



*Конфигурирование в системе
"1С:Предприятие 8".
Решение расчетных задач
Версия 8.3*

*Методические материалы
для слушателей сертифицированного курса*

Март, 2014

ПРАВО ТИРАЖИРОВАНИЯ И РАСПРОСТРАНЕНИЯ
МЕТОДИЧЕСКИХ МАТЕРИАЛОВ
ПРИНАДЛЕЖИТ ФИРМЕ "1С"

Получив настоящие материалы для обучения, Вы тем самым даете согласие
не допускать их копирования без письменного
разрешения фирмы "1С"

© ООО "1С", 2014 г.

Фирма "1С", Москва, 123056, а/я 64
Отдел продаж: ул. Селезневская, д.21,
телефон: (495)737-92-57,
факс: (495) 681-44-07,
e-mail: 1c@1c.ru,
URL: <http://www.1c.ru>

03-14

Автор материалов: ООО "1С-Учебный центр №3",
(495) 253-58-38, uc3@1c.ru, www.1c-uc3.ru

Предложения по совершенствованию методических материалов
просьба направлять в группу организации обучения фирмы "1С"
e-mail: cso@1c.ru

Содержание

1. ПРЕДВАРИТЕЛЬНЫЕ ЗАМЕЧАНИЯ	6
2. СОГЛАШЕНИЯ О ТЕРМИНАХ, ОБОЗНАЧЕНИЯХ И ДОПОЛНИТЕЛЬНЫЕ СОГЛАШЕНИЯ	7
3. ОБЗОР ОСОБЕННОСТЕЙ РАСЧЕТА ЗАРПЛАТЫ.....	8
3.1. Первая особенность: строгая периодичность расчета зарплаты. Понятие Период регистрации.....	8
3.2. Кажущаяся простота расчетных формул	9
3.3. Базовая зависимость	9
3.3.1. Понятие базовой зависимости	9
3.3.2. Многоуровневая базовая зависимость	10
3.4. Начисления и удержания	11
3.5. Понятие "Период действия". Зависимость по периоду действия	12
3.6. Понятие "Фактический период действия"	13
3.7. Краткие выводы из обзора особенностей расчета зарплаты	14
3.8. Состав расчетных механизмов платформы "1С:Предприятие" 8.....	14
4. ПОСТАНОВКА УЧЕБНОЙ ЗАДАЧИ	16
4.1. Техническое задание, первая очередь.....	16
4.2. Исходные данные и необходимые данные	17
4.3. Окончательная формулировка первой очереди технического задания	17
4.4. Создание учебной конфигурации	18
5. РАСЧЕТНЫЕ ОБЪЕКТЫ КОНФИГУРАЦИИ.....	19
5.1. Планы видов расчета	19
5.2. Создание плана видов расчета "ОсновныеНачисления"	19
5.2.1. Создание объекта конфигурации "ПланВидовРасчета"	19
5.2.2. Создание и настройка реквизитов плана видов расчета	21
5.2.3. Настройка расчетных свойств плана видов расчета	23
5.2.4. Формы плана видов расчета.....	27
5.2.5. Занесение видов расчета в план видов расчета "ОсновныеНачисления"	30
6. РЕГИСТРЫ РАСЧЕТА.....	34
6.1. Назначение и логическое устройство регистра расчета.....	34
6.2. Структура полей таблицы регистра расчета	36
6.2.1. Стандартные поля таблицы регистра расчета	36
6.2.2. Прикладные поля таблицы регистра расчета	37
6.3. Создание и настройка регистра расчета "ОсновныеНачисления"	38
6.3.1. Добавление нового объекта конфигурации "Регистр расчета"	38
6.3.2. Настройка основных свойств регистра расчета "ОсновныеНачисления"	39
6.3.3. Создание прикладных полей регистра расчета "ОсновныеНачисления"	42
6.3.4. Назначение регистратора регистру расчета	47
7. ШАГИ (ЭТАПЫ) СЕАНСА РАСЧЕТА ЗАРПЛАТЫ	48
7.1. Первый шаг сеанса расчета.....	48
7.2. Второй шаг сеанса расчета	49
7.3. Третий шаг сеанса расчета	49
7.4. Многократность выполнения второго и третьего шага сеанса расчета	49

8. РЕАЛИЗАЦИЯ РАСЧЕТНЫХ АЛГОРИТМОВ СЕАНСА РАСЧЕТА РЕГИСТРА "ОСНОВНЫЕ НАЧИСЛЕНИЯ"	51
8.1. Конфигурирование структуры документа "Расчет Зарплаты"	51
8.1.1. Создание табличной части "Основные Начисления"	51
8.1.2. Создание элемента формы для отображения табличной части	52
8.2. Краткое изложение принципов программной работы с регистром расчета	53
8.3. Процедура "Заполнить Набор Записей"	54
8.4. Программный код первого шага сеанса расчета	54
8.5. Особенность метода "Записать" набора записей регистра расчета	56
8.6. Создание экземпляра документа "Расчет зарплаты" и ввод в документ вида расчета "Оклад"	57
8.7. Проведение документа – запись исходных данных в регистр расчета	58
8.8. Программный код второго шага сеанса расчета регистра "Основные Начисления"	59
8.8.1. Виртуальная таблица "Данные Графика" регистра расчета. Расчет начислений со способом расчета "По Окладу"	59
8.8.2. Виртуальная таблица "База" регистра расчета. Расчет начислений со способом расчета "Процентом"	83
9. НАЛИЧИЕ У РЕГИСТРА РАСЧЕТА НЕСКОЛЬКИХ БАЗОВЫХ РЕГИСТРОВ	100
9.1. Техническое задание, вторая очередь	100
9.2. Создание второй пары расчетных объектов	101
9.2.1. План видов расчета "Премии"	101
9.2.2. Заполнение плана видов расчета "Премии"	103
9.2.3. Регистр расчета "Премии"	103
9.2.4. Доработка структуры документа "Начисление Зарплаты"	107
9.2.5. Код первого шага сеанса расчета	108
9.2.6. Ввод и запись вида расчета "Персональная премия"	110
9.2.7. Реализация расчетного кода способа расчета "Фиксированно"	111
9.2.8. Расчет начислений "Персональная премия"	114
9.2.9. Ввод и запись вида расчета "Месячная премия"	114
9.2.10. Запрос получения необходимых данных суммы базы для расчета премий	115
9.2.11. Пригодность составленного запроса для всего перечня премий	119
9.2.12. Код расчета для категории "Зависимое первого уровня"	119
9.2.13. Расчет начисления "Месячная премия"	121
9.2.14. Зависимые второго уровня. Начисление "Поощрительная надбавка"	121
9.2.15. "Частичное попадание" периода действия базового вида расчета в базовый период	123
10. ПОЛУЧЕНИЕ СУММЫ БАЗЫ В РАЗРЕЗЕ БАЗОВЫХ ВИДОВ РАСЧЕТА	125
10.1. Параметр "Разрезы" виртуальной таблицы "База"	126
10.2. Разработка запроса	126
11. РАЗРАБОТКА ОТЧЕТА ПО НАЧИСЛЕНИЯМ	130
12. КОРРЕКТИРОВКА РЕЗУЛЬТАТОВ РАСЧЕТА ПРОШЕДШИХ ПЕРИОДОВ	136
12.1. Невозможность исправления результатов расчета за прошедший период	136

12.2. Случай, когда должны рассчитываться начисления за прошедший период	137
12.3. Сторно	139
12.4. Средства платформы для расчета сторно-суммы. Метод "ПолучитьДополнение"	140
12.5. Цикл обработки таблицы сторно-записей	143
12.6. Процедура обработки строки таблицы сторно-записей	144
13. ПЕРЕРАСЧЕТЫ	156
13.1. Зависимость по перерасчету	156
13.2. Зависимость по перерасчету шире, чем зависимости по периоду действия и по базе.....	156
13.3. Стандартная табличная часть "ВедущиеВидыРасчета"	157
13.4. Заполнение табличной части "Ведущие виды расчета"	157
13.5. Объект конфигурации "Перерасчет"	160
13.5.1. Структура таблицы информационной базы объекта "Перерасчет"	161
13.5.2. Создание объекта "ПерерасчетПремий"	161
13.5.3. Создание измерения перерасчета	162
13.5.4. Установление соответствия измерений перерасчета измерениям регистров расчета.....	162
13.5.5. Создание объекта "ПерерасчетОсновныхНачислений"	165
13.6. Объектная модель работы с перерасчетом	166
13.7. Создание перерасчетной ситуации и первая проблема программной реализации перерасчета.....	168
13.8. Вторая проблема программной реализации перерасчета	170
13.9. Запрос для получения необходимых данных записей, подлежащих перерасчету	171
13.9.1. "Задающий" запрос – первый запрос пакета	173
13.9.2. Окончательный запрос	175
13.9.3. План дальнейших действий	176

1. Предварительные замечания

Механизмы сложных периодических расчетов платформы "1С:Предприятие 8", или просто расчетные механизмы, разработаны фирмой "1С" в основном для решения задач расчета заработной платы, хотя они с успехом могут применяться для решения задач любой предметной области, где требуется применение подобного функционала.

Следует обратить внимание, что в процессе изучения расчетных механизмов приходится оперировать понятиями, которые не встречаются ни в обыденной жизни, ни в других предметных областях. Это делает изучение расчетных механизмов несколько более напряженным и трудоемким, чем изучение других механизмов платформы "1С:Предприятие 8".

Поэтому можно надеяться, что к изучению механизмов расчета учащийся приступает, уже имея некоторый опыт разработки на платформе 1С 8 и обладая некоторыми знаниями встроенного языка платформы и языка запросов.

А именно, крайне желательно:

- иметь представление об основных прикладных объектах платформы: "Справочник", "Документ", "Перечисление", "ТабличнаяЧасть";
- иметь представление о программных объектах типа <Массив>, <СписокЗначений>, <Запрос>;
- иметь опыт программирования проведения регистров других классов (например, регистров накопления);
- владеть языком запросов и уметь работать в Конструкторе запроса.

2. Соглашения о терминах, обозначениях и дополнительные соглашения

Названия диалоговых кнопок, закладок диалоговых панелей, названия пунктов меню, имена других объектов, будут даваться в двойных кавычках, например: "ОК", "Услуги", "Предприятие", "Контрагент" и т.д.

Типы данных будут даваться в угловых скобках: <дата>, <СправочникСсылка.Организации>.

Обращение к пункту меню будет даваться в последовательном перечислении родительских пунктов через значок стрелки вправо "/", например, "Конфигурация/ Поддержка/ Обновить конфигурацию".

Практические задания делятся на Практикумы (задания, выполняемые самостоятельно), упражнения (выполняемые вместе с преподавателем). Практические задания оформляются так:

Практикум № _____

Ссылки на процедуры и функции в основном тексте будут даваться в двойных кавычках "" без скобок.

Важные дополнения к материалу даются:

Важно! _____

Текст, заключенный в рамку (врезки, таблицы и т.п.) может иметь продолжение на следующей странице.

Будьте внимательны!

3. Обзор особенностей расчета зарплаты

Предметная область расчета заработной платы имеет ряд специфических особенностей, которые отсутствуют в других предметных областях. Эти особенности и явились причиной того, что изучение расчетных механизмов является таким непривычным и не очень легким.

3.1. Первая особенность: строгая периодичность расчета зарплаты. Понятие Период регистрации

Мы как-то не очень задумываемся над вопросом строгой периодичности расчета заработной платы, пока нам ее не начинают задерживать. Тогда мы вспоминаем, что зарплата должна выплачиваться строго ежемесячно...

А вот проверяющие органы (налоговая инспекция и другие) об этом не забывают никогда. Для целей проверки они используют бухгалтерскую отчетность предприятия, в которой должны быть сведения о начислении заработной платы за каждый отчетный период (проще говоря, за каждый месяц).

Поэтому любой результат расчета заработной платы необходимо снабжать сведениями о том, в каком периоде (месяце) этот результат предполагается провести по бухучету, по кредиту счета 70 ("Расчеты с персоналом по оплате труда").

Важно!

Период, месяц, за который результаты месячного расчета зарплаты предполагается отразить в бухучете, называется **периодом регистрации**.

Иногда для него встречается название "учетный период".

Пример отражения результатов расчета зарплаты в бухучете с указанием периода регистрации:

Период регистрации	Счет дебета	Счет кредита	Сумма
февраль 2014	(20, 23, 25, 26 и т.д.)	(70)	•••
март 2014	(20, 23, 25, 26 и т.д.)	(70)	•••
апрель 2014	(20, 23, 25, 26 и т.д.)	(70)	•••

Т.е. период регистрации – это "бухгалтерский" период проводок по расчету зарплаты, на первый взгляд не связанный непосредственно с расчетом зарплаты. Но на самом деле связь все же имеется. И мы ее обязательно рассмотрим в нашем курсе.

3.2. Кажущаяся простота расчетных формул

Когда идет речь о расчете зарплаты, то вначале задача представляется достаточно простой. Математические формулы расчета зарплаты крайне просты и для их понимания вполне достаточно знаний арифметики в рамках начальной школы.

Например, сумма месячного заработка сотрудника, работающего повременно, обычно рассчитывается по такой формуле:

$$\text{[Сумма начисления]} = \text{[Ставка оклада]} * \text{[Число отработанных дней]} / \text{[Число рабочих дней в месяце]}$$

А если сотрудник помимо оклада получает еще и премию, то расчет его заработка можно выразить тремя формулами:

- 1) $\text{[Сумма оклада]} = \text{[Ставка оклада]} * \text{[Число отработанных дней]} / \text{[Число рабочих дней в месяце]}$
- 2) $\text{[Сумма премии]} = \text{[Сумма оклада]} * \text{[Процент премии]} / 100$
- 3) $\text{[Сумма начисления]} = \text{[Сумма оклада]} + \text{[Сумма премии]}$

И, как ни усложняй систему начислений, расчетные формулы никогда не будут содержать никакого аппарата высшей математики, что делает их понимание крайне простым.

Отметим, что оклад и премия имеют разные расчетные формулы, принципиально отличающиеся друг от друга. В общем случае таких составляющих (слагаемых) суммы начисления может быть несколько, и они могут иметь разные расчетные формулы.

Важно!

В зарплатных решениях фирмы "1С" такие составляющие суммы начисления, которые рассчитываются сотрудникам при начислении зарплаты и каждый из которых имеет собственную расчетную формулу (алгоритм), называются **видами расчета**. В нашем простом примере "Оклад" и "Премия" – это виды расчета.

Однако стоит приглядеться повнимательнее к этим простым формулам, как появляются некоторые вопросы, на которые необходимо получить ответ.

3.3. Базовая зависимость

3.3.1. Понятие базовой зависимости

Присмотримся ко второй формуле нашего примера расчета оклада и премии сотрудника. В общем случае сумма, которая умножается на процент премии, может состоять не только из оклада, но в нее могут входить результаты и других видов расчета: например, сдельного заработка (если сотрудник в течение месяца успел поработать также и сдельно), различных доплат (за вредность, за квалификацию и т.п.).

Чаще всего так и бывает. Эти виды расчета, результаты которых должны быть получены предварительно, по отношению к данному виду расчета (в нашем примере – премии) называются **базовыми видами расчета**. У нас в расчете

премии участвует результат только одного вида расчета – "Оклад", значит, "Оклад" в нашем простом примере является базовым видом расчета для вида расчета "Премия".

Важно!

Эта сумма, складывающаяся из результатов других видов расчета, по отношению к данному виду расчета ("Премия") называется суммой базовых начислений или более просто: расчетной базой, базовой суммой, суммой базы или совсем просто – базой.

А сам алгоритм расчета (расчетная формула) называется **расчетом по базе**.

Расчет по базе – очень широко распространенный расчетный алгоритм. Так рассчитываются не только премия, но и многие другие виды расчета, например, различные доплаты за вредные условия работы, надбавки отдельным сотрудникам или категориям сотрудников и т.п. А состав базовых видов расчета для каждого такого вида расчета определяется отдельно и может быть различным. Состав базовых видов расчета определяется руководством предприятия (расчетной политикой предприятия) и в принципе ничем не регламентируется.

3.3.2. Многоуровневая базовая зависимость

И здесь появляется первая трудность, на которую при рассмотрении простоты формулы расчета по базе мы внимания не обратили.

Дело в том, что, поскольку состав базовых видов расчета для каждого отдельного вида расчета определяется произвольно, то в этот состав могут также войти виды расчета, рассчитываемые по такому же алгоритму – т.е. по базе. А в состав базы этих видов расчета могут входить такие, которые тоже рассчитываются по базе... и т.д., и т.д. ... В принципе эта "цепочка" зависимости одних видов расчета от других по базе может быть сколь угодно длинной, т.е. базовая зависимость может быть многоуровневой. А визуально эту зависимость можно представить в виде иерархической структуры (дерева).

Например, пусть на некотором предприятии применяется вид расчета "Директорская надбавка", который рассчитывается по базе. В состав его базы входят некоторые виды расчета, которые тоже рассчитываются по базе и имеют базовые виды расчета, и т.д. Тогда базовую зависимость можно изобразить в виде примерно такого дерева:

Директорская надбавка
Оплата авторских договоров
Оплата по договорам подряда
Районный коэффициент
Доплата за вечерние часы
Квартальная премия
Оклад по дням
Оплата сверхурочных часов
Месячная премия
Доплата за работу в праздники и выходные
Оклад по дням
Оплата по производственным нарядам
Оплата праздничных и выходных дней
Оплата сверхурочных часов

Северная надбавка
Доплата за вечерние часы
Доплата за ночные часы
Доплата за работу в праздники и выходные

Т.е, в состав базы вида расчета "Директорская надбавка" непосредственно входят виды расчета: "Оплата авторских договоров", "Оплата по договорам подряда", "Районный коэффициент" и "Северная надбавка". А в состав базы вида расчета "Районный коэффициент" в свою очередь входят: "Доплата за вечерние часы", "Квартальная премия", "Месячная премия", "Оплата праздничных и выходных дней" и "Оплата сверхурочных часов". Среди них, например, "Квартальная премия" тоже рассчитывается по базе и тоже имеет свой состав базовых видов расчета, и т.д.

Из всего этого следует один важный вывод. **Прежде чем приступить к расчету, необходимо иметь результаты расчета базы всех видов расчета, которые мы хотим рассчитать.**

Таким образом, для выполнения этого расчета необходимо знать:

- состав базовых видов расчета;
- порядок, в котором осуществляется их расчет.

Очевидно, что расчет должен выполняться по очереди "снизу вверх" по уровням дерева базовой зависимости, т.е. сначала рассчитываются виды расчета, которые находятся на самом нижнем уровне дерева, потом те, которые являются для них "родительскими" и т.д., до самого высшего уровня.

Эту же задачу непременно должен решать и программист, разрабатывающий зарплатную систему. Т.е., нужно обеспечить:

- хранение состава базовых видов расчета;
- необходимый порядок расчета: "снизу вверх по дереву базовой зависимости".

3.4. Начисления и удержания

Когда производится расчет зарплаты, то результаты одних видов расчета складываются с общей суммой, причитающейся сотруднику (оклады, премии, надбавки разные, командировка, больничный лист и т.п.). Такие виды расчета называются начислениями.

А результаты других видов расчета вычитаются из общей суммы (налоги, штрафы, алименты, выплата по банковским кредитам и т.п.). Такие виды расчета называются удержаниями.

Важно!

Итак, начисления – это такие виды расчета, результаты которых складываются с общей суммой рассчитанной зарплаты.

А удержания – это такие виды расчета, результаты которых вычитаются из общей суммой начисленной зарплаты.

3.5. Понятие "Период действия". Зависимость по периоду действия

Состав зависимостей между видами расчета не исчерпывается базовой зависимостью. Существует еще одна очень важная зависимость видов расчета от других видов расчета. Она проявляется на тех видах расчета, которые по расчетной формуле зависят от отработанного времени, например, "Оклад".

Дело в том, что сотрудник не может одновременно находиться в двух или более удаленных друг от друга местах и одновременно выполнять там некоторые виды работ. Так, если сотрудник находится в командировке, то он физически не может работать на своем рабочем месте в офисе и, следовательно, не должен получать деньги за это. Таким образом, если сотрудник в командировке, то вид расчета Оклад ему не должен рассчитываться за период командировки, а должен рассчитываться вид расчета "Командировка". Также, если сотрудник заболел и взял больничный, то ему за период болезни должен рассчитываться только вид расчета "Больничный лист", а вид расчета "Оклад" опять же не должен рассчитываться за этот период.

Если изобразить длительность предполагаемого действия нескольких событий, которые происходили с сотрудником на протяжении месяца, скажем, ноября, то может получиться примерно такая картина:

Ноябрь	1	3	10	11	...	16	17	...	22	23	26	27	30
Оклад	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Командировка						■	■	■	■						
Больничный												■	■	■	
Больничный							■								

В начале месяца не предполагалось, что сотрудник поедет в командировку. Предполагалось, что он будет работать на своем рабочем месте целый месяц, и, следовательно, вид расчета "Оклад" будет действовать на протяжении всего месяца. Опять же, когда сотруднику выписывалась командировка, никто не мог предположить, что он в командировке заболеет, не сможет работать и ему придется взять больничный.

В обоих случаях для видов расчета "Оклад" и "Командировка" изначально мыслится некий предполагаемый период, известный заранее, и который может расцениваться как некий "плановый" период, в который вид расчета будет "действовать". Для "Оклад" это с 1 по 30 число, для "Командировка" – с 11 по 16 число.

Важно!

Этот "плановый", заранее известный период называется периодом действия вида расчета.

Используя неформальный язык, можно сказать, что период действия – это тот месяц (или другой период, например, для командировки), за работу в котором

сотруднику производится расчет. Если мы слышим слова: "мой оклад за ноябрь", то это значит, что речь идет о сумме, начисленной сотруднику за работу в ноябре – т.е. период действия этого начисления равен 1.11 – 30.11, и т.п.

Здесь назрело одно важное рассуждение.

Ранее мы ввели понятие "Период регистрации" – период, в котором результат начисления предполагается провести по бухучету. Теперь у нас появилось понятие "Период действия" – период, за который сотруднику начислена сумма по данному виду расчета. Возникает вопрос: период действия обязан совпадать с периодом регистрации или нет?

Конечно же, не обязан. Можно, к примеру, рассчитать сотруднику оклад за работу в мае. Это – период действия. Но провести эту сумму по бухучету можно предложить бухгалтеру в июне или другом месяце. Это – период регистрации.

Важно!

"Период действия" и "Период регистрации" имеют разное смысловое содержание и совсем не обязаны совпадать, хотя в частном случае такое вполне возможно.

Но, как мы знаем, могут происходить некоторые события, о которых заранее не могло быть известно и которые способны внести коррективы в "рабочий график" сотрудника. Сотрудник отправился в командировку или заболел – и вот, он уже не работает на своем рабочем месте, следовательно, "действие" вида расчета "Оклад" прерывается на период, равный периоду действия вида расчета "Командировка" (с 11 по 16 число). А если сотрудник заболел в командировке, то действие вида расчета "Командировка" прерывается на время болезни – т.е. на период, равный периоду действия вида расчета "Больничный" (с 13 по 14 число). А если сотрудник, прибыв из командировки, снова заболел, то действие вида расчета "Оклад" снова прерывается – уже на время второй болезни, т.е. на период с 23 по 26 число.

Поскольку результат этих видов расчета зависит от отработанного по ним времени, то и сумма результата изменяется под действием других видов расчета, которые прерывают их действие, или, как принято говорить, вытесняют эти виды расчета. Таким образом, виды расчета, зависящие по расчетной формуле от времени, тоже могут зависеть друг от друга. Эта зависимость называется зависимостью по периоду действия, а виды расчета, прерывающие действие других видов расчета, называются по отношению к ним вытесняющими видами расчета.

3.6. Понятие "Фактический период действия"

Таким образом, можно констатировать, что период действия некоего вида расчета (плановый, предполагаемый период) при наличии вытесняющих видов расчета претерпевает изменения: он дробится на несмежные интервалы, в промежутках между которыми этот вид расчета "не действует".

Важно!

Совокупность несмежных интервалов, на которые распадается изначальный непрерывный период действия вида расчета, называется фактическим периодом действия этого вида расчета.

Арифметически это означает, что фактически отработанное время для этого вида расчета становится меньше, следовательно, и сумма начисления должна быть меньше – ведь она зависит от отработанного времени. Например, для оклада:

$$[\text{Сумма начисления}] = [\text{Ставка оклада}] * [\text{Число отработанных дней}] / [\text{Число рабочих дней в месяце}]$$

А именно: **[Число отработанных дней]** будет равно сумме рабочих дней по всем интервалам фактического периода действия. А **[Число рабочих дней в месяце]** будет равно сумме рабочих дней в периоде действия.

Состав вытесняющих видов расчета, как и состав базовых видов расчета, задается для каждого вида расчета индивидуально и также определяется расчетной политикой руководства конкретного предприятия.

3.7. Краткие выводы из обзора особенностей расчета зарплаты

Таким образом, математически крайне простые расчетные формулы в реализации оказываются не так уж и просты, как нам показалось на первый взгляд.

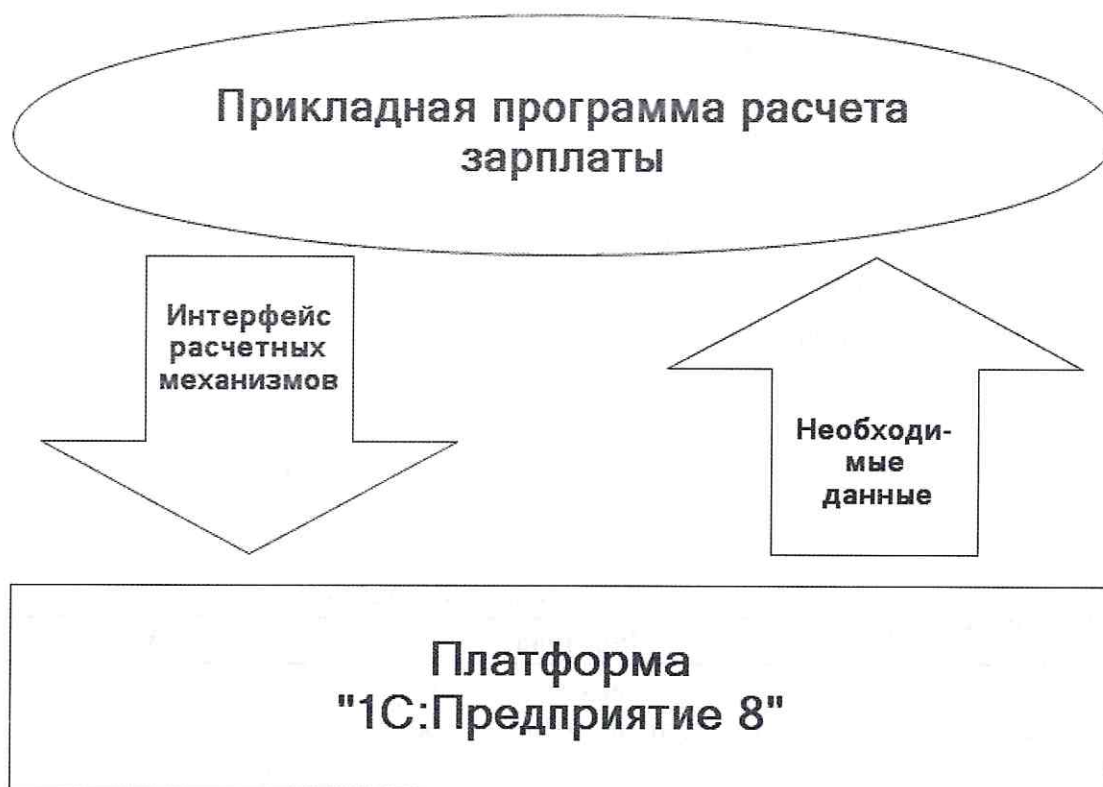
Чтобы получить правильный результат для вида расчета, который рассчитывается по базе, нужно знать состав его базовых видов расчета, а также придерживаться строгой очередности расчета. Ведь результаты базовых видов расчета должны быть получены до того, как начнется расчет того вида, для которого они являются базовыми, иначе сумма базы не может быть получена..

А чтобы получить правильный результат для вида расчета, который зависит от отработанного времени, нужно знать состав его вытесняющих видов расчета, чтобы, в случае их наличия за расчетный период, уметь определить интервалы фактического периода действия и просуммировать по ним фактически отработанное время.

Те учащиеся, которые имеют опыт работы в универсальных программных средах (Visual C++, Visual Basic, Delphi и других подобных), уже, видимо, поняли, какой трудоемкой может оказаться разработка системы расчета зарплаты, если самостоятельно разрабатывать эти два алгоритма: получение суммы базовых начислений и, особенно, получение суммы рабочего времени по фактическому периоду действия.

3.8. Состав расчетных механизмов платформы "1С:Предприятие" 8

Поэтому фирма "1С" сделала разработчикам систем расчета зарплаты весьма ценный подарок. Разработчики платформы встроили реализацию этих двух алгоритмов в платформу. Таким образом, вместо того, чтобы писать на встроенном языке чрезвычайно сложные программы получения суммы базы и суммы фактически отработанного времени, программист должен всего лишь правильно обратиться к этим расчетным механизмам через специальные средства платформы – и платформа, рассчитав эти данные, передаст их программе расчета зарплаты- так, как показано на рисунке:



А в самой программе расчета зарплаты остается реализовать только довольно простые алгоритмы, которые воспользуются этими данными.

Важно!

Итак, состав расчетных механизмов платформы следующий:

- механизм получения суммы базовых начислений;
- механизм получения суммы рабочего времени по фактическому периоду действия, а также по периоду действия.

Изучение этих механизмов и средств обращения к ним и составляет весь предмет нашего курса.

А теперь можно приступить к практическим действиям по изучению расчетных механизмов платформы 1С 8.

4. Постановка учебной задачи

4.1. Техническое задание, первая очередь

Начнем с того, что изложим "техническое задание" (ТЗ) на разработку простой расчетной конфигурации, которая будет реализовывать расчет зарплаты на основании некоторого состава видов расчета.

Для ясности понимания мы изложим наше маленькое ТЗ в двух частях; сначала реализуем первую часть ("первая очередь ТЗ"), а потом вторую – это будет "вторая очередь ТЗ".

ТЗ будет представлять собой перечень видов расчета, которые должна уметь рассчитывать наша конфигурация, и их расчетных формул. Представим перечень этих видов расчета "первой очереди" в виде следующей таблицы:

Вид расчета	Способ расчета	
	Формула	Имя
Оклад	$[\text{Результат}] = [\text{Ставка оклада}] * [\text{Число отработанных дней}] / [\text{Число рабочих дней в месяце}]$	По окладу
Оклад на выезде	$[\text{Результат}] = [\text{Ставка оклада}] * [\text{Число отработанных дней}] / [\text{Число рабочих дней в месяце}]$	По окладу
Доплата за квалификацию	$[\text{Результат}] = [\text{Процент доплаты}] * [\text{Сумма базы}] / 100$	Процентом

В этой таблице "Способ расчета" – это расчетная формула, снабженная содержательным наименованием (именем), чтобы можно было расчетную формулу идентифицировать.

Эти расчетные формулы нам уже знакомы.

Вид расчета "Оклад" – начисляется сотруднику, когда он работает на своем рабочем месте (в офисе).

Вид расчета "Оклад на выезде" отличается от вида расчета "Оклад" повышенной ставкой оклада. Он начисляется, когда сотрудник работает на выезде (например, выезжает в налоговую инспекцию или на другую территорию, где не очень комфортные условия труда). Очевидно, когда сотрудник находится на выезде, ему должен начисляться только "Оклад на выезде", а когда сотрудник находится на своем рабочем месте, то ему должен начисляться только "Оклад" – следовательно, вид расчета Оклад на выезде является вытесняющим видом расчета для "Оклад".

А вид расчета "Доплата за квалификацию" рассчитывается по базе. В сумму базы должны войти результаты расчета "Оклад" и "Оклад на выезде".

4.2. Исходные данные и необходимые данные

Присмотримся к операндам наших расчетных формул.

[Ставка оклада] и [Процент доплаты] – очевидно, должны быть известны до начала расчета. И сам сотрудник, и расчетчик знают, за какой оклад и за какой процент доплаты сотрудник работает.

А вот [Число отработанных дней], [Число рабочих дней в месяце] и [Сумма базы] – видимо, до начала расчета не известны, и их нам предстоит получить в процессе расчета, обратившись к расчетным механизмам платформы.

Важно!

Данные, которые известны до начала расчета зарплаты, мы будем отныне называть **исходными данными**.

Данные, которые предстоит получить в течение самого сеанса расчета зарплаты, мы будем отныне называть **необходимыми данными**.

4.3. Окончательная формулировка первой очереди технического задания

С учетом проведенного анализа мы можем дополнить наше ТЗ данными о базовой зависимости и зависимости по периоду действия наших видов расчета, указав в таблице состав базовых и вытесняющих видов расчета (ВР) там, где данная зависимость фигурирует в расчетной формуле:

Вид расчета	Способ расчета		Зависимости	
	Формула	Имя	Вытесняющие ВР	Базовые ВР
Оклад	$\text{[Результат]} = \frac{\text{[Ставка оклада]} * \text{[Число отработанных дней]}}{\text{[Число рабочих дней в месяце]}}$	По окладу	Оклад на выезде	
Оклад на выезде	$\text{[Результат]} = \frac{\text{[Ставка оклада]} * \text{[Число отработанных дней]}}{\text{[Число рабочих дней в месяце]}}$	По окладу		
Доплата за квалификацию	$\text{[Результат]} = \frac{\text{[Процент доплаты]} * \text{[Сумма базы]}}{100}$	Процентом		1. Оклад 2. Оклад на выезде

4.4. Создание учебной конфигурации

Создадим, например, на диске С: (или любом другом доступном носителе) папку, в которой будем разрабатывать нашу конфигурацию, назовем ее Курс Решение расчетных задач 1С 8. Запустим "1С:Предприятие 8.3" и в созданной папке создадим пустую информационную базу файлового режима. Таким образом, путь к базе у нас будет таким:

С:\Курс Решение расчетных задач 1С 8\

А имя нашей базы в списке информационных баз пусть будет таким:

Курс "Решение расчетных задач"

Загрузим в созданную базу выгрузку так называемой каркасной конфигурации. Каркасная конфигурация, точнее, каркасная база, содержит некоторые объекты общего назначения: справочники, регистры сведений и т.п., которые будут использоваться нами в нашей разработке, но которые непосредственно не относятся к расчетным механизмам. Таким образом, загрузив каркасную базу, мы не будем тратить время на создание объектов общего назначения.

5. Расчетные объекты конфигурации

5.1. Планы видов расчета

Очевидно, наш состав видов расчета с относящимися к ним данными (способ расчета и другие данные) необходимо хранить в информационной базе как некоторую нормативно-справочную информацию.

Мы могли бы хранить наш перечень видов расчета, например, в каком-нибудь справочнике, одном или нескольких. Но справочник – это объект метаданных общего назначения, а виды расчета должны иметь некоторые дополнительные свойства, чтобы расчетные механизмы платформы могли функционировать.

Поэтому состав видов расчета в расчетных решениях 1С хранится не в справочниках общего вида, а в специальных объектах конфигурации, очень похожих на справочник, но имеющих дополнительные специфические свойства метаданных.

Такой объект конфигурации называется **план видов расчета**. А элементы плана видов расчета (его "строки") называются **видами расчета**.

План видов расчета, как и справочник, является ссылочным объектом. Он тоже имеет два стандартных реквизита – "Код" и "Наименование", а также может иметь произвольное число прикладных (создаваемых разработчиком) реквизитов и произвольное число прикладных табличных частей. Однако, как мы увидим далее, план видов расчета имеет также от одной до трех стандартных табличных частей – т.е. таких, которые, подобно стандартным реквизитам, создаются и используются платформой.

Как и справочник, план видов расчета допускает наличие predetermined элементов – predetermined видов расчета.

В качестве отличий от справочника следует указать, что план видов расчета не поддерживает некоторых возможностей, которые поддерживаются платформой для справочника. план видов расчета не поддерживает:

- возможность автонумерации, т.е. автоматической генерации значения поля "Код" для нового вида расчета;
- возможность контроля уникальности кода платформой;
- иерархическую организацию видов расчета, ни в каком виде – ни типа "группа-элемент", ни типа "элемент-элемент".

5.2. Создание плана видов расчета "Основные Начисления"

5.2.1. Создание объекта конфигурации "ПланВидовРасчета"

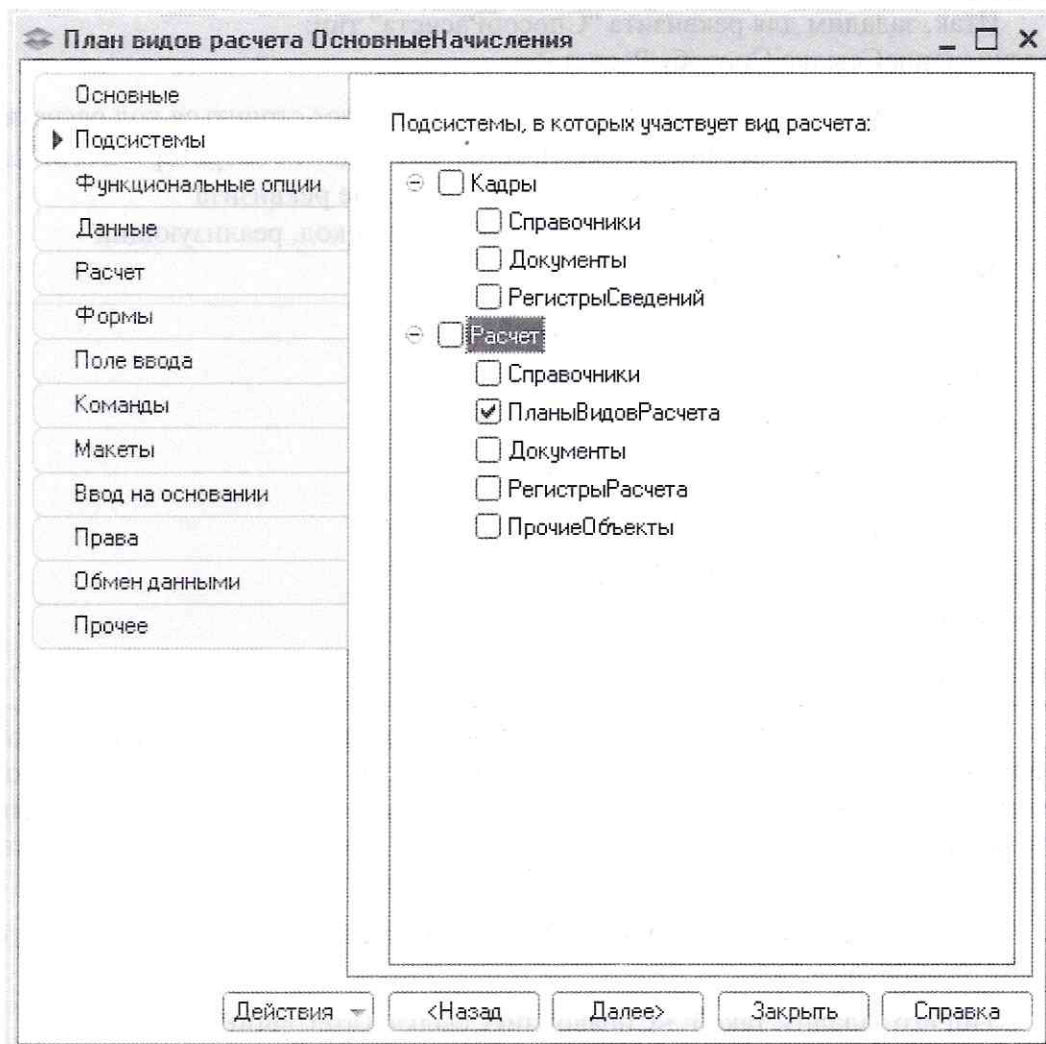
Разработку нашей конфигурации мы и начнем с создания такого "справочника" специального вида – плана видов расчета. Мы назовем его "Основные Начисления".

Итак, откроем нашу конфигурацию "Курс Решение расчетных задач" в режиме Конфигуратор и обратим свое внимание на ветвь дерева метаданных "Планы видов расчета". В этой ветви создаются объекты, о которых мы говорим.

Добавим новый объект в ветвь "Планы видов расчета" (кнопка "Добавить" панели дерева конфигурации). В открывшемся окне редактора объекта конфигурации зададим имя нашего объекта: "ОсновныеНачисления". (Так принято называть в типовых расчетных конфигурациях план видов расчета, который наиболее часто используется, будем следовать этой традиции).

The screenshot shows a window titled "План видов расчета ОсновныеНачисления". On the left is a tree view with categories: Основные, Подсистемы, Функциональные опции, Данные, Расчет, Формы, Поле ввода, Команды, Макеты, Ввод на основании, Права, Обмен данными, and Прочее. The "Основные" category is selected. On the right, there are several input fields: "Имя:" (filled with "ОсновныеНачисления"), "Синоним:" (filled with "Основные начисления"), "Комментарий:" (empty), "Представление объекта:" (empty), "Расширенное представление объекта:" (empty), "Представление списка:" (empty), "Расширенное представление списка:" (empty), and "Пояснение:" (empty text area). At the bottom, there are buttons: "Действия" (dropdown), "<Назад", "Далее>", "Закреть", and "Справка".

На закладке "Основные" редактора объекта конфигурации нам больше ничего делать не надо. Перейдем на закладку "Подсистемы" и зададим связь нашего объекта с подсистемой "ПланыВидовРасчета" раздела "Расчет" нашей конфигурации:



5.2.2. Создание и настройка реквизитов плана видов расчета

Теперь перейдем на закладку "Данные" редактора объекта конфигурации.

Здесь сначала изменим свойства стандартных реквизитов "Код" и "Наименование". Длину кода сделаем равной 5, а длину наименования сделаем равной 100 – так мы будем чувствовать себя более комфортно, когда будем создавать виды расчета в режиме пользователя.

Теперь создадим прикладные реквизиты нашего объекта. Реквизиты объекта конфигурации создаются для того, чтобы хранить дополнительные данные, относящиеся к элементу данных.

Продумаем состав реквизитов.

1. Нам в первую очередь нужно хранить информацию о расчетной формуле, по которой будет рассчитываться каждый вид расчета. Для этого **создадим реквизит**, который так и будет называться: "СпособРасчета". Вспомним, что способ расчета – это идентификатор (имя) расчетной формулы, под которым его знаем мы и по которому его будет идентифицировать наше программное обеспечение.

Имена способов расчета мы будем задавать не строкой, а значением перечисления "СпособыРасчета", которое уже имеется в каркасной конфигурации. Перечисление уже содержит несколько значений; в процессе разработки нашей конфигурации в него будут добавляться новые значения.

Итак, зададим для реквизита "СпособРасчета" тип:
<ПеречислениеСсылка.СпособыРасчета>

Забегая вперед, скажем пару слов о том, как будет строиться код расчетной программы. Он у нас будет представлять собой ветвящийся оператор "Если ... Тогда ...", ветви которого будут проверять значение реквизита "СпособРасчета", а в гнездах ветвей будет находиться код, реализующий расчетные формулы, примерно так:

```
Если СпособРасчета = Способ_А Тогда
    Сумма = <расчетная формула А>;
ИначеЕсли СпособРасчета = Способ_В Тогда
    Сумма = <расчетная формула В>;
ИначеЕсли СпособРасчета = Способ_В Тогда
    Сумма = <расчетная формула В>;
    . . . . .
КонецЕсли;
```

2. Вспомним, что на видах расчета может существовать базовая зависимость. В этом случае предполагается, что необходимо задавать очередность расчета видов расчета во времени, чтобы к моменту начала расчета такого вида, в расчетную формулу которого входит сумма базы, все виды расчета, результаты которых должны входить в сумму базы, были уже рассчитаны.

Поэтому создадим еще один реквизит плана видов расчета и назовем его так: "КатегорияРасчета".

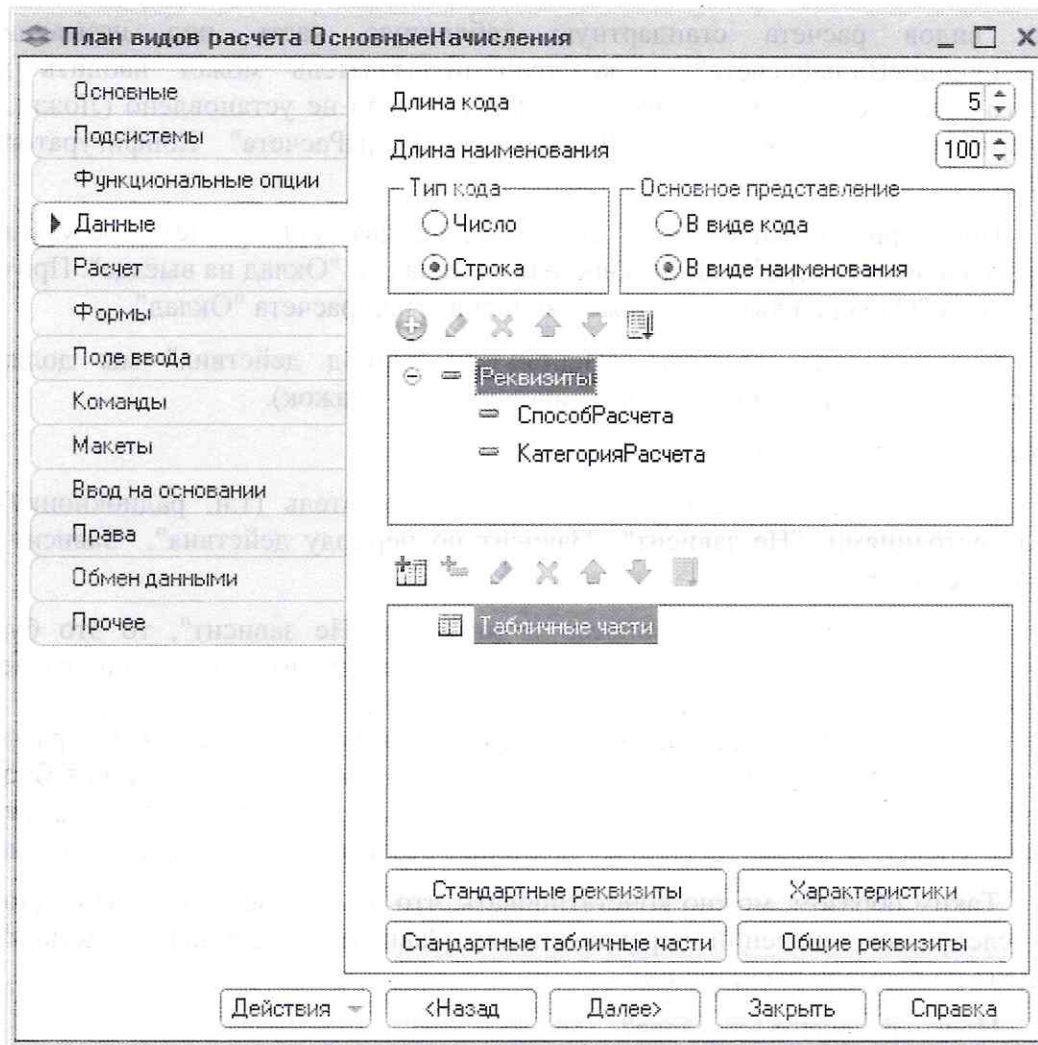
Тип его зададим такой: <СправочникСсылка.КатегорииРасчета>

Для целей хранения порядка очередности расчета в каркасной конфигурации имеется справочник "КатегорииРасчета". Коды его элементов (типа <Число>) суть значения очередности расчета, а наименования отражают содержание каждого "значения очередности".

Поясним. Виды расчета, которым будет назначена категория расчета с кодом 0 ("Первичное") будут рассчитываться в первую очередь: наша программа, которую мы напишем, определит это по значению реквизита "КатегорияРасчета". Это будут такие виды расчета, результаты которых входят в сумму базы других видов расчета, которым мы назначим категорию расчета с кодом 1 ("Зависимое первого уровня"). Таким образом, к моменту, когда будет начат расчет этих видов расчета, сумму базы для них будет уже возможно получить, поскольку результаты их расчета будут известны. А для тех видов расчета, в сумму базы которых должны входить результаты "зависимых первого уровня", мы назначим категорию расчета с кодом 2 ("Зависимые второго уровня"), и т.д.

Таким образом, наше программное обеспечение, которое мы разработаем впоследствии, будет "знать", в каком порядке по времени рассчитывать тот или иной вид расчета, и к моменту начала его расчета можно будет получить сумму базы для подстановки в расчетную формулу.

Итак, в результате нашей работы по созданию и настройке реквизитов плана видов расчета закладка "Данные" редактора объекта конфигурации будет выглядеть так:



Отметим, что мы не будем создавать для наших планов видов расчета прикладных табличных частей, они нам не нужны. Но мы вскоре убедимся, что для объекта "План видов расчета" Конфигуратор автоматически создает стандартные табличные части, которые будет заполнять пользователь, а данные их будут использовать расчетные механизмы платформы.

5.2.3. Настройка расчетных свойств плана видов расчета

Перейдем на закладку "Расчет". Этой закладки нет, когда мы конфигурируем объект типа "Справочник". На ней настраиваются специальные расчетные свойства, значения которых используются расчетными механизмами платформы.

Свойство "Использует период действия"

Это свойство имеет тип <Булево>. Оно "говорит" расчетным механизмам, будут ли в нашем плане видов расчета храниться такие, которые по расчетной формуле зависят от отработанного времени, или же нет. Если это так, то, следовательно, для таких видов расчета может быть задана зависимость по периоду действия от других – т.е. для каждого такого вида расчета может быть задан состав вытесняющих видов расчета, которые будут "рвать" период действия данного вида расчета.

Если свойство "Использует период действия" установлено в значение Истина (флажок установлен), то Конфигуратор автоматически создаст для нашего

плана видов расчета стандартную табличную часть под названием "ВытесняющиеВидыРасчета", в которую пользователь может вводить для каждого вида расчета состав таковых. Если свойство не установлено (Ложь), то стандартная табличная часть "ВытесняющиеВидыРасчета" Конфигуратором создана не будет.

Посмотрим в наше ТЗ. У нас имеются два вида расчета, результаты которых зависят от отработанного времени: "Оклад" и "Оклад на выезде". Причем вид расчета "Оклад на выезде" должен вытеснять вид расчета "Оклад".

Следовательно, свойство "Использует период действия" мы должны установить в значение Истина (т.е. установить этот флажок).

Свойство "Зависимость от базы"

Это свойство представляет собой переключатель (т.н. радиокнопку) с тремя состояниями: "Не зависит", "Зависит по периоду действия", "Зависит по периоду регистрации".

Если установить это свойство в значение "Не зависит", то это будет означать, что в данном плане видов расчета нельзя хранить такие виды расчета, формула которых зависит от суммы базы. Если взглянуть а наше ТЗ, то мы увидим, что нас это не устраивает: в списке видов расчета имеется вид расчета "Доплата за квалификацию", формула которого содержит операнд [Сумма базы], а сама сумма базы, как явствует из ТЗ, должна складываться из результатов расчета "Оклад" и "Оклад на выезде" (колонка "Базовые ВР" таблицы техзадания).

Таким образом, можно констатировать, что нам придется выбрать одно из двух следующих значений переключателя: "Зависит по периоду действия" и "Зависит по периоду регистрации".

Но что означают эти слова?

Понятие базового периода

Подумаем: если нам нужно рассчитать сотруднику вид расчета "Доплата за квалификацию", скажем, за январь 2014г., то будет абсолютно неправильно для получения суммы базы суммировать результаты "Оклад" и "Оклад на выезде" этого сотрудника за все время его работы на предприятии, например, за 10 лет... Следовательно, нам нужно задаться каким-то логически обоснованным интервалом, за который надо получать сумму этих начислений. Если вид расчета "Доплата за квалификацию" рассчитывается сотрудникам ежемесячно, то логично будет суммировать результаты базовых видов расчета именно за этот месяц. Если, например, мы собираемся рассчитать сотруднику вид расчета "Доплата за квалификацию" за январь 2014г., то для получения суммы базы и нужно суммировать результаты видов расчета "Оклад" и "Оклад на выезде" этого сотрудника за этот период, т.е. за интервал с 01.01.2014 по 31.01.2014.

Важно!

Интервал, который задается для выборки и суммирования результатов базовых видов расчета, называется **базовым периодом**

Важно!

По ходу дела подытожим, сколько у нас появилось понятий со словом "период" в названии. Их четыре:

- "Период регистрации";
- "Период действия";
- "Фактический период действия";
- "Базовый период".

Определения этих понятий – см. выше.

Выбор критерия попадания в базовый период – задание значения свойства "Зависимость от базы"

Итак, для функционирования расчетного механизма получения суммы базы необходимо задавать базовый период.

Но возникает следующий вопрос. Ну, вот, допустим, расчетный механизм обнаружил цифру результата сотрудника Иванова И.И. по виду расчета "Оклад". Механизму перед началом его работы заданы границы интервала базового периода. А как этот расчетный механизм будет определять, относится этот результат к базовому периоду или нет, т.е. суммировать его или пропустить?

У расчетного механизма получения базы выбор невелик. Ему "известны" только два понятия: **период действия** начисления и **период регистрации** начисления. Поэтому механизму получения суммы базы можно задать только один из вариантов критерия суммирования результата базового вида расчета:

- **попадание периода действия этого результата расчета в базовый период;**
- **попадание периода регистрации этого результата расчета в базовый период.**

Это и есть два значения свойства "Зависимость от базы": "Зависит по периоду действия" и "Зависит по периоду регистрации".

Ну, а нам какой критерий выбрать, какое значение этого свойства задать?

Поскольку до сих пор мы вели разговор о видах расчета, которые являются начислениями, то, видимо, для них логично использовать критерий попадания периода действия в базовый период. Если сотрудник получает оклад за январь 2014г., то и для расчета январской доплаты за квалификацию разумно использовать значение именно этого же, январского оклада, а не прошлогоднего декабрьского или сентябрьского. Так и есть. Мы установим для нашего плана видов расчета "Основные Начисления" свойство "Зависимость от базы" – т.е. критерий попадания в базовый период – в значение "Зависит по периоду действия". Это, как мы только что вывели, вполне разумно и логически обоснованно. Пусть для всех видов расчета, которые используют расчетную формулу с суммой базы за некий базовый период, расчетный механизм суммирует результаты базовых видов расчета именно за этот период.

А когда же может пригодиться второй критерий попадания в базовый период – "Зависит по периоду регистрации"? – Для расчета удержаний.

Дело в том, что правильность расчета удержаний проверяется внешними организациями (налоговой инспекцией, пенсионным фондом и т.п.). А они для этой проверки используют данные бухгалтерской отчетности предприятия, других данных у них просто нет. А какой период указывается в бухгалтерской

отчетности? – **Период регистрации**. Если мы будем использовать для расчета удержаний критерий попадания в базовый период периода действия, то внешние организации не смогут проверить правильность расчета удержаний, поскольку им известен только период регистрации (из бухгалтерности), а период действия не известен. Поэтому для расчета удержаний (на следующих этапах нашего курса) мы будем использовать критерий попадания в базовый период периода регистрации.

Важно!

Для планов видов расчета, в которых хранятся виды начислений, свойство "Зависимость от базы" устанавливается в значение "Зависит по периоду действия".

Для планов видов расчета, в которых хранятся виды удержания, свойство "Зависимость от базы" устанавливается в значение "Зависит по периоду регистрации".

Итак, после столь длительного обсуждения мы установим свойство "Зависимость от базы" плана видов расчета "Основные Начисления" в значение "Зависит по периоду действия".

Свойство "Базовые планы видов расчета"

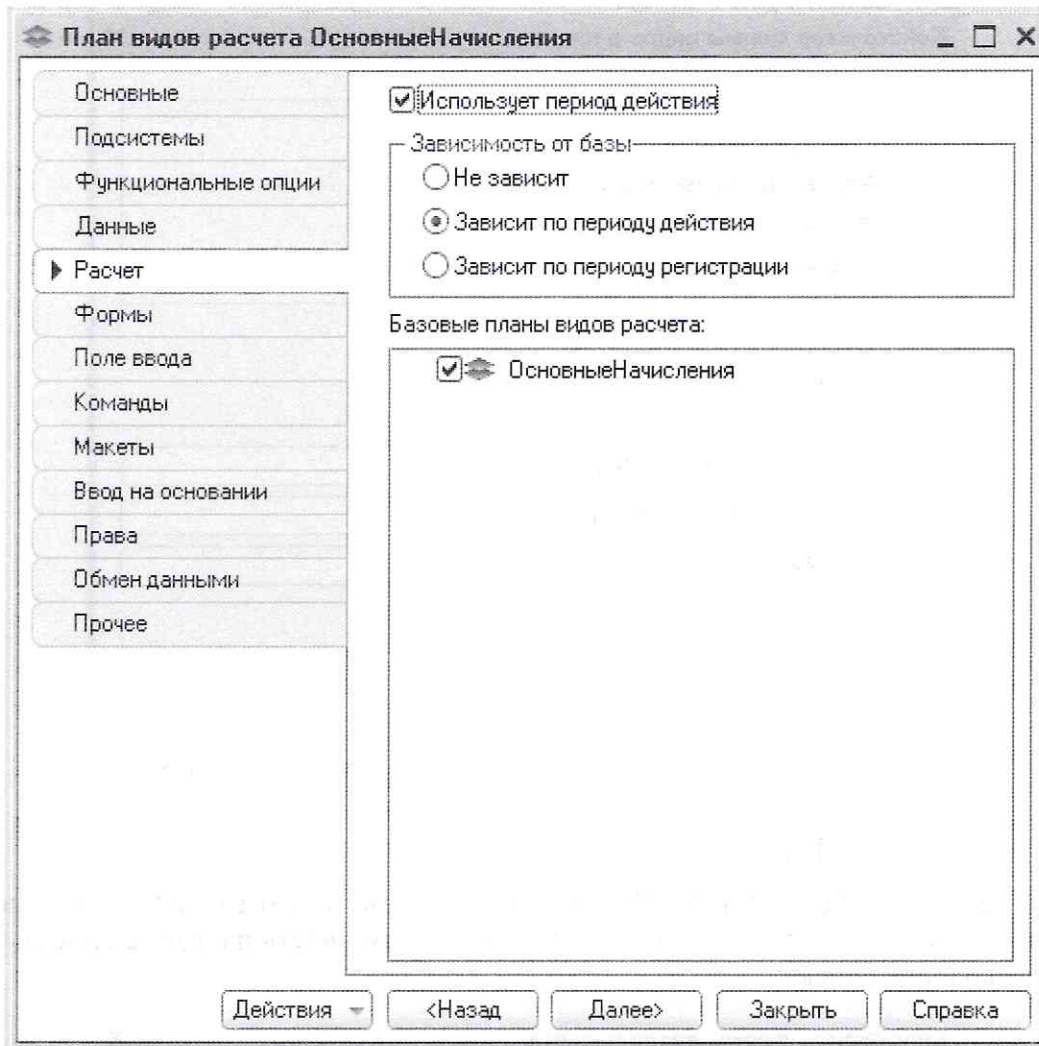
Где могут для какого-либо вида расчета, зависящего от базы, находиться базовые планы видов расчета? В конфигурации планов видов расчета может быть несколько, и в любом из них могут располагаться базовые виды расчета для наших видов расчета.

Но мы должны в настройках плана видов расчета указать их все. Для этого служит свойство "Базовые планы видов расчета" на закладке "Расчет". Оно представляет собой список всех имеющихся в конфигурации планов видов расчета с флажками. Мы должны установить флажки возле имен тех планов видов расчета, в которых собираемся хранить базовые виды расчета для нашего плана.

У нас в конфигурации пока только один план видов расчета – "Основные Начисления". Согласно нашему техзаданию, базовые виды расчета для вида "Доплата за квалификацию" – "Оклад" и "Оклад на выезде" – будут храниться в этом же плане видов расчета, в "Основные Начисления". Значит, нам достаточно для него установить флажок возле имени "Основные Начисления", т.е. он сам для себя служит базовым планом видов расчета.

Так мы и сделаем.

Итак, окончательная настройка расчетных свойств нашего плана видов расчета "Основные Начисления" выглядит таким образом:



5.2.4. Формы плана видов расчета

Для удобства работы пользователя можно создать формы плана видов расчета. Создадим две наиболее употребительные формы – форму списка и форму вида расчета (для объекта "Справочник" последняя называется формой элемента).

Форма списка

Создадим форму списка нашего плана видов расчета: перейдем на закладку "Формы" окна редактора объекта конфигурации и нажмем "Добавить".

В открывшейся форме Конструктора форм укажем при помощи радиокнопки тип создаваемой формы – "Форма списка плана видов расчета":

Конструктор формы видов расчета

Выберите тип формы:

- Форма вида расчета
- Форма списка плана видов расчета
- Форма выбора плана видов расчета
- Произвольная форма

Назначить форму основной

Имя:

Синоним:

Комментарий:

< Назад Далее > Готово Отмена Справка

и нажмем "Далее".

В следующей форме Конструктора форм отметим флажками поля (реквизиты) плана видов расчета, которые мы хотим видеть на форме списка, т.е. состав колонок списка:

Конструктор формы видов расчета

Располагать элементы управления в колонку(и)

Реквизиты

- Ссылка
- Код
- Наименование
- ПометкаУдаления
- Предопределенный
- ПериодДействияБазовый
- СпособРасчета
- КатегорияРасчета

< Назад Далее > Готово Отмена Справка

Обратим только внимание на стандартный реквизит "ПериодДействияБазовый". Мы подробно обсудим его назначение и применение позже, а сейчас только упомянем, что оно имеет тип <Булево>.

Нажмем "Готово", и форма списка создана – открывается окно редактора управляемой формы. Редактировать форму мы не будем, поэтому окно редактора следует закрыть.

Форма вида расчета

Создадим форму вида расчета. Снова нажмем "Добавить" на закладке "Формы", зададим тип формы "Форма вида расчета" и сразу нажмем "Готово". Откроется редактор управляемой формы.

Обратим внимание на структуру основного реквизита формы под названием "Объект" (поле вверху справа):

Реквизит	Использовать всегда
Объект	<input checked="" type="checkbox"/>
Ссылка	<input checked="" type="checkbox"/>
Код	<input checked="" type="checkbox"/>
Наименование	<input checked="" type="checkbox"/>
ПометкаУдаления	<input checked="" type="checkbox"/>
Предопределенный	<input checked="" type="checkbox"/>
ИмяПредопределенныхДанных	<input checked="" type="checkbox"/>
ПериодДействияБазовый	<input checked="" type="checkbox"/>
СпособРасчета	<input checked="" type="checkbox"/>
КатегорияРасчета	<input checked="" type="checkbox"/>
БазовыеВидыРасчета	<input checked="" type="checkbox"/>
ВедущиеВидыРасчета	<input checked="" type="checkbox"/>
ВытесняющиеВидыРасчета	<input checked="" type="checkbox"/>

Мы видим в структуре объекта три стандартные табличные части: "БазовыеВидыРасчета", "ВедущиеВидыРасчета", "ВытесняющиеВидыРасчета". Это и есть стандартные табличные части, которые Конфигуратор создал автоматически и в которые пользователь будет вносить виды расчета, от которых данный вид расчета (открытый в форме) зависит соответствующим образом. О назначении стандартных табличных частей "БазовыеВидыРасчета" и "ВытесняющиеВидыРасчета" мы уже знаем – в них будут в режиме "1С:Предприятие" заноситься списки видов расчета, от которых данный вид расчета зависит по базе или по периоду действия.

О назначении третьей табличной части – "ВедущиеВидыРасчета" – мы будем говорить значительно позже. Она имеет отношение к выполнению перерасчетов.

Отметим, что план видов расчета не всегда имеет ровно три стандартные табличные части.

Табличная часть "ВытесняющиеВидыРасчета" не будет создана, если свойство "Использует период действия" плана видов расчета не установлено (у нас в курсе еще будет такой план видов расчета).

Табличная часть "БазовыеВидыРасчета" не будет создана, если свойство "Зависимость от базы" плана видов расчета установлено в положение "Не зависит".

Табличная часть "ВедущиеВидыРасчета" создается всегда.

5.2.5. Занесение видов расчета в план видов расчета "Основные Начисления"

Теперь надо заполнить наш план видов расчета согласно техзаданию

Предварительно повторим, что план видов расчета может иметь предопределенные элементы, как и справочник. Но мы пока не будем использовать эту возможность.

Итак, обновим конфигурацию (актуализируем наши изменения) и запустим режим "1С:Предприятие" (по умолчанию наша конфигурация запускается с интерфейсом "Такси").

После открытия окна "1С:Предприятие" перейдем в раздел "Расчет", откроем форму списка плана видов расчета "Основные начисления" и нажмем в ней кнопку "Создать". Откроется форма нового вида расчета. Заполним поля реквизитов вида расчета значениями для вида расчета "Оклад" следующим образом

Код: 00001

Наименование: Оклад

Базовый период как период действия: *Ложь* (флажок не установлен)

Способ расчета: "По окладу" (выбираем из списка значений перечисления "Способы расчета")

Категория расчета: "Первичное" (выбираем из справочника "Категории расчета").

Т.е., форма вида расчета примет следующий вид:

Оклад (Основные начисления) *

Записать и закрыть Записать Еще ▾

Код: 00001

Наименование: Оклад

Базовый период как период действия:

Способ расчета: По окладу ▾

Категория расчета: Первичное ▾

Базовые виды расчетов Ведущие виды расчетов Вытесняющие виды расчетов

Добавить Еще ▾

Вид расчета

Мы выбрали категорию расчета "Первичное", т.е. вид расчета "Оклад" будут рассчитываться в самую первую очередь. Ведь он, согласно расчетной формуле из техзадания, сам от базы не зависит, но от него зависят другие виды расчета ("Доплата за квалификацию"). Следовательно, его результат должен быть получен до того момента, когда начнется выполнение расчета "Доплата за квалификацию", поэтому мы назначаем ему категорию расчета "Первичное".

Пока у нас вид расчета "Оклад" – только один, поэтому никаких зависимостей мы на этом этапе задать не можем (т.е. не можем заполнить стандартные табличные части). Поэтому нажмем кнопку "Записать и закрыть". Форма вида расчета закроется, а вид расчета "Оклад" появится в форме списка нашего плана видов расчета.

Таким же образом создадим вид расчета "Оклад на выезде". Форма вида расчета для него должна иметь следующий вид:

Т.е. настройка реквизитов этого вида расчета, за исключением "Код" и "Наименование", идентична настройке реквизитов вида расчета "Оклад".

Также нажмем "Записать и закрыть", и новый вид расчета будет записан в информационную базу, а в форме списка появится вторая строка.

Теперь пора вспомнить, что вид расчета "Оклад на выезде" является вытесняющим для вида расчета "Оклад", поскольку начисляется за время "местных командировок" сотрудника. Снова откроем форму вида расчета "Оклад" (двойным щелчком по строке в форме списка), перейдем на закладку "Вытесняющие виды расчета", добавим в табличную часть новую строку и поместим в нее вид расчета "Оклад на выезде":

Оклад (Основные начисления) *

Записать и закрыть Записать Еще ▾

Код: 50001

Наименование: Оклад

Базовый период как период действия:

Способ расчета: По окладу ▾

Категория расчета: Первичное ▾

Базовые виды расчетов Ведущие виды расчетов Вытесняющие виды расчетов

Добавить Еще ▾

Вид расчета

Оклад на выезде

И вновь нажмем "Записать и закрыть", чтобы сохранить сделанные изменения в настройках вида расчета.

Важно!

Вытесняющие виды расчета не могут находиться в другом плане видов расчета, их можно назначить только из этого же плана видов расчета. Такое ограничение накладывает платформа.

Таким же образом создадим новый вид расчета "Доплата за квалификацию" и заполним значения его реквизитов следующим образом:

Какие отличия настроек этого вида расчета от настроек предыдущих двух?

Способ расчета - выбираем значение "Процентом", согласно техзаданию.

Этот вид расчета уже зависит по расчетной формуле от суммы базы, поэтому для него задаются уже базовые виды расчета. В табличную часть "Базовые виды расчета" на одноименной закладке вносим виды расчета "Оклад" и "Оклад по дням".

Поскольку оба базовых вида расчета имеют категорию расчета "Первичное", то наш новый вид расчета должен рассчитываться по очереди сразу же после них. Поэтому выбираем для него категорию расчета "Зависимое первого уровня".

Вспомним, что базовые виды расчета, в отличие от вытесняющих, могут располагаться в других планах видов расчета, которые имеются в конфигурации.

Табличную часть "Вытесняющие виды расчета" для него не заполняем, поскольку расчетная формула (способ расчета) "Процентом" не включает данные об отработанном времени.

Запишем новый вид расчета.

Виды расчета из плана видов расчета "Основные начисления" созданы.

6. Регистры расчета

6.1. Назначение и логическое устройство регистра расчета

План видов расчета, как мы узнали, является вариантом справочника, в котором список используемых на предприятии видов расчета всего лишь хранится.

Можно провести аналогию с решением оперативных задач: имеется справочник "Номенклатура", в котором хранятся позиции номенклатуры, которая находится в обороте предприятия. А получение результирующих показателей оперативного учета – остатков и оборотов по номенклатуре в различных периодах и разрезах – выполняет другой объект конфигурации, а именно регистр накопления. "Регистр накопления" – объект конфигурации, который реализует специализированный функционал: получение показателей типа "Остаток" и "Оборот". Т.е. регистр накопления несет основную функциональную нагрузку в системах оперативного учета.

Подобным образом обстоит дело и с расчетными механизмами. План видов расчета – это вспомогательный объект, хранящий, подобно справочнику, состав видов расчета. А основную функциональную нагрузку расчетных механизмов несет другой объект конфигурации – **регистр расчета**.

Регистр расчета – это один из четырех классов регистров конфигурации 1С. Как и регистры остальных трех классов, регистр расчета можно логически представить в виде двух компонентов:

- **хранилище первичной информации;**
- **"калькулятор результирующих показателей".**

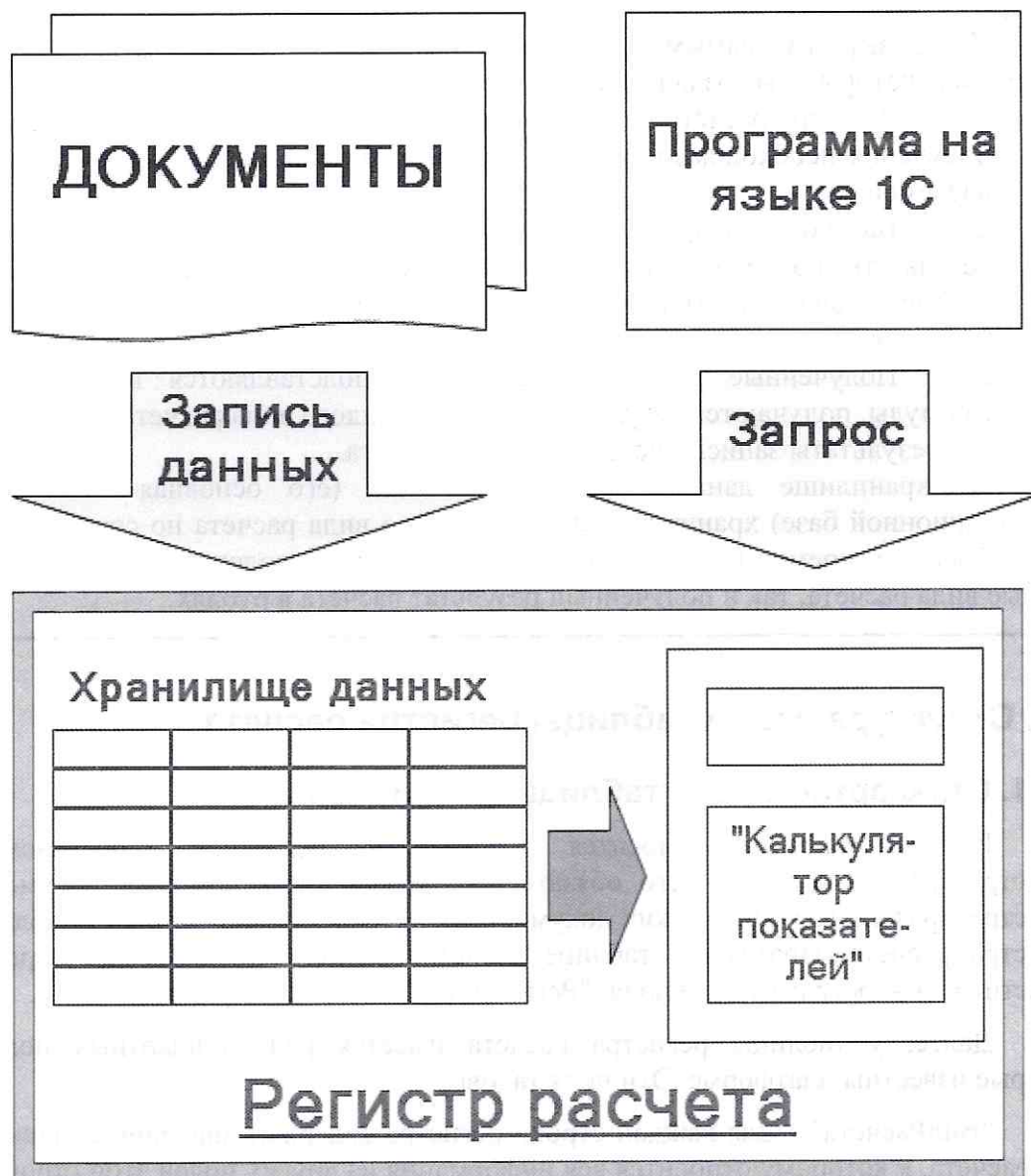
Только результирующие показатели у регистра расчета, понятно, уже не остатки и обороты, как у регистра накопления и регистра бухгалтерии. Каковы же результирующие показатели, которые выдает "калькулятор показателей" регистра расчета?

Вспомним, что мы в начале курса называли необходимыми данными. Это значения операндов расчетных формул, которые неизвестны к началу расчета зарплаты, а реализация алгоритмов получения которых на встроенном языке платформы довольно сложна для прикладного программиста 1С. Это такие данные:

- **количество рабочего времени по плану в рассчитываемом периоде;**
- **количество фактически отработанного времени;**
- **сумма базовых начислений.**

Вот эти данные и получает "калькулятор показателей" регистра расчета, это его функционал.

Итак, функционирование регистра расчета можно условно представить следующим рисунком:



Взаимодействие программного обеспечения с регистром расчета осуществляется по общей схеме для всех регистров:

- объекты "Документ" делают записи (движения) в "хранилище данных" регистра расчета;
- прикладные программы выдают запросы к виртуальным таблицам регистра расчета для получения вышеуказанных результирующих показателей суммы базы и отработанного времени, которые подлежат подстановке в расчетные формулы.

Важно!

Итак, выразим самыми общими словами, как будут работать наши программы, которые мы создадим в процессе обучения.

1. Регистру расчета передаются исходные данные, которые нужны для получения необходимых данных: период действия, состав базовых видов расчета и т.п.
2. Выдаются запросы к виртуальным таблицам регистра расчета, результаты которых будут содержать необходимые данные: количество рабочего времени (плановое и фактическое) и/или сумму базовых начислений.
3. Полученные необходимые данные подставляются в расчетные формулы, получаются результаты расчета каждого вида расчета в рублях, и эти результаты записываются в регистр расчета.

А "хранилище данных" регистра расчета (его основная таблица в информационной базе) хранит записи для каждого вида расчета по сотрудникам (оклад Иванова, премия Петровой и т.д.). Каждая запись содержит как исходные данные вида расчета, так и полученный результат расчета в рублях.

6.2. Структура полей таблицы регистра расчета

6.2.1. Стандартные поля таблицы регистра расчета

Регистр расчета относится к разряду регистров, управляемых регистраторами – т.е. для него обязательно должен быть назначен документ-регистратор (или несколько таких документов), которые делают записи в таблицу регистра расчета. Поэтому в таблице регистра расчета имеются два хорошо известных 1С-программистам поля: "Регистратор" и "НомерСтроки".

Далее, у таблицы регистра расчета имеется ряд стандартных полей, которые известны платформе. Эти поля таковы.

"ВидРасчета" – для каждой строки регистра содержит значение ссылки на вид расчета, к которому относится вся информация из других полей этой строки – как исходная, так и результирующая.

"ПериодРегистрации" – поле типа <Дата>, представляющее период регистрации. Поскольку длительность периода регистрации всегда определена – месяц, квартал, год – то для задания периода регистрации достаточно только даты начала периода. В этом поле всегда находится дата первого числа периода регистрации.

"ПериодДействияНачало" и "ПериодДействияКонец" – поля типа <Дата>, задающие границы периода действия, для которого рассчитывается вид расчета. Период действия не обязан начинаться с первого числа месяца (например, период действия вида расчета "Командировка" может начинаться с любого числа), то для задания периода действия, в отличие от периода регистрации, существуют два поля.

"БазовыйПериодНачало" и "БазовыйПериодКонец" – поля типа <Дата>, задающие границы базового периода, который может иметь произвольные границы.

"Сторно" – поле типа <Булево>. На данном этапе нашего обучения мы раскрывать назначение этого поля не будем, мы вернемся к нему гораздо позже.

6.2.2. Прикладные поля таблицы регистра расчета

Как и любой другой регистр, регистр расчета может иметь (и всегда имеет) ряд прикладных полей, создаваемых разработчиком, т.е. нами: измерения, ресурсы и реквизиты.

Любой регистр расчета имеет как минимум одно измерение для указания сотрудника, к которому относится эта строка регистра расчета. Ведь не может быть "оклада вообще", это обязательно оклад Иванова, Петрова и т.п.

У регистра расчета может быть и более одного измерения, если расчеты сотрудника требуется дополнительно детализировать – например, по должностям, если сотрудник является внутренним совместителем.. Так, если г-жа Петрова совмещает должности бухгалтера и программиста, то оклад рассчитывается отдельно Петровой – бухгалтеру и Петровой – программисту.

Для хранения результатов расчета создаются ресурсы, которые для регистра расчета всегда имеют тип <Число>. Ресурсов может быть несколько, например, если требуется производить расчеты и хранить результаты расчетов в нескольких валютах. Ресурсы – это целевые поля регистров, значения в них используются платформой для получения результирующих показателей, свойственных данному классу регистра.

Регистр расчета также может иметь реквизиты. **Реквизиты** – это вспомогательные поля, туда прикладные программы могут записывать информацию, относящуюся к конкретной записи. Обычно реквизиты не участвуют в работе функционала регистров, но регистр расчета представляет исключение из этого правила – у него некоторые реквизиты участвуют в работе его функционала, в чем мы убедимся в дальнейшем.

6.3. Создание и настройка регистра расчета "ОсновныеНачисления"

6.3.1. Добавление нового объекта конфигурации "Регистр расчета"

Создадим объект конфигурации "Регистр расчета". Установим выделение на ветвь "Регистры расчета" дерева конфигурации и нажмем "Добавить". Откроется редактор объекта метаданных "Регистр расчета":

Регистр расчета РегистрРасчета1

Основные

Подсистемы

Функциональные опции

Данные

Перерасчеты

Регистраторы

Формы

Команды

Макеты

Права

Обмен данными

Прочее

Имя: РегистрРасчета1

Синоним:

Комментарий:

План видов расчета: ... X

Период действия

График: ... X

Значение графика: ... X

Дата графика: ... X

Базовый период

Периодичность: Месяц

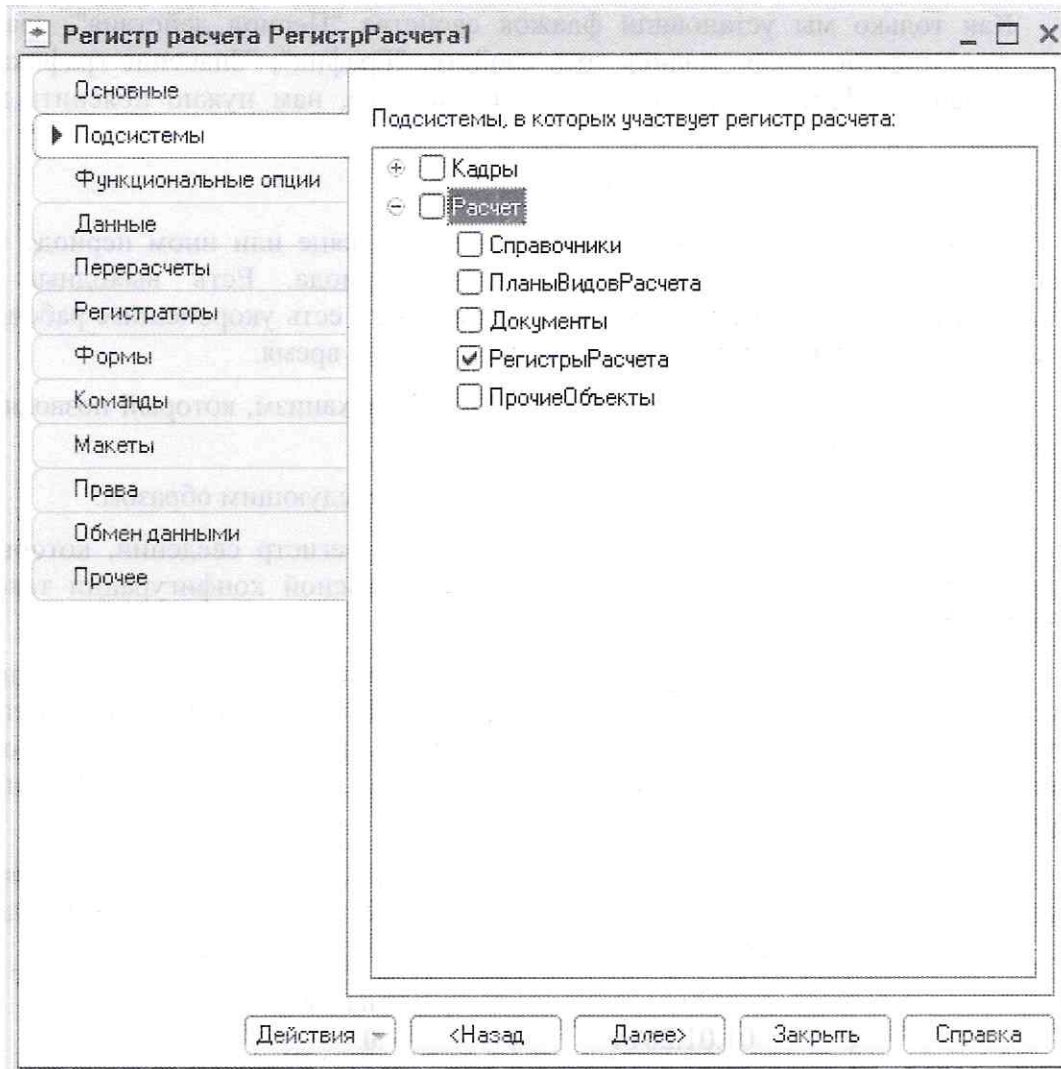
Представление списка:

Расширенное представление списка:

Пояснение:

Действия <Назад Далее> Закреть Справка

В отличие от плана видов расчета, на закладке "Основные" регистра расчета нам предстоит выполнить довольно много работы по настройке. Поэтому давайте сразу перейдем на закладку "Подсистемы" и свяжем наш регистр с подсистемой "РегистрыРасчета" раздела "Расчет", как показано на следующем рисунке:



А потом снова вернемся на закладку "Основные".

6.3.2. Настройка основных свойств регистра расчета "ОсновныеНачисления"

Свойство "Имя" – введем имя "ОсновныеНачисления". Т.е. наш регистр расчета будет называться так же, как план видов расчета, с которым он будет функционально связан.

Свойство "План видов расчета". Сюда надо ввести ссылку на план видов расчета, с которым будет связан наш регистр, т.е. какой план видов расчета будет регистр "обслуживать". Нажмем кнопку выбора, откроем список и выберем наш план видов расчета "ОсновныеНачисления".

Свойство "Период действия". Если установить этот флажок, то наш регистр расчета будет снабжен функционалом получения количества рабочего времени – планового и фактического. Если флажок не устанавливать, то регистр не сможет выдавать эти показатели, а попытка обратиться к нему для их получения вызовет программную ошибку.

Установим флажок свойства "Период действия", потому что в нашем регистре расчета будут рассчитываться виды расчета "Оклад" и "Оклад на выезде", которые зависят от количества отработанного времени.

Как только мы установили флажок свойства "Период действия", сразу стали доступны поля для задания трех свойств: "График", "Значение графика", "Дата графика". Перед тем, как задать эти свойства, нам нужно пояснить их назначение.

Календарь – регистр сведений "ГрафикиРаботы"

Очевидно, что количество рабочих дней в месяце или ином периоде не равно количеству календарных дней этого периода. Есть выходные и праздничные дни, в которые сотрудники не работают, есть укороченные рабочие дни... Для расчета должно учитываться только рабочее время.

Следовательно, регистру расчета необходим механизм, который позволяет ему учитывать только рабочее время.

В платформе 1С 8 этот механизм реализуется следующим образом.

В конфигурации создается неперiodический регистр сведений, который обычно называют "ГрафикиРаботы". В нашей каркасной конфигурации такой регистр уже есть.

В простейшем варианте регистр имеет одно измерение типа <Дата> и один ресурс типа <Число>. Измерение называется "Дата", а ресурс называется "Значение". В этом регистре мы должны обеспечить на каждую дату расчетного периода запись, значение измерения которой – календарная дата, а значение ресурса – длительность рабочего времени этой даты.

Если принять, что рабочее время исчисляется в днях, тогда значение ресурса "Значение" для каждой даты рабочего дня будет равно 1, а для каждой даты выходного дня будет равно 0. Например – для января 2014г.:

Дата	Значение
01.01.2014	0
02.01.2014	0
03.01.2014	0
04.01.2014	0
05.01.2014	0
06.01.2014	0
07.01.2014	0
08.01.2014	1
09.01.2014	1
10.01.2014	1
11.01.2014	0
12.01.2014	0
13.01.2014	1
14.01.2014	1
15.01.2014	1
16.01.2014	1
17.01.2014	1
18.01.2014	0
19.01.2014	0
20.01.2014	1
21.01.2014	1
22.01.2014	1
23.01.2014	1

Дата	Значение
24.01.2014	1
25.01.2014	0
26.01.2014	0
27.01.2014	1
28.01.2014	1
29.01.2014	1
30.01.2014	1
31.01.2014	1

Если регистр расчета будет связан с таким регистром-календарем, то он сможет получить количество рабочих дней за заданный период, отобразив записи с датами, входящими в указанный период и сложив цифры в ресурсе "Значение". Получится количество рабочих дней в указанном периоде.

Вот так (пока – в простейшем виде) устроен календарь рабочего времени в расчетных механизмах платформы 1С 8.

Мы должны внести в свойство "График" имя регистра сведений "ГрафикиРаботы", выбрав его из списка.

В свойство "Значение" графика нужно занести имя ресурса, значения которого будут суммироваться для подсчета рабочего времени, а в свойство "Дата графика" нужно занести имя измерения, в котором находятся календарные даты. Оба имени выбираются из списков.

Таким образом, значения этих трех свойств будут такими:

График: *ГрафикиРаботы*

Значение графика: *Значение*

Дата графика: *Дата*

(В регистре сведений "ГрафикиРаботы" мы можем заметить еще одно измерение: "ГрафикРаботы". О его назначении мы поговорим позже).

Свойство "Базовый период". Его назначение аналогично свойству "Период действия", только касается функционала получения суммы базы. Если флажок этого свойства установить, то регистр расчета будет обладать функционалом получения суммы базы, если не устанавливать, то такого функционала в регистре не будет, а попытка обратиться к этой возможности регистра расчета вызовет программную ошибку.

Установим флажок свойства "Базовый период", потому что в нашем регистре будут рассчитываться результаты вида расчета "Доплата за квалификацию", который зависит от базы.

Свойство "Периодичность". Здесь можно выбрать из списка значение периодичности регистра расчета. Месячные начисления будут рассчитываться в регистре с периодичностью "Месяц", а, например, квартальные премии можно рассчитывать в другом регистре расчета – с периодичностью "Квартал".

Оставим для нашего регистра значение этого свойства по умолчанию – "Месяц".

Итак, после того, как мы сделали все описанные настройки, закладка "Основные" редактора объекта конфигурации примет следующий вид:

6.3.3. Создание прикладных полей регистра расчета "ОсновныеНачисления"

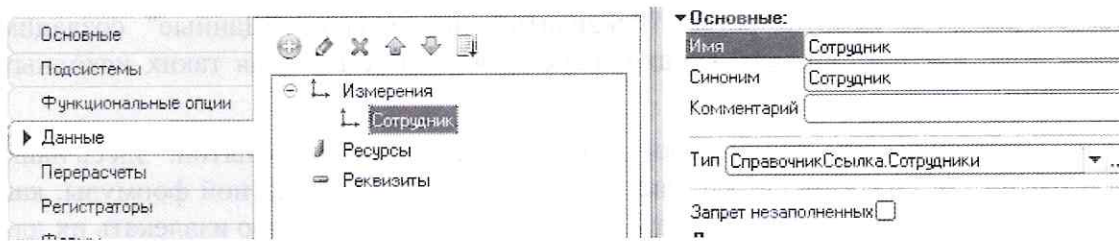
Вспомним, что прикладными полями регистра, в отличие от стандартных полей, называются те, которые создает программист-разработчик: измерения, ресурсы и реквизиты.

Измерения

Продумаем состав измерений нашего регистра. Измерения любого регистра 1С – это ключевые поля, относительно (или, как говорят, в разрезе) значений которых хранится информация в регистре.

Мы уже вскользь упоминали, что регистр расчета должен иметь как минимум одно измерение, значение которого будет указывать лицо, для которого рассчитывается данная строка регистра.

Перейдем на закладку данные редактора объекта метаданных. В ветвь "Измерения" добавим новое измерение, назовем его "Сотрудник". Тип измерения зададим следующий: <СправочникСсылка.Сотрудники> – в нашей каркасной конфигурации есть справочник "Сотрудники", элементы которого представляют информацию о физических лицах, получающих зарплату:



Итак, минимальный состав измерений регистра расчета мы обеспечили.

Теперь подумаем о возможности внутреннего совместительства, которую предоставляет в настоящее время Трудовой кодекс. Один и тот же сотрудник может работать на нескольких должностях или исполнять несколько профессиональных обязанностей, постоянно или в течение какого-то периода. Например, Иванов работал целый месяц в должности менеджера, а с 10 по 25 число месяца также исполнял обязанности консультанта. Следовательно, необходимо иметь возможность рассчитывать зарплату в разрезе "Сотрудник : Должность."

Для этого создадим еще одно измерение - "Должность", а тип ему установим такой: <СправочникСсылка.Должности> – в каркасной конфигурации имеется справочник "Должности", в который пользователь заносит должности и обязанности сотрудников предприятия.

Пока нам достаточно двух измерений регистра расчета.

Ресурсы

Ресурсы – это целевые поля регистра. В них хранится информация, относящаяся к комбинациям значений измерений, а также функционал регистра получает на основании значений ресурсов получает результирующие показатели.

Нам достаточно одного ресурса, в котором будет храниться результат расчета по данной строке регистра.

В ветвь "Ресурсы" на закладке "Данные" добавим новый ресурс и назовем его "Результат".

Ресурсы регистра расчета всегда имеют тип <Число>. Только давайте установим точность этого числа 2 (два знака после запятой), поскольку денежные величины принято хранить и отображать "с копейками".

Таким образом, в регистре расчета у нас будет один ресурс – "Результат".

Реквизиты

Реквизиты – это вспомогательные поля регистра, которые в общем случае не участвуют в работе функционала регистра. Поэтому в реквизитах обычно хранится информация, которая должна быть "под рукой", когда прикладная программа оперирует с конкретной записью регистра.

Реквизит "Размер"

Если проанализировать расчетные формулы (способы расчета), то мы обнаружим, что в формуле чаще всего имеется какая-то "опорная" числовая величина, смысл и содержание которой для разных формул различны. Так, для формулы расчета оклада это месячная ставка оклада, для формулы премии это процент премии, который берется от суммы расчетной базы. Вот это значение удобно хранить прямо в записи регистра расчета для удобства подстановки в расчетную формулу.

Для этой цели в ветви "Реквизиты" на закладке "Данные" создадим реквизит с именем "Размер" (подходящее обобщенное имя для таких исходных значений).

Установим ему тип <Число> и точность 2 знака после запятой. Здесь наша программа и будет помещать такие "опорные" значения расчетной формулы, как размер оклада, процент премии и т.п., чтобы можно было быстро извлекать их для подстановки в формулу.

Реквизит "ГрафикРаботы"

Настало время снова вернуться к вопросу получения количества рабочего времени.

Когда мы обсуждали назначение регистра-календаря и связь регистра расчета с календарем, мы намеренно остановились на простейшей схеме устройства календаря: каждой календарной дате у нас соответствовало только одно значение длительности рабочего времени, 1 для рабочего дня, 0 для выходного или праздничного.

Такая простейшая схема устанавливает жесткое ограничение на т.н. график работы предприятия и сотрудников. Так, если в рабочей неделе два выходных – суббота и воскресенье, то график называется пятидневкой (по числу рабочих дней в неделе). Если суббота является рабочим днем, то график называется шестидневкой. А еще рабочее время может исчисляться не в днях, а в часах... Вариантов графиков работы может быть достаточно много, они могут быть разными для отдельных сотрудников в разное время и для разных видов расчета.

Как платформа 1С позволяет нам обеспечить получение количества рабочего времени при наличии множества графиков? - Для этого платформа предлагает следующий подход.

В регистр-календарь добавляется еще одно измерение – "ГрафикРаботы", которое указывает вариант графика, к которому относится данная строка регистра-календаря: "Пятидневка", "Шестидневка", "Почасовка" и т.д. В этом случае для одной календарной даты в регистре-календаре будет уже не одна запись, а несколько – по числу используемых графиков работы. Например, для января 2014г. записи регистра-календаря могут быть такими:

Дата	ГрафикРаботы	Значение
01.01.2014	Пятидневка	0
01.01.2014	Шестидневка	0
02.01.2014	Пятидневка	0
02.01.2014	Шестидневка	0
03.01.2014	Пятидневка	0
03.01.2014	Шестидневка	0
04.01.2014	Пятидневка	0
04.01.2014	Шестидневка	0
05.01.2014	Пятидневка	0
05.01.2014	Шестидневка	0
06.01.2014	Пятидневка	0
06.01.2014	Шестидневка	0
07.01.2014	Пятидневка	0
07.01.2014	Шестидневка	0
08.01.2014	Пятидневка	1

Дата	ГрафикРаботы	Значение
08.01.2014	Шестидневка	1
09.01.2014	Пятидневка	1
09.01.2014	Шестидневка	1
10.01.2014	Пятидневка	1
10.01.2014	Шестидневка	1
11.01.2014	Пятидневка	0
11.01.2014	Шестидневка	1
12.01.2014	Пятидневка	0
12.01.2014	Шестидневка	0
13.01.2014	Пятидневка	1
13.01.2014	Шестидневка	1
14.01.2014	Пятидневка	1
14.01.2014	Шестидневка	1
.....
27.01.2014	Пятидневка	1
27.01.2014	Шестидневка	1
28.01.2014	Пятидневка	1
28.01.2014	Шестидневка	1
29.01.2014	Пятидневка	1
29.01.2014	Шестидневка	1
30.01.2014	Пятидневка	1
30.01.2014	Шестидневка	1
31.01.2014	Пятидневка	1
31.01.2014	Шестидневка	1

Мы видим, что для даты 11.01.2014 (это суббота) по графику "Пятидневка" длительность рабочего времени равна 0 (выходной, а по графику "Шестидневка" она равна 1 (рабочий день).

Становится понятно, как должен работать механизм получения количества рабочего времени регистра расчета. Кроме интервала, этому механизму нужно передавать еще значение графика работы – тогда механизм отфильтрует в указанном интервале записи с этим значением графика и просуммирует ресурс "Значение" только для них.

Важно!

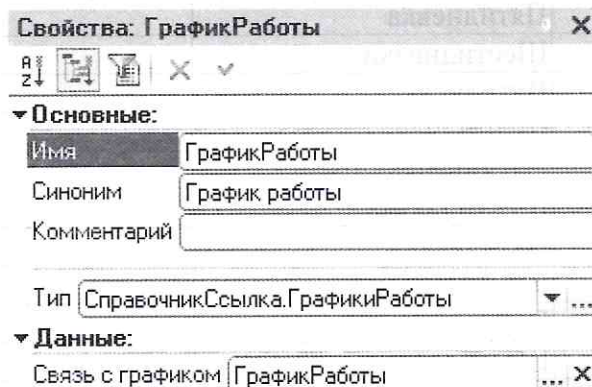
Для того чтобы можно было передать значение графика работы механизму расчета количества рабочего времени, в регистре расчета создается реквизит со специальной настройкой.

Мы уже упоминали, что реквизиты регистра расчета, в отличие от реквизитов других трех регистров, иногда принимают участие в работе функционала регистра. Как раз это тот самый случай.

Итак, создадим еще один реквизит регистра "ОсновныеНачисления", назовем его "ГрафикРаботы". Тип реквизита установим: <СправочникСсылка.ГрафикиРаботы> – в каркасной конфигурации есть справочник "ГрафикиРаботы", каждый элемент соответствует отдельному графику работы.

А чтобы реквизит мог служить для передачи значения графика механизму регистра расчета, установим значение свойства этого реквизита, которое

называется "Связь с графиком". В этом свойстве надо указать имя измерения регистра-календаря, в котором содержатся значения графиков. Нажмем на кнопку выбора и из списка измерений регистра "ГрафикиРаботы" выберем имя измерения "ГрафикРаботы":



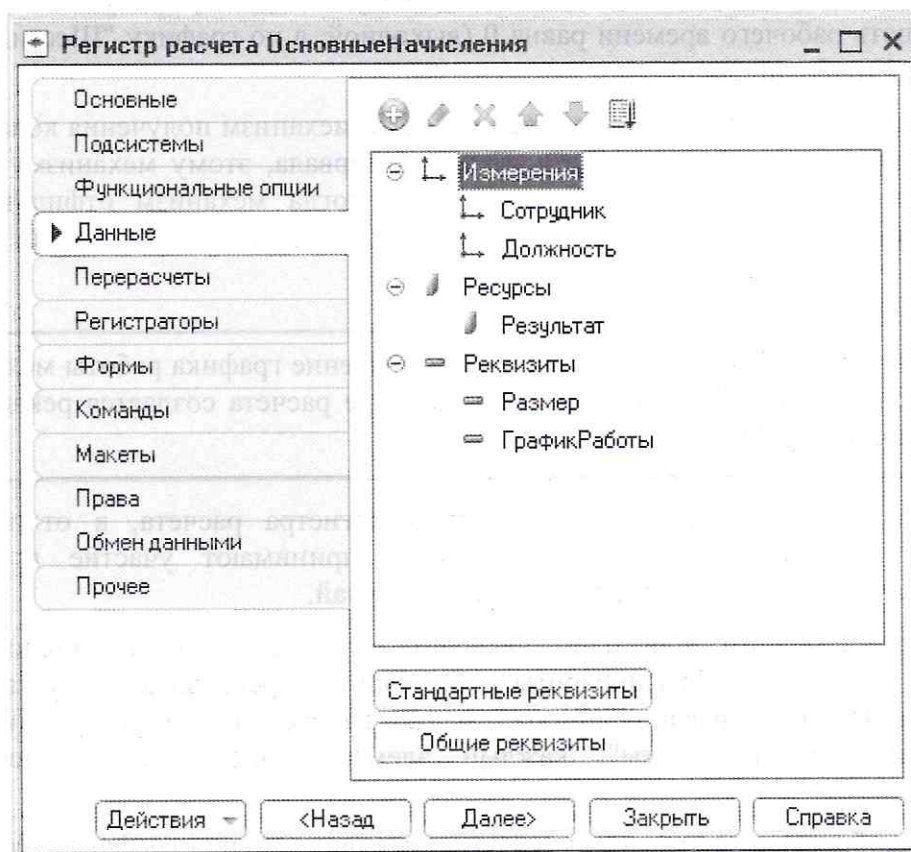
Важно!

Понятие "Данные графика"

После того, как мы ввели понятие графика работы и настроили регистр расчета для работы по нескольким графикам, введем для количества планового рабочего времени (получаемого по периоду действия) и количества фактически отработанного времени (получаемого по фактическому периоду действия) обобщающее определение – **"Данные графика"**.

Этот термин мы и будем использовать в дальнейшем.

Итак, мы создали структуру прикладных полей регистра расчета "ОсновныеНачисления". Закладка "Данные" теперь выглядит так:

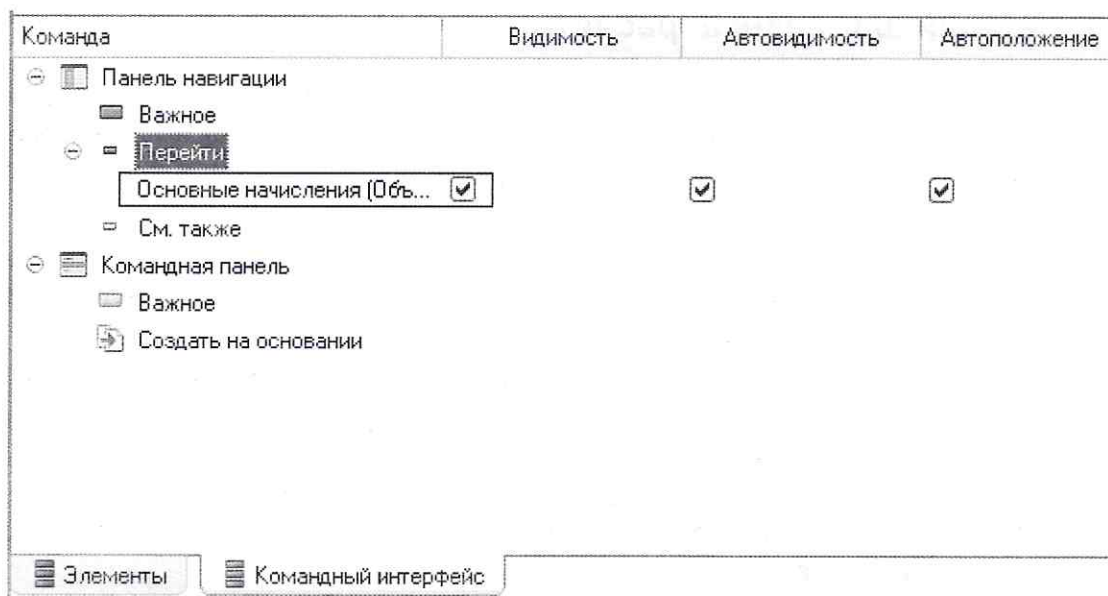


6.3.4. Назначение регистратора регистру расчета

В каркасной конфигурации у нас имеется "заготовка" расчетного документа – документ "РасчетЗарплаты". Мы будем в дальнейшем дорабатывать этот документ, а сейчас назначим его регистратором для нашего регистра расчета, как и должно быть по нашим замыслам.

А сейчас перейдем на закладку "Регистраторы" и установим флажок возле документа "НачислениеЗарплаты" – т.е. назначим этот документ регистратором для нашего регистра.

После этого для удобства надо обеспечить возможность просмотра наборов записей документа. Для этого нужно открыть редактор формы документа, перейти на закладку "Командный интерфейс", раскрыть группу "Перейти" и установить флажок "Видимость" в строке нашего регистра расчета - "Основные начисления", как показано на рисунке:



Теперь в режиме "1С:Предприятие" в форме документа, будет появляться надпись "Основные начисления". Если на нее нажать мышью, то появится форма списка регистра, в которой будет отражен набор записей данного документа.

Обновим конфигурацию базы данных.

Конфигурирование плана видов расчета и регистра расчета окончено.

7. Шаги (этапы) сеанса расчета зарплаты

Под сеансом расчета зарплаты мы будем понимать действия программного обеспечения от "нажатия кнопки" пользователем до записи в информационную базу результатов расчета.

Важно!

Желательно постараться как можно глубже усвоить себе понятия, которые излагаются в этом разделе. От этого зависит успешность освоения всей практической работы с расчетными механизмами платформы 1С 8.

Попробуем путем рассуждений установить, что должна сделать программа за время сеанса расчета.

7.1. Первый шаг сеанса расчета

1. Мы знаем, что расчетные формулы сами по себе весьма просты, но, чтобы их применить, предварительно нужно иметь необходимые данные, которые являются операндами формул: сумму базы и/или количество рабочего времени.
2. Для получения необходимых данных нужно обратиться к регистру расчета через специальный интерфейс, изучение которого, собственно, и составляет предмет нашего курса.
3. Чтобы сформировать и выдать нашей программе необходимые данные, регистр расчета должен иметь значения исходных данных – таких, как границы периода действия, границы базового периода, значение графика работы, на основании которых он и формирует необходимые данные. Например, количество отработанного времени рассчитывается в интервале периода действия.
4. Функционал регистра расчета может получить исходные данные только из полей самих записей регистра расчета. Для этого и служат стандартные поля записи регистра расчета: "ПериодДействияНачало", "ПериодДействияКонец" и т.д. Также для целей передачи значения графика работы мы создали реквизит "ГрафикРаботы" и указали в Конфигураторе, что этот реквизит содержит значение графика работы.

А из этой цепочки рассуждений выводится одно важнейшее следствие.

Важно!

Прежде, чем обратиться к регистру расчета для получения **необходимых данных**, в регистр расчета должны быть записаны подлежащие расчету строки с **исходными данными**.

Это и есть первый шаг сеанса расчета зарплаты: **создание записей регистра расчета со стандартными полями, содержащими исходные данные**.

7.2. Второй шаг сеанса расчета

После того, как выполнен первый шаг, следующим шагом будет получение необходимых данных. Программа обращается к **функционалу регистра расчета**. Забегая вперед, скажем, что это делается путем выполнения запросов к виртуальным таблицам регистра расчета.

По окончании работы функционала регистра расчета программа получает в свое распоряжение **необходимые данные** в виде результата запроса (т.е. табличной сущности).

Состав запрашиваемых необходимых данных – данные графика, сумма базы или же то и другое – определяется составом способов расчета (расчетных формул), которые соответствуют видам расчета в записях регистра, созданных на первом шаге сеанса расчета.

Необходимые данные будут получены для каждой записи регистра расчета, которая была создана в результате выполнения первого шага сеанса расчета – причем таким образом, что **программа сможет в процессе расчета результатов для каждой записи регистра**, которая рассчитывается, **найти соответствующую ей запись в полученной из запроса таблице необходимых данных**.

Что для этого нужно, мы рассмотрим подробно в соответствующих параграфах нашего курса.

7.3. Третий шаг сеанса расчета

Третий шаг сеанса расчета состоит в последовательном обходе записей регистра расчета и применении расчетной формулы, соответствующей каждой записи. Для каждой записи устанавливается соответствие со строкой таблицы необходимых данных, полученной на втором шаге, из этой строки таблицы необходимых данных извлекаются значения необходимых данных и подставляются в выражение расчетной формулы.

Полученный результат (рассчитанная сумма) помещается в ресурс записи регистра расчета (мы назвали его "Результат"), после этого записи с рассчитанными результатами снова записываются в информационную базу. Таким образом, на **третьем шаге мы получаем уже рассчитанные суммы**.

7.4. Многократность выполнения второго и третьего шага сеанса расчета

Если бы не было зависимости по базе одних видов расчета от других, то второй и третий шаги сеанса расчета достаточно было бы выполнить по одному разу. Но, поскольку практически всегда имеются виды расчета, рассчитываемые по базе, то второй и третий шаги выполняются не один раз, а по числу категорий расчета.

Например, рассчитываются несколько видов расчета с категориями "Первичное", "Зависимое первого уровня", "Зависимое второго уровня". Напомним, что это означает: виды расчета с категорией "Зависимое первого уровня" зависят по базе от видов расчета с категорией "Первичное", а виды

расчета с категорией "Зависимое второго уровня" зависят по базе от видов расчета с категорией "Зависимое первого уровня" (и, возможно, от "первичных" тоже),

Следовательно, вначале нужно рассчитать и записать в регистр расчета результаты видов расчета с категорией "Первичное", тогда можно будет получить необходимые данные - суммы базы - для категории "Зависимое первого уровня", рассчитать и записать уже их. Когда результаты видов расчета с категорией "Зависимое первого уровня" будут рассчитаны и записаны в регистр, то можно получить необходимые данные для видов расчета с категорией "Зависимое второго уровня", рассчитать и записать их – и так далее, по числу категорий расчета.

А первый шаг сеанса расчета выполняется только один раз – потому что исходные данные, которые записываются при его выполнении, никак в процессе расчета не изменяются.

Важно!

Таким образом, **первый шаг сеанса расчета выполняется один раз, а второй и третий шаги – по числу категорий расчета, которые участвуют в расчете.**

8. Реализация расчетных алгоритмов сеанса расчета регистра "Основные Начисления"

Мы приступаем к реализации программы расчета зарплаты с использованием плана видов расчета и регистра расчета "Основные Начисления".

В каркасной конфигурации имеется документ "Расчет Зарплаты", который мы назначили регистратором для регистра расчета "Основные Начисления". Пока в структуре документа почти ничего нет, кроме реквизита "Период Регистрации" типа <Дата> и пустой процедуры "ОбработкаПроведения" в модуле объекта.

Этот документ будет у нас основной "рабочей лошадью" сеанса расчета – в модуле объекта его мы и будем писать расчетные программы на встроенном языке.

Пользователь будет создавать экземпляр этого документа и вводить в его форме все исходные данные сеанса расчета: виды расчета для сотрудников, период действия, базовый период и т.д. Затем пользователь будет проводить документ, и при проведении документа будет выполняться весь сеанс расчета. Процедура "ОбработкаПроведения" больше всего подходит для этого, поскольку при проведении как раз и осуществляется создание записей в регистре.

8.1. Конфигурирование структуры документа "Расчет Зарплаты"

8.1.1. Создание табличной части "Основные Начисления"

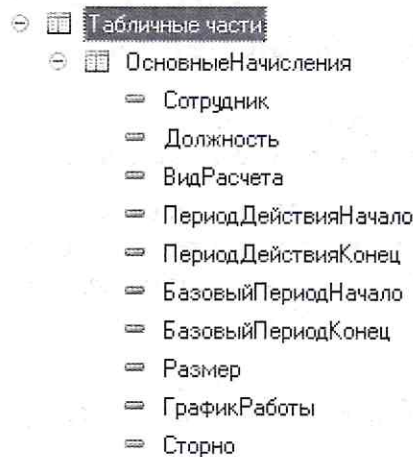
Для целей ввода исходных данных по расчету зарплаты удобно создать в документе табличную часть, каждая строка которой будет содержать исходные данные для одного вида расчета.

Откроем редактор метаданных документа "Расчет Зарплаты", на закладке "Данные" добавим новую табличную часть с именем "Основные Начисления". Создадим реквизиты табличной части в следующем составе:

Имя реквизита	Тип
Сотрудник	<СправочникСсылка.Сотрудники>
Должность	<СправочникСсылка.Должности>
ВидРасчета	<ПланВидовРасчетаСсылка.ОсновныеНачисления>
ПериодДействияНачало	<Дата>
ПериодДействияКонец	<Дата>
БазовыйПериодНачало	<Дата>
БазовыйПериодКонец	<Дата>
Размер	<Число (10, 2)>
ГрафикРаботы	<СправочникСсылка.ГрафикиРаботы>
Сторно	<Булево>

Как можно видеть, в каждой строке табличной части можно описать вид расчета со всеми исходными данными.

Структура табличной части на закладке "Данные" будет выглядеть так:

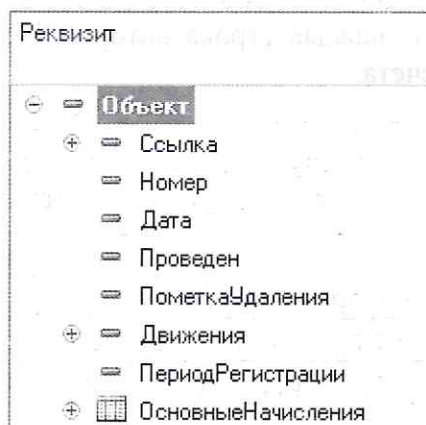


8.1.2. Создание элемента формы для отображения табличной части

Поскольку эту табличную часть будет заполнять пользователь, нужно создать элемент формы для нее – табличное поле.

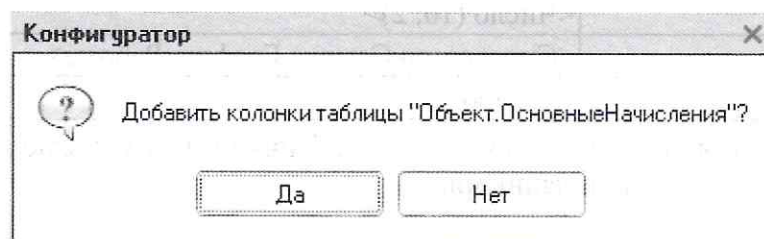
Откроем форму документа для редактирования. В списке элементов формы мы видим предварительно созданные группы: "ЗакладкаОсновныеНачисления", "ЗакладкаПремии", "ЗакладкаУдержания". Эти группы отображаются на форме как закладки, но, поскольку в этих группах еще нет элементов, то группы пока на форме в режиме пользователя невидимы.

На закладке "Реквизиты" редактора форм (справа вверху) развернем структуру основного реквизита формы – "Объект":

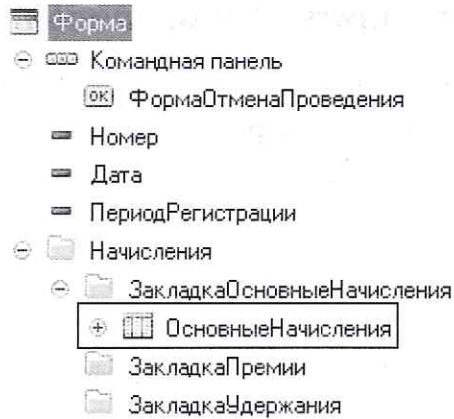


и выполним перетаскивание элемента "ОсновныеНачисления" налево, на группу "ОсновныеНачисления".

Будет задан вопрос:



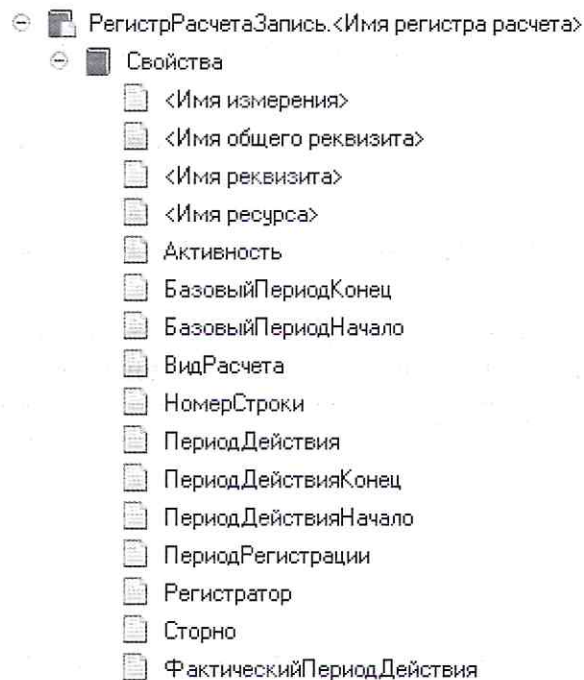
Нужно ответить "Да", и в составе группы "ОсновныеНачисления" формы документа появится табличное поле "ОсновныеНачисления":



Теперь после обновления конфигурации форма документа будет содержать табличное поле для заполнения табличной части "Основные начисления".

8.2. Краткое изложение принципов программной работы с регистром расчета

Работа с регистром расчета на встроенном языке в основном аналогична работе с другими классами регистров. Для каждого регистра расчета можно получить объект типа "РегистрРасчетаНаборЗаписей", через который осуществляется создание, редактирование и удаление записей регистра. Для добавления новой записи (объект типа "РегистрРасчетаЗапись") имеется метод "Добавить", после чего производится заполнение полей новой записи соответствующими значениями. К полям записи регистра расчета обращаются через одноименные свойства объекта "РегистрРасчетаЗапись", состав которых можно посмотреть в Синтакс-помощнике:



Запись набора записей регистра расчета выполняется, как и для других регистров, путем вызова метода "Записать" объекта "РегистрРасчетаНаборЗаписей".

8.3. Процедура "ЗаполнитьНаборЗаписей"

В нашей каркасной конфигурации есть общий модуль "Расчеты", в котором, в частности, имеется процедура "ЗаполнитьНаборЗаписей":

```
Процедура ЗаполнитьНаборЗаписей(ТекстЗапроса, ДокументСсылка,
НаборЗаписей) Экспорт
    Запрос = Новый Запрос(ТекстЗапроса);
    Запрос.УстановитьПараметр("Ссылка", ДокументСсылка);
    Выборка = Запрос.Выполнить().Выбрать();
    Пока Выборка.Следующий() Цикл
        НоваяЗапись = НаборЗаписей.Добавить();
        ЗаполнитьЗначенияСвойств(НоваяЗапись, Выборка);
    КонецЦикла;
КонецПроцедуры
```

Эта процедура предназначена для выполнения действий первого шага сеанса расчета – создания записей с исходными данными, которые одинаковы для всех типов расчетных документов.

В качестве аргументов процедуре передается текст запроса, который получит исходные данные из документа, ссылка на документ, которая нужна для установки параметра запроса, и сам набор записей регистра расчета, в котором записи будут созданы.

Код процедуры очень прост. Процедура выполняет переданный запрос, получает выборку результата запроса, в цикле создает новую запись для каждой позиции выборки и заполняет ее поля (свойства) с использованием процедуры глобального контекста платформы "ЗаполнитьЗначенияСвойств".

8.4. Программный код первого шага сеанса расчета

Итак, мы хотим в процессе проведения документа получить исходные данные для расчета из структуры документа и создать на их основе набор записей регистра расчета.

Для этого откроем модуль объекта документа "РасчетЗарплаты" и в теле процедуры "ОбработкаПроведения" напишем следующий программный код:

```
Процедура ОбработкаПроведения(Отказ, РежимПроведения)
    НаборОсновныеНачисления = Движения.ОсновныеНачисления;
    ТекстЗапроса =
        "ВЫБРАТЬ
        |
        | РасчетЗарплатыОсновныеНачисления.Ссылка.ПериодРегистрации,
        |   РасчетЗарплатыОсновныеНачисления.Сотрудник,
        |   РасчетЗарплатыОсновныеНачисления.Должность,
        |   РасчетЗарплатыОсновныеНачисления.ВидРасчета,
```

```
|
РасчетЗарплатыОсновныеНачисления.ПериодДействияНачало,
|
КОНЕЦПЕРИОДА (РасчетЗарплатыОсновныеНачисления.ПериодДействи
яКонец, ДЕНЬ) КАК ПериодДействияКонец,
|
РасчетЗарплатыОсновныеНачисления.БазовыйПериодНачало,
|
ВЫБОР
|
КОГДА
РасчетЗарплатыОсновныеНачисления.БазовыйПериодКонец =
ДАТАВРЕМЯ(1, 1, 1)
|
ТОГДА ДАТАВРЕМЯ(1, 1, 1)
|
ИНАЧЕ
КОНЕЦПЕРИОДА (РасчетЗарплатыОсновныеНачисления.БазовыйПериодКонец
, ДЕНЬ)
|
КОНЕЦ КАК БазовыйПериодКонец,
|
РасчетЗарплатыОсновныеНачисления.Размер,
|
РасчетЗарплатыОсновныеНачисления.ГрафикРаботы,
|
РасчетЗарплатыОсновныеНачисления.Сторно
|ИЗ
|
Документ.РасчетЗарплаты.ОсновныеНачисления КАК
РасчетЗарплатыОсновныеНачисления
|ГДЕ
|
РасчетЗарплатыОсновныеНачисления.Ссылка = &Ссылка
|
|УПОРЯДОЧИТЬ ПО
|
РасчетЗарплатыОсновныеНачисления.НомерСтроки";
Расчеты.ЗаполнитьНаборЗаписей(ТекстЗапроса, Ссылка,
НаборОсновныеНачисления);
НаборОсновныеНачисления.Записать(, Ложь);
КонецПроцедуры
```

(Текст запроса, понятно, легче построить при помощи Конструктора запросов).

Программный код несложен, и в нем легко может разобраться даже начинающий программист.

Как видно, формируется текст запроса для выборки данных, на основании которых будет заполняться набор записей. Состав полей запроса, как можно видеть, содержит все исходные данные, которые требуются для заполнения записи регистра расчета.

Обратим только внимание, как в запросе обрабатываются поля исходной табличной части "ПериодДействияКонец" и "БазовыйПериодКонец": при помощи функции КОНЕЦПЕРИОДА языка запросов они приравниваются к концу дня (устанавливается время 23:59:59), чтобы включить в состав соответствующего периода последний день полностью.

Также обратим внимание на оператор ВЫБОР, при помощи которого формируется значение поля "БазовыйПериодКонец": если это поле не заполнено, то приведение его значения к концу дня не выполняется.

Еще обратим внимание на операцию УПОРЯДОЧИТЬ ПО в запросе: результат запроса упорядочивается по номеру строки табличной части. Это приводит к тому, что порядок следования строк табличной части и записей в регистре расчета будет одинаковый. Так удобнее визуальнo сверять данные табличной части и регистра.

Далее вызывается процедура "ЗаполнитьНаборЗаписей" общего модуля "Расчеты", которую мы обсуждали выше, она создает записи регистра расчета и заполняет их поля.

В завершение происходит запись набора записей регистра в информационную базу – вызывается метод "Записать" объекта "РегистрРасчетаНаборЗаписей". И вот здесь надо обратить особое внимание на одно важное отличие работы метода "Записать" набора записей регистра расчета от работы одноименного метода наборов записей других трех классов регистров платформы "1С:Предприятие 8".

8.5. Особенность метода "Записать" набора записей регистра расчета

Метод "Записать" регистров накопления, бухгалтерии и сведений имеет один аргумент, который называется "Замещать" и имеет тип <Булево>. Он указывает, следует ли заместить прежний набор записей новыми записями, или же новые записи нужно дописывать в конец прежних. По умолчанию его значение равно Истина.

Метод "Записать" набора записей регистра расчета тоже имеет этот аргумент, его назначение и значение по умолчанию – такие же, как и для других классов регистров, поэтому в нашем коде мы используем значение по умолчанию – Истина.

Но в методе "Записать" набора записей регистра расчета используется также второй аргумент, которого нет у метода "Записать" других регистров. Этот аргумент называется "ТолькоЗапись" и тоже имеет тип <Булево>. Его назначение – указать, будет ли набор записей записан без каких-либо дополнительных действий (значение Истина), или же при записи будет также определены состав и границы интервалов фактического периода действия записей набора (значение Ложь).

Очевидно, при выполнении первого шага сеанса расчета необходимо, чтобы этот второй аргумент имел значение Ложь. Ведь после шага записи исходных данных будет выполняться второй шаг сеанса расчета, на котором программа будет обращаться к регистру расчета для получения данных о фактически отработанном времени. А чтобы эти данные были актуальны, предварительно должен быть получен полный состав интервалов фактического периода действия, соответствующий составу вытесняющих видов расчета, введенных в нашем документе.

Поэтому при выполнении первого шага сеанса расчета мы должны задать значение Ложь для второго аргумента метода "Записать".

А когда будет выполняться третий шаг, т.е. запись набора с уже полученными результатами расчета? Тогда, очевидно, нет никакой необходимости рассчитывать состав и границы интервалов фактического периода действия – ведь период действия в процессе расчета не изменяется, следовательно, не может измениться и фактический период действия.

Поэтому при записи набора с готовыми результатами расчета метод "Записать" вызывается с вторым аргументом, равным Истина – происходит просто запись набора без каких-либо дополнительных действий.

Важно!

Для выполнения первого шага сеанса расчета метод "Записать" набора записей регистра расчета вызывается со вторым аргументом, равным Ложь.

Для выполнения третьего шага сеанса расчета метод "Записать" набора записей регистра расчета вызывается со вторым аргументом, равным Истина.

Вызов метода "Записать" со вторым аргументом Ложь на третьем шаге сеанса расчета, конечно же, не может отрицательно повлиять на результат расчета – просто будет лишний раз пересчитан фактический период действия... На самом деле, хотя это и так, настоятельно рекомендуется вызывать метод "Записать" на третьем шаге со вторым аргументом Истина.

Важно!

Дело в том, что процедура пересчета фактического периода действия довольно ресурсоемка и длительна, поэтому в интересах повышения производительности расчетной системы крайне желательно ее не выполнять тогда, когда этого не требуется.

А значение по умолчанию второго аргумента – Ложь, так что легко ошибиться, машинально закрыв скобку. Поэтому лучше всегда помнить о важности значения второго аргумента, для этого лучше даже не пользоваться значением Ложь по умолчанию на первом шаге, а указывать это значение в явном виде, как сделано у нас.

На этом разработка кода для выполнения первого шага сеанса расчета заканчивается. Код первого шага довольно прост – это просто перезапись данных документа в набор записей регистра расчета и запись набора в информационную базу.

Теперь следует обновить конфигурацию базы данных (актуализировать сделанные нами доработки конфигурации) и запустить конфигурацию в режиме "1С:Предприятие" для создания документа "Расчет зарплаты" в режиме пользователя.

8.6. Создание экземпляра документа "Расчет зарплаты" и ввод в документ вида расчета "Оклад"

Запустив нашу конфигурацию в режиме пользователя, перейдем в раздел "Расчет" и откроем форму списка документа "Расчет зарплаты". У нас пока нет ни одного экземпляра этого документа. Нажмем кнопку "Создать" формы списка документа и в открывшейся форме нового документа введем следующие данные.

Поле "Период регистрации": 01.01.2014 (совершенно не обязательно вводить именно этот период, можно вводить первое число любого месяца).

Табличное поле "Основные начисления" заполним следующими данными:

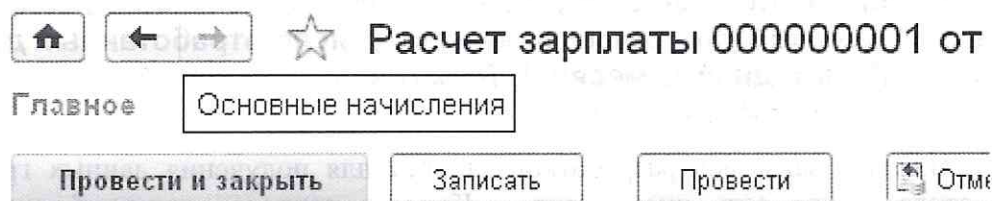
№	Сотрудник	Должность	Вид расчета	Период действия начало	Период действия конец	Базовый период начало	Базовый период конец	Размер	График работы	Сторно
1	Ковалев Александр Петрович	Директор	Оклад	01.01.2014	31.01.2014			70 000,00	Пятидневка	Нет
2	Троицкая Татьяна Сергеевна	Бухгалтер	Оклад	01.01.2014	31.01.2014			60 000,00	Пятидневка	Нет
3	Ильин Сергей Викторович	Менеджер	Оклад	01.01.2014	31.01.2014			50 000,00	Пятидневка	Нет

1. Значения полей "Сотрудник" и "Должность" каждой строки табличного поля выбираем из справочников – соответственно "Сотрудники" и "Должности".
2. Значение поля "Вид расчета" - "Оклад" - выбираем из плана видов расчета "Основные начисления".
3. Границы базового периода оставляем пустыми – вид расчета "Оклад" не зависит от базы.
4. Значение поля "График работы" выбираем из справочника "Графики работы".
5. Поле "Сторно" оставляем пустым (по умолчанию – Нет, т.е. Ложь).

8.7. Проведение документа – запись исходных данных в регистр расчета

После того, как мы выполнили ввод данных в форму документа, нажмем кнопку Провести на форме. Если мы не сделали ошибок конфигурирования и программирования, то документ будет проведен и в регистре расчета Основные начисления появятся записи.

Чтобы их посмотреть, можно нажать на надпись (кнопку) Основные начисления сверху формы документа:



(Вспомним, как настраивается возможность просмотра набора записей по регистрам: в редакторе формы документа, на закладке "Командный интерфейс", в группе "Перейти").

Появится форма списка регистра "Основные начисления", в которой будет показан набор записей нашего документа.

Мы видим, что наш код первого шага сеанса расчета выполнил свою задачу: В регистре расчета появились записи – по одной на каждую строчку заполненной табличной части. Поля записей регистра имеют соответствующие значения.

А поле "Результат" пока нулевое (пустое), и это правильно – ведь мы еще не реализовали расчетный код, код второго и третьего шага сеанса расчета.

8.8. Программный код второго шага сеанса расчета регистра "Основные Начисления"

Итак, мы имеем все необходимые условия для того, чтобы обратиться к функционалу регистра расчета и получить недостающие данные для подстановки в расчетную формулу оклада – необходимые данные. Вспомним, формулу для расчета оклада:

$$\text{[Сумма оклада]} = \text{[Ставка оклада]} * \text{[Число отработанных дней]} / \text{[Число рабочих дней в месяце]}$$

Из операндов этой формулы нам известна только [Ставка оклада] – это исходное данное, оно находится в поле реквизита Размер записей регистра расчета. Следовательно, [Число отработанных дней] и [Число рабочих дней в месяце] – суть необходимые данные, которые нам надлежит получить для каждой записи регистра с видом расчета Оклад. Вот эти необходимые данные нам и нужно получить от регистра расчета.

Важно! _____

Мы приступаем к самой сути нашего курса – начинаем изучать обращение к функционалу регистра расчета.

8.8.1. Виртуальная таблица "ДанныеГрафика" регистра расчета. Расчет начислений со способом расчета "ПоОкладу"

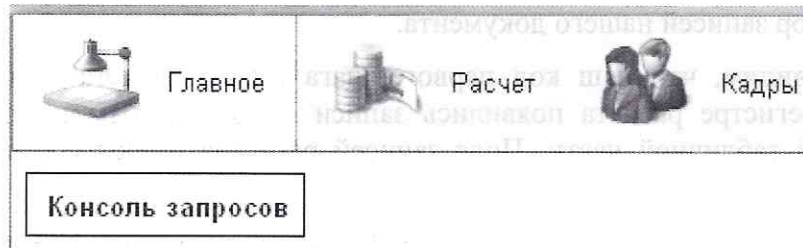
Мы уже упоминали, что самый эффективный способ обратиться к функционалу любого класса регистров 1С – накопления, бухгалтерии, расчета, сведений – это использовать возможности языка запросов, обращаясь в запросах к виртуальным таблицам регистров.

Поэтому без долгих рассуждений займемся изучением первой из виртуальных таблиц регистра расчета – виртуальной таблицы "ДанныеГрафика".

Вспомним, что данные графика – это обобщающее название, которое мы ввели для двух операндов расчетной формулы: **[Число отработанных дней]** и **[Число рабочих дней в месяце]**, Поскольку они должны быть получены с учетом заданного графика работы.

Вначале займемся разработкой запроса для получения данных графика. Для этого у нас есть инструмент – Консоль запросов, который позволяет разрабатывать и отлаживать запросы в режиме "1С:Предприятие".

Запустим нашу конфигурацию в режиме "1С:Предприятие", в главном разделе (по-другому он называется "рабочий стол") нажмем кнопку "Консоль запросов", которая располагается сразу же под закладками с именами разделов:



Откроется обработка Консоль запросов.

Разработка запроса для получения данных графика

Запрос, который мы построим для получения данных графика, будет таким:

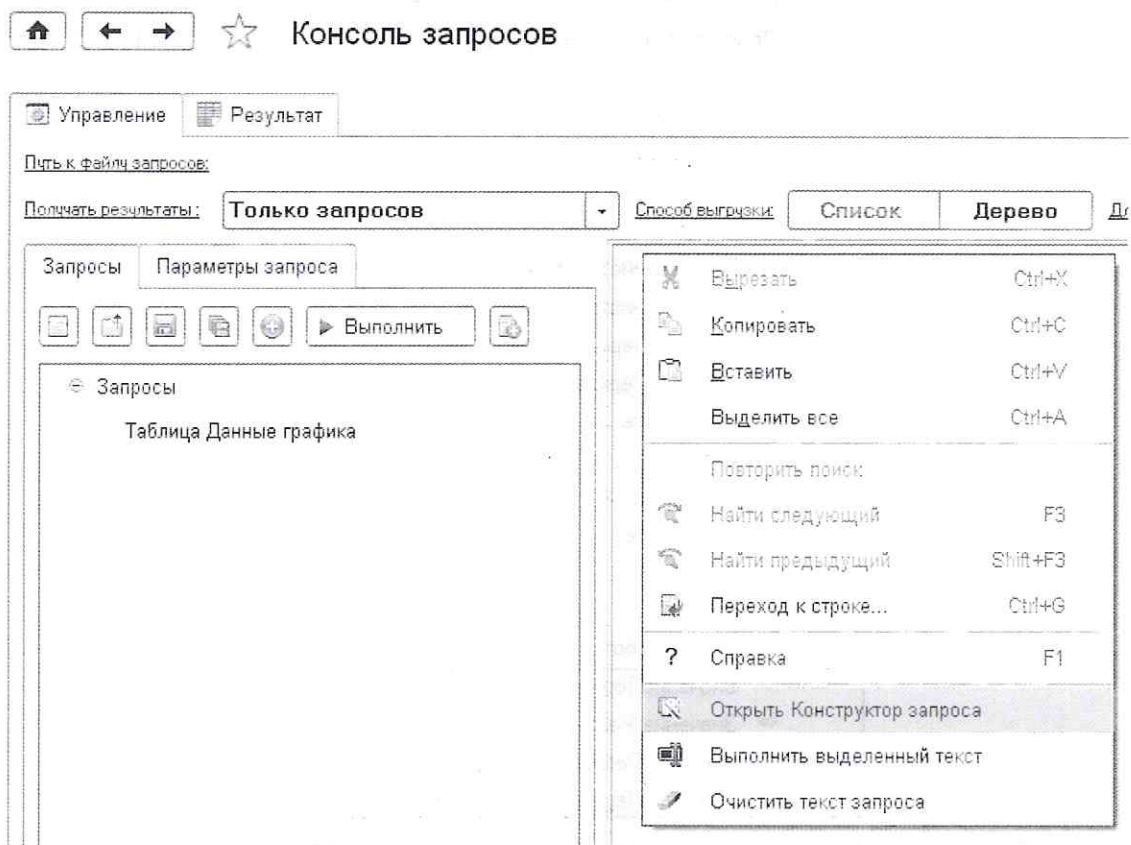
```
ВЫБРАТЬ
    ОсновныеНачисленияДанныеГрафика.НомерСтроки КАК
НомерСтроки,
    ОсновныеНачисленияДанныеГрафика.ЗначениеПериодДействия,
    ОсновныеНачисленияДанныеГрафика.ЗначениеФактическийПериодДействия
ИЗ
    РегистрРасчета.ОсновныеНачисления.ДанныеГрафика (
        Регистратор = &Регистратор
        И ВидРасчета.КатегорияРасчета =
&КатегорияРасчета) КАК ОсновныеНачисленияДанныеГрафика

УПОРЯДОЧИТЬ ПО
    НомерСтроки
```

Чтобы не заниматься ручным написанием текста запроса, мы воспользуемся Конструктором запроса, а заодно и обсудим смысл наших действий по построению этого запроса.

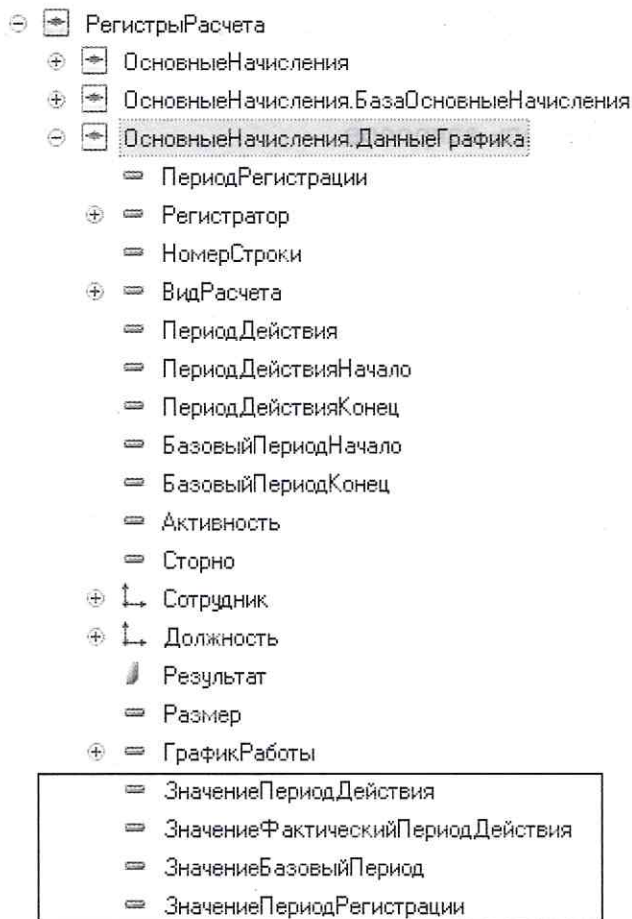
В списке запросов (древовидна структура слева) нажмем кнопку "Добавить", зададим имя нового запроса, например, "Таблица Данные графика"

и, щелкнув правой клавишей мыши на поле текста запроса (справа), откроем контекстное меню:



Выберем в контекстном меню пункт "Открыть" Конструктор запроса – откроется окно Конструктора запроса, инструмента для интерактивного построения текста запроса.

В списке доступных таблиц найдем виртуальную таблицу "ДанныеГрафика" регистра расчета "Основные Начисления" и развернем список ее полей:



Исследуем поля виртуальной таблицы "ДанныеГрафика" регистра расчета.

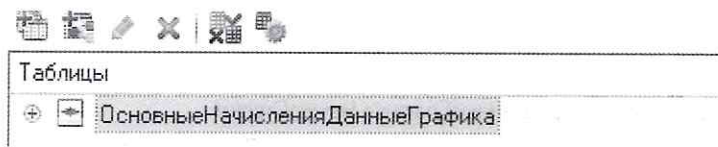
Прежде всего мы видим, что виртуальная таблица воспроизводит полностью всю структуру регистра расчета – своего "хозяина": от "ПериодРегистрации" до "ГрафикРаботы".

А далее мы видим в списке полей виртуальной таблицы четыре поля результирующих показателей виртуальной таблицы – поля "ЗначениеПериодДействия", "ЗначениеФактическийПериодДействия", "ЗначениеБазовыйПериод", "ЗначениеПериодРегистрации". Каков смысл их названий и каково содержание этих полей?

Эти поля представляют собой количество рабочего времени по ресурсу "Значение" регистра-календаря "ГрафикиРаботы" (первая часть имени поля), с которым связан наш регистр расчета по четырем периодам, названия которых мы видим во второй части имени каждого поля.

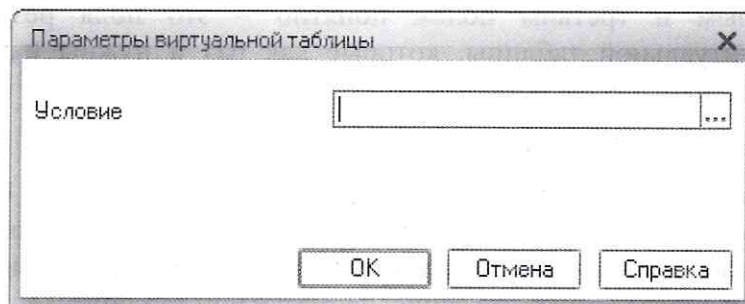
Очевидно, для наших целей поля количества рабочего времени по периоду регистрации и по базовому периоду не нужны, а нужны только поля "ЗначениеФактическийПериодДействия" и "ЗначениеПериодДействия", которые соответственно дадут нам для каждой записи регистра значения операндов расчетной формулы: **[Число отработанных дней]** и **[Число рабочих дней в месяце]**.

После этих рассуждений выберем виртуальную таблицу "ДанныеГрафика" в список таблиц нашего запроса:



После чего рассмотрим и установим параметры виртуальной таблицы.

Откроем форму параметров виртуальной таблицы:



Мы видим, что виртуальная таблица "ДанныеГрафика" имеет только один параметр – "Условие", в котором можно задать условие предварительного отбора, т.е. перед получением данных виртуальной таблицы из регистра расчета предварительно будут отобраны записи, удовлетворяющие этому условию.

Какое условие имеет смысл задавать для наших целей?

Во-первых, мы рассчитываем записи только нашего документа, следовательно, нужно наложить условие на поле "Регистратор".

Во-вторых, нужно задать условие на категорию расчета, поскольку один шаг расчетного алгоритма можно получить результаты видов расчета только одной категории расчета.

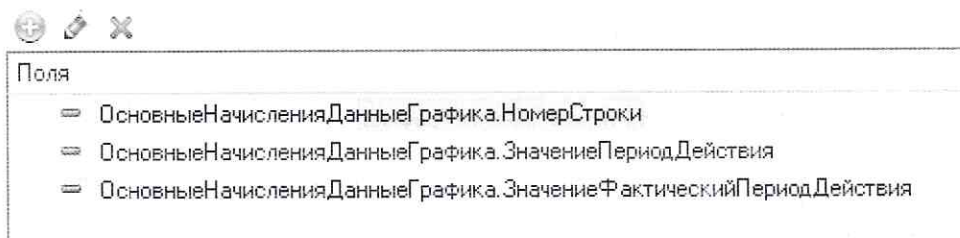
Откроем Конструктор выражений в поле "Условие" (кнопка выбора справа) и в поле выражения зададим следующее условие:

```
Регистратор = &Регистратор И ВидРасчета.КатегорияРасчета = &КатегорияРасчета
```

Для обращения к полю реквизита "КатегорияРасчета" мы воспользовались операцией разыменования ссылки – "точка", а значения условий будем задавать через параметры запроса "&Регистратор" и "&КатегорияРасчета".

После этого нажмем "ОК" в Конструкторе выражений, после его закрытия нажмем "ОК" в форме параметров виртуальной таблицы и приступим к формированию списка выбираемых полей нашего запроса.

Выберем в список полей запроса такие поля таблицы "ДанныеГрафика": "НомерСтроки", "ЗначениеФактическийПериодДействия", "ЗначениеПериодДействия":



Со вторым и третьим полем понятно – это поля результирующих показателей виртуальной таблицы, которые как раз и нужны нам в качестве необходимых данных для расчета оклада.

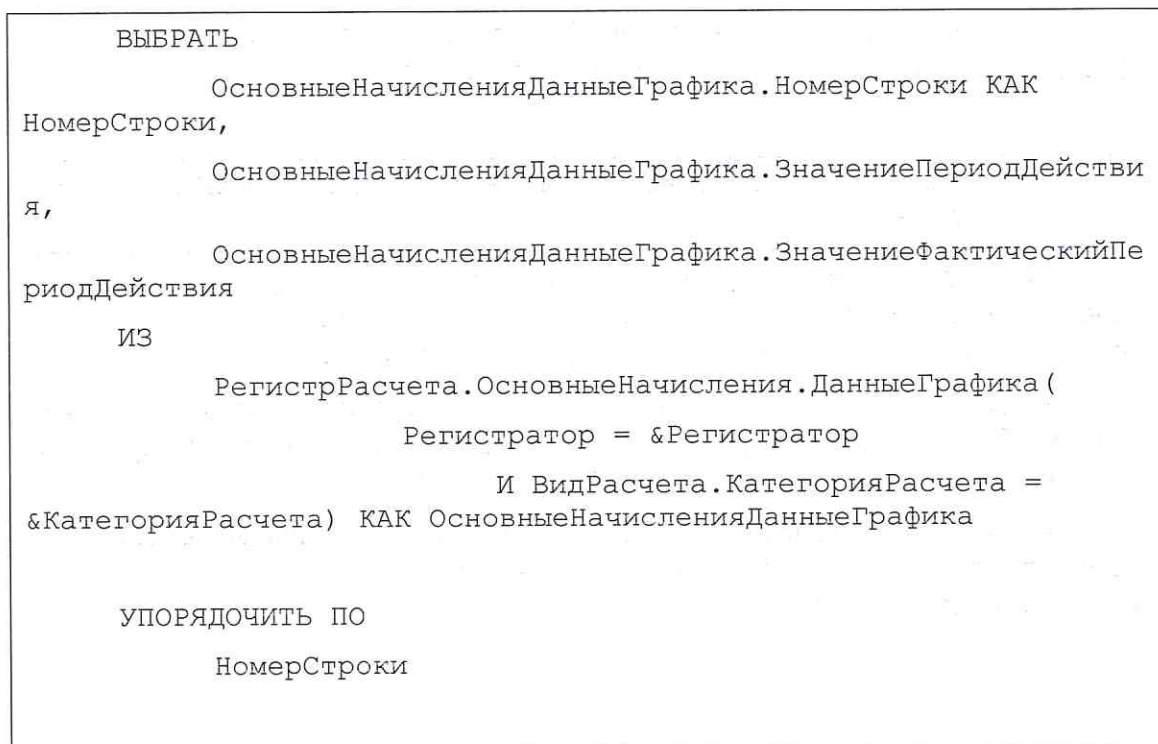
А зачем мы выбираем поле "НомерСтроки"?

А вот этот вопрос мы подробно разберем позже, когда будем писать программный код для расчета. Пока вкратце скажем, что поле "НомерСтроки" – это связующее поле записи результата виртуальной таблицы с записью регистра расчета, которой соответствует эта запись виртуальной таблицы.

В завершение построения нашего запроса на закладке Порядок Конструктора запроса зададим упорядочивание по возрастанию поля "НомерСтроки" – так будет удобнее сопоставлять записи результата запроса с записями регистра, которым эти записи виртуальной таблицы соответствуют:



Нажмем "ОК" в форме Конструктора запроса. Конструктор закроется и запрос будет помещен в поле текста запроса Консоли запросов:



Настройка и выполнение запроса

Прежде чем в Консоли запросов выполнить наш запрос, надо задать значения его параметров.

Перейдем на закладку "Параметры" запроса в форме Консоли запросов. Нажмем кнопку "Заполнить" – Консоль запросов выберет из текста запроса параметры и поместит их в таблицу параметров. А мы выберем значения параметров:

"Регистратор" – наш документ "Расчет зарплаты" (он у нас пока в единственном экземпляре);

"КатегорияРасчета" – "Первичное", поскольку мы собираемся рассчитывать вид расчета "Оклад", категория расчета у которого "Первичное".

Особый тип	Имя параметра	Значение параметра
<input type="checkbox"/>	Регистратор	Расчет зарплаты 000000001 от 01.01.2014 0:00:00
<input type="checkbox"/>	КатегорияРасчета	Первичное

После этого нажмем кнопку "Выполнить". Запрос будет выполнен, форма Консоли запросов перейдет на закладку "Результат" и мы увидим результат выполнения нашего запроса:

НомерСтроки	ЗначениеПериодДействия	ЗначениеФактическийПериодДействия
1	23	23
2	23	23
3	23	23

Наш результат пока очень прост, поэтому и смысл его раскрывается просто. Для строк набора записей нашего документа с номерами 1, 2 и 3 получены следующие, пока что одинаковые данные графика: число рабочих дней в месяце (поле "ЗначениеПериодДействия") – 23, число отработанных дней (поле "ЗначениеФактическийПериодДействия") – 23.

Именно эта сумма соответствует "количеству единиц" в поле "Значение" записей регистра сведений "ДанныеГрафика" в период с 01.01.2014 0:00:00 по 31.01.2014 23:59:59 – т.е. в границах периода действия, которое мы ввели для вида расчета "Оклад" всех наших сотрудников.

А поскольку все наши сотрудники отработали свой оклад за весь месяц полностью (мы пока не вводили вытесняющих видов расчета), то количество фактически отработанных дней равно этому количеству, т.е. 23.

Ввод вытесняющего вида расчета и влияние его на результат запроса

Сотрудник Троицкая Татьяна Сергеевна – бухгалтер. Она по роду своей работы периодически выезжает в различные контролирующие организации, вроде налоговой инспекции. Там она по целым дням "треплет себе нервы", общаясь с милыми и обходительными сотрудниками этих организаций. Поэтому руководство приняло решение оплачивать ей эти дни "местных командировок" по повышенной ставке. Когда наш бухгалтер выполняет такую работу, она не находится на своем рабочем месте. Поэтому дни, когда она бывает во внешних организациях, ей оплачиваются начислением вида расчета "Оклад на выезде".

Напомним, что это за вид расчета, который мы ранее ввели в наш план видов расчета. Расчетная формула (способ расчета) у него такая же, как у вида расчета "Оклад". Но он является вытесняющим видом расчета для "Оклад", а также предполагается, что ставка оклада ("Размер") для этого вида расчета должна быть больше, чем у вида расчета "Оклад". Таким образом руководство предприятия решает вопрос о повышенной ставке оплаты "местных командировок".

Допустим, наш бухгалтер была в "местной командировке" в период с 20.01.2014 по 27.01.2014.

Снова откроем форму нашего документа "Расчет зарплаты". Введем в табличную часть новую строку с такими значениями полей:

Сотрудник: *Троицкая Татьяна Сергеевна*

Вид расчета: *Оклад на выезде*

Период действия начало: *20.01.2014*

Период действия конец: *27.01.2014*

Размер: *75000,00*

График работы: *Пятидневка*

Остальные поля оставляем со значениями по умолчанию.

Проведем документ. В наборе записей регистра "Основные начисления" появится четвертая запись с номером строки 4.

А теперь снова выполним наш запрос к виртуальной таблице "ДанныеГрафика". На закладке "Результат" Консоли запросов мы увидим следующий результат:

НомерСтроки	ЗначениеПериодДействия	ЗначениеФактическийПериодДействия
1	23	23
2	23	17
3	23	23
4	23	6

Теперь все внимание - на результат запроса!

В нем появилась строчка со значением поля "НомерСтроки", равным 4. Это понятно, поскольку мы ввели еще одну строчку с новым видом расчета, и в наборе записей регистра она имеет номер 4. Также мы видим, что фактически отработанное время – поле "ЗначениеФактическийПериодДействия" – для нее

равно 6. Это количество рабочих дней за период с 20.01.2014 по 27.01.2014 включительно.

А теперь посмотрим на строку результата, где "НомерСтроки" равен 2. Заглянем в набор записей регистра: строка с номером 2 – это оклад бухгалтера Троицкой Т.С. Мы видим, что количество отработанного времени для этой строки изменилось – было 23, а стало 17, т.е. уменьшилось на 6 дней. Это как раз те самые 6 дней отработанного времени в строке 4 – в строке "Оклад на выезде".

Почему это произошло? Да потому, что "Оклад на выезде" мы назначили вытесняющим видом расчета для "Оклад". Произошло вытеснение: из периода действия вида расчета "Оклад" выпал интервал с 20.01.2014 по 27.01.2014 включительно, когда Троицкая Т.С. была "в местной командировке". Поэтому вид расчета "Оклад" в этом периоде "не действует" – его действие остановлено вытесняющим видом расчета "Оклад на выезде" как раз на длительность периода действия последнего.

Вот так функционал регистра расчета расправился с задачей, которую многим программистам реализовать просто не под силу: он получил количество рабочего времени по фактическому периоду действия вида расчета, для которого введен вытесняющий.

Итак, можно торжествовать: мы имеем все операнды расчетной формулы оклада:

$$\text{[Результат]} = \text{[Ставка оклада]} * \text{[Число отработанных дней]} / \text{[Число рабочих дней в месяце]}$$

И можно, взяв в руки калькулятор, получить следующие результаты:

Номер строки	Вид расчета	Сотрудник и должность	Результат
1	Оклад	Ковалев А.П., директор	$70000,00 * 23 / 23 = 70000,00$
2	Оклад	Троицкая Т.С., бухгалтер	$60000,00 * 17 / 23 = 44\ 347,83$
3	Оклад	Ильин С.В., менеджер	$50000,00 * 23 / 23 = 50000,00$
4	Оклад на выезде	Троицкая Т.С., бухгалтер	$75000,00 * 6 / 23 = 19\ 565,22$

Заметим, что г-жа Троицкая Т.С. получила в сумме по обоим видам расчета 63 913,05 руб. – т.е. больше, чем если бы она не ездила в "местную командировку", т.е. 60000,00 руб.

Общая схема расчетного алгоритма. Поле "НомерСтроки" – ключ соответствия записи регистра и ее необходимых данных

Наверное, мы не для того строили запрос к виртуальной таблице регистра расчета, чтобы считать зарплату вручную на калькуляторе. Теперь надо написать программный код, который будет делать это автоматически и записывать результаты расчета в информационную базу – в таблицу регистра расчета.

Однако это продумывание "с калькулятором в руке" имело все же некоторую пользу, а именно, у нас в голове сложилась некая общая схема, как должен строиться расчетный алгоритм.

Итак, мы выяснили, что для получения результата для каждой строки набора записей регистра расчета нужно:

а) установить расчетную формулу для этой строки (для этого у нас для каждого вида расчета имеется реквизит "СпособРасчета"; пока расчетная формула у нас одинаковая для всех записей регистра);

б) найти соответствующую ей строку в таблице результата запроса – т.е. в таблице необходимых данных, которую сформировал нам функционал регистра расчета;

в) прочитать из найденной строки содержащиеся в ней необходимые данные и подставить в расчетную формулу, получая результат расчета для рассчитываемой строки регистра.

Выше мы упоминали, как на встроенном языке реализуется пункт а) нашего алгоритма: для этого используется ветвящийся оператор "Если... ", каждая ветвь которого проверяет значение реквизита "СпособРасчета" на одно из значений способов расчета, а в гнезде каждой ветви содержится код расчетной формулы. Это вроде понятно.

Но как установить соответствие для данной строки набора записей регистра расчета и строки результата запроса с необходимыми данными? Т.е., как реализовать пункт б) нашего алгоритма?

Вспомним, что мы в запросе выбрали поле "НомерСтроки" виртуальной таблицы. Поле "НомерСтроки" в регистре расчета играет особую роль. Это – поле соответствия строки набора записей и строки необходимых данных, полученных для нее.

Важно! _____

Найти строку в таблице необходимых данных соответствующую записи регистра расчета – значит, найти в ней таблице необходимых данных такую строку, поле "НомерСтроки" которой равно полю "НомерСтроки" записи регистра расчета.

Программистских подходов к нахождению строки необходимых данных по номеру записи может быть множество, здесь каждый программист волен применять тот подход, который ему нравится. Можно, например, выгрузить результат запроса с необходимыми данными в таблицу значений и использовать метод "Найти" для колонки "НомерСтроки" таблицы значений. Можно использовать для поиска не таблицу значений, а объект "ВыборкаИзРезультатаЗапроса", который тоже позволяет вести поиск.

А можно применить и более изящный подход, который мы и применим в нашем курсе.

Запрос для использования в программном коде

Но сначала построим тот вариант запроса к виртуальной таблице "ДанныеГрафика", который мы будем использовать для решения нашей задачи.

Он будет модификацией того запроса, который мы разработали и выполнили для нашего "эксперимента" в Консоли запросов.

В нашем файле Консоли запросов добавим новую ветвь в список запросов (еще лучше скопировать наш первый вариант запроса, нажав кнопку "Скопировать запрос"). Дадим имя запросу, например, "Данные графика для расчета", а затем построим такой запрос:

```
ВЫБРАТЬ
    ОсновныеНачисления.НомерСтроки КАК НомерСтроки,
    ОсновныеНачисления.ВидРасчета.СпособРасчета КАК
СпособРасчета,
    ЕСТЬNULL (ОсновныеНачисленияДанныеГрафика.ЗначениеПери
одДействия, 0) КАК План,
    ЕСТЬNULL (ОсновныеНачисленияДанныеГрафика.ЗначениеФакт
ическийПериодДействия, 0) КАК Отработано
ИЗ
    РегистрРасчета.ОсновныеНачисления КАК
ОсновныеНачисления
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.ОсновныеНачисления.ДанныеГрафика (
    Регистратор = &Регистратор
    И ВидРасчета.КатегорияРасчета =
&КатегорияРасчета) КАК ОсновныеНачисленияДанныеГрафика
ПО ОсновныеНачисления.НомерСтроки =
ОсновныеНачисленияДанныеГрафика.НомерСтроки
ГДЕ
    ОсновныеНачисления.Регистратор = &Регистратор
    И ОсновныеНачисления.ВидРасчета.КатегорияРасчета =
&КатегорияРасчета
УПОРЯДОЧИТЬ ПО
    НомерСтроки
```

Отличия нашего "прикладного" варианта запроса от "экспериментального" следующие.

1. В запрос добавлена персистентная (основная, исходная) таблица регистра расчета "ОсновныеНачисления", которая соединяется левым соединением с виртуальной таблицей "ДанныеГрафика" по равенству полей "НомерСтроки" в обеих таблицах. Соответственно для наложения на персистентную таблицу условия по регистратору и категории расчета использована секция ГДЕ.
2. Поле "НомерСтроки" в списке полей выбора теперь выбирается из персистентной (левой) таблицы запроса.

3. Из левой (персистентной) таблицы выбирается также поле "СпособРасчета" путем разыменования ("точка") ссылочного поля "ВидРасчета" (вспомним, что "СпособРасчета" – это реквизит плана видов расчета).
4. К полям виртуальной таблицы "ЗначениеПериодДействия" и "ЗначениеФактическийПериодДействия" применяется функция языка запросов ЕСТЬNULL, которая производит замену NULL в этих полях на числовой ноль. Этим полям соответственно даны псевдонимы "План" и "Отработано".

Сначала, задав те же самые значения параметров, выполним запрос. Получим следующий результат:

НомерСтроки	СпособРасчета	План	Отработано
1	По окладу	23	23
2	По окладу	23	17
3	По окладу	23	23
4	По окладу	23	6

Результат практически идентичен результату нашего "экспериментального" запроса, только появилось еще одно выбранное поле – "СпособРасчета".

Зачем же нам понадобилась такая модификация исходного запроса?

Во-первых, поясним, зачем нам в запросе понадобилась вторая (персистентная) таблица регистра расчета.

Важно!

Виртуальные таблицы регистров накопления, бухгалтерии и расчета обладают одной особенностью: она не возвращают нулевых значений результирующих показателей.

Для нашего регистра: если, например, оба показателя для строки набора записей с номером 2 будут равны нулю, то в результате виртуальной таблицы строки с номером 2 просто не будет.

Такое может быть, например, если на рассчитываемый период (в нашем примере – январь 2014г.) нет записей в регистре "Данные графика".

Чтобы установить этот факт, мы применили стандартный прием разработки запросов в 1С: использовали в запросе персистентную таблицу регистра расчета, которая по определению содержит всю полноту строк набора записей, и соединили ее левым соединением с виртуальной таблицей. Для приведенного выше примера: если для строки номер 2 виртуальной таблицы не вернула записи (т.е. нет данных в регистре-календаре), то в результате левого соединения поля данных графика будут NULL. По этому признаку можно установить факт отсутствия записей в регистре "Данные графика" за расчетный период.

Чтобы не было ошибок при вычислениях, мы в запросе заменяем возможные NULL поля результата запроса на числовой ноль, используя функцию ЕСТЬNULL.

Во-вторых, скажем, зачем нам понадобилось в результате запроса поле "СпособРасчета". Собственно, уже ничего и пояснять не надо: по значению этого поля для каждой строки мы будем устанавливать расчетную формулу и подставлять в нее операнды – исходные и необходимые данные.

Подход к установлению соответствия строки регистра и строки необходимых данных

Можно было применить "классический" способ, какой описан выше. А именно:

- выгрузить результат запроса необходимых данных в таблицу значений;
- обходить в цикле набор записей регистра расчета как коллекцию (цикл "Для каждого..."), на каждом проходе цикла получать очередную запись и использовать метод "Найти" таблицы значений для поиска соответствующей строки с необходимыми данными по полю "НомерСтроки".

А можно применить другой прием. Обходить в цикле не набор записей регистра, а наоборот: выборку результата запроса необходимых данных, вычислять на основании значения поля "НомерСтроки" каждой позиции выборки индекс соответствующей записи в коллекции набора записей (индекс записи в коллекции равен [НомерСтроки – 1]) и получать запись по индексу из коллекции. Такой способ несколько необычен и программный код его реализации выглядит более компактно, чем код, использующий методы типа "Найти".

Мы имеем все основания обходить в цикле не набор записей регистра, а выборку запроса необходимых данных, потому что запрос содержит персистентную таблицу, стоящую слева, и номера строк по этой причине будут те же самые, что и в наборе записей: от 1 до N (т.е. до последнего).

Поэтому мы с полным основанием можем применить такой способ установления соответствия между записью регистра и строкой таблицы необходимых данных:

```
Пока ВыборкаДляРасчета.Следующий() Цикл
    // Получаем запись регистра из набора записей по
индексу:
    ЗаписьРегистра =
НаборЗаписей[ВыборкаДляРасчета.НомерСтроки - 1];
    // Реализуем расчетные формулы для каждого способа
расчета;
    . . . . .
    . . . . .
КонецЦикла;
```

Здесь:

"ВыборкаДляРасчета" – объект "ВыборкаИзРезультатаЗапроса", полученный для результата нашего запроса необходимых данных;

"НаборЗаписей" – объект "РегистрРасчетаНаборЗаписей" для документа, записи которого мы рассчитываем.

Как мы видим, в каждом проходе цикла на основании значения поля "НомерСтроки" текущей позиции выборки вычисляется (путем вычитания единицы) индекс соответствующей записи из рассчитываемого набора записей. После этого запись извлекается по индексу из набора записей как объекта-коллекции.

В результате программа имеет доступ:

- к свойствам (полям) рассчитываемой записи;
- к позиции выборки, содержащей необходимые данные для расчета этой записи.

Т.е., в распоряжении программы есть доступ ко всем операндам расчетной формулы.

Процедура "РассчитатьЗаписиРегистраРасчета" общего модуля "Расчеты"

Откроем текст общего модуля "Расчеты". Мы найдем там такую процедуру:

```
// Универсальная расчетная процедура
Процедура РассчитатьЗаписиРегистраРасчета (НаборЗаписей,
Выборка) Экспорт
    Результат = 0; // создаем промежуточную переменную
    // Обходим в цикле выборку необходимых данных:
    Пока Выборка.Следующий () Цикл
        // Получаем рассчитываемую запись регистра
        // из набора записей по индексу:
        ЗаписьРегистра =
НаборЗаписей [Выборка.НомерСтроки - 1];
        // Реализуем расчетные формулы
        // для каждого имеющегося способа расчета;
        // результат запоминаем в
        // промежуточной переменной Результат:
        // ( . . . . . )
        // Помещаем результат в запись регистра:
        ЗаписьРегистра.Результат = Результат;
    КонецЦикла ;
КонецПроцедуры
```

Точнее, это еще не процедура, а некая "заготовка" процедуры, предназначенной для выполнения расчета набора записей.

Процедура получает в качестве аргументов подлежащий расчету набор записей и выборку необходимых данных – результат выполнения запроса к виртуальным таблицам регистра расчета.

Выборка обходится в цикле, на каждом проходе цикла выполняется расчет очередной записи регистра. Результирующая цифра расчета помещается в поле ресурса "Результат" записи регистра.

Обратим внимание на строчку комментария с жирными точками (.). Это то самое место программы, где мы должны будем реализовать расчетные формулы, т.е. написать на встроенном языке выражения с участием свойств исходных данных в записи регистра (например, "Размер") и свойств текущей позиции выборки необходимых данных (например: "План", "Отработано"). Вспомним, что для этого надо использовать ветвящийся оператор "Если ... Тогда ... ИначеЕсли ... Тогда..."

Реализация расчетного кода способа расчета "ПоОкладу"

Резюмируем, что нам вообще нужно будет написать:

- 1) в модуле документа в процедуре "ОбработкаПроведения" после написанного нами ранее кода первого шага расчетного сеанса написать код, реализующий выполнение запроса для получения необходимых данных, получения выборки результата этого запроса и вызова процедуры общего модуля "РассчитатьЗаписиРегистраРасчета";
- 2) в теле процедуры "РассчитатьЗаписиРегистраРасчета" общего модуля написать оператор "Если ... Тогда ... ", который будет проверять способ расчета и применять расчетную формулу.

Начнем.

Откроем текст модуля документа "РасчетЗарплаты". В самом конце процедуры "ОбработкаПроведения" найдем предложение:

```
НаборОсновныеНачисления.Записать ( , Ложь ) ;
```

Это последнее предложение кода первого шага расчетного сеанса. Сразу после него напишем следующий код:

```
Запрос = Новый Запрос ( ) ;
Запрос.Текст =
"ВЫБРАТЬ
| ОсновныеНачисления.НомерСтроки КАК НомерСтроки,
| ОсновныеНачисления.ВидРасчета.СпособРасчета КАК
СпособРасчета,
|
| ЕСТЬNULL (ОсновныеНачисленияДанныеГрафика.ЗначениеПериодДейс
твия, 0) КАК План,
|
| ЕСТЬNULL (ОсновныеНачисленияДанныеГрафика.ЗначениеФактически
йПериодДействия, 0) КАК Отработано
```

```
| ИЗ
|     РегистрРасчета.ОсновныеНачисления КАК
ОсновныеНачисления
|     ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.ОсновныеНачисления.ДанныеГрафика (
|     Регистратор = &Регистратор
|     И ВидРасчета.КатегорияРасчета =
&КатегорияРасчета) КАК ОсновныеНачисленияДанныеГрафика
|     ПО ОсновныеНачисления.НомерСтроки =
ОсновныеНачисленияДанныеГрафика.НомерСтроки
| ГДЕ
|     ОсновныеНачисления.Регистратор = &Регистратор
|     И ОсновныеНачисления.ВидРасчета.КатегорияРасчета =
&КатегорияРасчета";

Запрос.УстановитьПараметр ("Регистратор", Ссылка);
Запрос.УстановитьПараметр ("КатегорияРасчета",
Справочники.КатегорииРасчета.Первичное);

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

Расчеты.РассчитатьЗаписиРегистраРасчета (НаборОсновныеНачи
сления, ВыборкаДляРасчета);

НаборОсновныеНачисления.Записать(, Истина);
```

Несмотря на свою громоздкость, код довольно прост.

Вначале создается объект "Запрос", который инициализируется текстом того самого запроса получения необходимых данных (данных графика), который мы разработали и отладили в Консоли запросов.

Затем устанавливаются значения параметров запроса. В запросе у нас два параметра: "&Регистратор" и "&КатегорияРасчета". Первый инициализируется значением ссылки документа, второй – значением ссылки на элемент "Первичное" справочника "КатегорииРасчетов" (этот элемент является предопределенным, так что его ссылку получаем сразу по программному имени элемента).

Потом запрос выполняется и получается выборка его результата (переменная "ВыборкаДляРасчета").

После чего вызывается процедура общего модуля "РассчитатьЗаписиРегистраРасчета", которой передаются полученная выборка и набор записей регистра "ОсновныеНачисления", который доступен в модуле объекта документа через коллекцию "Движения". Мы уже знаем, что эта процедура и должна выполнять расчет, помещая результат расчета в поле Результат рассчитываемой записи регистра расчета.

После вызова расчетной процедуры общего модуля набор записей регистра расчета записывается в информационную базу, причем подчеркнем, что метод "Записать" вызывается со вторым аргументом, равным Истина – поскольку

выполняется запись набора с результатами расчета, и от пересчета фактического периода действия при записи следует отказаться.

Вот теперь пришло время написания кода реализации расчетной формулы в процедуре "РасчитатьЗаписиРегистраРасчета".

Откроем на редактирование текст общего модуля "Расчеты", найдем в его процедуре "РасчитатьЗаписиРегистраРасчета" то самое место – комментарий вида

```
// (.....)
```

И прямо под ним напишем следующий код:

```
// (.....)
Если Выборка.СпособРасчета =
Перечисления.СпособыРасчета.ПоОкладу Тогда
    Если Выборка.План > 0 Тогда
        Результат = ЗаписьРегистра.Размер *
Выборка.Отработано / Выборка.План;
    КонецЕсли;

Иначе
    Результат = 0;
КонецЕсли;
```

Это и есть тот оператор "Если ...", который мы будем "разветвлять" по мере того, как будем реализовывать новые способы расчета.

Как мы видим, выражение реализует расчетную формулу для оклада и помещает результат вычислений в промежуточную переменную Результат (предварительно проверяется значение "Выборка.План", чтобы не было деления на 0). Потом мы будем добавлять новые ветви "ИначеЕсли ..." для каждого очередного способа расчета (там, где пустая строка), а "для подстраховки" в ветви Иначе обнуляем переменную "Результат": вдруг забудем реализовать какой-либо способ расчета – нулевой результат это сразу покажет.

Для контроля приведем код процедуры обработки проведения документа и код процедуры "РасчитатьЗаписиРегистраРасчета" общего модуля "Расчеты".

Код процедуры "ОбработкаПроведения" документа "РасчетЗарплаты":

```
Процедура ОбработкаПроведения (Отказ, РежимПроведения)
    НаборОсновныеНачисления =
Движения.ОсновныеНачисления;
    ТекстЗапроса =
        "ВЫБРАТЬ
        |
        РасчетЗарплатыОсновныеНачисления.Ссылка.ПериодРегистрации,
        |      РасчетЗарплатыОсновныеНачисления.Сотрудник,
```

```

        |      РасчетЗарплатыОсновныеНачисления.Должность,
        |      РасчетЗарплатыОсновныеНачисления.ВидРасчета,
        |
        |      РасчетЗарплатыОсновныеНачисления.ПериодДействияНачало,
        |
        |      КОНЕЦПЕРИОДА (РасчетЗарплатыОсновныеНачисления.ПериодДействи
        |      яКонец, ДЕНЬ) КАК ПериодДействияКонец,
        |
        |      РасчетЗарплатыОсновныеНачисления.БазовыйПериодНачало,
        |      ВЫБОР
        |
        |      КОГДА
        |      РасчетЗарплатыОсновныеНачисления.БазовыйПериодКонец =
        |      ДАТАВРЕМЯ(1, 1, 1)
        |
        |      ТОГДА ДАТАВРЕМЯ(1, 1, 1)
        |
        |      ИНАЧЕ
        |      КОНЕЦПЕРИОДА (РасчетЗарплатыОсновныеНачисления.БазовыйПериодКонец
        |      , ДЕНЬ)
        |
        |      КОНЕЦ КАК БазовыйПериодКонец,
        |      РасчетЗарплатыОсновныеНачисления.Размер,
        |      РасчетЗарплатыОсновныеНачисления.ГрафикРаботы,
        |      РасчетЗарплатыОсновныеНачисления.Сторно
        |      ИЗ
        |      Документ.РасчетЗарплаты.ОсновныеНачисления КАК
        |      РасчетЗарплатыОсновныеНачисления
        |      ГДЕ
        |      РасчетЗарплатыОсновныеНачисления.Ссылка =
        |      &Ссылка
        |
        |      УПОРЯДОЧИТЬ ПО
        |      РасчетЗарплатыОсновныеНачисления.НомерСтроки";
        |
        |      Расчеты.ЗаполнитьНаборЗаписей(ТекстЗапроса, Ссылка,
        |      НаборОсновныеНачисления);
        |
        |      НаборОсновныеНачисления.Записать(, Ложь);
        |
        |      Запрос = Новый Запрос();
        |      Запрос.Текст =
        |      "ВЫБРАТЬ
        |      |      ОсновныеНачисления.НомерСтроки КАК НомерСтроки,
        |      |      ОсновныеНачисления.ВидРасчета.СпособРасчета КАК
        |      |      СпособРасчета,
```

```
|
  ЕСТЬNULL (ОсновныеНачисленияДанныеГрафика.ЗначениеПериодДейс
твия, 0) КАК План,
|
  ЕСТЬNULL (ОсновныеНачисленияДанныеГрафика.ЗначениеФактически
йПериодДействия, 0) КАК Отработано
|ИЗ
|   РегистрРасчета.ОсновныеНачисления КАК
ОсновныеНачисления
|       ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.ОсновныеНачисления.ДанныеГрафика (
|               Регистратор = &Регистратор
|               И
ВидРасчета.КатегорияРасчета = &КатегорияРасчета) КАК
ОсновныеНачисленияДанныеГрафика
|               ПО ОсновныеНачисления.НомерСтроки =
ОсновныеНачисленияДанныеГрафика.НомерСтроки
|ГДЕ
|   ОсновныеНачисления.Регистратор = &Регистратор
|   И ОсновныеНачисления.ВидРасчета.КатегорияРасчета
= &КатегорияРасчета";

Запрос.УстановитьПараметр ("Регистратор", Ссылка);
Запрос.УстановитьПараметр ("КатегорияРасчета",
Справочники.КатегорииРасчета.Первичное);

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

Расчеты.РассчитатьЗаписиРегистраРасчета (НаборОсновны
еНачисления, ВыборкаДляРасчета);

НаборОсновныеНачисления.Записать (, Истина);

КонецПроцедуры
```

Код процедуры "РассчитатьЗаписиРегистраРасчета" общего модуля "Расчеты":

```
// Универсальная расчетная процедура
Процедура РассчитатьЗаписиРегистраРасчета (НаборЗаписей,
Выборка) Экспорт
    Результат = 0; // создаем промежуточную переменную
    // Обходим в цикле выборку необходимых данных:
    Пока Выборка.Следующий () Цикл
        // Получаем рассчитываемую запись регистра
        // из набора записей по индексу:
        ЗаписьРегистра =
```

```

НаборЗаписей[Выборка.НомерСтроки - 1];

        // Реализуем расчетные формулы
        // для каждого имеющегося способа расчета;
        // результат запоминаем в
        // промежуточной переменной Результат:
        Если Выборка.СпособРасчета =
Перечисления.СпособыРасчета.ПоОкладу Тогда
            Если Выборка.План > 0 Тогда
                Результат = ЗаписьРегистра.Размер *
Выборка.Отработано / Выборка.План;
            КонецЕсли;

        Иначе
            Результат = 0;
        КонецЕсли;
        // Помещаем результат в запись регистра:
        ЗаписьРегистра.Результат = Результат;
    КонецЦикла;
КонецПроцедуры
    
```

Выполнение расчетного кода. Расчет начислений

Обновим конфигурацию базы данных. Запустим ее в режиме "1С:Предприятие" и проведем наш документ. Мы увидим в регистре "Основные начисления" следующие результаты:

Номер строки	Вид расчета	Сотрудник	Должность	Результат
1	Оклад	Ковалев Александр Петрович	Директор	70 000,00
2	Оклад	Троицкая Татьяна Сергеевна	Бухгалтер	44 347,83
3	Оклад	Ильин Сергей Викторович	Менеджер	50 000,00
4	Оклад на выезде	Троицкая Татьяна Сергеевна	Бухгалтер	19 565,22

Мы видим, что результаты работы нашего расчетного алгоритма полностью совпадают с полученными выше результатами ручного расчета.

Практикум № 1

В каркасной базе имеется значение графика работы "Почасовка" и наполнение регистра-календаря для этого графика (рабочее время вычисляется в часах).

Реализуйте в конфигурации новый способ расчета: "ЧасовойТариф". Расчетная формула такова:

[Фактически отработанное время в часах] * [Часовая тарифная ставка]

Часовая тарифная (в рублях) ставка известна до начала расчета.

Создайте вид расчета "По тарифу" с этим способом расчета, введите и рассчитайте его для какого-либо сотрудника на произвольной должности за произвольный период (можно добавить в справочники новые значения).

Выбор плана видов расчета и другие решения по реализации здесь и в дальнейшем принимаются самостоятельно.

Практикум № 2

Реализуйте в конфигурации новый способ расчета: "СвободныйГрафик". Алгоритм расчета таков.

Расчетная формула:

[Фактически отработанное время в днях] * [Дневная тарифная ставка]

Дневная тарифная ставка известна до начала расчета. Если фактически отработанное время составляет 80% и более от планового рабочего времени, то дневная тарифная ставка увеличивается на 10%.

Создайте вид расчета "По свободному графику" с этим способом расчета, введите и рассчитайте его для произвольного сотрудника на произвольной должности.

Практикум № 3

Реализуйте способ расчета: "ШтрафЗаПрогул". Расчетная формула:

-N руб. за каждый день прогула.

Пусть значение N хранится в константе.

Создайте вид расчета "Штраф" с этим способом расчета, который вытесняет вид расчета "Оклад". Введите и рассчитайте его для произвольного сотрудника на произвольной должности.

Программное получение интервалов фактического периода действия. Виртуальная таблица "ФактическийПериодДействия"

Виртуальная таблица "ДанныеГрафика" дает нам количество отработанного времени, суммируя рабочие дни по всем интервалам фактического периода действия.

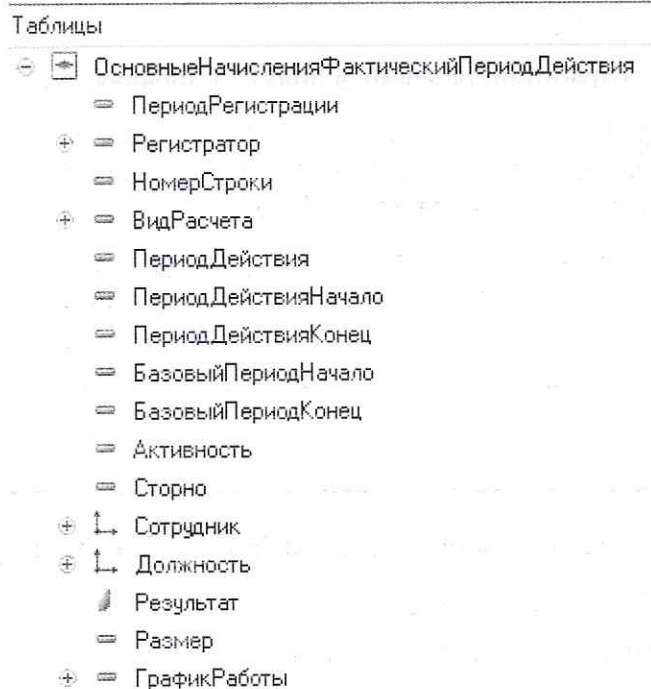
Но может случиться так, что какая-нибудь очень экзотическая расчетная формула потребует доступа не только к количеству рабочего времени за фактический период действия, но и множество самих интервалов фактического периода действия, т.е. их количество и границы этих интервалов.

Границы интервалов фактического периода действия можно получить запросом, обращаясь к виртуальной таблице "ФактическийПериодДействия" регистра расчета. Для знакомства с ней откроем Консоль запросов, добавим список запросов еще один запрос и назовем его "ФПД" (фактический период действия).

Откроем Конструктор запроса и выберем из списка доступных таблиц (ветвь "Регистры расчета") в состав таблиц нашего запроса таблицу, которая называется так:

"ОсновныеНачисления.ФактическийПериодДействия"

Эта таблица как раз и даст нам список интервалов фактического периода действия. Развернем структуру ее полей и увидим, что эта таблица полностью повторяет структуру полей регистра расчета "ОсновныеНачисления" – и все, больше в ней никаких полей нет:



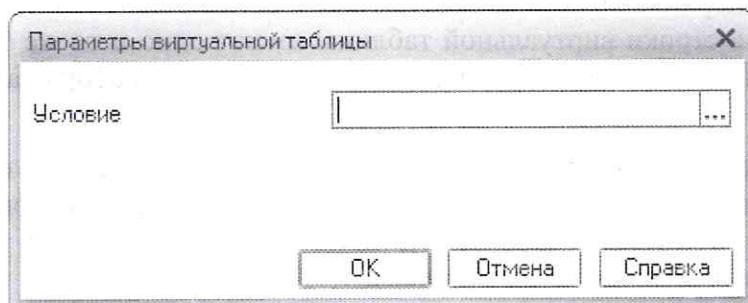
Какую же информацию может тогда дать эта таблица, если в ней нет никаких результирующих полей?

Все дело в том, что каждая запись этой таблицы соответствует одному интервалу фактического периода действия вида расчета. А границы этого

интервала находятся в полях "ПериодДействияНачало" и "ПериодДействияКонец" записи. Т.е., для каждого вида расчета эта таблица содержит столько записей, на сколько интервалов распался его изначальный период действия под воздействием вытеснения.

Зададим для таблицы псевдоним "ФПД" – исключительно для большей компактности текста запроса.

Откроем форму параметров этой виртуальной таблицы. Мы увидим, что она принимает только один (необязательный) параметр – "Условие", представляющий собой логическое выражение условия предварительного отбора:



Зададим условие на поле "Сотрудник" регистра расчета:

```
Сотрудник = &Сотрудник
```

В состав полей запроса выберем следующие поля:

- "НомерСтроки";
- "Сотрудник";
- "ВидРасчета";
- "ПериодДействияНачало";
- "ПериодДействияКонец"

Получится вот такой запрос:

```
ВЫБРАТЬ
    ФПД.НомерСтроки,
    ФПД.Сотрудник,
    ФПД.ВидРасчета,
    ФПД.ПериодДействияНачало,
    ФПД.ПериодДействияКонец
ИЗ
    РегистрРасчета.ОсновныеНачисления.ФактическийПериодДе
йствия (Сотрудник = &Сотрудник) КАК ФПД
```

Нажмем ОК в форме Конструктора запроса и зададим значение параметра "&Сотрудник", выберем сотрудника:

Троицкая Татьяна Сергеевна

Теперь выполним запрос. Получим такой результат запроса:

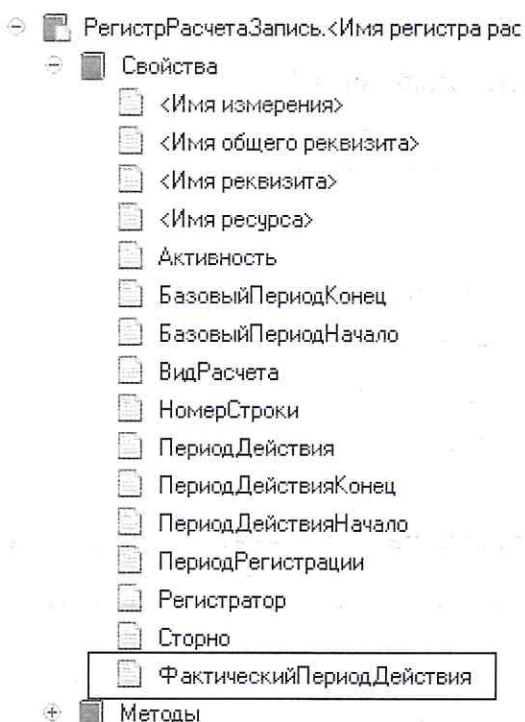
Номер Строки	Сотрудник	ВидРасчета	ПериодДействияНачало	ПериодДействияКонец
4	Троицкая Татьяна Сергеевна	Оклад на выезде	20.01.2014 0:00:00	27.01.2014 23:59:59
2	Троицкая Татьяна Сергеевна	Оклад	28.01.2014 0:00:00	31.01.2014 23:59:59
2	Троицкая Татьяна Сергеевна	Оклад	01.01.2014 0:00:00	19.01.2014 23:59:59

Мы видим, что виду расчета "Оклад" соответствуют два интервала фактического периода действия: с 01.01.2014 по 19.01.2014 и с 28.01.2014 по 31.01.2014. Обе строки виртуальной таблицы имеют одно и то же значение поля "НомерСтроки": 2. Т.е. обе строки относятся к строке 2 набора записей нашего документа.

Троицкая Т.С. отработала с 01.01.2014 по 19.01.2014, затем была в "местной командировке", а потом вернулась и доработала до конца месяца, с 28.01.2014 по 31.01.2014. Это и отражает состав интервалов фактического периода действия для вида расчета "Оклад".

А за время местной командировки ей начислялся "Оклад на выезде" – вытесняющий вид расчета для "Оклад". Его период действия - с 20.01.2014 по 28.01.2014 – как раз соответствует периоду, "выпавшему" из исходного периода действия вида расчета "Оклад". Поскольку для "Оклад" на выезде вытесняющих видов расчета не задавалось, то его период действия изменений не претерпел.

Завершая разговор об интервалах фактического периода действия, надо отметить, что их состав можно получить не только запросом, но и через объектную модель, т.е. на встроенном языке. Посмотрим в Синтакс-помощнике описание объекта "РегистрРасчетаЗапись". Мы увидим там свойство "ФактическийПериодДействия":



Прочитав справку об этом свойстве, мы узнаем, что оно имеет тип <ФиксированныйМассив>, а каждый элемент этого массива имеет тип

<ЭлементФактическогоПериодаДействия>. Последний объект имеет свойства: "ДатаНачала" и "ДатаОкончания". Таким образом, обходя этот массив в цикле, можно получить доступ к границам интервалов фактического периода действия данной записи.

8.8.2. Виртуальная таблица "База" регистра расчета. Расчет начислений со способом расчета "Процентом"

Мы познакомились с функциональностью регистра расчета, позволяющей получить первый род необходимых данных – данные графика. Теперь рассмотрим, что нужно сделать, чтобы получить второй род необходимых данных – сумму базы для тех видов расчета, в расчетных формулах которых присутствует этот операнд.

Ввод начисления "Доплата за квалификацию"

Ранее мы создали вид расчета "Доплата за квалификацию" и назначили ему базовые виды расчета: "Оклад" и "Оклад на выезде", а также назначили категорию расчета: "Зависимое первого уровня" – потому что его базовые виды расчета имеют категорию "Первичное".

Расчетная формула для этого вида расчета будет такой:

$$[\text{Сумма}] = [\text{Сумма базы}] * [\text{Процент}] / 100$$

Здесь [Процент] – это процент доплаты за квалификацию, он задается целым числом меньше 100, поэтому результат расчета и делится на 100. Это – исходное данное, для хранения которого будет служить реквизит "Размер" регистра расчета. А [Сумма базы] - это необходимое данное, которое следует получить, используя функционал регистра расчета. Способ расчета для этой расчетной формулы мы назвали "Процентом" и назначили его виду расчета "Доплата за квалификацию".

Откроем в режиме "1С:Предприятие" наш документ "Расчет зарплаты". Введем в табличную часть "Основные начисления" новую строку и заполним ее поля такими значениями:

Сотрудник: *Троицкая Татьяна Сергеевна*

Должность: *Бухгалтер*

Вид расчета: *Доплата за квалификацию*

Период действия начало: *01.01.2014*

Период действия конец: *31.01.2014*

Базовый период начало: *01.01.2014*

Базовый период конец: *31.01.2014*

Размер: *10,00*

Остальные поля оставим пустыми по умолчанию.

Проанализируем эти значения. Очевидно, поскольку вид расчета требует получения суммы базы, то мы вполне обоснованно задали для него границы базового периода. Поскольку мы рассчитываем доплату за квалификацию за январь, то и должны будем получить сумму результатов базовых видов расчета тоже за этот период.

Но зачем мы задали границы периода действия? Ведь расчетная формула не содержит данных графика, следовательно, для получения необходимых данных этого вида расчета мы не будем обращаться к таблице "ДанныеГрафика".

Дело в том, что мы в настройках нашего регистра установили флажок "Период действия", т.е. снабдили его функционалом получения данных графика. А если имеется такая настройка регистра расчета, то мы должны обязательно задавать в каждой его записи границы периода действия, иначе набор записей просто не будет записан в информационную базу. Таково требование платформы.

Важно!

Если в метаданных регистра расчета установлено свойство Период действия, то в каждой записи регистра обязательно требуется задавать границы периода действия, даже если расчетная формула для этой записи не требует получения данных графика.

Иначе будет выдано сообщение об ошибке и набор записей не будет записан в информационную базу.

Проведем документ. В нашем регистре расчета появится запись о виде расчета Доплата за квалификацию с исходными данными, но поле "Результат", понятно, буде пустым, поскольку у нас еще не реализован расчетный код для способа расчета "Процентом".

Виртуальные таблицы "База" регистра расчета

Вторая часть функционала регистра расчета – получение суммы базы – реализуется путем обращения к виртуальной таблице "База" регистра расчета.

В первую очередь надо сказать, что виртуальных таблиц "База" у регистра расчета может быть несколько. Число этих таблиц равно числу базовых регистров – т.е. таких регистров расчета, в которых у данного регистра расчета могут находиться результаты базовых видов расчета.

Сколько может быть базовых регистров? Не менее, чем число базовых планов видов расчета у того плана видов расчета, с которым связан данный регистр расчета – потому что каждый план видов расчета может быть связан с несколькими регистрами расчета.

Важно!

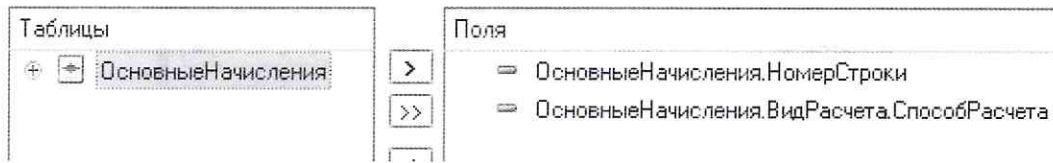
Таблиц "База" у регистра расчета может быть несколько – по числу его базовых регистров расчета.

Разработка запроса для получения расчетной базы. Параметры виртуальной таблицы "База"

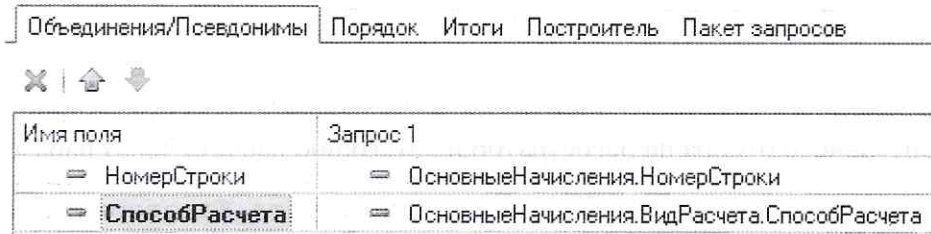
Как и в прошлый раз, мы разработаем запрос сначала в Консоли запросов, получим его результат и убедимся в его правильности. А потом используем этот запрос в расчетном коде нашей программы.

Откроем Консоль запросов, добавим в список запросов новый запрос, назовем, его, скажем, так: "Данные базы регистра Основные начисления". Откроем Конструктор запроса.

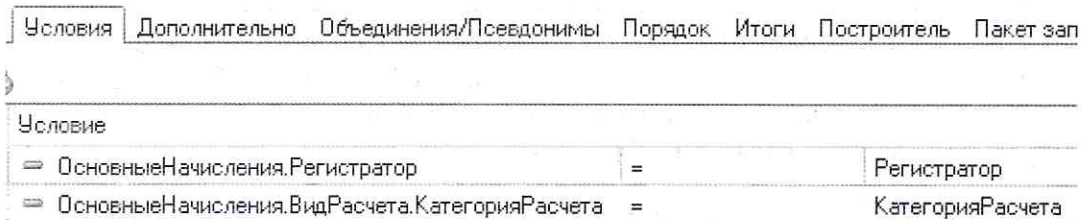
Запрос будем строить сразу в окончателно варианте. Поэтому, как в прошлый раз, добавим в запрос персистентную таблицу регистра расчета "ОсновныеНачисления" и выберем из нее поля "НомерСтроки" и "ВидРасчета.СпособРасчета":



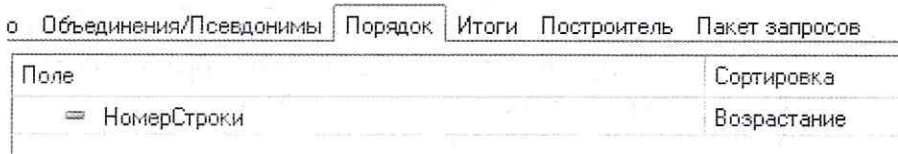
Последнему полю дадим псевдоним СпособРасчета:



Сформируем условия через параметры, такие же, как и для предыдущего запроса:



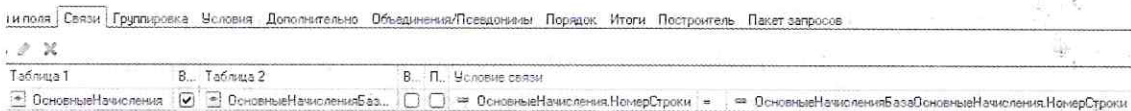
И сразу зададим упорядочивание по "НомерСтроки":



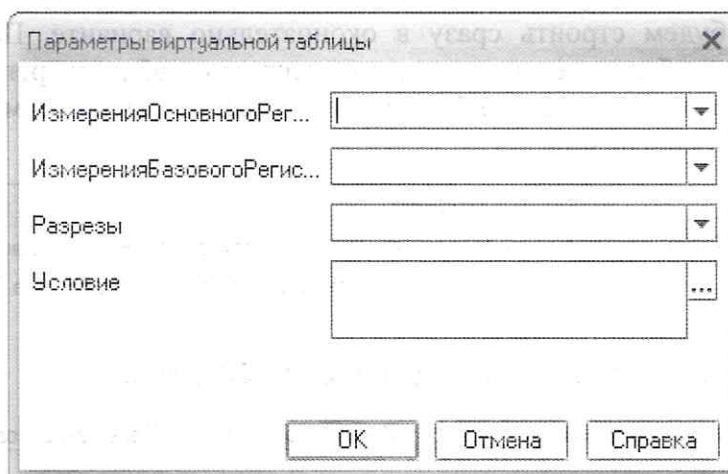
Теперь выберем в список таблиц нашего запроса виртуальную таблицу, которая называется так: "ОсновныеНачисления.БазаОсновныеНачисления" – это и есть таблица базы регистра "ОсновныеНачисления". Отметим, что у этого регистра расчета одна таблица базы, потому что у него один базовый регистр – он сам.

Обратим также внимание, как именуются таблицы "База". Имя начинается со слова База и следом идет имя одного из базовых регистров регистра расчета – "хозяина".

На закладке "Связи" сразу создадим левое соединение с условием по равенству полей "НомерСтроки" в обеих таблицах:



Теперь займемся параметрами виртуальной таблицы "База". Откроем форму параметров виртуальной таблицы:



Начнем со знакомого параметра – "Условие". Зададим в нем условие предварительного отбора на регистратор и категорию расчета через параметры:

```
Регистратор = &Регистратор И ВидРасчета.КатегорияРасчета = &КатегорияРасчета
```

Параметр "Разрезы" мы пока использовать не будем, поговорим о нем позже.

А параметры "ИзмеренияОсновногоРегистра" и "ИзмеренияБазовогоРегистра" – нужно задавать обязательно. (Это единственный случай для виртуальных таблиц регистров платформы 1С 8, когда параметры виртуальной таблицы являются обязательными).

Зачем нужны эти два параметра?

Если взять общий случай – наличие нескольких базовых регистров, то имена измерений в базовом регистра могут отличаться от имен измерений "основного" регистра – т.е. регистра, для которого получается сумма базы. А виртуальной таблице для получения суммы базы нужно брать только те записи базового регистра, в которых значения измерений совпадают со значениями соответствующих измерений основного регистра. Так, для расчета премии бухгалтера Троицкой нужно брать результаты базовых видов расчета только бухгалтера Троицкой и никакого другого сотрудника. Поэтому виртуальная таблица должна производить отбор записей базового регистра по значениям измерений той записи основного регистра, для которой база получается. Но, если имена измерений основного и базового регистра не совпадают, то виртуальная таблица не может определить соответствие измерений основного и базового регистра...

Для того, чтобы установить соответствие имен измерений основного и базового регистров, и служат первые два параметра таблицы "База". Они задаются объектами типа либо <Массив>, либо <СписокЗначений>. (В Консоли запросов доступен только тип <СписокЗначений>).

В параметре "ИзмеренияОсновногоРегистра" в качестве элементов размещаются строковые имена измерений основного регистра. Например, для нашего регистра это будут элементы со значениями "Сотрудник" и "Должность".

В параметре "ИзмеренияБазовогоРегистра" в том же порядке (обязательно!) размещаются элементы с именами соответствующих измерений базового регистра, таблица базы которого параметризуется. Таким образом,

механизм виртуальной таблицы, сопоставляя элементы с именами основного и базового регистра, может установить, какое измерение базового регистра соответствует какому измерению основного регистра.

Важно!

Параметры "ИзмеренияОсновногоРегистра" и "ИзмеренияБазовогоРегистра" должны задаваться обязательно, даже если имена измерений основного и базового регистра совпадают. Умолчания для этих параметров нет.

Но, поскольку в нашем частном случае измерения основного и базового регистра называются одинаково (это один и тот же регистр), то мы можем задать оба списка имен измерений одним параметром - "&Измерения" и будем заполнять только один массив (или список значений):

Параметры виртуальной таблицы

ИзмеренияОсновногоРег... &Измерения

ИзмеренияБазовогоРегис... &Измерения

Разрезы

Условие
Регистратор = &Регистратор
И
ВидРасчета.КатегорияРасчета =
&КатегорияРасчета

OK Отмена Справка

Параметры виртуальной таблицы созданы, нажмем "OK" в форме параметров.

Исследуем теперь состав полей таблицы "База":

- [-] [-] ОсновныеНачисленияБазаОсновныеНачисления
 - = ПериодРегистрации
 - + = Регистратор
 - = НомерСтроки
 - + = ВидРасчета
 - = ПериодДействия
 - = ПериодДействияНачало
 - = ПериодДействияКонец
 - = БазовыйПериодНачало
 - = БазовыйПериодКонец
 - = Активность
 - = Сторно
 - + ↕ Сотрудник
 - + ↕ Должность
 - 📄 Результат
 - = Размер
 - + = ГрафикРаботы
 - 📄 **РезультатБаза**
 - = ПериодРегистрацииРазрез
 - + = РегистраторРазрез
 - = НомерСтрокиРазрез
 - + = ВидРасчетаРазрез
 - = ПериодДействияРазрез
 - + ↕ СотрудникРазрез
 - + ↕ ДолжностьРазрез
 - = РазмерРазрез
 - + = ГрафикРаботыРазрез

Мы видим, что сначала идут поля регистра расчета – "хозяина"; а также видим результирующее поле таблицы – "РезультатБаза". Результирующие поля рассчитываются по всем ресурсам базового регистра, а названия их состоят из имени ресурса и суффикса "База".

Поскольку в запросе имеется левое соединение, то возможны случаи, когда поля правой таблицы будут NULL. Чтобы избежать появления NULL в результате запроса, мы выберем поле "РезультатБаза" в список полей запроса не непосредственно, а напишем в списке полей выбора выражение:

ЕСТЬNULL(ОсновныеНачисленияБазаОсновныеНачисления.РезультатБаза, 0)

Дадим этому полю псевдоним "СуммаБазы":

	Объединения/Псевдонимы	Порядок	Итоги	Построитель	Пакет запросов
✕ ⬆ ⬇					
ээ ду...]	Запрос 1				
	Имя поля				
	= НомерСтроки			= ОсновныеНачисления.НомерСтроки	
	= СпособРасчета			= ОсновныеНачисления.ВидРасчета.СпособРасчета	
	f(3) СуммаБазы			f(3) ЕСТЬNULL(ОсновныеНачисленияБазаОсновныеНачисления.РезультатБаза, 0)	

Запрос готов:

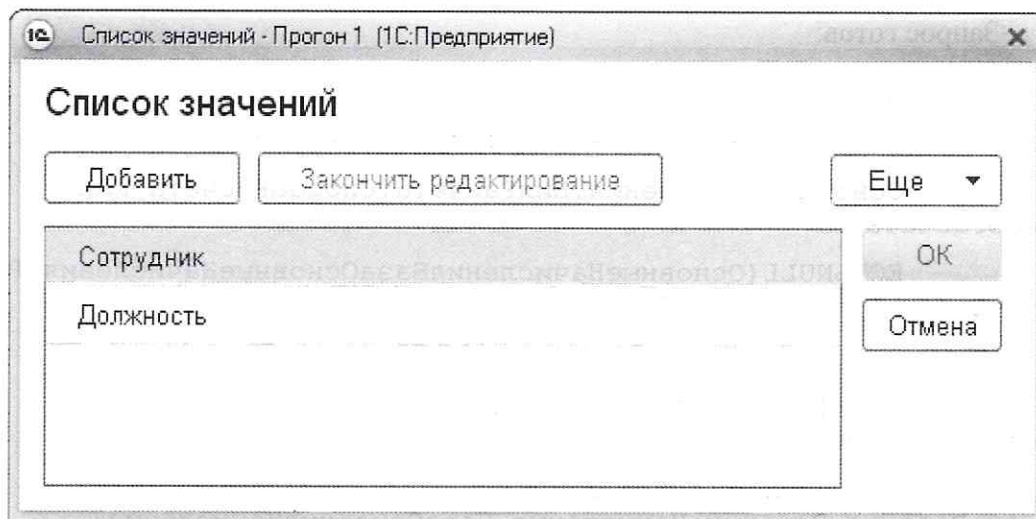
```
ВЫБРАТЬ
    ОсновныеНачисления.НомерСтроки КАК НомерСтроки,
    ОсновныеНачисления.ВидРасчета.СпособРасчета КАК
СпособРасчета,
    ЕСТЬNULL (ОсновныеНачисленияБазаОсновныеНачисления.Рез
ультатБаза, 0) КАК СуммаБазы
ИЗ
    РегистрРасчета.ОсновныеНачисления КАК
ОсновныеНачисления
    ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.ОсновныеНачисления.БазаОсновныеНачисления (
        &Измерения,
        &Измерения,
        ,
        Регистратор = &Регистратор
        И ВидРасчета.КатегорияРасчета =
&КатегорияРасчета) КАК ОсновныеНачисленияБазаОсновныеНачисления
    ПО ОсновныеНачисления.НомерСтроки =
ОсновныеНачисленияБазаОсновныеНачисления.НомерСтроки
ГДЕ
    ОсновныеНачисления.Регистратор = &Регистратор
    И ОсновныеНачисления.ВидРасчета.КатегорияРасчета =
&КатегорияРасчета
УПОРЯДОЧИТЬ ПО
    НомерСтроки
```

Зададим в Консоли запросов значения параметров запроса:

Регистратор: *Расчет зарплаты 000000001*

КатегорияРасчета: *Зависимое первого уровня*

Для параметра "&Измерения" зададим тип <СписокЗначений> и заполним список значений именами измерений нашего регистра расчета:



Выполним запрос, получим такой результат:

НомерСтроки	СпособРасчета	СуммаБазы
5	Процентом	63 913,05

Строка №5 в регистре расчета – это вид расчета "Доплата за квалификацию" бухгалтера Троицкой. А число 63 913,05 – это сумма результатов базовых видов расчета для него:

Оклад: 44 347,83 руб.

Оклад на выезде: 19 565,22 руб.

44 347,83 руб. + 19 565,22 руб. = 63 913,05 руб.

Запрос для получения необходимых данных работает правильно. Можно приступать к разработке программного кода, реализующего способ расчета "Процентом".

Реализация расчетного кода способа расчета "Процентом"

Еще раз вспомним, что в самом конце программы расчета оклада выполняет запись набора записей регистра расчета в информационную базу:

```
НаборЗаписей.Записать (, Истина);
```

Найдем это предложение в процедуре "ОбработкаПроведения" нашего документа и сразу после него напишем код для расчета начислений с категорией "ЗависимоеПервогоУровня":

```
Запрос.Текст =
"ВЫБРАТЬ
|     ОсновныеНачисления.НомерСтроки КАК НомерСтроки,
|     ОсновныеНачисления.ВидРасчета.СпособРасчета КАК
СпособРасчета,
|
|     ЕСТЬNULL (ОсновныеНачисленияБазаОсновныеНачисления.Результат
База, 0) КАК СуммаБазы
```

```
| ИЗ
|   РегистрРасчета.ОсновныеНачисления КАК
ОсновныеНачисления
|   ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.ОсновныеНачисления.БазаОсновныеНачисления (
|   &ИзмеренияОсновногоРегистра,
|   &ИзмеренияБазовогоРегистра,
|   ,
|   Регистратор = &Регистратор
|   И ВидРасчета.КатегорияРасчета =
&КатегорияРасчета) КАК ОсновныеНачисленияБазаОсновныеНачисления
|   ПО ОсновныеНачисления.НомерСтроки =
ОсновныеНачисленияБазаОсновныеНачисления.НомерСтроки
| ГДЕ
|   ОсновныеНачисления.Регистратор = &Регистратор
|   И ОсновныеНачисления.ВидРасчета.КатегорияРасчета =
&КатегорияРасчета
|
| УПОРЯДОЧИТЬ ПО
|   НомерСтроки";

Запрос.УстановитьПараметр ("КатегорияРасчета",
Справочники.КатегорииРасчета.ЗависимоеПервогоУровня);
// массив имен измерений
Измерения = Новый Массив ();
Измерения.Добавить ("Сотрудник");
Измерения.Добавить ("Должность");

Запрос.УстановитьПараметр ("ИзмеренияОсновногоРегистра",
Измерения);
Запрос.УстановитьПараметр ("ИзмеренияБазовогоРегистра",
Измерения);

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

Расчеты.РассчитатьЗаписиРегистраРасчета (НаборОсновныеНачисл
ения, Выборка);

НаборОсновныеНачисления.Записать (, Истина);
```

Структура кода нам уже знакома. Вначале выполняется запрос для получения необходимых данных (суммы базовых начислений) для начислений с категорией "ЗависимоеПервогоУровня" и получается выборка результата запроса. Затем выборка вместе с набором записей передается в процедуру общего модуля, которая должна выполнить расчет. После этого набор записей записывается, снова без пересчета фактического периода действия.

Обратим внимание, что для задания параметров виртуальной таблицы "База" мы использовали объект <Массив>, а, поскольку имена измерений "основного" и "базового" регистров совпадают (это один и тот же регистр), то можно использовать один и тот же экземпляр объекта <Массив> для задания параметров "ИзмеренияОсновногоРегистра" и "ИзмеренияБазовогоРегистра".

Также мы использовали тот же экземпляр объекта <Запрос>, что и для первого случая. Нет необходимости создавать новый экземпляр объекта, можно только поменять текст запроса. К тому же значение параметра "Регистратор" было уже задано, так что нужно только изменить значение параметра "КатегорияРасчета" и задать значения двух новых параметров – списков имен измерений.

Теперь допишем расчетный код процедуры общего модуля: добавим ветвь для расчета по способу "Процентом".

В тело оператора "Если", реализующего расчетные формулы, добавим еще одну ветвь "ИначеЕсли" и запишем в ней выражение для расчетной формулы по способу расчета "Процентом":

```
ИначеЕсли Выборка.СпособРасчета =  
Перечисления.СпособыРасчета.Процентом Тогда  
    Результат = ЗаписьРегистра.Размер * Выборка.СуммаБазы  
    / 100;
```

Снова для выполнения формулы имеются все необходимые операнды: значение процента от базы (исходное данное) – в реквизите "Размер" записи регистра, сумма базовых начислений (необходимое данное) – в позиции выборки, соответствующей рассчитываемой записи регистра.

Приведем полный код модуля документа "РасчетЗарплаты" и код расчетной процедуры общего модуля:

Код процедуры "ОбработкаПроведения":

```
Процедура ОбработкаПроведения (Отказ, РежимПроведения)  
    НаборОсновныеНачисления =  
    Движения.ОсновныеНачисления;  
    ТекстЗапроса =  
    "ВЫБРАТЬ  
    |  
    РасчетЗарплатыОсновныеНачисления.Ссылка.ПериодРегистрации,  
    |     РасчетЗарплатыОсновныеНачисления.Сотрудник,  
    |     РасчетЗарплатыОсновныеНачисления.Должность,
```

Реализация расчетных алгоритмов сеанса расчета регистра
"Основные Начисления"

```
| РасчетЗарплатыОсновныеНачисления.ВидРасчета,  
|  
РасчетЗарплатыОсновныеНачисления.ПериодДействияНачало,  
|  
КОНЕЦПЕРИОДА (РасчетЗарплатыОсновныеНачисления.ПериодДействи  
яКонец, ДЕНЬ) КАК ПериодДействияКонец,  
|  
РасчетЗарплатыОсновныеНачисления.БазовыйПериодНачало,  
|  
| ВЫБОР  
|  
| КОГДА  
РасчетЗарплатыОсновныеНачисления.БазовыйПериодКонец =  
ДАТАВРЕМЯ(1, 1, 1)  
|  
| ТОГДА ДАТАВРЕМЯ(1, 1, 1)  
|  
| ИНАЧЕ  
КОНЕЦПЕРИОДА (РасчетЗарплатыОсновныеНачисления.БазовыйПериодКонец  
, ДЕНЬ)  
|  
| КОНЕЦ КАК БазовыйПериодКонец,  
|  
| РасчетЗарплатыОсновныеНачисления.Размер,  
|  
| РасчетЗарплатыОсновныеНачисления.ГрафикРаботы,  
|  
| РасчетЗарплатыОсновныеНачисления.Сторно  
| ИЗ  
|  
| Документ.РасчетЗарплаты.ОсновныеНачисления КАК  
РасчетЗарплатыОсновныеНачисления  
| ГДЕ  
|  
| РасчетЗарплатыОсновныеНачисления.Ссылка =  
&Ссылка  
|  
| УПОРЯДОЧИТЬ ПО  
|  
| РасчетЗарплатыОсновныеНачисления.НомерСтроки";  
  
Расчеты.ЗаполнитьНаборЗаписей(ТекстЗапроса, Ссылка,  
НаборОсновныеНачисления);  
  
НаборОсновныеНачисления.Записать(, Ложь);  
  
Запрос = Новый Запрос();  
Запрос.Текст =  
"ВЫБРАТЬ  
|  
| ОсновныеНачисления.НомерСтроки КАК НомерСтроки,  
|  
| ОсновныеНачисления.ВидРасчета.СпособРасчета КАК  
СпособРасчета,  
|  
| ЕСТЬNULL (ОсновныеНачисленияДанныеГрафика.ЗначениеПериодДейс
```

```

твия, 0) КАК План,
    |
    ЕСТЬNULL (ОсновныеНачисленияДанныеГрафика.ЗначениеФактически
йПериодДействия, 0) КАК Отработано
    | ИЗ
    |     РегистрРасчета.ОсновныеНачисления КАК
ОсновныеНачисления
    |     ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.ОсновныеНачисления.ДанныеГрафика (
    |     Регистратор = &Регистратор
    |     И
ВидРасчета.КатегорияРасчета = &КатегорияРасчета) КАК
ОсновныеНачисленияДанныеГрафика
    |     ПО ОсновныеНачисления.НомерСтроки =
ОсновныеНачисленияДанныеГрафика.НомерСтроки
    | ГДЕ
    |     ОсновныеНачисления.Регистратор = &Регистратор
    |     И ОсновныеНачисления.ВидРасчета.КатегорияРасчета
= &КатегорияРасчета";

    Запрос.УстановитьПараметр ("Регистратор", Ссылка);
Запрос.УстановитьПараметр ("КатегорияРасчета",
Справочники.КатегорииРасчета.Первичное);

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

    Расчеты.РассчитатьЗаписиРегистраРасчета (НаборОсновные
Начисления, ВыборкаДляРасчета);
    НаборОсновныеНачисления.Записать (, Истина);

    Запрос.Текст =
    "ВЫБРАТЬ
    |     ОсновныеНачисления.НомерСтроки КАК НомерСтроки,
    |     ОсновныеНачисления.ВидРасчета.СпособРасчета КАК
СпособРасчета,
    |
    ЕСТЬNULL (ОсновныеНачисленияБазаОсновныеНачисления.Результат
База, 0) КАК СуммаБазы
    | ИЗ
    |     РегистрРасчета.ОсновныеНачисления КАК
ОсновныеНачисления
    |     ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.ОсновныеНачисления.БазаОсновныеНачисления (

```

*Реализация расчетных алгоритмов сеанса расчета регистра
"Основные Начисления"*

```
|
|      &ИзмеренияОсновногоРегистра,
|      &ИзмеренияБазовогоРегистра,
|      ,
|      Регистратор = &Регистратор
|      И
ВидРасчета.КатегорияРасчета = &КатегорияРасчета) КАК
ОсновныеНачисленияБазаОсновныеНачисления
|      ПО ОсновныеНачисления.НомерСтроки =
ОсновныеНачисленияБазаОсновныеНачисления.НомерСтроки
| ГДЕ
|      ОсновныеНачисления.Регистратор = &Регистратор
|      И ОсновныеНачисления.ВидРасчета.КатегорияРасчета
= &КатегорияРасчета
|
| УПОРЯДОЧИТЬ ПО
|      НомерСтроки";
|
|      Запрос.УстановитьПараметр ("КатегорияРасчета",
Справочники.КатегорииРасчета.ЗависимоеПервогоУровня);
|      // массив имен измерений
Измерения = Новый Массив ();
Измерения.Добавить ("Сотрудник");
Измерения.Добавить ("Должность");
|
|      Запрос.УстановитьПараметр ("ИзмеренияОсновногоРегистра
", Измерения);
|      Запрос.УстановитьПараметр ("ИзмеренияБазовогоРегистра"
, Измерения);
|
|      Выборка = Запрос.Выполнить ().Выбрать ();
|
|      Расчеты.РассчитатьЗаписиРегистраРасчета (НаборОсновные
Начисления, Выборка);
|
|      НаборОсновныеНачисления.Записать (, Истина);
|
|      КонецПроцедуры
```

Код процедуры "РассчитатьЗаписиРегистраРасчета" общего модуля:

```
Процедура РассчитатьЗаписиРегистраРасчета (НаборЗаписей,  
Выборка) Экспорт  
    Результат = 0; // создаем промежуточную переменную  
    // Обходим в цикле выборку необходимых данных:  
    Пока Выборка.Следующий() Цикл  
        // Получаем рассчитываемую запись регистра  
        // из набора записей по индексу:  
        ЗаписьРегистра =  
НаборЗаписей[Выборка.НомерСтроки - 1];  
        // Реализуем расчетные формулы  
        // для каждого имеющегося способа расчета;  
        // результат запоминаем в  
        // промежуточной переменной Результат:  
        Если Выборка.СпособРасчета =  
Перечисления.СпособыРасчета.ПоОкладу Тогда  
            Если Выборка.План > 0 Тогда  
                Результат = ЗаписьРегистра.Размер *  
Выборка.Отработано / Выборка.План;  
            КонецЕсли;  
            ИначеЕсли Выборка.СпособРасчета =  
Перечисления.СпособыРасчета.Процентом Тогда  
                Результат = ЗаписьРегистра.Размер *  
Выборка.СуммаБазы / 100;  
            Иначе  
                Результат = 0;  
            КонецЕсли;  
        // Помещаем результат в запись регистра:  
        ЗаписьРегистра.Результат = Результат;  
    КонецЦикла;  
КонецПроцедуры
```

Выполнение расчетного кода

Обновим конфигурацию базы данных и запустим на выполнение. Проведем документ расчета. В регистре расчета "ОсновныеНачисления" получим, в дополнение к уже имеющимся результатам расчета первичных начислений результат начисления, зависящего по базе:

Номер строки	Вид расчета	Сотрудник	Должность	Результат
1	Оклад	Ковалев Александр Петрович	Директор	70 000,00
2	Оклад	Троицкая Татьяна Сергеевна	Бухгалтер	44 347,83
3	Оклад	Ильин Сергей Викторович	Менеджер	50 000,00
4	Оклад на выезде	Троицкая Татьяна Сергеевна	Бухгалтер	19 565,22
5	Доплата за квалификацию	Троицкая Татьяна Сергеевна	Бухгалтер	6 391,31

В строке 5 мы видим результат расчета по способу Процентом: 6391,31 руб. Это те самые 10% от суммы базы для вида расчета Доплата за квалификацию, указанные в поле Размер строки табличной части документа.

Итак, мы уже умеем получать необходимые данные из регистра расчета, используя две его виртуальные таблицы: "ДанныеГрафика" и "База".

Практикум № 4

Реализуйте способ расчета: "ВечернееДежурство" (работник после рабочего дня остается дежурить вечером, продолжительность дежурства при расчете не учитывается). Расчетная формула:

[Сумма базовых начислений] * 1.10

(т.е. сумма базы + 10%). Базовые виды расчета: "Оклад", "По тарифу").

Предполагается, что сумма базы берется пропорционально количеству рабочих дней, в которые сотрудник дежурил.

Создайте вид расчета "Вечернее дежурство", введите и рассчитайте его для произвольного сотрудника за произвольный интервал (или несколько интервалов) дежурства.

Использование периода действия в качестве базового периода

Когда мы вводили в документ вид расчета "Доплата за квалификацию" для г-жи Троицкой, мы, вероятно, обратили внимание, что вводятся и границы периода действия, и границы базового периода.

Период действия для этого вида расчета, вообще говоря, не нужен – ведь он рассчитывается процентом от базы и его расчетная формула не включает данные графика.

Но отказаться от ввода периода действия основного начисления мы не можем. Дело в том, что регистр расчета "Основные Начисления" имеет функционал получения данных графика (свойство "Период действия" регистра расчета установлено), а в этом случае платформа требует обязательного задания границ периода действия для каждой записи регистра – иначе возникнет ошибка при записи набора в информационную базу.

Но есть способ избежать ввода одинаковых значений границ интервала периода действия и интервала базового периода.

С одной стороны, ввод границ базового периода платформой не контролируется и при записи набора с незаполненными границами базового периода ошибки не возникает. С другой стороны, у видов расчета, принадлежащих плану видов расчета Основные начисления, есть стандартный реквизит: "ПериодДействияБазовый", который имеет тип <Булево> и на форме вида расчета обозначен подписью "Базовый период как период действия".

Если этот реквизит установить, то механизм регистра расчета получения суммы базовых начислений будет использовать заданные границы периода действия в качестве границ базового периода. При этом значения, заданные в полях "БазовыйПериодНачало" и "БазовыйПериодКонец" записей регистра будут игнорироваться.

Таким образом, если для вида расчета, который требует получения суммы базовых начислений, установить свойство "ПериодДействияБазовый" в значение Истина, то для указания границ базового периода можно будет использовать период действия, а поля границ базового периода можно оставлять пустыми.

В режиме "1С:Предприятие" откроем форму вида расчета "Доплата за квалификацию" и установим флажок "Базовый период как период действия":

Доплата за квалификацию (Основные начисления)

Записать и закрыть Записать Еще ▾

Код: 00003

Наименование: Доплата за квалификацию

Базовый период как период действия:

Способ расчета: Процентом ▾

Категория расчета: Зависимое первого уровня ▾

Базовые виды расчетов Ведущие виды расчетов Вытесняющие виды расчетов

Добавить Еще ▾

Вид расчета
Оклад
Оклад на выезде

Теперь откроем наш документ "Расчет зарплаты" и в строке табличной части "Доплата за квалификацию" для г-жи Троицкой очистим поля границ базового периода:

*Реализация расчетных алгоритмов сеанса расчета регистра
"Основные Начисления"*

Вид расчета	Период действия начало	Период действия конец	Базовый период начало	Базовый период конец
Оклад	01.01.2014 0:00:00	31.01.2014 0:00:00		
Оклад	01.01.2014 0:00:00	31.01.2014 0:00:00		
Оклад	01.01.2014 0:00:00	31.01.2014 0:00:00		
Оклад на выезде	20.01.2014 0:00:00	27.01.2014 0:00:00		
Доплата за квалификацию	01.01.2014 0:00:00	31.01.2014 0:00:00		

Проведем документ. Несмотря на то, что границы базового периода в явном виде не заданы, мы все равно получили тот же результат расчета вида "Доплата за квалификацию":

Номер строки	Вид расчета	Сотрудник	Должность	Результат	Дата начала периода действия	Дата окончания периода действия	Дата начала базового периода	Дата окончания базового периода
	Оклад	Ковалев Александр Петрович	Директор	70 000,00	01.01.2014 0:00:00	31.01.2014 23:59:59		
	Оклад	Троицкая Татьяна Сергеевна	Бухгалтер	44 347,83	01.01.2014 0:00:00	31.01.2014 23:59:59		
	Оклад	Ильин Сергей Викторович	Менеджер	50 000,00	01.01.2014 0:00:00	31.01.2014 23:59:59		
	Оклад на выезде	Троицкая Татьяна Сергеевна	Бухгалтер	19 565,22	20.01.2014 0:00:00	27.01.2014 23:59:59		
	Доплата за квалификацию	Троицкая Татьяна Сергеевна	Бухгалтер	6 391,31	01.01.2014 0:00:00	31.01.2014 23:59:59		

9. Наличие у регистра расчета нескольких базовых регистров

Создадим принципиально новую расчетную ситуацию – когда у регистра расчета имеется в наличии несколько базовых регистров, следовательно, несколько виртуальных таблиц "База". Принципиальной сложности решения расчетной задачи это не создает, однако возникает особенность построения запроса для получения необходимых данных, которую следует рассмотреть и изучить.

9.1. Техническое задание, вторая очередь

Поставим задачу реализовать следующий состав видов расчета, который можно объединить общим названием "Премии":

Вид расчета	Способ расчета		Зависимости	
	Формула	Имя	Базовые ВР	Базовый период
Персональная премия	$[Результат] = [Размер]$	Фиксированно		
Месячная премия	$[Результат] = [Размер] * [СуммаБазы] / 100$	Процентом	1. Персональная премия 2. Оклад 3. Оклад на выезде	Расчетный месяц
Поощрительная надбавка	$[Результат] = [Размер] * [СуммаБазы] / 100$	Процентом	Месячная премия	Расчетный месяц
Премия за период	$[Результат] = [Размер] * [СуммаБазы] / 100$	Процентом	Оклад	Переменный, в пределах расчетного месяца

Расчетная формула способа расчета "Фиксировано" совсем проста: результат расчета равен исходному значению размера. Значит, этот способ расчета не требует получения необходимых данных.

А способ расчета "Процентом" нам уже знаком.

Обратим внимание на следующую особенность второй очереди технического задания.

Ни одна расчетная формула не содержит в качестве операндов ни плановое количество рабочего времени, ни количество фактически отработанного времени.

Следовательно, на примере второй очереди технического задания можно изучить наличие нескольких регистров расчета, создав для этих видов расчета новый регистр и новый план видов расчета.

9.2. Создание второй пары расчетных объектов

9.2.1. План видов расчета "Премии"

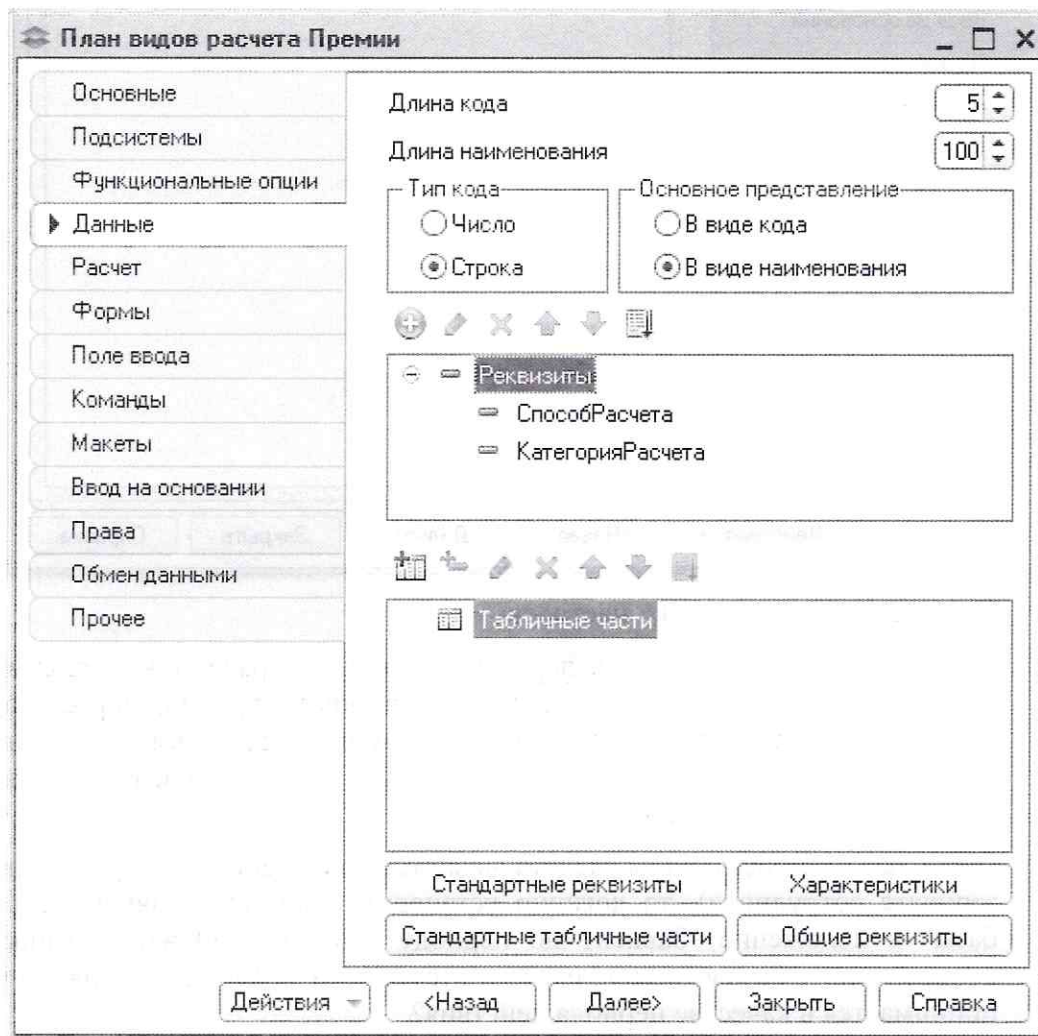
Добавим в конфигурацию новый план видов расчета, дадим ему имя "Премии". Свяжем его с подсистемой "ПланыВидовРасчета" из раздела "Расчет".

Также, как и для плана видов расчета "ОсновныеНачисления", на закладке "Данные" установим длину кода 5, длину наименования 100.

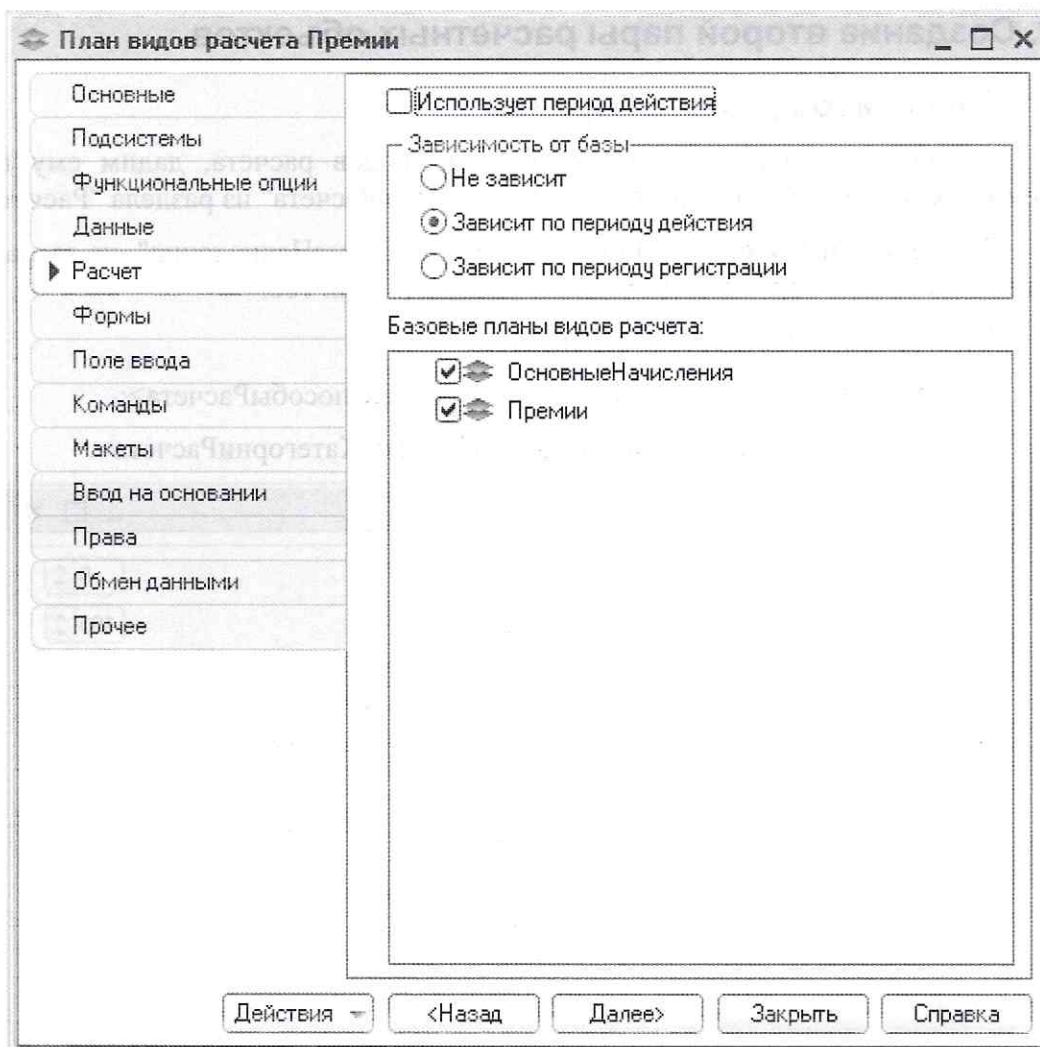
Создадим такой же состав реквизитов:

"СпособРасчета", тип <ПеречислениеСсылка.СпособыРасчета>;

"КатегорияРасчета", тип <СправочникСсылка.КатегорииРасчета>.



На закладке "Расчет" установим следующие настройки:



Дадим обоснование таких настроек.

1. Поскольку ни одна расчетная формула новых видов расчета не использует в качестве операнда данные графика, то наш новый план видов расчета не будет использовать период действия. Напомним: это значит, что у него будет отсутствовать стандартная табличная часть "ВытесняющиеВидыРасчета".
2. Поскольку все новые виды расчета являются начислениями (идут "в плюс" зарплате сотрудника), то логично установить свойство "Зависимость от базы" в положение "Зависит по периоду действия". Однако возникает вопрос: если у базового вида расчета отсутствует период действия, то что принимается в качестве периода действия?

Важно!

Если у базового вида расчета период действия отсутствует, то при определении попадания его результата в сумму базовых начислений считается, что **период действия равен периоду регистрации**

3. У вида расчета "Месячная премия" базовые виды расчета находятся в разных планах видов расчета: "Персональная премия" – в "Премии", а "Оклад" и "Оклад на выезде" – в "ОсновныеНачисления". Следовательно,

нужно указать оба плана видов расчета в качестве базовых для плана видов расчета "Премии".

Аналогичным образом, как для плана видов расчета "Основные Начисления", создадим формы: форму списка и форму вида расчета.

При создании формы вида расчета обратим внимание, что создаются только два табличных поля на двух закладках формы – табличная часть "Вытесняющие Виды Расчета" отсутствует.

9.2.2. Заполнение плана видов расчета "Премии"

Запустим нашу конфигурацию в режиме "1С:Предприятие", откроем план видов расчета "Премии" и внесем туда виды расчета в соответствии со следующей таблицей:

Вид расчета	Способ расчета	Категория расчета	Базовые виды расчета	Базовый план видов расчета
Персональная премия	Фиксировано	Первичное		
Месячная премия	Процентом	Зависимое первого уровня	Персональная премия	Премии
			Оклад	Основные начисления
			Оклад на выезде	Основные начисления
Премия за период	Процентом	Зависимое первого уровня	Оклад	Основные начисления
Поощрительная надбавка	Процентом	Зависимое второго уровня	Месячная премия	Премии

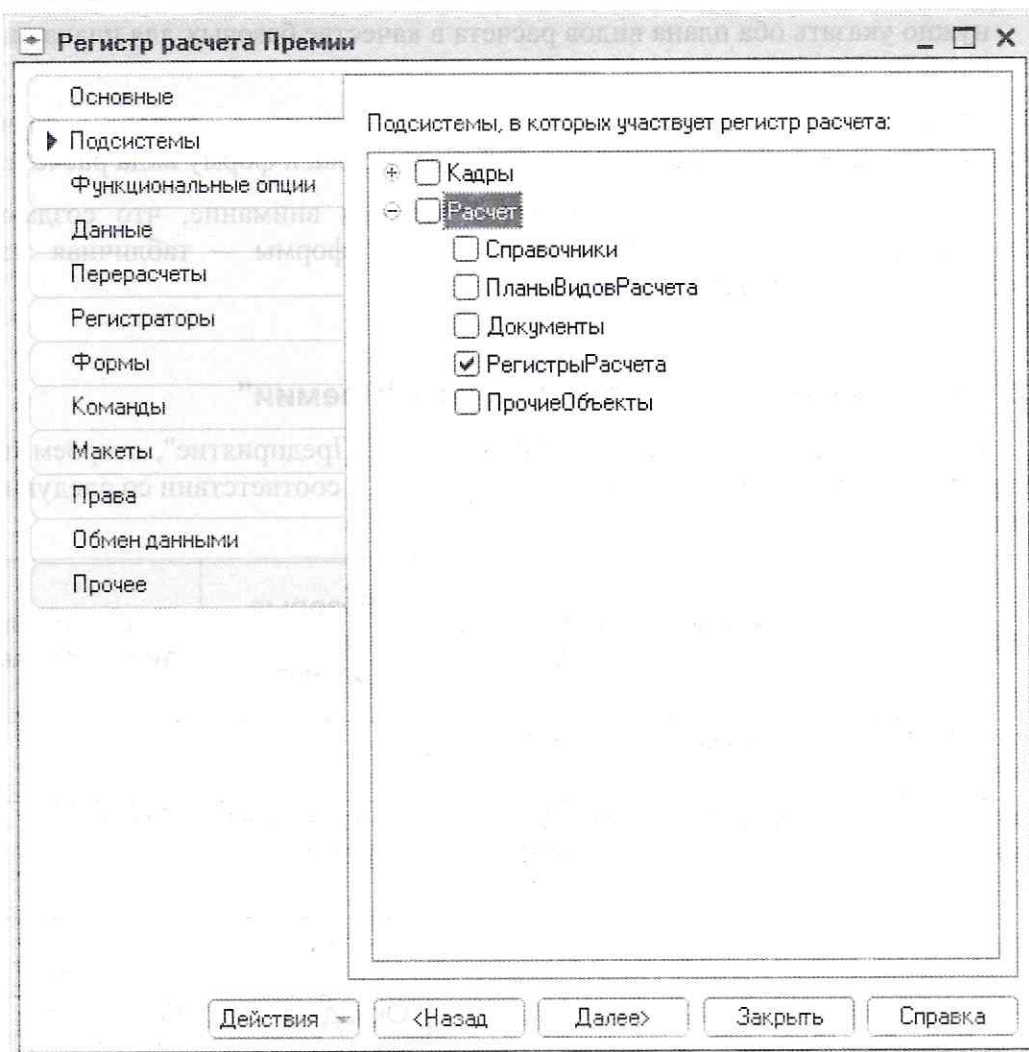
Отметим, что используются базовые виды расчета из двух планов видов расчета: "Основные начисления" и "Премии".

Вид расчета "Поощрительная надбавка" имеет категорию расчета "Зависимое второго уровня", поскольку зависит по базе от вида расчета "Месячная премия" (зависимое первого уровня).

9.2.3. Регистр расчета "Премии"

Добавим в конфигурацию новый регистр расчета, назовем его "Премии".

Свяжем его с подсистемой "РегистрыРасчета" раздела "Расчет":



А на закладке "Основные" сделаем следующие настройки:

Регистр расчета Премии

Основные

Имя: Премии

Синоним: Премии

Комментарий:

План видов расчета: Премии

Период действия

График:

Значение графика:

Дата графика:

Базовый период

Периодичность: Месяц

Представление списка:

Расширенное представление списка:

Пояснение:

Действия <Назад Далее> Закрыть Справка

Поясним установленные значения свойств.

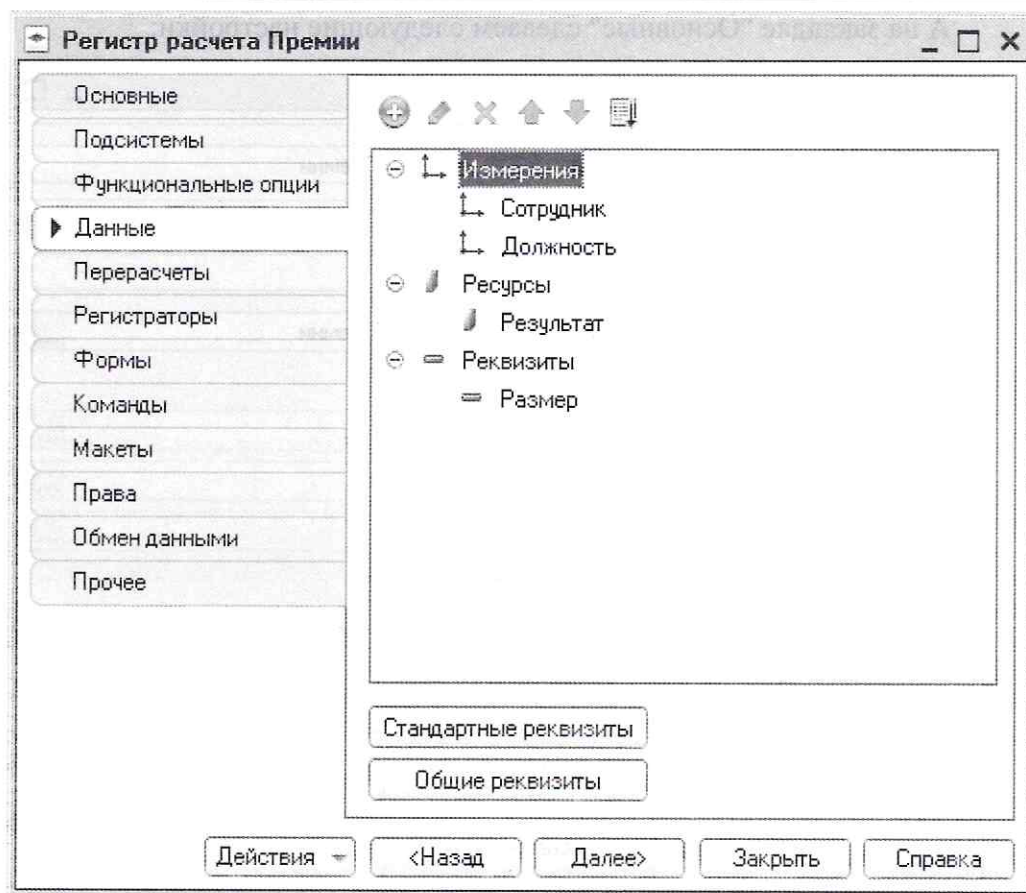
Новый регистр расчета предназначен для расчета начислений, которые содержатся в плане видов расчета "Премии", поэтому мы связали регистр с планом видов расчета "Премии".

Ни один из этих видов расчета не требует для реализации своей расчетной формулы данных графика, следовательно, свойство "Период действия" регистра мы не устанавливали и, соответственно, регистр сведений для получения данных графика не задавали.

Виды расчета из плана видов расчета "Премии" в основном рассчитываются с использованием расчетной базы, следовательно, мы установили свойство "Базовый период" регистра расчета.

Регистр будет рассчитывать месячные премии, значит, оставляем значение свойства "Периодичность" по умолчанию – "Месяц".

Состав прикладных полей регистра расчета "Премии" будет таким:



Т.е., состав измерений и ресурсов будет совпадать с составом этих полей для регистра расчета "ОсновныеНачисления", а из реквизитов будет только один – "Размер", поскольку новый регистр расчета не обладает функциональностью получения данных графика, и реквизит "ДанныеГрафика" ему не нужен.

Типы прикладных полей – те же самые, что и для регистра "ОсновныеНачисления":

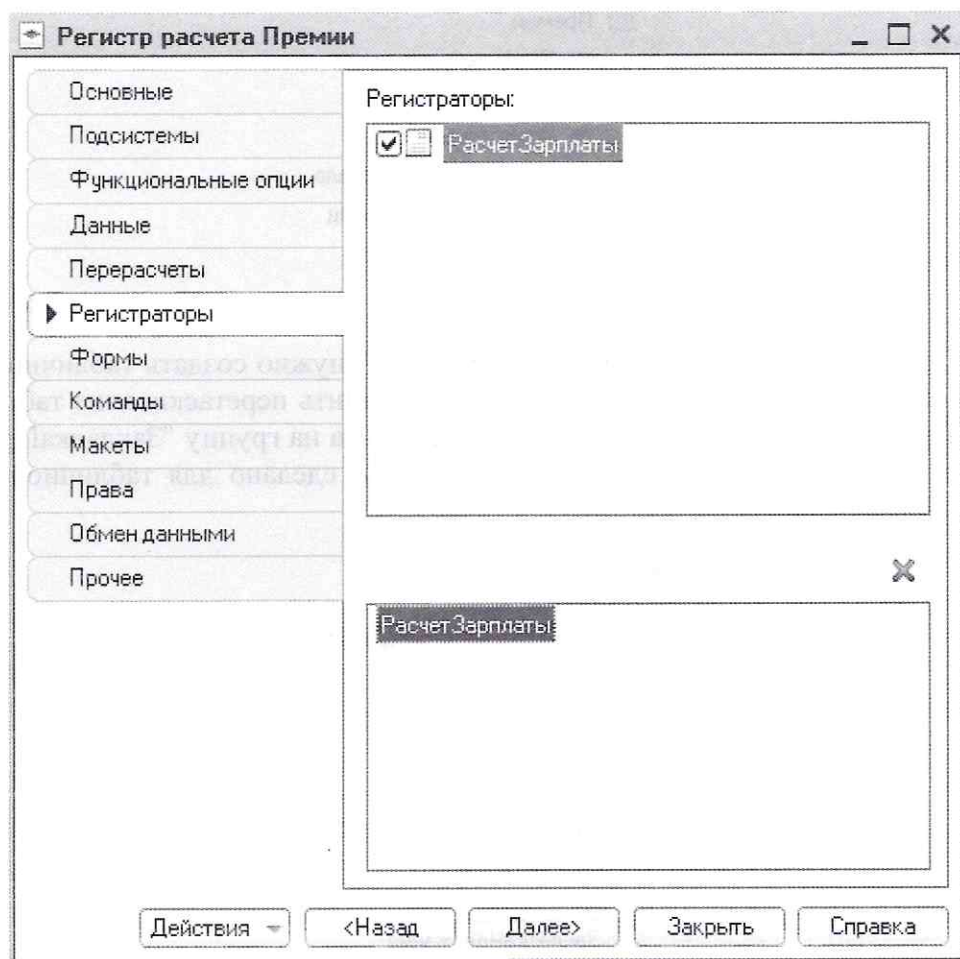
"Сотрудник": <СправочникСсылка.Сотрудники>;

"Должность": <СправочникСсылка.Должности>;

"Результат": <Число (10, 2)>;

"Размер": <Число (10, 2)>.

Регистратором для нового регистра расчета назначим тот же документ – "РасчетЗарплаты":



Настройка второй пары расчетных объектов завершена.

9.2.4. Доработка структуры документа "НачислениеЗарплаты"

В структуру документа добавим новую табличную часть – "Премии". Состав и типы реквизитов новой табличной части будут такими:

Имя реквизита	Тип
Сотрудник	<СправочникСсылка.Сотрудники>
Должность	<СправочникСсылка.Должности>
ВидРасчета	<ПланВидовРасчетаСсылка.Премии>
БазовыйПериодНачало	<Дата>
БазовыйПериодКонец	<Дата>
Размер	<Число (10, 2)>
Сторно	<Булево>

Количество реквизитов у новой табличной части будет меньше, чем у табличной части "ОсновныеНачисления". Не будет реквизитов "ПериодДействияНачало" и "ПериодДействияКонец", а также реквизита "ГрафикРаботы" – по той причине, что для расчета премий данные графика использоваться не будут.

Структура табличной части на закладке "Данные" будет выглядеть так:

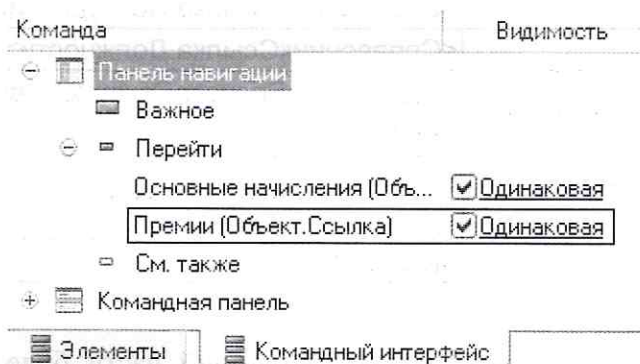
- ☰ Премии
 - ☞ Сотрудник
 - ☞ Должность
 - ☞ ВидРасчета
 - ☞ БазовыйПериодНачало
 - ☞ БазовыйПериодКонец
 - ☞ Размер
 - ☞ Сторно

После создания табличной части "Премии" нужно создать табличное поле на форме документа для ее отображения – выполнить перетаскивание табличной части "Премии" из структуры реквизитов документа на группу "ЗакладкаПремии" в перечне элементов формы, как это ранее было сделано для табличной части "ОсновныеНачисления":

- ☰ Форма
 - ☰ Командная панель
 - ☑ ФормаОтменаПроведения
 - ☞ Номер
 - ☞ Дата
 - ☞ ПериодРегистрации
 - ☰ Начисления
 - ☞ ЗакладкаОсновныеНачисления
 - ☞ ЗакладкаПремии
 - ☞ Премии
 - ☞ ЗакладкаУдержания

Теперь после обновления конфигурации форма документа будет содержать табличное поле для заполнения табличной части "Премии".

В довершение сделаем, чтобы набор записей регистра "Премии" можно было видеть из формы документа. На закладке "Командный интерфейс" редактора формы раскроем группу "Перейти" и установим флажок "Видимость" напротив ветви "Премии":



9.2.5. Код первого шага сеанса расчета

Вспомним, что на первом шаге сеанса расчета для каждого регистра расчета необходимо создать набор записей с исходными данными для расчета.

Поместим код первого сеанса для нашего нового регистра сразу же под кодом первого сеанса регистра "ОсновныеНачисления".

Откроем модуль объекта документа "РасчетЗарплаты", найдем вызов метода "Записать" набора записей регистра "ОсновныеНачисления" со вторым параметром Ложь:

```
НаборОсновныеНачисления.Записать (, Ложь);
```

И сразу же после этой строчки напишем следующий код:

```
НаборПремии = Движения.Премии;  
ТекстЗапроса =  
"ВЫБРАТЬ  
| РасчетЗарплатыПремии.Ссылка.ПериодРегистрации,  
| РасчетЗарплатыПремии.Сотрудник,  
| РасчетЗарплатыПремии.Должность,  
| РасчетЗарплатыПремии.ВидРасчета,  
| РасчетЗарплатыПремии.БазовыйПериодНачало,  
| ВЫБОР  
| КОГДА РасчетЗарплатыПремии.БазовыйПериодКонец =  
ДАТАВРЕМЯ(1, 1, 1)  
| ТОГДА  
РасчетЗарплатыПремии.БазовыйПериодКонец  
| ИНАЧЕ  
КОНЕЦПЕРИОДА(РасчетЗарплатыПремии.БазовыйПериодКонец, ДЕНЬ)  
| КОНЕЦ КАК БазовыйПериодКонец,  
| РасчетЗарплатыПремии.Размер,  
| РасчетЗарплатыПремии.Сторно  
| ИЗ  
| Документ.РасчетЗарплаты.Премии КАК  
РасчетЗарплатыПремии  
| ГДЕ  
| РасчетЗарплатыПремии.Ссылка = &Ссылка  
|  
| УПОРЯДОЧИТЬ ПО  
| РасчетЗарплатыПремии.НомерСтроки";  
  
Расчеты.ЗаполнитьНаборЗаписей(ТекстЗапроса, Ссылка,  
НаборПремии);  
НаборПремии.Записать (, Ложь);
```

Код первого этапа для регистра "Премии" очень похож на код первого этапа для регистра "ОсновныеНачисления". Мы выделяем для удобства в отдельную переменную "НаборПремии" набор записей регистра "Премии",

составляем текст запроса по табличной части "Премии" и вызываем процедуру общего модуля "ЗаполнитьНаборЗаписей", после чего записываем набор записей со вторым параметром Ложь.

Текст запроса по табличной части тоже не сильно отличается от варианта для основных начислений – только по понятной причине в результате отсутствуют поля "ПериодДействияНачало", "ПериодДействияКонец" и "ГрафикРаботы".

Здесь может возникнуть вопрос: а какая разница для регистра "Премии", с каким значением второго параметра записывать набор записей – ведь этот регистр не связан с периодом действия? Да, вообще говоря, в этом случае разницы нет. Но запись исходных данных со вторым параметром Ложь является хорошим стилем разработки. Может потребоваться впоследствии придать регистру расчета функционал получения данных графика, и тогда, если мы не будем придерживаться этого правила, нам придется исправлять код программы, возможно, в нескольких местах. Поэтому лучше сразу подстраховаться – хуже не будет, если мы всегда будем записывать исходные данные регистра расчета со вторым параметром Ложь.

9.2.6. Ввод и запись вида расчета "Персональная премия"

Обновим конфигурацию базы данных, запустим ее в режиме "1С:Предприятие" и в табличное поле на закладке "Премии" введем такие строки:

Сотрудник	Должность	Вид расчета	Базовый период начало	Базовый период конец	Размер
Ильин Сергей Викторович	Менеджер	Персональная премия			10 000,00
Троицкая Татьяна Сергеевна	Бухгалтер	Персональная премия			10 000,00

Отметим, что, поскольку для способа расчета "Фиксированно" получение необходимых данных не требуется, поля границ базового периода можно оставить пустыми – эти значения не оказывают никакого влияния на результат расчета по способу "Фиксированно".

Проведем документ, нажмем в форме документа на надпись "Премии" вверху формы:

Расчет зарплаты 000000001

Главное Основные начисления Премии

Провести и закрыть Записать Провести

Номер: 000000001

Откроется список записей регистра "Премии", и мы увидим следующие результаты:

Вид расчета	Сотрудник	Должность	Результат	Размер
Персональная премия	Ильин Сергей Викторович	Менеджер	0	10 000,00
Персональная премия	Троицкая Татьяна Сергеевна	Бухгалтер	0	10 000,00

Итак, первый шаг сеанса расчета выполнен – строки регистра "Премии" с исходными данными сформированы и записаны.

9.2.7. Реализация расчетного кода способа расчета "Фиксированно"

Реализация расчетной формулы

Расчетная формула способа расчета "Фиксированно" является вырожденной: суммой начисления является просто исходное значение поля "Размер", т.е. сумма начисления известна уже до начала расчетного сеанса.

Реализуем расчетную формулу этого способа расчета. Откроем общий модуль Расчеты, в оператор "Если..." процедуры "РассчитатьЗаписиРегистраРасчета" добавим еще одну ветвь:

```
ИначеЕсли Выборка.СпособРасчета =
Перечисления.СпособыРасчета.Фиксированно Тогда
    Результат = ЗаписьРегистра.Размер;
```

Таким образом, код процедуры "РассчитатьЗаписиРегистраРасчета" теперь будет таким:

```
Процедура РассчитатьЗаписиРегистраРасчета (НаборЗаписей,
Выборка) Экспорт
    Результат = 0; // создаем промежуточную переменную
    // Обходим в цикле выборку необходимых данных:
    Пока Выборка.Следующий() Цикл
        // Получаем рассчитываемую запись регистра
        // из набора записей по индексу:
        ЗаписьРегистра =
НаборЗаписей[Выборка.НомерСтроки - 1];
        // Реализуем расчетные формулы
        // для каждого имеющегося способа расчета;
        // результат запоминаем в
        // промежуточной переменной Результат:
        Если Выборка.СпособРасчета =
Перечисления.СпособыРасчета.ПоОкладу Тогда
            Если Выборка.План > 0 Тогда
                Результат = ЗаписьРегистра.Размер *
Выборка.Отработано / Выборка.План;
            КонецЕсли;
```

```
ИначеЕсли Выборка.СпособРасчета =
Перечисления.СпособыРасчета.Процентом Тогда
    Результат = ЗаписьРегистра.Размер *
Выборка.СуммаБазы / 100;
ИначеЕсли Выборка.СпособРасчета =
Перечисления.СпособыРасчета.Фиксированно Тогда
    Результат = ЗаписьРегистра.Размер;
Иначе
    Результат = 0;
КонецЕсли;
// Помещаем результат в запись регистра:
ЗаписьРегистра.Результат = Результат;
КонецЦикла;
КонецПроцедуры
```

Запрос для получения необходимых данных и расчет

Как мы уже отмечали, необходимые данные для способа расчета "Фиксированно" в принципе не нужны, поскольку результат расчета известен заранее. Но мы должны следовать разработанной нами единой схеме расчетного алгоритма: для каждой категории расчета должен быть выполнен запрос по регистру расчета, в результате которого обязательно должно быть поле "НомерСтроки" и поле "СпособРасчета" – этого требует наша универсальная процедура "РассчитатьЗаписиРегистраРасчета".

Так что придется разработать и выполнить запрос и для такого простого способа расчета, как "Фиксировано".

Итак, вспомним: для каждой категории расчета нужно выполнить запрос для получения необходимых данных, передать его выборку вместе с рассчитываемым набором записей в процедуру "РассчитатьЗаписиРегистраРасчета", которая реализует расчетные формулы, а потом записать результат расчета в информационную базу.

Код для расчета регистра "Премии" можно начать писать сразу же после кода расчета регистра "ОсновныеНачисления" всех уровней, т.е. фактически в самом конце процедуры "ОбработкаПроведения" модуля документа. Так мы и сделаем.

Однако есть одно перспективное предложение: оформить код расчета премий в виде отдельной процедуры, которой будет передаваться значение категории расчета. Мы назовем процедуру "РассчитатьПремии". Мы вскоре увидим, что при таком подходе нас ожидает приятный сюрприз...

Запишем вызов этой, пока не написанной еще, процедуры, в конце процедуры "ОбработкаПроведения" для расчета премий с категорией "Первичное" – как раз эту категорию зависимости имеет введенный нами вид расчета Персональная премия:

Процедура ОбработкаПроведения (Отказ, РежимПроведения)

.....

РассчитатьПремии (НаборПремии, Запрос,
Справочники. КатегорииРасчета. Первичное);

КонецПроцедуры

Процедуре будут передаваться набор записей "Премии", объект <Запрос>, который у нас уже работал для получения необходимых данных регистра "ОсновныеНачисления", и значение категории расчета.

Важно!

Конечно, более рационально было бы вынести переменные набора записей и запроса на уровень программного модуля, чтобы они были "видны" всем процедурам этого модуля, тогда в процедуру надо было бы передать только один аргумент - категорию расчета. Но мы на в данный момент не будем уделять этому внимания.

А после процедуры "ОбработкаПроведения" напишем саму процедуру "РассчитатьПремии":

Процедура РассчитатьПремии (НаборЗаписей, Запрос,
КатегорияРасчета)

Запрос.Текст =

"ВЫБРАТЬ

| Премии.НомерСтроки КАК НомерСтроки,

| Премии.ВидРасчета.СпособРасчета КАК

СпособРасчета

| ИЗ

| РегистрРасчета.Премии КАК Премии

| ГДЕ

| Премии.Регистратор = &Регистратор

| И Премии.ВидРасчета.КатегорияРасчета =

&КатегорияРасчета

|

| УПОРЯДОЧИТЬ ПО

| НомерСтроки";

Запрос.УстановитьПараметр ("КатегорияРасчета",
КатегорияРасчета);

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

Расчеты.РассчитатьЗаписиРегистраРасчета (НаборЗаписей,
Выборка);

НаборЗаписей.Записать(, Истина);

КонецПроцедуры

Даже без Консоли запросов видно, что запрос для передачи в расчетную процедуру пока крайне прост: он выбирает из регистра "Премии" только поля "НомерСтроки" и "ВидРасчета.СпособРасчета" для записей указанного регистратора и указанной категории расчета – они передаются через параметры. (Напомним, что в процедуру передается уже частично настроенный объект <Запрос>, в котором значение параметра "&Регистратор" уже было установлено при работе с основными начислениями, поэтому процедура устанавливает только значение параметра "&КатегорияРасчета").

Выборка выполненного запроса вместе с набором записей передается расчетной процедуре общего модуля, которая рассчитывает записи набора, после чего выполняется запись набора с полученными результатами в информационную базу.

9.2.8. Расчет начислений "Персональная премия"

Обновим конфигурацию базы данных, запустим ее в режиме "1С:Предприятие", откроем форму документа и проведем его. Как мы это делали выше, посмотрим в набор записей регистра "Премии" и увидим результаты расчета начисления "Персональная премия":

Вид расчета	Сотрудник	Должность	Результат	Размер
Персональная премия	Ильин Сергей Викторович	Менеджер	10 000,00	10 000,00
Персональная премия	Троицкая Татьяна Сергеевна	Бухгалтер	10 000,00	10 000,00

Результаты расчета премий с категорией "Первичное" получены.

9.2.9. Ввод и запись вида расчета "Месячная премия"

Вид расчета Месячная премия имеет категорию расчета "Зависимое первого уровня", поскольку зависит по базе от результатов видов расчета с категорией Первичное: "Оклад" и "Оклад на выезде" (из состава основных начислений) и "Персональная премия" (из состава премий).

Запустим конфигурацию в режиме "1С:Предприятие", откроем форму документа "Расчет зарплаты" и добавим в табличную часть "Премии" следующие строки:

Сотрудник	Должность	Вид расчета	Базовый период начало	Базовый период конец	Размер
Ильин Сергей Викторович	Менеджер	Месячная премия	01.01.2014 0:00:00	31.01.2014 0:00:00	20,00
Троицкая Татьяна Сергеевна	Бухгалтер	Месячная премия	01.01.2014 0:00:00	31.01.2014 0:00:00	25,00

Проведем документ. В регистре "Премии" увидим следующее:

Номер строки	Вид расчета	Сотрудник	Должность	Результат	Размер	Дата начала базового периода	Дата окончания базового периода
1	Персональная премия	Ильин Сергей Викторович	Менеджер	10 000,00	10 000,00		
2	Персональная премия	Троицкая Татьяна Сергеевна	Бухгалтер	10 000,00	10 000,00		
3	Месячная премия	Ильин Сергей Викторович	Менеджер	0	20,00	01.01.2014 0:00:00	31.01.2014 23:59:59
4	Месячная премия	Троицкая Татьяна Сергеевна	Бухгалтер	0	25,00	01.01.2014 0:00:00	31.01.2014 23:59:59

В регистре появились записи с видом расчета "Месячная премия" – пока с нулевым результатом расчета, поскольку код получения необходимых данных и вызова расчетной процедуры для категории "Зависимое первого уровня" еще не написан.

9.2.10. Запрос получения необходимых данных суммы базы для расчета премий

Вернемся к списку видов расчета "Премии" с настройками их реквизитов:

Вид расчета	Способ расчета	Категория расчета	Базовые виды расчета	Базовый план видов расчета
Персональная премия	Фиксированно	Первичное		
Месячная премия	Процентом	Зависимое первого уровня	Персональная премия	Премии
			Оклад	Основные начисления
			Оклад на выезде	Основные начисления
Премия за период	Процентом	Зависимое первого уровня	Оклад	Основные начисления

Поощрительная надбавка	Процентом	Зависимое второго уровня	Месячная премия	Премии
------------------------	-----------	--------------------------	-----------------	--------

Для всех из них, кроме "Персональная премия", указан уже нам знакомый и реализованный нами способ расчета – "Процентом". Однако вспомним, что для регистра расчета "Премии" имеет место новая расчетная ситуация: у него два регистра базы: результаты базовых видов расчета "Оклад" и "Оклад" на выезде выбираются из регистра "ОсновныеНачисления", а результат базового вида расчета "Персональная премия" – из регистра "Премии". Следовательно, в запросе для получения необходимых данных каждой записи документа в регистре "Премии" должны соответствовать строки двух таблиц базы – база по регистру "ОсновныеНачисления" и база по регистру "Премии". В языке запросов они носят такие имена:

"Премии.БазаОсновныеНачисления"

"Премии.БазаПремии"

Это задача для языка запросов. Нужно в запросе получить таким образом соотнесенные записи исходной (персистентной) таблицы регистра "Премии" для нашего документа и соответствующие им записи обеих таблиц базы.

Один из способов построения такого запроса состоит в использовании операции СОЕДИНЕНИЕ языка запросов: условием соединения будет равенство полей "НомерСтроки" исходных таблиц запроса. Но необходимо принять во внимание следующее важнейшее обстоятельство, которое мы уже затрагивали в нашем курсе:

Важно!

Виртуальные таблицы регистров платформы 1С не возвращают записей с нулевыми результатами.

(Это не касается только виртуальных таблиц регистров сведений).

Если, например, для некоторой записи регистра расчета по некоторому базовому регистру сумма базы равна нулю, то виртуальная таблица "База" просто не вернет строки с этим номером.

Из этого следует, что запрос для получения суммы базы по нескольким базовым регистрам нужно строить с учетом этого обстоятельства, иначе можно, при неправильном применении операции соединения, потерять сумму базы по какому-либо базовому регистру.

Чтобы это учесть, чаще всего применяют такой прием. В запросе выбирают персистентную таблицу регистра, поскольку она содержит данные всех записей документа – с первой по последнюю. А с этой таблицей левым соединением соединяют таблицы "База" тех регистров расчета, которые для рассчитываемого регистра содержат базовые суммы.

Этот прием применим и мы.

В режиме "1С:Предприятие" откроем Консоль запросов, добавим в наш список запросов новый запрос, назовем его "Данные базы регистра Премии" и откроем Конструктор запроса. Построим следующий запрос:

```
ВЫБРАТЬ
    Премии.НомерСтроки КАК НомерСтроки,
    Премии.ВидРасчета.СпособРасчета КАК СпособРасчета,
    ЕСТЬNULL (ПремииБазаОсновныеНачисления.РезультатБаза,
0) + ЕСТЬNULL (ПремииБазаПремии.РезультатБаза, 0) КАК СуммаБазы
ИЗ
    РегистрРасчета.Премии КАК Премии
        ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.Премии.БазаОсновныеНачисления (
            &ИзмеренияОсновногоРегистра,
            &ИзмеренияБазовогоРегистра,
            ,
            Регистратор = &Регистратор
                И ВидРасчета.КатегорияРасчета =
&КатегорияРасчета) КАК ПремииБазаОсновныеНачисления
        ПО Премии.НомерСтроки =
ПремииБазаОсновныеНачисления.НомерСтроки
        ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.Премии.БазаПремии (
            &ИзмеренияОсновногоРегистра,
            &ИзмеренияБазовогоРегистра,
            ,
            Регистратор = &Регистратор
                И ВидРасчета.КатегорияРасчета =
&КатегорияРасчета) КАК ПремииБазаПремии
        ПО Премии.НомерСтроки =
ПремииБазаПремии.НомерСтроки
ГДЕ
    Премии.Регистратор = &Регистратор
    И Премии.ВидРасчета.КатегорияРасчета =
&КатегорияРасчета

УПОРЯДОЧИТЬ ПО
    НомерСтроки
```

Кратко поясним, что мы сделали.

Запрос выполняет левое соединение персистентной таблицы регистра "Премии" с обеими виртуальными таблицами "База" этого регистра по условию равенства поля "НомерСтроки". Сумма базы каждой строки результата рассчитывается в запросе как сумма полей "РезультатБаза" обеих виртуальных таблиц. Поскольку соединение левое, то в результате запроса будут присутствовать номера всех строк набора записей нашего документа, с первой по

последнюю (они выбираются из самой левой, персистентной таблицы). А, если в одной или в обеих виртуальных таблицах сумма базы по какой-либо строке регистра равна нулю, то результатом левого соединения с этой таблицы будет NULL в поле "РезультатБаза" этой таблицы. Применяя функцию языка запросов "ЕСТЬNULL", мы заменяем NULL на числовой ноль, чтобы не было ошибки в результате сложения. Таким образом, поле "СуммаБазы" результата запроса будет содержать правильную сумму базы для каждой строки результата запроса.

Зададим значения параметров запроса:

&ИзмеренияОсновногоРегистра: Список значений (Сотрудник, Должность);

&ИзмеренияБазовогоРегистра: Список значений (Сотрудник, Должность);

&Регистратор: Документ Расчет зарплаты 000000001

&КатегорияРасчета: Зависимое первого уровня

Эти же значения параметров мы задавали для запроса получения базы регистра "ОсновныеНачисления".

Выполним запрос. Получим следующий результат:

НомерСтроки	СпособРасчета	СуммаБазы
3	Процентом	60 000
4	Процентом	73 913,05

Строка 3 – запись с видом расчета "Месячная премия" для г-жи Троицкой Т.С. Полученный результат суммы базовых начислений: 73913,05 руб.

Проверяем по начислениям г-жи Троицкой:

Оклад: 44 347,83 руб.

Оклад на выезде: 19 565,22 руб.

Персональная премия: 10 000,00 руб.

Итого: 73 913,05 руб.

Результат правильный.

Строка 4 - запись с видом расчета "Месячная премия" для г-на Ильина С.В.

Полученный результат суммы базовых начислений: 60 000,00 руб.

Проверяем по начислениям г-на Ильина:

Оклад: 50 000,00 руб.

Персональная премия: 10 000,00 руб.

Итого: 60 000,00 руб.

Результат правильный.

9.2.11. Пригодность составленного запроса для всего перечня премий

А теперь проведем небольшой эксперимент. Изменим значение параметра "&КатегорияРасчета" на "Первичное". Выполним запрос, получим результат:

НомерСтроки	СпособРасчета	СуммаБазы
1	Фиксировано	0
2	Фиксировано	0

Это – результат запроса для первых двух строк, которые содержат записи с видом расчета "Персональная премия". Следовательно, для получения необходимых данных регистра "Премии" можно выполнять один и тот же запрос, меняя только значение параметра категории расчета.

Вот поэтому мы и создали процедуру "РассчитатьПремии". Стоит вместо имеющегося там запроса поместить туда тот, который мы только что разработали – и процедура будет пригодна для расчета всех премий и ее можно будет вызывать, передавая ей разные значения категории расчета. Что мы сейчас и сделаем.

9.2.12. Код расчета для категории "Зависимое первого уровня"

Откроем модуль документа "РасчетЗарплаты", найдем процедуру "РассчитатьПремии". Заменим текст запроса, который присваивается свойству "Текст" объекта <Запрос>, на тот, который мы только что разработали и протестировали в Консоли запросов. Код процедуры теперь будет таким:

```

Процедура РассчитатьПремии (НаборЗаписей, Запрос,
КатегорияРасчета)
    Запрос.Текст =
        "ВЫБРАТЬ
        |     Премии.НомерСтроки КАК НомерСтроки,
        |     Премии.ВидРасчета.СпособРасчета КАК
СпособРасчета,
        |
        |     ЕСТЬNULL (ПремииБазаОсновныеНачисления.РезультатБаза, 0) +
ЕСТЬNULL (ПремииБазаПремии.РезультатБаза, 0) КАК СуммаБазы
        |ИЗ
        |     РегистрРасчета.Премии КАК Премии
        |
        |     ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.Премии.БазаОсновныеНачисления (
        |
        |         &ИзмеренияОсновногоРегистра,
        |         &ИзмеренияБазовогоРегистра,
        |         ,
        |         Регистратор = &Регистратор
        |
        |         И
ВидРасчета.КатегорияРасчета = &КатегорияРасчета) КАК
ПремииБазаОсновныеНачисления
        |
        |     ПО Премии.НомерСтроки =

```

```

ПремииБазаОсновныеНачисления . НомерСтроки
|
|          ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета . Премии . БазаПремии (
|
|          &ИзмеренияОсновногоРегистра ,
|
|          &ИзмеренияБазовогоРегистра ,
|
|          ,
|
|          Регистратор = &Регистратор
|
|          И
ВидРасчета . КатегорияРасчета = &КатегорияРасчета) КАК
ПремииБазаПремии
|
|          ПО Премии . НомерСтроки =
ПремииБазаПремии . НомерСтроки
|ГДЕ
|
|          Премии . Регистратор = &Регистратор
|
|          И Премии . ВидРасчета . КатегорияРасчета =
&КатегорияРасчета
|
|УПОРЯДОЧИТЬ ПО
|
|          НомерСтроки";

Запрос . УстановитьПараметр ("КатегорияРасчета",
КатегорияРасчета) ;
Выборка = Запрос . Выполнить () . Выбрать () ;
Расчеты . РассчитатьЗаписиРегистраРасчета (НаборЗаписей,
Выборка) ;
НаборЗаписей . Записать (, Истина) ;

КонецПроцедуры
    
```

И напишем вызов этой процедуры для расчета категории "Зависимое первого уровня" для регистра "Премии" – сразу же после ее вызова для расчета категории "Первичные":

```

Процедура ОбработкаПроведения (Отказ, РежимПроведения)
. . . . .
    РассчитатьПремии (НаборПремии, Запрос,
Справочники . КатегорииРасчета . Первичное) ;
    РассчитатьПремии (НаборПремии, Запрос,
Справочники . КатегорииРасчета . ЗависимоеПервогоУровня) ;
КонецПроцедуры
    
```


9.2.13. Расчет начисления "Месячная премия"

Обновим конфигурацию базы данных, запустим ее в режиме "1С:Предприятие", откроем форму документа и проведем его. Посмотрим в набор записей регистра "Премии" и увидим результаты расчета начислений:

Вид расчета	Сотрудник	Должность	Результат	Размер
Месячная премия	Ильин Сергей Викторович	Менеджер	12 000,00	20,00
Месячная премия	Троицкая Татьяна Сергеевна	Бухгалтер	18 478,26	25,00

Проверим результаты. Подставим в расчетную формулу "Процентом" полученные выше суммы базовых начислений и значения процента премии:

Ильин С.В.: $60\,000,00 * 20,00 / 100 = 12\,000,00$ руб.

Троицкая Т.С.: $73\,913,05 * 25,00 / 100 = 18\,478,26$ руб.

Результаты расчетов правильны.

Практикум № 5

Реализуйте альтернативный вариант запроса для получения необходимых данных: с использованием не операции СОЕДИНЕНИЕ, а операции ОБЪЕДИНИТЬ языка запросов.

9.2.14. Зависимые второго уровня. Начисление "Поощрительная надбавка"

Способ расчета у премии под названием "Поощрительная надбавка" такой же – "Процентом", а категория расчета – "Зависимое второго уровня", поскольку базовым видом расчета для нее назначен вид расчета "Месячная премия".

Следовательно, чтобы обеспечить расчет этого начисления, нужно в коде расчета премий еще раз написать вызов процедуры "РассчитатьПремии", передав ей значение категории расчета "Зависимое второго уровня".

Так и сделаем:

```

Процедура ОбработкаПроведения (Отказ, РежимПроведения)
. . . . .
    РассчитатьПремии (НаборПремии, Запрос,
Справочники.КатегорииРасчета.Первичное);
    РассчитатьПремии (НаборПремии, Запрос,
Справочники.КатегорииРасчета.ЗависимоеПервогоУровня);
    РассчитатьПремии (НаборПремии, Запрос,

```

Справочники . КатегорииРасчета . ЗависимоеВторогоУровня) ;

КонецПроцедуры

Обновим конфигурацию базы данных, запустим ее в режиме "1С:Предприятие".

Откроем документ "Расчет зарплаты", в табличную часть "Премии" добавим начисление "Поощрительная надбавка" для г-на Ильина С.В.:

Сотрудник	Должность	Вид расчета	Базовый период начало	Базовый период конец	Размер
Ильин Сергей Викторович	Менеджер	Поощрительная надбавка	01.01.2014 0:00:00	31.01.2014 0:00:00	10,00

Проведем документ. В регистре "Премии" увидим результат расчета нового начисления:

Вид расчета	Сотрудник	Должность	Результат	Размер
Поощрительная надбавка	Ильин Сергей Викторович	Менеджер	1 200,00	10,00

Проверим правильность результата.

Базовый вид расчета один: Месячная премия. Сумма базы по результатам этого вида расчета за базовый период: 12000,00 руб.

$$12\ 000,00 * 10,00 / 100 = 1\ 200,00 \text{ руб.}$$

Результат расчета правильный.

Практикум № 6

Создайте новую пару расчетных объектов: план видов расчета и регистр расчета "Удержания". Вспомните об особенностях настройки ПВР для удержаний.

Введите условный вид расчета "Алименты". Алименты и другие удержания будут рассчитываться по способу расчета "Процентом"; в состав базовых видов расчета входят все начисления из ПВР "Основные начисления" и "Премии", кроме "Штраф за прогул". Создайте все необходимые интерфейсные и программные объекты для реализации расчета удержаний, напишите программный код. Введите вид расчета "Алименты" какому-либо из сотрудников (значение процента в расчетной формуле - произвольное) и рассчитайте его.

9.2.15. "Частичное попадание" периода действия базового вида расчета в базовый период

В план видов расчета мы ввели вид расчета "Премия за период". Способ расчета у него "Процентом", базовый вид расчета только один - "Оклад", поэтому категория расчета у него "Зависимое первого уровня".

Как мы видим, сам по себе этот вид расчета интереса не представляет. Но мы создали его, чтобы продемонстрировать новую расчетную ситуацию, когда базовый период вида расчета частично "укладывается" в период действия базового вида расчета.

Что будет, если устанавливать базовый период не целый месяц, а, скажем, только с 10 по 15 число месяца? Очевидно, сумма базовых начислений должна сократиться пропорционально уменьшению количества рабочего времени в базовом периоде, иначе мы получим "несправедливо" большой результат расчета по базе.

Так оно и происходит с суммой базовых начислений по тем базовым видам расчета, которые обладают периодом действия, т.е. по расчетной формуле зависят от количества рабочего времени.

В режиме "1С:Предприятие" откроем документ "Расчет зарплаты" и в табличную часть "Премии" добавим строку с видом расчета "Премия за период" для сотрудника Ильин С.В., причем укажем базовый период длительностью только одну рабочую неделю, с 13.01.2014 по 17.01.2014:

Сотрудник	Должность	Вид расчета	Базовый период начало	Базовый период конец	Размер
Ильин Сергей Викторович	Менеджер	Премия за период	13.01.2014 0:00:00	17.01.2014 23:59:59	20,00

Проведем документ, получим результат:

Вид расчета	Сотрудник	Должность	Результат	Размер
Премия за период	Ильин Сергей Викторович	Менеджер	2 173,91	20,00

Чтобы уяснить себе полученный результат расчета, выполним в Консоли запросов запрос для получения суммы базовых начислений для видов расчета "Премии" с категорией расчета "Зависимое первого уровня". Мы увидим такой результат запроса:

НомерСтроки	СпособРасчета	СуммаБазы
3	Процентом	60 000
4	Процентом	73 913,05
6	Процентом	10 869,565217

Заглянем в список движений по регистру "Премии". Строка 6 – это начисление "Премия" за период г-на Ильина С.В., результат по его базовому виду расчета "Оклад" равен, как мы уже видели ранее, 50 000,00 руб. Но сумма базовых начислений для него получена в размере 10 869,565217 руб. Это произошло потому, что сумма базовых начислений получена не в полном размере, а в

пропорции к количеству рабочего времени в базовом периоде, с 13.01.2014 по 17.01.2014, . т.е. 5 дней.

Количество рабочих дней в расчетном месяце (январь 2014г.), по нашим данным графика, равно 23 дня, а количество рабочих дней в базовом периоде равно 5 дней.

Проверим результат расчета.

Сумма базовых начислений: $50\,000,00 * 5 / 23 = 10\,869,565217$ руб.

Результат расчета: $10\,869,565217 * 20,00 / 100 = 2\,173,91$ руб.

Итак, мы выяснили, что, когда базовый период лишь "частично" попадает в границы периода действия базового вида расчета, то сумма базовых начислений считается пропорционально количеству рабочего времени в базовом периоде.

Важно! _____

Но это действительно только для тех базовых видов расчета, которые обладают периодом действия, т.е. зависят по расчетной формуле от отработанного времени.

На базовые виды расчета, которые не обладают периодом действия, это не распространяется. Можно убедиться в этом самостоятельно, добавив такой вид расчета в состав базовых для начисления "Премия за период".

Практикум № 7 _____

Реализуйте способ расчета: "ПропорциональноБазе". Расчетная формула:

$$\frac{[\text{Сумма базовых начислений}] * [\text{Фактически отработанное время}]}{[\text{Плановое рабочее время}]}$$

Базовые виды расчета: "Оклад", "Доплата за квалификацию". Требование: результаты обоих видов расчета должны включаться в сумму базы полностью, т.е. сумма базы не должна зависеть от фактически отработанного времени.

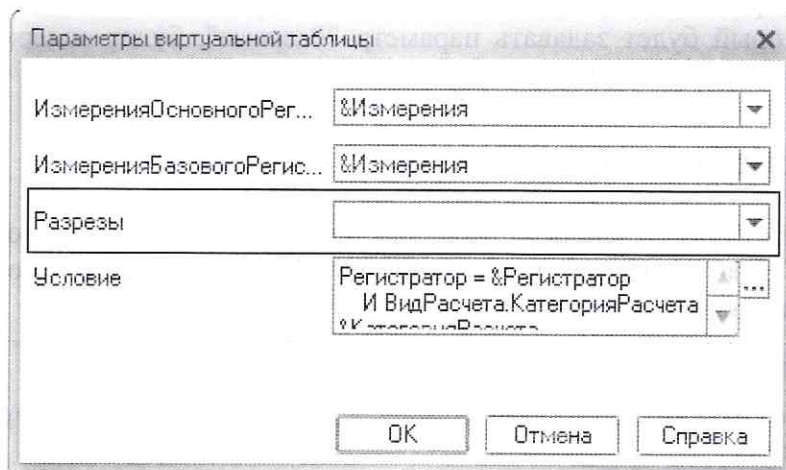
Практикум № 8 _____

Введите вид расчета "Спецкомандировка". Предполагается, что иногда сотрудники выезжают на выполнение некоторых конфиденциальных заданий; такая командировка оплачивается "Фиксированно" (по договоренности с руководством). Вид расчета вытесняет вид расчета "Оклад".

Вводить и рассчитывать новое начисление кому-либо из сотрудников не обязательно.

10. Получение суммы базы в разрезе базовых видов расчета

Вернемся к рассмотрению запроса к таблице "База" регистра расчета. Ранее мы не обращали внимание на третий параметр этой виртуальной таблицы, под названием "Разрезы":



а также на поля этой таблицы, имеющие в названии суффикс "Разрез":

- ⊖ = ПериодРегистрацииРазрез
- + ⊖ = РегистраторРазрез
- ⊖ = НомерСтрокиРазрез
- + ⊖ = ВидРасчетаРазрез
- ⊖ = ПериодДействияРазрез
- + ⊖ = СотрудникРазрез
- + ⊖ = ДолжностьРазрез
- ⊖ = РазмерРазрез
- + ⊖ = ГрафикРаботыРазрез

Эти средства позволяют развернуть сумму базовых начислений любого вида расчета в разрезе указанных полей, а также в разрезе комбинаций этих полей.

Нас прежде всего интересует разрез суммы базы по базовым видам расчета. Например, вид расчета "Месячная премия" из плана видов расчета "Премии" имеет три базовых вида расчета: "Оклад", "Оклад на выезде", "Персональная премия". Как узнать, чему равна сумма базы по результату каждого из этих начислений, т.е. сколько каждый базовый вид расчета "привнес" в общую сумму базы для месячной премии? Это позволяют сделать указанные средства настройки виртуальной таблицы "База" регистра расчета.

Для этого нужно в запросе выбрать одно или несколько полей этой таблицы, имеющих суффикс "Разрез" в названии – именно ту комбинацию полей, по которой мы хотим "развернуть" сумму базы для каждого вида расчета. Но одного этого недостаточно, нужно задать в запросе для виртуальной таблицы "База" третий параметр – "Разрезы".

10.1. Параметр "Разрезы" виртуальной таблицы "База"

Параметр "Разрезы" задается объектом типа <Массив> или <СписокЗначений>. Элементами массива являются строковые имена полей-разрезов (без суффикса "Разрез"), по которым мы хотим развернуть сумму базы.

Мы ставим задачу получить запросом в Консоли запросов сумму базы для каждого вида расчета в разрезе базовых видов расчета. Значит, наш список значений, который будет задавать параметр "Разрезы", будет содержать только один элемент – *ВидРасчета*.

10.2. Разработка запроса

Откроем Консоль запросов. Добавим в наш список запросов новый запрос, назовем его "База в разрезе" и при помощи Конструктора запросов построим следующий запрос:

```
ВЫБРАТЬ
    ОсновныеНачисленияБазаОсновныеНачисления.Сотрудник
КАК Сотрудник,
    ОсновныеНачисленияБазаОсновныеНачисления.ВидРасчета
КАК ВидРасчета,
    ОсновныеНачисленияБазаОсновныеНачисления.ВидРасчетаРазрез
КАК БазовыйВидРасчета,
    ОсновныеНачисленияБазаОсновныеНачисления.РезультатБаза
КАК РезультатБаза
ИЗ
    РегистрРасчета.ОсновныеНачисления.БазаОсновныеНачисления
    (&Измерения, &Измерения, &Разрезы, Регистратор =
    &Регистратор) КАК ОсновныеНачисленияБазаОсновныеНачисления

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ
    ПремииБазаОсновныеНачисления.Сотрудник,
    ПремииБазаОсновныеНачисления.ВидРасчета,
    ПремииБазаОсновныеНачисления.ВидРасчетаРазрез,
    ПремииБазаОсновныеНачисления.РезультатБаза
ИЗ
    РегистрРасчета.Премии.БазаОсновныеНачисления (&Измерения,
    &Измерения, &Разрезы, Регистратор = &Регистратор) КАК
    ПремииБазаОсновныеНачисления

ОБЪЕДИНИТЬ ВСЕ
```

ВЫБРАТЬ
ПремииБазаПремии.Сотрудник,
ПремииБазаПремии.ВидРасчета,
ПремииБазаПремии.ВидРасчетаРазрез,
ПремииБазаПремии.РезультатБаза
ИЗ
РегистрРасчета.Премии.БазаПремии (&Измерения, &Измерения, &Разрезы, Регистратор = &Регистратор) КАК ПремииБазаПремии
ИТОГИ
СУММА (РезультатБаза)
ПО
Сотрудник, ВидРасчета

Поясним наши действия.

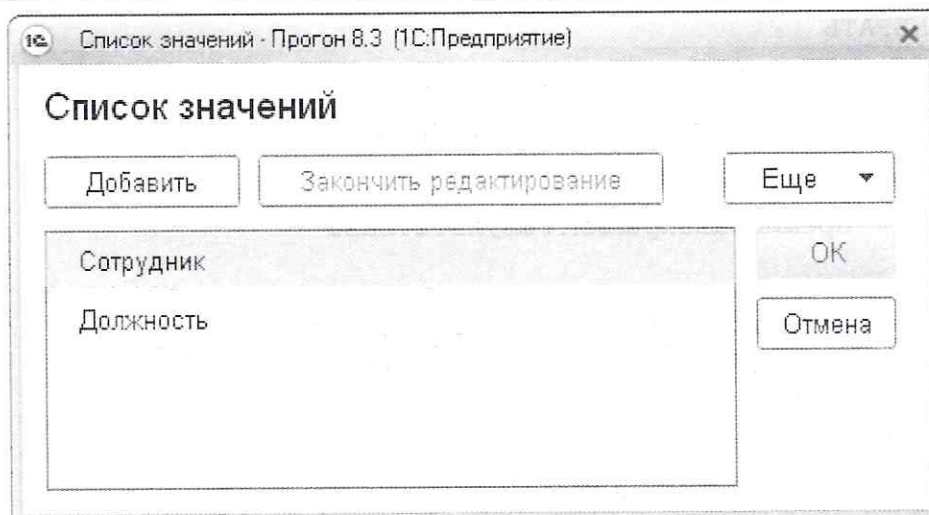
Мы объединяем в запросе выборки из виртуальных таблиц "База" обоих наших регистров расчета. Напомним, что у регистра "ОсновныеНачисления" одна таблица "База", а у регистра "Премии" – две.

Мы хотим получить результат для одного конкретного расчетного документа, поэтому в параметре "Условие" виртуальных таблиц через параметр "&Регистратор" будем задавать условие на документ.

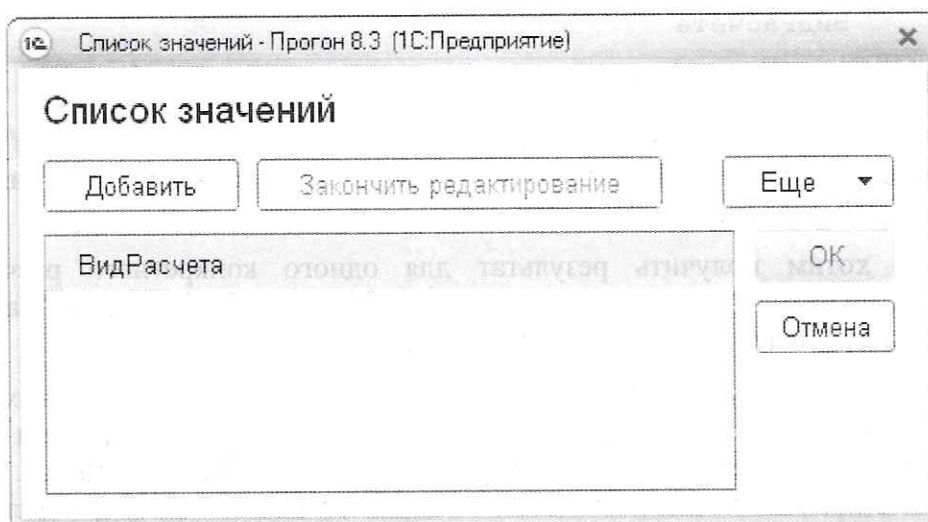
Мы хотим получить суммы базовых начислений в разрезе базовых видов расчета для каждого сотрудника, поэтому в запросах выбираем поля: "Сотрудник", "ВидРасчета", "ВидРасчетаРазрез" и "РезультатБаза". Поле результата запроса для полей "ВидРасчетаРазрез" для большей содержательности результата будет именоваться "БазовыйВидРасчета".

Чтобы получить сумму базовых начислений для каждого вида расчета воспользуемся секцией "Итоги" запроса: для каждого вида расчета просуммируем значения поля "РезультатБаза" его базовых видов расчета.

Зададим значения параметров запроса. Назначение первых двух параметров виртуальной таблицы "База" нам известно – это список значений со строковыми именами измерений основного и базового регистров расчета:

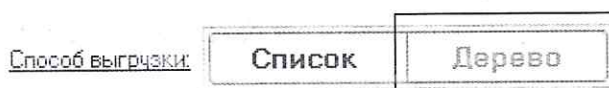


Для параметра "&Разрезы" тоже зададим тип <СписокЗначений> и внесем в список одно строковое значение – имя разреза *ВидРасчета*:



Для параметра "&Регистратор" зададим ссылку на наш документ "Расчет зарплаты".

Поскольку запрос содержит секцию ИТОГИ, его результат удобно просматривать в иерархическом виде. Для этого переключатель "Способ выгрузки" Консоли запросов установим в положение "Дерево":



Выполним запрос, развернем все итоговые ветви поля результата запроса и увидим следующий результат:

Сотрудник	ВидРасчета	БазовыйВидРасчета	РезультатБаза
Троицкая Татьяна Сергеевна	NULL	NULL	137 826,1
Троицкая Татьяна Сергеевна	Доплата за квалификацию	NULL	63 913,05
Троицкая Татьяна Сергеевна	Доплата за квалификацию	Оклад	44 347,83
Троицкая Татьяна Сергеевна	Доплата за квалификацию	Оклад на выезде	19 565,22
Троицкая Татьяна Сергеевна	Месячная премия	NULL	73 913,05
Троицкая Татьяна Сергеевна	Месячная премия	Оклад	44 347,83
Троицкая Татьяна Сергеевна	Месячная премия	Оклад на выезде	19 565,22
Троицкая Татьяна Сергеевна	Месячная премия	Персональная премия	10 000

Получение суммы базы в разрезе базовых видов расчета

Сотрудник	ВидРасчета	БазовыйВидРасчета	РезультатБаза
Ильин Сергей Викторович	NULL	NULL	82 869,565217
Ильин Сергей Викторович	Месячная премия	NULL	60 000
Ильин Сергей Викторович	Месячная премия	Оклад	50 000
Ильин Сергей Викторович	Месячная премия	Персональная премия	10 000
Ильин Сергей Викторович	Премия за период	NULL	10 869,565217
Ильин Сергей Викторович	Премия за период	Оклад	10 869,565217
Ильин Сергей Викторович	Поощрительная надбавка	NULL	12 000
Ильин Сергей Викторович	Поощрительная надбавка	Месячная премия	12 000

В результате запроса хорошо видно, сколько каждый базовый вид расчета "принес" в сумму базовых начислений каждого вида расчета, а также в итоговых записях результата запроса видна эта сумма базовых начислений (строки с NULL в поле "БазовыйВидРасчета").

Важно! _____

Возможность получать сумму базы в разрезе дает довольно ценную информацию для учета, а также для выяснений отношений с сотрудниками по суммам начислений и удержаний. Например, сотрудник пришел в бухгалтерию с претензией, что ему неправильно начислили премию. Получив сумму базовых начислений для премии в разрезе базовых видов расчета премии, можно показать сотруднику его неправоту, либо найти ошибку расчетчика при расчете премии.

11. Разработка отчета по начислениям

Разработаем отчет по начислениям для наших сотрудников, который будет иметь вот такой вид:

Суммы начислений

Сотрудник	Должность	Сумма
Регистр расчета		
Вид расчета		
Ильин Сергей Викторович	Менеджер	75 373,91
Основные начисления		50 000,00
Оклад		50 000,00
Премии		25 373,91
Месячная премия		12 000,00
Персональная премия		10 000,00
Поощрительная надбавка		1 200,00
Премия за период		2 173,91
Ковалев Александр Петрович	Директор	70 000,00
Основные начисления		70 000,00
Оклад		70 000,00
Троицкая Татьяна Сергеевна	Бухгалтер	98 782,62
Основные начисления		70 304,36
Доплата за квалификацию		6 391,31
Оклад		44 347,83
Оклад на выезде		19 565,22
Премии		28 478,26
Месячная премия		18 478,26
Персональная премия		10 000,00
Итого		244 156,53

Т.е., для каждого сотрудника покажем по регистрам расчета состав и результаты расчета начислений за заданный учетный период.

В Конфигураторе создадим внешний отчет.

Создадим основную схему компоновки данных.

Создадим набор данных – объединение, назовем его "Суммы".

Для набора-объединения создадим два набора данных типа <Запрос>, назовем их "ОсновныеНачисления" и "Премии":



Для набора данных "ОсновныеНачисления" напишем следующий запрос:

```

ВЫБРАТЬ
    &ОсновныеНачисления КАК РегистрРасчета,
    ОсновныеНачисления.Сотрудник КАК Сотрудник,
    ОсновныеНачисления.Должность,
    ОсновныеНачисления.ВидРасчета,
    ОсновныеНачисления.Результат КАК Результат
ИЗ
    РегистрРасчета.ОсновныеНачисления КАК
ОсновныеНачисления
ГДЕ
    ОсновныеНачисления.ПериодРегистрации =
НАЧАЛОПЕРИОДА (&ПериодРегистрации, МЕСЯЦ)
    
```

Т.е. выбираем данные из регистра расчета "ОсновныеНачисления" за заданный период регистрации.

Чтобы иметь возможность образовывать группировку отчета по имени регистра расчета, мы ввели в запрос параметр "&ОсновныеНачисления", в котором будет задаваться строковое название регистра расчета – *Основные начисления*.

Для набора данных "Премии" напишем аналогичный запрос:

```

ВЫБРАТЬ
    &Премии КАК РегистрРасчета,
    Премии.Сотрудник КАК Сотрудник,
    Премии.Должность,
    Премии.ВидРасчета,
    Премии.Результат КАК Результат
ИЗ
    РегистрРасчета.Премии КАК Премии
ГДЕ
    Премии.ПериодРегистрации =
НАЧАЛОПЕРИОДА (&ПериодРегистрации, МЕСЯЦ)
    
```

Точно так же для цели группировки по регистру расчета мы ввели в запроса параметр "&Премии", в котором будет задаваться строковое название регистра "Премии".

Для набора данных – объединения зададим для поля "Результат" заголовок - "Сумма".

На закладке "Ресурсы" редактора СКД зададим состав ресурсов схемы компоновки данных. Ресурс будет один – сумма по полю "Результат":

поля	Ресурсы	Параметры	Макеты	Вкл
Поле	Выражение			
⇒ Результат	Сумма(Результат)			

На закладке "Параметры" редактора СКД зададим следующие настройки параметров:

Наборы данных	Связи наборов данных	Вычисляемые поля	Ресурсы	Параметры	Макеты	Вложенные схемы	Настройки				
Имя	Заголовок	Тип	Д.	Д.	Значение	Выражение	Па.	Вкло.	Ограничен.	Запрещать не.	Использование
ПериодРегистрации	Период реги...	Дата		<input type="checkbox"/>				<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Всегда
ОсновныеНачисления	Основные начисления	Число		<input type="checkbox"/>	1 "Основные начисления"			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Всегда
Премии	Премии	Число		<input type="checkbox"/>	2 "Премии"			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Всегда

Поясним сделанные настройки.

Для параметров "ОсновныеНачисления" и "Премии" зададим строковые выражения, которые позволят выводить название регистра расчета и группировать по нему данные – соответственно *Основные начисления* и *Премии*.

Для этих двух параметров установим свойство "Ограничение доступности", так как пользователь не должен управлять значениями этих параметров.

Для параметра "ПериодРегистрации" свойство "Использование" установим в значение "Всегда". В этом случае флажок "Использование" для этого параметра в пользовательских настройках не будет доступен и пользователь будет должен обязательно задавать значение параметра.

На закладке "Настройки" зададим следующую структуру группировок отчета:

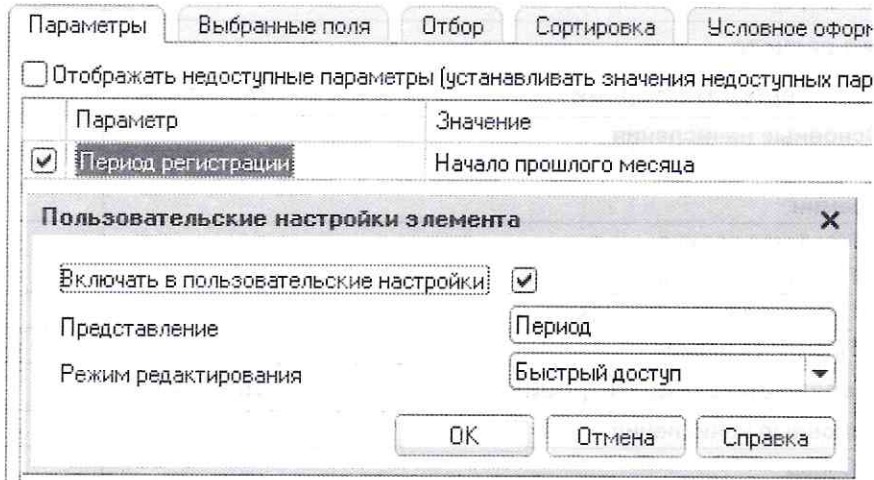
Отчет
<ul style="list-style-type: none"> ☑ Сотрудник, Должность ☑ РегистрРасчета ☑ ВидРасчета

Обратим внимание, что среди группировок имеется составная группировка, по двум полям: "Сотрудник" и "Должность".

Там же, на внутренней закладке "Выбранные поля" поместим в состав выбранных полей только одно поле – ресурс "Результат":

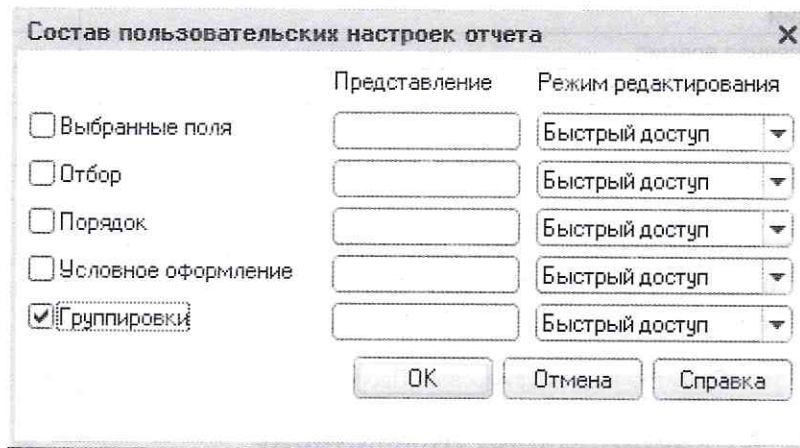
Параметры	Выбранные поля	Отбор	Сортировка
Доступные поля		Поле	
<ul style="list-style-type: none"> ☑ ВидРасчета ☑ Должность ☑ РегистрРасчета ☑ Сотрудник ☑ Результат ☑ СистемныеПоля ☑ ПараметрыДанных 		<ul style="list-style-type: none"> ☑ Выбранные поля ☑ Результат 	

Там же, на внутренней закладке "Параметры" установим для параметра "Период регистрации" флажок свойства "Включать в пользовательские настройки":



- чтобы пользователь мог задавать значение учетного периода, за который выбираются начисления.

Для отчета в целом зададим вывод состава и структуры группировок в пользовательские настройки, чтобы пользователь мог управлять группировками:



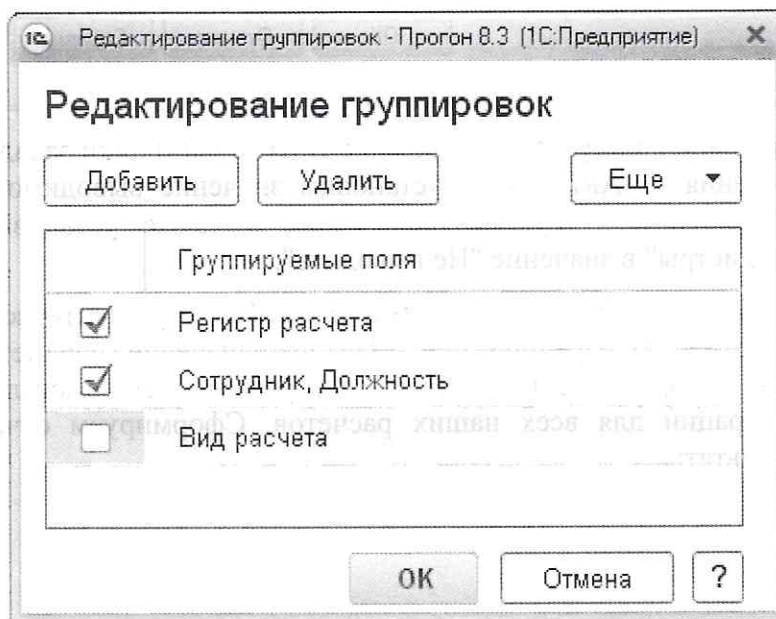
В завершение на внутренней закладке Другие настройки зададим значение макета оформления – "Античный", установим значение выводимого заголовка (свойство "Заголовок") – *Суммы начислений*, а также установим свойство "Выводить параметры" в значение "Не выводить".

Отчет готов. Сохраним его с именем файла "СуммыНачислений.erf" и откроем в режиме "1С:Предприятие". Зададим значение параметра "Период регистрации" – первое число того месяца, который мы использовали в качестве периода регистрации для всех наших расчетов. Сформируем отчет, получим желаемый результат:

Суммы начислений

Сотрудник	Должность	Сумма
Регистр расчета		
Вид расчета		
Ильин Сергей Викторович	Менеджер	75 373,91
Основные начисления		50 000,00
Оклад		50 000,00
Премии		25 373,91
Месячная премия		12 000,00
Персональная премия		10 000,00
Поощрительная надбавка		1 200,00
Премия за период		2 173,91
Ковалев Александр Петрович	Директор	70 000,00
Основные начисления		70 000,00
Оклад		70 000,00
Троицкая Татьяна Сергеевна	Бухгалтер	98 782,62
Основные начисления		70 304,36
Доплата за квалификацию		6 391,31
Оклад		44 347,83
Оклад на выезде		19 565,22
Премии		28 478,26
Месячная премия		18 478,26
Персональная премия		10 000,00
Итого		244 156,53

Через пользовательские настройки можно управлять составом и порядком группировок отчета. Например, если мы зададим такой состав и порядок группировок: "Регистр расчета", "(Сотрудник, Должность)":



то мы получим вот такой результат отчета:

Суммы начислений

Регистр расчета		Сумма
Сотрудник	Должность	
Основные начисления		190 304,36
Ильин Сергей Викторович	Менеджер	50 000,00
Ковалев Александр Петрович	Директор	70 000,00
Троицкая Татьяна Сергеевна	Бухгалтер	70 304,36
Премии		53 852,17
Ильин Сергей Викторович	Менеджер	25 373,91
Троицкая Татьяна Сергеевна	Бухгалтер	28 478,26
Итого		244 156,53

12. Корректировка результатов расчета прошедших периодов

12.1. Невозможность исправления результатов расчета за прошедший период

До сих пор мы не обращали внимания на значение свойства "ПериодРегистрации" записей регистра расчета. Нас интересовали только период действия и базовый период.

Мы начали наш курс с того, что пояснили назначение периода регистрации – это период бухгалтерских проводок с суммами результатов расчета зарплаты. Поэтому период регистрации до сих пор представляется нам чем-то "внешним", не имеющим отношения непосредственно к выполнению расчетов, к использованию расчетных механизмов платформы. В самом деле, ни в одном обращении к механизмам регистра расчета, будь то получение данных графика или получение суммы базовых начислений, период регистрации не фигурирует в качестве исходного данного.

Настало время раскрыть значение и смысл этого свойства записи регистра расчета.

Расчет заработной платы выполняется, естественно, не для того, чтобы его результаты просто сохранить в базе данных. На основании полученных результатов расчета:

а) сотрудникам предприятия выплачивается сумма денег, соответственно результатам расчета зарплаты;

б) составляются бухгалтерские отчеты по фонду зарплаты для предоставления в фискальные и другие контролирующие организации для взыскания соответствующих отчислений (налоги и т.д.) как с предприятия, так и с самих сотрудников – физических лиц.

Эта бухгалтерская отчетность, как и сам расчет зарплаты, составляется со строгим соблюдением установленной периодичности, что было отмечено в самом начале нашего курса. При этом отчетность, предоставляемая в контролирующие организации, датируется периодом, который и есть то, что мы называем периодом регистрации.

Важно!

С этим связано одно очень серьезное обстоятельство: **сданная бухгалтерская отчетность корректировке и изменению не подлежит**. Из этого следует, что, если в отчетности за некий прошедший период в текущем периоде обнаружена ошибка, то ее исправление и повторная подача за тот же прошедший период невозможна.

В самом деле, контролирующие организации физически не способны обеспечить повторный прием и анализ отчетности за прошлые периоды. Если отчетность за период сдана, то про этот период говорят: период закрыт, и никакие изменения в отчетности за этот период уже невозможны.

Но ошибки в расчетах и отчетности, естественно, возможны и делаются постоянно. Только для их исправления, очевидно, должна использоваться не повторная подача исправленной отчетности за "закрытый" период, а какие-то другие подходы. Бухгалтеры очень хорошо знают, как исправляются ошибки закрытых периодов. Подобный подход необходимо изучить и нам.

12.2. Случай, когда должны рассчитываться начисления за прошедший период

Мы ранее создали вид расчета "Спецкомандировка". Это очень простой вид расчета, рассчитываемый по способу "Фиксированно", т.е. не требующий получения никаких необходимых данных. Но, если сотрудник получает оклад, то этот вид расчета должен обладать периодом действия и быть вытесняющим для вида расчета "Оклад", так как сотрудник во время нахождения в спецкомандировке отсутствует на рабочем месте и не выполняет ту работу, за которую ему начисляется оклад.

Предположим, что сотрудник Ильин С.В. выезжал на выполнение особых поручений руководства в период с 15.01.2014 по 22.01.2014. Он договорился с руководством, что за выполнение этих поручений он получит 30 000 рублей.

Итак, г-н Ильин С.В. с 15.01.2014 по 22.01.2014 отсутствовал на своем рабочем месте, а по приезде из спецкомандировки получил на руки свои 30 000 рублей. Но по какой-то причине информация о том, что он был в командировке, не поступила расчетчику в январе, а поступила только в феврале. Следовательно, расчетчик должен ввести этот вид расчета уже в феврале, а период действия у него будет, естественно, январский - с 15.01.2014 по 22.01.2014.

Значит, нужно создать новый экземпляр документа "Расчет зарплаты", задать для него период регистрации уже 01.02.2014, ввести в него вид расчета "Спецкомандировка" для г-на Ильина с периодом действия с 15.01.2014 по 22.01.2014 и размером 30 000,00 и рассчитать его.

Сделаем это.

Создадим новый документ "Расчет зарплаты" (он, видимо, получит номер 000000002), в его форме зададим период регистрации 01.02.2014, а в табличную часть "Основные начисления" введем следующую строку:

Сотрудник	Должность	Вид расчета	Период действия начало	Период действия конец	Размер
Ильин Сергей Викторович	Менеджер	Спецкомандировка	15.01.2014 0:00:00	22.01.2014 0:00:00	30 000,00

Проведем документ. В регистре расчета *Основные начисления* увидим запись:

Период регистрации	Регистратор	Вид расчета	Сотрудник	Должность	Результат	Размер	Дата начала периода действия	Дата окончания периода действия
01.02.2014 0:00:00	Расчет заработной платы 000000002	Спецкомандировка	Ильин Сергей Викторович	Менеджер	30000,00	30000,00	15.01.2014 0:00:00	22.01.2014 23:59:59

Т.е., вид расчета "Спецкомандировка" рассчитан.

Теперь внимательно посмотрим на его период действия. Он – январский, он лежит в пределах периода действия вида расчета "Оклад" для г-на Ильина, который мы рассчитали ему за январь. "Спецкомандировка" – вытесняющий вид расчета для "Оклад", следовательно... по идее, если пересчитать первый (январский) документ, то должно произойти вытеснение вида расчета "Оклад" для г-на Ильина, и его результат расчета должен уменьшиться...

Откроем форму нашего первого документа "Расчет заработной платы 000000001" и проведем его заново. Посмотрим в регистр расчета "Основные начисления"... Что такое??? Сумма оклада г-на Ильина по прежнему 50000,00 руб., следовательно, никакого вытеснения не произошло...

Отменим проведение документа "Расчет заработной платы 000000002". Введем запись с этим видом расчета в документ 000000001 и рассчитаем его. Теперь все в порядке: результат расчета оклада для г-на Ильина 36 956,52 руб., т.е. явно видно, что механизм вытеснения работает. Можно убедиться в этом, выполнив в Консоли запросов запрос к таблице "ДанныеГрафика" для документа 000000001 – время по фактическому периоду действия для вида расчета "Оклад" у г-на Ильина составляет 17 дней против 23 по периоду действия.

Вернем ситуацию, удалим введенную строку из документа 000000001, проведем документ, а затем снова повторим эксперимент: проведем документ 000000002, а следом проведем документ 000000001. Вновь видим, что оклад у г-на Ильина не изменился...

Что же происходит?

У нас возникла принципиально новая расчетная ситуация: период регистрации вида расчета "Спецкомандировка" (документ "Расчет заработной платы 000000002") больше его периода действия: период регистрации – 01.02.2014, а период действия – с 15.01.2014 по 22.01.2014. Это означает, что делается попытка изменить результаты расчета документа прошедшего (закрытого) периода, что недопустимо. И расчетные механизмы платформы отслеживают это.

Важно!

Если период регистрации вытесняющего вида расчета больше его периода действия, то расчетные механизмы запрещают вытеснение.

Ибо это означало бы изменение результатов закрытого периода, который считается по периоду регистрации.

Т.е. платформа сама следит, чтобы попытка пересчета закрытого периода не привела к изменению его результатов.

12.3. Сторно

Но что же нам делать? Ведь г-н Ильин С.В. "переполучил" в январе зарплату по виду расчета "Оклад": за период спецкомандировки с 15.01.2014 по 22.01.2014 он не работал на своем рабочем месте, следовательно, за этот период оклад ему начисляться не должен. Как восстановить справедливость? Как исправить ошибку расчета уже закрытого периода?

Бухгалтер хорошо знает, как исправляются ошибки закрытого периода.

Предположим, бухгалтер в прошедшем (закрытом) периоде, в январе, сделал ошибочную проводку, ну, хотя бы начислил кому-то лишнюю 1000 рублей по зарплате:

Период	Дебет	Кредит	Сумма
январь 2014г.	(26)	(70)	1000,00

Закрытый период исправлять нельзя. Значит, бухгалтер должен действовать косвенно, а именно, сделать уже в новом отчетном периоде следующую проводку:

Период	Дебет	Кредит	Сумма
февраль 2014г.	(26)	(70)	– 1000,00

- т.е. сделать проводку с той же корреспонденцией счетов, но с отрицательной суммой, составляющей ошибку прошлого периода.

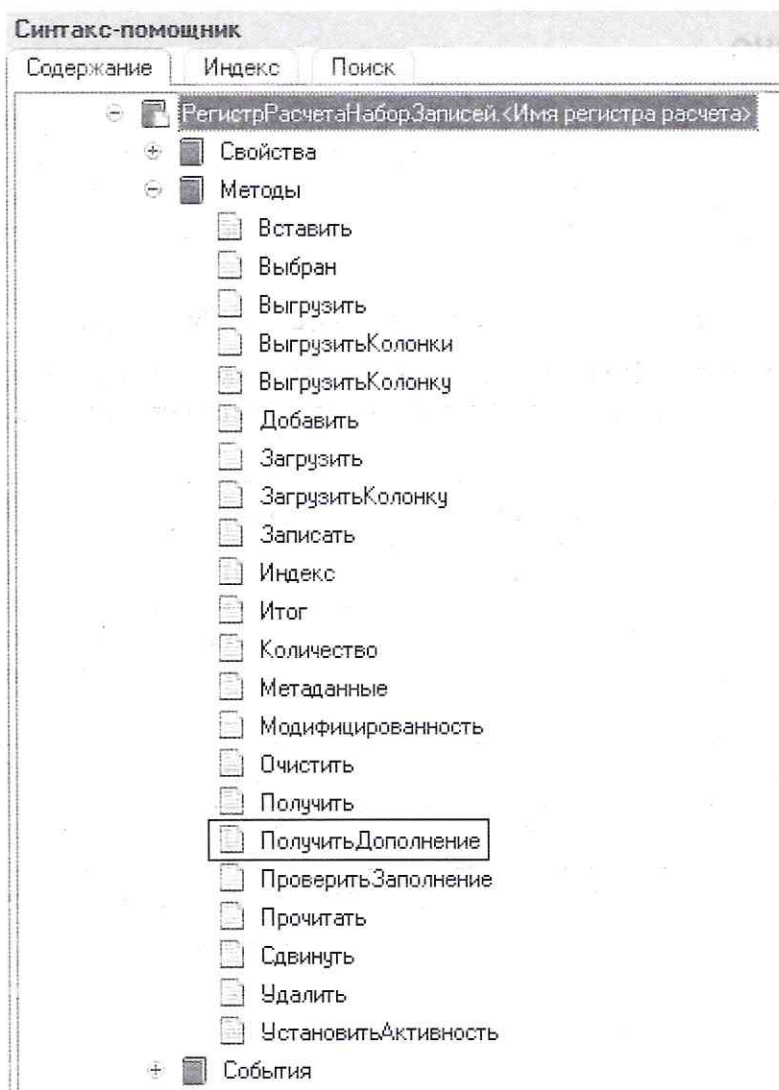
Такая проводка называется в бухгалтерии **сторно** (сброс). Вот так бухгалтер и исправляет ошибки закрытых периодов, и баланс приходит в актуальное состояние.

Аналогично должны поступить и мы. Г-н Ильин переполучил сумму оклада из расчета его месячной ставки (50000,00 руб.) за количество рабочих дней в период с 15.01.2014 по 22.01.2014. Значит, нужно рассчитать ему оклад исходя из этой ставки за период с 15.01.2014 по 22.01.2014 и в следующем расчетном периоде (февраль) записать этот результат в регистр расчета с отрицательным знаком. Это и будет операция сторно – аналог сторно бухгалтерского.

12.4. Средства платформы для расчета сторно-суммы. Метод "ПолучитьДополнение"

В общем случае периоды, за которые нужно рассчитать сторно-суммы, заранее не известны. Платформа дает в наше распоряжение средство получения всей необходимой информации, в том числе и периодов сторнирования, для расчета сторнируемых сумм.

В Конфигураторе откроем Синтакс-помощник, найдем описание объекта "РегистрРасчетаНаборЗаписей", заглянем в состав его методов. Мы найдем там метод "ПолучитьДополнение" – подобного метода нет у наборов записей других классов регистров платформы 1С 8:



Это метод-функция, который возвращает объект типа <ТаблицаЗначений>. Каждая строка этой таблицы соответствует ситуации, когда платформа запрещает вытеснение по причине того, что период действия вытесняющего вида расчета меньше периода регистрации.

Это как раз наш случай. В таблице значений будет столько строк, сколько раз платформа запретила вытеснение, а каждая ее строка будет содержать всю необходимую информацию для расчета сторно-суммы.

Значит, мы должны перед расчетом сторно-суммы вызывать метод "ПолучитьДополнение" и обработать полученную таблицу значений, создавая для

каждой ее строки сторно-запись регистра расчета. Остается определить, в каком месте программного кода должен вызываться этот метод.

Подумаем: когда механизм вытеснения платформы может определить, что вытеснение запрещено? Очевидно, в тот момент, когда он актуализирует все вытеснения набора записей, иными словами, когда определяется состав и границы интервалов фактического периода действия для каждого вида расчета.

Мы уже знаем, в какой момент это происходит. Это происходит тогда, когда программа выполняет запись набора с исходными данными – т.е. когда метод "Записать" набора записей вызывается со вторым аргументом, равным Ложь. Следовательно, после вызова этого метода и надо вызвать метод "ПолучитьДополнение": если к этому моменту получен актуальный фактический период действия – результат вытеснения, то, следовательно, регистр расчета к этому моменту имеет информацию о тех случаях, когда вытеснение было запрещено.

Значит, после строки процедуры "ОбработкаПроведения"

```
НаборОсновныеНачисления.Записать (, Ложь);
```

и нужно вызвать метод "ПолучитьДополнение" с последующей обработкой, т.е. написать:

```
ТаблицаСторноЗаписей =  
НаборОсновныеНачисления.ПолучитьДополнение ();  
..... // (обработка таблицы сторно-  
записей)
```

Про то, как обрабатывать полученную таблицу сторно-записей, поговорим позже. А пока исследуем ее визуально: посмотрим, какую информацию она дает.

Для этого можно воспользоваться методом "ВыбратьСтроку" объекта <ТаблицаЗначений>. Этот метод-функция предназначен для интерактивного выбора строки таблицы значений: он открывает модальное окно, содержащее таблицу значений, и возвращает выбранную пользователем строку. Однако нам не нужно выбирать строку, нам нужно только посмотреть структуру и содержание таблицы значений. Значит, можно вызвать этот метод-функцию "как процедуру", т.е. не получая возвращаемое значение.

Итак, в процедуре "ОбработкаПроведения" документа "РасчетЗарплаты" после записи набора "ОсновныеНачисления" с исходными данными напомним то, что показано жирным шрифтом:

```
Процедура ОбработкаПроведения (Отказ, РежимПроведения)  
.....  
НаборОсновныеНачисления.Записать (, Ложь);  
ТаблицаСторноЗаписей =  
НаборОсновныеНачисления.ПолучитьДополнение ();  
ТаблицаСторноЗаписей.ВыбратьСтроку ();  
.....  
КонецПроцедуры
```

Обновим конфигурацию базы данных и запустим ее в режиме "1С:Предприятие". Откроем форму документа "Расчет зарплаты 000000002" – того, который рассчитывает вид расчета "Спецкомандировка" с январским периодом действия и февральским периодом регистрации.

Нажмем "Провести". При проведении сработает наш код и мы увидим на экране окно с таблицей значений такого содержания (для компактности отображения на рисунке таблица "транспонирована" – развернута на 90° влево):

ПериодРегистрации	01.01.2014 0:00:00
Регистратор	Расчет зарплаты 000000001 от 31.01.2014 0:00:00
НомерСтроки	3
ВидРасчета	Оклад
ПериодДействия	01.01.2014 0:00:00
ПериодДействияНачало	01.01.2014 0:00:00
ПериодДействияКонец	31.01.2014 23:59:59
БазовыйПериодНачало	01.01.0001 0:00:00
БазовыйПериодКонец	01.01.0001 0:00:00
Активность	Истина
Сторно	Ложь
Сотрудник	Ильин Сергей Викторович
Должность	Менеджер
Результат	50 000
Размер	50 000
ГрафикРаботы	Пятидневка
ПериодРегистрацииСторно	01.02.2014 0:00:00
ПериодДействияНачалоСторно	15.01.2014 0:00:00
ПериодДействияКонецСторно	22.01.2014 23:59:59

Исследуем содержание таблицы. Сначала (первые 16 колонок) идет полное повторение строки "Оклад" для Ильина С.В. за январь – т.е. того вида расчета, для которого была сделана попытка вытеснения видом расчета "Спецкомандировка", запрещенная платформой. А следом идут три колонки:

"ПериодРегистрацииСторно" – период регистрации того вида расчета ("Спецкомандировка"), который сделал "неудавшуюся" попытку вытеснения (01.02.2014);

"ПериодДействияНачалоСторно" – начало периода действия этого вида расчета (15.01.2014);

"ПериодДействияКонецСторно" – конец периода действия этого вида расчета (22.01.2014).

Таким образом, мы имеем всю исходную информацию для того, чтобы рассчитать для г-на Ильина сторнируемую в феврале сумму оклада, которую он переполучил в январе. Для расчета нужно:

- создать запись регистра расчета с исходными данными, соответствующими полученным в строке таблицы, но период регистрации будет таким, который указан в поле "ПериодРегистрацииСторно", а границы периода действия будут такими, какие указаны в полях строки таблицы

"ПериодДействияНачалоСторно" и "ПериодДействияКонецСторно"
соответственно;

- рассчитать результат этой записи;
- изменить знак полученного результата записи на отрицательный.

Так будет получена сторнирующая запись, относящаяся к новому учетному периоду.

Это сделает код, который мы напишем вместо вызова метода "ВыбратьСтроку"(), который нужно предварительно удалить или закомментировать.

12.5. Цикл обработки таблицы сторно-записей

Теперь нам ясно, что делать с этой таблицей. Ее нужно обработать в цикле, построчно. Для каждой строки нужно создать новую запись в регистре расчета, заполняя ее поля значениями из полей строки таблицы сторно-записей, а затем снова записать набор записей и рассчитать его.

Итак, закомментируем вызов метода "ВыбратьСтроку" таблицы значений и вместо нее напишем следующий цикл:

```
Процедура ОбработкаПроведения (Отказ, РежимПроведения)
. . . . .
//ТаблицаСторноЗаписей.ВыбратьСтроку ();
Для каждого СтрокаСторно из ТаблицаСторноЗаписей Цикл
    ДобавитьСтрокуСторно (НаборОсновныеНачисления,
СтрокаСторно, ОсновныеНачисления);
    КонецЦикла;
    Записать ();
    НаборОсновныеНачисления.Записать (, Ложь);
. . . . .
КонецПроцедуры
```

Поясним содержание нового фрагмента кода.

В цикле обходятся последовательно строки полученной таблицы сторно-записей. На каждом проходе цикла вызывается процедура "ДобавитьСтрокуСторно", которую мы сейчас напишем. Она для каждой строки таблицы сторно-записей создает новую запись в наборе записей регистра расчета и заполняет ее поля значениями полей строки таблицы.

По окончании цикла повторно записывается документ (для этого вызывается метод "Записать" документа, доступный в модуле объекта; необходимость этого мы обоснуем, когда будем писать процедуру обработки строки таблицы).

В завершение обработки таблицы сторно-записей набор записей регистра расчета повторно записывается с пересчетом фактического периода действия (второй параметр метода "Записать" имеет значение Ложь), потому что новые записи регистра расчета могут оказать влияние на фактический период действия.

12.6. Процедура обработки строки таблицы сторно-записей

Теперь напишем саму процедуру "ДобавитьСтрокуСторно". В модуле документа после конца процедуры "ОбработкаПроведения" напишем следующий код:

```
Процедура ДобавитьСтрокуСторно (НаборЗаписей, СтрокаСторно,
ТабличнаяЧасть)
```

```
    // Добавление записи в набор записей
```

```
    ЗаписьРегистра = НаборЗаписей.Добавить ();
```

```
    ЗаписьРегистра.ПериодРегистрации =
СтрокаСторно.ПериодРегистрацииСторно;
```

```
    ЗаписьРегистра.ПериодДействияНачало =
СтрокаСторно.ПериодДействияНачалоСторно;
```

```
    ЗаписьРегистра.ПериодДействияКонец =
СтрокаСторно.ПериодДействияКонецСторно;
```

```
    ЗаписьРегистра.БазовыйПериодНачало =
СтрокаСторно.БазовыйПериодНачало;
```

```
    ЗаписьРегистра.БазовыйПериодКонец =
СтрокаСторно.БазовыйПериодКонец;
```

```
    ЗаписьРегистра.ВидРасчета = СтрокаСторно.ВидРасчета;
```

```
    ЗаписьРегистра.Сторно = Истина;
```

```
    ЗаписьРегистра.Сотрудник = СтрокаСторно.Сотрудник;
```

```
    ЗаписьРегистра.Должность = СтрокаСторно.Должность;
```

```
    ЗаписьРегистра.Размер = СтрокаСторно.Размер;
```

```
    ЗаписьРегистра.ГрафикРаботы =
СтрокаСторно.ГрафикРаботы;
```

```
КонецПроцедуры
```

Поясним смысл написанного кода.

Процедура вызывается на каждом проходе цикла обработки таблицы сторно-записей и ей передаются аргументы: набор записей регистра расчета – "НаборЗаписей" и полученная в цикле строка таблицы значений – "СтрокаСторно", а также объект табличной части – "ТабличнаяЧасть" (зачем нужен этот третий аргумент – скоро увидим).

Процедура добавляет новую запись в набор записей регистра и заполняет ее значениями полей полученной строки таблицы значений – в соответствии с логикой, которую мы обсудили выше.

Отметим только, что поле "Сторно" записи регистра устанавливается в Истина. Это будет признак для процедуры расчета общего модуля, что знак результата надо изменить на отрицательный.

Изменим код процедуры общего модуля "РассчитатьЗаписиРегистраРасчета". Найдем в ней строку:

```
ЗаписьРегистра.Результат = Результат;
```

и изменим ее следующим образом:

```
ЗаписьРегистра.Результат = Результат *  
? (ЗаписьРегистра.Сторно, -1, 1);
```

Мы ввели в выражение для результата сторно-множитель, значение которого вырабатывается т.н. тернарной операцией – операцией "вопросительный знак". Тернарной операции передается три аргумента: первый имеет тип <Булево>, второй и третий имеют произвольные типы. Тернарная операция возвращает значение второго аргумента, если значение первого аргумента равно Истина, и значение третьего аргумента, если значение первого аргумента равно Ложь.

Таким образом, тернарная операция вернет –1, если значение поля "Сторно" записи регистра будет Истина, и вернет 1 в противном случае. Для этого процедура "ДобавитьСтрокуСторно" устанавливает поле "Сторно" создаваемой записи регистра расчета в значение Истина. Как следствие, знак результата будет отрицательным, если поле "Сторно" записи регистра равно Истина, чего мы и добивались.

Задача почти выполнена. Осталось подумать вот над чем. Сторнирующая запись в регистре создается автоматически, при обработке таблицы сторно-записей, а не записывается в регистр из табличной части. Таким образом, такой сторно-строки в табличной части у нас нет. Это может вызвать недоумение у пользователя: в регистре строка есть, а в табличную часть он ее не вводил...

Чтобы не загадывать пользователю эту загадку, нужно автоматически добавить эту сторно-строку и в табличную часть, и тогда между записями регистра расчета и строками табличной части будет полное соответствие.

С этой целью допишем в процедуру следующий код:

```
Процедура ДобавитьСтрокуСторно (НаборЗаписей, СтрокаСторно,  
ТабличнаяЧасть)
```

```
// Добавление записи в набор записей
```

```
ЗаписьРегистра = НаборЗаписей.Добавить ();
```

```
ЗаписьРегистра.ПериодРегистрации =  
СтрокаСторно.ПериодРегистрацииСторно;  
ЗаписьРегистра.ПериодДействияНачало =  
СтрокаСторно.ПериодДействияНачалоСторно;  
ЗаписьРегистра.ПериодДействияКонец =  
СтрокаСторно.ПериодДействияКонецСторно;  
ЗаписьРегистра.БазовыйПериодНачало =  
СтрокаСторно.БазовыйПериодНачало;  
ЗаписьРегистра.БазовыйПериодКонец =  
СтрокаСторно.БазовыйПериодКонец;  
ЗаписьРегистра.ВидРасчета = СтрокаСторно.ВидРасчета;  
ЗаписьРегистра.Сторно = Истина;  
  
ЗаписьРегистра.Сотрудник = СтрокаСторно.Сотрудник;  
ЗаписьРегистра.Должность = СтрокаСторно.Должность;  
  
ЗаписьРегистра.Размер = СтрокаСторно.Размер;  
ЗаписьРегистра.ГрафикРаботы =  
СтрокаСторно.ГрафикРаботы;  
  
// Добавление строки в табличную часть  
СтрокаТЧ = ТабличнаяЧасть.Добавить ();  
  
СтрокаТЧ.ПериодДействияНачало =  
СтрокаСторно.ПериодДействияНачалоСторно;  
СтрокаТЧ.ПериодДействияКонец =  
СтрокаСторно.ПериодДействияКонецСторно;  
СтрокаТЧ.БазовыйПериодНачало =  
СтрокаСторно.БазовыйПериодНачало;  
СтрокаТЧ.БазовыйПериодКонец =  
СтрокаСторно.БазовыйПериодКонец;  
СтрокаТЧ.ВидРасчета = СтрокаСторно.ВидРасчета;  
СтрокаТЧ.Сторно = Истина;  
  
СтрокаТЧ.Сотрудник = СтрокаСторно.Сотрудник;  
СтрокаТЧ.Должность = СтрокаСторно.Должность;  
  
СтрокаТЧ.Размер = СтрокаСторно.Размер;  
СтрокаТЧ.ГрафикРаботы = СтрокаСторно.ГрафикРаботы;
```

Корректировка результатов расчета прошедших периодов

КонецПроцедуры

Т.е. мы добавляем новую строку уже в табличную часть (для этого в процедуру и передается третий аргумент – "ТабличнаяЧасть") и заполняем ее точно такими же значениями полей, какими заполнили запись регистра расчета. В результате в табличной части автоматически появится новая строка.

Готово.

Обновим конфигурацию базы данных и запустим ее в режиме "1С:Предприятие". Откроем документ "Расчет зарплаты 000000002" и нажмем "Провести".

В табличной части "Основные начисления" мы увидим автоматически появившуюся вторую строку:

Сотрудник	Должность	Вид расчета	Период действия начало	Период действия конец	Размер	График работы	Сторно
Ильин Сергей Викторович	Менеджер	Спецкомандировка	15.01.2014 0:00:00	22.01.2014 0:00:00	30 000,00		Нет
Ильин Сергей Викторович	Менеджер	Оклад	15.01.2014 0:00:00	22.01.2014 23:59:59	50 000,00	Пятидневка	Да

А результат расчета в регистре будет таким:

Период регистрации	Вид расчета	Сторно	Сотрудник	Должность	Результат	Размер	Дата начала периода действия	Дата окончания периода действия
01.02.2014 0:00:00	Спецкомандировка	Нет	Ильин Сергей Викторович	Менеджер	30 000,00	30 000,00	15.01.2014 0:00:00	22.01.2014 23:59:59
01.02.2014 0:00:00	Оклад	Да	Ильин Сергей Викторович	Менеджер	-13 043,48	50 000,00	15.01.2014 0:00:00	22.01.2014 23:59:59

Вот эту сумму, 13043,48 руб., и переполучил сотрудник Ильин С.В. в предыдущем периоде из-за того, что вид расчета "Спецкомандировка" не был

рассчитан вовремя. Эту сумму мы с него теперь и удерживаем в новом учетном периоде – "начисляем" со знаком минус.

Для контроля приведем полный текст модуля документа "РасчетЗарплаты" к данному моменту:

```
Процедура ОбработкаПроведения (Отказ, РежимПроведения)
    НаборОсновныеНачисления =
Движения.ОсновныеНачисления;
    НаборПремии = Движения.Премии;

    // 1. Создание наборов записей с исходными данными
    // 1.1. Основные начисления
    ТекстЗапроса =
        "ВЫБРАТЬ
        |
        РасчетЗарплатыОсновныеНачисления.Ссылка.ПериодРегистрации,
        |     РасчетЗарплатыОсновныеНачисления.Сотрудник,
        |     РасчетЗарплатыОсновныеНачисления.Должность,
        |     РасчетЗарплатыОсновныеНачисления.ВидРасчета,
        |
        РасчетЗарплатыОсновныеНачисления.ПериодДействияНачало,
        |
        КОНЕЦПЕРИОДА (РасчетЗарплатыОсновныеНачисления.ПериодДействи
яКонец, ДЕНЬ) КАК ПериодДействияКонец,
        |
        РасчетЗарплатыОсновныеНачисления.БазовыйПериодНачало,
        |     ВЫБОР
        |         КОГДА
        РасчетЗарплатыОсновныеНачисления.БазовыйПериодКонец =
ДАТАВРЕМЯ(1, 1, 1)
        |
        ТОГДА ДАТАВРЕМЯ(1, 1, 1)
        |
        ИНАЧЕ
        КОНЕЦПЕРИОДА (РасчетЗарплатыОсновныеНачисления.БазовыйПериодКонец
, ДЕНЬ)
        |
        КОНЕЦ КАК БазовыйПериодКонец,
        |
        РасчетЗарплатыОсновныеНачисления.Размер,
        |
        РасчетЗарплатыОсновныеНачисления.ГрафикРаботы,
        |
        РасчетЗарплатыОсновныеНачисления.Сторно
        |ИЗ
        |
        Документ.РасчетЗарплаты.ОсновныеНачисления КАК
        РасчетЗарплатыОсновныеНачисления
        |ГДЕ
```

```

&Ссылка | РасчетЗарплатыОсновныеНачисления.Ссылка =
|
| УПОРЯДОЧИТЬ ПО
| РасчетЗарплатыОсновныеНачисления.НомерСтроки";

Расчеты.ЗаполнитьНаборЗаписей(ТекстЗапроса, Ссылка,
НаборОсновныеНачисления);

НаборОсновныеНачисления.Записать(, Ложь);

ТаблицаСторноЗаписей =
НаборОсновныеНачисления.ПолучитьДополнение();
//ТаблицаСторноЗаписей.ВыбратьСтроку();
Для каждого СтрокаСторно из ТаблицаСторноЗаписей Цикл
ДобавитьСтрокуСторно(НаборОсновныеНачисления,
СтрокаСторно, ОсновныеНачисления);
КонецЦикла;
Записать();
НаборОсновныеНачисления.Записать(, Ложь);

// 1.2. Премии
ТекстЗапроса =
"ВЫБРАТЬ
| РасчетЗарплатыПремии.Ссылка.ПериодРегистрации,
| РасчетЗарплатыПремии.Сотрудник,
| РасчетЗарплатыПремии.Должность,
| РасчетЗарплатыПремии.ВидРасчета,
| РасчетЗарплатыПремии.БазовыйПериодНачало,
| ВЫБОР
| КОГДА
РасчетЗарплатыПремии.БазовыйПериодКонец = ДАТАВРЕМЯ(1, 1, 1)
| ТОГДА
РасчетЗарплатыПремии.БазовыйПериодКонец
| ИНАЧЕ
КОНЕЦПЕРИОДА(РасчетЗарплатыПремии.БазовыйПериодКонец, ДЕНЬ)
| КОНЕЦ КАК БазовыйПериодКонец,
| РасчетЗарплатыПремии.Размер,
| РасчетЗарплатыПремии.Сторно
| ИЗ
| Документ.РасчетЗарплаты.Премии КАК

```

```
РасчетЗарплатыПремии
    |ГДЕ
    |    РасчетЗарплатыПремии.Ссылка = &Ссылка
    |
    |УПОРЯДОЧИТЬ ПО
    |    РасчетЗарплатыПремии.НомерСтроки";

    Расчеты.ЗаполнитьНаборЗаписей(ТекстЗапроса, Ссылка,
НаборПремии);

    НаборПремии.Записать(, Ложь);

    // 2. Расчет наборов записей регистров
    Запрос = Новый Запрос();
    Запрос.УстановитьПараметр("Регистратор", Ссылка); //
для всех запросов
    // 2.1. Основные начисления
    // 2.1.1. "Первичные"
    Запрос.Текст =
"ВЫБРАТЬ
    |    ОсновныеНачисления.НомерСтроки КАК НомерСтроки,
    |    ОсновныеНачисления.ВидРасчета.СпособРасчета КАК
СпособРасчета,
    |
    |    ЕСТЬNULL(ОсновныеНачисленияДанныеГрафика.ЗначениеПериодДейс
твия, 0) КАК План,
    |
    |    ЕСТЬNULL(ОсновныеНачисленияДанныеГрафика.ЗначениеФактически
йПериодДействия, 0) КАК Отработано
    |ИЗ
    |    РегистрРасчета.ОсновныеНачисления КАК
ОсновныеНачисления
    |
    |    ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.ОсновныеНачисления.ДанныеГрафика (
    |
    |    Регистратор = &Регистратор
    |
    |    И
ВидРасчета.КатегорияРасчета = &КатегорияРасчета) КАК
ОсновныеНачисленияДанныеГрафика
    |
    |    ПО ОсновныеНачисления.НомерСтроки =
ОсновныеНачисленияДанныеГрафика.НомерСтроки
    |ГДЕ
    |    ОсновныеНачисления.Регистратор = &Регистратор
```

```

|      И ОсновныеНачисления.ВидРасчета.КатегорияРасчета
= &КатегорияРасчета
|
|      УПОРЯДОЧИТЬ ПО
|      НомерСтроки";

Запрос.УстановитьПараметр ("КатегорияРасчета",
Справочники.КатегорииРасчета.Первичное);

Выборка = Запрос.Выполнить ().Выбрать ();
Расчеты.РассчитатьЗаписиРегистраРасчета (НаборОсновные
Начисления, Выборка);
НаборОсновныеНачисления.Записать (, Истина);

// 2.1.2. "Зависимые первого уровня"
Запрос.Текст =
"ВЫБРАТЬ
|      ОсновныеНачисления.НомерСтроки КАК НомерСтроки,
|      ОсновныеНачисления.ВидРасчета.СпособРасчета КАК
СпособРасчета,
|
|      ЕСТЬNULL (ОсновныеНачисленияБазаОсновныеНачисления.Результат
База, 0) КАК СуммаБазы
|ИЗ
|      РегистрРасчета.ОсновныеНачисления КАК
ОсновныеНачисления
|      ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.ОсновныеНачисления.БазаОсновныеНачисления (
|
|      &ИзмеренияОсновногоРегистра,
|      &ИзмеренияБазовогоРегистра,
|      ,
|      Регистратор = &Регистратор
|
|      И
ВидРасчета.КатегорияРасчета = &КатегорияРасчета) КАК
ОсновныеНачисленияБазаОсновныеНачисления
|      ПО ОсновныеНачисления.НомерСтроки =
ОсновныеНачисленияБазаОсновныеНачисления.НомерСтроки
|ГДЕ
|      ОсновныеНачисления.Регистратор = &Регистратор
|      И ОсновныеНачисления.ВидРасчета.КатегорияРасчета
= &КатегорияРасчета
|

```

```
| УПОРЯДОЧИТЬ ПО
|     НомерСтроки";

    Запрос.УстановитьПараметр ("КатегорияРасчета",
Справочники.КатегорииРасчета.ЗависимоеПервогоУровня);

    Измерения = Новый Массив ();
    Измерения.Добавить ("Сотрудник");
    Измерения.Добавить ("Должность");

    Запрос.УстановитьПараметр ("ИзмеренияОсновногоРегистра
", Измерения);
    Запрос.УстановитьПараметр ("ИзмеренияБазовогоРегистра"
, Измерения);

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

    Расчеты.РассчитатьЗаписиРегистраРасчета (НаборОсновные
Начисления, Выборка);

    НаборОсновныеНачисления.Записать (, Истина);

// 2.2. Премии
// 2.2.1. "Первичные"
    РассчитатьПремии (НаборПремии, Запрос,
Справочники.КатегорииРасчета.Первичное);
// 2.2.2. "Зависимые первого уровня"
    РассчитатьПремии (НаборПремии, Запрос,
Справочники.КатегорииРасчета.ЗависимоеПервогоУровня);
// 2.2.3. "Зависимые второго уровня"
    РассчитатьПремии (НаборПремии, Запрос,
Справочники.КатегорииРасчета.ЗависимоеВторогоУровня);
    КонецПроцедуры

Процедура РассчитатьПремии (НаборЗаписей, Запрос,
КатегорияРасчета)
    Запрос.Текст =
    "ВЫБРАТЬ
    |     Премии.НомерСтроки КАК НомерСтроки,
    |     Премии.ВидРасчета.СпособРасчета КАК
СпособРасчета,
```



```

|
|      ЕСТЬNULL (ПремииБазаОсновныеНачисления.РезультатБаза, 0) +
|      ЕСТЬNULL (ПремииБазаПремии.РезультатБаза, 0) КАК СуммаБазы
|
|      ИЗ
|
|      РегистрРасчета.Премии КАК Премии
|
|      ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.Премии.БазаОсновныеНачисления (
|
|      &ИзмеренияОсновногоРегистра,
|
|      &ИзмеренияБазовогоРегистра,
|
|      ,
|
|      Регистратор = &Регистратор
|
|      И
ВидРасчета.КатегорияРасчета = &КатегорияРасчета) КАК
ПремииБазаОсновныеНачисления
|
|      ПО Премии.НомерСтроки =
ПремииБазаОсновныеНачисления.НомерСтроки
|
|      ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.Премии.БазаПремии (
|
|      &ИзмеренияОсновногоРегистра,
|
|      &ИзмеренияБазовогоРегистра,
|
|      ,
|
|      Регистратор = &Регистратор
|
|      И
ВидРасчета.КатегорияРасчета = &КатегорияРасчета) КАК
ПремииБазаПремии
|
|      ПО Премии.НомерСтроки =
ПремииБазаПремии.НомерСтроки
|
|      ГДЕ
|
|      Премии.Регистратор = &Регистратор
|
|      И Премии.ВидРасчета.КатегорияРасчета =
&КатегорияРасчета
|
|      |УПОРЯДОЧИТЬ ПО
|
|      НомерСтроки";
|
|
|      Запрос.УстановитьПараметр ("КатегорияРасчета",
КатегорияРасчета);
|
|
|      Выборка = Запрос.Выполнить ().Выбрать ();
|
|
|      Расчеты.РассчитатьЗаписиРегистраРасчета (НаборЗаписей,
Выборка);

```

```
НаборЗаписей.Записать (, Истина);

КонецПроцедуры

Процедура ДобавитьСтрокуСторно (НаборЗаписей, СтрокаСторно,
ТабличнаяЧасть)

// Добавление записи в набор записей
ЗаписьРегистра = НаборЗаписей.Добавить ();

ЗаписьРегистра.ПериодРегистрации =
СтрокаСторно.ПериодРегистрацииСторно;
ЗаписьРегистра.ПериодДействияНачало =
СтрокаСторно.ПериодДействияНачалоСторно;
ЗаписьРегистра.ПериодДействияКонец =
СтрокаСторно.ПериодДействияКонецСторно;
ЗаписьРегистра.БазовыйПериодНачало =
СтрокаСторно.БазовыйПериодНачало;
ЗаписьРегистра.БазовыйПериодКонец =
СтрокаСторно.БазовыйПериодКонец;
ЗаписьРегистра.ВидРасчета = СтрокаСторно.ВидРасчета;
ЗаписьРегистра.Сторно = Истина;

ЗаписьРегистра.Сотрудник = СтрокаСторно.Сотрудник;
ЗаписьРегистра.Должность = СтрокаСторно.Должность;

ЗаписьРегистра.Размер = СтрокаСторно.Размер;
ЗаписьРегистра.ГрафикРаботы =
СтрокаСторно.ГрафикРаботы;

// Добавление строки в табличную часть
СтрокаТЧ = ТабличнаяЧасть.Добавить ();

СтрокаТЧ.ПериодДействияНачало =
СтрокаСторно.ПериодДействияНачалоСторно;
СтрокаТЧ.ПериодДействияКонец =
СтрокаСторно.ПериодДействияКонецСторно;
СтрокаТЧ.БазовыйПериодНачало =
СтрокаСторно.БазовыйПериодНачало;
СтрокаТЧ.БазовыйПериодКонец =
СтрокаСторно.БазовыйПериодКонец;
```

СтрокаТЧ.ВидРасчета = СтрокаСторно.ВидРасчета;

СтрокаТЧ.Сторно = Истина;

СтрокаТЧ.Сотрудник = СтрокаСторно.Сотрудник;

СтрокаТЧ.Должность = СтрокаСторно.Должность;

СтрокаТЧ.Размер = СтрокаСторно.Размер;

СтрокаТЧ.ГрафикРаботы = СтрокаСторно.ГрафикРаботы;

КонецПроцедуры

13. Перерасчеты

13.1. Зависимость по перерасчету

Как мы могли убедиться неоднократно, результаты одних видов расчета могут зависеть от результатов других видов расчета.

Отсюда следует очевидный вывод: если возникает необходимость пересчитать те виды расчета, от которых зависят другие, то мы должны пересчитать и все зависящие от них виды расчета.

Например, если после расчета зарплаты была обнаружена ошибка ввода исходных данных вида расчета "Оклад" для сотрудника Ильина С.В., то нужно пересчитать не только оклад, но также премии и все зависящие от оклада виды расчета Ильина С.В., и т.п.

Важно!

Будем говорить, что, если после пересчете вида расчета "А" необходимо пересчитать также вид расчета "Б", то вид расчета "Б" **зависит по перерасчету** от вида расчета "А".

13.2. Зависимость по перерасчету шире, чем зависимости по периоду действия и по базе

Нам известны две основные зависимости, которыми могут быть связаны виды расчета: зависимость по базе и зависимость по периоду действия.

Очевидно, что, если вид расчета "Оклад" является базовым для вида расчета "Доплата за квалификацию", то "Доплата за квалификацию" зависит по перерасчету от "Оклад". При пересчете оклада, скорее всего, изменилась сумма результата оклада, а это влечет за собой изменение суммы базы для "Доплата за квалификацию", что делает необходимым пересчет также вида расчета "Доплата за квалификацию".

Также, если было обнаружено, что период действия вида расчета "Оклад" на выезде введен неправильно, то после перерасчета "Оклад" на выезде надо пересчитать также "Оклад", поскольку у него, скорее всего, изменится фактический период действия.

Исчерпывается ли зависимость по перерасчету двумя известными нам зависимостями? – Нет.

Пример. Мы знаем, что зависимость по базе может быть многоуровневой. Например, вид расчета "Поощрительная надбавка" зависит по базе только от вида расчета "Месячная премия". От вида расчета "Оклад" он по базе не зависит – в его табличной части "Базовые виды расчета" нет вида расчета "Оклад". Однако кто будет спорить, что, если был пересчитан оклад, то надо пересчитывать также и поощрительную надбавку? Ведь она, хотя непосредственно не зависит по базе от оклада, зависит от него опосредованно – через базовую зависимость своего базового вида расчета "Месячная премия".

Возможны и другие случаи, когда виды расчета непосредственно не связаны друг с другом ни зависимостью по базе, ни зависимостью по периоду действия, но при перерасчете одного вида расчета надо пересчитывать и другой.

Все это говорит о том, что зависимость по перерасчету шире зависимостей по базе и по периоду действия.

А поскольку реализация перерасчетов на практике при всей их интуитивной понятности довольно сложна, мы в методических целях наметим только пути решения некоторых проблем, которые возникают по ходу реализации механизма перерасчетов в прикладном решении. Иными словами, мы сделаем некий "модельный пример", который даст нам общее представление о путях реализации перерасчетов и трудностях, которые на этих путях возникают.

Конкретно: мы решим задачу получения запросом необходимых данных для выполнения перерасчетов. А дальнейший путь программирования перерасчетов при нашем алгоритмическом подходе будет уже очевиден.

Нам также придется принять ряд условных **учебных допущений**, без которых учебный материал по перерасчетам может неоправданно усложниться.

13.3. Стандартная табличная часть "ВедущиеВидыРасчета"

Поскольку в силу широты зависимости по перерасчету платформе установить ее автоматически затруднительно, то ответственность за установление этой зависимости возлагается на пользователя.

Для этой цели объект конфигурации "План видов расчета" имеет в своем составе стандартную табличную часть, которая называется "ВедущиеВидыРасчета".

Табличная часть "ВедущиеВидыРасчета", в отличие от двух других уже знакомых нам стандартных табличных частей, присутствует в плане видов расчета всегда, ее наличие никак не зависит от настроек свойств метаданных плана видов расчета. Таким образом, план видов расчета как объект конфигурации всегда имеет от одной до трех стандартных табличных частей (и сколько угодно прикладных).

13.4. Заполнение табличной части "Ведущие виды расчета"

Сделаем первые учебные допущения.

Необходимость перерасчета при изменении результатов базовых видов расчета наиболее очевидна. В связи с этим мы исследуем перерасчеты именно на базовых видах расчета для начислений г-на Ильина С.В.

В плане видов расчета "Основные начисления" вид расчета "Доплата за квалификацию" г-ну Ильину не начислялся. Следовательно, первое, что нужно сделать в планах видов расчета "Основные начисления" и "Премии" – это заполнить для зависящих от базы видов расчета, которые мы рассчитывали Ильину С.В., табличную часть "Ведущие виды расчета" составом их базовых видов расчета.

Это такие виды расчета из плана видов расчета "Премии":

Вид расчета	Ведущие виды расчета
Месячная премия	Оклад (Основные начисления)
	Оклад на выезде (Основные начисления)
	Персональная премия (Премии)
Премия за период	Оклад (Основные начисления)

В режиме "1С:Предприятие" откроем план видов расчета "Премии" и заполним табличную часть "Ведущие виды расчета" для указанных двух видов расчета:

Месячная премия (Премии)

Записать и закрыть Записать Еще ▾

Код: 00002

Наименование: Месячная премия

Способ расчета: Процентом ▾

Категория расчета: Зависимое первого уровня ▾

Базовые виды расчетов Ведущие виды расчетов

Добавить Еще ▾

Вид расчета	Ведущие виды расчета
Оклад	
Оклад на выезде	
Персональная премия	

Премия за период (Премии)

Записать и закрыть Записать Еще ▾

Код: 00004

Наименование: Премия за период

Способ расчета: Процентом ▾

Категория расчета: Зависимое первого уровня ▾

Базовые виды расчетов Ведущие виды расчетов

Добавить Еще ▾

Вид расчета
Оклад

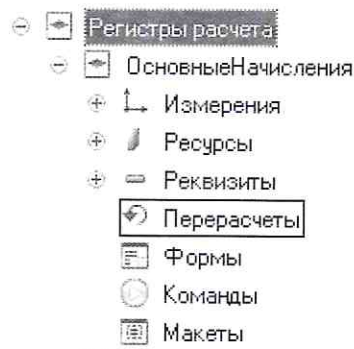
То же самое сделаем и для другого плана видов расчета – "Основные начисления".

Там нас интересует вид расчета "Доплата за квалификацию". У него два базовых вида расчета: "Оклад" и "Оклад на выезде", причем оба из одного плана видов расчета – "Основные начисления". Их мы поместим в табличную часть "Ведущие виды расчета":

13.5. Объект конфигурации "Перерасчет"

Платформа оказывает прикладному программисту содействие в программировании механизмов перерасчета. Для этого в конфигурации создаются и настраиваются объекты "Перерасчет".

Объект метаданных "Перерасчет" логически подчиняется объекту "РегистрРасчета". В дереве метаданных у регистра расчета имеется ветвь "Перерасчеты":



В эту ветвь добавляются объекты конфигурации "Перерасчет".

Объект "Перерасчет" – это таблица информационной базы, т.е. перерасчет является персистентным объектом. Работа платформы с перерасчетом происходит так. Когда происходит перерасчет результата вида расчета "А", который находится в табличной части "ВедущиеВидыРасчета" другого вида расчета "Б", то платформа

помещает строку с видом расчета "Б" и сопутствующей информацией в таблицу перерасчета. Подразумевается, что прикладная программа должна прочитать данные таблицы перерасчета и перерассчитать вид расчета "Б".

Как мы видим, платформа сама ничего не перерасчитывает, а только дает исходную информацию о видах расчета, которые надо бы перерассчитать. А вся ответственность за выполнение перерасчета возлагается на прикладную программу. Это так называемый **уведомительный принцип**, который мы уже видели при создании сторно-записей: платформа сама ничего не сторнирует, но только сообщает прикладной программе в таблице сторно-записей всю необходимую информацию.

13.5.1. Структура таблицы информационной базы объекта "Перерасчет"

Объект конфигурации "Перерасчет" – это таблица, которая имеет следующую структуру полей:

"ВидРасчета";

"ОбъектПерерасчета";

<Измерение>;

<Измерение>;

.

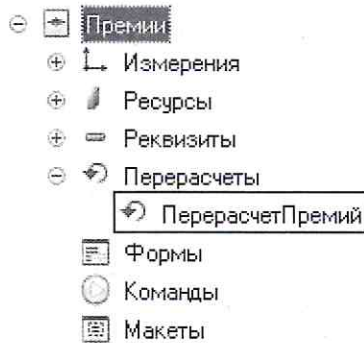
Поля "ВидРасчета" и "ОбъектПерерасчета" – стандартные, они создаются Конфигуратором автоматически

В каждой строке таблицы перерасчета "ВидРасчета" – это вид расчета, который предлагается перерассчитать, а "ОбъектПерерасчета" – ссылка на документ-регистратор, который сделал в регистре расчета движение с данным видом расчета.

Естественно, что "просто вид расчета" пересчитывать смысла нет, вид расчета принадлежит определенному сотруднику. Поэтому, как и у регистра расчета, у перерасчета надо создать набор измерений, который соответствует набору измерений регистра расчета – "хозяина" перерасчета. Так, перерасчитывать имеет смысл не просто вид расчета "Месячная премия", а "Месячная премия" для сотрудника Ильин С.В., работающего в должности "Менеджер". Такой состав измерений нужно создать и у перерасчета.

13.5.2. Создание объекта "ПерерасчетПремий"

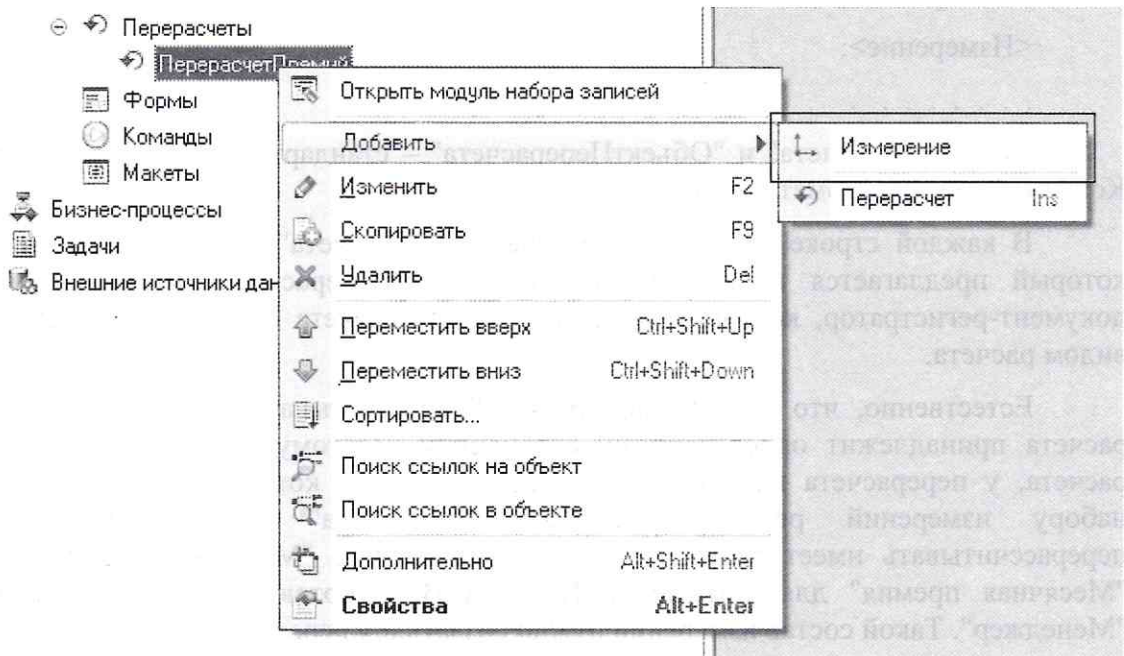
В Конфигураторе установим выделение на ветвь "Перерасчеты" регистра расчета "Премии" и новый перерасчет. В палитре свойства дадим ему имя "ПерерасчетПремий":



Сам по себе объект "Перерасчет", кроме задания имени, настроек от нас не потребует. Однако надо создать измерения перерасчета – по числу измерений регистра "Премии": "Сотрудник" и "Должность".

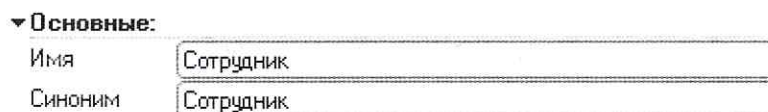
13.5.3. Создание измерения перерасчета

Измерения перерасчета создаются через контекстное меню. Правой клавишей мыши откроем контекстное меню на объекте "ПерерасчетПремий" и выберем пункт меню "Добавить" / "Измерение":



Будет создано измерение перерасчета. Настроим его свойства.

В палитре свойств измерения перерасчета зададим ему имя: "Сотрудник".

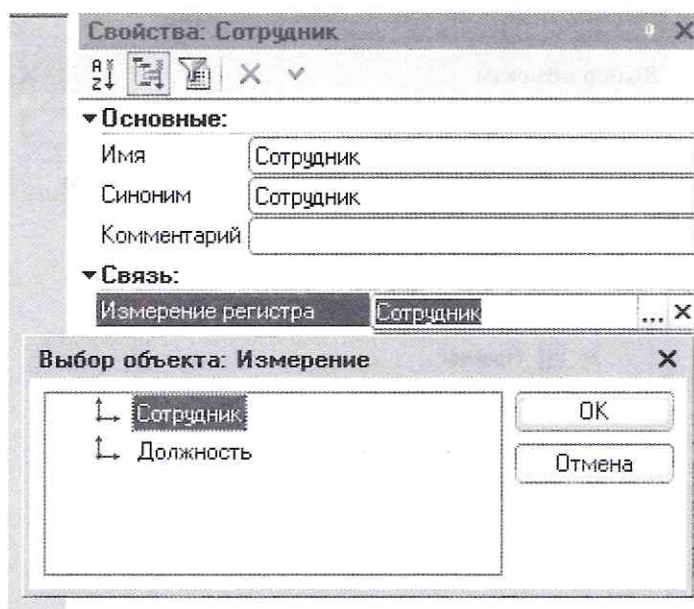


13.5.4. Установление соответствия измерений перерасчета измерениям регистров расчета

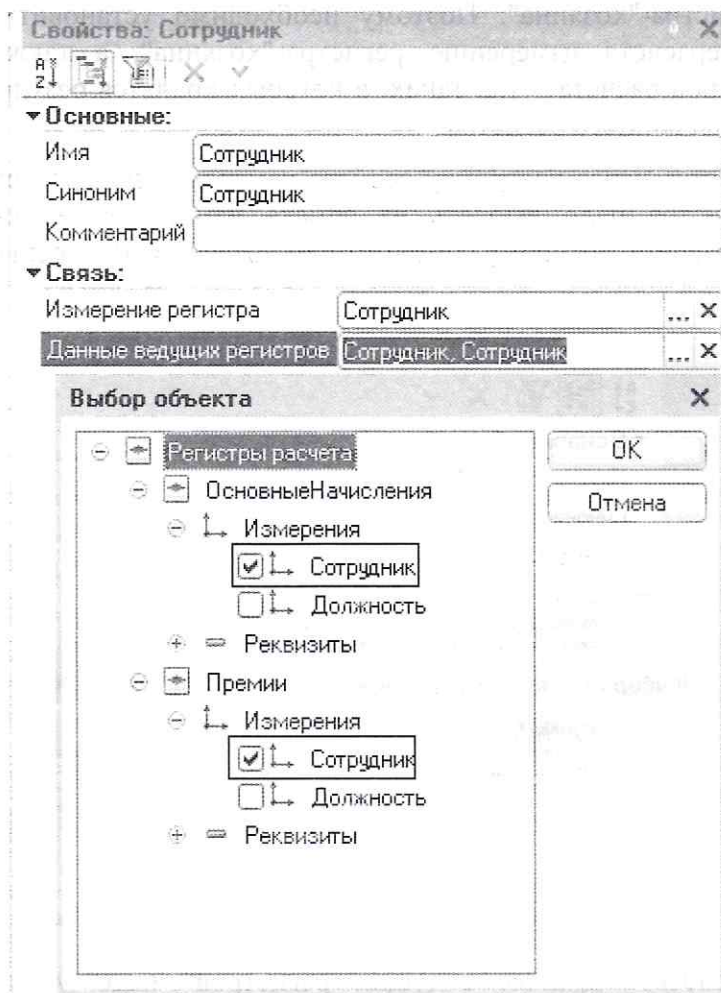
При создании измерений перерасчета возникает вопрос, с которым мы уже сталкивались, когда настраивали в запросе виртуальную таблицу "База" регистра расчета. А именно: имена измерений перерасчета не обязаны совпадать с именами

измерений регистра-"хозяина". Поэтому необходимо установить соответствие изменения перерасчета измерению регистра-"хозяина", а также измерениям ведущих регистров расчета – т.е. таких, в которых хранятся результаты ведущих видов расчета для данного регистра.

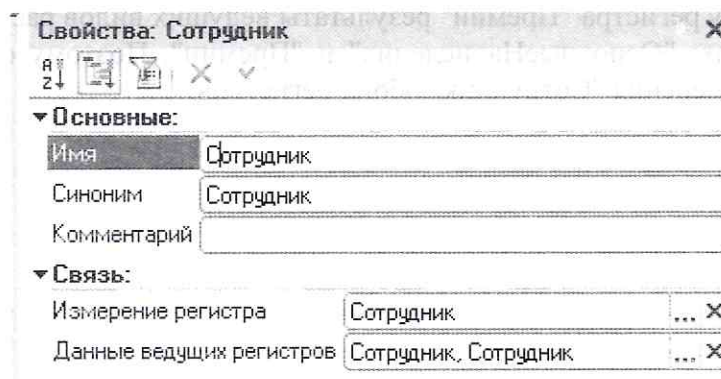
Это делается через настройку свойств измерения перерасчета. В свойстве Измерение регистра надо выбрать из списка измерение регистра расчета - "хозяина" перерасчета, которому соответствует созданное измерение перерасчета: "Сотрудник":



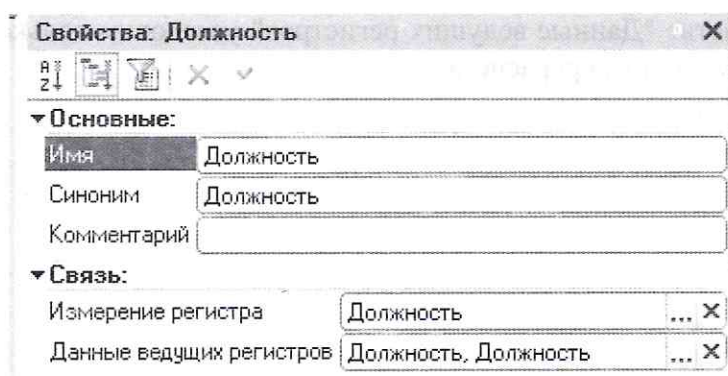
А в свойстве "Данные ведущих регистров" надо указать флажками соответствие этого измерения перерасчета с измерениями всех ведущих регистров для данного регистра расчета. Форма структуры ведущих регистров открывается нажатием на кнопку выбора в поле свойства "Данные ведущих регистров". Напомним, что у регистра "Премии" результаты ведущих видов расчета находятся в двух регистрах: "ОсновныеНачисления" и "Премии". Поэтому флажки нужно установить у измерения "Сотрудник" обоих регистров:



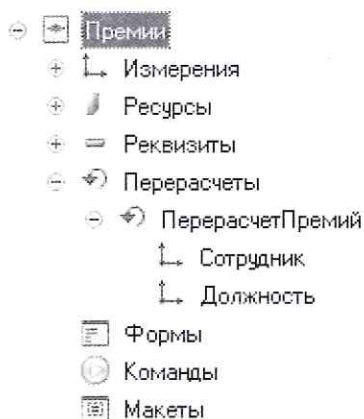
Настройки свойств измерения перерасчета "Сотрудник" примут следующий вид:



Точно таким же образом создается и настраивается второе измерение перерасчета – "Должность":



В результате создания и настройки перерасчета структура регистра "Премии" в дереве метаданных конфигурации стала такой:

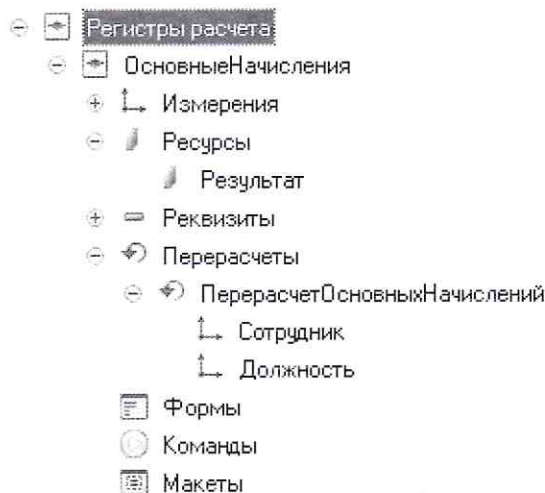


Конфигурирование объекта "ПерерасчетПремий" завершено.

13.5.5. Создание объекта "ПерерасчетОсновныхНачислений"

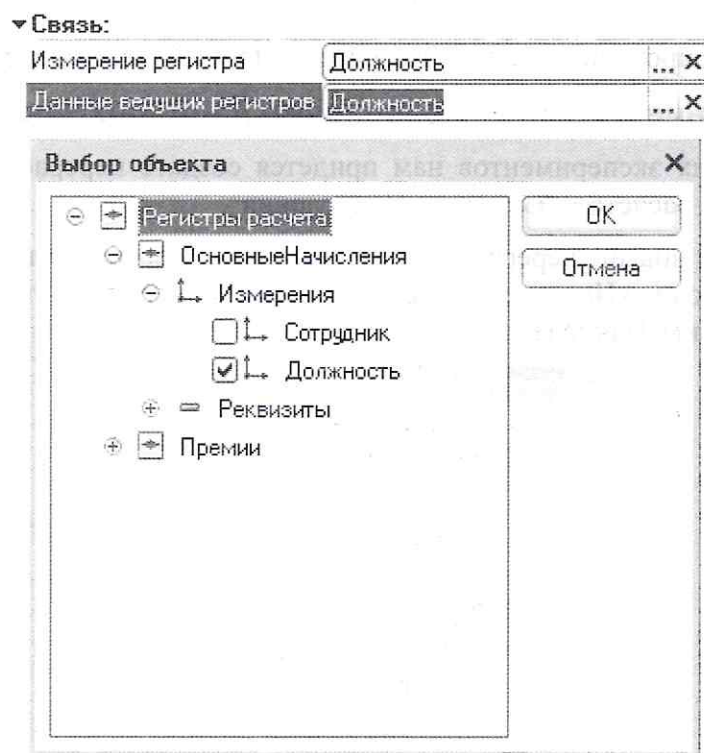
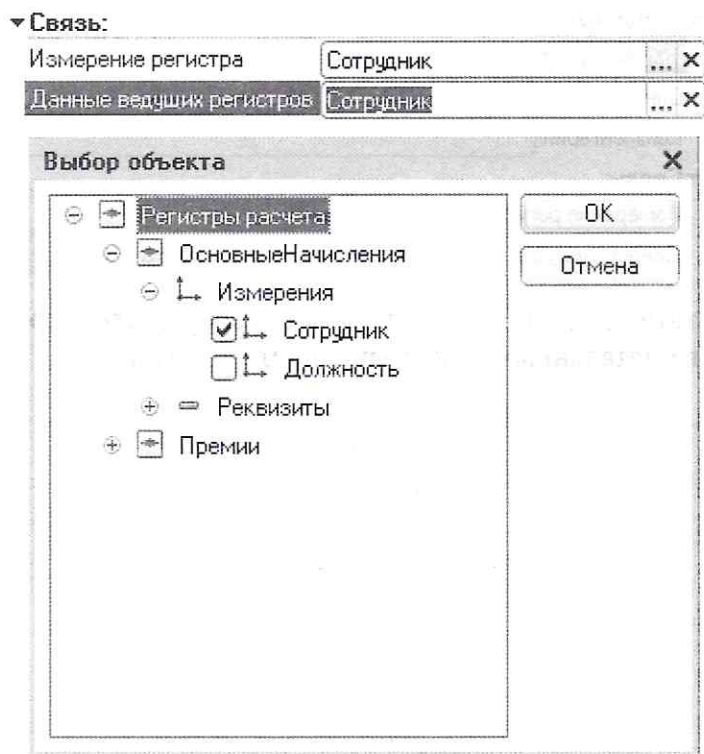
Для наших экспериментов нам придется создать перерасчет и у другого нашего регистра расчета – "ОсновныеНачисления".

Добавим новый перерасчет в его ветвь "Перерасчеты", назовем его "ПерерасчетОсновныхНачислений", создадим ему измерения – такие же, как и у регистра "ОсновныеНачисления":



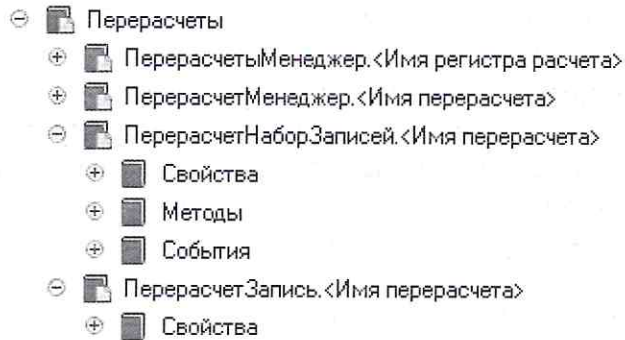
Таким же образом, как мы делали выше, зададим соответствие измерений перерасчета измерениям регистра-"хозяина" и ведущих регистров. Только вспомним, что у регистра "ОсновныеНачисления" ведущий регистр один – он сам.

Поэтому в свойстве "Данные ведущих регистров" установим только один флажок для каждого измерения перерасчета:



13.6. Объектная модель работы с перерасчетом

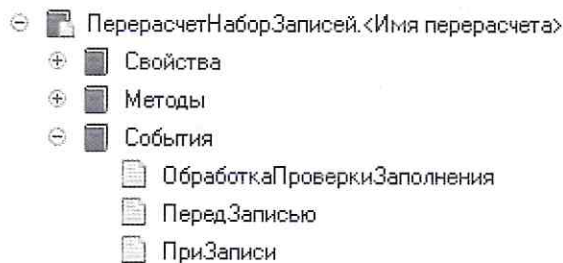
Если посмотреть в Синтакс-помощник, то можно заметить почти полное сходство объектной модели перерасчета с объектной моделью любого регистра платформы 1С:



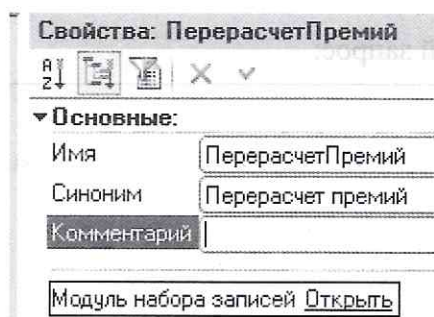
Имеется объект – менеджер перерасчета, объект "ПерерасчетНаборЗаписей", который, подобно набору записей регистра, является коллекцией объектов "ПерерасчетЗапись".

Объект "ПерерасчетНаборЗаписей" имеет метод "Добавить", который позволяет из прикладной программы создать новую запись перерасчета и заполнить ее свойства, метод "Записать" для записи набора перерасчета в информационную базу, метод "Удалить" для удаления записи из перерасчета.

Объект "ПерерасчетНаборЗаписей" также позволяет программировать обработчики событий, а состав событий для этого объекта аналогичен составу событий для регистра:



Модуль набора записей перерасчета, где можно запрограммировать обработчики событий, открывается из палитры свойств перерасчета в Конфигураторе:



Короче, программная работа с перерасчетом аналогична программной работе с регистром. Из этого следует, что не только платформа, но и прикладная программа может изменять состав и содержание записей перерасчета.

Для чего дана такая возможность?

Предположим, на предприятии некоторым сотрудникам начисляется зарплата по виду расчета "Агентское вознаграждение". Т.е. эти сотрудники занимаются заключением сделок, например, по продаже недвижимости, а результат этого вида расчета представляет собой определенный процент от оборота суммы заключенных агентом за расчетный месяц сделок.

Очевидно, обороты по сделкам с недвижимостью хранятся не в регистре расчета, а, скорее всего, в каком-то оборотном регистре накопления. Следовательно, механизму перерасчета непосредственно не может быть известно об изменении суммы сделок, которое произошло, например, при перепроведении документа по регистру оборотов сделок – регистры расчета не имеют никакой связи с регистрами накопления. Значит, запись в таблицу перерасчета для пересчета результата вида расчета "Агентское вознаграждение" автоматически не попадет и может быть сформирована только прикладным кодом, который должен получать управление после перепроведения документа по регистру накопления.

13.7. Создание перерасчетной ситуации и первая проблема программной реализации перерасчета

Мы пока обойдемся без программного вмешательства в состав записей перерасчета.

Сначала разработаем запрос, который позволит нам анализировать таблицу перерасчета. Откроем Консоль запросов, добавим в наш список запросов новый запрос, назовем его "Перерасчет".

Открыв Конструктор запросов, мы увидим, что таблицы перерасчета доступны как исходные таблицы запроса:

- + [] РегистрыРасчета
- [] Перерасчеты
 - [] Премии.ПерерасчетПремий
 - + [] ОбъектПерерасчета
 - + [] ВидРасчета
 - + [] Сотрудник
 - + [] Должность

Создадим запрос-объединение выборок всех полей из обеих таблиц перерасчета.

Получится вот такой запрос:

```
ВЫБРАТЬ
    ПерерасчетПремий.ОбъектПерерасчета,
    ПерерасчетПремий.ВидРасчета,
    ПерерасчетПремий.Сотрудник,
    ПерерасчетПремий.Должность
ИЗ
    РегистрРасчета.Премии.ПерерасчетПремий          КАК
ПерерасчетПремий

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ
    ПерерасчетОсновныхНачислений.ОбъектПерерасчета,
```


ПерерасчетОсновныхНачислений.ВидРасчета, ПерерасчетОсновныхНачислений.Сотрудник, ПерерасчетОсновныхНачислений.Должность ИЗ РегистрРасчета.ОсновныеНачисления.ПерерасчетОсновныхНачислений КАК ПерерасчетОсновныхНачислений
--

Теперь создадим перерасчетную ситуацию.

Откроем документ "Расчет зарплаты 000000002" – документ, в котором мы рассчитывали вид расчета "Спецпоручение" и сторно-оклад за прошлый месяц для Ильина С.В. Перепроведем документ, при проведении он будет перерасчитан.

Теперь выполним наш запрос к таблице перерасчета. Получим результат запроса:

ОбъектПерерасчета	ВидРасчета	Сотрудник	Должность
Расчет зарплаты 000000001	Месячная премия	Ильин Сергей Викторович	Менеджер
Расчет зарплаты 000000001	Премия за период	Ильин Сергей Викторович	Менеджер

Результат вполне обоснован. Документ, который мы только что пересчитали, содержит вид расчета "Оклад" (сторно). Период действия этого вида расчета лежит в пределах периода действия вида расчета "Оклад" для менеджера Ильина С.В. за прошедший месяц. Вид расчета "Оклад" является ведущим видом расчета для видов расчета "Месячная премия" и "Премия за период". Вот эти виды расчета платформа и поместила в таблицу перерасчета. Вполне логично: мог измениться базовый вид расчета "Оклад", который мы ввели как ведущий вид расчета для премий – вот нам платформа и предлагает перерассчитать эти премии.

Ну, раз просят пересчитать, давайте пересчитаем.

Откроем документ "Расчет зарплаты 000000001", в котором мы рассчитывали премии для г-на Ильина, и перепроведем его. Т.е., будет выполнен перерасчет этих видов расчета.

Теперь снова выполним запрос по таблице перерасчета.

И мы увидим результат, который не может не поставить нас в тупик:

ОбъектПерерасчета	ВидРасчета	Сотрудник	Должность
Расчет зарплаты 000000002	Оклад	Ильин Сергей Викторович	Менеджер

В результат запроса попала запись документа "Расчет зарплаты 000000002", которым мы рассчитывали сторно-оклад для г-на Ильина.

Чтобы исключить случайность, точно так же снова пересчитаем документ "Расчет зарплаты 000000002". Потом снова выполним наш запрос. И снова увидим:

ОбъектПерерасчета	ВидРасчета	Сотрудник	Должность
Расчет зарплаты 000000001	Месячная премия	Ильин Сергей Викторович	Менеджер
Расчет зарплаты 000000001	Премия за период	Ильин Сергей Викторович	Менеджер

Очевидно, получается какой-то порочный круг. Пересчитываем документ 000000001 – в перерасчет попадают записи документа 000000002,

пересчитываем документ 000000002 – в перерасчет снова попадают те же записи документа 000000001...

Видимо, мы допускаем какую-то логическую ошибку. Какую же?

Результат запроса показывает, что таблицы перерасчета содержат записи только о двух видах расчета – "Месячная премия" и "Премия за период". А мы что делаем? – Мы пересчитываем весь документ 000000001. В том числе и вид расчета "Оклад" для Ильина, который нас никто не просил пересчитывать и который пересекается по периоду действия с видом расчета "Оклад" из документа 000000002. Вот этот вид расчета и попадает в таблицу перерасчета. И понятно, что, если мы снова пересчитаем весь документ 000000002 с этим видом расчета, то цепочка событий повторится: в таблицу перерасчета снова попадут премии, для которых вид расчета "Оклад" является ведущим.

Итак, мы имеем первую проблему, которую надо решать при программной реализации перерасчетов.

Важно!

Необходимо пересчитывать не весь документ, который указан в поле "ОбъектПерерасчета" таблицы перерасчета, а **только те виды расчета, которые попали в перерасчет.**

13.8. Вторая проблема программной реализации перерасчета

Предположим, что менеджеру Ильину С.В. вид расчета "Премия" за период начислялся за месяц два раза – ну мог же он два раза в месяц как-то особо отличиться. Реализуем этот случай: добавим в табличную часть "Премии" документа "Расчет зарплаты 000000001" еще одну строку:

Сотрудник	Должность	Вид расчета	Базовый период начало	Базовый период конец	Размер
Ильин Сергей Викторович	Менеджер	Премия за период	27.01.2014 0:00:00	31.01.2014 0:00:00	15,00

Т.е. менеджер Ильин С.В. получает премию за два периода в одном и том же месяце, второй раз за период с 27 по 31 января. Проведем этот документ, увидим, что результат расчета по этому начислению составляет 1 630,43 руб.

Теперь воспроизведем перерасчетную ситуацию, как мы это делали: проведем заново документ "Расчет зарплаты 000000002".

Выполним наш запрос по таблицам перерасчетов. Увидим уже знакомый результат:

ОбъектПерерасчета	ВидРасчета	Сотрудник	Должность
Расчет зарплаты 000000001	Месячная премия	Ильин Сергей Викторович	Менеджер
Расчет зарплаты 000000001	Премия за период	Ильин Сергей Викторович	Менеджер

Как мы видим, в перерасчете имеется только одна строка "Премия за период", хотя мы начисляли ее сотруднику за месяц два раза. И не мудрено: ведь

в таблице перерасчета нет поля "НомерСтроки", поэтому в ней нет возможности указать записи регистра, подлежащие перерасчету.

Имеем вторую проблему, которую надо решать.

Важно!

Таблица перерасчета идентифицирует не запись регистра расчета, которую надо перерассчитать, а только вид расчета. Если таких видов расчета в наборе записей регистра несколько, то пересчитать надо их все.

Конечно, это не столько "проблема", сколько логическое следствие из структуры таблицы перерасчета. Однако здесь можно допустить программную ошибку: легко забыть о том, что в регистре расчета для сотрудника может быть несколько записей с одинаковым видом расчета и на основании единственности записи в таблице перерасчета и пересчитать только одну запись регистра с этим видом расчета.

13.9. Запрос для получения необходимых данных записей, подлежащих перерасчету

К счастью, при той схеме расчетного алгоритма, которую мы приняли в нашем курсе, составить запрос для перерасчета не является такой уж сложной задачей.

Вспомним, как мы строили запрос, который давал нам выборку необходимых данных, обрабатываемую процедурой общего модуля "РассчитатьЗаписиРегистраРасчета". Это было левое соединение персистентной таблицы регистра расчета с виртуальными таблицами. Персистентная таблица обеспечивала состав (номера) строк набора записей рассчитываемого документа и способ расчета, применяемый к каждой строке, а виртуальные таблицы давали необходимые данные. Например:

```
ВЫБРАТЬ
    Премии.НомерСтроки КАК НомерСтроки,
    Премии.ВидРасчета.СпособРасчета КАК СпособРасчета,
    ЕСТЬNULL (ПремииБазаОсновныеНачисления.РезультатБаза,
0) + ЕСТЬNULL (ПремииБазаПремии.РезультатБаза, 0) КАК СуммаБазы
ИЗ
    РегистрРасчета.Премии КАК Премии
        ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.Премии.БазаОсновныеНачисления (
            &ИзмеренияОсновногоРегистра,
            &ИзмеренияБазовогоРегистра,
            ,
            Регистратор = &Регистратор
            И ВидРасчета.КатегорияРасчета =
&КатегорияРасчета) КАК ПремииБазаОсновныеНачисления
        ПО Премии.НомерСтроки =
ПремииБазаОсновныеНачисления.НомерСтроки
        ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.Премии.БазаПремии (
            &ИзмеренияОсновногоРегистра,
            &ИзмеренияБазовогоРегистра,
            ,
            Регистратор = &Регистратор
            И ВидРасчета.КатегорияРасчета =
&КатегорияРасчета) КАК ПремииБазаПремии
        ПО Премии.НомерСтроки =
ПремииБазаПремии.НомерСтроки
ГДЕ
    Премии.Регистратор = &Регистратор
    И Премии.ВидРасчета.КатегорияРасчета =
&КатегорияРасчета

УПОРЯДОЧИТЬ ПО
    НомерСтроки
```

И общее содержание результата запроса:

НомерСтроки	СпособРасчета	< Не о б х о д и м ы е д а н н ы е ... >		
n1	ПоОкладу
n2	ПоОкладу
...	Процентом
...

Еще раз подчеркнем "задающую" роль левой, персистентной, таблицы запроса: состав номеров записей регистра, подлежащих расчету, и способов расчета для этих записей задает именно эта – левая таблица.

Следовательно, если эта левая таблица будет содержать не все записи набора рассчитываемого документа с 1 по N, а только подлежащие перерасчету, то такой запрос решит первую обозначенную нами проблему.

Как можно получить такую левую таблицу? В нашем распоряжении есть таблицы перерасчетов с видами расчета, которые мы должны перерассчитать. Вспомним, какой результат дает операция ВНУТРЕННЕЕ СОЕДИНЕНИЕ языка запросов... Получается, выход найден: нужно получить выборку данных таблиц перерасчета и сделать внутреннее соединение с персистентной таблицей регистра по равенству полей "ВидРасчета" в обеих таблицах и полей "Регистратор" и "ОбъектПерерасчета" соответственно в таблице регистра и перерасчета, а также всех измерений регистра и перерасчета.

Результат такого запроса даст нам состав как раз тех записей регистра расчета, которые подлежат перерасчету, с их способами расчета.

И уже таблицу этого результата можно использовать в качестве "задающей" (левой) таблицы для получения необходимых данных.

В качестве примера построим запрос для получения записей, подлежащих перерасчету, для регистра "Премии". Пойдем по этапам.

13.9.1. "Задающий" запрос – первый запрос пакета

На первом этапе построим в Консоли запросов "задающий" запрос, который даст нам состав только тех записей регистра, которые попали в перерасчет. Назовем его в списке запросов "Подлежат перерасчету":

```

ВЫБРАТЬ
    Премии.НомерСтроки,
    Премии.ВидРасчета.СпособРасчета КАК СпособРасчета
ИЗ
    РегистрРасчета.Премии КАК Премии
    ВНУТРЕННЕЕ СОЕДИНЕНИЕ
    РегистрРасчета.Премии.ПерерасчетПремий КАК ПерерасчетПремий
    ПО Премии.Регистратор =
    ПерерасчетПремий.ОбъектПерерасчета
        И Премии.ВидРасчета =
    ПерерасчетПремий.ВидРасчета
        И Премии.Сотрудник =
    ПерерасчетПремий.Сотрудник
        И Премии.Должность =
    ПерерасчетПремий.Должность
ГДЕ
    Премии.Регистратор = &Регистратор
    И Премии.ВидРасчета.КатегорияРасчета =
    &КатегорияРасчета
    
```

Еще раз выполним запрос по таблицам перерасчетов. Убедимся, что в таблицах перерасчетов присутствуют записи по двум видам расчетов из регистра Премии:

ОбъектПерерасчета	ВидРасчета	Сотрудник	Должность
Расчет зарплаты 000000001	Месячная премия	Ильин Сергей Викторович	Менеджер
Расчет зарплаты 000000001	Премия за период	Ильин Сергей Викторович	Менеджер

А теперь выполним запрос "Подлежат перерасчету":

НомерСтроки	СпособРасчета
3	Процентом
7	Процентом
6	Процентом

Строка 3 – это, как можно удостовериться, запись по виду расчета Месячная премия для менеджера Ильина С.В.

А строки 6 и 7 – это обе записи по виду расчета "Премия за период" менеджера Ильина С.В. Подчеркнем: **обе!** Как мы видим, разработанный нами запрос построен так, что дает все записи по виду расчета, попавшему в таблицу перерасчета. Итак, попутно решена и вторая поставленная нами проблема: наш запрос выдает номера строк по всем записям видов расчета, попавших в перерасчет.

Итак, "задающая" (левая) таблица запроса необходимых данных для выполнения перерасчета в принципе получена.

13.9.2. Окончательный запрос

Дальнейшие шаги уже очевидны. Оформим разработанный запрос как запрос на получение временной таблицы и построим второй запрос пакета:

```

ВЫБРАТЬ
    Премии.НомерСтроки,
    Премии.ВидРасчета.СпособРасчета КАК СпособРасчета
ПОМЕСТИТЬ ПодлежатПерерасчету
ИЗ
    РегистрРасчета.Премии КАК Премии
    ВНУТРЕННЕЕ СОЕДИНЕНИЕ
РегистрРасчета.Премии.ПерерасчетПремий КАК ПерерасчетПремий
    ПО Премии.Регистратор =
ПерерасчетПремий.ОбъектПерерасчета
        И Премии.ВидРасчета =
ПерерасчетПремий.ВидРасчета
        И Премии.Сотрудник =
ПерерасчетПремий.Сотрудник
        И Премии.Должность =
ПерерасчетПремий.Должность
ГДЕ
    Премии.Регистратор = &Регистратор
    И Премии.ВидРасчета.КатегорияРасчета =
&КатегорияРасчета
;

////////////////////////////////////
////////

ВЫБРАТЬ
    ПодлежатПерерасчету.НомерСтроки КАК НомерСтроки,
    ПодлежатПерерасчету.СпособРасчета,
    ЕСТЬNULL (ПремииБазаОсновныеНачисления.РезультатБаза,
0) + ЕСТЬNULL (ПремииБазаПремии.РезультатБаза, 0) КАК СуммаБазы
ИЗ
    ПодлежатПерерасчету КАК ПодлежатПерерасчету
    ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.Премии.БазаОсновныеНачисления (
        &Измерения,
        &Измерения,
        ,
        Регистратор = &Регистратор

```

```

И ВидРасчета.КатегорияРасчета =
&КатегорияРасчета) КАК ПремииБазаОсновныеНачисления
    ПО ПодлежатПерерасчету.НомерСтроки =
ПремииБазаОсновныеНачисления.НомерСтроки
        ЛЕВОЕ СОЕДИНЕНИЕ
РегистрРасчета.Премии.БазаПремии (
            &Измерения,
            &Измерения,
            ,
            Регистратор = &Регистратор
                И ВидРасчета.КатегорияРасчета =
&КатегорияРасчета) КАК ПремииБазаПремии
                    ПО ПодлежатПерерасчету.НомерСтроки =
ПремииБазаПремии.НомерСтроки

УПОРЯДОЧИТЬ ПО
    НомерСтроки
    
```

Зададим уже знакомые значения параметров запроса, получим результат:

НомерСтроки	СпособРасчета	СуммаБазы
3	Процентом	60 000
6	Процентом	10 869,565217
7	Процентом	10 869,565217

Результат нашего запроса имеет ту же самую структуру, какую имеет результат запроса для выполнения расчета документа, выборку которого мы передаем в процедуру "РассчитатьЗаписиРегистраРасчета", и содержит данные точно такого же характера. Следовательно, можно использовать для выполнения перерасчетов ту же самую процедуру.

Практикум № 9

Напишите запрос, который будет получать необходимые данные для перерасчета регистра "ОсновныеНачисления".

13.9.3. План дальнейших действий

Итак, мы, как и условились в начале этого раздела, только наметили принципиальные пути решения нетривиальных задач, которые возникают при программировании перерасчетов. Поскольку мы получили запрос, возвращающий состав записей регистра, подлежащих перерасчету, а содержание результата этого запроса по структуре аналогично содержанию результата запроса для выполнения основного расчетного алгоритма, то становится ясно, что в принципе при том подходе к построению расчетного алгоритма, который мы так удачно выбрали, алгоритмы расчета и перерасчета отличаются только текстом запросов.

Наметим план доведения до конца задачи программирования перерасчетов.

Мы должны принять во внимание, что перерасчету могут подлежать записи регистра расчета прошедшего (закрытого) периода. Поэтому при практической реализации перерасчетов изменения в результаты существующих записей регистра расчета не вносятся. Вместо этого подлежащие перерасчету суммы сторнируются и создаются новые записи регистра расчета, в которых рассчитываются новые результаты.

Поэтому разумным представляется разработать в Конфигураторе специальный документ, который будет выполнять перерасчеты. Документ будет получать из таблиц перерасчетов записи, подлежащие перерасчету, с суммами результатов, формировать в наборе записей сторнирующие записи и создавать новые записи, в которые будут помещаться результаты перерасчетов.

Для заметок
