

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

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

02.03.03 МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ И АДМИНИСТРИРОВАНИЕ
ИНФОРМАЦИОННЫХ СИСТЕМ

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

БАКАЛАВРСКАЯ РАБОТА

на тему «Автоматизация процесса распределения персонала (на примере Самарской гуманитарной академии)»

Студент	С.С. Васильева	_____
Руководитель	А.И. Туищев	_____
Консультант по аннотации	Н.В. Яценко	_____

Допустить к защите
Заведующий кафедрой к.тех.н., доцент, А.В. Очеповский _____

« _____ » _____ 2017 г.

Тольятти 2017

АННОТАЦИЯ

Название бакалаврской работы «Автоматизация процесса распределения персонала (на примере Самарской гуманитарной академии)».

Объектом бакалаврской работы является процесс распределения персонала. Предметом бакалаврской работы является автоматизация процесса распределения персонала. Цель работы – программная реализация автоматизированной системы управления процессом распределения персонала.

В бакалаврской работе исследуется процесс распределения персонала. Особое внимание уделяется возможности математического моделирования этого процесса.

Создается и описывается математическая задача оптимального распределения персонала. Затем проводится анализ программных аналогов для выявления их достоинств и недостатков.

Разрабатывается концептуальная модель процесса распределения персонала. Далее происходит разработка интерфейса программного продукта.

Результатом работы является программная реализация автоматизированной системы управления процессом распределения персонала. Результаты исследования показывают, что программа оказала положительное влияние на организацию процесса распределения персонала в организации.

Данная бакалаврская работа состоит из пояснительной записки на 67 стр., включая 26 рисунков, 7 таблиц, списка 20 источников, в том числе 5 источников на иностранном языке и 3 приложения.

ABSTRACT

The title of the given graduation work is «Automation of the process of personnel distribution (on the example of the Samara Humanitarian Academy)».

The object of the graduation work is the process of staff distribution. The subject of the graduation work is the automation of the process of staff distribution. The aim of the work is the software implementation of an automated system for managing the process of staff distribution.

In the graduation work, the process of staff distribution is explored. Much attention is given to the possibility of mathematical modeling of this process.

A mathematical problem of the optimal distribution of personnel is created and described. Then, software analogs are analyzed for revealing their advantages and disadvantages.

A conceptual model of the staff distribution process is developed. Next, the interface of the software product is created.

The result of the work is the software implementation of the automated personnel distribution management system. The results of the study show that program had a positive impact on organization of the process of staff distribution in the enterprise.

The graduation work consists of an explanatory note on 67 pages, including 26 figures, 7 tables, the list of 20 references including 5 foreign sources and 3 appendices.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
Глава 1 АНАЛИЗ ИСПОЛЬЗОВАНИЯ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА РАСПРЕДЕЛЕНИЯ ПЕРСОНАЛА	6
1.1 Техничко-экономическая характеристика образовательного учреждения	6
1.2 Анализ принципов управления человеческими ресурсами для решения задачи оптимизации их распределения	9
1.3 Описание математической модели процесса оптимального распределения персонала на выполнение задач	14
1.3.1 Математическое моделирование процесса управления человеческими ресурсами	14
1.3.2 Постановка математической задачи о назначении	16
1.3.3 Решение задачи о назначении с использованием венгерского метода	22
1.3.4 Реализации задачи о назначении с использованием табличного редактора	25
1.4 Обзор программных аналогов	28
Вывод по первой главе	33
Глава 2 МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ ПЕРСОНАЛА НА ВЫПОЛНЕНИЕ ЗАДАЧ	34
2.1 Обзор и выбор средств программирования	34
2.2 Разработка концептуальной модели процесса распределения персонала .	39
2.3 Разработка интерфейса программного продукта	41
2.4 Описание проектируемой базы данных	44
2.5 Программная реализация автоматизации процесса распределения персонала	46
Вывод по второй главе	50
Глава 3 РЕАЛИЗАЦИЯ АВТОМАТИЗАЦИИ ПРОЦЕССА РАСПРЕДЕЛЕНИЯ ПЕРСОНАЛА В ОРГАНИЗАЦИИ	52

3.1 Описание работы программной реализации автоматизации процесса распределения персонала	52
3.2 Оценка и обоснование экономической эффективности программного приложения	55
Вывод по третьей главе	58
ЗАКЛЮЧЕНИЕ	59
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	60
ПРИЛОЖЕНИЕ А Код главного класса	62
ПРИЛОЖЕНИЕ Б Код основных функций программного приложения	63
ПРИЛОЖЕНИЕ В Конструкторы классов	67

ВВЕДЕНИЕ

Регулировка работы кадров – одна из наиболее важных сфер управления предприятием, направленная на обеспечение полного и гармоничного развития организации. Для успешного выполнения работы необходимо, чтобы ей занимался грамотный специалист, имеющий представление о сути и деталях своего дела.

В настоящее время наблюдается значительное сокращение рабочих мест, что увеличивает конкуренцию и, соответственно, усложняет процесс подбора персонала. При выборе нового сотрудника руководитель должен опираться не только на профессиональные качества кандидата, но и на его возможность и способность работать на конкретно рассматриваемой должности. При этом следует учитывать множество различных факторов, влияющих на дальнейшую работу специалиста, держать в голове качества каждого отдельно взятого претендента для их общего сравнения. Эти же критерии могут использоваться при дальнейшем отборе наиболее подходящих сотрудников на повышение или для поездки за рубеж от лица организации.

Необходимость учета различных факторов создает определенные трудности при подборе персонала и общего продвижения кадров в пределах организации. Современные методы работы с информацией позволяют значительно облегчить данный процесс. В связи с этим большое значение обретает задача автоматизации процесса распределения персонала.

Объект бакалаврской работы: процесс распределения персонала.

Предмет бакалаврской работы: автоматизация процесса распределения персонала.

Целью бакалаврской работы является программная реализация автоматизированной системы управления процессом распределения персонала.

Задачами бакалаврской работы, исходя из поставленной цели, являются:

- изучение аппарата математического моделирования;
- изучение процесса распределения персонала;

- программная реализация автоматизации процесса распределения персонала.

Первая глава работы посвящена анализу теоретических основ математического моделирования. Также изучаются различные системы автоматизации процесса распределения персонала.

Вторая глава описывает реализацию автоматизации процесса распределения персонала: алгоритм работы программы, формулирование требований к программе, логика программы.

В третьей главе представлена программная реализация автоматизации процесса распределения персонала.

Результатом работы будет являться программный продукт, являющийся реализацией автоматизации процесса распределения персонала в организации.

Глава 1 АНАЛИЗ ИСПОЛЬЗОВАНИЯ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА РАСПРЕДЕЛЕНИЯ ПЕРСОНАЛА

1.1 Техничко-экономическая характеристика образовательного учреждения

Самарская гуманитарная академия (СаГА) – это частное образовательное учреждение высшего образования, которое осуществляет подготовку по направлениям и специальностям на пяти разных факультетах: юриспруденция, экономика, психология, филология и менеджмент. Обучение осуществляется по очной, заочной и очно-заочной формам, в том числе с использованием дистанционных технологий и ускоренного обучения.

В общей сложности СаГА и ее филиалы насчитывают около 2500 студентов. В СаГА работают 184 преподавателя, в том числе 25 докторов наук и профессоров, 89 кандидатов наук.

Филиал Самарской гуманитарной академии в г. Тольятти был основан в 1994 году. Образовательная деятельность в Филиале осуществляется с 1995 года. Изначально в структуре Филиала были выделены два факультета – гуманитарный и экономико-правовой, реализовывавшие основные образовательные программы.

Информация о количестве преподавателей помогла заочно распределить границы работ по управлению человеческими ресурсами.

Целями образовательного процесса СаГА являются прежде всего:

- формирование общей культуры личности;
- адаптация личности к жизни в обществе;
- создание основы для осознанного выбора и освоения профессиональных образовательных программ;
- обеспечение качественного воспитания и подготовки обучающихся.

Структурная форма организации показана на рисунке 1.1.

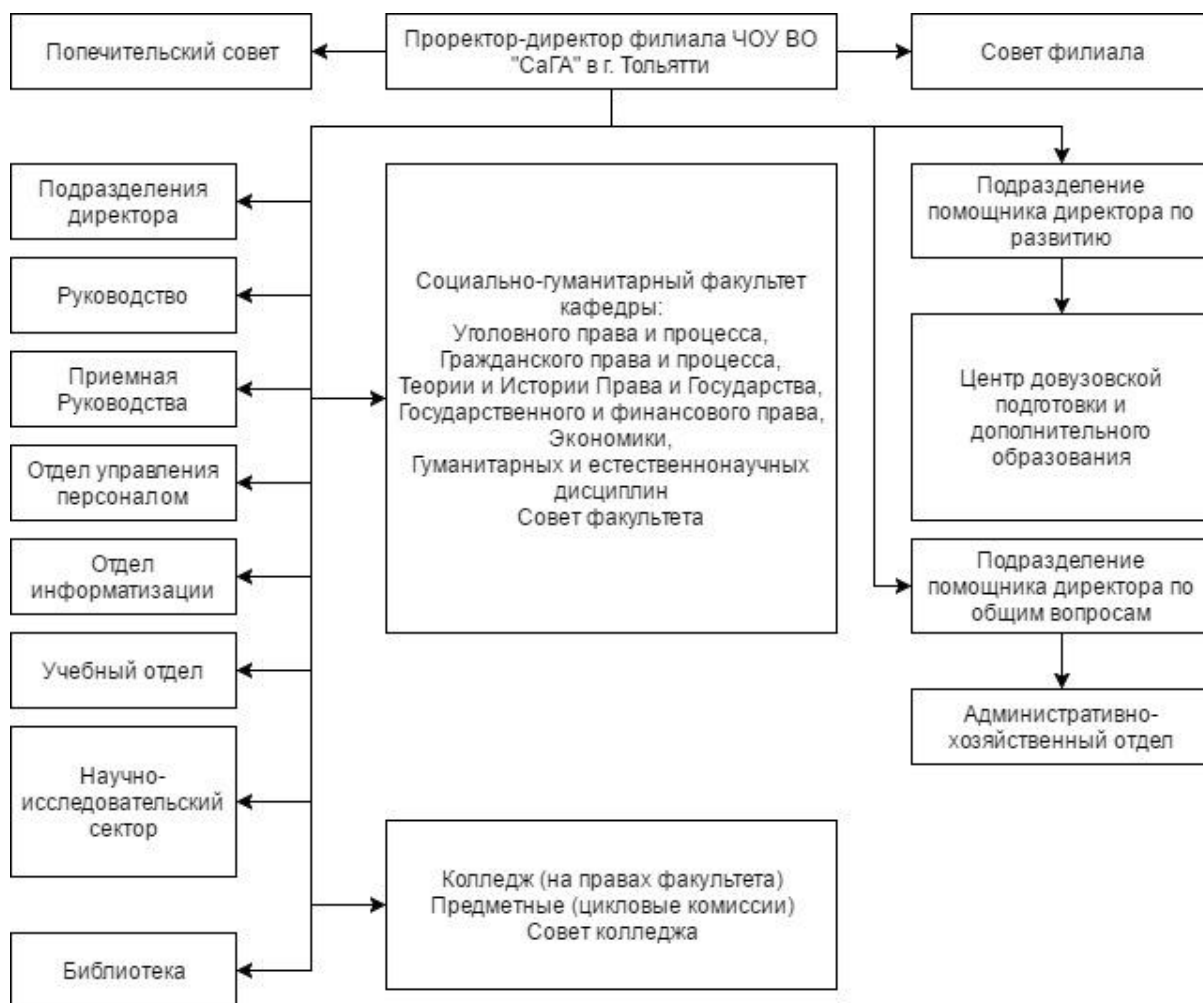


Рисунок 1.1 – Структурная форма организации

Управление деятельностью Филиала Самарской гуманитарной академии осуществляется директором Филиала, который назначается на должность приказом ректора Академии.

Директор Филиала, на основании доверенности, выданной ректором академии, руководит всей деятельностью Филиала, представляет его интересы и интересы Академии, заключает договоры и дает указания, обязательные для всех работников Филиала.

Директор Филиала по должности является членом Совета Академии. По усмотрению ректора Академии, лицо, работающее в должности Директора Филиала, может именоваться Директором Филиала – Проректором ЧОУ ВО «СаГА».

Ежегодно директор Филиала подготавливает и доводит до сведения ректора Академии отчет о работе Филиала, его учебной, учебно-методической, научной и хозяйственной деятельности. При назначении на должность директора Филиала с ним заключается трудовой договор, в котором определяются права, обязанности, условия оплаты его труда, срок действия, условия освобождения от занимаемой должности. Директор Филиала может быть освобожден от занимаемой должности до истечения срока действия трудового договора по основаниям, предусмотренным действующим законодательством РФ.

Директор Филиала исполняет следующие обязанности по управлению:

- по доверенности принимает на работу сотрудников Филиала, в том числе профессорско-преподавательский состав, заключает и расторгает с ними соответствующие договоры и издает приказы о приеме и увольнении на работу;
- подписывает договоры поручительства и кредитные договоры в кредитных учреждениях;
- издает распоряжения по вопросам деятельности Филиала, заключает договоры по вопросам финансово-хозяйственной деятельности Филиала, в том числе договоры аренды зданий и сооружений, не требующие нотариального удостоверения и государственной регистрации;
- организует учебно-методическое и материальное обеспечение учебного процесса в Филиале;
- вносит дополнения, касающиеся деятельности Филиала к правилам внутреннего трудового распорядка и правилам внутреннего распорядка Академии. Вносимые дополнения не должны противоречить соответствующим правилам Академии;
- осуществляет другую деятельность и совершает иные действия в пределах его компетенции в соответствии с предоставленными полномочиями.

Филиал частного образовательного учреждения «СаГА» в г. Тольятти насчитывает более 50 сотрудников.

На данный момент все решения о продвижении сотрудников, их повышениях и трудоустройстве новых кадров принимаются руководителем филиала. Данное обстоятельство значительно тормозит работу организации, т.к. требуется большое количество времени для изучения информации о заслугах кандидата, его опыте работы, возрасте, производительности, а также время на анализ данной информации относительно других сотрудников и принятие решения.

Соответственно, на данный момент в организации СаГА существует необходимость оптимизации процесса взаимодействия с сотрудниками компании.

1.2 Анализ принципов управления человеческими ресурсами для решения задачи оптимизации их распределения

Для оптимизации взаимодействия начальника с персоналом, необходимо изучить, что собой представляет данный процесс. Сотрудников в организации можно рассмотреть, как ресурс, обеспечивающий выполнение поставленных задач, а взаимодействие с ними можно представить, как процесс управления человеческими ресурсами.

Управление человеческими ресурсами – это особый подход к управлению персоналом организации, при котором люди рассматриваются как наиболее ценный ресурс в условиях конкурентной борьбы, при этом их надо мотивировать и развивать для достижения стратегических целей организации [3].

Основной целью управления человеческими ресурсами является обеспечение организации такими работниками, которые позволят наиболее эффективно достигать поставленных целей.

В современных условиях труда, когда сотрудники предприятия оказываются важнейшим ресурсом, одной из основных задач руководителей служб по управлению человеческими ресурсами состоит в формировании наиболее эффективной стратегии предприятия.

Конкретные цели управления человеческими ресурсами могут изменяться в зависимости от специфики целей организации.

На рисунке 1.2 приведены задачи, которые решает управление человеческими ресурсами в организации.



Рисунок 1.2 – Основные задачи управления человеческими ресурсами

Для решения обозначенных задач используют различные методы и подходы к управлению человеческими ресурсами в организации. Наиболее распространены подходы «рынок труда» и «человеческий капитал», специфика управления в которых определяется тремя методами: инвестиционным, организационно-правовым и социально-психологическим. Более подробно об этом писала Новаковская О.А. в учебном пособии «Управление человеческими ресурсами в неустойчивых деловых организациях» [7].

Подход к управлению с позиции модели «рынок труда» ориентирован на приспособление организации к изменениям внешней среды. В основе данной модели заключена рыночная концепция, которая исходит из того, что

человеческие ресурсы требуемого качества и количества могут быть найдены на рынке труда (таблица 1.1).

Таблица 1.1 – Зависимость деятельности от метода управления человеческими ресурсами в подходе «рынок труда»

Методы	Деятельность по управлению человеческими ресурсами
Инвестиционный	<ul style="list-style-type: none"> • Небольшие затраты на обучение и подготовку сотрудников, большая часть затрат направлена на внешний рынок (поиск сотрудников, подбор, расстановка). • Оценка на входе чисто формальная, оценивается в основном соответствие рабочему месту. • Обучение направлено на адаптацию к новому рабочему месту.
Организационно-правовой	<ul style="list-style-type: none"> • Краткосрочное трудоустройство. • Организация труда основывается на узкой специализации с индивидуальными заданиями и контролем. • Профессиональное продвижение является узкоспециализированным.
Социально-психологический	<ul style="list-style-type: none"> • В процессе привлечения используются в основном внешние стимулы, такие как реклама и имидж организации. • Для закрепления работников используются экономические факторы – высокая заработная плата.

Рыночный подход к трудоустройству сотрудников и развитию потенциала организации не придает особого значения функциям подбора, обучения и расстановки человеческих ресурсов. Он минимизирует издержки организации на поиск и набор человеческих ресурсов нужного профиля и квалификации.

Рыночный подход выгоден с финансовой точки зрения, однако наряду с низкими затратами на обслуживание трудоустройства, продвижения, развития человеческих ресурсов возникает потребность в частой замене работников.

В силу этого, рыночный подход, с одной стороны, сопровождается высоким оборотом человеческих ресурсов (так называемой «текучестью кадров»), и с другой – невысокой степенью удовлетворенности трудом и низкой склонностью к технологическим переменам, а также конфликтам между работниками, высокой стоимостью инноваций. Значительно уменьшается внимание к мотивационным, поведенческим, социальным факторам, всем тем вещам, которые создают организационную, корпоративную культуру.

В итоге, наряду с выгодой, получаемой при экономии на качестве человеческих ресурсов, их обучении и правильной расстановке, возникают

проблемы с низкой производительностью, качеством работы и необходимостью частой замены сотрудников.

В подходе с позиции «человеческий капитал» важнейшие механизмы связаны с внутренними вопросами развития человеческих ресурсов. Концепция «человеческого капитала» подразумевает, что человеческие ресурсы требуемого профессионального уровня необходимо развивать собственными средствами обучения непосредственно внутри самого предприятия.

Организации, меняющие свою деятельность под воздействием внешних условий, а не только приспособляющиеся к ним, связывают основной успех с развитием человеческих ресурсов. Классический подход «человеческий капитал» приведен в таблице 1.2.

Таблица 1.2 – Зависимость деятельности от метода управления человеческими ресурсами в подходе «человеческий капитал»

Методы	Деятельность по управлению человеческими ресурсами
Инвестиционный	<ul style="list-style-type: none"> • Крупные вклады в обучение и общую подготовку. Большая часть затрат направлена на общее и специальное образование сотрудников. • При подборе дается неформализованная оценка работоспособности, готовности принять нормы организации способности к групповой деятельности. • Ведется общее, специальное обучение, обучение с элементами группового поведения.
Организационно-правовой	<ul style="list-style-type: none"> • Долгосрочное или пожизненное трудоустройство сотрудников, групповые формы организации труда, которые обеспечивают максимальное использование потенциала человека. • Неспециализированное профессиональное развитие по вертикали и горизонтали.
Социально-психологический	<ul style="list-style-type: none"> • Внутренние стимулы преданности фирме, такие как признание культурных ценностей организации. • Оплата труда ставится в зависимости от стажа работы в организации. • Участие сотрудников в управлении, прибылях.

Ориентация на развитие человеческих ресурсов сопровождается, прежде всего, их низкой оборачиваемостью, низкой стоимостью и высокой эффективностью инноваций. Также процесс работы включает слабую

отчуждаемость сотрудников, добросовестность в учебе, их будущее связывается с «карьерной лестницей» организации. Высокие издержки, связанные с постоянным обучением и ростом квалификации, высокая мобильность человеческих ресурсов внутри организации быстро компенсируются творческой приспособляемостью к технологическим и организационным новшествам.

Данные подходы к управлению человеческими ресурсами не являются взаимоисключающими, наоборот, в подавляющем большинстве случаев на практике наиболее часто наблюдается совмещение этих методов, что позволяет сочетать экономическую выгоду и ориентированность на качество работы сотрудников.

Исходя из представленной информации можно сделать вывод, что процесс управления человеческими ресурсами – это сложная структура, для представления которой требуется использование определенных инструментальных решений. Далее будет рассматриваться только область распределения персонала.

1.3 Описание математической модели процесса оптимального распределения персонала на выполнение задач

1.3.1 Математическое моделирование процесса распределения персонала

Процесс распределения персонала возможно представить, используя математическую модель. Это позволит упростить условное представление процесса руководства кадрами компании и облегчит разработку программного продукта.

Вид модели и степень ее детализации определяется свойствами моделируемого объекта и целью, с которой выполняется моделирование. Поэтому процесс разработки модели состоит в пошаговом анализе и моделировании отдельных ее подсистем с последующим установлением связей между этими подсистемами. Подробно об этом писали Егоров А.Ф., Бельков В.П. и Савицкая Т.В. Бельков В.П., Савицкая Т.В. при описании курса

«Математическое моделирование и методы синтеза гибких химических производств» [1].

Процесс построения математической модели управления человеческими ресурсами представлен на рисунке 1.3.



Рисунок 1.3 – Процесс создания математической модели

В начале построения модели происходит определение рассматриваемого объекта моделирования в качестве системы – выделяются основные признаки, которые характеризуют систему и ее элементы, а также отношения, на которых происходит реализация этих признаков [19, 20].

Далее определяется цель моделирования системы. После чего происходит разработка математической модели, разделенная на несколько уровней детализации.

На первом, содержательном этапе построения модели создается концептуальная или содержательная модель – изучается исходная система и происходит ее содержательное описание [18]. Такое описание называют

концептуальной моделью, представляющей собой словесную формулировку математической задачи.

На структурном или топологическом этапе определяются балансы и ограничения проектируемой системы, происходит математическая постановка задачи [4, 9].

Третий этап называется алгоритмическим или функциональным. На этом этапе строится алгоритм решения описанной модели.

Программная реализация созданного алгоритма происходит на четвертом уровне детализации – параметрическом, где определяются параметры модели.

На последнем этапе происходит проверка адекватности модели объекту моделирования [16, 17].

На основе этой информации будет строиться математическая модель задачи о назначении преподавателя на решение определенной задачи, сформулированной руководителями.

1.3.2 Постановка математической задачи о назначении

Рассмотрим применение данного алгоритма для решения задачи о назначении сотрудника образовательного учреждения (преподавателя) на выполнение определенной задачи, с которой он может справиться максимально быстро и эффективно. Для этого сформулируем задачу оптимизации в форме математической модели булева линейного программирования.

Постановка задачи: перед руководителем стоит n видов задач, которые могут быть выполнены n преподавателями, n – натуральное число. При этом каждый преподаватель может выполнить только одну задачу, а каждая задача может быть выполнена только одним сотрудником. Были найдены эффективности C_{ij} ($\forall i, j \in 1, 2 \dots n$) выполнения каждым преподавателем всех рассматриваемых задач. Требуется так распределить преподавателей на все задачи, чтобы общая эффективность выполнения была наилучшей.

$$\begin{aligned}
x_{11} + x_{12} + \dots + x_{1r} &= 1; \\
x_{21} + x_{22} + \dots + x_{2r} &= 1; \\
&\dots \\
x_{m1} + x_{m2} + \dots + x_{mr} &= 1; \\
x_{11} + x_{21} + \dots + x_{m1} &= 1; \\
x_{12} + x_{22} + \dots + x_{m2} &= 1; \\
&\dots \\
x_{1r} + x_{2r} + \dots + x_{mr} &= 1; \\
x_{ij} \in 0,1, \quad \forall i \in 1,2 \dots r, j \in 1,2 \dots m.
\end{aligned}
\tag{6}$$

Для нахождения целевой функции необходимо рассчитать показатели эффективности, которые будут использоваться в задаче о назначении. Для расчетов будет использоваться информация, представленная в виде базы данных Excel с рядом характеристик того или иного работника (рисунок 1.4).

1	2	3	4	5	6	7	8	9	10
Персональный состав педагогических работников									
№ п/п	Ф.И.О.	Должность	Преподаваемые дисциплины	Ученая степень	Ученое звание	Направлен ие подготовк и и (или) специальности	Повышение квалификации и (или) профессиональная подготовка	общий стаж работы	стаж работы по специальности
1.	Баранов Андрей Андреевич	Доцент кафедры экономика	Введение в профессию; Менеджмент; Практика по получению первичных профессиональных умений и навыков; Преддипломная практика; Управление человеческими ресурсами; Экономика; Государственная итоговая аттестация	к.э.н.	доцент	Специальность- промышленное и гражданское строительство	2013 г. РГСУ, программа переподготовки «Правовое обеспечение деятельности организации», 510 часов, Диплом ППЗ № 004940; 2013 г. РГСУ, программа «Инновационные модели обеспечения качества высшего образования в современном образовательном комплексе», 72 часа, Удостоверение №36040;	38	14
2.	Бурда Светлана Юрьевна	Доцент кафедры естественна учных и гуманитарных дисциплин	Конфликтология; Основы консультативной психологии; Педагогика; Практикум специализации; Психолингвистика; Психология рекламы; Социальная психология; Основы консультативной психологии; Психология семьи; Этнопсихология	к.псх.н.		Специальность- психология	2015 г. Стажировка в "Coral clab International" представительстве в г. Тольятти. Справка о прохождении стажировки; 2016 г. ЧОУ ВО «СаГА» Основы работы в системе дистанционного обучения «MOODLE», 72 часа. Удостоверение 632401650515	36	5
3.	Быков Сергей Владимирович	Профессор	Методологические основы	д.псх.н.	доцент	Специальность	2015 г. Стажировка в "Coral clab	34	16

Рисунок 1.4 – Исходный вид данных

Представим в виде множества $S = (s_1, s_2, \dots, s_d)$ критерии, положенные в основу оценки сотрудника. Данным, по которым возможно произвести оценку, присваивается числовой коэффициент, имеющий определенный вес в дальнейших расчетах.

Например, в графе «Ученая степень» следует учесть, является сотрудник кандидатом или доктором наук; при расчете оптимальных показателей учена

степень кандидата наук имеет коэффициент «1», доктора – коэффициент «2». Сила коэффициента меняется для каждого отдельного сотрудника.

В таблице 1.3 представлены коэффициенты исходных данных, которые будут положены в основу вычисления функции поиска эффективности кандидата при выполнении конкретной задачи.

Таблица 1.3 – Присвоение коэффициентов исходным данным

Критерий	Исходный вид	Присваиваемый коэффициент	Описание
Ученая степень	Нет данных	0	Отсутствие ученой степени приравнивается к нулевому показателю рассматриваемого критерия
	к.н.	1	Кандидат наук – ученая степень первой ступени, присуждается диссертационным советом. Для получения достаточно положительного решения совета
	д.н.	2	Доктор наук – ученая степень второй, высшей ступени. Для получения требуется положительное решение диссертационного совета и положительное заключение экспертного совета
Ученое звание	Нет данных	0	Отсутствие ученого звания приравнивается к нулевому показателю рассматриваемого критерия
	Доцент	1	Ученое звание доцента присваивается работникам высших учебных заведений за научно-педагогическую деятельность
	Профессор	2	Ученое звание профессора присваивается работникам высших учебных заведений за научно-педагогическую деятельность и подготовку аспирантов
	доцент-академик	3	Доцент, входящий в состав российской академии наук
	профессор-академик	4	Профессор, входящий в состав российской академии наук

Продолжение таблицы 1.3.

Преподаваемые дисциплины	Прокурорский надзор; Судебный контроль за предварительным следствием; Производственная практика	3	Подсчитывается количество преподаваемых дисциплин
Участие в конференциях	Участие в научно-практической конференции "Актуальные проблемы теории и практики применения уголовного закона"	1	Информация о количестве участия в конференциях берется из описания сотрудника

Информация о научных, учебных и производственных «титулах» сотрудников вуза взята на сайте Сазонова В. Ф. «Кинезиолог» [14].

Для столбцов «Общий стаж работы» и «Стаж работы по специальности» исходный вид данных (количество лет) не меняется.

Зададим в виде множества $V = (v_1, v_2, \dots, v_d)$ параметры, характеризующие степень важности рассматриваемых критериев. Степень важности определяет, насколько важно учитывать каждый параметр при решении поставленной задачи.

Например, при поиске кандидата на повышение, руководитель желает учитывать в первую очередь его стаж работы по специальности. Тогда при расчете показателей этой графе присваивается коэффициент важности «1», остальным критериям – меньше. Важность коэффициента является константой, одинаковой для каждого сотрудника.

С учетом этих данных составлена функция (7), которая рассчитывает коэффициент наибольшего соответствия поставленной задачи для каждого отдельно взятого сотрудника с учетом всех критериев.

$$a = \sum_{i=0}^d s_i * v_i, \quad (7)$$

где a – коэффициент соответствия, s_i – значение критерия, v_i – вес критерия, d – количество критериев.

Так как для каждого критерия задается отдельное значение, диапазоны минимума и максимума которых разнятся между собой, то вычисление коэффициента соответствия лучше проводить с учетом нормализации. Тогда функция (7) видоизменяется в функцию (8):

$$a = \frac{d}{i=0} \frac{s_i - \min_{1 < i < d} s_i}{\max_{1 < i < d} s_i - \min_{1 < i < d} s_i} * v_i, \quad (8)$$

Формула находит коэффициент соответствия для одного сотрудника. Информация о нормировке данных взята на сайте «Подготовка и обработка, анализ и представление данных» [13].

Множество $E = (a_1, a_2, \dots, a_d)$ содержит все рассчитанные коэффициенты эффективности.

Приведенная формула поможет сортировать данные по выбранному коэффициенту. При дальнейшей реализации программного продукта в качестве основы будет использоваться созданная математическая модель.

Для оптимизации процесса назначения задач следует реализовать математическую модель, представив ее в виде программного продукта, который позволит автоматически подбирать кандидатов для поставленной цели, что избавит от случайных ошибок и воздействия человеческого фактора на принятие решения, а также позволит сократить временные затраты на принятие решения.

Требуется определить основные показатели, характеризующие сотрудника, на основании которых будут проведены дальнейшие расчеты поиска оптимальной кандидатуры. Итак, при поиске коэффициентов эффективности для решения задачи о назначении (поездка на конференцию) расчетная формула (9) будет выглядеть следующим образом:

$$A = 5*s_1 + 4*s_2 + 3*s_3 + 10*s_4 + 3*s_5 + 7*s_6, \quad (9)$$

где s_1, s_2, \dots, s_d – значения критериев для каждого отдельного сотрудника.

При реализации представленной математической модели могут быть получены результаты, представленные на рисунке 1.5.

ализация	Повышение	Поездка на конференцию	Отпуск	Выступление с докладом	Пользов-ский
Имя		{0.5, 0.4, 0.3, 1, 0.3, 0.7}	Должность		Специальность
Быков Сергей Владимирович			Профессор кафедры гуманитарных и		Инженерная психология
Дуюнов Владимир Кузьмич			Профессор кафедры уголовного права и		Правоведение
Ищенко Станислав Андреевич			Профессор кафедры государственного и		Правоведение
Гогина Галина Николаевна			Профессор, заведующий кафедрой		Планирование народного хозяйства
Калукова Ольга Макаровна			Доцент кафедры гуманитарных и		Математика
Кулишов Евгений Николаевич			Доцент кафедры экономика		Экономика труда
Ильин Николай Егорович			Старший преподаватель кафедры		Итальянский язык и литература
Бурда Светлана Юрьевна			Доцент кафедры естественнонаучных и		Психология
Закомолдина Тамара Олеговна			Доцент кафедры государственного и		Правоведение
Закомолдин Руслан Валериевич			Доцент кафедры уголовного права и		Юриспруденция
Некрасова Екатерина Владимировна			Доцент кафедры гуманитарных и		Психология
Баранов Андрей Андреевич			Доцент кафедры экономика		Промышленное и гражданское строитель
Воробьев Павел Викторович			Доцент кафедры экономика		Водоснабжение и канализация
Маштаков Игорь Владимирович			Доцент, заведующий кафедрой		Юриспруденция
Грибанов Дмитрий Александрович			Доцент кафедры теории и истории права и		Юриспруденция

Рисунок 1.5 – Пример вывода при реализации математической модели

Полученные данные будут определять вес преподавателя при назначении его на выполнение определенного круга задач.

Задачу о назначении можно решить при помощи венгерского метода, специально разработанного для решения задач данного типа.

1.3.3 Решение задачи о назначении с использованием венгерского метода

Метод основан на следующем свойстве матрицы эффективности: если ко всем элементам c_{ij} i -ой строки прибавить произвольную константу u_i , и ко всем элементам j -ого столбца прибавить произвольную константу v_j , то будет получена новая матрица эффективности с элементами $d_{ij} = a_{ij} + u_i + v_j (\forall i, j \in 1, 2, \dots, n)$. Если умножить обе части этого выражения на x_{ij} и учесть ограничения, заданные формулами (2) и (3), то будет получено выражение:

$$\sum_{i=1}^n \sum_{j=1}^n a_{ij} x_{ij} = \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} - \sum_{i=1}^n u_i - \sum_{j=1}^n v_j. \quad (10)$$

Из этого следует, что задача максимизации целевой функции (1) с заданными ограничениями (2) – (4) эквивалентна задаче минимизации некоторой измененной функции с такой же системой ограничений. При этом измененная целевая функция определяется следующим способом:

$$\sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \rightarrow \min_{x \in \Delta \beta}, \quad (11)$$

Так как венгерский метод рассчитан на поиск минимизации, то для задач, ориентированных на максимизацию, все элементы целевой матрицы умножаются на -1 , а затем складываются с достаточно большим числом, чтобы избежать отрицательных знаков. После этого исходная задача решается как задача минимизации.

Исходя из этого, идея решения задачи распределения венгерским методом заключается в последовательном вычитании из элементов каждой строки и каждого столбца исходной матрицы затрат их наименьшие элементы. После этого проводится анализ на достижение допустимого решения. Если получено решение, которому соответствуют нули в измененной матрице затрат, то оно является оптимальным решением задачи. Если решение недопустимо, то выполняется дальнейшее изменение матрицы затрат.

В общем случае венгерский алгоритм основан на итерациях и заключается в следующем пошаговом выполнении:

1. Редукция строк и столбцов. Для задачи о назначении, представленной в симметрической форме (1) – (4) в каждой строке исходной матрицы из всех элементов вычитывается наименьший, после чего в каждом столбце из всех элементов также вычитывается наименьший. Итогом будет некоторая редуцированная матрица затрат задачи о назначении с элементами d_{ij} , нулевые значения которых будут соответствовать возможным назначениям сотрудников на сформулированные задачи.

2. Анализ допустимости найденного решения. Заключается в проверки выполнения ограничений (2) – (4) для соответствующих значений x_{ij} . Для анализа выполняется отдельная процедура вычеркивания строк и столбцов. Если после выполнения этой процедуры итогом является полное допустимое значение, то оно рассматривается как оптимальное решение задачи. Алгоритм завершается. Если полученное значение является неполным, выполняется шаг 3.

3. Модификация редуцированной матрицы затрат. Если полученное решение является неполным, то следует упростить рассматриваемую матрицу

для получения новых нулевых элементов. Для этого проводится процедура редукции элементов. Для проверки допустимости нового решения, выполняется шаг 2.

Далее описывается процедура вычеркивания строк и столбцов, используемая в шаге 2:

1. Поиск в матрице строки, содержащий один невычеркнутый нулевой элемент. Для этой строки производится назначение в соответствии с невычеркнутым элементом, после чего вся строка и столбец, в котором находится этот элемент, вычеркиваются.

2. Поиск в матрице столбца, содержащего один невычеркнутый нулевой элемент. Для этого столбца производится назначение в соответствии с невычеркнутым элементом, после чего весь столбец и строка, в которой находится этот элемент, вычеркиваются.

3. Шаги 1 и 2 проводятся до тех пор, пока в матрице не останется строк или столбцов, которые содержат невычеркнутые нулевые элементы. Если все строки и столбцы при этом вычеркнуты, то найденное решение оптимально. Иначе решение является недопустимым.

Далее описывается процедура редукции ненулевых элементов:

1. Посчитать количество нулей в невычеркнутых строках и столбцах матрицы.

2. Вычеркнуть столбец или строку с наибольшим числом нулей. Если их несколько, вычеркивается любая строка или столбец.

3. Выполнять шаги 1 и 2 до тех пор, пока не будут вычеркнуты все нулевые элементы.

4. Из всех невычеркнутых ненулевых элементов вычесть минимальный и прибавить его к каждому из элементов, расположенных на пересечении каждой пары линейки вычеркивания.

Венгерский алгоритм, как и используемые в нем процедуры, имеет конечный характер. Он может быть представлен графически в виде диаграммы деятельности (рисунок 1.6).

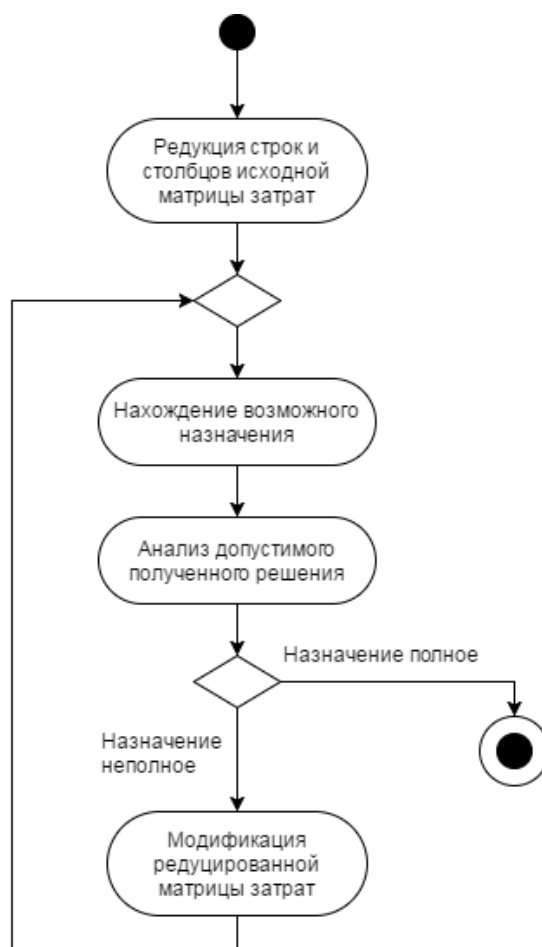


Рисунок 1.6 – Диаграмма деятельности (алгоритм венгерского метода)

Задачу оптимизации распределения человеческих ресурсов на выполнение задач можно решить несколькими способами: при помощи программы MS Excel, в которую заложен алгоритм нелинейной оптимизации Generalized Reduced Gradient (GRG2), разработанный Леоном Ласдоном и Аланом Уореном, или переложить в программную разработку, которая будет основана на использовании алгоритма решения задачи о назначении.

Определим сложность решения задачи о назначении, применив описанный алгоритм в среде табличного редактора.

1.3.4 Реализация задачи о назначении с использованием табличного редактора

Рассмотрим решение задачи о назначении преподавателей в программе Excel на примере максимизации общей эффективности при выполнении работ.

Показатели эффективности сотрудников, которые будут определяющими при выборе преподавателя для решения обозначенных задач z_{ij} ($\forall i, j \in 1,2,3,4$), заданы таблицей 1.4.

Таблица 1.4 – Оптимизационные показатели преподавателей для выполнения ими конкретных задач

Задача	Ищенко	Гогина	Дуюнов	Быков
<i>Экскурсия со студентами</i>	2,35	2,06	1,92	1,78
<i>Поездка на конференцию</i>	1,61	1,47	2,27	2,39
<i>Составление вопросов на экзамен</i>	1,21	1,26	1,67	2,06
<i>Выступление с докладом</i>	1,98	1,2	2,03	2,27

Тогда задача о назначении для данного примера может быть записана следующим образом:

$$\begin{aligned}
 & 2,35x_{11} + 2,06x_{12} + 1,92x_{13} + 1,78x_{14} + \\
 & + 1,61x_{21} + 1,47x_{22} + 2,27x_{23} + 2,39x_{24} + \\
 & + 1,21x_{31} + 1,26x_{32} + 1,67x_{33} + 2,06x_{34} + \\
 & + 1,98x_{41} + 1,2x_{42} + 2,03x_{43} + 2,27x_{44} \rightarrow \max_{x \in \Delta\beta},
 \end{aligned} \tag{12}$$

где множество допустимых альтернатив $\Delta\beta$ задано следующей системой:

$$\begin{aligned}
 & x_{11} + x_{12} + x_{13} + x_{14} = 1; \\
 & x_{21} + x_{22} + x_{23} + x_{24} = 1; \\
 & x_{31} + x_{32} + x_{33} + x_{34} = 1; \\
 & x_{41} + x_{42} + x_{43} + x_{44} = 1; \\
 & x_{11} + x_{21} + x_{31} + x_{41} = 1; \\
 & x_{11} + x_{22} + x_{32} + x_{42} = 1; \\
 & x_{11} + x_{23} + x_{33} + x_{43} = 1; \\
 & x_{11} + x_{24} + x_{34} + x_{44} = 1; \\
 & x_{ij} \in 0,1 \quad \forall i, j \in 1,2,3,4.
 \end{aligned} \tag{13}$$

Причем первые четыре ограничения соответствуют общему ограничению (2), следующие четыре ограничения соответствуют ограничению (3), а последнее – ограничению (4).

Решим сформулированную задачу о назначении при помощи программы Excel, задав все начальные данные и ограничения в виде формул.

Также введем параметры и базовые ограничения для задачи о назначении в мастер поиска решения (рисунок 1.7).

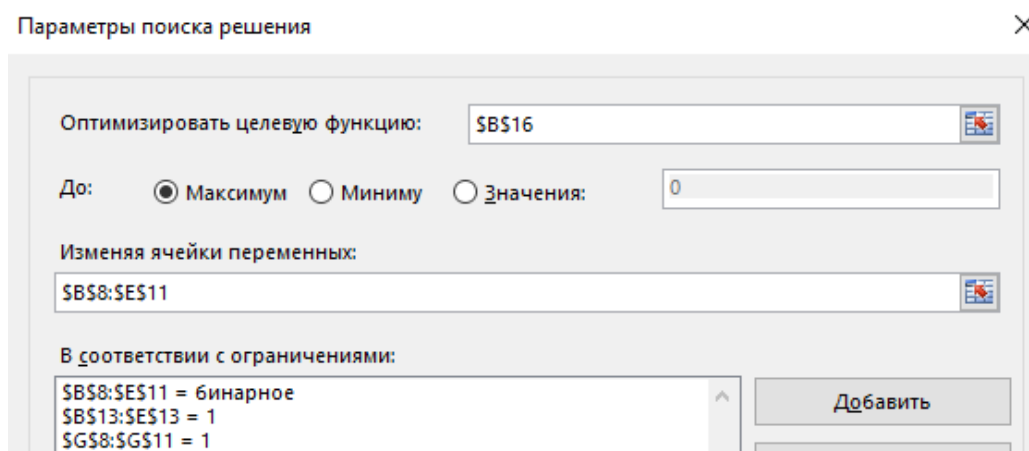


Рисунок 1.7 – Параметры и ограничения задачи

После выполнения расчетов программой Excel будет найдено количественное решение, показанное на рисунке 1.8.

Переменные:	Ищенко	Гогина	Дуюнов	Быков
Экскурсия со студентами	0	1	0	0
Поездка на конференцию	0	0	1	0
Составление вопросов на экзамен	0	0	0	1
Выступление с докладом	1	0	0	0

Рисунок 1.8 – Результат решения задачи о назначении

Результатом вычислений являются следующие оптимальные значения переменных: $x_{12} = 1$, $x_{23} = 1$, $x_{34} = 1$, $x_{41} = 1$, остальные переменные равняются 0. Также найдено максимально значение целевой функции $f = 8,37$.

Анализ решения задачи о назначении показывает, что на экскурсию со студентами рекомендуется назначить Гогину, для поездки на конференцию – Дуюнова, составление вопросов на экзамен – Быкова, выступления с докладом – Ищенко.

В образовательном учреждении «СаГА» ежедневно решаются задачи о назначении с большим числом сотрудников. Создание новой математической модели для решения каждой задачи в табличном редакторе не продуктивно, т.к.

уходит много времени на подсчет новых коэффициентов и распределение данных в таблице. Выходом может служить программная система для автоматизации этого процесса.

Программа при этом должна иметь возможность подключения к базе данных, в которой хранится информация о сотрудниках предприятия (такая как ФИО, квалификация сотрудника, предметы, которые он ведет, стаж работы сотрудника и сколько раз он проходил курсы повышения квалификации), обработки данных, осуществления расчетов коэффициентов по формулам, сортировки данных и вывода их на экран в доступном пользователю виде.

Итак, информационная система должна обладать такими функциями как:

- автоматизация процесса распределения персонала;
- экономия времени руководителя организации;
- исключение ошибок, возникающих из-за человеческого фактора, из процесса распределения персонала.

Для успешного взаимодействия пользователя с программой следует разработать понятный интерфейс, содержащий поля настройки программы и кнопки ее запуска. Результат работы программы будет выводиться на экран.

1.4 Обзор программных аналогов

Так как организация СаГА насчитывает большое количество сотрудников, то существует проблема высоких временных затрат на обработку данных сотрудников и отсутствия оптимизации процесса распределения персонала.

В настоящий момент требуется решить такие задачи как:

- сокращение временных затрат на процесс распределения персонала;
- оптимизация работы с сотрудниками;
- устранение необходимости анализа и сравнения большого количества личной информации о сотрудниках.

Для решения представленных задач следует воспользоваться автоматизированными системами или программой. На данный момент

существуют специализированные программы для регулировки работы кадров на предприятии.

Одним из таких приложений является программа «Мини-Кадры» – простое бесплатное приложение по учету персонала [12]. Позволяет хранить данные о сотрудниках предприятия, формировать различные приказы кадровой службы, учитывать время работы сотрудников и т.д.

Программа позволяет выполнять функции, описанные в таблице 1.5.

Таблица 1.5 – Функции программы «Мини-Кадры»

Функция	Пояснения
Полный учет информации о работниках	<ul style="list-style-type: none"> • основные данные (ФИО, паспорт, ИНН, адрес, телефон и т.п.); • изображение; • образование (полный список всех законченных учебных заведений); • владение иностранными языками; • полный список профессий; • данные о воинском учете; • полный состав семьи.
Уникальный режим «Календарь кадровика»	<ul style="list-style-type: none"> • отображение кадрового состава в виде календаря; • отображение клеток разными цветами, в зависимости от «состояния» работника в конкретный день: работал, отсутствовал, болел, был в отпуске и т.д.; • отображение и ввод отработанного времени; • отображение дня рождения сотрудника с указанием возраста.
Учет приказов	<ul style="list-style-type: none"> • о трудоустройстве; • об увольнении; • об уходе сотрудника в отпуск; • о командировке; • фиксирование всех основных атрибутов приказов (даты, номера и т.п.); • возможность настройки печатной формы приказа; • печать приказов.

Данная программа предназначена для небольших организаций и может обрабатывать данные не более, чем у семи сотрудников. Фирмы с большим числом работников не смогут пользоваться данным приложением.

Интерфейс программы «Мини-Кадры» представлен на рисунке 1.9.

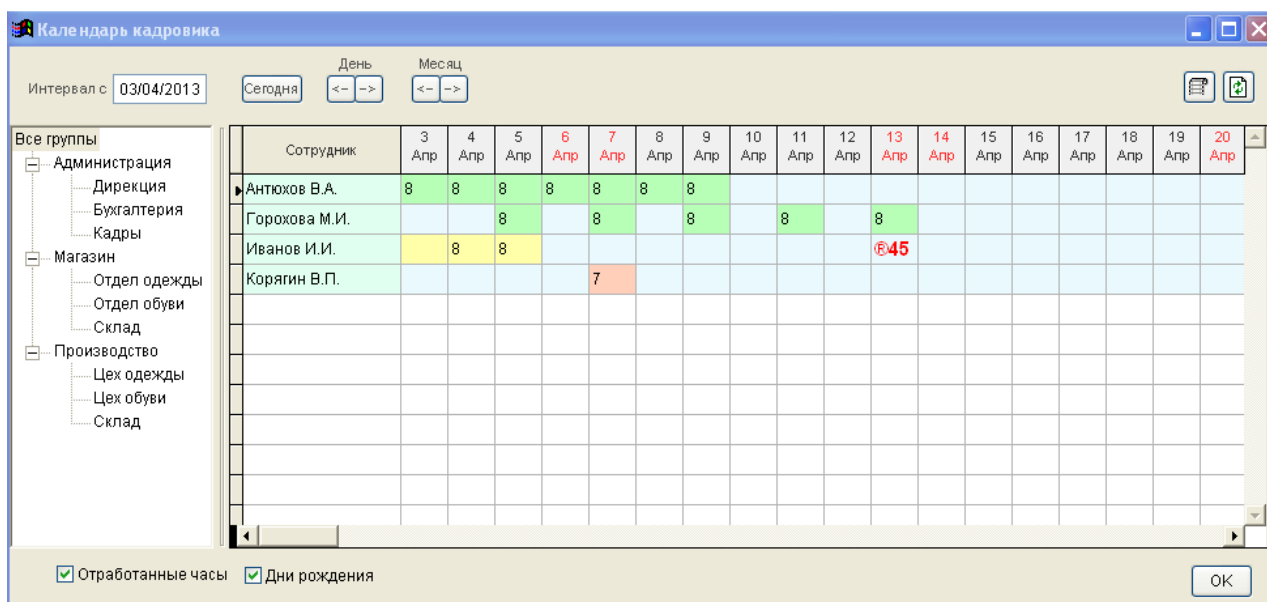


Рисунок 1.9 – Интерфейс программы «Мини-Кадры»

Другим программным аналогом является «LeaderTask» – программа для управления задачами и поручениями. Интерфейс программы показан на рисунке 1.10.

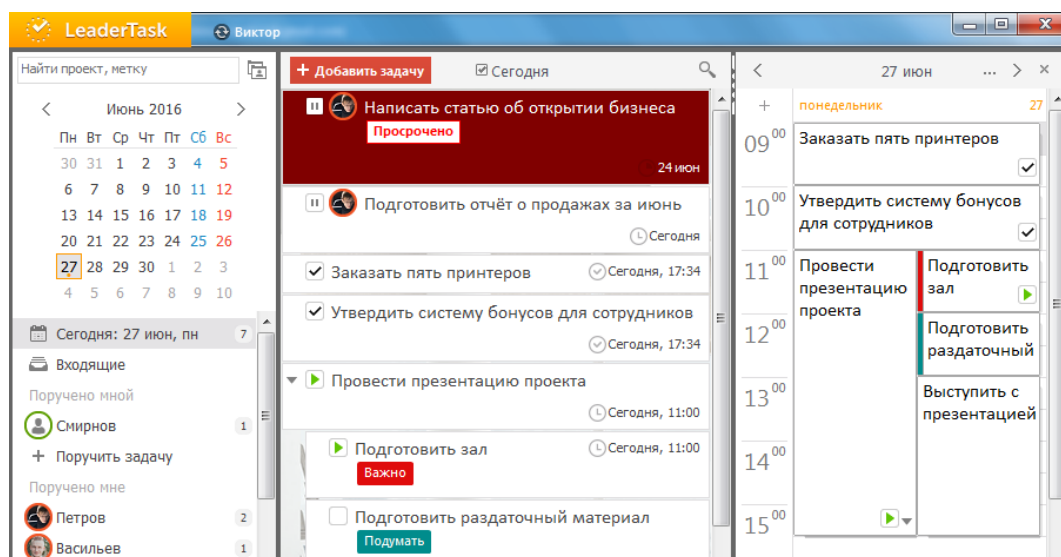


Рисунок 1.10 – Интерфейс программы «LeaderTask»

Данный сервис позволяет составлять списки дел на различные промежутки времени, раздавать поручения и контролировать исполнение, а также управлять проектами [15].

Программа «LeaderTask» позволяет решать следующие задачи:

- раздавать и контролировать поручения;
- отправлять задачи на доработку;
- управлять проектами;
- устанавливать напоминания;
- создавать планы работ.

Данная программа не предусматривает автоматизацию распределения задач и не распространяется бесплатно (стоимость от 2400 рублей на одного пользователя). Данные обстоятельства являются недостатком рассматриваемого приложения.

Похожим программным продуктом, облегчающим процесс распределения персонала, является программа «Teambox».

Данный программный продукт обладает доступной ценой и предоставляет такие функции как:

- отображение текущих задач и статуса их выполнения;
- назначение сотрудника на выполнение определенной задачи;
- присвоение задачам определенного статуса;
- возможность устанавливать напоминания.

Экранная форма интерфейса «Teambox» представлен на рисунке 1.11.

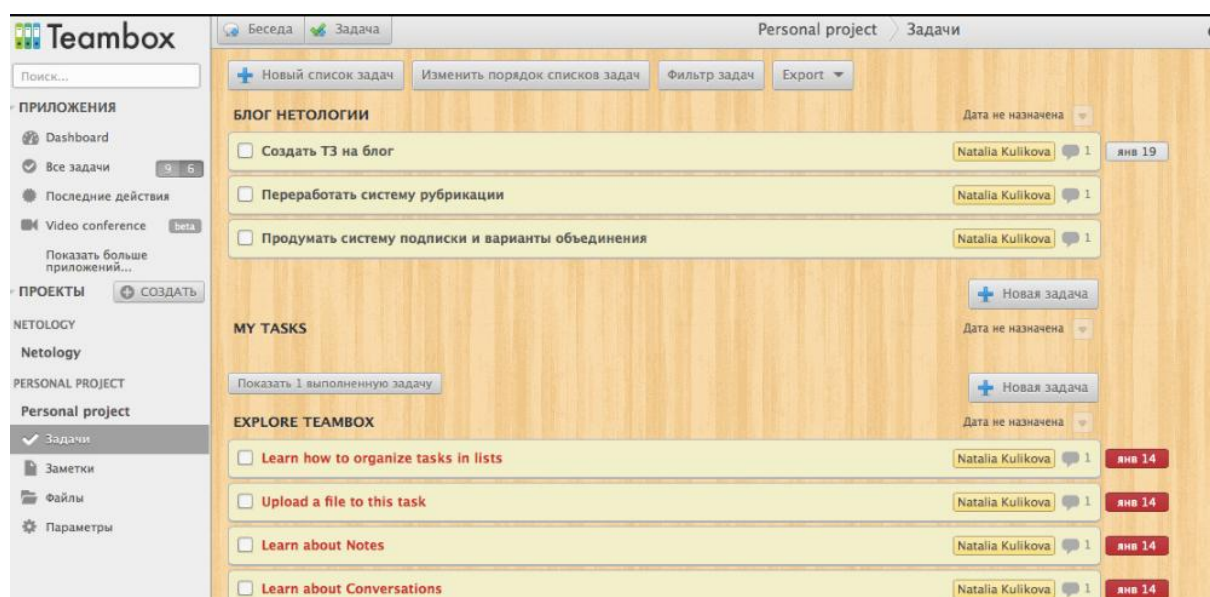


Рисунок 1.11 – Интерфейс программы «Teambox»

Программа «Teambox» обладает минимальным функционалом – ответственные, крайний срок выполнения, вложения. Существует возможность добавить к каждому проекту обсуждения и заметки. Достоинством данной программы является то, что программа имеет возможность запуска через мобильные приложения и ее невысокая цена – полная версия стоит меньше 300 рублей в месяц.

Недостатком данного продукта является отсутствие автоматизации и ориентация на компании с невысоким числом сотрудников.

В таблице 1.6 приведен результат сравнения программных аналогов.

Таблица 1.6 – Сравнение программных аналогов

Программа	Программа «Мини-Кадры»	Программа «LeaderTask»	Программа «Teambox»
Разработчик	ООО «Корс-Софт»	ООО «Органайзер ЛидерТаск»	Redbooth
Сложность использования	Средняя	Средняя	Низкая
Стоимость	Бесплатно	От 2 400 рублей	В зависимости от числа сотрудников – до 285 рублей в месяц
Автоматизация	Нет	Нет	Нет
Масштаб	Работа с организациями, состоящими не более чем из 7 сотрудников	Работа с организациями любого масштаба	Работа с небольшими организациями, с умеренным числом сотрудников

При изучении существующих программных продуктов была выделена идея создания обобщенной модели для организаций различного размера и с варьирующимся количеством сотрудников, а также реализации сортировки сотрудников по различным критериям для поиска наиболее квалифицированного работника, способного решить ту или иную задачу.

Представленные аналоги имеют свои достоинства, но предполагают, что руководитель принимает решения по распределению кадров самостоятельно, с учетом особенностей и работоспособности каждого отдельного сотрудника, что является затруднительным в крупных организациях. Данный подход не решает

проблемы высоких временных затрат и избавления от случайных ошибок, связанных с человеческим фактором.

Следует разработать собственный программный продукт, адаптированный под нужды организации и решающий поставленные задачи. Программа должна реализовывать автоматизацию распределения сотрудников, быть проста для понимания и позволять руководителю сразу получать результат обработки данных.

Вывод по первой главе

После анализа деятельности предприятия было выявлено, что в настоящее время управление человеческими ресурсами в Самарской гуманитарной академии не автоматизировано, все решения принимаются руководителем организации, в обязанности которого входит множество других задач. Такой способ управления человеческими ресурсами не гарантирует высокой скорости работы, а также защиты от случайных ошибок.

Автоматизация процесса управления человеческими ресурсами организации позволит значительно сократить время обработки данных и избавит от ошибок, связанных с человеческим фактором, что обеспечит облегчение деятельности по управлению предприятием.

На основании этой информации было принято решение о необходимости создания математической модели управления человеческими ресурсами организации и дальнейшей ее реализации в виде программного продукта. Для достижения поставленной цели требуется выбрать средства и технологию программирования, после чего реализовать математическую модель.

Глава 2 МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ ПЕРСОНАЛА НА ВЫПОЛНЕНИЕ ЗАДАЧ

2.1 Описание архитектуры информационной системы

При проектировании информационной системы, следует выбрать ее архитектуру. Приложение для реализации математической модели будет локальным, т.к. создается для использования только руководителем организации. Все данные, необходимые для запуска приложения будут храниться на одном компьютере.

В связи с этим при разработке программы будет использоваться архитектура типа «файл-сервер» (рисунок 2.1).

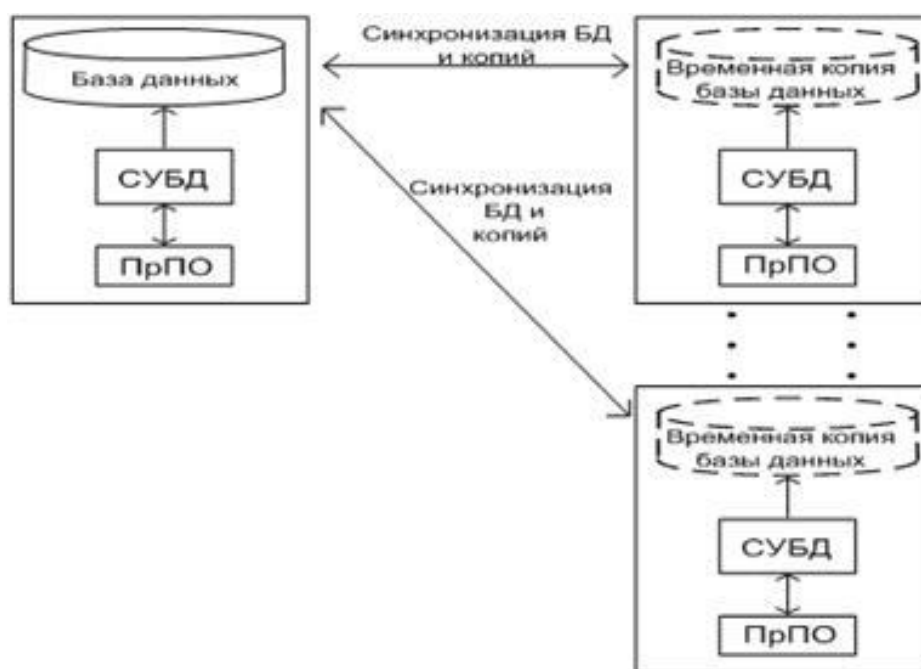


Рисунок 2.1 – Архитектура «файл-сервер»

Данный тип архитектуры является наиболее простым в реализации, в то время как основные его недостатки не актуальны при использовании программы только компьютером-сервером (т.к. отсутствует необходимость передачи данных на ПК-клиенты).

Также перед проектированием программного приложения следует рассмотреть некоторые языки программирования.

2.2 Обзор и выбор средств программирования

В настоящее время существует множество потребности реализации какой-либо системы посредством программирования, в связи с чем создано множество языков, которые можно для этого использовать. Наиболее распространенными среди языков программирования являются языки Java, C++ и C#.

Язык программирования Java является строго типизированным объектно-ориентированным языком. Данный язык был разработан компанией Sun Microsystems, которую в последствии приобрела компания Oracle.

Java-приложения обычно транслируются в специальный байт-код с помощью виртуальной Java-машины (JVM) – программы, которая обрабатывает байтовый код и передает инструкции оборудованию в качестве интерпретатора.

Подобный способ выполнения программ обладает таким достоинством как полная независимость байт-кода от операционной системы или оборудования. Данное обстоятельство обеспечивает выполнение Java-приложений на любом устройстве, для которого существует виртуальная машина.

Технологии Java также обладают другой важной особенностью, которой является гибкая система безопасности. В рамках этой системы выполнение программы целиком контролируется виртуальной машиной. Если операции превышают установленные полномочия программы (например, происходит попытка несанкционированного соединения с другим компьютером или доступа к защищенным данным), то вызывается немедленное прерывание.

К сожалению, концепция виртуальной машины содержит в себе такой недостаток как снижение производительности. Однако ряд разработанных улучшений несколько ускорил процесс выполнения программ на Java:

- применение JIT-технологии, подразумевающей трансляцию байт-кода в машинный код непосредственно в процессе работы программы, с учетом возможности сохранения версий класса в машинном коде,
- свободное использование платформенно-ориентированного кода, называемого native-кодом, в стандартных библиотеках,
- аппаратные средства, которые способны обеспечить более быструю обработку байт-кода [2].

Язык программирования C++ является компилируемым строго типизированным языком общего назначения. Он поддерживает такие парадигмы программирования как процедурная, обобщенная, функциональная; при этом наибольшее внимание уделено объектно-ориентированному программированию.

Язык C++ обладает следующими достоинствами:

1. Масштабируемость. Данный язык программирования позволяет разрабатывать программы для широкого спектра платформ и систем.
2. Возможность работать с памятью, адресами, портами на низком уровне. Однако при неосторожном использовании, это достоинство становится недостатком.
3. Возможность создавать обобщенные алгоритмы для различных типов данных, их специализация, и вычисление на этапе компиляции, с использованием шаблонов.

Однако, при всем этом, язык программирования C++ имеет ряд недостатков:

1. Трудности в поиске ошибок. C++ унаследовало большинство проблем типобезопасности от языка C, к тому же, создатель языка отказался от идеи использования автоматического управления памятью (такого как сборка мусора). Из-за этих факторов язык C++ обладает уязвимостями типа "переполнение буфера".

2. Плохая поддержка модульности. Подключение внешнего интерфейса через заголовочный файл (`#include`) серьезно замедляет компиляцию, если использовать большое количество модулей.

3. Нехватка информации о типах данных в процессе компиляции.

4. Язык C++ сложен для изучения и компиляции.

5. Некоторые преобразования типов требуют внимательности разработчика. К примеру, операция над беззнаковыми и знаковыми числами выдает беззнаковый результат.

6. Примитивность препроцессора C++ (унаследованного от C) приводит к сложностям при осуществлении некоторых задач метапрограммирования, к тому же он часто приводит к ошибкам и требует много действий для того, чтобы обойти потенциальные проблемы.

7. На практике в языке C++ отсутствует поддержка функционального программирования. Данный пробел можно частично устранить различными библиотеками метапрограммирования, но качество таких решений будет значительно уступать качеству решений, встроенных в функциональные языки. Более подробную информацию можно найти в справочном руководстве по языку C++ М. Эллиса и Б. Строуструпа [6].

Другим популярным языком программирования является C# – язык программирования, который сочетает концепции объектно-ориентированного и аспектно-ориентированного программирования. Данный язык был разработан в 1998–2001 годах в компании Microsoft в качестве основного языка для разработки приложений платформы Microsoft.NET. В стандартную установку самой .NET входит компилятор с C#, поэтому программы на этом языке можно создавать и без таких инструментальных средств как Visual Studio.

Синтаксис языка C# наиболее близок к C++ и Java. C# имеет строгую статическую типизацию, осуществляет поддержку полиморфизмов, перегрузки операторов, указателей на функции-члены классов, атрибутов, событий, свойств, исключений и комментариев в XML-формате. C# исключает некоторые модели, используемые другими языками и зарекомендовавшие

себя как проблематичные, в ходе разработки программных систем: например, язык C#, в отличие от C++, не поддерживает множественное наследование классов.

Язык программирования C# имеет следующие основные достоинства:

1. C# является подлинным объектно-ориентированным языком программирования;
2. Имеется компонентно-ориентированный подход к написанию программ, который способствует меньшей зависимости результирующего программного кода от архитектуры и аппаратных средств, а также гибкости, переносимости и простоте повторного использования (фрагментов) программ;
3. Язык C# ориентирован на безопасность кода (в сравнении с языками C и C++);
4. Унифицированная система типизации;
5. Широкая поддержка событийно-ориентированного написания программ.

Несмотря на перечисленные достоинства, язык C# имеет также некоторые недостатки, такие как:

1. Довольно сложный синтаксис;
2. Малое количество свежих концептуальных идей;
3. Относительно невысокая производительность (C# сравним с Java, но гораздо более медленный, чем язык C);

1. Язык C# не является кроссплатформенным языком. Так как разработчиком языка является компания Microsoft, и работает он только под конкретной операционной системой Windows, хотя в данный момент идет разработка кроссплатформенная версия данного языка программирования. Информация взята из книги «Разработка приложений для Windows 8 на языке C#» Пугачева С., Шериева А. и Кичинского К. [8].

После рассмотрения предложенных примеров, было принято решение использовать Java в качестве основного языка программирования, так как

данный язык не зависит от архитектуры и аппаратных средств, а также обладает гибкой системой безопасности.

В качестве среды программирования будет использована свободная интегрированная среда NetBeans IDE, которая позволяет разрабатывать приложения на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада и ряде других.

Для работы в среде NetBeans и для успешной инсталляции и функционирования самой среды NetBeans следует предварительно установить Sun JDK или J2EE SDK соответствующей версии. Данная среда разработки поддерживает разработку на платформах J2SE и J2EE. Начиная с версии 6.0 NetBeans включает в себя поддержку разработки для мобильных платформ J2ME, C++ и PHP без необходимости устанавливать дополнительные компоненты.

Для реализации математической модели в качестве основного языка программирования был выбран язык Java, в качестве среды программирования – среда NetBeans.

2.3 Разработка концептуальной модели процесса распределения персонала

Для разработки концептуальной модели процесса распределения персонала нужно построить диаграмму вариантов использования в рамках исследуемой области.

Данные диаграммы описывают функциональное назначение исходной системы или функции, которые система должна исполнять. Разработка диаграммы вариантов использования преследует следующие цели:

- определение общих границ и контекста моделируемой предметной области;
- формулировка общих требований к функциональному поведению проектируемой системы;

- разработка исходной концептуальной модели системы для ее последующей детализации в форме логических и физических моделей;
- подготовка исходной документации для взаимодействия разработчика системы с заказчиком и пользователями [5].

Для создания концептуальной модели будет использоваться язык графического описания UML (англ. Unified Modeling Language – унифицированный язык моделирования), который используется для объектного моделирования в области разработки программного обеспечения, системного проектирования, моделирования бизнес-процессов и отображения организационных структур.

Для создания концептуальной модели используется онлайн-сервис Gliffy, созданный для построения диаграмм графиков, а также для различных схем, алгоритмов, планов.

Для разработки концептуальной модели необходимо определить, кто взаимодействует с системой, и каким образом.

В рамках управления персоналом директор организации выполняет такие функции как:

- определение соответствия уровня компетенции кандидатов выдвинутым требованиям – необходим сбор и анализ данных о сотруднике для принятия решения о степени его успешности по качеству выполненной работы;
- принятие решений о профессиональной карьере сотрудника (назначение сотрудника на выполнение определенной задачи);
- начисление зарплаты сотрудникам.

Из этого следует, что концептуальная модель будет содержать двух актеров – руководителя организации и сотрудника.

Сотрудник предоставляет личную информацию о себе для дальнейшей обработки. На основе предоставленной информации и ее анализа руководитель организации определяет, насколько сотрудник соответствует заданным требованиям, принимает решения о его профессиональной деятельности

(например, направлении сотрудника на конференцию) и начисляет заработную плату.

На основании представленных данных была разработана диаграмма вариантов использования, необходимая для дальнейшего проектирования системы (рис. 2.2).



Рисунок 2.2 – Диаграмма вариантов использования

В данном представлении диаграмма вариантов использования рассмотрена с позиции выполнения человеком определенных функций.

Созданная диаграмма вариантов использования будет учитываться при разработке программного продукта, основанного на созданной математической модели.

2.3 Разработка интерфейса программного продукта

Для реализации программного продукта, направленного на автоматизацию процесса распределения персонала, следует выбрать наиболее удобный для пользователя вариант интерфейса.

В ходе моделирования приложения были рассмотрены несколько возможных вариантов интерфейса программы.

В первом варианте меню пользователя располагается в верхней части окна, в нижней части выводится результат работы приложения (рисунок 2.3).



Рисунок 2.3 – Интерфейс основного меню, вариант 1

На рисунке кнопки от 1 до n обозначают функции поиска, которые может реализовать программа, например, поиск наилучшего кандидата на «Повышение», «Отпуск», «Выступление с докладом» и т.д. Для каждой функции важность критериев задается автоматически, поэтому вывод результатов происходит сразу, в поле вывода таблицы.

Кнопка «Пользовательский» создана для самостоятельного ввода важности критериев пользователем, поэтому для нее существует отдельное меню ввода (рисунок 2.4).



Рисунок 2.4 – Интерфейс пользовательского меню, вариант 1

В меню ввода отображаются полосы с числовыми значениями, где пользователь может задать для каждого критерия уровень важности от 0 до 10. Вывод таблицы происходит после нажатия кнопки «Применить».

В представленном варианте вся информация отображается в пределах одного окна, что может быть трудным для восприятия пользователем. Следует рассмотреть возможность «пошагового» отображения результатов работы программы, когда каждое действие приводит к запуску отдельного окна.

Во втором варианте при запуске программы пользователь видит основное меню, на котором отображаются основные кнопки и кнопка «Пользовательский» (рисунок 2.5).

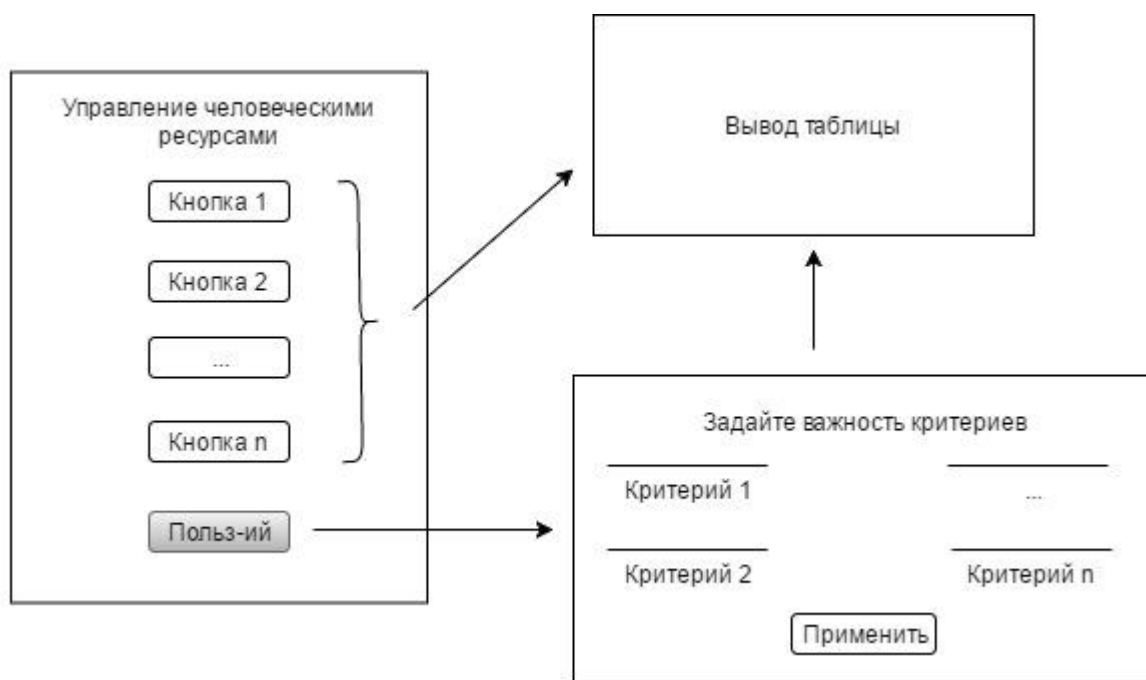


Рисунок 2.5 – Интерфейс программы, вариант 2

При нажатии одной из основных кнопок в отдельном окне происходит вывод таблицы результатов. При выборе пользовательских настроек открывается отдельное меню, где задается важность критериев. После нажатия кнопки «Применить», вывод таблицы результатов также отображается в отдельном окне.

После рассмотрения возможных вариантов интерфейса было принято решение использовать второй вариант, т.к. пошаговое отображение рабочих окон будет более удобным для восприятия пользователем.

2.4 Описание проектируемой базы данных

Для успешной реализации математической модели следует спроектировать базу данных, с которой будет взаимодействовать программа. Для этого создадим концептуальную диаграмму базы данных (рисунок 2.6).

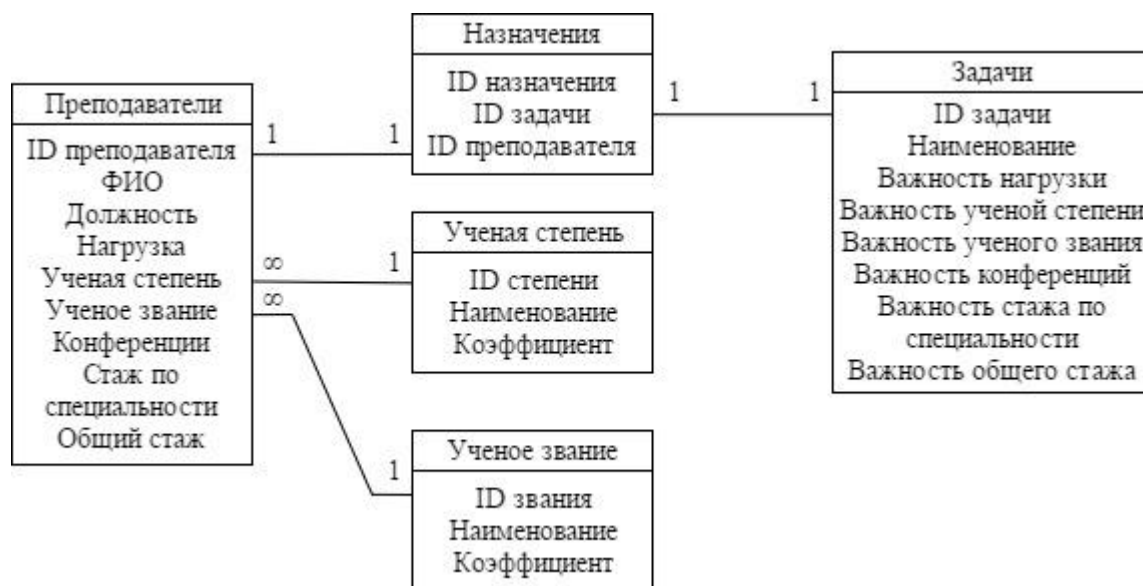


Рисунок 2.6 – Концептуальная модель данных

База данных содержит информацию о сотрудниках, которая используется в программе при поиске кандидатов для выполнения какой-либо задачи.

База данных содержит пять таблиц: «teachers», «appointments», «tasks», «academicdegreecof» и «academictitlecof».

Таблица «teachers» включает в себя поля со следующей информацией: id сотрудника, ФИО сотрудника, его должность, количество преподаваемых дисциплин, id ученой степени, id ученого звания, направление специальности, количество поездок на конференции, информация о повышении квалификации, общий стаж работы, стаж работы по специальности.

Внешний вид таблицы представлен на рисунке 2.7.

id	name	position	discipline	academic_degree	academic_title	training_direction	conferences	qualification	servic...	sped...
10	Закомолдин Алексей Валери...	Доцент кафедры у...	7	1	1	Юриспруденция	0	2013 г. Российск...	12	12
11	Закомолдин Руслан Валериевич	Доцент кафедры у...	6	1	3	Юриспруденция	0	2010 г. Государс...	19	17
12	Закомолдина Тамара Олеговна	Доцент кафедры г...	2	1	1	Правоведение	0	2013 г. РГУС пр...	46	21
13	Ильин Николай Егорович	Старший преподав...	1	3	5	Итальянский яз...	0		44	44
14	Ищенко Станислав Андреевич	Профессор кафедр...	3	2	4	Правоведение	0	2013 г. Российск...	39	38
15	Калукова Ольга Макаровна	Доцент кафедры г...	10	1	1	Математика	0	2013 г. ФГБОУ В...	41	35
16	Капелюшный Эдуард Дмитр...	Доцент кафедры э...	4	1	5	Строительство ...	0	2013 г. РГУС, пр...	28	11
17	Кулишов Евгений Николаевич	Доцент кафедры э...	8	1	1	Экономика труда	0	2015 г. ООО АФ ...	38	37
18	Кац Татьяна Владимировна	Старший преподав...	7	3	5	Филология	0	2016 г. ЧОУ ВО ...	13	10
19	Лапшина Ольга Сергеевна	Доцент, заведую...	8	1	5	Юриспруденция	0	2013 г. Российск...	10	10
20	Маштаков Игорь Владимирович	Доцент, заведую...	5	1	1	Юриспруденция	0	2013 г. НОУ ВП...	20	18
21	Некрасова Екатерина Влади...	Доцент кафедры г...	16	1	5	Психология	0	2013 г. НОУ ВП...	13	13
22	Пантыкина Марина Ивановна	Профессор кафедр...	2	2	1	Философия	0	2014 г. ФГБОУ В...	30	28
23	Пересыпкин Владимир Алекс...	Доцент кафедры г...	6	1	5	Учебно-профил...	0	*2013 г. НОУ ВП...	44	11
24	Пимахин Михаил Александро...	Преподаватель ка...	3	3	5	Юриспруденция	0	Является члено...	7	2
25	Плотникова Татьяна Юрьевна	Доцент кафедры г...	7	1	5	Психология	0	2013 г. НОУ ВП...	10	8
26	Румянцева Елена Леонидовна	Доцент, заведую...	7	1	1	Автоматизация ...	0	*2013 г. НОУ ВП...	22	21
27	Сахчинская Наталья Серге...	Профессор кафедр...	7	2	1	Бухгалтерский у...	0	2011г. Присужде...	13	13
28	Селюжицкая Наталья Никол...	Старший преподав...	11	3	5	Планирование п...	0	*2010г. ОУ НУ Д...	28	28
29	Семенов Денис Евгеньевич	Доцент кафедры э...	11	1	1	Бухгалтерский у...	0	*2010 г. Стажир...	10	10
30	Соколова Анастасия Алексан...	Доцент кафедры э...	5	1	1	Бухгалтерский у...	0	*2009 г. Присуж...	11	11
31	Стародубова Людмила Вале...	Старший преподав...	8	3	5	Юриспруденция	0	*2016 г. ЧОУ ВО...	22	15
32	Сюлина Светлана Павловна	Доцент кафедры э...	9	1	1	Бухгалтерский у...	0	*2010 г. Присво...	22	21
33	Торопова Наталья Юрьевна	Старший преподав...	8	3	5	Юриспруденция	0	*2013 г. НОУ ВП...	14	8

Рисунок 2.7 – Таблица «teachers»

Таблица «tasks» содержит информацию о задачах, которые требуется решить, такую как наименование задачи и коэффициенты важности, которые используются при распределении.

Таблица «appointments» хранит итоговые назначения преподавателей на задания.

Таблица «academicdegreesof» служит для распознавания обозначений ученой степени сотрудника и содержит поле «id ученой степени», по которому производится связь с таблицей «teachers», название ученой степени и коэффициент, который присваивается ей при включении в формулу (рисунок 2.8).

id	name	coefficient
1	К.Н.	1
2	Д.Н.	2
3	(NULL)	0

Рисунок 2.8 – Таблица с информацией об ученой степени сотрудника

Таблица «academictitlesof» построена аналогичным образом и служит для распознавания обозначений ученого звания сотрудника.

Физическая модель базы данных представлена на рисунке 2.9.

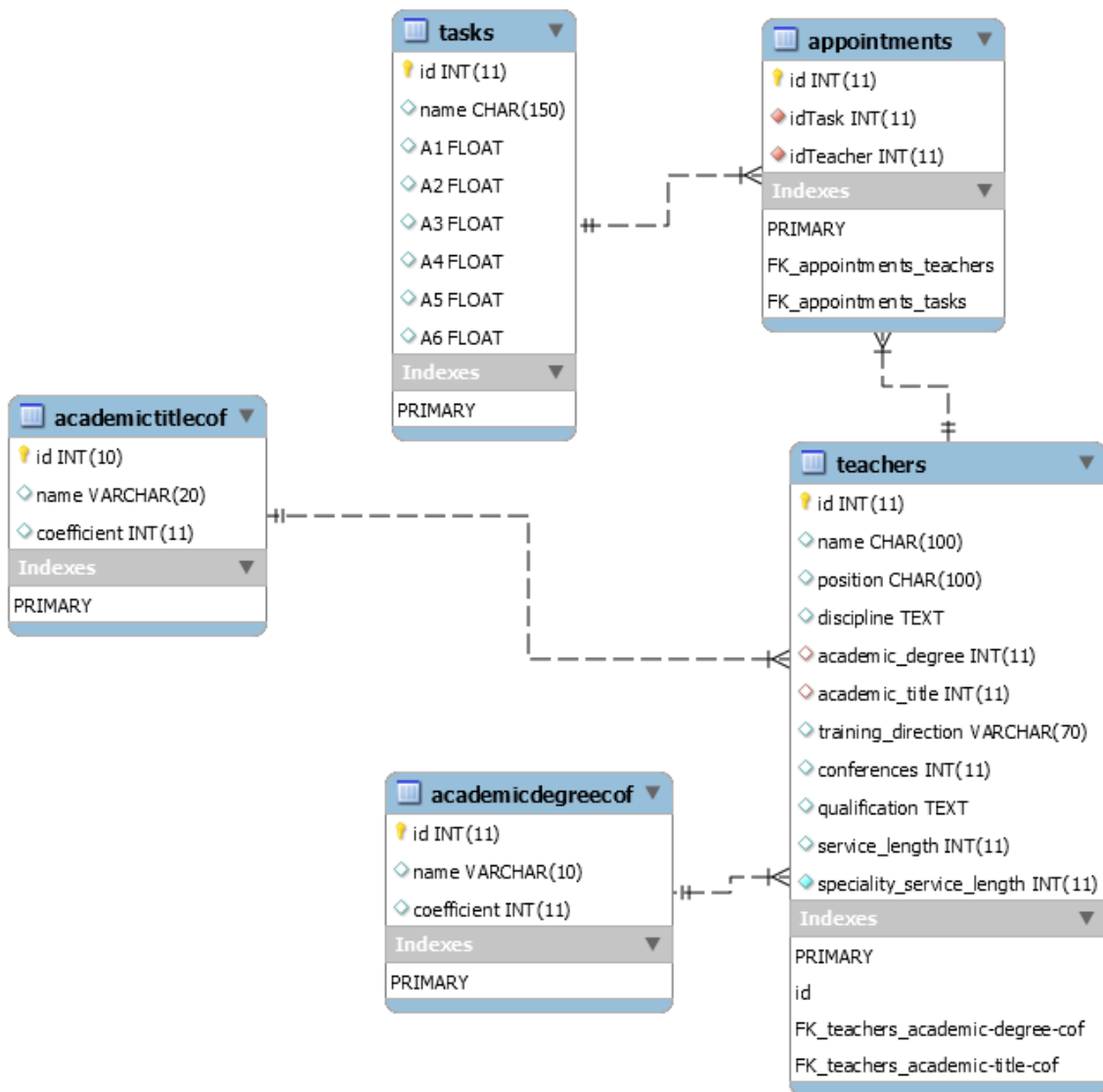


Рисунок 2.9 – Физическая модель данных

Описанная база данных будет использоваться для обработки данных в программном продукте.

2.5 Программная реализация автоматизации процесса распределения персонала

Для того, чтобы реализовать математическую модель, требуется построить диаграмму классов, наглядно демонстрирующую, какие методы и функции содержат классы и как они взаимодействуют между собой. Схема взаимодействия классов представлена на рисунке 2.10.

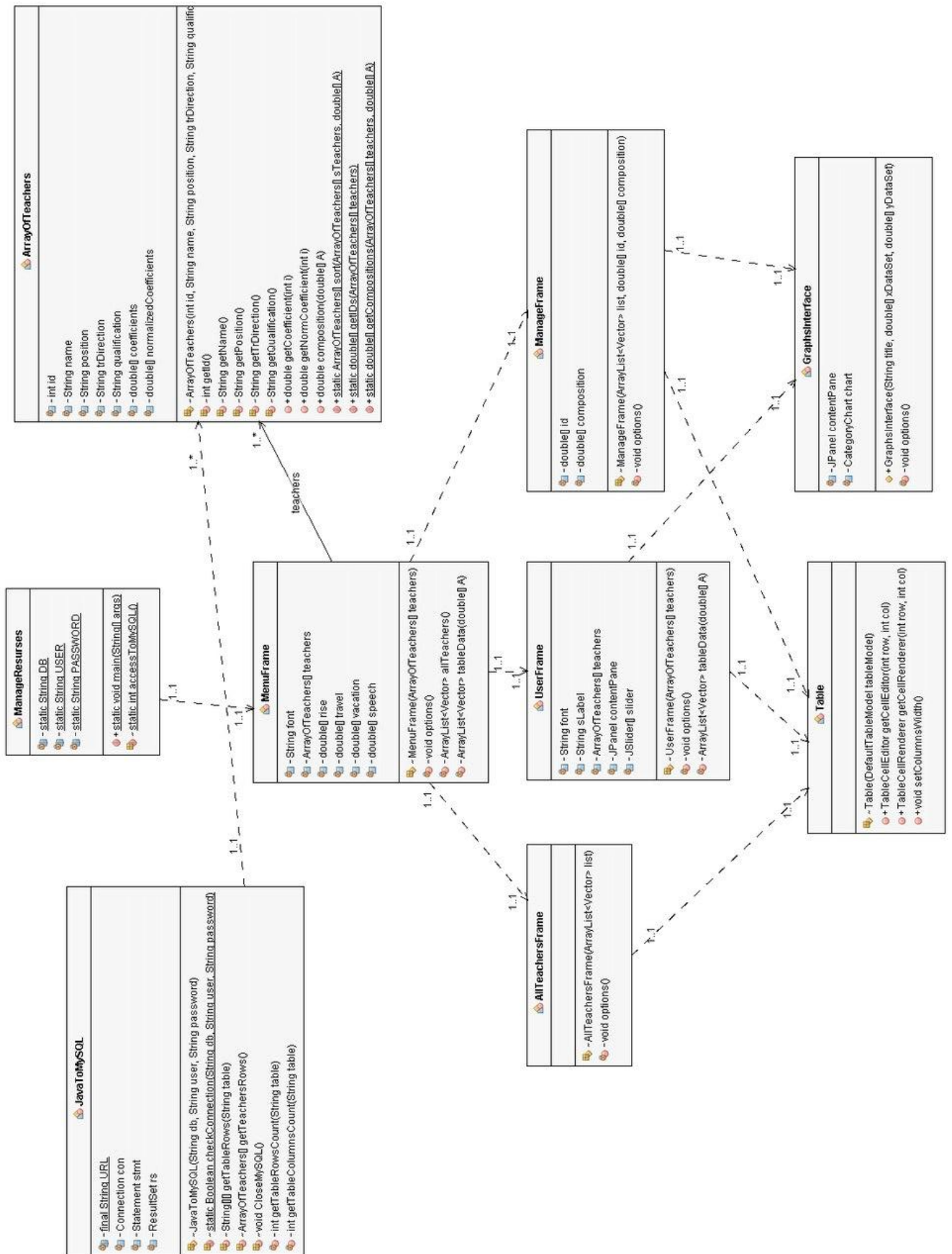


Рисунок 2.10 – Диаграмма классов

В теле главного класса (Приложение А) с помощью функции «accessToMySQL» (Приложение Б) происходит создание формы подключения к базе данных. Данная форма позволяет задать название используемой БД, логин и пароль. В случае, если пользователь нажимает кнопку «ок», давая согласие на попытку создания соединения, то функция возвращает значение 0, иначе – 1.

Соединение с базой данных устанавливается с помощью JDBC – платформенно-независимым промышленным стандартом взаимодействия Java-приложений с различными СУБД [11].

Соединение с базой данных будет установлено если выполнены следующие условия:

- функция «accessToMySQL» возвращает значение 0;
- данные введены верно, и создание соединения возможно, что проверяется с помощью статического метода «checkConection» (Приложение Б) класса «JavaToMySQL».

Если данные введены неверно, то пользователь увидит сообщение об ошибке. Иначе создается объект «db» класса «JavaToMySQL», хранящий данные соединения с БД.

После подключения к базе, метод «getTeachersRows» (Приложение Б) объекта «db» передает в конструктор класса «MenuFrame» (Приложение В) данные типа «ArrayOfTeachers[]».

Метод «getTeachersRows» обрабатывает информацию из базы данных и заносит ее в массив объектов класса «ArrayOfTeachers», предназначенный для удобного хранения и обработки данных.

Затем соединение с базой данных разрывается, так как все необходимые данные получены. Происходит отображение окна меню.

Шрифт текста меню Times New Roman, для кнопок 14пт, для заголовка 16пт.

«MenuFrame» является расширенной реализацией класса «JFrame» из библиотеки «swing» и его задачей является создание меню и дальнейшая работа с данными. В конструкторе класса информация таблицы копируется в

локальную переменную, затем происходит нормализация данных с помощью статичной функции «normalization» (Приложение Б) класса «ArrayOfTeachers», затем инициализируются и добавляются к фрейму объекты типа «JLabel» и «JButton».

К каждой кнопке привязывается класс-наследник от «ActionListener»-а. «ActionListener» отслеживает нажатие кнопки и выполняет действие после нажатия.

Кнопка «Все пользователи» после нажатия передает в конструктор класса «AllTeachersFrame» (Приложение В) данные типа «ArrayList<Vector>», созданные с помощью метода фрейма «allTeachers» (Приложение Б).

Данный метод использует нормализованные данные и представляет их в виде списка из двух векторов:

- вектора «columnNames», хранящего названия для столбцов таблиц;
- вектора «data», хранящего данные об учителях построчно.

Подобное необходимо для корректной работы класса «Table», отображающего информацию об учителях в виде таблицы.

Конструктор «AllTeachersFrame» отображает на экране таблицу данных по всем преподавателям. Класс «Table» дополняет реализацию класса «JTable» библиотеки «swing».

Четыре последующих кнопки «Повышение», «Поездка на конференцию», «Отпуск», «Выступление с докладом» с помощью «ActionListener»-ов и коэффициентов важности, изначально заданных в «MenuFrame», отображают отсортированную таблицу из 15 записей с наивысшим коэффициентом.

«ActionListener»-ы кнопок используют:

- функцию «sort» (Приложение Б) класса «ArrayOfTeachers», которая сортирует данные массива «ArrayOfTeachers» в зависимости от переданных коэффициентов важности;
- функцию «tableData» (Приложение Б), позволяющую генерировать разные «ArrayList<Vector>» в зависимости от переданных коэффициентов важности;

- класс «ManageFrame» (Приложение В), отображающий отсортированную таблицу данных.

После нажатия одной из кнопок меню в конструктор класса «ManageFrame» передается список, созданный функцией «tableData», и массивы типа «double», сгенерированные с помощью функций «getIDs» (Приложение Б) и «getCompositions» (Приложение Б) класса «ArrayOfTeachers».

Данные массивы содержат идентификаторы списка отсортированных преподавателей и значения вычисленной пригодности. Они необходимы для создания диаграммы.

Создание диаграммы происходит путем нажатия кнопки «Построение диаграммы». «ActionListener» данной кнопки передаст массивы типа «double» конструктору класса «GraphsInterface» (Приложение В). «GraphsInterface» является расширенной реализацией класса «JFrame». Для создания диаграммы используется сторонняя, бесплатно распространяемая библиотека «xchart».

Последняя кнопка меню позволяет пользователю самому задать важность критериев.

При нажатии кнопки создается окно фрейма «UserFrame», который содержит слайдеры регулировки. Они позволяют численно задавать важность критериев.

Затем, после выбора всех необходимых критериев и нажатия кнопки «Принять», создается окно отсортированных данных «ManageFrame».

В ходе проделанной работы был написан код для реализации математической модели управления человеческими ресурсами.

Вывод по второй главе

В ходе анализа различных языков программирования в качестве языка разработки был выбран Java.

Для разработки концептуальной модели системы управления человеческими ресурсами была построена диаграмма вариантов использования

в рамках исследуемой области. Данная модель наглядно демонстрирует функции, которые должно будет выполнять разработанное приложение.

Далее на основе нескольких шаблонов был выбран наиболее удобный для пользователя вариант интерфейса. Также была описана исходная база данных и написан код для реализации математической модели управления человеческими ресурсами.

Для проверки эффективности реализации математической модели требуется сделать пробный запуск разработанного приложения и оценить его работу.

Глава 3 РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ РАСПРЕДЕЛЕНИЯ ПЕРСОНАЛА В ОРГАНИЗАЦИИ

3.1 Описание работы программной реализации автоматизации процесса распределения персонала

При запуске приложения пользователь видит окно меню, в котором отображены кнопки имеющихся функций (рисунок 3.1). При выборе одного из пунктов меню происходит расчет показателей эффективности по формуле 2, показанной в первой главе.

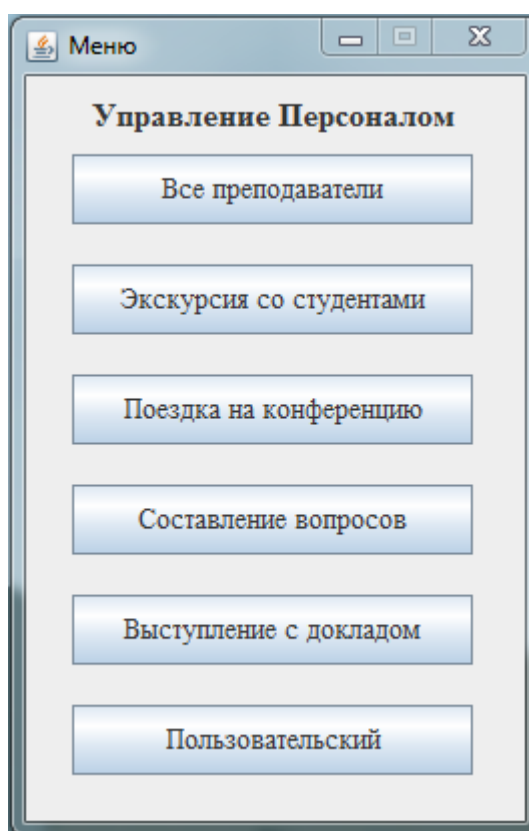


Рисунок 3.1 – Окно основного меню

Результатом расчетов является таблица сотрудников, в которой отображены пятнадцать наиболее подходящих кандидатов для выбранной функции (рисунок 3.2).

id	Имя	Должность	Специальность	Значение функции
3	Быков Сергей Владимирович	Профессор кафедры	Инженерная	2.27
9	Дуюнов Владимир Кузьмич	Профессор кафедры	Правоведение	2.03
14	Ищенко Станислав Андреевич	Профессор кафедры	Правоведение	1.98
22	Пантыкина Марина Ивановна	Профессор кафедры	Философия	1.29
2	Бурда Светлана Юрьевна	Доцент кафедры	Психология	1.26
27	Сахчинская Наталья Сергеевна	Профессор кафедры	Бухгалтерский учет,	1.25
11	Закомолдин Руслан Валериевич	Доцент кафедры	Юриспруденция	1.22
6	Гогина Галина Николаевна	Профессор,	Планирование	1.2
15	Калукова Ольга Макаровна	Доцент кафедры	Математика	1.01
17	Кулишов Евгений Николаевич	Доцент кафедры	Экономика труда	0.99
32	Сюлина Светлана Павловна	Доцент кафедры	Бухгалтерский учет и	0.88
1	Баранов Андрей Андреевич	Доцент кафедры	Промышленное и	0.86
26	Румянцева Елена Леонидовна	Доцент, заведующий	Автоматизация	0.86
12	Закомолдина Тамара Олеговна	Доцент кафедры	Правоведение	0.85
4	Воробьев Павел Викторович	Доцент кафедры	Водоснабжение и	0.83

Построение Диаграммы

Рисунок 3.2 – Окно вывода результатов

Внизу каждой таблицы находится кнопка «Построение Диаграммы», отображающая окно с диаграммой для наглядного отображения информации.

Пример диаграммы показан на рисунке 3.3.

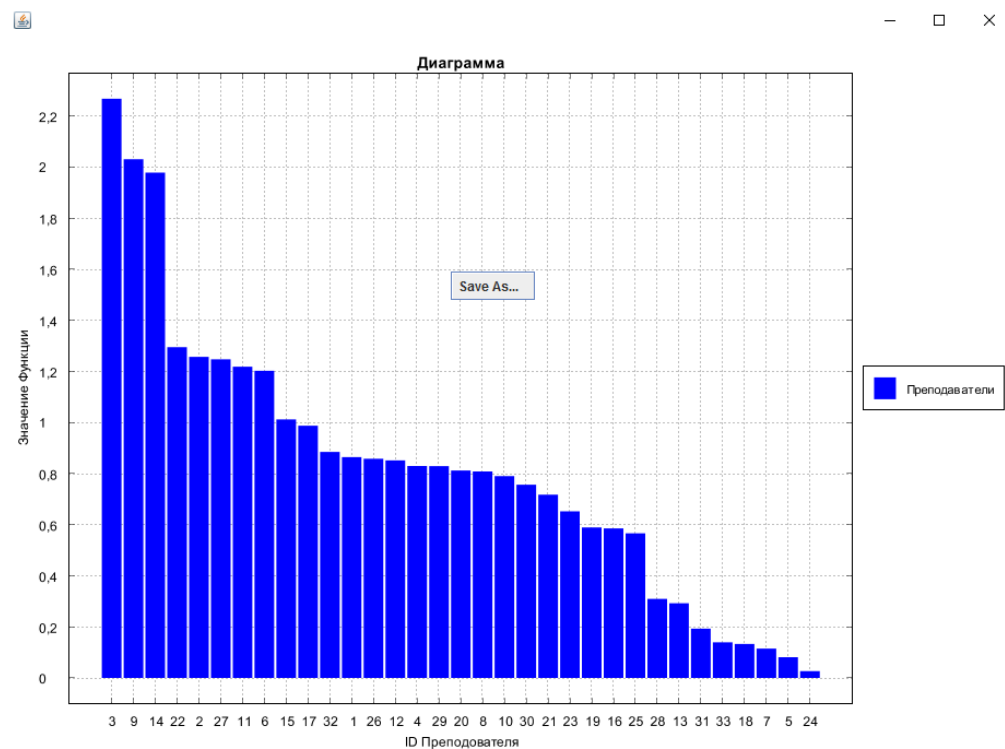


Рисунок 3.3 – Окно диаграммы

Каждую диаграмму можно сохранить в качестве рисунка, если нажать на ней правой кнопкой мыши.

Пользовательский режим позволяет задавать настройки важности критериев самостоятельно, при помощи окна настроек (рисунок 3.4).

После нажатия кнопки «Применить» дальнейший процесс проходит аналогичным образом – отображается окно с результатами расчетов, где расположена кнопка построения диаграммы.

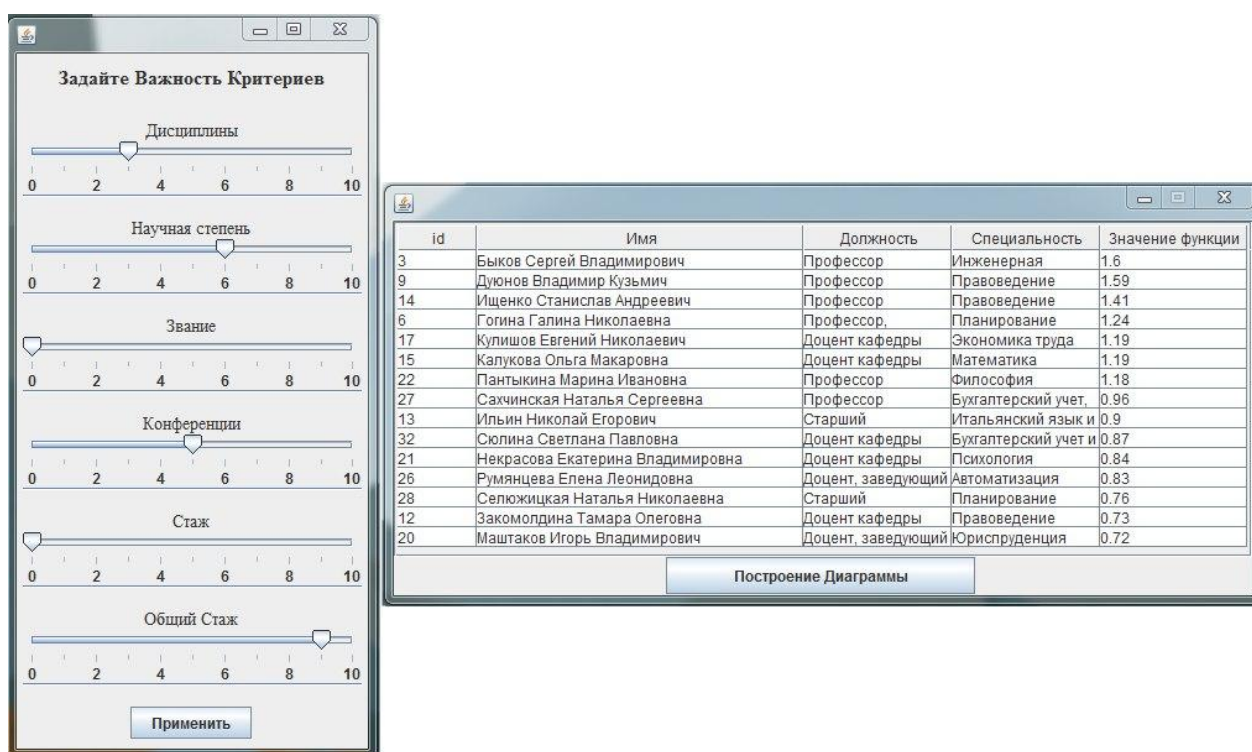


Рисунок 3.4 – Использование пользовательского режима

После подсчета всех критериев происходит решение задачи о назначении, результатом которой является окно, представленное на рисунке 3.5.

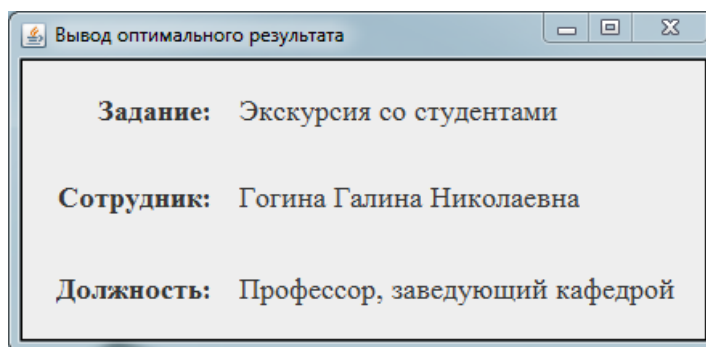


Рисунок 3.5 – Окно вывода оптимального результата

Программа прекращает работу после закрытия окна меню.

При запуске программы ошибок в выполнении не обнаружено, все данные отображаются корректно, вычисления по формулам производятся верно.

3.2 Оценка и обоснование экономической эффективности программного приложения

На принятие решения по управлению человеческими ресурсами в организации «СаГА» уходило примерно 90 минут. При использовании реализованной математической модели затрачиваемое время снижается примерно до 10 минут.

Для расчета прямой эффективности от использования приложения по управлению человеческими ресурсами, требуется расчет показателей стоимостных и трудовых затрат. Информация о формулах для расчета эффективности взята в работе Юртаевой А. Е. «Методические основы определения экономической эффективности» [10].

Абсолютное снижение трудовых затрат, рассчитывается по формуле 5:

$$\Delta T = T_0 - T_1 = 90 - 10 = 80(\text{мин}) \quad (5)$$

где T_0 – время, затрачиваемое на принятие решения до внедрения автоматизации;

T_1 – время, затрачиваемое на принятие решения после внедрения.

Коэффициент относительного снижения трудовых затрат KT (в процентах) рассчитывается по формуле 6:

$$KT = (\Delta T / T_0) * 100\% = (80/90) * 100\% = 88,89\% \quad (6)$$

Индекс снижения трудовых затрат рассчитывается по формуле 7:

$$YT = T_0 / T_1 = 90/10 = 9 \quad (7)$$

К стоимостным показателям относятся абсолютное снижение, коэффициент относительного снижения и индекс снижения стоимостных затрат.

Рассчитаем снижение стоимостных затрат при средней зарплате руководителя в 50 тысяч рублей (312,5 р/час).

Абсолютное снижение стоимостных затрат, рассчитывается по формуле 8:

$$C = C0 - C1 = 312,5 * 1,5 - 312,5 * (1/6) = 468,75 - 52,1 = 416,65 (\text{руб}) \quad (8)$$

где C0 является затратами на принятие решения до внедрения автоматизации;

C1 – при принятии решения после внедрения автоматизации (в расчете при зарплате руководителя в 50 тыс.р.).

Коэффициент относительного снижения стоимостных затрат, рассчитывается по формуле:

$$KC = (C / C0) * 100\% = 416,65 / 468,75 * 100\% = 88,89\% \quad (9)$$

Индекс снижения стоимостных затрат, рассчитывается по формуле:

$$YC = C0 / C1 = 468,75 / 52,1 = 8.997 \quad (10)$$

Коэффициент относительного снижения стоимости затрат KC и индекс снижения стоимостных затрат YC показывают рост производительности труда за счет внедрения автоматизации в рабочий процесс.

Далее требуется рассчитать показатели экономической эффективности, представленных в таблице 3.1.

Таблица 3.1 – Показатели эффективности от внедрения программного продукта

	Затраты		Абсолютное изменение затрат	Коэффициент изменения затрат	Индекс изменения затрат
	Базовый вариант	Проектный вариант			
Трудоемкость	T0(час)	T1(час)	T=T0-T1	KT=T/T0*100%	YT=T0/T1
	90	10	80	88,89	9
Стоимость	C0(руб)	C1(руб)	C=C0-C1	KC=C/C0*100%	YC=C0/C1
	468,75	52,1	416,65	88,89	8.997

Для наглядного представления расчетов экономии трудовых и стоимостных затрат, представленных в таблице 3.1, построим диаграммы. На рисунке 3.6 представлена диаграмма, которая показывает разницу между

показателями трудоемкости до внедрения автоматизации в процесс управления персоналом и после внедрения.

Представленная диаграмма показывает, что время, необходимое на принятие решения по управлению сотрудниками, снижено в 9 раз благодаря внедрению автоматизации в процесс управления человеческими ресурсами.



Рисунок 3.6 – Диаграмма изменения трудозатрат

На рисунке 3.7 представлена диаграмма изменения стоимостных затрат на прием и обработку заявки.

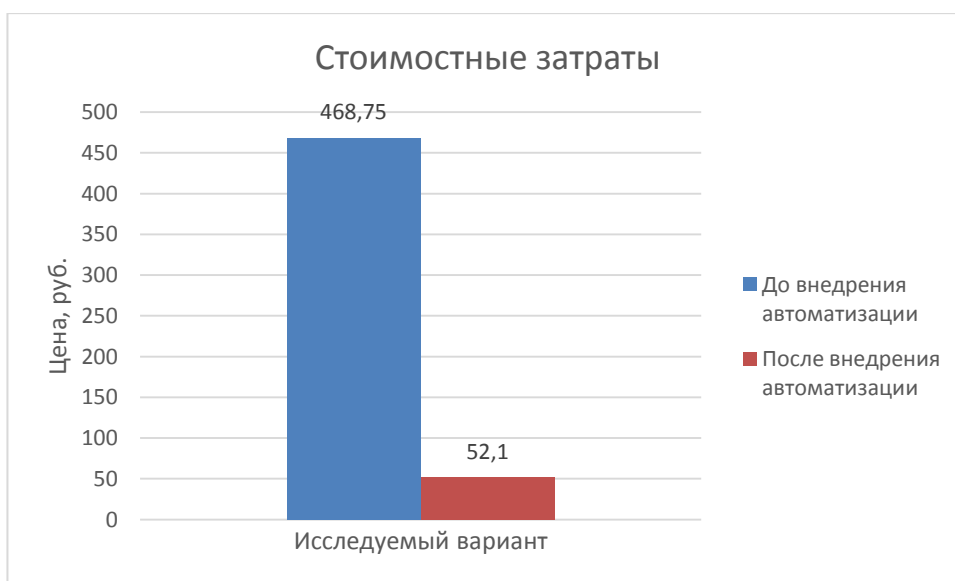


Рисунок 3.7 – Диаграмма изменения стоимостных затрат

Представленная диаграмма показывает, что стоимость затрат на принятие решений по управлению человеческими ресурсами, уменьшилось почти в 9 раз благодаря внедрению автоматизации.

Следовательно, реализованная математическая модель по управлению человеческими ресурсами для организации «СаГА» является актуальным решением для уменьшения временных и стоимостных затрат при управлении человеческими ресурсами.

Вывод по третьей главе

Реализованная математическая модель удовлетворила потребности руководителя организации. Осуществлена возможность автоматизированного распределения персонала.

Проведенный анализ экономической эффективности, показал, что внедрение реализованной математической модели, имеет большую значимость для организации.

ЗАКЛЮЧЕНИЕ

Использование автоматизации различных процессов становится неотъемлемой частью деятельности современных организаций.

В данной работе был выделен объект исследования, проанализирована предметная область, сформулированы цель и задачи. Был проведен анализ деятельности организации «СаГА» и выявлена необходимость оптимизации процесса распределения персонала.

Была сформулирована задача распределения персонала и изучены возможные способы ее решения, после чего сделан вывод, что использование табличного редактора занимает много времени, поэтому стоит воспользоваться венгерским алгоритмом, переложенным в программную модель.

После этого были рассмотрены существующие системы распределения персонала, выделены их основные преимущества и недостатки, после чего было принято решение разработать собственный программный продукт.

Были рассмотрены основные языки программирования и в качестве языка реализации математической модели был выбран язык Java. Для формирования общих требований к программному приложению были построены диаграмма вариантов использования, концептуальная и физическая модель базы данных и диаграмма классов на основе графического языка объектно-ориентированного моделирования – UML. Для управления данными выбрана СУБД MySQL.

В ходе работы была реализована математическая модель распределения персонала, выполняющая следующие функции: расчет эффективности сотрудников при выполнении задач, выбор рекомендуемых кандидатов для решения задачи. Также реализована возможность вывода полного списка сотрудников и самостоятельного ввода настроек, по которым осуществляется расчет эффективности.

Реализованное приложение позволит автоматизировать процесс распределения персонала, что уменьшит временные затраты на обработку информации и исключит возможные ошибки, связанные с человеческим фактором.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Егоров А.Ф., Бельков В.П., Савицкая Т.В. Методические указания к лабораторным работам по курсу “Математическое моделирование и методы синтеза гибких химических производств”. М.РХТУ, 2014. - 40с.
2. Даконта, М.; Саганич, А. XML и Java 2; СПб: Питер, 2010. - 384 с.
3. Кибанов А.Я. Основы управления персоналом: учеб. - М.: ИНФРА-М, 2011. - 304 с.
4. Колокольцов, В.Н. Математическое моделирование многоагентных систем конкуренции и кооперации (Теория игр для всех): Учебное пособие / В.Н. Колокольцов, О.А. Малафеев. - СПб.: Лань, 2012. - 624 с.
5. Леоненков, А.В. Самоучитель UML 2 / А.В. Леоненков. – СПб.: БХВ - Петербург, 2007. – 576с.
6. М. Эллис, Б. Строуструп. Справочное руководство по языку C++ с комментариями: Пер. с англ. - Москва: Мир, 1992. 445с.
7. Новаковская О.А. Управление человеческими ресурсами в неустойчивых деловых организациях: Учебное пособие для студентов экономических специальностей. - Улан-Удэ: ВСГТУ, 2007.
8. Пугачев С., Шериев А., Кичинский К. Разработка приложений для Windows 8 на языке C#. - СПб.: БХВ-Петербург, 2013. - 416 с.
9. Урубков, А.Р. Методы и модели оптимизации управленческих решений: Учебное пособие / А.Р. Урубков, И.В. Федотов. - М.: ИД Дело РАНХиГС, 2012. - 240 с.
10. Юртаева А.Е. Методические основы определения экономической эффективности. - М.: LAP Lambert Academic Publishing, 2011. - 136 с.

Электронные ресурсы

11. Java SE Technologies – Database // Oracle [Электронный ресурс]: <https://www.oracle.com/index.html> (дата обращения: 1.05.2017).

12. Мини-Кадры бесплатная программа автоматизации кадровой службы // КОРС-СОФТ программы для бизнеса [Электронный ресурс]: <http://www.kors-soft.ru/index.htm> (дата обращения: 21.03.2017).

13. Нормировка показателей // Подготовка и обработка, анализ и представление данных [Электронный ресурс]: <https://psyttest.wordpress.com> (дата обращения: 18.04.2017).

14. Сазонов В.Ф. Должности, степени и звания в вузе // Кинезиолог, 2009-2014 [Электронный ресурс]: понятие о научных, учебных и производственных "титулах" сотрудников вуза: <http://kineziolog.bodhy.ru> (дата обращения: 7.03.2017).

15. Система управления персоналом // Программа для распределения задач [Электронный ресурс]: <http://www.leadercommand.ru> (дата обращения: 20.04.17).

Литература на иностранных языках

16. "...the very game..." A Tutorial on Mathematical Modeling, 2016, Michael P. McLaughlin. – PP. 70–212

17. A Mathematical Model of Collective Cell Migration in a Three-Dimensional, Heterogeneous Environment, 2015, David P. Stonko, Lathiena Manning, Michelle Starz-Gaiano, Bradford E. Peercy. – PP. 89–111

18. Construction of a mathematical model for tuberculosis transmission in highly endemic regions of the Asia-pacific, 2014, James M. Trauera, Justin T. Denholmb, Emma S. McBrydea. – PP. 74–84

19. Finite element analysis and modeling of structure with bolted joints, 2014, Jeong Kima, Joo-Cheol Yoonb, Beom-Soo Kangc. – PP. 895–911

20. Modeling and simulation of a gas distribution pipeline network, 2013, A. Herrán-González, J.M. De La Cruz, B. De Andrés-Toro, J.L. Risco-Martín. – PP. 1584–1600

ПРИЛОЖЕНИЕ А

Код главного класса

```
public static void main(String[] args) {  
    if(accessToMySQL() == 0 &&  
        JavaToMySQL.checkConnection(DB, USER, PASSWORD) == true){  
        JavaToMySQL db = new JavaToMySQL(DB, USER, PASSWORD);  
        new MenuFrame(db.getTeachersRows()).setVisible(true);  
        db.CloseMySQL();  
    }  
}
```


ПРИЛОЖЕНИЕ Б

Код основных функций программного приложения

Код функции «accessToMySQL»

```
static int accessToMySQL(){
    JTextField db = new JTextField(10);
    JTextField user = new JTextField(10);
    JPasswordField password = new JPasswordField(20);
    db.setText(DB);
    user.setText(USER);
    password.setText(PASSWORD);
    password.setToolTipText("Ваш пароль должен содержать: "
        + "Цифры, буквы, знаки пунктуации, завязку развитие, "
        + "кульминацию и неожиданный финал с двумя шутками про хохлов.");
    JPanel myPanel = new JPanel();
    myPanel.setLayout(new BorderLayout(myPanel, BorderLayout.PAGE_AXIS));
    myPanel.add(new JLabel("База данных:"));
    myPanel.add(Box.createVerticalStrut(5));
    myPanel.add(db);
    myPanel.add(Box.createVerticalStrut(5));
    myPanel.add(new JLabel("Логин:"));
    myPanel.add(Box.createVerticalStrut(5));
    myPanel.add(user);
    myPanel.add(Box.createVerticalStrut(5));
    myPanel.add(new JLabel("Пароль:"));
    myPanel.add(Box.createVerticalStrut(5));
    myPanel.add(password);
    myPanel.add(Box.createVerticalStrut(10));
    int result = JOptionPane.showConfirmDialog(null, myPanel,
        "Подключение", JOptionPane.OK_CANCEL_OPTION);
    if (result == JOptionPane.OK_OPTION) {
        DB = db.getText(); USER = user.getText(); PASSWORD = password.getText();
    }
    return result;
}
```

Код функции «checkConection»

```
static Boolean checkConnection(String db, String user, String password){
    String sURL = URL + db + "?useSSL=false";
    try {
        // opening database connection to MySQL server
        Connection con = DriverManager.getConnection(sURL, user, password);
        try { con.close(); } catch(SQLException se) { /*can't do anything */ }
        return true;
    } catch (SQLException sqlEx) {
        JOptionPane.showMessageDialog(new JPanel(),
            "Не возможно установить соединение.", "Ошибка",
            JOptionPane.ERROR_MESSAGE);
        return false;
    }
}
```

Код функции «getTeachersRows»

```
ArrayOfTeachers[] getTeachersRows() {
    int row = getTableRowsCount("teachers");
    ArrayOfTeachers[] teachers = new ArrayOfTeachers[row];
    if(row != 0){
        String query = "SELECT "
            + "teachers.id, " + "teachers.name, "
            + "teachers.position, " + "teachers.training_direction, "
            + "teachers.qualification, " + "teachers.discipline, "
            + "academicDegreeCof.coefficient, " + "academicTitleCof.coefficient, "
            + "teachers.conferences, " + "teachers.service_length, "
            + "teachers.speciality_service_length " + "FROM teachers "
            + "JOIN academicDegreeCof on academicDegreeCof.id = teachers.academic_degree "
            + "JOIN academicTitleCof on academicTitleCof.id = teachers.academic_title "
            + "ORDER BY id";

        try {
            stmt = con.createStatement();
            rs = stmt.executeQuery(query);
            int i = 0;
            while (rs.next()) {
                int id = rs.getInt(1);
                String name = rs.getString(2);
                String position = rs.getString(3);
                String trDirection = rs.getString(4);
                String qualification = rs.getString(5);
                int discipline = rs.getInt(6);
                int degree = rs.getInt(7);
                int tittle = rs.getInt(8);
                int conferences = rs.getInt(9);
                int servLength = rs.getInt(10);
                int specServLength = rs.getInt(11);
                teachers[i] = new ArrayOfTeachers(id, name, position, trDirection, qualification,
                    discipline, degree, tittle, conferences, servLength, specServLength);
                i++;
            }
        } catch (SQLException sqlEx) {
            sqlEx.printStackTrace();
        }
    }
    return teachers;
}
```

Код функции «normalization»

```
public static void normalization(ArrayOfTeachers[] teachers) {
    int N = teachers.length;
    for(int k = 0; k < 6; k++){
        double max = teachers[0].coefficients[k];
        double min = teachers[0].coefficients[k];
        for(int i = 1; i < N; i++){
            if(teachers[i].coefficients[k] < min)
                min = teachers[i].coefficients[k];
            if(teachers[i].coefficients[k] > max)
                max = teachers[i].coefficients[k];
        }
        for(int i = 0; i < N; i++){
            teachers[i].normalizedCoefficients[k] =
                (teachers[i].coefficients[k] - min) / (max - min);
        }
    }
}
```

Код функции «getCompositions»

```
public static double[] getCompositions(ArrayOfTeachers[] teachers, double[] A){
    double[] buff = new double[teachers.length];
    for(int i = 0; i < teachers.length; i++){
        buff[i] = teachers[i].composition(A);
    }
    return buff;
}
```

Код функции «allTeachers»

```
private ArrayList<Vector> allTeachers(){
    ArrayList<Vector> list = new ArrayList<Vector>();
    Vector columnNames = new Vector();
    Vector data = new Vector();
    columnNames.add("id");
    columnNames.add("Имя");
    columnNames.add("Должность");
    columnNames.add("Квалификация");
    columnNames.add("Специальность");
    for(int i = 0; i < teachers.length; i++){
        Vector obj = new Vector();
        obj.add(String.format("%d", teachers[i].getId()));
        obj.add(teachers[i].getName());
        obj.add(teachers[i].getPosition());
        obj.add(teachers[i].getQualification());
        obj.add(teachers[i].getTrDirection());
        data.add(obj);
    }
    list.add(data);
    list.add(columnNames);
    return list;
}
```

Код функции «sort»

```
public static ArrayOfTeachers[] sort(ArrayOfTeachers[] sTeachers, double[] A){
    ArrayOfTeachers[] teachers = Arrays.copyOf(sTeachers, sTeachers.length);
    for (int i = teachers.length - 1; i >= 0; i--) {
        for (int j = 0; j < i; j++) {
            if (teachers[j].composition(A) < teachers[j + 1].composition(A)) {
                ArrayOfTeachers t = teachers[j];
                teachers[j] = teachers[j + 1];
                teachers[j + 1] = t;
            } else {
            }
        }
    }
    return teachers;
}
```

Код функции «tableData»

```
private ArrayList<Vector> tableData(double[] A){
    ArrayOfTeachers[] fTeachers = ArrayOfTeachers.sort(this.teachers, A);
    ArrayList<Vector> list = new ArrayList<Vector>();

    Vector columnNames = new Vector();
    Vector data = new Vector();

    columnNames.add("id");
    columnNames.add("Имя");
    columnNames.add("Должность");
    columnNames.add("Специальность");
    columnNames.add("Значение функции");

    double t = 0;
    for(int i = 0; i < 15; i++){
        Vector obj = new Vector();
        obj.add(Integer.toString(fTeachers[i].getId()));
        obj.add(fTeachers[i].getName());
        obj.add(fTeachers[i].getPosition());
        obj.add(fTeachers[i].getTrDirection());
        t = Math rint(100.0 * fTeachers[i].composition(A)) / 100.0;
        obj.add(Double.toString(t));

        data.add(obj);
    }
    list.add(data);
    list.add(columnNames);
    return list;
}
```

Код функции «getIDs»

```
public static double[] getIDs(ArrayOfTeachers[] teachers){
    double[] buff = new double[teachers.length];
    for(int i = 0; i < teachers.length; i++)
        buff[i] = teachers[i].getId();
    return buff;
}
```

ПРИЛОЖЕНИЕ В

Конструкторы классов

Код конструктора класса «MenuFrame»

```
MenuFrame(ArrayOfTeachers[] teachers){
    super("Меню");
    this.teachers = teachers;
    ArrayOfTeachers.normalization(this.teachers);
    options();
    String sJButton[] = {"Все преподаватели", "Повышение",
                        "Поездка на конференцию", "Отпуск",
                        "Выступление с докладом", "Пользовательский"};
    String sTitle[] = {"Управление Человеческими", "Ресурсами"};
    JButton button[] = new JButton[sJButton.length];
    JLabel title[] = new JLabel[sTitle.length];
    for(int i = 0; i < sTitle.length; i++){
        title[i] = new JLabel(sTitle[i]);
        title[i].setAlignmentX(Component.CENTER_ALIGNMENT);
        title[i].setFont(new Font(font, Font.BOLD, 16));
        title[i].setBorder(new CompoundBorder(title[i].getBorder(),
            new EmptyBorder(10, 10, i * 10, i * (-10) + 10)));
        getContentPane().add(title[i]);
    }
    for(int i = 0; i < sJButton.length; i++){
        button[i] = new JButton(sJButton[i]);
        button[i].setFont(new Font(font, Font.PLAIN, 14));
        button[i].setAlignmentX(Component.CENTER_ALIGNMENT);
        button[i].setPreferredSize(new Dimension(200, 35));
        button[i].setMaximumSize(new Dimension(200, 35));
        getContentPane().add(button[i]);
        getContentPane().add(Box.createHorizontalStrut(6));
    }
    button[0].addActionListener(new ActionListenerAll());
    button[1].addActionListener(new ActionListenerRise());
    button[2].addActionListener(new ActionListenerTravel());
    button[3].addActionListener(new ActionListenerVacation());
    button[4].addActionListener(new ActionListenerSpeech());
    button[5].addActionListener(new ActionListenerUser());
}
```

Код конструктора класса «AllTeachersFrame»

```
AllTeachersFrame(ArrayList<Vector> list){
    options();
    DefaultTableModel tableModel = new DefaultTableModel(list.get(0), list.get(1));
    Table table = new Table(tableModel);
    JScrollPane scrollPane = new JScrollPane(table);
    Border margin = new EmptyBorder(2, 2, 2, 2);
    scrollPane.setBorder(new CompoundBorder(scrollPane.getBorder(), margin));

    add(scrollPane);
}
```

Код конструктора класса «ManageFrame»

```
ManageFrame(ArrayList<Vector> list, double[] id, double[] composition){
    //super("-", "...", "-");
    this.id = id;
    this.composition = composition;
    options();
    DefaultTableModel tableModel = new DefaultTableModel(list.get(0), list.get(1));
    Table table = new Table(tableModel);
    JScrollPane scrollPane = new JScrollPane(table);
    Border margin = new EmptyBorder(2,2,2,2);
    scrollPane.setBorder(new CompoundBorder(scrollPane.getBorder(), margin));

    JButton chartButton = new JButton("Построение Диаграммы");
    chartButton.addActionListener(new ButtonListener());
    chartButton.setAlignmentX(Component.CENTER_ALIGNMENT);
    chartButton.setPreferredSize(new Dimension(250, 33));
    chartButton.setMaximumSize(new Dimension(250, 33));

    add(scrollPane);
    add(chartButton);
}
```

Код конструктора класса «GraphsInterface»

```
public GraphsInterface(String title, double[] xDataSet, double[] yDataSet) {
    options();
    chart.addSeries(title, xDataSet, yDataSet);
    contentPane.add(new XChartPanel(chart));
    contentPane.validate();
}
```