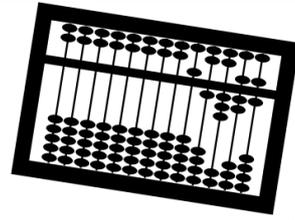


Глава 6



Первый расчет

После кадровика к работе приступает расчетчик. Зарплата уже была посчитана в семерке, и теперь предстояло пересчитать ее в восьмерке. Расчетчик ввел первичные документы и занялся проверкой расчетов...

6.1. Расчетные листки

Настройка расчетных листов... А в чем проблема-то? Взял нажал на кнопку — и список расчетных листов сотрудников готов! Так, да не так...

Расчетчик: Уже два часа сижу, ничего не получается, помогите!

Внедренец: Что случилось?

Расчетчик: Да вот, проверяю, правильно ли сделан расчет зарплаты в старой версии и в новой. Только в старой программе расчетные листки формировались по алфавиту, а в новой — в порядке подразделений. Поэтому приходится перелистывать все пятьдесят листов, пока найдешь нужного человека.

Внедренец: Понятно. Вообще-то в новой программе сделано все правильно, ведь расчетки вы все равно по подразделениям раздаете?

Расчетчик: Ну, да.

Внедренец: Может распечатать расчетные листы из старой программы, а в новой формировать расчетный листок по одному человеку?

Расчетчик: Да, я так уже и начала делать, только все равно, приходится искать человека в списке, нажимать на кнопку...

Внедренец. Ну, хорошо, проблема понятна, сейчас что-нибудь придумаем...

Формировать расчетки по подразделениям удобно при раздаче, а не при проверке (например, для сверки с данными бухгалтерии). Искать расчетку человека в "простыне" из 50 листов, мягко говоря, не очень удобно, если не знаешь, в каком подразделении находится сотрудник.

? Требуется предоставить возможность сделать две формы этого отчета: первую как в программе, вторую — упорядоченную по физическим лицам без разбивки по подразделениям.

Естественно, меняем не сам отчет, а делаем его копию в виде внешнего отчета и уже его настраиваем так, как нам нужно.

6.1.1. Добавляем флажок

На форме размещаем флажок **ГруппироватьПоПодразделениям** (рис. 6.1).

В функции ДействияФормыСформировать (Кнопка) изменяем порядок группировки (листинг 6.1).

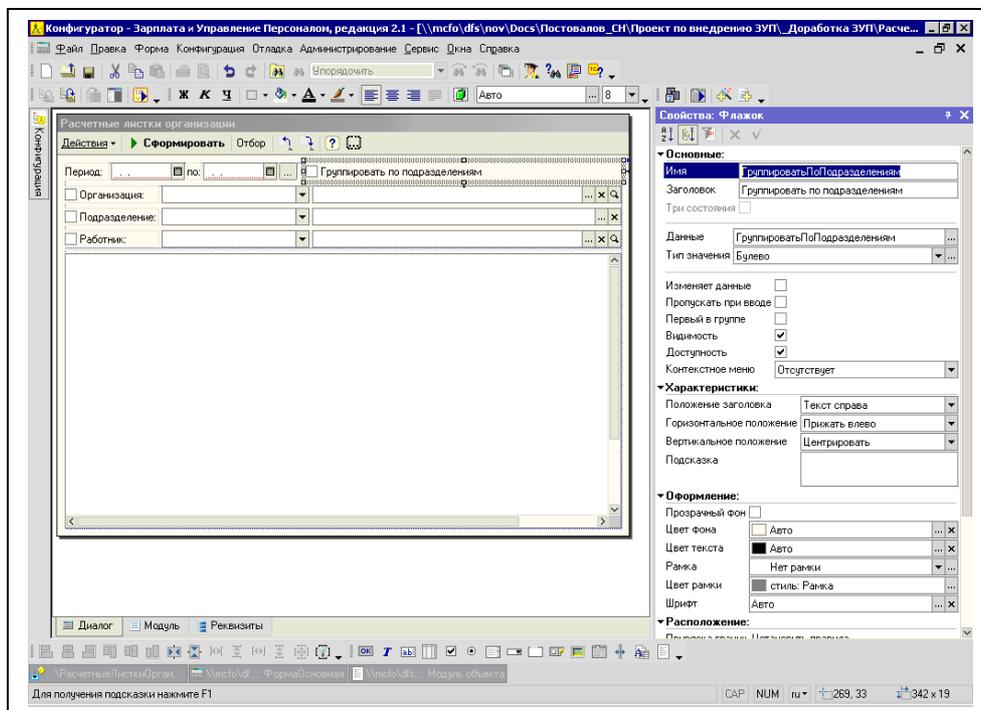


Рис. 6.1. Настройка формы отчета **РасчетныеЛисткиОрганизаций**

Листинг 6.1. Настройка порядка группировки

```
// Группировать = Истина;
// Постовалов, 14.05.07
Группировать = ЭлементыФормы.ГруппироватьПоПодразделениям.Значение;
// ---
```

6.1.2. Ищем запрос

Последовательно проходим цепочку вызовов, начиная с кнопки **Сформировать**:

- ДействияФормыСформировать (Кнопка)
- СформироватьОтчет (ЭлементыФормы. ДокументРезультат)
- ПолучитьРезультатЗапроса ()

В последней функции формируется запрос с названием "ЗапросРЛ". Текст запроса очень большой и несвязанный, поэтому в отладчике ставим точку остановки на строке:

```
ЗапросРЛ.Текст = ТекстЗапроса;
```

и смотрим значение переменной `ТекстЗапроса` (более подробно выполняемые действия описаны в *разд. 2.3*). Запрос состоит из объединения нескольких подзапросов, но нас интересует только упорядочивание в конце текста запроса (листинг 6.2).

Листинг 6.2. Часть запроса, связанная с упорядочиванием данных

```
УПОРЯДОЧИТЬ ПО
    ПериодРегистрации,
    ОрганизацияНаименование,
    ФИОФизлица,
    ВидДвижения,
    ПорядокВидаРасчета,
    ПериодДействияНачало
ИТОГИ
    МИНИМУМ (ПериодДействияНачало) ,
    МАКСИМУМ (ПериодДействияКонец) ,
    МАКСИМУМ (ОтработаноДней) ,
    МАКСИМУМ (ОтработаноЧасов) ,
    СУММА (Начисления) ,
```

```

СУММА (Удержания) ,
СУММА (ВычетНаФизлицо) ,
СУММА (ВычетНаДетей) ,
МАКСИМУМ (ВычетИмущественный) ,
МАКСИМУМ (СуммаВзаиморасчетовНачальныйОстаток) ,
МАКСИМУМ (СуммаВзаиморасчетовКонечныйОстаток)

```

ПО

```

ПериодРегистрации,
Организация,
ПодразделениеОрганизации,
ФИОФизлица, // для сортировки по алфавиту
ФизЛицо

```

В случае если группировка отсутствует, то меняем порядок итогов (листинг 6.3). Переменная Группировать является реквизитом отчета, поэтому она может быть использована в модуле отчета.

Листинг 6.3. Изменение полей группировки

```

Если Группировать Тогда
    ТекстЗапроса = ТекстЗапроса      +
    "
    |ПО
    |ПериодРегистрации,
    |Организация,
    |ПодразделениеОрганизации,
    |ФИОФизлица, // для сортировки по алфавиту
    |ФизЛицо";
Иначе
    ТекстЗапроса = ТекстЗапроса      +
    "
    |ПО
    |ПериодРегистрации,
    |Организация,
    |ФИОФизлица, // для сортировки по алфавиту
    |ПодразделениеОрганизации,
    |ФизЛицо";
КонецЕсли;

```

Теперь осталось внести отчет в список дополнительных внешних отчетов и научить расчетчика пользоваться им, а не встроенным отчетом.



Время на выполнение работы — 1 час.

ЗАМЕЧАНИЕ

Эта проблема с расчетными листками была в ЗиУП редакции 2.1. В редакции 2.5 отчет **Расчетные листки организаций** был существенно переработан. Теперь в нем можно по кнопке **Настройки** задавать произвольные поля для группировки, отбора и упорядочивания данных.

6.2. Районный коэффициент и пособие по уходу за ребенком

Рассматриваемая организация имеет распределенную структуру. Головной офис находится в г. Новосибирске, а обособленные подразделения — в Сочи, Москве, Красноярске, Хабаровске и других городах. Поскольку все эти подразделения не имеют отдельного баланса, мы при переносе данных обозначили их как обычные подразделения.

При анализе расчета пособий по уходу за ребенком возникла следующая проблема. Согласно частям 3 и 4 ст. 15 Федерального закона от 19 мая 1995 г. № 81-ФЗ "О государственных пособиях гражданам, имеющим детей": "В районах и местностях, в которых в установленном порядке применяются районные коэффициенты к заработной плате, минимальный и максимальный размеры указанного пособия определяются с учетом этих коэффициентов".

В связи с этим на все пособия сотрудницам, находящимся в отпуске по уходу за ребенком, "накручивался" федеральный районный коэффициент, установленный для Новосибирска равным 1,2. Так как подобных сотрудниц было немного, то мы рекомендовали расчетчику выполнять корректировку пособий вручную в документе "Начисление зарплаты работникам организаций".

В редакции 2.5 появилась возможность разделять подразделения организации на обычные и обособленные. Для этого в форме подразделения на вкладке **Для обособленных подразделений** нужно указать ОКАТО и другие параметры, в том числе и районный коэффициент.

Так мы и сделали. Ввели ОКАТО для Сочи и установили федеральный районный коэффициент равным 1,0. Сделали расчет по этой сотруднице и получили такой результат (листинг 6.4). Полужирным шрифтом выделены строки, в которых показано, что берется все равно районный коэффициент Новосибирска.

Листинг 6.4. Комментарий расчета пособия по уходу за ребенком

□ Расчет Пособие по уходу за ребенком до 3-х лет

Способ расчета: Пособие по уходу за ребенком до 3-х лет

Порядок расчета: <Пособие по уходу за ребенком до 3-х лет>

Календарных дней в месяце: 30

Оплачиваемые дни отпуска: 10

Учтенный районный коэффициент: 1,2

Размер пособия за месяц: 60

Результат расчета: 20

□ Расчет Пособие по уходу за ребенком до 1.5 лет

Способ расчета: Пособие по уходу за ребенком до 1.5 лет

Порядок расчета: <Средний дневной заработок> * 30.4 / <Число календарных дней в месяце> * <Число календарных дней выплаты пособия>

Размер пособия в процентах от среднего заработка: 40

Размер пособия за месяц исходя из среднего заработка: 9 331,58

Учтенный районный коэффициент: 1,2

Минимальный размер пособия: 3 600

Максимальный размер пособия: 7 200

Расчет производится исходя из максимального размера пособия

Календарных дней в месяце: 30

Оплачиваемые дни отпуска: 10

Результат расчета: 2 400



Требуется найти причину неверного расчета районного коэффициента при вычислении пособия по уходу за ребенком.

Поиск по документации и форумам не дал никаких результатов. Тогда приступаем к поиску по конфигурации. Нам нужно найти, откуда берется районный коэффициент при расчете. Из комментария расчета копируем строку Учтенный районный коэффициент и выполняем глобальный поиск по конфигурации, в результате которого было найдено несколько мест конфигурации, где встречается эта строка (рис. 6.2). Анализируя текст, приходим к выводу, что подходящей является вторая строка. Информация в комментарий расчета берется из переменной ИсходныеДанные.РайонныйКoeffициентРФОрганизации.

Теперь глобальным поиском ищем строку РайонныйКoeffициентРФОрганизации и находим следующий фрагмент (листинг 6.5).

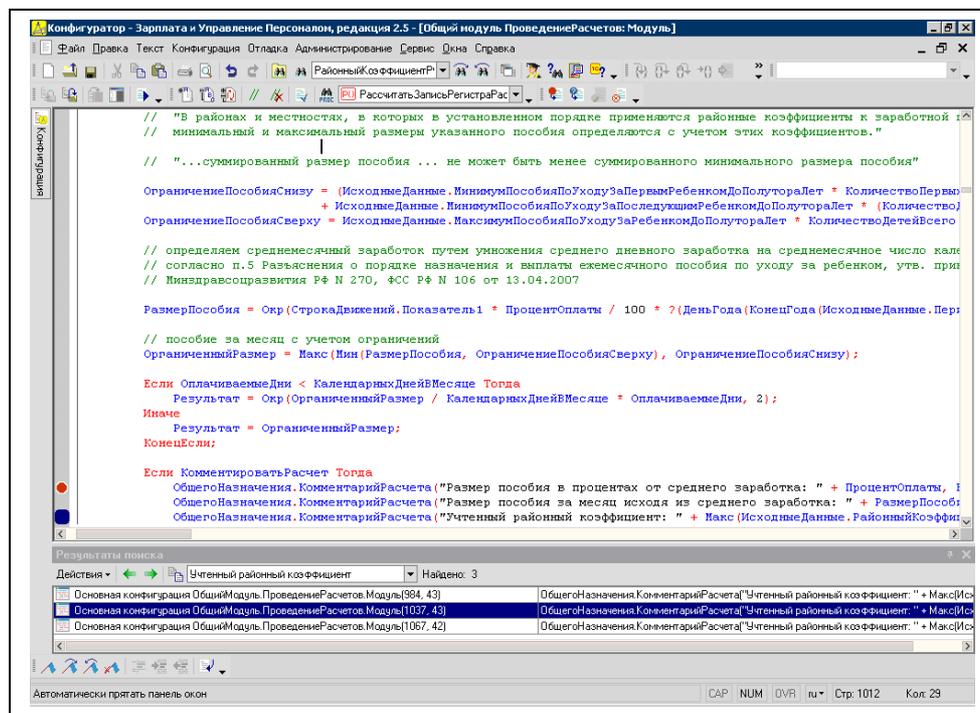


Рис. 6.2. Поиск комментария выполнения расчета

Данные о районном коэффициенте берутся или из реквизита Подразделение-Организация, если ОКАТО не пустое, и районный коэффициент больше единицы, или из реквизита ОбособленноеПодразделение. Но в нашем случае районный коэффициент Сочи равен 1,0, поэтому условие, выделенное в листинге 6.5 полужирным шрифтом, оказывается ложным, и районный коэффициент берется из реквизита ОбособленноеПодразделение, которое у нас совпадает с самой организацией.

Если бы было наоборот, районный коэффициент у организации равен 1,0, а у подразделения — 1,2, то все работало бы правильно. Значит, в листинге 6.5 в указанном условии ошибка!

Чтобы исправить ошибку, нужно в данном условии вместо >1.0 записать $>=1.0$ (листинг 6.6).

Корректируем конфигурацию и отправляем сообщение об этом на хотлайн, надеясь на то, что в следующих релизах конфигурации ошибка будет исправлена.

После этого проверяем результат расчета (листинг 6.7). Все верно, для расчета был взят районный коэффициент 1,0.

Листинг 6.5. Текст запроса по получению районного коэффициента

```

Если НеобходимыеДанные.РайонныйКoeffициентРФОрганизации Тогда
// размер районного коэффициента, установленного для организации
ТекстЗапроса = ТекстЗапроса + ",
|ВЫБОР
| КОГДА Основной.ПодразделениеОрганизации.КодПоОКАТО <> "
| И Основной.ПодразделениеОрганизации.РайонныйКoeffициентРФ > 1
| ТОГДА Основной.ПодразделениеОрганизации.РайонныйКoeffициентРФ
| КОГДА Основной.ОбособленноеПодразделение.РайонныйКoeffициентРФ > 1
| ТОГДА Основной.ОбособленноеПодразделение.РайонныйКoeffициентРФ
| ИНАЧЕ 1
| КОНЕЦ КАК РайонныйКoeffициентРФОрганизации";
Если КомментироватьРасчет Тогда
ПоляЗапроса.Добавить ("РайонныйКoeffициентРФОрганизации");
КонецЕсли;
КонецЕсли;

```

Листинг 6.6. Исправленный текст запроса по получению районного коэффициента

```

Если НеобходимыеДанные.РайонныйКoeffициентРФОрганизации Тогда
// размер районного коэффициента, установленного для организации
ТекстЗапроса = ТекстЗапроса + ",
|ВЫБОР
| КОГДА Основной.ПодразделениеОрганизации.КодПоОКАТО <> "
//| И Основной.ПодразделениеОрганизации.РайонныйКoeffициентРФ > 1
// Постовалов, 08.10.2007
| И Основной.ПодразделениеОрганизации.РайонныйКoeffициентРФ >= 1
// ---
| ТОГДА Основной.ПодразделениеОрганизации.РайонныйКoeffициентРФ
| КОГДА Основной.ОбособленноеПодразделение.РайонныйКoeffициентРФ > 1
| ТОГДА Основной.ОбособленноеПодразделение.РайонныйКoeffициентРФ
| ИНАЧЕ 1
| КОНЕЦ КАК РайонныйКoeffициентРФОрганизации";
Если КомментироватьРасчет Тогда
ПоляЗапроса.Добавить ("РайонныйКoeffициентРФОрганизации");
КонецЕсли;

```

КонецЕсли;

Листинг 6.7. Комментарий расчета после исправления ошибки

Расчет Пособие по уходу за ребенком до 3 лет

Способ расчета: Пособие по уходу за ребенком до 3 лет

Порядок расчета: <Пособие по уходу за ребенком до 3х лет>

Календарных дней в месяце: 30

Оплачиваемые дни отпуска: 10

Учтенный районный коэффициент: 1

Размер пособия за месяц: 50

Результат расчета: 16,67

Расчет Пособие по уходу за ребенком до 1,5 лет

Способ расчета: Пособие по уходу за ребенком до 1.5 лет

Порядок расчета: <Средний дневной заработок> * 30.4 / <Число календарных дней в месяце> * <Число календарных дней выплаты пособия>

Размер пособия в процентах от среднего заработка: 40

Размер пособия за месяц исходя из среднего заработка: 9 331,58

Учтенный районный коэффициент: 1

Минимальный размер пособия: 3 000

Максимальный размер пособия: 6 000

Расчет производится исходя из максимального размера пособия

Календарных дней в месяце: 30

Оплачиваемые дни отпуска: 10

Результат расчета: 2 000



Время на выполнение работы — 2 часа.

6.3. Работа в праздничные и выходные дни

При сравнении расчетов оплаты работы в праздничные и выходные дни выяснилось, что алгоритмы расчета в старой и новой программах немного отличаются.

В ЗиУП при расчете оплаты за праздничный или выходной день используется среднемесячное число рабочих часов:

Среднемесячное число рабочих часов = Число рабочих часов за год / 12

Тарифная часовая = Оклад / Среднемесячное

ставка

число рабочих часов

В то же время при расчете оплаты работы в рабочий день берется число рабочих часов в месяце расчета, т. е. оплата работы в праздничный или выходной день может оказаться меньше, чем в обычный рабочий день.

Поэтому

в организации, в которой проходило внедрение, расчет выполнялся по следующей формуле:

$$\text{Тарифная часовая ставка} = \frac{\text{Оклад}}{\text{Число рабочих часов в месяце}}$$

? Требуется изменить алгоритм расчета оплаты работы в праздничные и выходные дни.

Для решения данной задачи воспользуемся механизмом внешних обработок заполнения табличных частей (см. главу 2). Сначала расчетчик заполняет документ "Оплата праздничных и выходных дней организаций" штатными средствами, а после нажатия кнопки **Заполнить** над табличной частью происходит перерасчет часовой тарифной ставки и суммы оплаты.

Порядок действий будет таков.

Шаг 1. Создаем новую внешнюю обработку заполнения табличной части на основании обработки `ЗаготовкаДляВнешнейОбработкиЗаполненияТЧ.erf`. Сохраняем файл под названием `РасчетЧасовойТарифнойСтавки.erf`.

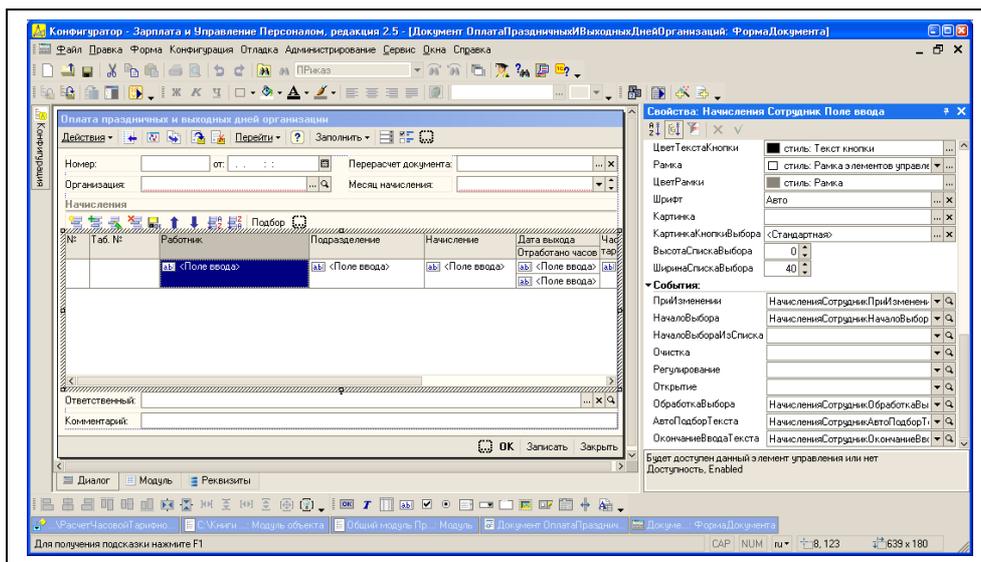


Рис. 6.3. Форма документа Оплата праздничных и выходных дней организаций

Шаг 2. Ищем штатную процедуру расчета часовой тарифной ставки. Для этого открываем документ `ОплатаПраздничныхИВыходныхДнейОрганизаций` в режиме конфигуратора. Открываем форму документа и встаем на поле ввода реквизита `Работник` табличной части (рис. 6.3). В поле свойств видим, что при выборе сотрудника вызывается процедура `НачисленияСотрудникПриИзменении()`.

Открываем ее, щелкнув по кнопке  справа от названия процедуры. Из этой процедуры вызывается другая — `ВыполнитьАвторасчетРеквизитовСтрокиНачисления()`, — выполняющая расчет строки. Нажимаем клавишу <F12> и переходим в эту процедуру. Анализируем ее текст и видим, что расчет часовой тарифной ставки выполняется в процедуре `ЧасоваяТарифнаяСтавкаРаботникаОрг()` общего модуля `ПроцедурыУправленияПерсоналом` (листинг 6.8). Строки, выделенные полужирным шрифтом, скопируем в процедуру `Инициализировать()`, которая перебирает все строки документа и заполняет тариф и результат расчета (листинг 6.9).

Листинг 6.8. Процедура расчета реквизитов строки начисления

```
Процедура ВыполнитьАвторасчетРеквизитовСтрокиНачисления (РасчитатьТариф =
Истина)
ДанныеСтроки = ЭлементыФормы.Начисления.ТекущиеДанные;
ДатаАктуальности = ? (ЗначениеЗаполнено (ДанныеСтроки.ДатаВыхода),
ДанныеСтроки.ДатаВыхода, Дата);
Если РассчитатьТариф Тогда
// Рассчитаем часовую тарифную ставку работника
ЧасовойТариф = ПроцедурыУправленияПерсоналом.ЧасоваяТарифнаяСтавка
РаботникаОрг ( ДанныеСтроки.Сотрудник, ДатаАктуальности,
НачалоМесяца (ПериодРегистрации) ,
Константы.ВалютаРегламентированногоУчета.Получить () );
ДанныеСтроки.Размер = ЧасовойТариф;
КонецЕсли;
// Расчет суммы начисления
ДанныеСтроки.Результат = ДанныеСтроки.Размер * ДанныеСтроки.ОтработаноЧасов;
КонецПроцедуры // ВыполнитьАвторасчетРеквизитовСтрокиНачисления ()
```

Листинг 6.9. Процедура заполнения табличной части

```
Процедура Инициализировать (Док, ТабЧасть, Расшифровка) Экспорт
Сообщить ("Заполнение табличной части "+ТабЧасть+" документа "+Док);
Конт = Док;
ТЧ = Док[ТабЧасть];
Для Каждого Стр Из ТЧ Цикл
```

```

Сообщить (" "+Стр.ФизЛицо) ;
ЧасовойТариф = ЧасоваяТарифнаяСтавкаРаботникаОрг (Стр.Сотрудник,
Док.Дата, НачалоМесяца (Док.Дата), Константы.ВалютаРегламентированногоУче-
та.Получить ());
Стр.Размер = ЧасовойТариф;
Стр.Результат = Стр.Размер*Стр.ОтработаноЧасов;
Сообщить (" ");
КонецЦикла;
КонецПроцедуры

```

Шаг 3. Копируем процедуру ЧасоваяТарифнаяСтавкаРаботникаОрг () из общего модуля ПроцедурыУправленияПерсоналом в модуль внешней обработки. К ссылкам на другие функции общего модуля добавляем в начало имя общего модуля и точку (листинг 6.10, выделено полужирным шрифтом).

Листинг 6.10. Процедура расчета часовой тарифной ставки

```

// Выполняет расчет часовой тарифной ставки работника
Функция ЧасоваяТарифнаяСтавкаРаботникаОрг (Знач Сотрудник,
Знач ДатаАктуальности, Знач ДатаВалютногоУчета,
ВалютаРегламентированногоУчета) Экспорт
//
// ...
//
// Пересчет месячной тарифной ставки в часовую
// определим среднемесячное количество часов
Запрос = Новый Запрос;
Запрос.УстановитьПараметр ("Год", Год (ДатаАктуальности));
Запрос.УстановитьПараметр ("РабочийДень",
Перечисления.ВидыДнейПроизводственногоКалендаря.Рабочий);
Запрос.УстановитьПараметр ("ПредпраздничныйДень",
Перечисления.ВидыДнейПроизводственногоКалендаря.Предпраздничный);
Запрос.Текст =
"ВЫБРАТЬ
| СУММА (ВЫБОР
| КОГДА РегламентированныйПроизводственныйКалендарь.ВидДня = &РабочийДень
| ТОГДА 1
| ИНАЧЕ 0
| КОНЕЦ) КАК ЧислоРабочихДней,
| СУММА (ВЫБОР

```

```

| КОГДА РегламентированныйПроизводственныйКалендарь.ВидДня = &Пред-
| праздничныйДень
| ТОГДА 1
| ИНАЧЕ 0
| КОНЕЦ) КАК ЧислоПредпраздничныхДней
| ИЗ
| РегистрСведений.РегламентированныйПроизводственныйКалендарь КАК Регла-
| ментированныйПроизводственныйКалендарь
| ГДЕ
| РегламентированныйПроизводственныйКалендарь.Год = &Год";
ВыборкаКалендаря = Запрос.Выполнить().Выбрать();
Если ВыборкаКалендаря.Следующий() Тогда
    ЧислоРабочихДнейВГоду = ?(ВыборкаКалендаря.ЧислоРабочихДней = NULL, 0,
    ВыборкаКалендаря.ЧислоРабочихДней);
    ЧислоПредпраздничныхДнейВГоду =
    ?(ВыборкаКалендаря.ЧислоПредпраздничныхДней= NULL, 0,
    ВыборкаКалендаря.ЧислоПредпраздничныхДней);
Иначе
    ЧислоРабочихДнейВГоду = 0;
    ЧислоПредпраздничныхДнейВГоду = 0;
КонецЕсли;
Ставка =
ПроцедурыУправленияПерсоналом. ПолучитьЧасовуюТарифнуюСтавкуНаОсновеМе-
сячнойСтавки (
    Выборка.ОкладТариф, ПродолжительностьРабочейНедели,
    ЧислоРабочихДнейВГоду, ЧислоПредпраздничныхДнейВГоду);
//
// ...
//
Возврат Ставка;
КонецФункции // ЧасоваяТарифнаяСтавкаРаботникаОрг()

```

Шаг 4. Подключаем внешнюю обработку (рис. 6.4) и проверяем, чтобы не было ошибок при выполнении нашей обработки. Если ошибки есть, то они могут быть обусловлены только наличием неопределенных ссылок, связанных с переносом процедуры из общего модуля в модуль обработки.

Шаг 5. Анализируем в процедуре расчета часовой тарифной ставки алгоритм расчета среднемесячного числа рабочих дней. Он заключается в том, что формируется запрос к регламентированному производственному календарю

и определяется число рабочих дней за год (условие `РегламентированныйПроизводственныйКалендарь.Год = &Год`). Нам нужно получить число рабочих дней за месяц, поэтому добавим условие `РегламентированныйПроизводственныйКалендарь.ДатаКалендаря` между `&НачДата` и `&КонДата`, а параметры `НачДата` и `КонДата` установим как начало и конец месяца.

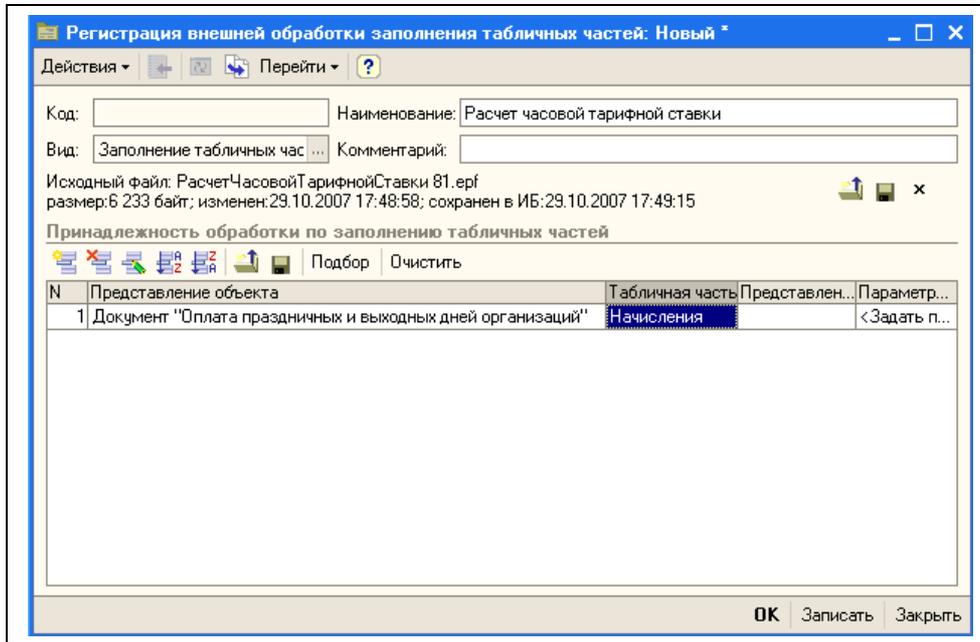


Рис. 6.4. Подключение внешней обработки заполнения табличной части

В результате получаем следующий алгоритм расчета числа рабочих дней (листинг 6.11). Изменения выделены полужирным шрифтом.

Листинг 6.11. Алгоритм расчета числа рабочих часов в месяце

```
// Пересчет месячной тарифной ставки в часовую
// определим среднемесячное количество часов
Запрос = Новый Запрос;
Запрос.УстановитьПараметр ("Год", Год (ДатаАктуальности) );
Запрос.УстановитьПараметр ("РабочийДень", Перечисления.
ВидыДнейПроизводственногоКалендаря.Рабочий) ;
```

```

Запрос.УстановитьПараметр ("ПредпраздничныйДень",
Перечисления.ВидыДнейПроизводственногоКалендаря.Предпраздничный);
// Постовалов, меняем период выборки с года на месяц
Запрос.УстановитьПараметр ("НачДата", НачалоМесяца (ДатаАктуальности));
Запрос.УстановитьПараметр ("КонДата", КонецМесяца (ДатаАктуальности));
// ---
Запрос.Текст =
"ВЫБРАТЬ
|СУММА (ВЫБОР
|КОГДА РегламентированныйПроизводственныйКалендарь.ВидДня = &РабочийДень
|ТОГДА 1
|ИНАЧЕ 0
|КОНЕЦ) КАК ЧислоРабочихДней,
|СУММА (ВЫБОР
|КОГДА РегламентированныйПроизводственныйКалендарь.ВидДня = &Пред-
праздничныйДень
|ТОГДА 1
|ИНАЧЕ 0
|КОНЕЦ) КАК ЧислоПредпраздничныхДней
|ИЗ
|РегистрСведений.РегламентированныйПроизводственныйКалендарь КАК Регла-
ментированныйПроизводственныйКалендарь
|ГДЕ
|РегламентированныйПроизводственныйКалендарь.Год = &Год
// Постовалов, вычисляем число рабочих дней за месяц:
|И РегламентированныйПроизводственныйКалендарь.ДатаКалендаря МЕЖДУ
&НачДата И &КонДата
// ---
|";
ВыборкаКалендаря = Запрос.Выполнить().Выбрать();
Если ВыборкаКалендаря.Следующий() Тогда
    ЧислоРабочихДнейВГоду = ? (ВыборкаКалендаря.ЧислоРабочихДней = NULL, 0,
    ВыборкаКалендаря.ЧислоРабочихДней);
    ЧислоПредпраздничныхДнейВГоду =
    ? (ВыборкаКалендаря.ЧислоПредпраздничныхДней= NULL, 0,
    ВыборкаКалендаря.ЧислоПредпраздничныхДней);
Иначе
    ЧислоРабочихДнейВГоду = 0;
    ЧислоПредпраздничныхДнейВГоду = 0;
КонецЕсли;

```

// Постовалов, переносим все предпраздничные дни на рабочие!

```
ЧислоРабочихДнейВГоду = ЧислоРабочихДнейВГоду +
ЧислоПредпраздничныхДнейВГоду;
```

```
ЧислоПредпраздничныхДнейВГоду = 0;
```

```
Ставка = ПроцедурыУправленияПерсоналом.
```

```
ПолучитьЧасовуюТарифнуюСтавкуНаОсновеМесячнойСтавки (Выборка.ОкладТариф,
ПродолжительностьРабочейНедели, ЧислоРабочихДнейВГоду*12,
ЧислоПредпраздничныхДнейВГоду*0);
```

Теперь осталось только сохранить изменения и проверить работоспособность алгоритма.



Время на выполнение работы — 3 часа.

6.4. Подбор списка сотрудников в платежную ведомость

При переносе списка зарплатных счетов сотрудников (см. главу 4), мы их распределили по документам ЗаявкаНаОткрытиеСчета таким образом, чтобы в одном документе находились сотрудники, получающие зарплату в одинаковом банке. Тогда при заполнении документа ЗарплатаКВыплатеОрганизаций потребуется добавить возможность подбора сотрудников по документу ЗаявкаНаОткрытиеСчета.



Требуется разработать внешнюю обработку заполнения табличной части документа ЗарплатаКВыплатеОрганизаций списком сотрудников из документа ЗаявкаНаОткрытиеСчета.

Для решения данной задачи воспользуемся механизмом внешних обработок заполнения табличных частей.

Шаг 1. Создаем новую внешнюю обработку заполнения табличной части на основании обработки ЗаготовкаДляВнешнейОбработкиЗаполненияТЧ.erf. Сохраняем файл под названием ЗаполнениеВыплатыПоЛицевымСчетам.erf.

Шаг 2. Меняем процедуру Инициализировать (), как показано в листинге 6.12.

Листинг 6.12. Процедура заполнения табличной части Зарплата из документа ЗарплатаКВыплатеОрганизаций

```
Процедура Инициализировать (Док, ТабЧасть, Расшифровка) Экспорт
```

```
ТЧ = Док[Табчасть];
```

```
Форма = Документы.ЗаявкаНаОткрытиеСчетов.ПолучитьФормуВыбора ();
Форма.РежимВыбора = Истина;
Рез = Форма.ОткрытьМодально ();
Если Рез<>Неопределено Тогда
    Для Каждого Стр Из Рез.РаботникиОрганизации Цикл
        НовСтр = ТЧ.Добавить ();
        НовСтр.ФизЛицо = Стр.ФизЛицо;
        НовСтр.АвтоРасчет = Истина;
    КонецЦикла;
    // в комментарий записываем, из какого документа
    // взяли список сотрудников
    Док.Комментарий=СокрЛП (Док.Комментарий) +" "+СокрЛП (Рез.Комментарий);
КонецЕсли;
КонецПроцедуры
```

Шаг 3. Подключаем внешнюю обработку и проверяем работу процедуры заполнения.



Время на выполнение работы — 1 час.

6.5. Удаление строк с нулевой суммой

После заполнения документа "Зарплата к выплате организаций" списком сотрудников из документа "Заявка на открытие счета" и расчета суммы зарплаты может оказаться, что в документе останутся строки с нулевой суммой. Такие строки нужно удалить.



Требуется разработать внешнюю обработку заполнения табличной части документа `ЗарплатаКВыплатеОрганизаций`, которая будет удалять строки с нулевой суммой.

Для решения данной задачи воспользуемся механизмом внешних обработок заполнения табличных частей.

Шаг 1. Создаем новую внешнюю обработку заполнения табличной части на основании обработки `ЗаготовкаДляВнешнейОбработкиЗаполненияТЧ.erf`. Сохраняем файл под названием `ЗарплатаКВыплатеУбратьНулевыеСтроки.erf`.

Шаг 2. Меняем процедуру `Инициализировать()`, как показано в листинге 6.13.

Листинг 6.13. Процедура заполнения табличной части Зарплата из документа ЗарплатаКВыплатеОрганизаций

```
Процедура Инициализировать (Док, ТабЧасть, Расшифровка) Экспорт
ТЧ = Док[Табчасть];
КолСтрок = ТЧ.Количество();
Для Н = 1 По КолСтрок Цикл
    Стр = ТЧ.Получить (КолСтрок-Н);
    Если Стр.Сумма = 0 Тогда
        Сообщить ("Удалили строку "+(КолСтрок-Н));
        ТЧ.Удалить (Стр);
    КонецЕсли;
КонецЦикла;
КонецПроцедуры
```

Шаг 3. Подключаем внешнюю обработку и проверяем работу процедуры заполнения.



Время на выполнение работы — 30 мин.

6.6. Выгрузка в Сбербанк

Несмотря на то, что в ЗиУП есть механизм выгрузки зарплаты на карточные счета в Сбербанк РФ, существует еще много других банков, которые устанавливают свои форматы файлов на зачисление зарплаты на карточные счета сотрудников организации. В нашем случае из-за распределенной структуры организации зарплата перечисляется в разных форматах в разные банки по месту жительства сотрудников.

Мы рассмотрим только один пример выгрузки, остальные делаются аналогично.



Требуется разработать процедуру формирования файла на перечисление заработной платы на карточные счета.

На основании платежных ведомостей, которые формируются документами "Зарплата к выплате организаций", вводится документ "Расходный кассовый ордер" при наличной выплате зарплаты либо "Платежное поручение исходящее" — при перечислении зарплаты на карточные счета.

Таким образом, файл выгрузки в банк должен формироваться из документа "Платежное Поручение Исходящее". Одновременно с выгрузкой будем печатать список сумм на зачисление на карточные счета.

Для формирования файла выгрузки нам потребуется некоторая информация о банке:

- номер договора;
- номер отделения банка;
- номер филиала.

Список банков хранится в справочнике контрагентов. Для ввода этой информации добавим в разделе **Адреса и телефоны** соответствующие виды контактной информации (рис. 6.5). Информация будет находиться в регистре сведений КонтактнаяИнформация.

Ко...: Сибирский банк Сбербанка России г. Новосибирск * _ □ ×

Действия ▾ Перейти ▾ ?

Наименование: Сибирский банк Сбербанка России г. Нов Код: 000000001

Полностью: Сибирский банк Сбербанка России г. Новосибирск

Группа: Банки для перечисления зарплаты ... x

ИНН: КПП:

Основной банковский счет:

Счет: Visa Classic ... Q

Основной договор контрагента:

Договор: Договор № 1111 от 06.10.2004 (Classic) ... Q

Адреса и телефоны:

Действия ▾ Основная

	Тип	Вид	Представление
	Другое	ОтделениеБанка	1234
	Другое	НомерФилиала	1234
	Другое	НомерДоговора	1111
	Адрес	Фактический адр...	

OK Записать Закрыть

Рис. 6.5. Настройка контактной информации

Для печати списка на зачисление и выгрузки файла воспользуемся механизмом внешних печатных форм (см. главу 2).

Шаг 1. Создаем новую внешнюю печатную основании обработки `ЗаготовкаДляВнешнейПечатнойФормы.epf`. Сохраняем файл под названием `ПечатьВедомостиВСбербанк.epf`.

Шаг 2. Изменяем процедуру `Печать()` так, как показано в листинге 6.14.

Шаг 3. Создаем макет, как показано на рис. 6.6.

	1	2	3	4	5	6	7	
Шапка	1						Приложение №3	
	2						<к договору (Банк_Договор)>	
	3	СПИСОК						
	4	НА ЗАЧИСЛЕНИЕ СРЕДСТВ НА СЧЕТА КАРТ АС СБЕРКАРТ "ЗАРПЛАТНАЯ"						
	5							
	6	Всего листов _____						
	7	Лист № _____						
	8							
	9	Дата			<ДатаДок>			
	10	Наименование предприятия			<Константа_СокращенноеНаименованиеОрганизации>			
	11	Номер расчетного (текущего) счета предприятия			<Банк_НомерСчетаПредприятия>			
	12	Наименование подразделения Сбербанка-получателя			<Банк_ПолноеНаименование>			
	13	Дополнительная информация						
	14	Количество операций						
	15	Общая сумма			<Сумма>			
	16							
	17	№	Ф.И.О.		№ счета		Сумма	
Строка	18	рПП>	<Сотрудник_Наименование>		<НомерЗарплатногоСчета>		<Сумма>	
Подвал	19				ИТОГО:		<Сумма>	
	20	Руководитель Предприятия			</Руководитель/>			
	21	Главный бухгалтер Предприятия			</ГлБухгалтер/>			
	22							
	23							

Рис. 6.6. Макет печатной формы списка на зачисление средств на карточные счета

Шаг 4. Пишем процедуру выгрузки в файл (листинг 6.15). При выгрузке используется функция выбора каталога (листинг 6.16) и получения контактной информации (листинг 6.17).

Шаг 5. Подключаем внешнюю печатную форму к документу ПлатежноеПоручениеИсходящее и проверяем работу процедуры выгрузки.



Время на выполнение работы — 6 часов.

Листинг 6.14. Процедура печати списка на зачисление средств на карточные счета

```

Функция Печать () Экспорт
Докум = ССылкаНаОбъект;
ОтделениеБанка = Докум.Контрагент;
ДоговорСБанком = Докум.Контрагент.ОсновнойДоговорКонтрагента;
// Ищем руководителя организации
Отбор = Новый Структура;
Отбор.Вставить ("СтруктурнаяЕдиница", ССылкаНаОбъект.Организация);
Отбор.Вставить ("ОтветственноеЛицо",
Перечисления.ОтветственныеЛицаОрганизаций.Руководитель);

```

```
Руководитель = РегистрыСведений.ОтветственныеЛицаОрганизаций.  
ПолучитьПоследнее ( СсылкаНаОбъект.Дата,Отбор) .ФизическоеЛицо.Наименование;  
// Ищем главного бухгалтера  
Отбор = Новый Структура;  
Отбор.Вставить ("СтруктурнаяЕдиница", СсылкаНаОбъект.Организация);  
Отбор.Вставить ("ОтветственноеЛицо",  
Перечисления.ОтветственныеЛицаОрганизаций.ГлавныйБухгалтер);  
ГлБухгалтер = РегистрыСведений.ОтветственныеЛицаОрганизаций.  
ПолучитьПоследнее ( СсылкаНаОбъект.Дата,Отбор) .ФизическоеЛицо.Наименование;  
ТаблицаПечати = Новый ТабличныйДокумент;  
Макет = ЭтотОбъект.ПолучитьМакет ("СписокСумм");  
ОбластьШапки = Макет.ПолучитьОбласть ("Шапка");  
ОбластьШапки.Параметры.Банк_Договор = СсылкаНаОбъект.Контрагент.Основ-  
нойДоговорКонтрагента;  
ОбластьШапки.Параметры.ДатаДок = СсылкаНаОбъект.Дата;  
ОбластьШапки.Параметры.Константа_СокращенноеНаименованиеОрганизации =  
СсылкаНаОбъект.Организация.Наименование;  
ОбластьШапки.Параметры.Банк_НомерСчетаПредприятия =  
СсылкаНаОбъект.СчетОрганизации;  
ОбластьШапки.Параметры.Банк_ПолноеНаименование =  
СсылкаНаОбъект.Контрагент.НаименованиеПолное;  
ОбластьШапки.Параметры.Сумма =  
ЧислоПрописью (СсылкаНаОбъект.ПеречислениеЗарботнойПлаты.Итог (  
"СуммаКВывплате"), "L=ru_RU", "рубль, рубля, рублей, м, копейка, копейки,  
копеек, ж, 2");  
ТаблицаПечати.Вывести (ОбластьШапки);  
ТаблицаПечати.ТолькоПросмотр = Истина;  
НомерПП = 0;  
Итого=0;  
Для Каждого Стр Из СсылкаНаОбъект.ПеречислениеЗарботнойПлаты Цикл  
Ведомость = Стр.Ведомость;  
Для Каждого Сотрудник Из Ведомость.Зарплата Цикл  
НомерПП = НомерПП+1;  
Отбор = Новый Структура;  
Отбор.Вставить ("ФизЛицо", Сотрудник.Физлицо);  
Отбор.Вставить ("Организация", СсылкаНаОбъект.Организация);  
Отбор.Вставить ("Банк", СсылкаНаОбъект.Контрагент);  
Запись = РегистрыСведений.ЛицевыеСчетаРаботниковОрганизации.Получить (Отбор);  
НомерЗарплатногоСчета = Запись.НомерЛицевогоСчета;  
ОбластьСтроки = Макет.ПолучитьОбласть ("Строка");  
ОбластьСтроки.Параметры.НомерПП = НомерПП;
```

```

ОбластьСтроки.Параметры.Сотрудник_Наименование =
    Сотрудник.Физлицо.Наименование;
ОбластьСтроки.Параметры.НомерЗарплатногоСчета =
    НомерЗарплатногоСчета;
ОбластьСтроки.Параметры.Сумма = Формат (Сотрудник.Сумма,
    "цц=15; чдц=2; чрд=.; чрг=' '");
Итого = Итого + Сотрудник.Сумма;
ТаблицаПечати.Вывести (ОбластьСтроки);
КонецЦикла;
КонецЦикла;
ОбластьПодвала = Макет.ПолучитьОбласть ("Подвал");
ОбластьПодвала.Параметры.Сумма =
    Формат (Итого, "цц=15; чдц=2; чрд=.; чрг=' '");
ОбластьПодвала.Параметры.Руководитель = Руководитель;
ОбластьПодвала.Параметры.ГлБухгалтер = ГлБухгалтер;
ТаблицаПечати.Вывести (ОбластьПодвала);
ТаблицаПечати.ТолькоПросмотр = Истина;
ВыгрузкаВФайл (); // выгружаем сведения в файл
Возврат ТаблицаПечати;
КонецФункции

```

Листинг 6.15. Процедура выгрузки в файл

```

Процедура ВыгрузкаВФайл ()
Каталог = ВыборКаталога ();
ВидКонтактнойИнформации_НомерДоговора = Справочники.
ВидыКонтактнойИнформации.НайтиПоНаименованию ("НомерДоговора");
ВидКонтактнойИнформации_ОтделениеБанка = Справочники.
ВидыКонтактнойИнформации.НайтиПоНаименованию ("ОтделениеБанка");
ВидКонтактнойИнформации_НомерФилиала = Справочники.
ВидыКонтактнойИнформации.НайтиПоНаименованию ("НомерФилиала");
Пробелы = " ";
НомерДоговора = ПолучитьКонтактнуюИнформацию (СсылкаНаОбъект.Контрагент,
    ВидКонтактнойИнформации_НомерДоговора);
ОтделениеБанка = ПолучитьКонтактнуюИнформацию (СсылкаНаОбъект.Контрагент,
    ВидКонтактнойИнформации_ОтделениеБанка);
НомерФилиала = ПолучитьКонтактнуюИнформацию (СсылкаНаОбъект.Контрагент,
    ВидКонтактнойИнформации_НомерФилиала);
Сообщить ("Выгрузка данных: "+НомерДоговора+" "+ОтделениеБанка+
    " "+НомерФилиала);

```

```

Текст= Новый ТекстовыйДокумент;
НомДоговора = Лев (НомерДоговора+Пробелы, 4) ;
НомОСБИмяФайла = Лев (ОтделениеБанка+Пробелы, 4) ;
НомОСБ = Лев (ОтделениеБанка+Пробелы, 4) ;
НомФилОСБ = Прав ("00000"+НомерФилиала, 5) ;
Службное = Лев (Пробелы, 19) ;
КодВалюты = "810"; //рубли
Службное2 = Лев (Пробелы, 29) ;
ДеньПеречисления = Формат (День (СсылкаНаОбъект.Дата), "чц=2; ЧВН=") ;
НомерПП = 0;
ИтогСумма=0;
// Перебираем платежные ведомости
Для Каждого Стр Из СсылкаНаОбъект.ПеречислениеЗарботнойПлаты Цикл
    Ведомость = Стр.Ведомость;
    // Перебираем сотрудников из ведомости
    Для Каждого Сотрудник Из Ведомость.Зарплата Цикл
        Отбор = Новый Структура;
        Отбор.Вставить ("ФизЛицо", Сотрудник.ФизЛицо) ;
        Отбор.Вставить ("Организация", СсылкаНаОбъект.Организация) ;
        Отбор.Вставить ("Банк", СсылкаНаОбъект.Контрагент) ;
        Запись = РегистрыСведений.ЛицевыеСчетаРаботниковОрганизации.
            Получить (Отбор) ;
        НомерЗарплатногоСчета = Запись.НомерЛицевогоСчета;
        Если (СокрЛП (НомерЗарплатногоСчета)= "") Тогда
            Сообщить ("У сотрудника "+Сотрудник.ФизЛицо.Наименование+
                " не задан номер зарплатного счета!");
            Продолжить;
        КонецЕсли;
        Состояние (НомерПП) ;
        НомерПП = НомерПП+1;
        Отбор = Новый Структура;
        Отбор.Вставить ("ФизЛицо", Сотрудник.ФизЛицо) ;
        Запись = РегистрыСведений.ФИОФизЛиц.ПолучитьПоследнее (
            СсылкаНаОбъект.Дата, Отбор) ;
        Фамилия = Запись.Фамилия;
        Имя = Запись.Имя;
        Отчество = Запись.Отчество;
        СуммаОперации = Формат (Сотрудник.Сумма,
            "чц=15; чдц=2; чрд=.; чг=0");

```

```
Строка = Лев (" "+НомерПП+Пробелы, 10) +
    НомОСБ+
    НомФилОСБ+
    Службное+
    Лев (Врег (Фамилия) +Пробелы, 30) +
    Лев (Врег (Имя) +Пробелы, 30) +
    Лев (Врег (Отчество) +Пробелы, 30) +
    Прав (Пробелы+СокрЛП (НомерЗарплатногоСчета) , 25) +
    Лев (ТипСчета+Пробелы, 8) +
    Прав (Пробелы+СуммаОперации, 15) +
    КодВалюты+
    Службное2;
Сообщить (Строка) ;
Текст.ДобавитьСтроку (Строка) ;
ИтогСумма = ИтогСумма + Сотрудник.Сумма;
КонецЦикла;
КонецЦикла;
// Записываем текст в файл
ИмяФайла = СокрЛП (Каталог) + "\ "+НомОСБИмяФайла+НомДоговора+".s"+
    ДеньПеречисления;
ФС = Новый Файл (ИмяФайла) ;
Если ФС.Существует () Тогда
Если Вопрос ("Файл под именем "+ИмяФайла+" существует, заменить?",
    РежимДиалогаВопрос.ДаНет) =КодВозвратаДиалога.Да Тогда
    Текст.Записать (ИмяФайла, КодировкаТекста.ОЕМ) ;
КонецЕсли;
Иначе
    Текст.Записать (ИмяФайла, КодировкаТекста.ОЕМ) ;
КонецЕсли;
Сообщить ("Файл выгружен: "+ИмяФайла) ;

// второй файл – сводная информация
Текст= Новый ТекстовыйДокумент;
НазваниеОрганизации = Лев (Врег (СсылкаНаОбъект.Организация.Наименование)
+Пробелы, 30) ;
Текст.ДобавитьСтроку (
    Лев (Врег (СсылкаНаОбъект.Организация.Наименование) +Пробелы, 30) +
    НомДоговора+
    Прав (Пробелы+НомерПП, 7) +
```

```

Прав (Пробелы+Формат (ИтогСумма, "ЧЦ=15; ЧДЦ=2; ЧРД=. ; ЧГ=0"), 15));
ИмяФайлаК = СокрЛП (Каталог) + "\" + НомОСВИмяФайла + НомДоговора + ".
к" + ДеньПеречисления;
ФС = Новый Файл (ИмяФайлаК);
Если ФС.Существует () Тогда
    Если Вопрос ("Файл под именем "+ИмяФайлаК+" существует, заменить?",
        РежимДиалогаВопрос.ДаНет) = КодВозвратаДиалога.Да Тогда
        Текст.Записать (ИмяФайлаК, КодировкаТекста.ОЕМ);
    КонецЕсли;
Иначе
    Текст.Записать (ИмяФайлаК, КодировкаТекста.ОЕМ);
КонецЕсли;
Сообщить ("Файл выгружен: "+ИмяФайлаК);
Сообщить ("Сумма итого: "+ИтогСумма);
Предупреждение ("Файлы успешно сформированы.
    |Имена файлов: "+ИмяФайла+", "+ИмяФайлаК);
КонецПроцедуры

```

Листинг 6.16. Функция выбора каталога

```

Функция ВыборКаталога ()
    Диалог = Новый ДиалогВыбораФайла (РежимДиалогаВыбораФайла.ВыборКаталога);
    Диалог.Заголовок = "Выберите каталог";
    Если Диалог.Выбрать () Тогда
        Возврат Диалог.Каталог;
    КонецЕсли;
    Возврат "";
КонецФункции

```

Листинг 6.17. Функция поиска контактной информации

```

Функция ПолучитьКонтактнуюИнформацию (Контрагент, ВидКонтактнойИнформации);
    Отбор = Новый Структура;
    Отбор.Вставить ("Объект", Контрагент);
    Отбор.Вставить ("Тип", Перечисления.ТипыКонтактнойИнформации.Другое);
    Отбор.Вставить ("Вид", ВидКонтактнойИнформации);
    Запись = РегистрыСведений.КонтактнаяИнформация.Получить (Отбор);
    Возврат Запись.Представление;
КонецФункции

```

