2012 год 1 тур

Требования

Крупная финансовая компания, представляющая собой группу взаимосвязанных организаций (порядка 150), ведет бухгалтерский учет в единой информационной базе в программе «1С:Бухгалтерия 8». Многофирменный учет в программе организован с использованием измерения регистра бухгалтерии «Организации».

Задолженность организаций при внутрикорпоративных расчетах отражается (условно) только на счете 76 по субконто «Организации». Значения измерения «Организации» также как и субконто вида «Организации» определяются в едином справочнике «Организации».

В процессе проведения финансовых операций возникают задолженности между организациями, входящими в состав компании. Если организация Х является должником организации Y, то это отражается в учете у организации Х как кредитовое сальдо счета 76 по субконто «Организация Y» на сумму долга, одновременно у организации Y на счете 76 по субконто «Организация Х» отражается дебетовое сальдо на эту же сумму.

Необходимо по данным бухгалтерского учета:

1. Определить внутрикорпоративные долги для организаций и общую сумму долгов, представить их в виде таблицы следующего вида.

Долги

|  |  |  |
| --- | --- | --- |
| **Должник** | **Кредитор** | **Сумма долга** |
| Орг-01 | Орг-02 | 500,00 |
| Орг-02 | Орг-03 | 200,00 |
| Орг-02 | Орг-04 | 150,00 |
| Орг-03 | Орг-01 | 250,00 |
| Орг-04 | Орг-01 | 100,00 |
| Итог |  | 1200,00 |

2. Для каждой организации определить сальдо взаиморасчетов как разницу между общей суммой долгов, причитающихся к получению и общей суммы долгов, причитающихся к возврату, в следующем виде.

Сольдо взаиморасчетов

|  |  |  |
| --- | --- | --- |
| **Организация** | **Сальдо  взаиморасчетов** |  |
| Орг-01 | -150,00 |  |
| Орг-02 | 150,00 |  |
| Орг-03 | -50,00 |  |
| Орг-04 | 50,00 |  |

3. Предложить вариант зачета взаимных требований, такой, чтобы общая сумма долгов всех организаций была бы минимальной, при этом для каждой организации сальдо взаиморасчетов оставалось бы неизменным. Результат представить в виде таблицы, аналогичной исходной.

Долги после взаимозачетов

|  |  |  |
| --- | --- | --- |
| **Должник** | **Кредитор** | **Сумма долга** |
| Орг-01 | Орг-02 | 100,00 |
| Орг-01 | Орг-04 | 50,00 |
| Орг-03 | Орг-02 | 50,00 |
| Итог |  | 200,00 |

Каркасная конфигурация

Задание необходимо выполнить в каркасной конфигурации, в которой имеется:

* план счетов, содержащий счета 76 "Расчеты с прочими дебиторами и кредиторами" и 0 "Вспомогательный";
* справочник «Организации»;
* документ для ввода данных о задолженности организаций, при помощи которого введено несколько проводок в дебет и кредит счета 76 в корреспонденции со вспомогательным счетом 0.

ВНИМАНИЕ! Для отладки модуля отчета Вы можете добавлять в информационную базу каркасной конфигурации дополнительные организации и проводки. Жюри проводит тестирование на собственных наборах данных, которые будут отличаться от данных, содержащихся в исходной информационной базе каркасной конфигурации.

Решение

Причина больших сумм долгов внутри организации кроется в циклах и цепочках задолжностей. Как пример цикла – организация 1 должна организации 2, та в свою очередь должна организации 3, а та должна организации 1. Пример цепочки долгов (цепочки видны при отсутствии циклов) – организация 1 должна организации 2, а та должна организации 4. Для уменьшения общей суммы долгов достаточно устранить циклы и цепочки долгов, чтобы все долги сводились к тому, что один занял другому ни у кого не занимая сам. Для устранения циклов достаточно избавиться от одного из составляющих его звеньев. Для устранения цепи (первый – второй – третий) достаточно, чтобы первый отдавал напрямую третьему (за второго) и второму (если остается ещё должен).

Рассмотрим ход решения на представленных в требованиях данных.

Шаг 1 – устранение циклов.

Ищем циклы, первый цикл – (1 – 2 – 3 – 1). Для устранения цикла нужно избавиться от одного из звеньев. Для этого найдем минимальный долг:

|  |  |  |
| --- | --- | --- |
| Кто | Кому | Сколько |
| Орг 1 | Орг 2 | 500 |
| Орг 2 | Орг 3 | 200 |
| Орг 3 | Орг 1 | 250 |

Из таблицы видно, что минимальный долг у организации 2 для организации 3. Теперь можно уничтожить это звено (считая что все три организации прощают друг другу 200, никто ничего не теряет). Получается следующее состояние по организации:

|  |  |  |
| --- | --- | --- |
| Кто | Кому | Сколько |
| Орг 1 | Орг 2 | 300 |
| Орг 2 | Орг 4 | 150 |
| Орг 3 | Орг 1 | 50 |
| Орг 4 | Орг 1 | 100 |

Проверяем на наличие циклов, тут он присутствует, это цикл – (1 – 2 – 4 – 1). Находим минимальный долг:

|  |  |  |
| --- | --- | --- |
| Кто | Кому | Сколько |
| Орг 1 | Орг 2 | 300 |
| Орг 2 | Орг 4 | 150 |
| Орг 4 | Орг 1 | 100 |

Минимальный долгу у организации 4 для организации 1. После уничтожения звена получается состояние:

|  |  |  |
| --- | --- | --- |
| Кто | Кому | Сколько |
| Орг 1 | Орг 2 | 200 |
| Орг 2 | Орг 4 | 50 |
| Орг 3 | Орг 1 | 50 |

Проверяем состояние на наличие циклов, и видим что они отсутствуют.

Шаг 2 – устранение цепочек.

Ищем цепочки долгов (достаточно найти и избавиться от всех цепей длиной 3, не рассматривая более длинные). Первая найденная цепочка – (1 – 2 – 4). Разбиваем её на два независимых долга. Организация 1 отдаст за организацию 2 50 организации 4, так как организации 2 все равно придется потратить 50 из полученных 200 от организации 1 на уплату долга. Так убираем посредника и снижаем общую сумму долга. После разбития цепочки получаем состояние:

|  |  |  |
| --- | --- | --- |
| Кто | Кому | Сколько |
| Орг 1 | Орг 2 | 150 |
| Орг 1 | Орг 4 | 50 |
| Орг 3 | Орг 1 | 50 |

Проверяем состояние на наличие цепочек, находим – (3 – 1 – 2). Разбиваем цепочку. Организация 3 уплатит организации 2 50, а организация 1 останется должна только 100.

|  |  |  |
| --- | --- | --- |
| Кто | Кому | Сколько |
| Орг 1 | Орг 2 | 100 |
| Орг 1 | Орг 4 | 50 |
| Орг 3 | Орг 2 | 50 |

Проверяем состояние на наличие цепочек – их в состоянии не найдено, следовательно решение найдено, и долги организаций разбиты так, что организация даёт в займы не беря ни у кого в долг сама (организации 2 и 4).

Реализация

Процедура КнопкаСформироватьНажатие(Кнопка)

ТабДок = ЭлементыФормы.ПолеТабличногоДокумента;

Долги = Новый ТаблицаЗначений;

Долги.Колонки.Добавить("Кто");

Долги.Колонки.Добавить("Кому");

Долги.Колонки.Добавить("Сумма");

Сальдо = Новый ТаблицаЗначений;

Сальдо.Колонки.Добавить("Кто");

Сальдо.Колонки.Добавить("Сумма");

Взаиморасчеты = Новый ТаблицаЗначений;

Взаиморасчеты.Колонки.Добавить("Кто");

Взаиморасчеты.Колонки.Добавить("Кому");

Взаиморасчеты.Колонки.Добавить("Сумма");

// получаем долги

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ЖурналОперацийДвиженияССубконто.Организации,

| ЖурналОперацийДвиженияССубконто.СубконтоКт1,

| ЖурналОперацийДвиженияССубконто.Сумма

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.ДвиженияССубконто КАК ЖурналОперацийДвиженияССубконто

|ГДЕ

| ЖурналОперацийДвиженияССубконто.СчетКт.Код = ""76""";

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

Пока Выборка.Следующий() Цикл

ДолгиЗапись = Долги.Добавить();

ВзаиморасчетыЗапись = Взаиморасчеты.Добавить();

ДолгиЗапись.Кто = Выборка.Организации;

ДолгиЗапись.Кому = Выборка.СубконтоКт1;

ДолгиЗапись.Сумма = Выборка.Сумма;

ВзаиморасчетыЗапись.Кто = ДолгиЗапись.Кто;

ВзаиморасчетыЗапись.Кому = ДолгиЗапись.Кому;

ВзаиморасчетыЗапись.Сумма = ДолгиЗапись.Сумма;

КонецЦикла;

// получаем сальдо

Для Каждого Долг Из Долги Цикл

СальдоЗаписьКто = Сальдо.Найти(Долг.Кто,"Кто");

СальдоЗаписьКому = Сальдо.Найти(Долг.Кому,"Кто");

Если СальдоЗаписьКто = Неопределено Тогда

СальдоЗаписьКто = Сальдо.Добавить();

СальдоЗаписьКто.Кто = Долг.Кто;

СальдоЗаписьКто.Сумма = 0;

КонецЕсли;

Если СальдоЗаписьКому = Неопределено Тогда

СальдоЗаписьКому = Сальдо.Добавить();

СальдоЗаписьКому.Кто = Долг.Кому;

СальдоЗаписьКому.Сумма = 0;

КонецЕсли;

СальдоЗаписьКто.Сумма = СальдоЗаписьКто.Сумма - Долг.Сумма;

СальдоЗаписьКому.Сумма = СальдоЗаписьКому.Сумма + Долг.Сумма;

КонецЦикла;

// расчитываем взаиморасчеты

// ищем циклы

ЦиклРезультат = НайтиЦикл(Взаиморасчеты,Неопределено);

Пока ЦиклРезультат <> Неопределено Цикл

Минимальный = ЦиклРезультат[0];

Для Каждого Звено Из ЦиклРезультат Цикл

Если Звено.Сумма < Минимальный.Сумма Тогда

Минимальный = Звено;

КонецЕсли;

КонецЦикла;

// уничтожаем цикл

Для Каждого Звено Из ЦиклРезультат Цикл

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Если ВзаиморасчетыЗапись.Кто <> Звено.Кто Тогда

Продолжить;

КонецЕсли;

Если ВзаиморасчетыЗапись.Кому <> Звено.Кому Тогда

Продолжить;

КонецЕсли;

ВзаиморасчетыЗапись.Сумма = ВзаиморасчетыЗапись.Сумма - Минимальный.Сумма;

Если ВзаиморасчетыЗапись.Сумма = 0 Тогда

Взаиморасчеты.Удалить(ВзаиморасчетыЗапись);

КонецЕсли;

КонецЦикла;

КонецЦикла;

ЦиклРезультат = НайтиЦикл(Взаиморасчеты,Неопределено);

КонецЦикла;

// ищем цепи

ЦепьРезультат = НайтиЦепь(Взаиморасчеты,Неопределено);

Пока ЦепьРезультат <> Неопределено Цикл

Запись = Новый Массив(2);

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Для i = 0 по 1 Цикл

Если ВзаиморасчетыЗапись.Кто = ЦепьРезультат[i].Кто И ВзаиморасчетыЗапись.Кому = ЦепьРезультат[i].Кому Тогда

Запись[i] = ВзаиморасчетыЗапись;

КонецЕсли;

КонецЦикла;

КонецЦикла;

Если Запись[0].Сумма > Запись[1].Сумма Тогда

Второму = Запись[0].Сумма - Запись[1].Сумма;

Запись[0].Сумма = Второму;

Запись[1].Кто = Запись[0].Кто;

ИначеЕсли Запись[0].Сумма <= Запись[1].Сумма Тогда

Третьему = Запись[1].Сумма - Запись[0].Сумма;

Запись[0].Кому = Запись[1].Кому;

Запись[1].Сумма = Третьему;

КонецЕсли;

ЦепьРезультат = НайтиЦепь(Взаиморасчеты,Неопределено);

КонецЦикла;

Макет = ПолучитьМакет("Макет");

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьШапкаВзаиморасчеты = Макет.ПолучитьОбласть("ШапкаВзаиморасчеты");

ОбластьСтрокаВзаиморасчеты = Макет.ПолучитьОбласть("СтрокаВзаиморасчеты");

ОбластьШапкаСальдо = Макет.ПолучитьОбласть("ШапкаСальдо");

ОбластьСтрокаСальдо = Макет.ПолучитьОбласть("СтрокаСальдо");

ТабДок.Очистить();

ОбластьЗаголовок.Параметры[0] = "Долги";

ТабДок.Вывести(ОбластьЗаголовок);

ТабДок.Вывести(ОбластьШапкаВзаиморасчеты);

Для Каждого Запись Из Долги Цикл

ОбластьСтрокаВзаиморасчеты.Параметры[0] = Запись.Кто;

ОбластьСтрокаВзаиморасчеты.Параметры[1] = Запись.Кому;

ОбластьСтрокаВзаиморасчеты.Параметры[2] = Запись.Сумма;

ТабДок.Вывести(ОбластьСтрокаВзаиморасчеты);

КонецЦикла;

ОбластьЗаголовок.Параметры[0] = "Сальдо";

ТабДок.Вывести(ОбластьЗаголовок);

ТабДок.Вывести(ОбластьШапкаСальдо);

Для Каждого Запись Из Сальдо Цикл

ОбластьСтрокаСальдо.Параметры[0] = Запись.Кто;

ОбластьСтрокаСальдо.Параметры[1] = Запись.Сумма;

ТабДок.Вывести(ОбластьСтрокаСальдо);

КонецЦикла;

ОбластьЗаголовок.Параметры[0] = "Взаиморасчеты";

ТабДок.Вывести(ОбластьЗаголовок);

ТабДок.Вывести(ОбластьШапкаВзаиморасчеты);

Для Каждого Запись Из Взаиморасчеты Цикл

ОбластьСтрокаВзаиморасчеты.Параметры[0] = Запись.Кто;

ОбластьСтрокаВзаиморасчеты.Параметры[1] = Запись.Кому;

ОбластьСтрокаВзаиморасчеты.Параметры[2] = Запись.Сумма;

ТабДок.Вывести(ОбластьСтрокаВзаиморасчеты);

КонецЦикла;

КонецПроцедуры

Функция НайтиЦикл(Взаиморасчеты,Результат)

РезультатВременный = Неопределено;

Если Результат = Неопределено Тогда

РезультатВременный = Новый ТаблицаЗначений;

РезультатВременный.Колонки.Добавить("Кто");

РезультатВременный.Колонки.Добавить("Кому");

РезультатВременный.Колонки.Добавить("Сумма");

Иначе

РезультатВременный = Результат.Скопировать();

КонецЕсли;

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Количество = РезультатВременный.Количество();

Если Количество > 0 Тогда

Если РезультатВременный[Количество - 1].Кому <> ВзаиморасчетыЗапись.Кто Тогда

Продолжить;

КонецЕсли;

КонецЕсли;

РезультатЗапись = РезультатВременный.Добавить();

РезультатЗапись.Кто = ВзаиморасчетыЗапись.Кто;

РезультатЗапись.Кому = ВзаиморасчетыЗапись.Кому;

РезультатЗапись.Сумма = ВзаиморасчетыЗапись.Сумма;

Если РезультатВременный.Количество() > 1 Тогда

Если РезультатВременный[0].Кто = РезультатЗапись.Кому Тогда

// Цикл найден

Возврат РезультатВременный;

КонецЕсли;

КонецЕсли;

ЦиклР = НайтиЦикл(Взаиморасчеты,РезультатВременный);

Если ЦиклР <> Неопределено Тогда

Возврат ЦиклР;

Иначе

РезультатВременный.Удалить(РезультатЗапись);

КонецЕсли;

КонецЦикла;

Возврат Неопределено;

КонецФункции

Функция НайтиЦепь(Взаиморасчеты,Результат)

РезультатВременный = Неопределено;

Если Результат = Неопределено Тогда

РезультатВременный = Новый ТаблицаЗначений;

РезультатВременный.Колонки.Добавить("Кто");

РезультатВременный.Колонки.Добавить("Кому");

РезультатВременный.Колонки.Добавить("Сумма");

Иначе

РезультатВременный = Результат.Скопировать();

КонецЕсли;

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Количество = РезультатВременный.Количество();

Если Количество > 0 Тогда

Если РезультатВременный[Количество - 1].Кому <> ВзаиморасчетыЗапись.Кто Тогда

Продолжить;

КонецЕсли;

КонецЕсли;

РезультатЗапись = РезультатВременный.Добавить();

РезультатЗапись.Кто = ВзаиморасчетыЗапись.Кто;

РезультатЗапись.Кому = ВзаиморасчетыЗапись.Кому;

РезультатЗапись.Сумма = ВзаиморасчетыЗапись.Сумма;

Если РезультатВременный.Количество() > 1 Тогда

// цепь из больше чем двух элементов найдена

Возврат РезультатВременный;

КонецЕсли;

Цепь = НайтиЦепь(Взаиморасчеты,РезультатВременный);

Если Цепь <> Неопределено Тогда

Возврат Цепь;

Иначе

РезультатВременный.Удалить(РезультатЗапись);

КонецЕсли;

КонецЦикла;

Возврат Неопределено;

КонецФункции

Автор: Михайлов Алексей

Пояснения к коду:

Процедура КнопкаСформироватьНажатие(Кнопка)

ТабДок = ЭлементыФормы.ПолеТабличногоДокумента;

Долги = Новый ТаблицаЗначений;

Долги.Колонки.Добавить("Кто");

Долги.Колонки.Добавить("Кому");

Долги.Колонки.Добавить("Сумма");

Сальдо = Новый ТаблицаЗначений;

Сальдо.Колонки.Добавить("Кто");

Сальдо.Колонки.Добавить("Сумма");

Взаиморасчеты = Новый ТаблицаЗначений;

Взаиморасчеты.Колонки.Добавить("Кто");

Взаиморасчеты.Колонки.Добавить("Кому");

Взаиморасчеты.Колонки.Добавить("Сумма");

// получаем долги

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ЖурналОперацийДвиженияССубконто.Организации,

| ЖурналОперацийДвиженияССубконто.СубконтоКт1,

| ЖурналОперацийДвиженияССубконто.Сумма

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.ДвиженияССубконто КАК ЖурналОперацийДвиженияССубконто

|ГДЕ

| ЖурналОперацийДвиженияССубконто.СчетКт.Код = ""76""";

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

Пока Выборка.Следующий() Цикл

ДолгиЗапись = Долги.Добавить();

ВзаиморасчетыЗапись = Взаиморасчеты.Добавить();

ДолгиЗапись.Кто = Выборка.Организации;

ДолгиЗапись.Кому = Выборка.СубконтоКт1;

ДолгиЗапись.Сумма = Выборка.Сумма;

ВзаиморасчетыЗапись.Кто = ДолгиЗапись.Кто;

ВзаиморасчетыЗапись.Кому = ДолгиЗапись.Кому;

ВзаиморасчетыЗапись.Сумма = ДолгиЗапись.Сумма;

КонецЦикла;

// получаем сальдо

Для Каждого Долг Из Долги Цикл

СальдоЗаписьКто = Сальдо.Найти(Долг.Кто,"Кто");

СальдоЗаписьКому = Сальдо.Найти(Долг.Кому,"Кто");

Если СальдоЗаписьКто = Неопределено Тогда

СальдоЗаписьКто = Сальдо.Добавить();

СальдоЗаписьКто.Кто = Долг.Кто;

СальдоЗаписьКто.Сумма = 0;

КонецЕсли;

Если СальдоЗаписьКому = Неопределено Тогда

СальдоЗаписьКому = Сальдо.Добавить();

СальдоЗаписьКому.Кто = Долг.Кому;

СальдоЗаписьКому.Сумма = 0;

КонецЕсли;

СальдоЗаписьКто.Сумма = СальдоЗаписьКто.Сумма - Долг.Сумма;

СальдоЗаписьКому.Сумма = СальдоЗаписьКому.Сумма + Долг.Сумма;

КонецЦикла;

// расчитываем взаиморасчеты

// ищем циклы

ЦиклРезультат = НайтиЦикл(Взаиморасчеты,Неопределено);

Пока ЦиклРезультат <> Неопределено Цикл

Минимальный = ЦиклРезультат[0];

Для Каждого Звено Из ЦиклРезультат Цикл

Если Звено.Сумма < Минимальный.Сумма Тогда

Минимальный = Звено;

КонецЕсли;

КонецЦикла;

// уничтожаем цикл

Для Каждого Звено Из ЦиклРезультат Цикл

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Если ВзаиморасчетыЗапись.Кто <> Звено.Кто Тогда

Продолжить;

КонецЕсли;

Если ВзаиморасчетыЗапись.Кому <> Звено.Кому Тогда

Продолжить;

КонецЕсли;

ВзаиморасчетыЗапись.Сумма = ВзаиморасчетыЗапись.Сумма - Минимальный.Сумма;

Если ВзаиморасчетыЗапись.Сумма = 0 Тогда

Взаиморасчеты.Удалить(ВзаиморасчетыЗапись);

КонецЕсли;

КонецЦикла;

КонецЦикла;

ЦиклРезультат = НайтиЦикл(Взаиморасчеты,Неопределено);

КонецЦикла;

// ищем цепи

ЦепьРезультат = НайтиЦепь(Взаиморасчеты,Неопределено);

Пока ЦепьРезультат <> Неопределено Цикл

Запись = Новый Массив(2);

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Для i = 0 по 1 Цикл

Если ВзаиморасчетыЗапись.Кто = ЦепьРезультат[i].Кто И ВзаиморасчетыЗапись.Кому = ЦепьРезультат[i].Кому Тогда

Запись[i] = ВзаиморасчетыЗапись;

КонецЕсли;

КонецЦикла;

КонецЦикла;

Если Запись[0].Сумма > Запись[1].Сумма Тогда

Второму = Запись[0].Сумма - Запись[1].Сумма;

Запись[0].Сумма = Второму;

Запись[1].Кто = Запись[0].Кто;

ИначеЕсли Запись[0].Сумма <= Запись[1].Сумма Тогда

Третьему = Запись[1].Сумма - Запись[0].Сумма;

Запись[0].Кому = Запись[1].Кому;

Запись[1].Сумма = Третьему;

КонецЕсли;

ЦепьРезультат = НайтиЦепь(Взаиморасчеты,Неопределено);

КонецЦикла;

Макет = ПолучитьМакет("Макет");

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьШапкаВзаиморасчеты = Макет.ПолучитьОбласть("ШапкаВзаиморасчеты");

ОбластьСтрокаВзаиморасчеты = Макет.ПолучитьОбласть("СтрокаВзаиморасчеты");

ОбластьШапкаСальдо = Макет.ПолучитьОбласть("ШапкаСальдо");

ОбластьСтрокаСальдо = Макет.ПолучитьОбласть("СтрокаСальдо");

ТабДок.Очистить();

ОбластьЗаголовок.Параметры[0] = "Долги";

ТабДок.Вывести(ОбластьЗаголовок);

ТабДок.Вывести(ОбластьШапкаВзаиморасчеты);

Для Каждого Запись Из Долги Цикл

ОбластьСтрокаВзаиморасчеты.Параметры[0] = Запись.Кто;

ОбластьСтрокаВзаиморасчеты.Параметры[1] = Запись.Кому;

ОбластьСтрокаВзаиморасчеты.Параметры[2] = Запись.Сумма;

ТабДок.Вывести(ОбластьСтрокаВзаиморасчеты);

КонецЦикла;

ОбластьЗаголовок.Параметры[0] = "Сальдо";

ТабДок.Вывести(ОбластьЗаголовок);

ТабДок.Вывести(ОбластьШапкаСальдо);

Для Каждого Запись Из Сальдо Цикл

ОбластьСтрокаСальдо.Параметры[0] = Запись.Кто;

ОбластьСтрокаСальдо.Параметры[1] = Запись.Сумма;

ТабДок.Вывести(ОбластьСтрокаСальдо);

КонецЦикла;

ОбластьЗаголовок.Параметры[0] = "Взаиморасчеты";

ТабДок.Вывести(ОбластьЗаголовок);

ТабДок.Вывести(ОбластьШапкаВзаиморасчеты);

Для Каждого Запись Из Взаиморасчеты Цикл

ОбластьСтрокаВзаиморасчеты.Параметры[0] = Запись.Кто;

ОбластьСтрокаВзаиморасчеты.Параметры[1] = Запись.Кому;

ОбластьСтрокаВзаиморасчеты.Параметры[2] = Запись.Сумма;

ТабДок.Вывести(ОбластьСтрокаВзаиморасчеты);

КонецЦикла;

КонецПроцедуры

Основная процедура, в ней происходит получение долгов, определение сальдо и взаиморасчета, вывод всего результата в поле табличного документа.

ТабДок = ЭлементыФормы.ПолеТабличногоДокумента;

Долги = Новый ТаблицаЗначений;

Долги.Колонки.Добавить("Кто");

Долги.Колонки.Добавить("Кому");

Долги.Колонки.Добавить("Сумма");

Сальдо = Новый ТаблицаЗначений;

Сальдо.Колонки.Добавить("Кто");

Сальдо.Колонки.Добавить("Сумма");

Взаиморасчеты = Новый ТаблицаЗначений;

Взаиморасчеты.Колонки.Добавить("Кто");

Взаиморасчеты.Колонки.Добавить("Кому");

Взаиморасчеты.Колонки.Добавить("Сумма");

В данном блоке создаются переменные для хранения данных. ТабДок – поле табличного документа формы, для вывода результата. Долги – таблица значений с колонками Кто,Кому,Сумма для хранения долгов организаций. Сальдо – таблица значений с колонками Кто,Сумма для хранения сальдо взаиморасчетов. Взаиморасчеты – таблица значений с колонками Кто,Кому,Сумма для хранения взаиморасчетов организаций.

// получаем долги

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

| ЖурналОперацийДвиженияССубконто.Организации,

| ЖурналОперацийДвиженияССубконто.СубконтоКт1,

| ЖурналОперацийДвиженияССубконто.Сумма

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.ДвиженияССубконто КАК ЖурналОперацийДвиженияССубконто

|ГДЕ

| ЖурналОперацийДвиженияССубконто.СчетКт.Код = ""76""";

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

Пока Выборка.Следующий() Цикл

ДолгиЗапись = Долги.Добавить();

ВзаиморасчетыЗапись = Взаиморасчеты.Добавить();

ДолгиЗапись.Кто = Выборка.Организации;

ДолгиЗапись.Кому = Выборка.СубконтоКт1;

ДолгиЗапись.Сумма = Выборка.Сумма;

ВзаиморасчетыЗапись.Кто = ДолгиЗапись.Кто;

ВзаиморасчетыЗапись.Кому = ДолгиЗапись.Кому;

ВзаиморасчетыЗапись.Сумма = ДолгиЗапись.Сумма;

КонецЦикла;

В данном блоке происходит получение долгов организаций из базы данных. Так как долги записываются в журнал операций дважды (Как дебет одной организации в одной проводке и как кредит в другой) нужно выбрать один вариант, в данной реализации выбрано получение организации и её кредитора (СубконтоКт1), а чтобы не было пустых кредиторов (в проводках с дебетом) стоит условие на код счета кредита «76». После получения данных они обходятся и записываются в таблицы долги (для отображения в табличный документ) и взаиморасчеты (для дальнейшей обработки).

// получаем сальдо

Для Каждого Долг Из Долги Цикл

СальдоЗаписьКто = Сальдо.Найти(Долг.Кто,"Кто");

СальдоЗаписьКому = Сальдо.Найти(Долг.Кому,"Кто");

Если СальдоЗаписьКто = Неопределено Тогда

СальдоЗаписьКто = Сальдо.Добавить();

СальдоЗаписьКто.Кто = Долг.Кто;

СальдоЗаписьКто.Сумма = 0;

КонецЕсли;

Если СальдоЗаписьКому = Неопределено Тогда

СальдоЗаписьКому = Сальдо.Добавить();

СальдоЗаписьКому.Кто = Долг.Кому;

СальдоЗаписьКому.Сумма = 0;

КонецЕсли;

СальдоЗаписьКто.Сумма = СальдоЗаписьКто.Сумма - Долг.Сумма;

СальдоЗаписьКому.Сумма = СальдоЗаписьКому.Сумма + Долг.Сумма;

КонецЦикла;

В данном блоке определяется сальдо всех организаций. Для этого из каждой записи с информацией о долге берется кредитор и заемщик, они оба ищутся в таблице Сальдо (если они там есть, сумма будет увеличиваться/уменьшаться), если запись не найдена, то она добавляется. Для существующей записи заемщика сумма долга отнимается от его сальдо, для кредитора сумма долга прибавляется.

// расчитываем взаиморасчеты

// ищем циклы

ЦиклРезультат = НайтиЦикл(Взаиморасчеты,Неопределено);

Пока ЦиклРезультат <> Неопределено Цикл

Минимальный = ЦиклРезультат[0];

Для Каждого Звено Из ЦиклРезультат Цикл

Если Звено.Сумма < Минимальный.Сумма Тогда

Минимальный = Звено;

КонецЕсли;

КонецЦикла;

// уничтожаем цикл

Для Каждого Звено Из ЦиклРезультат Цикл

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Если ВзаиморасчетыЗапись.Кто <> Звено.Кто Тогда

Продолжить;

КонецЕсли;

Если ВзаиморасчетыЗапись.Кому <> Звено.Кому Тогда

Продолжить;

КонецЕсли;

ВзаиморасчетыЗапись.Сумма = ВзаиморасчетыЗапись.Сумма - Минимальный.Сумма;

Если ВзаиморасчетыЗапись.Сумма = 0 Тогда

Взаиморасчеты.Удалить(ВзаиморасчетыЗапись);

КонецЕсли;

КонецЦикла;

КонецЦикла;

ЦиклРезультат = НайтиЦикл(Взаиморасчеты,Неопределено);

КонецЦикла;

// ищем цепи

ЦепьРезультат = НайтиЦепь(Взаиморасчеты,Неопределено);

Пока ЦепьРезультат <> Неопределено Цикл

Запись = Новый Массив(2);

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Для i = 0 по 1 Цикл

Если ВзаиморасчетыЗапись.Кто = ЦепьРезультат[i].Кто И ВзаиморасчетыЗапись.Кому = ЦепьРезультат[i].Кому Тогда

Запись[i] = ВзаиморасчетыЗапись;

КонецЕсли;

КонецЦикла;

КонецЦикла;

Если Запись[0].Сумма > Запись[1].Сумма Тогда

Второму = Запись[0].Сумма - Запись[1].Сумма;

Запись[0].Сумма = Второму;

Запись[1].Кто = Запись[0].Кто;

ИначеЕсли Запись[0].Сумма <= Запись[1].Сумма Тогда

Третьему = Запись[1].Сумма - Запись[0].Сумма;

Запись[0].Кому = Запись[1].Кому;

Запись[1].Сумма = Третьему;

КонецЕсли;

ЦепьРезультат = НайтиЦепь(Взаиморасчеты,Неопределено);

КонецЦикла;

Основная часть процедуры, здесь происходит определение взаиморасчетов при которых сумма долга по предприятию минимальна. Для этого уничтожаются циклы, а после них уничтожаются цепи.

ЦиклРезультат = НайтиЦикл(Взаиморасчеты,Неопределено);

Пока ЦиклРезультат <> Неопределено Цикл

Минимальный = ЦиклРезультат[0];

Для Каждого Звено Из ЦиклРезультат Цикл

Если Звено.Сумма < Минимальный.Сумма Тогда

Минимальный = Звено;

КонецЕсли;

КонецЦикла;

// уничтожаем цикл

Для Каждого Звено Из ЦиклРезультат Цикл

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Если ВзаиморасчетыЗапись.Кто <> Звено.Кто Тогда

Продолжить;

КонецЕсли;

Если ВзаиморасчетыЗапись.Кому <> Звено.Кому Тогда

Продолжить;

КонецЕсли;

ВзаиморасчетыЗапись.Сумма = ВзаиморасчетыЗапись.Сумма - Минимальный.Сумма;

Если ВзаиморасчетыЗапись.Сумма = 0 Тогда

Взаиморасчеты.Удалить(ВзаиморасчетыЗапись);

КонецЕсли;

КонецЦикла;

КонецЦикла;

ЦиклРезультат = НайтиЦикл(Взаиморасчеты,Неопределено);

КонецЦикла;

Данный блок отвечает за уничтожение циклов, с помощью функции НайтиЦикл, с параметром Взаиморасчеты (текущие взаиморасчеты) и вторым параметром Неопределено (этот параметр необходим самой функции, так как она рекурсивна, и его значение появится при поиске цикла). Для каждого найденного цикла находится минимальное звено (долг с минимальной суммой), и после этого цикл разбивается – каждое звено цикла находится в таблице взаиморасчетов и уменьшается на сумму минимального долга, если сумма = 0 (а она будет равна нулю, у найденного звена) то звено удаляется из взаиморасчетов. После этого повторно вызывается функция НайтиЦикл, для следующей итерации цикла (возможно, что не все циклы уничтожены во взаиморасчетах).

ЦепьРезультат = НайтиЦепь(Взаиморасчеты,Неопределено);

Пока ЦепьРезультат <> Неопределено Цикл

Запись = Новый Массив(2);

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Для i = 0 по 1 Цикл

Если ВзаиморасчетыЗапись.Кто = ЦепьРезультат[i].Кто И ВзаиморасчетыЗапись.Кому = ЦепьРезультат[i].Кому Тогда

Запись[i] = ВзаиморасчетыЗапись;

КонецЕсли;

КонецЦикла;

КонецЦикла;

Если Запись[0].Сумма > Запись[1].Сумма Тогда

Второму = Запись[0].Сумма - Запись[1].Сумма;

Запись[0].Сумма = Второму;

Запись[1].Кто = Запись[0].Кто;

ИначеЕсли Запись[0].Сумма <= Запись[1].Сумма Тогда

Третьему = Запись[1].Сумма - Запись[0].Сумма;

Запись[0].Кому = Запись[1].Кому;

Запись[1].Сумма = Третьему;

КонецЕсли;

ЦепьРезультат = НайтиЦепь(Взаиморасчеты,Неопределено);

КонецЦикла;

В данном блоке уничтожаются цепи. Для поиска цепей используется функция НайтиЦепь, с параметром Взаиморасчеты (текущие взаиморасчеты) и вторым параметром Неопределено (этот параметр необходим самой функции, так как она рекурсивна, и его значение появится при поиске цепи). Если цепь найдена, то её звенья находятся в таблице Взаиморасчеты и записываются в массив Запись. Далее определяется как именно разбить цепь. Если первый должен второму больше, чем второй должен третьему, то первый должен второму часть, и отдает долг третьему за второго. Если же первый должен второму меньше либо столько же, сколько второй третьему, то первый отдает долг третьему и ничего больше не должен второму, а второй отдает свой остаток третьему. После уничтожения цепи повторно вызывается функция НайтиЦепь для следующей итерации.

Макет = ПолучитьМакет("Макет");

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьШапкаВзаиморасчеты = Макет.ПолучитьОбласть("ШапкаВзаиморасчеты");

ОбластьСтрокаВзаиморасчеты = Макет.ПолучитьОбласть("СтрокаВзаиморасчеты");

ОбластьШапкаСальдо = Макет.ПолучитьОбласть("ШапкаСальдо");

ОбластьСтрокаСальдо = Макет.ПолучитьОбласть("СтрокаСальдо");

В данном блоке получается макет для вывода табличного документа, и его области.

ТабДок.Очистить();

ОбластьЗаголовок.Параметры[0] = "Долги";

ТабДок.Вывести(ОбластьЗаголовок);

ТабДок.Вывести(ОбластьШапкаВзаиморасчеты);

Для Каждого Запись Из Долги Цикл

ОбластьСтрокаВзаиморасчеты.Параметры[0] = Запись.Кто;

ОбластьСтрокаВзаиморасчеты.Параметры[1] = Запись.Кому;

ОбластьСтрокаВзаиморасчеты.Параметры[2] = Запись.Сумма;

ТабДок.Вывести(ОбластьСтрокаВзаиморасчеты);

КонецЦикла;

ОбластьЗаголовок.Параметры[0] = "Сальдо";

ТабДок.Вывести(ОбластьЗаголовок);

ТабДок.Вывести(ОбластьШапкаСальдо);

Для Каждого Запись Из Сальдо Цикл

ОбластьСтрокаСальдо.Параметры[0] = Запись.Кто;

ОбластьСтрокаСальдо.Параметры[1] = Запись.Сумма;

ТабДок.Вывести(ОбластьСтрокаСальдо);

КонецЦикла;

ОбластьЗаголовок.Параметры[0] = "Взаиморасчеты";

ТабДок.Вывести(ОбластьЗаголовок);

ТабДок.Вывести(ОбластьШапкаВзаиморасчеты);

Для Каждого Запись Из Взаиморасчеты Цикл

ОбластьСтрокаВзаиморасчеты.Параметры[0] = Запись.Кто;

ОбластьСтрокаВзаиморасчеты.Параметры[1] = Запись.Кому;

ОбластьСтрокаВзаиморасчеты.Параметры[2] = Запись.Сумма;

ТабДок.Вывести(ОбластьСтрокаВзаиморасчеты);

КонецЦикла;

В данном блоке происходит очистка табличного документа от старых данных и вывод новых.

Функция НайтиЦикл(Взаиморасчеты,Результат)

РезультатВременный = Неопределено;

Если Результат = Неопределено Тогда

РезультатВременный = Новый ТаблицаЗначений;

РезультатВременный.Колонки.Добавить("Кто");

РезультатВременный.Колонки.Добавить("Кому");

РезультатВременный.Колонки.Добавить("Сумма");

Иначе

РезультатВременный = Результат.Скопировать();

КонецЕсли;

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Количество = РезультатВременный.Количество();

Если Количество > 0 Тогда

Если РезультатВременный[Количество - 1].Кому <> ВзаиморасчетыЗапись.Кто Тогда

Продолжить;

КонецЕсли;

КонецЕсли;

РезультатЗапись = РезультатВременный.Добавить();

РезультатЗапись.Кто = ВзаиморасчетыЗапись.Кто;

РезультатЗапись.Кому = ВзаиморасчетыЗапись.Кому;

РезультатЗапись.Сумма = ВзаиморасчетыЗапись.Сумма;

Если РезультатВременный.Количество() > 1 Тогда

Если РезультатВременный[0].Кто = РезультатЗапись.Кому Тогда

// Цикл найден

Возврат РезультатВременный;

КонецЕсли;

КонецЕсли;

ЦиклР = НайтиЦикл(Взаиморасчеты,РезультатВременный);

Если ЦиклР <> Неопределено Тогда

Возврат ЦиклР;

Иначе

РезультатВременный.Удалить(РезультатЗапись);

КонецЕсли;

КонецЦикла;

Возврат Неопределено;

КонецФункции

Данная функция отвечает за поиск циклов в таблице Взаиморасчеты. Функция рекурсивна (вызывает сама себя). Второй параметр используется для хранения текущего пути обхода графа взаиморасчетов. Если он Неопределен, то таблица для хранения пути создается, и будет заполнена при обходе первым попавшимся долгом. Далее, если в таблице результата уже больше одной записи, то проверяется не совпадает ли первый должник, с тем, кому должен последний (условие цикличности долгов). Если совпадает, значит цикл найден и таблица результата возвращается. Если количество не больше 1, либо ещё не определен цикл, то вызывается НайтиЦикл, с указанием текущего пройденного пути (для дальнейшего поиска), если цикл будет найден – он будет возвращен как результат, если нет – данная ветвь не приводит к решению и удаляется из таблицы результата. Если же были проверены все ветви и нигде не были найдены циклы значит цикла нет и возвращается Неопределено.

Функция НайтиЦепь(Взаиморасчеты,Результат)

РезультатВременный = Неопределено;

Если Результат = Неопределено Тогда

РезультатВременный = Новый ТаблицаЗначений;

РезультатВременный.Колонки.Добавить("Кто");

РезультатВременный.Колонки.Добавить("Кому");

РезультатВременный.Колонки.Добавить("Сумма");

Иначе

РезультатВременный = Результат.Скопировать();

КонецЕсли;

Для Каждого ВзаиморасчетыЗапись Из Взаиморасчеты Цикл

Количество = РезультатВременный.Количество();

Если Количество > 0 Тогда

Если РезультатВременный[Количество - 1].Кому <> ВзаиморасчетыЗапись.Кто Тогда

Продолжить;

КонецЕсли;

КонецЕсли;

РезультатЗапись = РезультатВременный.Добавить();

РезультатЗапись.Кто = ВзаиморасчетыЗапись.Кто;

РезультатЗапись.Кому = ВзаиморасчетыЗапись.Кому;

РезультатЗапись.Сумма = ВзаиморасчетыЗапись.Сумма;

Если РезультатВременный.Количество() > 1 Тогда

// цепь из больше чем двух элементов найдена

Возврат РезультатВременный;

КонецЕсли;

Цепь = НайтиЦепь(Взаиморасчеты,РезультатВременный);

Если Цепь <> Неопределено Тогда

Возврат Цепь;

Иначе

РезультатВременный.Удалить(РезультатЗапись);

КонецЕсли;

КонецЦикла;

Возврат Неопределено;

КонецФункции

Данная функция используется для поиска цепочек долгов, она реализована так же как и поиск циклов, с той лишь разницей, что тут не должно быть замкнутого круга и достаточно чтобы одна запись долга продолжалась другой (первый должен второму, а второй третьему – 2 записи о долге).

Сформированный отчет

