

Содержание

Перечень сокращений, условных обозначений, терминов	9
Введение.....	10
1 Техническое задание на создание системы	12
1.1 Назначение и цели создания системы	12
1.2 Характеристика объекта автоматизации.....	12
1.2.1 Общее описание	12
1.2.2 Структура и принципы функционирования	14
1.2.3 Существующая информационная система и ее недостатки	15
1.2.4 Анализ аналогичных разработок	16
1.2.5 Актуальность проводимой разработки	16
1.3 Общие требования к системе	16
1.3.1 Требования к структуре и функционированию системы.....	17
1.3.2 Дополнительные требования	18
1.4 Требования к функциям, выполняемым системой.....	18
1.4.1 Сбор информации.....	18
1.4.2 Функция привязки данных	19
1.4.3 Функция начальной поставки данных организации.....	19
1.4.4 Формирование списка интеграционных запросов.....	19
1.4.5 Формирование ссылок на файлы документов организации	20
1.4.6 Функция обеспечения обмена данными	20
1.4.7 Функция отображения списка запросов	20
1.4.8 Функция отображения истории выполнения запросов	20
1.4.9 Функция отображения статистики	21
1.5 Требования к видам обеспечения	21
1.5.1 Требования к алгоритмическому обеспечению	21
1.5.2 Требования к информационному обеспечению.....	22
1.5.3 Требования к программному обеспечению.....	22

Подп. и дата								
Инв. № дубл.								
Взаим. инв. №								
Подп. и дата								
Инв. № подл.	Изм.	Лист	№ докум.	Подп.	Дата	<p><i>ВКР-УЛГТУ-09.03.02-11/286-2015 ПЗ</i></p> <p><i>Пояснительная записка</i></p> <p><i>ИСТ88-41</i></p>		
	Разраб.	Желепов А.С.						
	Пров.	Кандаулов В.М.						
	Н.контр.	Степчева З.В.						
	Утв.							

5.2.1	Структура прикладного программного обеспечения	56
5.2.2	Модуль обмена данными.....	58
5.2.3	Модуль привязки данных	61
5.2.4	Модуль сбора данных	61
5.2.5	Модуль формирования запросов	63
5.2.6	Модуль формирования файлового хранилища	63
5.2.7	Модуль обеспечения бесперебойной передачи данных.....	64
5.2.8	Вспомогательный модуль.....	65
5.2.9	Модуль перехвата сообщений	66
5.2.10	Модуль конфигурации.....	67
5.2.11	Модуль панели управления интеграцией	68
5.3	Особенности эксплуатации и сопровождения системы	68
5.4	Интерфейс пользователя с системой	70
5.4.1	Руководство пользователя.....	70
5.4.2	Инсталляция и настройка	76
5.4.3	Исключительные ситуации и их обработка.....	76
6	Тестирование системы	77
6.1	Условия тестирования.....	77
6.2	Процесс тестирования	77
6.3	Исходные данные для контрольных примеров	78
6.4	Результаты тестирования.....	81
7	Экономический раздел.....	82
7.1	Расчет показателя трудоемкости для разработанного программного продукта	82
7.2	Расчет затрат на материальные ресурсы и сырье.....	84
7.3	Расчет затрат на оплату труда	86
7.4	Расчет отчислений в социальные фонды	87
7.5	Расчет амортизации оборудования.....	87

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
					Изм.	Лист	№ докум.	Подп.	Дата	

7.6	Расчет себестоимости разработки.....	88
7.7	Расчет плановой прибыли.....	88
7.8	Расчет основных технико-экономических показателей и эффективности использования программного продукта	89
8	Безопасность и экологичность проекта.....	91
8.1	Исходные данные	91
8.2	Перечень нормативных документов и актов	92
8.3	Анализ потенциальных опасностей.....	93
8.3.1	Анализ вредных и опасных производственных факторов.....	94
8.3.2	Анализ воздействия на окружающую среду	101
8.3.3	Анализ возможных чрезвычайных ситуаций	101
8.4	Мероприятия по охране труда.....	103
8.4.1	Мероприятия по обеспечению комфортных условий труда	104
8.4.2	Мероприятия по защите от опасных и вредных производственных факторов	106
8.5	Мероприятия по охране окружающей среды	106
8.6	Мероприятия по защите от чрезвычайных ситуаций	107
8.7	Расчетная часть	108
8.7.1	Расчет уровня шума на рабочем месте.....	108
8.7.2	Расчет величины освещенности рабочего пространства	109
8.8	Оценка эффективности принятых решений	111
	Заключение	112
	Список литературы	114
	Приложение А	116
	Приложение Б	134

Инв. № подл.	Взам. инв. №	Инв. № дудл.	Подп. и дата							
					ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

Перечень сокращений, условных обозначений, терминов

АИС – автоматизированная информационная система;

БД – база данных;

ЖКХ – жилищно-коммунальное хозяйство;

ИС – информационная система;

ОУ – объектовый учет;

ПО – программное обеспечение;

Реформа – федеральный портал «Реформа ЖКХ»;

СУБД – система управления базами данных;

УК – управляющая компания;

ЯВУ – язык высокого уровня;

GUID – globally unique identifier;

ID – идентификатор;

IDE – интегрированная среда разработки;

ORM – Object Relational Mapping;

SOAP – Simple Object Access Protocol;

WCF – Windows Communication Foundation.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

Введение

В последнее время сфера жилищно-коммунального хозяйства стала востребованной площадкой для разработки информационных проектов. Основной целью этих программных продуктов является автоматизация процессов домоуправления и контроля решений текущих коммунальных проблем и задач.

Число информационных систем, которыми пользуются организации сферы ЖКХ, постоянно увеличивается. Такая тенденция имеет как положительные, так и отрицательные стороны. Положительная составляющая проявляется в возможности автоматизации процессов сферы ЖКХ. Отрицательная сторона заключается в том, что управляющие компании часто не раскрывают данные процессов домоуправления. В первую очередь, эта проблема сказывается на гражданах-собственниках жилья, так как в вопросах сферы ЖКХ они всегда заинтересованы в получении свежих и актуальных данных.

Правительство Российской Федерации, решая проблему раскрытия информации, в сентябре 2010 года издало постановление № 731 «Об утверждении стандарта раскрытия информации организациями, осуществляющими деятельность в сфере управления многоквартирными домами». Согласно постановлению был разработана информационная система «Реформа ЖКХ». Основная функциональность портала заключается в загрузке и отображении данных:

- а) процессов домоуправления, которые загружаются организациями;
- б) всероссийского рейтинга управляющих компаний;
- в) всероссийского мониторинга решения задачи переселения граждан из аварийного и ветхого жилья.

Исторически сложилось, что управляющие компании часто пользуются информационными решениями частных фирм для автоматизации процессов работы. В Ульяновской области и во многих других регионах Российской Федерации компаниями сферы ЖКХ используется продукт фирмы АИС: Город

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	
Изм.	Лист
№ докум.	Подп.
Дата	
ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ	
Лист	

«АИС: Объектовый учет». Данная информационная система многофункциональна, при этом один из ее модулей полностью реализует процесс управления многоквартирными домами.

Задача интеграции данных процессов домоуправления на федеральный портал «Реформа ЖКХ» была поставлена по многочисленным просьбам управляющих компаний, использующих систему объектового учета. Причина постановки задачи заключается в обязанности управляющих компаний раскрывать свои данные на федеральном портале согласно постановлению Правительства.

Информационная система «Реформа ЖКХ» поддерживает два режима раскрытия данных:

- а) ручной, когда оператор управляющей компании самостоятельно следит за новизной данных и обновляет их;
- б) автоматический, при котором возможно создание системы взаимодействия федерального портала и внешней системы на базе специального API-интерфейса.

Первым способом раскрытия данных пользуется наибольшее число управляющих компаний. Однако поддерживать полную синхронизацию данных для «АИС: Объектовый учет» и федерального портала вручную практически невозможно. Это объясняется тем, что процессы домоуправления динамичны, и данные по ним часто претерпевают изменения. Поэтому сопоставление информации в двух ИС – длительный и трудоемкий процесс.

Разработанная подсистема интеграции решает эту проблему и обеспечивает своевременное обновление данных управляющих компаний на федеральном портале.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

1 Техническое задание на создание системы

В данном разделе приводится техническое задание на разработку подсистемы интеграции данных с федеральной системой «Реформа ЖКХ» для «АИС: Объектовый учет».

1.1 Назначение и цели создания системы

Назначением разработки выпускной квалификационной работы является обеспечение раскрытия данных домоуправления управляющими компаниями на федеральном портале «Реформа ЖКХ».

Целями создания программного продукта являются:

- а) автоматизация процесса загрузки данных управляющих компаний из «АИС: Объектовый учет» на сайт портала «Реформа ЖКХ»;
- б) создание решения, модули которого в дальнейшем могут быть использованы для налаживания интеграционного взаимодействия с другими ИС.

1.2 Характеристика объекта автоматизации

Объектом автоматизации выпускной квалификационной работы является процесс обмена данными между домоуправляющими компаниями и федеральным порталом «Реформа ЖКХ».

1.2.1 Общее описание

Сфера жилищно-коммунального хозяйства является одной из востребованных площадок для разработки программных продуктов. Это объясняется возможностью программного решения существующих задач автоматизации процессов и предоставления открытого доступа к информации. Особенно остро вопрос прозрачности данных стоит для собственников жилья, которые заинтересованы в квалифицированных и своевременных решениях коммунальных проблем управляющими компаниями.

Подп. и дата	
Инв. № аудл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

					ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		

Правительство Российской Федерации поддерживает инициативы по внедрению информационных технологий в сфере ЖКХ. Законодательно эта поддержка оформлена постановлением № 731 "Об утверждении стандарта раскрытия информации организациями, осуществляющими деятельность в сфере управления многоквартирными домами" [5]. Согласно принятому постановлению управляющие компании обязаны раскрывать информацию о реальном состоянии процессов домоуправления на федеральном портале «Реформа ЖКХ».

Реформа представляет собой интернет-ресурс, при пользовании, которым пользователь имеет возможность ознакомиться с данными:

- а) процессов домоуправления, которые загружаются организациями;
- б) всероссийского рейтинга управляющих компаний;
- в) всероссийского мониторинга решения задачи переселения граждан из аварийного и ветхого жилья.

Для загрузки данных процессов домоуправления на информационный портал «Реформа ЖКХ» компании необходимо подать заявку на регистрацию, содержащую полную информацию об организации. После проверки данных администратором портала организации предоставляется доступ к личному кабинету. В нем оператор управляющей компании имеет возможность загружать и изменять данные:

- а) конструктивных элементов, состояния и параметров объекта;
- б) списка домов, находящихся в управлении;
- в) прикрепленных файлов нормативных актов и документов.

Федеральный проект «Реформа ЖКХ» не единственный, который решает задачи автоматизации процессов домоуправления. В настоящее время многие частные фирмы разрабатывают и продвигают свои информационные продукты в сфере жилищно-коммунального хозяйства. Эти программные решения зачастую более функциональны, чем относительно новый проект «Реформа ЖКХ».

Управляющие организации используют такие программные продукты на

Инф. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					

протяжении многих лет. Соответственно, передача данных на портал приводит к сложности синхронизации предоставляемой информации. Проблема возникает во время одновременной загрузки данных в несколько источников (в региональную систему и в Реформу), что приводит к дополнительным временным затратам на сопоставление данных. Разработчики федеральной системы предусмотрели этот вариант событий и предоставили API для интеграции с другими ИС.

Для организации обмена данными посредством API-интерфейса создатели портала использовали технологию web-служб [9, 10]. Такой модуль приложения предоставляет разработчикам возможность удаленного взаимодействия с источником данных, в данном случае с базой данных федеральной ИС. Передача информации осуществляется через протокол обмена xml-сообщениями SOAP. Web-служба поддерживает более двадцати API-методов, которые подразделяются на три типа:

- а) GET-методы, реализующие возможность получения данных с Реформы;
- б) SET-методы, обеспечивающие непосредственную загрузку данных на сайт портала;
- в) Методы аутентификации, производящие открытие сессии для обмена данными.

1.2.2 Структура и принципы функционирования

Управляющая организация, использующая региональную систему объектового учета «АИС: Объектовый учет» должна иметь возможность подать заявку на обмен данными с информационной системой «Реформа ЖКХ» из личного кабинета. Поданная заявка должна быть рассмотрена администратором федерального портала и может быть принята или отклонена. Статус заявки должен быть виден из личного кабинета пользователя управляющей компании.

При успешной регистрации должен происходить процесс инициализации компании на портале «Реформа ЖКХ», подразумевающий загрузку всех данных организации по процессам домоуправления.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					

Пользователь управляющей компании часто вносит изменения в данные по домоуправлению. Например, добавление нового конструктивного элемента для дома, загрузка документа по капитальному ремонту объекта и т.д. Эти изменения должны отслеживаться подсистемой интеграции, которая оповещает о них информационный портал «Реформа ЖКХ», обеспечивая загрузку или синхронизацию данных.

Для просмотра проводимых операций по обмену данными пользователю должен быть предоставлен доступ к списку интеграционных запросов его организации. По каждому из них должна отображаться история вызова, где можно увидеть тексты отправляемых и принимаемых SOAP-сообщений, статус и время выполнения запроса на обмен данными. Если запрос выполнен с ошибкой, то пользователь должен иметь возможность внести соответствующие коррективы в свои данные, согласно информации, содержащейся в сообщении об ошибке или сообщить, о возникшей проблеме службе поддержки проекта «АИС: Объектовый учет».

1.2.3 Существующая информационная система и ее недостатки

На текущий момент организации, производящие выгрузку данных по процессам домоуправления на федеральный портал «Реформа ЖКХ» производят все операции вручную.

Управляющие компании, использующие информационную систему объектового учета «АИС: Объектовый учет», делают двойную работу, одновременно производя сохранение данных в региональной и федеральной системах.

Основной недостаток помимо траты большого количества времени на загрузку заключается в сопоставлении данных в обеих информационных системах. Процессы домоуправления динамичны, поэтому сверять данные приходится часто из-за постоянно меняющейся информации.

Инф. № подл.	Подп. и дата	Взам. инб. №	Инб. № дубл.	Подп. и дата						
					ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

Инб. № подл.	Подп. и дата	Взам. инб. №	Инб. № дубл.	Подп. и дата

Формат А4

«АИС: Объектовый учет».

1.3.1 Требования к структуре и функционированию системы

Приложение интеграции должно быть разработано с использованием модульного подхода к созданию программного обеспечения. Применение этой методологии объясняется необходимостью дальнейшего построения интеграционных подсистем для «АИС: Объектовый учет» с другими информационными решениями сферы ЖКХ помимо Реформы.

Преимущество модульного подхода заключается в переиспользовании компонентов, что дает возможность настраивать новую систему интеграции из готовых программных модулей и библиотек.

Основными модулями, разрабатываемыми в рамках программного решения для автоматизации обмена данными с федеральным порталом «Реформа ЖКХ», должны стать:

- а) модуль сбора данных, осуществляющий поиск и представление в табличном виде информации согласно определенному набору полей данных;
- б) модуль привязки к объекту определенного типа, позволяющий на основании данных, полученных в табличном виде, формировать объект API-класса с любым уровнем вложенностей;
- в) модуль для работы с API-интерфейсом, использующийся для непосредственной поставки данных;
- г) модуль управления интеграцией для личного кабинета пользователя управляющей организации;
- д) модуль сохранения действий пользователя на сайте региональной системы объектового учета. С его помощью должно происходить формирование списка запросов на изменение данных на сайте портала «Реформа ЖКХ»;
- е) модуль интеграции файлов, подразумевающий разработку хранилища

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>

Инб. № подл.	Подп. и дата	Взам. инб. №	Инб. № дубл.	Подп. и дата

Требования к выходным данным: создание записей в таблице запросов на выгрузку/синхронизацию данных с Реформой.

					ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		

1.4.5 Формирование ссылок на файлы документов организации

При запуске функции должно формироваться хранилище, в котором содержатся ссылки на файлы документов, хранящиеся на сервере.

Приоритет выполнения: средний.

Требования к входным данным: управляющая организация.

Требования к выходным данным: выходными данными является таблица, хранящая сведения о файлах документов, выгружаемых управляющей организацией.

1.4.6 Функция обеспечения обмена данными

Функция должна осуществлять выполнение интеграционных API-методов, описанных в таблице запросов, для каждой организации.

Приоритет выполнения: высокий.

Требования к входным данным: таблица запросов, управляющая организация.

Требования к выходным данным: выходные данные представляются ответными SOAP-сообщениями от ИС «Реформа ЖКХ». Они содержат информацию о статусе выполнения каждого запроса и текста возможной ошибки.

1.4.7 Функция отображения списка запросов

Функция должна обеспечивать отображение списка запросов в личном кабинете пользователя управляющей компании.

Приоритет выполнения: средний.

Требование к входным данным: управляющая организация.

Требование к выходным данным: отображение списка запросов на html-странице web-приложения.

1.4.8 Функция отображения истории выполнения запросов

Функция должна обеспечивать отображение истории выполнения каждого

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

е) составление файлового хранилища для интеграции.

1.5.2 Требования к информационному обеспечению

При разработке приложения интеграции необходимо разработать базу данных, которая будет хранить информацию об интеграционных запросах организаций, истории их выполнения и файловом хранилище. Дополнительно должно быть организовано хранение и обновление справочных данных Реформы. Эта информация должна заполняться согласно документации о API-интерфейсе.

1.5.3 Требования к программному обеспечению

Создание приложения интеграции должно производиться инструментальными средствами среды Visual Studio 13 и .NET Framework версии 4.0 [7, 8]. Служба обмена данными должна функционировать на компьютере с операционной системой Windows Server 2008 R2 Standard. В качестве СУБД необходимо использовать Microsoft SQL Server и язык доступа к данным Transact-SQL.

Программный код должен быть хорошо структурирован и понятен в дальнейшем сопровождении, для этого необходимо использовать паттерны программирования. Видимая часть проекта – панель управления интеграцией для пользователей должна быть разработана в соответствии со спецификацией шаблона проектирования Model-View-Controller [12].

1.5.4 Требования к техническому обеспечению

Подсистема интеграции для федерального портала «Реформа ЖКХ» и «АИС: Объектовый учет» должна функционировать на компьютере-сервере. Его технические характеристики соответствуют следующим параметрам:

- а) процессор – Intel(R) Xeon(R) CPU E5-2680 с частотой 2.70GHz;
- б) установленная память (ОЗУ) – 3,08 Гб;
- в) жесткий диск – объемом 1 Тб с интерфейсом обмена данными SATA;
- г) сетевой адаптер пропускной способностью 1 Гбит/с.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

2 Модель исходной информационной системы

В разделе рассматривается функциональная модель подсистемы интеграции и диаграммы ее декомпозиции, разработанные с помощью стандарта моделирования IDEF0.

2.1 IDEF0-модель подсистемы интеграции

На рисунке 2.1 представлена диаграмма функциональной модели подсистемы интеграции для «АИС: Объектовый учет» и федерального портала «Реформа ЖКХ».

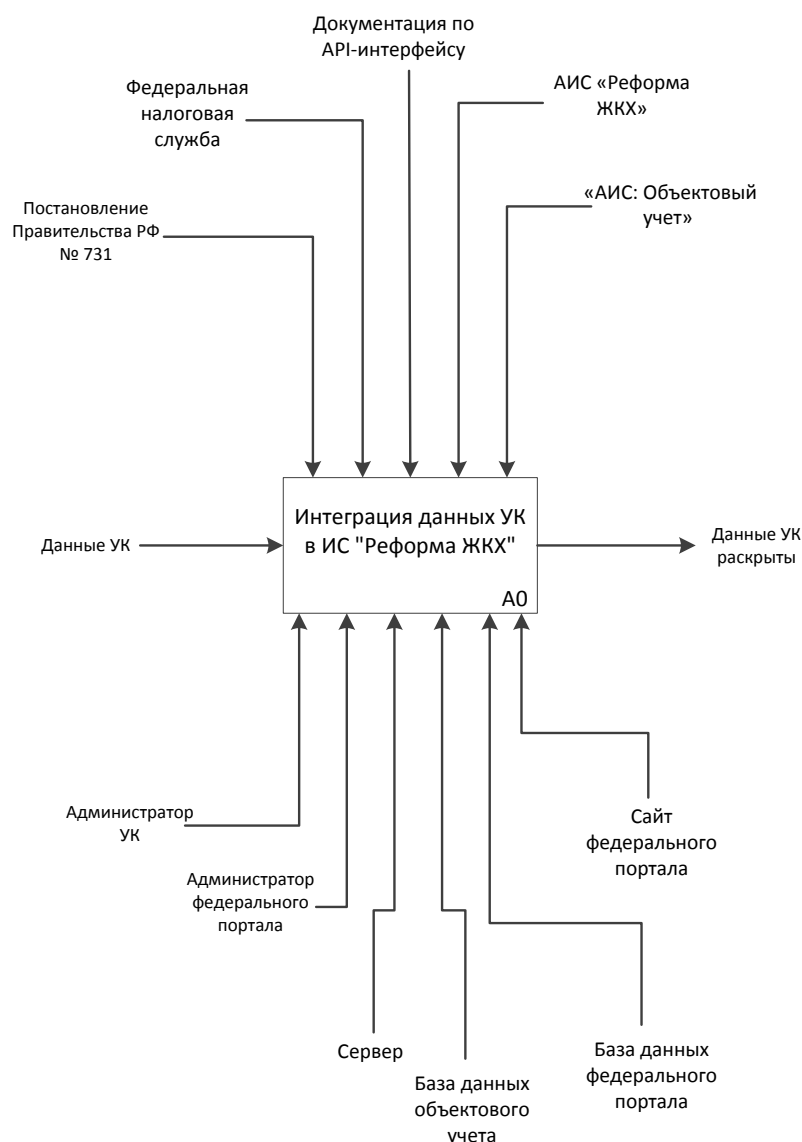


Рисунок 2.1 – IDEF0 диаграмма функциональной модели подсистемы

Назначение подсистемы – это интеграция данных процессов домоуправления управляющих компаний, зарегистрированных в информационной системе «АИС: Объектовый учет», на федеральный портал «Реформа ЖКХ».

Ресурсами, с помощью которых выполняется процесс интеграции данных, являются:

- а) администратор УК;
- б) администратор федерального портала;
- в) сервер, использующийся для развертки подсистемы интеграции;
- г) база данных ОУ, используемая для хранения данных интеграционного процесса;
- д) база данных федерального портала для хранения данных процессов домоуправления организаций.

Механизмами управления являются:

- а) постановление Правительства Российской Федерации № 731, гласящее о том, что управляющие организации обязаны раскрывать данные по домоуправлению на сайте портала «Реформа ЖКХ»;
- б) сайт федеральной налоговой службы, где администратор портала проверяет существование управляющей компании, которая подала заявку на раскрытие своих данных;
- в) документация API-интерфейса, описывающая стандарты и данные для интеграции;
- г) Требования к реализации интеграции портала «Реформа ЖКХ» и «АИС: Объектовый учет».

На рисунке 2.2 представлена диаграмма декомпозиции функциональной модели подсистемы интеграции.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

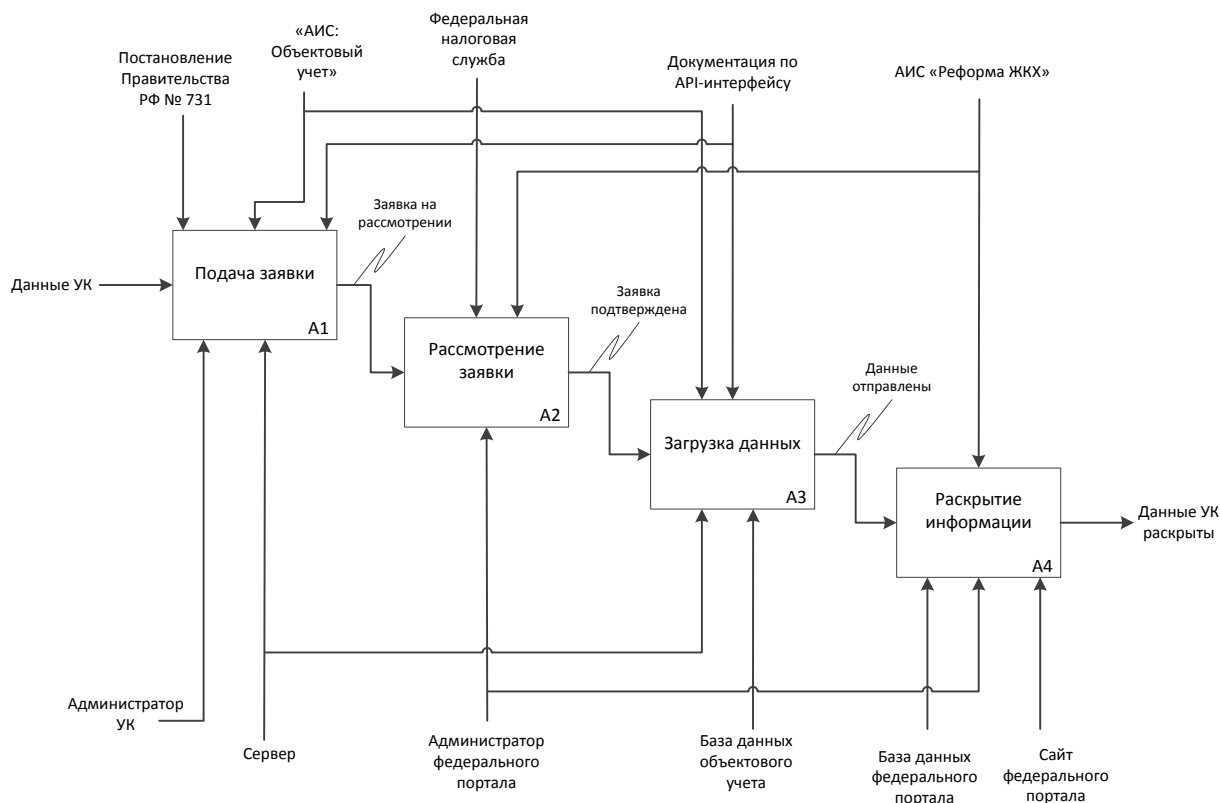


Рисунок 2.2 – IDEF0 диаграмма декомпозиции функциональной модели системы

Процесс интеграции данных состоит из четырех этапов:

- подача заявки администратором управляющей компании о согласии организации раскрывать данные процессов домоуправления на сайте портала;
- осуществление проверки данных организации (ИНН, ОГРН и др.) администратором федерального портала, после которой принимается решение о предоставлении доступа;
- открытие возможности интеграции данных управляющей компании при наличии подтвержденной заявки из системы регионального объектового учета;
- публикация данных УК администратором федерального портала.

На рисунке 2.3 представлена диаграмма декомпозиции процесса «Поддача заявки».

Изм.	Лист	№ докум.	Подп.	Дата
Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № подл.	Подп. и дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Копировал

Формат А4

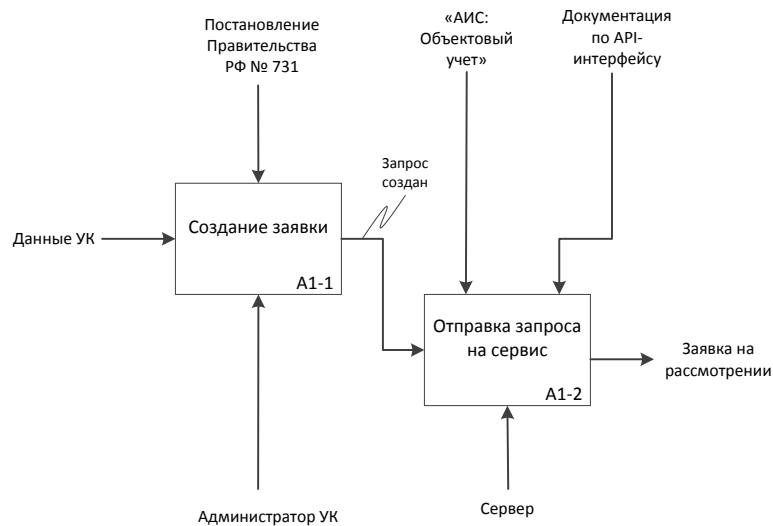


Рисунок 2.3 – IDEF0 диаграмма процесса подачи заявки

Процесс подачи заявки состоит из двух подпроцессов:

- а) оповещение администратором управляющей компании системы объектового учета о том, что организация согласна являться поставщиком данных для федерального портала «Реформа ЖКХ»;
- б) осуществление сервером, на котором развернута подсистема интеграции, отправки данных организации сервису Реформы.

На рисунке 2.4 представлена декомпозиция процесса рассмотрения заявки администратором федерального портала.



Рисунок 2.4 – IDEF0 диаграмма процесса рассмотрения заявки

Подп. и дата	
Инв. № аудл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Процесс рассмотрения заявки состоит из двух подпроцессов:

- а) осуществление проверки существования организации через реестр федеральной налоговой службы;
- б) принятие решения администратором о предоставлении возможности интеграции данных управляющей компании.

На рисунке 2.5 представлена декомпозиция процесса загрузки данных управляющей организации на федеральный портал.

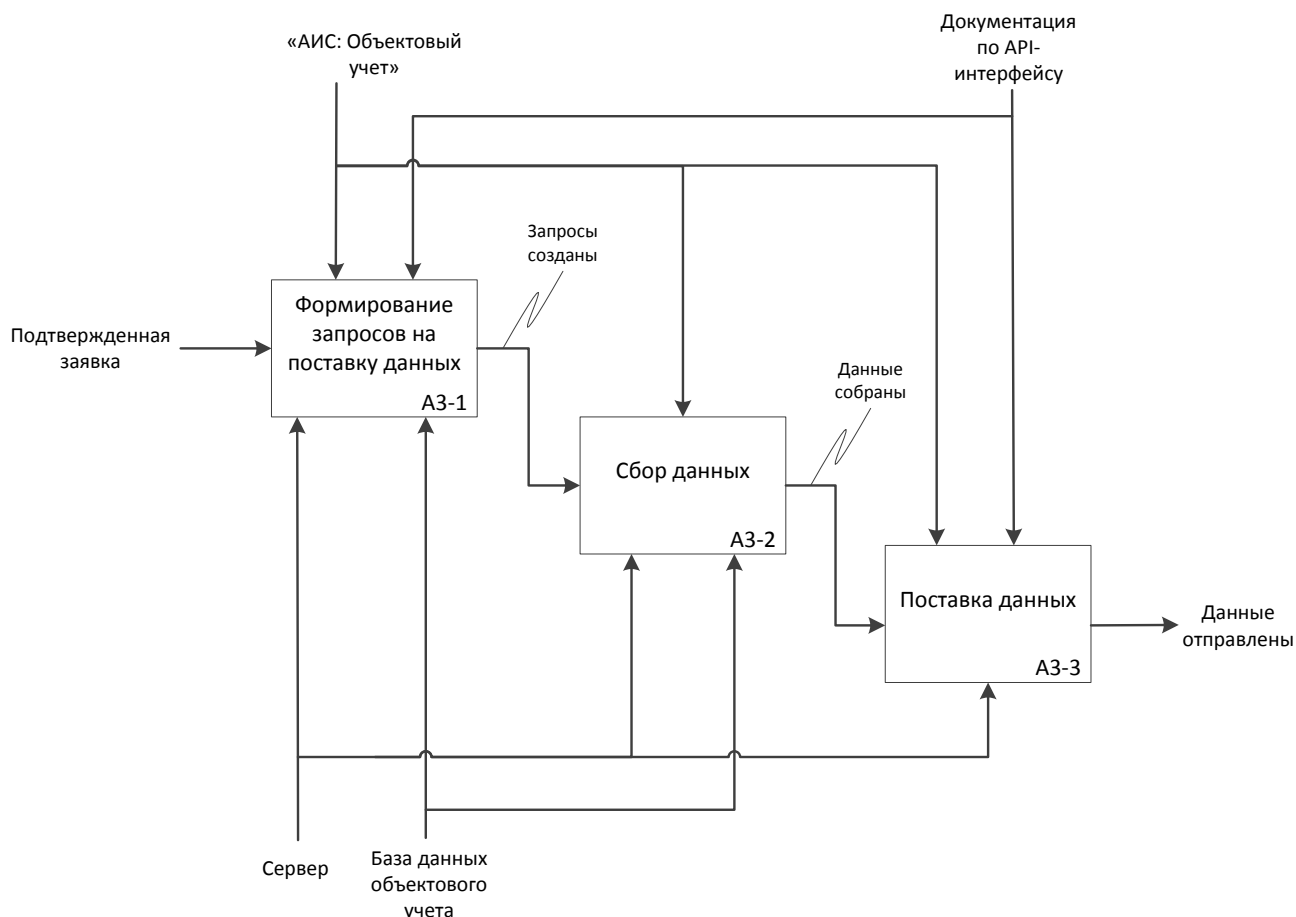


Рисунок 2.5 – IDEF0 диаграмма процесса загрузки данных

Процесс загрузки данных состоит из трех подпроцессов:

- а) формирование запросов на раскрытие данных;
- б) формирование объектов с информацией для пересылки данных на основании запросов;
- в) отправка данных на федеральный портал при помощи методов API-интерфейса.

Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

На рисунке 2.6 представлена диаграмма декомпозиции процесса раскрытия данных управляющей компании на портале «Реформа ЖКХ».

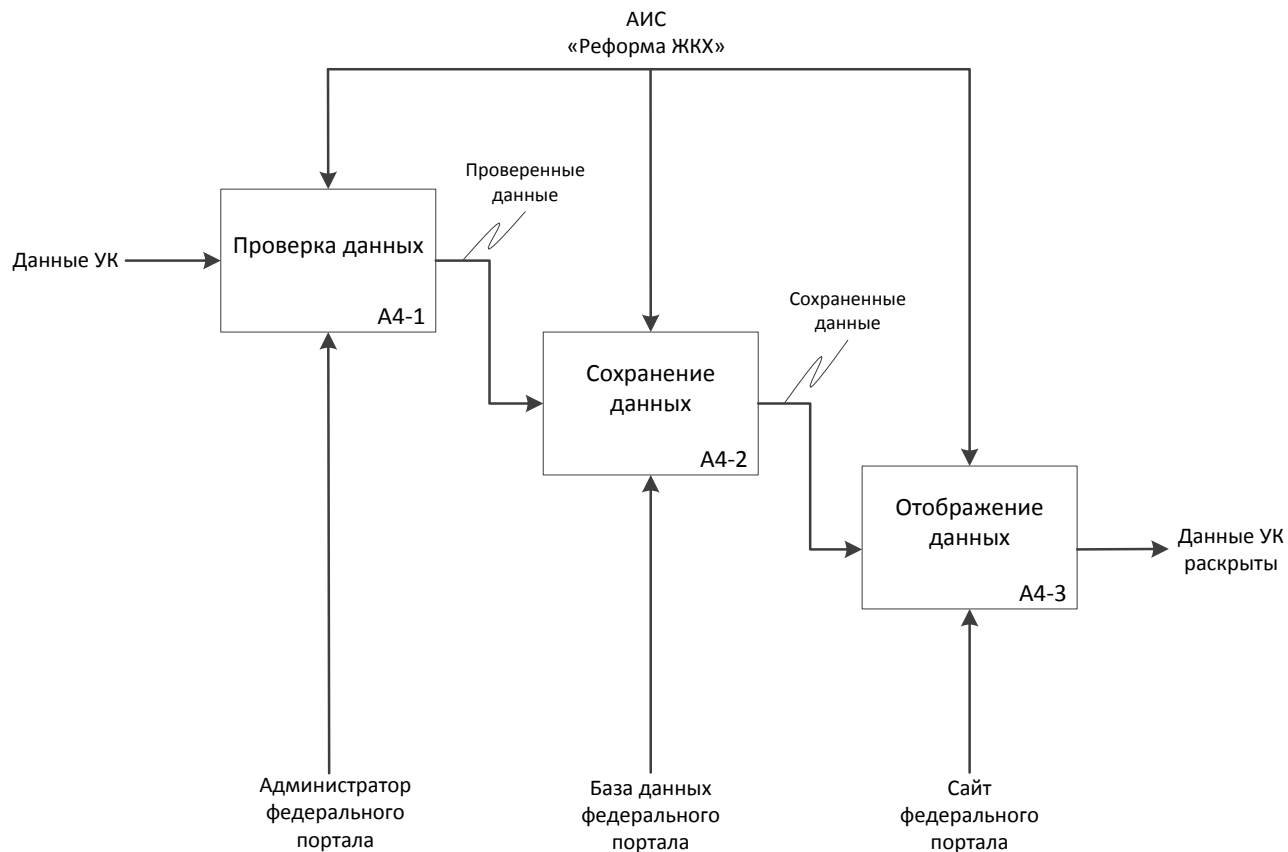


Рисунок 2.6 – IDEF0 диаграмма процесса раскрытия данных управляющей компании

Процесс раскрытия данных управляющей компании состоит из трех подпроцессов:

- а) проверка поставленных данных, осуществляемая администратором федерального портала;
- б) сохранение данных в базу данных федерального портала;
- в) отображение данных УК на сайте федерального портала, где собственники жилья могут ознакомиться с интересующей их информацией процессов домоуправления.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Инб. № подл.	Подп. и дата	Взам. инб. №	Инб. № дубл.	Подп. и дата

В соответствии с требованием технического задания о том, что проект подсистемы интеграции должен быть независимым от объектового учета и перевода проектов на концепцию общей модели данных, окончательный выбор был сделан в пользу использования ORM EF4.

Луст

3.2 Проектирование базы данных

В подразделе приводится подробное описание модели данных подсистемы интеграции, структуры таблиц и представлений.

3.2.1 Физическая модель данных

На рисунке 3.1 представлена физическая модель данных для подсистемы интеграции.

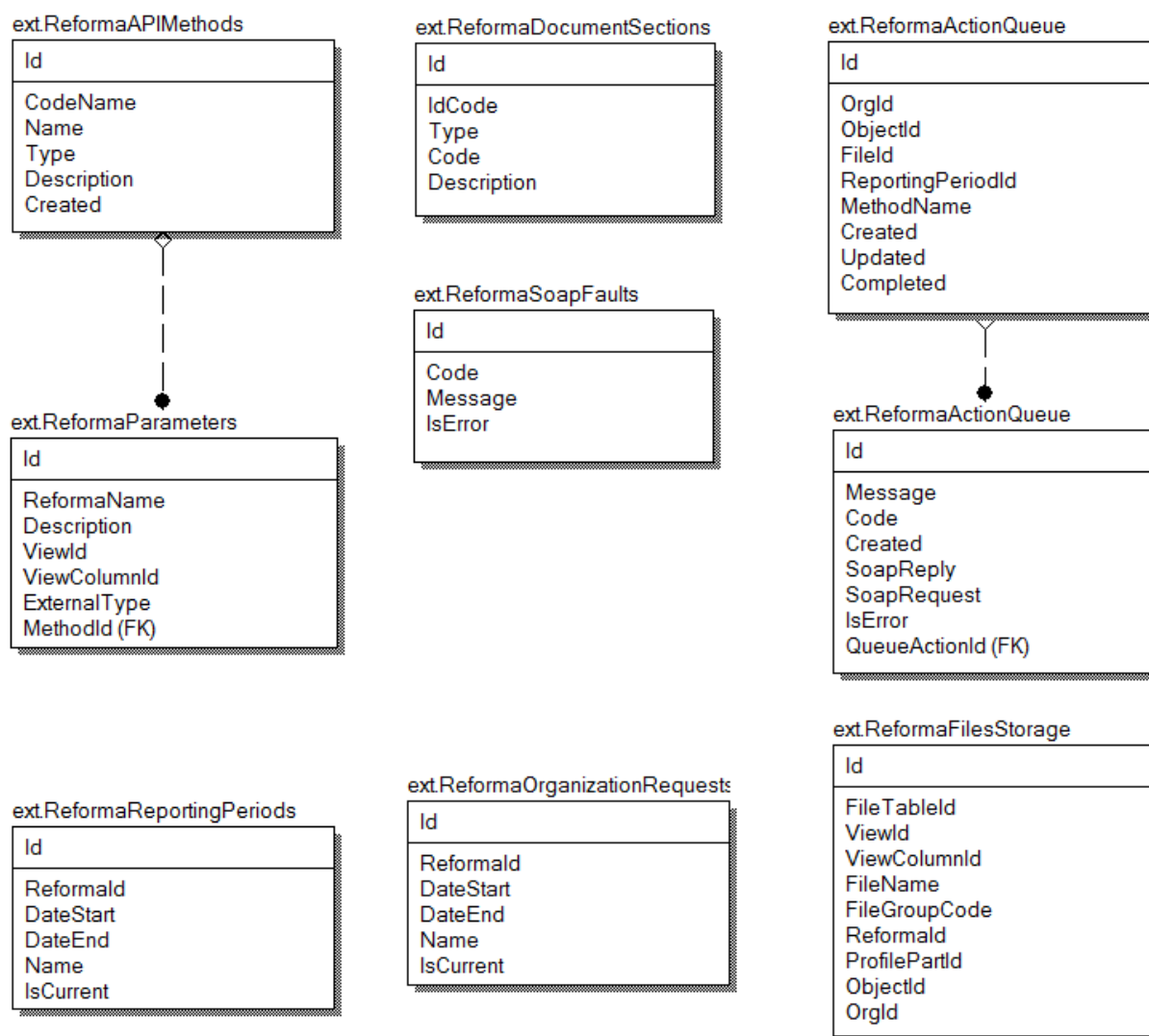


Рисунок 3.1 – Физическая модель базы данных

Представленная структура базы данных может быть подразделена на четыре группы таблиц:

а) таблицы-справочники, содержащие данные из документации по

интеграции с Реформой;

б) таблицы-справочники, обновление которых происходит автоматически с помощью API-методов;

в) таблицы, реализующие модель «Очередь запросов»;

г) служебная таблица, предназначенная для сопоставления хранящихся файлов в базах данных интегрируемых информационных систем.

3.2.1.1 Таблицы для хранения справочных данных

Документация по API-интерфейсу Реформы включает в себя большое количество информации о данных и структурах для интеграции, описании API-методов и т.д. Данные сведения хранятся в специальных таблицах-справочниках:

а) ext.ReformaAPIMethods. Таблица содержит описания методов API-интерфейса.

б) ext.ReformaParameters. Таблица содержит описания набора параметров данных для каждого API-метода.

в) ext.ReformaDocumentSections. Таблица хранит информацию о разделах анкет дома и организации.

г) ext.ReformaSoapFaults. Таблица содержит коды ошибок и их описания, которые могут возникнуть в процессе интеграционного обмена.

В таблице 3.1 рассматривается структура SQL-таблицы ext.ReformaAPIMethods.

Таблица 3.1 – Структура таблицы ext.ReformaAPIMethods

Наименование поля	Тип поля	Описание поля
1. Id	PK, INT, NOT NULL	ID API-метода.
2. CodeName	NVARCHAR(50), NOT NULL	Код API-метода (используется в программном коде).
3. Name	NVARCHAR(200), NOT NULL	Наименование API-метода.
4. Type	NVARCHAR(50), NOT NULL	Тип API-метода.
5. Created	DATETIME2(2), NOT NULL	Дата создания записи.
6. Description	NVARCHAR(1500), NULL	Описание API-метода.

В таблице 3.2 рассматривается структура SQL-таблицы ext.ReformaParameters.

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

/лист

Таблица 3.2 – Структура таблицы ext.ReformaParameters

Наименование поля	Тип поля	Описание поля
1. Id	PK, INT, NOT NULL	ID параметра.
2. MethodId	FK, INT, NOT NULL	ID API-метода, который используется для интеграции параметра.
3. ReformaName	NVARCHAR(200), NOT NULL	Наименование параметра в документации.
4. Description	NVARCHAR(400), NULL	Описание параметра.
5. ViewId	INT, NULL	ID представления, содержащее параметр.
6. ViewColumn	INT, NULL	ID поля, в котором хранится значение параметра.
7. ExternalType	NVARCHAR(100), NULL	Тип параметра, описанный в документации.

В таблице 3.3 рассматривается структура SQL-таблицы ext.ReformaDocumentSections.

Таблица 3.3 – Структура таблицы ext. ReformaDocumentSections

Наименование поля	Тип поля	Описание поля
1. Id	PK, INT, NOT NULL	ID раздела анкеты.
2. IdCode	INT, NOT NULL	ID анкеты, определенный в Реформе.
3. Type	NVARCHAR(100), NOT NULL	Тип анкеты.
4. Code	NVARCHAR(100), NOT NULL	Код раздела анкеты (используется в программном коде).
5. Description	NVARCHAR(500), NOT NULL	Описание раздела анкеты.

В таблице 3.4 рассматривается структура SQL-таблицы ext.ReformaSoapFaults.

Таблица 3.4 – Структура таблицы ext. ReformaSoapFaults

Наименование поля	Тип поля	Описание поля
1. Id	PK, INT, NOT NULL	ID сообщения.
2. Code	NVARCHAR(50), NOT NULL	Код сообщения.
3. Message	NVARCHAR(500), NOT NULL	Текст сообщения.
4. IsError	BIT, NOT NULL	Статус сообщения.

Некоторые справочные данные периодически обновляются администраторами портала «Реформа ЖКХ». Для своевременной синхронизации этих данных были созданы таблицы, которые перезаписываются каждый раз при запуске приложения подсистемы интеграции. Этими таблицами являются:

а) ext.ReformaReportingPeriods. Таблица содержит информацию об отчетных

Подп. и дата

Инв. № дубл.

Взам. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

периодах управляющих организаций.

б) ext.ReformaOrganizationRequests. Таблица содержит информацию о статусах заявок от организаций на раскрытие данных по процессам домоуправления.

В таблице 3.5 рассматривается структура SQL-таблицы ext.ReformaReportingPeriods.

Таблица 3.5 – Структура таблицы ext. ReformaReportingPeriods

Наименование поля	Тип поля	Описание поля
1. Id	PK, INT, NOT NULL	ID отчетного периода.
2. ReformaId	INT, NOT NULL	ID отчетного периода в базе данных Реформы.
3. DateStart	DATETIME2(2), NOT NULL	Время начала отчетного периода.
4. DateEnd	DATETIME2(2), NOT NULL	Время окончания отчетного периода.
5. Name	NVARCHAR(200), NOT NULL	Наименование отчетного периода.
6. IsCurrent	BIT, NOT NULL	Обозначает текущий отчетный период.

В таблице 3.6 рассматривается структура SQL-таблицы ext.ReformaOrganizationRequests.

Таблица 3.6 – Структура таблицы ext. ReformaOrganizationRequests

Наименование поля	Тип поля	Описание поля
1. Inn	PK, NVARCHAR(200), NOT NULL	ИНН организации.
2. Status	INT, NOT NULL	Статус заявки на раскрытие данных.
3. Created	DATETIME2(2), NOT NULL	Время первого рассмотрения заявки.
4. Updated	DATETIME2(2), NULL	Время последнего рассмотрения заявки.

3.2.1.2 Таблицы для хранения данных процесса интеграции

Для стандартизации обмена данными между федеральной системой «Реформа ЖКХ» и «АИС: Объектовый учет» была разработана модель «Очередь запросов». Данные, которыми оперирует программная реализация модели, содержатся в таблицах:

а) ext.ReformaActionQueue. Таблица хранит запросы пользователей на

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. № дубл.	Подп. и дата						/Лист
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					

интеграцию данных;

б) ext.ReformaActionQueueLog. Таблица хранит данные ответных от сервиса Реформы сообщений-статусов выполнения API-методов.

Таблицы объединены связью типа «один-ко-многим». Это позволяет неоднократно выполнять конкретный запрос и фиксировать историю его вызова.

В таблице 3.7 рассматривается структура SQL-таблицы ext.ReformaActionQueue.

Таблица 3.7 – Структура таблицы ext. ReformaActionQueue

Наименование поля	Тип поля	Описание поля
1. Id	PK, INT, NOT NULL	ID запроса.
2. OrgId	INT, NOT NULL	ID организации.
3. ObjectId	INT, NULL	ID объекта (дома)
4. FileId	INT, NULL	ID файла
5. ReportingPeriodId	INT, NULL	ID отчетного периода.
6. MethodName	NVARCHAR(200), NOT NULL	Наименование API-метода.
7. Created	DATETIME2(2), NOT NULL	Время формирования запроса
8. Updated	DATETIME2(2), NULL	Время последнего выполнения запроса.
9. Completed	DATETIME2(2), NULL	Время последнего успешного выполнения запроса.

В таблице 3.8 рассматривается структура SQL-таблицы ext.ReformaActionQueueLog.

Таблица 3.8 – Структура таблицы ext. ReformaActionQueueLog

Наименование поля	Тип поля	Описание поля
1. Id	PK, INT, NOT NULL	ID записи истории вызова запроса.
2. QueueActionId	FK, INT, NOT NULL	ID запроса.
3. Code	INT, NULL	Код ответного сообщения.
4. Created	DATETIME2(2), NOT NULL	Время создания записи о вызове запроса.
5. Message	NVARCHAR(400), NULL	Текст ответного сообщения.
6. SoapRequest	NVARCHAR(MAX), NULL	XML-сообщение, сформированное запросом.
7. SoapReply	NVARCHAR(MAX), NULL	Ответное XML-сообщение.
8. IsError	BIT, NOT NULL	Статус ответного сообщения.

Организация возможности интеграции файлов реализована с помощью служебной таблицы ext.ReformaFilesStorage. Она агрегирует необходимые файлы документов и актов из разных таблиц базы данных объектового учета и обеспечивает их синхронизацию с данными о файлах, хранящихся на сайте

Подп. и дата

Инв. № дубл.

Взам. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

портала Реформы.

В таблице 3.9 рассматривается структура SQL-таблицы ext.ReformaActionQueueLog.

Таблица 3.9 – Структура таблицы ext.ReformaActionQueueLog

Наименование поля	Тип поля	Описание поля
1. Id	PK, INT, NOT NULL	ID записи о файле.
2. FileTableId	INT, NOT NULL	ID файла в таблице.
3. ViewId	INT, NOT NULL	ID представления, где хранится файл.
4. ViewColumnId	INT, NULL	ID поля представления, по которому можно идентифицировать существование файла.
5. FileName	NVARCHAR(500), NULL	Наименование файла.
6. FileGroupCode	NVARCHAR(200), NULL	Наименование группы, в которую входит файл.
7. ReformaId	INT, NULL	ID файла в Реформе.
8. ProfilePartId	INT, NOT NULL	ID раздела анкеты, в который прикрепляется файл.
9. ObjectId	INT, NULL	ID объекта, к информации о котором прикреплен файл.
10. OrgId	INT, NOT NULL	ID организации, к информации о которой прикреплен файл.

3.2.2 Применение технологии SQL-представлений

База данных проекта «АИС: Объектовый учет» имеет сложную архитектуру, состоящую из множества таблиц. Для упрощения процесса доступа к данным применяется механизм построения SQL-представлений. Представление – это виртуальная таблица, динамически вычисляемая на основании данных реальных таблиц или тех же представлений. Данная методология нашла широкое применение при разработке архитектуры базы данных подсистемы интеграции. Преимущество представлений заключается в возможности расширения базовой таблицы новыми полями данных.

Таблицы, представленные в модели данных, были расширены с применением технологии представлений. Ниже рассматриваются структуры представлений, построенных на базе таблиц модели данных подсистемы интеграции. Базовые поля таблиц не приводятся.

Подп. и дата
Инв. № аудл.
Взам. инв. №
Подп. и дата
Инв. № подл.

					ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		

В таблице 3.10 приводится описание структуры представления ext.vw_ReformaActionQueue.

Таблица 3.10 – Структура представления ext.vw_ReformaActionQueue

Название поля	Источник	Тип	Описание
Inn	no.cmn\$Organization	NVARCHAR(200), NOT NULL	ИНН организации.
StructureId	no.cmn\$Object	INT, NULL	ID строения.
ObjectReformaId	no.cmn\$Object	INT, NULL	ID объекта в Реформе.
FileReformaID	ext.ReformaFilesStorage	INT, NULL	ID файла в Реформе.
MethodShortDescription	ext.ReformaAPIMethods	NVARCHAR(1500), NULL	Описание API-метода.

В таблице 3.11 приводится описание структуры представления ext.vw_ReformaFilesStorage.

Таблица 3.11 – Структура представления ext.vw_ReformaFilesStorage

Название поля	Источник	Тип	Описание
ViewName	no.vw_meta\$View	NVARCHAR(200), NOT NULL	Наименование представления, в котором хранится запись о файле.
ViewColumnName	no.vw_meta\$ViewColumn	NVARCHAR(200), NOT NULL	Наименование поля, которое идентифицирует файл.

В таблице 3.12 приводится описание структуры представления ext.vw_ReformaOrganizationRequests.

Таблица 3.12 – Структура представления ext.vw_ReformaOrganizationRequests

Название поля	Источник	Тип	Описание
OrgFullName	no.cmn\$Organization	NVARCHAR(200), NOT NULL	Полное наименование организации.
OrgShortName	no.cmn\$Organization	NVARCHAR(200), NULL	Название организации.
Message	ext.ReformaSoapFaults	NVARCHAR(500), NULL	Ответное сообщение.
IsError	ext.ReformaSoapFaults	BIT, NOT NULL	Статус сообщения.

В таблице 3.13 приводится описание структуры представления ext.vw_ReformaParameters.

Таблица 3.13 – Структура представления ext.vw_ReformaParameters

Название поля	Источник	Тип	Описание
ViewName	no.vw_meta\$View	NVARCHAR(200), NOT NULL	Наименование представления, в котором хранится

Подп. и дата

Инв. № докл.

Взам. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Название поля	Источник	Тип	Описание
			запись о файле.
ViewColumnName	no.vw_meta\$ViewColumn	NVARCHAR(200), NOT NULL	Наименование поля, которое идентифицирует файл.
IsStructure	no.vw_meta\$View	BIT, NULL	Обозначает: описывает ли представление строение.

Часто приходится использовать представления в чистом виде. Они не имеют базовой таблицы как основания, а строятся из набора разобщенных таблиц.

В таблице 3.14 описывается структура представления ext.vw_ReformaObjectToOrgLink, которая устанавливает соответствие домов и управляющих организаций.

Таблица 3.14 – Структура представления ext.vw_ReformaObjectToOrgLink

Название поля	Источник	Тип	Описание
ComposedFullName	no.cmn\$Object	NVARCHAR(200)	Полный адрес объекта.
ObjDateStart	no.cmn\$Object	DATETIME, NOT NULL	Дата строительства объекта.
ObjectId	no.cmn\$Object	INT, NOT NULL	ID объекта.
StructureId	no.cmn\$Object	INT, NOT NULL	ID строения.
OrganizationId	no.cmn\$ObjectToOrganizationLink	INT, NOT NULL	ID управляющей организации.
MngDateStart	no.cmn\$ObjectToOrganizationLink	DATETIME, NULL	Время начала управления домом.
MngDateEnd	no.cmn\$ObjectToOrganizationLink	DATETIME, NULL	Время окончания управления домом.
ShortName	no.cmn\$Organization	NVARCHAR(200), NULL	Название управляющей организации.
Name	no.cmn\$ObjectToOrganizationLink	NVARCHAR(200), NOT NULL	Наименование типа обслуживания.
LinkCode	no.cmn\$ObjectToOrganizationLink	NVARCHAR(50), NULL	Код типа обслуживания.
LinkId	no.cmn\$ObjectToOrganizationLink	INT, NOT NULL	ID типа обслуживания.
OrgTypeCode	no.cmn\$OrganizationType	NVARCHAR(50), NULL	Код типа управляющей организации.
Sort	Поле формируется в представлении	BIGINT, NULL	Приоритет управления (для случая одновременного управления домом несколькими УК).

Подп. и дата

Инв. № аудл.

Взам. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

3.2.3 Взаимодействие с базой данных «АИС: Объектовый учет»

Рассмотренная модель данных обеспечивает работу только подсистемы интеграции. Дополнительно приложение взаимодействует с источником данных проекта «АИС: Объектовый учет». На рисунке 3.2 показан порядок взаимодействия подсистемы интеграции и «АИС: Объектовый учет» на уровне базы данных.

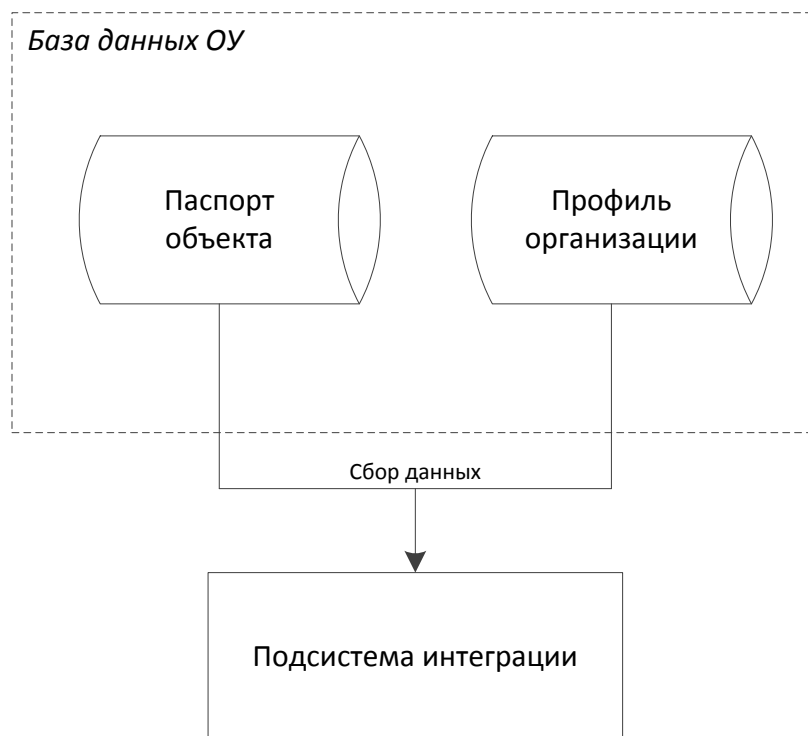


Рисунок 3.2 – Механизм взаимодействия подсистемы интеграции и БД ОУ

Подсистема интеграции взаимодействует с таблицами и представлениями, описывающими паспорт объекта и профиль организации. В таблице 3.15 приведен список представлений, формирующих паспорт объекта.

Таблица 3.15 – Список представлений паспорта объекта

Название представления	Описание
1. no.vw_od\$HouseHistoryCurrent	Текущие показания счетчиков на ресурсы ЖКХ (вода, газ, электричество).
2. no.vw_od\$StructureExtensionsCommon	Общие параметры строения объекта (тип проекта дома, год постройки и др.).
3. no.vw_od\$StructureExtensionsUninhabited	Параметры нежилой площади строения (общая площадь, площадь лестничных клеток и др.).
4. no.vw_cmn\$Structure	Общие сведения о строении (координаты строения, тип строения и др.).

Название представления	Описание
5. no.vw_cmn\$Object	Описание параметров объекта (адрес, тип дома и др.).
6. no.vw_od\$StructureComponentWall_alt	Описание параметров стен дома (площадь, материал и др.).
7. no.vw_od\$StructureComponentOverlap	Описание параметров домовых перекрытий (площадь, материал и др.).
8. no.vw_od\$LandExplicationAreas	Описание придомовой территории (площадь спортивных площадок, застроенная площадь и др.).
9. fin.vw_Account	Описание параметров начислений по дому (номер счета в банке, доходы, расходы на содержание дома и др.).
10. no.vw_od\$EnergyCharCommon	Описание классов энергопотребления.
11. no.vw_od\$WorkEstimate	Оценка выполненных УК работ (затраты на ремонт и др.).
12. no.vw_od\$LastBigRepairYear	Описание подробностей последнего капитального ремонта элемента строения (год, элемент строения и др.).
13. no.vw_od\$StructureConditions	Описание текущего состояния строения (год признания дома аварийным, текущее состояние и т.д.).
14. no.vw_od\$StructureComponentFacade	Описание параметров фасада строения (оштукатуренная площадь, материал и др.).
15. no.vw_od\$StructureComponentFacadeDetails	Описание типа фасада.
16. no.vw_od\$StructureComponentAperture	Описание параметров проемов (площадь дверных проемов, оконных и др.).
17. no.vw_od\$StructureComponentRoof	Описание параметров крыши (площадь, тип и др.).
19. no.vw_od\$RoofMaterialTypeLink	Соответствие типа крыши и материала.
20. no.vw_od\$StructureComponentBasement	Описание параметров подвала строения (площадь, объем и др.).
21. no.vw_od\$EngineerSystemChute	Описание параметров мусоропровода (длина, пропускная способность и др.).
22. no.vw_od\$EngineerSystemHeating	Описание параметров центральной системы отопления (длина труб, степень износа и др.).
23. no.vw_od\$EngineerSystemHotWater	Описание параметров системы горячего водоснабжения (тип труб, длина и др.).
24. no.vw_od\$EngineerSystemColdWater	Описание параметров системы холодного водоснабжения (тип труб, количество точек подачи воды и др.).
25. no.vw_od\$EngineerSystemSewage	Описание параметров системы канализации (длина труб и др.).
26. no.vw_od\$EngineerSystemPowerSupply	Описание параметров энергосети (напряжение, тип проводов и др.).
27. no.vw_od\$EngineerSystemGasSupply	Описание параметров системы газоснабжения (длина труб, тип отопления и др.).
28. no.vw_od\$EngineerSystemLift	Описание параметров лифтов (количество этажей, количество лифтов на подъезд и др.).
29. no.vw_od\$ManagementService	Описание предоставляемых УК услуг (тип услуги, название УК и др.).
30. no.vw_cmn\$Organization	Описание управляющей организации (название,

Подп. и дата

Инв. № аудл.

Взам. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Таблица 3.17 – Особенности паттерна «Очередь запросов» и их реализация

Особенность	Реализация
Паттерн позволяет хранить данные о запросах в отдельной таблице.	В таблице ext.ReformaActionQueue организовано хранение интеграционных запросов пользователей. Структура таблицы позволяет однозначным образом идентифицировать запрос.
Паттерн предусматривает отдельный модуль сбора данных для запросов.	В реализации подсистемы интеграции предусмотрен модуль сбора данных. Его работа основана на применении хранимых процедур, каждая из которых осуществляет выборку определенных параметров для соответствующего запроса.
Паттерн организует связь между таблицей с запросами и программной реализацией.	В подсистеме интеграции эта связь реализована с помощью ORM. Такой подход позволяет осуществлять вызов хранимых процедур для сбора информации из программной реализации, а затем использовать полученные данные в качестве аргументов методов API-интерфейса.

Сбор данных для интеграционных запросов организован с помощью хранимых процедур:

- а) ext.Reforma_ExtractData, определяет структуру выборки и какую хранимую процедуру для сбора данных необходимо вызвать;
- б) ext.Reforma_GetHouseInfo, осуществляет выборку параметров адреса объекта;
- в) ext.Reforma_SetHouseProfile, осуществляет выборку данных о доме из паспорта объекта;
- г) ext.Reforma_SetNewCompany, осуществляет выборку данных из профиля управляющей компании;
- д) ext.Reforma_SetCompanyProfile, осуществляет выборку отчетных данных управляющей компании.

Каждой хранимой процедуре соответствует один или несколько методов API-интерфейса, для которых процедурой осуществляется сбор данных. В таблице 3.18 приведено это сопоставление.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата							
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист	

Таблица 3.18 Сопоставление хранимых процедур и API-методов

Хранимая процедура	Методы API-интерфейса
ext.Reforma_GetHouseInfo	GetHouseInfo, SetNewHouse
ext.Reforma_SetHouseProfile	SetHouseProfile
ext.Reforma_SetNewCompany	SetNewCompany
ext.Reforma_SetCompanyProfile	SetCompanyProfile

3.3.1.2 Устройство хранимой процедуры сбора данных

Хранимая процедура сбора данных осуществляет выборку параметров для интеграции двумя способами: с использованием динамического SQL и с применение обычных запросов.

Представление ext.vw_ReformaParameters, справочник параметров для интеграции, имеет в своей структуре поля ViewName и ViewColumnName, которые обозначают из какого представления и какого поля может быть осуществлена выборка каждого параметра. При наличии этой информации возможно составление динамического SQL-запроса и его вызова.

Стандартные SQL-запросы применяются в случаях, когда составление динамического запроса невозможно из-за особенностей структуры базы данных.

3.3.2 Организация обработки информации

Процесс обработки информации заключается в преобразовании выборки данных, получаемых после выполнения хранимой процедуры, для сбора параметра запроса. Преобразование данных происходит в программном коде приложения, где коллекция записей конвертируется в объект API-класса. Этот объект в дальнейшем используется в качестве аргумента метода API-интерфейса для интеграции данных на портал «Реформа ЖКХ».

3.3.3 Организация передачи информации

Передача информации между «АИС: Объектовый учет» и порталом «Реформа ЖКХ» осуществляется посредством API-интерфейса, реализованного на базе протокола обмена данными SOAP. Принцип его действия основывается на обмене структурированными xml-сообщениями, в чем и заключается основное

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
					ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

преимущество протокола, так как при помощи xml-разметки возможно представление структуры данных любой сложности.

На момент разработки выпускной квалификационной работы API-интерфейс портала «Реформа ЖКХ» поддерживал 23 метода, подразделяющихся на три группы:

- а) GET-методы, осуществляющие загрузку данных с федерального портала;
- б) SET-методы, осуществляющие загрузку данных на портал «Реформа ЖКХ»;
- в) методы аутентификации, осуществляющие открытие и закрытие сессии подключения интеграции данных.

В таблице 3.19 приведено описание GET-методов API-интерфейса.

Таблица 3.19 – GET-методы API-интерфейса

Наименование метода	Описание
GetRequestList	Возвращает список запросов подписки на управляющую организацию, поданных внешней системой
GetReportingPeriodList	Возвращает список отчетных периодов системы.
GetHouseList	Метод получения списка домов, находящихся в управлении организации с соответствующим ИНН.
GetHouseInfo	Метод получения идентификатора дома с соответствующим адресом по ФИАС.
GetFilesInfoFromCompanyProfile	Метод получения списка файлов из заданного раздела текущей/архивной анкеты организации с соответствующим ИНН за указанный отчетный период.
GetFilesInfoFromHouseProfile	Метод получения списка файлов из заданного раздела текущей анкеты дома с соответствующим идентификатором дома.
GetFileByID	Метод получения файла с соответствующим идентификатором.
GetCompanyProfile	Метод получения данных анкеты (текущей и архивной) управляющей организации с соответствующим ИНН за указанный отчетный период.
GetHouseProfile	Метод получения данных анкеты (текущей) дома с соответствующим идентификатором дома.
GetCompanyProfileSF	Метод получения данных анкеты (текущей и архивной) управляющей организации по указанному субъекту федерации за указанный отчетный период.

Инв. № подл.	Взам. инв. №	Инв. № дубл.	Подп. и дата						Лист	
					ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

Наименование метода	Описание
GetHouseProfileSF	Метод получения данных анкеты (текущей) дома по указанному субъекту федерации.

В таблице 3.20 представлено описание SET-методов API-интерфейса.

Таблица 3.20 – SET-методы API-интерфейса

Наименование метода	Описание
SetRequestForSubmit	Метод подачи запроса на раскрытие данных. Внешняя система подает на вход список ИНН управляющих организаций, по которым собирается раскрывать данные.
SetCompanyProfile	Изменяет данные по текущей/архивной анкете управляющей организации с соответствующим ИНН за указанный отчетный период.
SetNewCompany	Метод подачи запроса на регистрацию новой управляющей организации.
SetUnlinkFromOrganization	Метод прекращения управления домом с соответствующим идентификатором.
SetHouseLinkToOrganization	Метод добавления дома, с соответствующим идентификатором, в управление организации с соответствующим ИНН.
SetHouseProfile	Изменяет данные текущей анкеты дома с соответствующим идентификатором дома.
SetNewHouse	Метод добавления нового дома с соответствующим адресом по ФИАС (GUID).
SetFileToCompanyProfile	Метод добавления файла в заданный раздел текущей/архивной анкеты организации с соответствующим ИНН за указанный отчетный период.
SetFileToHouseProfile	Метод добавления нового файла в заданный раздел текущей анкеты дома с соответствующим идентификатором дома.
SetFileDeleted	Метод удаления файла с соответствующим идентификатором.

В таблице 3.21 представлено описание SET-методов API-интерфейса.

Таблица 3.21 – Методы аутентификации API-интерфейса

Наименование метода	Описание
Login	Выполняет авторизацию внешней системы и открывает сеанс работы.
Logout	Завершает авторизованный сеанс работы внешней системы.

3.3.4 Организация выдачи информации

Выдача информации о проходящем интеграционном процессе производится на специальной html-странице в личном кабинете пользователя управляющей компании. Пользователю отображается подробная статистика, список

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

интеграционных запросов его компании и история статусов их выполнения.

3.3.5 Механизмы развертывания данных

Согласно требованиям технического задания на разработку должна обеспечиваться поддержка своевременного обновления справочных данных. Для решения этого вопроса применяется использование процедур развертки.

Функция развертки реализуется в виде хранимой процедуры, которая обеспечивает заполнение данными таблиц-справочников. Алгоритм процесса заполнения состоит из следующих этапов:

- а) создание временной таблицы, структура которой совпадает со структурой таблицы-справочника;
- б) сохранение данных во временную таблицу;
- в) произведение анализа сопоставления данных с помощью SQL-операций сравнения множеств, на основании которого некоторые записи из таблиц справочников подлежат удалению, обновлению или добавлению из временной таблицы;
- г) удаление временной таблицы.

В таблице 3.22 приводятся реализованные процедуры развертки и их назначение.

Таблица 3.22 – Процедуры развертки таблиц-справочников

Название	Таблица-справочник	Назначение
deploy.ReformaAPIMethods	ext.ReformaAPIMethods	Развертывание данных об API-методах.
deploy.ReformaParameters	ext.ReformaAPIParameters	Развертывание данных о параметрах интеграции.
deploy. ReformaDocumentSections	ext.ReformaDocumentSections	Развертывание данных о разделах анкет.
deploy.ReformaSoapFaults	ext.ReformaSoapFaults	Развертывание данных об ответных сообщениях сервиса.

Подп. и дата

Инв. № ауд.

Взам. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

4 Алгоритмическое обеспечение системы

В разделе рассматриваются алгоритмы, разработанные в процессе создания подсистемы интеграции для федерального портала «Реформа ЖКХ» и региональной системы объектового учета «АИС: Объектовый учет». Алгоритмы делятся на две группы: разработанные с помощью ЯВУ С# и языка доступа к данным Transact-SQL. Реализация модулей, разработанных с помощью ЯВУ С# представлена в приложении А, с помощью Transact-SQL – в приложении Б.

4.1 Алгоритм привязки данных

Назначением алгоритма привязки данных является формирование объекта API-класса, который используется методом API-интерфейса в качестве отправляемых данных Реформе.

Структура алгоритма состоит из двух частей. Первая из них представляет собой функцию, задача которой заключается в вызове подпрограммы установки значения свойству API-класса соответствующего значения элемента из входного списка параметров. Структура этого списка указана в таблице 4.1.

Таблица 4.1 – Структура данных списка

Наименование свойства	Назначение
1. ParameterName	Обозначает наименование параметра. В случае если параметр является вложенным по отношению к другому применяется обозначение вида: «параметр1.параметр2» .
2. Value	Обозначает значение параметра.
3. ArrayIndex	Обозначает порядок параметра, если его типом является коллекция объектов.

Вторая часть алгоритма представляет собой подпрограмму привязки значения свойству API-класса. Алгоритм является рекурсивным, потому что подпрограмма на основании указанного положения свойства определяет, куда необходимо установить значение. Рекурсивный подход позволяет применять алгоритм для класса, имеющего любую структуру.

На рисунке 4.1 представлена схема подпрограммы первой части алгоритма, обеспечивающей вызов подпрограммы установки значения из коллекции

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

соответствующему свойству API-класса.

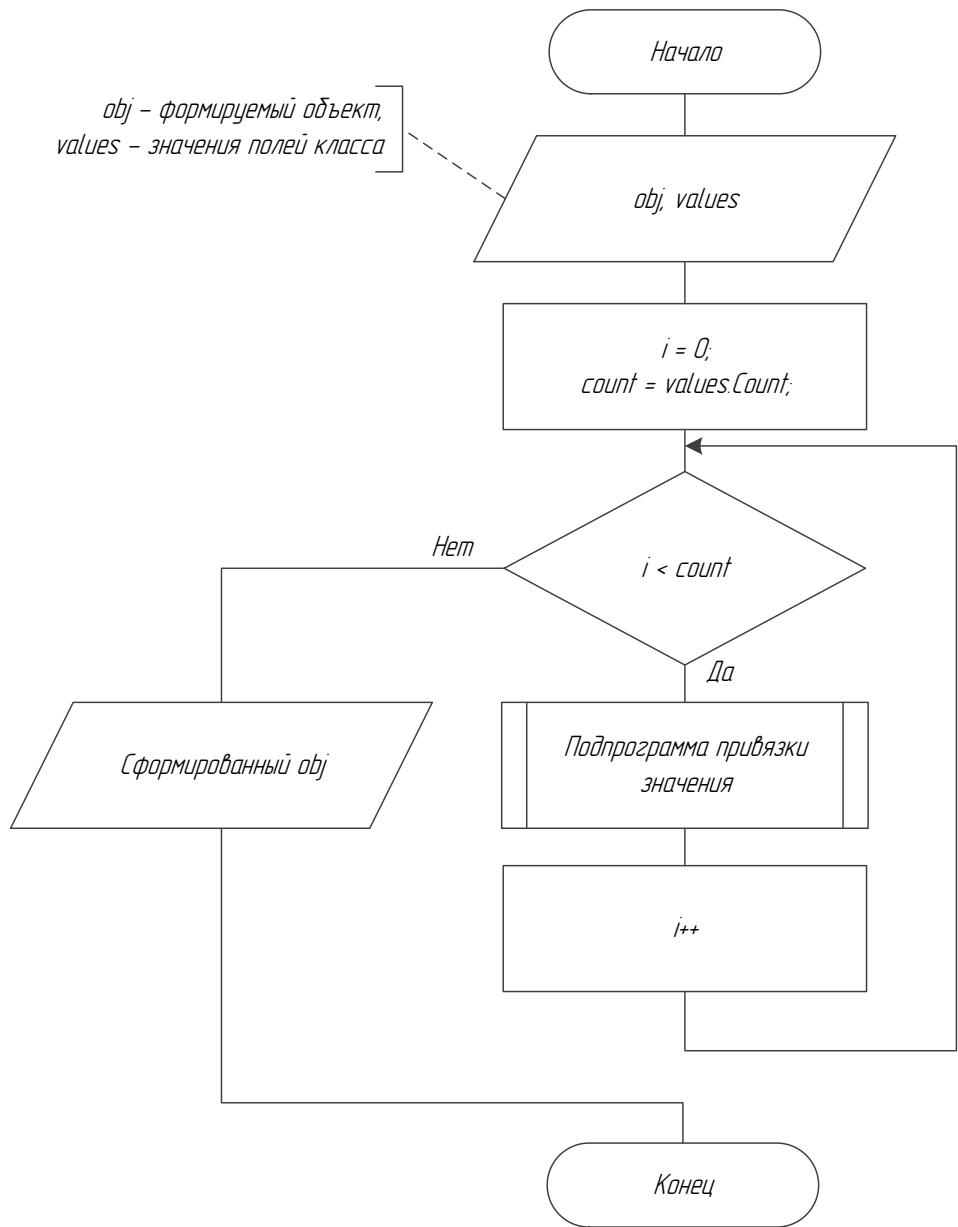


Рисунок 4.1 – Схема подпрограммы вызова алгоритма для привязки значения свойству API-класса

На вход функции подается два параметра: пустой объект требуемого API-класса и коллекция значений его свойств. В цикле, условием выхода из которого является перебор всех элементов коллекции, осуществляется вызов подпрограммы привязки данных. Выходным значением алгоритма является сформированный объект API-класса, готовый к отправке методом API-интерфейса в Реформу.

Подп. и дата	Инв. № аудл.	Взам. инв. №	Подп. и дата	Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Схема подпрограммы привязки значения показана на рисунке 4.2.

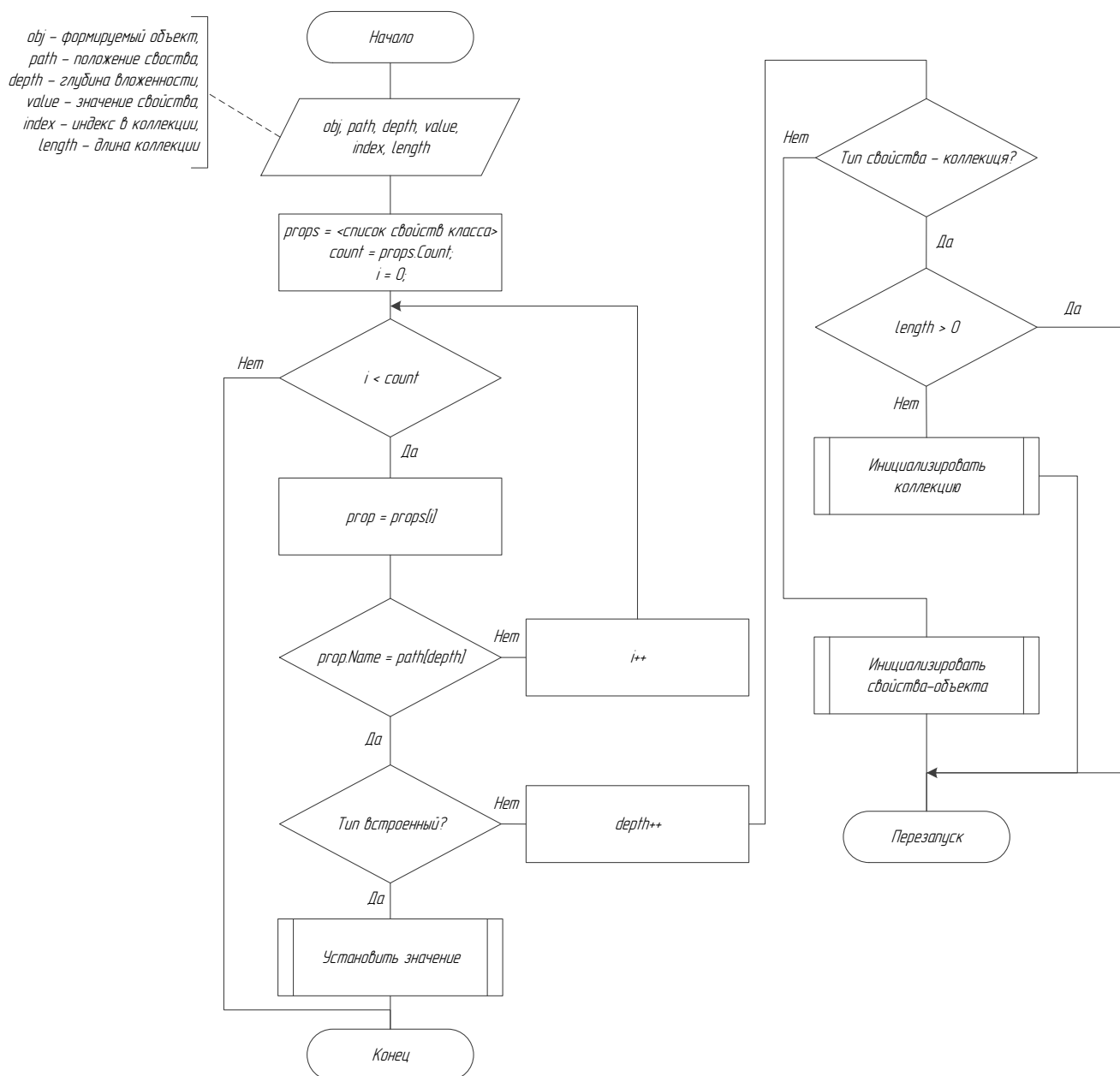


Рисунок 4.2 – Схема подпрограммы привязки значения

Особенность алгоритма – это его реализация с помощью механизма рефлексии ЯВУ С# [13]. Рефлексия или отражение – это процесс, во время которого программа может отслеживать и производить модификацию собственной структуры. В данном случае возможности механизма рефлексии были использованы для создания объекта класса на основе SQL-выборки данных.

В самом начале выполнения алгоритма запускается цикл по всем свойствам класса. При условии, что название параметра или первого из вложенных уровней

Подп. и дата

Инд. № аудл.

Взаим. инд. №

Подп. и дата

Инд. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Копировал

Формат А4

совпадает с наименованием поля класса, производится разветвление алгоритма. Если тип свойства – встроенный, один из базовых типов ЯВУ С#, то производится установка значения и выход из алгоритма. Если типом свойства является коллекция или классовый тип, то идентификатор уровня вложенности увеличивается на единицу. Затем происходит инициализация свойства-коллекции или свойства-объекта. Далее алгоритм продолжает работу рекурсивно до тех пор, пока значение не будет установлено на свое место в структуре свойств API-класса. При каждом рекурсивном вызове подпрограмма оперирует одним и тем же объектом. Это приводит к тому, что по окончании действия алгоритма на выходе получается экземпляр API-класса с установленным значением указанного свойства.

4.2 Алгоритм обеспечения процесса интеграции

Алгоритм обеспечения процесса обмена данными является ключевым для разработанной подсистемы интеграции. Просматривая записи, обозначающие запросы пользователей, алгоритм вызывает необходимые API-методы для отправки данных организаций Реформе.

Входные данные для алгоритма представляются коллекцией интеграционных запросов, поданных управляющими компаниями. На основании этой коллекции программа составляет список организаций, участвующих в информационном обмене [11]. Следующий шаг алгоритма обеспечивает загрузку или обновление справочников Реформы в базе данных «АИС: Объектовый учет». Затем происходит исполнение цикла по организациям, в теле которого выбираются запросы компании, выполняется проверка подачи заявки организации на раскрытие данных. Если заявка не была подана, то алгоритм обеспечивает процесс регистрации организации на федеральном портале. Затем производится выгрузку данных домоуправления в случае одобрения заявки администратором Реформы. Если организация уже зарегистрирована, выполняется обычная процедура поставки данных. Алгоритм завершает свою

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

работу после обработки актуальных на момент запросов всех организаций.

На рисунке 4.3 представлена схема подпрограммы обеспечения процесса интеграции.

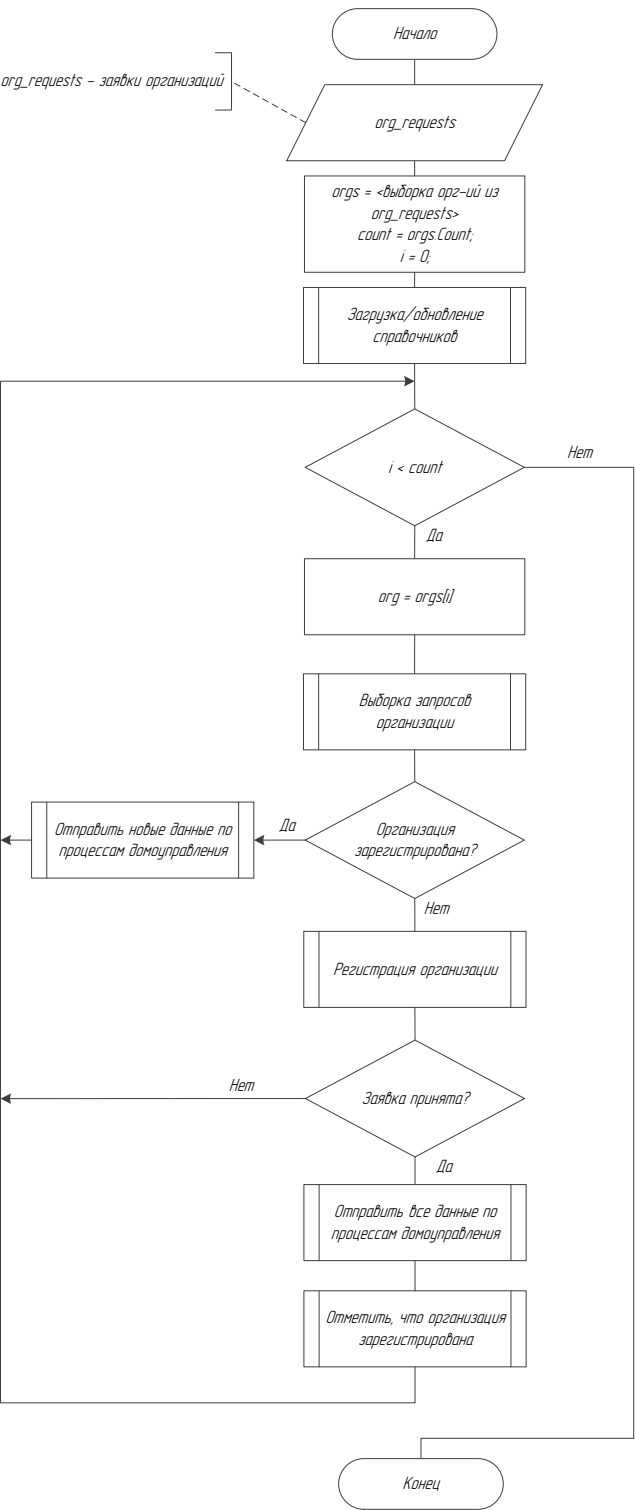


Рисунок 4.3 – Схема подпрограммы обеспечения процесса интеграции

Инд. № подл.	Подп. и дата
Взам. инд. №	Инд. № аудл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

схема подпрограммы сохранения истории вызова запроса на интеграцию данных.

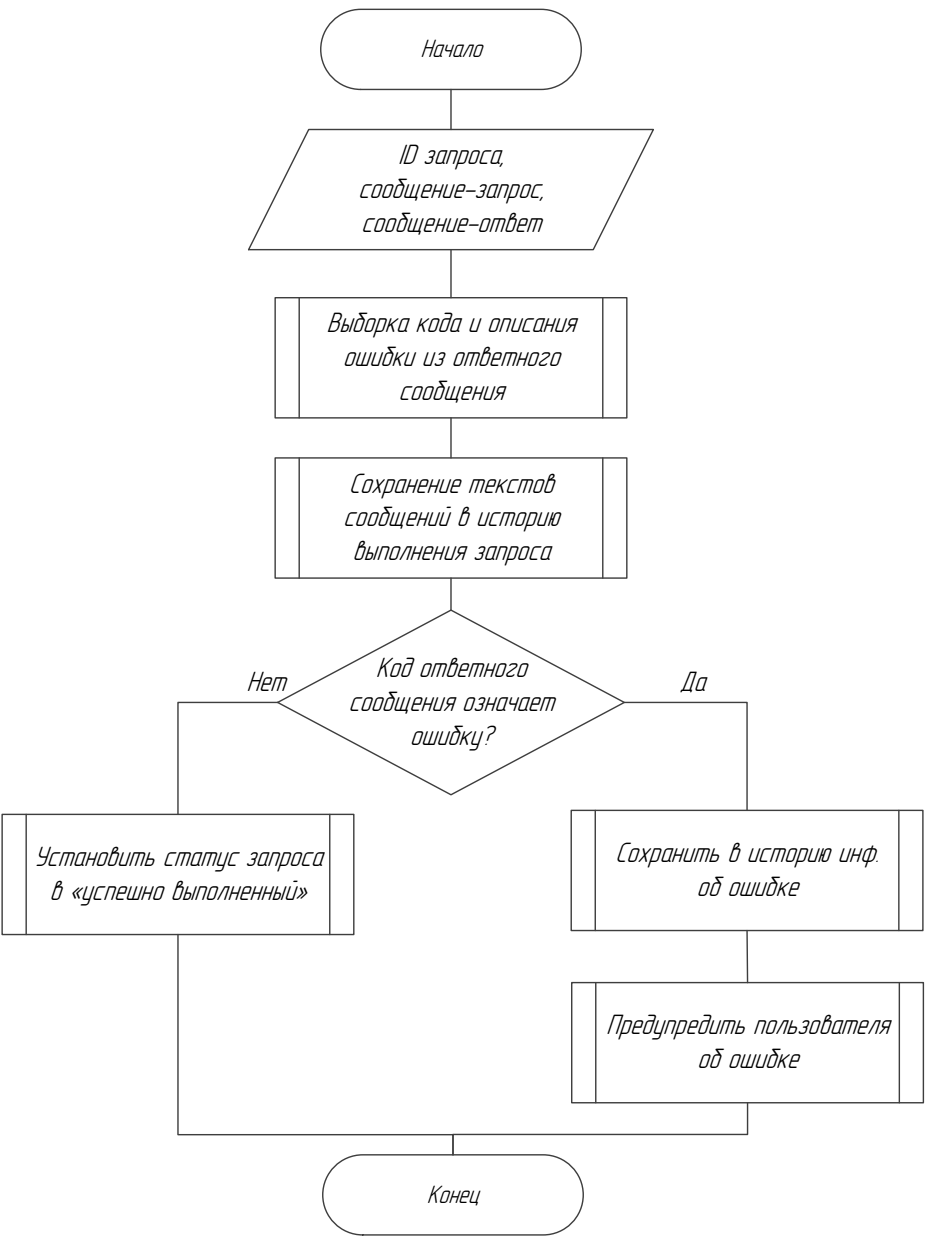


Рисунок 4.4 – Блок-схема алгоритма сохранения истории для интеграционного запроса

4.4 Алгоритм формирования запросов

Алгоритм формирования запросов отслеживает действия пользователя в системе «АИС: Объектовый учет» и создает соответствующие интеграционные запросы. Действия пользователя управляющей организации могут быть двух различных типов: изменение или добавление новых данных. Информация об этих

Инд. № подл.	Подп. и дата
Взам. инд. №	Инд. № подл.
Подп. и дата	Инд. № подл.
Инд. № подл.	Инд. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

событиях сохраняется в SQL-таблицу audit.Logs. Входными данными для алгоритма являются ID управляющей организации и дата, после которой необходимо учитывать внесенные пользователем изменения. Алгоритм реализован при помощи языка обращения к данным Transact-SQL.

Результатом выполнения алгоритма является создание новых записей о запросах на интеграцию в таблице ext.ReformaActionQueue.

4.4.1 «Быстрый» поиск первой записи в таблице о действиях
пользователя

Быстрота алгоритма формирования запросов зависит от того, сколько записей таблицы о действиях пользователя ему необходимо обработать. Алгоритм «быстрого» поиска ID первой записи, удовлетворяющей условию: «дата события больше или равна установленной дате», позволяет практически мгновенно находить результат при обработке большого количества записей.

В листинге 4.1 представлена реализация алгоритма при помощи языка доступа к данным Transact-SQL.

Листинг 4.1 – Реализация алгоритма «быстрого» поиска

```
DECLARE @aId BIGINT, @currentId BIGINT
DECLARE @currentDate DATETIME2
DECLARE @pace INT = -1000000

SELECT TOP 1 @aId = AuditID FROM audit.vw_Logs ORDER BY AuditID DESC
SET @currentId = @aId
WHILE 1 = 1 BEGIN
    SELECT TOP 1 @currentDate = al.UpdateDate FROM audit.vw_Logs AS al WHERE al.AuditID =
    @currentId

    IF @pace < 0 AND @currentDate < @startDate
        SET @pace = @pace / 2
    ELSE
        SET @aId = @currentId

    IF @pace = 0 BEGIN
        SELECT @aId = MIN(al.AuditID)
        FROM audit.vw_Logs AS al
    WHERE al.AuditID <= @aId AND al.AuditID > @currentId AND al.UpdateDate >= @startDate
        BREAK
    END
    SET @currentId = @aId + @pace
END
RETURN @aId
```

Подп. и дата
Инв. № аудл.
Взам. инв. №
Подп. и дата
Инв. № подл.

5 Программное обеспечение системы

В разделе приводится описание инструментальных средств разработки, применяемых в процессе создания подсистемы интеграции, основных модулей приложения, руководства пользования подсистемой и процесса ее установки и настройки.

5.1 Выбор компонентов программного обеспечения

В подразделе рассматриваются основные и вспомогательные инструментальные средства разработки, которые были использованы в процессе создания приложения интеграции.

5.1.1 Инструментальные средства разработки и язык программирования

Подсистема интеграции для региональной системы объектового учета «АИС: Объектовый учет» и федерального портала «Реформа ЖКХ» была разработана с применением технологий платформы .NET.

Серверная часть подсистемы, обеспечивающая сбор и обмен данными с Реформой, реализована в виде консольного приложения, хотя в требованиях упоминается о создании службы интеграции. Смена подхода объясняется тем, что в процессе разработки возникла необходимость полноценной отладки создаваемого программного обеспечения. Технология служб .NET не поддерживает эту особенность, поэтому выбор был сделан в пользу консольного приложения. Интерфейс панели управления подсистемой интеграции реализован с применением технологии ASP.NET MVC 4.

В ходе разработки подсистемы интеграции были использованы следующие языки программирования:

- а) С# 4.0., с помощью которого были созданы подсистема обмена данными и серверная часть панели управления интеграции для пользователя управляющей организации;
- б) язык гипертекстовой разметки HTML 5 для создания web-страниц;

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № ауд.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

- в) язык описания внешнего вида web-страниц CSS 3;
- г) встраиваемый язык программирования JavaScript для реализации интерактивных сценариев на web-страницах;
- д) язык доступа к данным Transact-SQL для организации процессов управления данными на стороне СУБД.

В качестве основной IDE использовалась Visual Studio 13 версии. Программный пакет используется в компании «АИС: Город» при разработке программных решений, так как фирма является сертифицированным партнером Microsoft.

5.1.2 Вспомогательное программное обеспечение

Для управления проектом базы данных использовалась программа SQL Server Management Studio фирмы Microsoft. Утилита представляет собой графическую среду, которая позволяет работать с объектами и настройками сервера базы данных.

Функциональность Management Studio может быть расширена специальными дополнениями. Они упрощают процесс создания архитектуры базы данных и написания SQL-скриптов на выборку записей из таблиц. При проектировании базы данных проекта использовались следующие расширения для Management Studio:

- а) SQL Prompt, производящий автозаполнение при написании кодов SQL-запросов;
- б) SQL Multi Script, позволяющий выполнять один SQL-скрипт на нескольких выбранных базах данных одновременно.

Разработка квалификационной работы проводилась на предприятии, где применяются программы SVN и GIT. Они предназначены для организации возможности ведения коллективной разработки.

Коллективная разработка создавала дополнительную сложность в проектировании архитектуры источника данных проекта. Она заключалась в том,

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					

что у каждого разработчика есть своя база, куда он вносит изменения. После этого программист должен сравнивать изменения, установленные другими разработчиками и разрешать возникающие конфликты. Этот процесс был автоматизирован с применением программного пакета SQL Compare производства компании Redgate Software. Программа позволяет сравнивать структуры нескольких баз данных и формировать искомую схему, в которой учитываются все новые изменения.

Проект подсистемы интеграции обеспечивает обмен данными между «АИС: Объектовый учет» и «Реформа ЖКХ» для нескольких регионов, каждый из которых представлен собственной базой данных. При этом записи таблиц-справочников должны быть идентичными. Для автоматизации проверки эквивалентности этих данных применялся программный продукт SQL Data Compare также производства компании Redgate Software.

Для тестирования процесса взаимодействия с сервисом Реформы использовалась открытая программа soapUI версии 4.5.2. С ее помощью можно отправлять xml-сообщения на указанный адрес сервиса и получать на них ответы.

5.2 Разработка прикладного программного обеспечения

В данном подразделе приводится описание структуры разработанной подсистемы интеграции в целом и ее каждого модуля в отдельности.

5.2.1 Структура прикладного программного обеспечения

Подсистема интеграции для федерального портала «Реформа ЖКХ» и «АИС: Объектовый учет» представляет собой приложение со сложной архитектурой. Архитектуру можно подразделить на две смысловые части:

- а) подсистема интеграции – это часть проекта, реализованная в виде консольного приложения, обеспечивающая обмен данными с Реформой;
- б) панель управления интеграцией – это часть проекта, реализованная в рамках web-приложения «АИС: Объектовый учет». Ее функция заключается

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

в демонстрации хода процесса интеграции данных пользователям управляющих компаний.

На рисунке 5.1 представлена архитектура подсистемы интеграции.

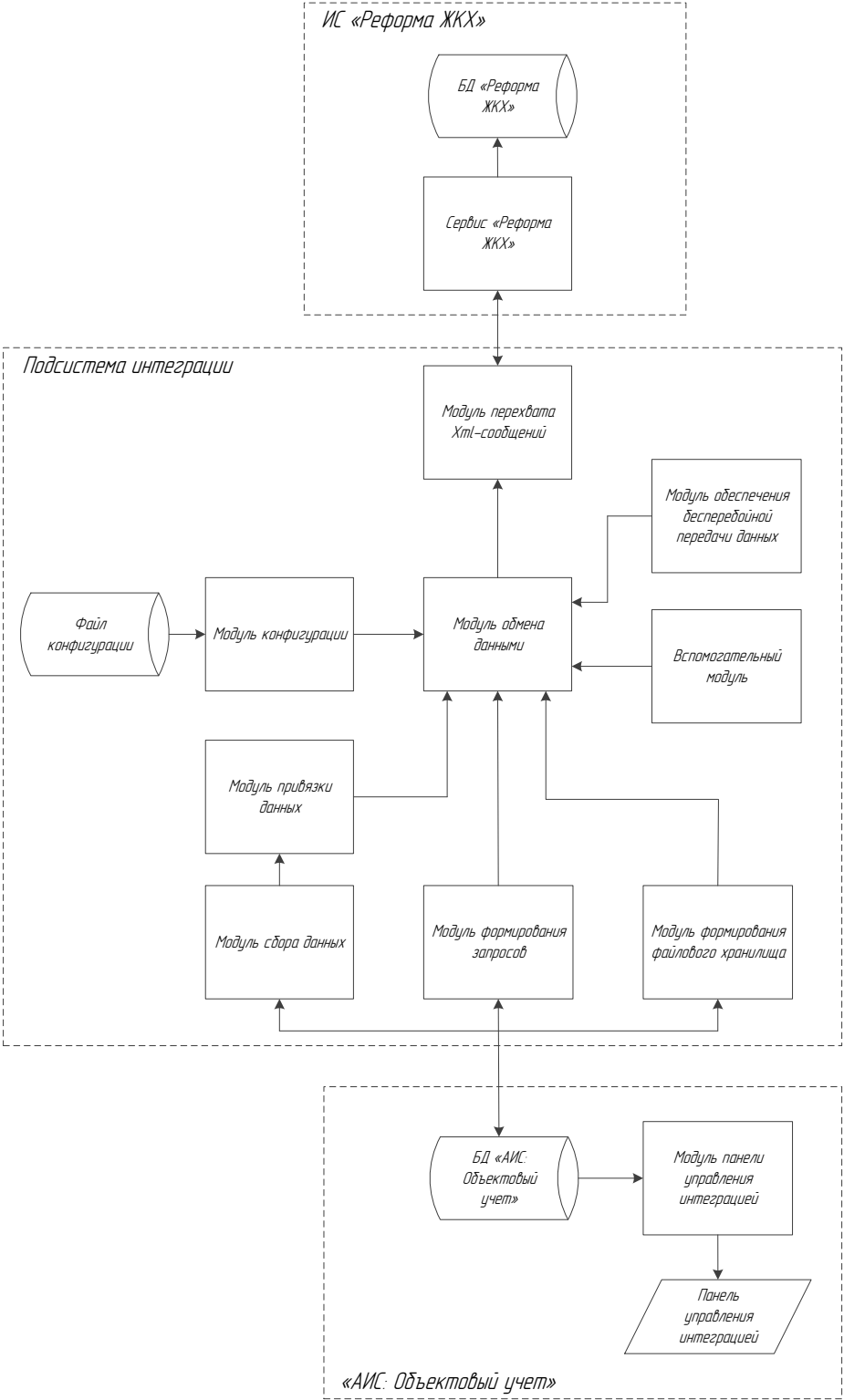


Рисунок 5.1 – Архитектура приложения подсистемы интеграции

Инд. № подл.	Подп. и дата
Взам. инд. №	Инд. № дубл.
Подп. и дата	Подп. и дата
Инд. № подл.	Инд. № дубл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Таблица 5.1 – Спецификация класса APIProvider

Название и тип элемента	Описание
Глобальные переменные	
1. private DB db	Объект для взаимодействия с данными базы данных как с объектами.
2. private DataBinder binder	Объект для обеспечения доступа к методам модуля привязки данных.
3. private ApiSoapPortClient client	Объект, при помощи которого осуществляется вызов методов API-интерфейса Реформы.
4. private LoginResponse loginResponse	Объект, содержащий информацию о ключе сессии подключения к сервису Реформы.
5. private ReloginHelper reloginner	Объект, обеспечивающий доступ к методам модуля бесперебойной передачи данных.
6. private bool isSoapFaultMessage	Переменная-флаг, обозначающая статус ответного сообщения.
7. private bool isLoginRepeatedly	Переменная-флаг, обозначающая момент выполнения API-метода Login() для получения нового ключа сессии подключения.
8. private int? activeQueueId	ID выполняемого интеграционного запроса
9. private string activeMethodName	Наименование выполняемого API-метода
10. private int daysRange = 7	Переменная устанавливает количество дней, в течение которых от текущей даты модуль формирования запросов идентифицирует изменения пользователей и создает соответствующие интеграционные запросы.
11. private AuthHeaderBehavior behavior	Определяет дополнительное «поведение» подключению для создания возможности перехвата сообщений.
Свойства	
1. public string ActiveMethodName	Свойство доступа к глобальной переменной класса activeMethodName.
2. public int? ActiveQueueId	Свойство доступа к глобальной переменной класса activeQueueId.
Конструкторы	
1. public APIProvider(DB _db)	_db – объект для взаимодействия с данными базы данных как с объектами.
2. public APIProvider(DB _db, string _endPointAddress)	_db – объект для взаимодействия с данными базы данных как с объектами; _endPointAddress – адрес сервиса Реформы.
Методы	
1. public void CommonInit(DB _db)	<u>Параметры:</u> _db – объект для взаимодействия с данными базы. <u>Назначение:</u> Используется для инициализации общих для всех конструкторов параметров.
2. public void HandleFaultReply(Message _reply)	<u>Параметры:</u> _reply – ответное xml-сообщение. <u>Назначение:</u> Callback-метод для обработки приходящего сообщения.
3. public void Login()	<u>Назначение:</u> Расширение стандартного API-метода аутентификации на сервисе Реформы.
4. public void Logout()	<u>Назначение:</u> Расширение стандартного API-метода закрытия канала обмена данными.
5. private void SetRequestForSubmit(IEnumerable<vw	<u>Параметры:</u> _queueItems – список запросов организации. <u>Назначение:</u> Расширение стандартного API-метода подачи

Подп. и дата

Инв. № аудл.

Взаим. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Название и тип элемента	Описание
<u>ReformaActionQueue</u> > <u>queueItems</u>)	заявки на раскрытие данных организацией
6. private void SetNewCompany(IEnumerable<vw_ReformaActionQueue> <u>queueItems</u>)	<u>Параметры:</u> <u>queueItems</u> – список запросов организации. <u>Назначение:</u> Расширение стандартного API-метода создания новой организации.
7. private void SetHouseLinkToOrganization(IEnumerable<vw_ReformaActionQueue> <u>queueItems</u>)	<u>Параметры:</u> <u>queueItems</u> – список запросов организации. <u>Назначение:</u> Расширение стандартного API-метода добавления дома в управление организацией.
8. private void SetUnlinkFromOrganization(IEnumerable<vw_ReformaActionQueue> <u>queueItems</u>)	<u>Параметры:</u> <u>queueItems</u> – список запросов организации. <u>Назначение:</u> Расширение стандартного API-метода удаления дома из управления организацией.
9. public void SetNewHouse(IEnumerable<vw_ReformaActionQueue> <u>queueItems</u>)	<u>Параметры:</u> <u>queueItems</u> – список запросов организации. <u>Назначение:</u> Расширение стандартного API-метода добавления нового дома.
10. private void SetHouseProfile(IEnumerable<vw_ReformaActionQueue> <u>queueItems</u>)	<u>Параметры:</u> <u>queueItems</u> – список запросов организации. <u>Назначение:</u> Расширение стандартного API-метода для поставки данных о доме.
11. private void SetFileToHouseProfile(IEnumerable<vw_ReformaActionQueue> <u>queueItems</u>)	<u>Параметры:</u> <u>queueItems</u> – список запросов организации. <u>Назначение:</u> Расширение стандартного API-метода для загрузки файла в анкету дома.
12. public void SetFileToCompanyProfile(IEnumerable<vw_ReformaActionQueue> <u>queueItems</u>)	<u>Параметры:</u> <u>queueItems</u> – список запросов организации. <u>Назначение:</u> Расширение стандартного API-метода для загрузки файла в анкету организации.
13. public void SetFileDeleted(IEnumerable<vw_ReformaActionQueue> <u>queueItems</u>)	<u>Параметры:</u> <u>queueItems</u> – список запросов организации. <u>Назначение:</u> Расширение стандартного API-метода удаления файла.
14. private void GetRequestList()	<u>Назначение:</u> Расширение стандартного API-метода для получения и сохранения списка статусов заявок организаций на раскрытие данных.
15. private void GetHouseList()	<u>Назначение:</u> Расширение стандартного API-метода для синхронизации объектов, находящихся в управлении организаций.
16. private void GetHouseInfo(bool? <u>updateAllObjects</u> = false, int? <u>orgId</u> = null)	<u>Параметры:</u> <u>updateAllObjects</u> – если параметр принимает значение true, тогда осуществляется синхронизация объектов еще не сопоставленных объектов. В противном случае синхронизация происходит для всех объектов. <u>orgId</u> – обозначает ID управляющей организации. При установке данного параметра будет осуществлена синхронизация только тех домов, которые находятся в управлении у данной организации. <u>Назначение:</u> Расширение стандартного API-метода для синхронизации всех объектов, хранящихся в базах данных обеих ИС.
17. private void GetReportingPeriodList()	<u>Назначение:</u> Расширение стандартного API-метода для загрузки/обновления данных справочника об отчетных периодах.
18. public void LaunchQueue()	<u>Назначение:</u> Метод обеспечивает процесс обмена данными между системами.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Инб. № подл.	Подп. и дата	Взам. инб. №	Инб. № дубл.	Подп. и дата

Таблица 5.2 – Спецификация класса DataBinder

Название и тип элемента	Описание
Глобальные переменные	
private char nestedParametersSplitter	Обозначает разделитель вложенных параметров.
Конструкторы	
public DataBinder(char splitter)	<u>Параметры:</u> splitter – символ-разделитель вложенных параметров.
Методы	
public void BindData(ref object _obj, List<Reforma_ExtractData_Result> _data)	Параметры: _obj – пустой объект API-класса. _data – коллекция данных, полученная в результате выборки данных с помощью вызова хранимой процедуры. <u>Назначение:</u> Создание объекта API-класса на основе коллекции данных.
private void SetValue(ref object _obj, string[] _path, int _depth, string _value, int? _index, int? _length)	<u>Параметры:</u> _obj – объект API-класса. _path – наименования уровней вложенности. _depth – номер текущего уровня вложенности. _value – значение свойства класса. _index – индекс элемента коллекции, в который нужно установить значение. _length – длина коллекции. <u>Назначение:</u> Установка значения свойству класса.

Модуль сбора данных производит выборку данных для интеграционного запроса. Работа модуля целиком построена на стороне базы данных, и его структура полностью состоит из иерархически взаимосвязанных хранимых процедур. На рисунке 5.2. представлен механизм очередности вызова процедур в рамках работы модуля.

					ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		

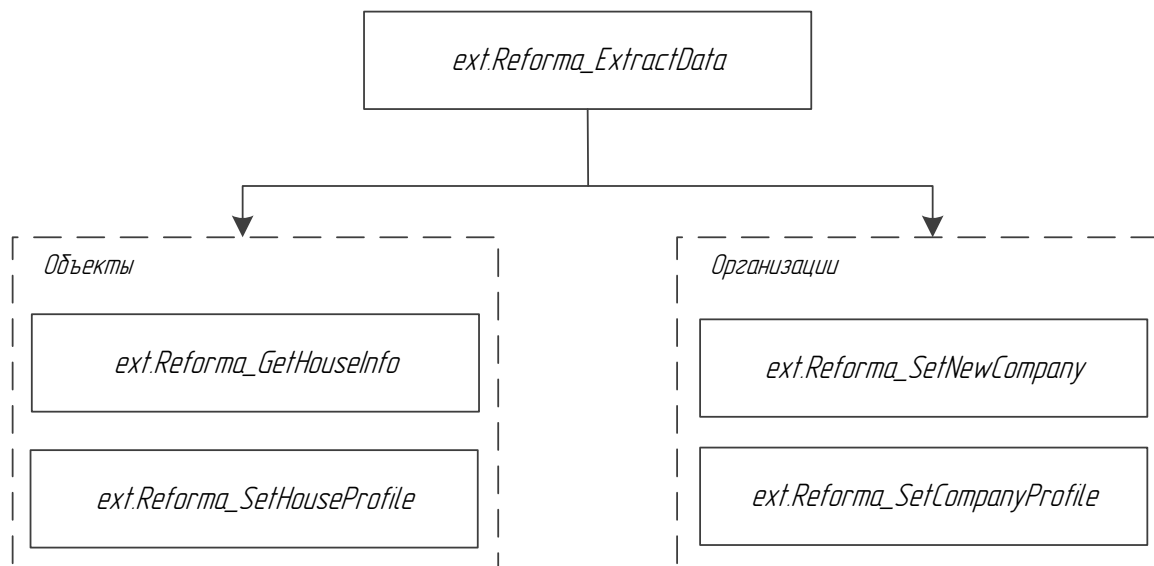


Рисунок 5.2 – Механизм работы модуля сбора данных

Согласно представленной схеме при сборе данных первоначально осуществляется вызов корневой хранимой процедуры `ext.Reforma_ExtractData`. Она определяет структуру возвращаемой выборки данных и создает сложный тип в ORM-системе. Непосредственный сбор информации происходит в остальных хранимых процедурах, выбор которых зависит от наименования API-метода, одного из аргументов корневой процедуры.

Количество значимых строк кода – 1741. В таблице 5.3 представлена спецификация модуля сбора данных.

Таблица 5.3 – Спецификация модуля сбора данных

Наименование хранимой процедуры	Описание
<code>ext.Reforma_ExtractData</code>	<p><u>Параметры:</u> @objId INT – ID объекта. @structureId INT – ID структуры объекта. @orgId INT – ID управляющей организации. @methodName NVARCHAR(MAX) – наименование API-метода.</p> <p><u>Назначение:</u> Корневая хранимая процедура в иерархии выборки данных для дальнейшей поставки в Реформу.</p>
<code>ext.Reforma_GetHouseInfo</code>	<p><u>Параметры:</u> @addressId INT – ID адреса объекта.</p> <p><u>Назначение:</u> Обеспечивает получение адресных данных объекта. Возвращаемые данные используются для API-метода <code>GetHouseInfo()</code>.</p>
<code>ext.Reforma_SetHouseProfile</code>	<p><u>Параметры:</u> @objId INT – ID объекта. @structureId INT – ID структуры объекта. @orgId INT – ID управляющей организации.</p>

данных. Его структура состоит из двух хранимых процедур: ext.Reforma_UpdateFilesRequestsForActionQueue и ext.Reforma_UpdateFilesStorage.

Количество значимых строк кода – 222. В таблице 5.5 представлена спецификация модуля формирования файлового хранилища.

Таблица 5.5 Спецификация модуля формирования файлового хранилища

Наименование хранимой процедуры	Описание
ext.Reforma_UpdateFilesRequestsForActionQueue	<u>Параметры:</u> @orgId INT – ID организации. <u>Назначение:</u> Формирование группы интеграционных запросов на сохранение и удаление файлов в базе данных федерального портала.
ext.Reforma_UpdateFilesStorage	<u>Параметры:</u> @orgId INT – ID организации. <u>Назначение:</u> Формирование записей о добавленных организацией файлах в таблице хранилища.

5.2.7 Модуль обеспечения бесперебойной передачи данных

Модуль обеспечения бесперебойной передачи данных был разработан с целью продления сессии подключения. Согласно регламенту взаимодействия с сервисом Реформы время пересылки данных в рамках одной сессии ограничено до 10 минут. Модуль производит переподключение к сервису каждые 8 минут для устранения возникновения исключительных ситуаций.

Структура модуля состоит из класса ReloginHelper. Количество значимых строк кода модуля – 8. В таблице 5.6 приведена спецификация класса ReloginHelper.

Таблица 5.6 – Спецификация класса ReloginHelper

Название и тип элемента	Описание
Глобальные переменные	
private APIProvider provider	Объект класса модуля обеспечения обмена данными (APIProvider).
private DateTime sessionStart	Время начала поставки данных.
Свойства	
public bool IsSessionExpired	Свойство, определяющее закончилось ли время подключения.
Конструкторы	
public ReloginHelper(APIProvider provider)	<u>Параметры:</u> provider – объект класса модуля обеспечения обмена данными (APIProvider).
Методы	
public void StartTimer()	<u>Назначение:</u> Производит запуск таймера,

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Название и тип элемента	Описание
	отмеряющего время подключения.
public void Relogin()	<u>Назначение:</u> Производит переподключение к сервису Реформу с помощью API-методов аутентификации.

5.2.8 Вспомогательный модуль

Вспомогательный модуль реализован в виде статического класса Servant. Класс содержит набор методов для обеспечения поддержки интеграционного процесса.

Количество значимых строк кода – 92. В таблице 5.7 представлена спецификация класса Servant.

Таблица 5.7 – Спецификация класса Servant

Название и тип элемента	Описание
Методы	
public static int GetRequiredObjectId(DB _db, HouseData _address)	<u>Параметры:</u> _db – объект для взаимодействия с данными базы данных как с объектами. _address – API-объект, обозначающий адрес дома. <u>Назначение:</u> Получение ID объекта в базе данных ОУ по адресу из Реформы.
public static int[] GetRequiredObjectTypes(DB _db)	<u>Параметры:</u> _db – объект для взаимодействия с данными базы данных как с объектами. <u>Назначение:</u> Получение идентификаторов типов объектов.
public static IEnumerable<vw_ReformaActionQueue> FilterQueueActionsByMethodName(DB _db, IEnumerable<vw_ReformaActionQueue> _actions, string _methodName)	<u>Параметры:</u> _db – объект для взаимодействия с данными базы данных как с объектами. _actions – коллекция запросов. _methodName – наименование API-метода. <u>Назначение:</u> Получение информации об определенных запросах из общей коллекции.
public static FileObject EncodeFileContent(DB _db, vw_ReformaFilesStorage _doc)	<u>Параметры:</u> _db – объект для взаимодействия с данными базы данных как с объектами. _doc – коллекция запросов. <u>Назначение:</u> Создание объекта для интеграции данных о файле.
public static bool IsAllActionsSucceed(IEnumerable<vw_ReformaActionQueue> _actions, string _methodName)	<u>Параметры:</u> _actions – коллекция запросов. _methodName – наименование API-метода. <u>Назначение:</u> Проверяет выполнены ли

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

Название и тип элемента	Описание
	успешно все методы определенного типа.
public static bool IsBidApproved(DB _db, int _orgId)	<u>Параметры:</u> _db – объект для взаимодействия с данными базы данных как с объектами. _orgId – ID организации. <u>Назначение:</u> Проверяет подтверждена ли заявка на раскрытие данных организацией.
public static IEnumerable<vw_cmnn_Organization> GetSynchronizedOrgs(DB _db)	<u>Параметры:</u> _db – объект для взаимодействия с данными базы данных как с объектами. <u>Назначение:</u> Получение списка организаций, осуществляющих интеграционный обмен данными.
public static void SaveLog(DB _db, string _soapReply, string _soapRequest, int _activeActionId, bool _isError, int? _code = null, string _message = "")	<u>Параметры:</u> _db – объект для взаимодействия с данными базы данных как с объектами. _soapReply – xml-сообщение ответа от Реформы. _soapRequest – xml-сообщение запроса. _activeActionId – ID выполненного запроса. _isError – обозначает статус выполнения запроса. _code – код сообщения. _message – текст ответного сообщения. <u>Назначение:</u> Сохранение истории выполнения запроса.

5.2.9 Модуль перехвата сообщений

Модуль перехвата сообщений был разработан с целью установки ключа сессии подключения в заголовки отправляемых сообщений и для их сохранения.

Структура модуля состоит из класса AuthHeaderBehavior, реализующего интерфейсы IEndpointBehavior и IClientMessageInspector. Количество значимых строк кода составляет 22. В таблице 5.8 представлена спецификация класса AuthHeaderBehavior.

Таблица 5.8 – Спецификация класса AuthHeaderBehavior

Название и тип элемента	Описание
Свойства	
public string SoapReply	Хранит ответное сообщение от Реформы.
public string SoapRequest	Хранит xml-сообщение запроса.
public TokenProvider TokenProvider	Объект класса используется для хранения ключа сессии.
public APIProvider ApiProvider	Объект класса модуля обеспечения обмена

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Название и тип элемента	Описание
	данными (APIProvider).
Конструкторы	
public AuthHeaderBehavior()	Инициализирует экземпляр класса.
public void AfterReceiveReply(ref System.ServiceModel.Channels.Message reply, object correlationState)	<u>Параметры:</u> reply – ответное xml-сообщение, приходящие от сервиса Реформы. correlationState – данные состояния корреляции. <u>Назначение:</u> Метод срабатывает в момент прихода ответного xml-сообщения и вызывает callback-функцию на стороне модуля обеспечения обмена данными для сохранения истории запроса.
public object BeforeSendRequest(ref System.ServiceModel.Channels.Message request, System.ServiceModel.IClientChannel channel)	<u>Параметры:</u> request – ответное xml-сообщение, приходящие от сервиса Реформы. channel – канал объекта клиента WCF. <u>Назначение:</u> Метод срабатывает в момент отправки xml-сообщения и устанавливает ключ сессии подключения в заголовок сообщения.

5.2.10 Модуль конфигурации

Назначение модуля конфигурации заключается в настройке подсистемы интеграции для работы с несколькими базами данных. Структура модуля состоит из классов CommonConfiguration и RegionConfigurationHandler. Статический класс CommonConfiguration описывает основные настройки конфигурации. Класс RegionConfigurationHandler обеспечивает доступ к настройкам, хранящимся в xml-файле конфигурации. Количество значимых строк кода модуля – 44.

В таблицах 5.9 и 5.10 представлены спецификации классов CommonConfiguration и RegionConfigurationHandler соответственно.

Таблица 5.9 – Спецификация класса CommonConfiguration

Название и тип элемента	Описание
Свойства	
public static string regionName	Название региона.
public static string connectionString	Строка подключения к базе данных.
public static string people_site_root_url	URL-адрес папки с файлами на сервере.
public static string people_site_path	Путь до папки с файлами на сервере.
Методы	
public static void DropSettings()	<u>Назначение:</u> Обеспечивает сброс настроек конфигурации.

Таблица 5.10 – Спецификация класса RegionConfigurationHandler

Название и тип элемента	Описание
Глобальные переменные	
private string configFilePath	Обозначает имя xml-файла конфигурации. Значение по умолчанию: @"RegionsConfig.xml"
private List<string> handledRegions	Список наименований регионов, с которыми

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

для этого настройка запуска программы осуществляется в меню диспетчера сервера, представленного на рисунке 5.3.

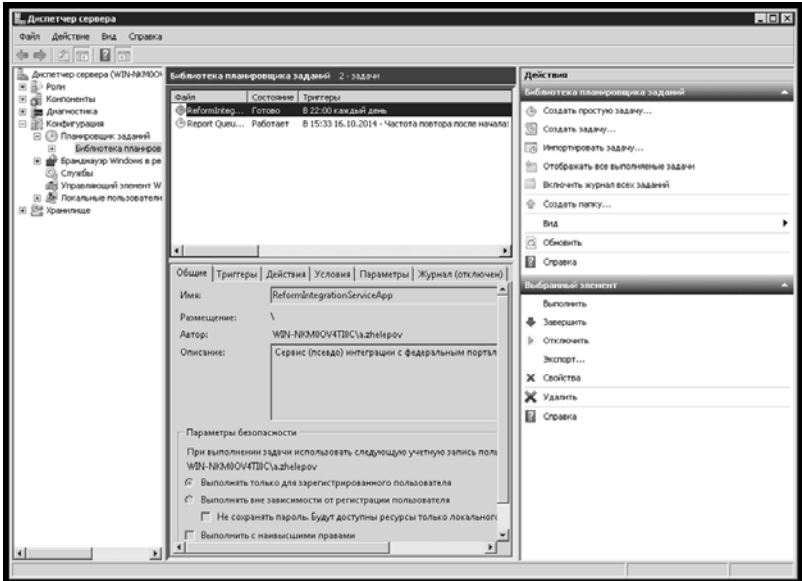


Рисунок 5.3 – Диспетчер сервера

Чтобы не влиять на скорость работы web-приложения «АИС: Объектовый учет», подсистема интеграции настроена на выполнение процесса обмена данными в ночное время, когда с системой объектового учета взаимодействует минимальное число пользователей.

Информационные системы для сферы ЖКХ быстро меняются под влиянием поступающих требований от заказчиков или правительства. Подсистема интеграции устроена таким образом, чтобы при переходе к новой версии максимально минимизировать затраты времени. Выпуск новой версии программы для обмена данными возможен при возникновении следующих ситуаций:

- а) Реформа изменила API-интерфейс. В этом случае доработка подсистемы интеграции сводится к адаптации модулей сбора данных и обмена данными для обеспечения поддержки новых возможностей API-интерфейса;
- б) Реформа добавила новые параметры интеграции. В этом случае доработка подсистемы интеграции сводится к адаптации модуля сбора данных и определению новых параметров в таблице-справочнике.

Подп. и дата	
Инв. № ауд.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

5.4 Интерфейс пользователя с системой

В данном подразделе рассматриваются возможности пользовательского интерфейса панели управления интеграцией, процесс установки для использования пользовательской части приложения интеграции и обработка возникающих в процессе работы web-приложения исключительных ситуаций.

5.4.1 Руководство пользователя

Интерфейс панели управления интеграцией состоит из четырех html-страниц:

- а) главная страница;
- б) страница со списком запросов организации;
- в) страница с историей вызова конкретного запроса;
- г) страница отображения статистики процесса интеграции.

Чтобы попасть на главную страницу пользователю управляющей компании необходимо авторизоваться в системе и зайти в личный кабинет. На рисунке 5.4 отображено положение меню доступа на главную страницу панели управления интеграцией.

Тарифы и услуги

Услуги
Здесь Вы можете указать услуги, оказываемые управляющей компанией.

Использование услуг
Здесь Вы можете указать использование оказываемых услуг, предоставляемых без договоров.

Тарифы на услуги
В этом разделе можно указать стоимость предоставляемых услуг.

Описание услуг

Описание услуг
Здесь вы можете описать результат выполнения работ федеральным законом, иным организацией), указание конс имущества многоквартирного

Нарушения

Информация о количестве случаев снижения платы*
Здесь Вы можете указать количество случаев снижения платы за нарушение качества содержания и ремонта общего имущества, коммунальных услуг, или за превышение продолжительности перерывов в оказании коммунальных услуг. Данные будут видны жителям.

Информация о привлечении к административной ответственности*
В случае привлечения управляющей организации к административной ответственности за нарушения в сфере управления многоквартирными домами, Вы можете указать количество случаев, копии документов о применении мер административного воздействия, а также меры, принятые для устранения нарушений, повлекших применение административных санкций. Данные будут видны жителям.

Федеральная система "Реформа ЖКХ"

Панель управления
Здесь Вы сможете отслеживать процесс интеграции Ваших данных с Реформой ЖКХ.

Рисунок 5.4 – Положение меню доступа на главную страницу интеграции

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

После перехода по ссылке пользователь попадает на главную страницу панели управления, представленную на рисунке 5.5.

Рисунок 5.5 – Главная страница панели управления интеграцией

На главной странице пользователь УК может прочитать правила интеграции данных и подать заявку на раскрытие данных своей организации.

После нажатия на кнопку о подаче заявки пользователь увидит диалоговое окно, представленное на рисунке 5.6.

Рисунок 5.6 – Диалоговое окно подачи заявки на раскрытие данных
Диалоговое окно создано для подтверждения процесса подачи

Подп. и дата	
Инв. № дубл.	
Взаим. инв. №	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

пользователем заявки, после которого статус главной страницы панели управления изменится, как указано на рисунке 5.7.

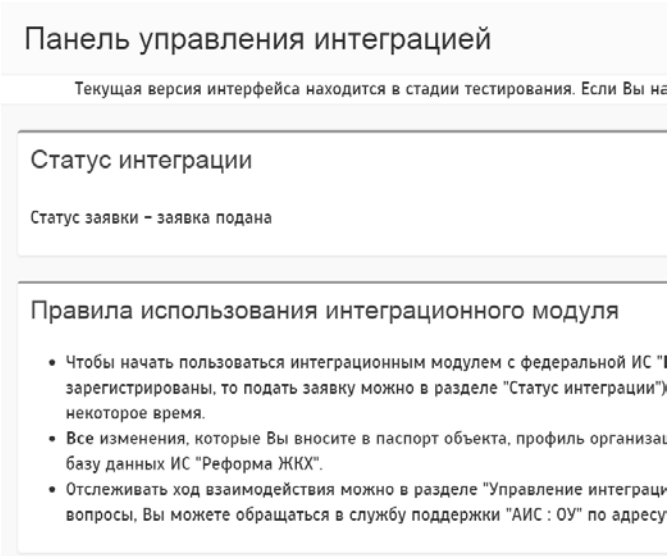


Рисунок 5.7 – Изменение статуса заявки после ее подачи

После рассмотрения и подтверждения заявки администратором портала, интерфейс главной страницы будет выглядеть, как представлено на рисунке 5.8.

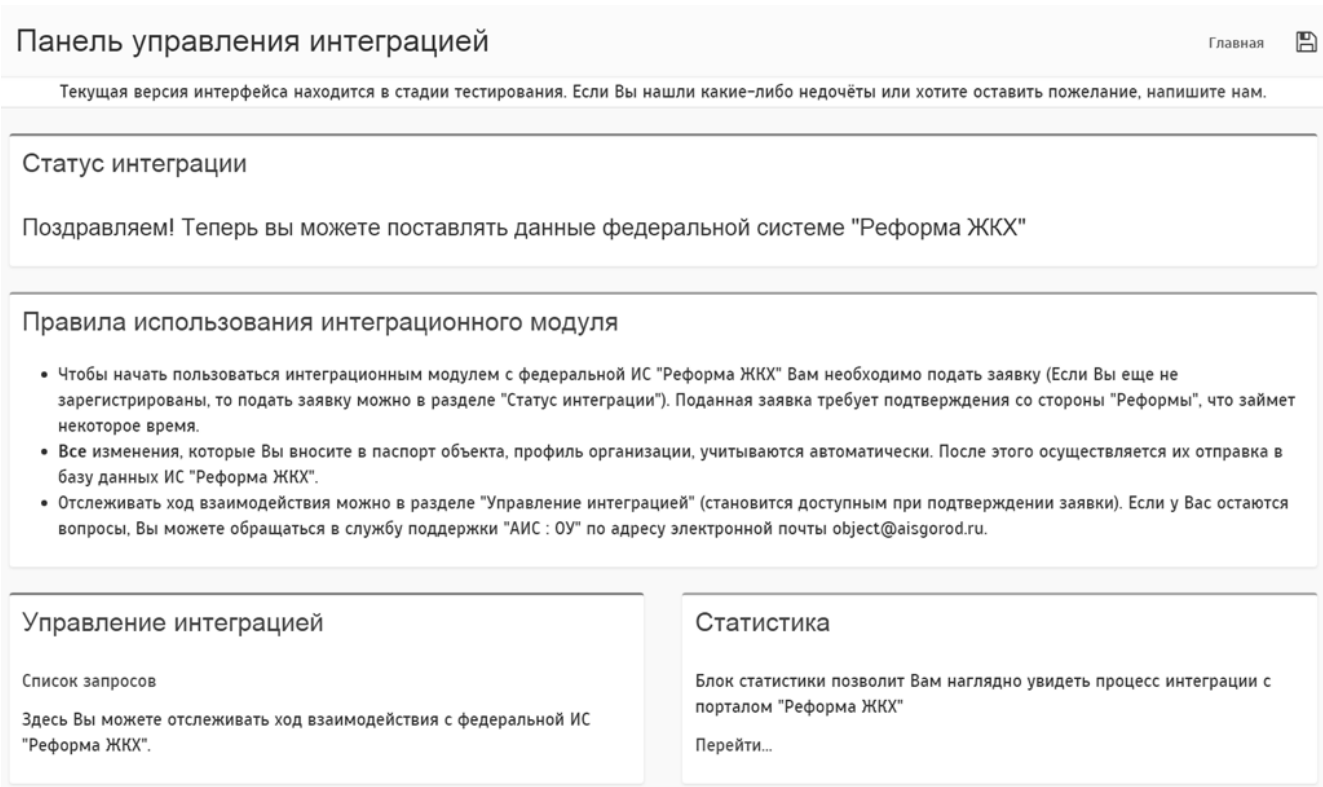


Рисунок 5.8 – Интерфейс панели управления после подтверждения заявки

В случае отклонения заявки на раскрытие данных, статус на главной

Подп. и дата
Инв. № дубл.
Взаим. инв. №
Подп. и дата
Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

странице также оповестит об этом пользователя. Вариант построения интерфейса в этом случае представлен на рисунке 5.9.

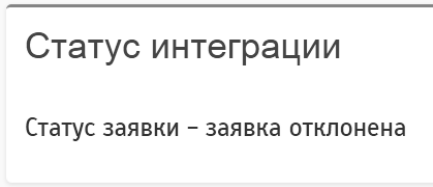


Рисунок 5.9 – Статус отклонения заявки

Подменю «Управление интеграцией» позволяет пользователю УК ознакомиться со списком интеграционных запросов, с помощью которых осуществляется раскрытие данных организации. Страница списка интеграционных запросов показана на рисунке 5.10.

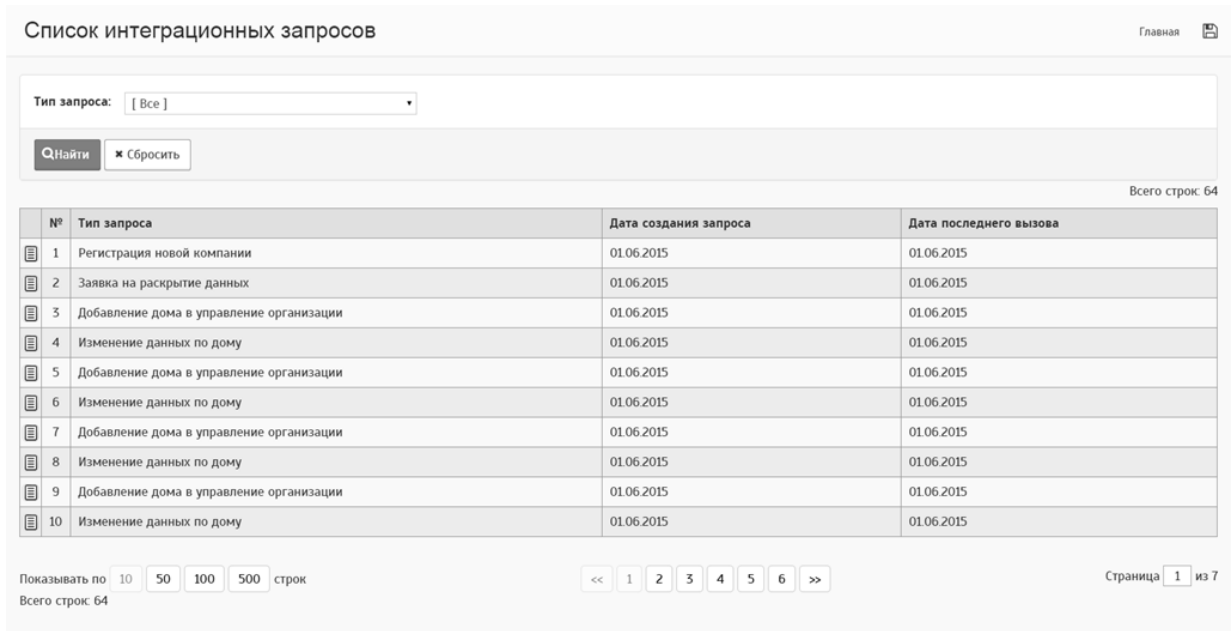


Рисунок 5.10 – Страница для отображения списка интеграционных запросов

Для удобства просмотра списка на странице есть фильтр, позволяющий отображать только определенные виды запросов. Для смены отображения групп и количества запросов пользователь может использовать встроенную пагинацию.

Для просмотра страницы истории вызова определенного запроса пользователь должен перейти по ссылке, которая представляет собой иконку «блокнота» и находится слева от названия интеграционного запроса в списке.

На рисунке 5.11 отображена страница истории вызова запроса.

Подп. и дата
Инв. № ауд.
Взам. инв. №
Подп. и дата
Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист
Формат А4



Рисунок 5.11 – Страница отображения истории вызова запроса

История вызова содержит записи о датах выполнения запроса и статусах их выполнения, а также информации об ошибках или предупреждениях, возникших при пересылке данных.

Страница статистики интеграции позволяет наглядно демонстрировать пользователям УК процесс интеграции их данных. На рисунке 5.12 отображена страница статистики интеграции.



Рисунок 5.12 – Страница отображения статистики интеграции

Подп. и дата

Инв. № аудл.

Взаим. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Всего на странице представлено несколько графиков, каждый из которых отражает определенную характеристику процесса интеграции. График синхронизации объектов показывает соотношение сопоставленных объектов для «АИС: Объектовый учет» и «Реформа ЖКХ», находящихся в управлении организации. На рисунке 5.13 отображен рассматриваемый график.

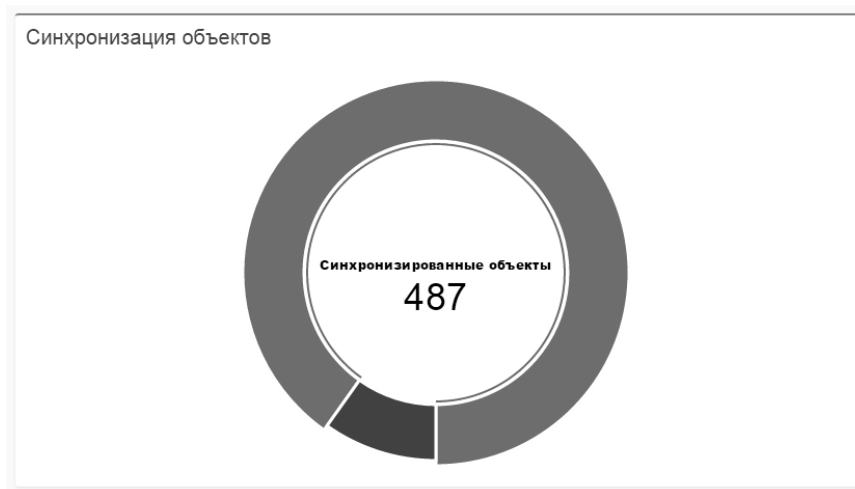


Рисунок 5.13 – График синхронизации объектов

Следующий за ним график отражает активность организации за последние семь дней. Он показывает количество выполненных успешно и невыполненных интеграционных запросов по дням. На рисунке 5.14 отображен рассматриваемый график.

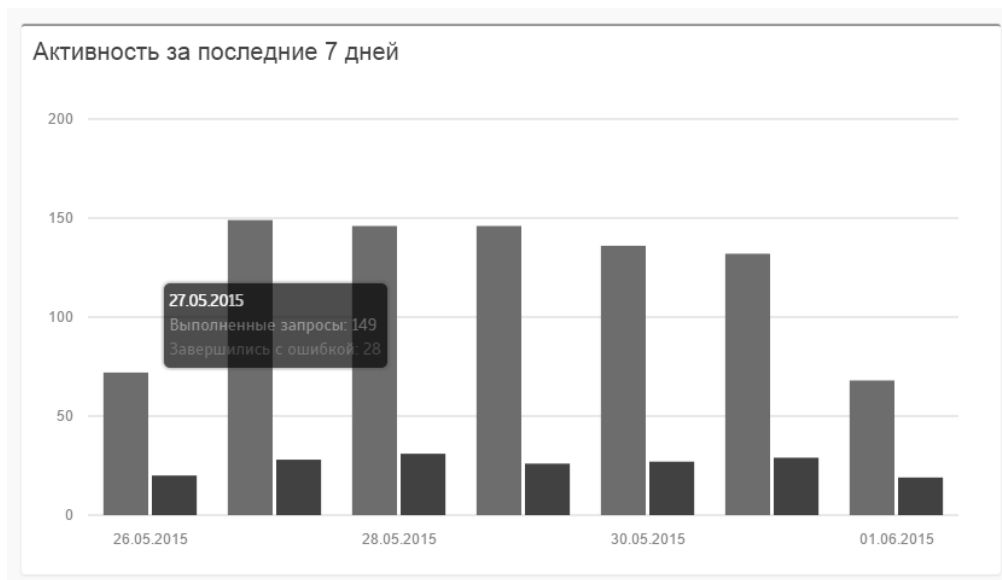


Рисунок 5.14 – График активности управляющей организации

Подп. и дата	
Инв. № аудл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

№№. № подл.	Подл. и дата	Взам. унб. №	Унб. № дудл.	Подл. и дата

6 Тестирование системы

6.1 Условия тестирования

Реформа ЖКХ поддерживает два сервиса обмена данными: тестовый и рабочий. Тестовый контур используется для тестирования процесса взаимодействия Реформы и внешней системы. Рабочий сервис используется для приема реальных данных управляющих компаний и дальнейшего их сохранения в базу данных Реформы.

Для начала обмена данными с федеральным порталом необходимо пройти два уровня проверки внешней системы интеграции: тестовый и испытательный. Первый уровень предполагает, что внешняя система должна быть подключена к тестовому сервису, с которым будет производиться обмен данными. Тестовый сервис предоставляет идентичный набор API-функций, что и рабочий. Это позволяет разработать подсистему с учетом всех особенностей интеграции.

Испытательный уровень наступает в момент, когда компания-разработчик внешней системы присылает заявку Реформе, в которой говорится о готовности программного решения для интеграции данных. Администратор портала открывает доступ на рабочий контур и в течение установленного срока проверяет поставляемые данные. Если проверка данных проходит успешно, то внешней системе окончательно открывается доступ на пересылку данных управляющих компаний.

6.2 Процесс тестирования

Процесс тестирования подсистемы интеграции данных условно разделяется на три основных этапа:

- а) тестирование в процессе разработки;
- б) альфа-тестирование;
- в) бета-тестирование.

Тестирование во время разработки подсистемы интеграции велось с

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № аудл.	Подп. и дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

применением методологии «белого ящика». Использование методологии позволило провести тщательную отладку решения интеграции.

Альфа-тестирование проводилось на этапе готовности подсистемы. В этом случае была выбрана методология «серого ящика». Использование методологии позволило лучше скоординировать действия сотрудника отдела качества компании «АИС: Город» и разработчика подсистемы.

Бета-тестирование подсистемы проводилось с непосредственным участием заказчиков подсистемы интеграции данных – пользователями управляющих компаний. В результате были исправлены недочеты и расширены возможности пользовательского интерфейса подсистемы интеграции.

6.3 Исходные данные для контрольных примеров

В процессе проверки корректной работоспособности подсистемы интеграции наибольшее внимание было уделено информационному обмену. Тестирование было проведено для всех методов API-интерфейса.

В листинге 6.1 представлен контрольный пример xml-сообщения, сформированного API-методом установки дома в управление организацией (SetHouseToOrganizationLink()).

Листинг 6.1 – Xml-сообщение запроса на установку дома в управление

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <authenticate>s5nj2gii3f2n9dmvj0u06kllc4</authenticate>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SetHouseLinkToOrganization xmlns="http://api-beta.reformagkh.ru/api_document_literal">
      <house_id xmlns="">6949302</house_id>
      <inn xmlns="">7325064546</inn>
      <date_start xmlns="">2007-12-01</date_start>
      <plan_date_stop xmlns="">2014-12-01</plan_date_stop>
    </SetHouseLinkToOrganization>
  </s:Body>
</s:Envelope>
```

В листинге 6.2 указывается ответ сервиса на запрос об установке дома в управление.

Листинг 6.2 – Ответное сообщение сервиса Реформы

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

Инв. № подл.	Взаим. инв. №	Инв. № дубл.	Подп. и дата						Лист	
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					

```

xmlns:ns1="http://api-beta.reformagkh.ru/api_document_literal"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <s:Header xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" />
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>House under the management</faultstring>
      <detail xsi:type="ns1:ErrorDetails">
        <code>1012</code>
        <name>House under the management</name>
        <description>Дом уже под управлением</description>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

В данном случае сервис вернул сообщение об ошибке выполнения метода. В сообщении говорится о том, что указанный в запросе дом уже находится под управлением.

В листинге 6.3 представлен контрольный пример xml-сообщения, сформированного API-методом для регистрации новой компании (SetNewCompany()).

Листинг 6.3 – Сообщение запроса о регистрации новой организации

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <authenticate>s5nj2gii3f2n9dmvj0u06kllc4</authenticate>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SetNewCompany xmlns="http://api-beta.reformagkh.ru/api_document_literal">
      <inn xmlns="">7325064546</inn>
      <newCompanyData xmlns="">
        <name_full>ОАО "Городская управляющая компания Ленинского района"</name_full>
        <name_short>ГУК Ленинского района</name_short>
        <okopf>1 10 00</okopf>
        <firstname>Андрей</firstname>
        <surname>Ворожецов</surname>
        <middlename>Владимирович</middlename>
        <position>Директор</position>
        <ogrn xsi:nil="true" />
        <date_assignment_ogrn>2010-06-07</date_assignment_ogrn>
        <name_authority_assigning_ogrn xsi:nil="true" />
        <legal_address>
          <city_id>bebfd75d-a0da-4bf9-8307-2e2c85eac463</city_id>
          <street_id>34d3d69a-6e9c-4b23-8652-4fdb60486103</street_id>
          <house_number>3</house_number>
          <building xsi:nil="true" />
          <block xsi:nil="true" />
          <room_number xsi:nil="true" />
        </legal_address>
        <actual_address>
          <city_id>bebfd75d-a0da-4bf9-8307-2e2c85eac463</city_id>
          <street_id>34d3d69a-6e9c-4b23-8652-4fdb60486103</street_id>
          <house_number>3</house_number>
          <building xsi:nil="true" />
          <block xsi:nil="true" />
          <room_number xsi:nil="true" />
        </actual_address>
      </SetNewCompany>
    </s:Body>
  </s:Envelope>

```

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. № дубл.	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					

```

</actual_address>
<post_address>
  <city_id>bebfd75d-a0da-4bf9-8307-2e2c85eac463</city_id>
  <street_id>34d3d69a-6e9c-4b23-8652-4fdb60486103</street_id>
  <house_number>3</house_number>
  <building xsi:nil="true" />
  <block xsi:nil="true" />
  <room_number xsi:nil="true" />
</post_address>
<phone>8(8422)272720</phone>
<site xsi:nil="true" />
<proportion_sf xsi:nil="true" />
<proportion_mo xsi:nil="true" />
</newCompanyData>
</SetNewCompany>
</s:Body>
</s:Envelope>

```

В листинге 6.4 указывается ответ сервиса на запрос о регистрации новой компании.

Листинг 6.4 – Ответное сообщение сервиса Реформы

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns1="http://api-beta.reformagkh.ru/api_document_literal"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <s:Header xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" />
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>The company with INN already exist</faultstring>
      <detail xsi:type="ns1:ErrorDetails">
        <code>1032</code>
        <name>The company with INN already exist</name>
        <description>Управляющая организация с указанным ИНН уже зарегистрирована в
системе</description>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Ответ сервиса Реформы содержит информацию о том, что регистрация компании необязательна, так как необходимые данные о ней уже занесены в базу данных.

В листинге 6.5 представлен контрольный пример xml-сообщения, сформированного запросом о подаче управляющей компанией заявки на раскрытие данных.

Листинг 6.5 – Сообщение запроса о подаче заявки

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <authenticate>s5nj2gii3f2n9dmvj0u06kllc4</authenticate>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SetRequestForSubmit xmlns="http://api-beta.reformagkh.ru/api_document_literal">
      <inns xmlns="">

```

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. № дубл.	Подп. и дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
					Изм.	Лист	№ докум.	Подп.	Дата	

```

        <item>7325064546</item>
      </inns>
    </SetRequestForSubmit>
  </s:Body>
</s:Envelope>

```

В листинге 6.6 указывается ответ сервиса на запрос о подачи заявки организацией.

Листинг 6.6 – Ответное сообщение сервиса Реформы

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api-beta.reformagkh.ru/api_document_literal">
  <s:Header xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" />
  <SOAP-ENV:Body>
    <ns1:SetRequestForSubmitResponse>
      <SetRequestForSubmitResult>
        <item>
          <inn>7325064546</inn>
          <status>3</status>
        </item>
      </SetRequestForSubmitResult>
    </ns1:SetRequestForSubmitResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Ответное сообщение содержит информацию о том, что организация с индивидуальным налоговым номером 7325064546 уже подавала ранее запрос на раскрытие данных. Это подтверждается значением статуса сообщения.

6.4 Результаты тестирования

Выбранная последовательность применения различных подходов к тестированию программного обеспечения показала свою эффективность.

На этапе разработки подсистемы интеграции постоянное тестирование модулей приложения с помощью методологии «белого ящика» позволило построить надежную архитектуру.

Альфа-тестирование подсистемы позволило выявить недоработки модуля сбора данных из паспорта объекта и профиля организации.

На этапе бета-тестирования подсистемы интеграции принимали участие непосредственные заказчики – пользователи управляющих организаций. По результатам этой стадии тестирования были внесены правки в пользовательский интерфейс приложения интеграции и окончательно проверена правильность работы модуля обмена данными.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					

7 Экономический раздел

7.1 Расчет показателя трудоемкости для разработанного программного продукта

Трудоемкость работ — это показатель, характеризующий затраты живого труда, выраженные в рабочем времени, затраченном на производство продукции или услуг. Величина данного параметра напрямую зависит от продолжительности периодов времени, занимаемых каждым из этапов проектирования программного продукта.

Чтобы выполнить разработку интеграционного слоя информационных систем необходимо начать с анализа предметной области, в которой будет использоваться создаваемый программный продукт.

После детального изучения сферы применения наступает время процесса прогнозирования временных затрат для каждого из этапов проектирования. Подходить к этим расчетам нужно ответственно, чтобы свести к минимуму погрешности в оценке трудоемкости работ по проекту.

В настоящее время для определения трудоемкости разработки информационных приложений применяется способ оценки работ в человеко-часах. Эффективность методики подтверждена ведущими современными IT-компаниями.

Величина параметра трудоемкости для разрабатываемого программного решения состоит из суммы значений трудоемкости для каждого этапа разработки и рассчитывается по формуле 7.1:

$$T_{об} = \sum_{i=1}^n t_i, \quad (7.1)$$

где $T_{об}$ – общая трудоемкость разработки программного продукта,

t_i – трудоемкость разработки i -го этапа проектирования,

n – общее количество этапов проектирования.

Проанализировав формулу n , можно утверждать, что если проект разделен

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

на большее количество стадий разработки, то искомая оценка трудоемкости выполняемых работ будет точнее. В таблице 7.1 приведены данные о расчете величины параметра трудоемкости для каждого из этапов проектирования и для всего проекта в целом.

Таблица 7.1 – Поэтапная и общая оценка трудоемкости программного решения

Этап разработки	Вид работ	Длительность работ (чел. × час.)
Формирование требований к системе	Исследование предметной области объекта проектирования. Анализ требований пользователей	40
Разработка технического задания	Написание документов технического задания на систему. Утверждение технического задания	50
Изучение принципов и методологий информационных обменов	Выбор методики обмена сообщениями и изучение принципов ее работы	40
Реализация программного решения	Разработка информационного продукта на языке программирования	250
Альфа-тестирование системы	Проведение тестирования разработанного программного решения на тестовых данных. Устранение ошибок	60
Бета-тестирование	Проведение мер по тестированию с использованием реальных данных пользователей.	40
Написание рабочей документации	Разработка сопроводительной документации на систему	50
Процесс внедрения	Поставка готового решения пользователям	10
Итого:		540

Общая продолжительность рабочего времени, затраченного на разработку программного продукта составила 540 ч., что примерно составляет 68 8-ми часовых рабочих дней.

В таблице 7.2 представлен график проведения работ по проекту.

Таблица 7.2 – График проведения работ по проекту

Вид работ	Исполнитель	Трудоемкость, чел.-час	Кол -во дней	Продолжительность работы									
1. Формирование	Инженер -	40	5										

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

Вид работ	Исполнитель	Трудоемкость, чел.-час	Кол -во дней	Продолжительность работы							
требований к системе	программист										
2. Разработка технического задания	Инженер - программист	50	7								
3. Изучение принципов и методологий информационных обменов	Инженер - программист	40	5								
4. Реализация программного решения	Инженер - программист	250	32								
5. Альфа-тестирование системы	Инженер - программист и сотрудник отдела QA	60	8								
6. Бета-тестирование	Инженер - программист, сотрудник отдела QA, заказчики системы	40	5								
7. Написание рабочей документации	Инженер - программист	50	7								
8. Процесс внедрения	Инженер-программист	10	2								
Общая трудоемкость и длительность проведения работ по проекту		540	71								

7.2 Расчет затрат на материальные ресурсы и сырье

Материальные ресурсы – это различные виды сырья, материалов, топлива, энергии, комплектующих и полуфабрикатов, которые хозяйствующий субъект закупает для использования в хозяйственной деятельности с целью выпуска продукции, оказания услуг и выполнения работ.

Процесс дипломного проектирования требовал определенный ресурс в виде материальных и сырьевых затрат. Расчет стоимости необходимых материалов производился с помощью формулы 7.2:

$$Z_m = \sum_{i=1}^n P_i \times C_i, \quad (7.2)$$

где P_i – расход i -го вида материального ресурса, натуральные единицы,

C_i – цена за единицу i -го вида материального ресурса,

i – вид материального ресурса,

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

/лист

Изм. /лист № докум. Подп. Дата

n – общее количество всех видов материальных ресурсов.

Результаты расчетов затрат на материальные ресурсы приведены в таблице 7.3.

Таблица 7.3 – Сумма затрат на материальные ресурсы

Наименование	Единица измерения	Количество израсходованного материала	Цена за единицу, руб.	Сумма, руб
1. Ноутбук	шт.	1	26 700	26 700
2. Принтер	шт.	1	4 750	4 750
3. Канцелярские принадлежности	шт.	1	500	500
Полная сумма затрат на материальные ресурсы				31 950

Общая стоимость расходных материалов рассчитывается также по формуле 7.2. Необходимые расчеты отображены в таблице 7.4.

Таблица 7.4 – Расчет стоимости затрат на расходные материалы

Наименование	Единица измерения	Количество израсходованного материала	Цена за единицу, руб.	Сумма, руб
1. Оплата услуг интернет-провайдера	руб./мес.	4	350	1400
2. Бумага для принтера	упак.	1	200	200
Полная сумма затрат на расходные материалы				1 600

Расчет стоимости затраченной электроэнергии в процессе написания дипломного проекта производится на основе действующих тарифов на электроэнергию, устанавливаемых региональными энергетическими комиссиями. Общая сумма энергетических затрат рассчитывается по формуле 7.3:

$$Z_э = \sum_{i=1}^n P_i \times T_i \times Ц, \quad (7.3)$$

где $Z_э$ – сумма затрат на электроэнергию,

P_i – паспортная мощность электрооборудования i -го вида, измеряется в кВт,

$Ц$ – тариф электроэнергии, руб./кВт × ч.

i – вид прибора электрооборудования,

n – общее число электроприборов.

Необходимые расчеты затрат на электроэнергию приведены в таблице 7.5.

Инв. № подл.	Взам. инв. №	Инв. № дубл.	Подп. и дата						Лист	
Инв. № подл.	Взам. инв. №	Инв. № дубл.	Подп. и дата	Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ	Лист

Таблица 7.5 – Расчет затрат на электроэнергию

Наименование единицы оборудования	Паспортная мощность, кВт.	Время работы оборудования, ч	Тариф электроэнергии, руб/кВт × ч	Сумма, руб
Ноутбук	0,1	540	2,9	156,6
Принтер	0,1	1		0,29
Освещение рабочего места	0,03	200		17,4
Итого за электроэнергию				174.29

7.3 Расчет затрат на оплату труда

Заработная плата начисляется, исходя из установленных на предприятии тарифов, сдельных расценок, окладов и сведений о фактически отработанном работниками времени или сведений об объемах выпущенной продукции. Расчет зарплаты производится на основании таких документов, как штатное расписание, положение об оплате труда, приказы о приеме на работу, трудовые договора и дополнения к ним.

Величину отчислений на социальные нужды определяют исходя из включенной в смету суммы заработной платы и установленных норм отчислений на эти цели.

Расчет суммы затрат на оплату труда ($Z_{\text{тр}}$) производится по формуле 7.4:

$$Z_{\text{тр}} = \sum_{i=1}^n \text{ЧС}_i \times T, \quad (7.4)$$

где T – количество часов, затраченных на дипломное проектирование,

ЧС_i – часовая ставка сотрудника,

i – номер сотрудника,

n – общее количество сотрудников.

Расчеты заработных отчислений приведены в таблице 7.6.

Таблица 7.6 – Расчет заработной платы сотрудников

Квалификация сотрудника	Трудоемкость, чел.× час.	Часовая ставка, руб.	Сумма, руб.
1. Инженер-программист	540	80	43 200
2. QA-специалист	10	100	1 000
Итого затрат на начисление заработных плат			44 200

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

7.4 Расчет отчислений в социальные фонды

Обязательное страхование составляет часть государственной системы социальной защиты граждан, спецификой которой является осуществление в соответствии с федеральным законом страхования работающих граждан Российской Федерации от возможного изменения материального и социального положений. Страховой случай наступает в случае нетрудоспособности гражданина.

В таблице 7.7 приведены расчеты обязательных страховых взносов.

Таблица 7.7 – Отчисления на обязательные страховые взносы

	Пенсионный фонд			ФФОМС	ФСС	Страхование от несчастных случаев	Налоговая нагрузка
	Для лиц 1966 г.р. и старше	Для лиц 1967 г.р. и моложе					
	Страховая часть	Страховая часть	Накопительная часть				
Общий режим	22	16	6	5,1	2,9	0,2	30,2
Плательщики, применяющие УСН							
Плательщики, переведенные на ЕНВД							
Начислено заработной платы, руб.							44 200
Сумма взносов, руб.							13 348,4

7.5 Расчет амортизации оборудования

Амортизация оборудования - это исчисляемый в денежном выражении износ основных средств в процессе их производственного использования. Расходы на амортизацию (P_a) выражаются при помощи формулы 7.5:

$$P_a = \sum_{i=1}^n \frac{C_i \times H_{ai} \times T_{pi}}{100 \times T_{\varepsilon i}}, \quad (7.5)$$

где C_i – стоимость i -го оборудования,

H_{ai} – годовая норма амортизации i -го оборудования, %,

T_{pi} – время работы i -го оборудования за весь период разработки дипломного проекта, ч,

$T_{\varepsilon i}$ – эффективный фонд времени работы i -го оборудования за год, ч/год,

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

/лист

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

i – вид оборудования,

n – общее число различного оборудования.

При установленной годовой норме амортизационных начислений в 20 % расходы составят:

$$P_a = \frac{26\,700 \times 20 \times 540}{100 \times 1\,750} + \frac{4\,750 \times 20 \times 1}{100 \times 1\,750} = 1649 \text{ руб.}$$

7.6 Расчет себестоимости разработки

Стоимость разработки информационной системы вычисляется из совокупности всех средств затраченных на материальные ресурсы и расходные материалы. При подсчете также учитывается заработная плата сотрудников и обязательные отчисления в фонды социального страхования.

Расчет стоимости программного продукта, разработанного в рамках дипломного проектирования приведен в таблице 7.8:

Таблица 7.8 – Расчет себестоимости программного обеспечения

Статьи затрат	Сумма, руб
1. Затраты на материальные ресурсы	31 950
2. Затраты на расходные материалы	1 600
3. Затраты на электроэнергию	174,29
4. Затраты на заработную плату сотрудникам	44 200
5. Обязательные страховые отчисления	13 348,4
6. Амортизация оборудования	1649
Итого:	92 921,7

7.7 Расчет плановой прибыли

От того, насколько достоверно определена плановая прибыль, будет зависеть успешная финансово-хозяйственная деятельность предприятия. Расчет плановой прибыли должен быть экономически обоснованным. Это позволит осуществлять своевременное и полное финансирование инвестиций, прироста собственных оборотных средств и соответствующих выплат сотрудникам.

Плановая прибыль реализации программного решения рассчитывается по формуле 7.6:

$$П = \frac{C_{\text{пол}} \times P_n}{100}, \quad (7.6)$$

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

где $C_{\text{пол}}$ – полная себестоимость, руб.,

R_n – норматив рентабельности.

При нормативе рентабельности, равном 30%, прибыль будет составлять 26 908,9 руб. С учетом налога на прибыль, составляющим 20 %, доход составит:

$$27\,876,5 - 0,2 \times 27\,876,5 = 22\,301,2 \text{ руб.}$$

7.8 Расчет основных технико-экономических показателей и эффективности использования программного продукта

Экономический эффект – это величина, характеризующая достигнутые благодаря созданию или совершенствованию ПО дополнительные экономические результаты.

Экономическая эффективность – результативность экономической деятельности, экономических программ и мероприятий, характеризуемая отношением полученного экономического эффекта к затратам факторов, обусловившим получение этого результата.

Основными пользователями информационной системы “Объектовый учет” являются управляющие компании, которые занимаются сбором данных по домам находящимся под их управлением и решением проблем сферы ЖКХ. Другая информационная система федерального масштаба “Реформа ЖКХ” заинтересована в получении и раскрытии данных управляющих компаний со всей территории РФ. Следует заметить, что обмен информацией с федеральной системой обязателен для управляющих компаний в соответствии с постановлением правительства Российской Федерации от 23.09.2010 года № 731 “Об утверждении стандарта раскрытия информации организациями, осуществляющими деятельность в сфере управления многоквартирными домами”.

До разработки интеграционного слоя между представленными информационными системами весь обмен данными между управляющими

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

компаниями и “Реформой ЖКХ” был не автоматизирован. Как правило, в управляющей компании был сотрудник, который занимался сбором необходимой информации и заполнял ее на сайте “Реформы”. Заработная плата среднестатистического офисного сотрудника составляет 20 000 руб. Помимо ее выплаты компания делает обязательные страховые отчисления в размере 30,2% от заработной платы работника. Несложно посчитать затраты управляющий компании за год:

$$ЗП = 12 \times (20\,000 + 20\,000 \times 0,302) = 312\,480 \text{ руб. (26\,040 руб./мес.)}$$

При согласии управляющих компаний использовать встроенный интеграционный модуль системы “Объектовый учет” работа по информационному обмену будет полностью автоматизирована. Учитывая стоимость разработки (92 921,7 руб.), программное решение покажет свою экономическую выгоду уже после 4 месяцев внедрения.

Чтобы рассчитать экономическую эффективность разработанного ПО необходимо воспользоваться формулой для расчета экономического эффекта 7.7:

$$\mathcal{E}_э = \frac{\Pi}{C_{\text{пол}}} \times 100\% = 24\%, \quad (7.7)$$

где $\mathcal{E}_э$ – экономический эффект, %,

Π – прибыль (с вычетом налога на прибыль), руб.,

$C_{\text{пол}}$ – себестоимость проекта, руб.

Экономический эффект равен:

$$\mathcal{E}_э = \frac{22\,301,2}{92\,921,7} \times 100\% = 24\%$$

Нормативное значение экономической эффективности составляет 15 %. Так как рассчитанный коэффициент превышает эту величину, то производство и внедрение интеграционной системы считается эффективным.

<div>Подп. и дата</div> <div>Инв. № дубл.</div> <div>Взам. инв. №</div> <div>Подп. и дата</div> <div>Инв. № подл.</div>									
	Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ			

8 Безопасность и экологичность проекта

8.1 Исходные данные

Исходные данные дипломного проектирования описаны в таблице 8.1.

Таблица 8.1 – Исходные данные

Данные	Название
1. Тема дипломного проекта	Подсистема интеграции с федеральной системой «Реформа ЖКХ» для «АИС: Объектовый учет».
2. Технологический процесс	Проектирование интеграционного модуля для двух ИС.
3. Оборудование, в т.ч. паспортные данные	Ноутбук, принтер.
4. Персонал (состав, профессии)	Инженер-программист, сотрудник отдела QA (отдел контроля качества).
5. Исходное состояние системы, ресурсы, материалы	Сервер – основной ресурс, на котором осуществляется хранение данных ИС «АИС: Объектовый учет», запуск интеграционной службы, обеспечивающей обмен данными.
6. Энергоносители (электричество, вода, пар, газ, уголь) и их характеристики	Энергоноситель – электрическая сеть с напряжением 220 В.
7. Расположение рабочего места, функции персонала	Рабочий персонал располагается в офисе. Каждый сотрудник имеет персональный компьютер, с помощью которого выполняет свои прямые обязанности. Инженер-программист проектирует программные решения и занимается автоматизацией процессов. Сотрудник отдела QA тестирует и выявляет ошибки программного обеспечения, разработанного программистом.
8. Признаки отнесения объекта к опасным промышленным объектам	-
9. Санитарная характеристика объекта	-
10. Характеристика помещений по электроопасности	Помещение без повышенной опасности.
11. Характеристика среды помещений	Сухие помещения.
12. Категория производства по взрывопожарной опасности	Д – пониженная пожароопасность.
13. Класс пожароопасной зоны	-
14. Класс взрывоопасной зоны	-
15. Рассматриваемые стадии «жизненного цикла» продукции	реализация программного решения, альфа-тестирование, бета-тестирование, написание рабочей документации, процесс внедрения
16. Классы условий труда в соответствии с Картой аттестации рабочего места по условиям труда: по вредности, по	По вредности – вредный (III класс). По травмоопасности – допустимый (II класс).

Подп. и дата

Инв. № ауд.

Взам. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

травмоопасности	
17. Вредные и опасные производственные факторы	См. рис. 8.2
18. Виды загрязнений окружающей среды	-
19. Возможные чрезвычайные ситуации	См. п. 9.3.3

8.2 Перечень нормативных документов и актов

В списке представлен перечень основных нормативных документов:

- а) Санитарные правила и нормы. СанПиН 2.2.2./2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы;
- б) «Руководство по гигиенической оценке факторов рабочей среды и трудовых процессов. Критерии и классификация условий труда». Р 2.2.2006-05;
- в) ГОСТ 12.0.003-74.ССБТ. (СТ СЭВ 790-77) Опасные и вредные производственные факторы. Классификация. М.: Изд-во стандартов, 1996;
- г) ГОСТ 12.1.003-83.ССБТ. Шум. Общие требования безопасности. М.: Изд-во стандартов.1996;
- д) ГОСТ 12.1.004-91.ССБТ. Пожарная безопасность. Общие требования. М.: Изд-во стандартов, 1996;
- е) ГОСТ 12.1.005-88.ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны. М.: Изд-во стандартов, 1996.
- ж) ГОСТ 12.1.006-88.ССБТ. Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля. М.: Изд-во стандартов, 1998;
- з) ГОСТ 12.1.019-79.ССБТ (СТ СЭВ 4880-84). Электробезопасность. Общие требования. М.: Изд-во стандартов, 1996;
- и) ГОСТ 12.1.030-81.ССБТ. Электробезопасность. Защитное заземление. М.: Изд-во стандартов, 1996;
- к) ГОСТ 12.1.038-82.ССБТ. Электробезопасность. Предельно-допустимые значения напряжений прикосновения и токов. М.: Изд-во стандартов, 1996;

Инв. № подл.	Взам. инв. №	Инв. № дубл.	Подп. и дата							
					ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

- л) Правила устройства электроустановок. М.: Энергия, 1987;
- м) Общесоюзные нормы технологического проектирования ОНТП 24-86., М.: МВД СССР, 1986;
- н) СНиП 2.01.02-85. Противопожарные нормы. М.: Стройиздат, 1986;
- о) СНиП 2.04.05-86. Отопление, вентиляция, кондиционирование воздуха. М.: Стройиздат, 1988;
- п) СНиП 23-05-95. Естественное и искусственное освещение. Анализ проектирования. М.: Энерго, 1996;
- р) Р 2.2.013-94. Гигиена труда. М.: Госкомсанэпиднадзор России, 1994;
- с) Правила пожарной безопасности в Российской Федерации – ППБ 01 03;
- т) Нормы пожарной безопасности – НПБ 88-2001. Установки пожаротушения и сигнализации. Нормы и правила проектирования.

8.3 Анализ потенциальных опасностей

Опасность – это возможность возникновения обстоятельств, при которых материя, поле, энергия, информация или их сочетание могут таким образом повлиять на сложную систему, что приведет к ухудшению или невозможности ее функционирования и развития; вероятность того, что может произойти какой-то вред.

Помимо множества определений понятия «опасности» существует аксиома, гласящая: «Ни в одном виде деятельности невозможно достичь абсолютной безопасности, поэтому любая деятельность потенциально опасна».

Следуя аксиоме логично предположить, что процесс проектирования дипломного проекта также таит в себе ряд опасностей. В основном они относятся к группе техногенных опасностей, потому что разработка связана с использованием офисного оборудования.

Во избежание возникновения негативных производственных факторов была разработана блок-схема обеспечения безопасности объекта проектирования, представленная на рисунке 8.1.

<div>Подп. и дата</div> <div>Инв. № дубл.</div> <div>Взам. инв. №</div> <div>Подп. и дата</div> <div>Инв. № подл.</div>						<div>ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ</div> <div>Изм. Лист № докум. Подп. Дата</div>	<div>Лист</div>

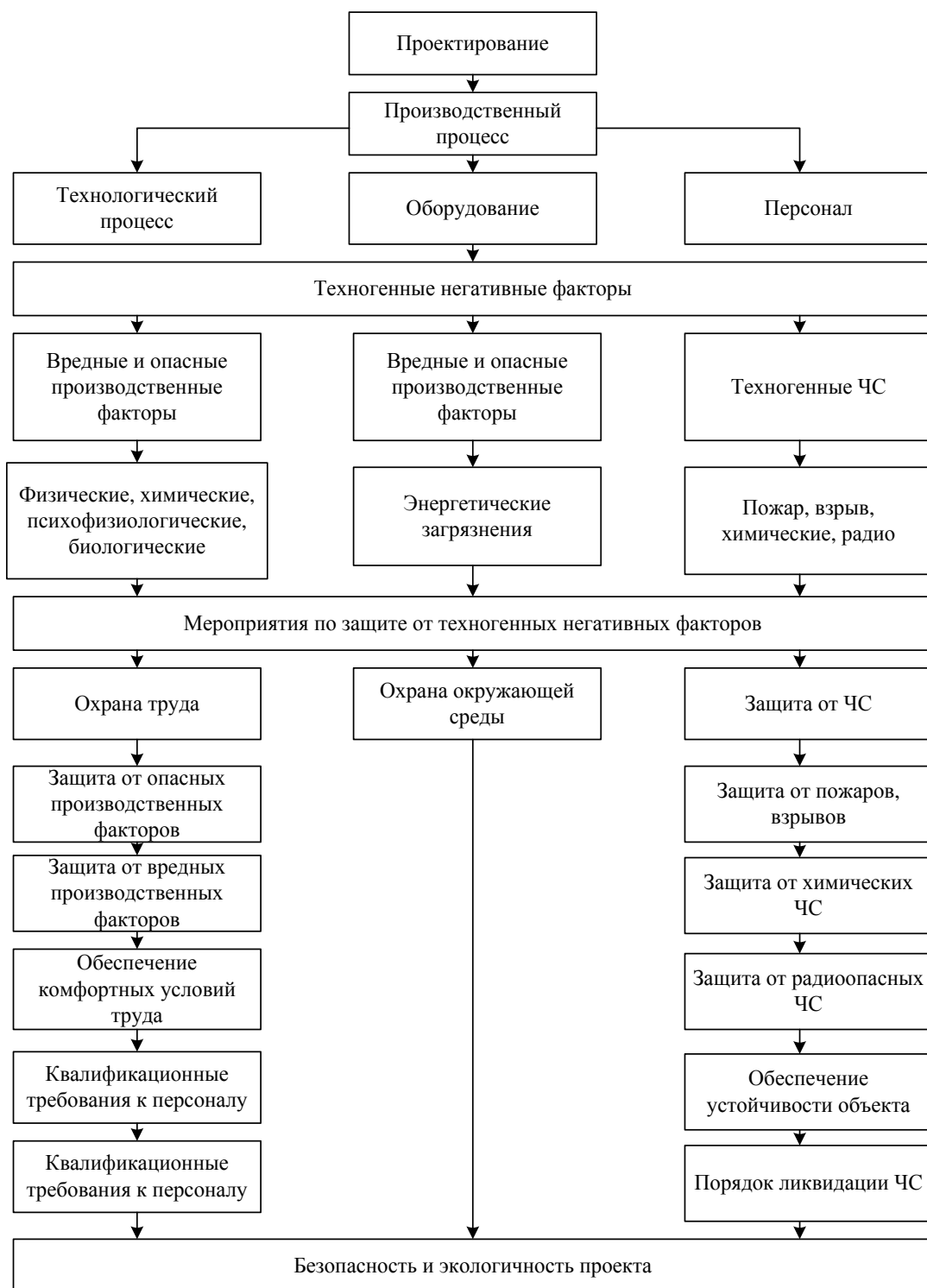


Рисунок 8.1 – Обеспечение безопасности объекта проектирования

8.3.1 Анализ вредных и опасных производственных факторов

В процессе трудовой деятельности человек может подвергаться вредным и опасным производственным факторам. Вредные факторы вызывают заболевания,

Подп. и дата

Инв. № аудл.

Взам. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Копировал

Формат А4

а опасные травмы.

ГОСТ 12.0.003-74 разделяет производственные факторы на четыре группы: физические, химические, биологические и психофизиологические. На рисунке 8.2 приведена их классификация [2].

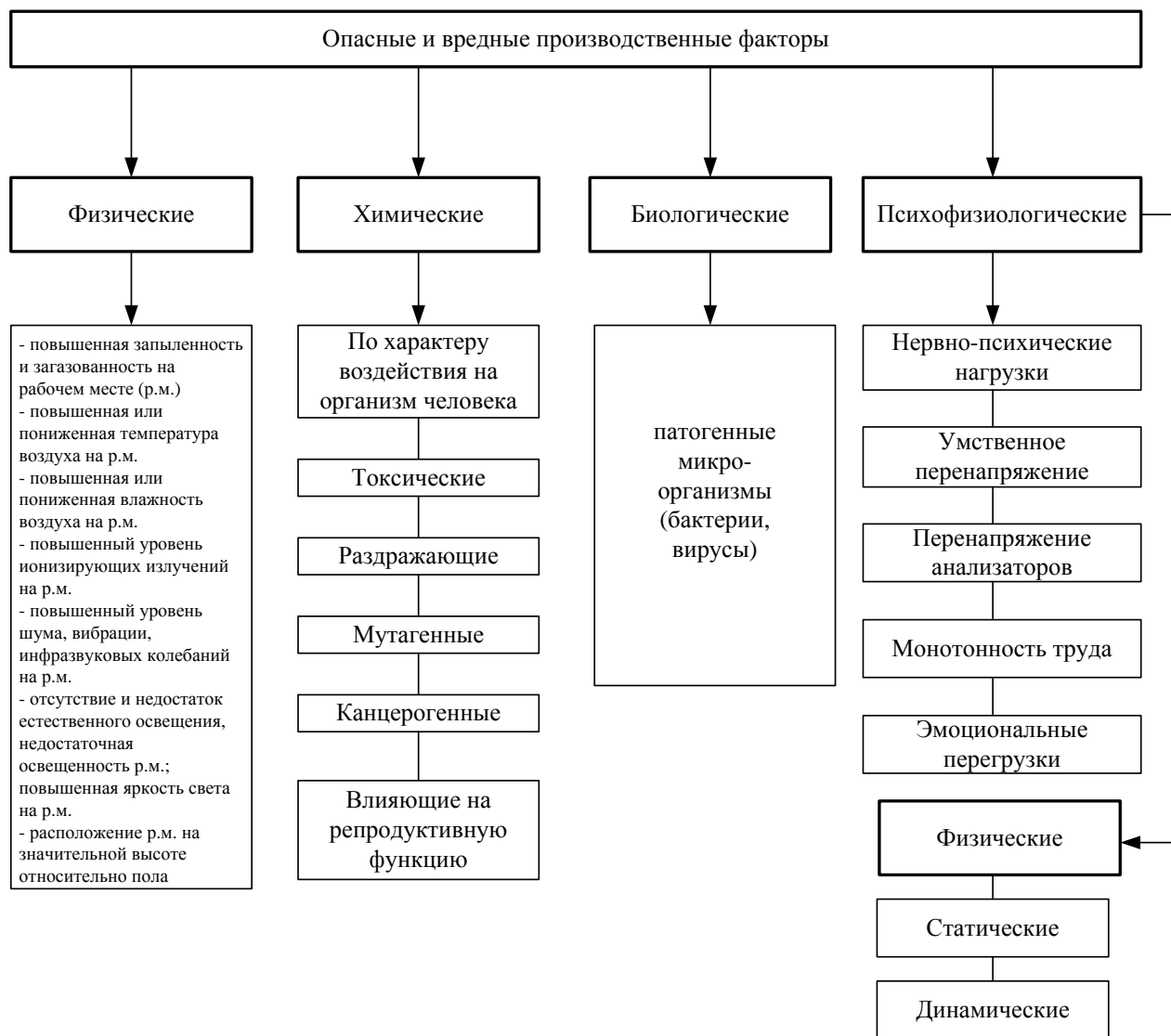


Рисунок 8.2 – Классификация опасных и вредных производственных факторов

На рабочее место инженера-программиста свое влияние оказывают, как правило, только две группы факторов: психофизиологические и физические.

Психофизиологические факторы проявляются в связи со специфическими условиями труда. В процессе трудовой деятельности программист испытывает:

а) умственное перенапряжение, потому что решаемые задачи часто требуют

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

нестандартных решений;

б) эмоциональные перегрузки, когда обнаруженная проблема или поставленная задача требуют немедленного решения;

в) монотонность труда, если процесс разработки однообразен;

г) нервно-психические перегрузки из-за систематического недосыпания.

При игнорировании или не проведении профилактических мероприятий (полноценный отдых, отвлечение от рабочего процесса) все перечисленные психофизиологические элементы могут привести к серьезным заболеваниям нервной системы и нарушениям мозговой деятельности.

Физические факторы оказывают негативное влияние на работника. На рабочем месте инженера-программиста таких факторов может быть несколько:

а) пыль на рабочем месте – при больших скоплениях может вызывать заложенность дыхательных путей или появление аллергических заболеваний (кашель, сыпь и т.д.);

б) недостаточная освещенность или излишняя яркость экрана – вызывает перенапряжение глаз, что в итоге приводит к сильному ухудшению зрения;

в) шум от работающего компьютерного вентилятора. Наличие постоянного шумового «сопровождения» притупляет работу органов слуха.

8.3.1.1 Гигиенические требования к организации рабочего места

Рабочее место, оснащенное ПЭВМ должно полностью соответствовать гигиеническим требованиям, описанным в документе СП 2.2.2/2.4.1340-03. Наиболее важное положение – это выбор помещения для рабочего места сотрудника, которое должно удовлетворять следующим критериям:

а) при выборе рабочего места, оборудованного компьютером, предпочтение следует отдавать помещениям, ориентированным на север или северо-восток;

б) не допускается оборудовать рабочее место для работы с ПК в помещениях, расположенных в цокольных и подвальных помещениях;

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

в) площадь на одно рабочее место с ПК на базе электронно-лучевой трубки должна составлять не менее 6 м², а при работе с жидкокристаллическими (плазменными) мониторами – 4,5 м²;

г) рабочий стол (рабочее место) размещают таким образом, чтобы монитор был ориентирован боковой стороной к окнам, а естественный свет падал преимущественно слева;

д) в целях защиты от прямых солнечных лучей на окнах должны быть солнцезащитные жалюзи или однотонные шторы из плотной ткани, которые выполняют и функцию звукопоглощения. Цвет штор должен гармонировать с цветом стен, ширина занавесей в 2 раза больше ширины окна, для того чтобы их можно было повесить в складки;

е) в помещении проводится ежедневная влажная уборка, включая уборку пыли с экранов мониторов, столов, подоконников, а также систематическое проветривание после каждого часа работы.

Конструктивные особенности рабочего места должны удовлетворять следующим правилам:

а) экран монитора должен находиться от глаз пользователя на расстоянии не менее 50 см (оптимально 60 - 70 см). Уровень глаз должен приходиться на центр или 2/3 высоты экрана;

б) конструкция рабочего места (стола, стула, кресла и т.п.) должна обеспечивать поддержание рациональной рабочей позы - корпус выпрямлен, сохранены естественные изгибы позвоночного столба, голова слегка наклонена вперед, для снятия статического напряжения с плечевого пояса и рук предплечья опираются на поверхность стола. Рациональная рабочая поза позволит снизить напряжение мышц шейно-плечевой области, мышц спины и предупредить развитие утомления;

в) рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию;

г) при наличии высокого стола и стула, не соответствующих росту работника, следует использовать регулируемую по высоте подставку для ног.

Важным гигиеническим фактором рабочего процесса является удобство восприятия картинки монитора ПЭВМ. Для этого экран должен удовлетворять требованиям:

- а) возможность изменения угла поворота в горизонтальной и вертикальной плоскостях с фиксацией в удобном для пользователя положении;
- б) возможность регулировки яркости и контрастности матрицы экрана;
- в) поверхность экрана должна быть матовой, что позволит избежать появления бликов.

В процессе работы за ПВЭМ необходимо строго соблюдать режим труда и отдыха. Максимальная непрерывная продолжительность работы за компьютером не должна составлять более 40 минут, после прохождения, которых необходимо выполнять профилактические упражнения для органов зрения и другие физкультурные мероприятия.

8.3.1.2 Микроклимат на рабочем месте

Микроклимат помещений – это климат внутренней среды помещений, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха, а также температуры окружающих поверхностей.

ГОСТ 12.1.005-88 описывает основные параметрические требования к микроклимату помещений для работы с ПЭВМ:

- а) температура помещения в теплый период должна составлять 20-24°C, в холодный – 22-24 °C, в переходный – 18-22 °C;

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

б) допуск температурных колебаний должен быть не более 4%;

в) подвижность воздуха – от 0,1 до 0,2 м/с;

г) влажность воздуха – 60-70%;

д) недопустимость запыленности и загазованности воздуха.

Нарушение указанных норм может привести к появлению ряда негативных последствий для человека:

а) снижение работоспособности;

б) нарушение самочувствия.

8.3.1.3 Освещенность рабочего места

Свет играет очень большую роль в рабочем процессе. Хорошо спроектированная система освещения способствует повышению работоспособности и оказывает положительное психологическое воздействие на трудящегося.

При недостатке освещенности на рабочем месте обостряются проблемы со зрением, так как человек, работающий за ПЭВМ, вынужден больше напрягать зрительные органы.

Существует три вида освещения рабочего пространства: естественное, искусственное и совмещенное.

Естественное освещение – это освещение, создаваемое направленным или рассеянным светом или светом неба, проникающим через световые проемы помещения.

Искусственным освещением называется освещение рабочего пространства, создаваемое при помощи осветительных приборов. Данный тип освещения применяется при работе в темное время суток и днем, когда не удастся обеспечить нормированные показатели естественного освещения. Одним из типов искусственного освещения является рабочее (остальные типы: аварийное, эвакуационное и охранное), которое подразделяется на общее и комбинированное. Комбинированный тип освещения должен применяться в

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 8.2 – Нормы освещенности рабочего места

	Общая освещенность, лк	Комбинированная освещенность, лк
Зрительная работа высокой точности	300	750
Зрительная работа средней точности	200	300

8.3.1.4 Шум на рабочем месте

На рабочем месте оператора ПЭВМ существует множество различных источников шума: компьютерный вентилятор, жесткий диск, печатающий принтер и т.д.

В соответствии с СанПиН 2.2.2./2.4.1340-03 допустимый уровень звукового давления при работе с ПЭВМ не должен быть больше 65 дБ. Если указанный шумовой порог превышает, то нормирование уровня звука достигается за счет установки малозумного оборудования и установки специальных звукопоглощающих панелей, корпусов для оборудования.

8.3.1.5 Электромагнитные и ионизирующие излучения на рабочем месте

Электромагнитное излучение – это излучения, которое прямо или косвенно

вызывает ионизацию среды. Для человека данный тип излучения представляет большую опасность и приводит к возникновению различных патологий.

Основным источником электромагнитного излучения на рабочем месте инженера-программиста является монитор, и чтобы защититься от его вредного воздействия необходимо следовать правилам:

- а) произвести замену ЭЛТ-монитора (электронно-лучевая трубка) на жидкокристаллический или LED, так как уровень их излучения на порядок ниже;
- б) установить монитор в угол, чтобы излучение поглощалось стенами;
- в) по возможности сокращать время работы за компьютером.

Ионизирующее излучение – это излучение, вызывающее ионизацию среды, очень опасно для человека, так как потоки заряженных частиц приводят к нарушениям работы организма на клеточном уровне.

Средствами защиты от ионизирующего излучения являются:

- а) регламентирование времени работы за компьютером согласно санитарным нормам;
 - б) нахождение на расстоянии 50-70 см и большем от экрана монитора.
- Излучение на этой дистанции составляет 0,08 мкР/ч, что соответствует допустимой нормы.

8.3.2 Анализ воздействия на окружающую среду

Проектирование квалификационной работы подразумевает под собой разработку программного продукта и написание технической документации. Никакого воздействия на окружающую среду не оказывается.

8.3.3 Анализ возможных чрезвычайных ситуаций

Чрезвычайной ситуацией (ЧС) называется обстановка на определенной территории, сложившаяся в результате аварии, опасного природного явления, катастрофы, стихийного или иного бедствия, которые могут повлечь или

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

повлекли за собой человеческие жертвы, ущерб здоровью людей или окружающей природной среде, значительные материальные потери и нарушение условий жизнедеятельности людей. Классификация типов ЧС представлена на рисунке 9.5.



Рисунок 9.5 Классификация типов ЧС

Инв. № подл.	Взаим. инв. №	Инв. № дубл.	Подп. и дата	
Инв. № подл.	Взаим. инв. №	Инв. № дубл.	Подп. и дата	
Изм.	Лист	№ докум.	Подп.	Дата
ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ				
Лист				

В процессе проектирования квалификационной работы существует вероятность возникновения ЧС природного и антропогенного характеров. Вероятности возникновения опасностей этих категорий выше на территории Ульяновской области.

К ЧС природного характера можно отнести неконтролируемые природные явления такие как: паводки в весенний период, штормовые ветра или оползень.

Антропогенные аварии возникают в результате хозяйственной деятельности человека. Дипломное проектирование проводится в здании, смежном со зданием старой городской почты. Помещения арендуются многими предприятиями, и как следствие этого высокая плотность постоянно работающей офисной и бытовой техники. Наличие большого числа электроприборов может привести к перегрузке энергосети, что повлечет за собой пожар.

К возможным техногенным авариям в процессе дипломного проектирования можно отнести:

- а) пожары, взрывы бытового газа;
- б) обрушение здания.

8.4 Мероприятия по охране труда

Охраной труда называется система сохранения жизни и здоровья работников в процессе их трудовой деятельности, включающая в себя правовые, социально-экономические, организационно-технические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия. Охрана труда условно подразделяется на четыре составляющие:

- а) правовая охрана труда (ПОТ);
- б) техника безопасности (ТБ);
- в) производственная санитария (ПС);
- г) пожарная безопасность (ПБ).

В соответствии со ст. 210 ТК РФ основными направлениями государственной политики в области охраны труда являются:

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

- а) обеспечение приоритета сохранения жизни и здоровья работников;
- б) принятие и реализация федеральных законов и иных нормативных правовых актов Российской Федерации, законов и иных нормативных правовых актов субъектов Российской Федерации в области охраны труда, а также федеральных целевых, ведомственных целевых и территориальных целевых программ улучшения условий и охраны труда;
- в) государственное управление охраной труда;
- г) государственный надзор и контроль над соблюдением государственных нормативных требований охраны труда;
- д) государственная экспертиза условий труда;
- е) профилактика несчастных случаев и повреждения здоровья работников;
- ж) расследование и учет несчастных случаев на производстве и профессиональных заболеваний;
- з) защита законных интересов работников, пострадавших от несчастных случаев на производстве и профессиональных заболеваний, а также членов их семей, на основе обязательного социального страхования работников от несчастных случаев на производстве и профессиональных заболеваний;
- и) установление компенсаций за тяжелую работу и работу с вредными и опасными условиями труда;
- к) обеспечение функционирования единой информационной системы охраны труда.

8.4.1 Мероприятия по обеспечению комфортных условий труда

Обеспечение комфортных условий для трудовой деятельности позволяет повысить качество и производительность труда, обеспечить хорошее самочувствие и наилучшие для сохранения здоровья параметры среды обитания и характеристики трудового процесса.

Согласно ТОИ Р-45-084-01 от министерства РФ по связи и информатизации комфортные условия труда для работников, взаимодействующих с ПЭВМ,

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

закljučаются в установлении режима работы в зависимости от вида их трудовой деятельности:

- а) группа «А» – работа по считыванию информации с монитора ПЭВМ с предварительным запросом;
- б) группа «Б» – работа по вводу информации;
- в) группа «В» - творческая работа в режиме диалога с ПЭВМ.

При выполнении в течение рабочей смены работ, относящихся к различным видам трудовой деятельности, за основную работу с компьютером следует принимать такую, которая занимает не менее 50% времени в течение рабочей смены или рабочего дня.

Для групп категорий устанавливаются три типа тяжести и напряженности работы с ПЭВМ:

- а) для группы «А» - по суммарному числу считываемых знаков за рабочую смену, но не более 60000 знаков за смену;
- б) для группы «Б» - по суммарному числу считываемых или вводимых знаков за рабочую смену, но не более 40000 знаков за смену;
- в) для группы «В» - по суммарному времени непосредственной работы с ПЭВМ за рабочую смену, но не более 6 часов за смену.

В зависимости от категории трудовой деятельности и уровня нагрузки за рабочую смену при работе с ПЭВМ устанавливается суммарное время регламентированных перерывов. Если у работника наблюдается ухудшение зрения или другие неприятные ощущения, то рекомендуется применять индивидуальный регламент времени работы за ПЭВМ.

Для предупреждения преждевременной утомляемости пользователей ПЭВМ рекомендуется организовывать рабочую смену путем чередования работ с использованием ПЭВМ и без него.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

8.4.2 Мероприятия по защите от опасных и вредных производственных факторов

Основные мероприятия по защите работника от опасных и вредных производственных факторов (ОВПФ) заключаются:

- а) в использовании новых технологий на производстве с целью снижения уровня ОВПФ;
- б) удаление на расстояние от источника ОВПФ;
- в) уменьшение времени нахождения в зоне действия ОВПФ;
- г) установка регламента перерывов для рабочего процесса;
- д) применение средств защиты (индивидуальной и коллективной).

На рабочем месте инженера-программиста основными ОВПФ являются шум, электромагнитное и ионизирующее виды излучения, постоянная нагрузка на зрительные органы и сидячая поза. Для защиты от воздействия перечисленных факторов необходимо использовать превентивные меры:

- а) при использовании устаревшего ЭЛТ-монитора произвести его замену на LED- или ЖК-дисплей;
- б) установить расстояние до экрана на 50-70 см и более;
- в) для защиты от шума использовать наушники, специальные звукопоглощающие контейнеры или произвести замену оборудования на малошумные аналоги;
- г) во время перерывов проводить специальные упражнения для снятия напряжения с органов зрения.

8.5 Мероприятия по охране окружающей среды

Охраной окружающей среды называется деятельность органов государственной власти РФ, органов местного самоуправления, общественных и иных некоммерческих объединений, юридических и физических лиц, направленная на сохранение и восстановление природной среды, рациональное

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

использование и воспроизводство природных ресурсов, предотвращение негативного воздействия хозяйственной или иной деятельности на окружающую среду и ликвидацию ее последствий.

В статье 1 федерального закона «Об охране окружающей среды» закреплены основные принципы охраны окружающей среды:

- а) соблюдение права человека на благоприятную окружающую среду;
- б) обеспечение благоприятных условий жизнедеятельности человека;
- в) обязательность оценки воздействия на окружающую среду при принятии решений об осуществлении хозяйственной и иной деятельности;
- г) обязательность проведения государственной экологической экспертизы проектов и иной документации, обосновывающих хозяйственную и иную деятельность, которая может оказать негативное воздействие на окружающую среду, создать угрозу жизни, здоровью и имуществу граждан.

Разрабатываемый дипломный проект оперирует цифровыми данными и, как следствие, не оказывает влияния на окружающую среду. Все права человека на трудовую деятельность в благоприятных условиях соблюдаются.

8.6 Мероприятия по защите от чрезвычайных ситуаций

Система защиты населения от чрезвычайных ситуаций подразделяется на три основных уровня:

- а) предупреждение ЧС – включает в себя меры, направленные на прогнозирование и профилактику к возникновению ЧС, контроль над природной средой и потенциально опасными объектами, проведение мероприятий, связанных с подготовкой и обучением населения к действиям во время ЧС;
- б) защита в ЧС – включает проведение комплекса защитных мероприятий и использование средств индивидуальной и коллективной защиты;
- в) ликвидация ЧС – включает в себя работы и другие неотложные мероприятия, направленные на устранения последствий чрезвычайной

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист

ситуации.

В самом начале процесса разработки дипломного проекта был проведен инструктаж по технике безопасности на рабочем месте. В течение беседы были подробно рассмотрены многие вопросы, касающиеся эвакуационных мероприятий в случае нештатной ситуации, правил пользования подручными и специализированными средствами пожаротушения. В рамках дисциплины «Безопасность жизнедеятельности» был успешно изучен комплекс мер по оказанию первой медицинской помощи пострадавшему.

8.7 Расчетная часть

В разделе приводятся расчеты уровня шума от электрооборудования и общей величины освещенности на рабочем месте.

8.7.1 Расчет уровня шума на рабочем месте

Шум является одним из неблагоприятных факторов на рабочем месте инженера-программиста. Его постоянное раздражающее воздействие негативно сказывается на умственной деятельности и производительности работника.

При работе за ПЭВМ очень трудно избежать шумового сопровождения, поэтому в данном случае можно лишь предпринять меры по снижению уровня громкости звука.

В помещении, в котором ведется дипломное проектирование, находятся несколько источников шума. В таблице 8.3 представлены данные, полученные в результате измерения уровня шума для каждого из них.

Таблица 8.3 – Уровни шума для каждого источника на рабочем месте

Источник шума	Уровень шума, дБ
Вентилятор	35
Принтер	45
Клавиатура	15
Жесткий диск	30

Источники шума некогерентные, поэтому при вычислении общего уровня шума можно воспользоваться формулой 8.1:

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

$$L = 10 \lg \sum_{i=1}^n 10^{0,1L_i}, \quad (8.1)$$

где L_i – уровень звукового давления i -го источника шума,

n – число источников шума.

Произведя необходимые расчеты при помощи формулы получаем результат:

$$L = 10 \lg(10^{3,5} + 10^{4,5} + 10^{1,5} + 10^3) = 46 \text{ дБ}$$

Допустимый уровень звукового давления согласно ГОСТ 12.1.003-83 составляет 65 дБ. Это означает, что шумовое воздействие не превышает норму.

8.7.2 Расчет величины освещенности рабочего пространства

В помещении, в котором ведется дипломное проектирование, преобладает искусственное освещение, которое обеспечивается люминесцентными лампами. Для расчета величины освещенности необходимо использовать формулу 8.2, описывающую метод светового потока:

$$F = \frac{E_{\text{норм}} \times K_3 \times S \times z}{q \times f}, \quad (8.2)$$

где F – световой поток, Лк,

$E_{\text{норм}}$ – нормированная минимальная освещенность, равная 200 Лк,

K_3 – коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации, равный 1,2 (для офисных помещений),

S – площадь помещения, м^2 ,

z – коэффициент неравномерности освещения, равный 1,15 (для люминесцентных ламп),

q – коэффициент использования светового потока (величина определяется с помощью индекса помещения и предполагаемых коэффициентов отражения поверхностей помещения),

f – коэффициент затемнения, равный 0,9.

Инв. № подл.	Взам. инв. №	Инв. № дубл.	Подп. и дата					
Инв. № подл.	Взам. инв. №	Инв. № дубл.	Подп. и дата					
				<div style="text-align: right;">/лист</div>				
				<div style="text-align: right;">ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ</div>				
Изм.	Лист	№ докум.	Подп.	Дата				

В таблице 8.4 представлены параметры помещения, в котором ведется дипломное проектирование.

Таблица 8.4 – Параметры помещения

Параметр	Значение
Длина, а, м	3,8
Ширина, b, м	6
Высота, h, м	3
Коэффициент отражения, R _{см}	30
Коэффициент отражения потолка, R _п	65

По измеренным параметрам определим площадь (S) помещения:

$$S = a \times b = 6 \times 3 = 18 \text{ м}^2$$

Величина площади необходима для расчета индекса помещения, который рассчитывается следующим образом:

$$i = \frac{S}{h \times (a + b)} = \frac{18}{3,8 \times (6 + 3)} = 0,53 \approx 0,6$$

При индексе помещения равном 0,6 коэффициент использования (q) составляет 0,28.

Произведем необходимые расчеты по нахождению величины параметра освещенности:

$$F = \frac{200 \times 1,2 \times 18 \times 1,15}{0,9 \times 0,28} = 19\,714 \text{ Лм}$$

Необходимое число ламп рассчитывается по формуле 8.3:

$$N = \frac{F}{F_{\text{л}}}, \quad (8.3)$$

где F – световой поток равный 19 714 Лм,

F_л – световой поток одной лампы.

Принимаем величину светового потока одной лампы равной 4450 Лм (модель лампы OSRAM HO 54 W/830 G5). В результате необходимого расчета количество ламп равняется:

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	<div style="text-align: right;"> <div>Лист</div> <div></div> </div>				
					<div style="text-align: center; font-size: 1.2em; font-weight: bold;">ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ</div>				
Изм.	Лист	№ докум.	Подп.	Дата					

$$N = \frac{19\,714}{4450} \approx 4,4 = 5$$

В действительности помещение оборудовано тремя светильниками по две лампы в каждом, что обеспечивает оптимальное освещение рабочего пространства.

8.8 Оценка эффективности принятых решений

В разделе «Безопасность и экологичность объекта проектирования» проведен полный анализ вредных и опасных факторов производства, характерных для процесса разработки выпускной квалификационной работы бакалавра по специальности «Информационные системы и технологии».

Разработанный проект подсистемы интеграции для федерального портала «Реформа ЖКХ» и «АИС: Объектовый учет» не оказывает никакого влияния на окружающую среду, так как является программным продуктом.

В процессе работы над проектом соблюдались все нормы трудовой деятельности:

- а) на предприятии было оборудовано полноценное рабочее место с персональным компьютером;
- б) использованный тип монитора – LED, оказывающий наименьшее воздействие на органы зрения человека;
- в) уровень шума на рабочем месте соответствовал норме;
- г) уровень освещенности рабочего места соответствовал норме;
- д) помещение, в котором производилась разработка проекта, оборудовано пожарной сигнализацией и ручными средствами пожаротушения – огнетушителями.

Приведенные вывод показывают, что условия труда, созданные работодателем, оптимальны и соответствовали всем санитарным нормам.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
					ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

Заключение

В результате проведения проектирования выпускной бакалаврской работы по теме «Подсистема интеграции для федерального портала «Реформа ЖКХ» и «АИС: Объектовый учет» были получены следующие результаты:

- а) все требования технического задания на разработку системы были соблюдены;
- б) проект реализован в полном объеме;
- в) налажен процесс интеграции данных на федеральный портал для управляющих компаний Ульяновской области, использующих «АИС: Объектовый учет».

Эффективность работы подсистемы интеграции подтверждается пользователями управляющих компаний, которые отмечают, что информация процессов домоуправления раскрывается в полном объеме.

Со времени начала запуска подсистемы интеграции заметно выросли следующие показатели раскрытия информации:

- а) скорость раскрытия информации по процессам домоуправления. Выгрузка данных осуществляется в автоматическом режиме, что значительным образом экономит время операторов управляющих компаний;
- б) частота обновления данных. В настоящий момент данные процессов домоуправления обновляются ежедневно. Если понадобится сократить этот временной интервал, то это можно легко выполнить с изменением соответствующих настроек приложения интеграции данных;
- в) правильность данных. Пользователю управляющей компании больше нет необходимости проводить сопоставление данных, хранящихся в объектовом учете и на федеральном портале, для синхронизации данных двух информационных источников. Подсистема интеграции выполняет этот процесс автоматически;

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

г) «дружелюбный» интерфейс (англ. friendly interface). Панель управления интеграцией позволяет пользователям управляющих компаний в системе «АИС: Объектовый учет» полностью контролировать процесс интеграции данных домоуправления. В случае ошибки при пересылке данных отображение соответствующего сообщения помогает оператору УК разрешить проблему самостоятельно. В случае возникновения трудностей всегда можно обратиться в службу поддержки проекта «АИС: Объектовый учет»;

д) автономность решения интеграции. Подсистема интеграции автоматически отслеживает все изменения данных, которые вносятся пользователями управляющих компаний, и обеспечивает формирование соответствующих интеграционных запросов.

В ходе разработки решения интеграции данных было разработано приложение с модульной архитектурой. Отдельные части программного решения планируется применить при проектировании подсистемы интеграции для ИС «ГИС ЖКХ» и «АИС: Объектовый учет». Применение паттерна проектирования «Очередь запросов» позволит реализовать любой процесс интеграции данных, основанный на запросах.

Проект подсистемы интеграции принимал участие в конкурсе программных продуктов на кафедре измерительно-вычислительных комплексов Ульяновского государственного технического университета. По результатам конкурса проект получил наивысшие баллы и был удостоен награды в виде диплома первой степени.

По проекту было опубликовано статьи: «Подсистема «Очередь запросов» как решение задачи интеграции региональной и федеральной систем объектового учета», «Организация межпрограммного взаимодействия федеральной и региональной систем объектового учета». Публикации размещены в сборнике научных трудов и докладов конференции ПИС-2015.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	<p>ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ</p>					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

Список литературы

1. Гамма, Э. Приемы объектно-ориентированного программирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб. : Питер, 2001. – 344 с.
2. Гончар, С.Т. Безопасность и экологичность объекта проектирования: учебное пособие по дипломному проектированию / С.Т. Гончар. – 2-е изд. – Ульяновск : УлГТУ, 2009. – 165 с.
3. ГОСТ 19.701-90. Единая система конструкторской документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. – М. : Стандартинформ, 2010.
4. ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам. – М. : Стандартинформ, 1996.
5. Постановление правительства Российской Федерации № 731 от 23 сентября 2010 г. «Об утверждении стандарта раскрытия информации организациями, осуществляющими деятельность в сфере управления многоквартирными домами» [Электронный ресурс] / КонсультантПлюс. – [Б. м. : б. и.], 2010. –
Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_169248/
(дата обращения: 12.05.2015).
6. Родионов В.В. Дипломное проектирование: учебно –методическое пособие для студентов специальности 23020165 «Информационные системы и технологии» / В.В. Родионов. – Ульяновск : УлГТУ, 2008. – 98 с.
7. Троелсен, Э. Язык программирования C# 2010 и платформа .NET 4 / Э. Троелсен. – 5-е изд. - М. : Вильямс, 2010. – 1392 с.
8. Фленов, М. Е. Библия C# / М.Е.Фленов. – 2-е изд. – СПб. : БХВ-Петербург, 2011. – 560 с.
9. Cheng, S. Microsoft Windows Communication Foundation 4.0 Cookbook for developing SOA Applications / S.Cheng. – Birmingham : Packt Publishing, 2010.

– 316 pp.

10. Lowy, J. Programming WCF Services / J. Lowy. – NY : O'Reilly – 2010. – 910 pp.

11. Mukherjee, S. Thinking in LINQ / S. Mukherjee. – NY : Appress, 2015. – 259 pp.

12. Palermo, J. ASP.NET MVC 4 in Action / J.Palermo, J. Bogard, E. Hexter, M. Hinze, J. Skinner. – Shelter Island, NY : Manning, 2012. – 440 pp.

13. Skeet, J. C# in depth. Third edition / J.Skeet. – Shelter Island, NY : Manning, 2014. – 614 pp.

Інб. № подл.	Подп. и дата				Інб. № аудл.	Взам. инб. №	Подп. и дата	Інб. № подл.	
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ				Лист

Приложение А

(обязательное)

Текст программы

Модуль обмена данными

```
using System;
using System.Linq;
using System.ServiceModel;
using AIS.HM.Model;
using System.Reflection;
using System.ServiceModel.Channels;
using System.Collections.Generic;
using System.Xml;
using System.Xml.Linq;
using System.Configuration;
using AIS.HM.ReformaGKHService.Classes;

#if DEBUG
using AIS.HM.ReformaGKHService.ReformaAPI_beta;
#else
using AIS.HM.ReformaGKHService.ReformaAPI;
#endif

namespace ReformaGKHService.Classes
{
    public class APIProvider
    {
        public delegate void ReplyCallback(Message _reply);
        private DB db;
        private DataBinder binder;
        private ApiSoapPortClient client;
        private ReloginHelper reloginner;
        private bool isSoapFaultMessage = false;
        private bool isLoginRepeatedly = false;
        private int? activeQueueId = null;
        private string activeMethodName = string.Empty;
        private int daysRange = 7;
        private AuthHeaderBehavior behavior = null;

        public string ActiveMethodName
        {
            get { return activeMethodName; }
            protected set { activeMethodName = value; }
        }
        public int? ActiveQueueId
        {
            get { return activeQueueId; }
            protected set { activeQueueId = value; }
        }

        #region Конструкторы
        public APIProvider(DB _db)
        {
            client = new ApiSoapPortClient();
            CommonInit(_db);
        }

        public APIProvider(DB _db, string _endPointAddress)
        {
            client = new ApiSoapPortClient(new BasicHttpBinding("ApiSoapBinding"), new
```

Подп. и дата	
Инв. № аудл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

					ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		


```

EndpointAddress(_endPointAddress));
    CommonInit(_db);
}

public void CommonInit(DB _db)
{
    int range;
    daysRange =
Int32.TryParse(ConfigurationManager.AppSettings["service_collector_days_range"], out range)
? range : daysRange;
    loginResponse = new LoginResponse();
    binder = new DataBinder('.');
    db = _db;
}
#endregion

#region Callbacks
public void HandleFaultReply(Message _reply)
{
    string description = string.Empty, errorCode = string.Empty;
    using (XmlDictionaryReader reader = _reply.GetReaderAtBodyContents())
    {
        XDocument xdoc = XDocument.Parse(reader.ReadOuterXml());
        errorCode = xdoc.Descendants().Where(i => i.Name == "code").FirstOrDefault().Value;
        description = xdoc.Descendants().Where(i => i.Name ==
"description").FirstOrDefault().Value;
    }
    bool isError = db.vw_ReformaSoapFaults.FirstOrDefault(i => i.Code ==
errorCode).IsError;
    if (isError)
    {
        if (activeQueueId.HasValue)
            Servant.SaveLog(db, behavior.S SoapReply, behavior.S SoapRequest,
activeQueueId.Value, true, Convert.ToInt32(errorCode), description);
        isSoapFaultMessage = true;
    }
    else
        isSoapFaultMessage = false;
}
#endregion

#region Аутентификация
public void Login()
{
    string login = string.Empty, password = string.Empty;
#if DEBUG
    login = ConfigurationManager.AppSettings["reforma_login_debug"];
    password = ConfigurationManager.AppSettings["reforma_password_debug"];
#else
    login = ConfigurationManager.AppSettings["reforma_login"];
    password = ConfigurationManager.AppSettings["reforma_password"];
#endif
    if (!isLoginRepeatedly)
    {
        client.Endpoint.Behaviors.Add(new AuthHeaderBehavior());
        reloginner = new ReloginHelper(this);
        reloginner.StartTimer();
        isLoginRepeatedly = true;
    }
    loginResponse.LoginResult = client.Login(login, password);
    Console.WriteLine("Логин... {0} - {1}",
ConfigurationManager.AppSettings["reforma_login"],
ConfigurationManager.AppSettings["reforma_password"]);
    Console.WriteLine("Пространство имен - {0}", client.GetType());
    AuthHeaderBehavior customBehavior = (client.Endpoint.Behaviors.First(i => i.GetType()

```

Подп. и дата

Инд. № ауд.

Взам. инд. №

Подп. и дата

Инд. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

```

== typeof(AuthHeaderBehavior)) as AuthHeaderBehavior);
    customBehavior.TokenProvider.LogKey = loginResponse.LoginResult;
    customBehavior.ApiProvider = this;
    behavior = customBehavior;
}
public void Logout()
{
    Console.WriteLine("Выход из системы...");
    (client.Endpoint.Behaviors.First(i => i.GetType() == typeof(AuthHeaderBehavior)) as
AuthHeaderBehavior).ApiProvider = null;
    try
    {
        client.Logout();
    }
    catch (Exception e)
    {
        Console.WriteLine("Обнаружены проблемы при выходе из системы. Причина: {0}",
e.Message);
    }
}
#endregion

#region Set-методы

#region Методы организации
private void SetRequestForSubmit(IEnumerable<vw_ReformaActionQueue> _queueItems)
{
    activeMethodName = MethodBase.GetCurrentMethod().Name;
    Console.WriteLine("{0}...", activeMethodName);
    _queueItems = Servant.FilterQueueActionsByMethodName(db, _queueItems,
activeMethodName);

    if (_queueItems.Count() == 0)
        return;

    if (reloginner.IsSessionExpired)
        reloginner.Relogin();

    SetRequestForSubmitInnStatus[] reformFeedback =
client.SetRequestForSubmit(_queueItems.Select(i => i.Inn).ToArray());
    IQueryable<vw_ReformaAPIFeedback> feedbackCodes = db.vw_ReformaAPIFeedback.Where(i =>
i.MethodName == activeMethodName);
    vw_ReformaAPIFeedback reply = null;
    vw_ReformaActionQueue queueItem = null;
    vw_ReformaActionQueueLog logItem = null;
    bool isError = false;

    foreach (var feedbackItem in reformFeedback)
    {
        reply = feedbackCodes.First(i => i.Code == feedbackItem.status);
        queueItem = _queueItems.First(i => i.Inn == feedbackItem.inn);
        isError = reply.IsError.HasValue ? reply.IsError.Value : false;
        Servant.SaveLog(db, behavior.SoapReply, behavior.SoapRequest, queueItem.Id,
isError, feedbackItem.status, reply.Message);

        queueItem.Updated = DateTime.Now;
        if (!reply.IsError.HasValue || !reply.IsError.Value)
            queueItem.Completed = DateTime.Now;
        db.Save(queueItem);
    }
    db.SaveChanges();
}

private void SetRequestForSubmit(IEnumerable<vw_ReformaActionQueue> _queueItems, int
_chunkSize)
{

```

Подп. и дата

Инд. № ауд.

Взаим. инд. №

Подп. и дата

Инд. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

```

        activeMethodName = MethodBase.GetCurrentMethod().Name;
        Console.WriteLine("{0}...", activeMethodName);
        _queueItems = Servant.FilterQueueActionsByMethodName(db, _queueItems,
activeMethodName).OrderBy(i => i.Inn);
        SetRequestForSubmitInnStatus[] requestsStatus = null; //отправляем пачками, иначе
time-out
        IQueryable<vw_ReformaAPIFeedback> feedbackCodes = db.vw_ReformaAPIFeedback.Where(i =>
i.MethodName == activeMethodName);
        List<string[]> innGroups = new List<string[]>();
        vw_ReformaAPIFeedback reply = null;
        vw_ReformaActionQueue queueItem = null;
        bool isError = false;
        int requestCount = (int)_queueItems.Count() / _chunkSize;
        int remainedCount = _queueItems.Count() - requestCount * _chunkSize;

        if (requestCount != 0)
        {
            for (int i = 0; i < requestCount; i++)
            {
                if (i != requestCount - 1)
                    innGroups.Add(_queueItems.Skip(i * _chunkSize).Take(_chunkSize).Select(n =>
n.Inn).ToArray());
                else
                    innGroups.Add(_queueItems.Skip(i * _chunkSize).Take(_chunkSize +
remainedCount).Select(n => n.Inn).ToArray());
            }
        }
        else
        {
            innGroups.Add(_queueItems.Select(i => i.Inn).ToArray());
        }

        foreach (var item in innGroups)
        {
            if (reloginner.IsSessionExpired)
                reloginner.ReLogin();
            requestsStatus = client.SetRequestForSubmit(item);
            foreach (var status in requestsStatus)
            {
                reply = feedbackCodes.First(i => i.Code == status.status);
                queueItem = _queueItems.First(i => i.Inn == status.inn);
                isError = reply.IsError.HasValue ? reply.IsError.Value : false;
                Servant.SaveLog(db, behavior.SoapReply, behavior.SoapRequest, queueItem.Id,
isError, status.status, reply.Message);

                if (!reply.IsError.HasValue || !reply.IsError.Value)
                    queueItem.Completed = DateTime.Now;
                db.Save(queueItem);
            }
            Console.WriteLine();
        }
        db.SaveChanges();
    }

    private void SetNewCompany(IEnumerable<vw_ReformaActionQueue> _queueItems)
    {
        activeMethodName = MethodBase.GetCurrentMethod().Name;
        Console.WriteLine("{0}...", activeMethodName);
        _queueItems = Servant.FilterQueueActionsByMethodName(db, _queueItems,
activeMethodName);
        List<Reforma_ExtractData_Result> data;
        object newCompanyData;
        foreach (var queueItem in _queueItems)
        {
            activeQueueId = queueItem.Id;
            data = db.Reforma_ExtractData(null, null, queueItem.OrgId,

```

Инд. № подл.	Подп. и дата	Взаим. инд. №	Инд. № ауд.	Подп. и дата	<div style="text-align: right;">/лст</div>				
					<div style="text-align: right;">ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ</div>				
					Изм.	Лист	№ докум.	Подп.	Дата

```

activeMethodName).ToList();
    newCompanyData = new NewCompanyProfileData();
    binder.BindData(ref newCompanyData, data);
    if (reloginner.IsSessionExpired)
        reloginner.Relogin();
    try
    {
        client.SetNewCompany(queueItem.Inn, newCompanyData as NewCompanyProfileData);
    }
    catch { }
    queueItem.Updated = DateTime.Now;

    if (!isSoapFaultMessage)
    {
        queueItem.Completed = DateTime.Now;
        Servant.SaveLog(db, behavior.S SoapReply, behavior.S SoapRequest,
activeQueueId.Value, false);
    }

    isSoapFaultMessage = false;
    db.Save(queueItem);
}
db.SaveChanges();
}

private void SetHouseLinkToOrganization(IEnumerable<vw_ReformaActionQueue> _queueItems)
{
    activeMethodName = MethodBase.GetCurrentMethod().Name;
    Console.WriteLine("{0}...", activeMethodName);
    _queueItems = Servant.FilterQueueActionsByMethodName(db, _queueItems,
activeMethodName);
    vw_RateContract contract = null;
    vw_ReformaActionQueueLog log = null;

    foreach (var queueItem in _queueItems)
    {
        activeQueueId = queueItem.Id;
        contract = db.vw_RateContract.Where(i => i.TypeCode == "Manage").FirstOrDefault(i
=> i.ObjectId == queueItem.ObjectId && i.OrgId == queueItem.OrgId);
        if (contract != null)
        {
            if (reloginner.IsSessionExpired)
                reloginner.Relogin();
            try
            {
                client.SetHouseLinkToOrganization(queueItem.ObjectReformaId.Value,
queueItem.Inn, contract.DateStart, contract.DateEnd.Value);
            }
            catch { }
            if (!isSoapFaultMessage)
            {
                queueItem.Completed = DateTime.Now;
                Servant.SaveLog(db, behavior.S SoapReply, behavior.S SoapRequest,
activeQueueId.Value, false);
            }

            isSoapFaultMessage = false;
        }
        else
        {
            Servant.SaveLog(db, string.Empty, string.Empty, activeQueueId.Value, true,
_message: "Контракт на управление домом не найден");
            queueItem.Updated = DateTime.Now;
            db.Save(queueItem);
        }
        db.SaveChanges();
    }
}

```

<div>Подп. и дата</div> <div>Инд. № ауд.</div> <div>Взам. инд. №</div> <div>Подп. и дата</div> <div>Инд. № подл.</div>						<div>ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ</div> <div>Изм. /лист</div> <div>№ докум.</div> <div>Подп.</div> <div>Дата</div>	<div>/лист</div>

```

private void SetUnlinkFromOrganization(IEnumerable<vw_ReformaActionQueue> _queueItems)
{
    activeMethodName = MethodBase.GetCurrentMethod().Name;
    Console.WriteLine("{0}...", activeMethodName);
    _queueItems = Servant.FilterQueueActionsByMethodName(db, _queueItems,
activeMethodName);
    vw_RateContract contract = null;
    DateTime date = DateTime.Now;
    string reason = string.Empty;
    int reasonStatus = 0;
    bool isHandled = false;

    foreach (var queueItem in _queueItems)
    {
        activeQueueId = queueItem.Id;
        contract = db.vw_RateContract.Where(i => i.TypeCode == "Manage").FirstOrDefault(i
=> i.ObjectId == queueItem.ObjectId && i.OrgId == queueItem.OrgId);
        if (contract != null)
        {
            if (contract.DateEnd.HasValue && contract.DateEnd.Value < DateTime.Now)
            {
                reason = "Истек срок действия договора";
                date = contract.DateEnd.Value;
                reasonStatus = 1;
                isHandled = true;
            }
            if (contract.TerminationDate.HasValue && contract.TerminationDate.Value <
DateTime.Now)
            {
                reason = contract.TerminationCause;
                date = contract.TerminationDate.Value;
                reasonStatus = 1;
                isHandled = true;
            }
        }
        else
        {
            date = DateTime.Now;
            reason = string.Empty;
            reasonStatus = 2;
            isHandled = true;
        }
        if (isHandled)
        {
            if (reloginner.IsSessionExpired)
                reloginner.Relogin();
            try
            {
                client.SetUnlinkFromOrganization(queueItem.ObjectReformaId.Value, date,
reasonStatus, reason);
            }
            catch { }
            queueItem.Updated = DateTime.Now;

            if (!isSoapFaultMessage)
            {
                queueItem.Completed = DateTime.Now;
                Servant.SaveLog(db, behavior.SoapReply, behavior.SoapRequest,
activeQueueId.Value, false);
            }
            isSoapFaultMessage = false;
            isHandled = false;
            db.Save(queueItem);
        }
    }
}

```

Инд. № подл.	Подп. и дата	Взаим. инд. №	Инд. № ауд.	Подп. и дата	<div style="text-align: right;">/лист</div> <div style="text-align: center; font-size: 1.2em; font-weight: bold;">ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ</div>				
					Изм.	Лист	№ докум.	Подп.	Дата


```

        if (reloginner.IsSessionExpired)
            reloginner.Relogin();
        try
        {
            client.SetHouseProfile(queueItem.ObjectReformaId.Value, houseData as
HouseProfileData);
        }
        catch { }
        queueItem.Updated = DateTime.Now;
        if (!isSoapFaultMessage)
        {
            queueItem.Completed = DateTime.Now;
            Servant.SaveLog(db, behavior.S SoapReply, behavior.S SoapRequest,
activeQueueId.Value, false);
        }
        isSoapFaultMessage = false;
        db.Save(queueItem);
    }
    db.SaveChanges();
}
#endregion

#region Интеграция файлов
private void SetFileToHouseProfile(IEnumerable<vw_ReformaActionQueue> _queueItems)
{
    activeMethodName = MethodBase.GetCurrentMethod().Name;
    Console.WriteLine("{0}...", activeMethodName);
    _queueItems = Servant.FilterQueueActionsByMethodName(db, _queueItems,
activeMethodName);
    vw_ReformaFilesStorage uploadedDoc = null;
    FileInfo fileInfo = null;

    foreach (var queueItem in _queueItems)
    {
        activeQueueId = queueItem.Id;
        uploadedDoc = db.vw_ReformaFilesStorage.FirstOrDefault(i => i.Id ==
queueItem.FileId);
        if (uploadedDoc != null)
        {
            if (!queueItem.ObjectReformaId.HasValue)
            {
                Servant.SaveLog(db, string.Empty, string.Empty, activeQueueId.Value, true,
_message: "Объект не синхронизирован");
                continue;
            }
            if (reloginner.IsSessionExpired)
                reloginner.Relogin();
            try
            {
                fileInfo = client.SetFileToHouseProfile(queueItem.ObjectReformaId.Value,
uploadedDoc.ProfilePartId, Servant.EncodeFileContent(db, uploadedDoc));
            }
            catch { }
            queueItem.Updated = DateTime.Now;
            if (!isSoapFaultMessage)
            {
                queueItem.Completed = DateTime.Now;
                uploadedDoc.ReformaId = fileInfo.file_id;
                db.Save(uploadedDoc);
                Servant.SaveLog(db, behavior.S SoapReply, behavior.S SoapRequest,
activeQueueId.Value, false);
            }
            isSoapFaultMessage = false;
            db.Save(queueItem);
        }
    }
}

```

<div>Инд. № подл.</div> <div>Взаим. инд. №</div> <div>Инд. № аудл.</div> <div>Подп. и дата</div>	Подп. и дата					<div>Инд. № подл.</div> <div>Взаим. инд. №</div> <div>Инд. № аудл.</div> <div>Подп. и дата</div>	<div>Изм.</div> <div>Лист</div> <div>№ докум.</div> <div>Подп.</div> <div>Дата</div>	<div>ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ</div>	Лист

```

        else
            continue;
    }
    db.SaveChanges();
}

public void SetFileToCompanyProfile(IEnumerable<vw_ReformaActionQueue> _queueItems)
{
    activeMethodName = MethodBase.GetCurrentMethod().Name;
    Console.WriteLine("{0}...", activeMethodName);
    _queueItems = Servant.FilterQueueActionsByMethodName(db, _queueItems,
activeMethodName);
    vw_ReformaFilesStorage uploadedDoc = null;
    FileInfo fileInfo = null;

    foreach (var queueItem in _queueItems)
    {
        activeQueueId = queueItem.Id;
        uploadedDoc = db.vw_ReformaFilesStorage.FirstOrDefault(i => i.Id ==
queueItem.FileId);
        if (uploadedDoc != null)
        {
            if (reloginner.IsSessionExpired)
                reloginner.Relogin();
            try
            {
                fileInfo = client.SetFileToCompanyProfile(queueItem.ReportingPeriodId.Value,
queueItem.Inn, uploadedDoc.ProfilePartId, Servant.EncodeFileContent(db, uploadedDoc));
            }
            catch { }
            queueItem.Updated = DateTime.Now;
            if (!isSoapFaultMessage)
            {
                queueItem.Completed = DateTime.Now;
                uploadedDoc.ReformaId = fileInfo.file_id;
                db.Save(uploadedDoc);
                Servant.SaveLog(db, behavior.S SoapReply, behavior.S SoapRequest,
activeQueueId.Value, false);
            }
            isSoapFaultMessage = false;
            db.Save(queueItem);
        }
        else
            continue;
    }
    db.SaveChanges();
}

```

```

public void SetFileDeleted(IEnumerable<vw_ReformaActionQueue> _queueItems)
{
    activeMethodName = MethodBase.GetCurrentMethod().Name;
    Console.WriteLine("{0}...", activeMethodName);
    _queueItems = Servant.FilterQueueActionsByMethodName(db, _queueItems,
activeMethodName);
    vw_ReformaFilesStorage deletedFile = null;

    foreach (var queueItem in _queueItems)
    {
        activeQueueId = queueItem.Id;
        deletedFile = db.vw_ReformaFilesStorage.FirstOrDefault(i => i.Id ==
queueItem.FileId);

        if (deletedFile != null && deletedFile.ReformaId.HasValue)
        {
            if (reloginner.IsSessionExpired)
                reloginner.Relogin();
        }
    }
}

```

Подп. и дата	
Инд. № ауд.	
Взам. инд. №	
Подп. и дата	
Инд. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист


```

        try
        {
            client.SetFileDeleted(deletedFile.ReformaId.Value);
        }
        catch { }
        queueItem.Updated = DateTime.Now;
        if (!isSoapFaultMessage)
        {
            queueItem.Completed = DateTime.Now;
            db.Delete(deletedFile);
            Servant.SaveLog(db, behavior.S SoapReply, behavior.S SoapRequest,
activeQueueId.Value, false);
        }
        isSoapFaultMessage = false;
        db.Save(queueItem);
    }
}
db.SaveChanges();
}
#endregion

#endregion

#region Get-методы
private void GetRequestList()
{
    activeMethodName = MethodBase.GetCurrentMethod().Name;
    Console.WriteLine("{0}...", activeMethodName);
    if (reloginner.IsSessionExpired)
        reloginner.Relogin();
    RequestState[] orgRequestStates = client.GetRequestList();
    IQueryable<vw_ReformaOrganizationsRequests> requests =
db.vw_ReformaOrganizationsRequests;
    foreach (var item in orgRequestStates)
    {
        vw_ReformaOrganizationsRequests requestReplyItem = requests.FirstOrDefault(i =>
i.Inn == item.inn);
        if (requestReplyItem != null)
        {
            requestReplyItem.Updated = item.update_date.Value;
            requestReplyItem.Status = item.status;
        }
        else
        {
            requestReplyItem = new vw_ReformaOrganizationsRequests()
            {
                Inn = item.inn,
                Status = item.status,
                Created = item.create_date,
                Updated = item.update_date.HasValue ? item.update_date.Value : (DateTime?)null
            };
            if (db.vw_cmn_Organization.FirstOrDefault(i => i.Inn == requestReplyItem.Inn) !=
null)
                db.Save(requestReplyItem);
        }
        db.SaveChanges();
    }

    private void GetHouseList()
    {
        activeMethodName = MethodBase.GetCurrentMethod().Name;
        Console.WriteLine("{0}...", activeMethodName);
        var organizations = db.vw_ReformaOrganizationsRequests.Where(i => i.IsError.HasValue
&& !i.IsError.Value).Select(i => new { Inn = i.Inn, orgName = i.OrgShortName }).ToArray();
        HouseData[] housesData = null;
    }
}

```

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № подл.	Подп. и дата	<div style="text-align: right;">/лист</div>				
					<div style="text-align: right;">ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ</div>				
					Изм.	Лист	№ докум.	Подп.	Дата

```

int[] allowedObjectTypes = Servant.GetRequiredObjectTypes(db);
int OCUpdated = 0, OCAddressNotFound = 0, OCNotFound = 0;

foreach (var org in organizations)
{
    if (reloginner.IsSessionExpired)
        reloginner.Relogin();
    try
    {
        housesData = client.GetHouseList(org.Inn);
        Console.WriteLine("Готовы к проверке {0} объекта организации {1}",
housesData.Length, org.orgName);
        foreach (var dataItem in housesData)
        {
            Console.Write("{0} - {1} - {2} ", dataItem.full_address.city1_formal_name,
dataItem.full_address.street_formal_name, dataItem.full_address.house_number);
            int addressId = Servant.GetRequiredObjectId(db, dataItem);
            vw_cmn_Object1 objItem = null;
            if (addressId != -1)
                objItem = db.vw_cmn_Object1.FirstOrDefault(i => i.StructureAdrId == addressId
&& allowedObjectTypes.Contains(i.TypeId));
            else
            {
                OCAddressNotFound++;
                Console.WriteLine("- Адрес не найден в ОУ");
                continue;
            }
            if (objItem != null)
            {
                if (objItem.ReformaId.HasValue)
                {
                    //Console.WriteLine("- {0}", objItem.ReformaId.Value);
                    Console.WriteLine("");
                    continue;
                }
                objItem.ReformaId = dataItem.house_id;
                db.Save(objItem);
                OCUpdated++;
                Console.WriteLine("- Объект найден");
            }
            else
            {
                OCNotFound++;
                //Console.WriteLine(addressId);
                Console.WriteLine("- Объект не найден");
                continue;
            }
        }
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message + " " + org.Inn + " - " + org.orgName);
    }
    Console.WriteLine("{0} - обновлены, {1} - не найдены адреса в а/п, {2} - не найдены
среди объектов", OCUpdated, OCAddressNotFound, OCNotFound);
    OCUpdated = 0; OCAddressNotFound = 0; OCNotFound = 0;
    db.SaveChanges();
}

private void GetHouseInfo(bool? _updateAllObjects = false, int? _orgId = null)
{
    activeMethodName = MethodBase.GetCurrentMethod().Name;
    Console.WriteLine("{0}...", activeMethodName);
    int[] allowedObjectTypes = Servant.GetRequiredObjectTypes(db);
    List<vw_cmn_Object1> objects = new List<vw_cmn_Object1>();

    if (_updateAllObjects.HasValue)
    {

```

Подп. и дата

Инд. № ауд.

Взам. инд. №

Подп. и дата

Инд. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

```

        if (_updateAllObjects.Value)
            objects = db.vw_cmn_Object1.Where(i =>
allowedObjectTypes.Contains(i.TypeId)).OrderBy(i => i.Id).ToList();
        else
            objects = db.vw_cmn_Object1.Where(i => !i.ReformaId.HasValue &&
allowedObjectTypes.Contains(i.TypeId)).OrderBy(i => i.Id).ToList();
    }
    else if (_orgId.HasValue)
        objects = (from obj in db.vw_cmn_Object1
            join link in db.vw_ReformaObjectToOrgLink on obj.Id equals link.ObjectId
            where link.OrganizationId == _orgId.Value
            orderby obj.Id
            select obj).ToList();

    if (objects.Any())
    {
        Console.WriteLine("Представлено объектов к обновлению: {0}", objects.Count());
        int count = 0, current = 0;
        HouseInfo[] houseInfo = null;
        foreach (var objItem in objects)
        {
            object houseAddress = new FiasAddress();
            List<Reforma_ExtractData_Result> data = db.Reforma_ExtractData(objItem.Id,
objItem.StructureId, null, activeMethodName).ToList();
            binder.BindData(ref houseAddress, data);

            if (reloginner.IsSessionExpired)
                reloginner.Relogin();
            try
            {
                houseInfo = client.GetHouseInfo(houseAddress as FiasAddress);
            }
            catch { }
            if (houseInfo != null)
            {
                count++;
                Console.WriteLine("{0} = {1}/{2} - {3}", objItem.Id, current, objects.Count(),
objItem.ComposedFullName);
                objItem.ReformaId = houseInfo.FirstOrDefault(i =>
!i.drifting_date.HasValue).house_id;
                db.Save(objItem);
            }
            current++;
            if (current % 30 == 0)
                db.SaveChanges();
        }
        db.SaveChanges();
        Console.WriteLine("{0}/{1} объектов обновлено", count, objects.Count());
    }

    private void GetReportingPeriodList()
    {
        activeMethodName = MethodBase.GetCurrentMethod().Name;
        Console.WriteLine("{0}...", activeMethodName);
        if (reloginner.IsSessionExpired)
            reloginner.Relogin();
        ReportingPeriod[] periods = client.GetReportingPeriodList();
        string[] periodNames = db.vw_ReformaReportingPeriods.Select(i => i.Name).ToArray();

        foreach (var period in periods)
        {
            vw_ReformaReportingPeriods item = db.vw_ReformaReportingPeriods.FirstOrDefault(i =>
i.Name == period.name);
            if (item == null)
            {

```

Подп. и дата	
Инд. № ауд.	
Взам. инд. №	
Подп. и дата	
Инд. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Луст

Модуль привязки данных

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Globalization;
using AIS.HM.Model;
#if DEBUG
using AIS.HM.ReformaGKHService.ReformaAPI_beta;
#else
using AIS.HM.ReformaGKHService.ReformaAPI;
#endif

namespace ReformaGKHService.Classes
{
    public class DataBinder
    {
        private char nestedParametersSplitter;
        public DataBinder(char splitter)
        {
            nestedParametersSplitter = splitter;
        }

        private void SetValue(ref object obj, string[] path, int depth, string value, int? index,
            int? length)
        {
            foreach (var property in obj.GetType().GetProperties())
            {
                if (property.Name == path[depth])
                {
                    if (!property.PropertyType.IsClass || Type.GetTypeCode(property.PropertyType) !=
                        TypeCode.Object)
                    {
                        {
                            if (String.IsNullOrEmpty(value))
                            {
                                property.SetValue(obj, null, null);
                                break;
                            }
                            if (property.PropertyType.IsGenericType &&
                                property.PropertyType.GetGenericTypeDefinition() == typeof(Nullable<>))
                            {
                                if (String.IsNullOrEmpty(value))
                                    property.SetValue(obj, null, null);
                                else
                                {
                                    Type t = Nullable.GetUnderlyingType(property.PropertyType) ??
                                    property.PropertyType;
                                    if (t == typeof(Boolean))
                                    {
                                        switch (value)
                                        {
                                            case "0":
                                                property.SetValue(obj, false, null);
                                                break;
                                            case "1":
                                                property.SetValue(obj, true, null);
                                                break;
                                        }
                                    }
                                    else
                                    {
                                        property.SetValue(obj, Convert.ChangeType(value, t,
                                            CultureInfo.InvariantCulture), null);
                                    }
                                }
                            }
                            else if (property.PropertyType.IsEnum)
                                property.SetValue(obj, Enum.Parse(property.PropertyType, value), null);
                        }
                    }
                }
            }
        }
    }
}
```

Подп. и дата	
Инв. № дудл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

```

        else
            property.SetValue(obj, Convert.ChangeType(value, property.PropertyType,
CultureInfo.InvariantCulture), null);
            break;
        }
        else
        {
            depth++;
            if (property.PropertyType.IsArray)
            {
                if (length.HasValue)
                {
                    if (property.GetValue(obj, null) == null)
                    {
                        Array array =
Array.CreateInstance(property.PropertyType.GetElementType(), length.Value);
                        property.SetValue(obj, array, null);
                        for (int i = 0; i < array.Length; i++)
                            array.SetValue(Activator.CreateInstance(property.PropertyType.GetElementType()), i);
                    }
                    object arrayItem = ((property.GetValue(obj, null)) as
Array).GetValue((long)index);
                    SetValue(ref arrayItem, path, depth, value, null, null);
                }
                else
                    break;
            }
            else
            {
                if (property.GetValue(obj, null) == null)
                    property.SetValue(obj, Activator.CreateInstance(property.PropertyType),
null);
                object nestedObj = property.GetValue(obj, null);
                SetValue(ref nestedObj, path, depth, value, null, null);
            }
        }
    }
}

public void BindData(ref object obj, List<Reforma_ExtractData_Result> data)
{
    foreach (var item in data)
    {
        string[] nestedParameters = item.ParameterName.Split(nestedParametersSplitter);
        if (item.ArrayIndex != null)
        {
            int length = data.Where(i => i.ParameterName == item.ParameterName).Max(i =>
i.ArrayIndex).Value + 1;
            SetValue(ref obj, nestedParameters, 0, item.Value, item.ArrayIndex, length);
        }
        else
            SetValue(ref obj, nestedParameters, 0, item.Value, item.ArrayIndex, null);
    }
}
}

```

Модуль обеспечения бесперебойной передачи данных

```

using System;
using ReformaGKHService.Classes;
using System.Configuration;

namespace AIS.HM.ReformaGKHService.Classes
{
    public class ReloginHelper

```

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

```

{
    private APIProvider provider;
    private DateTime sessionStart;
    public bool IsSessionExpired
    {
        get
        {
            return (DateTime.Now - sessionStart).TotalSeconds >
Double.Parse(ConfigurationManager.AppSettings["session_timeout"]) -
Double.Parse(ConfigurationManager.AppSettings["session_time_shift"]);
        }
    }
    public ReloginHelper(APIProvider provider)
    {
        this.provider = provider;
    }
    public void StartTimer()
    {
        sessionStart = DateTime.Now;
    }
    public void Relogin()
    {
        provider.Logout();
        provider.Login();
        this.StartTimer();
    }
}
}

```

Модуль перехвата сообщений

```

using System;
using System.ServiceModel.Channels;
using System.ServiceModel.Description;
using System.ServiceModel.Dispatcher;

namespace ReformaGKHSservice.Classes
{
    public sealed class AuthHeaderBehavior : IEndpointBehavior, IClientMessageInspector
    {
        public string SoapReply { get; set; }
        public string SoapRequest { get; set; }
        public TokenProvider TokenProvider { get; set; }
        public APIProvider ApiProvider { get; set; }
        public AuthHeaderBehavior()
        {
            TokenProvider = new TokenProvider(string.Empty);
        }
        public void AddBindingParameters(ServiceEndpoint endpoint,
System.ServiceModel.Channels.BindingParameterCollection bindingParameters) { }
        public void ApplyDispatchBehavior(ServiceEndpoint endpoint, EndpointDispatcher
endpointDispatcher) { }
        public void Validate(ServiceEndpoint endpoint) { }
        public void ApplyClientBehavior(ServiceEndpoint endpoint, ClientRuntime clientRuntime)
        {
            clientRuntime.MessageInspectors.Add(this);
        }
        public void AfterReceiveReply(ref System.ServiceModel.Channels.Message reply, object
correlationState)
        {
            SoapReply = reply.ToString();
            if (ApiProvider != null && reply.IsFault)
            {
                APIProvider.ReplyCallback callback = null;
                callback = new APIProvider.ReplyCallback(ApiProvider.HandleFaultReply);
                callback.Invoke(reply);
            }
        }
    }
}

```

Подп. и дата

Инд. № аудл.

Взаим. инд. №

Подп. и дата

Инд. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист

```

    }
    public object BeforeSendRequest(ref System.ServiceModel.Channels.Message request,
    System.ServiceModel.IClientChannel channel)
    {
        request.Headers.Clear();
        if (!String.IsNullOrEmpty(this.TokenProvider.LogKey))
            request.Headers.Add(MessageHeader.CreateHeader("authenticate", string.Empty,
this.TokenProvider.LogKey));
        SoapRequest = request.ToString();
        return null;
    }
}
}

```

Модуль конфигурации

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Xml.Linq;

```

```

namespace AIS.HM.ReformaGKHService.Classes
{

```

```

    public static class CommonConfiguration
    {
        public static string regionName;
        public static string connectionString;
        public static string people_site_root_url;
        public static string people_site_root_local;
        public static string people_site_path;
        public static string people_site_filedir;
    }

```

```

    public static void DropSettings()
    {
        connectionString = string.Empty;
        people_site_root_url = string.Empty;
        people_site_root_local = string.Empty;
        people_site_path = string.Empty;
        people_site_filedir = string.Empty;
    }
}

```

```

public class RegionsConfigurationHandler
{

```

```

    private string configFilePath = @"RegionsConfig.xml";
    private List<string> handledRegions = new List<string>();
    public RegionsConfigurationHandler() { }
    public bool SetConfiguration()
    {
        XDocument xmlDoc = XDocument.Load(Environment.CurrentDirectory + configFilePath);
        var regions = xmlDoc.Descendants("Regions").Descendants("Region").ToList();
        if (regions.Any())
        {
            var region = regions.FirstOrDefault(i =>
!handledRegions.Contains(i.Attribute("name").Value));
            //обработали все регионы?
            if (region == null)
                return false;
            handledRegions.Add(region.Attribute("name").Value); //добавили в список уже
"обработанных"
            var items = region.Descendants();
            try
            {
                CommonConfiguration.regionName = region.Attribute("name").Value;
                CommonConfiguration.connectionString = items.First(n => n.Name ==
"connection_string")
                    .Attribute("value")
                    .Value;
            }
            catch { }
        }
    }
}

```

Подп. и дата

Инв. № дудл.

Взам. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Лист


```

        CommonConfiguration.people_site_path = items.First(n => n.Name ==
"people_site_path")
                .Attribute("value")
                .Value;
        CommonConfiguration.people_site_root_local = items.First(n => n.Name ==
"people_site_root_local")
                .Attribute("value")
                .Value;
        CommonConfiguration.people_site_root_url = items.First(n => n.Name ==
"people_site_root_url")
                .Attribute("value")
                .Value;
        CommonConfiguration.people_site_filedir = items.First(n => n.Name ==
"people_site_filedir")
                .Attribute("value")
                .Value;
    }
    catch { }
    return true;
}
return false;
}
}
}

```

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № ауд.	Подп. и дата	<div>ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ</div>					Лист
Изм.	Лист	№ докум.	Подп.	Дата						

Приложение Б

(обязательное)

Тексты SQL-процедур

Хранимая процедура [ext].[Reforma_ExtractData]

```
ALTER PROCEDURE [ext].[Reforma_ExtractData]
    @objId INT,
    @structureId INT,
    @orgId INT,
    @methodName NVARCHAR(MAX)
AS
BEGIN
    --Для привязки данных в Entity Framework
    IF 1 = 2 BEGIN
        SELECT
            CAST(NULL AS NVARCHAR(MAX)) AS ParameterName,
            CAST(NULL AS NVARCHAR(MAX)) AS Value,
            CAST(NULL AS INT) AS ArrayIndex
        WHERE 1 = 2
        END

        SET NOCOUNT ON;
        CREATE TABLE #methodParameters ([Number] INT, [ParameterName] NVARCHAR(MAX),
[ViewName] NVARCHAR(MAX),
[ViewColumnName] NVARCHAR(MAX),
[IsStructure] BIT,
[Value] NVARCHAR(MAX),
[ArrayIndex] INT, [ExternalType] NVARCHAR(MAX))

        DECLARE @tmp INT
        --Заполнение временной таблицы параметрами и их координатами в БД ОУ
        INSERT INTO #methodParameters (Number, ParameterName, ViewName, ViewColumnName,
IsStructure, Value, ArrayIndex, ExternalType)
        SELECT RANK() OVER (ORDER BY p.Id), p.ReformaName, p.ViewName, p.ViewColumnName,
p.IsStructure, NULL, NULL, p.ExternalType
        FROM ext.vw_ReformaParameters AS p
            INNER JOIN ext.ReformaAPIMethods AS m ON m.Id = p.MethodId
        WHERE m.CodeName = @methodName

        --Вызов нужной хранимки
        IF (@methodName = 'SetHouseProfile')
            EXEC ext.Reforma_SetHouseProfile @objId = @objId, @structureId = @structureId,
@orgId = @orgId, @methodName = @methodName
        IF (@methodName = 'SetNewCompany')
            EXEC ext.Reforma_SetNewCompany @orgId = @orgId, @methodName = @methodName
        IF (@methodName = 'GetHouseInfo' OR @methodName = 'SetNewHouse')
            BEGIN
                SELECT TOP 1 @tmp = s.AdrId FROM no.cmn$Structure AS s WHERE s.Id =
@structureId
                EXEC ext.Reforma_GetHouseInfo @addressId = @tmp
            END
        IF (@methodName = 'SetCompanyProfile')
            EXEC ext.Reforma_SetCompanyProfile @orgId = @orgId, @methodName = @methodName

        --result-select
        SELECT ParameterName, Value, ArrayIndex
        FROM #methodParameters
    END
```

Подп. и дата

Инд. № ауд.

Взаим. инд. №

Подп. и дата

Инд. № подл.

Лист

ВКР-УлГТУ-09.03.02-11/286-2015 ПЗ

Изм. Лист № докум. Подп. Дата

Копирован

Формат А4

