2011 год 2 тур

Требование

Исходная ситуация

Торговая организация заключает договора с поставщиками и покупателями. В соответствии с этими договорами организация получает платежи по договорам с покупателями и производит платежи по договорам с поставщиками.

Задолженность перед поставщиками отражается информационной базе (ИБ) по кредиту счета 60 «Поставщики». Задолженность покупателей отражается в ИБ по дебету счета 62 «Покупатели». Аналитический учет на этих счетах ведется в разрезе субконто двух видов – «Контрагенты» и «Договоры». Видам субконто «Контрагенты» и «Договоры» соответствуют одноименные справочники.

Справочник «Контрагенты» содержит код и наименование контрагента, ему подчинен справочник «Договоры».

Справочник «Договоры» помимо реквизитов «Код» и «Наименование» содержит реквизиты – «Дата оплаты» и «Процент штрафа». Для договоров с покупателями реквизит «Дата оплаты» – это дата планируемого поступления денежных средств по договору. Для договоров с поставщиками реквизит «Дата оплаты» – это дата наступления штрафных санкций (день, начиная с которого ежедневно начисляются суммы штрафов). Реквизит «Процент штрафа» *(р)* указывает процент штрафа, начисляемого за каждый день просрочки платежа.

Начисление суммы штрафа при задержке оплаты поставщику

Сумма штрафа начисляется ежедневно, начиная со дня начала действия штрафных санкций, до дня окончательного погашения долга (включительно), и рассчитывается от суммы фактического остатка долга на начало текущего дня (*Sтек*) по формуле: *Sтек\*р/100.* Начисленная сумма штрафа увеличивает сумму долга, но она не участвует при исчислении суммы штрафа в последующие дни.

Погашение долга

Долг может погашаться частично или полностью. В любом случае при погашении долга сначала полностью погашается накопленная на этот момент сумма начисленного штрафа. Оставшаяся сумма (если осталась) идет на погашение фактического остатка долга, зафиксированного на начало текущего дня. Если сумма долга погашена не полностью, тогда на оставшуюся сумму долга в последующие дни также начисляются штрафы.

Требуется

На основании анализа данных ИБ по предполагаемому поступлению и расходованию денежных средств необходимо сформировать отчет, содержащий оптимальный график предстоящих платежей.

Для этого на момент составления отчета необходимо определить:

- фактическое наличие денежных средств на расчетном счете организации на дату составления отчета (сальдо счета 51);

- суммы задолженностей покупателей по заключенным с ними договорам, как текущие остатки по договорам, учтенные на счете 62 на дату составления отчета. Договора с просроченной оплатой при составлении графика учитываться не должны;

- сумму задолженности перед поставщиками, как текущий остаток на начало дня по договору на счете 60, увеличенный на сумму штрафа.

На основании этих данных в отчете необходимо определить последовательность и суммы предполагаемых поступлений денежных средств и планируемых платежей. При недостаточном наличии денежных средств на расчетном счете допускаются частичные погашения долгов, не приводящие к овердрафту на расчетном счете. График должен быть составлен таким образом, чтобы общая сумма уплачиваемых штрафов была бы минимальной. Форма представления отчета приведена в разделе «*Пример*».

Если общая сумма задолженности перед поставщиками превышает текущую сумму на расчетном счете и общую сумму планируемых поступлений денежных средств, то в таком случае составлять график не требуется, а следует выдать соответствующее сообщение.

Исходные данные

Задание необходимо выполнить в каркасной конфигурации, в которой имеется План счетов, содержащий счета 41, 51, 60, 62, 90, справочники «Контрагенты», «Договоры», «Товары», документы для ввода проводок, отражающих суммы задолженностей по договорам с покупателями и поставщиками, а также поступление денежных средств на расчетный счет.

При помощи документов введено несколько проводок по приходу товаров – в дебет счета 41 и кредит счета 60 и по продаже товаров – в дебет счета 62 и кредит счета 90.

Введено поступление денежных средств на расчетный счет для формирования начального остатка денежных средств. Проводки бухгалтерского учета, не относящиеся к сути задания на программирование, опущены.

Для отладки модуля отчета разрешается добавлять в информационную базу каркасной конфигурации дополнительные справочные данные и проводки.

Жюри проводит тестирование на собственных данных, которые будут отличаться от данных, содержащихся в исходной информационной базе каркасной конфигурации.

Пример

Пусть на начало дня 15 января 2011 года у предприятия на расчетном счете имеется 10 000 руб., при этом в учете зафиксирована следующая картина взаимных обязательств по договорам купли-продажи.

Договоры с покупателями

|  |  |  |  |
| --- | --- | --- | --- |
| Покупатель | Договор | сумма | Дата планируемого поступления денежных средств |
| Пок010 | Д10 | 30 000,00 | 16.01.2011 |
| Пок020 | Д20 | 35 000,00 | 18.01.2011 |
| Пок030 | Д30 | 45 000,00 | 21.01.2011 |
| Пок040 | Д40 | 60 000,00 | 25.01.2011 |
| ИТОГО |  | 170 000,00 |  |

Договоры с поставщиками

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поставщик | Договор | Cумма | Дата оплаты | % штрафа |
| Пост001 | Д1 | 20 000,00 | 15.01.2011 | 2,00 |
| Пост002 | Д2 | 35 000,00 | 15.01.2011 | 3,00 |
| Пост003 | Д3 | 25 000,00 | 17.01.2011 | 2,00 |
| Пост004 | Д4 | 60 000,00 | 19.01.2011 | 5,00 |
| ИТОГО |  | 140 000,00 |  |  |

 Необходимо предложить отчет, содержащий планируемый график платежей. Искомый график будет иметь следующий вид.

Планируемый график платежей по состоянию на 15.01.2011

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Дата | сальдо  счета 51 | Поступление по договорам  с покупателями (сч. 62) | | Выплаты по договорам  с поставщиками (сч. 60) | |
| договор | сумма | договор | Сумма |
| 15.01.2011 | 10 000,00 |  |  | Д2 | 10 000,00 |
| 16.01.2011 | 0,00 | Д10 | 30 000,00 |  |  |
| 17.01.2011 | 30 000,00 |  |  | Д2 | 27 613,00 |
|  | 2 387,00 |  |  | Д4 | 2 387,00 |
|  | 0,00 |  |  |  |  |
| 18.01.2011 | 0,00 | Д20 | 35 000,00 |  |  |
| 19.01.2011 | 35 000,00 |  |  | Д4 | 35 000,00 |
| 21.01.2011 | 0,00 | Д30 | 45 000,00 |  |  |
| 22.01.2011 | 45 000,00 |  |  | Д3 | 15 682,30 |
|  | 29 317,70 |  |  | Д4 | 29 317,70 |
| 25.01.2011 | 0,00 | Д40 | 60 000,00 |  |  |
| 26.01.2011 | 60 000,00 |  |  | Д1 | 24 800,00 |
|  | 35 200,00 |  |  | Д3 | 13 303,12 |
|  | 21 896,88 |  |  |  |  |
| ИТОГО: |  |  | 170 000,00 |  | 158 103,12 |

Сохранение выполненного задания

В результате выполнения задания должен быть разработан отчет, работающий в среде каркасной конфигурации. Данный отчет необходимо сохранить на как внешний отчет.

Решение

Суть данной задачи – найти в графе путь минимальной длинны. Для этого требуется пройти по всем возможным вариантам и выбрать тот, где сумма долгов будет минимальной. Вершины графа – состояния долгов, содержащие – дату, сольдо, ожидаемые получения от покупателей, требуемые выплаты поставщикам. Вес связи двух вершин – сумма затраченная на переход от состояния первой вершины, к состоянию второй (оплачиваемая сумма долга). Рассмотрим пример переходов по состояниям, приводящий к решению как в примере требований.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Состояние | Дата | Сальдо | Покупатели | | | | Поставщики | | | | | Оплачено |
| Договор | Сумма | | Дата | Договор | Сумма | Дата | Процент | Долг |
| S0 | 15.01.11 | 10000 | Д10 | 30000 | | 16.01.11 | Д1 | 20000 | 15.01.11 | 2 | 400 | 0 |
| Д20 | 35000 | | 18.01.11 | Д2 | 35000 | 15.01.11 | 3 | 1050 |
| Д30 | 45000 | | 21.01.11 | Д3 | 25000 | 17.01.11 | 2 | 0 |
| Д40 | 60000 | | 25.01.11 | Д4 | 60000 | 19.01.11 | 5 | 0 |
| S02 | 15.01.11 | 0 | Д10 | 30000 | | 16.01.11 | Д1 | 20000 | 15.01.11 | 2 | 400 | 10000 |
| Д20 | 35000 | | 18.01.11 | Д2 | 26050 | 15.01.11 | 3 | 0 |
| Д30 | 45000 | | 21.01.11 | Д3 | 25000 | 17.01.11 | 2 | 0 |
| Д40 | 60000 | | 25.01.11 | Д4 | 60000 | 19.01.11 | 5 | 0 |
| S022 | 17.01.11 | 2387 | Д20 | 35000 | | 18.01.11 | Д1 | 20000 | 15.01.11 | 2 | 1200 | 37613 |
| Д30 | 45000 | | 21.01.11 | Д3 | 25000 | 17.01.11 | 2 | 500 |
| Д40 | 60000 | | 25.01.11 | Д4 | 60000 | 19.01.11 | 5 | 0 |
| S0224 | 17.01.11 | 0 | Д20 | 35000 | | 18.01.11 | Д1 | 20000 | 15.01.11 | 2 | 1200 | 40000 |
| Д30 | 45000 | | 21.01.11 | Д3 | 25000 | 17.01.11 | 2 | 500 |
| Д40 | 60000 | | 25.01.11 | Д4 | 57613 | 19.01.11 | 5 | 0 |
| S02244 | 19.01.11 | 0 | Д30 | 45000 | | 21.01.11 | Д1 | 20000 | 15.01.11 | 2 | 2000 | 75000 |
| Д40 | 60000 | | 25.01.11 | Д3 | 25000 | 17.01.11 | 2 | 1500 |
| Д4 | 25493,65 | 19.01.11 | 5 | 0 |
| S022444 | 22.01.11 | 15682,30 | Д40 | | 60000 | 25.01.11 | Д1 | 20000 | 15.01.11 | 2 | 2800 | 105592,38 |
| Д3 | 25000 | 17.01.11 | 2 | 3000 |
| S0224443 | 22.01.11 | 0 | Д40 | | 60000 | 25.01.11 | Д1 | 20000 | 15.01.11 | 2 | 2800 | 120000 |
| Д3 | 12317,7 | 17.01.11 | 2 | 0 |
| S02244431 | 26.01.11 | 35200 |  | | | | Д3 | 12317,7 | 17.01.11 | 2 | 985,42 | 144800 |
| S022444313 | 26.01.11 | 21896,88 |  | | | |  | | | | | 158103,12 |

Это один из путей, начиная с состояния S0 существует несколько вариантов развития событий: оплата по договору Д1, либо по договору Д2, либо по Д3, либо по Д4. После оплаты по договору снова появляется несколько вариантов для оплаты. Для верного решения задачи требуется рассмотреть все варианты последовательности оплаты, и выбрать минимальный по сумме выплат. Для этого подойдет рекурсивная реализация обхода графа, рассмотрим алгоритм:

Функция ОбходГрафа(Граф)

Если вершин идущих из корня нет

Вернуть конец графа

Конец если

Для всех вершин (A) идущих из корня

Результат = ОбходГрафа(Граф начинающийся в вершине A)

Если Результат > Минимального результата Тогда

Минимальный результат = Результат

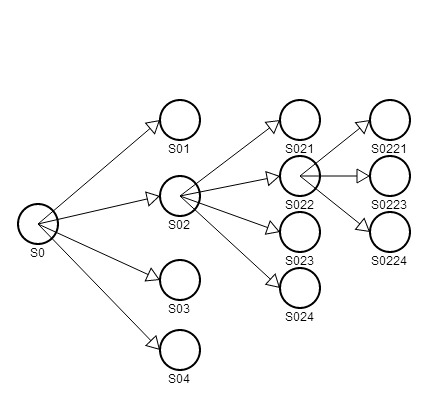
Конец если

Конец цикла

Вернуть минимальный результат

Конец функции

Функция вызывает сама себя, чтобы обойти подграфы. Рассмотрим граф:



S0 – корень графа, из него можно перейти в S01,S02,S03,S04. Функция вызыванная для графа с корнем S0, вызовет саму себя, для графов с корнями S01,S02,S03,S04, внутри опять вызовет саму себя для каждого из графов, с другими графами, и так далее. Так функция обойдет весь граф и, дойдя до вершины не имеющей дочерних вершин, остановится и соберет результат выбирая на наименьший из всех получившихся. Так будет найден путь, приводящий к требуемому результату.

Реализация

Процедура КнопкаСформироватьНажатие(Кнопка)

ТабДок = ЭлементыФормы.ПолеТабличногоДокумента;

Макет = ПолучитьМакет("Макет");

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьЗапись = Макет.ПолучитьОбласть("Запись");

ОбластьИтог = Макет.ПолучитьОбласть("Итог");

// Получаем данные из базы и формируем начальные структуры

СтартДанные = Новый Структура;

СтартДанные.Вставить("Сумма",0);

// Дата

СтартДата = Дата;

// Сальдо

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналОперацийОстатки.СуммаОстаток

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&Дата, Счет.Код = ""51"", , ) КАК ЖурналОперацийОстатки";

Запрос.УстановитьПараметр("Дата", Дата);

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

Результат.Следующий();

СтартСальдо = Результат.СуммаОстаток;

// Поставщики

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналОперацийОстатки.Субконто1,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто1),

| ЖурналОперацийОстатки.Субконто2 КАК Субконто2,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто2),

| ЖурналОперацийОстатки.СуммаОстатокКт,

| Договоры.ДатаОплаты КАК ДатаОплаты,

| Договоры.Процент

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&Дата, Счет.Код = ""60"", , ) КАК ЖурналОперацийОстатки

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Договоры КАК Договоры

| ПО ЖурналОперацийОстатки.Субконто2 = Договоры.Ссылка

|

|УПОРЯДОЧИТЬ ПО

| ДатаОплаты,

| Субконто2";

Запрос.УстановитьПараметр("Дата", Дата);

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

Поставщики = Новый Массив;

Пока Результат.Следующий() Цикл

Запись = Новый Структура;

Запись.Вставить("Дата",Результат.ДатаОплаты);

Запись.Вставить("Договор",Результат.Субконто2);

Запись.Вставить("Сумма",Результат.СуммаОстатокКт);

Запись.Вставить("Штраф",Результат.Процент);

Поставщики.Добавить(Запись);

КонецЦикла;

СтартДанные.Вставить("Поставщики",Поставщики);

// Покупатели

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналОперацийОстатки.Субконто1,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто1),

| ЖурналОперацийОстатки.Субконто2 КАК Субконто2,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто2),

| ЖурналОперацийОстатки.СуммаОстатокДт КАК СуммаОстатокДт,

| Договоры.ДатаОплаты КАК ДатаОплаты

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&Дата, Счет.Код = ""62"", , ) КАК ЖурналОперацийОстатки

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Договоры КАК Договоры

| ПО ЖурналОперацийОстатки.Субконто2 = Договоры.Ссылка

|ГДЕ

| Договоры.ДатаОплаты >= &Дата

|

|УПОРЯДОЧИТЬ ПО

| ДатаОплаты,

| Субконто2";

Запрос.УстановитьПараметр("Дата", Дата);

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

Покупатели = Новый Массив;

Пока Результат.Следующий() Цикл

Запись = Новый Структура;

Запись.Вставить("Дата",Результат.ДатаОплаты);

Запись.Вставить("Договор",Результат.Субконто2);

Запись.Вставить("Сумма",Результат.СуммаОстатокДт);

Покупатели.Добавить(Запись);

КонецЦикла;

СтартДанные.Вставить("Покупатели",Покупатели);

// Планируем расчеты

Результат = СпланироватьРасчеты(СтартДата,СтартСальдо,СтартДанные);

ТабДок.Очистить();

Если Результат = Неопределено Тогда

Сообщить("Не хватает средств для оплаты долгов");

Иначе

ОбластьЗаголовок.Параметры[0] = "Планируемый график платежей по состоянию на " + Дата;

ТабДок.Вывести(ОбластьЗаголовок);

ПОЛУЧЕНИЕ\_ДЕНЕГ = 0;

ВЫПЛАТА\_ДЕНЕГ = 1;

ПокупателиИтог = 0;

ПоставщикиИтог = 0;

Для Каждого Вершина Из Результат.Путь Цикл

ОбластьЗапись.Параметры[0] = Вершина.Дата;

ОбластьЗапись.Параметры[1] = Вершина.Сальдо;

Если Вершина.Действие = ПОЛУЧЕНИЕ\_ДЕНЕГ Тогда

ОбластьЗапись.Параметры[2] = Вершина.Покупатель.Договор;

ОбластьЗапись.Параметры[3] = Вершина.Покупатель.Сумма;

ОбластьЗапись.Параметры[4] = "";

ОбластьЗапись.Параметры[5] = "";

ПокупателиИтог = ПокупателиИтог + Вершина.Покупатель.Сумма;

Иначе

ОбластьЗапись.Параметры[2] = "";

ОбластьЗапись.Параметры[3] = "";

ОбластьЗапись.Параметры[4] = Вершина.Поставщик.Договор;

ОбластьЗапись.Параметры[5] = Вершина.Поставщик.Оплачено;

ПоставщикиИтог = ПоставщикиИтог + Вершина.Поставщик.Оплачено;

КонецЕсли;

ТабДок.Вывести(ОбластьЗапись);

КонецЦикла;

ОбластьИтог.Параметры[0] = ПокупателиИтог;

ОбластьИтог.Параметры[1] = ПоставщикиИтог;

ТабДок.Вывести(ОбластьИтог);

КонецЕсли;

КонецПроцедуры

Функция СпланироватьРасчеты(Дата,Сальдо,Данные)

ПОЛУЧЕНИЕ\_ДЕНЕГ = 0;

ВЫПЛАТА\_ДЕНЕГ = 1;

Результат = Новый Структура;

Путь = Новый Массив;

Результат.Вставить("Сумма",0);

Результат.Вставить("Путь",Путь);

// проверяем не завершены ли получения и выплаты

Если Данные.Покупатели.Количество() = 0 И Данные.Поставщики.Количество() = 0 Тогда

Возврат Результат;

КонецЕсли;

РДата = Дата;

РСальдо = Сальдо;

// проверяем есть ли деньги на счете для оплаты поставщикам

Если Сальдо = 0 Тогда

Если Данные.Покупатели.Количество() = 0 Тогда

Возврат Неопределено;

КонецЕсли;

// денег нет, выполняем ближайшую получку денег от покупателя и планируем дальше, начиная со следующего после получения денег дня

Покупатель = Данные.Покупатели[0];

РДата = НачалоДня(Покупатель.Дата) + (60\*60\*24);

РСальдо = Покупатель.Сумма;

// убираем оформленную получку денег

Данные.Покупатели.Удалить(0);

// добавляем в путь состояние с получением оплаты

НовоеСостояние = Новый Структура;

НовоеСостояние.Вставить("Дата",Покупатель.Дата);

НовоеСостояние.Вставить("Сальдо",Сальдо);

НовоеСостояние.Вставить("Действие",ПОЛУЧЕНИЕ\_ДЕНЕГ);

НовоеСостояние.Вставить("Покупатель",Покупатель);

Результат.Путь.Добавить(НовоеСостояние);

КонецЕсли;

ЛучшийРезультат = Неопределено;

ЛучшийНачало = Неопределено;

// пробуем разные выплаты поставщикам (обходим их по очереди, выбираем результат с минимальной суммой выплат)

Для Каждого Поставщик Из Данные.Поставщики Цикл

НовоеСальдо = 0;

Начало = Неопределено;

НовыеДанные = Новый Структура;

НовыеДанные.Вставить("Поставщики",Новый Массив);

НовыеДанные.Вставить("Покупатели",Новый Массив);

Для Каждого Запись Из Данные.Покупатели Цикл

НоваяЗапись = Новый Структура;

НоваяЗапись.Вставить("Дата",Запись.Дата);

НоваяЗапись.Вставить("Договор",Запись.Договор);

НоваяЗапись.Вставить("Сумма",Запись.Сумма);

НовыеДанные.Покупатели.Добавить(НоваяЗапись);

КонецЦикла;

Для Каждого Запись Из Данные.Поставщики Цикл

НоваяЗапись = Неопределено;

Если Поставщик = Запись Тогда

Штраф = 0;

Дни = 0;

Если РДата >= Запись.Дата Тогда

Дни = ((КонецДня(РДата)+1)-НачалоДня(Запись.Дата))/(24\*60\*60);

Штраф = (Запись.Сумма \* (Запись.Штраф/100)) \* Дни;

КонецЕсли;

Если (Поставщик.Сумма + Штраф) > РСальдо Тогда

НоваяЗапись = Новый Структура;

Если Штраф = 0 Тогда

НоваяЗапись.Вставить("Дата",Запись.Дата);

Иначе

НоваяЗапись.Вставить("Дата",(КонецДня(РДата)+1));

КонецЕсли;

НоваяЗапись.Вставить("Договор",Запись.Договор);

НоваяЗапись.Вставить("Сумма",Запись.Сумма + Штраф - РСальдо);

НоваяЗапись.Вставить("Штраф",Запись.Штраф);

НовоеСальдо = 0;

Начало = Новый Структура;

Начало.Вставить("Дата",Запись.Дата);

Начало.Вставить("Договор",Запись.Договор);

Начало.Вставить("Оплачено",РСальдо);

Иначе

НовоеСальдо = РСальдо - (Запись.Сумма + Штраф);

Начало = Новый Структура;

Начало.Вставить("Дата",Запись.Дата);

Начало.Вставить("Договор",Запись.Договор);

Начало.Вставить("Оплачено",Запись.Сумма + Штраф);

КонецЕсли;

Иначе

НоваяЗапись = Новый Структура;

НоваяЗапись.Вставить("Дата",Запись.Дата);

НоваяЗапись.Вставить("Договор",Запись.Договор);

НоваяЗапись.Вставить("Сумма",Запись.Сумма);

НоваяЗапись.Вставить("Штраф",Запись.Штраф);

КонецЕсли;

Если НоваяЗапись <> Неопределено Тогда

НовыеДанные.Поставщики.Добавить(НоваяЗапись);

КонецЕсли;

КонецЦикла;

РезультатВетки = СпланироватьРасчеты(РДата,НовоеСальдо,НовыеДанные);

Если ЛучшийРезультат <> Неопределено И РезультатВетки <> Неопределено Тогда

Если (ЛучшийНачало.Оплачено + ЛучшийРезультат.Сумма) > (Начало.Оплачено + РезультатВетки.Сумма) Тогда

ЛучшийРезультат = РезультатВетки;

ЛучшийНачало = Начало;

КонецЕсли;

Иначе

ЛучшийРезультат = РезультатВетки;

ЛучшийНачало = Начало;

КонецЕсли;

КонецЦикла;

Если ЛучшийРезультат = Неопределено Тогда

Возврат Неопределено;

КонецЕсли;

// найден лучший результат, составляем итоговый результат

Результат.Сумма = ЛучшийНачало.Оплачено + ЛучшийРезультат.Сумма;

НоваяВершина = Новый Структура;

НоваяВершина.Вставить("Дата",РДата);

НоваяВершина.Вставить("Сальдо",РСальдо);

НоваяВершина.Вставить("Действие",ВЫПЛАТА\_ДЕНЕГ);

НоваяВершина.Вставить("Поставщик",ЛучшийНачало);

Результат.Путь.Добавить(НоваяВершина);

Для Каждого Вершина Из ЛучшийРезультат.Путь Цикл

Результат.Путь.Добавить(Вершина);

КонецЦикла;

Возврат Результат;

КонецФункции

Процедура ПриОткрытии()

Дата = Дата(2011,01,15);

КонецПроцедуры

Автор: Михайлов Алексей

Пояснения к коду:

Процедура КнопкаСформироватьНажатие(Кнопка)

ТабДок = ЭлементыФормы.ПолеТабличногоДокумента;

Макет = ПолучитьМакет("Макет");

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьЗапись = Макет.ПолучитьОбласть("Запись");

ОбластьИтог = Макет.ПолучитьОбласть("Итог");

// Получаем данные из базы и формируем начальные структуры

СтартДанные = Новый Структура;

СтартДанные.Вставить("Сумма",0);

// Дата

СтартДата = Дата;

// Сальдо

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналОперацийОстатки.СуммаОстаток

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&Дата, Счет.Код = ""51"", , ) КАК ЖурналОперацийОстатки";

Запрос.УстановитьПараметр("Дата", Дата);

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

Результат.Следующий();

СтартСальдо = Результат.СуммаОстаток;

// Поставщики

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналОперацийОстатки.Субконто1,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто1),

| ЖурналОперацийОстатки.Субконто2 КАК Субконто2,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто2),

| ЖурналОперацийОстатки.СуммаОстатокКт,

| Договоры.ДатаОплаты КАК ДатаОплаты,

| Договоры.Процент

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&Дата, Счет.Код = ""60"", , ) КАК ЖурналОперацийОстатки

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Договоры КАК Договоры

| ПО ЖурналОперацийОстатки.Субконто2 = Договоры.Ссылка

|

|УПОРЯДОЧИТЬ ПО

| ДатаОплаты,

| Субконто2";

Запрос.УстановитьПараметр("Дата", Дата);

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

Поставщики = Новый Массив;

Пока Результат.Следующий() Цикл

Запись = Новый Структура;

Запись.Вставить("Дата",Результат.ДатаОплаты);

Запись.Вставить("Договор",Результат.Субконто2);

Запись.Вставить("Сумма",Результат.СуммаОстатокКт);

Запись.Вставить("Штраф",Результат.Процент);

Поставщики.Добавить(Запись);

КонецЦикла;

СтартДанные.Вставить("Поставщики",Поставщики);

// Покупатели

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналОперацийОстатки.Субконто1,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто1),

| ЖурналОперацийОстатки.Субконто2 КАК Субконто2,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто2),

| ЖурналОперацийОстатки.СуммаОстатокДт КАК СуммаОстатокДт,

| Договоры.ДатаОплаты КАК ДатаОплаты

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&Дата, Счет.Код = ""62"", , ) КАК ЖурналОперацийОстатки

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Договоры КАК Договоры

| ПО ЖурналОперацийОстатки.Субконто2 = Договоры.Ссылка

|ГДЕ

| Договоры.ДатаОплаты >= &Дата

|

|УПОРЯДОЧИТЬ ПО

| ДатаОплаты,

| Субконто2";

Запрос.УстановитьПараметр("Дата", Дата);

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

Покупатели = Новый Массив;

Пока Результат.Следующий() Цикл

Запись = Новый Структура;

Запись.Вставить("Дата",Результат.ДатаОплаты);

Запись.Вставить("Договор",Результат.Субконто2);

Запись.Вставить("Сумма",Результат.СуммаОстатокДт);

Покупатели.Добавить(Запись);

КонецЦикла;

СтартДанные.Вставить("Покупатели",Покупатели);

// Планируем расчеты

Результат = СпланироватьРасчеты(СтартДата,СтартСальдо,СтартДанные);

ТабДок.Очистить();

Если Результат = Неопределено Тогда

Сообщить("Не хватает средств для оплаты долгов");

Иначе

ОбластьЗаголовок.Параметры[0] = "Планируемый график платежей по состоянию на " + Дата;

ТабДок.Вывести(ОбластьЗаголовок);

ПОЛУЧЕНИЕ\_ДЕНЕГ = 0;

ВЫПЛАТА\_ДЕНЕГ = 1;

ПокупателиИтог = 0;

ПоставщикиИтог = 0;

Для Каждого Вершина Из Результат.Путь Цикл

ОбластьЗапись.Параметры[0] = Вершина.Дата;

ОбластьЗапись.Параметры[1] = Вершина.Сальдо;

Если Вершина.Действие = ПОЛУЧЕНИЕ\_ДЕНЕГ Тогда

ОбластьЗапись.Параметры[2] = Вершина.Покупатель.Договор;

ОбластьЗапись.Параметры[3] = Вершина.Покупатель.Сумма;

ОбластьЗапись.Параметры[4] = "";

ОбластьЗапись.Параметры[5] = "";

ПокупателиИтог = ПокупателиИтог + Вершина.Покупатель.Сумма;

Иначе

ОбластьЗапись.Параметры[2] = "";

ОбластьЗапись.Параметры[3] = "";

ОбластьЗапись.Параметры[4] = Вершина.Поставщик.Договор;

ОбластьЗапись.Параметры[5] = Вершина.Поставщик.Оплачено;

ПоставщикиИтог = ПоставщикиИтог + Вершина.Поставщик.Оплачено;

КонецЕсли;

ТабДок.Вывести(ОбластьЗапись);

КонецЦикла;

ОбластьИтог.Параметры[0] = ПокупателиИтог;

ОбластьИтог.Параметры[1] = ПоставщикиИтог;

ТабДок.Вывести(ОбластьИтог);

КонецЕсли;

КонецПроцедуры

Данная процедура выполняется при нажатии на кнопку сформировать отчет и выполняет загрузку данных из базы данных, заполнение графа, вызов функции обхода графа и вывод результата в поле табличного документа.

ТабДок = ЭлементыФормы.ПолеТабличногоДокумента;

Макет = ПолучитьМакет("Макет");

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьЗапись = Макет.ПолучитьОбласть("Запись");

ОбластьИтог = Макет.ПолучитьОбласть("Итог");

В данном блоке идет получение элемента формы ТабличныйДокумент и получение областей макета, по которым будет строиться результат.

// Получаем данные из базы и формируем начальные структуры

СтартДанные = Новый Структура;

СтартДанные.Вставить("Сумма",0);

// Дата

СтартДата = Дата;

В данном блоке создается структура начального состояния СтартДанные и переменная даты начала СтартДата.

// Сальдо

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналОперацийОстатки.СуммаОстаток

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&Дата, Счет.Код = ""51"", , ) КАК ЖурналОперацийОстатки";

Запрос.УстановитьПараметр("Дата", Дата);

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

Результат.Следующий();

СтартСальдо = Результат.СуммаОстаток;

В данном блоке получается Сальдо на дату начала и записывается в переменную СтартСальдо.

// Поставщики

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналОперацийОстатки.Субконто1,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто1),

| ЖурналОперацийОстатки.Субконто2 КАК Субконто2,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто2),

| ЖурналОперацийОстатки.СуммаОстатокКт,

| Договоры.ДатаОплаты КАК ДатаОплаты,

| Договоры.Процент

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&Дата, Счет.Код = ""60"", , ) КАК ЖурналОперацийОстатки

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Договоры КАК Договоры

| ПО ЖурналОперацийОстатки.Субконто2 = Договоры.Ссылка

|

|УПОРЯДОЧИТЬ ПО

| ДатаОплаты,

| Субконто2";

Запрос.УстановитьПараметр("Дата", Дата);

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

Поставщики = Новый Массив;

Пока Результат.Следующий() Цикл

Запись = Новый Структура;

Запись.Вставить("Дата",Результат.ДатаОплаты);

Запись.Вставить("Договор",Результат.Субконто2);

Запись.Вставить("Сумма",Результат.СуммаОстатокКт);

Запись.Вставить("Штраф",Результат.Процент);

Поставщики.Добавить(Запись);

КонецЦикла;

СтартДанные.Вставить("Поставщики",Поставщики);

В данном блоке получается список поставщиков, на дату начала. Полученные из базы данных записи добавляются в массив Поставщики, и массив сохраняется в начальном состоянии в поле Поставщики.

// Покупатели

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналОперацийОстатки.Субконто1,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто1),

| ЖурналОперацийОстатки.Субконто2 КАК Субконто2,

| ПРЕДСТАВЛЕНИЕ(ЖурналОперацийОстатки.Субконто2),

| ЖурналОперацийОстатки.СуммаОстатокДт КАК СуммаОстатокДт,

| Договоры.ДатаОплаты КАК ДатаОплаты

|ИЗ

| РегистрБухгалтерии.ЖурналОпераций.Остатки(&Дата, Счет.Код = ""62"", , ) КАК ЖурналОперацийОстатки

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Договоры КАК Договоры

| ПО ЖурналОперацийОстатки.Субконто2 = Договоры.Ссылка

|ГДЕ

| Договоры.ДатаОплаты >= &Дата

|

|УПОРЯДОЧИТЬ ПО

| ДатаОплаты,

| Субконто2";

Запрос.УстановитьПараметр("Дата", Дата);

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

Покупатели = Новый Массив;

Пока Результат.Следующий() Цикл

Запись = Новый Структура;

Запись.Вставить("Дата",Результат.ДатаОплаты);

Запись.Вставить("Договор",Результат.Субконто2);

Запись.Вставить("Сумма",Результат.СуммаОстатокДт);

Покупатели.Добавить(Запись);

КонецЦикла;

СтартДанные.Вставить("Покупатели",Покупатели);

В данном блоке получаются покупатели на дату начала. Данные записываются в массив, а массив добавляется в начальное состояние в поле Покупатели.

// Планируем расчеты

Результат = СпланироватьРасчеты(СтартДата,СтартСальдо,СтартДанные);

ТабДок.Очистить();

В данном блоке происходит вызов рекурсивной функции СпланироватьРасчеты, которая определит лучший путь. Как параметры указаны дата начала (СтартДата), начальное сальдо (СтартСальдо) и начальное состояние (СтартДанные), на основе которого будут строиться дальнейшие состояния. После завершения планирования идет отчистка табличного документа от старых данных.

Если Результат = Неопределено Тогда

Сообщить("Не хватает средств для оплаты долгов");

Иначе

ОбластьЗаголовок.Параметры[0] = "Планируемый график платежей по состоянию на " + Дата;

ТабДок.Вывести(ОбластьЗаголовок);

ПОЛУЧЕНИЕ\_ДЕНЕГ = 0;

ВЫПЛАТА\_ДЕНЕГ = 1;

ПокупателиИтог = 0;

ПоставщикиИтог = 0;

Для Каждого Вершина Из Результат.Путь Цикл

ОбластьЗапись.Параметры[0] = Вершина.Дата;

ОбластьЗапись.Параметры[1] = Вершина.Сальдо;

Если Вершина.Действие = ПОЛУЧЕНИЕ\_ДЕНЕГ Тогда

ОбластьЗапись.Параметры[2] = Вершина.Покупатель.Договор;

ОбластьЗапись.Параметры[3] = Вершина.Покупатель.Сумма;

ОбластьЗапись.Параметры[4] = "";

ОбластьЗапись.Параметры[5] = "";

ПокупателиИтог = ПокупателиИтог + Вершина.Покупатель.Сумма;

Иначе

ОбластьЗапись.Параметры[2] = "";

ОбластьЗапись.Параметры[3] = "";

ОбластьЗапись.Параметры[4] = Вершина.Поставщик.Договор;

ОбластьЗапись.Параметры[5] = Вершина.Поставщик.Оплачено;

ПоставщикиИтог = ПоставщикиИтог + Вершина.Поставщик.Оплачено;

КонецЕсли;

ТабДок.Вывести(ОбластьЗапись);

КонецЦикла;

ОбластьИтог.Параметры[0] = ПокупателиИтог;

ОбластьИтог.Параметры[1] = ПоставщикиИтог;

ТабДок.Вывести(ОбластьИтог);

КонецЕсли;

В данном блоке выводится найденный результат. Если результат неопределен, то выводится сообщение о нехватке средств для решения. Если результат найден, то выводится заголовок, и вершины (состояния) пути. В вершинах присутствует поле Действие, которое помогает определить, что именно делается на данном шаге (получение денег от покупателя, либо выплата денег поставщику). В зависимости от действия заполняются нужные поля области записи. После вывода всех вершин, выводится итог по покупателям и по поставщикам.

Функция СпланироватьРасчеты(Дата,Сальдо,Данные)

ПОЛУЧЕНИЕ\_ДЕНЕГ = 0;

ВЫПЛАТА\_ДЕНЕГ = 1;

Результат = Новый Структура;

Путь = Новый Массив;

Результат.Вставить("Сумма",0);

Результат.Вставить("Путь",Путь);

// проверяем не завершены ли получения и выплаты

Если Данные.Покупатели.Количество() = 0 И Данные.Поставщики.Количество() = 0 Тогда

Возврат Результат;

КонецЕсли;

РДата = Дата;

РСальдо = Сальдо;

// проверяем есть ли деньги на счете для оплаты поставщикам

Если Сальдо = 0 Тогда

Если Данные.Покупатели.Количество() = 0 Тогда

Возврат Неопределено;

КонецЕсли;

// денег нет, выполняем ближайшую получку денег от покупателя и планируем дальше, начиная со следующего после получения денег дня

Покупатель = Данные.Покупатели[0];

РДата = НачалоДня(Покупатель.Дата) + (60\*60\*24);

РСальдо = Покупатель.Сумма;

// убираем оформленную получку денег

Данные.Покупатели.Удалить(0);

// добавляем в путь состояние с получением оплаты

НовоеСостояние = Новый Структура;

НовоеСостояние.Вставить("Дата",Покупатель.Дата);

НовоеСостояние.Вставить("Сальдо",Сальдо);

НовоеСостояние.Вставить("Действие",ПОЛУЧЕНИЕ\_ДЕНЕГ);

НовоеСостояние.Вставить("Покупатель",Покупатель);

Результат.Путь.Добавить(НовоеСостояние);

КонецЕсли;

ЛучшийРезультат = Неопределено;

ЛучшийНачало = Неопределено;

// пробуем разные выплаты поставщикам (обходим их по очереди, выбираем результат с минимальной суммой выплат)

Для Каждого Поставщик Из Данные.Поставщики Цикл

НовоеСальдо = 0;

Начало = Неопределено;

НовыеДанные = Новый Структура;

НовыеДанные.Вставить("Поставщики",Новый Массив);

НовыеДанные.Вставить("Покупатели",Новый Массив);

Для Каждого Запись Из Данные.Покупатели Цикл

НоваяЗапись = Новый Структура;

НоваяЗапись.Вставить("Дата",Запись.Дата);

НоваяЗапись.Вставить("Договор",Запись.Договор);

НоваяЗапись.Вставить("Сумма",Запись.Сумма);

НовыеДанные.Покупатели.Добавить(НоваяЗапись);

КонецЦикла;

Для Каждого Запись Из Данные.Поставщики Цикл

НоваяЗапись = Неопределено;

Если Поставщик = Запись Тогда

Штраф = 0;

Дни = 0;

Если РДата >= Запись.Дата Тогда

Дни = ((КонецДня(РДата)+1)-НачалоДня(Запись.Дата))/(24\*60\*60);

Штраф = (Запись.Сумма \* (Запись.Штраф/100)) \* Дни;

КонецЕсли;

Если (Поставщик.Сумма + Штраф) > РСальдо Тогда

НоваяЗапись = Новый Структура;

Если Штраф = 0 Тогда

НоваяЗапись.Вставить("Дата",Запись.Дата);

Иначе

НоваяЗапись.Вставить("Дата",(КонецДня(РДата)+1));

КонецЕсли;

НоваяЗапись.Вставить("Договор",Запись.Договор);

НоваяЗапись.Вставить("Сумма",Запись.Сумма + Штраф - РСальдо);

НоваяЗапись.Вставить("Штраф",Запись.Штраф);

НовоеСальдо = 0;

Начало = Новый Структура;

Начало.Вставить("Дата",Запись.Дата);

Начало.Вставить("Договор",Запись.Договор);

Начало.Вставить("Оплачено",РСальдо);

Иначе

НовоеСальдо = РСальдо - (Запись.Сумма + Штраф);

Начало = Новый Структура;

Начало.Вставить("Дата",Запись.Дата);

Начало.Вставить("Договор",Запись.Договор);

Начало.Вставить("Оплачено",Запись.Сумма + Штраф);

КонецЕсли;

Иначе

НоваяЗапись = Новый Структура;

НоваяЗапись.Вставить("Дата",Запись.Дата);

НоваяЗапись.Вставить("Договор",Запись.Договор);

НоваяЗапись.Вставить("Сумма",Запись.Сумма);

НоваяЗапись.Вставить("Штраф",Запись.Штраф);

КонецЕсли;

Если НоваяЗапись <> Неопределено Тогда

НовыеДанные.Поставщики.Добавить(НоваяЗапись);

КонецЕсли;

КонецЦикла;

РезультатВетки = СпланироватьРасчеты(РДата,НовоеСальдо,НовыеДанные);

Если ЛучшийРезультат <> Неопределено И РезультатВетки <> Неопределено Тогда

Если (ЛучшийНачало.Оплачено + ЛучшийРезультат.Сумма) > (Начало.Оплачено + РезультатВетки.Сумма) Тогда

ЛучшийРезультат = РезультатВетки;

ЛучшийНачало = Начало;

КонецЕсли;

Иначе

ЛучшийРезультат = РезультатВетки;

ЛучшийНачало = Начало;

КонецЕсли;

КонецЦикла;

Если ЛучшийРезультат = Неопределено Тогда

Возврат Неопределено;

КонецЕсли;

// найден лучший результат, составляем итоговый результат

Результат.Сумма = ЛучшийНачало.Оплачено + ЛучшийРезультат.Сумма;

НоваяВершина = Новый Структура;

НоваяВершина.Вставить("Дата",РДата);

НоваяВершина.Вставить("Сальдо",РСальдо);

НоваяВершина.Вставить("Действие",ВЫПЛАТА\_ДЕНЕГ);

НоваяВершина.Вставить("Поставщик",ЛучшийНачало);

Результат.Путь.Добавить(НоваяВершина);

Для Каждого Вершина Из ЛучшийРезультат.Путь Цикл

Результат.Путь.Добавить(Вершина);

КонецЦикла;

Возврат Результат;

КонецФункции

Главная функция, производит поиск минимального пути в графе.

ПОЛУЧЕНИЕ\_ДЕНЕГ = 0;

ВЫПЛАТА\_ДЕНЕГ = 1;

В данном блоке определяются константы, для добавления вершин в результат (чтобы различать состояния с получением денег и состояния с выплатой денег).

Результат = Новый Структура;

Путь = Новый Массив;

Результат.Вставить("Сумма",0);

Результат.Вставить("Путь",Путь);

В данном блоке создаются переменные для результата выполнения функции, и хранения пути для результата.

// проверяем не завершены ли получения и выплаты

Если Данные.Покупатели.Количество() = 0 И Данные.Поставщики.Количество() = 0 Тогда

Возврат Результат;

КонецЕсли;

В данном блоке проверяется не завершен ли обход графа (закончились поступления и выплаты).

РДата = Дата;

РСальдо = Сальдо;

// проверяем есть ли деньги на счете для оплаты поставщикам

Если Сальдо = 0 Тогда

Если Данные.Покупатели.Количество() = 0 Тогда

Возврат Неопределено;

КонецЕсли;

// денег нет, выполняем ближайшую получку денег от покупателя и планируем дальше, начиная со следующего после получения денег дня

Покупатель = Данные.Покупатели[0];

РДата = НачалоДня(Покупатель.Дата) + (60\*60\*24);

РСальдо = Покупатель.Сумма;

// убираем оформленную получку денег

Данные.Покупатели.Удалить(0);

// добавляем в путь состояние с получением оплаты

НовоеСостояние = Новый Структура;

НовоеСостояние.Вставить("Дата",Покупатель.Дата);

НовоеСостояние.Вставить("Сальдо",Сальдо);

НовоеСостояние.Вставить("Действие",ПОЛУЧЕНИЕ\_ДЕНЕГ);

НовоеСостояние.Вставить("Покупатель",Покупатель);

Результат.Путь.Добавить(НовоеСостояние);

КонецЕсли;

В данном блоке определяются Дата и Сальдо с которым обходить исходящие из корня вершины. Если Сальдо равно нулю, то требуется получить деньги от покупателя. Если покупателей не осталось, то такой путь считается тупиковым и возвращается Неопределено. Иначе – запись покупателя убирается из данных, дата берется – следующий день после даты получения денег, а сальдо – количество полученых денег. Так же добавляется состояние отвечающее за получение денег от покупателя.

ЛучшийРезультат = Неопределено;

ЛучшийНачало = Неопределено;

// пробуем разные выплаты поставщикам (обходим их по очереди, выбираем результат с минимальной суммой выплат)

Для Каждого Поставщик Из Данные.Поставщики Цикл

НовоеСальдо = 0;

Начало = Неопределено;

НовыеДанные = Новый Структура;

НовыеДанные.Вставить("Поставщики",Новый Массив);

НовыеДанные.Вставить("Покупатели",Новый Массив);

Для Каждого Запись Из Данные.Покупатели Цикл

НоваяЗапись = Новый Структура;

НоваяЗапись.Вставить("Дата",Запись.Дата);

НоваяЗапись.Вставить("Договор",Запись.Договор);

НоваяЗапись.Вставить("Сумма",Запись.Сумма);

НовыеДанные.Покупатели.Добавить(НоваяЗапись);

КонецЦикла;

Для Каждого Запись Из Данные.Поставщики Цикл

НоваяЗапись = Неопределено;

Если Поставщик = Запись Тогда

Штраф = 0;

Дни = 0;

Если РДата >= Запись.Дата Тогда

Дни = ((КонецДня(РДата)+1)-НачалоДня(Запись.Дата))/(24\*60\*60);

Штраф = (Запись.Сумма \* (Запись.Штраф/100)) \* Дни;

КонецЕсли;

Если (Поставщик.Сумма + Штраф) > РСальдо Тогда

НоваяЗапись = Новый Структура;

Если Штраф = 0 Тогда

НоваяЗапись.Вставить("Дата",Запись.Дата);

Иначе

НоваяЗапись.Вставить("Дата",(КонецДня(РДата)+1));

КонецЕсли;

НоваяЗапись.Вставить("Договор",Запись.Договор);

НоваяЗапись.Вставить("Сумма",Запись.Сумма + Штраф - РСальдо);

НоваяЗапись.Вставить("Штраф",Запись.Штраф);

НовоеСальдо = 0;

Начало = Новый Структура;

Начало.Вставить("Дата",Запись.Дата);

Начало.Вставить("Договор",Запись.Договор);

Начало.Вставить("Оплачено",РСальдо);

Иначе

НовоеСальдо = РСальдо - (Запись.Сумма + Штраф);

Начало = Новый Структура;

Начало.Вставить("Дата",Запись.Дата);

Начало.Вставить("Договор",Запись.Договор);

Начало.Вставить("Оплачено",Запись.Сумма + Штраф);

КонецЕсли;

Иначе

НоваяЗапись = Новый Структура;

НоваяЗапись.Вставить("Дата",Запись.Дата);

НоваяЗапись.Вставить("Договор",Запись.Договор);

НоваяЗапись.Вставить("Сумма",Запись.Сумма);

НоваяЗапись.Вставить("Штраф",Запись.Штраф);

КонецЕсли;

Если НоваяЗапись <> Неопределено Тогда

НовыеДанные.Поставщики.Добавить(НоваяЗапись);

КонецЕсли;

КонецЦикла;

РезультатВетки = СпланироватьРасчеты(РДата,НовоеСальдо,НовыеДанные);

Если ЛучшийРезультат <> Неопределено И РезультатВетки <> Неопределено Тогда

Если (ЛучшийНачало.Оплачено + ЛучшийРезультат.Сумма) > (Начало.Оплачено + РезультатВетки.Сумма) Тогда

ЛучшийРезультат = РезультатВетки;

ЛучшийНачало = Начало;

КонецЕсли;

Иначе

ЛучшийРезультат = РезультатВетки;

ЛучшийНачало = Начало;

КонецЕсли;

КонецЦикла;

В данном блоке обходятся все возможные варианты оплаты из текущего состояния. Начальные данные копируются в НовыеДанные, чтобы все вытекающие состояния работали с начальными данными, а не испорченными предыдущим.

Далее составляем новое состояние, которое будет корнем графа, который нужно будет обойти рекурсивным вызовом. Все записи просто копируются, а та, по которой проводится оплата либо удаляется (если сумма полностью оплачена), либо сумма меняется на новую (сумма + штраф – сумма оплаты) и дата начала для расчета штрафа меняется на день после оплаты.

Далее вычисляется результат обхода графа, с новым корнем, если результат лучше сохраненного – сохраняем как лучший его.

Если ЛучшийРезультат = Неопределено Тогда

Возврат Неопределено;

КонецЕсли;

// найден лучший результат, составляем итоговый результат

Результат.Сумма = ЛучшийНачало.Оплачено + ЛучшийРезультат.Сумма;

НоваяВершина = Новый Структура;

НоваяВершина.Вставить("Дата",РДата);

НоваяВершина.Вставить("Сальдо",РСальдо);

НоваяВершина.Вставить("Действие",ВЫПЛАТА\_ДЕНЕГ);

НоваяВершина.Вставить("Поставщик",ЛучшийНачало);

Результат.Путь.Добавить(НоваяВершина);

Для Каждого Вершина Из ЛучшийРезультат.Путь Цикл

Результат.Путь.Добавить(Вершина);

КонецЦикла;

Возврат Результат;

Если лучший результат не найден – считаем ветку тупиковой и возвращаем Неопределено. Иначе – добавляем в путь результата вершину, которая приводит к лучшему результату, и все вершины пути лучшего результата.

Процедура ПриОткрытии()

Дата = Дата(2011,01,15);

КонецПроцедуры

Данный блок необходим для задания даты по умолчанию – 15 января 2011.

Сформированный отчет

