

Сгруппированно = 0;

// Обходим получившуюся корреспонденцию, отмечая сгруппированные счета

Для Индекс = 0 по СчетаПроводок.ВГраница() Цикл

// проверяем не сгруппирован ли уже счет

Если СчетСгруппирован[Индекс] = Истина Тогда

Продолжить;

КонецЕсли;

// получаем данные

Счет = СчетаПроводок.Получить(Индекс);

// устанавливаем данные перед обходом

СчетаВГруппе = Новый Массив();

СчетаВГруппе.Добавить(Индекс);

СчетСгруппирован[Индекс] = Истина;

Сгруппированно = Сгруппированно + 1;

// отладка

//Сообщить("Начало группы: " + Счет);

// обходим счета, в поиске отсутствующих в корреспонденции

Для Индекс2 = 0 по СчетаПроводок.ВГраница() Цикл

// проверяем не сгруппирован ли уже счет

Если СчетСгруппирован[Индекс2] = Истина Тогда

Продолжить;

КонецЕсли;

// получаем данные

СчетКор = СчетаПроводок.Получить(Индекс2);

// отладка

//Сообщить("Проверяем корреспонденцию группы с " + СчетКор);

// обходим группу

НеКорреспондирует = 0;

Для Каждого СчетГруппы Из СчетаВГруппе Цикл

// получаем его массив корреспонденции

МассивКорреспонденции = ПроводкиСоответствие.Получить(СчетГруппы);

// проверяем наличие корреспонденции

Если МассивКорреспонденции.Найти(СчетКор) = Неопределено Тогда

// запоминаем что не корреспондирует

НеКорреспондирует = НеКорреспондирует + 1;

КонецЕсли;

КонецЦикла;

// проверяем, со всеми ли не корреспондирует

Если НеКорреспондирует = СчетаВГруппе.Количество() Тогда

// отладка

//Сообщить("Кореспонденция не найдена");

// добавляем в группу

СчетаВГруппе.Добавить(Индекс2);

// помечаем сгруппированным

СчетСгруппирован[Индекс2] = Истина;

Сгруппированно = Сгруппированно + 1;

КонецЕсли;

КонецЦикла;

ГруппыСтр = Строка(Счет);

Для Индекс3 = 1 по СчетаВГруппе.ВГраница() Цикл

ИндексСчета = СчетаВГруппе[Индекс3];

ГруппыСтр = ГруппыСтр + ", " + СчетаПроводок.Получить(ИндексСчета);

КонецЦикла;

// выводим в табличный документ

ОбластьДанные.Параметры.НомерГруппы = НомерГруппы;

ОбластьДанные.Параметры.Счета = ГруппыСтр;

табДок.Вывести(ОбластьДанные);

// следующая группа

НомерГруппы = НомерГруппы + 1;

// проверяем все ли сгруппированы, чтобы не продолжать обход

Если Сгруппированно = КолСчетов Тогда

Прервать;

КонецЕсли;

КонецЦикла;

// отладка

ВремяВыполнения = ТекущаяУниверсальнаяДата() - ВремяНачала;

Сообщить("Время выполнения: " + ВремяВыполнения + " мс");

КонецПроцедуры

Автор: Михайлов Алексей

Пояснения к коду:

Процедура КнопкаСформироватьНажатие(Кнопка)

табДок = Новый ТабличныйДокумент;

СформироватьОтчет(табДок);

табДок.Показать("Результат");

КонецПроцедуры

Данная процедура отвечает за обработку нажатия кнопки Сформировать. В ней создается новый объект – табличный документ, в который формируется отчет (на стороне сервера) и после этого документ показывается пользователю.

Макет = ОтчетОбъект.ПолучитьМакет("Макет");

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьДанные = Макет.ПолучитьОбласть("Данные");

табДок.Вывести(ОбластьЗаголовок);

В данном блоке происходит получение макета табличного документа, его областей для отображения и вывод заголовка.

// массив счетов, участвующих в проводках

СчетаПроводок = Новый Массив();

// массив соответствий корреспонденции

ПроводкиСоответствие = Новый Массив();

// массив "сгруппирован или нет"

СчетСгруппирован = Новый Массив();

В данном блоке создаются массивы для хранения данных. СчетаПроводок – список все счетов, участвующих в проводках и которые должны быть распределены. ПроводкиСоответствие – массив содержащий массивы счетов, которые корреспондируют с соответствующим счетом. СчетСгруппирован – массив отвечающий за хранение логической переменной, отвечающей сгруппирован ли уже счет или нет.

Пример данных:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| СчетаПроводок | | ПроводкиСоответствие | | | СчетСгруппирован | |
| Индекс | Счет | Индекс | Счета | | Индекс | Значение |
| 1 | 10 | 1 | 1 | 11 | 1 | Ложь |
| 2 | 21 |
| 2 | 11 | 2 | 1 | 10 | 2 | Ложь |
| 2 | 20 |
| 3 | 30 |
| 3 | 20 | 3 | 1 | 11 | 3 | Ложь |
| 2 | 21 |
| 4 | 21 | 4 | 1 | 10 | 4 | Ложь |
| 2 | 20 |
| 5 | 30 | 5 | 1 | 11 | 5 | Ложь |

// получаем счета имеющиеся в проводках

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ РАЗЛИЧНЫЕ

| ПланСчетов.Код КАК Код,

| ПланСчетов.Родитель.Код

|ИЗ

| ПланСчетов.ПланСчетов КАК ПланСчетов,

| (ВЫБРАТЬ

| РегистрБухгалтерии.СчетДт.Код КАК СчетДтКод,

| РегистрБухгалтерии.СчетКт.Код КАК СчетКтКод

| ИЗ

| РегистрБухгалтерии.РегистрБухгалтерии КАК РегистрБухгалтерии) КАК ВложенныйЗапрос

|ГДЕ

| ПланСчетов.Код В (ВложенныйЗапрос.СчетДтКод, ВложенныйЗапрос.СчетКтКод)

|

|УПОРЯДОЧИТЬ ПО

| Код";

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

Пока Выборка.Следующий() Цикл

// получаем код синтетического счета

СинтетическийСчет = Выборка.Код;

Если Выборка.РодительКод <> null Тогда

СинтетическийСчет = Выборка.РодительКод;

КонецЕсли;

КодСинтетическогоСчета = Число(СтрЗаменить(СинтетическийСчет,".",""));

// проверяем не добавлен ли уже счет в массив

Если СчетаПроводок.Найти(КодСинтетическогоСчета) = Неопределено Тогда

СчетаПроводок.Добавить(КодСинтетическогоСчета);

// создаем массив для корреспонденции

МассивКорреспонденции = Новый Массив();

ПроводкиСоответствие.Добавить(МассивКорреспонденции);

// отладка

//Сообщить("Добавлен счет: " + КодСинтетическогоСчета);

СчетСгруппирован.Добавить(Ложь);

КонецЕсли;

КонецЦикла;

В данном блоке идет получение счетов из базы данных и запись их в массив СчетаПроводок. Выбираются только счета участвующие в проводках, с сортировкой по Коду. Получаем Код счета и Код родителя счета, чтобы отбросить субсчета и взять только синтетические. После получения данных идет обход результата, если имеется родитель – берем его код, далее преобразуем строковый код к числу и добавляем в массив СчетаПроводок (предварительно проверив нет ли уже в массиве данного счета). При добавлении так же создаем массив для корреспонденции и добавляем Ложь в СчетСгруппирован (чтобы один и тот же индекс отвечал во всех трех массивах за данные об одном счете).

// считываем корреспонденцию

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| РегистрБухгалтерии.СчетДт.Код,

| РегистрБухгалтерии.СчетДт.Родитель.Код,

| РегистрБухгалтерии.СчетКт.Код,

| РегистрБухгалтерии.СчетКт.Родитель.Код,

| РегистрБухгалтерии.Сумма

|ИЗ

| РегистрБухгалтерии.РегистрБухгалтерии КАК РегистрБухгалтерии

|ГДЕ

| РегистрБухгалтерии.Сумма <> 0";

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

// заполняем массив счетов, учавствующих в проводках

Пока Выборка.Следующий() Цикл

// СчетДТ

СинтетическийСчетДТ = Выборка.СчетДтКод;

Если Выборка.СчетДтРодительКод <> null Тогда

СинтетическийСчетДТ = Выборка.СчетДтРодительКод;

КонецЕсли;

КодСинтетическогоСчетаДТ = Число(СтрЗаменить(СинтетическийСчетДТ,".",""));

// СчетКТ

СинтетическийСчетКТ = Выборка.СчетКтКод;

Если Выборка.СчетКтРодительКод <> null Тогда

СинтетическийСчетКТ = Выборка.СчетКтРодительКод;

КонецЕсли;

КодСинтетическогоСчетаКТ = Число(СтрЗаменить(СинтетическийСчетКТ,".",""));

// отладка

//Сообщить("синтетические счета: " + КодСинтетическогоСчетаДТ + "," + КодСинтетическогоСчетаКТ);

// проверяем разные ли счета

Если КодСинтетическогоСчетаДТ <> КодСинтетическогоСчетаКТ Тогда

// ищем счета

СчетДТ = СчетаПроводок.Найти(КодСинтетическогоСчетаДТ);

СчетКТ = СчетаПроводок.Найти(КодСинтетическогоСчетаКТ);

// получаем массивы корреспонденции

МассивКорреспонденцииДТ = ПроводкиСоответствие.Получить(СчетДТ);

МассивКорреспонденцииКТ = ПроводкиСоответствие.Получить(СчетКТ);

// проверяем не было ли между ними корреспонденции

Если МассивКорреспонденцииДТ.Найти(КодСинтетическогоСчетаКТ) = Неопределено Тогда

// добавляем в массивы данные

МассивКорреспонденцииДТ.Добавить(КодСинтетическогоСчетаКТ);

МассивКорреспонденцииКТ.Добавить(КодСинтетическогоСчетаДТ);

//Сообщить("Добавлена корреспонденция: " + КодСинтетическогоСчетаДТ + "," + КодСинтетическогоСчетаКТ);

КонецЕсли;

КонецЕсли;

КонецЦикла;

В данном блоке происходит получение данных по проводкам из базы данных и заполняется массив корреспонденции. При получении данных берутся все проводки с ненулевыми суммами (одно из условий задачи). Далее при обходе результата получаются численные коды синтетических счетов дебета и кредита, и в случае если они разные (также одно из условий задачи), определяется индекс в массиве СчетаПроводок отвечающий за оба счета, берется массив корреспонденции каждого счета (расположенный по найденному индексу) и если в корреспонденции ещё нет записи о том, что эти два счета корреспондируют друг с другом – добавляются эти записи.

НомерГруппы = 1;

КолСчетов = СчетаПроводок.Количество();

Сгруппированно = 0;

// Обходим получившуюся корреспонденцию, отмечая сгруппированные счета

Для Индекс = 0 по СчетаПроводок.ВГраница() Цикл

// проверяем не сгруппирован ли уже счет

Если СчетСгруппирован[Индекс] = Истина Тогда

Продолжить;

КонецЕсли;

// получаем данные

Счет = СчетаПроводок.Получить(Индекс);

// устанавливаем данные перед обходом

СчетаВГруппе = Новый Массив();

СчетаВГруппе.Добавить(Индекс);

СчетСгруппирован[Индекс] = Истина;

Сгруппированно = Сгруппированно + 1;

// отладка

//Сообщить("Начало группы: " + Счет);

// обходим счета, в поиске отсутствующих в корреспонденции

Для Индекс2 = 0 по СчетаПроводок.ВГраница() Цикл

// проверяем не сгруппирован ли уже счет

Если СчетСгруппирован[Индекс2] = Истина Тогда

Продолжить;

КонецЕсли;

// получаем данные

СчетКор = СчетаПроводок.Получить(Индекс2);

// отладка

//Сообщить("Проверяем корреспонденцию группы с " + СчетКор);

// обходим группу

НеКорреспондирует = 0;

Для Каждого СчетГруппы Из СчетаВГруппе Цикл

// получаем его массив корреспонденции

МассивКорреспонденции = ПроводкиСоответствие.Получить(СчетГруппы);

// проверяем наличие корреспонденции

Если МассивКорреспонденции.Найти(СчетКор) = Неопределено Тогда

// запоминаем что не корреспондирует

НеКорреспондирует = НеКорреспондирует + 1;

КонецЕсли;

КонецЦикла;

// проверяем, со всеми ли не корреспондирует

Если НеКорреспондирует = СчетаВГруппе.Количество() Тогда

// отладка

//Сообщить("Кореспонденция не найдена");

// добавляем в группу

СчетаВГруппе.Добавить(Индекс2);

// помечаем сгруппированным

СчетСгруппирован[Индекс2] = Истина;

Сгруппированно = Сгруппированно + 1;

КонецЕсли;

КонецЦикла;

ГруппыСтр = Строка(Счет);

Для Индекс3 = 1 по СчетаВГруппе.ВГраница() Цикл

ИндексСчета = СчетаВГруппе[Индекс3];

ГруппыСтр = ГруппыСтр + ", " + СчетаПроводок.Получить(ИндексСчета);

КонецЦикла;

// выводим в табличный документ

ОбластьДанные.Параметры.НомерГруппы = НомерГруппы;

ОбластьДанные.Параметры.Счета = ГруппыСтр;

табДок.Вывести(ОбластьДанные);

// следующая группа

НомерГруппы = НомерГруппы + 1;

// проверяем все ли сгруппированы, чтобы не продолжать обход

Если Сгруппированно = КолСчетов Тогда

Прервать;

КонецЕсли;

КонецЦикла;

Главный блок – реализация алгоритма распределения по группам, вместе с выводом в табличный документ. Переменная НомерГруппы – отвечает за номер группы распределения, который выводится в табличный документ. КолСчетов – общее количество счетов для группировки. Сгруппировано – количество уже сгруппированных счетов. В начале итерации происходит проверка – сгруппирован ли уже текущий счет, или нет. В случае, если он уже сгруппирован – счет пропускается. Далее инициализируется массив счетов содержащихся в группе, в начале он состоит только из текущего счета. И текущий счет сразу помечается сгруппированным. Далее происходит обход всех счетов, так же пропуская уже сгруппированные и проверяется – корреспондирует ли счет с любым из счетов в массиве СчетаВГруппе. Если счет не корреспондирует ни с одним счетом из группы (переменная НеКорреспондирует = количеству счетов в группе), то счет добавляется в группу и помечается сгруппированным. При этом повышается значение счетчика Сгруппированно. После завершения составления группы она выводится в табличный документ (предварительно из массива с номерами счетов группы составляется строка), повышается счетчик НомерГруппы. В конце проверяется сгруппированы ли уже все счета, для прекращения обхода.

Сформированный отчет

