

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

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему: «Математическая модель и алгоритм принятия решений для HR-менеджеров в условиях неопределенности»

Обучающийся

А.В. Рязанцева

(И.О. Фамилия)

(личная подпись)

Руководитель

М.А. Тренина

(ученая степень, звание, И.О. Фамилия)

Консультант

к.п.н., доцент С.А. Гудкова

(ученая степень, звание, И.О. Фамилия)

Аннотация

Тема данной выпускной квалификационной работы «Математическая модель и алгоритм принятия решений для HR-менеджеров в условиях неопределенности».

Объектом исследования является бизнес-процесс выбора кандидата на должность в ООО «Стратегия Роста».

Предмет исследования – математическая модель и алгоритм принятия решений в бизнес-процессе выбора кандидата на должность.

Целью работы является разработка математической модели и алгоритма принятия решений для HR-менеджеров в условиях неопределенности.

Бакалаврская работа посвящена построению математической модели и разработке алгоритма принятия решений для HR-менеджеров в условиях неопределенности. Она состоит из введения, двух глав, заключения, списка используемой литературы. Во введении раскрывается актуальность темы, формулируется цель и поставленные задачи для ее достижения.

В первой главе проанализирована проблема выбора кандидата на должность, поставлена многокритериальная задача в условиях неопределенности, построена математическая модель, выбран и исследован метод решения, разработан алгоритм. Во второй главе представлено описание программной реализации и тестирования приложения с графическим интерфейсом пользователя.

В заключении приведены выводы по проделанной работе.

Результатами работы являются математическая модель и приложение с графическим интерфейсом пользователя для выбора кандидата на должность, при помощи которого можно быстро и эффективно принимать обоснованные решения.

В бакалаврской работе 52 страницы, содержащих 19 рисунков, 8 таблиц и список используемой литературы из 25 источников.

Abstract

The title of the graduation work is «Mathematical model and decision-making algorithm for HR managers under conditions of uncertainty».

The relevance of the topic of this work is due to the importance of high-quality personnel selection for the successful operation of the organization. Highly qualified and talented specialists contribute to the company's growth.

The object of the research is the business process of selecting a candidate for a position at Strategia Rosta LLC.

The subject of the graduation work is a mathematical model and decision-making algorithm in the business process of selecting a candidate for a position.

The aim of the work is to develop a mathematical model and decision-making algorithm for HR managers under conditions of uncertainty.

The graduation work is devoted to the construction of a mathematical model and the development of a decision-making algorithm for HR managers under conditions of uncertainty. It consists of an introduction, two chapters, a conclusion, a list of references.

The introduction reveals the relevance of the topic, formulates the aim and the tasks set to achieve it.

In the first chapter, the problem of selecting a candidate for a position is analyzed, a multi-criterion problem is posed under conditions of uncertainty, a mathematical model is built, a solution method is selected and studied, and an algorithm is developed. The second chapter described the software implementation and testing of the application with a graphical user interface. In conclusion, conclusions on the work done are presented.

The results of the work are a mathematical model and an application with a graphical user interface for selecting a candidate for a position, with which you can quickly and efficiently make informed decisions.

The graduation work consists of 52 pages, contains 8 tables, 19 illustrations, the list of 25 references including 5 foreign sources.

Оглавление

Введение.....	5
Глава 1 Анализ предметной области.....	7
1.1 Проблема выбора кандидата на должность	7
1.2 Постановка и классификация задачи принятия решений.....	8
1.3 Формирование множества критериев задачи принятия решений....	13
1.4 Построение математической модели принятия решений.....	17
1.5 Методы решения задачи выбора на основе экспертной оценки	19
1.6 Исследование метода анализа иерархий	21
1.7 Разработка алгоритма принятия решений	25
1.8 Решение задачи выбора кандидата на должность	26
Глава 2 Программная реализация алгоритма принятия решений.....	33
2.1 Функциональное моделирование приложения для выбора кандидата на должность.....	33
2.2 Выбор программных средств создания графического интерфейса приложения.....	36
2.3 Реализация алгоритма и интерфейса приложения на Java	38
2.4 Тестирование разработанного приложения	43
Заключение	48
Список используемой литературы и используемых источников.....	49

Введение

Принятие решений играет ключевую роль в различных сферах жизни человека и организаций. От выбора оптимального решения зависит успешность достижения поставленных целей, эффективность использования ресурсов и общий успех проекта или бизнеса.

Подбор персонала – это вспомогательный бизнес-процесс организации. Тем не менее, в современном бизнесе, в условиях неопределенности и высокой конкуренции на рынке труда роль HR-менеджеров (менеджеров по персоналу) является существенной. Качественно подобранный персонал один из основных факторов успешной работы организации.

Часто при выборе кандидата на должность бывает сложно принять правильное, обоснованное решение, этот процесс требует больших временных затрат. В связи с этим данный вид процессов следует максимально оптимизировать и автоматизировать.

Таким образом, актуальность темы выпускной квалификационной работы (ВКР) обусловлена необходимостью совершенствования бизнес-процесса подбора персонала, а именно выбора наиболее подходящего кандидата на должность.

Объектом исследования является бизнес-процесс выбора кандидата на должность в ООО «Стратегия Роста».

Предмет исследования – математическая модель и алгоритм принятия решений в бизнес-процессе выбора кандидата на должность.

Целью ВКР является разработка математической модели и алгоритма принятия решений для HR-менеджеров в условиях неопределенности.

Для достижения цели, нужно решить следующие задачи:

- исследовать предметную область;
- построить математическую модель задачи выбора наиболее подходящего кандидата на должность;
- выбрать метод для решения данной задачи;

- разработать алгоритм принятия решений на основе построенной модели и выбранного метода;
- функционально смоделировать программу для выбора кандидата на должность;
- выбрать программные средства для создания графического интерфейса;
- реализовать алгоритм и интерфейс с помощью языка программирования Java;
- протестировать разработанное приложение.

Глава 1 Анализ предметной области

1.1 Проблема выбора кандидата на должность

В современном бизнесе одной из ключевых задач является выбор правильного кандидата на открытую должность. Процесс подбора персонала напрямую влияет на успех организации: высококвалифицированные и талантливые специалисты способствуют росту компании. Поэтому необходимо проанализировать эту проблему.

Отбором кандидатов на вакантную должность в ООО «Стратегия Роста» занимается HR-менеджер, на компьютере которого хранятся резюме. Их присылают претенденты по электронной почте. Процесс отбора проводится по этапам, на каждом из которых отсеивается часть кандидатов. Как правило, в данном процессе применяют собеседование, анализ документов, тестирование.

После того, как кандидаты будут отобраны, может возникнуть сложность в сравнении их между собой, поскольку одно резюме может выделяться на одних критериях, в то время как другое – на других.

При изучении проблемы было найдено несколько причин ее возникновения. Во-первых, традиционные подходы к выбору кандидата на должность (например, собеседования) могут быть исключительно субъективными и зависеть от индивидуальных предвзятостей HR-менеджеров.

Во-вторых, выбор кандидата на должность – это многокритериальная задача, где каждый критерий вносит свой вклад в формирование оценки. С ростом количества критериев оценки, задача усложняется. Таким образом, этот процесс может занять большое количество времени.

Анализ проблемы позволяет выявить недостатки существующего бизнес-процесса и определить необходимые изменения. Для решения проблемы выбора кандидата на должность можно обратиться к теории

принятия решений.

Математическая модель обеспечит снижение фактора субъективности при принятии решения, а алгоритм для выбора кандидата на должность позволит автоматизировать процесс. Этот подход помогает оценить все возможные варианты, принять обоснованное решение и сократить временные затраты.

1.2 Постановка и классификация задачи принятия решений

Теория принятия решений (ТПР) изучает такие задачи наилучшего выбора, когда у лица, принимающего решение (ЛПР), есть несколько вариантов, и он может выбрать наиболее подходящий из них, учитывая цели и предпочтения.

Лицом, принимающим решение, называют человека (или группу людей), который осуществляет выбор и отвечает за его последствия. Существует индивидуальное и групповое ЛПР. В контексте задачи важны лишь те его характеристики, которые могут быть связаны с процессом выбора (например, опыт в данной области). Далее рассматриваться задача индивидуального ЛПР, то есть будут учитываться предпочтения одного лица. Эксперты (специалисты в сфере решения определенной проблемы) помогают ЛПР собирать и анализировать информацию, формировать решение.

Задачи принятия решений (ЗПР) встречаются часто в компаниях, например: работники банков могут выбирать объекты для инвестирования, руководители различных подразделений выбирать более подходящую линию поведения и т.д. ТПР учит делать выбор обоснованно, эффективно пользоваться информацией о целях, предпочтениях. Теория также помогает избежать негативных последствий непродуманных решений и находить наилучшие варианты.

Выбор кандидата на должность является задачей принятия решений, которую в общем виде можно представить «Пусть дано..., нужно найти

лучшую альтернативу...». Следовательно, данную задачу можно представить в форме:

$$\langle t, X, R, A, F, G, D \rangle, \quad (1)$$

где t – тип задачи;

X – множество альтернатив (допустимых решений);

R – множество критериев выбора;

A – множество шкал для измерения критериев;

F – отображение множества допустимых решений в множество оценок критериев;

G – система предпочтений лица, принимающего решение;

D – решающее правило (методы принятия решений) [3].

Соответственно этой форме будет использоваться классификация задач принятия решений, представленная в виде схемы на рисунке 1.



Рисунок 1 – Классификация задач принятия решений

Задачи принятия решений могут быть разных типов, которые характеризуют цели ЛПР. В данном случае – упорядочивание альтернатив по важности и затем нахождение наилучшей из них.

Для принятия обоснованного решения необходимо задать набор решений, из которого будет совершен выбор – множество возможных (допустимых) решений X . Допустимым может быть решение, которое удовлетворяет ограничениям (например, правовым или ресурсным). Синонимами понятия «решение», могут быть такие термины, как альтернатива, стратегия, план, вариант. Множество допустимых решений может быть и бесконечным, и конечным, но должно содержать минимально два элемента.

Множество X содержит подмножество $C(X)$. Это множество выбираемых решений, оно может состоять не только из одного элемента (может быть набор решений), но должен быть хотя бы один элемент, чтобы выбор мог осуществиться. В некоторых задачах $C(X)$ бесконечное. Задачу выбора можно назвать решенной, если найдено множество $C(X)$, $C(X) \subset X$.

Критерии в ЗПР – это показатели, которые определяют привлекательность (или непривлекательность) альтернатив для ЛПР. Это основа для выбора наилучшего варианта. В зависимости от количества существуют задачи принятия решений со скалярным критерием и многокритериальные ЗПР (с векторным критерием). Сегодня в теории принятия решений главное место занимают многокритериальные ЗПР. Такие задачи считаются наиболее приближенными к реальности. Далее будут рассмотрен этот тип задач.

Основной сложностью многокритериальной задачи является невозможность априорно определить, что назвать наилучшим вариантом. Само понятие «наилучшего» зависит от психологического восприятия ситуации человеком и от множества факторов, которые на данный момент развития науки и математического аппарата невозможно учесть в модели [4].

Важной частью многокритериальной задачи выбора является

векторный критерий вида:

$$f = (f_1, f_2 \dots f_n), n \geq 2. \quad (2)$$

Числовые функции $f_1, f_2 \dots f_n$ (определенные на множестве X), называют критериями оптимальности. Векторный критерий f принимает значения в пространстве n -мерных векторов R^n . Такое пространство именуют критериальным пространством (пространством оценок), а любое значение $f(x) = (f_1(x), f_2(x), \dots, f_n(x)) \in R^n$ векторного критерия f при определенном $x \in X$ называют векторной оценкой допустимого решения x . Все возможные векторные оценки образуют множество возможных оценок критериев $Y = f(X) = \{y \in Y \mid y = f(x) \text{ при } x \in X\}$ [16].

Множество выбираемых оценок $C(Y) = f(C(X)) = \{y \in Y \mid y = f(x) \text{ при } x \in C(X)\}$ представляет собой подмножество множества Y .

Обычно между множествами возможных решений X и соответствующими векторами Y существует взаимно однозначное соответствие F . Это означает, что каждому решению соответствует определенный вектор, и наоборот, каждому вектору соответствует определенное решение. Поэтому выбор решения математически эквивалентен выбору вектора.

По виду отображения F (детерминированное, вероятностное и неопределенное) различают ЗПР в условиях определенности, риска и неопределенности соответственно. В условиях определенности для принятия решения доступна полная и достоверная информация о проблемной ситуации, все факты, альтернативы и их последствия точно известны. ЗПР в условиях риска известны вероятности возможных исходов, но точных результатов неизвестно. В условиях неопределенности доступна неясная информация для выбора наилучшей альтернативы, а вероятности исходов не определены.

Неопределенность может возникать при принятии финансовых

решений (например, инвестирование), при разработке стратегий в сфере маркетинга или при управлении проектами (например, из-за изменения требований заказчика). Также существует неопределенность и в процессе подбора персонала, она может быть вызвана изменениями в бизнесе (например, смена стратегии компании или применение новых технологий), или высокой конкуренцией на рынке труда. При оценке кандидата на должность используется субъективная информация, которая тоже является причиной неопределенности в задаче принятия решений.

Векторный критерий помогает определить локальные цели, которые часто оказываются противоречивыми. Но их одновременное достижение почти невозможно, поэтому для нахождения компромисса требуется дополнительная информация о предпочтениях ЛПР. Для этой цели в многокритериальную задачу необходимо включить дополнительный элемент, который позволит учесть и описать эти предпочтения.

Пусть существует два возможных решения x' и x'' и предположено, что ЛПР из этих решений выбирает (то есть предпочитает) первое. Это можно представить в виде $x' \succ_X x''$. Знак \succ_X обозначает предпочтение данного ЛПР, которое выражается отношением строгого предпочтения.

Отношение предпочтения \succ_Y на множестве возможных векторов Y , обусловленное отношением предпочтения на множестве решений X , определяется так

$$f(x') \succ_Y f(x'') \leftrightarrow x' \succ_X x'', x' \in X \text{ и } x'' \in X. \quad (3)$$

Вектор $y' = f(x')$ предпочтительнее вектора $y'' = f(x'')$, при условии, что соответствующее решение x' предпочтительнее решения x'' .

Предпочтения ЛПР представляют упорядоченное в порядке важности отношение на множестве критериев. Для измерения степени важности используют различные шкалы: порядка, интервальные, отношений, наименований и другие.

Таким образом, суть многокритериальной задачи выбора состоит в нахождении множества выбираемых решений $C(X)$, $C(X) \subset X$, ввиду его отношения предпочтения \succ_X на основании векторного критерия f , который отражает цели ЛПР.

Руководителю компании (ЛПР) необходимо выбрать наиболее подходящего кандидата на должность из множества. Пусть X – множество претендентов, тогда требования к кандидатам представляются в виде векторного критерия f в виде (2). Такую задачу можно классифицировать как:

- многокритериальную задачу (необходимо оптимизировать несколько критериев одновременно);
- задачу в условиях неопределенности. Оценка кандидатов из-за субъективных предпочтений, ограниченной информации может быть подвержена неопределенности;
- задачу, которая представляет интересы одного лица.

Таким образом, была представлена постановка задачи выбора кандидата на должность, а также классификация по основным элементам. Это позволяет рассмотреть проблемную ситуацию с различных сторон, лучше понимать сущность задачи и подобрать методы для ее решения.

1.3 Формирование множества критериев задачи принятия решений

После постановки задачи и определения цели принятия решения необходимо сформулировать набор критериев, по которому будет происходить оценка альтернатив. Критериям следует отражать все аспекты поставленной цели, однако они не должны быть избыточны. Несмотря на то, что такая задача может быть сложна для формализации, можно подчеркнуть важные свойства, которым следует обладать набору критериев.

Первое свойство – полнота. Это свойство характеризуется тем, что если к полному набору критериев добавить дополнительные критерии, результат

принятия решения не меняется, а исключение хотя бы одного критерия из полного набора вызывает его изменение. То есть необходимо отбросить критерии, которые не имеют влияния на поставленную задачу.

Следующим свойством является избыточность. Различные критерии из набора охватывают различные аспекты последствий. Например, рассматривая задачу оптимизации перевозки грузов, нет необходимости учитывать расходы на перевозку на определенном участке дороги и расход топлива на нем одновременно, так как второй показатель участвует в формировании первого [13].

Третье свойство – минимальность. Количество критериев в наборе должно быть как можно меньше. Существует некоторое противоречие свойства минимальности свойству полноты, но следует находить компромисс между бесосновательной сложностью и полнотой описания характеристик задачи.

Следующее свойство – прозрачность. Критерии должны быть понятны и одинаково интерпретироваться для всех участников процесса.

Последним свойством является измеримость. Каждый критерий в наборе должен иметь возможность оценки на какой-либо шкале. Оценка критерия может быть получена с помощью физических измерений или проводиться экспертным путем.

Выделяют количественные и качественные критерии оценки. Первые из них можно точно измерить численно, например, такие как показатели цены, прибыли (рубли), времени (минуты). Но не всегда критерий можно связать с конкретным численным значением. Тогда он является качественным и описывается терминами сравнения (например, «лучше-хуже», «больше-меньше»).

Чтобы провести математический анализ качественных критериев нужно преобразовать их в количественный вид. Для этого специалисты в данной области могут применять экспертное оценивание, которое позволяет оценить важность каждого критерия и влияние его на принятие решения.

Так как эксперт компетентен в рамках своей области и нейтрально оценивает ситуацию, то его оценки являются наиболее объективными. Такая информация ценна, но стоит подчеркнуть, что их мнение имеет лишь рекомендательный характер и не является конечным решением. На основе своих предпочтений ЛПР принимает решение и несет за него ответственность. Мнение эксперта и ЛПР в общем случае могут различаться.

Экспертные оценки бывают индивидуальными и коллективными. Первый вид основан на мнении одного эксперта или нескольких экспертов, каждый из которых независим от других. Коллективная оценка основана на коллективном мнении экспертов. В данной работе для решения задачи используется индивидуальная экспертная оценка. Этот вид оценки имеет преимущества такие как оперативность, низкие затраты, возможность полностью использовать способности эксперта.

Важно различать, какие критерии являются количественными, а какие качественными, так как от этого зависит подход к решению задачи. Задачи с количественной информацией наиболее удобны для анализа, так как допустимо измерение показателей критериев в виде числа и использование количественных методов оценки и выбора альтернатив. Задачи с качественной информацией требуют использования субъективных оценок, выражающихся в порядковых или номинальных шкалах. Это делает процесс принятия решения сложным и требующим помощи экспертов и использования специальных методов. В большей части реальных задач оценка альтернатив происходит и по количественным, и по качественным критериям [16].

Для оценки кандидатов было выявлено пять основных критериев:

- уровень образования – качественный критерий, информация о полученных степенях (среднее профессиональное, бакалавриат, специалитет, магистратура);
- опыт работы – количественный критерий, определяющий, сколько лет кандидат работает в определенной области (принимает любые

положительные значения или ноль);

- профессиональные навыки – качественный критерий, который отражает знания и умения кандидата в рамках профессии. Определяются по результатам выполнения практического задания: удовлетворительно, хорошо или отлично;
- уровень владения английским языком – качественный критерий, выражающий способность кандидата понимать устную и письменную речь, говорить и писать на данном иностранном языке (начальный, средний, продвинутый уровни знания);
- технические навыки – качественный критерий, который отражает умение кандидата работать с программами (текстовыми, табличными редакторами, программным обеспечением для создания и просмотра презентаций, системами управления базами данных). Может быть начинающий, уверенный и продвинутый компьютерный пользователь.

Данный набор критериев обладает всеми вышеперечисленными свойствами. Он наиболее полно позволяет оценить кандидатов, так как затрагивает основные аспекты, которые требуются для успешной работы на определенной должности.

Уровень образования и опыт работы имеют прямую связь с качеством работы сотрудника. Если первый критерий дает понятие о теоретических знаниях кандидата, то опыт работы – насколько знакома предметная область и сколько времени человек занимается данной деятельностью. Профессиональные навыки также влияют на эффективность работы сотрудника. Они определяют, с какими рабочими задачами может справиться кандидат. Знание английского языка и умение работать с компьютерными программами также являются важными навыками в современном бизнесе.

Таким образом, были сформулированы критерии оценки кандидатов. Следующим этапом процесса принятия решений является построение математической модели.

1.4 Построение математической модели принятия решений

Математическая модель – это описание исследуемого экономического явления или процесса с помощью абстрактных математических соотношений [2]. Это способ формализации задачи принятия решений, который является непростым процессом. Проблема заключается в том, что следует учитывать важные детали реальной задачи и одновременно модель не должна быть слишком сложной, с целью успешного применения уже разработанных математических методов, ее исследования и решения.

Задачи выбора имеют тесную связь с ЛПР, поэтому при формировании множества альтернатив и критериев обращается внимание на его советы, указания. Также следует помнить, что математическая модель представляет упрощение действительности, по этой причине она не будет в точности соответствовать реальной ситуации. Важными являются детали, наиболее влияющие на конечный выбор наилучшего решения.

Пусть на должность претендуют пять кандидатов. Каждый из кандидатов оценивается по пяти критериям (пункт 1.3):

- уровень владения английским языком;
- опыт работы;
- технические навыки;

Каждый критерий имеет свой вес (приоритет), определяющий его

о
т
н
о
с
и
т
е
л
ь

Таблица 1 – Шкала предпочтений

Степень важности	Объяснение
1	Равная важность
3	Слабое превосходство
5	Сильное превосходство
7	Очень сильное превосходство
9	Абсолютное превосходство
2, 4, 6, 8	Промежуточные значения, которые попадают между ранее определенными степенями важности

Матрица сравнений строится следующим образом: если элемент i более важный, чем элемент j , то в матрицу заносится целое положительное число из таблицы 1. Иначе заносится обратное число. В результате получается матрица, на главной диагонали которой расположены только единицы, а на симметричных клетках – обратные числа. Матрица сравнений должна быть согласованной, то есть отношения элементов матрицы не противоречивые.

Далее вычисляется собственный вектор матрицы сравнений. Вектор v

Аналогично находятся оценки кандидатов. Можно представить их в виде матрицы C . Элемент матрицы c_{ij} определяет оценку i -го кандидата по j -му критерию. Умножив матрицу оценок кандидатов C на вектор весов критериев, получим вектор весов кандидатов

$$p = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_5 \end{pmatrix} = C \times \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_5 \end{pmatrix} = \begin{pmatrix} c_{11} \cdot w_1 + c_{12} \cdot w_2 + \dots + c_{15} \cdot w_5 \\ c_{21} \cdot w_1 + c_{22} \cdot w_2 + \dots + c_{25} \cdot w_5 \\ \vdots \\ c_{51} \cdot w_1 + c_{52} \cdot w_2 + \dots + c_{55} \cdot w_5 \end{pmatrix}. \quad (4)$$

Решением задачи будет кандидат, соответствующий наибольшему компоненту вектора p

Таким образом, была рассмотрена математическая модель задачи выбора наиболее подходящего кандидата на должность. Она является моделью линейного упорядочивания, где всякому объекту x_i сопоставлен некоторый показатель p_i , который оценивает результаты его сравнений с другими объектами, затем объекты составляются в порядке убывания значений p_i . Задачи с данной математической моделью можно решить методами из теории принятия решений [1].

1.5 Методы решения задачи выбора на основе экспертной оценки

Связи с использованием качественной информации, для решения задачи выбора наилучшего кандидата необходимо применение методов принятия решений, основанных на экспертном мнении. Из них наиболее распространенными являются метод нечеткой логики и метод анализа иерархий (МАИ). Рассмотрим и выберем лучший из них для решения многокритериальной задачи в условиях неопределенности.

Нечеткая логика («fuzzy logic») построена на теории нечетких множеств, которую в 1965 году разработал ученый Лотфи Заде. По сравнению с классической, где всякий элемент принадлежит одному множеству, в нечеткой логике элемент может принадлежать множеству с некоторой степенью принадлежности. Она выражается действительным числом между нулем и единицей и вычисляется с помощью функции принадлежности. Чем выше эта степень, тем более элемент соответствует свойствам этого множества. С этой точки зрения данная логика напоминает человеческие рассуждения, которые часто нечетки и включают не только два значения

истинности, а могут находиться в их диапазоне.

В методе нечеткой логики для создания нечеткой системы необходимо определение всех функций принадлежности: знания экспертов кодируются в нечеткие правила. В этом состоит главный и существенный недостаток данного метода [6]. Метод нечеткой логики является непростым для понимания и интерпретации, чем традиционные методы принятия решений, тем усложняет процесс экспертной оценки, так как требует от специалиста подготовки и опыта в этой области.

Метод анализа иерархий тоже один из распространенных подходов к принятию решений в условиях неопределенности, основанный на экспертных оценках. Он был разработан математиком Томасом Саати в 1970 году и до сих пор активно применяется в управлении. Этот метод универсален и используется для решения многих задач в различных областях (например, в бизнесе, промышленности, здравоохранении). МАИ заключается в иерархической декомпозиции проблемы на более мелкие простые части и затем обработки суждений с помощью парного сравнения.

Метод анализа иерархий имеет ряд преимуществ. Одним из них является то, что он за счет применения иерархии задачи отражает способ мышления человека (выделить объекты и установить связь между ними), это помогает адекватно и полно выявить предпочтения ЛПР. Иерархия позволяет описать влияние элементов задачи одного уровня над элементами другого уровня, а также отличается в устойчивости (небольшие изменения в иерархии не способствуют ее разрушению). Однако недостатком иерархий может быть ее громоздкость.

В МАИ существует способ выявления противоречий мнения эксперта, что повышает качество получаемого результата. Они возникают, когда различные критерии или альтернативы не согласуются. Например, эксперт принимает, что первый фактор превосходит второй, а второй фактор, в тоже время, превосходит третий фактор. Тогда при сравнении третий фактор не должен превосходить первый фактор, иначе логика в суждениях эксперта

нарушается, что может препятствовать принятию обоснованного решения.

Метод анализа иерархий предоставляет возможность использовать и количественные, и качественные критерии одновременно путем применения шкал сравнения. Это преимущество позволяет решать многие реальные задачи принятия решений, где присутствуют оба вида критериев.

Недостатком МАИ может быть отсутствие средств, проверяющих достоверность данных. Однако этот метод применяется в основном, когда может и вовсе не быть объективных данных, а главными при принятии решений являются предпочтения людей. Объемные вычисления также затрудняют использование метода анализа иерархий.

Таким образом, рассмотрев методы принятия решений на основе экспертной оценки, учитывая их преимущества и недостатки, для задачи выбора кандидата на должность наиболее подходящим из них является МАИ.

1.6 Исследование метода анализа иерархий

Метод анализа иерархий применяется для решения многокритериальных задач в условиях неопределенности. Этот метод строится на следующих принципах.

Принцип декомпозиции заключается в организации проблемы в форме иерархии. С помощью нее возможно получение отношений между элементами задачи.

В общем случае иерархия является полной (каждый элемент уровня связан с каждым элементом следующего уровня) и содержит три уровня:

- цель (главный критерий);
- критерии;
- альтернативы.

На рисунке 2 представлена обобщенная иерархия проблемы.

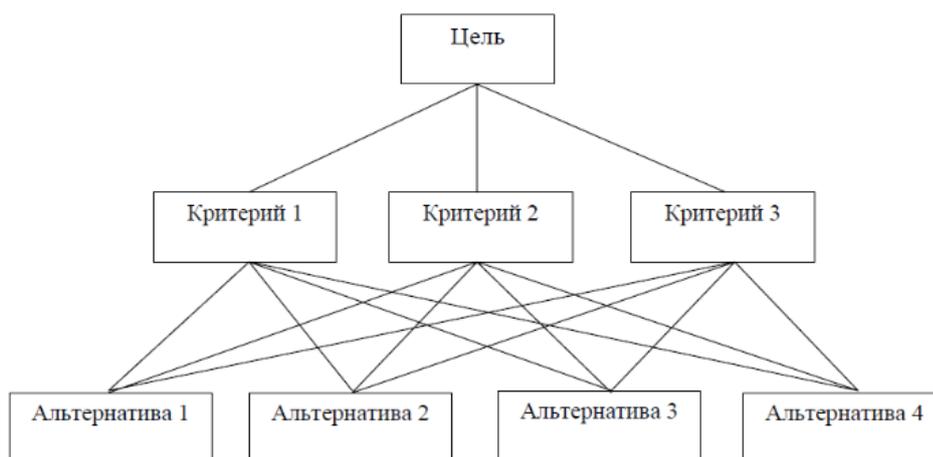


Рисунок 2 – Иерархия в методе анализа иерархий

Принцип сравнительных суждений состоит в попарном сравнении критериев и альтернатив задачи относительно их важности в задаче принятия решений.

В методе анализа иерархий применяют матричную алгебру. После попарных сравнений образуется квадратная матрица A , которая является положительной и обратнo-симметричной. В процессе определения локальных приоритетов вычисляется собственный вектор $v \neq 0$ такой, что выполняется равенство $Av = \lambda v$, в соответствии которому поставлено собственное значение (число) λ этой матрицы. Обычно нахождение λ , сводится к решению характеристического уравнения $|A - \lambda E| = 0$, где E обозначает единичную матрицу, то есть квадратную матрицу с единицами на главной диагонали и с остальными элементами равными нулю. У матрицы размера $n \times n$ имеется ровно n собственных значений.

В МАИ используют главное собственное значение λ_{max} и соответствующий ему собственный вектор v . Эти понятия объясняет следующая теорема.

Теорема Перрона. Пусть дана положительная матрица A , тогда:

- матрица A имеет действительное, положительное собственное значение, которое по модулю строго больше любого другого

собственного значения [15];

- собственный вектор матрицы A , соответствующий собственному значению, имеет положительные компоненты и, с точностью до постоянного множителя, единственен [15].

Так как точное вычисление главного собственного вектора v может быть трудоемко, МАИ предлагает использовать приближенные методы расчета. Один из них включает нахождение среднего геометрического для каждой i -ой строки матрицы парного сравнения A (размерностью n)

$$v_i = \left(\prod_{j=1}^n a_{ij} \right)^{\frac{1}{n}}. \quad (6)$$

Затем для нормирования собственного вектора каждой матрицы необходимо вычислить коэффициент

$$r = \sum_{i=1}^n v_i. \quad (7)$$

В итоге, получается вектор локальных приоритетов (вектор весов критериев или кандидатов)

$$w_i = \frac{v_i}{r}. \quad (8)$$

В методе анализа иерархий контролируется согласованность матрицы для выявления противоречий в суждениях эксперта. Для этого вычисляется собственное значение λ_{max} , которое соответствует вектору v

$$\lambda_{max} = \sum_{i=1}^n \left(\left(\sum_{j=1}^n a_{ij} \right) \cdot w_i \right). \quad (9)$$

Далее определяют индекс согласованности (ИС) по формуле

$$ИС = \frac{\lambda_{max} - n}{n - 1}. \quad (10)$$

Матрица A является согласованной, когда главное собственное число $\lambda_{max} = n$. Следовательно, чем ближе величина λ_{max} к n , тем более непротиворечивы суждения эксперта [21].

Затем ИС нужно сравнить с той же величиной, только полученной при случайном выборе суждений, которая называется случайным индексом согласованности (ИС). Этот показатель известен и представлен в таблице 2.

Таблица 2 – Значения случайного индекса согласованности

n	1	2	3	4	5	6	7	8	9	10
СИ	0	0	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,49

Формула для нахождения отношения согласованности:

$$ОС = \frac{ИС}{СИ}. \quad (11)$$

Допустимым является значение $ОС \leq 0,1$. Если для матрицы парных сравнений $ОС > 0,1$, то суждения эксперта считаются несогласованными и следует пересмотреть матрицу [9].

Принцип синтеза приоритетов предполагает вычисление локальных приоритетов, выражающих относительное влияние элементов заданного уровня на элемент уровнем выше, и дальнейший их синтез для получения глобальных приоритетов (приоритетов альтернатив).

Таким образом, был рассмотрен метод анализа иерархий, который основан на экспертных оценках и широко применяется при решении задач в условиях неопределенности.

1.7 Разработка алгоритма принятия решений

На основе математической модели и метода анализа иерархий возможно решение многокритериальной задачи выбора в условиях неопределенности. Необходимо выполнить следующие шаги:

- определение проблемы и формулирование цели принятия решения;
- выбор критериев и альтернатив;
- формирование иерархии. Необходимо структурировать задачу, начиная с цели решения (самый верхний уровень), проходя через уровни критериев и заканчивая уровнем альтернатив;
- построение матриц парных сравнений. Критерии сравниваются относительно цели, альтернативы – относительно критериев. На данном этапе эксперт для оценки критериев и альтернатив пользуется качественной девятибалльной шкалой (таблица 1);
- определение локальных приоритетов путем вычисления собственных векторов матриц парных сравнений;
- оценка согласованности суждений эксперта;
- вычисление глобальных приоритетов с помощью синтеза приоритетов.

Опираясь на рассмотренные выше этапы, разработан алгоритм принятия решений, блок-схема которого представлена на рисунке 3.

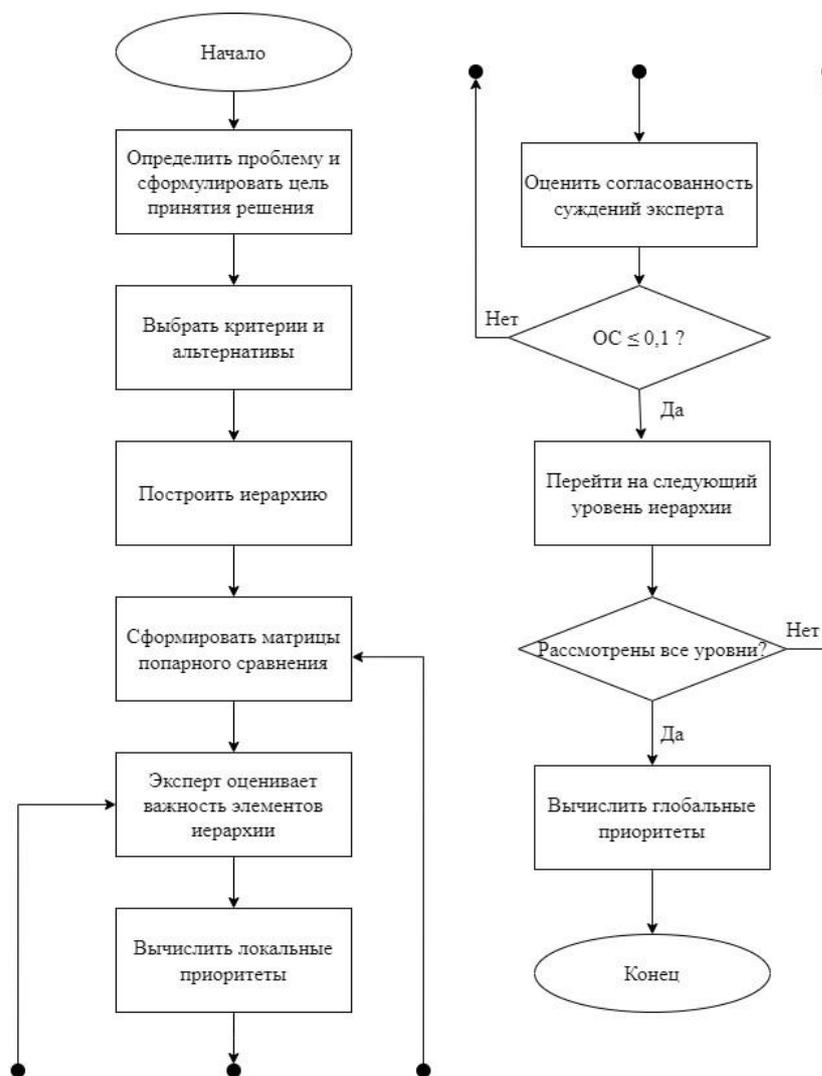


Рисунок 3 – Блок-схема алгоритма принятия решений

Далее применим построенный алгоритм для решения многокритериальной задачи выбора кандидата на должность.

1.8 Решение задачи выбора кандидата на должность

Цель задачи – выбрать наилучшего кандидата на должность. Пусть необходимо выбрать одного из пяти кандидатов, которые оцениваются по критериям из пункта 1.3:

Кандидат 1:

- уровень образования: специалитет;

- уровень владения английским языком: средний;
- опыт работы: менее одного года;
- технические навыки: уверенный компьютерный пользователь.
- профессиональные навыки: отлично выполнил практическое задание.

Кандидат 2:

- уровень образования: среднее профессиональное;
- уровень владения английским языком: продвинутый;
- опыт работы: пять лет;
- технические навыки: продвинутый компьютерный пользователь;
- профессиональные навыки: хорошо выполнил практическое задание.

Кандидат 3:

- уровень образования: бакалавриат;
- уровень владения английским языком: начальный;
- опыт работы: один год;
- технические навыки: уверенный компьютерный пользователь;
- профессиональные навыки: удовлетворительно выполнил практическое задание.

Кандидат 4:

- уровень образования: специалитет;
- уровень владения английским языком: средний;
- опыт работы: двенадцать лет;
- технические навыки: начинающий компьютерный пользователь;
- профессиональные навыки: отлично выполнил практическое задание.

Кандидат 5:

- уровень образования: магистратура;
- уровень владения английским языком: начальный;

- опыт работы: семь лет;
- технические навыки: продвинутый компьютерный пользователь;
- профессиональные навыки: хорошо выполнил практическое задание.

С помощью разработанного алгоритма решим данную многокритериальную задачу. Для начала необходимо построить иерархию из определенных ранее ее элементов (рисунок 4).

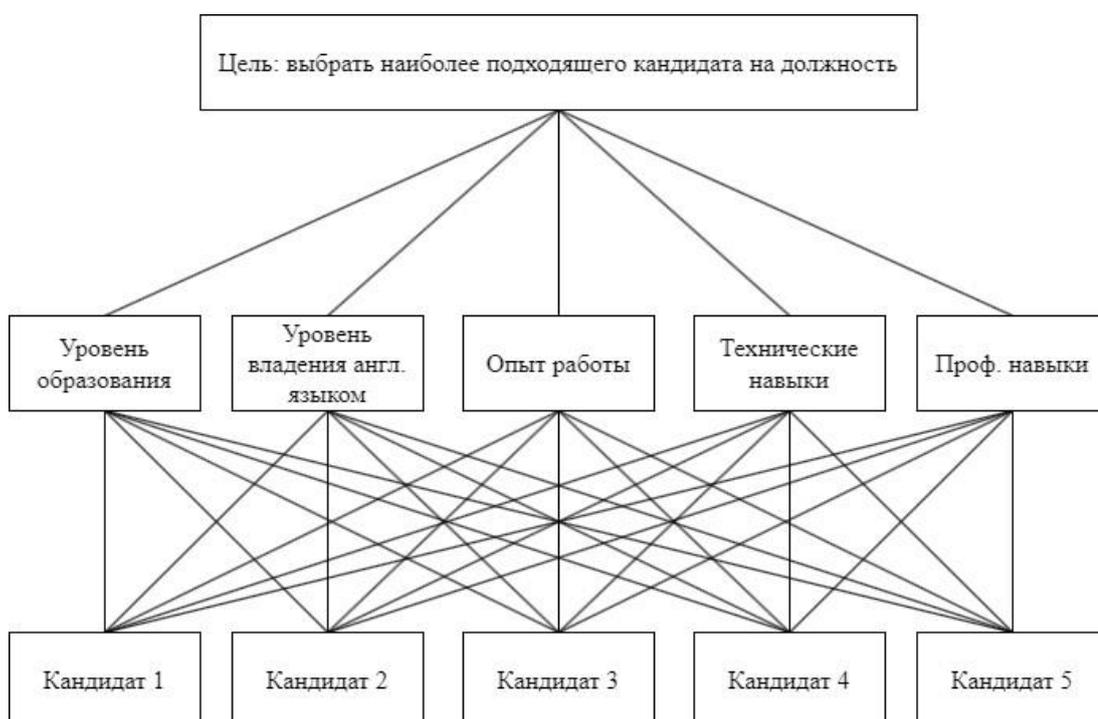


Рисунок 4 – Представление проблемы задачи в виде иерархии

Далее эксперт выполняет попарное сравнение всех элементов иерархии согласно девятибалльной шкале (пункт 1.4), для каждой из полученных матриц (таблицы 3-8) вычисляется вектор локальных приоритетов, главное собственное значение λ_{max} . Важно, чтобы матрицы попарных сравнений были согласованы. Проверка на непротиворечивость суждений происходит при вычислении отношения согласованности. Ранее в пункте 1.6 были рассмотрены все необходимые для этого формулы.

Таблица 3 – Матрица сравнения критериев относительно цели

	Уровень образования	Уровень владения англ. яз.	Опыт работы	Тех. навыки	Проф. навыки
Уровень образования	1,000	9,000	5,000	8,000	3,000
Уровень владения англ. яз.	0,111	1,000	0,200	0,333	0,167
Опыт работы	0,200	5,000	1,000	3,000	0,500
Тех. навыки	0,125	3,000	0,333	1,000	0,250
Проф. навыки	0,333	6,000	2,000	4,000	1,000

Главный собственный вектор: $V = (4,043; 0,262; 1,084; 0,500; 1,741)$.

Вектор локальных приоритетов: $W = (0,530; 0,034; 0,142; 0,066; 0,228)$.

Главное собственное значение: $\lambda_{max} = 5,167$.

Индекс согласованности: ИС = 0,042.

Отношение согласованности: ОС = 0,037.

Таблица 4 – Матрица попарного сравнения альтернатив относительно критерия «Уровень образования»

	Кандидат 1	Кандидат 2	Кандидат 3	Кандидат 4	Кандидат 5
Кандидат 1	1,000	9,000	5,000	1,000	2,000
Кандидат 2	0,111	1,000	0,167	0,125	0,143
Кандидат 3	0,200	6,000	1,000	0,333	0,500
Кандидат 4	1,000	8,000	3,000	1,000	2,000
Кандидат 5	0,500	7,000	2,000	0,500	1,000

Главный собственный вектор: $V = (2,460; 0,201; 0,725; 2,169; 1,285)$.

Вектор локальных приоритетов: $W = (0,360; 0,029; 0,106; 0,317; 0,188)$.

Главное собственное значение: $\lambda_{max} = 5,105$.

Индекс согласованности: ИС = 0,026.

Отношение согласованности: ОС = 0,023.

Таблица 5 – Матрица попарного сравнения альтернатив относительно критерия «Уровень владения английским языком»

	Кандидат 1	Кандидат 2	Кандидат 3	Кандидат 4	Кандидат 5
Кандидат 1	1,000	0,333	5,000	1,000	5,000
Кандидат 2	3,000	1,000	7,000	3,000	7,000
Кандидат 3	0,200	0,143	1,000	0,200	1,000
Кандидат 4	1,000	0,333	5,000	1,000	5,000
Кандидат 5	0,200	0,143	1,000	0,200	1,000

Главный собственный вектор: $V = (1,528; 3,380; 0,356; 1,528; 0,356)$.

Вектор локальных приоритетов: $W = (0,223; 0,494; 0,052; 0,223; 0,052)$.

Главное собственное значение: $\lambda_{max} = 5,356$.

Индекс согласованности: ИС = 0,090.

Отношение согласованности: ОС = 0,080.

Таблица 6 – Матрица попарного сравнения альтернатив относительно критерия «Опыт работы»

	Кандидат 1	Кандидат 2	Кандидат 3	Кандидат 4	Кандидат 5
Кандидат 1	1,000	0,333	0,500	0,143	0,200
Кандидат 2	3,000	1,000	3,000	0,200	0,250
Кандидат 3	2,000	0,333	1,000	0,167	0,200
Кандидат 4	7,000	5,000	6,000	1,000	2,000
Кандидат 5	5,000	4,000	5,000	0,500	1,000

Главный собственный вектор: $V = (0,343; 0,852; 0,467; 3,347; 1,187)$.

Вектор локальных приоритетов: $W = (0,050; 0,125; 0,068; 0,489; 0,320)$.

Главное собственное значение: $\lambda_{max} = 5,442$.

Индекс согласованности: ИС = 0,110.

Отношение согласованности: ОС = 0,100.

Таблица 7 – Матрица попарного сравнения альтернатив относительно критерия «Технические навыки»

	Кандидат 1	Кандидат 2	Кандидат 3	Кандидат 4	Кандидат 5
Кандидат 1	1,000	0,250	1,000	6,000	0,250
Кандидат 2	4,000	1,000	4,000	8,000	1,000
Кандидат 3	1,000	0,250	1,000	6,000	0,250
Кандидат 4	0,167	0,125	0,167	1,000	0,125
Кандидат 5	4,000	1,000	4,000	8,000	1,000

Главный собственный вектор: $V = (0,822; 2,639; 0,822; 0,213; 2,639)$.

Вектор локальных приоритетов: $W = (0,120; 0,386; 0,120; 0,031; 0,386)$.

Главное собственное значение: $\lambda_{max} = 5,371$.

Индекс согласованности: ИС = 0,090.

Отношение согласованности: ОС = 0,080.

Таблица 8 – Матрица попарного сравнения альтернатив относительно критерия «Профессиональные навыки»

	Кандидат 1	Кандидат 2	Кандидат 3	Кандидат 4	Кандидат 5
Кандидат 1	1,000	4,000	8,000	1,000	4,000
Кандидат 2	0,250	1,000	6,000	0,250	1,000
Кандидат 3	0,125	0,167	1,000	0,125	0,167
Кандидат 4	1,000	4,000	8,000	1,000	4,000
Кандидат 5	0,250	1,000	6,000	0,250	1,000

Главный собственный вектор: $V = (2,639; 0,822; 0,213; 2,639; 0,822)$.

Вектор локальных приоритетов: $W = (0,386; 0,120; 0,031; 0,386; 0,120)$.

Главное собственное значение: $\lambda_{max} = 5,371$.

Индекс согласованности: ИС = 0,090.

Отношение согласованности: ОС = 0,080.

Необходимо выполнить иерархический синтез для нахождения глобального вектора приоритетов согласно формуле (4)

$$\begin{pmatrix} 0,360 & 0,223 & 0,050 & 0,120 & 0,386 \\ 0,029 & 0,494 & 0,125 & 0,386 & 0,120 \\ 0,106 & 0,052 & 0,068 & 0,120 & 0,031 \\ 0,317 & 0,223 & 0,489 & 0,031 & 0,386 \\ 0,188 & 0,052 & 0,320 & 0,386 & 0,120 \end{pmatrix} \cdot \begin{pmatrix} 0,53 \\ 0,034 \\ 0,142 \\ 0,066 \\ 0,228 \end{pmatrix} = \begin{pmatrix} 0,301 \\ 0,103 \\ 0,083 \\ 0,335 \\ 0,199 \end{pmatrix}.$$

Согласно формуле (5) оптимальным является альтернатива с максимальным значением приоритета, то есть «Кандидат 4».

Таким образом, был рассмотрен конкретный пример работы алгоритма для решения многокритериальной задачи в условиях неопределенности.

Вывод по первой главе. Данная глава посвящена исследованию предметной области ВКР. В результате анализа проблемы выбора кандидата на должность в ООО «Стратегия Роста», был найден подход к принятию обоснованного решения с помощью теории принятия решений, поставлена многокритериальная задача в условиях неопределенности.

Для формализации данной задачи и возможности применения математических методов была построена математическая модель на основе выбранных критериев оценки кандидатов.

В результате исследования методов принятия решений в условиях неопределенности, наиболее подходящим оказался метод анализа иерархий. На основе построенной математической модели и МАИ разработан алгоритм, который был применен для решения конкретной задачи. Процесс вычислений вручную занял много времени, поэтому необходимо его автоматизировать.

Глава 2 Программная реализация алгоритма принятия решений

2.1 Функциональное моделирование приложения для выбора кандидата на должность

Отбор кандидатов был ранее представлен в виде многокритериальной задачи, где каждый критерий вносит свой вклад в формирование оценки кандидата на должность. С ростом количества критериев оценки, задача выбора решения для ЛПР усложняется. По этой причине предлагается программная реализация алгоритма выбора кандидатов на вакантную должность.

Перед проектированием приложения необходим анализ требований, который заключается в общении с заказчиком для понимания его потребностей, в сборе и изучении информации. В итоге был сформирован следующий список. Приложение должно быть:

- разработано на языке программирования Java;
- предназначено для использования на персональных компьютерах;
- иметь понятный и удобный графический интерфейс пользователя.

Функциональными требованиями называют конкретные утверждения о том, что нужно делать приложению. К ним относятся:

- построение матрицы попарного сравнения для критериев (ввод и корректировка элементов матрицы);
- построение матриц попарного сравнения для альтернатив (ввод и корректировка элементов матриц);
- проверка матриц попарного сравнения на согласованность, что включает расчет главного собственного значения, индекса согласованности и отношения согласованности (ОС), сопоставление ОС с допустимыми значениями, возможность изменения элементов матриц в случае несогласованности и повторный расчет показателей;

- вычисление вектора локальных приоритетов матриц попарного сравнения: расчет и вывод на экран;
- синтез локальных приоритетов, то есть расчет и вывод вектора глобальных приоритетов;
- визуализация результатов: построение круговой диаграммы на основе полученного вектора глобальных приоритетов, вывод наиболее подходящего кандидата.

Модель IDEF0 (Integration Definition for Function Modeling) является инструментом для моделирования процессов и их взаимосвязей во время проектирования приложений. Главным элементом нотации IDEF0 является графический язык, который передает определенную информацию.

Данная модель визуально представляется в виде диаграмм, состоящих из пронумерованных блоков, каждый из которых изображается в виде прямоугольников и олицетворяет собой определенную функцию приложения. Согласно нотации IDEF0 название блока должно включать глагол (например, «построить матрицы попарного сравнения»).

С каждой стороны прямоугольника находятся стрелки: верхние обозначают управление (правила, стандарты или методы), нижние – механизм (персонал, оборудование и другие ресурсы). С левой стороны блока располагаются стрелки входа, а с правой стороны – стрелки выхода. Они включают материалы или данные, которые соответственно преобразуются или производятся функцией.

Контекстная диаграмма с обозначением «А-0», является самой верхней, объект моделирования на ней определяется одним блоком, который описывает главное назначение приложения. На рисунке 5 представлена данная диаграмма для проектируемого приложения.

Построение диаграмм модели IDEF0 базируется на принципе декомпозиции: сложный процесс разбивается на составляющие его функции. Декомпозиция позволяет постепенно и структурированно представлять модель системы в виде иерархической структуры отдельных диаграмм, что

делает ее менее перегруженной и легко усваиваемой [14]. Результат декомпозиции контекстной диаграммы изображен на рисунке 6.



Рисунок 5 – Контекстная диаграмма

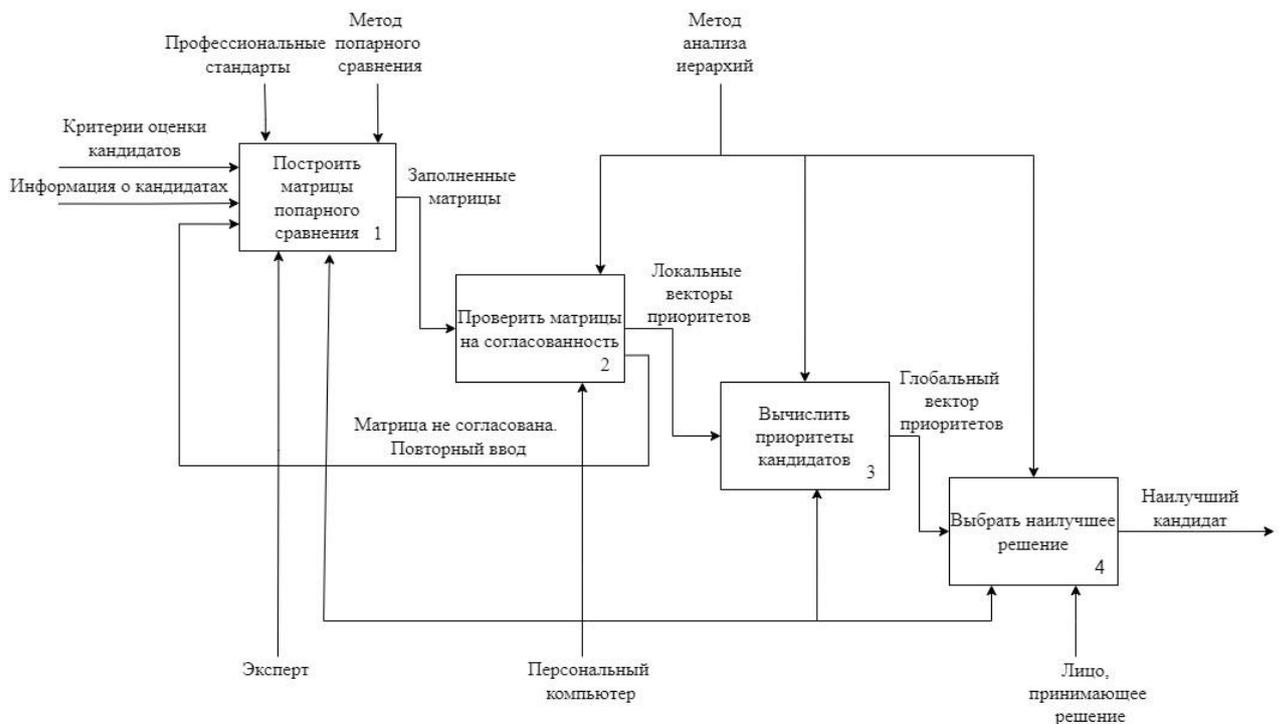


Рисунок 6 – Диаграмма первого уровня декомпозиции

Из функциональной диаграммы (рисунок 6) понятно, что персональный компьютер необходим для реализации всех функций. При построении матриц попарного сравнения также участвует эксперт (HR-менеджер), который вводит оценки в матрицы, исходя из информации о кандидатах и критериях. Проверяются матрицы на согласованность и вычисляются приоритеты исключительно с помощью персонального компьютера. Он также выводит лучшее с его точки зрения решение, но окончательный выбор все равно делает лицо, принимающее решение (руководитель компании).

Применение диаграмм IDEF0 способствовало более подробному и структурированному описанию функциональных требований заказчика, подчеркнуть ключевые процессы и взаимосвязи между ними.

Таким образом, необходимо разработать приложение для выбора кандидата на должность, которое соответствует построенной функциональной модели IDEF0, а, следовательно, и требованиям заказчика. Программа должна иметь графический интерфейс пользователя, поэтому далее нужно выбрать средства его разработки.

2.2 Выбор программных средств создания графического интерфейса приложения

Для разработки графического пользовательского интерфейса на языке программирования Java существует множество библиотек, из которых наиболее популярными являются Java Swing и JavaFX. Обе библиотеки дают возможность создавать интерактивные приложения с графическим интерфейсом, но имеют ряд различий. Необходимо рассмотреть преимущества и недостатки Java Swing и JavaFX и выбрать наиболее подходящую из них.

Библиотека Java Swing была разработана в конце 1990-х годов компанией Sun Microsystems. Предлагается большой выбор компонентов (кнопок, текстовых полей, таблиц, списков и многое другое) для построения

графических интерфейсов, настройки внешнего вида и обработки событий [7]. Можно выявить несколько преимуществ, таких как:

- Java Swing не требует установки, так как поставляется вместе с платформой Java SE;
- подробная документация и широкая база пользователей, что упрощает поиск решений проблем;
- низкое потребление системных ресурсов, используется небольшой объем оперативной памяти.

Из недостатков библиотеки Java Swing следует отметить устаревающую архитектуру и низкую производительность, в связи с этим рекомендуется применять более современные технологии для создания графического интерфейса.

В настоящее время библиотека JavaFX является более актуальной, технология предоставляет большие возможности для создания кроссплатформенных приложений с графическим интерфейсом. Основные преимущества этой библиотеки:

- JavaFX позволяет разработчикам писать понятный, управляемый код на Java, который удобно обновлять и отлаживать;
- существует встроенная поддержка архитектурного шаблона Model-View-Controller (MVC) для работы над разными аспектами приложения независимо друг от друга;
- возможность применения языка разметки FXML для упрощения работы с графическим интерфейсом [19].

Данная технология пришла на замену устаревающей Java Swing, поэтому очевиден выбор в пользу JavaFX. Далее для дальнейшей программной реализации необходимо подробнее рассмотреть принцип работы с этой библиотекой.

Архитектурный шаблон MVC, тесно связанный с JavaFX, помогает организовать программный код и предлагает разделить его на три компонента. Модель (Model) хранит классы данных, методы для работы с

ними и бизнес-логику. Этот компонент является самым независимым в системе, то есть не должен ничего знать о других модулях. Представление (View), отвечает за отображение данных пользователю. Контроллер (Controller) необходим для обработки пользовательского ввода, управления взаимодействия между моделью и представлением. Взаимодействие компонентов шаблона MVC представлено на рисунке 7.

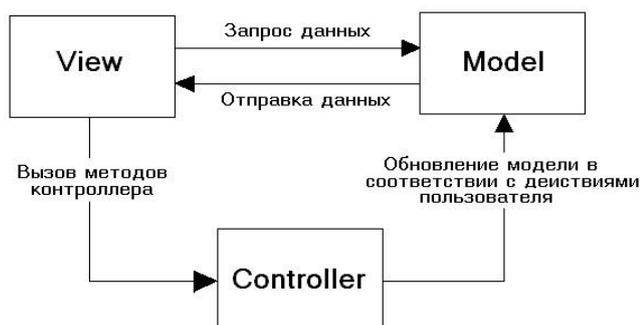


Рисунок 7 – Компоненты шаблона MVC и их взаимодействие

В Представлениях часто используют язык разметки FXML, созданный на основе языка XML. Описание графических компонентов приложения происходит декларативно, то есть предоставляются инструкции о желаемом результате, а система сама выбирает способы достижения этого результата [20].

Таким образом, была выбрана библиотека для создания графического интерфейса JavaFX и рассмотрен принцип создания архитектуры приложения с помощью шаблона MVC и языка разметки FXML. Следующим этапом будет написание исходного кода программы.

2.3 Реализация алгоритма и интерфейса приложения на Java

В пункте 1.7 рассмотрен алгоритм принятия решений, а в пункте 2.1 построена функциональная модель, на основе которых будет строиться приложение для выбора кандидата на должность. Была определена

последовательность действий в программе, поэтому можно приступить к программной реализации.

Разработку программного кода приложения с графическим интерфейсом пользователя следует начать с построения непосредственно его структуры и дизайна. Создадим макет интерфейса приложения, чтобы согласовать образ программы с заказчиком и определиться с его компонентами, с их расположением.

Алгоритм предполагает построение матрицы попарного сравнения, то есть необходима таблица, в которую пользователь будет вводить оценки с помощью переключателей и шкалы с бегунком. Для вывода значений локального вектора приоритетов и названий объектов также применяется таблица, а для рассчитанных показателей главного собственного значения, индекса согласованности и отношения согласованности – метки. Переход между страницами осуществляется с помощью кнопок «Назад» и «Далее» (рисунок 8).

Попарное сравнение альтернатив относительно фактора уровень образования

Какая альтернатива предпочтительнее?

Альтернатива 1 Альтернатива 2

Оцените степень предпочтения:

Абсолютное превосходство
Сильное превосходство
Умеренное превосходство
Слабое превосходство
Равная важность

№	Названия	1	2	3	4	5	Приоритет
1	Кандидат 1						
2	Кандидат 2						
3	Кандидат 3						
4	Кандидат 4						
5	Кандидат 5						

$\lambda =$ ИС = ОС =
max

Рисунок 8 – Макет интерфейса формы экспертного оценивания

Вывод результатов работы программы происходит через таблицу, которая состоит из значений глобального вектора приоритетов. Для иллюстрации применим круговую диаграмму. Наиболее подходящий

кандидат будет определен компонентом метка. На рисунке 9 изображена данная спроектированная форма.



Рисунок 9 – Макет интерфейса формы вывода результатов

С помощью визуального редактора Scene Builder созданы в режиме конструктора элементы интерфейса, определены их свойства и расположение на экране. В результате этого этапа были разработаны файлы с разрешением «.fxml». Они представляют собой представления в шаблоне MVC и содержат описание структуры интерфейса на языке FXML (рисунок 10).

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.control.Button?>
4 <?import javafx.scene.control.Label?>
5 <?import javafx.scene.control.RadioButton?>
6 <?import javafx.scene.control.Slider?>
7 <?import javafx.scene.control.TableView?>
8 <?import javafx.scene.control.ToggleGroup?>
9 <?import javafx.scene.layout.AnchorPane?>
10 <?import javafx.scene.text.Font?>
11
12 <AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="540.0"
13 <children>
14 <Label layoutX="56.0" layoutY="116.0" text="Какой из критериев предпочтительнее?">
15 <font>
16 <Font size="14.0" />
17 </font></Label>
18 <Label layoutX="80.0" layoutY="208.0" text="Оцените степень предпочтения:">
19 <font>
20 <Font size="14.0" />
21 </font></Label>
22 <Button fx:id="NextButton" layoutX="146.0" layoutY="55.0" mnemonicParsing="false" onAction="#NextButtonClick"
23 <font>

```

Рисунок 10 – Фрагмент кода представления приложения

Далее пишется код для обработки событий, возникающих при взаимодействии пользователя с интерфейсом (нажатие кнопки, перемещение бегунка и так далее). Разработанный код хранится в классах Java, называемыми контроллерами, которые связываются с FXML файлами (представлениями). На рисунке 11 представлен фрагмент кода контроллера, отвечающий за обработку событий переключателей.

```
121 @FXML
122 void ClickedRadioButton() {
123
124     if(RadioButton1.isSelected()){ // Проверяет, выбран ли первый переключатель
125         chg = false;
126         SliderCrit.setDisable(false); // Разблокирует элемент
127     }
128     else if(RadioButton2.isSelected()){ // Проверяет, выбран ли второй переключатель
129         chg = true;
130         SliderCrit.setDisable(false); // Разблокирует элемент
131     }
132 }
```

Рисунок 11 – Фрагмент кода контроллера приложения

Следующим этапом необходимо написать программный код для управления бизнес-логикой программы, то есть создать класс, называемый моделью. Класс «Criterion» содержит данные о матрице попарных сравнений и результаты вычислений связанных с ней параметров (рисунок 12), а также методы для управления ими.

```
1 package application;
2
3 public class Criterion {
4
5     double[][] criterion; // матрица попарных сравнений
6     double[] vCriterion; // среднее геометрическое каждой i-ой строки матрицы
7     double rCriterion; // нормирующий коэффициент
8     double[] wCriterion; // вектор приоритетов матрицы
9     double[] sCriterion; // сумма каждого j-го столбца матрицы
10    double[] pCriterion; // результат умножения вектора s на вектор w
11    double hmax; // главное собственное значение
12    double IS; // индекс согласованности
13    double OS; // отношение согласованности
14 }
```

Рисунок 12 – Данные класса «Criterion»

Основной метод «change(int i, int j, double val)» принимает на вход выбранное пользователем новое значение элемента, номер его строки, столбца и реализует заполнение матрицы попарного сравнения. С помощью вызова вспомогательного метода «calc()» рассчитываются нужные для алгоритма параметры в соответствии с формулами (6)-(11). На рисунке 13 изображена блок-схема метода «change», а на рисунке 14 – программный код.

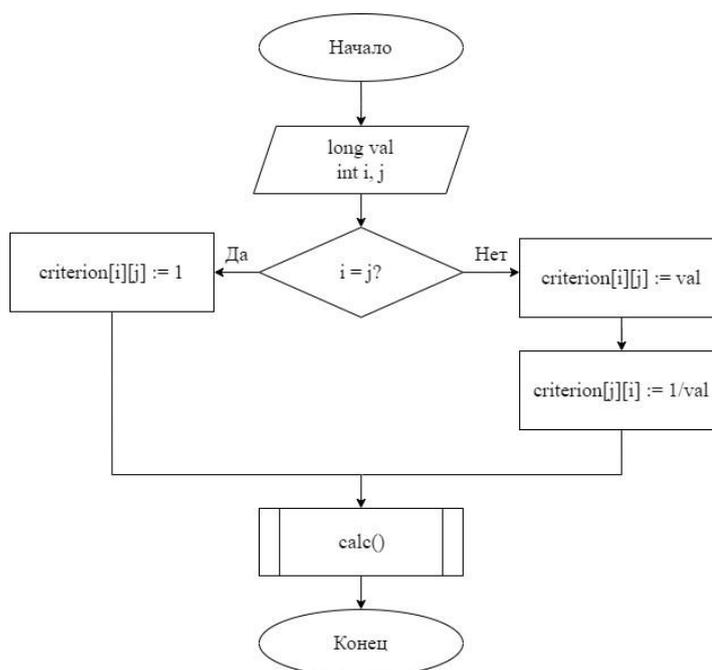


Рисунок 13 – Блок-схема метода «change»

```

55 // заполнение введенных пользователем симметричных относительно главной диагонали элементов матрицы
56 public void change(int i, int j, double val){
57     criterion[i][j] = val;
58     criterion[j][i] = Math.round((1.0/val)*1000.0)/1000.0;
59
60     calc();
61 }
62
63 // вычисление локального вектора приоритетов, hmax, IS, OS
64 public void calc(){
65     vCriterion = v(criterion);
66     rCriterion = r(vCriterion);
67     wCriterion = w(vCriterion, rCriterion);
68     sCriterion = s(criterion);
69     pCriterion = p(sCriterion, wCriterion);
70     hmax = hmax(pCriterion);
71     IS = IS(hmax, pCriterion);
72     OS = OS(pCriterion, IS);
73 }
74
  
```

Рисунок 14 – Программный код методов «change» и «calc»

В технологии JavaFX класс «Main» содержит методы, отвечающие за запуск графического интерфейса. Также здесь создается метод «calcQ()» (рисунок 15), который выполняет заключительный шаг алгоритма – вычисление глобального вектора приоритетов согласно (4).

```

public static void calcQ(){
    double[] wCriterion = c[0].getwCriterion(); // матрица сравнения критериев
    double[] wCriterionK1 = c[1].getwCriterion(); // матрица сравнения кандидатов по первому критерию
    double[] wCriterionK2 = c[2].getwCriterion(); // матрица сравнения кандидатов по второму критерию
    double[] wCriterionK3 = c[3].getwCriterion(); // матрица сравнения кандидатов по третьему критерию
    double[] wCriterionK4 = c[4].getwCriterion(); // матрица сравнения кандидатов по четвертому критерию
    double[] wCriterionK5 = c[5].getwCriterion(); // матрица сравнения кандидатов по пятому критерию

    // вычисление глобального вектора приоритетов
    Q1 = wCriterionK1[0] * wCriterion[0] + wCriterionK2[0] * wCriterion[1] + wCriterionK3[0] * wCriterion[2]
    + wCriterionK4[0] * wCriterion[3] + wCriterionK5[0] * wCriterion[4];
    Q2 = wCriterionK1[1] * wCriterion[0] + wCriterionK2[1] * wCriterion[1] + wCriterionK3[1] * wCriterion[2]
    + wCriterionK4[1] * wCriterion[3] + wCriterionK5[1] * wCriterion[4];
    Q3 = wCriterionK1[2] * wCriterion[0] + wCriterionK2[2] * wCriterion[1] + wCriterionK3[2] * wCriterion[2]
    + wCriterionK4[2] * wCriterion[3] + wCriterionK5[2] * wCriterion[4];
    Q4 = wCriterionK1[3] * wCriterion[0] + wCriterionK2[3] * wCriterion[1] + wCriterionK3[3] * wCriterion[2]
    + wCriterionK4[3] * wCriterion[3] + wCriterionK5[3] * wCriterion[4];
    Q5 = wCriterionK1[4] * wCriterion[0] + wCriterionK2[4] * wCriterion[1] + wCriterionK3[4] * wCriterion[2]
    + wCriterionK4[4] * wCriterion[3] + wCriterionK5[4] * wCriterion[4];
}

```

Рисунок 15 – Программный код метода «calcQ»

Таким образом, были рассмотрены этапы программной реализации алгоритма, начиная от проектирования, заканчивая описанием основных методов кода. После разработки приложения необходимо его протестировать, выявить неполадки в работе и устранить их.

2.4 Тестирование разработанного приложения

Завершением программной реализации приложения для выбора кандидата на должность будет его тестирование. Необходимо проверить соответствие ранее рассмотренным требованиям (пункт 2.1) и правильность полученных результатов, поэтому рассмотрим работу программы на исходных данных из пункта 1.7.

После запуска приложения появляется форма, в которой можно заполнить матрицу попарного сравнения критериев. Переключатели и шкала

с бегунком на этом этапе работы с приложением скрыты, пользователь нажимает на ячейку таблицы, переключатели активируются, он выбирает из двух критериев более предпочтительный, становится доступной шкала с бегунком. После выбора значение заносится в нужную ячейку, одновременно с этим изменяется и значение симметричной относительно главной диагонали ячейки. Также вычисляются необходимые показатели: вектор локальных приоритетов, главное собственное значение, индекс согласованности (ИС) и отношение согласованности (ОС) (рисунок 16).

Выбор кандидата на должность

Далее ->

Парное сравнение критериев относительно фактора цель

Какой из критериев предпочтительнее?

Ур. образования Ур. владения англ. яз.

Оцените степень предпочтения:

Абсолютное превосходство

Сильное превосходство

Умеренное превосходство

Слабое превосходство

Равная важность

№	Названия	1	2	3	4	5	Приоритет
1	Ур. образования	1.0	9.0	5.0	8.0	3.0	0.53
2	Ур. владения англ. яз.	0.111	1.0	0.2	0.333	0.167	0.034
3	Опыт работы	0.2	5.0	1.0	3.0	0.5	0.142
4	Тех. навыки	0.125	3.0	0.333	1.0	0.25	0.066
5	Проф. навыки	0.333	6.0	2.0	4.0	1.0	0.228

* Для сравнения критериев клик на ячейке матрицы сравнения

$\lambda_{\max} = 5,166$ ИС = 0,042 ОС = 0,037

Рисунок 16 – Форма для заполнения матрицы критериев

При недопустимом значении ОС, соответствующая метка загорается красным цветом (рисунок 17).

Выбор кандидата на должность

Далее ->

Парное сравнение критериев относительно фактора цель

Какой из критериев предпочтительнее?

Ур. владения англ. яз. Тех. навыки

Оцените степень предпочтения:

Абсолютное превосходство

Сильное превосходство

Умеренное превосходство

Слабое превосходство

Равная важность

№	Названия	1	2	3	4	5	Приоритет
1	Ур. образования	1.0	1.0	1.0	1.0	1.0	0.194
2	Ур. владения англ. яз.	1.0	1.0	1.0	7.0	1.0	0.286
3	Опыт работы	1.0	1.0	1.0	1.0	1.0	0.194
4	Тех. навыки	1.0	0.143	1.0	1.0	1.0	0.132
5	Проф. навыки	1.0	1.0	1.0	1.0	1.0	0.194

* Для сравнения критериев клик на ячейке матрицы сравнения

$\lambda_{\max} = 5,544$ ИС = 0,136 ОС = 0,121

Рисунок 17 – Значение ОС превышает допустимое

Пользователь нажимает на кнопку «Далее», появляется похожая форма только для формирования матрицы попарного сравнения кандидатов по первому критерию (рисунок 18).

Выбор кандидата на должность

<- Назад Далее ->

Попарное сравнение альтернатив относительно фактора уровень образования

Какая альтернатива предпочтительнее?

Кандидат 1 Кандидат 3

Оцените степень предпочтения:

Абсолютное превосходство

Сильное превосходство

Умеренное превосходство

Слабое превосходство

Равная важность

№	Названия	1	2	3	4	5	Приоритет
1	Кандидат 1	1.0	1.0	3.0	1.0	1.0	0.247
2	Кандидат 2	1.0	1.0	1.0	1.0	1.0	0.198
3	Кандидат 3	0.333	1.0	1.0	1.0	1.0	0.159
4	Кандидат 4	1.0	1.0	1.0	1.0	1.0	0.198
5	Кандидат 5	1.0	1.0	1.0	1.0	1.0	0.198

* Для сравнения альтернатив по критерию клик на ячейке матрицы сравнения

$\lambda_{\max} = 5,153$ ИС = 0,038 ОС = 0,034

Рисунок 18 – Форма для заполнения матрицы кандидатов по критерию «Уровень образования»

Пользователь с помощью кнопок «Далее» и «Назад» переходит от одной формы заполнения матриц к другой, пока не отобразятся результаты расчета алгоритма. Вычисленный глобальный вектор приоритетов представлен в таблице, ниже определен кандидат с наивысшим значением, для удобства восприятия результаты оформлены в виде круговой диаграммы (рисунок 19).

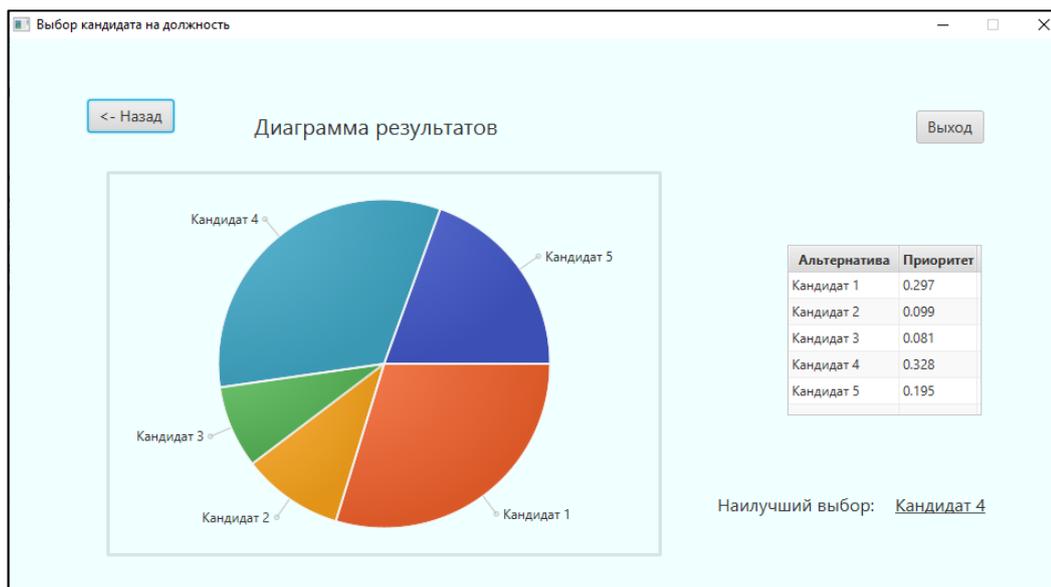


Рисунок 19 – Форма для вывода результатов

Из круговой диаграммы видно, что наиболее предпочтительной является альтернатива «Кандидат 4». Такой же результат был и получен при расчете в пункте 1.7. Возможны небольшие отклонения числовых значений, так как при вычислениях вручную производилось округление до тысячных.

Таким образом, был изложен принцип взаимодействия пользователя с приложением на конкретном примере, рассмотрены результаты работы программы, которые оказались аналогичны полученным вручную. Из этого следует, что все требования заказчика к приложению выполнены: приложение выполняет заявленные ранее функции, пользователю интуитивно понятно и удобно работать с компонентами графического интерфейса.

Вывод по второй главе. В данной главе рассмотрен процесс программной реализации алгоритма выбора кандидата на должность, который начинается с выявления требований заказчика.

Построение диаграмм IDEF0 способствовало более подробному и структурированному описанию функциональных требований, подчеркнуть ключевые процессы и взаимосвязи между ними. Для определения структуры и дизайна графического интерфейса были созданы макеты и согласованы с заказчиком.

Анализ программных средств создания графического интерфейса приложения показал преимущество библиотеки JavaFX. Рассмотрен принцип создания архитектуры приложения с помощью шаблона MVC и языка разметки FXML для дальнейшего написания кода.

В итоге, создано приложение для выбора кандидата на должность на Java на основе ранее разработанного алгоритма принятия решений. Тестирование программы показало, что она работает верно, соответствует требованиям заказчика.

Заключение

Выпускная квалифицированная работа посвящена построению математической модели и разработке алгоритма принятия решений для HR-менеджеров в условиях неопределенности.

Во время выполнения ВКР была исследована предметная область. Для этого проанализирована проблема выбора кандидата на должность в ООО «Стратегия Роста», выявлен подход к принятию обоснованного решения с помощью теории принятия решений. После постановки многокритериальной задачи выбора была построена математическая модель, которая структурировала и формализовала данный бизнес-процесс. Также на основе математической модели и метода анализа иерархий разработан алгоритм принятия решений.

С ростом количества критериев оценки, задача выбора решения для ЛПР усложняется. По этой причине предложена программная реализация разработанного алгоритма выбора кандидатов на вакантную должность. На основе выявленных требований заказчика была спроектирована функциональная модель для отображения производимых приложением действий. Для определения структуры и дизайна графического интерфейса были созданы макеты и согласованы с заказчиком.

В результате было создано приложение с графическим интерфейсом пользователя для выбора кандидата на вакантную должность, при помощи которого можно быстро и эффективно принимать обоснованные решения. Тестирование программы показало, что она соответствует требованиям заказчика и выводит правильные значения.

Таким образом, все задачи были выполнены, а, следовательно, цель выпускной квалифицированной работы достигнута.

Список используемой литературы

1. Богданова П.А. Обзор методов многокритериальной оптимизации в задачах принятия решений / П. А. Богданова, Т. В. Васильева // Инновационные аспекты развития науки и техники: Материалы IX Международной научно-практической конференции, Саратов, 31 мая 2021 года / НОО «Цифровая наука». – М.: “КДУ”, “Добросвет”, 2021. – С. 153-157. URL: <https://www.elibrary.ru/item.asp?id=46624415> (дата обращения: 01.02.2024).
2. Волкова В. Н. Теория систем и системный анализ: учебник для вузов / В.Н. Волкова, А.А. Денисов. – Москва: Издательство Юрайт, 2024. – 562с.
3. Гребнева О.А. Теория принятия решений: учебное пособие / О.А. Гребнева. – Иркутск: Издательство ИРНИТУ, 2019. – 164 с.
4. Егорушкина Т. Н. Проблемы применения математических моделей при принятии решений в условиях неопределенности / Т. Н. Егорушкина, Е. В. Этова, Т. Б. Белякова // European science forum: Сборник статей V Международного научного форума, Петрозаводск, 07 октября 2020 года. – Петрозаводск: Международный центр научного партнерства «Новая Наука», 2020. – С. 25-29. URL: <https://elibrary.ru/item.asp?id=44060810> (дата обращения: 15.02.2024).
5. Елиферов В.Г. Бизнес-процессы: регламентация и управление: учебник / В.Г. Елиферов, В.В. Репин. – Москва: Издательство ИНФРА-М, 2019. – 319 с.
6. Зак Ю. А. Принятие решений в условиях нечетких и размытых данных: Fuzzy-Технологии / Ю.А. Зак. – Москва: Издательство: Ленанд, 2022. – 352 с.
7. Замышляев В.П. Проблема актуальности графических интерфейсов на Java / В.П. Замышляев, Л.Е. Попок // Информационное общество: современное состояние и перспективы развития: сборник

материалов XII международного форума, Краснодар, 15–20 июля 2019 года. – Краснодар: ФГБОУ ВО «Кубанский государственный аграрный университет имени И. Т. Трубилина», 2019. – С. 320-321. URL: <https://www.elibrary.ru/item.asp?id=41133831&pf=1> (дата обращения: 01.04.2024).

8. Качала В.В. Общая теория систем и системный анализ: учебник для вузов / В.В. Качала. – Москва: Издательство Горячая линия-Телеком, 2020. – 204 с.

9. Кузьмин, А. М. Метод анализа иерархий – один из инструментов выработки и принятия решений / А. М. Кузьмин, Е. А. Высоковская // Методы менеджмента качества. – 2019. – № 7. – С. 55. URL: <https://www.elibrary.ru/item.asp?id=38334390> (дата обращения: 08.02.2024).

10. Лиманова Н.И. Теория принятия решений: учебное пособие / Н.И. Лиманова. – Самара: ПГУТИ, 2020. –198 с.

11. Лихолетов В.В. Управление предприятием (организацией). Методы решения задач и принятия управленческих решений: учебное пособие / В.В. Лихолетов, Я.Д. Гельруд. – Челябинск: Издательский центр ЮУрГУ, 2020. – 360 с.

12. Определение интерфейса в FXML [Электронный ресурс]. URL: <https://metanit.com/java/javafx/2.1.php> (дата обращения: 11.04.2024).

13. Подиновский В.В. Многокритериальные задачи принятия решений: теория и методы анализа: учебник для вузов / В.В. Подиновский. – Москва: Издательство Юрайт, 2024. – 486 с.

14. Подиновский В.В. Идеи и методы теории важности критериев в многокритериальных задачах принятия решений / В.В. Подиновский. – Москва: Издательство Наука, 2019. – 103 с.

15. Постановка задачи принятия решения [Электронный ресурс]. URL: http://systematy.ru/articles/41_postanovka_zadachi_prinyatiya_resheniya (дата обращения: 07.02.2024).

16. Проектирование информационных систем [Электронный ресурс]. URL: <https://intuit.ru/studies/courses/16088/55/lecture/1630> (дата обращения: 05.04.2024).
17. Турлыгул Н. К. Многокритериальный выбор в условиях неопределённости / Н. К. Турлыгул // Синергия Наук. – 2020. – № 54. – С. 754-769. URL: <https://elibrary.ru/item.asp?id=44793235> (дата обращения: 12.03.2024).
18. Формирование множества критериев [Электронный ресурс]. URL: https://studme.org/292579/informatika/formirovanie_mnozhestva_kriteriev (дата обращения: 12.02.2024).
19. Шихалёв А.М. Многокритериальные задачи принятия решений в управлении качеством: учебное пособие / А.М. Шихалёв, И.И. Хафизов. – Казань: Издательство Казанского университета, 2020. – 92 с.
20. Щебленкова К.А. Алгоритм разработки цифрового HR-продукта в организации / К.А. Щебленкова, С.В. Кошевенко // Экономика и социум. 2020. – № 4(71). – С. 1171-1173. URL: <https://www.elibrary.ru/item.asp?id=43360472> (дата обращения: 21.01.2024).
21. Mitra M. Fuzzy Logic Approach to Assess Web Learner's Joint Skills / M. Mitra, A. Das // International Journal of Modern Education and Computer Science. – 2015. – Vol. 7, No. 9. – P. 14-21. URL: <https://elibrary.ru/item.asp?id=44874767> (дата обращения: 23.01.2024).
22. Mouludi I. Decision Support System Menggunakan Analytic Hierarchy Process dan Analytical Network Process Pada Pemilihan Supplier Bahan Baku / I. Mouludi, M. Ramdhanti, F. Jamsan // INVENTORY: Industrial Vocational E-Journal On Agroindustry. – 2022. – Vol. 3, No. 1. – P. 26. URL: <https://elibrary.ru/item.asp?id=59199587> (дата обращения: 01.02.2024).
23. Schildt H. Java: The Complete Reference, Thirteenth Edition / H. Schildt, D. Coward: McGraw-Hill Education, 2023. – 1280 p.
24. Sharan K. Learn JavaFX 17: Building User Experience and Interfaces with Java, Second Edition / K. Sharan: Apress, 2022. – 942 p.

25. Schiavon L. L. P. Use of the Analytic Hierarchy Process Method in the Variety Selection Process for Sugarcane Planting / L. L. P. Schiavon, P. A. B. Lima, A. F. Crepaldi, E. B. Mariano // Eng. – 2023. – Vol. 4, No. 1. – P. 602-614. URL: <https://elibrary.ru/item.asp?id=61226027> (дата обращения: 05.03.2024).