

## Практическая работа № 16

### *План видов расчета, регистр расчета (1:00)*

В этой работе мы познакомимся с объектами конфигурации *План видов расчета* и *Регистр расчета* и узнаем об основных понятиях, используемых при создании сложных периодических расчетов. В конце работы мы создадим план видов расчета и регистр расчета, на основе которых в следующей работе продемонстрируем работу механизмов периодических расчетов.

Такие расчеты используются, прежде всего, при расчете зарплаты. Поэтому дальнейшее их рассмотрение мы будем строить на примере расчета зарплаты сотрудников нашей фирмы.

Сумма зарплаты складывается из множества частей (оплата по окладу, премии, штрафы, больничные, разовые выплаты и т.д.), которые могут зависеть друг от друга. Назовем такие части *вид расчета*.

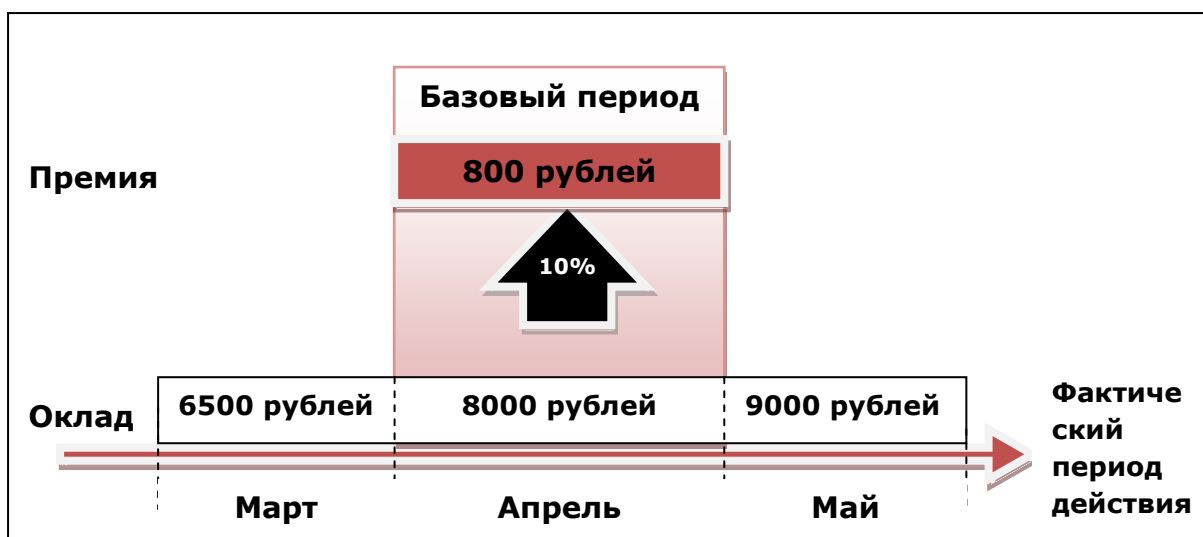
Алгоритм каждого расчета опирается на две категории параметров: период, за который нужно получить конечные данные, и набор некоторых исходных данных, используемых при расчете.

В реальной жизни различные виды расчета оказывают некоторое влияние на другие виды расчета.

- Это влияние может быть на исходные данные, используемые при расчете.

В качестве примера можно привести начисление премии в виде процента от оплаты по окладу. При изменении оклада размер премии тоже должен быть пересчитан. Т.е. сумма начисленного оклада является базой для расчета премии.

Причем оклад рассчитывается для некоторого периода, при расчете премии нам интересно знать не значение оклада вообще, а сумму за определенный период, влияющий на расчет премии. Такой период мы будем называть *базовым*, а подобную зависимость между видами расчета – *зависимостью по базовому периоду*.

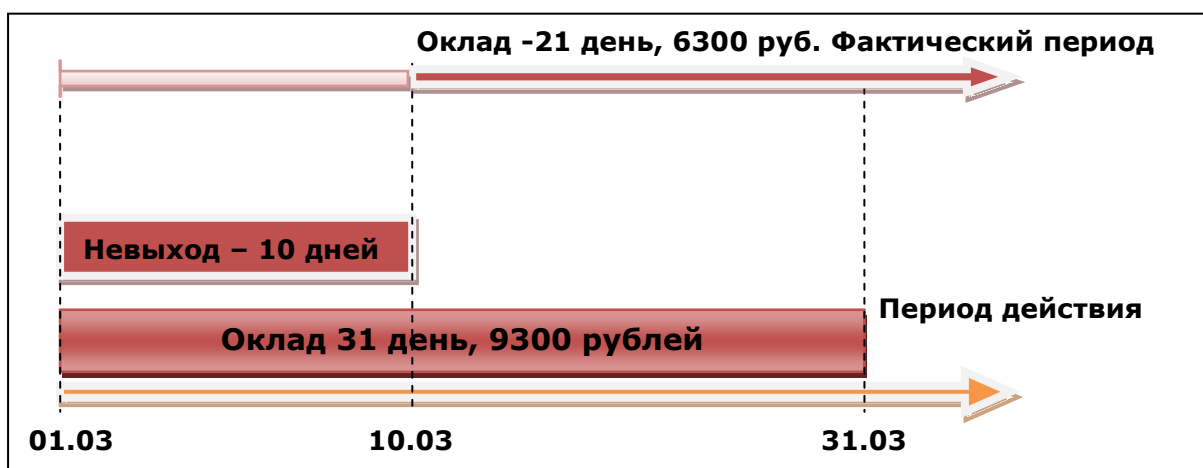


- Это влияние может быть не на исходные данные, а на сам период, за который производится расчет.

В качестве примера можно привести расчет оплаты по окладу и невыход на работу. Предположим, мы начислили сотруднику оплату по окладу за март. В этом случае период действия расчета будет с 01.03.2010 по 31.03.2010. После этого мы получили информацию от руководителя отдела, что сотрудник отсутствовал с 1 по 10 марта по неизвестной причине. В этом случае нам нужно будет произвести расчет **Невыход**, в котором можно рассчитать какие-то удержания с сотрудника. Но кроме этого необходимо будет пересчитать и оклад сотрудника исходя из фактического периода действия – с 11.03 по 31.03.

Такое влияние будем называть *вытеснение по периоду действия*.

В результате если за полный месяц работы сотруднику должно быть начислено 9300 рублей, то теперь за фактический период работы, начисление составит 6300 рублей.



Таким образом, с каждым видом расчета будет связано три периода:

- Период действия является «запрашиваемым». Т.е. указывая период действия, мы говорим: «Мы хотели бы, чтобы результат действовал в этом периоде».
- Фактический период – это то, что получилось из периода действия после анализа всех периодов действия расчетов, которые вытесняют наш по периоду действия.
- Базовый период – в котором мы анализируем результаты других расчетов, влияющих на наш по базовому периоду (календарный месяц).

### План видов расчета

Объект *План видов расчета* предназначен для описания структуры хранения информации о возможных видах расчетов. На основе *Плана видов расчета* платформа создает в базе данных таблицу, в которой будет храниться информация о том, какие существуют виды расчета и какие взаимосвязи между ними.

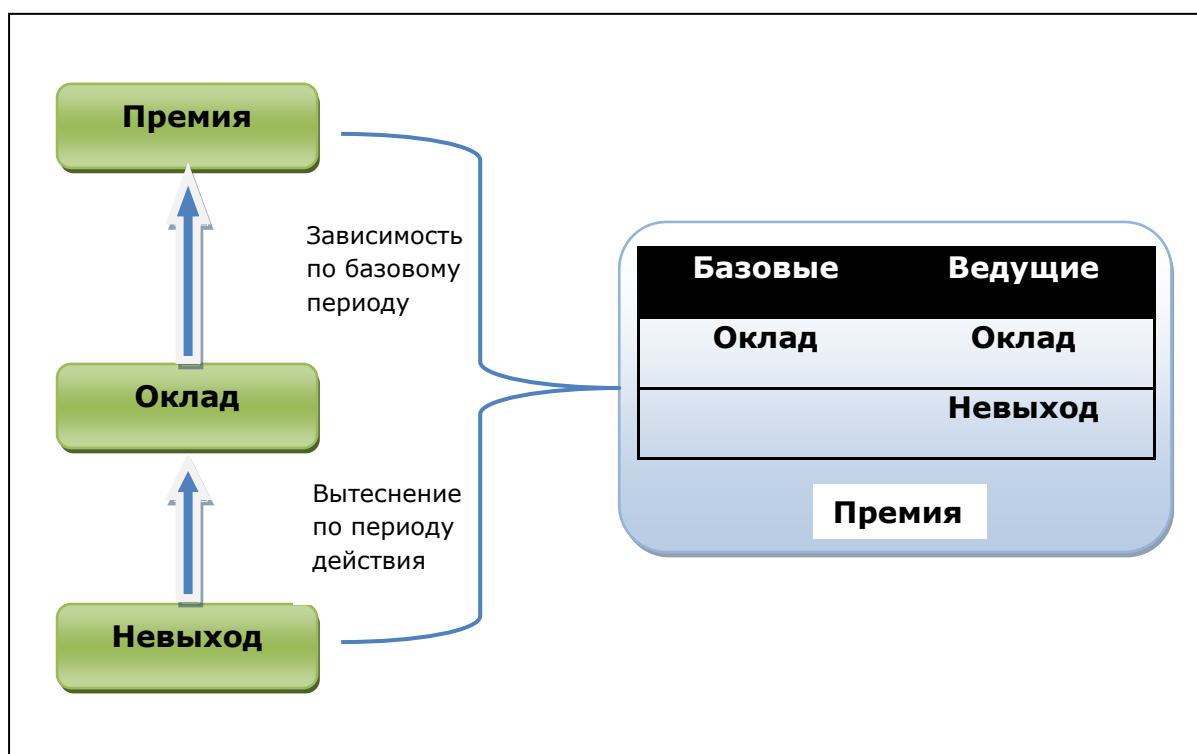
Отличительной особенностью плана видов расчета является то, что пользователь в процессе работы может добавлять новые виды расчета.

План видов расчета имеет свойства:

- **Использует период действия** – определяет, будут ли в этом плане находиться виды расчета, которые могут быть вытеснены по периоду действия.
- **Зависимость от базы** – определяет, будут ли в этом плане находиться зависимые по базовому периоду виды расчета.

Еще одной важной особенностью плана видов расчета является возможность создания предопределенных видов расчета и описания их взаимного влияния. При этом разработчик имеет возможность указать три категории видов расчета, влияющих на предопределенный вид расчета:

- *Базовые* – их результаты должны быть использованы при перерасчете этого вида расчета.
- *Вытесняющие* – вытесняют этот вид расчета по периоду действия.
- *Ведущие* – изменение их результатов должно приводить к необходимости перерасчета этого вида расчета.



## Добавление плана видов расчета

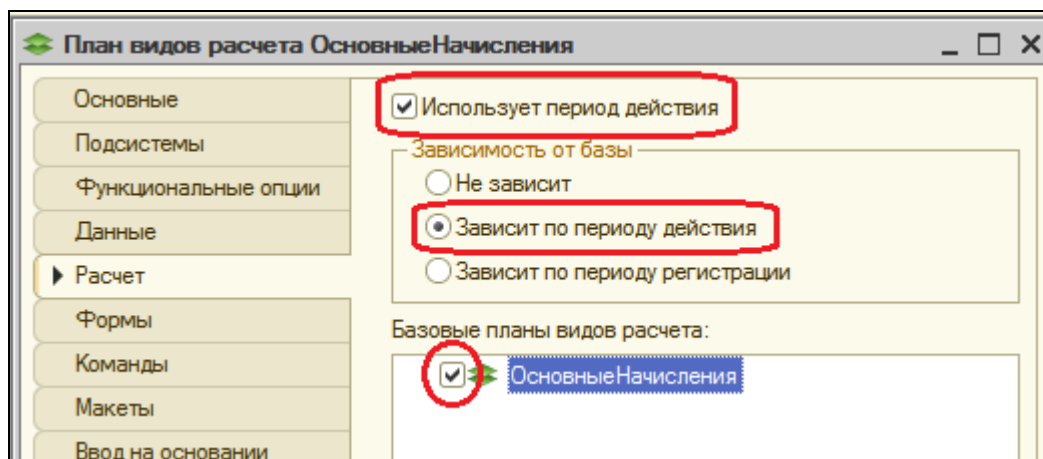
### В режиме Конфигуратор

Создайте новый объект конфигурации План видов расчета с именем **ОсновныеНачисления**. Представление списка задайте как **Виды расчетов**.

На закладке **Подсистемы** укажите, что план видов расчета будет отображаться в подсистеме **РасчетЗарплаты**.

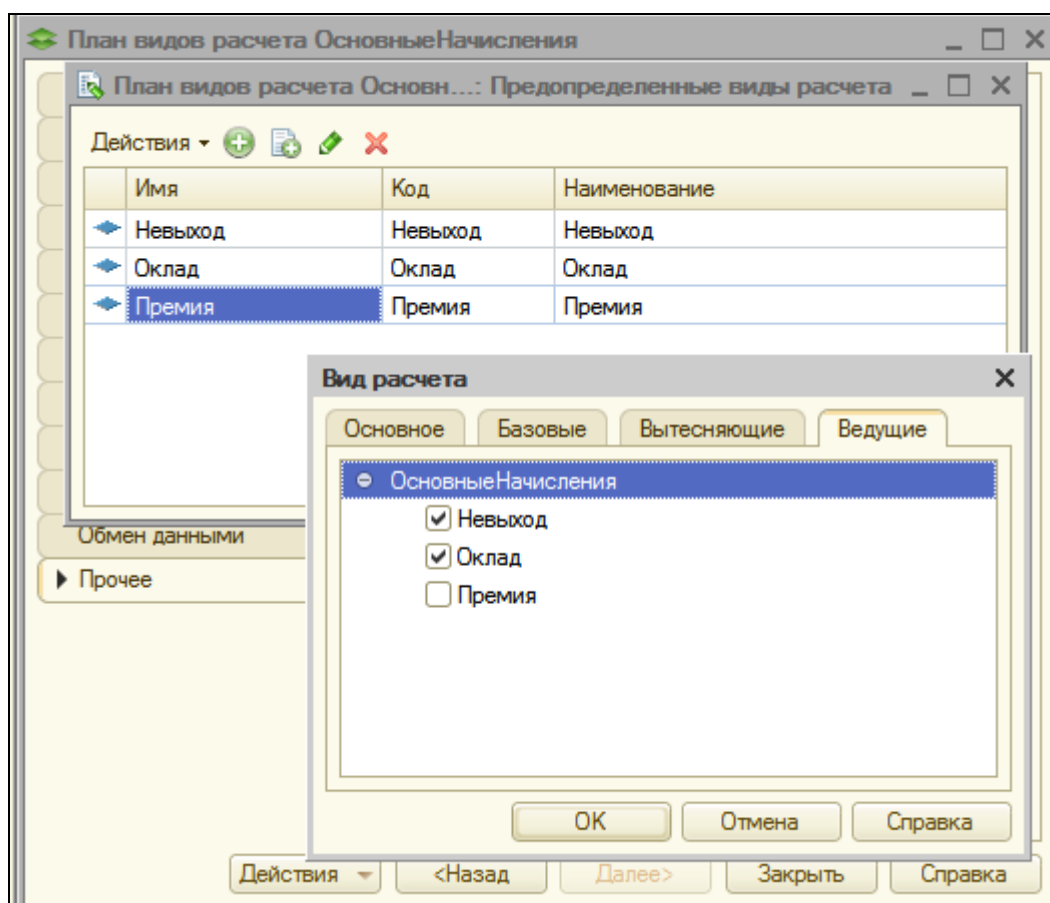
На закладке **Расчет** укажите, что он будет использовать период действия и зависеть от базы по периоду действия.

В качестве базового плана видов расчета укажем его самого, поскольку все наши виды расчетов будут храниться в единственном плане видов расчета.



На закладке Прочее зададим предопределенные виды расчета:

- **Невыход** – с именем, кодом и наименованием **Невыход**;
- **Оклад** – с именем, кодом и наименованием **Оклад** и вытесняющим его видом расчета **Невыход**;
- **Премия** – с именем, кодом и наименованием **Премия**, с базовым видом расчета **Оклад** и ведущими видами расчета **Невыход** и **Оклад**.



## Что такое Регистр расчета

Объект конфигурации *Регистр расчета* предназначен для описания структуры накопления данных, являющихся результатами расчетов. На основе регистра расчета платформа создает в базе данных таблицы, в которых будут накапливаться данные, формируемые различными объектами базы данных.

Отличительные особенности регистра расчета:

- **Он не предназначен для интерактивного редактирования пользователем.** Разработчик может при необходимости предоставить пользователю возможность редактирования, но изначально регистр расчета для этого не предназначен.
- **Периодичность.** Определяет промежуток времени, к которому будет относиться каждая запись регистра.
- **Возможность использования механизмов вытеснения по периоду действия.** При этом для каждой записи регистр формирует фактический период действия, который является совокупностью нескольких периодов внутри периода действия.
- **Возможность использования механизмов вытеснения по зависимости по базовому периоду.** Этот механизм позволяет основывать расчет зависимых (вторичных) записей регистра на данных, полученных в результате расчета первичных записей.
  - **Зависимость по периоду действия.** При анализе базовых записей будут выбираться те, в которых найдено пересечение фактического периода действия и указанного базового периода.
  - **Зависимость по периоду регистрации.** При анализе базовых записей будут выбираться те, которые попадают в указанный базовый период значением своего поля Период регистрации.
- **Связь с планом видов расчета.** На основе этой связи работают механизмы вытеснения по периоду действия и зависимости по базовому периоду, поскольку в плане видов расчета описано взаимное влияние видов расчета друг на друга.

У регистра расчета могут существовать подчиненные объекты **Перерасчет**. Они предназначены для регистрации фактов появления в регистре записей, влияющих на результат расчета уже существующих записей регистра. **Перерасчет** может иметь несколько измерений, каждое из которых устанавливает связь между измерениями данного регистра расчета и влияющих регистров расчета. В частном случае это может быть один и тот же регистр.

Таблицы перерасчета заполняются автоматически на основании записей регистров расчета, затронутых ведущими видами расчета и на основании записей регистра расчета, для которых изменился фактический период действия.

Регистр расчета может быть связан с графиком времени. Такой график времени должен представлять собой регистр сведений (непериодический, с обязательным измерением типа Дата и ресурсом типа Число), в котором содержится временная схема исходных данных, участвующих в расчетах. Измерениями этого графика могут быть, например, график работы (ссылка на справочник) и дата, а ресурсом – количество рабочих часов в этой дате. В этом случае можно будет связать запись регистра расчета с каким-либо конкретным графиком работы (указав в качестве реквизита записи ссылку на справочник ВидыГрафиковРаботы) и в дальнейшем средствами встроенного языка получать информацию о количестве рабочих часов в периоде действия, фактическом периоде действия или периоде регистрации этой записи.

### **Добавление регистра расчета**

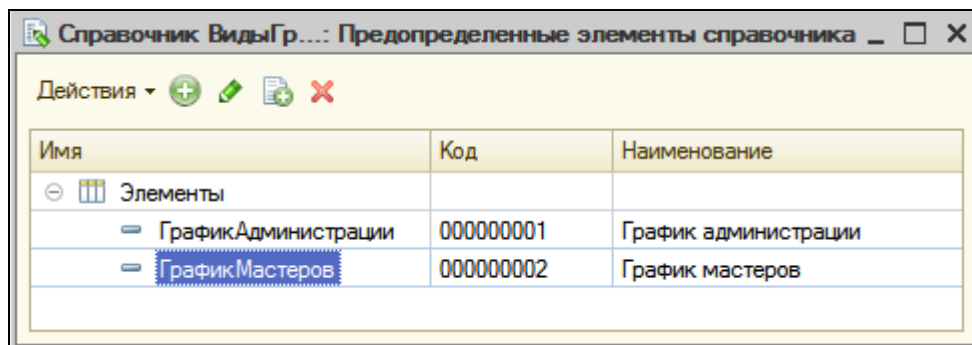
Прежде чем мы начнем создавать объект конфигурации Регистр расчета **Начисления**, нам потребуется создать два дополнительных объекта конфигурации:

- Регистр сведений **ГрафикиРаботы**,
- Справочник **ВидыГрафиковРаботы**.

Справочник понадобится нам для хранения информации о том, какие графики работы существуют в нашей фирме, а регистр сведений – для указания того, какие дни в месяце являются рабочими, поскольку сумма оплаты по окладу будет рассчитываться исходя из отработанных дней в месяце сотрудником.

## В режиме Конфигуратор

Создайте новый объект Справочник с именем **ВидыГрафиковРаботы**. На закладке **Подсистемы** укажите, что справочник будет отображаться в подсистеме **РасчетЗарплаты**. На закладке **Прочее** создайте для справочника два predetermined графика работы – **ГрафикАдминистрации** и **ГрафикМастеров**.



После этого создадим объект конфигурации Регистр сведений с именем **ГрафикиРаботы**. Этот регистр будет иметь два измерения:

- **ГрафикРаботы**, тип **СправочникСсылка.ВидыГрафиковРаботы**;
- **Дата**, тип **Дата**.


Затем создадим единственный ресурс регистра – **Значение**, тип **Число**, длина **1**.

На закладке **Подсистемы** укажем **РасчетЗарплаты**.

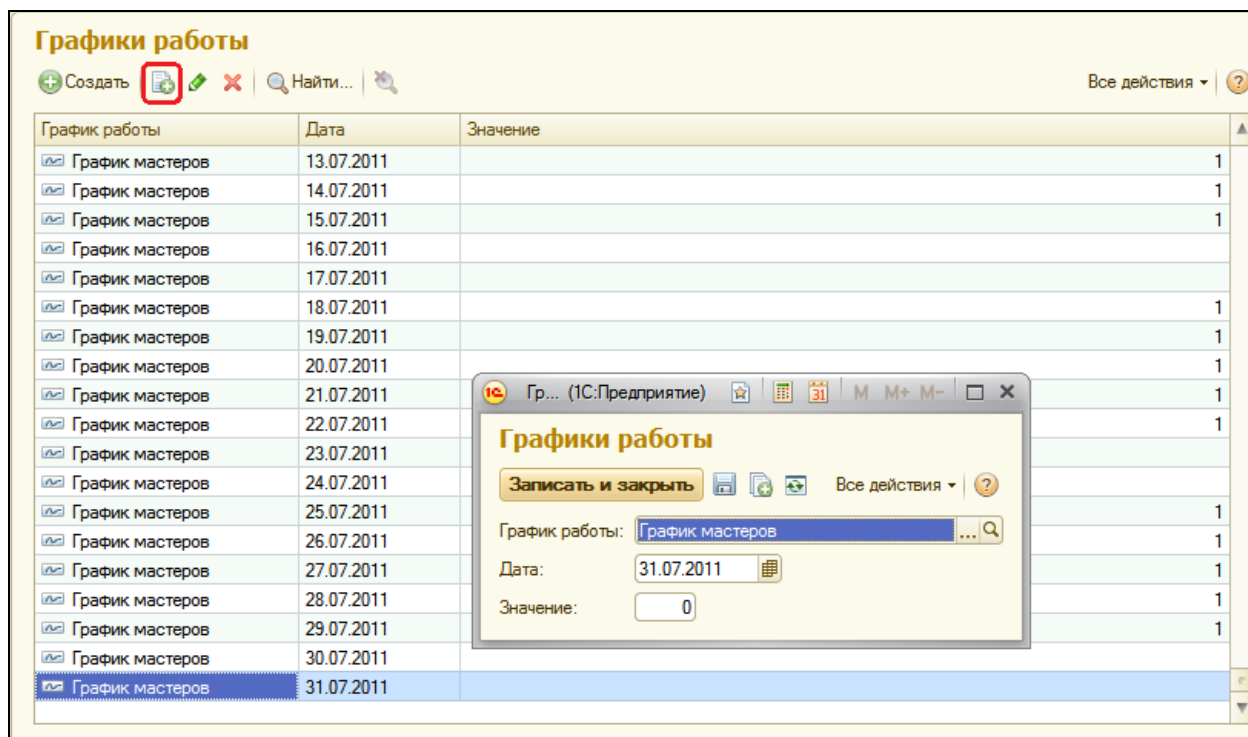
Теперь заполним регистр сведений **ГрафикиРаботы** данными о рабочих днях июля графика мастеров.

(Если произошла ошибка Frame.dll, настройте родительское окно в режим Соединяемое, Свободное.)

## В режиме 1С:Предприятие

Запустите режим отладки и в разделе **Расчет зарплаты** выполните команду **Графики работы**. Поочередно создайте 31 запись в регистре для июля. Для облегчения работы воспользуйтесь кнопкой  **Создать новый элемент копированием текущего**. В качестве измерения **ГрафикРаботы** нашего регистра выберем predetermined элемент **ГрафикМастеров** справочника **ВидыГрафиковРаботы**. В качестве ресурса **Значение** у рабочих дней поставим 1, выходных – 0.





Теперь все готово для создания регистра расчета.

## В режиме Конфигуратор

Добавим новый объект конфигурации *Регистр расчета* с именем **Начисления**. Зададим **Расширенное представление списка** как **Движения в регистре Начисления**.

В качестве плана видов расчета, используемого регистром, выберем **Основные Начисления**.

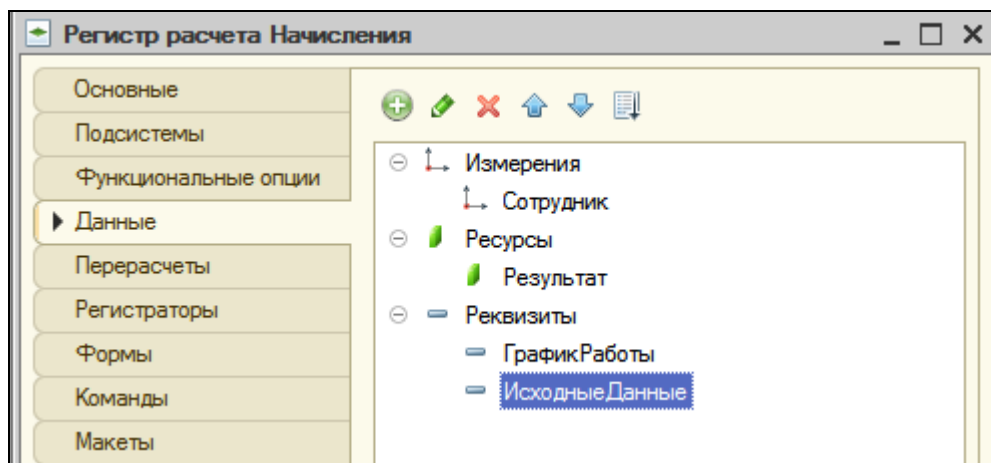
Установим, что регистр будет использовать период действия, график будет задаваться в регистре сведений **Графики Работы**, значение графика будет находиться в ресурсе **Значение**, а дата графика – в измерении **Дата**. Укажем, что регистр будет использовать базовый период и периодичностью регистра будет **Месяц**.

На закладке **Подсистемы** укажем **Расчет Зарплаты**.

На закладке **Данные** создадим:

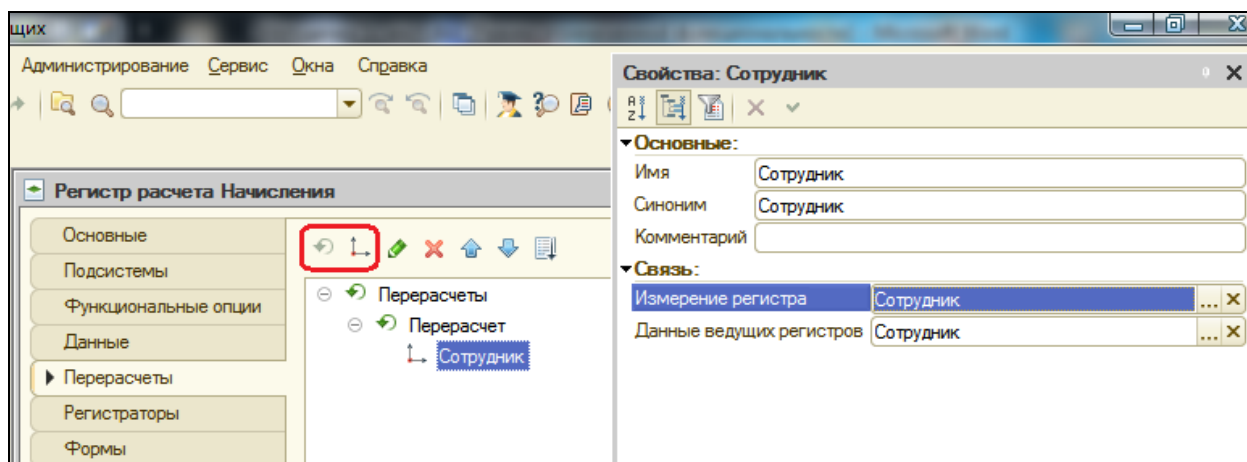
- Измерение **Сотрудник**, тип **СправочникСсылка.Сотрудники**, **Базовое**;
- Ресурс **Результат**, тип **Число**, длина **15**, точность **2**;
- Реквизит **ГрафикРаботы**, тип **СправочникСсылка.ВидыГрафиковРаботы**, в разделе свойств **Данные** зададим свойство **Связь с графиком** по измерению **ГрафикРаботы**;
- Реквизит **ИсходныеДанные**, тип **Число**, длина **15**, точность **2**.

Реквизит **ГрафикРаботы** будем использовать, чтобы связать запись регистра с используемым графиком работы, а реквизит **ИсходныеДанные** – чтобы хранить в нем данные для расчета или перерасчета.



Перейдем на закладку **Перерасчеты**. Создадим объект конфигурации **Перерасчет** с таким же именем. У него будет измерение **Сотрудник**, для которого в разделе **Связь** укажем:

- Измерение регистра – **Сотрудник**;
- Данные ведущих регистров – выберем то же самое измерение **Сотрудник** регистра расчета **Начисления**.



В заключение отредактируем командный интерфейс, чтобы в подсистеме **РасчетЗарплаты** была доступна команда для просмотра записей регистра расчета.

Для этого в дереве объектов конфигурации выберите подсистему **РасчетЗарплаты**, вызовите контекстное меню и выберите **Открыть командный интерфейс**. В группе **Панель навигации.Обычное** включите видимость команды **Начисления**.

На этом создание объекта **Регистр расчета Начисления** завершено.

## Контрольные вопросы

- ✓ Как использовать план видов характеристик для организации.
- ✓ Что такое вид расчета, база.
- ✓ Какая разница между базовым периодом, периодом действия и фактическим периодом.
- ✓ Что такое зависимость по базовому периоду.
- ✓ Что такое вытеснение по периоду действия.
- ✓ Для чего предназначен объект конфигурации План видов расчета.
- ✓ Каковы основные свойства плана видов расчета.
- ✓ Какая разница между базовыми, вытесняющими и ведущими видами расчетов.
- ✓ Как создать план видов расчета.
- ✓ Что такое объект конфигурации Регистр расчета.
- ✓ Каковы отличительные особенности регистра расчета.
- ✓ Что такое перерасчет.
- ✓ Как создать регистр расчета.

## Практическая работа № 17

### *Использование регистра расчета*

В этой работе мы создадим документ, с помощью которого будут выполняться различные виды начислений, посмотрим, как и когда платформа формирует записи перерасчета, увидим, как работают механизмы вытеснения по периоду действия и зависимости по базовому периоду.

Кроме этого, мы создадим отчет, показывающий начисления сотрудникам нашей фирмы, и сделаем так, чтобы данные расчетов можно было поддерживать в актуальном состоянии. Всё это нам нужно для системы расчета зарплаты.

В заключение мы познакомимся с новым элементом формы – *Диаграмма Ганта* и с его помощью наглядно покажем работу некоторых механизмов расчета.

#### **Добавление документа о начислениях**

Откройте конфигуратор и добавьте новый объект Документ с именем **НачисленияСотрудникам** и представлением **Начисление сотрудникам**. На закладке **Нумерация** установим:

- **Тип номера** – **Число**;
- **Длина номера** – **5**.

На закладке **Подсистемы** укажем, что документ будет отображаться в подсистеме **РасчетЗарплаты**.

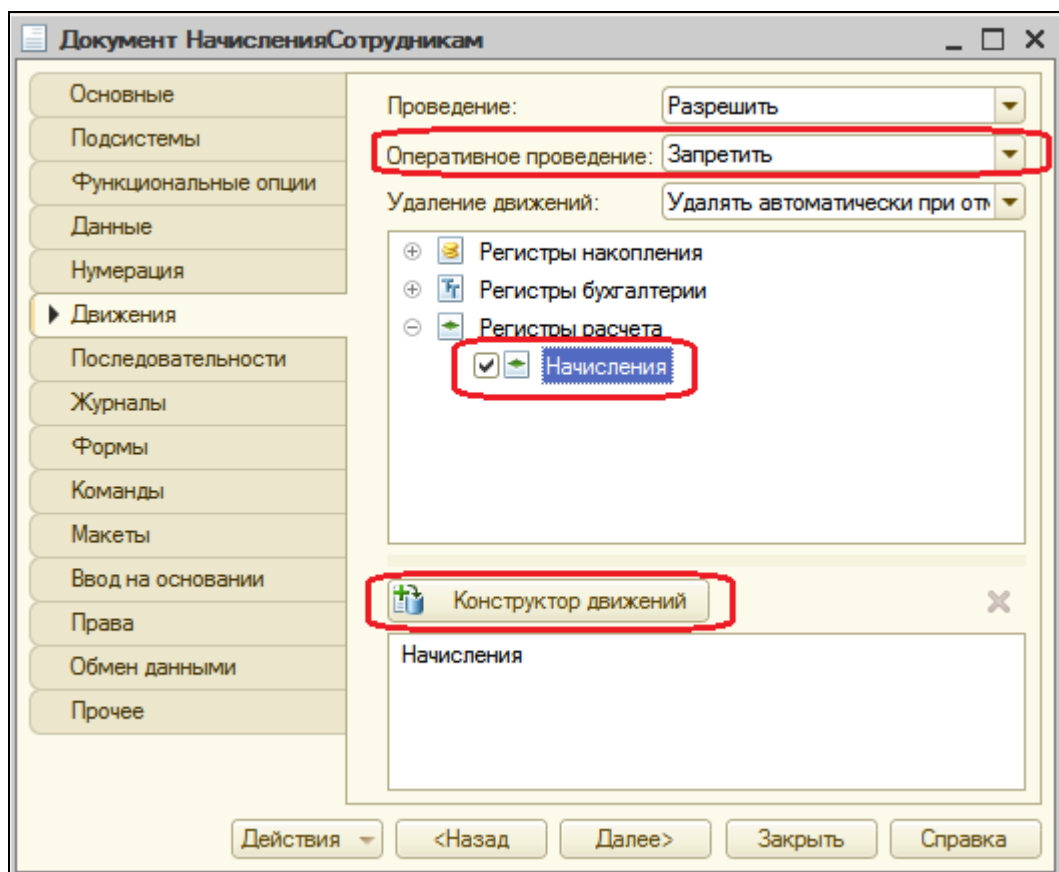
На закладке **Данные** укажем, что этот документ будет иметь табличную часть **Начисления**, содержащую следующие реквизиты:

- **Сотрудник**, тип **СправочникСсылка.Сотрудник**;
- **ГрафикРаботы**, тип **СправочникСсылка.ВидыГрафиковРаботы**;
- **ДатаНачала**, тип **Дата**;
- **ДатаОкончания**, тип **Дата**;
- **ВидРасчета**, тип **ПланВидовРасчетаСсылка.ОсновныеНачисления**;
- **Начислено**, **Число**, **длина 15**, **точность 2**.

Реквизиты **ДатаНачала** и **ДатаОкончания** понадобятся нам для задания периода, в котором должна действовать запись расчета.

На закладке **Движения** запретим оперативное проведение документа.

Отметим, что документ будет создавать движения по регистру расчета **Начисления**, и запустим конструктор движений.



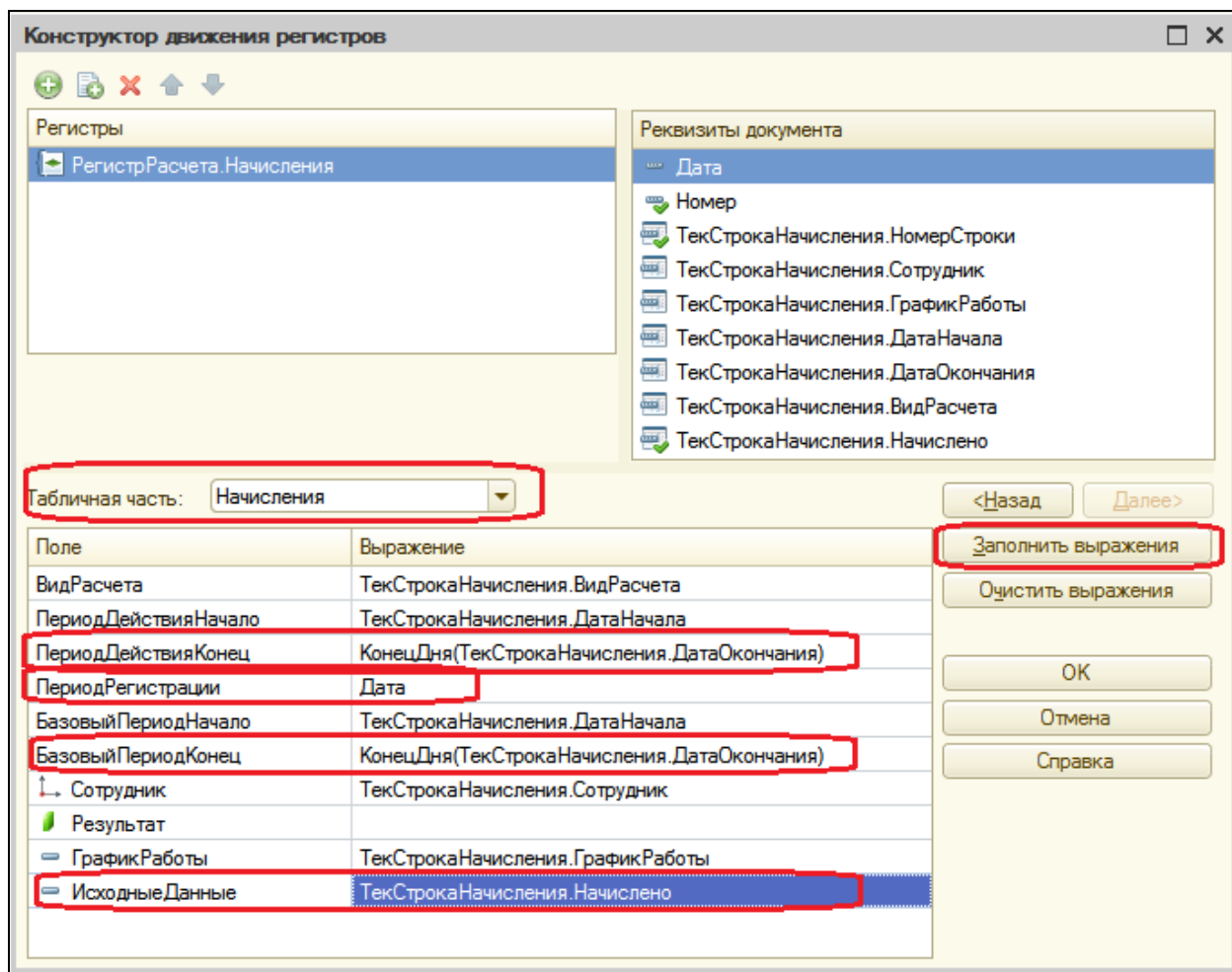
В окне конструктора выберем табличную часть **Начисления** и нажмем **Заполнить выражения**.

Для реквизитов **ПериодДействияКонец** и **БазовыйПериодКонец** укажем выражение **КонецДня(ТекСтрокаНачисления.ДатаОкончания)**.

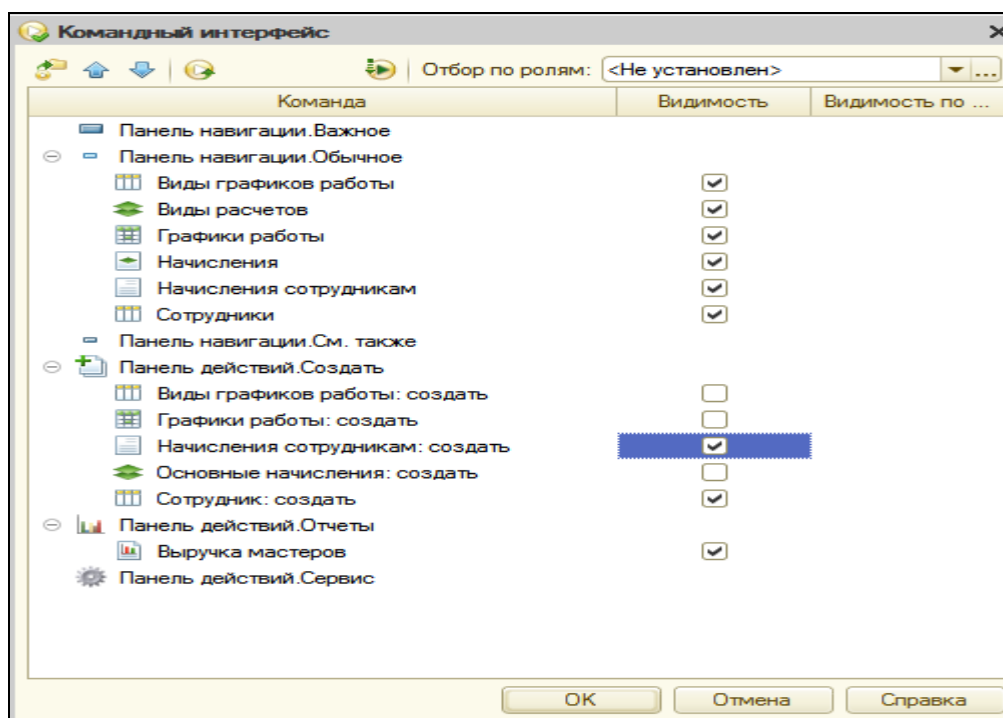
Для поля **ПериодРегистрации** укажем выражение **Дата**.

Реквизиту **ИсходныеДанные** поставим в соответствие реквизит табличной части **Начислено - ТекСтрокаНачисления.Начислено**, а для ресурса **Результат** оставим пустое выражение, т.к. мы будем его потом рассчитывать.

Нажмите **ОК** и посмотрим текст обработчика, созданный конструктором.



В заключение отредактируем командный интерфейс, чтобы в подсистеме **РасчетЗарплаты** была доступна команда создания новых документов.



## В режиме 1С:Предприятие

Запустим режим отладки и посмотрим, как работает наш документ.

В панели действий раздела **Расчет зарплаты** выполним команду **Начисление сотрудникам** и начислим оклад за июль всем сотрудникам фирмы.

Период начисления должен совпадать с периодом, введенным в регистр сведений **Графики работы**.

Начисления сотрудникам (создание) \*

Провести и закрыть Провести

Номер: 0

Дата: 09.05.2011 0:00:00

Добавить

N	Сотрудник	График работы	Дата начала	Дата окончания	Вид расчета	Начислено
1	Гусаков Николай Д...	График мастеров	01.07.2011	31.07.2011	Оклад	10 000,00
2	Деловой Иван Серге...	График мастеров	01.07.2011	31.07.2011	Оклад	8 000,00
3	Симонов Валерий М...	График мастеров	01.07.2011	31.07.2011	Оклад	3 000,00

Проведем документ и посмотрим, какие движения он сформировал в регистре **Начисления**.

Движения в регистре Начисления

Найти...

Период ре...	Регистратор	Н...	Вид расчета	С...	Сотрудник	Р...	График работы	Исходные данные	Период действия
01.05.2011...	Начисления сотру...	1	Оклад		Гусаков Николай Дм...		График мастеров	10 000,00	01.07.2011 0:00:0
01.05.2011...	Начисления сотру...	2	Оклад		Деловой Иван Серге...		График мастеров	8 000,00	01.07.2011 0:00:0
01.05.2011...	Начисления сотру...	3	Оклад		Симонов Валерий Ми...		График мастеров	3 000,00	01.07.2011 0:00:0

Обратите внимание, что платформа привела период регистрации каждой записи к началу периода регистра расчета.

Кроме этого, в каждой записи мы сохранили в реквизите **ИсходныеДанные** размер оклада сотрудника, введенный в документе, чтобы в дальнейшем рассчитать сумму оплаты по окладу.



Для дальнейшего изучения работы регистра расчета нам понадобится служебный отчет, с помощью которого мы сможем посмотреть содержимое записей перерасчета.

### Иллюстрация механизмов вытеснения и зависимости от базы

#### Отчет по перерасчетам

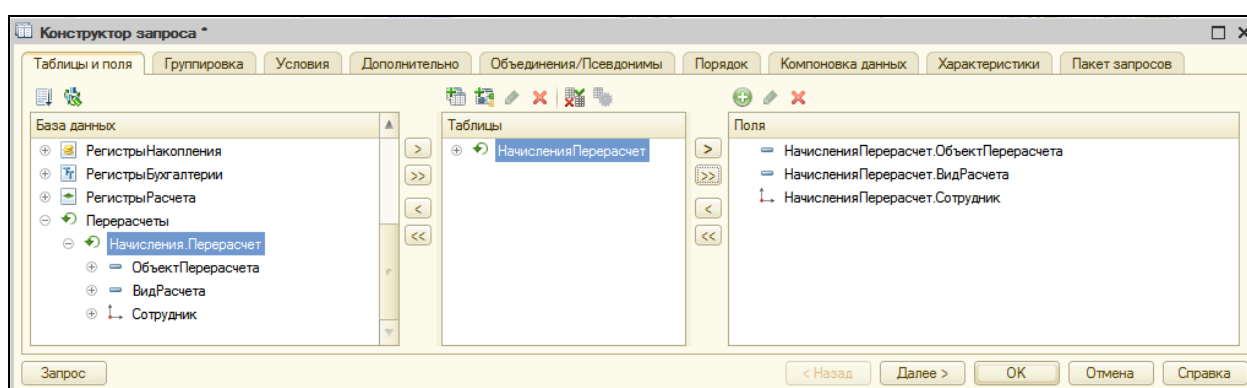
В режиме Конфигуратор

Создадим новый объект **Отчет** с именем **Перерасчет**. Создадим новую схему компоновки данных, добавим источник данных – запрос и откроем конструктор запроса.

В списке **База данных** раскроем ветвь **Перерасчеты** и из виртуальной таблицы перерасчета **Начисления.Перерасчет** выберем все поля:

- **ОбъектПерерасчета,**
- **ВидРасчета,**
- **Сотрудник.**

На этом создание запроса закончено, нажмите ОК.



Перейдем на закладку **Настройки** и добавим группировку детальных записей. На закладке **Выбранные поля** выберем для вывода в отчет поля **ОбъектПерерасчета**, **ВидРасчета** и **Сотрудник**.

На этом создание схемы компоновки закончено, закройте ее.

В окне редактирования отчета **Перерасчет** на закладке **Подсистемы** укажите **РасчетЗарплаты**.

#### Зависимость по базовому периоду

В режиме 1С:Предприятие

Если сейчас мы выполним отчет в режиме 1С:Предприятие, то мы увидим, что ни один перерасчет еще не выполнялся.

Поэтому создадим новый документ **Начисление сотрудникам №2**, в котором начислим премию за июль Гусакову и Деловому.

N	Сотрудник	График работы	Дата начала	Дата окончания	Вид расчета	Начислено
1	Гусаков Николай Д...	График мастеров	01.07.2011	31.07.2011	Премия	
2	Деловой Иван Серг...	График мастеров	01.07.2011	31.07.2011	Премия	

Этим документом мы зафиксируем факт, что сотрудникам Гусакову и Деловому нужно начислить премию по итогам работы за июль. Поскольку размер премии нам неизвестен (он будет рассчитываться), поле Начислено мы оставили пустым. Нажмите **Провести и закрыть**.

Теперь откроем документ **Начисление сотрудникам №1** (Начисления сотрудникам в навигационной панели) и изменим оклад Гусакова с 10000 на 7000. Нажмите **Провести и закрыть**.

Сформируйте отчет **Перерасчет**.

Регистратор	Вид расчета	Сотрудник
Начисления сотрудникам 2 от 09.05.2011 13:40:58	Премия	Гусаков Николай Дмитриевич
Начисления сотрудникам 2 от 09.05.2011 13:40:58	Премия	Деловой Иван Сергеевич

Как видите, отчет содержит какие-то данные. В самом деле, вид расчета **Премия** зависит у нас по базовому периоду от вида расчета **Оклад**. Как только мы изменили существовавшие в регистре записи по виду расчета **Оклад**, платформа сразу же сформировала набор записей перерасчета, которые должны быть рассчитаны заново, т.к. изменилась их база.

В перерасчет попали записи как про Делового, так и про Гусакова, потому что платформа не отслеживает конкретные изменения, а

отслеживает факт изменения набора записей регистра расчета в результате проведения документа. Поэтому она включает информацию обо всех записях регистра, значение ресурсов которых может измениться в результате перепроведения документа, создавшего базовые записи регистра.

Перепроведем документ **Начисления сотрудникам №2** и сформируем отчет **Перерасчет**.

Он снова не содержит никаких данных – система отметила, что мы пересчитали зависимые записи, и очистила таблицу перерасчета.

На этом примере мы познакомились с работой механизма поддержки зависимости по базовому периоду у регистра расчета.

### Вытеснение по периоду действия

В режиме 1С:Предприятие

Теперь посмотрим, как работает механизм вытеснения по периоду действия. Для этого нам понадобится создать документ **Начисления сотрудникам №3**.

Начисления сотрудникам (создание) - Пособие для начинающих \* (1С:Предприятие)

Начислен...

Перейти  
Начисления

**Начисления сотрудникам (создание) \***

Провести и закрыть | Провести

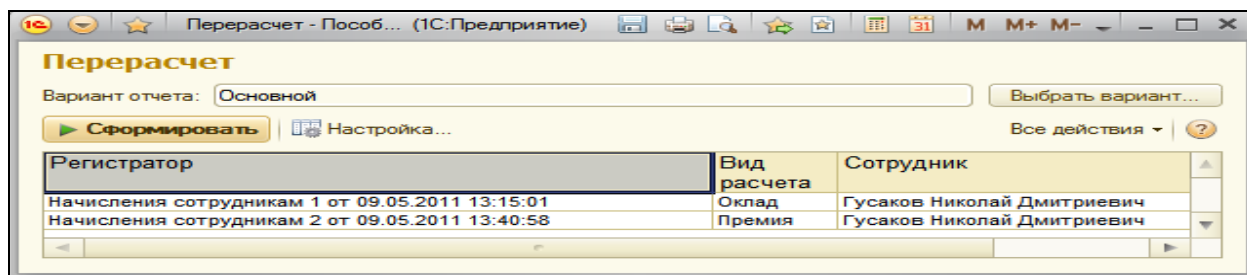
Номер: 0

Дата: 09.05.2011 0:00:00

+ Добавить | - Удалить | ↑ Вверх | ↓ Вниз

N	Сотрудник	График работы	Дата начала	Дата окончания	Вид расчета	Начислено
1	Гусаков Николай Д...	График мастеров	01.07.2011	10.07.2011	Невыход	

Этим документом мы зафиксируем факт невыхода на работу Гусакова с 1 по 10 июля. В этом случае потребуется пересчитать его оплату по окладу и как следствие начисленную премию. Нажмите **Провести и закрыть** и сформируйте отчет **Перерасчет**.



Как видите, в перерасчет попала запись о начислении оклада Гусакову. Это явилось результатом работы механизма вытеснения по периоду действия, ведь вид расчета **Невыход** вытесняет у нас вид расчета **Оклад**.

Обратите внимание, что в перерасчет попала и запись о начислении премии Гусакову. При создании predetermined видов расчета мы указали, что результат вида расчета **Премия** будет зависеть от изменения результата вида расчета **Невыход**. Платформа отследила эту зависимость.

Перепроведем документы **Начисления сотрудникам №1** и **№2** и убедимся, что таблица перерасчета очистилась, сформировав отчет.

### Процедура расчета записей регистра расчета

#### В режиме Конфигуратор

До сих пор мы просто заносили в регистр расчета **Начисления** записи о том, что необходимо выполнить какой-либо вид расчета, но каким именно образом получать эти результаты, мы не говорили.

Теперь настало время описать алгоритмы формирования различных видов расчета. Поскольку эти алгоритмы нам нужно будет использовать не только в документе **Начисления сотрудникам**, удобнее всего будет разместить их в отдельном общем модуле.

Откроем в конфигураторе текст обработчика проведения документа **НачислениеСотрудникам** и добавим в него после завершения создания движений в регистре **Начисления** вызов процедуры **РассчитатьНачисления()** из общего модуля **ПроведениеРасчетов**.

```
Процедура ОбработкаПроведения(Отказ, Режим)
```

```
...
```

```
КонецЦикла;
```

```
    //}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

```
    //Записываем движения регистров.  
    Движения.Начисления.Записать();
```

```

//Получим список всех сотрудников, содержащихся в документе.
Запрос = Новый Запрос(
    "ВЫБРАТЬ РАЗЛИЧНЫЕ
    | НачисленияСотрудникамНачисления.Сотрудник
    | ИЗ
    | Документ.НачисленияСотрудникам.Начисления
    | КАК НачисленияСотрудникамНачисления
    | ГДЕ
    | НачисленияСотрудникамНачисления.Ссылка =
&ТекущийДокумент");

Запрос.УстановитьПараметр("ТекущийДокумент", Ссылка);

//Сформируем список сотрудников.
ТаблЗнач = Запрос.Выполнить().Выгрузить();
МассивСотрудников = ТаблЗнач.ВыгрузитьКолонку("Сотрудник");

//Вызов процедуры РассчитатьНачисления из общего модуля.
ПроведениеРасчетов.РассчитатьНачисления(Движения.Начисления,
    ПланыВидовРасчета.ОсновныеНачисления.Оклад,
МассивСотрудников);
    Движения.Начисления.Записать(, Истина);
ПроведениеРасчетов.РассчитатьНачисления(Движения.Начисления,
    ПланыВидовРасчета.ОсновныеНачисления.Премия,
МассивСотрудников);
    Движения.Начисления.Записать(, Истина);
КонецПроцедуры

```

Обратите внимание: при проведении документа мы сначала записываем движения, сформированные документом, в регистр (`Движения.Начисления.Записать()`), а затем передаем этот набор записей регистра в процедуру расчета **РассчитатьНачисления()**, которую мы создадим в общем модуле **ПроведениеРасчетов**.

Эту процедуру мы вызываем сначала для расчета первичных записей (**Оклад**), а затем для расчета вторичных (**Премия**).

После того, как ресурсы будут рассчитаны, мы перезаписываем набор записей регистра без формирования записей перерасчета (второй параметр в методе **Записать()** – **Истина**).

Перед вызовом процедуры из общего модуля мы с помощью запроса формируем массив сотрудников, перечисленных в документе, чтобы передать его в вызываемую процедуру.

Для параметра запроса **ТекущийДокумент** устанавливаем значение стандартного реквизита документа – **Ссылка**. Используя метод запроса **Запрос.Выполнить().Выгрузить()**, выгружаем результат запроса в таблицу значений (переменную **ТаблЗнач**). Затем формируем массив

**МассивСотрудников**, содержащий колонку **Сотрудник** из этой таблицы значений.

Теперь создадим в ветке **Общие** новый общий модуль **ПроведениеРасчетов**.

Установим флажок **Вызов сервера** для видимости его экспортных процедур и функций.

Добавим в него следующий текст процедуры **РассчитатьНачисления**:

Процедура РассчитатьНачисления(НаборЗаписейРегистра, ТребуемыйВидРасчета, СписокСотрудников) Экспорт

```
Регистратор=НаборЗаписейРегистра.Отбор.Регистратор.Значение;
```

```
// Рассчитать первичные записи
```

```
Если ТребуемыйВидРасчета = ПланыВидовРасчета.ОсновныеНачисления.Оклад
```

```
Тогда
```

```
Запрос = Новый Запрос;
```

```
Запрос.Текст =
```

```
"ВЫБРАТЬ
```

```
| НачисленияДанныеГрафика.ЗначениеПериодДействия КАК Норма,
```

```
| НачисленияДанныеГрафика.ЗначениеФактическийПериодДействия КАК Факт,
```

```
| НачисленияДанныеГрафика.НомерСтроки КАК НомерСтроки
```

```
| ИЗ
```

```
| РегистрРасчета.Начисления.ДанныеГрафика(Регистратор = &Регистратор И
```

```
| ВидРасчета = &ВидРасчета И Сотрудник В (&СписокСотрудников))
```

```
| КАК НачисленияДанныеГрафика";
```

```
Запрос.УстановитьПараметр("Регистратор", Регистратор);
```

```
Запрос.УстановитьПараметр("ВидРасчета", ТребуемыйВидРасчета);
```

```
Запрос.УстановитьПараметр("СписокСотрудников", СписокСотрудников);
```

```

ВыборкаРезультата = Запрос.Выполнить().Выбрать();
    // Рассчитать вторичные записи
    ИначеЕсли ТребуемыйВидРасчета =
ПланыВидовРасчета.ОсновныеНачисления.Премия Тогда
        КонечЕсли;

КонечПроцедуры

```

В этом запросе мы выбираем из виртуальной таблицы данных графика регистра расчета значение графика для периода действия и для фактического периода действия. При задании параметров виртуальной таблицы мы ограничиваем выборку регистратором, нужным нам видом расчета и списком сотрудников, по которым нужно получить значения графика.

Теперь добавим обход переданного в процедуру набора записей и расчет записей, для которых получены значения графика.

```

Запрос.УстановитьПараметр("Регистратор", Регистратор);
Запрос.УстановитьПараметр("ВидРасчета", ТребуемыйВидРасчета);
Запрос.УстановитьПараметр("СписокСотрудников", СписокСотрудников);

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

Для Каждого ЗаписьРегистра Из НаборЗаписейРегистра Цикл
    СтруктураНомер = Новый Структура("НомерСтроки");
    СтруктураНомер.НомерСтроки = ЗаписьРегистра.НомерСтроки;
    ВыборкаРезультата.Сбросить();
    Если ВыборкаРезультата.НайтиСледующий(СтруктураНомер) Тогда
        Если ВыборкаРезультата.Норма = 0 Тогда
            Сообщение = Новый СообщениеПользователю;
            Сообщение.Текст = "Вид расчета: Оклад – Нет рабочих
дней в заданном периоде";
            Сообщение.Сообщить();
            ЗаписьРегистра.Результат = 0;
        Иначе

            // Рассчитать оклад по фактическому периоду и
исходным данным
            ЗаписьРегистра.Результат =
(ЗаписьРегистра.ИсходныеДанные /ВыборкаРезультата.Норма) *
ВыборкаРезультата.Факт;
            Сообщение = Новый СообщениеПользователю;
            Сообщение.Текст = "Выполнен расчет" +
ЗаписьРегистра.Регистратор + " – " + ЗаписьРегистра.ВидРасчета + " – "
            + ЗаписьРегистра.Сотрудник;
            Сообщение.Сообщить();
        КонечЕсли;
    КонечЕсли;
КонечЦикла;

```

```

// Рассчитать вторичные записи
ИначеЕсли ТребуемыйВидРасчета =
ПланыВидовРасчета.ОсновныеНачисления.Премия Тогда
    КонечЕсли;

КонечПроцедуры

```

Для каждой записи из набора записей регистра расчета мы получаем номер строки, идентифицирующий начисление для конкретного сотрудника, и по этому номеру ищем соответствующую запись в выборке из результатов запроса.

Если в результате запроса есть запись с таким номером строки, мы рассчитываем результат записи регистра расчета. Т.е. мы получаем начисление по окладу для каждого сотрудника как результат от деления начисленной суммы (поле регистра **ИсходныеДанные**) на количество рабочих дней в месяце (**Норма**) и умножения на фактически отработанные рабочие дни (**Факт**).

Добавим текст запроса во вторую ветку условия с той лишь разницей, что теперь мы будем получать значения базы, используя виртуальную таблицу регистра расчета **РегистрРасчета.Начисления.БазаНачисления**.

```

// Рассчитать вторичные записи
ИначеЕсли ТребуемыйВидРасчета = ПланыВидовРасчета.ОсновныеНачисления.Премия
Тогда

    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        | НачисленияБазаНачисления.РезультатБаза КАК База,
        | НачисленияБазаНачисления.НомерСтроки КАК НомерСтроки
        | ИЗ
        | РегистрРасчета.Начисления.БазаНачисления(&ИзмеренияОсновного,
        | &ИзмеренияБазового, , Регистратор = &Регистратор И ВидРасчета =
        &ВидРасчета И
        | Сотрудник В (&СписокСотрудников)) КАК НачисленияБазаНачисления";

    Измер = Новый Массив(1);
    Измер[0] = "Сотрудник";

    Запрос.УстановитьПараметр("ИзмеренияОсновного", Измер);
    Запрос.УстановитьПараметр("ИзмеренияБазового", Измер);
    Запрос.УстановитьПараметр("Регистратор", Регистратор);
    Запрос.УстановитьПараметр("ВидРасчета", ТребуемыйВидРасчета);
    Запрос.УстановитьПараметр("СписокСотрудников", СписокСотрудников);

```



```

ВыборкаРезультата = Запрос.Выполнить().Выбрать();
    КонецЕсли;

КонецПроцедуры

```

В параметрах виртуальной таблицы запроса мы кроме регистратора, вида расчета и списка сотрудников задаем еще измерения основного и базового регистров. В нашем случае это будет один и тот же регистр **Начисления**, а нужное нам измерение – **Сотрудник**.

В заключение осталось добавить во второе условие обход набора записей регистра расчета и вычисление результата вторичных записей.

```

// Рассчитать вторичные записи
ИначеЕсли ТребуемыйВидРасчета = ПланыВидовРасчета.ОсновныеНачисления.Премия
Тогда
...
ВыборкаРезультата = Запрос.Выполнить().Выбрать();

Для Каждого ЗаписьРегистра Из НаборЗаписейРегистра Цикл
    СтруктураНомер = Новый Структура("НомерСтроки");
    СтруктураНомер.НомерСтроки = ЗаписьРегистра.НомерСтроки;
    ВыборкаРезультата.Сбросить();
    Если ВыборкаРезультата.НайтиСледующий(СтруктураНомер) Тогда
        ЗаписьРегистра.Результат = ВыборкаРезультата.База * (10 / 100);
        Сообщение = Новый СообщениеПользователю;
        Сообщение.Текст = "Выполнен расчет" +
ЗаписьРегистра.Регистратор + " – " + ЗаписьРегистра.ВидРасчета + " – " +
ЗаписьРегистра.Сотрудник;
        Сообщение.Сообщить();
    КонецЕсли;
КонецЦикла;

    КонецЕсли;

КонецПроцедуры

```

Сумму начисленной премии мы рассчитываем как 10% от рассчитанной оплаты по окладу.

Общий вид на данный момент общего модуля:

```

Процедура РассчитатьНачисления(НаборЗаписейРегистра, ТребуемыйВидРасчета,
СписокСотрудников) Экспорт

```

```

    Регистратор=НаборЗаписейРегистра.Отбор.Регистратор.Значение;

```

```

    // Рассчитать первичные записи

```

```

Если ТребуемыйВидРасчета = ПланыВидовРасчета.ОсновныеНачисления.Оклад
Тогда

    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
          | НачисленияДанныеГрафика.ЗначениеПериодДействия КАК Норма,
          |
НачисленияДанныеГрафика.ЗначениеФактическийПериодДействия КАК Факт,
          | НачисленияДанныеГрафика.НомерСтроки КАК НомерСтроки
          |ИЗ
          | РегистрРасчета.Начисления.ДанныеГрафика(Регистратор =
&Регистратор И
          | ВидРасчета = &ВидРасчета И Сотрудник В (&СписокСотрудников))
          | КАК НачисленияДанныеГрафика";

    Запрос.УстановитьПараметр("Регистратор", Регистратор);
    Запрос.УстановитьПараметр("ВидРасчета", ТребуемыйВидРасчета);
    Запрос.УстановитьПараметр("СписокСотрудников", СписокСотрудников);

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

    Для Каждого ЗаписьРегистра Из НаборЗаписейРегистра Цикл
        СтруктураНомер = Новый Структура("НомерСтроки");
        СтруктураНомер.НомерСтроки = ЗаписьРегистра.НомерСтроки;
        ВыборкаРезультата.Сбросить();
        Если ВыборкаРезультата.НайтиСледующий(СтруктураНомер) Тогда
            Если ВыборкаРезультата.Норма = 0 Тогда
                Сообщение = Новый СообщениеПользователю;
                Сообщение.Текст = "Вид расчета: Оклад – Нет
рабочих дней в заданном периоде";
                Сообщение.Сообщить();
                ЗаписьРегистра.Результат = 0;
            Иначе

```

```

// Рассчитать оклад по фактическому периоду и исходным данным

        ЗаписьРегистра.Результат =
(ЗаписьРегистра.ИсходныеДанные /ВыборкаРезультата.Норма) *
ВыборкаРезультата.Факт;

        Сообщение = Новый СообщениеПользователю;

        Сообщение.Текст = "Выполнен расчет" +
ЗаписьРегистра.Регистратор + " - " + ЗаписьРегистра.ВидРасчета + " - "
        + ЗаписьРегистра.Сотрудник;

        Сообщение.Сообщить();

        КонецЕсли;

    КонецЕсли;

КонецЦикла;

// Рассчитать вторичные записи

    ИначеЕсли ТребуемыйВидРасчета =
ПланыВидовРасчета.ОсновныеНачисления.Премия Тогда

        Запрос = Новый Запрос;

        Запрос.Текст =

            "ВЫБРАТЬ
            | НачисленияБазаНачисления.РезультатБаза КАК База,
            | НачисленияБазаНачисления.НомерСтроки КАК НомерСтроки
            | ИЗ
            | РегистрРасчета.Начисления.БазаНачисления(&ИзмеренияОсновного,
            | &ИзмеренияБазового, , Регистратор = &Регистратор И ВидРасчета =
&ВидРасчета И
            | Сотрудник В (&СписокСотрудников)) КАК НачисленияБазаНачисления";

        Измер = Новый Массив(1);
        Измер[0] = "Сотрудник";

        Запрос.УстановитьПараметр("ИзмеренияОсновного", Измер);
        Запрос.УстановитьПараметр("ИзмеренияБазового", Измер);
        Запрос.УстановитьПараметр("Регистратор", Регистратор);
        Запрос.УстановитьПараметр("ВидРасчета", ТребуемыйВидРасчета);
        Запрос.УстановитьПараметр("СписокСотрудников", СписокСотрудников);

```

```

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

Для Каждого ЗаписьРегистра Из НаборЗаписейРегистра Цикл
СтруктураНомер = Новый Структура("НомерСтроки");
СтруктураНомер.НомерСтроки = ЗаписьРегистра.НомерСтроки;
ВыборкаРезультата.Сбросить();
Если ВыборкаРезультата.НайтиСледующий(СтруктураНомер) Тогда
    ЗаписьРегистра.Результат = ВыборкаРезультата.База * (10 / 100);
    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = "Выполнен расчет" + ЗаписьРегистра.Регистратор + "
- " + ЗаписьРегистра.ВидРасчета + " - " + ЗаписьРегистра.Сотрудник;
    Сообщение.Сообщить();
КонецЕсли;
КонецЦикла;
КонецЕсли;
КонецПроцедуры

```

## В режиме 1С:Предприятие

Запустите режим отладки и проверим правильность работы процедуры расчета.

Отменим проведение документа **Начисления сотрудникам №3 (Все действия – Отмена проведения)** и перепроведем документы **Начисления сотрудникам №1** и **№2**. Регистр расчета **Начисления** должен выглядеть так:

Движения в регистре Начисления									
Период регистрации	Регистратор	Н...	Вид ра...	С...	Сотрудник	Результат	График раб...	Исходн...	Период дейст
01.05.2011 0:00:00	Начисления сотрудн...	1	Оклад		Гусаков Николай Дми...	7 000,00	График ма...	7 000,00	01.07.2011 0:00:00
01.05.2011 0:00:00	Начисления сотрудн...	2	Оклад		Деловой Иван Серге...	8 000,00	График ма...	8 000,00	01.07.2011 0:00:00
01.05.2011 0:00:00	Начисления сотрудн...	3	Оклад		Симонов Валерий Мих...	3 000,00	График ма...	3 000,00	01.07.2011 0:00:00
01.05.2011 0:00:00	Начисления сотрудн...	1	Премия		Гусаков Николай Дми...	700,00	График ма...		01.07.2011 0:00:00
01.05.2011 0:00:00	Начисления сотрудн...	2	Премия		Деловой Иван Серге...	800,00	График ма...		01.07.2011 0:00:00

Движения в регистре Начисления

↔

🔍

Найти...

🔗

Все действия ▾

твия	Дата начала периода действия	Дата окончания периода действия	Дата начала базового периода	Дата окончания базового периода
00:00	01.07.2011 0:00:00	31.07.2011 23:59:59	01.07.2011 0:00:00	31.07.2011 23:59:59
00:00	01.07.2011 0:00:00	31.07.2011 23:59:59	01.07.2011 0:00:00	31.07.2011 23:59:59
00:00	01.07.2011 0:00:00	31.07.2011 23:59:59	01.07.2011 0:00:00	31.07.2011 23:59:59
00:00	01.07.2011 0:00:00	31.07.2011 23:59:59	01.07.2011 0:00:00	31.07.2011 23:59:59
00:00	01.07.2011 0:00:00	31.07.2011 23:59:59	01.07.2011 0:00:00	31.07.2011 23:59:59

Мы видим, что всем сотрудникам произведены начисления по окладу (поле **Результат**) за полный месяц в соответствии с исходными данными (поле **Исходные данные**).

Сотрудникам Гусакову и Деловому начислена премия 10% от оклада.

Проведем документы **Начисление сотрудникам №3**, затем **№1** и **№2**. При этом отчет Перерасчет должен быть пуст.

Состояние регистра изменится следующим образом (только для июля 2011):

Движения в регистре Начисления

Найти...

Все действия ▾

Период регистрации	Регистратор	Н...	Вид расчета	С..	Сотрудник	Результат	График работы	Исходные да
01.05.2011 0:00:00	Начисления сотрудн...	1	Оклад		Гусаков Николай Дми...	5 000,00	График мастеров	
01.05.2011 0:00:00	Начисления сотрудн...	2	Оклад		Деловой Иван Серге...	8 000,00	График мастеров	
01.05.2011 0:00:00	Начисления сотрудн...	3	Оклад		Симонов Валерий Мих...	3 000,00	График мастеров	
01.05.2011 0:00:00	Начисления сотрудн...	1	Премия		Гусаков Николай Дми...	500,00	График мастеров	
01.05.2011 0:00:00	Начисления сотрудн...	2	Премия		Деловой Иван Серге...	800,00	График мастеров	
01.05.2011 0:00:00	Начисления сотрудн...	1	Невыход		Гусаков Николай Дми...		График мастеров	

Движения в регистре Начисления

↔

🔍

Найти...

🔍

Все действия ▾

аные	Период действия	Дата начала периода действия	Дата окончания периода действия	Дата начала базового периода
7 000,00	01.07.2011 0:00:00	01.07.2011 0:00:00	31.07.2011 23:59:59	01.07.2011 0:00:00
8 000,00	01.07.2011 0:00:00	01.07.2011 0:00:00	31.07.2011 23:59:59	01.07.2011 0:00:00
3 000,00	01.07.2011 0:00:00	01.07.2011 0:00:00	31.07.2011 23:59:59	01.07.2011 0:00:00
	01.07.2011 0:00:00	01.07.2011 0:00:00	31.07.2011 23:59:59	01.07.2011 0:00:00
	01.07.2011 0:00:00	01.07.2011 0:00:00	31.07.2011 23:59:59	01.07.2011 0:00:00
	01.07.2011 0:00:00	01.07.2011 0:00:00	10.07.2011 23:59:59	01.07.2011 0:00:00

В результате невыхода на работу Гусакова сумма оплаты по окладу будет уменьшена и соответственно уменьшится премия.

### Отчет о начислениях сотрудникам

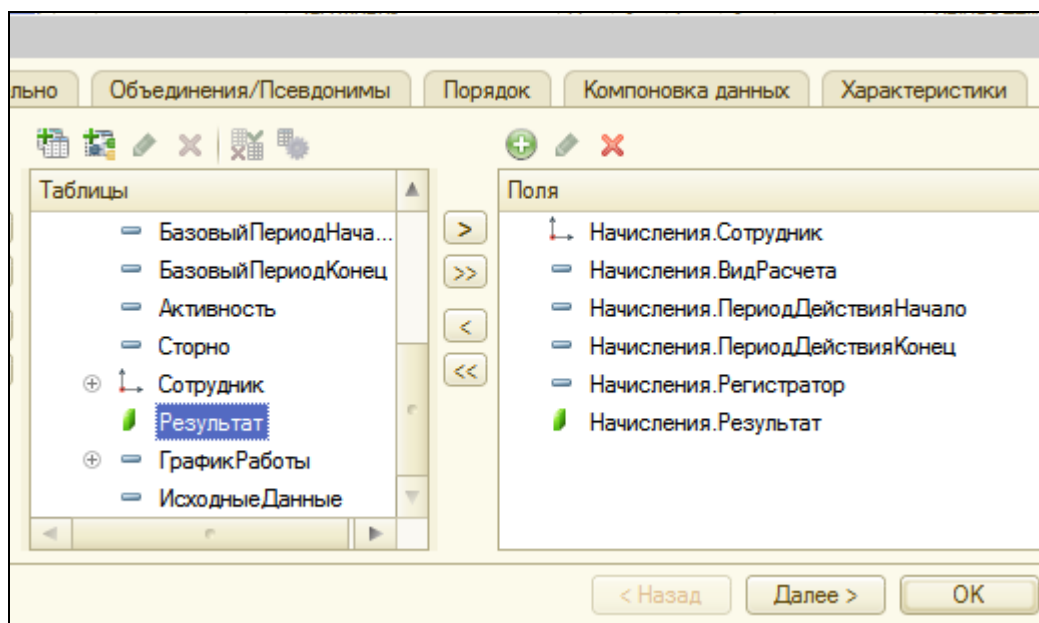
Теперь мы посмотрим, каким образом можно использовать данные, хранящиеся в регистре расчета, для получения в отчете итоговой информации о начислениях сотрудникам.

## В режиме Конфигуратор

Создадим новый объект **Отчет** с именем **НачисленияСотрудникам**. Создадим основную схему компоновки данных, добавим новый **Набор данных – запрос**, откроем конструктор запроса.

Выберем таблицу регистра расчета **Начисления**. Из нее выберем поля:

- Сотрудник,
- ВидРасчета,
- ПериодДействияНачало,
- ПериодДействияКонец,
- Регистратор,
- Результат.



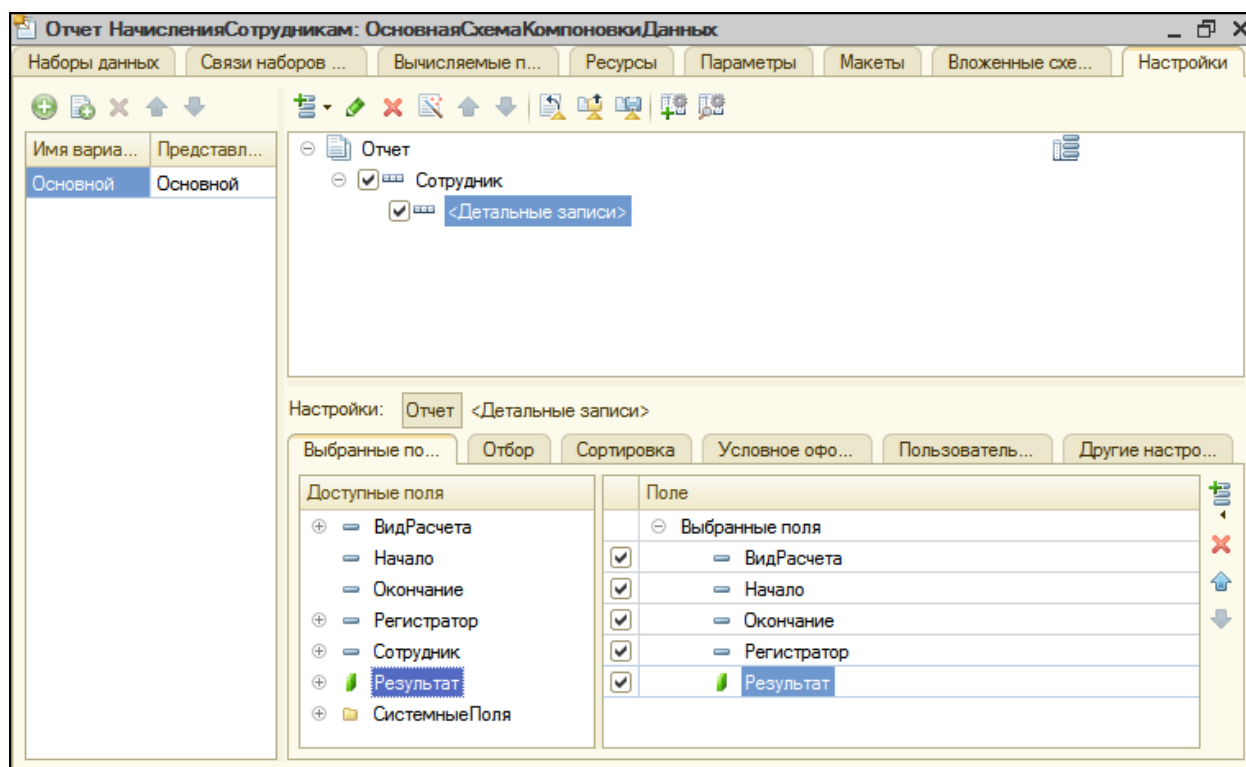
На закладке **Объединения/Псевдонимы** определим псевдонимы полей **ПериодДействияНачало** и **ПериодДействияКонец** как **Начало** и **Окончание** соответственно.

На этом создание запроса закончено, нажмите ОК.

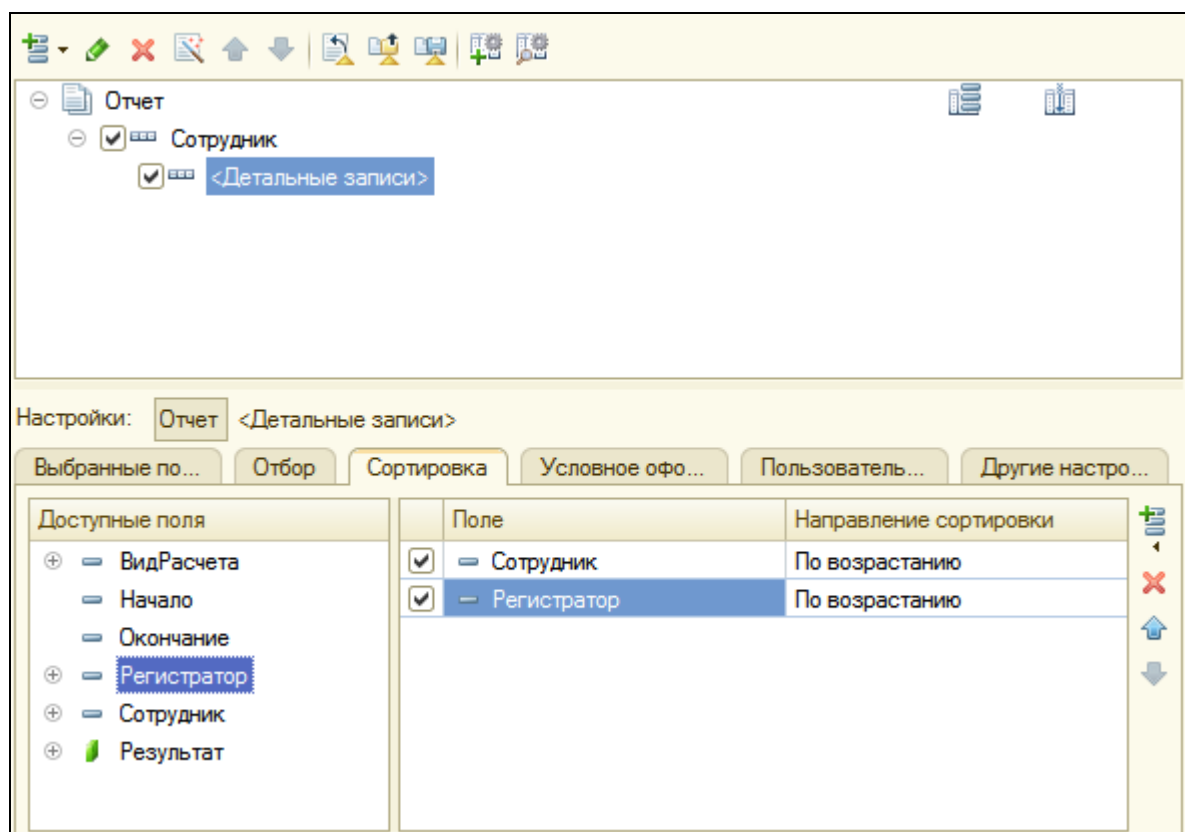
Перейдите на закладку **Ресурсы** и укажите, что должна быть рассчитана сумма по полю результат, дважды щелкнув на нем.

На закладке **Настройки** добавим группировку по полю **Сотрудник** и в ней – подчиненную группировку детальных записей.

В качестве полей, выводимых в отчет, выберем **ВидРасчета**, **Начало**, **Окончание**, **Регистратор** и **Результат**.



На закладке **Сортировка** укажем, что она должна выполняться по возрастанию значения поля **Сотрудник** и **Регистратор**.

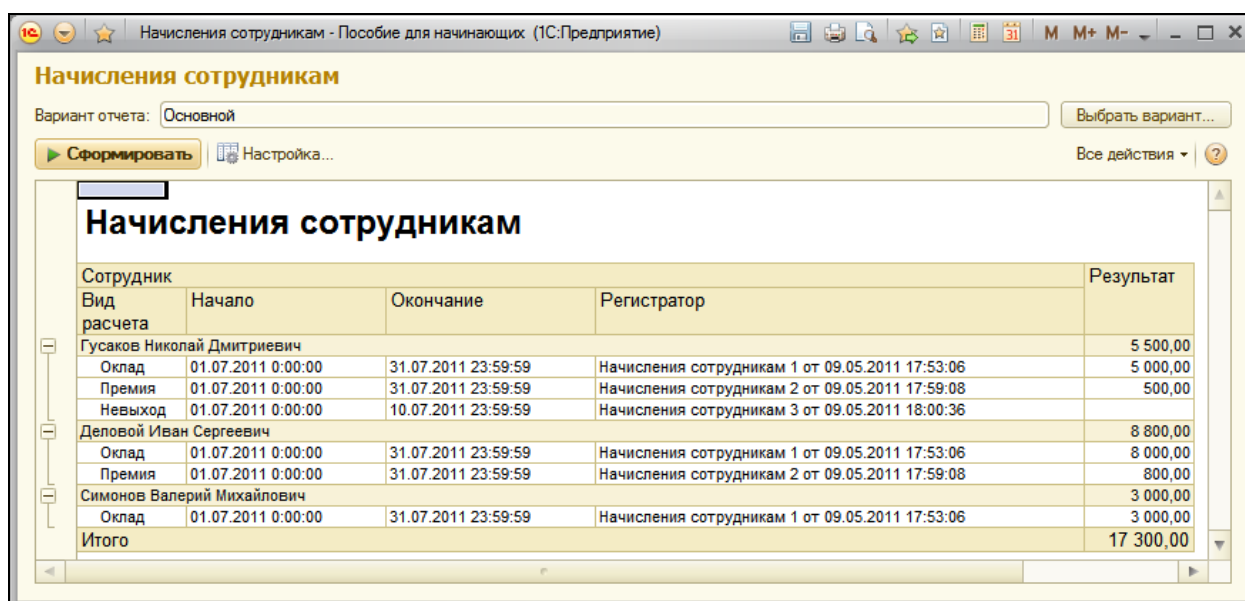


На закладке **Другие настройки** зададим заголовок отчета – **Начисления сотрудникам**. Закройте схему компоновки.

В окне редактирования отчета на закладке **Подсистемы** укажите **РасчетЗарплаты** и **Бухгалтерия**.

В режиме 1С:Предприятие

Запустите режим отладки. В командной панели раздела **Расчет зарплаты** выполните команду создания отчета **Начисления сотрудникам** и сформируйте его.



Сотрудник	Вид расчета	Начало	Окончание	Регистратор	Результат
Гусаков Николай Дмитриевич	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	5 500,00
	Премия	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 2 от 09.05.2011 17:59:08	500,00
	Невыход	01.07.2011 0:00:00	10.07.2011 23:59:59	Начисления сотрудникам 3 от 09.05.2011 18:00:36	
Деловой Иван Сергеевич	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	8 800,00
	Премия	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 2 от 09.05.2011 17:59:08	800,00
Симонов Валерий Михайлович	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	3 000,00
Итого					17 300,00

## Перерасчет


В нашем алгоритме осталось одно узкое место – контроль актуальности данных в регистре расчета.

До сих пор мы использовали служебный отчет **Перерасчет** для определения актуальности данных (актуальны, если отчет пуст).


Поскольку единственным способом получения итоговой информации о начислениях сотрудникам в нашей конфигурации является отчет **НачисленияСотрудникам**, для вызова этой процедуры мы создадим основную форму этого отчета и добавим в командную панель формы кнопку **Перерассчитать**, по которой будет выполняться перерасчет данных регистра.



## В режиме Конфигуратор

В окне редактирования отчета **НачисленияСотрудникам** перейдем на закладку **Формы**, нажмем кнопку открытия  и создадим основную форму отчета.

В правом верхнем окне редактора форм перейдем на закладку **Команды** и на закладке **Команды формы** создадим команду формы **Перерассчитать**.

Теперь нужно установить **Действие** для этой команды. Для этого нажмите кнопку открытия  в строке **Действия**.

В модуле формы будет создан шаблон процедуры **Перерассчитать()**, в которую мы поместим вызов процедуры **ПерерассчитатьНачисления()** из общего модуля **ПроведениеРасчетов**.

```
&НаКлиенте
Процедура Перерассчитать(Команда)
    ПроведениеРасчетов.ПерерассчитатьНачисления(ПредопределенноеЗначение
        ("ПланВидовРасчета.ОсновныеНачисления.Оклад"));
    ПроведениеРасчетов.ПерерассчитатьНачисления(ПредопределенноеЗначение
        ("ПланВидовРасчета.ОсновныеНачисления.Премия"));
КонецПроцедуры
```

Саму процедуру перерасчета поместим в общем модуле **ПроведениеРасчетов** после процедуры **РассчитатьНачисления()**.

```
Процедура ПерерассчитатьНачисления(ТребуемыйВидРасчета) Экспорт

    // Здесь следует выбрать из набора записей перерасчета записи в следующей
    // последовательности:
    // записи документа1 для сотрудников из списка,
    // записи документа2 для сотрудников из списка, и т. д.
    Запрос = Новый Запрос(
        "ВЫБРАТЬ
        | НачисленияПерерасчет.ОбъектПерерасчета,
        | НачисленияПерерасчет.Сотрудник
        | ИЗ
        | РегистрРасчета.Начисления.Перерасчет КАК НачисленияПерерасчет
        |
        | ГДЕ
        | НачисленияПерерасчет.ВидРасчета = &ТребуемыйВидРасчета
        |
        | ИТОГИ ПО
        | НачисленияПерерасчет.ОбъектПерерасчета");

    Запрос.УстановитьПараметр("ТребуемыйВидРасчета", ТребуемыйВидРасчета);
    СписокСотрудников = Новый СписокЗначений;

    // Перебрать группировку по регистратору.
```

```

        ВыборкаПоРегистратору =
        Запрос.Выполнить().Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
        Пока ВыборкаПоРегистратору.Следующий() Цикл
            Регистратор = ВыборкаПоРегистратору.ОбъектПерерасчета;

            // Перебрать группировку по сотрудникам для выбранного регистратора
            // и создать список сотрудников.
            ВыборкаПоСотрудникам = ВыборкаПоРегистратору.Выбрать();
            СписокСотрудников.Очистить();

            Пока ВыборкаПоСотрудникам.Следующий() Цикл
                СписокСотрудников.Добавить(ВыборкаПоСотрудникам.Сотрудник);
            КонецЦикла;

            // Получить набор записей регистра расчета для выбранного
            регистратора.
            НаборЗаписей = РегистрыРасчета.Начисления.СоздатьНаборЗаписей();
            НаборЗаписей.Отбор.Регистратор.Значение = Регистратор;
            НаборЗаписей.Прочитать();

            РассчитатьНачисления(НаборЗаписей, ТребуемыйВидРасчета,
            СписокСотрудников);
            НаборЗаписей.Записать( , Истина);

            // Очистить перерассчитанные записи в перерасчете.
            НаборЗаписейПерерасчета =
            РегистрыРасчета.Начисления.Перерасчеты.Перерасчет.СоздатьНаборЗаписей();
            НаборЗаписейПерерасчета.Отбор.ОбъектПерерасчета.Значение =
            Регистратор;
            НаборЗаписейПерерасчета.Записать();
            КонецЦикла;
        КонецПроцедуры

```

В самом начале процедуры мы запросом выбираем данные о записях перерасчетов, содержащие переданный вид расчета и сгруппированные по объекту перерасчета.

Далее, при обходе результата запроса, мы формируем для каждого объекта перерасчета список сотрудников, читаем соответствующие записи регистра расчета и вызываем процедуру **РассчитатьНачисления**, которая использовалась нами при расчете записей документа **НачисленияСотрудникам**.

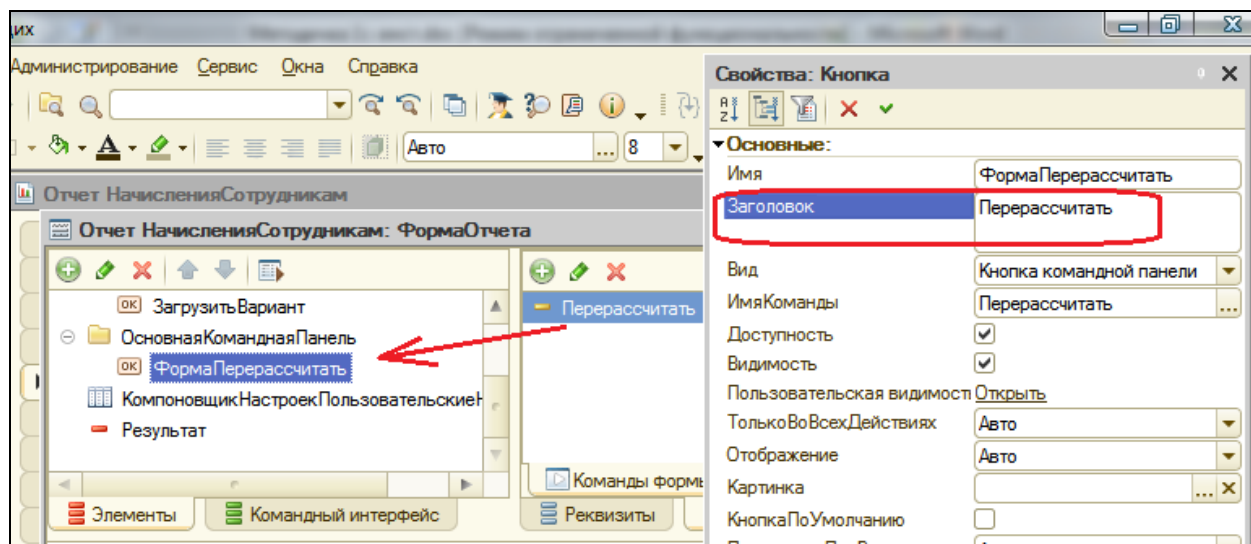
После выполнения расчета записей, мы записываем набор записей без формирования записей перерасчета и очищаем записи перерасчета по тому объекту, который только что обработали.

Вернемся в форму отчета **НачислениеСотрудникам**. Мы указали для команды **Перерассчитать** действие, т.е. процедуру ее выполнения.

Но чтобы воспользоваться этой командой, нужно создать в форме кнопку и связать ее с этой командой (в строке **Команда**).

Проще всего это сделать перетаскиванием команды из окна **Команды формы** в окно элементов формы.

Перетащите мышью команду **Перерассчитать** в группу элементов формы **ОсновнаяКоманднаяПанель**. Задайте заголовок кнопки - **Перерассчитать**.

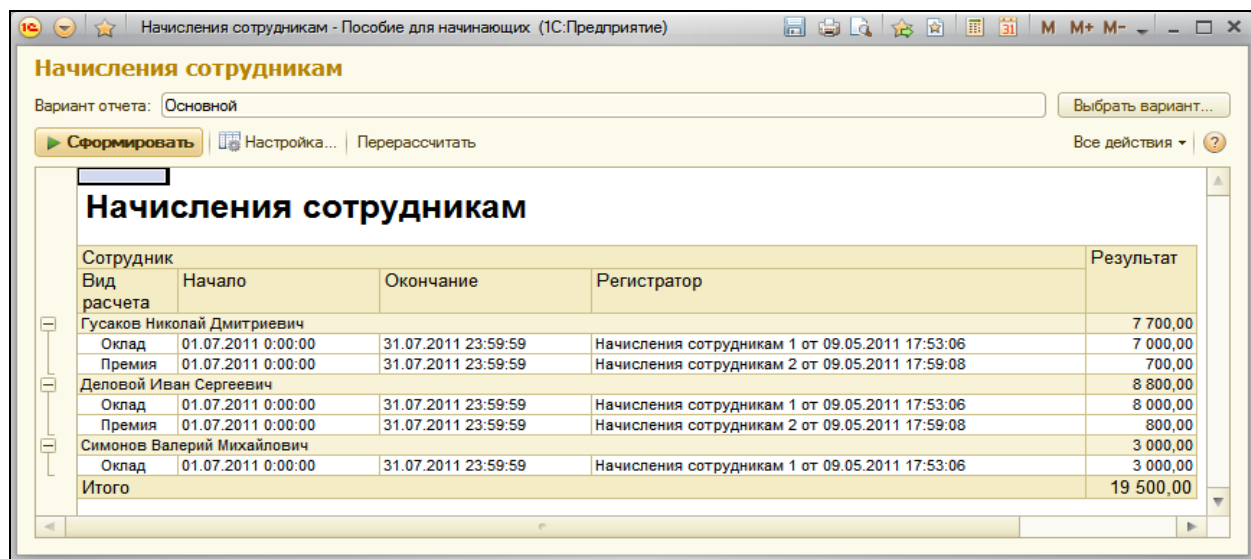


В режиме 1С:Предприятие

Запустим отладку и проверим как выполняется перерасчет.

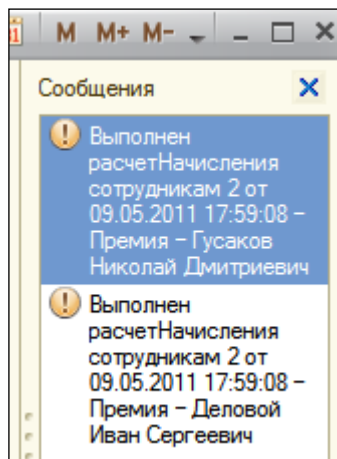
Отменим проведение всех документов **Начисления сотрудникам** и проведем документ **Начисления сотрудникам №1** и затем **№2**.

Сформируем отчет **Начисления сотрудникам**.



Теперь откроем документ **Начисления сотрудникам №1**, изменим оклад Гусакова на 10000 и проведем документ.

В отчете Начисления сотрудникам нажмите **Перерассчитать**. Будет выполнен перерасчет премии Гусакову и Деловому.



Чтобы увидеть в отчете актуальные данные, нажмите **Сформировать**. Результат работы будет содержать новые значения премии для Гусакова.

Начисления сотрудникам - Пособие для начинающих (1С:Предприятие)

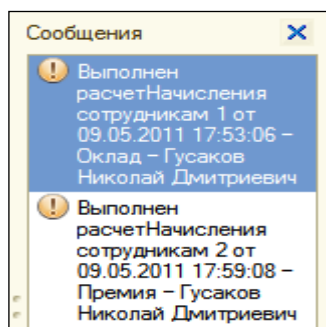
Вариант отчета: Основной Выбрать вариант...

Сформировать Настройка... Перерассчитать Все действия

### Начисления сотрудникам

сотрудник	ид	Начало	Окончание	Регистратор	Результат
Гусаков Николай Дмитриевич	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	11 000,00
	Премия	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 2 от 09.05.2011 17:59:08	1 000,00
Деловой Иван Сергеевич	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	8 800,00
	Премия	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 2 от 09.05.2011 17:59:08	800,00
Имонов Валерий Михайлович	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	3 000,00
	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	3 000,00
<b>того</b>					<b>22 800,00</b>

Наконец, проведем документ **Начисления сотрудникам №3** и нажмем **Перерассчитать** в отчете. Снова будет произведен перерасчет оклада и премии Гусакова.



Нажмите **Сформировать**. Данные отчета будут содержать актуальные значения начисления оклада и премии.

Начисления сотрудникам - Пособие для начинающих (1С:Предприятие)

Начисления сотрудникам

Вариант отчета: Основной

Сформировать Настройка... Перерассчитать

Выбрать вариант... Все действия

### Начисления сотрудникам

Сотрудник	Вид расчета	Начало	Окончание	Регистратор	Результат
Гусakov Николай Дмитриевич	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	7 857,15
	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	7 142,86
	Премия	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 2 от 09.05.2011 17:59:08	714,29
	Невыход	01.07.2011 0:00:00	10.07.2011 23:59:59	Начисления сотрудникам 3 от 09.05.2011 18:00:36	
Деловой Иван Сергеевич	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	8 800,00
	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	8 000,00
	Премия	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 2 от 09.05.2011 17:59:08	800,00
Симонов Валерий Михайлович	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	3 000,00
	Оклад	01.07.2011 0:00:00	31.07.2011 23:59:59	Начисления сотрудникам 1 от 09.05.2011 17:53:06	3 000,00
Итого					19 657,15

## Диаграмма Ганта

В конце этой работы создадим отчет, который в графическом виде будет показывать фактический период действия записей расчета.

Помимо наглядной демонстрации работы механизма вытеснения записей по периоду действия этот отчет позволит познакомиться с элементом формы, позволяющим создавать *диаграммы Ганта*.

*Диаграмма Ганта* представляет собой диаграмму интервалов на шкале времени и отражает использование объектами (точками) ресурсов (серий).

Эта диаграмма будет отображать для каждого сотрудника фактический период действия записи по каждому из видов расчета, имеющих место для этого сотрудника.

В нашем случае точками диаграммы являются сотрудники, а сериями – виды расчетов. Т.о. для каждого сотрудника существует некоторое значение диаграммы по каждой из серий, т.е. по каждому виду расчета.

Все интервалы всех значений диаграммы располагаются с привязкой к единой оси времени, что дает возможность видеть их взаимное расположение.

В качестве исходных данных для построения диаграммы, мы возьмем данные регистра расчета **Начисления**. Каждая запись этого регистра уже содержит все необходимое для построения диаграммы: сотрудника,

вид расчета, начало и конец интервала. Остается средствами встроенного языка разместить все это в диаграмме.

## В режиме Конфигуратор

Создайте новый отчет **ДиаграммаНачислений**. Мы не будем создавать схему компоновки, а создадим основную форму отчета и обеспечим формирование и настройку диаграммы Ганта с помощью кода.

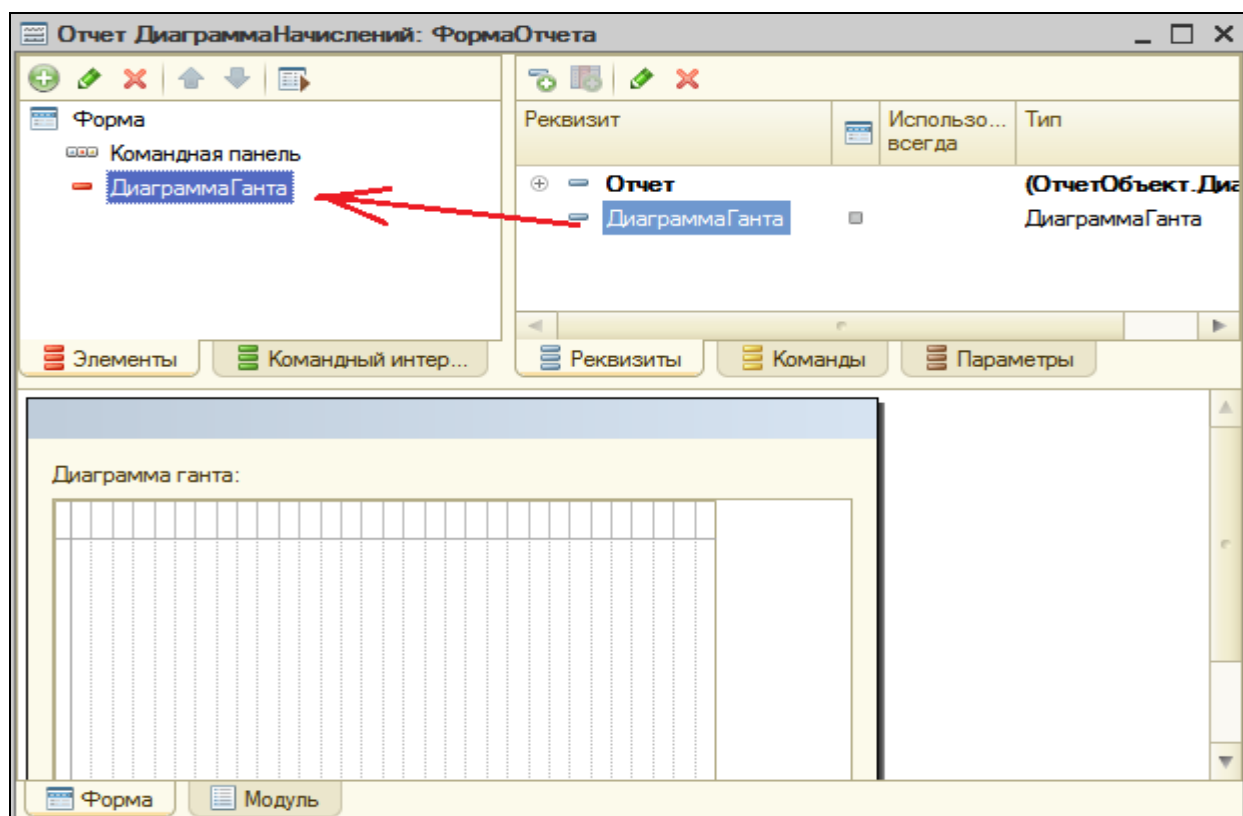
Для этого в окне редактирования отчета перейдите на закладку **Формы**, нажмите кнопку открытия и создайте основную форму отчета.

В правом верхнем углу редактора форм на закладке **Реквизиты** находятся реквизиты формы. Мы видим здесь основной реквизит формы **Отчет**, который был создан автоматически.

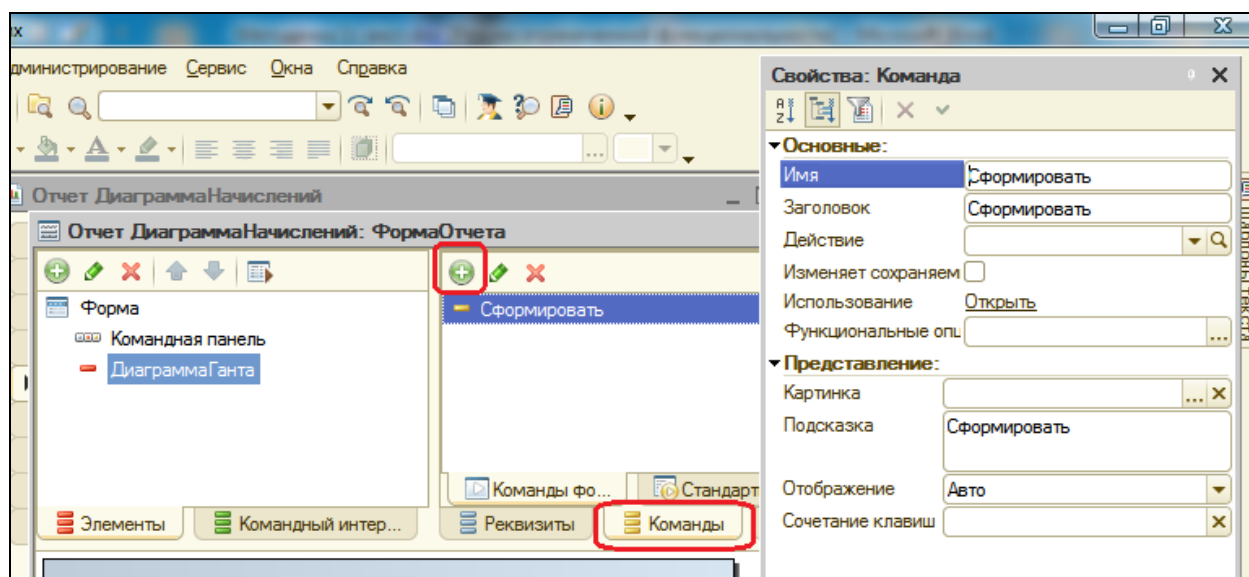
Нажмите кнопку **Добавить** и добавим новый реквизит формы. Назовем его **ДиаграммаГанта** с типом **ДиаграммаГанта**.

Теперь перетащим новый реквизит в окно элементов формы, которое пока пусто.

В окне элементов формы будет создано новое поле для отображения диаграммы Ганта, а в нижнем окне просмотра формы мы сразу увидим поле диаграммы.



На закладке **Команды** создадим команду формы **Сформировать**.



Теперь нужно установить **Действие** для этой команды. Для этого нажмите кнопку открытия в строке **Действие**.

В модуле формы будет создан шаблон процедуры **Сформировать()**, в которую мы поместим вызов процедуры **СформироватьНаСервере()** и в качестве параметра передадим в нее ссылку на реквизит формы **ДиаграммаГанта**.

```
&НаКлиенте
Процедура Сформировать(Команда)
    СформироватьНаСервере(ДиаграммаГанта)
КонецПроцедуры
```

Процедуру **СформироватьНаСервере()** мы поместим также в модуле формы и предварим ее директивой исполнения **&НаСервереБезКонтекста**. В нее мы вставим заготовку запроса.

```
&НаСервереБезКонтекста
Процедура СформироватьНаСервере(Диаграмма)

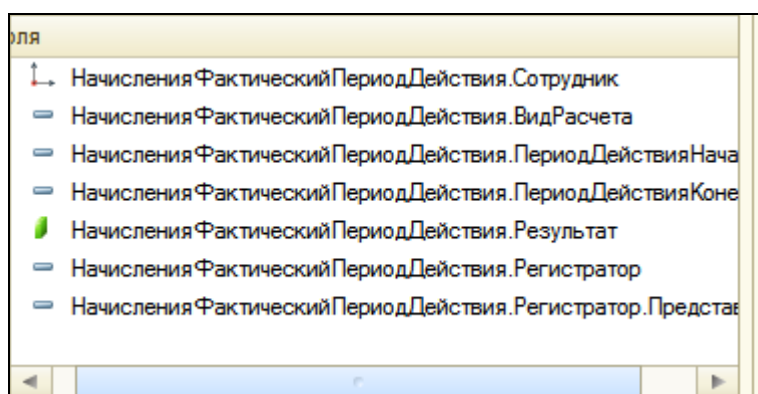
    Запрос = Новый Запрос;
    Запрос.Текст = ;

КонецПроцедуры
```

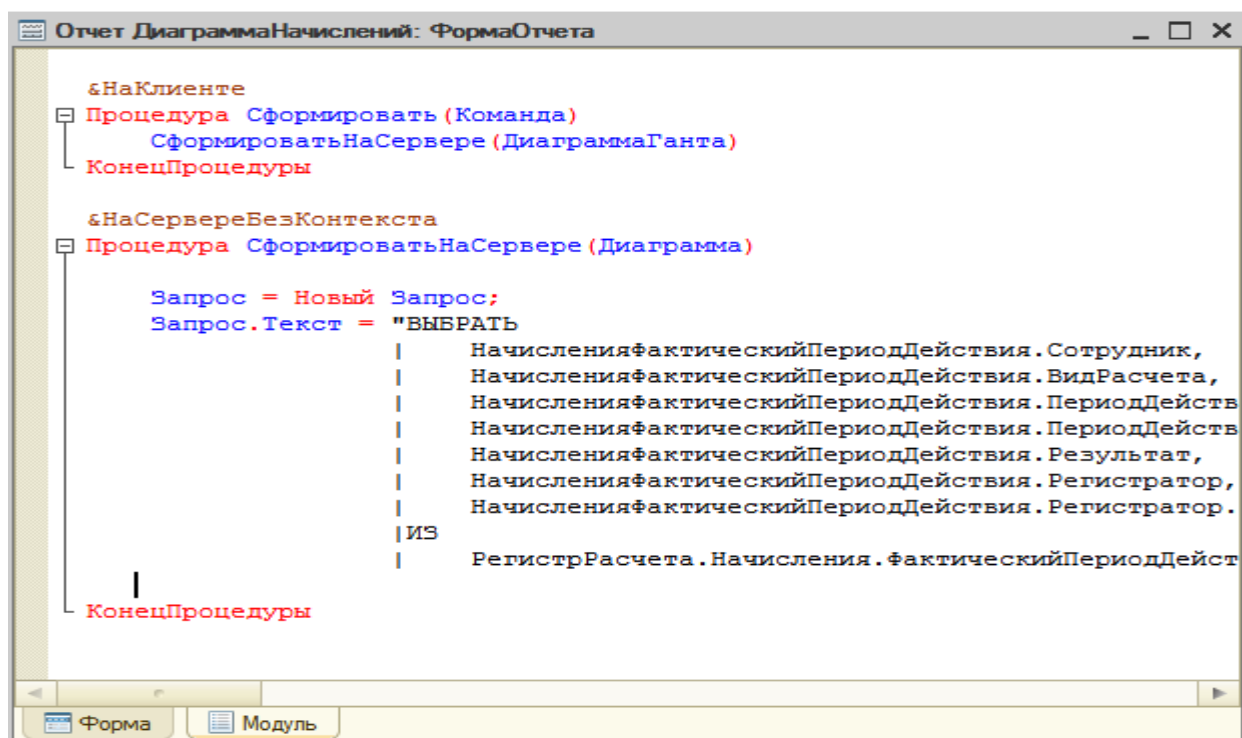
Установите курсор после равно перед точкой с запятой, вызовите контекстное меню, откройте конструктор запроса и создайте новый запрос.

Выберем виртуальную таблицу регистра расчета Начисления, **ФактическийПериодДействия**. Выберем из нее следующие поля:

- **Сотрудник,**
- **ВидРасчета,**
- **ПериодДействияНачало,**
- **ПериодДействияКонец,**
- **Результат,**
- **Регистратор,**
- **Регистратор.Представление.**



Так будет выглядеть созданный запрос:





Нажмите ОК и после текста запроса добавим в процедуру текст:

```
        | РегистрРасчета.Начисления.ФактическийПериодДействия КАК
НачисленияФактическийПериодДействия";

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

// Запретить обновление диаграммы
Диаграмма.Обновление = Ложь;

Диаграмма.Очистить();
Диаграмма.ОтображатьЗаголовок = Ложь;

// Заполнить диаграмму
Пока ВыборкаРезультата.Следующий() цикл
    // Получить серию, точку и значение для них
    ТекущаяСерия =
Диаграмма.УстановитьСерию(ВыборкаРезультата.ВидРасчета);
    ТекущаяТочка =
Диаграмма.УстановитьТочку(ВыборкаРезультата.Сотрудник);
    ТекущееЗначение = Диаграмма.ПолучитьЗначение(ТекущаяТочка,
ТекущаяСерия);

    // Создать нужные нам интервалы в значении
    ТекущийИнтервал = ТекущееЗначение.Добавить();
    ТекущийИнтервал.Начало = ВыборкаРезультата.ПериодДействияНачало;
    ТекущийИнтервал.Конец = ВыборкаРезультата.ПериодДействияКонец;
    ТекущийИнтервал.Текст =
ВыборкаРезультата.РегистраторПредставление;
    ТекущийИнтервал.Расшифровка = ВыборкаРезультата.Регистратор;
КонецЦикла;

// Раскрасить серии своими цветами
Для Каждого Серия из Диаграмма.Серии Цикл
    Если Серия.Значение = ПланыВидовРасчета.ОсновныеНачисления.Оклад
Тогда
        Серия.Цвет = WEBЦвета.Желтый;
    ИначеЕсли Серия.Значение =
ПланыВидовРасчета.ОсновныеНачисления.Премия Тогда
        Серия.Цвет = WEBЦвета.Зеленый;
    ИначеЕсли Серия.Значение =
ПланыВидовРасчета.ОсновныеНачисления.Невыход Тогда
        Серия.Цвет = WEBЦвета.Красный;
    КонецЕсли;
КонецЦикла;

// Разрешить обновление диаграммы
Диаграмма.Обновление = Истина;

КонецПроцедуры
```

Сначала мы запрещаем обновление диаграммы на то время, пока мы будем заполнять ее данными. Это нужно, чтобы в процессе заполнения не выполнялись пересчеты при каждом изменении данных диаграммы.

После окончания заполнения диаграммы мы разрешим обновление, и все пересчеты будут выполнены один раз.

Затем в цикле по выборке запроса мы заполняем диаграмму.

Сначала, используя методы **УстановитьСерию()** и **УстановитьТочку()**, мы получаем либо существующие, либо новые точку и серию. Точки и серии у нас сотрудник и вид расчета из запроса.

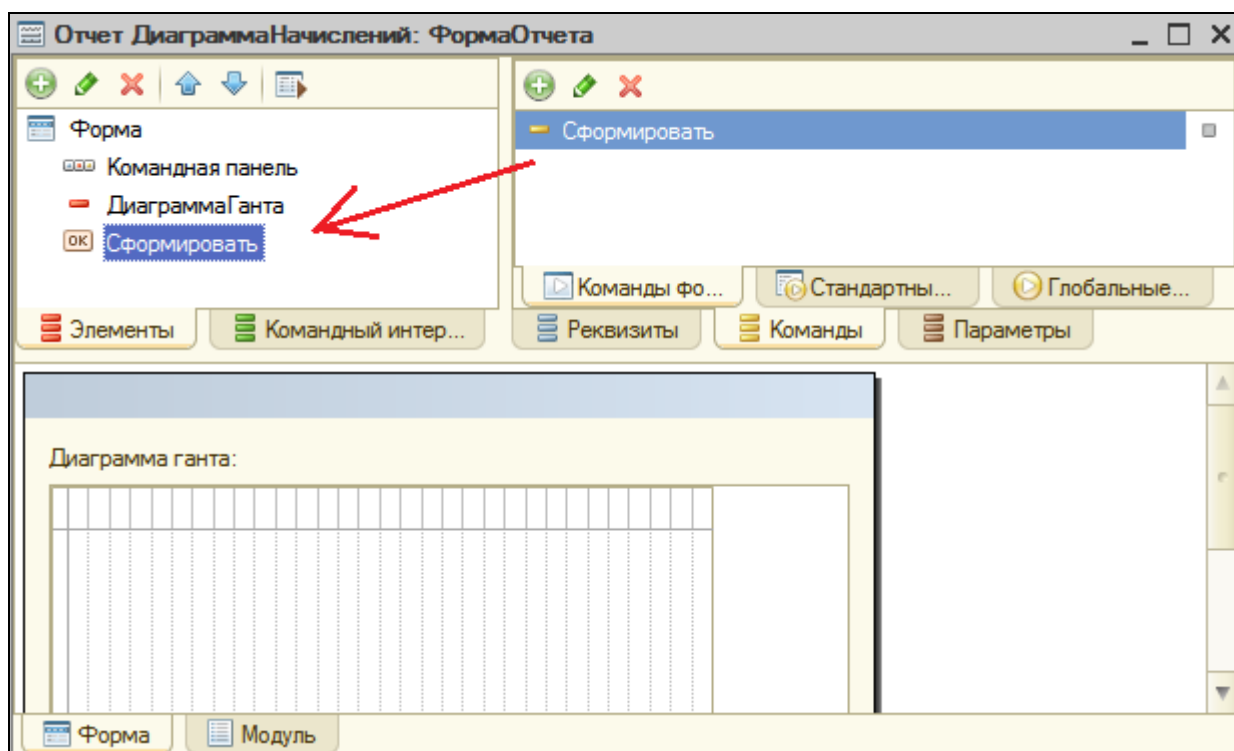
После получения точек и серий, с помощью метода **ПолучитьЗначение()**, мы получаем соответствующее им значение диаграммы.

Затем мы добавляем в значение диаграммы новый интервал, задаем его начало и конец, текст для всплывающей подсказки, расшифровку интервала, которая будет выполняться при двойном щелчке мыши на интервале.

После этого мы раскрашиваем серии своими цветами.

Вернемся в форму и добавим для нее кнопку для выполнения команды **Сформировать**.

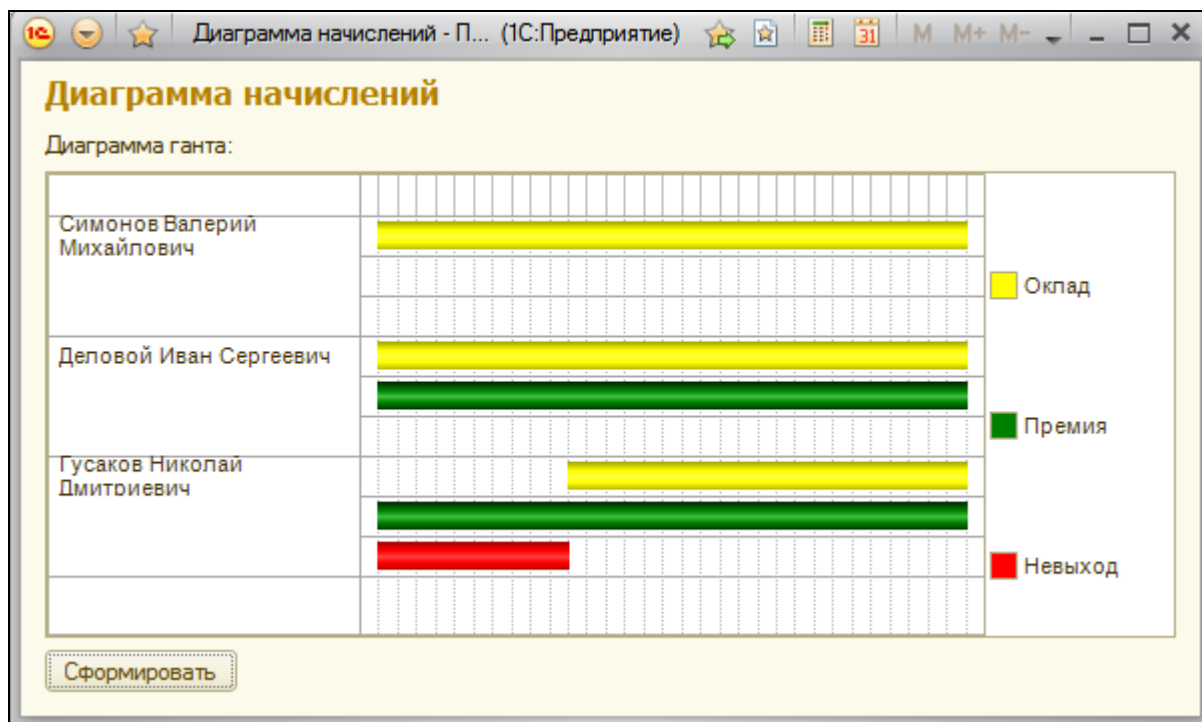
Для этого перетащим мышью команду **Сформировать** из окна **Команды формы** в окно элементов формы.



В заключение в окне редактирования отчета **ДиаграммаНачислений** укажем, что отчет будет вызываться из подсистемы **РасчетЗарплаты**.

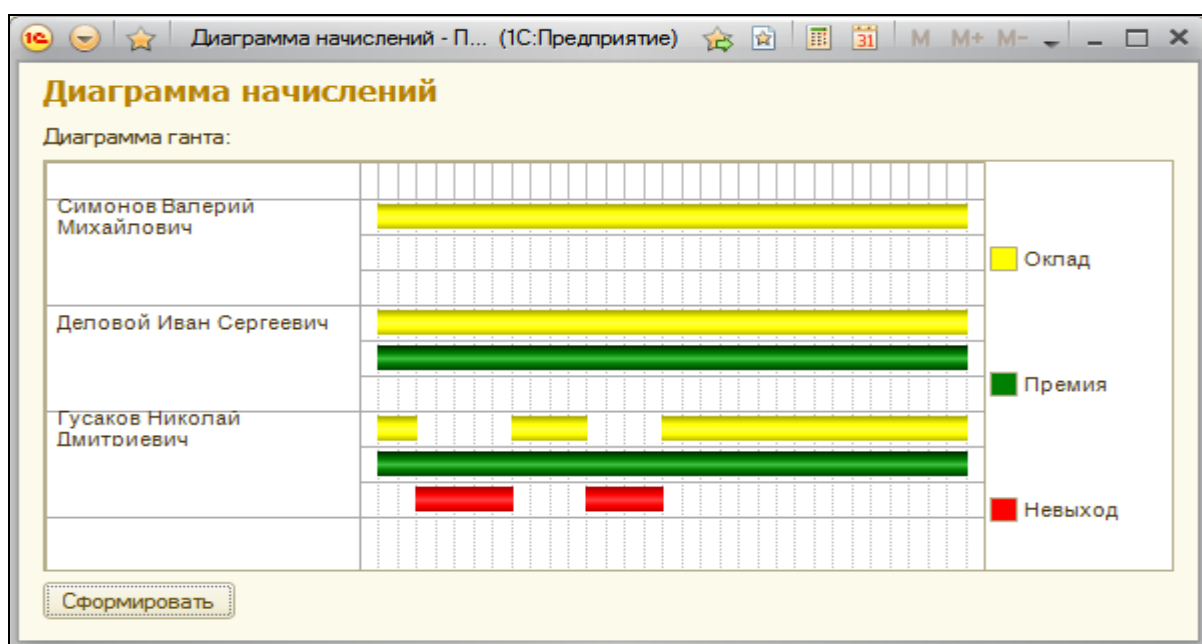
## В режиме 1С:Предприятие

Запустите режим отладки и посмотрите на результат работы отчета.



А теперь посмотрим, как выглядит механизм вытеснения по периоду действия в работе.

Откройте документ **Начисления сотрудникам №3** и вместо одного прогула с 1 по 10 число, зададим Гусакову два прогула: с 3 по 7 число и с 12 по 15 число. Проведем документ и снова нажмем **Сформировать** в нашем отчете.



## **Контрольные вопросы**

- ✓ Как запросом получить записи перерасчета.
- ✓ Как работает перерасчет.
- ✓ Как рассчитать записи регистра расчета.
- ✓ Как запросом получить данные графика и базы.
- ✓ Как выполнить перерасчет отдельных записей регистра расчета.
- ✓ Как устроена и для чего используется диаграмма Ганта.
- ✓ Как заполнить диаграмму Ганта данными.