

УДК 004.415

РАЗРАБОТКА ВЫХОДНЫХ ДОКУМЕНТОВ ПРОГРАММНОГО КОМПЛЕКСА «ЗАРПЛАТА»

Черепков А.В.¹, Зинченко В.А.², Зеленева И.Я.³

1,3 - Донецкий национальный технический университет

2 - ПАО «ИПС ПЛЮС», отдел ПО

cherepkov0905@mail.ru

Аннотация

Черепков А.В., Зинченко В.А., Зеленева И.Я. Разработка выходных документов программного комплекса «Зарплата». Проанализирован ряд важных вопросов, которые касаются автоматизации работы в отделе бухгалтерии и отделе кадров на предприятиях с хорошо развитой ИТ-структурой. В статье рассмотрены преимущества и области применения современного универсального фреймворка iReport. Предложена система классов и шаблонов для создания выходных документов для упрощения работы бухгалтеров и сотрудников отдела кадров.

Введение

В современном мире существует очень много организаций и производственных подразделений с большим количеством персонала. В каждом из них необходимо предоставлять сотрудникам всевозможные справки и документы, а также рассчитывать зарплату и все «родственные» расчеты. Подобные задачи стоят даже перед самыми мелкими организациями. И в таких организациях не всегда есть возможность обучить сотрудников работе с профессиональными программными продуктами. Существующие программные комплексы типа «1С. Зарплата и Кадры» не предлагают собственные выходные документы, и для создания отчетов необходимо иметь навыки разработки в данной программе с собственным внутренним набором функций, что создает трудности для сотрудников, не компетентных в программировании. Таким образом, актуальной является разработка удобного для обычного пользователя программного комплекса с большим набором распространенных выходных документов.

Применение iReport для создания шаблона отчета

Существует несколько способов добавления отчетов в приложение. Можно разработать веб-страницу, которая будет хорошо выглядеть на экране, но будет неудобна для печати. Для создания отчетов PDF-средствами необходимо писать большое количество кода, который в дальнейшем трудно поддерживать. Java предоставляет широкий API для печати, но необходимо выполнить еще много работы, чтобы написать специальный код для каждого формата. Оптимальным является использование библиотеки отчетов. JasperReports наиболее популярная в мире библиотека Java отчетов с открытым исходным кодом. А iReport – это визуальный дизайнер отчетов для JasperReports. Библиотека представляет собой генератор отчетов, которые могут быть интегрированы в открытые или коммерческие приложения. Библиотека позволяет отображать отчеты на экране или экспортировать их в определенный конечный формат. Поддерживаются такие форматы как PDF, OpenOffice, DOCX и многие другие. Кроме того, вы можете передавать результат через веб-приложение или отправить итоговый документ непосредственно на принтер.

При создании отчета с помощью iReport создается JRXML-файл, который представляет собой XML-документ. Данный документ содержит определение отчета. Но

шаблон полностью разработан визуально, что позволяет игнорировать реальную структуру JRXML-файла. Перед выполнением отчета, JRXML-файл должен быть скомпилирован в двоичный объект, называемый файлом Jasper. Этот файл будет создан из соображений производительности. Jasper файлы – это то, что необходимо прикреплять к приложению для запуска отчетов.

Источником данных для шаблона JasperReports могут быть запросы SQL, файлы XML, CSV-файл, HQL(Hibernate Query Language) запросы, коллекция JavaBeans, и т.д. Также есть возможность написать свой собственный источник данных. Так как программный комплекс «Зарплата» написан на Java и использует сервер баз данных MySQL для хранения всех данных источником для шаблона является HQL запрос.

Создание шаблонов JasperReports и классов Java для получения выходных документов

Для того, чтобы сделать выходной документ, необходимо выполнить следующие действия:

- Написать код файла главных данных, которые будут выводиться в шаблон отчета.
- Установить iReport.
- Создать шаблон выходного документа в iReport.
- Скомпилировать JRXML-файл.
- Написать сущность определенного отчета.
- Создать код формы для редактирования входных данных.
- Создать визуальную форму входных данных.
- Создать главное ядро отчета.
- Написать слушатель визуальной формы.

Файл главных данных называется <Название отчета> DataBean.java. В нем определяем объекты определенного типа, которые используются в конкретном выходном документе. Для каждого объекта написаны метод установщик(setter) значений объекта и метод получатель(getter).

Программный продукт iReport 4.5.1 предоставляется бесплатно и его можно найти на различных интернет-ресурсах.

Для создания шаблона был интегрирован в пакет приложения новый лист. С параметрами листа можно работать как и во всех известных офисных приложениях для работы с документацией. В новом листе создаем поля с такими же названиями и типами, как и объекты класса DataBean.java. Эти поля перетаскиваем на лист, после чего iReport добавляет к ним специальные префиксы(Рис. 1).

Также были созданы параметры, в которые можно напрямую из приложения отправлять данные, такие как расчетный месяц, выбранный из списка департамент или вид удержаний. В итоге были разработаны шаблоны для отчетов, которые в дальнейшем будут принимать определенные данные из приложения.

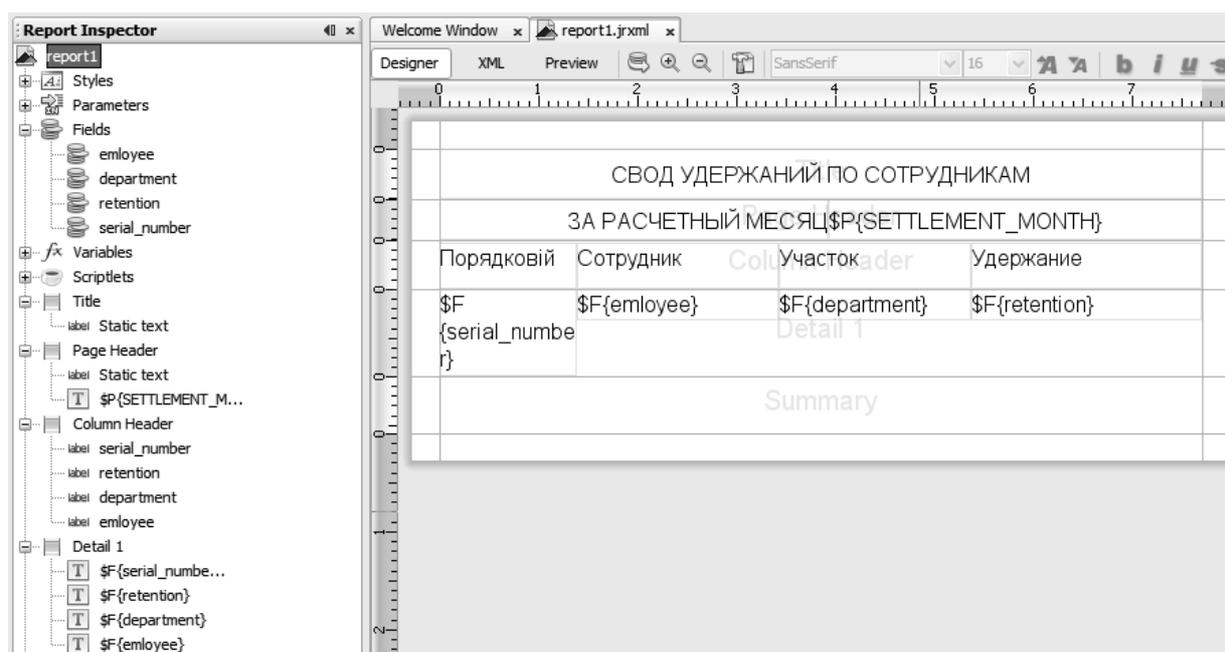


Рисунок 1 – Шаблон выходного документа

Далее файлы шаблонов JRXML скомпилированы средствами Java и подключенной библиотекой `net.sf.jasperreports.engine.Jasper` в файлы `Jasper`. Это выполняется в каждом тесте печати отчета следующей командой:

```
JasperCompileManager
    .compileReportToFile
        ("src/reports/report_vault_retention/report_vault_retention.jrxml");
```

В течении разработки создаются наборы полей(сущность) выходных документов, соответствующие полям базы данных. В этих файлах описаны объекты, которые будут отображаться для выбора входных данных, а также методы работы с ними.

В форме редактирования осуществляется связь с визуальной формой. Также происходит инициализация компонент формы. В этом файле устанавливаются обработчики клавиш и связываются элементы управления.

Визуальная форма была создана в визуальном редакторе среды разработки NetBeans IDE.

Ядро заполнения в каждом пакете отчета называется `<Название отчета> fillCore.java`. В этом файле были написаны запросы, которые осуществляют выборку из базы данных необходимых полей. После этого мы заполняем поля шаблона с помощью `setter` и `getter` файла `DataBean` следующим образом:

```
for(int i=0; i<rates_payment_list.size(); i++){
    Float sum_retent = (((BigDecimal)rates_payment_list.get(i)[1]).floatValue());
    Long pay_type_num = (((BigInteger)rates_payment_list.get(i)[2]).longValue());
    String pay_type_title = ((String)rates_payment_list.get(i)[3]);

    o = new ReportVaultRetentionByEnterpriseDataBean();

    o.setCorresponding_accounts("0");
    o.setType_retention_num(pay_type_num);
    o.setType_retention_title(pay_type_title);
```

```
o.setSum_retention(sum_retent);
rows_sum_list.add(o);
}
```

Обработчик кнопок управления визуальной формы непосредственно обрабатывает действия пользователя, а также заполняет отчет и подает его на просмотр. Это предложено сделать с помощью следующих команд:

```
JasperPrint jasperPrint =
    JasperFillManager.fillReport(
        "src/reports/report_vault_retention/report_vault_retention.jasper",
        parameters,
        jrb
    );
// подаем отчет на просмотр
JasperViewerExtended jrview = new JasperViewerExtended(jasperPrint);
```

В рамках возможностей рассмотренного инструментария авторами выполнена разработка соответствующей системы классов Java и шаблонов iReport. Разработанные выходные документы упрощают работу бухгалтеров и сотрудников отдела кадров. Выходные документы доступные в приложении «Зарплата» работают на основании справочников и информации по кадрам. То есть сотрудникам необходимо предоставить программному комплексу только лишь личные данные персонала.

Заключение

Результатом работы предложенного программного решения является целый ряд выходных документов, разработанных на кроссплатформенном языке Java с использованием бесплатного фреймворка построения отчетов iReport. Разработка учитывает существующее законодательство при расчете авансов и выплат сотруднику, а также дает возможность выдавать отчеты на основании введенных данных по каждому сотруднику, таких как удержание алиментов, удержание кредитов, выплаты через банк, своды итоговых сумм и прочие исполнительные листки и ведомости. Также созданы выходные документы, которые предоставляют отчеты по участкам, местам работы и по предприятию в целом, что позволяет получить любую информацию необходимую как сотруднику, так и отделу бухгалтерии в короткие сроки и с минимальными трудозатратами.

Список литературы

1. Java 2: The Complete Reference, П. Ноутон, Г. Шилдт., Издательство: БХВ-Петербург, 2008. – 996 с.
2. Техническая документация iReport // Режим доступа: <http://jasperforge.org/projects/ireport>
3. Техническая документация Hibernate // Режим доступа: <http://docs.jboss.org/hibernate/orm/4.1/devguide/en-US/html/>
4. Бухгалтерский учет. Учебник, Н. П. Кондраков, Издательство: Инфра-М, 2007. – 592с.