**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Факультет *ИСТ* Кафедра *ИВК а*

Специальность *информационные системы и технологии а*

УТВЕРЖДАЮ:

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г

.

ЗАДАНИЕ

**по бакалаврской работе студента**

*Желепова Алексея Сергеевича, гр. ИСТбд-41* а

(Ф.И.О., группа)

1. Тема проекта: *Подсистема интеграции с федеральной системой «Реформа ЖКХ» для «АИС: Объектовый учет»*

утверждена приказом по университету № \_\_\_ от «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

2. Срок сдачи студентом законченного проекта: «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

3. Исходные данные к проекту: *разработать модуль обмена данными для федеральной системы «Реформа ЖКХ» и «АИС: Объектовый учет».*

4. Содержание пояснительной записки (перечень подлежащих разработке вопросов): *техническое задание; разработка БД; разработка алгоритмов, математических моделей и методов; разработка ПО; проектирование комплекса технических средств; эксплуатационный раздел; тестирование системы.п*

5. Перечень графического материала (с точным указанием обязательных чертежей):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Консультанты по проекту, с указанием относящихся к ним разделов проекта: *Рыбкина М.В. – Экономический раздел, Куклев В.А. – Безопасность и экологичность\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

|  |  |  |  |
| --- | --- | --- | --- |
| Раздел | Консультант | Подпись, дата | |
| задание выдал | задание принял |
| Экономический раздел | Рыбкина М.В. |  |  |
| Безопасность и экологичность объекта проектирования | Куклев В.А. |  |  |
|  |  |  |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. Дата выдачи задания: «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Фамилия, имя, отчество

руководителя (полностью): *Кандаулов Валерий Михайлович*

Занимаемая должность: *к.т.н., доцент каф. ИВК*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_а

(подпись)

Задание принял к исполнению

(подпись)

# АННОТАЦИЯ

# СОДЕРЖАНИЕ

[ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СОКРАЩЕНИЙ И ТЕРМИНОВ 9](#_Toc420686741)

[ВВЕДЕНИЕ 10](#_Toc420686742)

[1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА СОЗДАНИЕ СИСТЕМЫ 11](#_Toc420686743)

[1.1 Назначение и цели создания системы 11](#_Toc420686744)

[1.2 Характеристика объекта автоматизации 11](#_Toc420686745)

[1.2.1 Общее описание 11](#_Toc420686746)

[1.2.2 Структура и принципы функционирования 13](#_Toc420686747)

[1.2.3 Существующая информационная система и ее недостатки 14](#_Toc420686748)

[1.2.4 Анализ аналогичных разработок 14](#_Toc420686749)

[1.2.5 Актуальность проводимой разработки 15](#_Toc420686750)

[1.3 Общие требования к системе 15](#_Toc420686751)

[1.3.1 Требования к структуре и функционированию системы 15](#_Toc420686752)

[1.3.2 Дополнительные требования 16](#_Toc420686753)

[1.4 Требования к функциям, выполняемым системой 17](#_Toc420686754)

[1.4.1 Сбор информации для интеграции по каждому из API-методов 17](#_Toc420686755)

[1.4.2 Функция привязки данных 18](#_Toc420686756)

[1.4.3 Начальная поставка данных организацией 18](#_Toc420686757)

[1.4.4 Формирование списка интеграционных запросов на поставку данных 18](#_Toc420686758)

[1.4.5 Формирование ссылок на файлы документов организации 19](#_Toc420686759)

[1.4.6 Очередь запросов 19](#_Toc420686760)

[1.4.7 Отображение списка запросов к ИС «Реформа ЖКХ» 19](#_Toc420686761)

[1.4.8 Формирование отображения истории выполнения запросов 20](#_Toc420686762)

[1.4.9 Формирование статистики 20](#_Toc420686763)

[1.5 Требования к видам обеспечения 20](#_Toc420686764)

[1.5.1 Требования к алгоритмическому обеспечению 20](#_Toc420686765)

[1.5.2 Требования к информационному обеспечению 21](#_Toc420686766)

[1.5.3 Требования к программному обеспечению 21](#_Toc420686767)

[1.5.4 Требования к техническому обеспечению 21](#_Toc420686768)

[2. МОДЕЛЬ ДАННЫХ СИСТЕМЫ 23](#_Toc420686769)

[2.1 Стандарт функционального моделирования IDEF0 23](#_Toc420686770)

[2.2 IDEF0-модель приложения интеграции 24](#_Toc420686771)

[3. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ 28](#_Toc420686772)

[3.1 Выбор технологий управления данными 28](#_Toc420686773)

[3.1.1 Выбор СУБД 28](#_Toc420686774)

[3.1.2 Утилиты и инструменты проектирования базы данных 29](#_Toc420686775)

[3.1.3 Технологии ORM 30](#_Toc420686776)

[3.1.4 Сравнение ORM-фреймворков LINQ to SQL и Entity Framework 31](#_Toc420686777)

[3.1.5 Особенности применения ORM Entity Framework 4 31](#_Toc420686778)

[3.2 Проектирование базы данных 32](#_Toc420686779)

[3.2.1 Физическая модель данных 32](#_Toc420686780)

[3.2.2 Технология SQL-представлений 35](#_Toc420686781)

[3.2.3 Техника применения deploy-процедур 36](#_Toc420686782)

[3.3 Организация сбора, передачи, обработки и выдачи информации 38](#_Toc420686783)

[3.3.1 Организация общей концепции процесса сбора данных 38](#_Toc420686784)

[3.3.2 Методики сбора данных 40](#_Toc420686785)

[3.3.3 Организация обработки информации 41](#_Toc420686786)

[3.3.4 Организация передачи информации 42](#_Toc420686787)

[3.3.5 Организация выдачи информации 43](#_Toc420686788)

[4. АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ 44](#_Toc420686789)

[5. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ 45](#_Toc420686790)

[6. ТЕСТИРОВАНИЕ СИСТЕМЫ 46](#_Toc420686791)

[7. ЭКОНОМИЧЕСКИЙ РАЗДЕЛ 47](#_Toc420686792)

[7.1 Расчет показателя трудоемкости для разработанного программного продукта 47](#_Toc420686793)

[7.2 Расчет затрат на материальные ресурсы и сырье 50](#_Toc420686794)

[7.3 Расчет затрат на оплату труда 52](#_Toc420686795)

[7.4 Расчет отчислений в социальные фонды 53](#_Toc420686796)

[7.5 Расчет амортизации оборудования 54](#_Toc420686797)

[7.6 Расчет себестоимости разработки 55](#_Toc420686798)

[7.7 Расчет плановой прибыли 55](#_Toc420686799)

[7.8 Расчет основных технико-экономических показателей и эффективности использования программного продукта 56](#_Toc420686800)

[8. БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ ПРОЕКТА 59](#_Toc420686801)

[8.1 Исходные данные 59](#_Toc420686802)

[8.2 Перечень нормативных документов 60](#_Toc420686803)

[8.3 Анализ потенциальных опасностей 61](#_Toc420686804)

[8.3.1 Анализ вредных и опасных производственных факторов 64](#_Toc420686805)

[8.3.2 Анализ воздействия на окружающую среду 71](#_Toc420686806)

[8.3.3 Анализ возможных чрезвычайных ситуаций 71](#_Toc420686807)

[8.4 Мероприятия по охране труда 73](#_Toc420686808)

[8.4.1 Мероприятия по обеспечению комфортных условий труда 74](#_Toc420686809)

[8.4.2 Мероприятия по защите от опасных и вредных производственных факторов 76](#_Toc420686810)

[8.5 Мероприятия по охране окружающей среды 76](#_Toc420686811)

[8.6 Мероприятия по защите от чрезвычайных ситуаций 77](#_Toc420686812)

[8.7 Расчетная часть 78](#_Toc420686813)

[8.7.1 Расчет уровня шума на рабочем месте 78](#_Toc420686814)

[8.7.2 Расчет величины освещенности рабочего пространства 79](#_Toc420686815)

[8.8 Оценка эффективности принятых решений 81](#_Toc420686816)

[ЗАКЛЮЧЕНИЕ 82](#_Toc420686817)

# ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СОКРАЩЕНИЙ И ТЕРМИНОВ

ИС – информационная система

АИС – автоматизированная информационная система

ЖКХ – жилищно-коммунальное хозяйство

Реформа – «Реформа ЖКХ»

ПО – программное обеспечение

УК – управляющая компания

ОУ – объектовый учет

СУРБД – система управления реляционными базами данных

ORM – object relational mapping

ID - идентификатор

# ВВЕДЕНИЕ

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА СОЗДАНИЕ СИСТЕМЫ

## Назначение и цели создания системы

Основным назначением разработки дипломного проекта является обеспечение обмена данными между федеральным порталом «Реформа ЖКХ» и региональной информационной системой АИС: Объектовый учет.

Целями создания программного продукта являются:

1. автоматизация процесса загрузки данных управляющими компаниями на сайт портала «Реформа ЖКХ»;
2. создание универсального подхода к проблеме решения интеграции, который позволит не только обеспечивать обмен данными с системой «Реформа ЖКХ», но и может быть переиспользован при налаживании взаимодействия с другими ИС.

## Характеристика объекта автоматизации

Объектом автоматизации дипломного проектирования является процесс обмена данными между домоуправляющими компаниями и информационным порталом «Реформа ЖКХ».

### 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 квалифицированном и своевременном решении коммунальных проблем управляющими компаниями.

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". 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:

1. процессов домоуправления, которые загружаются организациями;
2. всероссийского рейтинга управляющих компаний;
3. 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:

1. конструктивных элементов, состояния и параметров объекта;
2. списка домов, находящихся в управлении;
3. прикрепленных файлов нормативных актов и документов.

Федеральный проект “Реформа ЖКХ” не единственный, который решает задачи автоматизации процессов домоуправления. В настоящие 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 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. 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 два типа: методы GET, реализующие возможность получения данных с Реформы, и POST, обеспечивающие непосредственную загрузку данных. Таким образом, существует возможность реализации двустороннего канала обмена информацией.

### Структура и принципы функционирования

Каждая управляющая организация, использующая региональную систему объектового учета АИС: Объектовый учет может оставить 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 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.

Пользователь управляющей компании часто вносит какие-либо 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 модулем интеграции, который 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.

### 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 системе объектового учета.

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

### Анализ аналогичных разработок

Существуют порядка 50-ти программных решений интеграции с федеральным порталом «Реформа ЖКХ». Все они были разработаны на основе API-интерфейса, предоставляемого разработчиками федерального портала. Можно сделать вывод о том, что все аналоги и рассматриваемая подсистема интеграции для АИС: Объектовый учет в какой-то степени похожи между собой, так как они решают общую задачу и используют одинаковый набор API-функций.

Возможно[[1]](#footnote-1), основными различиями аналогов разрабатываемого модуля интеграции для АИС: Объектовый учет являются:

1. подход к сбору информации для интеграции из источников данных;
2. способ формирования объектов для обмена данными посредством API-интерфейса;
3. независимость подсистемы интеграции от проекта регионального объектового учета;
4. другие особенности архитектуры приложения.

### Актуальность проводимой разработки

Согласно постановлению Правительства Российской федерации № 731 "Об утверждении стандарта раскрытия информации организациями, осуществляющими деятельность в сфере управления многоквартирными домами" домоуправляющие компании обязаны раскрывать информацию о реальном состоянии жилищно-коммунального хозяйства на сайте федерального портала «Реформа ЖКХ».

Модуль интеграции с порталом для региональной информационной системы АИС: Объектовый учет должен решать установленную в постановлении Правительства РФ задачу.

## Общие требования к системе

### Требования к структуре и функционированию системы

Приложение интеграции должно быть создано с использованием модульного подхода к разработке программного обеспечения. Его применение объясняется необходимостью дальнейшей интеграции региональной системы АИС: Объектовый учет с другими информационными решениями[[2]](#footnote-2) сферы ЖКХ помимо Реформы. Основное преимущество модульного похода заключается в переиспользовании компонентов, что даст возможность построить систему интеграции из готовых программных пакетов и библиотек.

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

1. модуль сбора данных, осуществляющий поиск и представление в табличном виде информации согласно необходимому набору полей, которые описаны в документации по интеграции с порталом;
2. модуль привязки к объекту типа, позволяющей на основании данных, полученных в табличном виде, формировать объект API-класса[[3]](#footnote-3) с любым уровнем вложенностей;
3. модуль для работы с API-интерфейсом, использующийся для непосредственной поставки данных;
4. раздел управления интеграцией для личного кабинета пользователя управляющей организации;
5. 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»;
6. модуль интеграции файлов, подразумевающий разработку хранилища данных, в котором будут находиться ссылки на файлы документов, хранящиеся в базе данных региональной и федеральной ИС.

### Дополнительные требования

Дополнительным требованием к разработке приложения интеграции должна являеться его независимость от основного проекта объектового учета. Программное решение должно только обеспечивать обмен данными и никаким образом не влиять на работу информационной системы АИС: Объектовый учет.

Это требование должно быть реализовано применением технологии создания службы .NET. В этом случае принцип работы интеграционного приложения будет сведен к следующему алгоритму действий:

1. запуск приложения службы;
2. анализ действий пользователя за определенный промежуток времени;
3. сбор необходимой информации для интеграции;
4. преобразование собранных данных в API-объекты и осуществление интеграционных запросов;
5. завершение работы службы.

Другим дополнительным требованием является добавление возможности функционирования приложения в рамках одной сессии запуска с несколькими базами данных.

## Требования к функциям, выполняемым системой

### Сбор информации для интеграции по каждому из API-методов

Функция должна осуществлять сбор данных для последующего формирования объекта[[4]](#footnote-4) на основании свойств API-класса.

Приоритет выполнения задачи: высокий.

Требования к входным данным: управляющая организация, объект домоуправления, наименование функции API-интерфейса.

Требования к выходным данным: совокупность данных должна быть представлена в виде таблицы. Ее полями являются название поля[[5]](#footnote-5), значение и возможный идентификатор элемента коллекции, если тип данных – массив объектов.

### Функция привязки данных

Функция должна выполнять сборку объекта API-класса из предоставленных данных.

Приоритет выполнения: высокий.

Требования к входным данным: совокупность данных в виде таблицы, пустой объект определенного API-класса.

Требования к выходным данным: сформированный объект указанного API-класса.

### Начальная поставка данных организацией

Функция должна выполняться при подаче пользователем управляющей компании заявки на раскрытие своих данных.

Приоритет выполнения задачи: высокий.

Требования к входным данным: управляющая организация.

Требования к выходным данным: выходные данные представляют собой записи в таблице, обозначающие, какие запросы необходимо произвести, чтобы полностью осуществить выгрузку данных организации в ИС «Реформа ЖКХ».

### Формирование списка интеграционных запросов на поставку данных

Функция должна производить мониторинг действий пользователя управляющей компании на сайте приложения АИС: Объектовый учет за указанный временной интервал и формирует список необходимых запросов на синхронизацию данных.

Приоритет выполнения задачи: высокий.

Требования к входным данным: управляющая организация, временной интервал.

Требования к выходным данным: создание записей в таблице запросов на выгрузку/синхронизацию данных с ИС «Реформа ЖКХ».

### 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.

### Очередь запросов

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

### Отображение списка запросов к ИС «Реформа ЖКХ»

Должна быть разработана панель отображения процесса интеграции. Страница должна находиться в личном кабинете пользователя управляющей компании, должна обеспечивать отображение списка запросов организации к ИС «Реформа ЖКХ».

Приоритет выполнения: средний.

Требование к входным данным: управляющая организация.

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

### Формирование отображения истории выполнения запросов

Панель управления интеграцией в личном кабинете пользователя управляющей компании должна обеспечивать отображение истории выполнения интеграционных запросов к ИС «Реформа ЖКХ».

Приоритет выполнения: средний.

Требование к входным данным: управляющая организация.

Требование к выходным данным: отображение списка ответных сообщений от ИС «Реформа ЖКХ» на web-странице.

### Формирование статистики

Панель управления интеграцией в личном кабинете пользователя управляющей компании должна обеспечивать отображение статистики интеграционного процесса.

Приоритет выполнения: низкий.

Требование к входным данным: управляющая организация, таблица запросов организации, таблица истории выполнения запросов.

Требование к выходным данным: отображение графиков синхронизации объектов домоуправления, активность организации и др.

## Требования к видам обеспечения

### Требования к алгоритмическому обеспечению

Для корректной работы приложения интеграции должны быть разработаны алгоритмы, которые:

1. обеспечивает сборку объектов API-классов на основании временной таблицы;
2. осуществляет запуск интеграционных запросов организации;
3. осуществляет сбор и отображение статистики интеграции для организации;
4. осуществляет поиск «быстрого» выделения множества записей, описывающих действия пользователей.

### 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 данных об интегрируемых полях и методах API-интерфейса. Эта информация должна заполняться согласно документации API ИС «Реформа ЖКХ». Процесс развертки должен быть организован с использованием deploy-процедур[[6]](#footnote-6).

АИС: Объектовый учет использует СУБД Microsoft SQL Server, поэтому необходимо использовать указанную технологию. Для получения данных из источника должен быть использован Transact-SQL.

### Требования к программному обеспечению

Создание приложения интеграции должно производиться инструментальными средствами среды Visual Studio и .NET Framework версии 4.0. Служба обмена данными должна функционировать на компьютере с операционной системой Windows Server 2008 R2 Standard.

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, которая разрабатывается в соответствии с шаблоном проектирования Model-View-Controller.

### Требования к техническому обеспечению

Приложение интеграции региональной и федеральной систем объектового учета должно функционировать на компьютере-сервере. Его технические характеристики соответствуют следующим параметрам:

1. процессор – Intel(R) Xeon(R) CPU E5-2680 с частотой 2.70GHz;
2. установленная память (ОЗУ) – 3,08 Гб;
3. тип системы – 64-разрядная операционная система;
4. жесткий диск – объемом 1 Тб с интерфейсом обмена данными SATA;
5. сетевой адаптер пропускной способностью 1 Гбит/с.

Требования аппаратной части позволяют хранить большие объемы данных и производить быстрое обращение к ним, что способствует ускорению процесс обмена данными между информационными системами.

# МОДЕЛЬ ДАННЫХ СИСТЕМЫ

## Стандарт функционального моделирования IDEF0

Одним из начальных этапов разработки любого программного продукта является построение функциональной модели будущего прототипа системы. Ее создание позволяет избежать многих ошибок проектирования на ранних этапах, повысить качество продукта и снизить временные сроки на его реализацию. Для построения функциональных диаграмм бизнес-процессов приложения принято использовать стандарт функционального моделирования IDEF0.

Методология IDEF0 применяется в различных сферах как эффективное средство проектирования и анализа. Основной структурной единицей модели IDEF0 является диаграмма, которая представляет графическое описание рассматриваемой предметной области. Каждая диаграмма подразделяется на взаимосвязанные компоненты – блоки.

Блоки IDEF0-диаграмм отражают задачи/процессы, в результате выполнения которых можно получить определенные результаты. Блок IDEF0-диаграммы отображается в виде прямоугольника, каждая сторона которого обозначает определенное назначение. На рисунке 2.1 представлен пример простейшей IDEF0-диаграммы, состоящей из единственного логического блока.

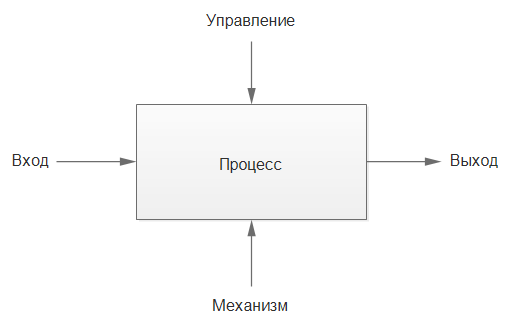


Рисунок 2.1. Простейшая IDEF0-диаграмма

Из рисунка видно, что стороны блока заключают в себе связи следующих типов:

1. вход (левая сторона блока) – описывает входные данные или информацию, использующиеся при выполнении бизнес-процесса для получения определенного результата;
2. управление (верхняя сторона блока) – описывает условия, требования, правила или стандарты, влияющие на выполнение функции;
3. механизм (нижняя сторона блока) – описывает ресурсы, с помощью которых выполняется работа. Например, персонал, денежные средства и т.д.;
4. выход (правая сторона блока) – описывает результат выполнения функции.

Модель информационной системы в нотации стандарта IDEF0 представляется совокупностью иерархически взаимосвязанных диаграмм двух типов: контекстной и декомпозиции.

Контекстная диаграмма располагается в верхней части иерархии модели и представляет собой общее описание системы и ее взаимодействия с внешней средой.

Диаграмма декомпозиции описывает отдельный фрагмент системы. Фрагментом может быть как крупный модуль проекта, так и отдельно взятая его функция. Описание архитектурных частей приложения может быть представлено несколькими диаграммами декомпозиции, которые логически взаимосвязаны между собой.

## IDEF0-модель приложения интеграции

Для построения IDEF0-диаграмм модели подсистемы интеграции федеральной и региональной ИС объектового учета было выбрана программа Edraw Max. Данное программное обеспечение поддерживает построение схем различных стандартов моделирования, в том числе и IDEF0.

На рисунке 2.2 представлена диаграмма функциональной модели разрабатываемой в рамках дипломного проектирования подсистемы.

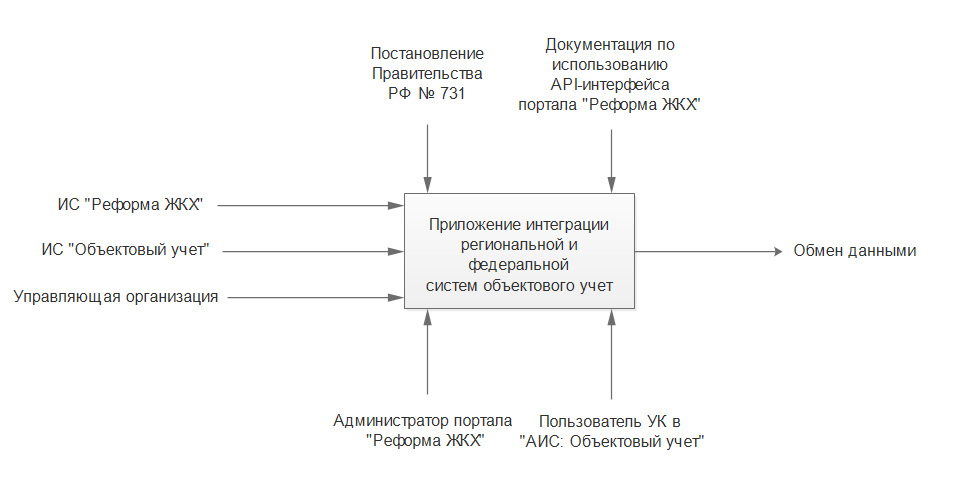


Рисунок 2.2. IDEF0-диаграмма функциональной модели

Из представленной схемы видно, что обязательными условиями работы приложения интеграции является наличие систем АИС: Объектовый учет и «Реформа ЖКХ», а также управляющей компании, заинтересованной в раскрытии данных процессов домоуправления. Среди входных требований выделяются два документа – это постановление Правительства РФ № 731, обязывающие домоуправляющие организации загружать данные на портал «Реформа ЖКХ», и документация по использованию API-интерфейса, предлагаемого разработчиками федеральной ИС, для взаимодействия с внешними системами.

Пользователи управляющих компаний в системе «АИС: Объектовый учет» осуществляют управление процессом интеграции, а администратор ИС «Реформа ЖКХ» следит за корректностью поставляемых данных.

На рисунке 2.3 представлена декомпозиция основных стадий поставки данных управляющей компании через подсистему интеграции.



Рисунок 2.3. Стадии раскрытия данных УК на портале Реформы

Как видно из рисунка, на первом этапе управляющая компания подает заявку на поставку данных. В случае ее одобрения администратором Реформы производится загрузка данных процессов домоуправления компании, другими словами ее «инициализация» на портале. Затем, по мере обновления/загрузки новых данных пользователем УК на сайт региональной системы ОУ производится раскрытие и этой информации.

На рисунке 2.4 изображен процесс отправки данных с помощью API-метода на сайт федерального портала.

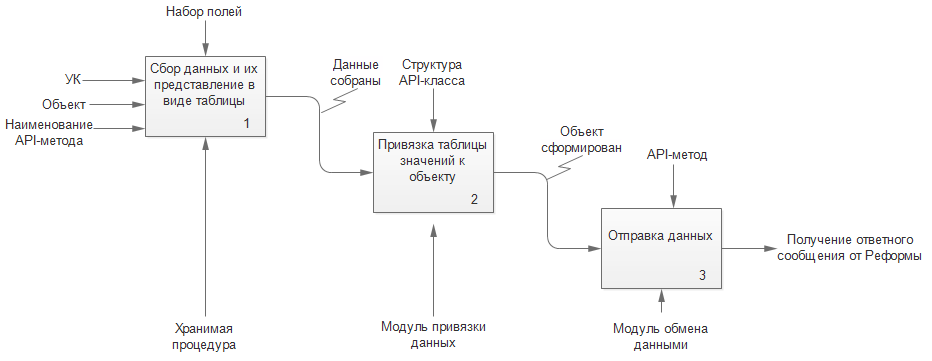


Рисунок 2.4. Диаграмма процесса поставки данных порталу «Реформа ЖКХ»

Перед выполнением API-метода на отправку информации Реформе, необходимо произвести сбор данных из источника. Это делается с помощью вызова специальной хранимой SQL-процедуры, входными параметрами которой являются идентификаторы организации и объекта, а также наименование API-метода. При успешном выполнении процедуры происходит формирование таблицы данных, которая затем трансформируется в объект API-класса с помощью модуля привязки. Если объект успешно собран, то происходит вызов API-метода и производится отправка данных. Результатом выполнения является ответное сообщение от портала Реформы, в котором указан статус выполненного запроса.

На рисунке 2.5 изображена диаграмма, описывающая процесса отправки xml-сообщения по протоколу передачи данных SOAP.

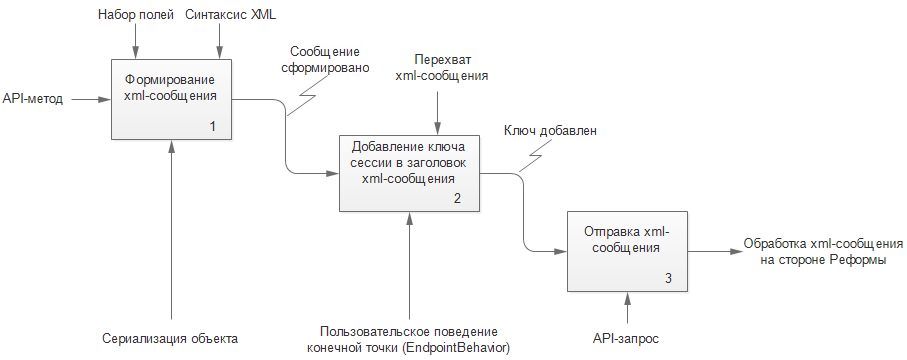


Рисунок 2.5. Диаграмма процесса отправки xml-сообщения Реформе

Согласно представленной диаграмме, процесс отправки Реформе xml-сообщения начинается с его формирования. Оно достигается путем сериализации объекта данных API-класса. После этого отправляемое сообщение вынужденно[[7]](#footnote-7) перехватывается для установки ключа сессии клиента. И только затем совершается отправка сообщения, результатом которой является его обработка на стороне сервера портала «Реформа ЖКХ».

# ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

## Выбор технологий управления данными

### Выбор СУБД

В качестве технологии управления данными была выбрана СУРБД Microsoft SQL Server (MSS). Причина такого выбора объясняется широким использованием технологии при проектировании и разработки баз данных для программных продуктов компании АИС Город.

Выбранная технология имеет как ряд преимуществ, так и недостатков. К положительным сторонам можно отнести:

1. существование многофункциональной утилиты SQL Server Management Studio для администрирования, управления и конфигурирования всех компонентов MSS;
2. MSS – один из наиболее популярных продуктов компании Microsoft, поэтому в интернете помимо сайта официальной документации также существует множество форумов сообществ, посещение которых часто помогает разрешать нетривиальные вопросы, возникающие в процессе разработки;
3. язык доступа к данным Transact-SQL, который является процедурным расширением языка SQL;
4. бесплатная версия MSS обладает теми же возможностями и функционалом, что и профессиональная. Отличие заключается только в ограничении максимально возможного объема данных, равного 10-ти Гб, для свободной версии ПО.

Основными недостатками рассматриваемой технологии являются:

1. для эффективного использования MSS нужно применять другие технологии программирования от фирмы Microsoft. Хотя в данном случае это скорее положительная сторона, так как проект объектового учета использует объектно-ориентированные технологии доступа к данным ADO.NET Entity Framework и LINQ to SQL;
2. высокая стоимость полной версии MSS.

### Утилиты и инструменты проектирования базы данных

Для управления проектом базы данных использовался программный пакет SQL Server Management Studio. Утилита представляет собой графическую среду, которая позволяет работать с объектами и настройками сервера базы данных. С Management Studio могут быть установлены расширения, значительно упрощающие процесс создания архитектуры базы данных и написания SQL-скриптов на выборку записей из таблиц. Примерами таких дополнений являются продукты компании Redgate Software:

1. SQL Data Compare, обеспечивающий автозаполнение при написании кодов SQL-запросов;
2. SQL Multi Script, позволяющий выполнять SQL-скрипт на нескольких выбранных базах данных одновременно.

Дипломное проектирование осуществлялось на предприятии, где применяются пакеты коллективной разработки SVN и GIT. Этот фактор создает дополнительную сложность в проектировании архитектуры источника данных проекта. Она заключается в том, что у каждого разработчика есть своя база, куда вносятся изменения. После этого программист должен сравнить изменения, установленные другими разработчиками, и разрешить возникшие конфликты. Этот процесс автоматизируется с применением программного пакета SQL Compare. Он позволяет сравнивать структуры нескольких баз данных и формировать искомую схему, в которой учитываются все новые изменения.

Проект подсистемы интеграции обеспечивает обмен данными между ОУ и Реформой для нескольких регионов, каждый из которых представлен собственной базой данных. При этом записи таблиц-справочников должны быть идентичными. Для автоматизации проверки эквивалентности этих данных применяется программный продукт SQL Data Compare также производства компании Redgate Software.

### Технологии ORM

Object Relational Mapping (объектно-реляционное отображение) – это технология, позволяющая связать базу данных с концепцией объектно-ориентированного подхода в программировании. Основное назначение технологии ORM – это упрощение процессов управления (обновление, удаление или создание) объектами в реляционной базе данных. ORM самостоятельно занимается преобразованием данных между двумя несовместимыми состояниями: объектом в программном коде и записью в источнике данных.

Принцип работы большинства ORM-систем основан на метаданных базы данных и объектов. Таким образом, достигается полная независимость компонентов: объекты ничего «не знают» о структуре источника данных, а база данных – ничего о том, как эти объекты организованы и используются в программе.

Применение технологии ORM освобождает разработчика от необходимости работы с SQL-запросами и написания большого количества кода доступа к объектам в базе данных. Генерируемый ORM код для этих целей уже проверен и, как правило, хорошо оптимизирован, что, несомненно, является плюсом применения технологии. Однако основным недостатком использования ORM является потеря производительности приложения. Это объясняется тем, что ORM-фреймворк поддерживает обработку широкого спектра сценариев управления данными, как правило, гораздо большего, чем предусмотрено в разрабатываемом приложении.

Вопрос о применение ORM-технологий обычно затрагивается только при разработке крупных и высоконагруженных проектов, потому что приходится выбирать между удобством и производительностью.

Подсистема интеграции двух систем объектового учета не является высоконагруженным приложением, поэтому выбор пал на использование ORM технологии.

### Сравнение ORM-фреймворков LINQ to SQL и Entity Framework

Технологии от фирмы Microsoft LINQ to SQL (LtS) и Entity Framework (EF) предназначены для решения проблем объектно-реляционного отображения. Обе ORM-системы имеют свои особенности, сравнение которых приведено в таблице 3.1.

**Таблица 3.1.** Сравнение возможностей ORM LINQ to SQL и Entity Framework

|  |  |  |
| --- | --- | --- |
| **Критерий** | **LINQ to SQL** | **Entity Framework** |
| Взаимодействие с СУРБД | Может взаимодействовать только с СУРБД Microsoft SQL Server. | Может взаимодействовать с различными СУРБД такими, как Oracle, DB2, MYSQL, Microsoft SQL Server и др. |
| Типы данных | Не поддерживает сложные типы данных. | Поддерживает сложные типы данных. |
| Подход к созданию модели данных | Не использует методологию Model First. | Поддерживает методологию Model First. |
| Отношения | Позволяет только отображать связь типа «один к одному» между классами сущностей и реляционными таблицами/представлениями. | Позволяет отображать связи типов «один к одному», «один ко многим», «многие ко многим» между классами сущностей и реляционными таблицами/представлениями. |
| Обеспечение доступа к данным | Базовый класс проводника доступа к данным – DataContext. | Классы проводника доступа к данным – EntitySQL, ObjectContext, DbContext. |

Проведя сравнительный анализ ORM, выбор был сделан в пользу EF. Это решение объясняется более широким спектром возможностей при использовании данной ORM-технологии и тем фактом, что уже существующий к тому моменту проект модели данных функционировал на основе EF четвертой версии.

### Особенности применения ORM Entity Framework 4

При разработке подсистемы интеграции возникла необходимость вызова в коде приложения хранимых процедур, написанных на языке Transact-SQL. Однако при попытке добавления процедуры в диаграмму ORM EF 4 по необъяснимым причинам не создавался тип объекта данных, возвращаемых процедурой.

Решение проблемы заключается в добавлении в реализацию хранимой процедуры «недосягаемой» части кода, в которой должны описываться метаданные возвращаемой временной таблицы. При проведении процесса сопоставления источника данных и ORM-схемы, EF формирует на основе недоступной части кода процедуры объект сложного типа. В листинге 3.2 приведен пример разобранного случая.

Листинг 3.2 Формирование сложного типа для EF на основе метаданных

--Для привязки данных в Entity Framework (недосягаемая часть кода)

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

--Создание временной таблицы

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))

--Результирующая выборка, используемая в коде приложения

SELECT ParameterName, Value, ArrayIndex

FROM #methodParameters

## Проектирование базы данных

### Физическая модель данных

Физическая модель данных приложения интеграции показана на рисунке 3.1.

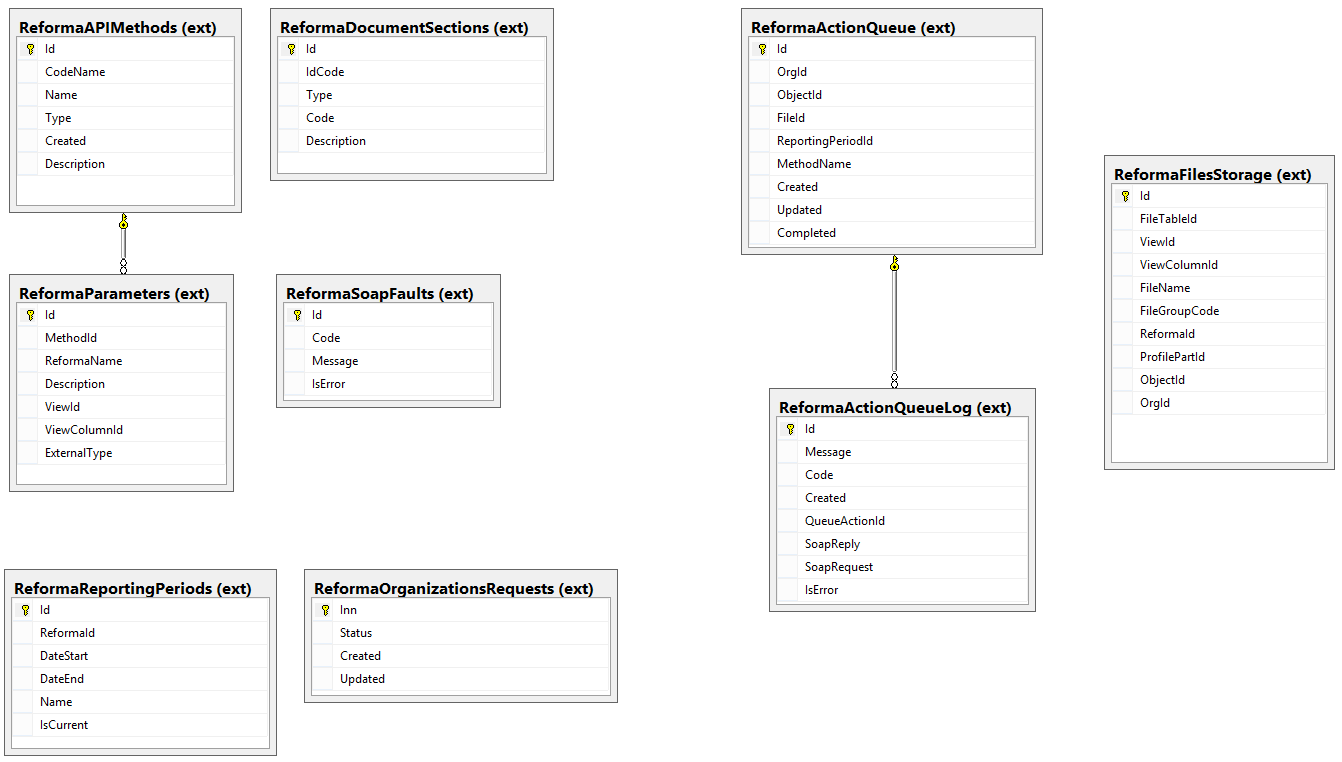


Рисунок 3.1 Физическая модель базы данных приложения интеграции

Представленную структуру базы данных можно разделить на четыре основные части:

1. таблицы-справочники, заполняемые на основании документации по интеграции с Реформой;
2. таблицы-справочники, обновление которых происходит автоматически с помощью API-методов;
3. таблицы, реализующие модель «Очередь запросов»[[8]](#footnote-8);
4. служебная таблица, предназначенная для сопоставления хранящихся файлов в базах данных интегрируемых информационных систем.

Необходимость хранения информации, представленной в документации об API-интерфейсе, объясняется тесной взаимосвязью служебных и таблиц, непосредственно осуществляющих хранение данных о процессе интеграции. Сведения документации хранятся в специальных таблицах-справочниках:

1. ext.ReformaAPIMethods содержит описания методов API-интерфейса из документации (см. таблицу);
2. ext.ReformaParameters содержит описания параметров, которые загружаются на сайт портала, для каждого из API-методов;
3. ext.ReformaDocumentSections обозначает части профиля анкет по дому или организации, в которые загружаются соответствующие файлы актов и других документов управляющими компаниями;
4. ext.ReformaSoapFault содержит информацию о кодах ошибок, которые могут возникнуть при интеграционном обмене.

Информация некоторых справочников периодически обновляется разработчиками портала «Реформа ЖКХ». Для своевременной синхронизации этих данных были созданы таблицы, которые перезаписываются каждый раз при запуске приложения интеграции. Этими таблицами являются:

1. ext.ReformaReportingPeriods содержит информацию об отчетных периодах управляющих организаций;
2. ext.ReformaOrganizationRequests содержит информацию о статусах заявок от организаций на раскрытие данных по домоуправлению.

Для стандартизации обмена данными между информационными системами была разработана модель «Очередь запросов» (подробно рассмотрена в пункте n части «Алгоритмическое обеспечение»). Данные, которыми оперирует алгоритм модели, хранятся в таблицах ext.ReformaActionQueue, предназначенной для хранения запросов, и ext.ReformaActionQueueLog, для сохранения сообщений-статусов выполнения API-методов. Таблицы объединены связью типа «один-ко-многим», что позволяет неоднократно выполнять конкретный запрос и фиксировать историю его вызова.

Организация возможности интеграции файлов реализована с помощью служебной таблицы ext.ReformaFilesStorage. Она агрегирует необходимые файлы документов и актов, а также обеспечивает их синхронизацию с данными о файлах, хранящихся на сайте портала Реформы.

### Технология SQL-представлений

Информационная система «АИС: Объектовый учет» представляет собой сложное приложение, состоящее из большого количества таблиц. Для примера, паспорт объекта, откуда осуществляется выборка данных для интеграции, состоит из порядка 150-ти взаимосвязанных между собой таблиц. Часто возникает необходимость получения данных одной таблицы через значения другой. В этой ситуации на помощь приходит механизм SQL-представлений, представляющих собой виртуальные таблицы, динамически вычисляемые на основании данных реальных таблиц.

В ходе проектирования базы данных дипломного проекта эта методика нашла широкое применение. Например, представление, описывающее список домов в управлении организации собирается при помощи SQL-скрипта из нескольких таблиц: организация, объект, договор управления домом.

Технология SQL-представлений позволяет более конкретизировать таблицы. Это достигается путем надстройки базовой структуры таблицы дополнительными полями. С использованием этой возможности была более конкретизирована таблица ext.ReformaActionQueue. Сравнение полей таблицы и соответствующего ему представления представлены в таблице 3.2.

**Таблица 3.2.** Сравнение полей таблицы ext.ReformaActionQueue и ее представления.

|  |  |  |
| --- | --- | --- |
| **Базовые поля таблицы ext.ReformaActionQueue** | **Представление ext.vw\_ReformaActionQueueLog** | **Описание полей** |
| Id | Id | ID сущности |
| OrgId | OrgId | ID организации |
| ObjectId | ObjectId | ID объекта управления |
| FileId | FileId | ID прикрепляемого файла |
| ReportingPeriodId | ReportingPeriodId | ID отчетного периода |
| MethodName | MethodName | Наименование метода |
| Created | Created | Дата создания запроса |
| Updated | Updated | Дата выполнения запроса |
| Completed | Completed | Дата последнего успешного выполнения запроса |
| - | Inn | ИНН организации |
| - | StructureId | ID структуры объекта |
| - | ObjectReformaId | ID объекта в базе данных «Реформа ЖКХ». Используется для синхронизации. |
| - | FileReformaId | ID файла в базе данных «Реформа ЖКХ». Используется для синхронизации. |
| - | MethodShortDescription | Краткое описание исполняемого API-метода |

Как правило, представления строятся с помощью применения специальных SQL-операторов объединения множеств данных. В листинге 3.3 приведен пример кода, осуществляющего формирование представления ext.vw\_ReformaActionQueue.

**Листинг 3.3.** Формирование SQL-представления ext.vw\_ReformaActionQueue.

CREATE VIEW [ext].[vw\_ReformaActionQueue]

AS

SELECT raq.Id,

raq.OrgId,

raq.ObjectId,

raq.Created,

raq.FileId,

raq.ReportingPeriodId,

raq.MethodName,

raq.Updated,

raq.Completed,

org.Inn,

obj.StructureId,

obj.ReformaId AS ObjectReformaId,

rfs.ReformaId AS FileReformaID,

meth.Name AS MethodShortDescription

FROM ext.ReformaActionQueue AS raq

INNER JOIN no.cmn$Organization AS org ON org.Id = raq.OrgId

LEFT OUTER JOIN no.cmn$Object AS obj ON obj.Id = raq.ObjectId

LEFT OUTER JOIN ext.vw\_ReformaFilesStorage AS rfs ON rfs.Id = raq.FileId

INNER JOIN ext.ReformaAPIMethods AS meth ON meth.CodeName = raq.MethodNam

### Техника применения deploy-процедур

Одним из требований к разработке модуля интеграции была возможность его одновременного функционирования с несколькими базами данных. С точки зрения подхода к разработке архитектуры источника данных это означает, что все справочные и другие статические данные должны быть идентичными для соответствующих таблиц разных баз данных.

Техника применения так называемых deploy-процедур (от англ. deploy – разворачивать, развертывать) позволяет решить эту проблему. Процедура представляет собой SQL-скрипт, выполняющийся каждый раз при обновлении структуры базы данных и заполняющий соответствующие таблицы данными.

Алгоритм такого заполнения состоит из этапов:

1. создания временной таблице, структура которой совпадает со структурой таблицы-справочника;
2. сохранения развертываемых данных во временную таблицу;
3. произведения анализа с помощью операций сравнения множеств в SQL, на основании которого некоторые данные из таблиц справочников подлежат удалению, обновлению или производится их запись из временной таблицы;
4. удаления временной таблицы.

В листинге 3.4 показаны основные шаги упомянутого алгоритма при его практическом применении.

**Листинг 3.4.** Deploy-процедура для справочника ext.ReformaDocumentSections

-- ШАГ 1. Создание временной таблицы

CREATE TABLE [#deploy\_ReformaDocumentSections]([IdCode] INT, [Type] NVARCHAR(MAX), [Code] NVARCHAR(MAX), [Description] NVARCHAR(MAX))

-- ШАГ 2. Заполнение данных во временную таблицу  
-- ...

INSERT INTO #deploy\_ReformaDocumentSections (IdCode, Type, Code, Description)

SELECT 2, 'CompanyProfile', 'AdditionalInformation', 'Дополнительная информация'

INSERT INTO #deploy\_ReformaDocumentSections (IdCode, Type, Code, Description)

SELECT 3, 'CompanyProfile', 'AnnualAccountReporting', 'Годовая бухгалтерская отчетность'

-- ...

-- ШАГ 3. Обновление/удаление/добавление таблицы-справочника, на основании данных временной таблицы

-- Обновление

UPDATE x

SET IdCode = d.IdCode, Type = d.Type, Code = d.Code, Description = d.Description

FROM [#deploy\_ReformaDocumentSections] AS d

INNER JOIN [ext].[ReformaDocumentSections] AS x ON x.IdCode = d.IdCode AND x.Type = d.Type AND x.Code = d.Code

--Удаление

DELETE x

FROM [ext].[ReformaDocumentSections] AS x

LEFT JOIN [#deploy\_ReformaDocumentSections] AS d ON x.IdCode = d.IdCode AND x.Type = d.Type AND x.Code = d.Code

WHERE d.IdCode IS NULL AND d.Code IS NULL AND d.Type IS NULL

--Добавление

INSERT INTO [ext].[ReformaDocumentSections] ( [IdCode], [Type], [Code], [Description])

SELECT IdCode = d.IdCode, Type = d.Type, Code = d.Code, Description = d.Description

FROM [#deploy\_ReformaDocumentSections] AS d

LEFT JOIN [ext].[ReformaDocumentSections] AS x ON x.IdCode = d.IdCode AND x.Type = d.Type AND x.Code = d.Code

WHERE x.IdCode IS NULL AND x.Code IS NULL AND x.Type IS NULL

-- ШАГ 4. Удаление временной таблицы

DROP TABLE [#deploy\_ReformaDocumentSections]

## Организация сбора, передачи, обработки и выдачи информации

### Организация общей концепции процесса сбора данных

Сбор данных – это ключевой момент при проектировании приложения интеграции информационных систем. Обмен информацией в подсистеме построен на основе применения паттерна «Очередь запросов». Идея этого шаблона проектирования заключается в том, что:

1. приложение интеграции представляет собой независимый продукт от интегрируемых информационных систем;
2. интегрируемые информационные системы классифицируются как «экспортер» и «импортер» данных. В конкретном случае интеграции системой-экспортером является АИС: Объектовый учет, а системой-импортером – федеральный портал «Реформа ЖКХ»;
3. обмен данными производится посредством API-интерфейса, разработанного «импортером»;
4. «экспортер» обеспечивает хранение данных о запросах на обмен данными, поступающих от пользователей системы;
5. система-экспортер обеспечивает процесс сбора данных в удобной для «импортера» форме;
6. обмен данными происходит не в «поступательном» режиме, а в определенный момент времени;
7. в момент обмена данными система-импортер должна реагировать, присылая ответные сообщения, сохранение которых необходимо обеспечить.

«Поступательный» режим обмена информацией подразумевает немедленную отправку новых данных после того, как пользователь внес изменения. Такая концепция требует того, чтобы приложение было постоянно в активном состоянии и осуществляло мониторинг очереди запросов на интеграцию. Недостатком такого подхода является то, что приложение постоянно занимает определенный ресурс памяти на сервере.

Подсистема интеграции реализована в виде службы, время и другие параметры запуска которой можно легко настроить в диспетчере приложений на сервере. Следовательно, служба и стратегия сохранения отложенных запросов пользователей на раскрытие их данных позволяет решить проблему загруженности сервера.

Согласно требованиям, указанным в техническом задании, при разработке подсистемы интеграции необходимо учитывать отдельный модуль сбора данных. Эта часть программы осуществляет поиск и подбор информации для ее последующей интеграции на сайт федерального портала. Особенностью этой части приложения является ее полная реализация на стороне базы данных при помощи SQL-скриптов, представляющих собой взаимосвязанную группу хранимых процедур (см. рисунок 3.5).



Рисунок 3.5. Иерархия вызова хранимых процедур для сбора данных

Согласно представленной схеме вызова, в первую очередь осуществляется выполнение «корневой» хранимой процедуры ext.Reforma\_ExtractData. Основная ее задача заключается в определении полей данных для интеграции, сложного типа для ORM-фреймворка[[9]](#footnote-9) и в возвращении временной таблицы с необходимыми данными. Входные данные рассматриваемой хранимой процедуры рассмотрены в таблице 3.6.

**Таблица 3.6.** Входные данные хранимой процедуры ext.Reforma\_ExtractData

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| @objId | INT | ID объекта управления |
| @structureId | INT | ID структуры объекта управления |
| @orgId | INT | ID организации, по инициативе которой делается запрос на обновление данных |
| @methodName | NVARCHAR(MAX) | Наименование API-запроса |

Следует отметить, что в соответствии со значением аргумента @methodName определяется вызов следующей в иерархии хранимой процедуры, в которой уже непосредственно ведется выборка нужных полей данных (см. листинг 3.7).

**Листинг 3.7** Вызов нужной процедуры выборки данных

-- ...

IF (@methodName = 'SetHouseProfile')

EXEC ext.Reforma\_SetHouseProfile @objId = @objId, @structureId = @structureId, @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

-- ...

### Методики сбора данных

Выборка полей данных для интеграции производится в хранимых процедурах, обозначенных ниже по иерархии относительно ext.Reforma\_ExtractData (см. рисунок 3.5). В процессе дипломного проектирования было разработано две методики получения информации: с применением динамического SQL и с использованием обычных запросов.

Для применения первой методики был реализован механизм поддержки метаданных. Он представляет собой две связанные таблицы, хранящие информацию об SQL-представлениях и полях данных содержащихся в них. На основании этих сведений была сформирована таблица-справочник полей для интеграции (ext.ReformaParameters). Структура представления, построенного на базе этой таблицы, рассматривается в таблице 3.8.

**Таблица 3.8.** Структура таблицы ext.vw\_ReformaParameters

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Id | INT | ID параметра |
| MethodId | INT | ID API-метода, в котором осуществляется отправка данного параметра |
| ReformaName | NVARCHAR(MAX) | Наименование поля данных в Реформе (берется из документации по интеграции) |
| Description | NVARCHAR(MAX) | Описание параметра (берется также из документации) |
| ViewId | INT | ID представления базы данных ОУ, в котором находится аналогичный параметр |
| ViewColumnId | INT | ID поля представления в ОУ, соответствующее аналогичному параметру |
| ViewName | NVARCHAR(MAX) | Наименование SQL-представления в базе данных ОУ |
| ViewColumnName | NVARCHAR(MAX) | Наименование поля представления |
| ExternalType | NVARCHAR(MAX) | Тип данных в источнике |

Структура таблицы отражает сопоставление полей источников данных Реформы и АИС: Объектовый учет. Поля представления ViewName и ViewColumnName определяют координаты, по которым возможно произвести выборку значения параметра динамическим способом. Случай, когда ViewName и ViewColumnName не имеют значений, означает, что прямое сопоставление параметров источников данных произвести невозможно и для получения параметра придется формировать запрос.

Алгоритм написания интеграционных динамических запросов рассматривается в пункте n части «Алгоритмическое обеспечение системы».

### Организация обработки информации

Обработка информации заключается в преобразовании данных, получаемых после выполнения хранимой процедуры выборки данных для указанного API-метода. Результатом такой выборки является временная таблица, которая в программном коде приложения представляет собой коллекцию записей, каждая из которых имеет структуру:

1. наименование поля – значение этого параметра должно быть идентично названию соответствующего поля класса. Если поле класса является частью вложенного объекта, то каждый уровень вложенности отмечается синтаксическим знаком, например, точкой;
2. значение, определяющее значение параметра;
3. индекс массива – данный параметр содержит значение индекса коллекции, при условии, что тип поля массив.

Данная коллекция преобразуется в объект API-класса, который затем отправляется с помощью метода API-интерфейса Реформе. Формирование объекта основывается на применение механизмов рефлексии ЯВУ C# и рекурсивного алгоритма. Его подробное описание приводится в части «Алгоритмическое обеспечение системы».

### Организация передачи информации

Передача информации между АИС: Объектовый учет и «Реформа ЖКХ» осуществляется посредством API-интерфейса, реализованного на базе протокола обмена данными SOAP (англ. Simple Object Access Protocol). Его принцип действия основывается на обмене структурированными xml-сообщениями, в чем и заключается основное преимущество протокола, так как при помощи xml-разметки возможно представление структуры данных любой сложности.

На момент разработки дипломного проекта портал «Реформа ЖКХ» поддерживал 23 метода API-интерфейса, подразделяющихся на два типа: GET, реализующих возможность загрузки данных и справочников с Реформы, и POST, обеспечивающих загрузку данных на сайт портала. В таблице 3.9 приведены наиболее часто используемые в ходе реализации подсистемы интеграции методы API-интерфейса.

Таблица 3.9 – Методы API-интерфейса Реформы

|  |  |
| --- | --- |
| **Наименование API-метода** | **Описание** |
| Login | Выполняет авторизацию внешней системы и открывает сеанс работы |
| SetRequestForSubmit | Осуществляет подачу запроса на раскрытие данных организацией |
| GetRequestList | Обеспечивает возвращение списка запросов подписки на управляющую организацию, поданных внешней системой |
| SetCompanyProfile | Изменяет данные по текущей/архивной анкете управляющей организации с соответствующим ИНН за указанный отчетный период |
| SetHouseLinkToOrganization | Добавляет дом с соответствующим идентификатором в управление организации с соответствующим ИНН |
| SetHouseProfile | Изменяет данные текущей анкеты дома с соответствующим идентификатором дома |
| SetFileToHouseProfile | Добавляет новый файл в заданный раздел текущей анкеты дома с соответствующим идентификатором дома |
| SetFileDeleted | Удаляет файл с соответствующим идентификатором |

Соответствующим идентификатором называется ID объекта/файла, служащий для синхронизации данных об объектах и файлах источников данных АИС: Объектовый учет и «Реформа ЖКХ». С помощью него сформирована таблица файлового хранилища (ext.ReformaFilesStorage) и организовано соответствие объектов, находящихся в управлении организаций.

### Организация выдачи информации

Выдача информации о проходящем интеграционном процессе производится на html-странице личного кабинета пользователя управляющей компании. Пользователю отображается подробная статистика, список API-запросов его компании и история статусов их выполнения.

# АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

# ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

# ТЕСТИРОВАНИЕ СИСТЕМЫ

# ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

## 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 8.1:

(8.1)

, где – общая трудоемкость разработки программного продукта,

– трудоемкость разработки i-го этапа проектирования,

*n* - общее количество этапов проектирования.

Проанализировав формулу n, можно утверждать, что если проект разделен на большее количество стадий разработки, то искомая оценка трудоемкости выполняемых работ будет точнее. В таблице 8.2 приведены данные о расчете величины параметра трудоемкости для каждого из этапов проектирования и для всего проекта в целом.

**Таблица 8.2** 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 8.3 представлен график проведения работ по проекту.

**Таблица 8.3** График проведения работ по проекту

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Вид работ** | **Исполнитель** | **Трудоемкость, чел.-час.** | **Кол-во дней** | **Продолжительность работы** | | | | | | | |
|
| 5 | 6 | 5 | 28 | 7 | 5 | 6 | 2 |
| 1 | Формирование требований к системе | Инженер - программист | 40 | 5 |  |  |  |  |  |  |  |  |
| 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 |  | | | | | | | |

## 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 8.4:

(8.4)

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

- цена за единицу i-го вида материального ресурса,

*i* - вид материального ресурса,

*n* - общее количество всех видов материальных ресурсов.

Результаты расчетов затрат на материальные ресурсы приведены в таблице 8.5.

**Таблица 8.5** Сумма затрат на материальные ресурсы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | 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 8.4. Необходимые расчеты отображены в таблице 8.6.

**Таблица 8.6** Расчет стоимости затрат на расходные материалы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Наименование | Единица измерения | Количество израсходованного материала | Цена за единицу, руб. | Сумма, руб |
| 1 | Оплата услуг интернет-провайлера | 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 8.7:

(8.7)

, где – сумма затрат на электроэнергию,

- паспортная мощность электрооборудования i-го вида, измерятся в кВт,

Ц - тариф электроэнергии, .

*i* - вид прибора электрооборудования,

*n* - общее число электроприборов.

Необходимые расчеты затрат на электроэнергию приведены в таблице 8.8.

**Таблица 8.8** Расчет затрат на электроэнергию

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Наименование единицы оборудования | Паспортная мощность, кВт. | Время работы оборудования, ч | Тариф электроэнергии, | Сумма, руб |
| 1 | Ноутбук | 0,1 | 540 | 2,9 | 156,6 |
| 2 | Принтер | 0,1 | 1 | 0,29 |
| 3 | Освещение рабочего места | 0,03 | 200 | 17,4 |
| Итого за электроэнергию | | | | | 174,29 |

## 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 () производится по формуле 8.9:

(8.9)

, где T – количество часов, затраченных на дипломное проектирование,

– часовая ставка сотрудника,

i – сотрудник,

n – общее количество сотрудников.

Расчеты заработных отчислений приведены в таблице 8.10.

**Таблица 8.10** Расчет заработной платы сотрудников

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Квалификация сотрудника | Трудоемкость, . | 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 |

## 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 8.11 приведены расчеты обязательных страховых взносов.

**Таблица 8.11** 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 |

## Расчет амортизации оборудования

Амортизация оборудования - это исчисляемый в денежном выражении износ основных средств в процессе их производственного использования. Расходы на амортизацию () выражаются при помощи формулы 8.12:

(8.12)

, где – стоимость i-го оборудования,

– годовая норма амортизации i-го оборудования, %,

– время работы i-го оборудования за весь период разработки дипломного проекта, ч.,

– эффективный фонд времени работы i-го оборудования за год, ч/год,

*i* – вид оборудования,

*n* – общее число различного оборудования.

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

## 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 8.13:

**Таблица 8.13** Расчет себестоимости программного решения

|  |  |  |
| --- | --- | --- |
| № п/п | 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 |

## 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 8.14:

(8.14)

, где – полная себестоимость, руб.,

– норматив рентабельности.

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

## 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 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:

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 8.15:

(8.15)

, где – экономический эффект, %,

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

– себестоимость проекта, руб.

Экономический эффект равен:

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

# БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ ПРОЕКТА

## Исходные данные

Исходные данные дипломного проектирования описаны в таблице 9.1.

**Таблица 9.1** Исходные данные

|  |  |  |
| --- | --- | --- |
| № п/п | 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 | - |

Окончание таблицы 9.1

|  |  |  |
| --- | --- | --- |
|  |  | - 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 травмоопасности | По вредности – вредный (III класс).  По травмоопасности – допустимый (II класс). |
| 17 | Вредные и опасные производственные факторы | См. рис. 9.3.1.1 |
| 18 | Виды загрязнений окружающей среды | - |
| 19 | Возможные чрезвычайные ситуации | См. п. 9.3.3 |

## Перечень нормативных документов

В списке представлен перечень основных нормативных документов:

1. Санитарные правила и нормы. СанПиН 2.2.2./2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы;
2. «Руководство по гигиенической оценке факторов рабочей среды и трудовых процессов. Критерии и классификация условий труда». Р 2.2.2006-05;
3. ГОСТ 12.0.003-74.ССБТ. (СТ СЭВ 790-77) Опасные и вредные производственные факторы. Классификация. М.: Изд-во стандартов, 1996;
4. ГОСТ 12.1.003-83.ССБТ. Шум. Общие требования безопасности. М.: Изд-во стандартов.1996;
5. ГОСТ 12.1.004-91.ССБТ. Пожарная безопасность. Общие требования. М.: Изд-во стандартов, 1996;
6. ГОСТ 12.1.005-88.ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны. М.: Изд-во стандартов, 1996.
7. ГОСТ 12.1.006-88.ССБТ. Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведения контроля. М.: Изд-во стандартов, 1998;
8. ГОСТ 12.1.019-79.ССБТ (СТ СЭВ 4880-84). Электробезопасность. Общие требования. М.: Изд-во стандартов, 1996;
9. ГОСТ 12.1.030-81.ССБТ. Электробезопасность. Защитное заземление зануление. М.: Изд-во стандартов, 1996;
10. ГОСТ 12.1.038-82.ССБТ. Электробезопасность. Предельно-допустимые значения напряжений прикосновения и токов. М.: Изд-во стандартов, 1996;
11. Правила устройства электроустановок. М.: Энергия, 1987;
12. Общесоюзные нормы технологического проектирования ОНТП 24-86., М.: МВД СССР, 1986;
13. СНиП 2.01.02-85. Противопожарные нормы. М.: Стройиздат,1986;
14. СНиП 2.04.05-86. Отопление, вентиляция, кондиционирование воздуха. М.: Стройиздат, 1988;
15. СНиП 23-05-95. Естественное и искусственное освещение. Анализ проектирования. М.: Энерго, 1996;
16. Р 2.2.013-94. Гигиена труда. М.: Госкомсанэпиднадзор России, 1994;
17. Правила пожарной безопасности в Российской Федерации – ППБ 01 03;
18. Нормы пожарной безопасности – НПБ 88-2001. Установки пожаротушения и сигнализации. Нормы и правила проектирования.

## Анализ потенциальных опасностей

Опасность – 1) 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; 2) 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. Блок-схема представлена на рисунке 9.2.



Рисунок 9.2 Блок-схема обеспечения безопасности объекта проектирования

### Анализ вредных и опасных производственных факторов

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

ГОСТ 12.0.003-74 разделяет производственные факторы на четыре группы: физические, химические, биологические и психофизиологические. На рисунке 9.3 приведена их классификация.



Рисунок 9.3 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:

1. умственное перенапряжение, потому что решаемые задачи часто требуют нестандартных решений;
2. эмоциональные перегрузки, когда обнаруженная проблема или поставленная задача требуют немедленного решения;
3. монотонность труда, если процесс разработки однообразен;
4. нервно-психические перегрузки из-за систематического недосыпания.

При игнорировании или не провождении 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:

1. пыль на рабочем месте – при больших скоплениях может вызывать заложенность дыхательных путей или появление аллергических заболеваний (кашель, сыпь и т.д.);
2. недостаточная освещенность или излишняя яркость экрана – вызывает перенапряжение глаз, что в итоге приводит к сильному ухудшению зрения;
3. шум от работающего компьютерного вентилятора. Наличие постоянного шумового «сопровождения» притупляет работу органов слуха.

#### 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:

1. при выборе рабочего места, оборудованного компьютером, предпочтение следует отдавать помещениям, ориентированным на север или северо-восток;
2. не допускается оборудовать рабочее место для работы с ПК в помещениях, расположенных в цокольных и подвальных помещениях;
3. 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;
4. рабочий стол (рабочее место) размещают таким образом, чтобы монитор был ориентирован боковой стороной к окнам, а естественный свет падал преимущественно слева;
5. 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;
6. 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:

1. 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;
2. 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;
3. рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию;
4. 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:

1. возможность изменения угла поворота в горизонтальной и вертикальной плоскостях с фиксацией в удобном для пользователя положении;
2. возможность регулировки яркости и контрастности матрицы экрана;
3. 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.

#### 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:

1. температура помещения в теплый период должна составлять 20-24°С, в холодный – 22-24 °С, в переходный – 18-22 °С;
2. допуск температурных колебаний должен быть не более 4%;
3. подвижность воздуха – от 0,1 до 0,2 м/с;
4. влажность воздуха – 60-70%;
5. недопустимость запыленности и загазованности воздуха.

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

1. снижение работоспособности;
2. нарушение самочувствия.

#### 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 удается 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 помещениях, где установлены ПЭВМ согласно документу СНиП II-4-79.

Единицей измерения освещенности помещения является «люкс». Требования к величине освещенности отображены в таблице 9.4.

**Таблица 9.4** Нормы освещенности рабочего пространства

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

Одно из важных гигиенических требований – это создание равномерного освещения рабочего пространства. Игнорирование или не следование данному правилу ведет к перенапряжению зрительных органов, а впоследствии к заметному ухудшению зрения.

#### Шум на рабочем месте

Шум – это общебиологический раздражитель, и в определенных условиях он может влиять на системы органов и организм человека в целом. Подвергаясь постоянному шумовому воздействию, страдают не только органы слуха, но и центральная нервная система и отделы головного мозга. Это приводит к быстрому утомлению, потере концентрации и снижению производительности.

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 превышается, то нормирование уровня звука достигается за счет установки малошумного оборудования и установки специальных звукопоглощающих панелей, корпусов для оборудования.

#### 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:

1. произвести замену ЭЛТ-монитора (электронно-лучевая трубка) на жидкокристаллический или LED, так как уровень их излучения на порядок ниже;
2. установить монитор в угол, чтобы излучение поглощалось стенами;
3. 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:

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

### Анализ воздействия на окружающую среду

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

### 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 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

В процессе дипломного проектирования 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:

1. пожары, взрывы бытового газа;
2. обрушение здания.

## 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:

1. правовая охрана труда (ПОТ);
2. техника безопасности (ТБ);
3. производственная санитария (ПС);
4. пожарная безопасность (ПБ).

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

1. обеспечение приоритета сохранения жизни и здоровья работников;
2. 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;
3. государственное управление охраной труда;
4. государственный надзор и контроль над соблюдением государственных нормативных требований охраны труда;
5. государственная экспертиза условий труда;
6. профилактика несчастных случаев и повреждения здоровья работников;
7. расследование и учет несчастных случаев на производстве и профессиональных заболеваний;
8. 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;
9. установление компенсаций за тяжелую работу и работу с вредными и опасными условиями труда;
10. обеспечение функционирования единой информационной системы охраны труда.

### 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, заключаются в установлении режима работы в зависимости от вида их трудовой деятельности:

1. группа «А» – работа по считыванию информации с монитора ПЭВМ с предварительным запросом;
2. группа «Б» – работа по вводу информации;
3. 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:

1. для группы «А» - по суммарному числу считываемых знаков за рабочую смену, но не более 60000 знаков за смену;
2. для группы «Б» - по суммарному числу считываемых или вводимых знаков за рабочую смену, но не более 40000 знаков за смену;
3. 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.

### Мероприятия по защите от опасных и вредных производственных факторов

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

1. в использовании новых технологий на производстве с целью снижения уровня ОВПФ;
2. удаление на расстояние от источника ОВПФ;
3. уменьшение времени нахождения в зоне действия ОВПФ;
4. установка регламента перерывов для рабочего процесса;
5. 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:

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

## 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 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:

1. соблюдение права человека на благоприятную окружающую среду;
2. обеспечение благоприятных условий жизнедеятельности человека;
3. обязательность оценки воздействия на окружающую среду при принятии решений об осуществлении хозяйственной и иной деятельности;
4. 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.

## Мероприятия по защите от чрезвычайных ситуаций

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

1. 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.
2. Защита в ЧС – включает проведение комплекса защитных мероприятий и использование средств индивидуальной и коллективной защиты.
3. Ликвидация ЧС – включает в себя работы и другие неотложные мероприятия, направленные на устранения последствий чрезвычайной 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.

## Расчетная часть

### 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 (описаны в пункте 9.3.1.4) по снижению уровня громкости звука.

В помещении, в котором ведется дипломное проектирование, находятся несколько источников шума. В таблице 9.6 представлены данные, полученные в результате измерения уровня шума для каждого из них.

**Таблица 9.6** 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 9.7:

(9.7)

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

*n* – число источников шума.

Произведя необходимые расчеты при помощи формулы получаем результат:

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

### 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 9.8, описывающую метод светового потока:

(9.8)

, где *F* – световой поток, Лк,

– нормированная минимальная освещенность, равная 200 Лк,

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

*S* – площадь помещения, м2,

*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.

В таблице 9.9 представлены параметры помещения, в котором ведется дипломное проектирование.

**Таблица 9.9** Параметры помещения

|  |  |
| --- | --- |
| **Параметр** | **Значение параметра** |
| Длина, a, м | 3,8 |
| Ширина, b, м | 6 |
| Высота, h, м | 3 |
| Коэффициент отражения стен, *Rст* | 30 |
| Коэффициент отражения потолка, *Rп* | 65 |

По измеренным параметрам определим площадь (S) помещения:

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

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

Произведем необходимые расчеты по нахождению величины параметра освещенности:

Необходимое число ламп рассчитывается по формуле 9.10:

(9.10)

, где *F* – световой поток равный 19 714 Лм,

– 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:

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

## Оценка эффективности принятых решений

В разделе «Безопасность и экологичность объекта проектирования» проведен анализ вредных и опасных факторов производства, характерных для процесса разработки дипломной работы бакалавра по специальности «Информационные системы и технологии». Приведенные расчеты уровня шума и освещенности рабочего помещения показывают, что условия труда, созданные работодателем, оптимальны и соответствуют всем санитарным нормам.

# ЗАКЛЮЧЕНИЕ

1. Особенности реализации аналогов неизвестны. [↑](#footnote-ref-1)
2. Например, <http://dom.gosuslugi.ru/> [↑](#footnote-ref-2)
3. Содержит в себе поля данных, которые поставляются в ИС «Реформа ЖКХ» посредством API-методов [↑](#footnote-ref-3)
4. Объект используется как аргумент API-метода. [↑](#footnote-ref-4)
5. Если имеет место быть вложенность типа «объект в объекте», то ее уровень указывается через разделитель. [↑](#footnote-ref-5)
6. Обеспечивает заполнение таблицы данными. [↑](#footnote-ref-6)
7. Реализация API-интерфейса не предусматривает установку ключа сессии в заголовок сообщения. [↑](#footnote-ref-7)
8. Рассматривается в пункте 3.3.1. [↑](#footnote-ref-8)
9. Рассматривалось в п. 3.1.5. [↑](#footnote-ref-9)