

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
( Н И У « Б е л Г У » )

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ

**ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ  
МЕЖДУ УПРАВЛЕНИЕМ ПЕНСИОННОГО ФОНДА РФ В Г.  
БЕЛГОРОД И ОТДЕЛЕНИЕМ ПО БЕЛГОРОДСКОЙ ОБЛАСТИ**

Выпускная квалификационная работа  
обучающейся по направлению подготовки 38.03.05 «Бизнес-информатика»  
очной формы обучения, группы 07001423  
Бородкиной Елены Владимировны

Научный руководитель:  
д-р физ.-мат. наук Ломазов В.А.

БЕЛГОРОД 2018

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Характеристика деятельности Отделения Пенсионного Фонда Российской Федерации по Белгородской области.....	5
1.1 Исследование деятельности Пенсионного Фонда Российской Федерации по Белгородской области .....	5
1.2 Анализ взаимодействия подразделений ОПФР и ФНС по вопросу принятия решения о возврате (взыскании) излишне уплаченных средств....	14
2 Проектирование программного комплекса «Возвраты».....	19
2.1 Проектирование бизнес-процессов, связанных с возвратом (взысканием) излишне уплаченных средств .....	19
2.2 Техническое задание.....	25
2.2.1 Назначение разработки .....	25
2.2.2 Требования к функциональным характеристикам .....	26
2.2.3 Требования к надёжности .....	26
3 Реализация программного комплекса «Возвраты».....	28
3.1 Процесс программной реализации комплекса «Возвраты» .....	28
3.1.1 Создание документа и вывод его на печать .....	28
3.1.2 Авторизация пользователей.....	33
3.2 Разработка БД.....	37
3.3 Расчет затрат на реализацию программного комплекса «Возвраты» .....	43
ЗАКЛЮЧЕНИЕ .....	53
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	55
ПРИЛОЖЕНИЕ А .....	59
ПРИЛОЖЕНИЕ Б.....	122

## **ВВЕДЕНИЕ**

Актуальность выпускной квалификационной работы связана с необходимостью совершенствования информационного взаимодействия региональных Отделений Пенсионного Фонда Российской Федерации, в том числе применительно к Белгородской области.

Объектом исследования выпускной квалификационной работы является документооборот между региональными подразделениями Пенсионного Фонда Российской Федерации.

Предметом исследования является информационное обеспечение документооборота между региональными подразделениями Пенсионного Фонда Российской Федерации.

Целью выпускной квалификационной работы является разработка информационного обеспечения документооборота между Управлением Пенсионного Фонда Российской Федерации и Отделением Пенсионного Фонда Российской Федерации в городе Белгород по осуществлению возвратов сумм излишне уплаченных (взысканных) денежных средств.

Для достижения поставленной цели в выпускной квалификационной работе последовательно решались следующие задачи:

1) Провести анализ взаимодействия подразделений Отделения Пенсионного Фонда Российской Федерации и Федеральной Налоговой Службой;

2) Определить проблемы, возникающие в рамках документооборота по возврату сумм излишне уплаченных (взысканных) денежных средств;

3) Выполнить проектирование программного комплекса «Возвраты» по обеспечению информационной поддержки документооборота по возврату плательщиком страховых взносов излишне уплаченных (взысканных) денежных средств;

4) Произвести расчет экономических затрат на разработку программного комплекса «Возвраты».

Разработка программного комплекса «Возвраты» предназначена для формирования запроса в ФНС о предоставлении сведений, а также ведения статистики возвратов (взысканий) и других аналитических материалов.

При выполнении выпускной квалификационной работы использовались общелогические методы и приёмы исследования, а также метод системного анализа, инструментарий проектирования и программной реализации информационной системы.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка используемой литературы и приложений.

Во введении обоснована актуальность темы, определены объект и предмет исследования, сформулированы цель и задачи.

В первой главе произведен анализ регионального Отделения Пенсионного Фонда Российской Федерации, сформулирована задача выпускной квалификационной работы и обоснована необходимость разработки программного комплекса «Возвраты» информационной поддержки документооборота между Отделением Пенсионного Фонда Российской Федерации и Управлением Пенсионного Фонда Российской Федерации по осуществлению возвратов сумм излишне уплаченных (взысканных) денежных средств.

Во второй главе выполнен проект программного комплекса «Возвраты» и сформулировано техническое задание на разработку.

Третья глава посвящена программной реализации исследовательского прототипа и обоснованию затрат на разработку программного комплекса «Возвраты».

В заключении представлены итоги проделанной работы и сформулированы окончательные выводы по рассматриваемой теме.

# **1 Характеристика деятельности Отделения Пенсионного Фонда Российской Федерации по Белгородской области**

## **1.1 Исследование деятельности Пенсионного Фонда Российской Федерации по Белгородской области**

Пенсионный фонд Российской Федерации (ПФР) - один из наиболее значимых социальных институтов страны. Это крупнейшая федеральная система оказания государственных услуг в области социального обеспечения в России. Устройство пенсионной системы во многом определяет уровень согласия в обществе, устойчивость и динамику экономического и социального развития.

Структура Пенсионного фонда России содержит 84 Отделения Пенсионного фонда в субъектах РФ, а также 2460 клиентских служб в территориальных органах ПФР. В системе ПФР трудится более 100 000 специалистов.

ПФР осуществляет ряд социально значимых функций, в том числе:

- установление и выплата страховых пенсий (по старости, по инвалидности, по случаю потери кормильца), накопительной пенсии, пенсии по государственному пенсионному обеспечению, пенсии военнослужащих и их семей, социальных пенсий, пенсии госслужащих;
- выдача государственных сертификатов на материнский (семейный) капитал;
- назначение и реализация социальных выплат отдельным категориям граждан: ветеранам, инвалидам, Героям Российской Федерации и др.;
- назначение и реализация федеральной социальной доплаты к пенсии до уровня прожиточного минимума пенсионера в регионе;
- ведение системы персонифицированного учета прав участников системы обязательного пенсионного страхования;
- формирование, инвестирование и выплата средств пенсионных накоплений;

— адресная помощь пенсионерам и софинансирование социальных программ субъектов РФ;

— реализация Программы государственного софинансирования пенсии;

— реализация международных соглашений.

ПФР и подчиненные ему органы составляют единую централизованную многоуровневую систему, управляющую средствами обязательного пенсионного страхования в России:

1) федеральный уровень - ПФР (правление, исполнительная дирекция);

2) федеральные округа Российской Федерации - управления ПФР в федеральных округах (структурные подразделения ПФР);

3) уровень субъектов Российской Федерации - территориальные отделения ПФР (в соответствующих регионах);

4) местный - городской (районный) уровень - управления (отделения) ПФР в городах (районах).

Нижестоящие органы подотчетны вышестоящим органам.

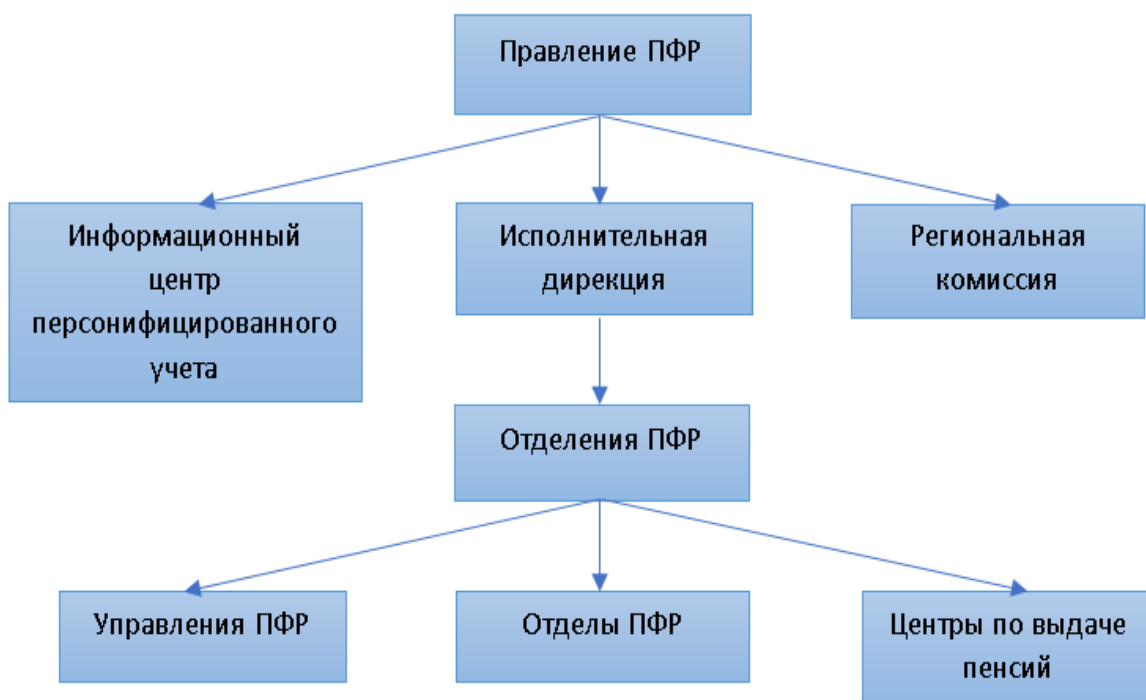


Рисунок 1.1 – Структура Пенсионного Фонда РФ

Управление всей пенсионной системой на федеральном уровне осуществляется правлением ПФР и его исполнительным органом - исполнительной дирекцией. В субъектах Российской Федерации действуют региональные отделения ПФР, являющиеся юридически самостоятельными для выполнения возложенных на них функций по назначению и выплате пенсий. Местный уровень пенсионной системы составляют пенсионные управления (отделения, отделы), расположенные в районах и городах. Они в основном выполняют функции по назначению и выплате пенсий.

В своей деятельности ПФР и его нижестоящие органы взаимодействуют со многими министерствами, ведомствами, иными организациями и их структурными подразделениями по вертикали. К их числу относятся: Федеральная Налоговая Служба; Минздравсоцразвития России и Минфин России; Минэкономразвития России; Россвязь; ФСС РФ и фонды обязательного медицинского страхования; Росстат; Минюст России; Банк России; Сбербанк России; Внешэкономбанк; управляющие пенсионные компании; Федерация независимых профсоюзов России и подведомственные всем им нижестоящие

органы. Именно благодаря взаимодействию ПФР с органами законодательной, исполнительной власти, органами местного самоуправления и вышеперечисленными организациями ПФР может осуществлять свою деятельность.

В субъектах Российской Федерации ПФР имеет свои региональные органы - территориальные отделения ПФР. Они действуют на основе положения о территориальных органах ПФР, которое утверждается правлением ПФР, и создаются для осуществления государственного управления средствами обязательного пенсионного страхования и организации пенсионного обеспечения.

В число основных обязанностей территориального отделения ПФР входят: исполнение бюджета отделения в соответствии с плановыми показателями, утвержденными правлением ПФР; организация работы по учету средств, поступающих по обязательному пенсионному страхованию, а также контроль за целевым использованием этих средств; финансирование расходов на выплату трудовых пенсий, пенсий по государственному пенсионному обеспечению и социальных пособий на погребение умерших пенсионеров, не работавших на день смерти; организация и ведение государственного банка данных по всем категориям страхователей, в том числе физических лиц, добровольно вступивших в правоотношения по обязательному пенсионному страхованию; организация работы по назначению (перерасчету) и доставке трудовых пенсий, пенсий по государственному пенсионному обеспечению, выплате трудовых пенсий, пенсий по государственному пенсионному обеспечению и социальных пособий на погребение умерших пенсионеров, не работавших на день смерти; организация работы по ведению базы данных по пенсионерам; организация и ведение индивидуального (персонифицированного) учета сведений обо всех категориях застрахованных лиц в соответствии с законодательством РФ о таком индивидуальном учете в системе пенсионного страхования; разрешение споров по вопросам уплаты взносов, назначения (перерасчета) и организации доставки трудовых пенсий,



пенсий по государственному пенсионному обеспечению, выплаты пенсий, социальных пособий на погребение умерших пенсионеров, не работавших на день смерти, удержания из пенсий, взыскания излишне выплаченных сумм пенсий и т.д.; ведение учета страховых взносов физических лиц, добровольно вступивших в правоотношения по обязательному пенсионному страхованию; решение других вопросов, отнесенных к компетенции отделения в соответствии с действующим законодательством РФ.

Обязательное пенсионное страхование на местах осуществляют районные (городские) управления, отделы ПФР (в дальнейшем - местные пенсионные органы). В отдельных населенных пунктах, например, в крупных поселках, городах, могут учреждаться должности уполномоченных ПФР, которые подчиняются местным органам ПФР. Местные пенсионные органы являются правопреемниками органов социальной защиты населения - в части назначения, перерасчета и выплаты трудовых пенсий и пенсий по государственному пенсионному обеспечению, а также социальных пособий на погребение умерших пенсионеров, не работавших на день смерти.

Местные пенсионные органы непосредственно подчиняются территориальному отделению ПФР. Содержание их работы разнообразно. Оно включает в себя самые различные по своему характеру виды работ, а именно: прием документов для назначения пенсий; назначение, перерасчеты, выплата пенсий; доставка пенсий; финансовое обеспечение пенсий; взаимодействие с иными государственными органами и учреждениями, организациями всех форм собственности; прием населения по пенсионным вопросам; работа с общественными организациями; ведение индивидуального (персонифицированного) учета; проверка соблюдения законодательства о пенсиях организациями всех форм собственности; организация и проведение учебно-методической работы по вопросам пенсионного обеспечения и т.д.

Для обеспечения своевременного финансирования выплат пенсий местные пенсионные органы (в отдельных регионах - центры по назначению и выплате пенсий) составляют и представляют в территориальные отделения

ПФР в установленные сроки заявки на право расходования средств для осуществления в очередном плановом периоде (год, квартал, месяц) выплат, финансируемых бюджетом ПФР согласно законодательству.

В свою очередь территориальные отделения фонда направляют заявки в его исполнительную дирекцию на выделение им необходимого лимита финансовых средств.

Территориальные органы ПФР действуют на основе положения о территориальных органах ПФР, которое утверждается правлением ПФР, и создаются для осуществления государственного управления средствами обязательного пенсионного страхования и организации пенсионного обеспечения.

Отделение Пенсионного Фонда Российской Федерации по Белгородской области образовано в соответствии с Постановлением Правительства ПФР от 22 марта 1991 года №59.

Во главе Отделения Пенсионного Фонда Российской Федерации стоит управляющий отделением, в подчинении у него находятся 4 заместителя управляющего, а у каждого из этих заместителей в подчинении находятся один либо несколько отделов. Непосредственно управляющему подчиняются помощник управляющего и начальники шести отделений: отдела по защите информации, отдела кадров, контрольно–ревизионного отдела, юридического отдела, отдела казначейства и группы по взаимодействию с СМИ. Применяется линейно-функциональная структура организации.

На рисунке 1.2 представлена организационная структура ОПФР по Белгородской области.

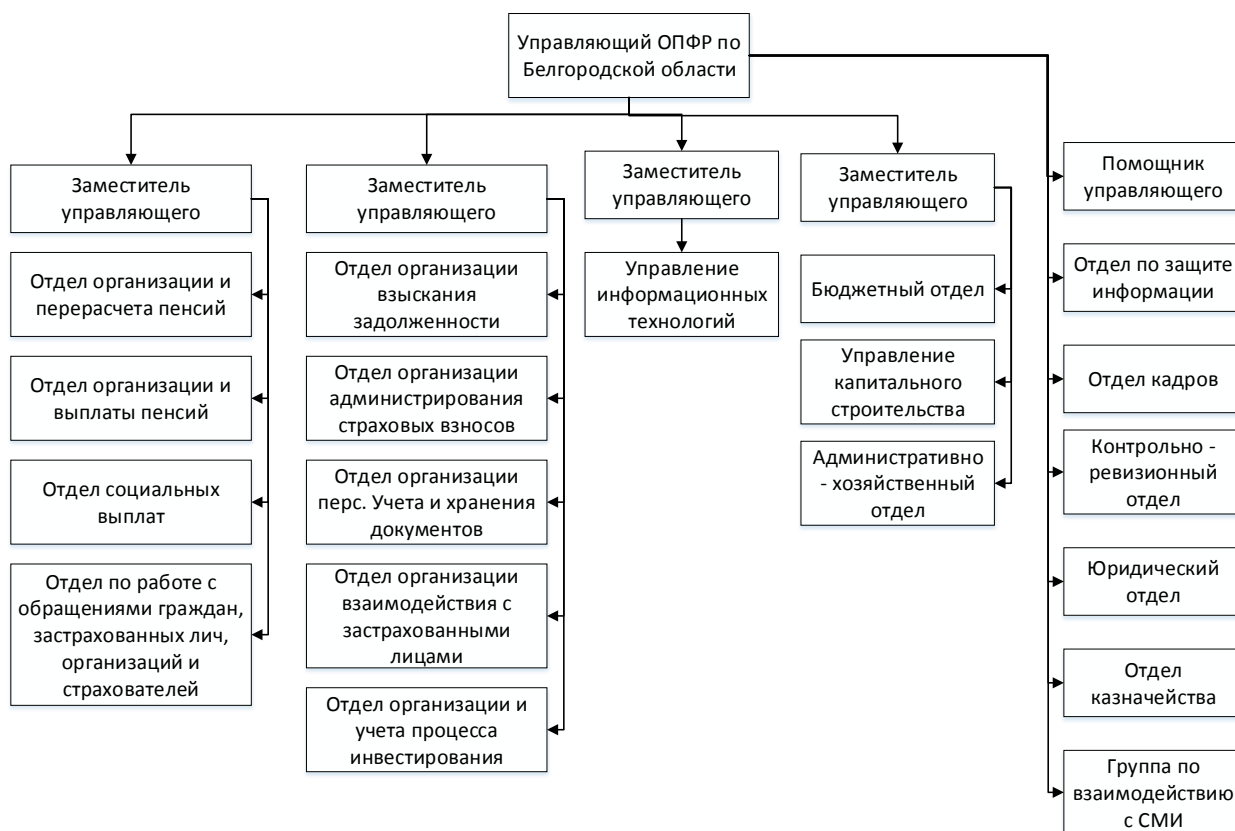


Рисунок 1.2 – Организационная структура ОПФР по Белгородской области

В структуре Отделения ПФР 22 Управления в городах и районах области, которые ведут персональные пенсионные счета более 1,6 миллиона человек, взаимодействуют с более чем 95 тысячами страхователей, из них более 48 тысяч - индивидуальные предприниматели и самозанятое население. Получателями пенсии в Белгородской области являются более 514 тысяч человек.

По данным, предоставленным на сайте пенсионного фонда на 1 апреля 2018 года:

— Средний размер пенсии в Белгородской области по состоянию на 1 апреля 2018 года составляет 13013 рублей 74 копейки;

— Средний размер социальной пенсии по состоянию на 1 апреля 2018 года составляет 8110 рублей 79 копеек;

— Средний размер страховой пенсии по старости по состоянию на 1 апреля 2018 года составляет 13825 рублей 26 копеек.

Сегодня Отделение Пенсионного фонда РФ по Белгородской области, в структуру которого входит Государственное учреждение Управление Пенсионного фонда РФ по г. Белгород, – многофункциональная система, непрерывно работающая над решением ряда социально значимых задач, затрагивающих интересы самых разных категорий населения:

- исполнение бюджета отделения в соответствии с плановыми показателями, утвержденными правлением ПФР;

- организация работы по учету средств, поступающих по обязательному пенсионному страхованию, а также контроль за целевым использованием этих средств;

- финансирование расходов на выплату трудовых пенсий, пенсий по государственному пенсионному обеспечению и социальных пособий;

- назначение и выплаты пенсий. Среди них трудовые пенсии (по старости, по инвалидности, по случаю потери кормильца), пенсии по государственному пенсионному обеспечению, пенсии военнослужащих и их семей, социальные пенсии, пенсии госслужащих;

- назначение и реализации социальных выплат отдельным категориям граждан: ветеранам, инвалидам, инвалидам вследствие военной травмы, Героям Советского Союза, Героям Российской Федерации и т.д.

- персонифицированный учет участников системы обязательного пенсионного страхования.

- взаимодействие с работодателями - плательщиками страховых пенсионных взносов;

- выдача сертификатов на получение материнского (семейного) капитала;

- управление средствами пенсионной системы, в том числе накопительной частью трудовой пенсии, которое осуществляется через государственную управляющую компанию и частные управляющие компании;

- реализация Программы государственного софинансирования пенсии.



## **1.2 Анализ взаимодействия подразделений ОПФР и ФНС по вопросу принятия решения о возврате (взыскании) излишне уплаченных средств**

22 февраля 2011 председатель Правления Пенсионного Фонда Российской Федерации Антон Дроздов и руководитель Федеральной Налоговой Службы России по г. Белгород Михаил Мишустин подписали соглашение о взаимодействии.

Соглашение затрагивает важнейшие вопросы взаимодействия двух ведомств, в том числе предусматривает широкий спектр информации, которой Федеральной Налоговой Службы России по г. Белгород и Пенсионного Фонда Российской Федерации будут обмениваться на регулярной основе в рамках администрирования налогов и страховых взносов.

Для повышения эффективности администрирования налогов и страховых взносов Федеральная Налоговая Служба России по г. Белгород по Соглашению будет предоставлять Пенсионному Фонду Российской Федерации на федеральном уровне данные статистической налоговой отчетности о поступлении налогов, сборов и иных обязательных платежей, задолженности по ним – по показателям, относящимся к деятельности Пенсионного Фонда Российской Федерации; информацию о налоговых агентах (организациях, индивидуальных предпринимателях). На региональном уровне налоговые органы предоставляют органам Пенсионного Фонда Российской Федерации акты камеральных проверок, выписки из актов выездных налоговых проверок по уплате страховых взносов.

Органы Пенсионного Фонда Российской Федерации со своей стороны обеспечивают передачу информации налоговым органам по решениям о привлечении к ответственности или об отказе от привлечения страхователей к ответственности за нарушение законодательства об обязательном пенсионном страховании по результатам контрольной работы налоговых органов и сведения о взыскании недоимки на основании переданной налоговыми органами информации, а также предоставляют сведения о начисленных и уплаченных

страховых взносах и другую информацию, необходимую для администрирования налогов.

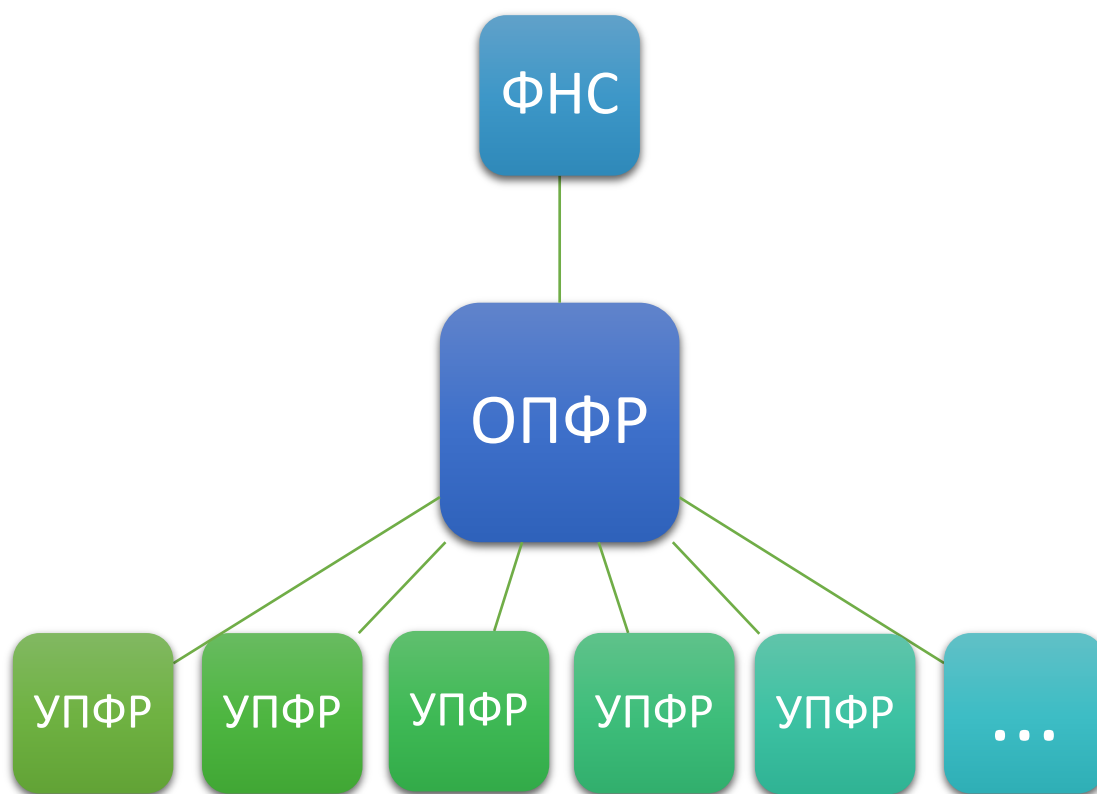


Рисунок 1.3 – Схема взаимодействия УПФР, ОПФР и ФНС

Взаимодействие подразделений ОПФР и ФНС по вопросу принятия решения о возврате (взыскании) излишне уплаченных средств, образовавшихся за расчетные (отчетные) периоды, истекшие до 01.01.2017, по причине поступления денежных средств после 01.01.2017 на коды бюджетной классификации с главой «182» на данный момент осуществляется следующим образом:

При поступлении в Управление Пенсионного Фонда Российской Федерации заявления плательщика страховых взносов о возврате излишне уплаченных (взысканных) денежных средств, образовавшихся за расчетные (отчетные) периоды, истекшие до 01.01.2017, по причине поступления денежных средств после 01.01.2017 на коды бюджетной классификации с главой «182», Управление Пенсионного Фонда Российской Федерации не позднее 2 рабочих дней после получения указанного заявления направляет в

Отделение Пенсионного Фонда Российской Федерации по Белгородской области на электронном носителе запрос о представлении сведений о принятом налоговыми органами по состоянию на текущую дату сальдо расчетов по страховым взносам, пеням, штрафам, сведений, переданных в рамках регулярного обмена, а также о суммах излишне уплаченных (взысканных) страховых взносов, образовавшихся за расчетные (отчетные) периоды, истекшие до 01.01.2017, соответствующих пеней и штрафов, подлежащих возврату, с указанием кода бюджетной классификации, на которые поступили вышеуказанные денежные средства.

В свою очередь Отделение Пенсионного Фонда Российской Федерации по Белгородской при получении запроса в срок не позднее 2 рабочих дней направляет этот запрос на электронном носителе, в Федеральную Налоговую Службу России по г. Белгород, внося номер запроса и даты направления.

Федеральная Налоговая Служба России по г. Белгород при получении запроса в срок не позднее 3 рабочих дней направляет в соответствующее Отделение Пенсионного Фонда Российской Федерации запрашиваемые сведения за подписью руководителя Управления на электронном носителе.

Отделение Пенсионного Фонда Российской Федерации в течение 2 рабочих дней после получения указанных сведений от Федеральной Налоговой Службы России по г. Белгород отправляет их на электронном носителе обратно в Управление Пенсионного Фонда Российской Федерации России по г. Белгород в рамках ответа на запрос.

Управление Пенсионного Фонда Российской Федерации в срок не позднее 2 рабочих дней с момента получения осуществляет сверку информации о состоянии расчетов с бюджетом по страховым взносам, содержащейся в информационных ресурсах Пенсионного Фонда Российской Федерации, с информацией от Федеральной Налоговой Службы и принимает решение о возврате сумм излишне уплаченных (взысканных) страховых взносов, пеней и штрафов и в рамках регулярного обмена информацией в соответствии с



пунктом 4 статьи 21 Федерального закона от 03.07.2016 № 250-ФЗ в срок не позднее 1 рабочего дня после принятия решения.

После принятия решения в срок не позднее 3 рабочих дней Управление Пенсионного Фонда Российской Федерации формирует решение о возврате суммы излишне уплаченных (взысканных) страховых взносов.

При поступлении в территориальный налоговый орган заявления плательщика страховых взносов о возврате излишне уплаченных (взысканных) страховых взносов, образовавшихся за расчетные (отчетные) периоды, истекшие до 01.01.2017, налоговый орган оставляет указанное заявление без рассмотрения и направляет в соответствующее отделение Пенсионного Фонда Российской Федерации уведомление (сообщение) с приложением копии заявления плательщика страховых взносов, о чем в срок не позднее 3 рабочих дней с момента получения заявления письменно уведомляет плательщика страховых взносов, поскольку в соответствии с положениями статьи 21 Федерального закона № 250-ФЗ решение о возврате сумм излишне уплаченных (взысканных) страховых взносов, пеней и штрафов за отчетные (расчетные) периоды, истекшие до 01.01.2017, принимается соответствующими органами Пенсионного Фонда Российской Федерации.

В результате проведенного анализа взаимодействия подразделений Отделения Пенсионного Фонда Российской Федерации и Федеральной Налоговой Службы по вопросу принятия решения о возврате (взыскании) излишне уплаченных средств были выявлены следующие проблемы:

- Большие временные затраты на формирование запроса, его отправку и обработку документов;
- Качество документооборота, связанное с ошибками ручного ввода;
- Утеря информации, так как не происходит ее фиксации.

Данный процесс потока информации занимает большое количество времени. Что бы сократить время взаимодействия подразделений Отделения Пенсионного Фонда России будет спроектирован программный комплекс «Возвраты», посредством которого будет осуществляться документооборот,

формироваться запрос в ФНС о предоставлении сведений, сводный отчёт на заданную дату и другие аналитические материалы.

В данной главе была исследована деятельность Пенсионного Фонда Российской Федерации по Белгородской области, был приведен ряд социально значимых задач, затрагивающих интересы самых разных категорий населения, а также был проведен анализ взаимодействия подразделений ОПФР и ФНС по вопросу принятия решения о возврате (взыскании) излишне уплаченных средств, образовавшихся за расчетные (отчетные) периоды, истекшие до 01.01.2017, по причине поступления денежных средств после 01.01.2017 на коды бюджетной классификации с главой «182» и выявлены основные проблемы.

## **2 Проектирование программного комплекса «Возвраты»**

### **2.1 Проектирование бизнес-процессов, связанных с возвратом (взысканием) излишне уплаченных средств**

Во время прохождения преддипломной практики в отделении пенсионного фонда России по Белгородской области, мною было проанализировано информационное взаимодействие между отделением и управлением пенсионного фонда в г. Белгород по осуществлению возвратов плательщиком страховых взносов излишне уплаченных (взысканных) денежных средств, образованных за расчётные (отчётные) периоды, истекшие до 01.01.2017, по причине поступления денежных средств после 01.01.2017 на коды бюджетной классификации с главой «182», в которой, согласно положениям Федерального закона от 03.07.2016 N 243-ФЗ "О внесении изменений в части первую и вторую Налогового кодекса Российской Федерации в связи с передачей налоговым органам полномочий по администрированию страховых взносов на обязательное пенсионное, социальное и медицинское страхование, начиная с 1 января 2017 года, на налоговые органы возложены полномочия по администрированию страховых взносов.

В соответствии с Законом налоговые органы должны осуществлять полномочия по администрированию страховых взносов, уплачиваемых за расчётные (отчетные) периоды до 01.01.2017, установленных Федеральным законом от 24.07.2009 N 212-ФЗ "О страховых взносах в Пенсионный фонд Российской Федерации, Фонд социального страхования Российской Федерации, Федеральный фонд обязательного медицинского страхования", так и страховых взносов, установленных Налоговым кодексом Российской Федерации.

В результате проведенного анализа были выявлены проблемы во взаимодействии между подразделениями Отделения Пенсионного Фонда Российской Федерации, посредством которого осуществляется документооборот.

Для проектирования бизнес – процессов использовалось CASE-средство верхнего уровня AllFusion ProcessModeler (BPwin), поддерживающее методологии:

- IDEF0 (функциональная модель);
- DFD (DataFlow Diagram);
- IDEF3 (Workflow Diagram).

В первую очередь нужно получить неформализованное описание в произвольной текстовой форме, которая является общепонятной для специалистов различных специальностей и профилей, которые в различной степени должны участвовать в построении модели. В построении диаграммы принимают участие разработчик в качестве системного аналитика и один или группа экспертов в предметной области, которые будут представлять описание процесса. В качестве подготовки к опросу эксперта, должны быть описаны и фиксированы границы моделирования, для определения необходимой глубины и полноты требуемого описания. К тому же, если точка зрения аналитика на процесс не соответствует обычной точке зрения для эксперта, это также должно быть зафиксировано. Как вариант, автор может предварительно подготовить набор вопросов таким же способом, как для интервьюирования. В результате работы экспертной группы, как правило, составляется текстовый документ, описывающий интересующий аналитика круг вопросов. В дополнение к нему может быть в наличии письменная документация, которая дает возможность определить природу изучаемого процесса. Независимо от того, какой является информация: текстовой или вербальной - она анализируется и разделяется частями речи для идентификации списка действий посредством глаголов и отглагольных существительных, составляющих процесс, и объектов - имен существительных, - участвующих в процессе. В ряде случаев приемлемо создание графической модели процесса в присутствии экспертов. Такая модель также может быть получена в результате сбора всей необходимой информации, что дает возможность сэкономить время экспертов на деталях форматирования получающихся диаграмм.

DFD расшифровывается как Data Flow Diagrams, что в переводе означает «диаграммы потоков данных».

С помощью DFD диаграмм показывается, как каждый процесс преобразует свои входные данные в выходные, и выявляются отношения между процессами. Для этого на диаграмме потоков данных присутствуют внешние сущности, накопители данных и потоки данных.

При DFD декомпозиции необходимо представить систему в виде отдельных процессов, которые представляют собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом.

Для наглядного представления документооборота между Управлением Пенсионного Фонда России, Отделением Пенсионного Фонда России и Федеральной Налоговой Службы, ниже приведена диаграмма DFD.

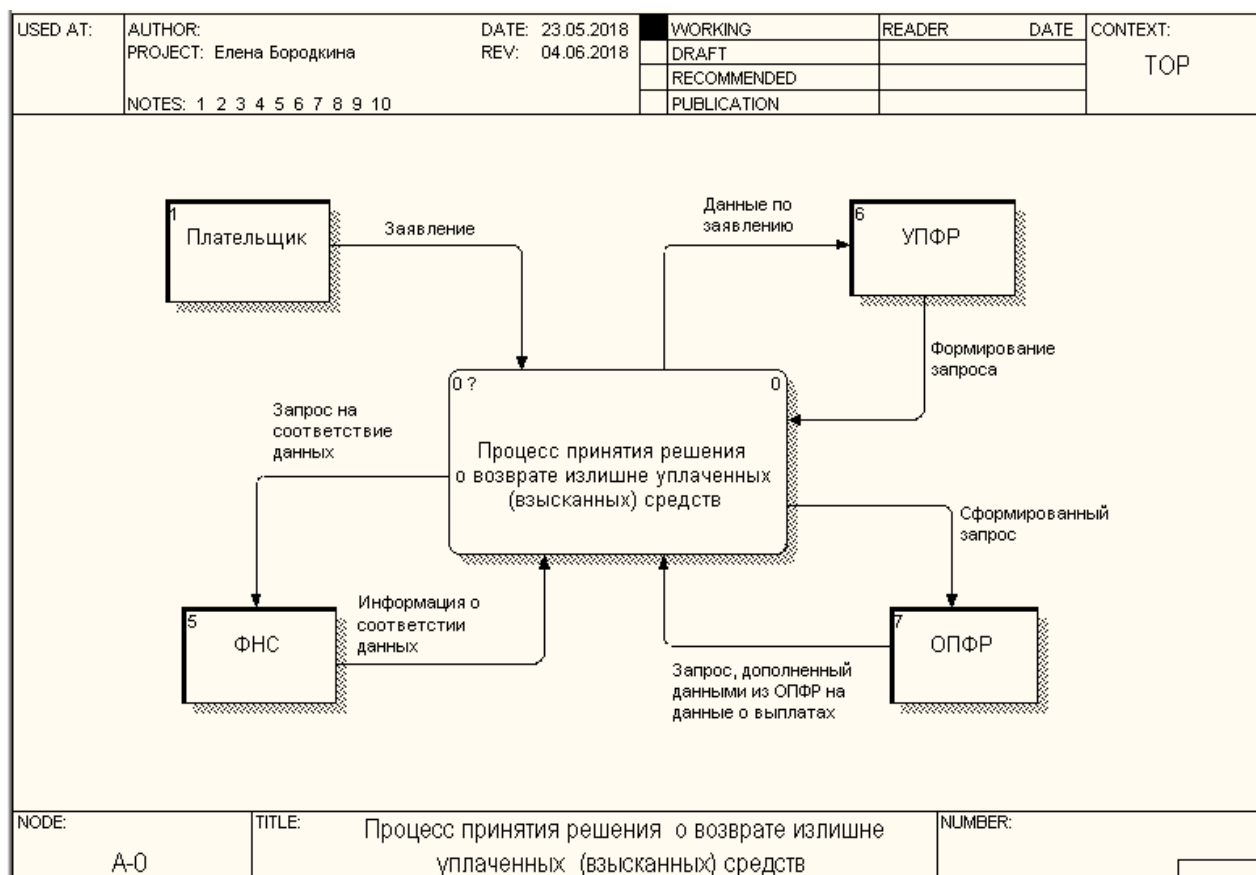


Рисунок 2.1 – Контекстная диаграмма «Процесс принятия решения о возврате излишне уплаченных (взысканных) средств» по стандарту DFD

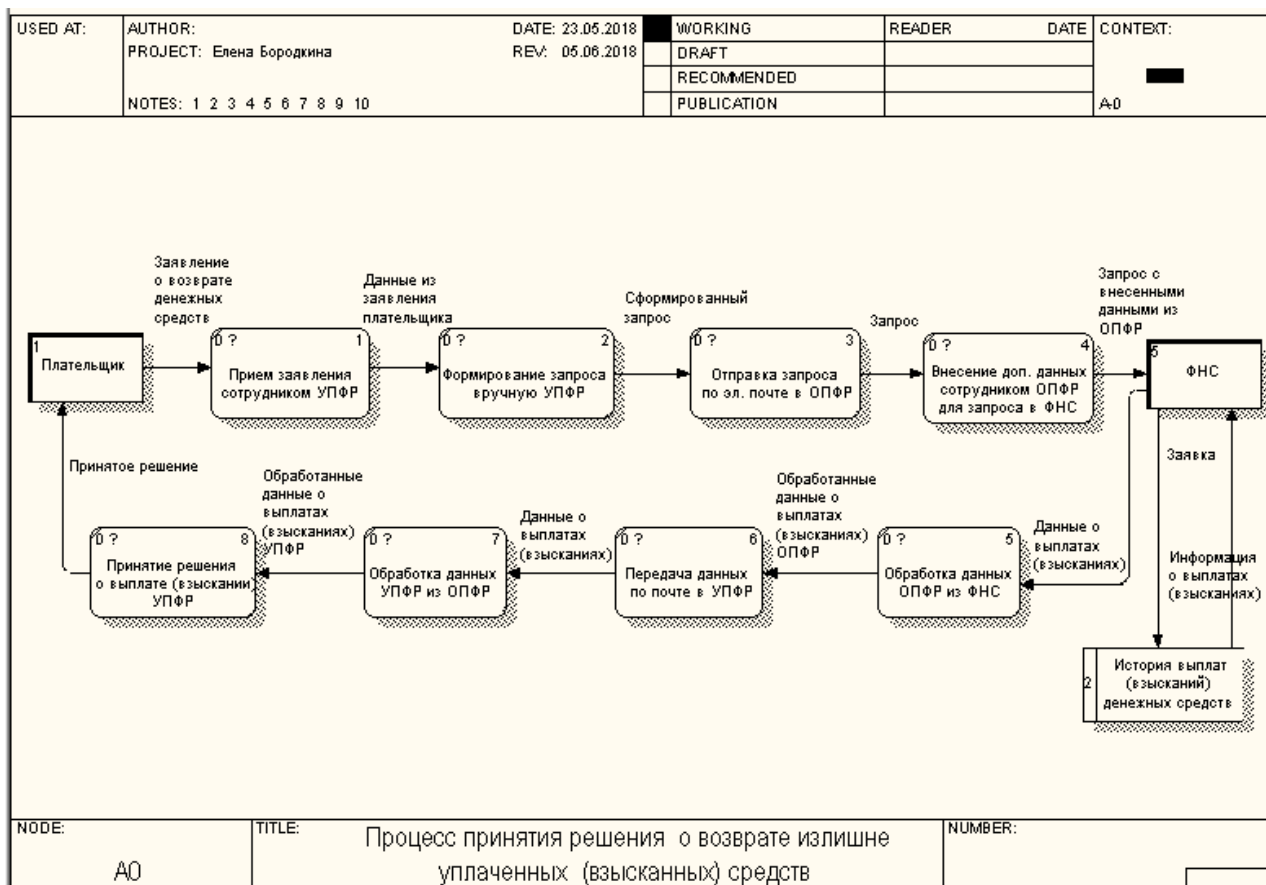


Рисунок 2.2 – Диаграмма «Процесс принятия решения о возврате излишне уплаченных (взысканных) средств» по стандарту DFD

Диаграмма содержит хранилище данных «история выплат (взысканий) денежных средств», а также процессы - прием заявления сотрудником УПФР, формирование запроса вручную УПФР, отправка запроса по эл. почте в ОПФР, внесение доп. данных сотрудником ОПФР для запроса в ФНС, обработка данных ОПФР из ФНС, передача данных по почте в УПФР, обработка данных УПФР из ОПФР и принятие решения о выплате (взыскании) УПФР. Внешние сущности – плательщик и ФНС. Так же отражены все потоки информации. Таким образом, мы видим основные потоки данных в ходе осуществления описываемого бизнес-процесса.

Для наглядного представления документооборота между Управлением Пенсионного Фонда России, Отделением Пенсионного Фонда России и Федеральной Налоговой Службы с использованием ПК «Возвраты», ниже приведена диаграмма DFD.

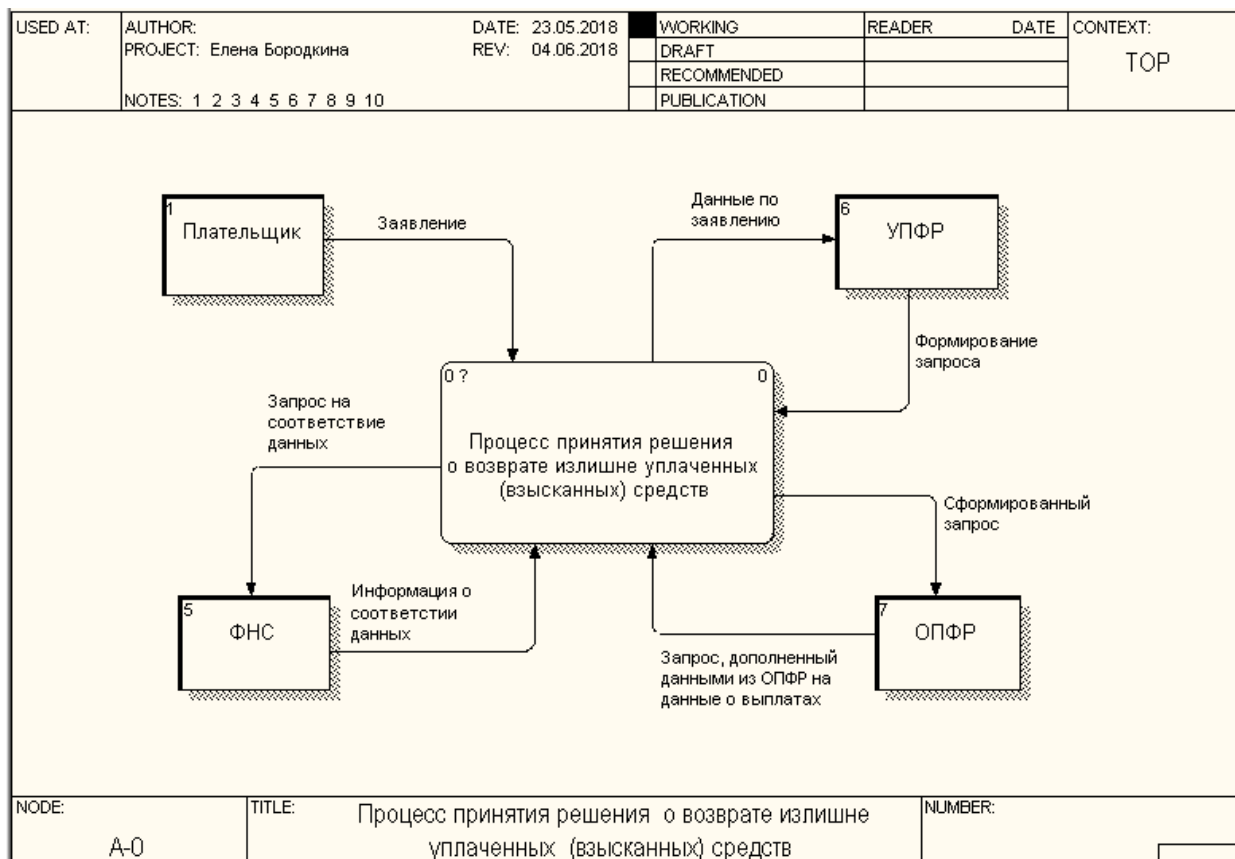


Рисунок 2.3 – Контекстная диаграмма «Процесс принятия решения о возврате излишне уплаченных (взысканных) средств с использованием ПК «Возвраты» по стандарту DFD

Ниже представленная диаграмма содержит хранилище данных «история выплат (взысканий) денежных средств» и «статистика принятых решений», а также процессы - прием заявления сотрудником УПФР, формирование запроса УПФР в ПК "Возвраты", внесение доп. данных сотрудником ОПФР по запросу в ФНС, внесение данных от ФНС ОПФР в ПК "Возвраты", принятие решения о выплате (взыскании) УПФР. Внешние сущности – плательщик, справочные данные и ФНС. Так же отражены все потоки информации. Таким образом, мы видим основные потоки данных в ходе осуществления описываемого бизнес-процесса.

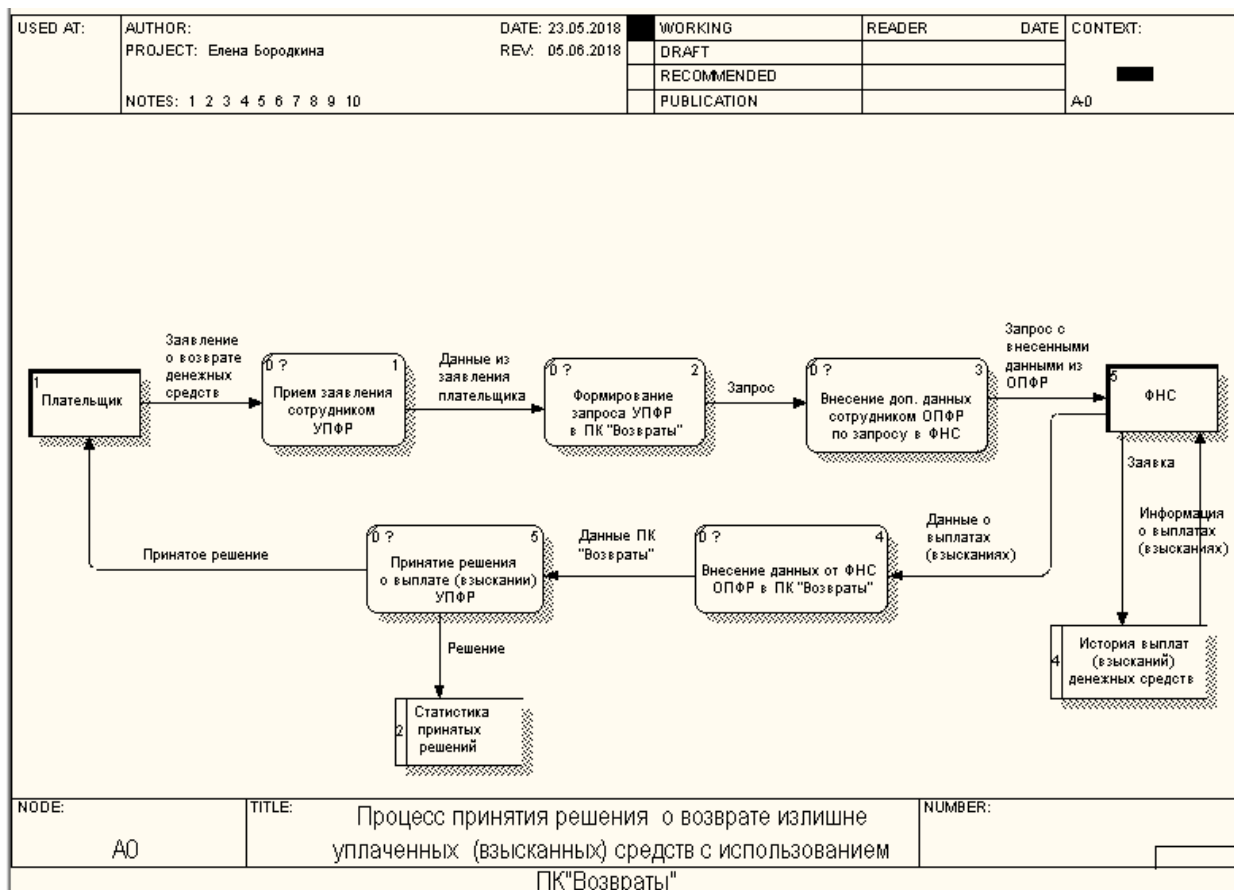


Рисунок 2.4 – Диаграмма «Формирование запроса в ФНС о возврате излишне уплаченных (взысканных) средств средствами ПК «Возвраты»» по стандарту DFD

Как можно увидеть из диаграмм, выполнение принятия решения о возврате (взыскании) денежных средств проходит значительно быстрее, чем вручную, так как процесс формирования запроса в ФНС о возврате излишне уплаченных (взысканных) средств с помощью ПК «Возвраты» осуществляется быстрее, как и поток данных от Федеральной Налоговой службы. Так же благодаря ПК «Возвраты» появляется возможность фиксирования возвратов (взысканий) и ведения статистики.

Основной задачей программного комплекса «Возвраты» является осуществление документооборота между Управлением Пенсионного фонда Российской Федерации в г. Белгород и Отделением по Белгородской области.

Программный комплекс позволит многократно сократить время по внесению данных сотрудниками, что будет способствовать увеличению



скорости работы и положительно скажется на производительности труда. будет давать точные данные о ходе принятия решения на возвраты (взыскания) денежных средств. Благодаря фиксированию данных в БД риски их потери при взаимодействии будут сведены к нулю, что позволит избежать штрафов. Также посредством программного комплекса «Возвраты» будет вестись статистика возвратов (взысканий). Программный комплекс позволит производить оперативный мониторинг работ по осуществлению возвратов плательщиком страховых взносов излишне уплаченных (взысканных) денежных средств, образованных за расчётные (отчётные) периоды, истекшие до 01.01.2017, по причине поступления денежных средств после 01.01.2017 на коды бюджетной классификации с главой «182».

## **2.2 Техническое задание**

### **2.2.1 Назначение разработки**

На разработку программного комплекса с целью оптимизации работы и повышения производительности труда по вопросу принятия Управлением ПФР решений о возврате сумм излишне уплаченных (взысканных) средств, образованные за расчетные периоды, истекшие до 01.01.2017, по причине поступления денежных средств после 01.01.2017 на коды бюджетной классификации с главой «182».

Основанием для разработки послужило индивидуальное задание во время прохождения преддипломной практики в Отделении ПФР на проектирование программного комплекса «Возвраты, посредством которого будет осуществляться взаимодействие между УПФР и ОПФР, формироваться запрос в ФНС о предоставлении сведений, а также сводный отчет на заданную дату и другие аналитические материалы.»

Срок сдачи дипломного проекта: «14» июня 2018 г.

### **2.2.2 Требования к функциональным характеристикам**

Разрабатываемый программный комплекс «Возвраты» предназначена для следующих задач:

- хранение в базе информации о наименовании района и его кода;
- хранение в базе информации о пользователях;
- хранение в базе информации о заявлениях;
- хранение в базе информации о состоянии запроса;
- хранение в базе информации о КБК и виде платежа;
- хранение в базе информации о сумме в запросе к ФНС и сумме, которая приходит в ответ от ФНС;
- хранение в базе информации о документах;
- хранение в базе информации о фиксации действий;
- хранение в базе информации о принятии решения о возврате;
- ввод, редактирование информации о плательщиках;
- ввод и редактирование информации о возвратах;
- добавление и редактирование пользователей;
- фиксация ввода и отправки данных с ФНС;
- формирование запроса в ФНС для печати.

### **2.2.3 Требования к надёжности**

Система должна сохранять работоспособность и обеспечивать ее восстановление в случаях возникновения следующих аварийных ситуаций:

- в ситуации сбоя электрического питания аппаратной части, которая приводит к перезагрузке операционной системы, восстановление программы должно осуществляться после перезапуска операционной системы и запуска исполняемого программного модуля системы;

— при ошибках в работе средств аппаратного уровня, за исключением носителей данных, восстановление функции системы возлагается на операционную систему;

— при ошибках, вызываемых программным обеспечением более низкого, аппаратного уровня (операционная система и драйверы устройств), восстановление работоспособности возлагается на операционную систему.

Для защиты компьютера от скачков напряжения электропитания и помех электрической природы должен использоваться источник бесперебойного питания.

Осуществить реализацию информационной с помощью среды программирования IntelliJ IDEA. Работа с данными производится посредством системы управления базами данных MySQL 5.5. Также при разработке используется язык программирования Java EE, сервера GlassFish, библиотекой Apache POI для работы с .docx и библиотекой Spring Security для защиты систем аутентификации и авторизации.

В главе были спроектированы бизнес – процессы, связанные с возвратом (взысканием) излишне уплаченных средств и составлено техническое задание на разработку программного комплекса «Возвраты».

## 3 Реализация программного комплекса «Возвраты»

### 3.1 Процесс программной реализации комплекса «Возвраты»

#### 3.1.1 Создание документа и вывод его на печать

Одной из ключевых функций программы должно быть автоматическое создание запросов в ФНС. Шаблон запроса представлен в приложении 1. В шаблоне необходимо заменить слова, начинающиеся с символа \$ на данные из заявления плательщика (наименование плательщика, ИНН, КПП и т.д.). Также необходимо сформировать поля таблицы и заполнить их данными о КБК, виде платежа и сумме. Сформированный запрос должен сохраниться на машине пользователя.

Для реализации данного функционала добавим в скриплет otopfr.jsp следующую строку.

Листинг 3.1 – Строка для добавления в скриплет otopfr.jsp

```
<a class="documentpechat" href=" ../dokumenpechat?id=<%=id_zayav%>"  
download>Заявление</a>
```

В данной строке создается ссылка при нажатии на которую должен открыться диалог для сохранения запроса. Для реализации этого функционала в ссылке присутствует аргумент download. Благодаря наличию этого аргумента при клике браузер не переходит по ссылке, а предлагает скачать документ, указанный в адресе ссылки. Также в ссылке передается параметр id равный переменной id\_zayav в которой хранится идентификационный номер заявления.

Сервлет DocumentPechat считывает значение параметра id в методе doPost. Из базы данных получаем необходимые данные для вставки в шаблон.

### Листинг 3.2 – Необходимые данные для вставки в шаблон

```
resultSet = statement.executeQuery("SELECT REGNUM, NAME, INN, KPP,  
DATE_ZAYA, id_STATUS " +  
    "from zayav where id_zayav=" + id);  
resultSet.next();
```

Шаблон документа хранится в таблице `dokument` в ячейке типа `longblob` это встроенный тип, который хранит двоичный большой объект в качестве значения столбца в строке таблицы базы данных.

### Листинг 3.3 – Хранение документа

```
String sql = "SELECT dokument FROM dokument WHERE id_dokument=94";  
ResultSet resultSet3 = statement3.executeQuery(sql);  
InputStream inputStream = null;  
while (resultSet3.next()) {  
    Blob blob = resultSet3.getBlob("dokument");  
    inputStream = blob.getBinaryStream();  
}
```

Необходимый шаблон считывается с помощью метода `getBlob` объекта `resultSet3` и записывается в объект `blob` класса `Blob`. В классе `Blob` имеется метод `getBinaryStream` с помощью которого можно вернуть `InputStream` объект. При помощи этого метода инициализируем переменную `inputStream`. Далее начинается работа с библиотекой `apache POI`. Сначала создается объект класса `XWPFDocument`. Он используется для создания MS-Word документа с форматом файла `.docx`. Происходит его инициализация с помощью конструктора принимающего тип `InputStream`. После инициализации идет поиск специальных символов в шаблоне и замена их на необходимые данные.

Данные действия осуществляются в цикле foreach который проходит по всем параграфам документа (XWPFParagraph). В параграфе есть много информации о стилизации, но фактический текст находится в дочернем классе XWPFRun. Поэтому в цикле по параграфам используется еще один цикл по пунктам. В этом цикле проверяется есть ли в вернувшейся строке специальный символ и в зависимости от того какая цифра находится после специального символа происходит его замена на необходимые данные.

#### Листинг 3.4 – Цикл проверки строки

```
XWPFDocument docxFile = null;
docxFile = new XWPFDocument(inputStream);
for (XWPFParagraph p : docxFile.getParagraphs()) {
    List<XWPFRun> runs = p.getRuns();
    if (runs != null) {
        for (XWPFRun r : runs) {
            String text = r.getText(0);
            if (text != null && text.contains("$")) {
                text = text.replace("$1", resultSet.getString("DATE_ZAYA"));
                text = text.replace("$2", resultSet.getString("REGNUM"));
                text = text.replace("$3", resultSet.getString("NAME"));
                text = text.replace("$4", resultSet.getString("INN"));
                text = text.replace("$5", resultSet.getString("KPP"));
                r.setText(text, 0);
            }
        }
    }
}
```

Заменяв данные произведем поиск таблицы в документе и сохраняем ее в переменной T класса XWPFTable. Также создается еще один документ типа

XWPFDocument в котором в дальнейшем будут создаваться параграфы с текстом и вставляться в таблицу. Выполняется запрос к базе данных для получения КБК, вида платежа и суммы.

#### Листинг 3.5 – Создание документа типа XWPFDocument

```
XWPFTable T = null;
for (XWPFTable tbl : docxFile.getTables()) {
    T = tbl;
}
XWPFDocument document= new XWPFDocument();
resultSet2 = statement2.executeQuery("select v.id_kbkvid, k.kbk, v.vidplat,
v.summ1/*, v.summ2*/ " +
    "from (select id_kbkvid, id_zayav, kbk " +
    "from kbk join kbkvid on kbk.id_kbk= kbkvid.id_kbk) k join " +
    "(select id_kbkvid, id_zayav, vidplat, summ1, summ2 from vidplat " +
    "join kbkvid on vidplat.id_vidplat= kbkvid.id_vidplat) v on v.id_kbkvid =
k.id_kbkvid " +
    "where v.id_zayav = " + id + " order by v.id_kbkvid");
```

В цикле while по результатам запроса создаются новые ячейки таблицы и заполняются данными. С помощью метода createRow создается новая строка в таблице. Во временном документе создается новый параграф, ему указывается размер текста, убирается жирность так как этот параметр устанавливается по умолчанию и присваивается текст. Затем методом getCell в нужную ячейку вставляется полученный параграф. Это действие повторяется для КБК, вида платежа и суммы.

### Листинг 3.6 – Применение метода createRow и getCell

```
while (resultSet2.next()) {  
    XWPFTableRow tableRowTwo = T.createRow();  
    XWPFParagraph paragraph = document.createParagraph();  
    XWPFRun run = paragraph.createRun();  
    run.setFontSize(13);  
    run.setBold(false);  
    run.setText(resultSet2.getString("k.kbk"));  
    tableRowTwo.getCell(0).setParagraph(paragraph);  
    XWPFParagraph paragraph2 = document.createParagraph();  
    XWPFRun run2 = paragraph2.createRun();  
    run2.setFontSize(13);  
    run2.setBold(false);  
    run2.setText(resultSet2.getString("v.vidplat"));  
    tableRowTwo.getCell(1).setParagraph(paragraph2);  
    XWPFParagraph paragraph3 = document.createParagraph();  
    XWPFRun run3 = paragraph3.createRun();  
    run3.setFontSize(13);  
    run3.setBold(false);  
    run3.setText(resultSet2.getString("v.summ1"));  
    tableRowTwo.getCell(2).setParagraph(paragraph3);  
}
```

После произведенных действий в объекте docxFile будет сформирован необходимый шаблон. Теперь его необходимо привести к необходимому формату и отправить объектом resp который предоставит метаданные для среды выполнения. Шаблон документа записывается с помощью метода write в объект bo класса ByteArrayOutputStream. Далее заполняются параметры объекта resp, тоесть что передается пользователю. Метод setContentType в которой можно указать mime тип передаваемых данных. APPLICATION/OCTET-



STREAM обозначает двоичные данные. В методе `setHeader` указывается имя файла заголовок `Content-Disposition` является индикатором того, что ожидаемый контент ответа будет отображаться в браузере, как веб-страница или часть веб-страницы, или же как вложение, которое затем может быть скачано и сохранено локально. Далее указывается размер передаваемого файла методом `setContentLength` в которые передается значение возвращаемое функцией `bo.size()`.

Листинг 3.7 – Приведение шаблона к необходимому формату

```
ByteArrayOutputStream bo= new ByteArrayOutputStream();
docxFile.write(bo);
resp.setContentType("application/octet-stream");
String headerKey = "Content-Disposition";
String headerValue = String.format("attachment; filename=\"%s\"",
"document.docx");
resp.setHeader(headerKey, headerValue);
resp.setContentLength(bo.size());
resp.getOutputStream().write(bo.toByteArray());
```

Таким образом был сформирован запрос и передан пользователю на сохранение.

### 3.1.2 Авторизация пользователей

При заходе в приложение первое что необходимо сделать это провести авторизацию пользователя, чтобы можно было фиксировать действия, совершенные им во время работы и предоставить доступ только к тем данным которые нужны ему для выполнения своих обязанностей.

Страница login.jsp представляет из себя простую форму с полями ввода логина и пароля кнопкой «Вход» которая отправит данные на сервлет Avt для проверки введенных данных и перенаправления на необходимую страницу.

Листинг 3.8 – Авторизацию пользователя

```
<form action='../avt' method='post'>
  <tr><td>Логин:</td> <td><input type='text' name='login'><br></td></tr>
  <tr><td>Пароль:</td> <td><input type='password'
name='password'><br></td></tr>
  <% if(session.getAttribute("user")!=null &&
session.getAttribute("user").equals("error")){//
  %>
  <tr><td colspan="2">Неверный логин и/или пароль!</td></tr>
  <% }%>
  <tr><td></td> <td><input type='submit' name='Вход'></td></tr>
</form>
```

В сервлете Avt в методе doPost происходит прием данных логина и пароля и создается сессия. В случае успешной проверки данных в сессии будут храниться данные пользователя.

Листинг 3.9 – Прием данных логина и пароля

```
req.setCharacterEncoding("UTF-8");
HttpSession session = req.getSession();
String login = req.getParameter("login");
String password = req.getParameter("password");
```

Следующим шагом является проверка на наличие введенных данных в базе данных.

### Листинг 3.10 – Запрос к базе проверяющий введенный логин и пароль

```
Connection connection = null;
Conn conn = new Conn();
    connection = conn.setConnection();
    String qwery ="SELECT * FROM USERS WHERE LOGIN=? and
PASSWORD=?";
    preparedStatement = connection.prepareStatement(qwery);
    preparedStatement.setString(1, login);
    preparedStatement.setString(2, password);
    ResultSet resultSet = preparedStatement.executeQuery();
    while (resultSet.next()) {
        statement = connection.createStatement();
        qwery ="SELECT namerayon FROM rayon WHERE
id_rayon="+resultSet.getInt("id_rayon");
        resultSet2 = statement.executeQuery(qwery);
        resultSet2.next();
        usersPFR = new UsersPFR(resultSet.getInt("id_user"),
resultSet.getString("login"),
        resultSet.getString("password"), resultSet.getString("name"),
        resultSet.getInt("id_rayon"),resultSet.getString("namerayon"));
    }
}
```

Для проверки на наличие введенных данных в базе данных создается подключение к базе данных ркв. Данные о подключении находятся в классе Conn. Соединение хранится в переменной connection класса Connection. Далее создается запрос к базе проверяющий введенный логин и пароль. Запрос реализуется с помощью класса PreparedStatement. PreparedStatement в отличии от класса Statement имеет защиту от SQL-инъекций путем встроенного экранирования кавычек и других специальных символов. Также предварительная компиляция и кэширование базы данных SQL-оператора

приводит к более быстрому выполнению и возможности повторного использования одного и того же оператора SQL в партиях. Если данные были найдены по `id_rayon` в таблице `USERS` происходит поиск названия района обычным `Statement` запросом т.к. здесь не требуется защита от SQL-инъекций. Полученные данные заносятся в модель `usersPFR` для дальнейшего использования ее в сессиях.

Дальше происходит перенаправление на нужную страницу в зависимости от результатов сравнения с базой данных.

#### Листинг 3.11 – Осуществление перенаправления на нужную страницу

```
if (usersPFR!=null){
    session.setAttribute("user", usersPFR);
    if (usersPFR.getRAYON()==2){
        resp.sendRedirect("pages/opfr.jsp");
    } else{
        session.setAttribute("param",1);//для определения вкладки
        resp.sendRedirect("pages/upfr.jsp");}
} else {
    session.setAttribute("user", "error");
    resp.sendRedirect("pages/login.jsp");
}
```

Если объект `usersPFR` не создан, то атрибут сессии `user` заполняется сообщением об ошибке, а пользователя возвращает на страницу `login.jsp` для повторного ввода логина и пароля где выводится сообщение, что введенные данные не верны. Иначе в атрибут `user` записывается объект `usersPFR` и в зависимости от параметра `getRAYON` пользователя перенаправляет на страницу `opfr.jsp` если `usersPFR.getRAYON()==2` в противном случае на страницу `upfr.jsp`.

## 3.2 Разработка БД

При осуществлении деятельности по принятию решений о возврате сумм излишне уплаченных денежных средств происходит обмен большим количеством данных между пользователями. Их необходимо хранить, предоставлять по требованию. Для этого необходимо разработать базу данных.

Программой будет использоваться в разных районах поэтому первым делом создается таблица rayon. В ней будут поля namerayon, kod для хранения наименования района и его кода в пфр.

Листинг 3.12 – Таблица rayon

```
CREATE TABLE `rayon` (  
  `id_rayon` INT(11) NOT NULL AUTO_INCREMENT,  
  `namerayon` VARCHAR(255) NOT NULL,  
  `kod` VARCHAR(4) NOT NULL,  
  PRIMARY KEY (`id_rayon`)  
)
```

В районах находятся работники, которые и будут пользоваться программой. Таблица users предназначена для хранения данных пользователей логина, пароля и имени, а также поле для хранения id района.

Первыми в программу будут заходить пользователи УПФР и вводить данные из заявлений плательщиков регистрационный номер, наименование плательщика, ИНН, КПП и дата заявления. Хранятся эти данные в таблице zaav. Еще в ней находятся индексы для связи с таблицей users и stat.

Листинг 3.13 – Таблица users

```
CREATE TABLE `users` (  
  `id_user` INT(11) NOT NULL AUTO_INCREMENT,  
  `login` VARCHAR(20) NOT NULL,  
  `password` VARCHAR(20) NOT NULL,  
  `name` VARCHAR(50) NOT NULL,  
  `id_rayon` INT(11) NOT NULL,  
  PRIMARY KEY (`id_user`),  
  INDEX `FK_users_rayon` (`id_rayon`),  
  CONSTRAINT `FK_users_rayon` FOREIGN KEY (`id_rayon`) REFERENCES `rayon`  
  (`id_rayon`) ON UPDATE CASCADE ON DELETE CASCADE  
)
```

Листинг 3.14 – Таблица zayav

```
CREATE TABLE `zayav` (  
  `id_zayav` INT(11) NOT NULL AUTO_INCREMENT,  
  `regnum` VARCHAR(10) NULL DEFAULT NULL,  
  `name` VARCHAR(250) NULL DEFAULT NULL,  
  `inn` VARCHAR(13) NULL DEFAULT NULL,  
  `kpp` VARCHAR(10) NULL DEFAULT NULL,  
  `date_zaya` DATE NULL DEFAULT NULL,  
  `id_status` TINYINT(4) NULL DEFAULT '1',  
  `id_users` INT(11) NULL DEFAULT NULL,  
  `date_zagr` DATETIME NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id_zayav`),  
  INDEX `FK_zayav_stat` (`id_status`),  
  INDEX `FK_zayav_users` (`id_users`),  
  CONSTRAINT `FK_zayav_stat` FOREIGN KEY (`id_status`) REFERENCES `stat`  
  (`id_stat`),  
  CONSTRAINT `FK_zayav_users` FOREIGN KEY (`id_users`) REFERENCES `users`  
  (`id_user`)  
)
```

Stat – это небольшая таблица для хранения состояния запроса. По ее индексу всегда можно будет узнать отправлен ли запрос в ФНС или по нему уже вынесено решение.

Листинг 3.15 – Таблица zaуav

```
CREATE TABLE `stat` (  
  `id_stat` TINYINT(4) NOT NULL AUTO_INCREMENT,  
  `stat` VARCHAR(250) NOT NULL,  
  PRIMARY KEY (`id_stat`)  
)
```

В базе данных должны храниться данные о КБК и виде платежа. Эти данные статичны задаются при создании таблиц и в программе меняться не будут, а будут использоваться при создании выпадающего списка. Они хранятся в таблицах vidplat и kbk.

Листинг 3.16 – Таблицы vidplat и kbk

```
CREATE TABLE `kbk` (  
  `id_kbk` INT(11) NOT NULL AUTO_INCREMENT,  
  `kbk` VARCHAR(250) NOT NULL,  
  PRIMARY KEY (`id_kbk`)  
)  
CREATE TABLE `vidplat` (  
  `id_vidplat` INT(11) NOT NULL AUTO_INCREMENT,  
  `vidplat` VARCHAR(250) NOT NULL,  
  PRIMARY KEY (`id_vidplat`)  
)
```

Связь между заявлением и видом платежа и КБК будет обеспечивать таблица kbkvid. Помимо идентификаторов таблиц zaуav, vidplat, kbk в ней

будет два поля для хранения сумм summ1 и summ2. summ1 – это сумма, которая указанная в запросе к ФНС. summ2 – сумма, которая приходит в ответ от ФНС.

Листинг 3.17 – Таблица kbkvid

```
CREATE TABLE `kbkvid` (  
  `id_kbkvid` INT(11) NOT NULL AUTO_INCREMENT,  
  `id_zayav` INT(11) NULL,  
  `id_kbk` INT(11) NULL,  
  `id_vidplat` INT(11) NULL,  
  `summ1` VARCHAR(10) NULL,  
  `summ2` VARCHAR(10) NULL DEFAULT NULL,  
  PRIMARY KEY (`id_kbkvid`),  
  INDEX `FK_kbkvid_zayav` (`id_zayav`),  
  INDEX `FK_kbkvid_kbk` (`id_kbk`),  
  INDEX `FK_kbkvid_vidplat` (`id_vidplat`),  
  CONSTRAINT `FK_kbkvid_kbk` FOREIGN KEY (`id_kbk`) REFERENCES `kbk`  
  (`id_kbk`),  
  CONSTRAINT `FK_kbkvid_vidplat` FOREIGN KEY (`id_vidplat`) REFERENCES `vidplat`  
  (`id_vidplat`),  
  CONSTRAINT `FK_kbkvid_zayav` FOREIGN KEY (`id_zayav`) REFERENCES `zayav`  
  (`id_zayav`)  
)
```

Для хранения документов будет использоваться таблица dokument. В этой таблице в будет храниться ответы от ФНС и файл шаблона, а также название файла. Для хранения документа у поля dokument устанавливается тип **LONGBLOB**. **LONGBLOB** - представляет собой большой файл, обычно изображение или звуковой файл, который должен обрабатываться (например, загружаться или храниться в базе данных) особым образом из-за его размера.



### Листинг 3.18 – Таблица dokument

```
CREATE TABLE `dokument` (  
  `id_dokument` INT(11) NOT NULL AUTO_INCREMENT,  
  `dokument` LONGBLOB NULL,  
  `name_dokument` VARCHAR(100) NULL DEFAULT NULL,  
  `name_file` VARCHAR(100) NULL DEFAULT NULL,  
  PRIMARY KEY (`id_dokument`),  
  UNIQUE INDEX `name_file` (`name_file`)  
)
```

### Листинг 3.19 – Таблица opfr

```
CREATE TABLE `opfr` (  
  `id_opfr` INT(11) NOT NULL AUTO_INCREMENT,  
  `nom_zap` VARCHAR(20) NOT NULL,  
  `data_otpr` DATE NOT NULL,  
  `nom_otv` VARCHAR(20) NULL DEFAULT NULL,  
  `data_otv` DATE NULL DEFAULT NULL,  
  `id_dokument` INT(11) NULL DEFAULT NULL,  
  `id_zayav` INT(11) NOT NULL,  
  `id_user` INT(11) NOT NULL,  
  PRIMARY KEY (`id_opfr`),  
  INDEX `FK_opfr_dokument` (`id_dokument`),  
  INDEX `FK_opfr_users` (`id_user`),  
  INDEX `FK_opfr_zayav` (`id_zayav`),  
  CONSTRAINT `FK_opfr_dokument` FOREIGN KEY (`id_dokument`) REFERENCES  
  `dokument` (`id_dokument`),  
  CONSTRAINT `FK_opfr_users` FOREIGN KEY (`id_user`) REFERENCES `users`  
  (`id_user`),  
  CONSTRAINT `FK_opfr_zayav` FOREIGN KEY (`id_zayav`) REFERENCES `zayav`  
  (`id_zayav`)  
)
```

Фиксация данных действий, совершенных в ОПФР будет храниться в таблице `opfr`. Основными полями в этой таблице являются `nom_zap` – номер запроса, `data_otpr` – дата отправки запроса в ФНС, `nom_otv` – номер ответа ФНС, `data_otv` – дата ответа ФНС. Также присутствуют три поля с идентификаторами для связи с таблицами `dokument`, `users`, `zayav`.

Создав эти таблицы в базу данных можно будет добавлять пользователей, вносить данные из заявления плательщика, фиксировать отправку и прием данных с ФНС. Нахватает лишь шага принятия решения по результатам ответа от ФНС. Для этого будет служить таблица `rech`. В ней будут поля для хранения номера решения и даты.

Листинг 3.20 – Таблица `rech`

```
CREATE TABLE `rech` (  
  `id_rech` INT(11) NOT NULL AUTO_INCREMENT,  
  `id_zayav` INT(11) NULL DEFAULT NULL,  
  `nom_rech` VARCHAR(25) NULL DEFAULT NULL,  
  `data_rech` DATE NULL DEFAULT NULL,  
  PRIMARY KEY (`id_rech`),  
  INDEX `FK_rech_zayav` (`id_zayav`),  
  CONSTRAINT `FK_rech_zayav` FOREIGN KEY (`id_zayav`) REFERENCES `zayav`  
  (`id_zayav`)  
)
```

После создания всех таблиц получим базу данных необходимую для решения поставленной задачи. Схема созданной базы данных полученная в СУБД DBeaver представлена на рисунке ниже.

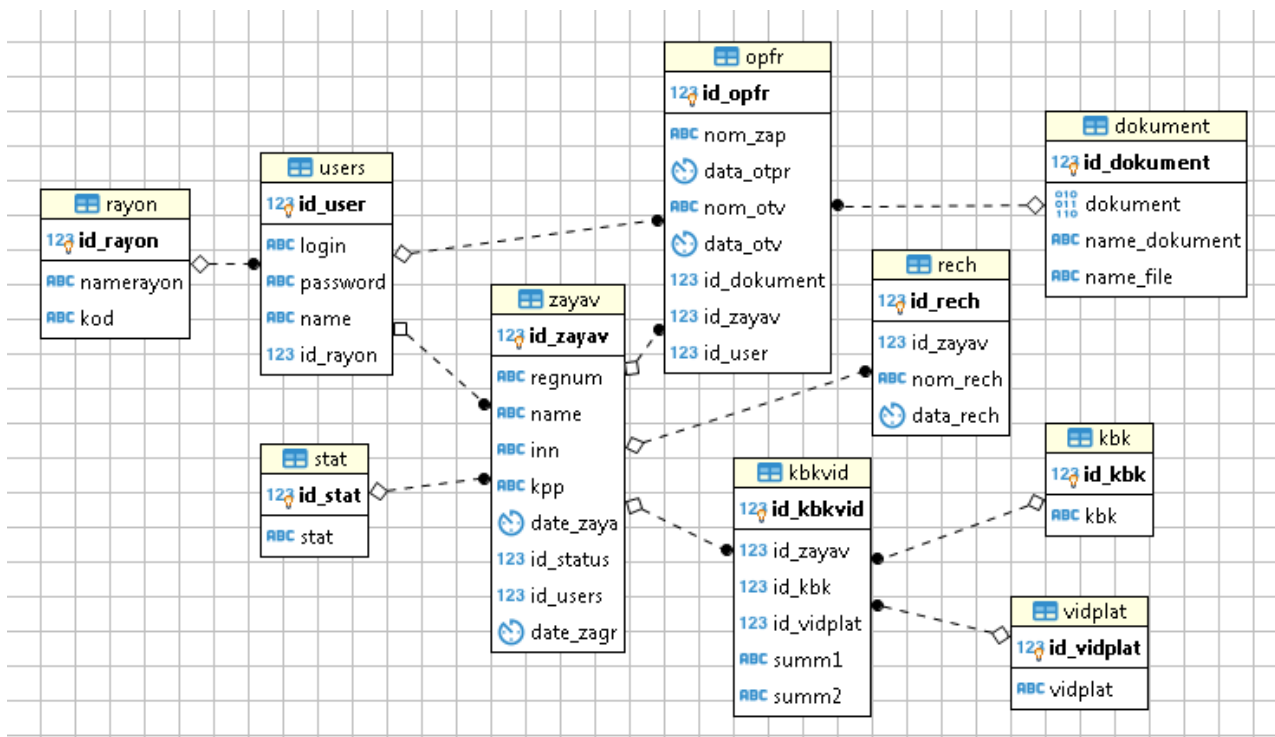


Рисунок 3.1 – Схема базы данных, сделанная в DBeaver

### 3.3 Расчет затрат на реализацию программного комплекса «Возвраты»

Разработан исследовательский прототип программного комплекса «Возвраты». Исследовательский прототип служит для апробации проектных решений, значит для вывода о целесообразности производственного программного продукта необходимо выполнить расчет затрат на его реализацию.

Ниже приведены те расчеты, которые необходимы для определения затрат на выполнение проекта:

- Расчет трудоемкости по стадиям проектирования программного комплекса;
- Расчет основной и дополнительной заработной платы;
- Отчисления на социальные нужды;
- Амортизация аппаратных средств вычислительной техники;
- Накладные расходы;

— Расходы на использование Интернет.

Единовременными затратами разработчика программного комплекса следует считать затраты на исследовательскую работу, постановку задачи, проектирование, создание программного кода, отладку, опытную эксплуатацию, оформление документов.

Трудоемкость разработки программного комплекса рассчитывается по формуле (3.1):

$$T_{об} = \sum_{i=1}^n T_i \quad (3.1)$$

где  $T_i$  - трудоемкость по стадиям, дн.; n – количество стадий разработки.

Таким образом,  $T_{об} = 10 + 10 + 15 + 30 + 15 = 80$  дней.

В таблице 3.1 показана трудоемкость по стадиям проектирования программного комплекса.

Таблица 3.1 – Содержание стадий разработки программного комплекса

№ п/п	Стадии НИР	Содержание работ	Трудоемкость	
			Дни	%
1	Техническое задание	Анализ предметной области с руководителем, утверждение технического задания и плана работ. Постановка задачи.	10	12,5
2	Эскизный проект	Рассмотрение теоретических аспектов по данной тематике. Планирование и предварительная разработка решения для входных и выходных данных. Проектирование процессов.	10	12,5
3	Технический проект	Разработка проектного решения. Подготовка сценария по которому будет осуществляться взаимодействие объектов.	15	18,7
4	Рабочий проект	Разработка программного комплекса, отладка системы.	30	37,6

5	Внедрение	Апробация результатов, подготовка отчета по дипломной работе.	15	18,7
Итого			80	100

В смету затрат на разработку ПО включаются:

- основная и дополнительная зарплата разработчика и руководителя;
- отчисления на социальные нужды;
- стоимость машинного времени на подготовку и отладку программ;
- стоимость инструментальных средств;
- затраты на использование сети Интернет;
- накладные расходы.

Основная заработная плата исполнителей разработки программного комплекса включает как зарплату разработчика, так и руководителя разработчика.

Для расчета основной заработной платы  $Z_{осн}$  применим формулу (3.2):

$$Z_{осн} = T_{об} \cdot Z_{ср.дн} \quad (3.2)$$

где  $Z_{ср.дн}$  – среднедневная зарплата одного работника, денежные единицы, руб.;  $T_{об}$  – общая трудоемкость проекта, дни.

На консультации запланировано:

Руководитель разработчика – 25 часов.

Среднечасовая заработная плата руководителя разработчика составляет 120 руб./ч.

Для расчёта заработной платы разработчика ( $Z_{раз}$ ) следует иметь в виду то, что научно-исследовательские работы производились на протяжении 80 дней.  $80 \cdot 8 = 640$  часов

Расчет основной заработной платы приведен в таблице 3.2.

Таблица 3.2 – Основная заработная плата

Исполнители	Время. Ч.	Оплата. руб/ч.	Итог. руб.
Руководитель	25	120	3000

Разработчик	640	100	64000
Всего			67000

Дополнительная заработная плата равна 10% от основной, следовательно:

$$З_{доп} = 0,1 * З_{осн} = 0,1 * 67000 = 6700 \text{ руб.}$$

Итого основная и дополнительная заработная плата составляет:

$$З_{общ} = 67000 + 6700 = 73700 \text{ руб.}$$

Рассчитать отчисления на заработную плату можно по формуле:

$$О_{соц} = З_{общ} * 0,302 = 73700 * 0,302 = 22257,4 \text{ руб.}$$

Для расчета стоимости машинного времени применяется формула (3.3):

$$T_m = 0,35 * T_{экс} + 0,6 * T_{тех \text{ пр}} + 0,8 * T_{раб \text{ пр}} + 0,6 * T_{вн} \quad (3.3)$$

$$T_m = 0,35 * 10 + 0,6 * 15 + 0,8 * 30 + 0,6 * 15 = 3,5 + 9 + 24 + 9 = 45,5 \text{ дня}$$

Для расчета затрат на электроэнергию возьмем следующую формулу (3.4):

$$Z_{эл} = C_{эл} * M_{эвм} * T_m * T_{сут} \quad (3.4)$$

где  $C_{эл}$  – стоимость 1 кВт/ч электроэнергии, руб;  $M_{эвм}$  – мощность ЭВМ;  $T_{сут}$  – время работы ЭВМ в сутки, ч.

$$Z_{эл} = 5,38 \text{ руб} * 0,4 \text{ кВт} * 45,5 \text{ дней} * 6 \text{ ч} = 587,496 \text{ руб.}$$

Амортизация аппаратных средств вычислительной техники рассчитывается следующим образом.

Используется формула (3.5):

$$A_m = \frac{O_{\phi} \cdot N_{ам}}{365 \cdot 100} \cdot t_{эвм} \quad (3.5)$$

где  $A_m$  – амортизационные отчисления, руб.;  $O_{\phi}$  – первоначальная стоимость ЭВМ и оборудования, руб.;  $N_{ам}$  – норма амортизации, %;  $t_{эвм}$  – время использования оборудования, дн.

Первоначальную стоимость ЭВМ  $O_{\phi} = 50000$ , получаем

$$A_m = \frac{50000 \cdot 30}{365 \cdot 100} \cdot 45.5 = 1869.9 \text{ руб.}$$

Стоимость машинного времени:

$$Зомв = 1869,9 + 587,5 = 2457,4 \text{ руб.}$$

Рассчитаем накладные расходы:

$$P_n = Зосн * 0,2 = 67000 * 0,2 = 13400 \text{ руб.}$$

Смета затрат на разработку программного средства приведена в таблице 3.6.

Таблица 3.6 – Смета затрат на разработку

№ п/п	Элемент затрат	Сметная стоимость, руб.
1	Основная и дополнительная заработная плата	73700
2	Отчисления на социальные нужды	22257,4
3	Амортизация ЭВМ и оборудования	1869,9
4	Затраты на электроэнергию	676,39
5	Накладные расходы	13400
6	Расходы на использование Интернет	1500
Итого:		113403,69

Сумма затрат на разработку распределяется по этапам. Она распределена и показана в таблице 3.7.

Таблица 3.7 – План реализации проекта

Этапы проектирования	Период (месяц)						Итого
	1	2	3	4	5	6	
Техническое задание	10						10
Эскизный проект	10						10
Технический проект	1	14					15

Рабочий проект		7	21	2			30
Внедрение				15			15
Итого	21	21	21	17			80

В соответствии с данным планом реализации проекта составляется инвестиционный план, который приведен в таблице 3.8.

Таблица 3.8 – План инвестиций

№ п/п	Этапы проектирования	Период (месяц)				Итого:
		1	2	3	4	
1	Тех. задание	10 300,51				10 300,51
2	Эскизный проект	18 050,41				18 050,41
3	Тех. проект	1 417,55	19 845,65			21 263,19
4	Рабочий проект		9 922,82	29 768,47	2 835,09	42 526,38
5	Внедрение				21 263,19	21 263,19
Итого:		29 768,47	29 768,47	29 768,47	24 098,28	113 403,69

Ниже представлены таблицы сравнения затрачиваемого времени на обработку одного обращения плетельщика в УПФР.

Таблица 3.9 – Временные затраты на формирования запроса в УПФР

№ п/п	Процесс формирования запроса в УПФР	Время выполнения вручную	Время выполнения посредством ПК «Возвраты»
1	2	3	4
1	Прием заявления плетельщика сотрудником УПФР	1 минута	1 минута
2	Открытие шаблона запроса на возврат (взыскание) излишне уплаченных средств	1 минута	-
3	Внесение вручную данных из	7 минут	-



	заявления плательщика в шаблон (регистрационный номер, наименование ФИО плательщика, ИНН, КПП, дата заявления, КБК, вид платежа, и сумма)		
--	---	--	--

Продолжение таблицы 3.9

1	2	3	4
4	Внесение данных из заявления плательщика в программный комплекс «Возвраты» (регистрационный номер, наименование ФИО плательщика, ИНН, КПП, дата заявления, КБК, вид платежа, и сумма)	-	3 минуты
5	Сохранение документа и отправка по электронной почте в ОПФР	3 минуты	-
6	Сохранение сформированного запроса в программном комплексе	-	1 минута
Итого:		12 минут	5 минут

Таблица 3.10 – Временные затраты на Процесс дополнения запроса ОПФР

№ п/п	Процесс дополнения запроса ОПФР	Время выполнения	Время выполнения посредством ПК «Возвраты»
1	Внесение номера запроса	1 минута	30 сек
2	Внесение даты направления	1 минута	30 сек
3	Отправка по почте в ФНС	3 минуты	3 минуты
	Итого:	5 минут	4 минуты

ФНС в свою очередь сравнивает выплаты (взыскания) и отправляет информацию в ОПФР.

Таблица 3.11 – Процесс дополнения запроса ОПФР данными из ФНС

№ п/п	Процесс дополнения запроса ОПФР данными из ФНС	Время выполнения	Время выполнения посредством ПК «Возвраты»
1	2	3	4
1	Дополняет запроса данными из ФНС (регистрационный номер, наименование ФИО плательщика, ИНН, КПП, дата заявления, КБК, вид платежа, и сумма)	7 минут	5 минут

Продолжение таблицы 3.11

1	2	3	4
2	Ставит входящий номер ответа от ФНС и его дату	1 минута	1 минута
3	Сохраняет документ и отправляет по электронной почте в УПФР	3 минуты	-
Итого:		11 минут	6 минут

Таблица 3.12 – Временные затраты на принятие решения о возврате (взыскании) излишне уплаченных средств УПФР

№ п/п	Принятие решения о возврате (взыскании) излишне уплаченных средств УПФР	Время выполнения	Время выполнения посредством ПК «Возвраты»
1	Сравнение суммы платежей	1 минута	1 минута
2	Принятие решения о возврате (взыскании) излишне уплаченных средств	1 минута	1 минута
3	вносит данные о решении в программный комплекс	-	3 минуты
Итого		2 минуты	5 минут

Таблица 3.13 – Общее время выполнения одного обращения

№ п/п	Общее время выполнения	Общее время выполнения посредством ПК «Возвраты»
1	30 минут	20 минут

Исходя из представленных таблиц видно, что время обработки обращения одного плательщика посредством ПК «Возвраты» происходит значительно быстрее и занимает на 10 минут меньше, чем время обработки обращения одного плательщика вручную. Сокращение времени позволяет обработать большее количество обращений плательщиков.

Внедрение программный комплекса «Возвраты» позволит многократно сократить время по внесению данных сотрудниками, что будет способствовать увеличению скорости работы и положительно скажется на производительности труда. будет давать точные данные о ходе принятия решения на возвраты (взыскания) денежных средств. Благодаря фиксированию данных в БД риски их потери при взаимодействии будут сведены к нулю, что позволит избежать штрафов. Также посредством программного комплекса «Возвраты» будет вестись статистика возвратов (взысканий). Программный комплекс позволит производить оперативный мониторинг работ по осуществлению возвратов плательщиком страховых взносов излишне уплаченных (взысканных) денежных средств, образованных за расчётные (отчётные) периоды, истекшие до 01.01.2017, по причине поступления денежных средств после 01.01.2017 на коды бюджетной классификации с главой «182».

Реализация программного комплекса «Возвраты» для деятельности ПФР является целесообразной, поскольку:

- разработанный исследовательский прототип показал оправданность предложенных проектных решений;
- проведенный расчет затрат показал, что их общий объем для регионального Отделения Пенсионного Фонда Российской Федерации является приемлемым.

Эффективность внедрения программного комплекса Возвраты определяет:

- снижение временных затрат на обработку документов;
- повышение качества документооборота связанных с ошибками ручного ввода;
- возможность анализа процессов возврата (взысканий) за счет хранения информации в БД.

В данной главе выпускной квалификационной работы была описана разработка базы данных программного комплекса «Возвраты», а также его программная реализация.

В главе были приведены план инвестиций и все необходимые расчеты для определения экономических затрат проекта:

- Расчет трудоемкости по стадиям проектирования программного комплекса;
- Расчет основной и дополнительной заработной платы;
- Отчисления на социальные нужды;
- Амортизация аппаратных средств вычислительной техники;
- Накладные расходы;
- Расходы на использование Интернет.

В главе были представлены таблицы сравнения затрачиваемого времени на обработку одного обращения плательщика в УПФР. Время обработки обращения одного плательщика посредством ПК «Возвраты» происходит значительно быстрее и занимает на 10 минут меньше, чем время обработки обращения одного плательщика вручную. Это позволяет обработать большее количество обращений плательщиков.

Также была обоснована целесообразность реализации программного комплекса «Возвраты» для деятельности ПФР.

## ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе была достигнута основная цель – был спроектирован программный комплекс «Возвраты», посредством которого осуществляется документооборот между Управлением Пенсионного Фонда Российской Федерации и Отделением Пенсионного Фонда Российской Федерации в городе Белгород по осуществлению возвратов сумм излишне уплаченных (взысканных) денежных средств.

При выполнении выпускной квалификационной работы были решены все поставленные задачи.

В ходе выполнения выпускной квалификационной работы был проведен анализ взаимодействия подразделений Отделения Пенсионного Фонда Российской Федерации.

На основе проведенного анализа были выявлены основные проблемы, возникающие в рамках документооборота по возврату сумм излишне уплаченных (взысканных) денежных средств, а именно:

- Большие временные затраты на формирование запроса, его отправку и обработку документов;
- Качество документооборота, связанное с ошибками ручного ввода;
- Утеря информации, так как не происходит ее фиксация.

Выполнено проектирование программного комплекса «Возвраты» по обеспечению информационной поддержки документооборота по возврату плательщиком страховых взносов излишне уплаченных (взысканных) денежных средств.

Произведены расчет экономических затрат на разработку программного комплекса «Возвраты».

Реализация программного комплекса «Возвраты» для деятельности Пенсионного Фонда Российской Федерации является целесообразной, поскольку:

- разработанный исследовательский прототип показал оправданность предложенных проектных решений;

— проведенный расчет затрат показал, что их общий объем для регионального Отделения Пенсионного Фонда Российской Федерации является приемлемым.

Эффективность внедрения программного комплекса Возвраты определяет:

- снижение временных затрат на обработку документов;
- повышение качества документооборота связанных с ошибками ручного ввода;
- возможность анализа процессов возврата (взысканий) за счет хранения информации в БД.

Программный комплекс «Возвраты» окажет положительное влияние на качество взаимодействия подразделений Отделения Пенсионного Фонда Российской Федерации.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 ПФРФ – Отделение по Белгородской области [Электронный ресурс] – Режим доступа: <http://pfrf.ru>
- 2 ПФРФ – Управление в г. Белгород [Электронный ресурс] – Режим доступа: <http://pfrf.ru>
- 3 ФНС – ФНС по Белгородской области [Электронный ресурс] – Режим доступа: <http://fns.ru>
- 4 Федеральная Налоговая Служба [Электронный ресурс] – Режим доступа: <https://www.nalog.ru/rn31>
- 5 Взыскание страховых взносов ПФР [Электронный ресурс] – Режим доступа: [https://www.nalog.ru/rn44/news/activities\\_fts/6797869/](https://www.nalog.ru/rn44/news/activities_fts/6797869/)
- 6 Страховые взносы с 1 января 2017 года [Электронный ресурс] – Режим доступа: [https://www.nalog.ru/rn31/ifns/ifns31\\_02/6404330/](https://www.nalog.ru/rn31/ifns/ifns31_02/6404330/)
- 7 Wikipedia – Государственные учреждения [Электронный ресурс] – Режим доступа: [http://ru.wikipedia.org/wiki/Гос\\_учреждения](http://ru.wikipedia.org/wiki/Гос_учреждения)
- 8 Wikipedia – ПФРФ [Электронный ресурс] – Режим доступа: <http://ru.wikipedia.org/wiki/пфрф>
- 9 Buhguru – КБК по пенсионным взносам в 2017г. [Электронный ресурс] – Режим доступа: <https://buhguru.com/strahovie-vznosy/kbk-pens-vzn-2017>
- 10 Федоров, Л.В. Пенсионный фонд Российской Федерации [Текст] / Л.В. Федоров – Москва: Учебник, 2009 – 135 с.
- 11 Ерусланова, Р. И. Пенсионное обеспечение в России [Текст] / Р.И. Ерусланова – Москва: Издательско-торговая корпорация «Дашков и Ко», 2009 г. – 308 с.
- 12 Барановская, Т.П. Информационные системы и технологии [Текст] / Т.П. Барановская. – М.: ЮНИТИ, 2003. – 416 с.
- 13 Тальянов, Г.Н. Моделирование, анализ, реорганизация и автоматизация бизнес – процессов [Текст] / Г.Н. Тальянов – Москва: Финансы и статистика, 2010. – 341 с.

- 14 Методология описания бизнес-процессов [Электронный ресурс] – Режим доступа: <http://piter-soft.ru/automation/more/glossary/process/metodologiya-opisaniya-biznes-protssessov>
- 15 DFD – Википедия [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/DFD>
- 16 Моделирование бизнес-процессов [Электронный ресурс] – Режим доступа: <http://www.script-coding.com/bp.html>
- 17 Цыренов, Д.Д. Информационные технологии управления бизнес-процессами: теория и практика [Текст] / Д.Д. Цыренов. – Улан-Удэ: Издательство Бурятского госуниверситета, 2013. – 160 с.
- 18 Мир знаний [Электронный ресурс] – Режим доступа: [http://mirznanii.com/a/261560/idef0-modelirovanie\\_bp](http://mirznanii.com/a/261560/idef0-modelirovanie_bp)
- 19 Козлов, А. С. Проектирование и исследование бизнес-процессов [Текст] / А.С. Козлов – Москва: Флинта: МПСИ, 2006 – 272 с.
- 20 Уваров, С.А. Документооборот: общая концепция, теория, практика [Текст] / С.А. Уваров. – Москва: Высшая школа, 2010 – 216 с.
- 21 Ковалёв, С. М., Методы анализа и оптимизации бизнес-процессов [Текст] / С. М. Ковалёв – Консультант директора, 2005 – 111 с.
- 22 Ковалёв, С. М., Технологии анализа и оптимизации бизнес-процессов. Реинжиниринг и постоянное совершенствование [Текст] / С. М. Ковалёв, – Консультант директора, 2005 – 21 с.
- 23 Анодина, Н.Н. Документооборот в организации. [Текст] / Н.Н. Анодина, – Москва: Омега-Л, 2009 – 176 с.
- 24 Байкова, И.Ю. Документооборот. Как организовать работу с документами [Текст] / И.Ю. Байкова, – М.: Эксмо, 2010 – 288 с.
- 25 Буренина, Т.А. Информационное обеспечение документооборота [Текст] / Т.А. Буренина, – Москва: Благовест-В, 2007 – 64 с.
- 26 Грехов, А.М. Составление Технического Задания [Текст] / А.М. Грехов, – К: Кондор, 2011 – 302 с.



- 27 Что такое Техническое задание [Электронный ресурс] – Режим доступа: <http://www.informicus.ru/default.aspx?SECTION=6&id=72>
- 28 Wikipedia – Техническое задание [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/Техническое\\_задание](https://ru.wikipedia.org/wiki/Техническое_задание)
- 29 Алексеев, Ю.Е. Практикум по программированию/ Ю.Е. Алексеев, А. С. Ваулин, А.В. Куров. – МГТУ им. Н. Э. Баумана, 2008. -255 с.
- 30 IBM Knowledge Center [Электронный ресурс] – Режим доступа: <https://www.ibm.com/support/knowledgecenter/ru>
- 31 Дэвид, Г. JavaEE. Библиотека профессионала [Текст] / Г. Дэвид – Москва: ООО «И.Д. Вильямс», 2011 – 544 с.
- 32 JavaEE: Разработка web-приложения. [Электронный ресурс] – Режим доступа: <http://onedevolver.ru/article?id=7>
- 33 Wikipedia – Скриплет [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/скриплет>
- 34 Wikipedia – Сервлет [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/сервлет>
- 35 Java Platform, Enterprise Edition. [Электронный ресурс] – Режим доступа: [http://ru.wikipedia.org/wiki/Java\\_EE](http://ru.wikipedia.org/wiki/Java_EE)
- 36 Фуфаев, Д.Э. Базы данных: учебное пособие [Текст] / Д.Э. Фуфаев, – Москва: Академия, 2013 – 320 с.
- 37 Глушаков, С.В. Базы данных [Текст] / С.В. Глушаков, – Харьков: Фолио, 2012 - 504 с.
- 38 MySQL. Самая популярная в мире база данных с открытым кодом [Электронный ресурс] – Режим доступа: <https://www.oracle.com/ru/mysql/index.html>
- 39 Wikipedia – Базы данных [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/Базы\\_данных](https://ru.wikipedia.org/wiki/Базы_данных)
- 40 Базы данных MySQL [Электронный ресурс] – Режим доступа: <http://feanor184.ru/web/bazyi-dannyih-mysql.html>

## **ПРИЛОЖЕНИЯ**

## ПРИЛОЖЕНИЕ А

### Код программы

FileBlobImpl

```
package classes;
```

```
import java.io.*;
import java.sql.Blob;
import java.sql.SQLException;
```

```
public class FileBlobImpl implements Blob {
```

```
    private File file;
```

```
    public FileBlobImpl(File file) {
        this.file = file;
    }
```

```
    @Override
    public long length() throws SQLException {
        return file.length();
    }
```

```
    @Override
    public byte[] getBytes(long pos, int length) throws SQLException {
        if (pos + length >= length())
            throw new SQLException("Bad lenght");
```

```
        try {
            RandomAccessFile raf = new RandomAccessFile(file, "r");
            raf.seek(pos);
            byte[] bytes = new byte[length];
            for (int i = 0; i < bytes.length; i++) {
                bytes[i] = raf.readByte();
            }
            return bytes;
        } catch (FileNotFoundException e) {
            throw new SQLException("File is not found :" + e.toString());
        } catch (IOException e) {
            throw new SQLException("IO Exception while reading this file:" + e.toString());
        }
    }
}
```

```
    @Override
    public InputStream getBinaryStream() throws SQLException {
        try {
            return new FileInputStream(file);
        } catch (FileNotFoundException e) {
            throw new SQLException("File is not found :" + e.toString());
        }
    }
}
```

```

    }

    @Override
    public long position(byte[] pattern, long start) throws SQLException {
        throw new UnsupportedOperationException();
    }

    @Override
    public long position(Blob pattern, long start) throws SQLException {
        throw new UnsupportedOperationException();
    }

    @Override
    public int setBytes(long pos, byte[] bytes) throws SQLException {
        throw new UnsupportedOperationException();
    }

    @Override
    public int setBytes(long pos, byte[] bytes, int offset, int len) throws SQLException {
        throw new UnsupportedOperationException();
    }

    @Override
    public OutputStream setBinaryStream(long pos) throws SQLException {
        throw new UnsupportedOperationException();
    }

    @Override
    public void truncate(long len) throws SQLException {
        throw new UnsupportedOperationException();
    }

    @Override
    public void free() throws SQLException {

    }

    @Override
    public InputStream getBinaryStream(long pos, long length) throws SQLException {
        return null;
    }
}

```

Translit

package classes;

```

public class Translit {
    public static String cyr2lat(char ch){
        switch (ch){
            case 'А': return "A";
            case 'Б': return "B";
            case 'В': return "V";

```

```
case 'Г': return "G";
case 'Д': return "D";
case 'Е': return "E";
case 'Ё': return "JE";
case 'Ж': return "ZH";
case 'З': return "Z";
case 'И': return "I";
case 'Й': return "Y";
case 'К': return "K";
case 'Л': return "L";
case 'М': return "M";
case 'Н': return "N";
case 'О': return "O";
case 'П': return "P";
case 'Р': return "R";
case 'С': return "S";
case 'Т': return "T";
case 'У': return "U";
case 'Ф': return "F";
case 'Х': return "KH";
case 'Ц': return "C";
case 'Ч': return "CH";
case 'Ш': return "SH";
case 'Щ': return "JSH";
case 'Ъ': return "HH";
case 'Ы': return "IH";
case 'Ь': return "JH";
case 'Э': return "EH";
case 'Ю': return "JU";
case 'Я': return "JA";
case 'а': return "a";
case 'б': return "b";
case 'в': return "v";
case 'г': return "g";
case 'д': return "d";
case 'е': return "e";
case 'ё': return "je";
case 'ж': return "zh";
case 'з': return "z";
case 'и': return "i";
case 'й': return "y";
case 'к': return "k";
case 'л': return "l";
case 'м': return "m";
case 'н': return "n";
case 'о': return "o";
case 'п': return "p";
case 'р': return "r";
case 'с': return "s";
case 'т': return "t";
case 'у': return "u";
case 'ф': return "f";
```

```

        case 'x': return "kh";
        case 'ц': return "c";
        case 'ч': return "ch";
        case 'ш': return "sh";
        case 'щ': return "jsh";
        case 'ь': return "hh";
        case 'ы': return "ih";
        case 'ъ': return "jh";
        case 'э': return "eh";
        case 'ю': return "ju";
        case 'я': return "ja";
        default: return String.valueOf(ch);
    }
}

public static String cyr2lat(String s){
    StringBuilder sb = new StringBuilder(s.length()*2);
    for(char ch: s.toCharArray()){
        sb.append(cyr2lat(ch));
    }
    return sb.toString();
}
}

Avt

package controller;

import dao.Conn;
import model.UsersPFR;

import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.sql.*;

@WebServlet("/avt")
public class Avt extends HttpServlet {
    //здесь авторизуется вользователь и создается сессия

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws IOException {
        req.setCharacterEncoding("UTF-8");
        HttpSession session = req.getSession();
        String login = req.getParameter("login");
        String password = req.getParameter("password");

        //логика проверки в бд
        UsersPFR usersPFR = null;

```

```

PreparedStatement preparedStatement = null;
Statement statement = null;
ResultSet resultSet2;
Connection connection = null;
Conn conn = new Conn();
try {
    connection = conn.setConnection();
    String qwery ="SELECT * FROM USERS WHERE LOGIN=? and PASSWORD=?";
    preparedStatement = connection.prepareStatement(qwery);
    preparedStatement.setString(1, login);
    preparedStatement.setString(2, password);
    ResultSet resultSet = preparedStatement.executeQuery();
    while (resultSet.next()) {
        statement = connection.createStatement();
        qwery ="SELECT namerayon FROM rayon WHERE
id_rayon="+resultSet.getInt("id_rayon");
        resultSet2 = statement.executeQuery(qwery);
        resultSet2.next();
        usersPFR = new UsersPFR(resultSet.getInt("id_user"), resultSet.getString("login"),
            resultSet.getString("password"), resultSet.getString("name"),
            resultSet.getInt("id_rayon"),resultSet2.getString("namerayon"));
    }
} catch (SQLException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} finally {
    if (preparedStatement != null) {
        try {
            preparedStatement.close();
            statement.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    if (connection != null) {
        try {
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

if (usersPFR!=null){
    session.setAttribute("user", usersPFR);
    if (usersPFR.getRAYON()==2){
        resp.sendRedirect("pages/opfr.jsp");
        //System.out.println("Район 2");
    } else{
        session.setAttribute("param",1);//для определения вкладки
        resp.sendRedirect("pages/upfr.jsp");}
}

```

```

        } else {
            session.setAttribute("user", "error");
            resp.sendRedirect("pages/login.jsp");
        }
    }
}

```

DocumentPechat

package controller;

import dao.Conn;

import org.apache.poi.xwpf.usermodel.\*;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.\*;

import java.net.URLEncoder;

import java.sql.\*;

import java.util.List;

@WebServlet("/dokumenpechat")

public class DocumentPechat extends HttpServlet {

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

doPost(req, resp);

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

req.setCharacterEncoding("UTF-8");

resp.setCharacterEncoding("UTF-8");

int id = 0;

try {

id = Integer.parseInt(req.getParameter("id"));

} catch (Exception e) {

}

try {

Conn conn = new Conn();

Connection connection = conn.setConnection();

Statement statement = connection.createStatement();

ResultSet resultSet;

ResultSet resultSet2;

Statement statement2 = connection.createStatement();



```

        resultSet = statement.executeQuery("SELECT REGNUM, NAME, INN, KPP,
DATE_ZAYA, id_STATUS " +
        "from zayav where id_zayav=" + id);
resultSet.next();

Statement statement3 = connection.createStatement();
String sql = "SELECT dokument FROM dokument WHERE id_dokument=94";
ResultSet resultSet3 = statement3.executeQuery(sql);
InputStream inputStream = null;

while (resultSet3.next()) {
    Blob blob = resultSet3.getBlob("dokument");
    inputStream = blob.getBinaryStream();
}

XWPFDocument docxFile = null;
docxFile = new XWPFDocument(inputStream);
// открываем файл и считываем его содержимое в объект XWPFDocument

for (XWPFParagraph p : docxFile.getParagraphs()) {
    List<XWPFRun> runs = p.getRuns();
    if (runs != null) {
        for (XWPFRun r : runs) {
            String text = r.getText(0);
            if (text != null && text.contains("$")) {
                text = text.replace("$1", resultSet.getString("DATE_ZAYA"));
                text = text.replace("$2", resultSet.getString("REGNUM"));
                text = text.replace("$3", resultSet.getString("NAME"));
                text = text.replace("$4", resultSet.getString("INN"));
                text = text.replace("$5", resultSet.getString("KPP"));
                r.setText(text, 0);
            }
        }
    }
}

//находим таблицу
XWPFTable T = null;
for (XWPFTable tbl : docxFile.getTables()) {
    T = tbl;
}

//пришлось создать новый документ для параграфов
XWPFDocument document= new XWPFDocument();

resultSet2 = statement2.executeQuery("select v.id_kbkvid, k.kbk, v.vidplat, v.summ1/*,
v.summ2*/ " +
        "from (select id_kbkvid, id_zayav, kbk " +
        "from kbk join kbkvid on kbk.id_kbk= kbkvid.id_kbk) k join " +
        "(select id_kbkvid, id_zayav, vidplat, summ1, summ2 from vidplat " +
        "join kbkvid on vidplat.id_vidplat= kbkvid.id_vidplat) v on v.id_kbkvid = k.id_kbkvid
" +

```

```

        "where v.id_zayav = " + id + " order by v.id_kbkvid");

while (resultSet2.next()) {
    XWPFTableRow tableRowTwo = T.createRow();
    XWPFPparagraph paragraph = document.createParagraph();
    XWPFRun run = paragraph.createRun();
    run.setFontSize(13);
    run.setBold(false);
    run.setText(resultSet2.getString("k.kbk"));
    tableRowTwo.getCell(0).setParagraph(paragraph);
    XWPFPparagraph paragraph2 = document.createParagraph();
    XWPFRun run2 = paragraph2.createRun();
    run2.setFontSize(13);
    run2.setBold(false);
    run2.setText(resultSet2.getString("v.vidplat"));
    tableRowTwo.getCell(1).setParagraph(paragraph2);
    XWPFPparagraph paragraph3 = document.createParagraph();
    XWPFRun run3 = paragraph3.createRun();
    run3.setFontSize(13);
    run3.setBold(false);
    run3.setText(resultSet2.getString("v.summ1"));
    tableRowTwo.getCell(2).setParagraph(paragraph3);

    //делать что то таблица кбкПрои
}

/* http://qaru.site/questions/238852/replacing-a-text-in-apache-poi-xwpf */

ByteArrayOutputStream bo= new ByteArrayOutputStream();
docxFile.write(bo);
resp.setContentType("application/octet-stream");
String headerKey = "Content-Disposition";
String headerValue = String.format("attachment; filename=\"%s\"", "document.docx");
resp.setHeader(headerKey, headerValue);
resp.setContentLength(bo.size());
resp.getOutputStream().write(bo.toByteArray());

inputStream.close();
bo.close();

} catch (SQLException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}
}
}

```

DokumentInsert

package controller;

```

import classes.FileBlobImpl;
import dao.Conn;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.ServletInputStream;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Part;
import java.io.*;

import java.nio.file.Paths;
import java.sql.Blob;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import static classes.Translit.cyr2lat;

@WebServlet("/dokumentinsert")
@MultipartConfig
public class DokumentInsert extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        doPost(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        req.setCharacterEncoding("UTF-8");
        resp.setCharacterEncoding("UTF-8");
        String fileName = "";
        FileInputStream fileContent = null;
        String description = null; // Retrieves <input type="text" name="description">
        String descriptionRU = null;
        Part filePart = null;
        try {
            descriptionRU = req.getParameter("description");
            description = cyr2lat(descriptionRU);
        } catch (Exception e) {
        }
        try {
            filePart = req.getPart("file"); // Retrieves <input type="file" name="file">
            fileName = Paths.get(filePart.getSubmittedFileName()).getFileName().toString(); // MSIE
fix.

```

```

    } catch (Exception e) {
    }
    if (descriptionRU == null || descriptionRU.equals("")) {
        description = fileName;
        descriptionRU = cyr2lat(fileName);
    }
    else { //регулярка чтобы добавить тип .doc
        Pattern p = Pattern.compile("^.+\\.");
        Matcher m = p.matcher(fileName);
        description = m.replaceAll(description+".");
    }
    fileContent = (FileInputStream) filePart.getInputStream();

    try {
        Conn conn = new Conn();
        Connection connection = conn.setConnection();
        connection.setAutoCommit(false);
        String sql = "INSERT INTO dokument (dokument, name_dokument, name_file) VALUES
(? , ? , ?)";
        PreparedStatement stmt = connection.prepareStatement(sql);
        stmt.setBinaryStream(1, fileContent);
        stmt.setString(2, description);
        stmt.setString(3, descriptionRU);
        stmt.executeUpdate();

        connection.commit();
        fileContent.close();
        connection.close();
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
    }
}

```

## DokumentRead

```
package controller;
```

```
import dao.Conn;
```

```

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.InputStream;

```

```

import java.io.OutputStream;
import java.sql.*;

@WebServlet("/dokumentread")
public class DokumentRead extends HttpServlet{
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        doPost(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        req.setCharacterEncoding("UTF-8");
        resp.setCharacterEncoding("UTF-8");

        int id = 0;
        try {
            id = Integer.parseInt(req.getParameter("id"));
        } catch (Exception e) {
        }

        try {
            Conn conn = new Conn();
            Connection connection = conn.setConnection();
            Statement statement = connection.createStatement();
            String sql = "SELECT dokument, name_file, name_dokument FROM dokument WHERE
id_dokument="+id;
            ResultSet resultSet = statement.executeQuery(sql);
            while (resultSet.next()){
                String fileName = resultSet.getString("name_dokument");
                Blob blob = resultSet.getBlob("dokument");
                InputStream inputStream = blob.getBinaryStream();

                int fileLength = inputStream.available();

                System.out.println("fileLength = " + fileLength);

                ServletContext context = getServletContext();

                // sets MIME type for the file download
                String mimeType = context.getMimeType(fileName);
                if (mimeType == null) {
                    mimeType = "application/octet-stream";
                }
                // set content properties and header attributes for the response
                resp.setContentType(mimeType);
                resp.setContentLength(fileLength);
                String headerKey = "Content-Disposition";
                String headerValue = String.format("attachment; filename=\"%s\"", fileName);
                resp.setHeader(headerKey, headerValue);
            }
        }
    }
}

```

```

        // writes the file to the client
        OutputStream outputStream = resp.getOutputStream();
        byte[] buffer = new byte[4096];
        int bytesRead = -1;
        while ((bytesRead = inputStream.read(buffer)) != -1) {
            outputStream.write(buffer, 0, bytesRead);
        }
        inputStream.close();
        outputStream.close();
    }
    connection.close();
} catch (SQLException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}
}
}

```

Exit

package controller;

```

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;

@WebServlet("/exit")
public class Exit extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        doPost(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        req.setCharacterEncoding("UTF-8");
        HttpSession session = req.getSession();
        session.invalidate();//удалить сессию
        resp.sendRedirect("/pkv-1.0-SNAPSHOT/pages/login.jsp");
    }
}
}

```

InsOPFR

```
package controller;
```

```
import dao.Conn;
```

```
import model.UsersPFR;
```

```
import javax.servlet.RequestDispatcher;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import javax.servlet.http.HttpSession;
```

```
import java.io.IOException;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.SQLException;
```

```
@WebServlet("/insopfr")
```

```
public class InsOPFR extends HttpServlet {
```

```
    @Override
```

```
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws  
ServletException, IOException {
```

```
        req.setCharacterEncoding("UTF-8");
```

```
        resp.setCharacterEncoding("UTF-8");
```

```
        HttpSession session = req.getSession();
```

```
        UsersPFR usersPFR = (UsersPFR) session.getAttribute("user");
```

```
        //вставить данные
```

```
        String id = "";
```

```
        String nom_zap = "";
```

```
        String data_otprav = "";
```

```
        String nom_otv = "";
```

```
        String data_otv = "";
```

```
        String vsebk = "";
```

```
        int docid = 0;
```

```
        try {
```

```
            id = req.getParameter("id");
```

```
        } catch (Exception e) {
```

```
        }
```

```
        try {
```

```
            nom_zap = req.getParameter("nom_zap");
```

```
        } catch (Exception e) {
```

```
        }
```

```
        try {
```

```
            data_otprav = req.getParameter("data_otprav");
```

```
        } catch (Exception e) {
```

```
        }
```

```
        try {
```

```

    nom_otv = req.getParameter("nom_otv");
} catch (Exception e) {
}
try {
    data_otv = req.getParameter("data_otv");
} catch (Exception e) {
}
try {
    vsebk = req.getParameter("vsebk");
} catch (Exception e) {
}
try {
    docid = Integer.parseInt(req.getParameter("docid"));
} catch (Exception e) {
}

PreparedStatement preparedStatement = null;
PreparedStatement preparedStatement2 = null;
PreparedStatement preparedStatement3 = null;
Connection connection = null;
try {
    Conn conn = new Conn();
    connection = conn.setConnection();
    connection.setAutoCommit(false); // транзакция
    String qwery;

    int status;

    if(nom_otv==null){ // проверка на пустые поля
        status = 2;
        // добавить 2 штуки
        qwery = "INSERT INTO opfr ( nom_zap, data_otpr, id_zayav, id_user) " +
            "VALUES (?, ?, ?, ?)";
        preparedStatement2 = connection.prepareStatement(qwery);
        preparedStatement2.setString(1, nom_zap);
        preparedStatement2.setString(2, data_otprav);
        preparedStatement2.setString(3, id);
        preparedStatement2.setInt(4, usersPFR.getID());
        preparedStatement2.executeUpdate();
        preparedStatement2.close();
    } else {
        status = 3;

        qwery = "UPDATE opfr SET nom_otv = ?, data_otv = ?, id_dokument = ? WHERE
id_zayav = ?";
        preparedStatement2 = connection.prepareStatement(qwery);
        preparedStatement2.setString(1, nom_otv);
        preparedStatement2.setString(2, data_otv);
        preparedStatement2.setInt(3, docid);
        preparedStatement2.setString(4, id);
        preparedStatement2.executeUpdate();
        preparedStatement2.close();
    }
}

```



```

//суммы кбк изменить
String vsekbk2[] = vsekbk.split(";");
for (int i = 0; i < vsekbk2.length; i+=2) {
    qwery = "UPDATE kbkvid SET summ2 = ? WHERE id_kbkvid = ?";
    preparedStatement3 = connection.prepareStatement(qwery);
    preparedStatement3.setString(1, vsekbk2[i+1]);
    preparedStatement3.setString(2, vsekbk2[i]);
    preparedStatement3.executeUpdate();
    preparedStatement3.close();
}

}
//увеличить id

qwery = "UPDATE zayav SET id_status=? WHERE id_zayav=?";
preparedStatement = connection.prepareStatement(qwery);
preparedStatement.setInt(1, status);
preparedStatement.setString(2, id);
preparedStatement.executeUpdate();
preparedStatement.close();

connection.commit();//завершить транзакцию
} catch (SQLException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} finally {
    if (connection != null) {
        try {
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

RequestDispatcher dispatcher = req.getRequestDispatcher("pages/opfrvvod.jsp");
dispatcher.include(req, resp);
}

@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
    doPost(req, resp);
}
}

```

InsResh

package controller;

```

import dao.Conn;
import model.UsersPFR;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

@WebServlet("/insresh")
public class InsResh extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        doPost(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        req.setCharacterEncoding("UTF-8");
        resp.setCharacterEncoding("UTF-8");

        String id = "";
        String data_rech = "";
        String nom_rech = "";

        try {
            id = req.getParameter("id");
        } catch (Exception e) {
        }
        try {
            nom_rech = req.getParameter("nom_rech");
        } catch (Exception e) {
        }
        try {
            data_rech = req.getParameter("data_rech");
        } catch (Exception e) {
        }

        PreparedStatement preparedStatement = null;
        PreparedStatement preparedStatement2 = null;
        Connection connection = null;

        Conn conn = new Conn();

```

```

try {
    connection = conn.setConnection();
    connection.setAutoCommit(false);//транзакция
    String qwery;
    qwery = "INSERT INTO rech ( nom_rech, data_rech, id_zayav) " +
        "VALUES (?,?,?)";
    preparedStatement = connection.prepareStatement(qwery);
    preparedStatement.setString(1, nom_rech);
    preparedStatement.setString(2, data_rech);
    preparedStatement.setString(3, id);
    preparedStatement.executeUpdate();
    preparedStatement.close();

    qwery = "UPDATE zayav SET id_status=? WHERE id_zayav=?";
    preparedStatement2 = connection.prepareStatement(qwery);
    preparedStatement2.setInt(1, 4);
    preparedStatement2.setString(2, id);
    preparedStatement2.executeUpdate();
    preparedStatement2.close();

    connection.commit();//завершить транзакцию
} catch (SQLException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}

RequestDispatcher dispatcher = req.getRequestDispatcher("pages/upfrizfns.jsp");
dispatcher.include(req, resp);
}
}

```

Conn

```
package dao;
```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Conn {
    String userName = "root";
    String password = "654321";
    String connectionURL = "jdbc:mysql://10.41.30.247:3306/pkv?" +
/*      "verifyServerCertificate=false" +
        "&useSSL=false" +
        "&requireSSL=false" +*/
        "&useLegacyDatetimeCode=false" +
        "&" +
        "&serverTimezone=UTC";
    Connection connection;
}

```

```

public Connection setConnection() throws SQLException, ClassNotFoundException {

    Class.forName("com.mysql.jdbc.Driver");
    connection = DriverManager.getConnection(connectionURL,userName,password);

    if(connection==null){
        System.out.println("Не удалось подключиться к БД");
    } else{
        System.out.println("Подключение к БД успешно");
    }
    return connection;
}
}

```

UsersPFR

```
package model;
```

```

public class UsersPFR {
    private int ID;
    private String LOGIN;
    private String NAME;
    private String PASSWORD;
    private int RAYON;
    private String NAMERAYON;

    public UsersPFR(int ID, String LOGIN, String PASSWORD, String NAME, int RAYON, String
NAMERAYON) {
        this.ID = ID;
        this.LOGIN = LOGIN;
        this.NAME = NAME;
        this.PASSWORD = PASSWORD;
        this.RAYON = RAYON;
        this.NAMERAYON = NAMERAYON;
    }

    public int getID() {
        return ID;
    }

    public void setID(int ID) {
        this.ID = ID;
    }

    public String getLOGIN() {
        return LOGIN;
    }

    public void setLOGIN(String LOGIN) {
        this.LOGIN = LOGIN;
    }
}

```

```

public String getName() {
    return NAME;
}

public void setName(String NAME) {
    this.NAME = NAME;
}

public String getPassword() {
    return PASSWORD;
}

public void setPassword(String PASSWORD) {
    this.PASSWORD = PASSWORD;
}

public int getRAYON() {
    return RAYON;
}

public void setRAYON(int RAYON) {
    this.RAYON = RAYON;
}

public String getNAMERAYON() {
    return NAMERAYON;
}

public void setNAMERAYON(String NAMERAYON) {
    this.NAMERAYON = NAMERAYON;
}

@Override
public String toString() {
    return "UsersPFR{" +
        "ID=" + ID +
        ", LOGIN=" + LOGIN + "\" +
        ", NAME=" + NAME + "\" +
        ", PASSWORD=" + PASSWORD + "\" +
        ", RAYON=" + RAYON +
        ", NAMERAYON=" + NAMERAYON +
        "}";
}

}

```

VidKbk

package model;

import java.util.Objects;

```

public class VidKbk {
    private int kbk;
    private int vid;
    private String summ;
    private String summ2;

    public String getSumm2() {
        return summ2;
    }

    public void setSumm2(String summ2) {
        this.summ2 = summ2;
    }

    public int getKbk() {
        return kbk;
    }

    public void setKbk(int kbk) {
        this.kbk = kbk;
    }

    public int getVid() {
        return vid;
    }

    public void setVid(int vid) {
        this.vid = vid;
    }

    public String getSumm() {
        return summ;
    }

    public void setSumm(String summ) {
        this.summ = summ;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        VidKbk vidKbk = (VidKbk) o;
        return kbk == vidKbk.kbk &&
            vid == vidKbk.vid &&
            Objects.equals(summ, vidKbk.summ) &&
            Objects.equals(summ2, vidKbk.summ2);
    }

    @Override
    public int hashCode() {

```

```

        return Objects.hash(kbk, vid, summ, summ2);
    }

    public VidKbk(String kbk, String vid, String summ) {
        this.kbk = Integer.parseInt(kbk);
        this.vid = Integer.parseInt(vid);
        this.summ = summ;
        this.summ2 = "";
    }

    public VidKbk(String kbk, String vid, String summ, String summ2) {
        this.kbk = Integer.parseInt(kbk);
        this.vid = Integer.parseInt(vid);
        this.summ = summ;
        this.summ2 = summ2;
    }
}

```

dokument.jsp

```

<% @ page import="java.sql.Connection" %>
<% @ page import="dao.Conn" %>
<% @ page import="java.sql.Statement" %>
<% @ page import="java.sql.ResultSet" %>
<% @ page import="java.sql.SQLException" %><%--
Created by IntelliJ IDEA.
User: 041BratchinAV
Date: 17.04.2018
Time: 14:52
To change this template use File | Settings | File Templates.
--%>

<%--Таблица документов--%>
<% @ page contentType="text/html;charset=UTF-8" language="java" %>
<a class="modalbox2" id="a_modal_form_insert" href="#">Добавить</a>
<table border="1" style="table-layout: fixed" width="600px" align="center" cellspacing="0">
    <tr>
        <td align="center">Название</td>
        <td align="center">Файл</td>
        <td align="center">Выбор</td>
    </tr>
</table>
<%
    Connection connection = null;
    try {
        Conn conn = new Conn();
        connection = conn.setConnection();
        Statement statement = connection.createStatement();
        ResultSet resultSet;
        resultSet = statement.executeQuery("SELECT id_dokument, name_dokument, name_file
from dokument");
        while (resultSet.next()) {
%>

```

```

        <tr id="<%=resultSet.getInt("id_dokument")%>">
            <td style="overflow: hidden"
align="center"><%=resultSet.getString("name_dokument")%></td>
            <td style="overflow: hidden" align="center"
class="namefile"><%=resultSet.getString("name_file")%></td>
            <td style="overflow: hidden" align="center"><a class="vibdoc"
name="<%=resultSet.getInt("id_dokument")%>" href="#">Выбрать</a></td>
        </tr>
    <%
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        if (connection != null) {
            try {
                connection.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
    %>
</table>

```

login.jsp

```

<%--
Created by IntelliJ IDEA.
User: 041BratchinAV
Date: 01.02.2018
Time: 9:12
To change this template use File | Settings | File Templates.
--%>
<% @ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>ПК Возвраты</title>
</head>
<body>
<table cellpadding="2" align="center">
    <tr><th colspan="2">Авторизация пользователя<br></th></tr>
    <form action='../avt' method='post'>
        <tr><td>Логин:</td>    <td><input type='text' name='login'><br></td></tr>
        <tr><td>Пароль:</td>    <td><input type='password' name='password'><br></td></tr>
        <% if(session.getAttribute("user")!=null && session.getAttribute("user").equals("error")){ //
        %>
        <tr><td colspan="2">Неверный логин и/или пароль!</td></tr>
        <% }%>
        <tr><td></td>    <td><input type='submit' name='Вход'></td></tr>
    </form>

```



```
</table>
</body>
</html>
```

opfr.jsp

```
<% @ page import="dao.Conn" %>
<% @ page import="java.sql.Connection" %>
<% @ page import="java.sql.Statement" %>
<% @ page import="java.sql.ResultSet" %>
<% @ page import="model.UsersPFR" %><%--
Created by IntelliJ IDEA.
User: 041BratchinAV
Date: 28.03.2018
Time: 9:12
To change this template use File | Settings | File Templates.
--%>
<% @ page contentType="text/html;charset=UTF-8" language="java" %>
<% UsersPFR usersPFR = (UsersPFR) session.getAttribute("user");%> <%--сессия--%>
<html>
<head>
  <title>ПК Возвраты</title>
</head>
<style>

  #modal_form, #div_modal_form_insert {
    width: 600px;
    height: 600px; /* Размеры должны быть фиксированы */
    border-radius: 5px;
    border: 3px #000 solid;
    background: #fff;
    position: fixed; /* чтобы окно было в видимой зоне в любом месте */
    top: 45%; /* отступаем сверху 45%, остальные 5% подвинет скрипт */
    left: 50%; /* половина экрана слева */
    margin-top: -300px;
    margin-left: -300px; /* тут вся магия центровки css, отступаем влево и вверх минус
половину ширины и высоты соответственно =) */
    display: none; /* в обычном состоянии окна не должно быть */
    opacity: 0; /* полностью прозрачно для анимирования */
    z-index: 5; /* окно должно быть наиболее большем слое */
    padding: 20px 10px;
  }

  #div_modal_form_insert {
    z-index: 7; /* окно должно быть наиболее большем слое */
    width: 400px;
    height: 100px; /* Размеры должны быть фиксированы */
    margin-left: -200px; /* тут вся магия центровки css, отступаем влево и вверх минус
половину ширины и высоты соответственно =) */
    margin-top: -50px;
  }

```

```

/* Кнопка закрыть для тех кто в танке) */
#modal_form #modal_close, #div_modal_form_insert #modal_close2 {
    width: 21px;
    height: 21px;
    position: absolute;
    top: 10px;
    right: 10px;
    cursor: pointer;
    display: block;
}

/* Подложка */
#overlay, #overlay2 {
    z-index: 3; /* подложка должна быть выше слоев элементов сайта, но ниже слоя
модального окна */
    position: fixed; /* всегда перекрывает весь сайт */
    background-color: #000; /* черная */
    opacity: 0.8; /* но немного прозрачна */
    -moz-opacity: 0.8; /* фикс прозрачности для старых браузеров */
    filter: alpha(opacity=80);
    width: 100%;
    height: 100%; /* размером во весь экран */
    top: 0; /* сверху и слева 0, обязательные свойства! */
    left: 0;
    cursor: pointer;
    display: none; /* в обычном состоянии её нет) */
}

#overlay2 {
    z-index: 6; /* подложка должна быть выше слоев элементов сайта, но ниже слоя
модального окна */
}

li {
    color: #1740ff;
    display: inline-block; /* Строчно-блочные элементы */
    /*background: #CA181A; /* Цвет фона */!*/
    margin-right: 50px; /* Расстояние между пунктами меню */
}
li:hover {
    /*background: #333; /* Цвет фона при наведении курсора мыши */!*/
    cursor: pointer;
    text-decoration-line: underline;
}
.fw {
    font-weight: bold;
}
</style>
<script type="text/javascript" src="../../lib/jquery-3.3.1.min.js"></script>
<script type="text/javascript">

```

```

var ANAMEDOC=0;

function closebutdocins () {
    //alert("sasdf");
    $('#div_modal_form_insert')
        .animate({opacity: 0, top: '45%'}, 200, // плавно меняем прозрачность на 0 и
одновременно двигаем окно вверх
        function () { // после анимации
            $(this).css('display', 'none'); // делаем ему display: none;
            $('#overlay2').fadeOut(400); // скрываем подложку
        }
    );
}

function closebutdocins2 () {
    $('#modal_form')
        .animate({opacity: 0, top: '45%'}, 200, // плавно меняем прозрачность на 0 и
одновременно двигаем окно вверх
        function () { // после анимации
            $(this).css('display', 'none'); // делаем ему display: none;
            $('#overlay').fadeOut(400); // скрываем подложку
        }
    );
}

$(document).ready(function () {
    $('#st1').addClass("fw");

    // обработка и отправка AJAX запроса при клике на кнопку upload_files
    $('#input[name="butdocins"]').on( 'click', function( event ){
        var data = new FormData();
        jQuery.each(jQuery('input[name="file"]')[0].files, function(i, file) {
            data.append('file', file);
        });
        data.append('description', $('#input[name="description"]').val());
        jQuery.ajax({
            url: './dokumentinsert',
            data: data,
            cache: false,
            contentType: false,
            processData: false,
            method: 'POST',
            type: 'POST', // For jQuery < 1.9
            success: function(){
                //очистить
                $('#input[name="description"]').val("");
                $('#input[name="file"]').val("");

                $("div[name='dliazameny']").load("/pkv-1.0-SNAPSHOT/pages/dokument.jsp");
                closebutdocins();
            }
        });
    });
});

```

```

$("div[name='dliazameny']").on('click','a',function () {
    //alert(this.getAttribute("id"));
    if(this.getAttribute("id")==="a_modal_form_insert"){
        event.preventDefault(); // выключаем стандартную роль элемента
        $('#overlay2').fadeIn(400, // сначала плавно показываем темную подложку
        function () { // после выполнения предыдущей анимации
            $('#div_modal_form_insert')
                .css('display', 'block') // убираем у модального окна display: none;
                .animate({opacity: 1, top: '50%'}, 200); // плавно прибавляем прозрачность
одновременно со съезжанием вниз
        });
    }
    if(this.getAttribute("class")==="vibdoc"){
        //alert(this.getAttribute("name"));
        var a = "tr#" + this.getAttribute("name") + " td.namefile";
        //alert($(a).text());
        $("a[name='"+ANAMEDOC+"']").attr("id",this.getAttribute("name"));
        $("a[name='"+ANAMEDOC+"']").text($(a).text());
        closebutdocins2();
    }
});

/*
$('#a_modal_form_insert').click(function (event) { // ловим клик по ссылке с id="go"
    event.preventDefault(); // выключаем стандартную роль элемента
    $('#overlay2').fadeIn(400, // сначала плавно показываем темную подложку
    function () { // после выполнения предыдущей анимации
        $('#div_modal_form_insert')
            .css('display', 'block') // убираем у модального окна display: none;
            .animate({opacity: 1, top: '50%'}, 200); // плавно прибавляем прозрачность
одновременно со съезжанием вниз
        });
    });*/
/* Закрытие модального окна, тут делаем то же самое но в обратном порядке */
$('#modal_close2, #overlay2').click(function (event) {closebutdocins2()});

$("table[name='table1']").on('click', 'a', function () {
    if (this.getAttribute("class") == "go") {
        //alert(this.getAttribute("name"));
        ANAMEDOC = this.getAttribute("name");
        event.preventDefault(); // выключаем стандартную роль элемента
        $('#overlay').fadeIn(400, // сначала плавно показываем темную подложку
        function () { // после выполнения предыдущей анимации
            $('#modal_form')
                .css('display', 'block') // убираем у модального окна display: none;
                .animate({opacity: 1, top: '50%'}, 200); // плавно прибавляем прозрачность
одновременно со съезжанием вниз
            });
        }
        if (this.getAttribute("class") == "documentinsert") {
            alert(this.getAttribute("name"));
        }
    });

```

```

/* $(a.go).click(function (event) { // ловим клик по ссылке с id="go"
//alert(this.getAttribute("name"));
ANAMEDOC = this.getAttribute("name");
event.preventDefault(); // выключаем стандартную роль элемента
$('#overlay').fadeIn(400, // сначала плавно показываем темную подложку
function () { // после выполнения предыдущей анимации
    $('#modal_form')
        .css('display', 'block') // убираем у модального окна display: none;
        .animate({ opacity: 1, top: '50%' }, 200); // плавно прибавляем прозрачность
одновременно со съезжанием вниз
    });
});*/
/* Закрытие модального окна, тут делаем то же самое но в обратном порядке */
$('#modal_close, #overlay').click(function () { // ловим клик по крестику или подложке
    closebutdocins2();
});

$("table[name='table1']").on('click', 'input', function () {

    if (this.getAttribute("class") == "button1") {
        var a = [];
        a = this.getAttribute("name").valueOf().split(",");
        var s = "";
        for (var i = 0; i < a.length; i++) {
            var ss = "input[name=" + a[i] + "]";
            var ssss = "table[name=" + this.getAttribute("id") + "]" + ss;
            s += a[i] + "=" + $(ssss).val() + "&";
        }
        s += ("id=" + this.getAttribute("id"));

        var t = "table[name=" + this.getAttribute("id") + "] table input:text";
        var list = $(t);
        var vsekbk = "";
        for (var i = 0; i < list.length; i++) {
            vsekbk += $(list[i]).attr('name') + ";" + $(list[i]).val() + ";";
        }
        s += "&vsekbk=" + vsekbk + "&docid=" + $("table[name=" + this.getAttribute("id") + "]
a.go").attr("id");
        $("table[name='table1']").load("../insopfr", s);
    }

/*
    if (this.getAttribute("class") == "button1") {
        var a = [];
        a = this.getAttribute("name").valueOf().split(",");
        var s = "";
        for (var i = 0; i < a.length; i++) {
            var ss = "input[name=" + a[i] + "]";
            var ssss = "table[name=" + this.getAttribute("id") + "]" + ss;
            s += a[i] + "=" + $(ssss).val() + "&";
        }
        s += ("id=" + this.getAttribute("id"));
    }
*/

```

```

var t = "table[name='" + this.getAttribute("id") + "]" table input:text";
var list = $(t);
var vsebk = "";
for (var i = 0; i < list.length; i++) {
    vsebk += $(list[i]).attr('name') + ";" + $(list[i]).val() + ";";
}
s += "&vsebk=" + vsebk + "&docid=" + $("table[name='" + this.getAttribute("id") + "]"
a.go").attr("id");
$("table[name='table1']").load("../insopfr", s);
}*/
});

```

```

$("ul").on('click','li',function () {

```

```

if(this.getAttribute("id")==="stn"){
    $("table[name='table1']").load("opfrvod.jsp","str="+$("#st1").html());
    $("#st1").html("1");
    $("#st2").html("2");
    $("#st3").html("3");
    $("#st4").html("4");
    $("#st5").html("5");
    $("#st1").addClass("fw");
    $("#st2").removeClass("fw");
    $("#st3").removeClass("fw");
    $("#st4").removeClass("fw");
    $("#st5").removeClass("fw");
}
if(this.getAttribute("id")==="st1"){
    $("table[name='table1']").load("opfrvod.jsp","str="+$("#st1").html());
    $("#st1").addClass("fw");
    $("#st2").removeClass("fw");
    $("#st3").removeClass("fw");
    $("#st4").removeClass("fw");
    $("#st5").removeClass("fw");
}
if(this.getAttribute("id")==="st2"){
    $("table[name='table1']").load("opfrvod.jsp","str="+$("#st2").html());
    $("#st1").removeClass("fw");
    $("#st2").addClass("fw");
    $("#st3").removeClass("fw");
    $("#st4").removeClass("fw");
    $("#st5").removeClass("fw");
}
if(this.getAttribute("id")==="st3"){
    $("table[name='table1']").load("opfrvod.jsp","str="+$("#st3").html());
    $("#st1").removeClass("fw");
    $("#st2").removeClass("fw");
    $("#st3").addClass("fw");
    $("#st4").removeClass("fw");
    $("#st5").removeClass("fw");
}
}

```

```

if(this.getAttribute("id")== "st4"){
    $("table[name='table1']").load("opfrvvod.jsp","str="+$("#st4").html());
    $("#st1").removeClass("fw");
    $("#st2").removeClass("fw");
    $("#st3").removeClass("fw");
    $("#st4").addClass("fw");
    $("#st5").removeClass("fw");
}
if(this.getAttribute("id")== "st5"){
    $("table[name='table1']").load("opfrvvod.jsp","str="+$("#st5").html());
    $("#st1").removeClass("fw");
    $("#st2").removeClass("fw");
    $("#st3").removeClass("fw");
    $("#st4").removeClass("fw");
    $("#st5").addClass("fw");
}
if(this.getAttribute("id")== "std"){
    var pos = $("#st5").html();
    $("table[name='table1']").load("opfrvvod.jsp","str="+1+(+pos));
    $("#st1").html(pos-1);
    $("#st2").html(pos);
    $("#st3").html(+pos+1);
    $("#st4").html(+pos+2);
    $("#st5").html(+pos+3);
    $("#st1").removeClass("fw");
    $("#st2").removeClass("fw");
    $("#st3").addClass("fw");
    $("#st4").removeClass("fw");
    $("#st5").removeClass("fw");
}
});
});
</script>

```

```
<body>
```

```
<jsp:include page="tlog.jsp"/>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<a href="/pkv-1.0-SNAPSHOT/pages/stat.jsp">Статистика</a>
```

```
<table border="1" align="center" cellspacing="0" name="table1" width="100%">
```

```
<%String pg = "opfrvvod.jsp";%>
```

```
<jsp:include page="<%=pg%>">
```

```
</table>
```

```
<ul align="center">
```

```

<li id="stn">В начало</li>
<li id="st1">1</li>
<li id="st2">2</li>
<li id="st3">3</li>
<li id="st4">4</li>
<li id="st5">5</li>
<li id="std">Дальше</li>
</ul>

```

```

<%-- Модальные окна --%>
<div id="modal_form"><!-- Само окно -->
  <span id="modal_close">X</span> <!-- Кнопка закрыть -->
  <!-- Тут любое содержимое -->
  <div name="dliazameny" >
    <jsp:include page="dokument.jsp"/>
  </div>
</div>
<div id="overlay"></div><!-- Подложка -->

<div id="div_modal_form_insert"><!-- Само окно -->
  <span id="modal_close2">X</span> <!-- Кнопка закрыть -->
  <form id='formx' enctype='multipart/form-data' method='post' action='../dokumentinsert'>
    <table>
      <tr>
        <td>Имя файла:</td>
        <td>
          <input type='text' name='description'>
        </td>
      </tr>
      <tr>
        <td>Выбрать файл:</td>
        <td>
          <input type='file' name='file' accept=".docx,.doc">
        </td>
      </tr>
    </table>
    <p>
      <input type='button' value='Send' name="butdocins">
    </p>
  </form>
</div>
<div id="overlay2"></div><!-- Подложка -->

</body>
</html>

```

opfrvvod.jsp

```

<% @ page import="model.UsersPFR" %>
<% @ page import="dao.Conn" %>
<% @ page import="java.sql.Connection" %>

```



```

<% @ page import="java.sql.Statement" %>
<% @ page import="java.sql.ResultSet" %>
<% @ page import="java.sql.SQLException" %><%--
Created by IntelliJ IDEA.
User: 041BratchinAV
Date: 30.03.2018
Time: 11:53
To change this template use File | Settings | File Templates.
--%>
<% @ page contentType="text/html;charset=UTF-8" language="java" %>

<% UsersPFR usersPFR = (UsersPFR) session.getAttribute("user");%>

<tr>
  <th align="center">ΥΠΦΡ</th>
  <th align="center">ΟΠΦΡ</th>
</tr>
<%
int str = 1;
try {
  str = Integer.parseInt(request.getParameter("str"));
} catch (Exception e) {
}
Connection connection = null;
try {
  Conn conn = new Conn();
  connection = conn.setConnection();
  Statement statement = connection.createStatement();
  ResultSet resultSet;
  resultSet = statement.executeQuery("SELECT id_zayav, id_status from zayav");
  int kolvonastr = 10;
  int strnach = kolvonastr * str - kolvonastr;
  int strkon = kolvonastr * str;
  int ii=1;
  while (resultSet.next()) {
    if(ii>=strnach && ii<=strkon){
%>

<tr id="<%=resultSet.getInt("id_zayav")%>">

  <td width="50%" align="center" name="upfr">
    <%String ss = "otupfr.jsp?id_zayav=" + resultSet.getInt("id_zayav");%>
    <jsp:include page="<%=ss%>" />
  </td>

  <%-- ОТВЕТ --%>
  <td width="50%" align="center" name="opfr">
    <%ss = "otopfr.jsp?id_zayav=" + resultSet.getInt("id_zayav");%>
    <jsp:include page="<%=ss%>" />
  </td>

</tr>

```

```

<%
    }ii++;
    }
} catch (SQLException e) {
    e.printStackTrace();
} finally {
    if (connection != null) {
        try {
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
%>

```

otopfr.jsp

```

<% @ page import="dao.Conn" %>
<% @ page import="java.sql.Connection" %>
<% @ page import="java.sql.Statement" %>
<% @ page import="java.sql.ResultSet" %>
<% @ page contentType="text/html;charset=UTF-8" language="java" %>

<%
    int id_zayav = 0;
    int id_status = 0;
    try {
        id_zayav = Integer.parseInt(request.getParameter("id_zayav"));
    } catch (Exception e) {
    }

    Conn conn = new Conn();
    Connection connection = conn.setConnection();
    Statement statement = connection.createStatement();
    ResultSet resultSet;
    ResultSet resultSet2;
    Statement statement2 = connection.createStatement();
    resultSet = statement.executeQuery("SELECT id_opfr, nom_zap, data_otpr, nom_otv, data_otv,
name_file, opfr.id_dokument from opfr " +
        "left join dokument on opfr.id_dokument=dokument.id_dokument where id_zayav=" +
id_zayav);
    resultSet.next();

    Statement statement3 = connection.createStatement();
    ResultSet resultSet3;
    resultSet3 = statement3.executeQuery("SELECT id_status " +
        "from zayav where id_zayav=" + id_zayav);
    resultSet3.next();
    id_status = resultSet3.getInt("id_status");
    if (id_status >= 3) {
        //Просто вывод

```

```

%>
<table align="center" cellspacing="0">
  <tr>
    <td>Номер запроса: <%=resultSet.getInt("nom_zap")%>
    </td>
  </tr>
  <tr>
    <td>Дата направления в ФНС: <%=resultSet.getString("data_otpr")%>
    </td>
  </tr>
  <tr>
    <td>Номер ответа от ФНС: <%=resultSet.getString("nom_otv")%>
    </td>
  </tr>
  <tr>
    <td>Дата ответа от ФНС: <%=resultSet.getString("data_otv")%>
    </td>
  </tr>
  <tr>
    <td>Документ:
      <a class="documentinsert" name="<%=resultSet.getString("opfr.id_dokument")%>"
        href=" ../dokumentread?id=<%=resultSet.getString("opfr.id_dokument")%>"
download>
      <%=resultSet.getString("name_file")%></a>
    </td>
  </tr>
<%
  resultSet2 = statement2.executeQuery("select v.id_kbkvid, k.kbk, v.vidplat, v.summ2 " +
    "from (select id_kbkvid, id_zayav, kbk " +
    "from kbk join kbkvid on kbk.id_kbk= kbkvid.id_kbk) k join " +
    "(select id_kbkvid, id_zayav, vidplat, summ1, summ2 from vidplat " +
    "join kbkvid on vidplat.id_vidplat= kbkvid.id_vidplat) v on v.id_kbkvid = k.id_kbkvid "
+
    "where v.id_zayav = " + id_zayav + " order by v.id_kbkvid");
  String skbk = "";
  String svidplat = "";
  String ssumm = "";
  while (resultSet2.next()) {
    skbk += resultSet2.getString("k.kbk") + "<br>";
    svidplat += resultSet2.getString("v.vidplat") + "<br>";
    ssumm += resultSet2.getString("v.summ2") + "<br>";
  }
%>
<tr>
  <td>
    <table name="kbk" border="1" align="center" cellspacing="0">
      <tr>
        <td align="center">КБК</td>
        <td align="center">Вид платежа</td>
        <td align="center">Сумма</td>
      </tr>
      <tr>

```

```

        <td><%=skbk%>
        </td>
        <td><%=svidplat%>
        </td>
        <td><%=ssumm%>
        </td>
    </tr>
</table>
</td>
</tr>
</table>

<%

} else {
    //для ввода данных
    String buttonname = "nom_zap,data_otprav";
%>

<table align="center" cellspacing="0" name='<%=id_zayav%>'>

    <%
        if (id_status == 1) {
    %>
    <tr>
        <td>Номер запроса:</td>
        <td><input type='text' name='nom_zap'></td>
    </tr>
    <tr>
        <td>Дата направления в ФНС:</td>
        <td><input type='date' name='data_otprav'></td>
    </tr>
    <% }
        if (id_status != 1) { //
            buttonname = "nom_otv,data_otv";
    %>
    <tr>
        <td>Номер ответа от ФНС:</td>
        <td><input type='text' name='nom_otv'></td>
    </tr>
    <tr>
        <td>Дата ответа от ФНС:</td>
        <td><input type='date' name='data_otv'></td>
    </tr>
    <tr>
        <td>Документ:</td>
        <td><a class="go" name="<%=id_zayav%>" href="#">Выбрать</a></td>
    </tr>
    <%
        resultSet2 = statement2.executeQuery("select v.id_kbkvid, k.kbk, v.vidplat, v.summ2 " +
            "from (select id_kbkvid, id_zayav, kbk " +
            "from kbk join kbkvid on kbk.id_kbk= kbkvid.id_kbk) k join " +

```

```

        "(select id_kbkvid, id_zayav, vidplat, summ1, summ2 from vidplat " +
        "join kbkvid on vidplat.id_vidplat= kbkvid.id_vidplat) v on v.id_kbkvid = k.id_kbkvid "
+
        "where v.id_zayav = " + id_zayav + " order by v.id_kbkvid");
String skbk = "";
String svidplat = "";
String input = "";
while (resultSet2.next()) {
    skbk += resultSet2.getString("k.kbk") + "<br>";
    svidplat += resultSet2.getString("v.vidplat") + "<br>";
    input += "<input type='text' name='" + resultSet2.getString("v.id_kbkvid") + "'>" + "<br>";
    //buttonname += ("," + resultSet2.getString("v.id_kbkvid"));
}
%>
<tr>
<td colspan="2">
<table border="1" align="center" cellspacing="0">
<tr>
<td align="center">КБК</td>
<td align="center">Вид платежа</td>
<td align="center">Сумма</td>
</tr>
<tr>
<td><%=skbk%>
</td>
<td><%=svidplat%>
</td>
<td><%=input%>
</td>
</tr>
</table>
</td>
</tr>
<% }%>
<tr>
<td>
<a class="documentpechat" href=" ../dokumenpechat?id=<%=id_zayav%>"
download>Заявление</a>
</td>
<td>
<input type='button' id='<%=id_zayav%>' value="Добавить" name="<%=buttonname%>"
class="button1">
</td>
</tr>
</table>

<%
}
%>

```

otupfr.jsp

```

<% @ page import="dao.Conn" %>
<% @ page import="java.sql.Connection" %>
<% @ page import="java.sql.Statement" %>
<% @ page import="java.sql.ResultSet" %>

<% @ page contentType="text/html;charset=UTF-8" language="java" %>

<%
    int id_zayav = 0;
    try {
        id_zayav = Integer.parseInt(request.getParameter("id_zayav"));
    } catch (Exception e) {
    }

    Conn conn = new Conn();
    Connection connection = conn.setConnection();
    Statement statement = connection.createStatement();
    ResultSet resultSet;
    ResultSet resultSet2;
    Statement statement2 = connection.createStatement();
    resultSet = statement.executeQuery("SELECT REGNUM, NAME, INN, KPP, DATE_ZAYA,
id_STATUS " +
        "from zayav where id_zayav="+id_zayav);
    resultSet.next();
%>
<table align="center" cellspacing="0" >
    <tr>
        <td>Регистрационный номер: <%=resultSet.getInt("REGNUM")%>
        </td>
    </tr>
    <tr>
        <td>Наименование: <%=resultSet.getString("NAME")%>
        </td>
    </tr>
    <tr>
        <td>ИНН: <%=resultSet.getString("INN")%>
        </td>
    </tr>
    <tr>
        <td>КПП: <%=resultSet.getString("KPP")%>
        </td>
    </tr>
    <tr>
        <td>Дата заявления: <%=resultSet.getDate("DATE_ZAYA")%>
        </td>
    </tr>
    <tr>
        <td>Статус: <%=resultSet.getString("id_STATUS")%>
        </td>
    </tr>
<%
    resultSet2 = statement2.executeQuery("select v.id_kbkvid, k.kbk, v.vidplat, v.summ1/*,

```

```

v.summ2*/ " +
    "from (select id_kbkvid, id_zayav, kbk " +
    "from kbk join kbkvid on kbk.id_kbk= kbkvid.id_kbk) k join " +
    "(select id_kbkvid, id_zayav, vidplat, summ1, summ2 from vidplat " +
    "join kbkvid on vidplat.id_vidplat= kbkvid.id_vidplat) v on v.id_kbkvid = k.id_kbkvid "
+
    "where v.id_zayav = " + id_zayav + " order by v.id_kbkvid");
String skbk = "";
String svidplat = "";
String ssumm = "";
while (resultSet2.next()) {
    skbk += resultSet2.getString("k.kbk") + "<br>";
    svidplat += resultSet2.getString("v.vidplat") + "<br>";
    ssumm += resultSet2.getString("v.summ1") + "<br>";
}
%>
<tr>
<td>
<table border="1" align="center" cellspacing="0">
<tr>
<td align="center">КБК</td>
<td align="center">Вид платежа</td>
<td align="center">Сумма</td>
</tr>
<tr>
<td><%=skbk%>
</td>
<td><%=svidplat%>
</td>
<td><%=ssumm%>
</td>
</tr>
</table>
</td>
</tr>
</table>

```

rech.jsp

```

<% @ page import="dao.Conn" %>
<% @ page import="java.sql.Connection" %>
<% @ page import="java.sql.ResultSet" %>
<% @ page import="java.sql.Statement" %><%--
Created by IntelliJ IDEA.
User: 041BratchinAV
Date: 29.03.2018
Time: 16:27
To change this template use File | Settings | File Templates.
--%>
<% @ page contentType="text/html;charset=UTF-8" language="java" %>
<%
    int id_zayav = 0;

```

```

int id_status = 0;
try {
    id_zayav = Integer.parseInt(request.getParameter("id_zayav"));
} catch (Exception e) {
}

Conn conn = new Conn();
Connection connection = conn.setConnection();

Statement statement3 = connection.createStatement();
ResultSet resultSet3;
resultSet3 = statement3.executeQuery("SELECT id_status " +
    "from zayav where id_zayav=" + id_zayav);
resultSet3.next();
id_status = resultSet3.getInt("id_status");
if (id_status == 3) {
    //ВВОД
    %>
    <table align="center" cellspacing="0" class="<%=id_zayav%>">
        <tr>
            <td>Номер решения:</td>
            <td><input type='text' name='nom_rech'></td>
        </tr>
        <tr>
            <td>Дата решения:</td>
            <td><input type='date' name='data_rech'></td>
        </tr>
        <tr>
            <td>
                <input type='button' value="Решение не принято" name="rechnull"
                id="<%=id_zayav%>">
            </td>
            <td>
                <input type='button' value="Добавить решения" name="buttonrech"
                id="<%=id_zayav%>">
            </td>
        </tr>
    </table>
    %>
} else if (id_status > 3) {
    Statement statement = connection.createStatement();
    ResultSet resultSet;
    resultSet = statement.executeQuery("SELECT nom_rech, data_rech " +
        "from rech where id_zayav=" + id_zayav);
    if (resultSet.next()) {
        if (resultSet.getString("nom_rech").equals("0")) {
            %>
            <table align="center" cellspacing="0">
                <tr>
                    <td>Решение не принято</td>
                </tr>
            </table>

```



```

    </tr>
</table>
<%
} else {

%>
<table align="center" cellspacing="0">
  <tr>
    <td>Номер решения:</td>
    <td><%=resultSet.getString("nom_rech")%>
    </td>
  </tr>
  <tr>
    <td>Дата решения:</td>
    <td><%=resultSet.getString("data_rech")%>
    </td>
  </tr>
</table>

<%
}
} else {
%>

<table align="center" cellspacing="0">
  <tr>
    <td>Решение не принято.</td>
  </tr>
</table>

<%
}
}

%>

```

stat.jsp

```

<% @ page import="dao.Conn" %>
<% @ page import="java.sql.Connection" %>
<% @ page import="java.sql.Statement" %>
<% @ page import="java.sql.ResultSet" %><!--
  Created by IntelliJ IDEA.
  User: 041BratchinAV
  Date: 23.05.2018
  Time: 14:17
  To change this template use File | Settings | File Templates.
--%>
<% @ page contentType="text/html; charset=UTF-8" language="java" %>

<html>
<head>

```

```

<title>ПК Возвраты</title>
</head>
<script type="text/javascript" src="../lib/jquery-3.3.1.min.js"></script>
<jsp:include page="tlog.jsp"/>
<br>
<br>
<br>
<table align="center" width="100%" border="1" cellspacing="0">
  <tr>
    <th colspan="8">Сводный отчет по возвратам</th>
  </tr>
  <tr>
    <td>№</td>
    <td>Код района</td>
    <td>Наименование района</td>
    <td>Кол-во</td>
    <td>Сумма</td>
    <td>Кол-во запросов в ФНС</td>
    <td>Кол-во ответов от ФНС</td>
    <td>Кол-во решений</td>
  </tr>
  <%
    System.out.println("asdfasdf");
    Conn conn;
    conn = new Conn();
    Connection connection = conn.setConnection();
    Statement statement = connection.createStatement();
    ResultSet resultSet;
    resultSet = statement.executeQuery("select kod, namerayon, count(id_rayon) kolvo, sum(ss)
summa," +
    " sum(if(id_status=1,1,0)) s1," +
    " sum(if(id_status=2,1,0)) s2," +
    " sum(if(id_status=3,1,0)) s3," +
    " sum(if(id_status=4,1,0)) s4" +
    " from" +
    "(select zayav.id_zayav, zayav.id_users, zayav.id_status, r.namerayon, kod, r.id_rayon
" +
    "from zayav join" +
    "(select users.id_user, rayon.namerayon, kod, rayon.id_rayon from users " +
    "join rayon on users.id_rayon=rayon.id_rayon) r on zayav.id_users=r.id_user) kol join
" +
    "(select sum(kbkvid.summ1) ss, kbkvid.id_zayav from kbkvid " +
    "group BY kbkvid.id_zayav) summa on kol.id_zayav=summa.id_zayav " +
    "group by id_rayon");
    int i=1;
    int sum1 = 0,sum3 = 0,sum4 = 0,sum5 = 0;
    double sum2 = 0;
    while (resultSet.next()) {
      //System.out.println(resultSet.getInt("id_zayav"));
    }
  <%>
  <tr>

```

```

        <td>
            <%=i++%>
        </td>
        <td>
            <%=resultSet.getString("kod")%>
        </td>
        <td>
            <%=resultSet.getString("namerayon")%>
        </td>
        <td>
            <% sum1+=resultSet.getInt("kolvo");%>
            <%=resultSet.getString("kolvo")%>
        </td>
        <td>
            <% sum2+=resultSet.getDouble("summa");%>
            <%=resultSet.getString("summa")%>
        </td>
        <td>
            <% sum3+=resultSet.getInt("s2");%>
            <%=resultSet.getString("s2")%>
        </td>
        <td>
            <% sum4+=resultSet.getInt("s3");%>
            <%=resultSet.getString("s3")%>
        </td>
        <td>
            <% sum5+=resultSet.getInt("s4");%>
            <%=resultSet.getString("s4")%>
        </td>
    </tr>
    <%
    }
    %>

    <tr>
        <td colspan="3">Итого по области</td>
        <td><%=sum1%></td>
        <td><%=sum2%></td>
        <td><%=sum3%></td>
        <td><%=sum4%></td>
        <td><%=sum5%></td>
    </tr>
</table>
</html>

```

tlog.jsp

```

<% @ page import="model.UsersPFR" %><%--
Created by IntelliJ IDEA.
User: 041BratchinAV
Date: 01.03.2018

```

Time: 9:55

To change this template use File | Settings | File Templates.

```
--%>
<% @ page contentType="text/html;charset=UTF-8" language="java" %>

<% UsersPFR usersPFR = (UsersPFR) session.getAttribute("user");%>

<script type="text/javascript">
    var iii=0;
    $(document).ready(function () {
        $("table[name='login']").on('click','a',function () {
            if(this.getAttribute("name")==="exit") {//добавить в БД
                window.location.href = './exit';
            }
        });
    });
</script>
<table cellpadding="2" align="right" name="login"> <%--Логин пароль--%>
    <tr>
        <td>Логин:</td>
        <td><%= usersPFR.getLogin()%>
        </td>
    </tr>
    <tr>
        <td>Район:</td>
        <td><%= usersPFR.getNAMERAYON()%>
        </td>
    </tr>
    <tr>
        <td></td>
        <td><a href="#" name="exit">Выход</a>
        </td>
    </tr>
</table>
```

upfr.jsp

```
<% @ page import="model.UsersPFR" %>

<%--
Created by IntelliJ IDEA.
User: 041BratchinAV
Date: 01.02.2018
Time: 9:30
To change this template use File | Settings | File Templates.
--%>

<% @ page contentType="text/html;charset=UTF-8" language="java" %>
<%
    UsersPFR usersPFR = (UsersPFR) session.getAttribute("user");
%> <%--сессия--%>
<html>
```

```

<head>
  <style>
    [name="history"] {
      margin-top: 20px;
    }

    #tkbkvvod {
      margin-top: 10px;
    }

    #tkbk {
      margin-top: 5px;
    }

    #kbkins, [name="delkbk"] {
      color: #1740ff;
    }

    #kbkins:hover, [name="delkbk"]:hover {
      text-decoration-line: underline; /* Цвет ссылок при наведении на них курсора мыши */
      cursor: pointer;
    }

    #kbkins:active, [name="delkbk"]:active {
      color: black; /* Цвет активных ссылок */
    }

    li {
      color: #1740ff;
      display: inline-block; /* Строчно-блочные элементы */
      /*background: #CA181A; /* Цвет фона */!*/
      margin-right: 50px; /* Расстояние между пунктами меню */
    }

    li:hover {
      /*background: #333; /* Цвет фона при наведении курсора мыши */!*/
      cursor: pointer;
      text-decoration-line: underline;
    }

    .fw {
      font-weight: bold;
    }
  </style>

  <script type="text/javascript" src="../lib/jquery-3.3.1.min.js"></script>
  <script type="text/javascript">
    var iii = 0;

    $(document).ready(function () {

      $("#li1").addClass("fw");

```

```

$("#st1").addClass("fw");

$("table[name='history']").on('click', 'input', function () {
    if (this.getAttribute("name") == "buttonrech") { //добавить в БД

        $("table[class='history']");
        var tt = "table[class='" + this.getAttribute("id") + "'] input[name='nom_rech']";
        var t = "table[class='" + this.getAttribute("id") + "'] input[name='data_rech']";
        var param = "id=" + this.getAttribute("id");
        param += "&nom_rech=" + $(tt).val() + "&data_rech=" + $(t).val();
        $("table[name='history']").load("../insresh", param);

    }

    if (this.getAttribute("name") == "rechnull") { //добавить в БД

        alert("a");
        $("table[class='history']");
        var param = "id=" + this.getAttribute("id");
        param += "&nom_rech=0&data_rech=01.01.1900";
        $("table[name='history']").load("../insresh", param);
        //alert("b");

    }

});

$("form").on('click', 'li', function () {
    $("table[name='history']").empty();
    if (this.getAttribute("id") == "li1") {
        $("#li1").addClass("fw");
        $("#li2").removeClass("fw");
        $("#li3").removeClass("fw");
        $("table[name='history']").load("upfrhistory.jsp", "userid=<%=usersPFR.getID()%>");
        $("#st1").html("1"); //обнулить строку перехода
        $("#st2").html("2");
        $("#st3").html("3");
        $("#st4").html("4");
        $("#st5").html("5");
        $("#st1").addClass("fw");
        $("#st2").removeClass("fw");
        $("#st3").removeClass("fw");
        $("#st4").removeClass("fw");
        $("#st5").removeClass("fw");
    }
    if (this.getAttribute("id") == "li2") {
        $("#li2").addClass("fw");
        $("#li3").removeClass("fw");
        $("#li1").removeClass("fw");
        $("table[name='history']").load("upfrizfns.jsp", "");
        $("#st1").html("1");
        $("#st2").html("2");
        $("#st3").html("3");
    }
});

```

```

    $("#st4").html("4");
    $("#st5").html("5");
    $("#st1").addClass("fw");
    $("#st2").removeClass("fw");
    $("#st3").removeClass("fw");
    $("#st4").removeClass("fw");
    $("#st5").removeClass("fw");
}
if (this.getAttribute("id") == "li3") {
    $("#li3").addClass("fw");
    $("#li2").removeClass("fw");
    $("#li1").removeClass("fw");
    $("table[name='history']").load("upfresh.jsp", "");
    $("#st1").html("1");
    $("#st2").html("2");
    $("#st3").html("3");
    $("#st4").html("4");
    $("#st5").html("5");
    $("#st1").addClass("fw");
    $("#st2").removeClass("fw");
    $("#st3").removeClass("fw");
    $("#st4").removeClass("fw");
    $("#st5").removeClass("fw");
}
});

$("#form").on('click', 'td', function () {

    if (this.getAttribute("id") == "kbbins") { //добавить кбк
        $("#tkbk").hide();
        $("#tkbkvvod").show();
    }
    if (this.getAttribute("name") == "delkbb") { // удалить кбк
        var s = $("tr[name=" + this.getAttribute("id") + "]").html();
        document.getElementById("textkbb").value =
            document.getElementById("textkbb").value.replace(this.getAttribute("id"), "");
        $("tr[name=" + this.getAttribute("id") + "]").remove();
        if (document.getElementById("textkbb").value == "") {
            $("#inputall").hide();
        }
    }
});

$("#form").on('click', 'input', function () {
    if (this.getAttribute("id") == "setkbb") { //добавить кбк
        var ikbk2 = document.getElementById('ikbk');
        var ividplat2 = document.getElementById('ividplat');
        var isumm = document.getElementById('isumm');
        iii = ikbk2.options[ikbk2.options.selectedIndex].value + ";" +
            ividplat2.options[ividplat2.options.selectedIndex].value + ";" +
            isumm.value + ";";
        if ($("tr[name=" + iii + "]").attr("name")) {
            alert("Такой элемент уже добавлен");
        }
    }
});

```

```

    }
    else {
        var tab = "<tr name=" + iii + "><td>" +
            ikbk2.options[ikbk2.options.selectedIndex].text + "</td><td>" +
            ividplat2.options[ivdplat2.options.selectedIndex].text + "</td><td>" +
            isumm.value + "</td>" + "<td align='center' name='delkbbk' id=" + iii +
">удалить</td></tr>";
        $("#kbbk1").after(tab); //добавить после
        document.getElementById("textkbbk").value +=
            ikbk2.options[ikbk2.options.selectedIndex].value + ";" +
            ividplat2.options[ivdplat2.options.selectedIndex].value + ";" +
            isumm.value + ";";

        $("#isumm").val("");

        $("#tkbkvvod").hide(); //скрыть
        $("#tkbk").show(); //показать
        $("#inputall").show(); //показать
    }
}

if (this.getAttribute("id") == "inputall") { //добавить в БД

    var regnom = $("input[name='regnom']").val();
    var fio = $("input[name='fio']").val();
    var inn = $("input[name='inn']").val();
    var kpp = $("input[name='kpp']").val();
    var data = $("input[name='data']").val();
    var textkbbk = $("input[name='textkbbk']").val();
    $("table[name='history']").load("upfrhistory.jsp",
        "regnom=" + regnom + "&fio=" + fio + "&inn=" + inn + "&kpp=" + kpp + "&data="
+ data + "&textkbbk=" + textkbbk + "&userid=" + <%=usersPFR.getID()%>");
    var a = [];
    a = $("input[name='textkbbk']").val().split(";");
    var s;
    for (var i = 0; i < a.length - 1; i = i + 3) {
        s = a[i] + ";" + a[i + 1] + ";" + a[i + 2] + ";";
        $("tr[name='" + s + "']").remove();
    }
    $("#isumm").val("");
    $("input[name='regnom']").val("");
    $("input[name='fio']").val("");
    $("input[name='inn']").val("");
    $("input[name='kpp']").val("");
    $("input[name='data']").val("");
    $("input[name='textkbbk']").val("");

    $("#inputall").hide();
    $("#tkbk").show();
    $("#tkbkvvod").hide();
}
});

```



```

$("ul").on('click', 'li', function () {

    if (this.getAttribute("id") == "stn") {
        $("#li1").hasClass("fw");
        $("#li2").hasClass("fw");
        $("#li3").hasClass("fw");
        console.log($("#li1").hasClass("fw"));
        console.log($("#li2").hasClass("fw"));
        console.log($("#li3").hasClass("fw"));

        if ($("#li1").hasClass("fw")) {
            $("#table[name='history']").load("upfrhistory.jsp",
"userid=<%=usersPFR.getID()%>&str=" + $("#st1").html());
        }
        if ($("#li2").hasClass("fw")) {
            $("#table[name='history']").load("upfrizfns.jsp", "str=" + $("#st1").html());
        }
        if ($("#li3").hasClass("fw")) {
            $("#table[name='history']").load("upfresh.jsp", "str=" + $("#st1").html());
        }
        //$("#table[name='table1']").load("opfrvod.jsp", "str="+$("#st1").html());
        $("#st1").html("1");
        $("#st2").html("2");
        $("#st3").html("3");
        $("#st4").html("4");
        $("#st5").html("5");
        $("#st1").addClass("fw");
        $("#st2").removeClass("fw");
        $("#st3").removeClass("fw");
        $("#st4").removeClass("fw");
        $("#st5").removeClass("fw");
    }
    if (this.getAttribute("id") == "st1") {
        if ($("#li1").hasClass("fw")) {
            $("#table[name='history']").load("upfrhistory.jsp",
"userid=<%=usersPFR.getID()%>&str=" + $("#st1").html());
        }
        if ($("#li2").hasClass("fw")) {
            $("#table[name='history']").load("upfrizfns.jsp", "str=" + $("#st1").html());
        }
        if ($("#li3").hasClass("fw")) {
            $("#table[name='history']").load("upfresh.jsp", "str=" + $("#st1").html());
        }
        //$("#table[name='table1']").load("opfrvod.jsp", "str="+$("#st1").html());
        $("#st1").addClass("fw");
        $("#st2").removeClass("fw");
        $("#st3").removeClass("fw");
        $("#st4").removeClass("fw");
        $("#st5").removeClass("fw");
    }
    if (this.getAttribute("id") == "st2") {

```

```

        if ($("#li1").hasClass("fw")) {
            $("table[name='history']").load("upfrhistory.jsp",
"userid=<%=usersPFR.getID()%>&str=" + $("#st2").html());
        }
        if ($("#li2").hasClass("fw")) {
            $("table[name='history']").load("upfrizfns.jsp", "str=" + $("#st2").html());
        }
        if ($("#li3").hasClass("fw")) {
            $("table[name='history']").load("upfresh.jsp", "str=" + $("#st2").html());
        }
        //$("#table[name='table1']").load("opfrvod.jsp","str="+$("#st2").html());
        $("#st1").removeClass("fw");
        $("#st2").addClass("fw");
        $("#st3").removeClass("fw");
        $("#st4").removeClass("fw");
        $("#st5").removeClass("fw");
    }
    if (this.getAttribute("id") == "st3") {
        if ($("#li1").hasClass("fw")) {
            $("table[name='history']").load("upfrhistory.jsp",
"userid=<%=usersPFR.getID()%>&str=" + $("#st3").html());
        }
        if ($("#li2").hasClass("fw")) {
            $("table[name='history']").load("upfrizfns.jsp", "str=" + $("#st3").html());
        }
        if ($("#li3").hasClass("fw")) {
            $("table[name='history']").load("upfresh.jsp", "str=" + $("#st3").html());
        }
        //$("#table[name='table1']").load("opfrvod.jsp","str="+$("#st3").html());
        $("#st1").removeClass("fw");
        $("#st2").removeClass("fw");
        $("#st3").addClass("fw");
        $("#st4").removeClass("fw");
        $("#st5").removeClass("fw");
    }
    if (this.getAttribute("id") == "st4") {
        if ($("#li1").hasClass("fw")) {
            $("table[name='history']").load("upfrhistory.jsp",
"userid=<%=usersPFR.getID()%>&str=" + $("#st4").html());
        }
        if ($("#li2").hasClass("fw")) {
            $("table[name='history']").load("upfrizfns.jsp", "str=" + $("#st4").html());
        }
        if ($("#li3").hasClass("fw")) {
            $("table[name='history']").load("upfresh.jsp", "str=" + $("#st4").html());
        }
        //$("#table[name='table1']").load("opfrvod.jsp","str="+$("#st4").html());
        $("#st1").removeClass("fw");
        $("#st2").removeClass("fw");
        $("#st3").removeClass("fw");
        $("#st4").addClass("fw");
        $("#st5").removeClass("fw");
    }

```

```

    }
    if (this.getAttribute("id") == "st5") {
        if ($("#li1").hasClass("fw")) {
            $("table[name='history']").load("upfrhistory.jsp",
"userid=<%=usersPFR.getID()%>&str=" + $("#st5").html());
        }
        if ($("#li2").hasClass("fw")) {
            $("table[name='history']").load("upfrizfns.jsp", "str=" + $("#st5").html());
        }
        if ($("#li3").hasClass("fw")) {
            $("table[name='history']").load("upfrfresh.jsp", "str=" + $("#st5").html());
        }
        //$("#table[name='table1']").load("opfrvod.jsp","str="+$("#st5").html());
        $("#st1").removeClass("fw");
        $("#st2").removeClass("fw");
        $("#st3").removeClass("fw");
        $("#st4").removeClass("fw");
        $("#st5").addClass("fw");
    }
    if (this.getAttribute("id") == "std") {
        var pos = $("#st5").html();
        if ($("#li1").hasClass("fw")) {
            $("table[name='history']").load("upfrhistory.jsp",
"userid=<%=usersPFR.getID()%>&str=" + 1+(+pos));
        }
        if ($("#li2").hasClass("fw")) {
            $("table[name='history']").load("upfrizfns.jsp", "str=" + 1+(+pos));
        }
        if ($("#li3").hasClass("fw")) {
            $("table[name='history']").load("upfrfresh.jsp", "str=" + 1+(+pos));
        }
        //$("#table[name='table1']").load("opfrvod.jsp","str="+1+(+pos));
        $("#st1").html(pos - 1);
        $("#st2").html(pos);
        $("#st3").html(+pos + 1);
        $("#st4").html(+pos + 2);
        $("#st5").html(+pos + 3);
        $("#st1").removeClass("fw");
        $("#st2").removeClass("fw");
        $("#st3").addClass("fw");
        $("#st4").removeClass("fw");
        $("#st5").removeClass("fw");
    }
});

});
</script>
<title>ПК Возвраты</title>
<style>
    select, input {
        width: 200px; /* Ширина списка в пикселях */
    }

```

```

        input[type="button"] {
            width: 100px;
        }
    </style>
</head>
<body>

<jsp:include page="tlog.jsp"/>
<%--пользователь сверху--%>

<br>
<br>
<br>

<form action='../insupfr' method='post'>
    <table cellpadding="3" align="center">
        <tr>
            <th colspan="2">Внести данные из заявления плательщика</th>
        </tr>
        <tr>
            <td>Регистрационный номер:</td>
            <td><input type='text' name='regnom'></td>
        </tr>
        <tr>
            <td>Наименование/ФИО плательщика:</td>
            <td><input type='text' name='fio'></td>
        </tr>
        <tr>
            <td>ИНН:</td>
            <td><input type='text' name='inn'></td>
        </tr>
        <tr>
            <td>КПП:</td>
            <td><input type='text' name='kpp'></td>
        </tr>
        <tr>
            <td>Дата заявления:</td>
            <td><input type='date' name='data'></td>
        </tr>
    </table>
    <br>

<jsp:include page="upfrkbksum.jsp"/>

<ul align="center">
    <li id="li1">Отправленные заявления</li>
    <li id="li2">Получен ответ от ФНС</li>
    <li id="li3">Решение принято</li>
</ul>

```

```

<table border="1" align="center" cellspacing="0" name="history" width="100%">

    <%
        String pg = "";
        int param = (Integer) session.getAttribute("param");
        if (param == 1) {
            pg = "upfrhistory.jsp?userid=" + usersPFR.getID();
        } else if (param == 2) {
            pg = "upfrifns.jsp";
        } else if (param == 3) {
            pg = "upfrfresh.jsp";
        }
    %>
    <jsp:include page="<%=pg%"/>

</table>

<ul align="center">
    <li id="stn">В начало</li>
    <li id="st1">1</li>
    <li id="st2">2</li>
    <li id="st3">3</li>
    <li id="st4">4</li>
    <li id="st5">5</li>
    <li id="std">Дальше</li>
</ul>

</form>
</body>
</html>

```

upfrhistory.jsp

```

<% @ page import="model.VidKbk" %>
<% @ page import="java.util.Set" %>
<% @ page import="java.util.HashSet" %>
<% @ page import="dao.Conn" %>
<% @ page import="java.sql.Connection" %>
<% @ page import="java.sql.PreparedStatement" %>
<% @ page import="java.sql.Statement" %>
<% @ page import="java.sql.ResultSet" %>

<%--
Created by IntelliJ IDEA.
User: 041BratchinAV
Date: 22.03.2018
Time: 10:40
To change this template use File | Settings | File Templates.
--%>

```

```
<% @ page contentType="text/html;charset=UTF-8" language="java" %>
```

```
<%
    request.setCharacterEncoding("UTF-8");
    session.setAttribute("param",1);//какая вкладка
    String regnom = "";
    String fio = "";
    String inn = "";
    String kpp = "";
    String data = "";
    int str = 1;
    try {
        str = Integer.parseInt(request.getParameter("str"));
    } catch (Exception e) {
    }
    String[] textkbk = null;
    int userid = 0;
    Set<VidKbk> vidKbks = null;
    try {
        regnom = request.getParameter("regnom")==null?"":request.getParameter("regnom");
        fio = request.getParameter("fio")==null?"":request.getParameter("fio");
        inn = request.getParameter("inn")==null?"":request.getParameter("inn");
        kpp = request.getParameter("kpp")==null?"":request.getParameter("kpp");
        data = request.getParameter("data")==null?"":request.getParameter("data");
        userid =
request.getParameter("userid")==null?0:Integer.parseInt(request.getParameter("userid"));
        textkbk =
request.getParameter("textkbk")==null?null:request.getParameter("textkbk").split(";");
        //обрабатываем kbk
        if(textkbk!=null){
            vidKbks= new HashSet<VidKbk>();
            for (int i = 0; i < textkbk.length; i=i+3) {
                vidKbks.add(new VidKbk(textkbk[i], textkbk[i + 1], textkbk[i + 2]));
            }
        }
    } catch (Exception e) {
//        System.out.println("Ошибка в приеме параметра:"+regnom+" "+fio+" "+inn+" "+kpp+"
"+data+" "+userid+" "+textkbk);
    }
    System.out.println("Параметры приняты:"+regnom+" "+fio+" "+inn+" "+kpp+" "+data+"
"+userid+" "+textkbk);
    //добавить в бд
    if(!regnom.equals("")){
        Conn conn = new Conn();
        Connection connection = conn.setConnection();
        PreparedStatement preparedStatement = null;
        PreparedStatement preparedStatement2 = null;
        connection.setAutoCommit(false);//транзакция
        String qwery = "INSERT INTO zayav (regnum, name, inn, kpp, date_zaya, ID_USERS) " +
            "VALUES (?,?,,?,?,?)";
        preparedStatement = connection.prepareStatement(qwery,
```

```

Statement.RETURN_GENERATED_KEYS);
    preparedStatement.setString(1, regnom);
    preparedStatement.setString(2, fio);
    preparedStatement.setString(3, inn);
    preparedStatement.setString(4, kpp);
    preparedStatement.setString(5, data);
    preparedStatement.setInt(6, userid);
    preparedStatement.executeUpdate();
    ResultSet rs = preparedStatement.getGeneratedKeys(); // получить сгенерированный индекс
    int idValue = 0;
    if (rs.next()) { // Значение ID
        idValue = rs.getInt(1);
    }
    String stringInsert = ""; // добавить kbkvid
    for (VidKbk v:
        vidKbks) {
        stringInsert="Insert into kbkvid (id_zayav,id_kbk,id_vidplat,summ1) " +
            "value (?,?,?,?)";
        preparedStatement2=connection.prepareStatement(stringInsert);
        preparedStatement2.setString(1, String.valueOf(idValue));
        preparedStatement2.setString(2, String.valueOf(v.getKbk()));
        preparedStatement2.setString(3, String.valueOf(v.getVid()));
        preparedStatement2.setString(4, v.getSumm());
        preparedStatement2.executeUpdate();
    }
    connection.commit();//завершить транзакцию
}

```

%>

```

<tr>
    <th colspan="10">Все заявления</th>
</tr>
<tr>
    <th align="center">Пер. номер</th>
    <th align="center">Наименование</th>
    <th align="center">ИНН</th>
    <th align="center">КПП</th>
    <th align="center">Дата заявления</th>
    <%--<th align="center">Дата загрузки</th>--%>
    <th align="center">КБК</th>
    <th align="center">Вид платежа</th>
    <th align="center">Сумма</th>
    <th align="center">Статус</th>
    <%--<th align="center">Сумма2</th>--%>
</tr>
<%

```

```

Conn conn = new Conn();
Connection connection = conn.setConnection();
Statement statement = connection.createStatement();
ResultSet resultSet;

```

```

Statement statement2 = connection.createStatement();
ResultSet resultSet2;
resultSet = statement.executeQuery("select id_zayav, regnum, name, inn, KPP, DATE_ZAYA,
v.id_stat, v.stat, DATE_ZAGR from " +
"(SELECT id_zayav,regnum, name, inn,KPP,DATE_ZAYA,id_status,DATE_ZAGR
from zayav " +
"WHERE ID_USERS="+userid+") k " +
"join (select id_stat, stat from stat) v on v.id_stat = k.id_status /*where v.id_stat<3*/");
int kolvonastr = 10;
int strnach = kolvonastr * str - kolvonastr;
int strkon = kolvonastr * str;
int ii=1;
while (resultSet.next()) {
    if(ii>=strnach && ii<=strkon){    %>
<tr>
    <td><%=resultSet.getInt("REGNUM")%>
    </td>
    <td><%=resultSet.getString("NAME")%>
    </td>
    <td><%=resultSet.getString("INN")%>
    </td>
    <td><%=resultSet.getString("KPP")%>
    </td>
    <td><%=resultSet.getDate("DATE_ZAYA")%>
    </td>
    <!--<td><%=resultSet.getDate("DATE_ZAGR")%>--%>
    <!--</td>--%>
    <td><%=
        resultSet2.executeQuery("select v.id_kbkvid, k.kbk, v.vidplat, v.summ1/*,
v.summ2*/ " +
        "from (select id_kbkvid, id_zayav, kbk " +
        "from kbk join kbkvid on kbk.id_kbk= kbkvid.id_kbk) k join " +
        "(select id_kbkvid, id_zayav, vidplat, summ1, summ2 from vidplat " +
        "join kbkvid on vidplat.id_vidplat= kbkvid.id_vidplat) v on v.id_kbkvid = k.id_kbkvid
" +
        "where v.id_zayav = " + resultSet.getInt("id_zayav") + " order by v.id_kbkvid");
String skbk = "";
String svidplat = "";
String ssumm = "";
String ssumm2 = "";
while (resultSet2.next()) {
    skbk += resultSet2.getString("k.kbk") + "<br>";
    svidplat += resultSet2.getString("v.vidplat") + "<br>";
    ssumm += resultSet2.getString("v.summ1") + "<br>";
//    ssumm2 += resultSet2.getString("v.summ2")=null?"":resultSet2.getString("v.summ2")
+ "<br>";
    }%>
    <%=skbk%>
    </td>
    <td><%=svidplat%>
    </td>
    <td><%=ssumm%>

```



```

        </td>
<%--      <td><%=ssumm2%>
        </td>--%>
        <%
            String statname = resultSet.getString("v.stat");
            int statid = resultSet.getInt("v.id_stat");
            resultSet.getInt("id_zayav");
            if(statid!=10){ // возможно будет не нужно!!!
//      <td id="kbbins" align="center">Добавить КБК</td>
        %>
        <td><%=statname%>
        </td>
        <% }else{ }
        %>
    </tr>
    <% }ii++;
    }%>

```

#### upfrizfns.jsp

```

<% @ page import="model.UsersPFR" %>
<% @ page import="java.sql.ResultSet" %>
<% @ page import="java.sql.Statement" %>
<% @ page import="java.sql.Connection" %>
<% @ page import="dao.Conn" %>
<%--
    Created by IntelliJ IDEA.
    User: 041BratchinAV
    Date: 29.03.2018
    Time: 9:15
    To change this template use File | Settings | File Templates.
--%>
<% @ page contentType="text/html; charset=UTF-8" language="java" %>
<%
    UsersPFR usersPFR = (UsersPFR) session.getAttribute("user");
    session.setAttribute("param",2); //какая вкладка
%>

<tr>
    <th align="center">УПФР</th>
    <th align="center">ОПФР</th>
    <th align="center">Решение</th>
</tr>
<%
    int str = 1;
    try {
        str = Integer.parseInt(request.getParameter("str"));
    } catch (Exception e) {
    }
    try {
        Conn conn = new Conn();
        Connection connection = conn.setConnection();

```

```

Statement statement = connection.createStatement();
ResultSet resultSet;
resultSet = statement.executeQuery("SELECT id_zayav " +
    "from zayav where id_users = " + usersPFR.getID() + " and id_status = 3");

int kolvonastr = 10;
int strnach = kolvonastr * str - kolvonastr;
int strkon = kolvonastr * str;
int ii=1;
while (resultSet.next()) {
    if(ii>=strnach && ii<=strkon){
        //System.out.println(resultSet.getInt("id_zayav"));
    }
}
%>

<%--<tr><td>drsg hdf</td></tr>--%>
<tr id="<%=resultSet.getInt("id_zayav")%>">

    <td width="33%" align="center" name="upfr">
        <%String ss = "otupfr.jsp?id_zayav=" + resultSet.getInt("id_zayav");%>
        <jsp:include page="<%=ss%>" />
    </td>

    <%--ОТВЕТ--%>
    <td width="33%" align="center" name="opfr">
        <%ss = "otopfr.jsp?id_zayav=" + resultSet.getInt("id_zayav");%>
        <jsp:include page="<%=ss%>" />
    </td>

    <td align="center" name="rech">
        <%ss = "rech.jsp?id_zayav=" + resultSet.getInt("id_zayav");%>
        <jsp:include page="<%=ss%>" />
    </td>

</tr>
<%
    }ii++;
}
} catch (Exception e) {
}
%>

```

upfrkbksum.jsp

```

<% @ page import="dao.Conn" %>
<% @ page import="java.sql.Connection" %>
<% @ page import="java.sql.Statement" %>
<% @ page import="java.sql.ResultSet" %><%--
Created by IntelliJ IDEA.
User: 041BratchinAV
Date: 19.03.2018
Time: 15:47
To change this template use File | Settings | File Templates.

```

```

--%>
<% @ page contentType="text/html;charset=UTF-8" language="java" %>

<table name="upfrkbksum" align="center" cellspacing="0" border="1" >
  <tr id="kbl1">
    <td width="200" align="center">КБК</td>
    <td width="200" align="center">Вид платежа</td>
    <td width="200" align="center">Сумма</td>
  </tr>
</table>

<table id="tkbk" align="center" cellspacing="0" >
  <tr>
    <td id="kblins" align="center">Добавить КБК</td>
  </tr>
  <tr >
    <td colspan="3" align="center">
      <input type="button" id="inputall" value="Внести данные" style="display: none;">
    </td>
  </tr>
</table>

<table align="center" id="tkbkvvod" style="display: none;">
  <tr>
    <td align="center">Выберите КБК</td>
    <td align="center">Выберите вид платежа</td>
    <td align="center">Сумма</td>
    <td align="center"></td>
  </tr>

  <tr id="kbl">
    <td>
      <select name="kbl" id="ikbl">
        <% Conn conn = new Conn();
          Connection connection = conn.setConnection();
          Statement statement = connection.createStatement();
          ResultSet resultSet = statement.executeQuery("SELECT id_kbl, kbl from kbl
ORDER BY kbl");

          while (resultSet.next()) {
            String s = resultSet.getString("КБК");
            int idkbl = resultSet.getInt("ID_КБК");
            %>
            <option value="<%=idkbl%>"><%=s%>
          </option>
          <% }%>
        </select>
      </td>
    <td>
      <select id="ividplat">
        <%

```

```

Statement statement2 = connection.createStatement();
ResultSet resultSet2;
resultSet = statement.executeQuery("SELECT id_vidplat, vidplat from vidplat");
while (resultSet.next()) {
    String s = resultSet.getString("vidplat");
    int idvidplat = resultSet.getInt("id_vidplat");
    %>
    <option value="<%=idvidplat%>"><%=s%>
    </option>
    <% }%>
</select>
</td>
<td>
    <input type='text' name='summ' id="isumm">
</td>
</tr>
<tr>
    <td colspan="3" align="right"><input type='button' id='setkbk' value="Добавить КБК">
    <input type="text" id="textkbk" name="textkbk" hidden></td>
</tr>
</table>

```

upfresh.jsp

```

<% @ page import="model.UsersPFR" %>
<% @ page import="dao.Conn" %>
<% @ page import="java.sql.Connection" %>
<% @ page import="java.sql.Statement" %>
<% @ page import="java.sql.ResultSet" %><%--
Created by IntelliJ IDEA.
User: 041BratchinAV
Date: 13.04.2018
Time: 9:16
To change this template use File | Settings | File Templates.
--%>
<% @ page contentType="text/html;charset=UTF-8" language="java" %>
<%
UsersPFR usersPFR = (UsersPFR) session.getAttribute("user");
session.setAttribute("param",3);//какая вкладка
%>

<tr>
    <th align="center">УПФР</th>
    <th align="center">ОПФР</th>
    <th align="center">Решение</th>
</tr>
<%--<tr><td>drsg hdf</td></tr>--%>

<%
int str = 1;
try {
    str = Integer.parseInt(request.getParameter("str"));

```

```

    } catch (Exception e) {
    }
    try {
        Conn conn;
        conn = new Conn();
        Connection connection = conn.setConnection();
        Statement statement = connection.createStatement();
        ResultSet resultSet;
        resultSet = statement.executeQuery("SELECT id_zayav " +
            "from zayav where id_users = " + usersPFR.getID() + " and id_status > 3");

        int kolvonastr = 10;
        int strnach = kolvonastr * str - kolvonastr;
        int strkon = kolvonastr * str;
        int ii=1;
        while (resultSet.next()) {
            if(ii>=strnach && ii<=strkon){
                //System.out.println(resultSet.getInt("id_zayav"));
            }
        }
    } catch (Exception e) {
    }
}
%>

```

```

<tr id="<%=resultSet.getInt("id_zayav")%>">

```

```

    <td width="33%" align="center" name="upfr">
        <%String ss = "otupfr.jsp?id_zayav=" + resultSet.getInt("id_zayav");%>
        <jsp:include page="<%=ss%>"/>
    </td>

```

```

<%--OTBET--%>
<td width="33%" align="center" name="opfr">
    <%ss = "otopfr.jsp?id_zayav=" + resultSet.getInt("id_zayav");%>
    <jsp:include page="<%=ss%>"/>
</td>

```

```

<td align="center" name="rech">
    <%ss = "rech.jsp?id_zayav=" + resultSet.getInt("id_zayav");%>
    <jsp:include page="<%=ss%>"/>
</td>

```

```

</tr>
<%
    }ii++;
    }
} catch (Exception e) {
}
%>

```

```

CREATE TABLE IF NOT EXISTS `dokument` (
    `id_dokument` int(11) NOT NULL AUTO_INCREMENT,
    `dokument` longblob,

```

```

`name_dokument` varchar(100) DEFAULT NULL,
`name_file` varchar(100) DEFAULT NULL,
PRIMARY KEY (`id_dokument`),
UNIQUE KEY `name_file` (`name_file`)
) ENGINE=InnoDB AUTO_INCREMENT=95 DEFAULT CHARSET=utf8mb4;
CREATE TABLE IF NOT EXISTS `kbk` (
  `id_kbk` int(11) NOT NULL AUTO_INCREMENT,
  `kbk` varchar(250) NOT NULL,
  PRIMARY KEY (`id_kbk`)
) ENGINE=InnoDB AUTO_INCREMENT=291 DEFAULT CHARSET=utf8mb4;
CREATE TABLE IF NOT EXISTS `kbkvid` (
  `id_kbkvid` int(11) NOT NULL AUTO_INCREMENT,
  `id_zayav` int(11),
  `id_kbk` int(11),
  `id_vidplat` int(11),
  `summ1` varchar(10),
  `summ2` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`id_kbkvid`),
  KEY `FK_kbkvid_zayav` (`id_zayav`),
  KEY `FK_kbkvid_kbk` (`id_kbk`),
  KEY `FK_kbkvid_vidplat` (`id_vidplat`),
  CONSTRAINT `FK_kbkvid_kbk` FOREIGN KEY (`id_kbk`) REFERENCES `kbk` (`id_kbk`),
  CONSTRAINT `FK_kbkvid_vidplat` FOREIGN KEY (`id_vidplat`) REFERENCES `vidplat`
(`id_vidplat`),
  CONSTRAINT `FK_kbkvid_zayav` FOREIGN KEY (`id_zayav`) REFERENCES `zayav`
(`id_zayav`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4;
CREATE TABLE IF NOT EXISTS `opfr` (
  `id_opfr` int(11) NOT NULL AUTO_INCREMENT,
  `nom_zap` varchar(20) NOT NULL,
  `data_otpr` date NOT NULL,
  `nom_otv` varchar(20) DEFAULT NULL,
  `data_otv` date DEFAULT NULL,
  `id_dokument` int(11) DEFAULT NULL,

```

```

`id_zayav` int(11) NOT NULL,
`id_user` int(11) NOT NULL,
PRIMARY KEY (`id_opfr`),
KEY `FK_opfr_dokument` (`id_dokument`),
KEY `FK_opfr_users` (`id_user`),
KEY `FK_opfr_zayav` (`id_zayav`),
CONSTRAINT `FK_opfr_dokument` FOREIGN KEY (`id_dokument`) REFERENCES
`dokument` (`id_dokument`),
CONSTRAINT `FK_opfr_users` FOREIGN KEY (`id_user`) REFERENCES `users` (`id_user`),
CONSTRAINT `FK_opfr_zayav` FOREIGN KEY (`id_zayav`) REFERENCES `zayav`
(`id_zayav`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4;
CREATE TABLE IF NOT EXISTS `rayon` (
`id_rayon` int(11) NOT NULL AUTO_INCREMENT,
`namerayon` varchar(255) NOT NULL,
`kod` varchar(4) NOT NULL,
PRIMARY KEY (`id_rayon`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4
COMMENT='районы или роли';
CREATE TABLE IF NOT EXISTS `rech` (
`id_rech` int(11) NOT NULL AUTO_INCREMENT,
`id_zayav` int(11) DEFAULT NULL,
`nom_rech` varchar(25) DEFAULT NULL,
`data_rech` date DEFAULT NULL,
PRIMARY KEY (`id_rech`),
KEY `FK_rech_zayav` (`id_zayav`),
CONSTRAINT `FK_rech_zayav` FOREIGN KEY (`id_zayav`) REFERENCES `zayav`
(`id_zayav`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4;
CREATE TABLE IF NOT EXISTS `stat` (
`id_stat` tinyint(4) NOT NULL AUTO_INCREMENT,
`stat` varchar(250) NOT NULL,
PRIMARY KEY (`id_stat`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4;
CREATE TABLE IF NOT EXISTS `users` (

```

```

`id_user` int(11) NOT NULL AUTO_INCREMENT,
`login` varchar(20) NOT NULL,
`password` varchar(20) NOT NULL,
`name` varchar(50) NOT NULL,
`id_rayon` int(11) NOT NULL,
PRIMARY KEY (`id_user`),
KEY `FK_users_rayon` (`id_rayon`),
CONSTRAINT `FK_users_rayon` FOREIGN KEY (`id_rayon`) REFERENCES `rayon`
(`id_rayon`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4
COMMENT='пользователи';
CREATE TABLE IF NOT EXISTS `vidplat` (
  `id_vidplat` int(11) NOT NULL AUTO_INCREMENT,
  `vidplat` varchar(250) NOT NULL,
  PRIMARY KEY (`id_vidplat`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4;
CREATE TABLE IF NOT EXISTS `zayav` (
  `id_zayav` int(11) NOT NULL AUTO_INCREMENT,
  `regnum` varchar(10) DEFAULT NULL,
  `name` varchar(250) DEFAULT NULL,
  `inn` varchar(13) DEFAULT NULL,
  `kpp` varchar(10) DEFAULT NULL,
  `date_zaya` date DEFAULT NULL,
  `id_status` tinyint(4) DEFAULT '1',
  `id_users` int(11) DEFAULT NULL,
  `date_zagr` datetime DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_zayav`),
  KEY `FK_zayav_stat` (`id_status`),
  KEY `FK_zayav_users` (`id_users`),
  CONSTRAINT `FK_zayav_stat` FOREIGN KEY (`id_status`) REFERENCES `stat` (`id_stat`),
  CONSTRAINT `FK_zayav_users` FOREIGN KEY (`id_users`) REFERENCES `users`
(`id_user`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4;

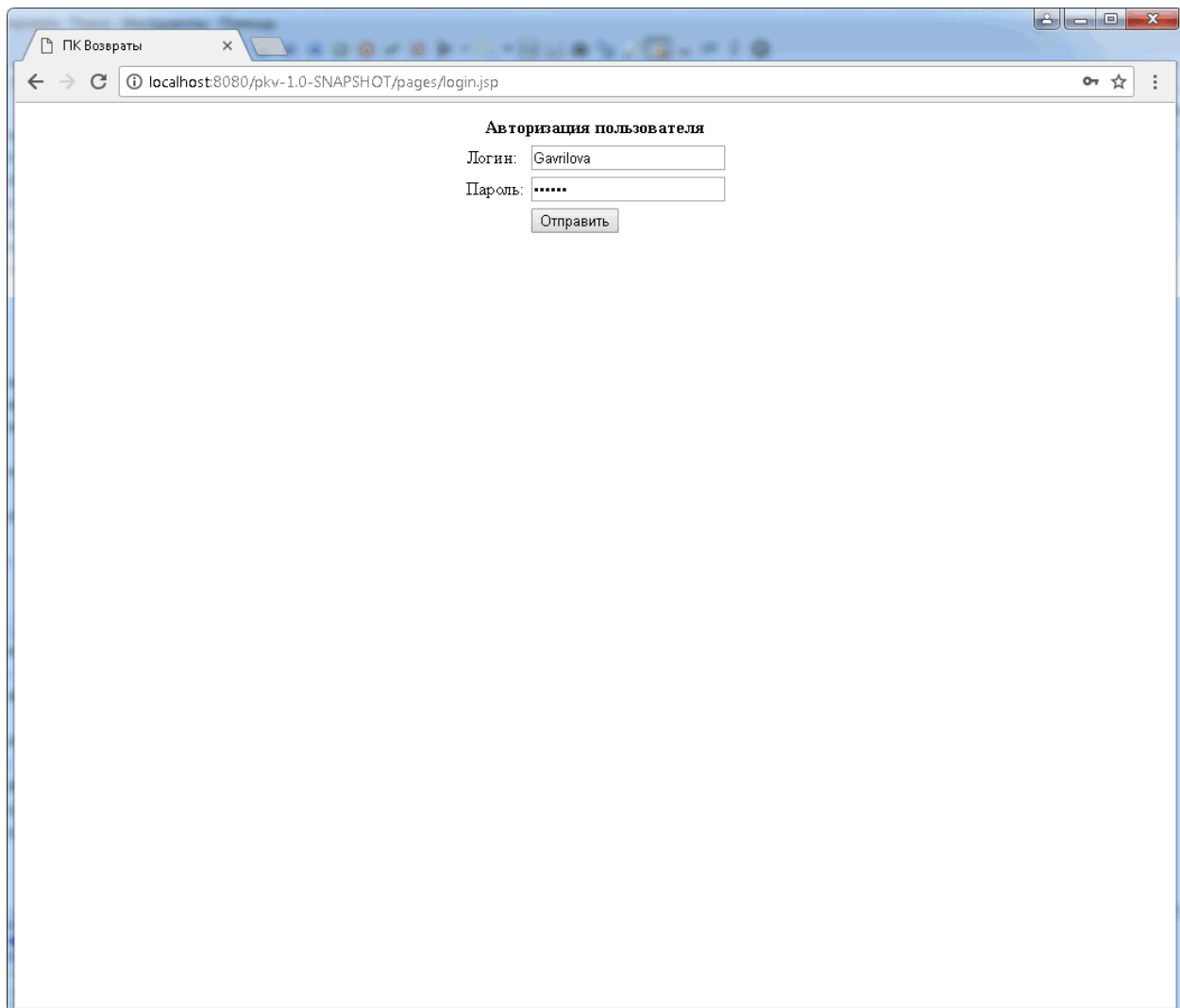
```





## ПРИЛОЖЕНИЕ Б

### Скриншоты программного комплекса «Возвраты»



The screenshot shows a web browser window with the following elements:

- Browser title: ПК Возвраты
- Address bar: localhost:8080/pkv-1.0-SNAPSHOT/pages/login.jsp
- Form title: **Авторизация пользователя**
- Form fields:
  - Логин:
  - Пароль:
- Submit button:

ПК Возвраты

localhost:8080/pkv-1.0-SNAPSHOT/pages/upfr.jsp

Логин: Gavrilova  
 Район: Белгород  
[Выход](#)

**Внести данные из заявления плательщика**

Регистрационный номер:

Наименование/ФИО плательщика:

ИНН:

КПП:

Дата заявления:

КБК	Вид платежа	Сумма
<a href="#">Добавить КБК</a>		

[Отправленные заявления](#)    [Получен ответ от ФНС](#)    [Решение принято](#)

Все заявления								
Рег. номер	Наименование	ИНН	КПП	Дата заявления	КБК	Вид платежа	Сумма	Статус
<a href="#">В начало</a> <a href="#">1</a> <a href="#">2</a> <a href="#">3</a> <a href="#">4</a> <a href="#">5</a> <a href="#">Дальше</a>								

ПК Возвраты

localhost:8080/pkv-1.0-SNAPSHOT/pages/upfr.jsp

Логин: Gavrilova  
 Район: Белгород  
[Выход](#)

**Внести данные из заявления плательщика**

Регистрационный номер:

Наименование/ФИО плательщика:

ИНН:

КПП:

Дата заявления:

КБК	Вид платежа	Сумма	
18210202010061000160	Пеня	3085.00	<a href="#">удалить</a>
18210202010061000000	Штраф	148.30	<a href="#">удалить</a>

[Добавить КБК](#)

[Отправленные заявления](#)    [Получен ответ от ФНС](#)    [Решение принято](#)

Все заявления								
Рег. номер	Наименование	ИНН	КПП	Дата заявления	КБК	Вид платежа	Сумма	Статус
<a href="#">В начало</a> <a href="#">1</a> <a href="#">2</a> <a href="#">3</a> <a href="#">4</a> <a href="#">5</a> <a href="#">Дальше</a>								

ПК Возвраты

localhost:8080/pkv-1.0-SNAPSHOT/pages/upfr.jsp

Логин: Gavrilova  
 Район: Белгород  
[Выход](#)

**Внести данные из заявления плательщика**

Регистрационный номер:

Наименование/ФИО плательщика:

ИНН:

КПП:

Дата заявления:

КБК	Вид платежа	Сумма
<a href="#">Добавить КБК</a>		

[Отправленные заявления](#)    [Получен ответ от ФНС](#)    [Решение принято](#)

Все заявления								
Рег. номер	Наименование	ИНН	КПП	Дата заявления	КБК	Вид платежа	Сумма	Статус
15487959	Российский алюминий	500100732259	773301001	2015-09-11	18210202010061000000	Штраф	148.30	Отправлено в ОПФР
					18210202010061000160	Пеня	3085.00	

[В начало](#)    [1](#)    [2](#)    [3](#)    [4](#)    [5](#)    [Дальше](#)

ПК Возвраты

localhost:8080/pkv-1.0-SNAPSHOT/pages/opfr.jsp

Логин: Morozov  
Район: ОПФР

[Статистика](#) [Выход](#)

УПФР			ОПФР	
Регистрационный номер: 15487959 Наименование: Российский алюминий ИНН: 500100732259 КПП: 773301001 Дата заявления: 2015-09-11 Статус: 1			Номер запроса: <input type="text"/> Дата направления в ФНС: <input type="text" value="дд.мм.гггг"/> <a href="#">Заявление</a> <input type="button" value="Добавить"/>	
КБК	Вид платежа	Сумма		
18210202010061000000	Штраф	148.30		
18210202010061000160	Пеня	3085.00		

В начало 1 2 3 4 5 [Дальше](#)

ПК Возвраты

localhost:8080/pkv-1.0-SNAPSHOT/pages/opfr.jsp

Логин: Morozov  
Район: ОПФР

[Статистика](#) [Выход](#)

УПФР			ОПФР	
Регистрационный номер: 15487959 Наименование: Российский алюминий ИНН: 500100732259 КПП: 773301001 Дата заявления: 2015-09-11 Статус: 1			Номер запроса: <input type="text" value="21564"/> Дата направления в ФНС: <input type="text" value="01.06.2018"/>	
			<a href="#">Заявление</a> <input type="button" value="Добавить"/>	
	КБК	Вид платежа	Сумма	
	18210202010061000000	Штраф	148.30	
	18210202010061000160	Пеня	3085.00	

В начало    1    2    3    4    5    [Дальше](#)

ПК Возвраты

localhost:8080/pkv-1.0-SNAPSHOT/pages/opfr.jsp

Логин: Morozov  
 Район: ОПФР  
 Выход

Статистика

УПФР      ОПФР

Регистрационный  
 Наименование:  
 ИНН: 500100732  
 КПП: 773301001  
 Дата заявления:  
 Статус: 2

КБК  
 1821020201006  
 1821020201006

Сумма

[Добавить](#) X

Название	Файл	Выбор
041009- PASSWORDS__2018_04_06_	123.docx	<a href="#">Выбрать</a>
17 IndPlan adapt Menjshnikov.doc	ihvolpihva.doc	<a href="#">Выбрать</a>
17 IndPlan adapt Borhsenko.doc	slon.doc	<a href="#">Выбрать</a>
slon3.doc	слон3	<a href="#">Выбрать</a>
osnova.docx	osnova.docx	<a href="#">Выбрать</a>
Шаблон.docx	SHablon.docx	<a href="#">Выбрать</a>



ПК Возвраты

localhost:8080/pkv-1.0-SNAPSHOT/pages/opfr.jsp#

Логин: Morozov  
Район: ОПФР

[Статистика](#) [Выход](#)

УПФР			ОПФР		
Регистрационный номер: 15487959 Наименование: Российский алюминий ИНН: 500100732259 КПП: 773301001 Дата заявления: 2015-09-11 Статус: 2			Номер ответа от ФНС: <input type="text" value="1547987"/> Дата ответа от ФНС: <input type="text" value="04.06.2018"/> Документ: <a href="#">osnova.docx</a>		
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
КБК	Вид платежа	Сумма			
18210202010061000000	Штраф	148.30			
18210202010061000160	Пеня	3085.00			

[Заявление](#)

В начало 1 2 3 4 5 Далее

ПК Возвраты

localhost:8080/pkv-1.0-SNAPSHOT/pages/opfr.jsp#

Логин: Morozov  
Район: ОПФР

[Статистика](#) [Выход](#)

УПФР			ОПФР		
Регистрационный номер: 15487959 Наименование: Российский алюминий ИНН: 500100732259 КПП: 773301001 Дата заявления: 2015-09-11 Статус: 3			Номер запроса: 3546 Дата направления в ФНС: 2018-06-01 Номер ответа от ФНС: 1547987 Дата ответа от ФНС: 2018-06-04 Документ: <a href="#">osnova.docx</a>		
КБК	Вид платежа	Сумма	КБК	Вид платежа	Сумма
18210202010061000000	Штраф	148.30	18210202010061000000	Штраф	148.30
18210202010061000160	Пеня	3085.00	18210202010061000160	Пеня	3085

В начало 1 2 3 4 5 [Дальше](#)

ПК Возвраты

localhost:8080/pkv-1.0-SNAPSHOT/pages/upfr.jsp

Логин: Gavrilova  
Район: Белгород [Выход](#)

**Внести данные из заявления плательщика**

Регистрационный номер:

Наименование/ФИО плательщика:

ИНН:

КПП:

Дата заявления:

КБК	Вид платежа	Сумма
<a href="#">Добавить КБК</a>		

[Отправленные заявления](#)    [Получен ответ от ФНС](#)    [Решение принято](#)

УПФР	ОПФР	Решение																		
Регистрационный номер: 15487959 Наименование: Российский алюминий ИНН: 500100732259 КПП: 773301001 Дата заявления: 2015-09-11 Статус: 3	Номер запроса: 3546 Дата направления в ФНС: 2018-06-01 Номер ответа от ФНС: 1547987 Дата ответа от ФНС: 2018-06-04 Документ: <a href="#">osnova.docx</a>	Номер решения: <input type="text"/> Дата решения: <input type="text" value="дд.мм.гггг"/> <input type="button" value="Решение не пр"/> <input type="button" value="Добавить реше"/>																		
<table border="1"> <thead> <tr> <th>КБК</th> <th>Вид платежа</th> <th>Сумма</th> </tr> </thead> <tbody> <tr> <td>18210202010061000000</td> <td>Штраф</td> <td>148.30</td> </tr> <tr> <td>18210202010061000160</td> <td>Пеня</td> <td>3085.00</td> </tr> </tbody> </table>	КБК	Вид платежа	Сумма	18210202010061000000	Штраф	148.30	18210202010061000160	Пеня	3085.00	<table border="1"> <thead> <tr> <th>КБК</th> <th>Вид платежа</th> <th>Сумма</th> </tr> </thead> <tbody> <tr> <td>18210202010061000000</td> <td>Штраф</td> <td>148.30</td> </tr> <tr> <td>18210202010061000160</td> <td>Пеня</td> <td>3085</td> </tr> </tbody> </table>	КБК	Вид платежа	Сумма	18210202010061000000	Штраф	148.30	18210202010061000160	Пеня	3085	
КБК	Вид платежа	Сумма																		
18210202010061000000	Штраф	148.30																		
18210202010061000160	Пеня	3085.00																		
КБК	Вид платежа	Сумма																		
18210202010061000000	Штраф	148.30																		
18210202010061000160	Пеня	3085																		

[В начало](#)    [1](#)    [2](#)    [3](#)    [4](#)    [5](#)    [Дальше](#)

ПК Возвраты

localhost:8080/pkv-1.0-SNAPSHOT/pages/upfr.jsp

Логин: Gavrilova  
Район: Белгород  
[Выход](#)

**Внести данные из заявления плательщика**

Регистрационный номер:

Наименование/ФИО плательщика:

ИНН:

КПП:

Дата заявления:

КБК	Вид платежа	Сумма
<a href="#">Добавить КБК</a>		

[Отправленные заявления](#)    [Получен ответ от ФНС](#)    [Решение принято](#)

УПФР	ОПФР	Решение																		
Регистрационный номер: 15487959 Наименование: Российский алюминий ИНН: 500100732259 КПП: 773301001 Дата заявления: 2015-09-11 Статус: 4	Номер запроса: 3546 Дата направления в ФНС: 2018-06-01 Номер ответа от ФНС: 1547987 Дата ответа от ФНС: 2018-06-04 Документ: <a href="#">osnova.docx</a>	Номер решения: 354687 Дата решения: 2018-06-13																		
<table border="1"> <thead> <tr> <th>КБК</th> <th>Вид платежа</th> <th>Сумма</th> </tr> </thead> <tbody> <tr> <td>18210202010061000000</td> <td>Штраф</td> <td>148.30</td> </tr> <tr> <td>18210202010061000160</td> <td>Пеня</td> <td>3085.00</td> </tr> </tbody> </table>	КБК	Вид платежа	Сумма	18210202010061000000	Штраф	148.30	18210202010061000160	Пеня	3085.00	<table border="1"> <thead> <tr> <th>КБК</th> <th>Вид платежа</th> <th>Сумма</th> </tr> </thead> <tbody> <tr> <td>18210202010061000000</td> <td>Штраф</td> <td>148.30</td> </tr> <tr> <td>18210202010061000160</td> <td>Пеня</td> <td>3085</td> </tr> </tbody> </table>	КБК	Вид платежа	Сумма	18210202010061000000	Штраф	148.30	18210202010061000160	Пеня	3085	
КБК	Вид платежа	Сумма																		
18210202010061000000	Штраф	148.30																		
18210202010061000160	Пеня	3085.00																		
КБК	Вид платежа	Сумма																		
18210202010061000000	Штраф	148.30																		
18210202010061000160	Пеня	3085																		

[В начало](#)    [1](#)    [2](#)    [3](#)    [4](#)    [5](#)    [Дальше](#)