

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

---

Кафедра Прикладная математика и информатика  
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем  
(код и наименование направления подготовки, специальности)

---

Мобильные и сетевые технологии  
(наименование профиля, специализации)

---

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему: «Разработка ПО для расчета стоимости коммунальных услуг»

Обучающийся	<u>Р.А. Шлегель</u> (И.О. Фамилия)	<u>_____</u> (личная подпись)
Руководитель	<u>канд. пед. наук, доцент, О.М. Гущина</u> (ученая степень, звание, И.О. Фамилия)	<u>_____</u>
Консультант	<u>канд. пед. наук, доцент, А.В. Егорова</u> (ученая степень, звание, И.О. Фамилия)	<u>_____</u>

Тольятти 2023

## **Аннотация**

Тема бакалаврской работы: «Разработка ПО для расчета стоимости коммунальных услуг».

Цель работы – разработка программного обеспечения автоматизации процесса управления расчетом услуг.

Объект – система оплаты коммунальных услуг.

Предмет – программное обеспечение автоматизации процесса управления расчетом услуг.

Выпускная квалификационная работа состоит из введения, 3-х разделов, заключения, списка литературы, состоящего из 20 источников. Работа содержит 42 страницы 3 таблиц и 23 рисунка.

Во введении обоснована актуальность темы, установлены цель, задачи, объект и предмет работы.

В первом разделе выполнены постановка задачи на разработку ПО для расчета стоимости коммунальных услуг, формирование требований к ПО и сравнительный анализ используемых аналогов.

Во втором разделе выполнено проектирование ПО для расчета стоимости коммунальных услуг, в ходе которого проведен выбор архитектуры ПО, разработаны логической модели ПО и выполнено проектирование базы данных.

В третьем разделе проведены выбор средств разработки, реализация и тестирование приложения.

В заключении представлены краткие результаты работы и дана оценка выполнения задач и достижения цели.

## **Abstract**

The title of the bachelor's thesis is: "Development of utility billing management software in the local network of a utility bill aggregator".

The aim of the work is to develop software to automate the process of managing the settlement of services.

The object - the system of payment for public utilities.

Subject - software to automate the process of managing the calculation of services.

Graduation qualification work consists of an introduction, 3 chapters, a conclusion, a list of references, which consists of 20 sources. Work contains 42 pages, 3 tables and 23 figures.

In the introduction the topicality of the theme is grounded, the purpose, tasks, object and subject of the work are set.

The first chapter sets the problem of developing software for calculating the cost of utilities, the formation of requirements for software and comparative analysis of used analogs.

In the second chapter the design of utility costing software was done, during which the choice of software architecture was made, logical models of the software were developed and the design of the database was done.

The third chapter deals with the selection of development tools, implementation and testing of the application.

A summary of the results of the work is presented in the conclusion and an assessment of the fulfilment of the tasks and the achievement of the objective is given.

## Содержание

Введение.....	5
1 Постановка задачи на разработку ПО для расчета стоимости коммунальных услуг.....	7
1.1 Анализ деятельности по расчету стоимости коммунальных услуг .....	7
1.2 Формирование требований к ПО для расчета стоимости коммунальных услуг.....	9
1.3 Сравнительный анализ аналогов ПО для расчета стоимости коммунальных услуг.....	11
2 Проектирование ПО для расчета стоимости коммунальных услуг .....	13
2.1 Выбор архитектуры ПО для расчета стоимости коммунальных услуг .	13
2.2 Концептуальная модель системы расчета стоимости коммунальных услуг.....	14
2.3 Разработка модели данных ПО для расчета стоимости коммунальных услуг.....	18
3 Реализация ПО для расчета стоимости коммунальных услуг .....	23
3.1 Выбор средств разработки ПО для расчета стоимости коммунальных услуг.....	23
3.2 Реализация приложения для расчета стоимости коммунальных услуг	25
3.3 Тестирование приложения для расчета стоимости коммунальных услуг .....	33
Заключение .....	39
Список используемой литературы .....	40

## Введение

Работа агрегатора коммунальных услуг во многом связана с необходимостью оперативного реагирования на изменения в тарифах, в объемах предоставляемых коммунальных услуг и в системе льгот по коммунальным платежам. Все это должно быть отражено в расчете стоимости коммунальных услуг по каждому объекту недвижимости.

В настоящее время внедрение информационных технологий в сферу платежей ЖКХ является важным и востребованным направлением. Информатизация сферы ЖКХ в целом позволяет создать качественно новую систему управления ЖКХ, а также систему регламентированного взаимодействия с исполнительными органами государственной власти, преодолеть разрыв между уже относительно развитой нормативной базой и правоприменительной практикой, а также повысить качество принимаемых решений, социальную защищенность населения и усилить контроль за жилищно-коммунальной сферой деятельности.

Следовательно, тема работы, направленная на разработку ПО для расчета стоимости коммунальных услуг, является актуальной.

Объект – система оплаты коммунальных услуг.

Предмет – программное обеспечение автоматизации процесса управления расчетом услуг.

Цель работы – разработка программного обеспечения автоматизации процесса управления расчетом услуг.

Для достижения цели в работе решены следующие задачи:

- поставить задачи на разработку;
- выполнить проектирование и моделирование программного обеспечения;
- провести разработку и тестирование программного обеспечения.

Во введении обоснована актуальность темы, установлены цель, задачи, объект и предмет работы.

В первом разделе выполнены постановка задачи на разработку ПО для расчета стоимости коммунальных услуг, формирование требований к ПО и сравнительный анализ используемых аналогов.

Во втором разделе выполнено проектирование ПО для расчета стоимости коммунальных услуг, в ходе которого проведен выбор архитектуры ПО, разработаны логической модели ПО и спроектирована база данных.

В третьем разделе проведены выбор средств разработки, реализация и тестирование приложения.

В заключении представлены краткие результаты работы и дана оценка выполнения задач и достижения цели.

Использование результатов работы обеспечит эффективный расчет стоимости коммунальных услуг.

# 1 Постановка задачи на разработку ПО для расчета стоимости коммунальных услуг

## 1.1 Анализ деятельности по расчету стоимости коммунальных услуг

Деятельность агрегатора коммунальных платежей предусматривает интенсивное информационное взаимодействие, в рамках которого выполняется сбор показаний приборов учета, формирование платежных документов, расчеты с поставщиками услуг, выставление счетов потребителям услуг, реализация тарифной политики с учетом различных льгот и правил федерального, регионального и муниципального уровней.

Основные функции и информационные потоки этой деятельности выделены с помощью диаграммы потоков данных DFD [16], показанной на рисунке 1.



Рисунок 1 – Диаграмма потоков данных деятельности агрегатора коммунальных платежей

В рамках функции Фиксация показаний приборов учета выполняется обмен запросами и показаниями приборов с потребителями услуг. Полученные от потребителей показания заносятся в базу данных для расчета оплаты.

В ходе управления платежами выполняется обмен с потребителями запросами на выполнение платежа (квитанциями на оплату услуг ЖКХ) и информацией о выполненных платежах. Выполненные платежи отражаются в базе данных, формируя величину неоплаченных услуг для каждого потребителя.

На основании величины неоплаченных услуг процесс Управление расчетом услуг формирует информацию о суммах к оплате для каждого потребителя и передает ее в процесс управления платежами. Также этот процесс осуществляет распределение услуг по потребителям, назначение и изменение стоимости платежей за услуги ЖКХ для каждого потребителя с учетом тарифов поставщиков услуг и существующих льгот для различных потребителей услуг ЖКХ. Процесс управления учетом услуг является наиболее информационно нагруженным и от его эффективности во многом зависит правильная работа агрегатора коммунальных платежей в целом.

На основании результатов преддипломной практики и консультаций со специалистами агрегатора коммунальных платежей «Квартплата 24» предметом выпускной квалификационной работы определена автоматизация процесса управления расчетом услуг.

Поставлена задача создать приложение на java с возможностью:

- редактирования справочника характеристики объекта недвижимости (площадь, количество прописанных, потребление горячей воды, потребление холодной воды, и т.д.);
- добавления новых объектов недвижимости с указанием характеристик из предыдущего пункта;
- добавления новых видов услуг;
- редактирование формулы расчета стоимости каждой услуги;

- привязывание нескольких услуг к объекту недвижимости;
- расчет стоимости услуг по каждому объекту недвижимости.

На итоговой экранной форме должно быть предоставлена возможность выбора объекта недвижимости. Для каждого объекта должны отображаться оказанные услуги, формула расчета по каждой услуге, итоговая (за все услуги) сумма.

## **1.2 Формирование требований к ПО для расчета стоимости коммунальных услуг**

Для формирования требований к приложению мы будем использовать классификацию FURPS+.

Данная классификация требований разработана и успешно применяется для создания приложений и информационных систем в настоящее время [9], [4].

Основа любого приложения или системы, согласно классификации FURPS+, это её функционал, то есть то, для чего предназначена система, и какие возможности в ней реализованы.

Классификация требований к системе FURPS+ была разработана Робертом Грэйди (Robert Grady) из Hewlett-Packard и предложена в 1992 году. Сокращение FURPS расшифровывается так:

- Functionality, функциональность
- Usability, удобство использования
- Reliability, надежность
- Performance, производительность
- Supportability, поддерживаемость [8]

Требования к проектируемому приложению представлены в таблице 1.

Таблица 1 – Требования к приложению

Требование	Приоритет	Описание
<b>Functionality</b>		
Ведение справочников характеристик объекта недвижимости	1	Пользователь должен иметь возможность добавить, удалить или отредактировать информацию в базе данных по объекту недвижимости
Добавление новых объектов недвижимости	1	Пользователь должен иметь возможность ввести новый объект недвижимости
Добавление новых видов услуг	1	Пользователь должен иметь возможность ввести новый вид услуг
Редактирование формулы расчета стоимости каждой услуги	1	Пользователь должен иметь возможность провести редактирование формулы расчета стоимости каждой услуги
Привязывание нескольких услуг к объекту недвижимости	1	Пользователь должен иметь возможность провести привязывание нескольких услуг к объекту недвижимости
Расчет стоимости услуг по каждому объекту недвижимости	1	Программа должна обеспечить расчет стоимости услуг по каждому объекту недвижимости
<b>Usability</b>		
Удобный и понятный интерфейс	1	Каждое действие должно быть интуитивно понятным, используются стандартные жесты для управления приложением и логичные пункты в меню
<b>Reliability</b>		
Доступность системы 24/7	2	Приложение (хотя бы частично) должно быть доступно клиенту вне зависимости от внешних факторов (работа сервера, баз данных, интернет-соединение)
<b>Performance</b>		
Время запуска приложения не более 3-х секунд	3	-
Время отклика приложения не более 1 секунды	2	-
<b>Supportability</b>		
Корректная работа функционала на различных устройствах	1	Приложение должно стабильно работать на любых устройствах, соответствующих минимальным системным требованиям

Приоритет реализации конкретных требований приложения представлен значениями от 1 до 5, где 1 – наивысший приоритет, 5 – наименьший.

### 1.3 Сравнительный анализ аналогов ПО для расчета стоимости коммунальных услуг

Для целесообразности разработки нового приложения необходимо проверить, если ли среди используемых аналогов то, что подходит под сформулированные требования. Наиболее популярными решениями в области формирования платежей ЖКХ являются приложения [13][12][10][11], представленные в таблице 2:

Таблица 2 – Сравнительный анализ аналогов

Функциональность	ГИС ЖКХ;	Контур ЖКХ	1С:Учет в управляющих компаниях ЖКХ	Домиленд	Инфокрафт ЖКХ 365
Прием платежей	+	-	-	+	-
История показаний приборов учёта	+	+	+	-	+
История платежей	+	+	-	+	+
Возможность редактирования формул платежей	-	-	-	-	-

Рассмотрены следующие информационные системы агрегации ЖКХ платежей:

- ГИС ЖКХ;
- Контур ЖКХ;
- 1С:Учет в управляющих компаниях ЖКХ;
- Домиленд;
- Инфокрафт ЖКХ 365.

Ключевая функция разрабатываемого приложения – возможность редактирования формул платежей – в выбранных для анализа аналогах отсутствует.

Выводы по разделу:

Исследована деятельность агрегатора коммунальных платежей и построена модель потоков данных, содержащая три основных функции – фиксация показаний приборов учета, управление платежами, управление расчетом стоимости коммунальных услуг. Обоснована актуальность автоматизации расчета стоимости коммунальных услуг.

Для проектируемого приложения сформированы требования по методологии FURPS+, указан приоритет реализации каждого требования.

Проведён анализ функциональных возможностей программ-аналогов: ГИС ЖКХ, Контур ЖКХ, 1С:Учет в управляющих компаниях ЖКХ, Домиленд и Инфокрафт ЖКХ 365. В результате анализа установлено, что ключевая функция разрабатываемого приложения – возможность редактирования формул платежей – в выбранных для анализа аналогах отсутствует. Это определяет необходимость разработки нового приложения.

## **2 Проектирование ПО для расчета стоимости коммунальных услуг**

### **2.1 Выбор архитектуры ПО для расчета стоимости коммунальных услуг**

Стандартным решением для организации связи Java-приложений с базой данных является JDBC. JDBC выполняет три основные функции работы с базами данных:

- установление связи с СУБД;
- передача базе данных SQL-запросов;
- обработка полученных результатов.

Архитектура JDBC подразумевает использование как минимум четырех частей: Java-приложения, которое вызывает методы JDBC; собственно, компонента JDBC, который распределяет клиентские вызовы по соответствующим базам данных; драйвера JDBC, который поддерживает сеанс связи с конкретной базой данных, и базы данных. Если драйвер JDBC поставляется производителем базы данных, то Java-приложение может получить доступ к ней через механизм JDBC без посредников. Этот вариант JDBC позволяет работать в двухуровневой архитектуре клиент-сервер.

«В целом, JDBC предлагает полный набор необходимых низкоуровневых функций для связи с базами данных, который может быть использован другими приложениями или трансляторами JSQL. Метод CallableStatement позволяет запускать внутренние процедуры СУБД, что весьма важно в тех случаях, когда пользователям запрещено напрямую изменять содержание базы данных, но разрешено работать с определенными функциями СУБД» [18].

На основании изложенного для разрабатываемого программного обеспечения выбираем двухуровневую архитектуру [2] (рисунок 2).

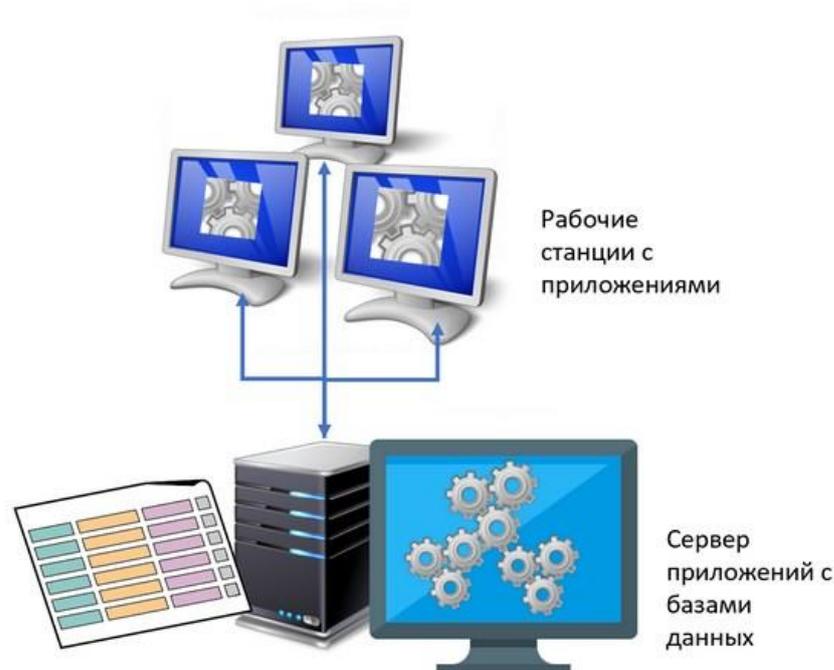


Рисунок 2 - Двухуровневая архитектура «клиент-сервер» [2]

Связь приложения-клиента с сервером базы данных будет реализована с помощью JDBC [5], [20].

## **2.2 Концептуальная модель системы расчета стоимости коммунальных услуг**

Центральной задачей проектируемого приложения является управление формулами для расчета стоимости услуг ЖКХ. Эти формулы различны для разных видов услуг, например, расчет стоимости горячего водоснабжения ведется по показаниям прибора учета горячей воды, расчет услуг по отоплению проводится на основании площади и только в холодное время года. Также формулы зависят от характеристик самого объекта, так горячее водоснабжение при отсутствии счетчика рассчитывается на основании числа прописанных, а для нежилых помещений без счетчиков – на основании площади.

Несмотря на разнообразие, любую формулу для расчета услуг ЖКХ можно представить, как произведение трех основных составляющих: базы, тарифа и коэффициента.

База относится к объекту недвижимости, например площадь, число прописанных человек, неоплаченные показания счетчика.

Тариф относится к услуге, например, тариф на горячее водоснабжение по счетчику, тариф горячего водоснабжения жилых объектов без счетчика, тариф горячего водоснабжения нежилых объектов без счетчика.

Коэффициент отражает особенности оказания услуги объекту и является характеристикой пары объект – услуга. Например, с помощью коэффициента можно отразить различие в расчете услуги вывоза мусора с объектов многоэтажной застройки и из частного сектора.

Операции изменения, добавления и удаления можно объединить общим термином «редактирование». Это позволит более кратко описать варианты использования, например прецеденты ввод нового объекта и изменение существующего объекта объединяются в один прецедент Редактирование объекта.

На один объект может быть назначено несколько услуг, и одна услуга может оказываться нескольким объектам. Подключением будем называть пару объект-услуга и, соответственно, операции по подключению, отключению и редактированию услуг объекту образуют вариант использования Редактирование подключения.

Редактирование базы в части изменения величин неоплаченных показаний приборов учета осуществляется внешними информационными системами – информационная система управления платежами вносит информацию об оплаченных услугах, информационная система фиксации показаний приборов учета вносит информацию о текущих показаниях счетчиков.

Диаграмма прецедентов [6] показана на рисунке 3.

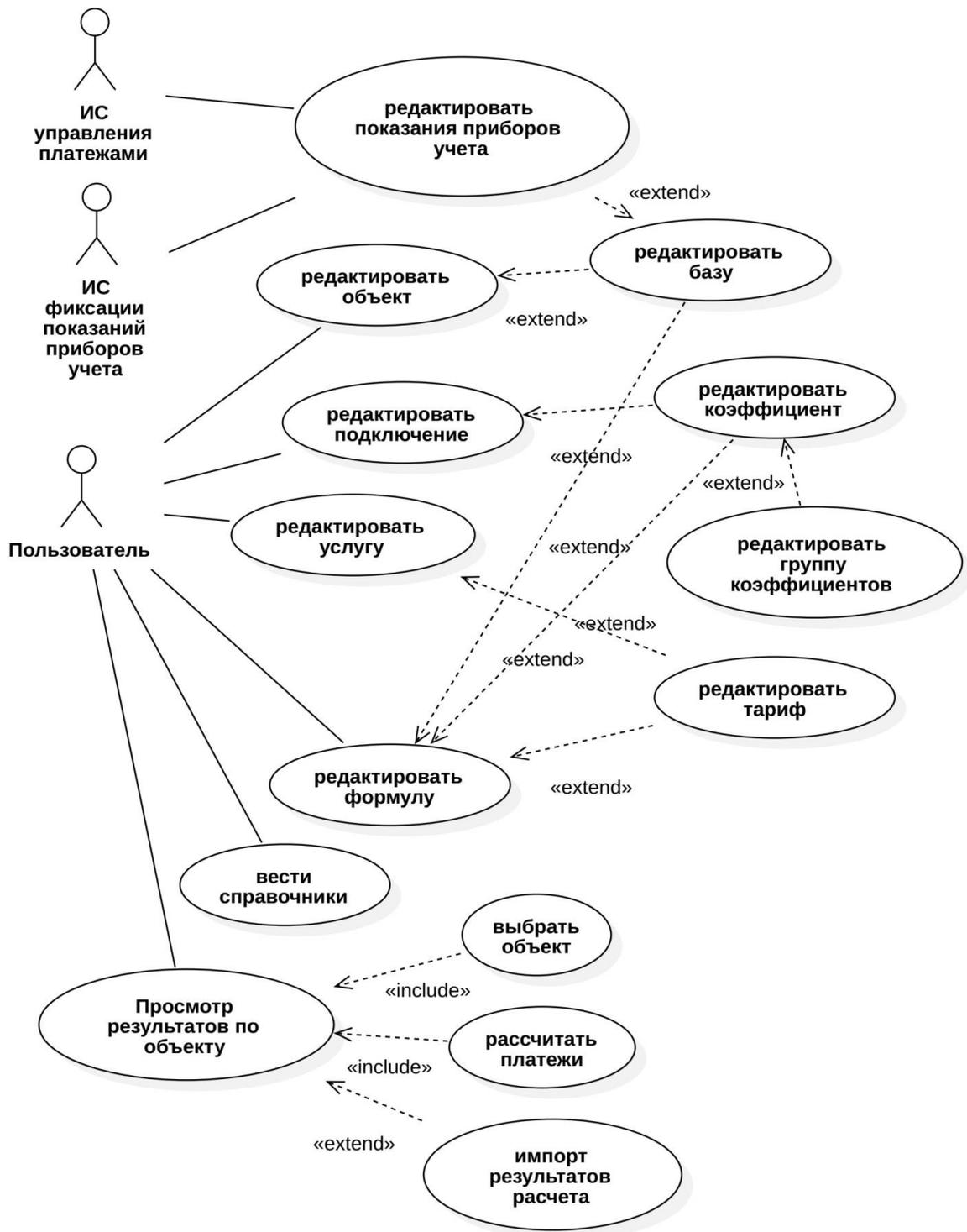


Рисунок 3 – Диаграмма прецедентов

Вторым актором является Пользователь, выполняющий основное количество вариантов использования.

При просмотре результатов по объекту пользователь необходимо выбрать объект и выполнить расчет. При этом есть возможность экспорта результатов.

Основные классы – объекты и услуги – содержат достаточно большое количество различных подклассов и их целесообразно классифицировать: услуги по видам услуг, объекты – по расположению, форме собственности и т.п. Ведение классификаторов или справочников также является вариантом использования проектируемого приложения.

Основные классы проектируемого программного обеспечения показаны на диаграмме классов [7] (рисунок 4).

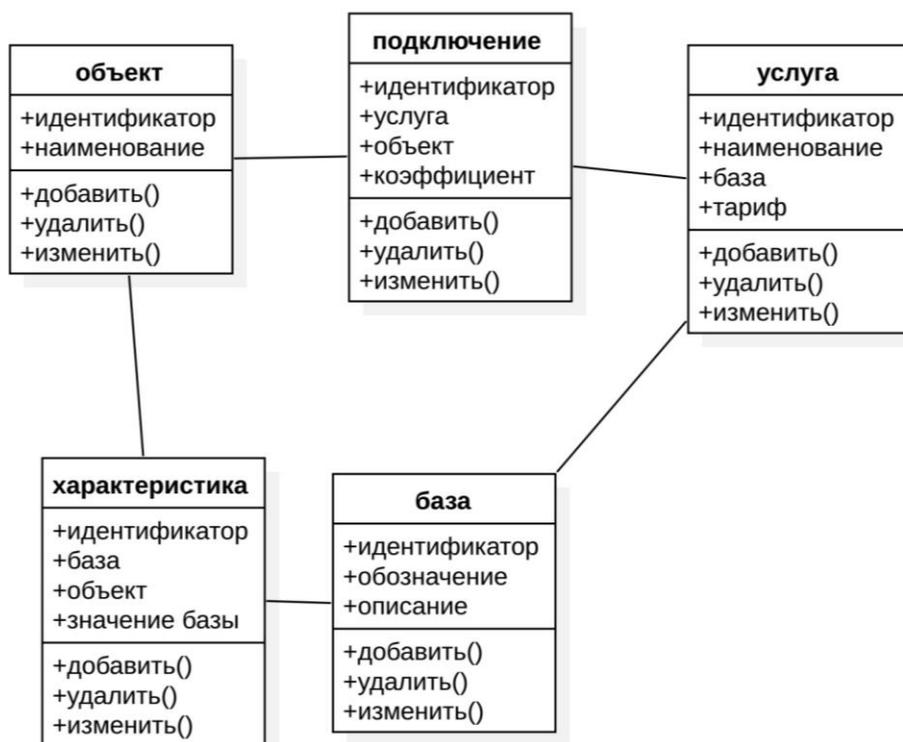


Рисунок 4 – Диаграмма классов

Набор наименований базовых величин для расчета вынесен в отдельный класс «база», а величины, необходимые для расчета представлены в классе «характеристика», который ассоциирован с классами «объект» и «база».

Данные о том какие услуги к какому объекту подключены хранятся в классе «подключение».

### 2.3 Разработка модели данных ПО для расчета стоимости коммунальных услуг

Логическая модель данных в нотации IDEF1X [107] показана на рисунке 5.

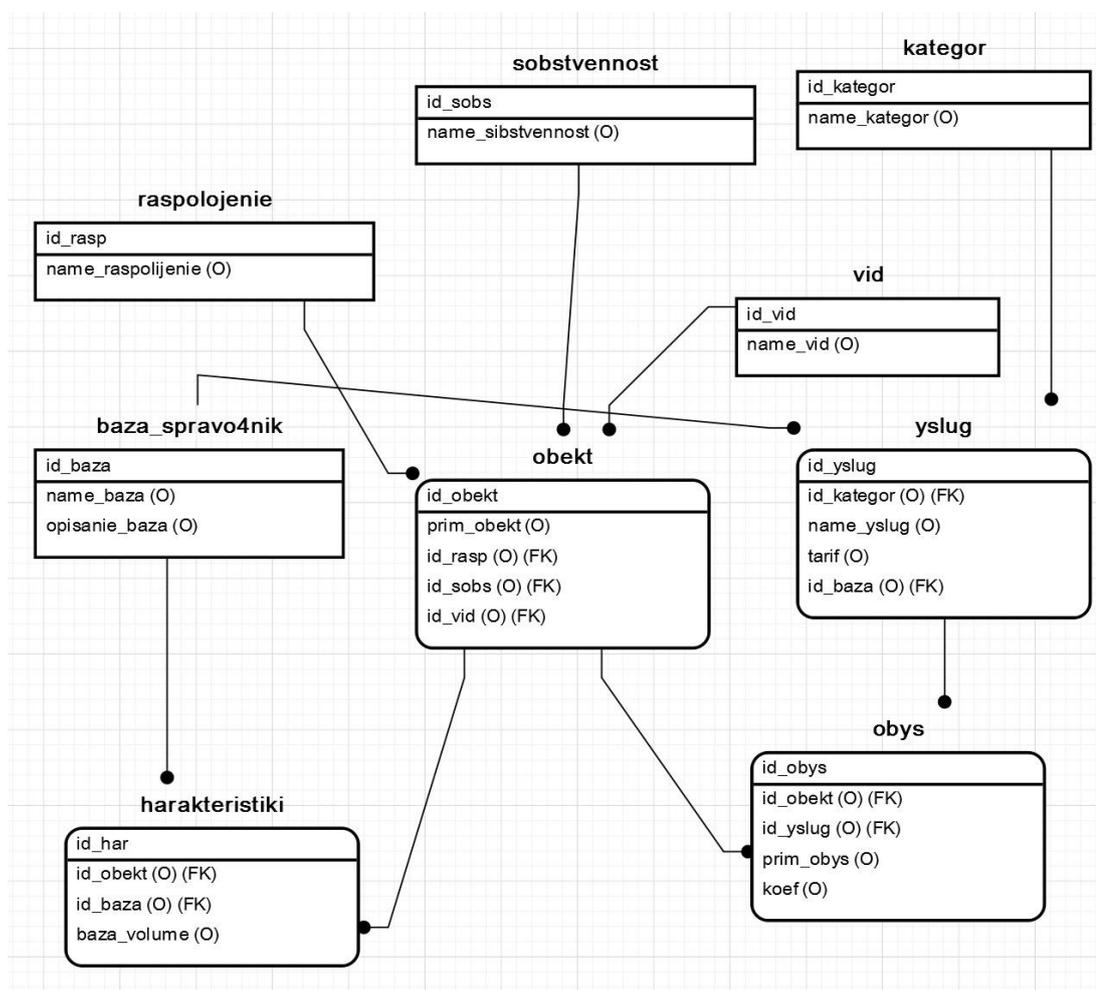


Рисунок 5 – Логическая модель данных

Отношение множества объектов к множеству услуг реализовано через промежуточную таблицу obys, которая помимо внешних ключей объектов и услуг содержит коэффициент, в котором учитывается особенность расчета

коммунальной услуги для конкретного подключения и поле для ввода примечаний.

В базе данных предусмотрена классификация объектов по таким признакам как расположение (многоэтажная застройка, частный сектор и т.п.), собственность (физическое лицо, юридическое лицо, муниципальная, федеральная и т.п.) и вид (жилое, нежилое, промышленное и т.п.). Использование этих классификационных признаков обеспечивает возможность группового изменения формул платежей для определенной категории объектов.

Для классификации услуг предусмотрена одна категория – вид (водоснабжение, водоотведение, вывоз мусора и т.п.). Применение данной классификации позволит управлять тарифами на группы услуг, например отключить все услуги вида «отопление» с окончанием отопительного сезона.

Тривиальное решение по хранению базовых величин для расчета оплаты (числа прописанных, площади, показаний счетчиков и т.п.) в данном случае не является оптимальным. Главным затруднением при таком подходе будет хранение обозначений баз в виде имен полей таблицы. При расчете тогда будет необходимо включать в запрос имена полей как переменные величины, например - если рассчитываем водоснабжение, то использовать поле с показанием счетчика, а если рассчитываем отопление, то использовать поле с площадью. Это сильно усложнит логику и затруднит редактирование и изменение формул расчета.

Предлагаемое решение – хранить все названия баз в таблице `baza_spravo4nik`, а числовые значения баз для каждой пары записей из справочника баз и объекта – в таблице `harakteristiki`.

Физическая модель в нотации Мартина [17] показана на рисунке 6. Синим показаны таблицы, голубым – представления.

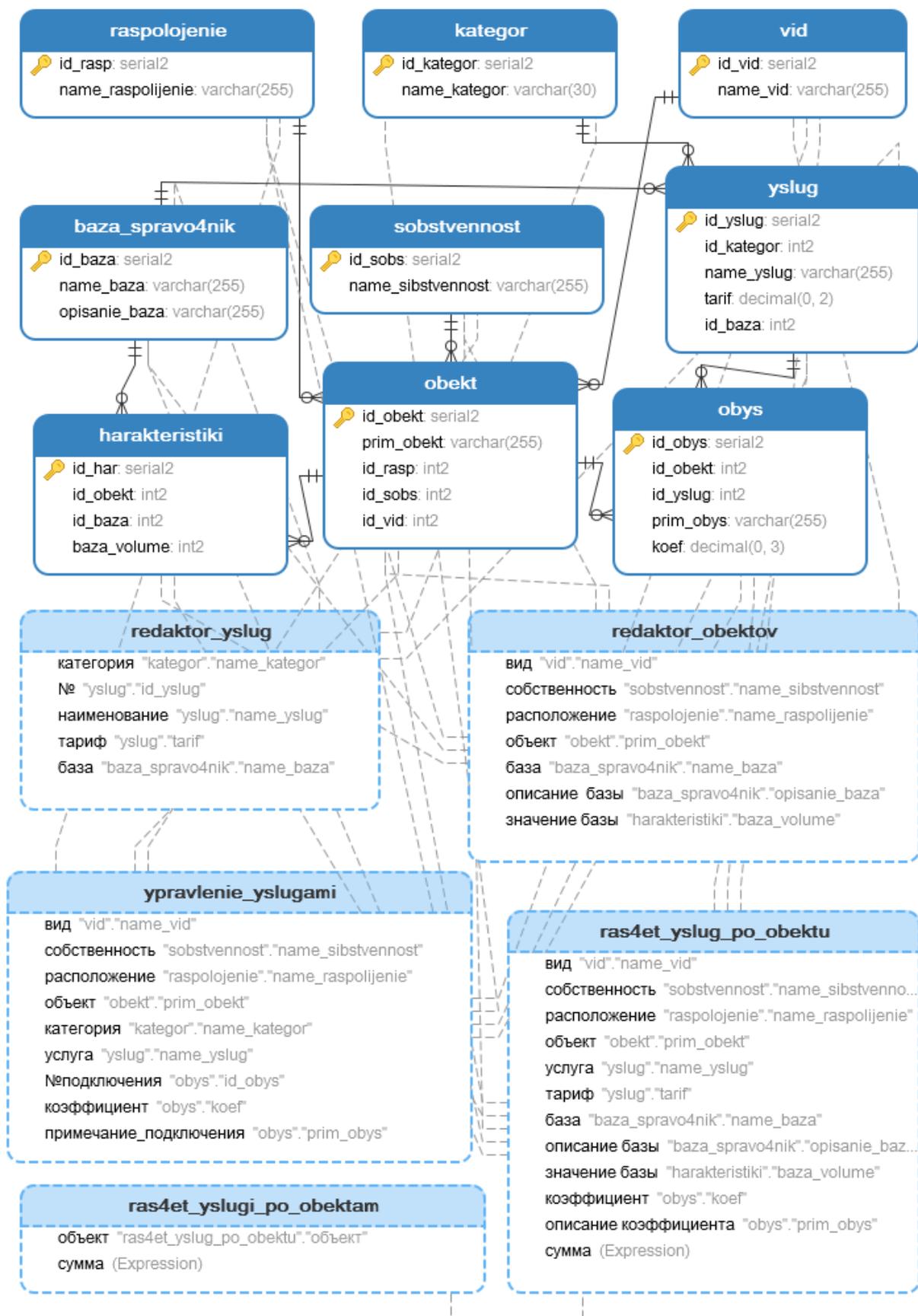


Рисунок 6 – Физическая модель данных

База является атрибутом услуги, так, например горячее водоснабжение по счетчику рассчитывается от базы показаний счетчиков, а горячее водоснабжение без счетчика – от базы числа прописанных. В общем случае для каждой услуги используется одна база для расчета. Поэтому в модель добавлена связь между таблицами справочника баз и услугами.

В целом необходимость эффективного редактирования формул расчета платежей привела к разнесению информации по связанным таблицам, что несколько затрудняет понимание при работе непосредственно с таблицами. Например, название объекта хранится в одной таблице, числовые значения характеристики «площадь» объекта – в другой, а наименование характеристики «площадь» в третьей.

Решение этого вопроса выполнено на этапе физического проектирования с использованием представлений.

Так для редактирования услуг удобнее использовать представление `redaktor_uslug`, которое формируется из таблиц услуги, справочник баз и категории.

Представление редактора объектов `redaktor_obektov` связано с таблицами вид, собственность, расположение, объект, справочник баз и характеристики.

Для управления подключением и отключением услуг к объекту используется представление `upravlenie_uslugami` от таблиц вид, собственность, расположение, объект, категория, услуги, справочник баз и подключения (`obus`).

Физическая модель служит основой для создания структуры таблиц и представлений базы данных в выбранной СУБД PostgreSQL.

Выводы по разделу

Во втором разделе выполнено обоснование выбора двухуровневой клиент-серверной архитектуры и механизма взаимодействия приложения с базой данных.

Построена концептуальная модель разрабатываемого программного обеспечения, включающая диаграмму прецедентов и диаграмму классов.

Предложена схема формализации формул услуг ЖКХ, представляющая любую формулу для расчета услуг, как произведение трех основных составляющих: базы, тарифа и коэффициента. Применение предложенной схемы обеспечивает возможно эффективное редактирование формул расчета услуг в разрабатываемом программном обеспечении.

Выполнено моделирование данных разрабатываемого программного обеспечения, в ходе которого разработаны логическая и физическая модели базы данных. В ходе физического моделирования для более удобного представления информации, разнесенной по разным таблицам, использованы представления.

### **3 Реализация ПО для расчета стоимости коммунальных услуг**

#### **3.1 Выбор средств разработки ПО для расчета стоимости коммунальных услуг**

При выборе средств разработки необходимо выбрать язык программирования, СУБД, IDE и, при необходимости, дополнительные средства для разработки и тестирования приложения.

«Java — язык программирования, который актуален уже почти 20 лет. Созданный Джеймсом Гослингом в 1995 году, он до сих пор входит в тройку лидеров многих рейтингов по популярности и востребованности. В Java много внимания уделено раннему обнаружению ошибок и динамической проверке во время работы программы, поэтому язык считается безопасным и на нем часто пишут важные системы: банковские терминалы, системы обработки транзакций, сервисы координации перелетов и другие» [14].

В качестве языка программирования выбрана Java SDK 8.0.2.

«MySQL - Это одна из самых популярных реляционных баз данных.

Плюсы MySQL:

- версию MySQL community edition (для сообщества) можно установить бесплатно;
- простой синтаксис и умеренная сложность;
- совместимость с облачными решениями» [1].

«СУБД PostgreSQL - разделяет свою популярность с MySQL. Это объектно-реляционная СУБД, в которой пользовательские объекты и табличные подходы объединяются для создания более сложных структур данных.»[2]

«СУБД MS SQL-сервер является полностью коммерческим инструментом и одной из самых популярных реляционных СУБД. Он хорошо справляется с эффективным хранением, изменением и управлением реляционными данными» [2].

В качестве СУБД выбрана PostgreSQL 9.5.25.

«Интегрированная среда разработки или по-английски Integrated development environment — IDE — это программа, которая содержит в себе инструменты для разработки программного обеспечения. Обычно современная среда разработки включает в себя:

- текстовый редактор с подсветкой кода;
- компилятор или интерпретатор;
- браузер классов, инспектор объектов и диаграмму иерархии классов;
- средства автоматизации сборки;
- отладчик;
- средства для интеграции с системами управления версиями (Git);
- инструменты для упрощения конструирования графического интерфейса пользователя» [3]/

«IDE IntelliJ IDEA с точки зрения возможностей и цены поставляется в двух вариантах: бесплатного Community edition, и платного Ultimate edition с расширенной функциональностью. Community edition предназначена для JVM- и Android-разработки. Бесплатная версия поддерживает Java, Kotlin, Groovy и Scala; Android; Maven, Gradle и SBT; работает с системами контроля версий Git, SVN, Mercurial и CVS, а Ultimate edition приспособлена для веб- и enterprise-разработки» [3].

«IDE Eclipse полностью бесплатная, с открытым исходным кодом, написанным преимущественно на Java. Тем не менее, её модульная архитектура позволяет использовать Eclipse и с другими языками.»[3].

«NetBeans IDE представляет собой бесплатную интегрированную среду разработки с открытым кодом для разработчиков программного обеспечения. Среда IDE NetBeans предоставляет все средства, необходимые для создания профессиональных приложений рабочей среды, корпоративных, мобильных и веб-приложений на языках Java, C/C++, а также на других динамических языках.»[19].

В качестве IDE выбрана NetBeans 8. В качестве средств разработки выбрано СУБД PostgreSQL 9.5.25, IDE NetBeans 8, java SDK 8.0.2.

### 3.2 Реализация приложения для расчета стоимости коммунальных услуг

В соответствии с функциональными требованиями в приложении необходимо реализовать следующие экранные формы:

- формы ведения справочников;
- редактирования услуг;
- редактирования объектов;
- управления подключением объекта к услугам;
- редактирования коэффициентов в расчетных формулах;
- расчета услуг по объекту.

Соотношение функциональных требований и экранных форм, на которых они реализуются, представлено в таблице 3.

Таблица 3 – Реализация функционала ПО на экранных формах

Функциональность	Экранная форма
Ведение справочников характеристик объекта недвижимости	Категории услуг, Базы для расчёта услуг, Расположение объекта, Собственность объекта, Вид объекта
Добавление новых объектов недвижимости	Редактор объектов
Добавление новых видов услуг	Редактор услуг
Редактирование формулы расчета стоимости каждой услуги	Редактор объектов, Редактор услуг, Управление подключением, Редактор коэффициентов
Привязывание нескольких услуг к объекту недвижимости	Управление подключением
Расчет стоимости услуг по каждому объекту недвижимости	Расчет услуг по объекту

Формы ведения справочников являются наиболее простыми и взаимодействуют с таблицами базы данных напрямую – без использования представлений.

Для внесения новой записи необходимо нажать кнопку «Новая», после чего в таблице появится пустая строка. Заполнить строку и нажать кнопку

«Сохранить», после чего введенные данные заносятся в таблицу базы данных. Кнопка «Удалить» удаляет выделенную запись из таблицы базы данных. Кнопка «Отменить» удаляет добавленную строку если до этого не нажата кнопка «Сохранить».

Форма ведения справочника баз для расчета услуг показана на рисунке 7.

№	Обозначение	Описание
1	XBC	холодное водоснабжение счетчик
2	ГВС	горячее водоснабжение счетчик
3	ВОС	водоотведение счетчик
4	прописано	количество прописанных человек

Рисунок 7 – Форма ведения справочника баз для расчета услуг

Форма ведения справочника категорий услуг показана на рисунке 8.

№	Наименование
3	отопление
4	вывоз мусора
5	домофон
6	новая категория услуг

Рисунок 8 – Форма ведения справочника категорий услуг

Форма ведения справочника расположений объектов показана на рисунке 9.

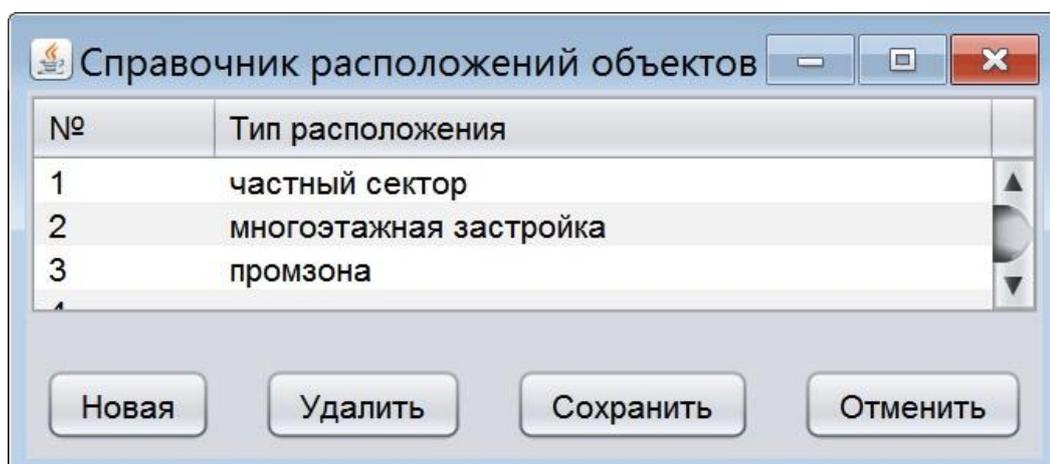


Рисунок 9 – Форма ведения справочника расположений объектов

Форма ведения справочника собственности объектов показана на рисунке 10.

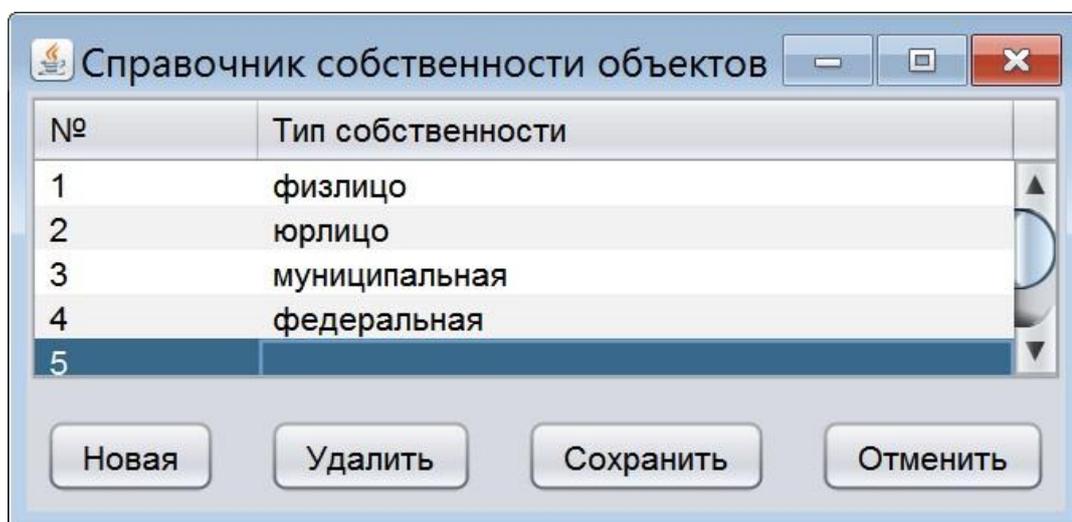


Рисунок 10 – Форма ведения справочника собственности объектов

Форма ведения справочника видов объектов показана на рисунке 11.

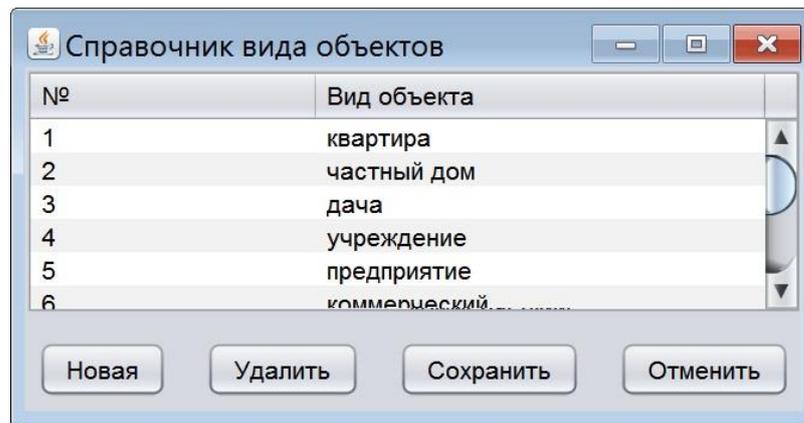


Рисунок 11 – Форма ведения справочника видов объектов

Редактор услуг (рисунок 12) выводит таблицу наименований услуг с возможностью фильтрации по категориям.

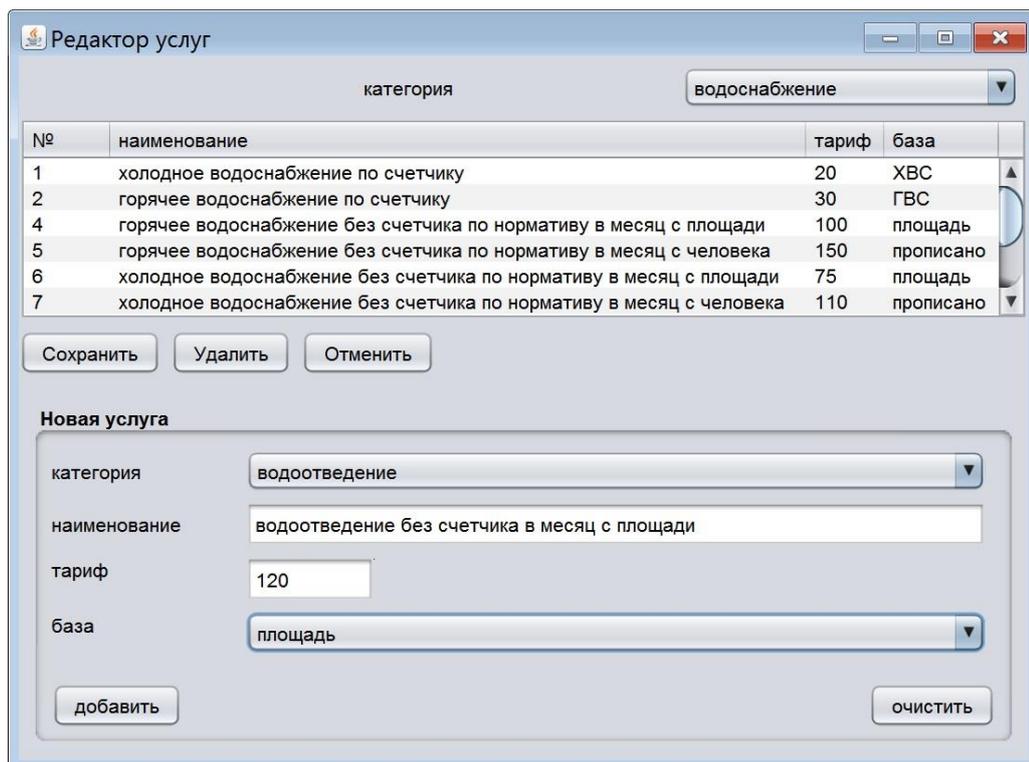


Рисунок 12 – Форма редактора услуг

Добавление новой услуги выполняется вводом в поля в нижней части формы наименования и тарифа услуги с выбором из списков категории и

базы. При нажатии кнопки «Добавить» строка добавляется в таблицу, после чего возможно отменить или занести строку в таблицу базы данных.

Удаление услуг выполняется путем выделения строки с услугой в таблице.

Редактирование тарифов выполняется непосредственно в таблице с возможностью сохранения или отмены.

Редактор объектов, показанный на рисунке 13, обеспечивает возможность добавления и удаления объектов недвижимости и редактирования их характеристик.

Редактор объектов

**Выбор и редактирование объекта**

вид: квартира

собственность: физлицо

расположение: многоэтажная застройка

Удалить Новый Сохранить Отменить

объект

- квартира 1
- квартира 2
- квартира 3
- квартира 4
- квартира 5
- квартира 6
- квартира 7

Редактировать характеристики объекта

описание	обозначение	значение
холодное водоснабжение счетчик м <sup>3</sup>	ХВС	2
горячее водоснабжение счетчик м <sup>3</sup>	ГВС	1
водоотведение счетчик м <sup>3</sup>	ВОС	3
число проживающих, человек	прописано	2
площадь объекта, м <sup>2</sup>	площадь	50

Сохранить значения Добавить характеристику Удалить характеристику

Отменить

Рисунок 13 – Форма редактора объектов

Для того чтобы уладить объект необходимо выбрать объект в одноколоночной таблице в правой части формы и нажать кнопку «Удалить». Добавление объекта выполняется нажатием на кнопку «Новый», вводом

названия объекта в добавившуюся строку таблицы и нажатием на кнопку «Добавить».

Для редактирования характеристик объекта необходимо вызвать таблицу характеристик выделенного в одноколоночной таблице объекта. Редактирование значений характеристик выполняется непосредственно в таблице с возможностью сохранения или отмены. Также имеется возможность удалить выделенную характеристику.

Подключение и отключение услуги к объекту выполняется с помощью формы управления подключением (рисунок 14).

Управление подключением

**Выбор объекта и категории услуг**

вид: квартира

собственность: физлицо

расположение: многоэтажная застройка

категория: все

объект:

- квартира 1
- квартира 2
- квартира 3
- квартира 4
- квартира 5
- квартира 6
- квартира 7

Показать подключенные услуги

наименование услуги	коэффициент	примечание
холодное водоснабжение по счетчику	1	
горячее водоснабжение по счетчику	1	
водоотведение по счетчику	1	

Сохранить коэффициенты    Удалить услугу    Подключить услугу

**Подключаемая услуга**

категория: электроснабжение

услуга: электроснабжение без счетчика

коэффициент: 1    примечание:

Рисунок 14 – Форма управления подключением

Список объектов формируется фильтрацией по виду, собственности и расположению, а также по категории услуг. Выделив объект в списке наименовании отфильтрованных объектов, можно вызвать список подключенных услуг. В данном списке можно изменять коэффициенты, а также удалять (отключать) услуги от объекта.

Для того чтобы подключить услугу к объекту необходимо в нижней части формы выбрать категорию и наименование подключаемой услуги, ввести коэффициент и примечание при необходимости. После этого при нажатии кнопки «Подключить услугу» информация переносится в список подключенных услуг.

Форма для группового редактирования коэффициентов показана на рисунке 15.

Групповой редактор коэффициентов

**Выбор объекта и категории услуг**

вид: все

собственность: все

расположение: все

категория: теплоснабжение

услуга: все

**Операции**

заменить на: 0

увеличить на %

увеличить на

уменьшить на %

уменьшить на

Показать текущее значение коэффициента по группе

Показать новое значение коэффициента по группе

от 0,9 до 1,2

от 0 до 0

Сохранить

Отменить

Рисунок 15 – Форма редактора коэффициентов

Использование данной формы позволяет выполнять групповое редактирование коэффициентов, например, обнулить коэффициенты

категории «теплоснабжение» по окончании отопительного сезона без отключения услуги от объектов.

Итоговая форма (рисунок 16) позволяет вывести суммы коммунальных платежей по отфильтрованным объектам и детализацию платежей по выбранному объекту. Имеется возможность импорта выводимых данных в файл Excel с использованием кнопок, расположенных в нижней части формы для дальнейшей обработки. Данная форма является главной и содержит меню для вызова остальных форм.

Расчет услуг по объекту

справочники услуги объекты подключения коэффициенты

**Выбор объекта**

вид: квартира

собственность: физлицо

расположение: многоэтажная застройка

Показать суммы по выборке

объект	сумма
квартира 1	300
квартира 2	200
квартира 3	250
квартира 4	50
квартира 5	100
квартира 6	300
квартира 7	400

Показать детализацию по объекту

наименование услуги	база	тариф	коэффициент	сумма
холодное водоснабжение по счетчику	2	10	1	20
горячее водоснабжение по счетчику	1	15	1	15
водоотведение по счетчику	3	5	1	15

Выгрузить суммы по объектам

Выгрузить выборку

Рисунок 16 – Форма расчета услуг по объекту

На следующем этапе необходимо выполнить функциональное тестирование приложения чтобы проверить как оно обеспечивает требуемую функциональность.

### 3.3 Тестирование приложения для расчета стоимости коммунальных услуг

Функциональное тестирование - это вид тестирования программного обеспечения, который проверяет программную систему на соответствие функциональным требованиям и спецификациям. Целью функциональных тестов является проверка каждой функции программного приложения путем предоставления соответствующих входных данных и проверки выходных данных на соответствие функциональным требованиям.

Функциональное тестирование в основном включает в себя тестирование «черного ящика» и не касается исходного кода приложения. Это тестирование проверяет пользовательский интерфейс, API, базу данных, безопасность, связь клиент/сервер и другие функциональные возможности тестируемого приложения. Тестирование может проводиться как вручную, так и с помощью автоматизации [21].

Для тестирования приложения на ввод данных взят случай добавления нового объекта – частный дом и подключения к нему оплаты газа по счетчику.

Сначала необходимо ввести показания газового счетчика в справочник баз для расчета (рисунок 17) и добавить новую категорию услуг «газоснабжение» в справочник категорий услуг (рисунок 18).

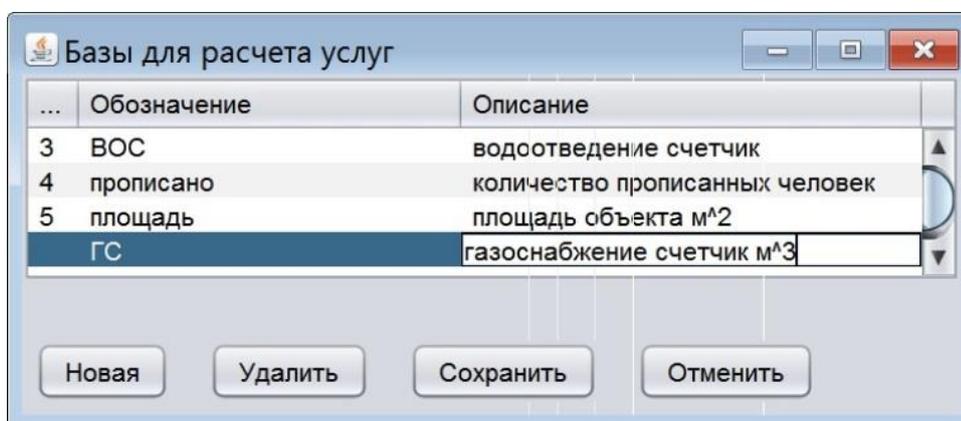


Рисунок 17 – Ввод данных в справочник баз расчета

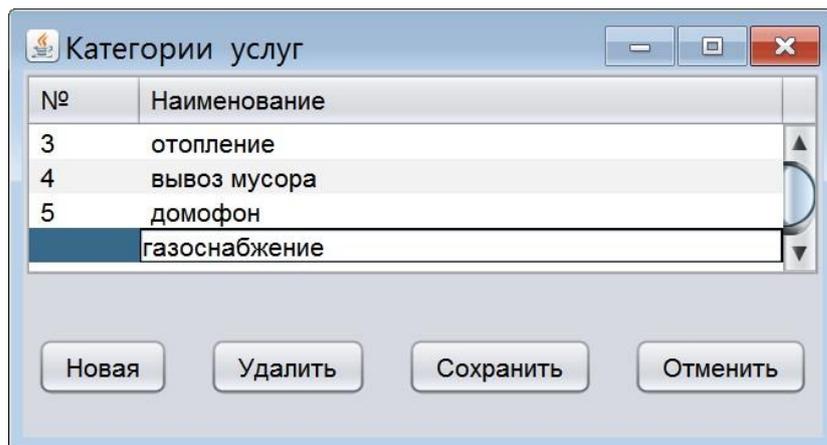


Рисунок 18 – Ввод новой категории услуг

Затем в редакторе услуг необходимо ввести новую услугу «газоснабжение физических лиц по счетчику» в категории «газоснабжение» с тарифом 25 рублей за кубометр (рисунок 19).

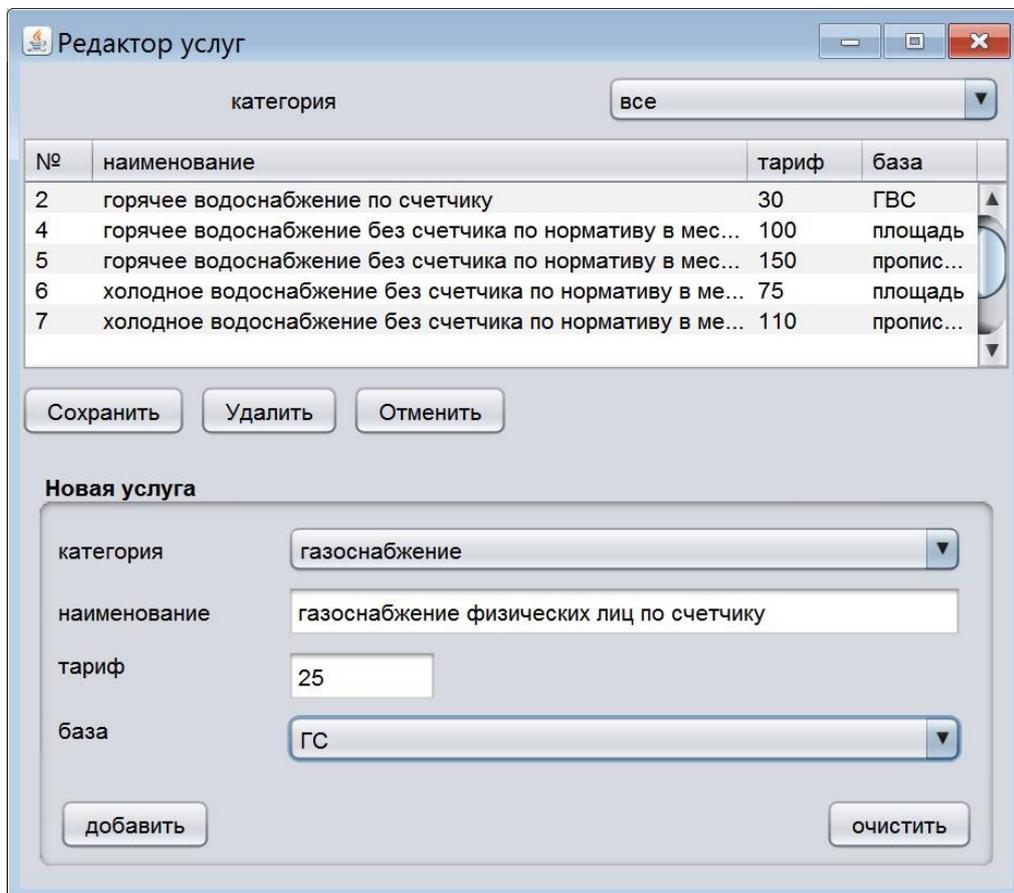


Рисунок 19 – Ввод новой услуги

На следующем шаге выполнен ввод объекта «частный дом 1» вида «частный дом», собственности «физлицо», расположение «частный сектор». Объекту добавлена характеристика «газоснабжение счетчик м<sup>3</sup>», которой присвоено значение 10 (рисунок 20).

Редактор объектов

Выбор и редактирование объекта

вид: частный дом

собственность: физлицо

расположение: частный сектор

Удалить Новый Сохранить Отменить

объект: частный дом 1

Редактировать характеристики объекта

описание	обозначение	значение
газоснабжение счетчик м <sup>3</sup>	ГС	10

Сохранить значения Добавить характеристику Удалить характеристику

Отменить

Рисунок 20 – Ввод объекта

Для подключения услуги объекту на форме управления подключением (рисунок 21) сначала выбран объект с фильтрацией по виду, собственности и расположению. В отфильтрованном списке содержится единственный объект, соответствующий настройке фильтра. В нижней части формы выбраны категория услуг «газоснабжение» и наименование услуги

«газоснабжение физических лиц по счетчику»), введен коэффициент единица, означающий что никаких особых изменений условий оплаты данное подключение не содержит. Поле «примечание» оставлено незаполненным.

Управление подключением

**Выбор объекта и категории услуг**

вид: частный дом

собственность: физлицо

расположение: частный сектор

категория: газоснабжение

объект: частный дом 1

Показать подключенные услуги

наименование услуги	коэффициент	примечание
---------------------	-------------	------------

Сохранить коэффициенты    Удалить услугу    Подключить услугу

**Подключаемая услуга**

категория: газоснабжение

услуга: газоснабжение физических лиц по счетчику

коэффициент: 1    примечание:

Рисунок 21 – Подключение услуги

Результат, показанный на рисунке 22, свидетельствует о том, что программа работает корректно – список отфильтрованных объектов показывает единственный частный дом, соответствующий настройкам фильтра, детализация содержит единственную подключенную к объекту

услугу – газоснабжение по счетчику. Суммы в отфильтрованном списке и в детализации равны.

The screenshot shows a software application window titled "Расчет услуг по объекту" (Calculation of services by object). The window has a menu bar with "справочники услуги объекты подключения коэффициенты" (reference services objects connection coefficients). Below the menu bar, there is a section "Выбор объекта" (Object selection) with three dropdown menus: "вид" (type) set to "частный дом" (private house), "собственность" (ownership) set to "физлицо" (individual), and "расположение" (location) set to "частный сектор" (private sector). To the right of these menus is a button "Показать суммы по выборке" (Show sums by selection). Below this is a small table with two columns: "объект" (object) and "сумма" (sum). The table contains one row: "частный дом 1" (private house 1) with a sum of 250. Below this table is a button "Показать детализацию по объекту" (Show details by object). At the bottom of the window, there is a large table with five columns: "наименование услуги" (service name), "база" (base), "тариф" (tariff), "коэффициент" (coefficient), and "сумма" (sum). The table contains one row: "газоснабжение физических лиц по счетчику" (gas supply for individuals by meter) with values 10, 25, 1, and 250. At the bottom of the window, there are two buttons: "Выгрузить суммы по объектам" (Export sums by objects) and "Выгрузить выборку" (Export selection).

наименование услуги	база	тариф	коэффициент	сумма
газоснабжение физических лиц по счетчику	10	25	1	250

Рисунок 22 – Расчет услуг газоснабжения по новому объекту

Тестирование разработанного программного обеспечения выполнено успешно.

Программа по результатам тестирования показала свою работоспособность. Встраивание разработанного ПО в информационную систему на основании диаграммы компонентов (рисунок 23) предусматривает взаимодействие модуля управления расчетов услуг с модулями фиксации показаний счетчиков и с модулем управления платежами через соответствующие интерфейсы.

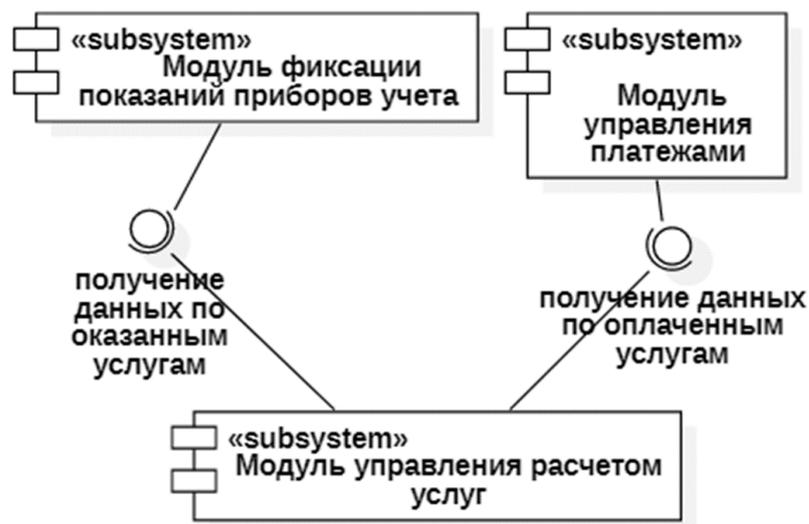


Рисунок 23 – Диаграмма компонентов

Выводы по разделу:

В третьем разделе выполнен выбор средств разработки, в ходе которого определены целесообразные для использования в данном проекте СУБД (PostgreSQL 9.5.25), IDE (NetBeans 8) и язык (java SDK 8.0.2).

Разработано программное обеспечение для расчета стоимости коммунальных услуг и позволяющее выполнять редактирование расчетных формул.

В результате функционального тестирования разработанного программного обеспечения показано, что оно позволяет вводить новые объекты и услуги, назначать (подключать и отключать) услуги объекту, редактировать формулы расчета услуг, проводить комплексную замену формул в различных режимах фильтрации объектов и услуг. В целом разработанное программное обеспечение по результатам тестирования работает корректно

## Заключение

Разработанное программное обеспечение позволяет эффективно осуществлять расчет стоимости коммунальных услуг.

В результате постановки задачи исследована деятельность агрегатора коммунальных платежей и построена модель потоков данных, содержащая три основных функции – фиксация показаний приборов учета, управление платежами, управление расчетом стоимости коммунальных услуг. Обоснована актуальность автоматизации расчета стоимости коммунальных услуг. Для проектируемого приложения сформированы требования по методологии FURPS+, указан приоритет реализации каждого требования. Проведён анализ аналогов, показавший отсутствие на рынке готового приложения, соответствующего предъявляемым требованиям.

Полученные в ходе проектирования архитектурные решения, модели и алгоритмы содержат всю необходимую информацию для разработки приложения. Разработанная база данных обеспечивает эффективное хранение информации и удобное структурированное представление информации. Физическая модель данных разработана для СУБД PostgreSQL и содержит таблицы и представления.

Разработанное java приложение обеспечивает редактирования справочников, добавления новых объектов недвижимости с указанием характеристик, добавления новых видов услуг, редактирование формулы расчета стоимости каждой услуги, привязывание нескольких услуг к объекту недвижимости и расчет стоимости услуг по каждому объекту недвижимости. По результатам тестирования разработанное приложение работает корректно.

Таким образом, все поставленные задачи были успешно выполнены, цель работы – достигнута.

## Список используемой литературы

1. Анализ популярных реляционных систем управления базами данных (2022 г) [Электронный ресурс]. URL: <https://drach.pro/blog/hi-tech/item/196-popular-relational-dbms-2022> (дата обращения: 01.02.2023).
2. Архитектура «Клиент-сервер» [Электронный ресурс]. URL: <https://itelon.ru/blog/arkhitektura-klient-server/?ysclid=193uf84y2c790984084>. (дата обращения: 29.02.2023).
3. Выбор IDE для Java-разработки. [Электронный ресурс]. URL: <https://javarush.com/groups/posts/1642-eclipse-netbeans-ili-intellij-idea-vihbiraem-ide-dlja-java-razrabotki> (дата обращения: 01.02.2023).
4. Заяц А. М., Васильев Н.П. Проектирование и разработка веб-приложений. Введение в frontend и backend разработку на JavaScript и node.js. М.: Издательство «Лань», 2020. 120 с.
5. Интернет-программирование : учеб. пособие / М.А. Колотилина. [Электронный ресурс]. URL: [https://lms2.sseu.ru/pluginfile.php/331799/mod\\_resource/content/2/Колотилина.pdf](https://lms2.sseu.ru/pluginfile.php/331799/mod_resource/content/2/Колотилина.pdf). (дата обращения: 29.08.2022).
6. Использование диаграммы вариантов использования UML при проектировании программного обеспечения [Электронный ресурс]. URL: <https://habr.com/ru/post/566218/> (дата обращения: 24.02.2023).
7. Использование диаграммы классов UML при проектировании и документировании программного обеспечения [Электронный ресурс]. URL: <https://habr.com/ru/post/572234/> (дата обращения: 31.01.2023)
8. Казиев, В. М. Введение в анализ, синтез и моделирование систем: учебное пособие. Москва, Саратов: ИнтернетУниверситет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. 270 с.
9. Классификация методов и моделей эффективности ИС [Электронный ресурс]. URL:

<https://sites.google.com/site/isefficiency/klassifikacia-izvestnyh-metodov-i-modelej>. (дата обращения: 19.02.2023).

10. Контур ЖКХ [Электронный ресурс]. URL: <https://kontur.ru/kontur-jkh-kvartplata> (дата обращения: 31.01.2023)

11. Обзор ПП для автоматизации деятельности ТСЖ, ЖКХ [Электронный ресурс]. URL: <https://infostart.ru/1c/articles/189465/> (дата обращения: 31.01.2023)

12. Расчет ЖКУ. Обзор программ для начисления квартплаты в 2022 г. [Электронный ресурс]. URL: <https://gkhhelp.ru/documents/articles/49208.html> (дата обращения: 31.01.2023)

13. Шесть лучших программ для учета ЖКХ (ТСЖ, ТСН и УК) [Электронный ресурс]. URL: <https://otzyvmarketing.ru/articles/8-best-accounting-software-zhkh-tszh-tsn-uk/> (дата обращения: 31.01.2023)

14. Язык программирования Java: особенности, синтаксис и идеи для первых проектов [Электронный ресурс]. URL: <https://ru.hexlet.io/blog/posts/yazyk-programirovaniya-java-osobennosti-sintaksis-i-idei-dlya-pervyh-proektov> (дата обращения: 01.02.2023).

15. Crow's Foot Notation – Relationship Symbols And How to Read Diagrams [Электронный ресурс]. URL: <https://www.freecodecamp.org/news/crows-foot-notation-relationship-symbols-and-how-to-read-diagrams/> (дата обращения: 31.01.2023)

16. Data Flow Diagram – A Comprehensive Tutorial [Электронный ресурс]. URL: <https://www.cybermedian.com/data-flow-diagram-a-comprehensive-tutorial> (дата обращения: 31.01.2023)

17. ISO/IEC/IEEE International Standard - Information technology -- Modeling Languages -- Part 2: Syntax and Semantics for IDEF1X97 (IDEFobject), in ISO/IEC/IEEE 31320-2:2012(E) , vol., no., pp.1-320, 30 Oct. 2012, doi: 10.1109/IEEESTD.2012.6357338.

18. Java и базы данных [Электронный ресурс]. URL: <https://www.osp.ru/news/articles/1997/0511/13032031> (дата обращения: 30.01.2023)

19. NetBeans IDE [Электронный ресурс]. URL: [https://ru.bmstu.wiki/NetBeans\\_IDE](https://ru.bmstu.wiki/NetBeans_IDE) (дата обращения: 30.01.2023)

20. Using Java Netbeans IDE Postgres database [Электронный ресурс]. URL: <https://quantlabs.net/blog/2015/03/using-java-netbeans-ide-postgres-database> (дата обращения: 31.01.2023)

21. What is Functional Testing? Types & Examples [Электронный ресурс]. URL: <https://www.guru99.com/functional-testing.html> (дата обращения: 12.05.2023)