

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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 кафедры)

01.03.02 Прикладная математика и информатика

(код и наименование направления подготовки, специальности)

Системное программирование и компьютерные технологии  
(направленность (профиль)/специализация)

## БАКАЛАВРСКАЯ РАБОТА

на тему **Моделирование процесса оптимального поиска объектов  
недвижимости на основе метода анализа иерархий**

Студент	<u>М.А. Логинова</u> (И.О. Фамилия)	_____ (личная подпись)
Руководитель	<u>Э.В. Егорова</u> (И.О. Фамилия)	_____ (личная подпись)
Консультанты	<u>М.А. Четаева</u> (И.О. Фамилия)	_____ (личная подпись)

**Допустить к защите**

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский  
(ученая степень, звание, И.О. Фамилия) \_\_\_\_\_ (личная подпись)  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г.

Тольятти 2018

## АННОТАЦИЯ

С. 54, рис. 46, табл. 3, лит. 28 источников

Выпускная работа посвящена моделированию процесса оптимального поиска объектов недвижимости на основе метода анализа иерархий.

Ключевые слова: метод анализа иерархий, оптимальный поиск, объект недвижимости, принятие решения.

Поиска объектов недвижимости специалистом рутинный процесс, который затрачивает много времени, поэтому его необходимо автоматизировать.

Метод анализа иерархий (МАИ) помогает лицам, принимающим решения, выбирать наилучшее решение из нескольких вариантов и критериев отбора. МАИ был разработан для оптимизации принятия решений, когда перед вами стоит сочетание качественных, количественных и иногда противоречивых факторов, которые учитываются.

Целью выпускной работы является разработка математической модели оптимального поиска объектов недвижимости и реализация ее программным путем.

В ходе выпускной работы была исследована предметная область, были определены конструктивные компоненты, обеспечивающие распределение приоритетных параметров при поиске жилья, описана математическая модель на основе метода анализа иерархий, обеспечивающего оптимальное решение задачи выбора жилья по требованиям клиента, а также реализована программными средствами описанная математическая модель.

Разработанное приложение оптимизирует работу риэлтора, увеличит коэффициент производительности работников риэлтерской компании, а также исключит возможность субъективного мнения риэлтора при поиске предложений для клиента.

## ABSTRACT

The bachelor's thesis is devoted to the modeling of the optimal search process for real estate objects on the basis of the Analytic Hierarchy Process.

Search for real estate by a specialist is a routine process that takes a lot of time, so it needs to be automated.

The Analytic Hierarchy Process (AHP) helps decision-makers choose the best solution from several options and selection criteria. AHP was developed to optimize decision making when one is faced with a mix of qualitative, quantitative, and sometimes conflicting factors that are taken into consideration.

The aim of the work was the development of a mathematical model for the optimal search for real estate and its implementation by software.

In the course of the work, the subject domain was investigated, structural components that provide the distribution of priority parameters in the search for apartments were identified, a mathematical model based on the analytic hierarchy process that provides the optimal solution to the problem of choosing apartment according to the client's requirements was described and also the described mathematical model was implemented by software.

The developed application will optimize the work of a realtor, will increase the productivity ratio of employees of a real estate company, and will exclude the possibility of a subjective opinion of a realtor when searching for proposals for the client.

The bachelor's thesis includes 3 tables, 46 figures, the list of references contains 28 references. The work consists of 54 pages.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	5
ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПОИСКА РЕШЕНИЯ ДЛЯ ОБОСНОВАНИЯ ВЫБОРА МЕТОДА ИССЛЕДОВАНИЯ.....	7
1.1 Характеристика процесса поиска объектов недвижимости в теории принятия решений .....	7
1.2 Выбор метода решения поиска объектов недвижимости .....	8
1.3 Построение математической модели поиска решения на основе метода анализа иерархий.....	11
ГЛАВА 2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ ПОИСКА РЕШЕНИЯ	26
2.1 Моделирование системы поиска решения для риэлтерской компании ..	26
2.2 Разработка алгоритма работы приложения для оптимального поиска объектов недвижимости .....	27
2.3 Проект работы реализованного приложения поиска объектов недвижимости.....	30
ГЛАВА 3 ОЦЕНКА ОПТИМАЛЬНОСТИ РЕАЛИЗОВАННОГО ПОИСКА ОБЪЕКТОВ НЕДВИЖИМОСТИ НА ОСНОВЕ МЕТОДА АНАЛИЗА ИЕРАРХИЙ .....	43
3.1 Обоснование выбора критериев оптимальности для поиска объектов недвижимости.....	43
3.2 Оценка выбранного метода построения поиска решения.....	46
ЗАКЛЮЧЕНИЕ .....	50
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	52
ПРИЛОЖЕНИЕ А ДИАГРАММА КЛАССОВ .....	55
ПРИЛОЖЕНИЕ Б ПРОНУМЕРОВАННЫЙ НАПРАВЛЕННЫЙ ГРАФ С УЧЕТОМ ВРЕМЕНИ.....	56
ПРИЛОЖЕНИЕ В КЛАСС CRITERION .....	57
ПРИЛОЖЕНИЕ Г ВЕРСТКА СТАРТОВОЙ СТРАНИЦЫ ПРИЛОЖЕНИЯ .....	58
ПРИЛОЖЕНИЕ Д ЧАСТЬ ДЕРЕВА ПРИНЯТИЯ РЕШЕНИЙ.....	59

## ВВЕДЕНИЕ

Риелторство – это вид бизнеса, занимающийся посредническими и консультационными услугами в сфере купли-продажи или аренды жилой и коммерческой недвижимости. В последнее время, благодаря бурному развитию строительства и улучшению финансового благополучия населения, любой может приобрести собственное жилье, выбирая при этом его местоположение, планировку, этажность, количество комнат и т. п.

Каждый день в риэлтерские компании обращаются клиенты с вопросами продажи-покупки или аренды жилья. Найти подходящий вариант для клиента – долгое и трудоемкое занятие. Так как риэлтерские компании обладают обширной базой предложений, риэлторы тратят большое количество времени, подыскивая оптимальный вариант, удовлетворяющий требования клиента. Оптимизация этого процесса облегчит работу риэлтерским компаниям, увеличит коэффициент производительности ее работников, а также исключит возможность влияния субъективного мнения риэлтора при поиске предложений.

Объектом исследования выпускной работы является процесс поиска недвижимости сотрудниками риэлтерской компании.

Предметом исследования выпускной работы является моделирование оптимального поиска решения методом анализа иерархий.

Целью выпускной работы является разработка математической модели оптимального поиска объектов недвижимости и реализация ее программным путем.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- исследовать предметную область для определения конструктивных компонентов, обеспечивающих распределение приоритетных параметров при поиске жилья;
- исследовать применимость метода анализа иерархии для поиска объектов недвижимости;

- описать математическую модель на основе метода анализа иерархий, обеспечивающего оптимальное решение задачи выбора жилого фонда по требованиям клиента;
- реализовать математическую модель программными средствами для минимизации временных затрат риэлтором;
- доказать оптимальность реализованного метода.

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

В первой главе формируются теоретические основы поиска решения для обоснования выбора метода исследования, характеризуется процесс поиска объектов недвижимости, рассматриваются методы поиска решения, строится математическая модель поиска решения на основе метода анализа иерархий.

Во второй главе моделируется система поиска решения для риэлтерской компании, разрабатывается алгоритм работы приложения, а также программно реализуется модель поиска решения.

В третьей главе оценивается оптимальность реализованного поиска объектов недвижимости на основе метода анализа иерархий.

В заключении подводятся итоги выпускной работы.

# **ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПОИСКА РЕШЕНИЯ ДЛЯ ОБОСНОВАНИЯ ВЫБОРА МЕТОДА ИССЛЕДОВАНИЯ**

## **1.1 Характеристика процесса поиска объектов недвижимости в теории принятия решений**

Решением называется выбор определенного сочетания цели, действий, которые направлены на достижение этой цели, и способов использования имеющихся ресурсов. Субъектом всякого решения является лицо, принимающее решение.

В узком смысле принятием решений называется заключительный акт анализа вариантов, то есть результат выбора. Процесс рационального или иррационального выбора альтернатив, имеющий целью достичь осознаваемого результата также называют принятием решения[16]. Задача принятия решения направлена на определение наилучшего или оптимального способа действий для достижения поставленных целей[6].

Под целью подразумевается конечный результат или предварительное идеальное представление желаемого результата. В данном случае в качестве цели будет получение в результате поиска удовлетворяющий требованиям клиента объект недвижимости.

В широком смысле поиском называют стремление добиться чего-либо, либо найти что-либо. Поиском объекта недвижимости называется стремление найти объект недвижимости, удовлетворяющий требованиям клиента. Так как поиск объекта недвижимости происходит во времени, то вводится понятие процесса поиска объектов недвижимости. Этот процесс состоит из последовательности этапов и процедур и направлен на получение подходящего объекта недвижимости.

Процесс принятия решений может быть подразделен на две операции. Это выработка рекомендаций специалистами по выбору лучшего варианта и принятие окончательного варианта непосредственно лицом, принимающим решение.

ООО «БИЭМ-Недвижимость» - это компания, активно работающая в области недвижимости города Тольятти. Она предоставляет услуги брокериджа и консультирует участников рынка недвижимости. Компания «БИЭМ-Недвижимость» работает для быстрого и качественного выполнения своих услуг в сфере недвижимости. Одно из направлений ее деятельности это поиск объектов недвижимости для аренды или купли/продажи. Следовательно, для нее максимально необходимо приложение, способное увеличить производительность ее сотрудников, улучшить качество обслуживания клиентов[15].

Приложение, осуществляющее поиск объектов недвижимости, позволит быть риэлтерской фирме лучшими на рынке недвижимости, предоставлять услуги высокого качества затрачивая на это меньше времени. Данное приложение позволит оставаться «БИЭМ-Недвижимости» надежной компанией, совершенствоваться и повышать качество услуг.

Для того чтобы смоделировать процесс поиска объектов недвижимости, необходимо для начала рассмотреть методы поиска решений.

## 1.2 Выбор метода решения поиска объектов недвижимости

Существуют различные методы поиска решений. В таблице 1.1 рассмотрены самые используемые методы поиска решения.

Таблица 1.1 – Методы поиска решений

Модель Карнеги	Модель мусорного ящика	Метод анализа иерархий
К поиску решения подключаются менеджеры, окончательный выбор они могут сделать только в коалиции, сообща, одинаково представляя себе цели и приоритеты проблемы. Однако менеджеры не обладают достаточным временем, чтобы переработать всю информацию по проблеме.	Модель разработана для поиска решения, где цели, задачи, альтернативы и решения плохо определены, неопределенность характерна для каждого шага процесса принятия решения, полная информация, которая необходима для выработки решения, недоступна.	Метод анализа иерархий представляет элементы, определяющие суть проблемы, иерархически, тем самым декомпозируя проблемы на более простые составляющие части и обрабатывая последовательности суждения лица, принимающего решения, по парным сравнением.
Часто решения принимаются не для	Процесс поиска решений не выглядит как	В результате получается численная интенсивность,



Модель Карнеги	Модель мусорного ящика	Метод анализа иерархий
нахождения оптимального решения проблемы, а для обеспечения удовлетворенности участников коалиции. При рассмотрении проблем коалиция примет то решение, которое воспринимается как удовлетворительное всеми ее членами, однако это не гарантирует оптимальность решения проблемы[14]. Достоинством данного метода является то, что наряду с количественными и рациональными показателями окончательное решение может включать и качественные, такие как неформальное мнение менеджеров, их опыт, взгляды и интуиция.	последовательность шагов, которые начинаются с проблемы, а заканчиваются решением. Проблемы, потенциальные решения, участники принятия решений и благоприятные возможности для выбора смешиваются в четыре «потока». Если проблема, решение и участник случайно соединятся в одной точке, то проблема может быть урегулирована, но если решение не подходит данной проблеме, она может остаться нерешенной. Таким образом, решения не поддаются упорядочиванию и не являются результатом пошаговой логической последовательности.	определяющая степень взаимодействия элементов в иерархии[19]. В ходе решения данного метода происходит синтез множественных суждений, получение приоритетности критериев и нахождение альтернативных решений.

Из таблицы 1.1 видно, что для поиска объектов недвижимости подходящим методом является метод анализа иерархий, так как цели, задачи, альтернативы хорошо определены, проблема должна быть решена обязательно, и для ее решения необходимо идентифицировать все показатели, переработать всю информацию, полученную от клиента.

Методом анализа иерархий называется математический инструмент системного подхода к сложным проблемам принятия решений. Он также является методологической основой для решения задач выбора альтернатив посредством их многокритериального рейтингования, то есть помогает выбирать наилучшее решение среди нескольких альтернатив относительно нескольких критериев. Данный метод не предписывает лицу, принимающему решение, какое-либо правильное решение, а позволяет самому в интерактивном режиме найти такой вариант, который наилучшим образом согласуется с его пониманием сути проблемы и требованиями к ее решению. Лицом, принимающим решение, является один человек или группа людей,

выбирающих наилучший вариант действий, и ответственных за принятое решение. В 1970 году данный метод был разработан Томасом Саати, получил широкое распространение и до сих пор активно используется в управленческой практике, развиваясь учеными. Метод был разработан для оптимизации принятия решений, когда необходимо учесть качественные, количественные и иногда противоречивые факторы[8]. Метод анализа иерархий имеет большое количество аналогий в различных областях.

Так, например, метод анализа иерархии схож с теорией вероятностей. Так как сумма приоритетов альтернатив равна единицы, и значения их положительны, то можно сравнить их с вероятностями выбора альтернатив. А вероятностями гипотез можно считать приоритеты факторов, влияющих на рейтинг альтернатив. Таким образом, формула полной вероятности похожа по применению на способ вычисления альтернатив.

Модель принятия решения в выбранном методе представляется в виде направленного графа, что схоже с теорией графов. Альтернативы, главный критерий рейтингования альтернатив и факторы, влияющие на рейтинг альтернатив, являются узлами графа. Связи, указывающие влияние одних узлов, приоритеты других узлов являются направленными дугами графа.

Также метод анализа иерархий аналогичен теории неотрицательных матриц. Так как, проводя расчет рейтингов, математически основываешься на методы расчетов собственных векторов для неотрицательных матриц.

Частным случаем применения выбранного метода является технология принятия решения с помощью экспертных систем, основанных на байесовском способе логического вывода.

По способу решения и проведения процедуры согласования обратная задача данного метода аналогична обучению нейросети, следовательно, можно сказать, что выбранный метод имеет схожесть и с идеологией искусственных нейросетей.

Достоинством метода анализа иерархий является высокая универсальность, так как метод используется в разнообразных ситуациях: при

принятии решений по управлению на межгосударственном уровне, по вопросам отраслевых и частных проблем, при анализе возможных сценариев развития ситуации, при распределении ресурсов, составлении рейтинга клиентов, принятии кадровых решений и других разнообразных проблем в бизнесе, промышленности, здравоохранении и образовании.

Также метод отражает естественный ход человеческого мышления, понятным и рациональным образом структурирует сложную проблему принятия решения в виде иерархии, сравнивает и выполняет количественную оценку альтернативных вариантов решения. Таким образом, метод дает не только способ выявления наиболее предпочтительного решения, но и позволяет количественно выразить степень предпочтительности посредством рейтингования. Это способствует полному и адекватному выявлению предпочтений лица, принимающего решение.

Недостатком выбранного метода является необходимость получения большого объема информации от экспертов. Данный метод подходит для тех случаев, когда основная часть данных основана на предпочтениях лица, принимающего решение, в процессе выбора наилучшего варианта решения из множества существующих альтернатив[9].

Постановка задачи в методе анализа иерархий выглядит следующим образом: имеется множество альтернатив, то есть вариантов решений  $B_1, B_2, \dots, B_k$ , каждая их альтернатив оценивается списком критериев  $K_1, K_2, \dots, K_n$ , требуется определить наилучшее решение [22].

Для того чтобы реализовать выбранный метод поиска решений необходимо построить его математическую модель.

### **1.3 Построение математической модели поиска решения на основе метода анализа иерархий**

Математической моделью является математическое представление алгоритма метода анализа иерархий. Алгоритм метода анализа иерархий [20]:

- 1) выделение проблемы и определение цели;

- 2) выделение основных критериев и альтернатив;
- 3) построение иерархий в виде дерева: от цели через критерии к альтернативам;
- 4) построение матрицы попарных сравнений критериев по цели и альтернатив по критериям;
- 5) применение методики анализа полученных матриц;
- 6) определение весов альтернатив по системе иерархий.

В первую очередь необходимо определить проблему и предполагаемый результат. В данном случае – это найти квартиру, удовлетворяющую требованиям клиента. Далее необходимо построить иерархическую структуру, включающую цель, критерии, альтернативы и другие рассматриваемые факторы, влияющие на выбор. Иерархической структурой называется графическое представление проблемы в виде перевернутого дерева, где каждый элемент, за исключением самого верхнего, зависит от одного или более вышеразположенных элементов. Данная структура отражает понимание проблемы лицом, принимающим решение, а также помогает проанализировать все аспекты проблемы и глубже вникнуть в суть задачи, тем самым являясь инструментом для качественного моделирования сложных проблем.

На рисунке 1.1 отображена иерархия для задачи оптимального поиска предложений по требованиям клиента, где вершиной иерархии является главная цель - квартира; элементы нижнего уровня представляют множество вариантов достижения цели (альтернативы), элементы промежуточных уровней соответствуют критериям (факторам), которые связывают цель с альтернативами - это стоимость, количество комнат, район, этаж, тип жилья (первичная/вторичная), наличие балкона; в иерархии каждая альтернатива связана с каждым из критериев.

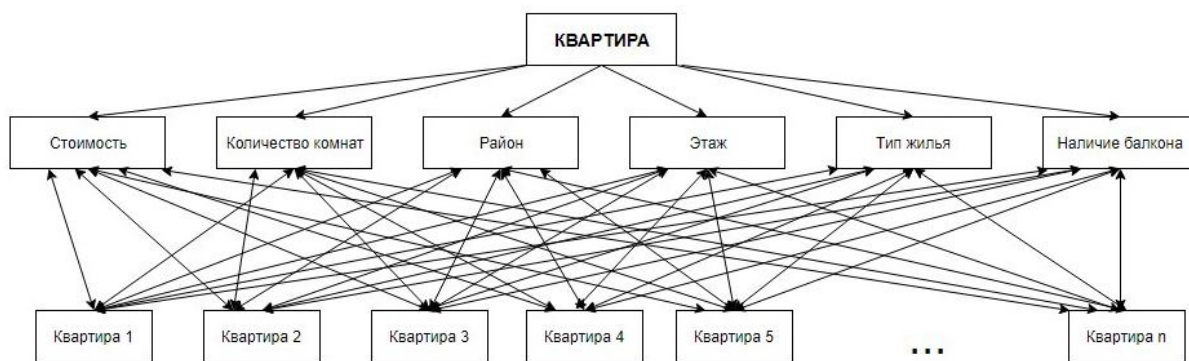


Рисунок 1.1 – Иерархия для задачи поиска оптимального решения

Следующим этапом после построения иерархии является определение приоритетов всех узлов структуры, представляющих относительную важность или предпочтительность узлов построенной иерархической модели. Узлом называются все элементы иерархии, то есть это все возможные решения (альтернативы), главная цель и все факторы. Под структурой принятия решения понимают совокупность всех узлов, сгруппированных по уровням, и всех связей между узлами, другими словами это и есть весь направленный граф.

Приоритетами называют числа, связанные с узлами иерархии и представляющие собой относительные веса элементов в каждой группе, то есть они служат для количественного выражения важности данного узла в кластере относительно остальных узлов кластера в соответствии с критерием, заключенным в вершине кластера. Безразмерные величины приоритетов, принимающие значения от 0 до 1, позволяют обоснованно сравнивать разнородные факторы, что является отличительной особенностью метода анализа иерархий[23]. Сумма приоритетов элементов, подчиненных одному элементу вышележащего уровня иерархии, равна 1. Приоритет цели по определению равен 1.

Информация для расстановки приоритетов узнается у клиента и математически обрабатывается с помощью процедуры попарных сравнений, то есть критерии сравниваются попарно по отношению к цели, а альтернативы

сравниваются попарно по отношению к каждому их критериев. На рисунке 1.2 изображена схема построения матриц попарных сравнений.

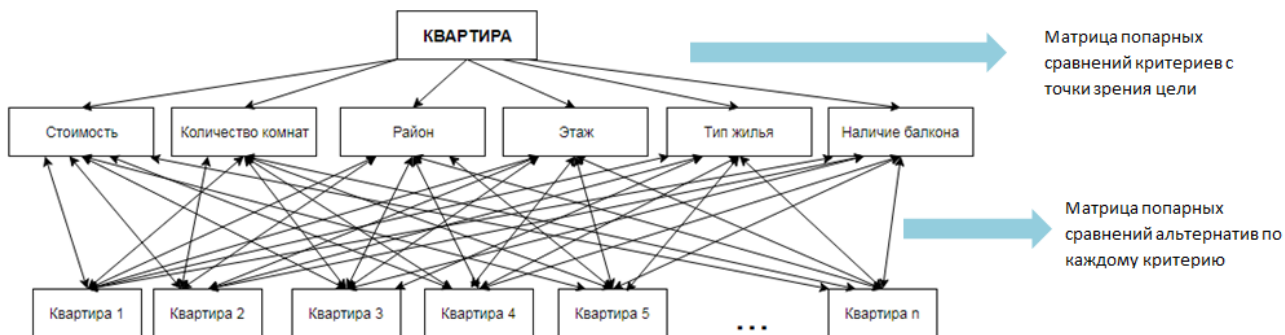


Рисунок 1.2 – Схема построения матриц попарных сравнений

Во время процедуры сравнения составляются матрицы попарных сравнений: одна матрица - для критериев,  $n$  матриц – для альтернатив, где  $n$  отображает количество критериев. На рисунке 1.3 показана матрица попарных сравнений критериев по отношению к цели, а на рисунке 1.4 показана матрица попарных сравнений альтернатив по отношению к  $n$ -ому критерию [4].

	$K_1$	$K_2$	...	$K_n$
$K_1$	1	8	...	3
$K_2$	1/8	1	...	1/5
...	...	...	1	
$K_n$	1/3	5	...	1

Рисунок 1.3 – Матрица попарных сравнений критериев по отношению к цели

$K_n$	$B_1$	$B_2$	...	$B_k$
$B_1$	1	2	...	1/6
$B_2$	1/2	1	...	1/7
...	...	...	1	
$B_k$	6	7	...	1

Рисунок 1.4 – Матрица попарных сравнений альтернатив по отношению к первому критерию

Из значений, расставленных клиентом, составляется матрица размером  $n \times n$ . Значения в матрицах парного сравнения отражают относительную значимость одного объекта над другим, находящимся на этом же уровне. Два объекта, находящиеся на одном уровне, сравниваются по своей относительной значимости для одного объекта высшего уровня по школе Саати, и результаты записываются в обратно симметричную матрицу. Данная шкала имеет 9 степеней предпочтения, имеющие следующий смысл:

- одинаково предпочтительны = 1;
- немного лучше (хуже) = 3 (1/3);
- лучше (хуже) = 5 (1/5);
- значительно лучше (хуже) = 7 (1/7);
- принципиально лучше (хуже) = 9 (1/9)
- при промежуточном мнении используются промежуточные баллы 2, 4, 6 или 8.

Числа из этой шкалы используются, чтобы показать, во сколько раз элемент с большей оценкой предпочтительности доминирует над элементом с меньшей оценкой относительно общего для них критерия или свойства. Таким образом, величина приоритета напрямую связана с оптимальностью решения.

Элементы матрицы рассчитываются относительно расставленных клиентов приоритетов. Составление матрицы происходит с использованием следующей математической модели: элементы главной диагонали всегда будут равны единицы, то есть  $a_{ii}=1$ , так как каждый критерий равен себе по важности; элементы, стоящие выше главной диагонали будут рассчитываться по формуле (1.1), а элементы, стоящие ниже главной диагонали будут вычисляться согласно формуле (1.2).

$$a_{ij} = \begin{cases} p_i - p_j, & \text{если } p_i > p_j \\ \frac{1}{p_j - p_i}, & \text{если } p_i < p_j \\ 1, & \text{если } p_i = p_j \end{cases}, \quad (1.1)$$

где

$a_{ij}$  – элемент матрицы,

$p_i, p_j$  – приоритеты, расставленные клиентом.

$$a_{ji} = \frac{1}{a_{ij}}, \quad (1.2)$$

где

$a_{ij}, a_{ji}$  – элементы матрицы.

То есть в процессе заполнения матрицы если элемент  $i$  важнее элемента  $j$ , то клетка  $(i, j)$ , соответствующая строке  $i$  и столбцу  $j$ , заполняется целым числом, а клетка  $(j, i)$ , соответствующая строке  $j$  и столбцу  $i$ , заполняется обратным числом (дробью). Заполнение таблицы проводится построчно. Чем важнее критерий, тем больше целочисленных оценок будет в соответствующей ему строке матрицы.

Матрица для сравнения альтернатив строиться таким же образом, то есть составляются аналогичные матрицы сравнения вариантов (альтернатив) по каждому критерию. Анализ матриц будет происходить следующим образом: находим сумму элементов каждого столбца по формуле (1.3), затем делим все элементы матрицы на полученную сумму элементов соответствующего столбца, как показано в формуле (1.4).

$$S_j = a_{1j} + a_{2j} + \dots + a_{nj}, \quad (1.3)$$

где

$S_j$  – сумма элементов по столбцам,

$a_{1j}, a_{2j}, \dots, a_{nj}$  – элементы матрицы попарного сравнения.

$$A_{ij} = \frac{a_{ij}}{S_j}, \quad (1.4)$$

где

$A_{ij}$  – нормированное значение элементов матрицы.

Таким образом, произвелась нормировка значений матрицы. Далее находим «веса» по формуле (1.5).



$$\omega_i = \sum_{j=1}^n \frac{A_{ij}}{n}, \quad (1.5)$$

где

$n$  – количество столбцов матрицы,

$\omega_i$  - среднее значение.

Полученные значения задают «веса» критериев с точки зрения поставленной цели. Эти значения – матрицу – также еще называют весовым столбцом критериев по цели. Затем повторяем те же действия для матриц попарного сравнения по критериям. В итоге получаем столбцы – значения – весовых коэффициентов объектов сравнения с точки зрения соответствия отдельным критериям.

Для того чтобы определить веса альтернатив необходимо умножить полученную на последнем шаге матрицу весовых коэффициентов объектов сравнения с точки зрения соответствия отдельным критериям на вектор, характеризующий веса критериев с точки зрения поставленной цели. В результате получатся веса альтернатив с точки зрения достижения цели. Альтернатива с наибольшим весом, считается самой выгодной. Но так как клиенту необходимо предоставить ни один конкретный вариант, а список вариантов в порядке убывания их соответствия его требованиям, то, следовательно, результатом работы алгоритма будет отсортированный список предложений в порядке убывания весов альтернатив.

Для дальнейшей проверки правильности вычислений, реализованных программным путем, рассчитаем вручную вычисления математической модели. Для наглядности вычислим для 16 риэлтерских предложений. На рисунке 1.5 представлены тестовые данные.

ID	VALUE	NUMBER_OF_ROOMS	DISTRICT	FLOOR	TYPE_OF_APARTMENT	IS_HAS_BALCONY	IS_HAS_ELEVATOR	IS_HAS_FURNITURE	ADDRESS	PHONE	CLIENT
1	3100000	2	Автозаводской	3	вторичный	да	да	нет	Спортивная, 17А	89024859602	Пудожников Ю. В.
2	1230000	1	Центральный	3	вторичный	нет	нет	да	Воздвиженская, 20	89094827153	Игошина Е. Г.
3	1300000	1	Ставропольский	1	вторичный	нет	нет	нет	Ореховый проезд, д. 1	89934857296	Созов Д. О.
4	5100000	3	Автозаводской	6	вторичный	да	да	да	Маршала Жукова ул, д. 8	89839022851	Пирогов Е. Ф.
5	2050000	3	Автозаводской	5	вторичный	да	да	нет	Свердлова ул, д. 68	89638491475	Сергеева И. Н.
6	5500000	4	Автозаводской	3	вторичный	да	нет	нет	Кулибина б-р, д. 6а	89092437281	Ватрушин Л. У.
7	1790000	1	Центральный	2	вторичный	да	да	нет	Мира 5-12	89003762156	Овечкина М. В.
8	8000	1	Шлюзовой	5	вторичный	да	нет	да	Зеленая, 6	89078543184	Дятлов Г. Л.
9	14000	2	Центральный	9	вторичный	да	да	да	Банькина, 28	89067853278	Морозова О. О.
10	10000	2	Комсомильский	1	вторичный	нет	нет	да	Мурысева, 91	89017483926	Алексеева А.А.
11	3100000	2	Центральный	3	первичный	да	да	нет	Комсомольская улица, 82	89127653153	Котов Г.В.
12	2500000	1	Автозаводской	5	первичный	да	да	да	40 лет Победы, 43А	89028475193	Егорова А. Г.
13	2500000	2	Автозаводской	5	вторичный	да	да	да	41 лет Победы, 43А	89935728173	Ульянов А.Н.
14	1999000	2	Автозаводской	8	первичный	да	да	нет	улица 40 лет Победы, 47	89425892057	Мельникова Е.Е.
15	3250000	4	Автозаводской	13	первичный	да	да	нет	проезд Оптимистов, 7	89345829478	Авенчук О.Я.
16	2313022	2	Центральный	5	первичный	да	да	нет	ул. Полякова	89324752298	Ковродкин И.И.

Рисунок 1.5 – Информация о рассматриваемых предложениях

Предположим, что клиент расставит такие приоритеты критериям поиска: стоимость – 5; количество комнат: 6; район – 9; этаж – 2; тип жилья – 3; наличие балкона – 1. Для расчета вычислений будем использовать Excel. Воспользовавшись формулой (1.1) вычислим элементы, стоящие над главной диагональю. На рисунке 1.6 представлен результат вычислений.

	Стоимость	Кол Комнат	Район	Этаж	Тип жилья	Налич балк
Стоимость	1	0,5	0,25	3	2	4
Кол Комнат		1	0,333333	4	3	5
Район			1	7	6	8
Этаж				1	0,5	2
Тип жилья					1	2
Налич балк						1

Рисунок 1.6 – Результат заполнения согласно формуле (1.1)

Далее воспользовавшись формулой (1.2) вычислим значения, находящиеся ниже главной диагонали. На рисунке 1.7 представлен результат вычислений.

	Стоимость	Кол Комнат	Район	Этаж	Тип жилья	Налич балк
Стоимость	1	0,5	0,25	3	2	4
Кол Комнат	2	1	0,333333	4	3	5
Район	4	3	1	7	6	8
Этаж	0,33333333	0,25	0,14286	1	0,5	2
Тип жилья	0,5	0,333333333	0,16667	2	1	2
Налич балк	0,25	0,2	0,125	0,5	0,5	1

Рисунок 1.7 – Результат вычислений согласно формуле (1.2)

Затем необходимо нормировать значения, для этого воспользуемся формулами (1.3) и (1.4). На рисунке 1.8 и 1.9 представлены результаты данных вычислений.

	Стоимость	Кол Комнат	Район	Этаж	Тип жилья	Налич балк
Стоимость	1	0,5	0,25	3	2	4
Кол Комнат	2	1	0,333333	4	3	5
Район	4	3	1	7	6	8
Этаж	0,33333333	0,25	0,14286	1	0,5	2
Тип жилья	0,5	0,333333333	0,16667	2	1	2
Налич балк	0,25	0,2	0,125	0,5	0,5	1
$S_j$	8,08333333	5,283333333	2,01786	17,5	13	22

Рисунок 1.8 – Результат вычислений согласно формуле (1.3)

	Стоимость	Кол Комнат	Район	Этаж	Тип жилья	Налич балк
Стоимость	0,1237113	0,094637224	0,12389	0,171429	0,1538462	0,18181818
Кол Комнат	0,2474227	0,189274448	0,16519	0,228571	0,2307692	0,22727273
Район	0,4948454	0,567823344	0,49558	0,4	0,4615385	0,36363636
Этаж	0,0412371	0,047318612	0,0708	0,057143	0,0384615	0,09090909
Тип жилья	0,0618557	0,063091483	0,0826	0,114286	0,0769231	0,09090909
Налич балк	0,0309278	0,03785489	0,06195	0,028571	0,0384615	0,04545455

Рисунок 1.9 – Нормированные значения матрицы

После нормировки значений матрицы, найдем веса критериев относительно поставленной цели, для этого воспользуемся формулой (1.5). На рисунке 1.10 отображены веса критериев поиска.

	Стоимость	Кол Комнат	Район	Этаж	Тип жилья	Налич балк	$w_i$
Стоимость	0,1237113	0,094637224	0,12389	0,171429	0,1538462	0,18181818	0,141555879
Кол Комнат	0,2474227	0,189274448	0,16519	0,228571	0,2307692	0,22727273	0,214750376
Район	0,4948454	0,567823344	0,49558	0,4	0,4615385	0,36363636	0,463903125
Этаж	0,0412371	0,047318612	0,0708	0,057143	0,0384615	0,09090909	0,057644279
Тип жилья	0,0618557	0,063091483	0,0826	0,114286	0,0769231	0,09090909	0,081610151
Налич балк	0,0309278	0,03785489	0,06195	0,028571	0,0384615	0,04545455	0,04053619

Рисунок 1.10 – «Веса» критериев поиска относительно цели

Далее предположим, что клиент расставил следующие уточняющие приоритеты:

- количество комнат: однокомнатная – 3; двухкомнатная – 5; трехкомнатная – 7; четырехкомнатная – 9;
- район: автозаводской – 9; центральный – 7; ставропольский – 3; шлюзовой – 4;
- тип жилья: первичный – 7; вторичный – 9;
- наличие балкона: да – 9; нет – 5.

Воспользовавшись Excel вычислим по формулам (1.1) и (1.2) значения матриц альтернатив относительно критериев. На рисунке 1.11 можно увидеть полученную матрицу для критерия стоимость, на рисунке 1.12 – количество комнат, на рисунке 1.13 – район, на рисунке 1.14 – этаж, на рисунке 1.15 – тип жилья и на рисунке 1.16 – наличие балкона.

Стоимость	Кв_1	Кв_2	Кв_3	Кв_4	Кв_5	Кв_6	Кв_7	Кв_8	Кв_9	Кв_10	Кв_11	Кв_12	Кв_13	Кв_14	Кв_15	Кв_16
Кв_1	1	5	4	0,5	2	0,5	3	2	0,5	1	1	2	2	2	1	2
Кв_2	0,2	1	0,5	0,142857	0,333333	0,142857	0,5	0,333333	0,142857	0,2	0,2	0,25	0,25	0,333333	0,2	0,25
Кв_3	0,25	2	1	0,166667	0,5	0,166667	0,5	0,5	0,166667	0,25	0,25	0,333333	0,333333	0,5	0,25	0,333333
Кв_4	2	7	6	1	4	1	5	4	1	2	2	3	3	4	2	3
Кв_5	0,5	3	2	0,25	1	0,25	2	1	0,25	0,5	0,5	0,5	0,5	1	0,5	0,5
Кв_6	2	7	6	1	4	1	5	4	1	2	2	3	3	4	2	3
Кв_7	0,333333	2	2	0,2	0,5	0,2	1	0,5	0,2	0,333333	0,333333	0,5	0,5	0,5	0,333333	0,5
Кв_8	0,5	3	2	0,25	1	0,25	2	1	0,25	0,5	0,5	0,5	0,5	1	0,5	0,5
Кв_9	2	7	6	1	4	1	5	4	1	2	2	3	3	4	2	3
Кв_10	1	5	4	0,5	2	0,5	3	2	0,5	1	1	2	2	2	1	2
Кв_11	1	5	4	0,5	2	0,5	3	2	0,5	1	1	2	2	2	1	2
Кв_12	0,5	4	3	0,333333	2	0,333333	2	2	0,333333	0,5	0,5	1	1	2	0,5	1
Кв_13	0,5	4	3	0,333333	2	0,333333	2	2	0,333333	0,5	0,5	1	1	2	0,5	1
Кв_14	0,5	3	2	0,25	1	0,25	2	1	0,25	0,5	0,5	0,5	0,5	1	0,5	0,5
Кв_15	1	5	4	0,5	2	0,5	3	2	0,5	1	1	2	2	2	1	2
Кв_16	0,5	4	3	0,333333	2	0,333333	2	2	0,333333	0,5	0,5	1	1	2	0,5	1

Рисунок 1.11 – Матрица попарных сравнений альтернатив согласно критерию поиска «Стоимость»

Кол комн	Кв_1	Кв_2	Кв_3	Кв_4	Кв_5	Кв_6	Кв_7	Кв_8	Кв_9	Кв_10	Кв_11	Кв_12	Кв_13	Кв_14	Кв_15	Кв_16
Кв_1	1	2	2	0,5	0,5	0,25	2	2	1	1	1	2	1	1	0,25	1
Кв_2	0,5	1	1	0,25	0,25	0,166667	1	1	0,5	0,5	0,5	1	0,5	0,5	0,166667	0,5
Кв_3	0,5	1	1	0,25	0,25	0,166667	1	1	0,5	0,5	0,5	1	0,5	0,5	0,166667	0,5
Кв_4	2	4	4	1	1	0,5	4	4	2	2	2	4	2	2	0,5	2
Кв_5	2	4	4	1	1	0,5	4	4	2	2	2	4	2	2	0,5	2
Кв_6	4	6	6	2	2	1	6	6	4	4	4	6	4	4	1	4
Кв_7	0,5	1	1	0,25	0,25	0,166667	1	1	0,5	0,5	0,5	1	0,5	0,5	0,166667	0,5
Кв_8	0,5	1	1	0,25	0,25	0,166667	1	1	0,5	0,5	0,5	1	0,5	0,5	0,166667	0,5
Кв_9	1	2	2	0,5	0,5	0,25	2	2	1	1	1	2	1	1	0,25	1
Кв_10	1	2	2	0,5	0,5	0,25	2	2	1	1	1	2	1	1	0,25	1
Кв_11	1	2	2	0,5	0,5	0,25	2	2	1	1	1	2	1	1	0,25	1
Кв_12	0,5	1	1	0,25	0,25	0,166667	1	1	0,5	0,5	0,5	1	0,5	0,5	0,166667	0,5
Кв_13	1	2	2	0,5	0,5	0,25	2	2	1	1	1	2	1	1	0,25	1
Кв_14	1	2	2	0,5	0,5	0,25	2	2	1	1	1	2	1	1	0,25	1
Кв_15	4	6	6	2	2	1	6	6	4	4	4	6	4	4	1	4
Кв_16	1	2	2	0,5	0,5	0,25	2	2	1	1	1	2	1	1	0,25	1

Рисунок 1.12 – Матрица попарных сравнений альтернатив согласно критерию поиска «Количество комнат»

Район	Кв_1	Кв_2	Кв_3	Кв_4	Кв_5	Кв_6	Кв_7	Кв_8	Кв_9	Кв_10	Кв_11	Кв_12	Кв_13	Кв_14	Кв_15	Кв_16
Кв_1	1	2	6	1	1	1	2	5	2	4	2	1	1	1	1	2
Кв_2	0,5	1	4	0,5	0,5	0,5	1	3	1	2	1	0,5	0,5	0,5	0,5	1
Кв_3	0,166667	0,25	1	0,166667	0,166667	0,166667	0,25	0,5	0,25	0,5	0,25	0,166667	0,166667	0,166667	0,166667	0,25
Кв_4	1	2	6	1	1	1	2	5	2	4	2	1	1	1	1	2
Кв_5	1	2	6	1	1	1	2	5	2	4	2	1	1	1	1	2
Кв_6	1	2	6	1	1	1	2	5	2	4	2	1	1	1	1	2
Кв_7	0,5	1	4	0,5	0,5	0,5	1	3	1	2	1	0,5	0,5	0,5	0,5	1
Кв_8	0,2	0,333333	2	0,2	0,2	0,2	0,333333	1	0,333333	0,5	0,333333	0,2	0,2	0,2	0,2	0,333333
Кв_9	0,5	1	4	0,5	0,5	0,5	1	3	1	2	1	0,5	0,5	0,5	0,5	1
Кв_10	0,25	0,5	2	0,25	0,25	0,25	0,5	2	0,5	1	0,5	0,25	0,25	0,25	0,25	0,5
Кв_11	0,5	1	4	0,5	0,5	0,5	1	3	1	2	1	0,5	0,5	0,5	0,5	1
Кв_12	1	2	6	1	1	1	2	5	2	4	2	1	1	1	1	2
Кв_13	1	2	6	1	1	1	2	5	2	4	2	1	1	1	1	2
Кв_14	1	2	6	1	1	1	2	5	2	4	2	1	1	1	1	2
Кв_15	1	2	6	1	1	1	2	5	2	4	2	1	1	1	1	2
Кв_16	0,5	1	4	0,5	0,5	0,5	1	3	1	2	1	0,5	0,5	0,5	0,5	1

Рисунок 1.13 – Матрица попарных сравнений альтернатив согласно критерию поиска «Район»

Этаж	Кв_1	Кв_2	Кв_3	Кв_4	Кв_5	Кв_6	Кв_7	Кв_8	Кв_9	Кв_10	Кв_11	Кв_12	Кв_13	Кв_14	Кв_15	Кв_16
Кв_1	1	1	3	0,5	0,5	1	2	1	0,5	3	2	0,5	0,5	0,333333	0,5	0,5
Кв_2	1	1	3	0,5	0,5	1	2	1	0,5	3	2	0,5	0,5	0,333333	0,5	0,5
Кв_3	0,333333	0,333333	1	0,25	0,2	0,333333	0,5	0,333333	0,25	1	0,5	0,2	0,2	0,166667	0,25	0,2
Кв_4	2	2	4	1	0,5	2	2	2	1	4	2	0,5	0,5	0,5	1	0,5
Кв_5	2	2	5	2	1	2	3	2	2	5	3	1	1	0,5	2	1
Кв_6	1	1	3	0,5	0,5	1	2	1	0,5	3	2	0,5	0,5	0,333333	0,5	0,5
Кв_7	0,5	0,5	2	0,5	0,333333	0,5	1	0,5	0,5	2	1	0,333333	0,333333	0,25	0,5	0,333333
Кв_8	1	1	3	0,5	0,5	1	2	1	0,5	3	2	0,5	0,5	0,333333	0,5	0,5
Кв_9	2	2	4	1	0,5	2	2	2	1	4	2	0,5	0,5	0,5	1	0,5
Кв_10	0,333333	0,333333	1	0,25	0,2	0,333333	0,5	0,333333	0,25	1	0,5	0,2	0,2	0,166667	0,25	0,2
Кв_11	0,5	0,5	2	0,5	0,333333	0,5	1	0,5	0,5	2	1	0,333333	0,333333	0,25	0,5	0,333333
Кв_12	2	2	5	2	1	2	3	2	2	5	3	1	1	0,5	2	1
Кв_13	2	2	5	2	1	2	3	2	2	5	3	1	1	0,5	2	1
Кв_14	3	3	6	2	2	3	4	3	2	6	4	2	2	1	2	2
Кв_15	2	2	4	1	0,5	2	2	2	1	4	2	0,5	0,5	0,5	1	0,5
Кв_16	2	2	5	2	1	2	3	2	2	5	3	1	1	0,5	2	1

Рисунок 1.14 – Матрица попарных сравнений альтернатив согласно критерию поиска «Этаж»

Тип жил	Кв_1	Кв_2	Кв_3	Кв_4	Кв_5	Кв_6	Кв_7	Кв_8	Кв_9	Кв_10	Кв_11	Кв_12	Кв_13	Кв_14	Кв_15	Кв_16
Кв_1	1	1	1	1	1	1	1	1	1	1	0,5	0,5	1	0,5	0,5	0,5
Кв_2	1	1	1	1	1	1	1	1	1	1	0,5	0,5	1	0,5	0,5	0,5
Кв_3	1	1	1	1	1	1	1	1	1	1	0,5	0,5	1	0,5	0,5	0,5
Кв_4	1	1	1	1	1	1	1	1	1	1	0,5	0,5	1	0,5	0,5	0,5
Кв_5	1	1	1	1	1	1	1	1	1	1	0,5	0,5	1	0,5	0,5	0,5
Кв_6	1	1	1	1	1	1	1	1	1	1	0,5	0,5	1	0,5	0,5	0,5
Кв_7	1	1	1	1	1	1	1	1	1	1	0,5	0,5	1	0,5	0,5	0,5
Кв_8	1	1	1	1	1	1	1	1	1	1	0,5	0,5	1	0,5	0,5	0,5
Кв_9	1	1	1	1	1	1	1	1	1	1	0,5	0,5	1	0,5	0,5	0,5
Кв_10	1	1	1	1	1	1	1	1	1	1	0,5	0,5	1	0,5	0,5	0,5
Кв_11	2	2	2	2	2	2	2	2	2	2	1	1	2	1	1	1
Кв_12	2	2	2	2	2	2	2	2	2	2	1	1	2	1	1	1
Кв_13	1	1	1	1	1	1	1	1	1	1	0,5	0,5	1	0,5	0,5	0,5
Кв_14	2	2	2	2	2	2	2	2	2	2	1	1	2	1	1	1
Кв_15	2	2	2	2	2	2	2	2	2	2	1	1	2	1	1	1
Кв_16	2	2	2	2	2	2	2	2	2	2	1	1	2	1	1	1

Рисунок 1.15 – Матрица попарных сравнений альтернатив согласно критерию поиска «Тип жилья»

Нал балк	Кв_1	Кв_2	Кв_3	Кв_4	Кв_5	Кв_6	Кв_7	Кв_8	Кв_9	Кв_10	Кв_11	Кв_12	Кв_13	Кв_14	Кв_15	Кв_16
Кв_1	1		4	4	1	1	1	1	1	4	1	1	1	1	1	1
Кв_2	0,25	1	1	1	0,25	0,25	0,25	0,25	0,25	1	0,25	0,25	0,25	0,25	0,25	0,25
Кв_3	0,25	1	1	1	0,25	0,25	0,25	0,25	0,25	1	0,25	0,25	0,25	0,25	0,25	0,25
Кв_4	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1
Кв_5	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1
Кв_6	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1
Кв_7	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1
Кв_8	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1
Кв_9	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1
Кв_10	0,25	1	1	1	0,25	0,25	0,25	0,25	0,25	1	0,25	0,25	0,25	0,25	0,25	0,25
Кв_11	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1
Кв_12	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1
Кв_13	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1
Кв_14	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1
Кв_15	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1
Кв_16	1	4	4	1	1	1	1	1	1	4	1	1	1	1	1	1

Рисунок 1.16 – Матрица попарных сравнений альтернатив согласно критерию поиска «Наличие балкона»

На рисунке 1.17 представлены нормированные значения матриц согласно формуле (1.4) и отображены рассчитанные по формуле (1.5) веса альтернатив согласно критерию «Стоимость», на рисунке 1.18 – критерию «Количество комнат», на рисунке 1.19 – «Район», на рисунке 1.20 – «Этаж», на рисунке 1.21 – «Тип жилья» и на рисунке 1.22 – «Наличие балкона».

Стоимост	Кв_1	Кв_2	Кв_3	Кв_4	Кв_5	Кв_6	Кв_7	Кв_8	Кв_9	Кв_10	Кв_11	Кв_12	Кв_13	Кв_14	Кв_15	Кв_16	w <sub>i</sub>
Кв_1	0,072551	0,074627	0,07619	0,068875	0,065934	0,068875	0,073171	0,065934	0,068875	0,072551	0,072551	0,088561	0,088561	0,065934	0,072551	0,088561	0,074019
Кв_2	0,01451	0,014925	0,009524	0,019679	0,010989	0,019679	0,012195	0,010989	0,019679	0,01451	0,01451	0,01107	0,01107	0,010989	0,01451	0,01107	0,0137437
Кв_3	0,018138	0,029851	0,019048	0,022958	0,016484	0,022958	0,012195	0,016484	0,022958	0,018138	0,018138	0,01476	0,01476	0,016484	0,018138	0,01476	0,0185157
Кв_4	0,145103	0,104478	0,114286	0,13775	0,131868	0,13775	0,121951	0,131868	0,13775	0,145103	0,145103	0,132841	0,132841	0,131868	0,145103	0,132841	0,1330315
Кв_5	0,036276	0,044776	0,038095	0,034438	0,032967	0,034438	0,04878	0,032967	0,034438	0,036276	0,036276	0,02214	0,02214	0,032967	0,036276	0,02214	0,0340868
Кв_6	0,145103	0,104478	0,114286	0,13775	0,131868	0,13775	0,121951	0,131868	0,13775	0,145103	0,145103	0,132841	0,132841	0,131868	0,145103	0,132841	0,1330315
Кв_7	0,024184	0,029851	0,038095	0,02755	0,016484	0,02755	0,02439	0,016484	0,02755	0,024184	0,024184	0,02214	0,02214	0,016484	0,024184	0,02214	0,0242245
Кв_8	0,036276	0,044776	0,038095	0,034438	0,032967	0,034438	0,04878	0,032967	0,034438	0,036276	0,036276	0,02214	0,02214	0,032967	0,036276	0,02214	0,0340868
Кв_9	0,145103	0,104478	0,114286	0,13775	0,131868	0,13775	0,121951	0,131868	0,13775	0,145103	0,145103	0,132841	0,132841	0,131868	0,145103	0,132841	0,1330315
Кв_10	0,072551	0,074627	0,07619	0,068875	0,065934	0,068875	0,073171	0,065934	0,068875	0,072551	0,072551	0,088561	0,088561	0,065934	0,072551	0,088561	0,074019
Кв_11	0,072551	0,074627	0,07619	0,068875	0,065934	0,068875	0,073171	0,065934	0,068875	0,072551	0,072551	0,088561	0,088561	0,065934	0,072551	0,088561	0,074019
Кв_12	0,036276	0,059701	0,057143	0,045917	0,065934	0,045917	0,04878	0,065934	0,045917	0,036276	0,036276	0,04428	0,04428	0,065934	0,036276	0,04428	0,0486951
Кв_13	0,036276	0,059701	0,057143	0,045917	0,065934	0,045917	0,04878	0,065934	0,045917	0,036276	0,036276	0,04428	0,04428	0,065934	0,036276	0,04428	0,0486951
Кв_14	0,036276	0,044776	0,038095	0,034438	0,032967	0,034438	0,04878	0,032967	0,034438	0,036276	0,036276	0,02214	0,02214	0,032967	0,036276	0,02214	0,0340868
Кв_15	0,072551	0,074627	0,07619	0,068875	0,065934	0,068875	0,073171	0,065934	0,068875	0,072551	0,072551	0,088561	0,088561	0,065934	0,072551	0,088561	0,074019
Кв_16	0,036276	0,059701	0,057143	0,045917	0,065934	0,045917	0,04878	0,065934	0,045917	0,036276	0,036276	0,04428	0,04428	0,065934	0,036276	0,04428	0,0486951

Рисунок 1.17 – Веса альтернатив согласно критерию поиска «Стоимость»

Кол комн	Кв_1	Кв_2	Кв_3	Кв_4	Кв_5	Кв_6	Кв_7	Кв_8	Кв_9	Кв_10	Кв_11	Кв_12	Кв_13	Кв_14	Кв_15	Кв_16	w <sub>i</sub>
Кв_1	0,046512	0,051282	0,051282	0,046512	0,046512	0,044776	0,051282	0,051282	0,046512	0,046512	0,046512	0,046512	0,046512	0,044776	0,046512	0,046512	0,0477854
Кв_2	0,023256	0,025641	0,025641	0,023256	0,023256	0,029851	0,025641	0,025641	0,023256	0,023256	0,023256	0,023256	0,025641	0,023256	0,029851	0,023256	0,0248256
Кв_3	0,023256	0,025641	0,025641	0,023256	0,023256	0,029851	0,025641	0,025641	0,023256	0,023256	0,023256	0,023256	0,025641	0,023256	0,029851	0,023256	0,0248256
Кв_4	0,093023	0,102564	0,102564	0,093023	0,093023	0,089552	0,102564	0,102564	0,093023	0,093023	0,093023	0,093023	0,102564	0,093023	0,089552	0,093023	0,0955709
Кв_5	0,093023	0,102564	0,102564	0,093023	0,093023	0,089552	0,102564	0,102564	0,093023	0,093023	0,093023	0,093023	0,102564	0,093023	0,089552	0,093023	0,0955709
Кв_6	0,186047	0,153846	0,153846	0,186047	0,186047	0,179104	0,153846	0,153846	0,186047	0,186047	0,186047	0,186047	0,153846	0,186047	0,179104	0,186047	0,1751161
Кв_7	0,023256	0,025641	0,025641	0,023256	0,023256	0,029851	0,025641	0,025641	0,023256	0,023256	0,023256	0,023256	0,025641	0,023256	0,029851	0,023256	0,0248256
Кв_8	0,023256	0,025641	0,025641	0,023256	0,023256	0,029851	0,025641	0,025641	0,023256	0,023256	0,023256	0,023256	0,025641	0,023256	0,029851	0,023256	0,0248256
Кв_9	0,046512	0,051282	0,051282	0,046512	0,046512	0,044776	0,051282	0,051282	0,046512	0,046512	0,046512	0,046512	0,051282	0,046512	0,044776	0,046512	0,0477854
Кв_10	0,046512	0,051282	0,051282	0,046512	0,046512	0,044776	0,051282	0,051282	0,046512	0,046512	0,046512	0,046512	0,051282	0,046512	0,044776	0,046512	0,0477854
Кв_11	0,046512	0,051282	0,051282	0,046512	0,046512	0,044776	0,051282	0,051282	0,046512	0,046512	0,046512	0,046512	0,051282	0,046512	0,044776	0,046512	0,0477854
Кв_12	0,023256	0,025641	0,025641	0,023256	0,023256	0,029851	0,025641	0,025641	0,023256	0,023256	0,023256	0,023256	0,025641	0,023256	0,029851	0,023256	0,0248256
Кв_13	0,046512	0,051282	0,051282	0,046512	0,046512	0,044776	0,051282	0,051282	0,046512	0,046512	0,046512	0,046512	0,051282	0,046512	0,044776	0,046512	0,0477854
Кв_14	0,046512	0,051282	0,051282	0,046512	0,046512	0,044776	0,051282	0,051282	0,046512	0,046512	0,046512	0,046512	0,051282	0,046512	0,044776	0,046512	0,0477854
Кв_15	0,186047	0,153846	0,153846	0,186047	0,186047	0,179104	0,153846	0,153846	0,186047	0,186047	0,186047	0,186047	0,153846	0,186047	0,179104	0,186047	0,1751161
Кв_16	0,046512	0,051282	0,051282	0,046512	0,046512	0,044776	0,051282	0,051282	0,046512	0,046512	0,046512	0,046512	0,051282	0,046512	0,044776	0,046512	0,0477854

Рисунок 1.18 – Веса альтернатив согласно критерию «Количество комнат»



Нал балк	Кв_1	Кв_2	Кв_3	Кв_4	Кв_5	Кв_6	Кв_7	Кв_8	Кв_9	Кв_10	Кв_11	Кв_12	Кв_13	Кв_14	Кв_15	Кв_16	w <sub>i</sub>
Кв_1	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_2	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182
Кв_3	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182
Кв_4	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_5	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_6	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_7	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_8	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_9	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_10	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182	0,018182
Кв_11	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_12	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_13	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_14	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_15	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727
Кв_16	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727	0,072727

Рисунок 1.22 – Веса альтернатив согласно критерию поиска «Наличие балкона»

Таким образом, матрица весов альтернатив согласно критериям отображена на рисунке 1.23.

Стоимость	Кол ком	Район	Этаж	Тип жил	Нал балк
0,074	0,048	0,089	0,046	0,048	0,073
0,014	0,025	0,046	0,046	0,048	0,018
0,019	0,025	0,013	0,017	0,048	0,018
0,133	0,096	0,089	0,066	0,048	0,073
0,034	0,096	0,089	0,096	0,048	0,073
0,133	0,175	0,089	0,046	0,048	0,073
0,024	0,025	0,046	0,030	0,048	0,073
0,034	0,025	0,017	0,046	0,048	0,073
0,133	0,048	0,046	0,066	0,048	0,073
0,074	0,048	0,024	0,017	0,048	0,018
0,074	0,048	0,046	0,030	0,095	0,073
0,049	0,025	0,089	0,096	0,095	0,073
0,049	0,048	0,089	0,096	0,048	0,073
0,034	0,048	0,089	0,141	0,095	0,073
0,074	0,175	0,089	0,066	0,095	0,073
0,049	0,048	0,046	0,096	0,095	0,073

Рисунок 1.23 – Веса альтернатив согласно критериям поиска

Полученные матрицы весов, отображенные на рисунке 1.9 и 1.23 необходимо перемножить для нахождения «весов» альтернатив согласно цели. На рисунке 1.24 представлен результат перемножений матриц, где в первом столбце числовое значение «веса» альтернативы, а во втором столбце номер альтернативы.



<i>Результат</i>	
0,072	1
0,036	2
0,020	3
0,092	4
0,079	5
0,107	6
0,039	7
0,028	8
0,061	9
0,037	10
0,055	11
0,070	12
0,071	13
0,075	14
0,104	15
0,055	16

Рисунок 1.24 – «Веса» альтернатив согласно цели

Отсортировав предложения по отображенным на рисунке 1.24 «весам» альтернатив относительно цели, получим список отсортированных предложений согласно требованиям клиента. То есть в верху списка будет предложение под номером 6, так как его вес наибольший, следовательно, квартира по адресу «Кулибина б-р, д. ба» будет возглавлять список, а квартира под номером 3 по адресу «Ореховый проспект д. 1» будет завершать список, так как у нее наименьшее значение веса, следовательно, она меньше всего подходит клиенту.

В дальнейшем данные вычисления будут реализованы программным путем. Однако, последовательность вычислений будет такой же, то есть сначала вычисляются «веса» критериев поиска относительно цели, затем вычисляются «веса» альтернатив относительно критериям поиска, а потом перемножив полученные ранее веса, находятся «веса» альтернатив относительно цели.

Таким образом, в данной главе были сформированы теоретические основы поиска решения для обоснования выбора метода исследования, характеризуется процесс поиска объектов недвижимости, рассмотрены методы поиска решения, построена математическая модель поиска решения на основе метода анализа иерархий.

# ГЛАВА 2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ ПОИСКА РЕШЕНИЯ

## 2.1 Моделирование системы поиска решения для риэлтерской компании

Воспользуемся нотацией BPMN для построения модели бизнес-процесса. Данная нотация используется для описания процессов нижнего уровня. Диаграмма процесса в нотации представляет собой алгоритм выполнения процесса. На ней могут быть определены события, исполнители, материальные и документальные потоки, сопровождающие выполнение процесса. Каждый процесс может быть декомпозирован на более низкие уровни[7]. В нотации BPMN выделяют 5 основных категорий элементов: элементы потока (события, процессы и шлюзы), данные (объекты данных и базы данных), соединяющие элементы (потоки управления (действий), потоки сообщений и ассоциации), зоны ответственности (пулы и дорожки) и артефакты (сноски). На рисунке 2.1 отображена модель бизнес-процесса[12].

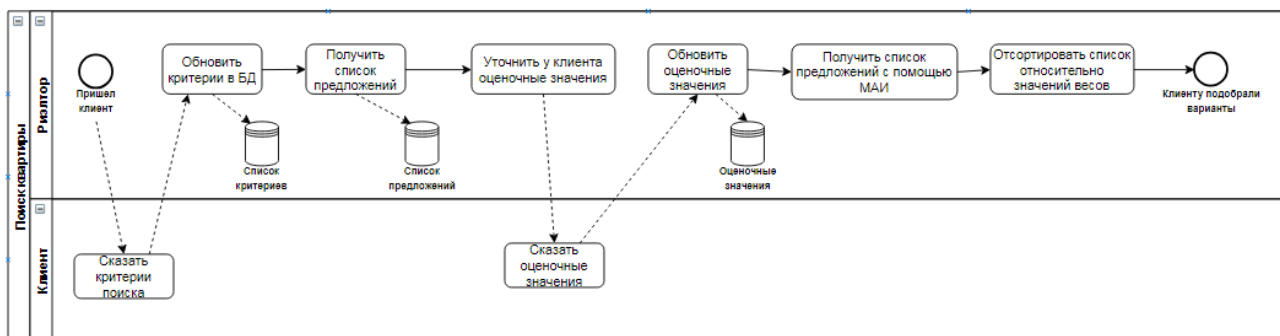


Рисунок 2.1 – Модель бизнес-процесса

В представленной модели BPMN на рисунке 2.1 предполагается использование хранилищ данных. Для хранения необходимой информации будет использоваться база данных MySQL. База данных будет состоять из трех таблиц. Первая таблица «apartments\_tlt» будет отвечать за хранение информации об объектах недвижимости риэлтерской компании. Вторая таблица «estimates» будет хранить оценочную информацию о предложениях. Третья таблица «criteria» хранит названия критериев поиска предложений и числовые значения этих критериев. На рисунке 2.2 представлена модель базы

данных, где отображены таблицы, в которых будет храниться необходимая информация, а также отображены связи между таблицами.

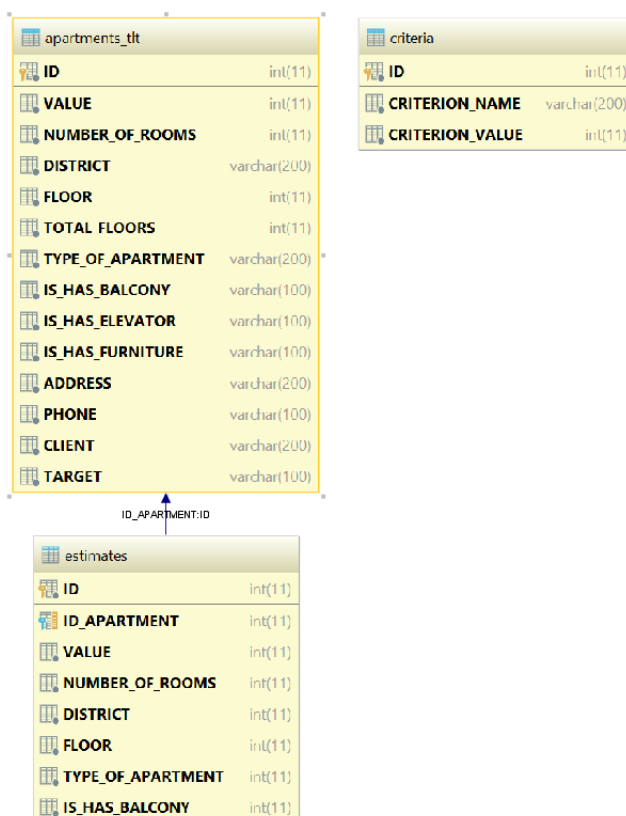


Рисунок 2.2 – Физическая модель данных

Данные для заполнения таблиц в базе данных предоставляется риэлтерской компанией, некоторые уточняются у клиента. Для того чтобы реализовать данное приложение необходимо продумать его алгоритм работы, а также выбрать программные средства с помощью которых он будет реализован.

## 2.2 Разработка алгоритма работы приложения для оптимального поиска объектов недвижимости

Приложение будет разработано с помощью Spring MVC, для взаимодействия с базой данных MySQL будет использоваться Hibernate.

Spring MVC является универсальным фреймворком, ориентированным на запросы, с открытым кодом для Java-платформы. Данный фреймворк построен на принципе модель-представление-контроллер. Модель объединяет данные приложения, в целом они будут состоять из Java-объектов. Вид или

представление отвечает за отображение данных Модели, генерируя обычно HTML-страницу[27]. Задачами контроллера являются обработка запроса пользователя, создание соответствующей Модели и передача её для отображения в Вид.

В разрабатываемом приложении Hibernate будет использоваться для освобождения от значительного объема низкоуровневого программирования при работе с реляционной базой данных. Его можно использовать при проектировании системы классов и таблиц «с нуля», а также с ним можно работать при уже существующей базе данных. Hibernate находится между традиционными Java-объектами и сервером базы данных для обработки всех запросов по сохранению этих объектов на основе соответствующих механизмов и шаблонов [18]. Данная библиотека решит задачу связи Java-классов с таблицами базы данных, и типов данных Java с типами данных SQL, также предоставит средства для автоматической генерации и обновления набора таблиц, построения запросов и обработки полученных данных, значительно уменьшит время разработки. Hibernate будет использоваться для автоматизации генерации SQL-запросов, освобождения от ручной обработки результирующего набора данных и преобразования объектов.

В приложении А представлена диаграмма классов разрабатываемого приложения. Данная диаграмма демонстрирует классы, их атрибуты, методы и взаимосвязи между ними. Так как в результате метода анализа иерархий получают веса альтернатив с точки зрения поставленной цели, относительно которых будет сортироваться лист приложений в порядке убывания весов, то основным методом разрабатываемого приложения будет метод, вычисляющий данные веса. Алгоритм метода вычисления весовых коэффициентов отображен на рисунке 2.3.

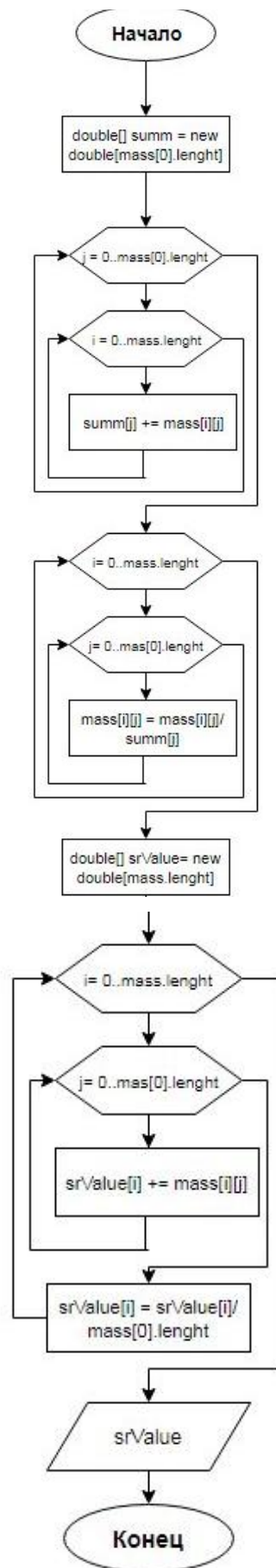


Рисунок 2.3 – Алгоритм вычисления весовых коэффициентов

В данном параграфе были выбраны программные средства, продуманы классы разрабатываемого приложения, их поля, методы, а также взаимодействие между собой. Также был отображен алгоритм основного этапа в алгоритме метода анализа иерархий. Далее необходимо реализовать рассмотренные классы и алгоритм выбранными программными средствами.

### **2.3 Проект работы реализованного приложения поиска объектов недвижимости**

На стартовой странице приложения будет предлагаться расставить предпочтения критериям поиска, проранжировав их. После того, как критерии будут оценены, необходимо будет уточнить некоторую дополнительную информацию о конкретных критериях, нажав кнопку «Уточнить». После того как приоритеты будут выбраны, дополнительная информация уточнена и нажата кнопка «Найти», данные о приоритетах обновятся в таблице базы данных. Таблица «criteria», отвечающая за хранение этих данных, отображена на рисунке 2.4. После того как значения в базе данных обновились, данные из таблицы базы данных конвертируются в модель, представленную в приложение В, то есть происходит сопоставление класса Criterion с таблицей «criteria» и типов данных Java с типами данных sql. Для этого в Hibernate используются аннотации, то есть пометки, с помощью которых указывается компилятору, что делать с участком кода помимо исполнения программы. Аннотация @Entity в приложении В указывает ему, что данный класс является сущностью, @Table показывает с какой именно таблицей базы данных необходимо связать данный класс, @Column определяет, к какому столбцу в таблице базы данных относится конкретное поле класса [21].

Затем с помощью класса CriterionDaoImpl будет обеспечиваться доступ к данным базы данных, то есть с помощью различных методов, отображенных в приложение Г, будет происходить взаимодействие с таблицей базы данных «criteria»[28]. С помощью метода listCriteria() получаем список всех критериев

и их значения, а с помощью метода `updateCriteria()` обновляем значения критерия в базе.

ID	CRITERION_NAME	CRITERION_VALUE
1	Стоимость	4
2	Количество комнат	6
3	Район	3
4	Этаж	4
5	Тип жилья	8
6	Наличие балкона	7

Рисунок 2.4 – Таблица «criteria», отвечающая за хранение числовых приоритетов критериев расставленных клиентом

Информация обо всех предложения риэлтерской компании также хранится в базе данных в таблице «apartments\_tlt». На рисунке 2.5 отображен фрагмент таблицы, в которой хранится данная информация. В приложение Д отображен фрагмент класс `Suggestion`, где происходит сопоставление класса `Suggestion` с таблицей «apartments\_tlt» и их типов данных[11].

ID	VALUE	NUMBER_OF_ROOMS	DISTRICT	FLOOR	TOTAL_FLOORS	TYPE_OF_APARTMENT	IS_HAS_BALCO	IS_HAS_ELEVAT	IS_HAS_FURNI
1	3100000	2	Автозаводской	3	10	вторичный	да	да	нет
2	1230000	1	Центральный	3	3	вторичный	нет	нет	да
3	1300000	1	Ставропольский	1	3	вторичный	нет	нет	нет
4	5100000	3	Автозаводской	6	11	вторичный	да	да	да
5	2050000	3	Автозаводской	5	9	вторичный	да	да	нет
6	5500000	4	Автозаводской	3	5	вторичный	да	нет	нет
7	1790000	1	Центральный	2	10	вторичный	да	да	нет
8	8000	1	Шлюзовой	5	5	вторичный	да	нет	да
9	14000	2	Центральный	9	9	вторичный	да	да	да
10	10000	2	Комсомольский	1	5	вторичный	нет	нет	да
11	3100000	2	Центральный	3	9	первичный	да	да	нет
12	2500000	1	Автозаводской	5	10	первичный	да	да	да
13	2500000	2	Автозаводской	5	10	вторичный	да	да	да
14	1999000	2	Автозаводской	8	17	первичный	да	да	нет
15	3250000	4	Автозаводской	13	17	первичный	да	да	нет

Рисунок 2.5 – Фрагмент таблицы базы данных, отвечающей за хранение информации о квартирах

После того как все необходимые данные были получены от клиента, критерии оценены, дополнительная информация уточнена, и все значения обновились в соответствующих таблицах базы данных, начинаются вычисления

метода анализа иерархий. На рисунке 2.6 отображен фрагмент кода, отвечающий за заполнение матрицы попарных сравнений, относительно которых потом будет производиться попарное сравнение. Матрица будет заполняться следующим образом: элементы, стоящие на главной диагонали всегда будут равны единицы, так как сравниваемые объекты равнозначны самому себе; элементы, стоящие выше главной диагонали будут вызывать метод `countPriority()`, а элементы ниже главной диагонали будут равны обратному значению соответствующего элемента, стоящего выше главной диагонали.

```
public double[][] initMatrixCriterion(List<Criterion> priorityOfCriteriaList) {
    int size = priorityOfCriteriaList.size();
    double[][] matrixPriority = new double[size][size];
    for (int i = 0; i < matrixPriority.length; i++) {
        for (int j = 0; j < matrixPriority[0].length; j++) {
            if (i == j)
                matrixPriority[i][j] = 1;
            if (i < j)
                matrixPriority[i][j] = countPriority(priorityOfCriteriaList.get(i).getCriterionValue(), priorityOfCriteriaList.get(j).getCriterionValue());
        }
    }
    for (int i = 0; i < matrixPriority.length; i++) {
        for (int j = 0; j < matrixPriority[0].length; j++) {
            if (i > j)
                matrixPriority[i][j] = 1 / matrixPriority[j][i];
        }
    }
    return matrixPriority;
}
```

Рисунок 2.6 – Фрагмент кода, отвечающий за заполнение матрицы

На рисунке 2.7 отображен фрагмент кода, отвечающий за простановку приоритетов в матрицу попарного сравнения относительно выставленной клиентом иерархии критериев. Так допустим, приоритет будет равен единице, если выставленные значения степеней предпочтения критериев одинаковые, если разница значений степеней предпочтения будет отрицательной, то возвращается модуль обратного значения, так как превосходство обратное, иначе в результате возвращается разница значений.

Основные вычисления метода анализа иерархий происходят в методе `count()`, в который в качестве параметра передается массив, отвечающий за матрицу, относительно которой происходит попарное сравнение, и вычисляются весовые коэффициенты. На рисунке 2.3 представлен алгоритм данного метода, а на рисунке 2.8 отображена реализация этого алгоритма[2].



```

private double countPriority(int previousValue, int currentValue) {
    double result = previousValue - currentValue;
    if (result == 0)
        return 1;
    if (result == 1)
        return 2;
    if (result > 1 && result < 9)
        return result;
    if (result == -1)
        return 1. / 2;
    return 1 / Math.abs(result);
}

```

Рисунок 2.7 – Фрагмент кода, отвечающий за простановку приоритетов в матрицу попарных сравнений относительно выставленной иерархии

Для нахождения весовых коэффициентов необходимо вначале посчитать сумму по столбцам, затем значения матрицы разделить на сумму элементов в столбце, таким образом, производится нормировка значений матрицы. Далее высчитывается «веса», то есть складываются значения матрицы по строкам, и полученная сумма делится на количество параметров[13]. Затем наибольший весовой коэффициент будет отображать наиболее значимый критерий отбора, либо наиболее лучшую альтернативу по конкретному критерию.

```

public double[] count(double[][] mass) {
    double[] summ = new double[mass[0].length]; //сумма столбцов

    for (int j = 0; j < mass[0].length; j++) {
        summ[j] = 0;
        for (int i = 0; i < mass.length; i++) {
            summ[j] += mass[i][j];
        }
    }

    for (int i = 0; i < mass.length; i++) {
        for (int j = 0; j < mass[0].length; j++)
            mass[i][j] = mass[i][j] / summ[j];
    }

    double[] srZnach = new double[mass.length];

    for (int i = 0; i < mass.length; i++) {
        srZnach[i] = 0;
        for (int j = 0; j < mass[0].length; j++)
            srZnach[i] += mass[i][j];
        srZnach[i] = srZnach[i] / mass[0].length;
    }

    return srZnach;
}

```

Рисунок 2.8 – Описание метода, отвечающего за вычисление «весов»

После нахождения весовых коэффициентов альтернатив по всем критериям необходимо полученную матрицу умножить справа на матрицу весовых коэффициентов критериев. На рисунке 2.9 отображен фрагмент кода, отвечающий за перемножение данных матриц.

```
public double[] countWeightAlternative(double[][] resultMatrix, double[] weightColumnOfCriteriaByPurpose) {
    if (resultMatrix[0].length == weightColumnOfCriteriaByPurpose.length) {
        double[] result = new double[resultMatrix.length];
        // перемножаем матрицы ( матрица весов (квартир по критериям) * матрицу критериев)
        for (int i = 0; i < resultMatrix.length; i++) {
            result[i] = 0;
            for (int j = 0; j < resultMatrix[0].length; j++) {
                result[i] += resultMatrix[i][j] * weightColumnOfCriteriaByPurpose[j];
            }
        }
        return result;
    }
    return null;
}
```

Рисунок 2.9 – Фрагмент кода, отвечающий за перемножение матриц

Если полученную матрицу отсортировать в порядке убывания ее значений, получится список предложений для клиента, в начале которого будут отображаться самые предпочтительные предложения относительно его требованиям.

В приложении Е отображен jsp файл, который отвечает за верстку стартовой страницы, где на 18 строчке подключается эмблема риэлтерской фирмы, на строчках 25-33 отрисовывается меню, на 35-37 строчках подключаются jsp файлы, отвечающие соответственно за отрисовку боковой панели, основного контента страницы и таблицы результатов. Основной контент страницы состоит из информационного и напоминающего сообщения, таблицы критериев и всплывающие окна, где необходимо уточнить некоторые критерии.

В приложении Ж показан фрагмент jsp файл, отвечающий за верстку основного контента страницы, где в начале отрисовывается напоминающее сообщение, затем таблица критериев и рядом с ней информационное сообщение, где расшифровываются значения приоритетов, ниже находятся кнопка «Найти», после нажатия на которую начинаются вычисления метода анализа иерархий, и кнопка «Сброс», после которой значения критерий

обнуляются и приложение можно использовать заново. В jsp файле, отвечающим за отрисовку таблицы предложений после вычислений МАИ рисуется только шапка таблицы, заполнение ее данными происходит в search.js, реализация которого представлена в приложении 3.

Рассмотрим поподробнее реализацию search.js, отображенную в приложении 3. На 6-11 строках описана функция, которая обрабатывает при нажатии на «X» в уточняющих всплывающих окнах. Данная функция обнуляет все проставленные уточняющие приоритеты. На строках 14-28 описана функция, которые обрабатывают при нажатии на кнопку «Сохранить» в уточняющем всплывающем окне относительно критерия «Количество комнат». Данная функция отвечают за обновление оценочных критериев в таблице базы данных. В приложении И отображена верстка всплывающего окна, где необходимо уточнить оценочные значения по критерию «Количество комнат». Остальные всплывающие окна имеют похожее строение. Данное всплывающее окно состоит из заголовка – строка 8, таблицы – строки 14–69, информационного сообщения – строки 70–79 и кнопок «Сохранить» и «Отмена» - строки 82 и 83 соответственно. Названия заголовков во всплывающих окнах зависят от наименования уточняющего критерия. Таблица всегда состоит из двух колонок: в первой колонки перечисляются значения, которые может принимать уточняющий критерий, во второй колонке необходимо выбрать приоритет для конкретного значения. Информационное сообщение и кнопки будут одинаковы во всех уточняющих всплывающих окнах.

На строках 31–33, 35–37 и 40–42 вызывается функция saveEstimates(), код которой представлен на рисунке 2.10. Данная функция сохраняет выбранные значения дополнительных критериев в массив и передает его в функцию updateEstimate(), показанную на рисунке 2.11, где он в виде json объекта передается в EstimateController в соответствующий метод. Определенный метод на контроллере обновляет конкретный параметр в

таблице «estimates» в базе данных. Метод, отвечающие за обновление критерия «Район» в таблице «estimates» представлены на рисунке 2.12.

```
function clean(select) {
  for (var j = 0; j < select.length; j++) {
    select[j].value = 1;
  }
}

function saveEstimates(estimate, estimateValue, path) {
  var certainEstimateValue = estimate,
      certainEstimateMassive = [],
      selectCertainEstimate = estimateValue;

  for (var i = 0; i < certainEstimateValue.length; i++) {
    var certainEstimate = {};
    certainEstimate.criterionName = certainEstimateValue[i].innerText;
    certainEstimate.criterionValue = parseInt(selectCertainEstimate[i].value);

    certainEstimateMassive[i] = certainEstimate;
  }
  updateEstimate(certainEstimateMassive, path);

  clean(estimateValue);
}
```

Рисунок 2.10 – Фрагмент кода, отвечающий за сохранение дополнительных числовых значений

```
function updateEstimate(estimatesMassive, criteriaPath) {
  $.ajax({
    type : "PUT",
    contentType : "application/json",
    url : window.location + criteriaPath,
    data : JSON.stringify(estimatesMassive),
    dataType : 'json',
    async: true,
    success : function(result) {
      console.log(result);
    },
    error : function(e) {
      console.log("ERROR: ", e);
    }
  });
}
```

Рисунок 2.11 – Фрагмент кода, отвечающий за передачу данных на контроллер

```

@Override
@Transactional
public void updateDistrictEstimates(Criterion estimateBalcon, List<Estimate> estimateList, List<Suggestion> suggestionList) {
    for (Suggestion suggestion: suggestionList) {
        for (Estimate estimate: estimateList) {
            if (suggestion.getId() == estimate.getIdApartment()) {
                if (suggestion.getDistrict().equals(estimateBalcon.getCriterionName())) {
                    estimate.setDistrict(estimateBalcon.getCriterionValue());
                    this.estimateDao.update(estimate);
                }
            }
        }
    }
}

```

Рисунок 2.12 – Метод, отвечающие за обновление оценочных значений в базе данных по критерию «Район»

При нажатии на кнопку «Найти», имеющую класс «findSuggestion» и находящуюся под таблицей критериев, выполняется функция, в которой с помощью цикла for критерии со значениями записываются в массив criteriaMassive, который передается в метод updateCriteriaValueAndMakeAlgorithm. Реализация этого метода представлена на рисунке 2.13, где с помощью ajax-запроса массив с обновленными значениями критериев передается по url «window.location + «algorithm»» на MaiController. Реализация данного контроллера показана в приложении К, где метод makeAlgorithm в качестве параметров принимается массив критериев, а возвращается лист отсортированных предложений.

```

function updateCriteriaValueAndMakeAlgorithm(criteriaList) {
    $.ajax({
        type: "PUT",
        contentType: "application/json",
        url: window.location + "algorithm",
        data: JSON.stringify(criteriaList),
        dataType: 'json',
        async: true,
        success: function(listSuggestions) {
            console.log(listSuggestions);
            drawResultTable(listSuggestions);
            $('resultTable')[0].classList.remove('invisible');
        },
        error: function(e) {
            console.log("ERROR: ", e);
        }
    });
}

```

Рисунок 2.13 – Реализация метода updateCriteriaValueAndMakeAlgorithm

В методе `makeAlgorithm()`, представленном в приложении К, происходит обращение к методу сервиса. Реализация сервиса `maiService` показана на рисунке 2.14, где метод `makeAlgorithm()` в качестве параметра принимает массив значений критериев, которые вначале обновляются, затем с помощью сервисов `suggestionService`, `criterionService` и `estimateService` получают соответственно листы предложений, критериев и оценок. Затем данные листы передаются в конструктор класса MAI для дальнейших вычислений метода анализа иерархий.

```
public class MaiService {  
  
    private SuggestionService suggestionService;  
    private CriterionService criterionService;  
    private EstimateService estimateService;  
  
    public void setSuggestionService(SuggestionService suggestionService) {  
        this.suggestionService = suggestionService;  
    }  
  
    public void setCriterionService(CriterionService criterionService) { this.criterionService = criterionService; }  
  
    public void setEstimateService(EstimateService estimateService) { this.estimateService = estimateService; }  
  
    public List<Suggestion> makeAlgorithm(Criterion[] criterionListForUpdate){  
        for(Criterion criteria: criterionListForUpdate){  
            this.criterionService.updateCriteria(criteria);  
        }  
        List<Suggestion> suggestionList = this.suggestionService.listSuggestions();  
        List<Criterion> criterionList = this.criterionService.listCriteria();  
        List<Estimate> estimateList = this.estimateService.listEstimates();  
  
        MAI mai = new MAI(suggestionList, criterionList, estimateList);  
        List<Suggestion> sortedListSuggestions = mai.algorithm();  
  
        return sortedListSuggestions;  
    }  
}
```

Рисунок 2.14 – Реализация `maiService`

После того как метод контроллера `MaiController` вернет лист отсортированных предложений в качестве результата аяк-запроса, вызывается метод `drawResultTable()`, отвечающий за отрисовку таблицы с заполненными данными. Реализация данного метода представлена в приложении Л[1].

После того как значения таблицы заполняются, скрытый ранее блок с таблицей отображается. После нажатия на кнопку «Сброс», имеющую класс «`eraseCriteria`» согласно функции, реализованной на строчках 61-69 в приложении З, блок с таблицей отсортированных предложений скроется, значения критериев сбросятся до 1, и появится вновь информационное сообщение, напоминающее о необходимости расставить приоритетность

критериев. Также в базе данных вернутся все оценочные значения дополнительных критериев в первоначальное дефолтное значение. То есть, на экране появится стартовая страница.

На рисунке 2.15 отображена стартовая страница, где будут расставляться предпочтения критериям поиска, то есть необходимо ранжировать критерии поиска. На данной странице можно увидеть в шапке сайта эмблему риэлтерской компании, справа боковое меню, а в центре страницы основной контент, то есть напоминающее и информационное сообщения, таблица с выпадающим списком возможных значений критериев, которые должны быть выбраны, а также кнопки «Уточнить», при нажатии на которые открываются окна, где требуется расставить дополнительные оценки тем или иным критериям. На рисунках 2.16 – 2.19 представлены окошки, открывающиеся после нажатия на кнопки «Уточнить», и в которых необходимо расставить дополнительные числовые значения, чтобы поиск квартир происходил согласно требованиям клиента.

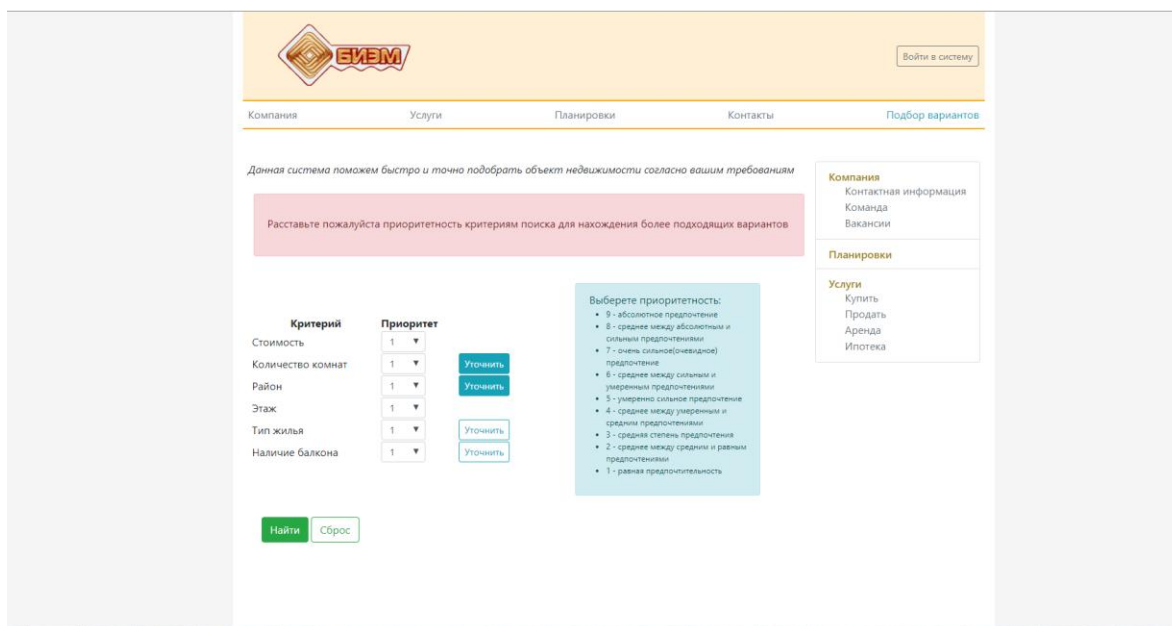


Рисунок 2.15 – Стартовая страница приложения

После нажатия на кнопку «Сохранить» данные значения обновляются в таблице «estimates» в базе данных. Данная таблица отвечает за оценочные значения критериев, и в дальнейшем эти значения будут использоваться при решении метода анализа иерархий. Правее таблицы критериев и кнопок

«Уточнить» отображается информационное сообщение, в котором разъясняются значения приоритетов согласно шкале Саати. Кроме этого на странице имеются кнопки «Найти» и «Сброс».

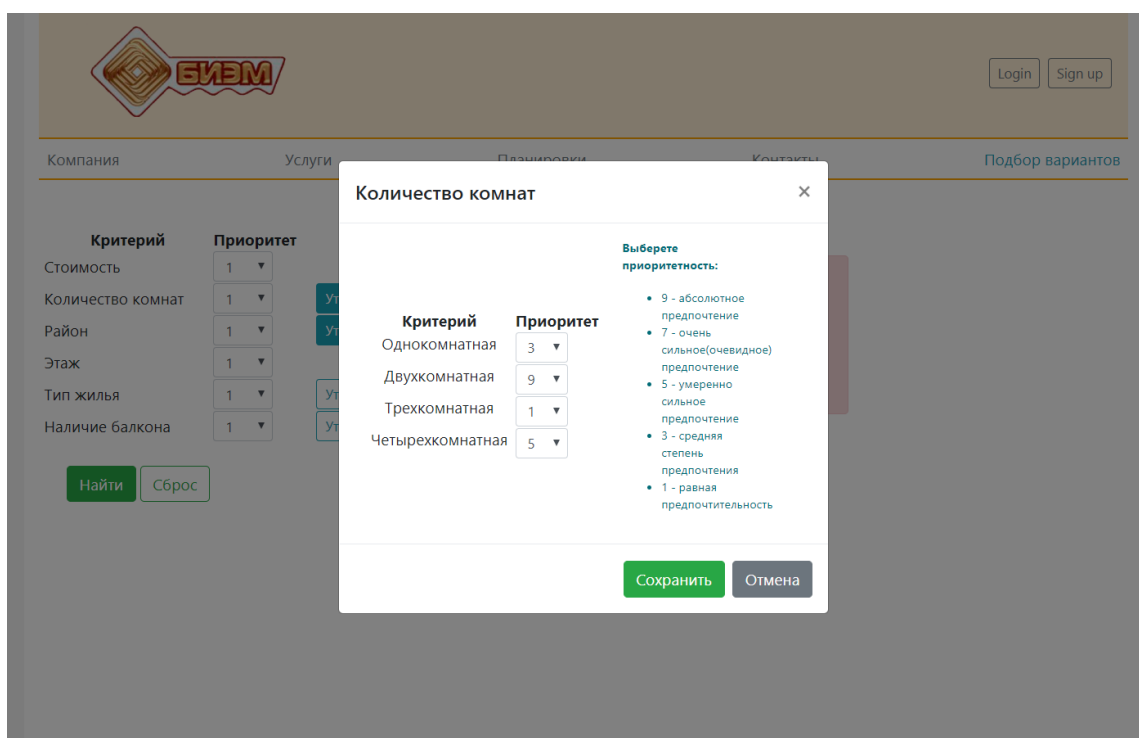


Рисунок 2.16 – Окошко, появляющееся при нажатии «Уточнить» в строке «Количество комнат»

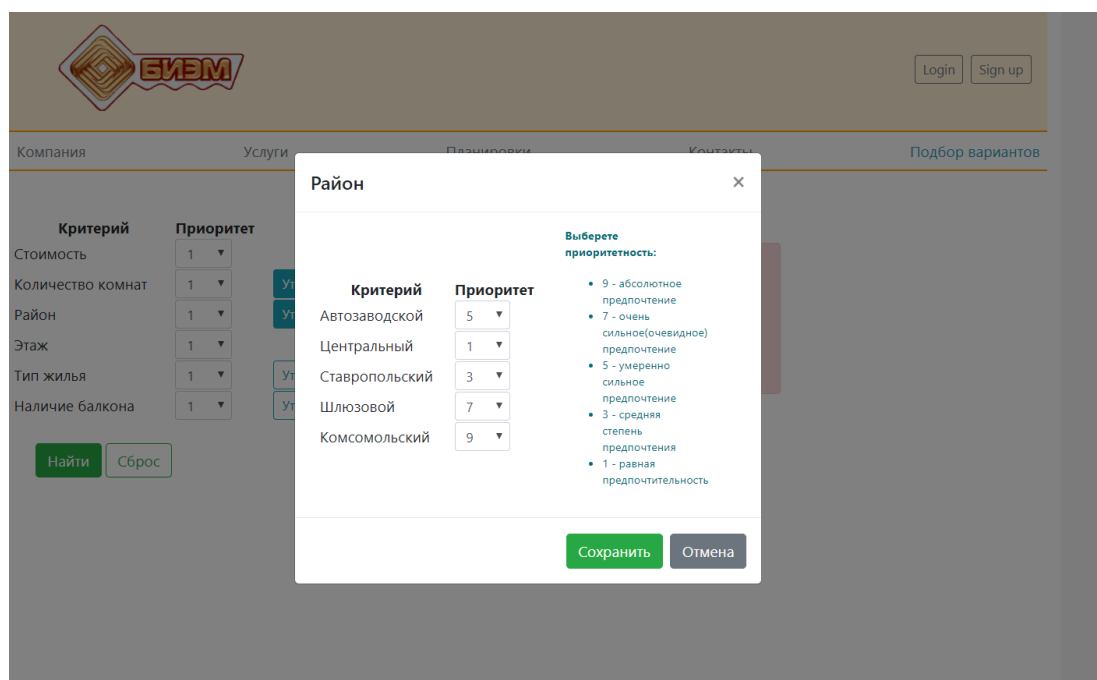


Рисунок 2.17 – Окошко, появляющееся при нажатии «Уточнить» в строке «Район»



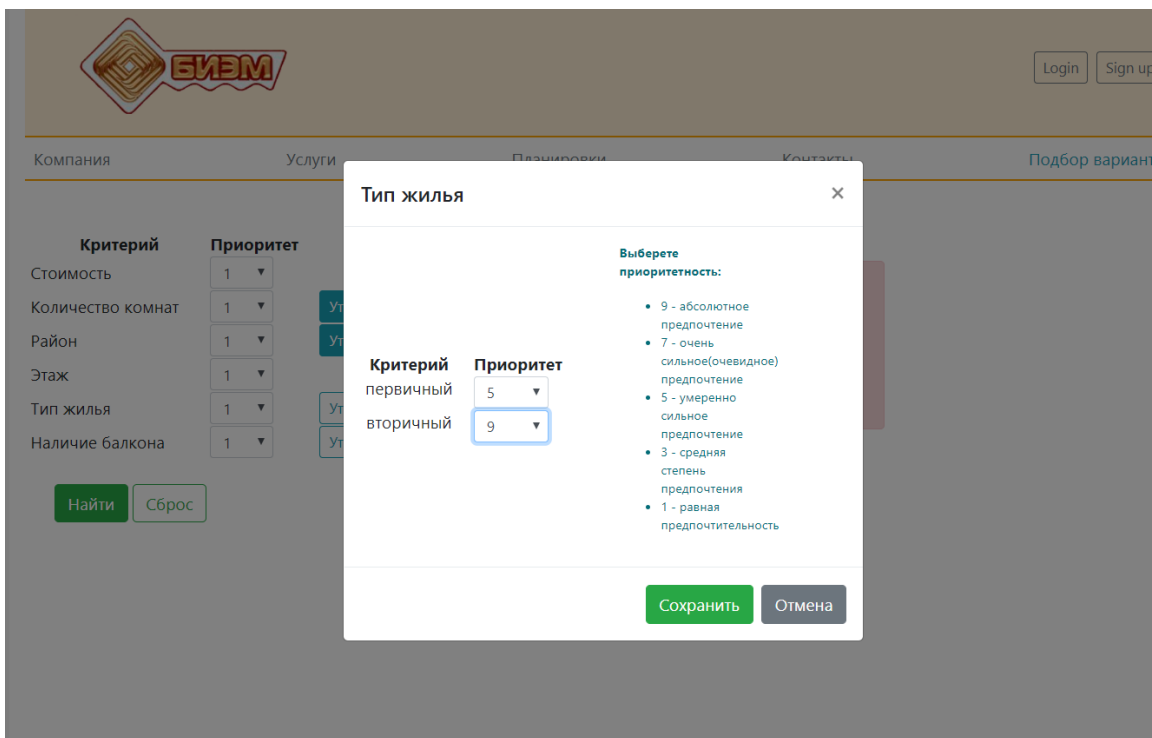


Рисунок 2.18 – Окошко, появляющееся при нажатии «Уточнить» в строке «Тип жилья»

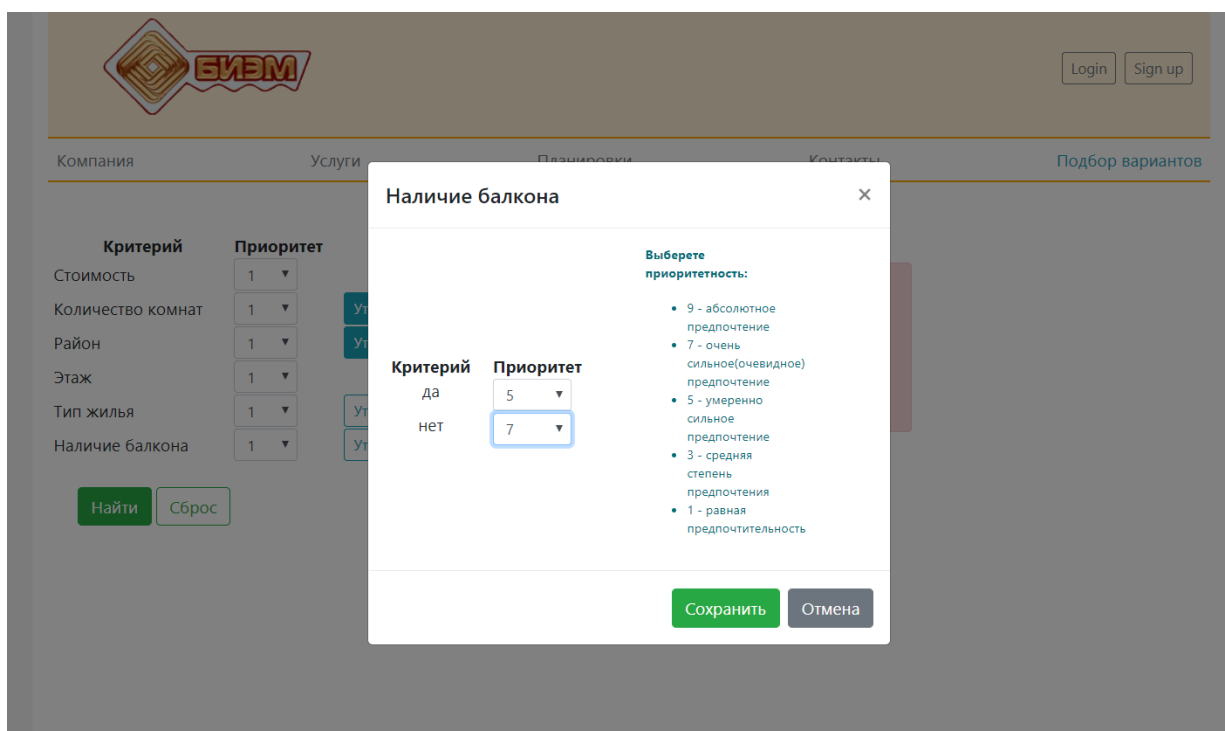


Рисунок 2.19 – Окошко, появляющееся при нажатии «Уточнить» в строке «Наличие балкона»

После проставления приоритетов и нажатие на кнопку «Найти», напоминающее сообщение исчезает, и внизу появляется таблица с сортированными предложениями. На рисунке 2.21 отображен результат поиска. После нажатие на кнопку «Сброс» таблица с предложениями исчезает, выставленные приоритеты обнуляются до 1, и информационное сообщение вновь появляется, таким образом, страница вернулась в стартовый вид, как показано на рисунке 2.15, то есть можно использовать приложение заново.

The screenshot shows a web application interface for real estate search. At the top, there is a navigation bar with the logo 'EUREM' and a 'Войти в систему' button. Below the navigation bar, there are tabs for 'Компания', 'Услуги', 'Планировки', 'Контакты', and 'Подбор вариантов'. The main content area is titled 'Данная система поможет быстро и точно подобрать объект недвижимости согласно вашим требованиям'. On the left, there are search filters for 'Критерий' (Cost, Rooms, District, Floor, Type of housing, Balcony) and 'Приоритет' (Priority) with 'Уточнить' buttons. A central box provides a legend for priority levels from 1 to 7. On the right, there are sections for 'Компания' (Company information, Team, Vacancies) and 'Планировки' (Services: Buy, Sell, Rent, Mortgage). At the bottom, there is a 'Найденные предложения' section with a table of results.

#	Стоимость	Количество комнат	Район	Этаж	Тип жилья	Наличие балкона	Наличие мебели	Адрес	Клиент	Контактная информация
1	5100000	3	Автозаводской	6	вторичный	да	да	Маршала Жукова ул. д. 8	Пирогов Е. Ф.	89039022851
2	5500000	4	Автозаводской	3	вторичный	да	нет	Клибуна Б-р д. 8а	Варушкин Л. Ю.	89062437281
3	3250000	4	Автозаводской	13	первичный	да	нет	просп. Октябрьский, 7	Авечук О.Я.	89348829478
4	1999000	2	Автозаводской	8	первичный	да	нет	40 лет Победы, 43А	Мельникова Е.Е.	89425892057
5	2500000	1	Автозаводской	5	первичный	да	да	40 лет Победы, 43А	Егорова А. Г.	89028475193
6	14200	2	Центральный	9	вторичный	да	да	Бачкина, 28	Морозова О. О.	89078553278
7	2500000	2	Автозаводской	5	вторичный	да	да	41 лет Победы, 43А	Ульянов А.Н.	89939728173

Рисунок 2.20 – Отображение отсортированных предложений

Таким образом, в данной главе была смоделирована система поиска решения для риэлтерской компании, разработан алгоритм работы приложения, а также программно реализована модель поиска решения. Но согласно требованиям заказчика реализованное приложение должно осуществлять оптимальный поиск объектов недвижимости, то есть, следовательно, надо оптимальность реализованного поиска объектов недвижимости на основе метода анализа иерархий.

# ГЛАВА 3 ОЦЕНКА ОПТИМАЛЬНОСТИ РЕАЛИЗОВАННОГО ПОИСКА ОБЪЕКТОВ НЕДВИЖИМОСТИ НА ОСНОВЕ МЕТОДА АНАЛИЗА ИЕРАРХИЙ

## 3.1 Обоснование выбора критериев оптимальности для поиска объектов недвижимости

Нахождение эффективного оптимального решения для клиента, согласно его требованиям – это цель риэлтерской компании. В данном случае, решение – это выбор альтернативы, то есть квартиры.

Обобщенной характеристикой решения является его эффективность. Эта характеристика включает эффект решения, определяющий степень достижения целей, отнесенный к затратам на их достижение. Решение тем эффективнее, чем больше степень достижения целей и меньше затраты на их реализацию.

Оптимизацией является процесс нахождения наилучшего или оптимального решения задачи при заданных критериях. Характеризуя различные методы поиска, сложно выбрать такой один критерий, который бы гарантировал оптимальность метода.

Для нахождения оптимального решения необходимо уточнить понятие принципа или критерия оптимальности. В общем случае, критерием оптимальности считается характерный показатель решения задачи, по значению которого оценивается оптимальность найденного решения, то есть максимальное удовлетворение поставленным требованиям. Правильный выбор критериев играет существенную роль в выборе оптимального решения. В теории принятия решений не найдено общего метода выбора критериев оптимальности [10]. В основном руководствуются опытом или рекомендациями. Проанализировав свою деятельность, желаемый результат, руководство риэлтерской компании «БИЭМ-Недвижимость» предоставила свои критерии оптимальности разрабатываемого приложения. Это результат, удовлетворяющий клиента, полученный за минимально короткое время.

Оптимальным решением считается такое решение, которое по тем или иным признакам предпочтительнее других. Однако, нахождение реальных оптимальных решений практически невозможно. В данном случае понятие нахождение оптимального решения предполагает, что в результате получен наиболее подходящий требованиям клиента вариант квартиры, а также то, что он был найден за минимальное время выполнения.

Существует множество формальных методик принятия решения, от простого голосования до применения структурированных подходов. Каждая из них имеет свои сильные и слабые стороны, и выбор конкретной методики зависит как от решаемой задачи, так и от стадии, на которой находится разработка в данный момент времени. Считается, что чем проще методика, тем лучше, но с другой стороны, решение комплексных проблем требует применения методик, позволяющих всесторонне оценивать предложения. Будем рассматривать следующие методики для решения поставленной задачи:

- поиск специалистом;
- анализ рынка;
- метод анализа иерархий;
- командный консенсус;
- дерево принятия решений.

Краткое описание методик, а также этапы каждой методики представлены в таблице 3.1[3].

Таблица 3.1 – Методики принятия решений

Поиск специалистом	Анализ рынка	Метод анализа иерархий	Командный консенсус	Дерево принятия решений
Специалисту необходимо узнать у клиента критерии поиска, затем найти в базе данных подходящие варианты квартир, предложить	Является методом оценки альтернатив на основе использования равнозначных критериев. Все критерии делятся на	Узнать числовые приоритеты критериев поиска предложений. Затем необходимо построить матрицы	Цель консенсуса – принять решение, отражающее мнения всех участников. Он берет за основу достижение согласия всех участников в том, что выбранное	Методика, помогающая выбрать решение между несколькими направлениями действий. Деревом решения является граф, в корне которого поставленная

Поиск специалистом	Анализ рынка	Метод анализа иерархий	Командный консенсус	Дерево принятия решений
<p>их клиенту. Уточнить у клиента дополнительную информацию. Затем откорректировать ранее найденные варианты, согласно полученной дополнительной информации. Описанная процедура повторяется до тех пор, пока найденные варианты не удовлетворят клиента полностью.</p>	<p>«обязательные» (да/нет) и «желательные». Если альтернатива не выполняет требования хотя бы одного из «обязательных» критериев, эта альтернатива автоматически убирается из рассмотрения. Оставшиеся варианты оцениваются по «желательным» критериям, и предложение, набравшее наибольшее количество баллов, «побеждает». Плюсом этой методики является простота, однако без определения важности критериев выбора может сложиться ситуация, когда решение будет принято на основании второстепенных критериев[24].</p>	<p>попарных сравнений для критериев, и для альтернатив. Перед началом вычислений, необходимо нормировать данные, затем вычисляются веса с т.з. достижения цели, по критериям. Затем полученные матрицы перемножаются, т.е. вычисляются веса альтернатив. Для получения списка предложений в порядке удовлетворения требованиям клиента, необходимо отсортировать список относительно вычисленным весам альтернатив. Альтернатива с наибольшим значением веса - самая подходящая под требования клиента.</p>	<p>решение является наиболее приемлемым для группы. В начале происходит «мозговой штурм», чтобы сократить список. Затем мультиголосование</p> <ol style="list-style-type: none"> <li>1) составление списка предложений</li> <li>2) объединение похожих пунктов, перенумерация списка</li> <li>3) голосование за предложения, которые считаются подходящими (количество предложений за которые можно голосовать ограничено)</li> <li>4) подсчет голосов</li> <li>5) сокращение списка предложений</li> <li>6) повторение шагов 2-5</li> </ol> <p>Минусом этой методики является большое количество затраченного времени.</p>	<p>проблема. Каждым узлом дерева является многовариантное решение, которое надо принять и которое определяет, в какой узел осуществляется дальнейший переход. Результат - путь от начального узла к конечному, состоящий из последовательности и решений. Этапы:</p> <ol style="list-style-type: none"> <li>1) создание дерева решений;</li> <li>2) оценка дерева решений - оценка полученного результат по параметрам: подходит ли клиенту предложений и вероятность реализации решения;</li> <li>3) получение решения - оптимальным является вариант с наиболее подходящим предложением и с наиболее возможным вариантом.</li> </ol>

На рисунке 3.1 представлен граф, отображающий этапы каждой методики, представленной в таблице 3.1. А на рисунке 3.2 представлена одна из четырех веток дерева принятия решения для решения данной задачи[17].



Рисунок 3.1 – Граф, отображающий этапы методик

Согласно графу, отображенному в приложении М, нельзя определить какая из рассмотренных методик является оптимальной. Для этого необходимо оценить более подробно рассмотренные методики.

### 3.2 Оценка выбранного метода построения поиска решения

Найдем оптимальную методику принятия решения согласно оговоренному выше определению оптимального решения. То есть в результате должен быть получен наиболее подходящий требованиям клиента вариант квартиры, а также он должен быть найден за минимальное время выполнения.

Первое условие оптимального решения будет выполняться во всех методиках, отображенных на рисунке 3.1, так как результатом решения каждой методики будет предложение, подходящее клиенту. Судя по рисунку 3.1, визуально дерево принятия решений выглядит как наименее затрачиваемая на выполнение времени методика. Однако, докажем это, используя метод нахождения кратчайшего пути, или воспользуемся задачей минимального пути[25]. Для этого расставим на графе, изображённого на рисунке 3.1, время в минутах, которое затрачивается для каждого этапа, а также для удобства и

наглядности пронумеруем вершины графа. Полученный граф изображен на рисунке 3.2. В Приложении Б отображен данный граф.

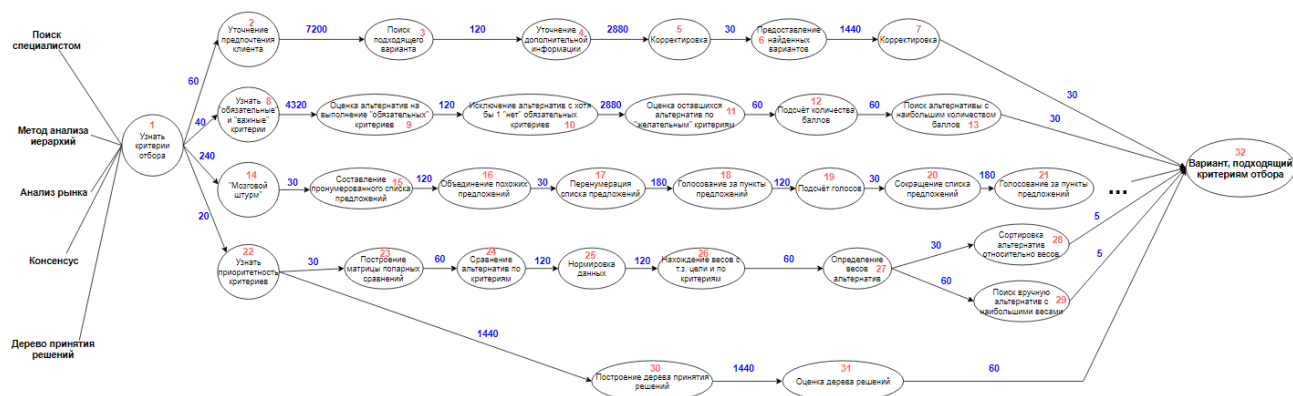


Рисунок 3.2 – Пронумерованный граф с отображением затраченного времени

Из вершины 1 попасть в вершину 32 можно несколькими способами – это 2-3-4-5-6-7, 8-9-10-11-12-13, 14-15-16-17-18-19-20-21-..., 22-23-24-25-26-27-28, 22-23-24-25-26-27-29, 22-30-31.

Путь 14-15-16-17-18-19-20-21-... сразу вычеркиваем, потому количество шагов заранее не известно, вершины 18, 19 и 20 могут повторяться каждый раз по-разному, количество повторений будет зависеть от конкретной задачи. Отообразим оставшиеся варианты в таблице 3.2 время, затраченное для каждого этапа[5].

Таблица 3.2 – Учет времени на каждом этапе

Начало дуги	Конец дуги	Время, затраченное на каждом этапе, мин
1	2	60
1	8	40
1	22	20
2	3	7200
3	4	120
4	5	2880
5	6	30
6	7	1440
7	32	30
8	9	4320
9	10	120
10	11	2880
11	12	60
12	13	60
13	32	30

Начало дуги	Конец дуги	Время, затраченное на каждом этапе, мин
22	23	30
22	30	1440
23	24	60
24	25	120
25	26	120
26	27	60
27	28	30
27	29	60
28	32	5
29	32	5
30	31	1440
31	32	120

Обозначим  $C(T)$  длину кратчайшего пути из 1 в вершину  $T$ . То есть, длину кратчайшего пути до вершины 32 будем искать по формуле (3.1).

$$C(32) = \min \{A; B; C; D; E\}, \quad (3.1)$$

где,

$$C(1) = 0,$$

$$A = C(1) + C(2) + C(3) + C(4) + C(5) + C(6) + C(7) + C(32),$$

$$B = C(1) + C(8) + C(9) + C(10) + C(11) + C(12) + C(13) + C(32),$$

$$C = C(1) + C(22) + C(23) + C(24) + C(25) + C(26) + C(27) + C(28) + C(32),$$

$$D = C(1) + C(22) + C(23) + C(24) + C(25) + C(26) + C(27) + C(29) + C(32),$$

$$E = C(1) + C(22) + C(30) + C(31) + C(32).$$

Подставляя свои значения, находим:

$$A = 60 + 7200 + 120 + 2880 + 30 + 1440 + 30 = 11760,$$

$$B = 40 + 4320 + 120 + 2880 + 60 + 60 + 30 = 7510,$$

$$C = 20 + 30 + 60 + 120 + 120 + 60 + 30 + 5 = 445,$$

$$D = 20 + 30 + 60 + 120 + 120 + 60 + 60 + 5 = 475,$$

$$E = 20 + 1440 + 1440 + 120 = 3020.$$

Таким образом, согласно формуле (3.1),  $C(32) = \min \{11760; 7510; 445; 475; 3020\} = 445$ .

Следовательно, путь 22-23-24-25-26-27-28 оказался наиболее коротким относительно потраченного времени для решения задачи. Данный путь



описывает алгоритм метода анализа иерархий. Следовательно, метод анализа иерархий затрачивает наименьшее количество времени на выполнение из всех разобранных методик. Так как он является самым наименее затратным относительно времени выполнения, и в результате его работы клиент получает удовлетворяющий его требованиям вариант, метод анализа иерархий является оптимальным методом решения поставленной задачи.

Так как оптимальность выбранного метода анализа иерархий была доказана в данной главе, следовательно, поставленная цель выпускной работы, а именно разработка математической модели оптимального поиска объектов недвижимости и реализация ее программным путем, была выполнена.

Таким образом, в данной главе было рассмотрено определение оптимального решения, были оценены методики принятия решения, а также была доказана оптимальность метода анализа иерархий, с помощью применения метода нахождения кратчайшего пути, то есть для доказательства оптимальности разработанного приложения воспользовались задачей минимального пути.

## ЗАКЛЮЧЕНИЕ

Поиска объектов недвижимости специалистом рутинный процесс, который затрачивает много времени, поэтому его необходимо автоматизировать.

В ходе выполнения выпускной работы был характеризован процесс поиска объектов недвижимости в теории принятия решений, была исследована предметная область для определения конструктивных компонентов, обеспечивающих распределение приоритетных параметров при поиске объектов недвижимости, построена математическая модель поиска решений на основе метода анализа иерархий.

Методом анализа иерархий называют методологическую основу для решения задач выбора альтернатив посредством их многокритериального рейтингования. Данный метод помогает выбирать наилучшее решение среди нескольких альтернатив относительно нескольких критериев, он не предписывает лицу, принимающему решение, какое-либо правильное решение, а позволяет самому в интерактивном режиме найти такой вариант, который наилучшим образом согласуется с его пониманием сути проблемы и требованиями к ее решению. Метод был разработан для оптимизации принятия решений, когда необходимо учесть качественные, количественные и иногда противоречивые факторы.

Метод анализа иерархий представляет элементы, определяющие суть проблемы, иерархически, тем самым декомпозируя проблемы на более простые составляющие части и обрабатывая последовательности суждения лица, принимающего решения, по парным сравнением. В результате получается численная интенсивность, определяющая степень взаимодействия элементов в иерархии.

Также в ходе выполнения выпускной работы была смоделирована система поиска решения для риэлтерской компании, был разработан алгоритм работы приложения, а также он был реализован программными средствами.

Кроме того были оценены методики принятия решения, а также была доказана оптимальность метода анализа иерархий, с помощью применения метода нахождения кратчайшего пути, то есть для доказательства оптимальности разработанного приложения воспользовались задачей минимального пути.

Разработанное приложение оптимизирует работу риэлтора, увеличит коэффициент производительности работников риэлтерской компании, исключит возможность субъективного мнения риэлтора при поиске предложений для клиента. Также разработанное приложение, осуществляющее поиск объектов недвижимости, позволит быть риэлтерской компании лучшей на рынке недвижимости, позволит предоставлять услуги высокого качества, затрачивая на это меньше времени. Данное приложение позволит оставаться «БИЭМ-Недвижимости» надежной компанией, совершенствоваться и повышать качество услуг.

В дальнейшем разработанное приложение планируется развивать. Используя его, можно будет искать объекты недвижимости не только для купли, но и для аренды. Это позволит автоматизировать деятельность риэлтора в еще одном направлении.

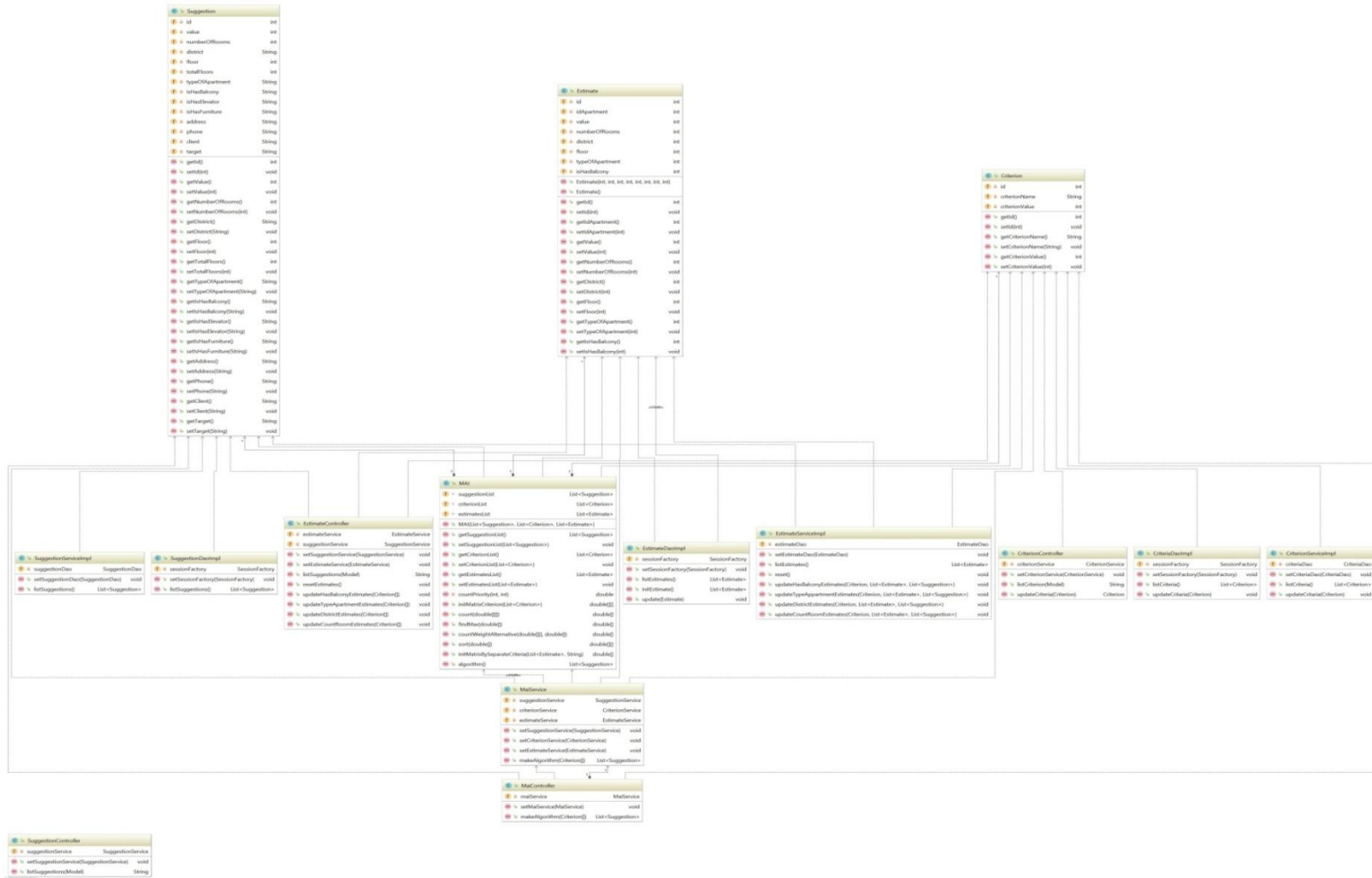
## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Flanagan D. JavaScript: Подробное руководство (Definitive Guide)/ D. Flanagan. – СПб: Символ-Плюс, 2015. – 1080 с.
2. Блох Д. Эффективное программирование/ Д. Блох – М., 2014. – 461 с.
3. Джоффрион А. Решение задач оптимизации при многих критериях на основе человеко-машинных процедур. Вопросы анализа и процедуры принятия решений. / А. Джоффрион, Дж. Дайер, А. Файнберг – М.: Мир, 2014. - 232 с.
4. Дэвид Г. Метод парных сравнений./Пер. с англ. под ред. Ю. Адлера. – М.: Статистика, 2015. – 114 с.
5. Ойстин О. Теория графов/ О. Ойстин. – М.: Наука, 2015. – 354 с.
6. Орлов А.И. Теория принятия решений. Учебное пособие. – М.: Издательство "Март", 2014. - 656 с.
7. Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов. — М.: Манн, Иванов и Фербер, 2013. - 136 с.
8. Саати Т. Л. Принятие решений при зависимостях и обратных связях: Аналитические сети / Т. Л. Саати – М.: Издательство ЛКИ, 2015. – 360 с
9. Саати Т. Принятие решений. Метод анализа иерархий / Пер. с англ. под ред. Р. Г. Вачнадзе Москва «Радио и связь», 2014.
10. Соболев И.М. Выбор оптимальных параметров в задачах со многими критериями. / И.М. Соболев, Р.Б. Статников – М.: "Наука", 2014. – 110 с.
11. Уоллс К. Spring in Action/ К. Уоллс - ДМК Пресс, 2015 – 754 с.
12. Федоров И. Моделирование бизнес-процессов в нотации BPMN 2.0 / И. Федоров. – М: МЭСИ, 2013. — 264 с.
13. Хорстманн К. «Java. Библиотека профессионала. Том 1».10-е издание/ К. Хорстманн, Г. Корнелл – М., 2016. – 864 с.
14. Моренин, А. Анализ математических методов поддержки принятия решений [Электронный ресурс]. – Режим доступа: <http://www.olap.ru/best/analysis.asp> (дата обращения: 15.12.2017)

15. Биэм-Недвижимость [Электронный ресурс]. – Режим доступа: <https://www.biemn.ru/about/> (дата обращения: 20.01.2018)
16. Царев, М. Введение в теорию принятия решений [Электронный ресурс]. – Режим доступа: <http://rain.ifmo.ru/cat/data/theory/unsorted/decision-making-2007/article.pdf> (дата обращения: 23.12.2017)
17. Шахиди А. Деревья решений – общие принципы работы [Электронный ресурс]. – Режим доступа: <http://www.basegroup.ru/library/analysis/tree/description> (дата обращения: 05.04.2018)
18. Bauer C. Java Persistence with Hibernate, 2nd Edition/ C. Bauer, G. Gregory, G. King, 2015
19. Chiara Mocenni, The Analytic Hierarchy Process [Электронный ресурс]. – Режим доступа: [http://www.dii.unisi.it/~mocenni/Note\\_AHP.pdf](http://www.dii.unisi.it/~mocenni/Note_AHP.pdf) (дата обращения: 02.02.2018)
20. Enrique Mu, Milagros Pereyra-Rojas, Understanding the Analytic Hierarchy Process [Электронный ресурс]. – Режим доступа: [https://books.google.ru/books?id=2f7cDAAAQBAJ&pg=PA7&lpg=PA7&dq=Chapter+2+Understanding+the+Analytic+Hierarchy+Process+In+this+chapter,+we+will+explain+the+fundamentals+of+the+Analytic+Hierarchy+Process.+The+reader+is+referred+to+the+original+Saaty%20%99s+\(2012\)+discussion+of+AHP+or+to+Brunelli%20%99s+\(2015\)+for+a+theoretical+introduction+to+the+method.+In+this+book,+AHP+concepts+will+be+explained+from+a+practical+point+of+view+using+examples+for+greater+clarity.&source=bl&ots=M8FtVhw-Z8&sig=k9GWZsdPXyQvQ6Fjh8\\_brxeOu3o&hl=ru&sa=X&ved=0ahUKEwi5x7XCtabaAhUBWSwKHQDvCnkQ6AEINjAC#v=onepage&q&f=false](https://books.google.ru/books?id=2f7cDAAAQBAJ&pg=PA7&lpg=PA7&dq=Chapter+2+Understanding+the+Analytic+Hierarchy+Process+In+this+chapter,+we+will+explain+the+fundamentals+of+the+Analytic+Hierarchy+Process.+The+reader+is+referred+to+the+original+Saaty%20%99s+(2012)+discussion+of+AHP+or+to+Brunelli%20%99s+(2015)+for+a+theoretical+introduction+to+the+method.+In+this+book,+AHP+concepts+will+be+explained+from+a+practical+point+of+view+using+examples+for+greater+clarity.&source=bl&ots=M8FtVhw-Z8&sig=k9GWZsdPXyQvQ6Fjh8_brxeOu3o&hl=ru&sa=X&ved=0ahUKEwi5x7XCtabaAhUBWSwKHQDvCnkQ6AEINjAC#v=onepage&q&f=false) (дата обращения: 13.03.2018)
21. Hibernate Tutorial [Электронный ресурс]. – Режим доступа: <https://www.tutorialspoint.com/hibernate/index.htm> (дата обращения: 15.02.2018)
22. Klutho S. Mathematical Decision Making An Overview of the Analytic Hierarchy Process/ S. Klutho, 2013

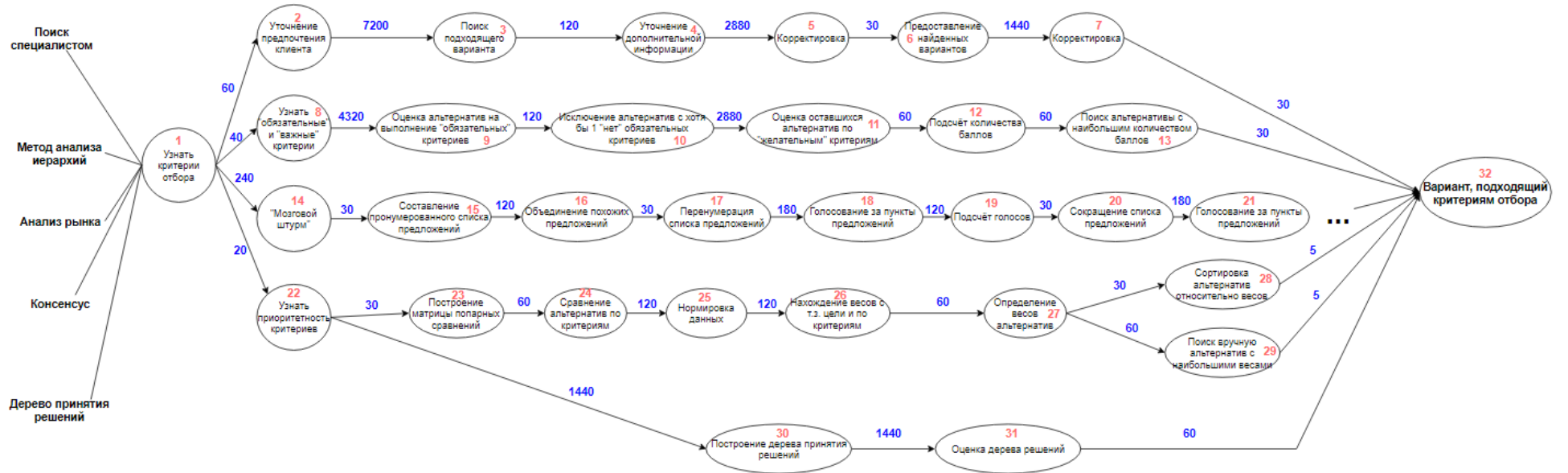
23. Melvin, A. Decision-Making using the Analytic Hierarchy Process (АНП) [Электронный ресурс]. – Режим доступа: <https://analytics.ncsu.edu/sesug/2012/SD-04.pdf> (дата обращения: 16.03.2018)
24. Goodwin P. Decision Analysis for Management Judgment, 3rd edition/ P. Goodwin, G. Wright. – Chichester: Wiley, 2014
25. Reinhard D. Graph Theory / D. Reinhard, 2014
26. Snyman, J.A. Practical Mathematical Optimization - Basic Optimization Theory and Gradient-Based Algorithms/ J.A Snyman, D.N. Wilke, 2017
27. Spring Framework Documentation [Электронный ресурс]. – Режим доступа: <https://docs.spring.io/spring/docs/current/spring-framework-reference/index.html> (дата обращения: 27.01.2018)
28. Spring Framework Tutorial [Электронный ресурс]. – Режим доступа: [https://www.tutorialspoint.com/spring/spring\\_web\\_mvc\\_framework.htm](https://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm) (дата обращения: 27.01.2018)

# ПРИЛОЖЕНИЕ А ДИАГРАММА КЛАССОВ



# ПРИЛОЖЕНИЕ Б

## ПРОНУМЕРОВАННЫЙ НАПРАВЛЕННЫЙ ГРАФ С УЧЕТОМ ВРЕМЕНИ





## ПРИЛОЖЕНИЕ В

### КЛАСС CRITERION

```
import javax.persistence.*;

@Entity
@Table(name = "criteria")
public class Criterion {
    @Id
    @Column(name = "ID")
    private int id;

    @Column(name = "CRITERION_NAME")
    private String criterionName;

    @Column(name = "CRITERION_VALUE")
    private int criterionValue;

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getCriterionName() { return criterionName; }

    public void setCriterionName(String criterionName) { this.criterionName = criterionName; }

    public int getCriterionValue() { return criterionValue; }

    public void setCriterionValue(int criterionValue) { this.criterionValue = criterionValue; }
}
```

# ПРИЛОЖЕНИЕ Г

## ВЕРСТКА СТАРТОВОЙ СТРАНИЦЫ ПРИЛОЖЕНИЯ

```
1 <%@ page contentType="text/html;charset=UTF-8" language="java" %>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3
4 <html>
5 <head>
6 <title>BIEM</title>
7 <base href="{pageContext.request.contextPath}">
8 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
9 <link href="{c:url value="/resources/bootstrap-4.1.1/dist/css/bootstrap.css"/}" rel="stylesheet" type="text/css">
10 <script src="{c:url value="/resources/bootstrap-4.1.1/dist/js/bootstrap.min.js"/}"></script>
11 <script src="{pageContext.request.contextPath}/resources/action.js" type="text/javascript"></script>
12 </head>
13 <body style="background-color:#F5F5F5">
14 <div class="container" style="...">
15 <div class="header py-3" style="background-color:#FFED5">
16 <div class="row flex-nowrap justify-content-between align-items-center">
17 <div class="col-8 ml-5">
18 
19 </div>
20 <div class="col-3 d-flex justify-content-end pr-4">
21 <a class="btn btn-sm btn-outline-secondary" href="#">Войти в систему</a>
22 </div>
23 </div>
24 </div>
25 <div class="nav-scroller" style="border-bottom: 2px solid orange; border-top: 2px solid orange">
26 <nav class="nav d-flex justify-content-between">
27 <a class="p-2 text-muted" href="#">Компания</a>
28 <a class="p-2 text-muted" href="#">Услуги</a>
29 <a class="p-2 text-secondary" href="#">Планировки</a>
30 <a class="p-2 text-secondary" href="#">Контакты</a>
31 <a class="p-2 text-info" href="#">Подбор вариантов</a>
32 </nav>
33 </div>
34 <div class="row">
35 <c:import url="WEB-INF/pages/menu.jsp"/>
36 <c:import url="WEB-INF/pages/criteria.jsp"/>
37 <c:import url="WEB-INF/pages/result.jsp"/>
38 </div>
39 </div>
40 <div class="blog-footer d-flex justify-content-center py-5">
41 <p>© ООО «БИЭМ Недвижимость», 2018 г</p>
42 </div>
43 </body>
44 </html>
```

# ПРИЛОЖЕНИЕ Д

## ЧАСТЬ ДЕРЕВА ПРИНЯТИЯ РЕШЕНИЙ

