

**Министерство образования и науки Российской Федерации**  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт кибернетики  
Направление подготовки Прикладная математика и информатика  
Кафедра прикладной математики

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

**Информационная система поддержки принятия решений в задачах  
организации учебного процесса 2го уровня**

УДК \_\_\_\_\_

Студент

8БМ41	Пискунова Татьяна Александровна			
Группа	ФИО	Подпись	Дата	

Руководитель

профессор	Берестнева О.Г.	Д.Т.Н.		
Должность	ФИО	Ученая степень, звание	Подпись	Дата

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

доцент	Конотопский В.Ю.	К.Э.Н.		
Должность	ФИО	Ученая степень, звание	Подпись	Дата

По разделу «Социальная ответственность»

доцент	Анищенко Ю.В.	К.Т.Н.		
Должность	ФИО	Ученая степень, звание	Подпись	Дата

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Зав. кафедрой	Гергет Ольга Михайловна	К.Т.Н.		
Должность	ФИО	Ученая степень, звание	Подпись	Дата

Томск – 2016 г.

**Министерство образования и науки Российской Федерации**  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт кибернетики  
Направление подготовки Прикладная математика и информатика  
Кафедра Прикладной математики

УТВЕРЖДАЮ:  
Зав. кафедрой

\_\_\_\_\_

(Подпись)

\_\_\_\_\_

(Дата)

\_\_\_\_\_

(Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

магистерской диссертации

Студенту:

Группа	ФИО
8БМ41	Пискуновой Татьяне Александровне

Тема работы:

Информационная система поддержки принятия решений в задачах организации учебного процесса 2го уровня

Утверждена приказом директора Института кибернетики (дата, номер)

Срок сдачи студентом выполненной работы: (дата)

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

**Исходные данные к работе**

*(наименование объекта исследования; документы конференции и отчеты НИИР; программное обеспечение).*

Объектом исследования являются учебная и научная деятельность студентов в процессе обучения в университете, а также их личностные и психологические характеристики. Литература по теме, Erwin Data Modeler 9 Community Edition, SQL Developer, Oracle 11g Database, Oracle Apex, данные об учащих в файлах различных форматов

<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b></p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования; содержательная, концептуальная, математическая структурная модель; разработка алгоритма решения задачи; выбор программного обеспечения; индивидуальные выводы о практической значимости проведенных исследований; дополнительные разделы, подлежащие разработке; заключение по работе).</i></p>	<p>аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования; разработка алгоритма решения задачи; выбор программного обеспечения; индивидуальные выводы о практической значимости проведенных исследований; разработка базы данных и интерактивной системы для решения задачи; заключение по работе.</p>
--	---

<p><b>Перечень графического материала</b></p> <p><i>(с точным указанием обязательных чертежей)</i></p>	
--	--

<p><b>Консультанты по разделам выпускной квалификационной работы</b></p> <p><i>(с указанием разделов)</i></p>	
---	--

Раздел	Консультант
Финансовый менеджмент	Конотопский В.Ю.
Ресурсоэффективность и ресурсосбережение	Конотопский В.Ю.
Социальная ответственность	Анищенко Ю.В.
Основная часть	Мокина Е.Е.
Английский язык	Сидоренко Т.В.

<p><b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b></p> <ol style="list-style-type: none"> <li>1. Построение дерева решений и базы знаний</li> <li>2. Построение базы данных</li> <li>3. Проектирование программного интерфейса</li> </ol>
---

<p><b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b></p>	
--	--

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
профессор	Берестнева О.Г.	Д.Т.Н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8БМ41	Пискунова Татьяна Александровна		

**Министерство образования и науки Российской Федерации**  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт кибернетики

Направление подготовки Прикладная математика и информатика

Уровень образования магистр

Кафедра прикладной математики

Период выполнения \_\_\_\_\_ осенний / весенний семестр 2015/2016 учебного года \_\_\_\_\_

Форма представления работы:

магистерская диссертация

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН**  
**выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
10.01.2015	Литературный обзор по теме диссертации	20
7.04.2016	Выполнение задания по теме диссертации	30
31.05.2016	Обсуждение результатов и оформление работы	50

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
профессор	Берестнева О.Г.	д.т.н.		

**СОГЛАСОВАНО:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
	Гергер. О.М.	к.т.н.		

## РЕФЕРАТ

Выпускная квалификационная работа \_\_\_\_\_ 110 \_\_\_\_\_ с., \_\_\_\_\_ 29 \_\_\_\_\_ рис., \_\_\_\_\_ 18 \_\_\_\_\_ табл.,  
\_\_\_\_\_ 91 \_\_\_\_\_ источников, \_\_\_\_\_ 4 \_\_\_\_\_ прил.

Ключевые слова: \_\_\_\_\_ СППР, классификация, регрессия, нечетка логика, data mining,  
нечеткие деревья решений, БД, базы знаний

Объектом исследования является (ются) \_\_\_\_\_ характеристики \_\_\_\_\_ и \_\_\_\_\_ деятельность  
студента во время обучения, влияющая на его учебно-научный потенциал

Цель работы – \_\_\_\_\_ создание ИСППР для оценки \_\_\_\_\_ учебно-научного \_\_\_\_\_ потенциала  
студентов

В процессе исследования проводились \_\_\_\_\_ обработка \_\_\_\_\_ и \_\_\_\_\_ анализ \_\_\_\_\_ экспериментальных  
данных о студентах ИК за последние пять лет

В результате исследования \_\_\_\_\_ проведено исследование экспериментальных данных и  
создана СППР, позволяющая просматривать и редактировать данные о студентах, составлять  
рейтинг по направлениям и проводить классификацию по успешности в учебной и научной  
деятельности

Степень внедрения: \_\_\_\_\_ работающий прототип системы, готов к внедрению

Область применения: \_\_\_\_\_ социальная сфера

Экономическая эффективность/значимость работы \_\_\_\_\_ повышение \_\_\_\_\_ эффективности  
принятых решений, снижение затрат на сбор и обработку информации

В будущем планируется \_\_\_\_\_ расширение функциональности за счет внедрения  
новых задач и методов

## Определения и ссылки

База данных – информационная модель, позволяющая упорядоченно хранить данные о группе объектов, обладающих одинаковым набором свойств. Программное обеспечение, предназначенное для работы с базами данных, называется система управления базами данных (СУБД).

База знаний – (БЗ; англ. knowledge base, KB) — база данных, содержащая правила вывода и информацию о человеческом опыте и знаниях в некоторой предметной области.

Дерево решений – средство поддержки принятия решений, используемое в статистике и анализе данных для прогнозных моделей. Структура дерева представляет собой «листья» и «ветки».

Классификация – особый случай применения логической операции деления объема понятия, представляющий собой некоторую совокупность делений.

Нечеткая логика – раздел математики, являющийся обобщением классической логики и теории множеств, базирующийся на понятии нечёткого множества как объекта с функцией принадлежности элемента к множеству, принимающей любые значения в интервале, а не только 0 и 1.

Регрессия – математическое выражение, отражающее зависимость зависимой переменной  $y$  от независимых переменных  $x$  при условии, что это выражение будет иметь статистическую значимость.

СППР (система поддержки принятия решений) – компьютерная автоматизированная система, целью которой является помощь людям, принимающим решение в сложных условиях для полного и объективного анализа предметной деятельности.

СУБД – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Data mining – собирательное название, используемое для обозначения совокупности методов обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.

## Оглавление

Введение	7
1. Обзор литературы	10
2. Постановка задачи	18
3. Объект и методы исследования	22
3.1. Объект исследования	22
3.2. Методы исследования	22
3.3. Выбор программного обеспечения	33
4. Математическое и программное обеспечение ИСППР	41
4.1. Построение онтологии предметной области	41
4.2. Выделение основных факторов и характеристик	47
4.2.1. Определение функциональности системы	47
4.2.2. Определение участвующих в решении переменных	49
4.3. Построение деревьев решений	55
4.3.1. Определение лингвистических переменных	55
4.3.2. Классификация	58
4.3.3. Регрессия	60
4.4. Построение базы данных	63
4.4.1. Выбор базовой структуры организации БД	63
4.4.2. Построение полной схемы БД	65
4.5. Проектирование программного интерфейса	69
4.5.1. Определение структуры приложения	69
4.5.2. Схема переходов	70
4.6. Обеспечение функциональности поддержки принятия решений	77
5. Результаты проведенного исследования	80
6. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	84
6.1. Организация и планирование работ	84
6.2. Расчет сметы затрат на выполнение проекта	88
6.3. Оценка экономической эффективности проекта	93
7. Социальная ответственность	95
7.1. Введение	95
7.2. Производственная безопасность	95
7.3. Экологическая безопасность	99
7.4. Безопасность в чрезвычайных ситуациях.	100
7.5. Правовые и организационные вопросы обеспечения безопасности.	101
Заключение	102
Список публикаций	105
Список литературы	106
Приложения	111

## Введение

**Актуальность темы.** Образование играет значительную роль в модернизации общества и экономики. Без конкурентоспособного образования переход к инновационной экономике невозможен, так как от уровня образования напрямую зависит качество трудовых ресурсов. Жесткая конкуренция даже на внутреннем рынке образовательных услуг заставляет учреждения совершенствовать качество своей работы путем поиска оптимальных управленческих решений и анализа деятельности уже поступивших студентов. Но увеличение объема информации, поступающей непосредственно к руководителям, необходимость учета большого числа взаимосвязанных факторов и быстро меняющейся обстановки настоятельно требуют использования ЭВМ при принятии решений. В связи с этим целесообразно внедрения системы поддержки принятия решений (СППР) в задачах организации учебного процесса, способной оценивать потенциал обучающихся для выбора альтернатив при принятии управленческих решений. Для успешной разработки ИСППР необходимо точно определить область знаний, в которой информация зачастую имеет слабоструктурированный характер, нуждающийся в формализации и формировании онтологической модели предметной области.

ВКР выполнялась в рамках инициативного проекта РФФИ «Модели и алгоритмы адаптации субъектов деятельности к измененным социальным условиям».

**Целью** работы является построение информационной системы поддержки принятия решений в задачах организации учебного процесса второго уровня.

В первую очередь, стоит отметить, что всю работу можно разделить на две отдельных компоненты, выполнение которых можно производить параллельно: создание информационной системы и математический аппарат



обработки данных. В рамках данного условия для достижения заданной цели необходимо решить ряд **задач**:

- 1) определить состав информационной базы для оценки учебного и научного потенциала студентов ВУЗа;
- 2) формализовать и описать предметную область деятельности и характеристик студентов университета на основе онтологической модели данных.
- 3) разработать и программно реализовать алгоритм оценки учебного и научного потенциала студентов, основанный на информационной базе знаний;
- 4) разработать и реализовать в составе интеллектуальной информационной системы методику поддержки принятия решений при оценке обучающихся.
- 5) спроектировать внешнюю оболочку для удобной работы с системой.

**Объект исследования** – учебная и научная деятельность студентов в процессе обучения в университете, а также их личностные и психологические характеристики.

**Предмет исследования** – модельное и алгоритмическое обеспечение процесса подготовки и принятия управленческих решений в высшем учебном учреждении.

**Методы исследования:** метод нечетких деревьев решений, онтологическое моделирование, теории принятия решений, программирование, разработка баз данных и информационных систем.

**Практическая значимость исследования.** Созданная ИСПП предоставляет лицу, принимающему решение, автоматизированный выбор оптимальной альтернативы согласно полученной базе знаний на основе построенных деревьев решений.

Основной результат исследования состоит в том, что разработанные на основе предложенной информационной базы алгоритмы позволяют оперативно и обоснованно проводить оценку студентов, анализ их успеваемости, научного потенциала и общей мотивации к обучению, а также принимать аргументированные управленческие решения, экономить время на процедуру оценки, получать информацию о взаимосвязях между уровнем компетенций и показателями деятельности студентов. Это предоставит возможность увидеть тенденции развития/деградации отдельно взятых студентов, а также общую динамику по подразделениям; определять дальнейшую индивидуальную траекторию для студентов согласно их текущей деятельности. Такая информационная поддержка позволит выпускникам лучше адаптироваться к дальнейшей деятельности, а способствует также увеличению качества образования. Повышение качества принятия решения приведет к уменьшению затрат на обучение студентов, ошибочно принятых в магистратуру или аспирантуру, и в итоге не закончивших обучение. Также, так как сократится нагрузка на учебно-вспомогательный персонал, возможно будет сократить расходы на выполнение функции сбора и обработки данных для принятия решений об отчислении, принятии для последующего обучения или на работу.

Основные результаты представлялись на конференциях, в том числе международных. По теме выпускной работы опубликовано 4 статьи.

## 1. Обзор литературы

Методы принятия управленческих решений делятся на три группы: неформальные (эвристические), коллективные и количественные [1-3].

Неформальные методы основаны на применении логических и творческих приемов мышления для определения лучших решений, сравнивая варианты и опираясь на опыт. Эвристические методы применяются на основе интуиции и аналитических способностях лиц, принимающих решения. Главным их преимуществом является оперативность, а недостатком отсутствие гарантии выбора лучшего варианта.

Коллективные методы предполагают участие сотрудников предприятия в принятии решений. Самыми распространенными методами такого типа являются интерактивные группы, номинальные группы, «мозговой штурм», метод Дельфы, а также японская система принятия решений.

Группа количественных методов принятия решений опирается на научно-практический подход для отыскания оптимального решения обработав (с использованием современных ЭВМ) большого объема информации и построения какой-либо математической модели.

В зависимости от математических функций, лежащих в основе моделей, выделяют: линейное моделирование; динамическое программирование, допускающее ввод новых переменных в процессе решения задач; вероятностные и статистические модели на базе теории массового обслуживания; теорию игр, позволяющую моделировать ситуации, в которых решения должны учитывать несовпадение интересов различных подразделений организации; имитационные модели, экспериментально реализующие решения, что позволит скорректировать исходные цели и уточнить требования к ним.

А также методы Data Mining, уже давно вышедшие за рамки традиционных статистических методов. Их основу составляют всевозможные методы классификации, моделирования и прогнозирования. Одно из важнейших назначений методов Data Mining состоит в наглядном представлении результатов вычислений, что позволяет использовать

инструментарий Data Mining людьми, не имеющих специальной математической подготовки, что значимо для данной задачи, т.к. системой будут пользоваться люди, не обладающие профильным образованием в данной области.

Знания, добываемые методами Data mining, принято представлять в виде моделей. **Модели представления знаний Data Mining**

- ассоциативные правила
- деревья решений
- кластеры
- метематические функции

Методам Data Mining включают:

1. эволюционное программирование и генетические алгоритмы
2. метод опорных векторов
3. искусственные нейронные сети
4. корреляционно-регрессионный анализ
5. неиерархические методы кластерного анализа, в том числе алгоритмы k-средних и k-медианы
6. методы поиска ассоциативных правил, в том числе алгоритм Apriori
7. иерархические методы кластерного анализа
8. методы ближайшего соседа и k-ближайшего соседа
9. байесовские сети
10. линейная регрессия
11. деревья решений, символьные правила
12. метод ограниченного перебора

Методы Data Mining определяются свойствами, важными при выборе метода анализа данных. В таблице 1.1 приведена сравнительная характеристика некоторых распространенных методов. Оценка каждой из характеристик проведена **в соответствии с** категориями: низкая, нейтральная, высокая, высокая и их промежуточными значениями. Как видно из таблицы ни один метод, при наличии своих преимуществ и недостатков, не способен обеспечить решение всех видов задач Data Mining [13, 17-19].

Таблица 1.1 – Сравнительная характеристика методов Data Mining

АЛГОРИТМ	ТОЧНОСТЬ	МАСШТАБИРУЕМОСТЬ	ИНТЕРПРЕТИРУЕМОСТЬ	ПРИГОДНОСТЬ К ИСП.
<i>линейная регрессия</i>	нейтральная	высокая	высокая / нейтральная	высокая
<i>нейронные сети</i>	высокая	низкая	низкая	низкая
<i>методы визуализации</i>	высокая	очень низкая	высокая	высокая
<i>деревья решений</i>	низкая	высокая	высокая	высокая / нейтральная
<i>нейронные сети</i>	высокая	нейтральная	низкая	высокая / нейтральная
<i>k-ближайшего соседа</i>	низкая	очень низкая	высокая / нейтральная	нейтральная

АЛГОРИТМ	ТРУДОЕМКОСТЬ	РАЗНОСТОРОННОСТЬ	БЫСТРОТА	ПОПУЛЯРНОСТЬ
<i>линейная регрессия</i>	нейтральная	нейтральная	высокая	низкая
<i>нейронные сети</i>	нейтральная	низкая	очень низкая	низкая
<i>методы визуализации</i>	очень высокая	низкая	чрезвычайно низкая	высокая / нейтральная
<i>деревья решений</i>	высокая	высокая	высокая / нейтральная	высокая / нейтральная
<i>нейронные сети</i>	низкая / нейтральная	нейтральная	низкая / нейтральная	нейтральная
<i>k-ближайшего соседа</i>	нейтральная / низкая	низкая	высокая	низкая

### Классификация методов

К логическим методам относятся нечеткие запросы и анализы; символьные правила; деревья решений; генетические алгоритмы [17-24].

Такие методы являются самыми интерпретируемыми, отображая полученные закономерности пользователя в достаточно прозрачном виде. Полученные правила могут включать непрерывные и дискретные переменные.

Методы кросс-табуляции: агенты, байесовские (доверительные) сети, кросс-табличная визуализация [16, 24, 36]. Последний метод не совсем отвечает свойству Data Mining по самостоятельному поиску закономерностей, однако предоставление информации в виде кросс-таблиц реализует основную задачу Data Mining – поиск шаблонов.

Методы на основе уравнений [16, 27]. Такие методы выражают выявленные закономерности в виде математических уравнений. Они способны обрабатывать исключительно числовые значения. Это ограничивает область

приложения методов данной группы, однако, они активно применяются для решения широкого спектра задач, особенно связанных с прогнозированием.

Методы Data Mining также можно распределить по задачам Data Mining на две группы. Первая группа методов направлена на решение задач сегментации (т.е. задачи классификации и кластеризации) и задач прогнозирования. Методы Data Mining из второй группы направлены на составление описаний и построение прогнозов.

Описательные методы ищут шаблоны или паттерны, которыми можно описать данные, интерпретируемые с точки зрения аналитика [16, 29-33]. К ним относят алгоритм k-средних, k-медианы, иерархические методы кластерного анализа, карты Кохонена, методы кросс-табличной визуализации.

Прогнозирующие методы используют значения входных переменных для предсказания/прогнозирования неизвестных (пропущенных) или будущих значений целевых переменных. Такие методы включают нейронные сети, деревья решений, линейную регрессию, методы ближайшего соседа и опорных векторов.

### **Деревья решений**

Деревья решений (decision trees) является одним из самых популярных методов для решения задач классификации и регрессии. Если целевая переменная описана дискретными значениями, то дерево решений строится для задачи классификации. В случае непрерывных значений решается задача численного прогнозирования [33, 29, 39, 56-59].

Дерево решений используется для отображений правил в иерархической, последовательной структуре. В ее узлах находятся ответы "Да" или "Нет" на ряд вопросов. Целью построения дерева решения в данном случае является определение значения категориальной зависимой переменной.

Задача классификации решается в два этапа: построение классификационной модели и ее использование. На этапе построения модели строится дерево классификации или создается набор неких правил. На этапе использования модели построенное дерево, или путь от его корня к одной из вершин, являющийся набором правил для конкретного объекта, используется для классификации новых объектов.

Внутренние узлы дерева являются атрибутами базы данных. Эти атрибуты называют прогнозирующими, или атрибутами расщепления (splitting attribute). Конечные узлы дерева, или листья, являются метками класса, определяющими значения зависимой переменной. Каждая ветвь дерева, идущая от внутреннего узла, отмечается предикатом расщепления. Он может относиться лишь к одному атрибуту расщепления данного узла. Суммарная информация об атрибутах и предикатах расщепления в узле называется критерием расщепления (splitting criterion) [33].

Для любой задачи может быть построено бесконечное множество деревьев решений, обладающих различной прогнозирующей точностью. Качество построенного дерева решения зависит от правильного выбора критерия расщепления. Над разработкой и усовершенствованием критериев работают многие исследователи. Метод деревьев решений часто называют "наивным" подходом [34, 43]. Но благодаря целому ряду преимуществ, данный метод является одним из наиболее популярных для решения задач классификации.

### **Преимущества деревьев решений**

Интуитивность деревьев решений. Результат работы деревьев решений, легко интерпретируется пользователем, в отличие, например, от нейронных сетей, являющихся "черными ящиками"[49-52].

Дерево решений позволяет понять и объяснить, почему конкретный объект относится к тому или иному классу. Они дают возможность отображать

правила из базы данных на естественном языке. Они позволяют создавать классификационные модели в тех областях, где аналитику достаточно сложно формализовать знания. Алгоритм конструирования дерева решений не требует от пользователя выбора входных атрибутов (независимых переменных). На вход алгоритма можно подавать все существующие атрибуты, алгоритм сам выберет наиболее значимые среди них, и только они будут использованы для построения дерева.

Точность моделей, созданных при помощи деревьев решений, сравнима с точностью прочих методов классификации, таких как статистические методы, нейронные сети и др [47-49].

Быстрый процесс обучения. Классификационные модели строятся значительно быстрее при помощи алгоритмов деревьев решений, чем, например, на обучение нейронных сетей. Большинство алгоритмов конструирования деревьев решений имеют возможность специальной обработки пропущенных значений. Деревья решений работают и с числовыми, и с категориальными типами данных. Деревья решений, в отличие от многих методов, строят непараметрические модели и способны обрабатывать категориальные значения. Таким образом, деревья решений способны решать такие задачи Data Mining, в которых отсутствует априорная информация о виде зависимости между исследуемыми данными.

Так как для решения исходной задачи, необходим метод, способный проводить и классификацию и прогнозирование, наилучшим выбором становится применение деревьев решений. Они также обладают рядом характеристик, выделяющих из среди прочих методов, такие как быстрота, и высокая масштабируемость, что является преимуществом, так как необходимо классифицировать большие объемы данных. А так же наглядность – очень важное свойство, потому что с программой будут работать пользователи, не имеющие специальных знаний и возможность объяснения решений на естественном языке крайне желательна.



## **Нечеткие множества**

Также в работе применен подход, использующий метод нечёткого вывода для решения задачи. В классической теории множеств принадлежность элементов множеству определяется бинарно: элемент либо принадлежит данному множеству, либо нет. Напротив, теория нечётких множеств разрешает градуированную оценку отношения принадлежности элементов множеству, описывая такое отношение при помощи функции принадлежности [50-54].

Нечеткий вывод осуществляется следующим образом: составляются правила и записываются в базу знаний интеллектуальной системы. Обработка этих данных осуществляется посредством процедур нечеткого логического вывода. Результатами работы системы являются нечеткое множество, полученное для заданного студента, и мера его сходства с возможными исходами, т. е. нечеткими множествами. В результате сортировки составляется рейтинг, который и позволит выделить лучших.

Преимуществом данного подхода является то, что формализация знаний с помощью правил позволяет учитывать различную важность критериев и самих правил для формирования требований. Для учета различной важности правил используются нормированные весовые коэффициенты, которые можно получить либо путем попарных сравнений, либо путем экспертного назначения весов. Применение нечеткой логики для решения данной задачи позволит формализовать знания экспертов [34, 65].

## **Нечеткие деревья решений**

Основная идея нечетких деревьев решений заключается в самом названии и состоит в сочетании достоинств нечеткой логики и деревьев решений. Здесь каждому атрибуту соответствуют несколько лингвистических термов и обозначаются степени принадлежности примеров к этим термам. Нечеткое дерево решений направлено на группировку степеней принадлежности примеров, а не их количества в конкретном узле. Безусловным

достоинством данного подхода является высокая точность классификации, достигаемая за счет сочетания достоинств нечеткой логики и деревьев решений. Процесс обучения происходит быстро, а результат прост для интерпретации. Однако для достаточной точности решения необходим репрезентативный набор обучающих примеров, в противном случае сгенерированное алгоритмом дерево решений будет слабо отражать действительность и, как следствие, выдавать ошибочные результаты [52-56, 63].

Прежде, чем применять описанные выше методы, необходимо создать базу знаний для разрабатываемой ЭС.

База знаний (БЗ) - это особая база данных, спроектированная для управления знаниями (метаданными). В ней хранится структурированная информация о некоторой области знаний для дальнейшего использования с определенной целью. Полноценные базы знаний содержат в себе фактическую информацию, а также и правила вывода, допускающие осуществление умозаключений о новых поступающих. Главная цель таких баз - помочь менее опытным людям найти уже существующее описание способа решения какой-либо проблемы [24, 32].

Иерархический способ представления набора объектов и их свойств и отношений в базе знаний называется онтологией. Онтология в информатике - это попытка всеобъемлющего и подробного отражения некоторой области знаний с помощью концептуальной схемы. Обычно такая схема состоит из структуры, содержащей данных все релевантные классы, их объектов связи и правила, принятые в этой области.

Анализ литературы по проблеме принятия решений, отбора и упорядочивания объектов, признаками которых являются элементы и компоненты компетентности, показал, что наиболее перспективный подход состоит в использовании нечетких моделей и алгоритмов нечеткой логики в сочетании с интеллектуальным анализом при помощи деревьев решений.

## 2. Постановка задачи

Большие объемы данных о студентах, поступающие к персоналу университета не позволяют провести тщательную их обработку и анализ для выявления паттернов, позволяющих принимать решения о дальнейшей перспективе их обучения в университете.

Комплексная информация о студентах позволит проводить качественный и количественный анализ их успеваемости, научного потенциала и общей мотивации к обучению. Это поможет увидеть тенденции развития/деградации отдельно взятых студентов, а также общую динамику по подразделениям; определять дальнейшую индивидуальную траекторию для студентов согласно их текущей деятельности (например, стоит ли данному обучающемуся поступать в магистратуру, т.к. он стремится заниматься наукой и теоретическими исследованиями, или же он более подходит для профессиональной деятельности на предприятиях города). Такая информационная поддержка позволит выпускникам лучше адаптироваться к дальнейшей деятельности (и не терять еще два года в магистратуре), а также увеличить качество образования (за счет улучшенного отбора).

Повышение качества принятия решения приведет к уменьшению затрат на обучение студентов, ошибочно принятых в магистратуру или аспирантуру, и в итоге не закончивших обучение. Также, так как сократится нагрузка на учебно-вспомогательный персонал, возможно будет сократить расходы на выполнение функции сбора и обработки данных для принятия решений об отчислении, принятии для последующего обучения или на работу.

Поэтому необходимо разработать информационную систему поддержки принятия решений в виде веб-приложения с математическим аппаратом обработки данных, которую можно использовать для анализа и обработки данных об учебной, научной, творческой, спортивной и общественной деятельности обучающихся университета, вместе с оценкой их личностных и

психологических характеристик с целью нахождения закономерностей, на основании которых возможно принятие решений о перспективности и успешности студентов.

Создание подобной ИСППР следует вести в двух направлениях: создание математического аппарата обработки, анализа данных для принятия решений и проектирование приложения, которое будет обрабатывать запросы пользователя и выдавать соответствующие им результаты.

К первой части можно прописать следующие задачи и требования:

- необходимо, используя математические методы обработки и анализа данных
  - определить интегральный показатель, отвечающий всем характеристикам студента для составления рейтинга студентов по группе/кафедре/направлению/институту.
  - определить каждого студента в один из классов для получения рекомендаций к дальнейшей его деятельности и поддержке в принятии решений о продолжении или остановке обучения.
- используемые методы должны быть устойчивы к отсутствию некоторых данных для части студентов
- используемые методы должны быстро обрабатывать большие объемы информации
- полученные результаты должны быть легко интерпретируемы и понятны даже человеку, не имеющему специального образования
- формирование правил в БЗ

Все вышеописанное приводит к постановке задач классификации и регрессии.

Основная цель классификации состоит в построении правила, в соответствии с которым устанавливается, к какому из классов объектов может быть отнесен классифицируемый объект. При этом под классом понимается некоторая совокупность объектов, обладающих близкими свойствами.

Пусть дано множество  $M$  объектов  $\omega$ ; на этом множестве существует разбиение на конечное число подмножеств (классов)  $\Omega_i$ ,  $i=1, \dots, m$ ,  $M = \bigcap_{i=1}^m \Omega_i$ . Разбиение  $M$  определено не полностью. Задана лишь некоторая информация  $I_0$  о классах  $\Omega_i$ . Объекты  $\omega$  задаются значениями некоторых признаков  $x_j$ ,  $j = 1, \dots, p$ . Совокупность значений признаков  $x_j$  определяет описание  $I(\omega)$  объекта  $\omega$ . Каждый из признаков может принимать значения из различных множеств допустимых значений, например, из следующих:  $(0,1)$  – признак отсутствует или присутствует, соответственно;  $(0,1, \Delta)$ ,  $\Delta$  – информация о признаке отсутствует;  $(0,1, \dots, d-1)$  – степень выраженности признака имеет различные градации,  $d > 2$ ;  $(a_1, \dots, a_d)$  – признак имеет конечное число значений,  $d > 2$ ; значениями признака  $x_j$  задаются функции некоторого класса; значениями признака  $x_j$  являются функции распределения некоторой случайной величины.

Задача классификации состоит в том, чтобы для данного объекта  $\omega$  и набора классов  $\Omega_i$ ,  $i=1, \dots, m$ , по обучающей информации  $I_0(\Omega_1, \dots, \Omega_m)$  о классах и описанию  $I(\omega)$  вычислить значения предикатов  $P_i(\omega) = " \omega \in \Omega_i "$ ,  $i=1, \dots, m$ . Информация о вхождении объекта  $\omega$  в класс  $i$  кодируется символами "1" ( $\omega \in \Omega_i$ ), "0" ( $\omega \notin \Omega_i$ ),  $\Delta$  – неизвестно, принадлежит ли  $\omega$  классу  $\Omega_i$  или нет.

Любая прикладная система классификации предназначена для классификации объектов и явлений конкретной предметной области и требует построения математической модели.

Задача регрессии ставится следующим образом:

Пусть дана совокупность результатов наблюдений множества  $M$  объектов  $\omega$ . В этой совокупности существует одна характеристика, для которой необходимо установить функциональную зависимость с параметрами объекта и среды, представленными значениями остальных характеристик. Требуется: установить количественную взаимосвязь между показателем и факторами. В этом случае задача регрессионного анализа ставится как задача выявления следующей функциональной зависимости  $f(x, K_n) = K_1 \cdot f_1(x) + K_2 \cdot f_2(x) + \dots + K_N \cdot f_N(x)$ , которая наилучшим образом описывает существующие экспериментальные данные.

Требования к системной части:

- комплекс задач, решаемых при помощи проектируемой системы, должен обеспечивать удобную, качественную и быструю работу сотрудника университета, ответственного за принятие описанных выше решений.
- создание информационной базы для автоматизированного процесса классификации и регрессии;
- загрузка данных в базу;
- открытость структуры данных, то есть возможность их редактирования;
- формирование рекомендаций для принятия решений;
- приложение должно быть доступно удаленно, т.е. обладать веб-интерфейсом

Входными данными для реализации задач системы являются:

- Данные о студентах;
- Параметры классификации;

Выходной информацией будет:

- Результаты классификации и регрессии.
- Отчет по результатам классификации;
- Интегральные показатели
- Отчет по данным

### **3. Объект и методы исследования**

#### **3.1. Объект исследования**

Стремительное развитие информационных технологий, в частности, прогресс в методах сбора, хранения и обработки данных позволил многим организациям собирать огромные массивы данных, которые необходимо анализировать. Объемы этих данных настолько велики, что возможностей экспертов уже не хватает, что породило спрос на методы автоматического исследования (анализа) данных, который с каждым годом постоянно увеличивается. Поэтому необходимо создание системы поддержки принятия управленческих решений, основанных на перечисленных выше данных.

Объектом исследования является учебная деятельность студента. К ней, безусловно, можно отнести успеваемость обучающегося и различные учебные достижения и достижения в науке. Также стоит рассмотреть для анализа и прочие факторы, например, личностные и психологические характеристики человека, способные косвенно повлиять на успешность в процессе обучения в университете.

В данной главе описаны методы и решения, выбранные для выполнения поставленных задач. В проекте будут использоваться модификация деревьев решения при помощи нечеткой логики. С их помощью производится регрессия и классификация объектов. Также в основе системы находится база данных на oracle, выбран паттерн проектирования MVC, его реализация производится при помощи Арх.

#### **3.2. Методы исследования**

##### **Построение дерева решений**

Рассматриваемая задача классификации относится к стратегии обучения с учителем, когда все объекты тренировочного набора данных заранее отнесены к одному из заранее определенных классов. Алгоритмы конструирования деревьев решений включают шаги "построения" дерева (tree building) и его "сокращения" (tree pruning). В процессе построения дерева

решаются вопросы выбора критерия расщепления и останова обучения. В процессе сокращения дерева решается вопрос отсечения ветвей.

### Критерий расщепления

Процесс построения дерева является нисходящим, т.е. проходит сверху вниз. В его ходе алгоритм ищет такой критерий расщепления, чтобы разбить множество на подмножества, согласованные с данным узлом. Каждому узлу соответствует определенный атрибут: он должен разбивать исходное множество данных так, чтобы объекты таких подмножеств были представителями одного класса или же близки к подобному разбиению.

Существуют различные критерии расщепления. Во многих алгоритмах ищется мера информативности подпространств атрибутов. Еще один критерий расщепления предложен Брейманом (Breiman) и др. для реализации алгоритма CART и называется индексом Gini. Здесь атрибут выбирается на основании расстояний между распределениями классов.

Если дано множество примеров  $M$ , с  $m$  классами, индекс Gini определяется следующим образом:

$$gini(T) = 1 - \sum_{j=1}^m p_j^2$$

где  $T$  - текущий узел,  $p_j$  - вероятность класса  $j$  в узле  $T$ ,  $m$  - количество классов.

Величина дерева влияет на его способность к классификации. Слишком маленькое дерево не производит достаточного разбиения, обобщая объекты с различными свойствами; а чрезмерно большое – определяет в полученные классы слишком малое количество объектов, уменьшается способность к обобщению объектов.

Для построения деревьев "подходящих размеров", чтобы они были просты, но одновременно достаточно сложны и учитывали информацию из



исследуемого набора данных [39], применяется две возможные стратегии. Первая регулирует рост дерева при достижении определенного размера согласно параметрам пользователя. Во второй стратегии используется ряд процедур по определению оптимальной величины дерева: отсечения ветвей у уже построенного дерева и использование правил останова обучения.

Согласно правилу останова решается, будет ли данный узел внутренним, или он является листом, и процесс разбиения на нем останавливается. При ранней остановке (prepruning) определяется целесообразность разбиения узла. Другой вариант – ограничение глубины дерева. Еще один способ – ограничение минимального количества примеров для дальнейшего разбиения [35]. Слишком ветвистое дерево сокращают отсечением (pruning) некоторых ветвей.

Качество классификационной модели определяется его точностью распознавания и ошибкой. Точность распознавания это отношение правильно классифицированных объектов к их общему количеству, а ошибка – как отношение неправильно классифицированных объектов к общему количеству. Отсечение ветвей проводят если при этом не возрастает ошибка модели. Каждый путь от корня дерева до его вершины или листа дает одно правило. Условиями правила являются проверки на внутренних узлах дерева.

На данный момент создано немало алгоритмов, реализующих деревья решений: CART, C4.5, CHAID, CN2, NewId, ITrule и т.д. В данной работе все деревья строятся на базе измененного алгоритма CART, традиционная версия которого представлена ниже.

CART (Classification and Regression Tree) – алгоритм, предназначенный для решения задач классификации и регрессии, был разработан в 1974-1984 годах. Алгоритм CART способен оперировать числовыми и категориальными атрибутами. Если атрибут числовой, то во внутреннем узле формируется правило вида  $x_i \leq c$ . В случае, когда атрибут является категориальной переменной, то во внутреннем узле определяется правило  $x_i \in V(x_i)$ , где  $V(x_i)$  -

некоторое непустое подмножество множества значений переменной  $x_i$  из обучающего набора данных.

Алгоритм CART принципиально отличается своим механизмом отсечения ветвей от прочих аналогичных алгоритмов. Метод заключается в проектировании последовательности уменьшающихся деревьев, из которых рассматриваются только лучшие. Перекрестная проверка (V-fold cross-validation) алгоритма CART отвечает за выбор окончательного дерева из уже созданных. Несмотря на все преимущества, чтобы построить качественную модель, необходимо понимать природу взаимосвязи между зависимыми и независимыми переменными и подготовить достаточный набор данных.

### **Нечеткий логический вывод**

При использовании нечеткого логического вывода необходимо построение функций принадлежности  $\mu_A(x)$  нечеткого множества  $A$  с элементами  $x$ , нечетко обладающими определяющим множеством свойством  $R$ .

Существуют прямые и косвенные методы построения функций принадлежности. Прямые методы, такие как параметрический, интервальный, методы относительных частот, целесообразно использовать для свойств, признаков и атрибутов, которые возможно измерить. При этом достаточно зафиксировать вид функции принадлежности и некоторые точки, по которым возможно построить непрерывную функцию на основе ее уже определенного дискретного отображения. В случае неизмеримых параметров прибегают к косвенным методам (наиболее известен метод парных сравнений). В свою очередь, в зависимости от числа привлеченных к опросу экспертов как прямые, так и косвенные методы делятся на одиночные и групповые.

Для отображения переменной при помощи нечеткой логики используются лингвистические переменные.

Лингвистической переменной (ЛП) называется набор  $(\beta, T, X, G, M)$ ,

где  $\beta$  – наименование лингвистической переменной;

$T$  – терм-множество значений лингвистической переменной, являющееся набором наименований нечетких переменных, с областью определения, равной  $X$ .

$G$  – синтаксическая процедура для произведения операций над элементами терм-множества  $T$ , например, позволяющая генерировать новые термы. Множество  $G(T)$ , где  $G\{T\}$  – множество сгенерированных термов или расширенный термом;

$M$  – семантическая процедура, формирующая соответствующее нечеткое множество при помощи процедур из  $G$ .

Механизм (алгоритм) нечеткого логического вывода - это процесс получения нечетких заключений на основе нечетких условий или предпосылок. С его помощью создается база правил нечетких высказываний. Справедливо выражение:

$$\langle \alpha \text{ есть } \alpha' \text{ и } \beta \text{ есть } \beta' \rangle \Rightarrow \langle (\alpha, \beta) \text{ есть } (\alpha' \cap \beta') \rangle.$$

Здесь  $\Rightarrow$  - знак подстановки,  $\alpha' \cap \beta'$  - значение лингвистической переменной  $(\alpha, \beta)$ , соответствующее исходному высказыванию  $\langle \alpha \text{ есть } \alpha' \text{ и } \beta \text{ есть } \beta' \rangle$ , которому на  $X \times Y$  ставится в соответствие нечеткое множество  $\hat{A} \cap \hat{B}$  функцией принадлежности

$$\mu_{\hat{A} \cap \hat{B}}(x, y) = \mu_{\hat{A}}(x, y) \wedge \mu_{\hat{B}}(x, y) = \mu_A(x) \wedge \mu_B(y).$$

Логический вывод осуществляется поэтапно.

1) Фаззификация (введение нечеткости) отвечает за определение соответствия между значением входной переменной и значением соответствующей ее терму функции принадлежности. Фаззификация окончена, когда найдены степени истинности  $\mu(a)$  всех элементарных логических высказываний вида « $\beta$  ЕСТЬ  $\alpha$ », входящих в антецеденты нечетких продукционных правил, где  $\alpha$  - некоторый терм с известной функцией

принадлежности  $\mu(x)$ ,  $\alpha$  - четкое численное значение, принадлежащее универсуму лингвистической переменной  $\beta$ .

2) Агрегирование – это процедура, использующая полученные на этапе фаззификации значения функций принадлежности термов лингвистических переменных, для определения степени истинности условий каждого правила в системе нечеткого логического вывода. В случае простого нечеткого высказывания, степень его истинности соответствует значению функции принадлежности соответствующего термина лингвистической переменной. Если же высказывание составное, то его степень истинности определяется из уже определенных значений истинности его элементарных высказываний.

3) Активизация отвечает за формирование функций принадлежности  $m(y)$  каждого из продукционных правил, которые находятся при помощи одного из методов нечеткой композиции:

- min-активизация -  $\mu(y) = \min\{c; \mu(x)\}$ ;
- prod-активизация -  $\mu(y) = c\mu(x)$ ;
- average-активизация -  $\mu(y) = 0,5(c + \mu(x))$ ;

Где  $c$  - степень истинности нечетких высказываний, образующих антецедент нечеткого продукционного правила.

4) Аккумуляция – это процесс определения функции принадлежности выходной лингвистической переменной путем объединения нечетких множеств всех подзаключений нечеткой базы правил (проводится как правило max-объединением  $\forall x \in X \mu_{A \cup B}(x) = \max\{\mu_A(x); \mu_B(x)\}$  ) относительно соответствующей лингвистической переменной.

5) В процессе дефаззификации функции принадлежности выходной лингвистической переменной  $\mu(y)$  ставится в соответствие её числовое значение  $y$ . Для этого применяется один из следующих методов:

- метод центра тяжести заключается в расчете центраоида площади:

$$\tilde{y} = \frac{\int_{y_{min}}^{y_{max}} y \mu(y) dy}{\int_{y_{min}}^{y_{max}} \mu(y) dy}, \text{ где } [y_{min}; y_{max}]$$

где  $[y_{min}; y_{max}]$ - носитель нечеткого множества выходной лингвистической переменной;

метод центра площади заключается в расчете абсциссы  $y$ , делящей площадь, ограниченную кривой функции принадлежности  $\mu(x)$ , так называемой биссектрисы площади

$$\int_{y_{min}}^{\tilde{y}} \mu(y) dy = \int_{\tilde{y}}^{y_{max}} \mu(y) dy ;$$

метод левого модального значения  $\tilde{y} = y_{min}$  ;

метод правого модального значения  $\tilde{y} = y_{max}$  .

Графическое представление нечеткого логического вывода представлено на рис. 3.1.

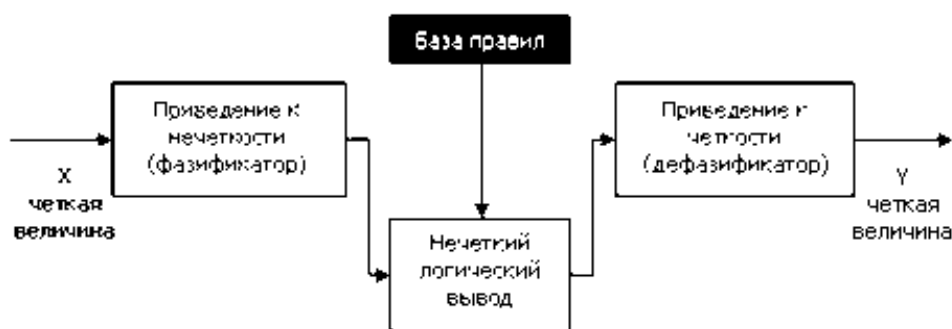


Рис. 3.1. Система нечеткого логического вывода

### Нечеткие деревья решений

Основная идея нечетких деревьев решений заключается в самом названии и состоит в сочетании достоинств нечеткой логики и деревьев решений. Здесь каждому атрибуту соответствуют несколько лингвистических термов и обозначаются степени принадлежности примеров к этим термам.

Нечеткое дерево решений направлено на группировку степеней принадлежности примеров, а не их количества в конкретном узле.

Вычисляется коэффициент как соотношение примеров узла  $N$  для целевого значения  $i$ , вычисляемый как:

$$P_i^N = \sum_{S^N} \min(\mu_N(D_j), \mu_i(D_j)),$$

где  $\mu_N(D_j)$  – степень принадлежности примера  $D_j$  к узлу  $N$ ,  $\mu_i(D_j)$  – степень принадлежности примера относительно целевого значения  $i$ ,  $S^N$  – множество всех примеров узла  $N$ . Затем находим коэффициент, обозначающий общие характеристики примеров узла  $N$ . В стандартном алгоритме дерева решений определяется отношение числа примеров, принадлежащих конкретному атрибуту, к общему числу примеров. Для нечетких деревьев используется отношение, для расчета которого учитывается степень принадлежности.

Выражение даёт оценку среднего количества информации для определения класса объекта из множества  $P^N$ .

$$E(S^N) = - \sum_i P_i^N / P^N \times \log_2 P_i^N / P^N.$$

На следующем шаге построения нечеткого дерева решений алгоритм вычисляет энтропию для разбиения по атрибуту  $A$  со значениями  $a_j$ :

$$E(S^N, A) = \sum_j P^N / P_i^N \times E(S^{N|j}),$$

, где узел  $N|j$  – дочерний для узла  $N$ .

Алгоритм выбирает атрибут с максимальным приростом информации.

$$G(S^N, A) = E(S^N) - E(S^N, A),$$

$$A^x = \operatorname{argmax}_A G(S, A).$$

Узел  $N$  разбивается на несколько подузлов  $N|j$ . Степень принадлежности примера  $D_k$  узла  $N|j$  вычисляется пошагово из узла  $N$  как

$$\mu_{N|j}(e_k) = \min(\mu_{N|j}(D_k), \mu_{N|j}(D_k, a_j))$$

где  $\mu_i(D_k, a_j)$  показывает степень принадлежности  $D_k$  к атрибуту  $a_j$ . Подузел  $N|j$  удаляется, если все примеры в нем имеют степень принадлежности, равную нулю. Алгоритм повторяется до тех пор, пока все примеры узла не будут классифицированы либо пока не будут использованы для разбиения все атрибуты. Принадлежность к целевому классу для новой записи находится по формуле,

$$\delta_j = \frac{\sum_i \sum_k P_k^i * \mu_i(D_j) * \chi_k}{\sum_i (\mu_i(D_j) * \sum_k P_k^i)}$$

где  $P_k^i$  – коэффициент соотношения примеров листа дерева  $l$  для значения целевого класса  $k$ ,  $\mu_i(D_j)$  – степень принадлежности примера к узлу  $l$ ,  $\chi_k$  – принадлежность значения целевого класса  $k$  к положительному значению исхода классификации.

К достоинствам нечетких деревьев можно отнести высокую точность классификации, полученную с внедрением нечеткой логики; быстрый процесс обучения, и простота результата для интерпретации, являющиеся неотъемлемым атрибутом деревьев решений. Способность алгоритма выдавать для нового объекта степень принадлежности к классу позволяет управлять порогом для классификации.

Однако для достижения необходимой точности необходим репрезентативный набор обучающих примеров, иначе получившееся дерево решений не будет способно корректно отражать действительность и, следовательно, выдавать ошибочные результаты.

## MVC

Model-view-controller (MVC, «модель-представление-контроллер», «модель-вид-контроллер») — схема использования нескольких шаблонов

проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области.

Модель (англ. Model) обеспечивает работу с знаниями. Предоставляет данные и методы оперирования этими данными, обрабатывает запросы, и согласно им изменяет своё состояние. Но модель не владеет информацией об отображении хранящихся в ней знаний.

Представление (англ. View) отвечает за визуализацию информации. Обычно представление является формой или окном с графическими элементами для отображения реакции системы на запросы пользователя.

Контроллер (англ. Controller) организует взаимодействие пользователя с системой: обрабатывает и проверяет данные, введенные пользователем, и управляет моделью и представлением для обеспечения необходимой реакции.

Главная идея применения этой концепции состоит в разделении бизнес-логики и её визуализации. Такое разделение приводит к повышению возможности повторного использования или когда пользователю необходимо получать одни и те же данные, но в различных контекстах или с различных точек зрения.

Структурная схема системы, построенной по его правилам, представлена на рис. 3.2.



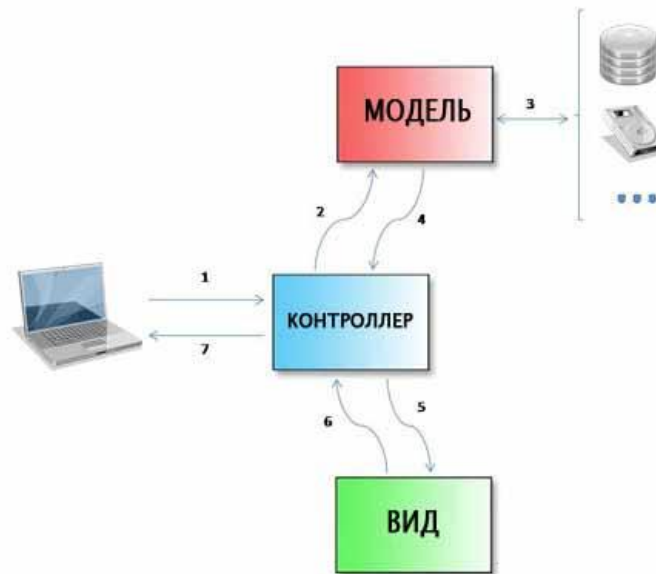


Рис. 3.2. Структурная схема информационных потоков в системе, организованной согласно MVC

Контроллер получает запрос пользователя [1] (запрос HTTP GET или POST). Можно организовать центральный контроллер, который получает запрос и вызывает периферийный контроллер.

Контроллер проверяет запрос и параметры, а затем вызывает модель, производит запрос [2].

Модель получает данные из базы (или из другого источника информации) [3], применяет фильтры и логику, отраженную в модели, а затем возвращает запрошенные данные [4].

Контроллер использует подходящий вид [5] для представления данных пользователю [6-7]. Если запрос приходит с мобильного телефона, используется вид для мобильного телефона; если пользователь использует определенное оформление интерфейса, то выбирается соответствующий вид, и так далее.

### **3.3.Выбор программного обеспечения**

#### **Построение онтологии**

Первая часть данной работы – создание онтологии для проектирования базы знаний профессий и их характеристик – будет решаться в программе для создания онтологий.

В последнее время количество общедоступных редакторов онтологий перевалило за 100. Но редко можно встретить универсальное и в то же время полезное средство. Рассмотрим существующие программы построения онтологий.

#### **Ontolingua**

Кроме собственно редактора онтологий, эта система содержит:

- сетевой компонент Webster, предназначенный для определения концептов;
- сервер, обеспечивающий доступ к онтологиям Ontolingua по протоколу OKBC (Open Knowledge Base Connectivity);
- Chimaera – инструментарий для анализа и объединения онтологий.

#### **Protege**

Это свободно распространяемая Java-программа, предназначенная для построения (создания, редактирования и просмотра) онтологий той или иной прикладной области. Она включает редактор онтологий, позволяющий проектировать онтологии, разворачивая иерархическую структуру абстрактных и конкретных классов и слотов. На основе сформированной онтологии Protege позволяет генерировать формы получения знаний для введения экземпляров классов и подклассов.

Данный инструмент поддерживает использование языка OWL и позволяет генерировать HTML-документы, отображающие структуру онтологий. Поскольку он использует фреймовую модель представления знаний ОКВС, это позволяет адаптировать его и для редактирования моделей предметных областей, представленных не в OWL, а в других форматах ( UML, XML, SHOE, DAML+OIL, RDF / RDFS и т.п.). Подробнее об этом редакторе будет рассказано далее в этой лекции.

## **DOE**

DOE (Differential Ontology Editor) – простой редактор, который позволяет пользователю создавать онтологии. Процесс спецификации онтологии состоит из трех этапов.

- На первом этапе пользователь строит таксономию понятий и отношений, явным образом очерчивая позицию каждого элемента (понятие) в иерархии. Затем пользователь указывает, в чем специфика понятия относительно его «родителя», и в чем это понятие подобно или отлично от его «братьев». Пользователь может также прибавить синонимы и энциклопедическое определение на нескольких языках для всех понятий.
- На втором этапе две таксономии рассматриваются с разных точек зрения. Пользователь может расширить их новыми объектами или добавить ограничения на области отношений.
- На третьем этапе онтология может быть переведена на язык представления знаний.

## **OntoEdit**

OntoEdit – инструментальное средство, обеспечивающее просмотр, проверку и модификацию онтологий. Оно поддерживает языки представления онтологии OIL и RDFS, а также внутренний язык представления знаний OXML,

основанный на XML. Как и Protege, это автономное Java-приложение, но его коды закрыты. Свободно распространяемая версия OntoEdit Free ограничена 50 концептами, 50 отношениями и 50 экземплярами.

### **OilEd**

OilEd – автономный графический редактор онтологий, разработанный в рамках проекта On-To-Knowledge. Он свободно распространяется по общедоступной лицензии GPL. Инструмент использует для представления онтологий язык OIL. В OilEd отсутствует поддержка экземпляров классов.

### **WebOnto**

WebOnto представляет собой Java-апплет и разработан для просмотра, создания и редактирования онтологий. Для моделирования онтологий он использует язык OCML (Operational Conceptual Modeling Language). Пользователь может создавать различные структуры, в том числе классы со множественным наследованием. Инструмент имеет ряд полезных особенностей: просмотр отношений, классов и правил, возможна совместная работа над онтологией нескольких пользователей.

### **ODE, WebODE**

ODE (Ontological Design Environment) взаимодействует с пользователями на концептуальном уровне, обеспечивает их набором таблиц для заполнения (концептов, атрибутов, отношений) и автоматически генерирует код на языках LOOM, Ontolingua и F-Logic. Данный инструмент получил свое развитие в редакторе онтологий WebODE, который интегрирует все сервисы ODE в единую архитектуру, сохраняя свои онтологии в реляционной БД.

Ниже приведена таблица, описывающая основные характеристики наиболее популярных редакторов онтологий.

**Таблица 3.1 – Сравнительная характеристика программных продуктов для построения онтологий**

	<b>OilEd</b>	<b>OntoEdit</b>	<b>Ontolingua</b>	<b>OntoSaurus</b>	<b>Protégé</b>	<b>WebODE</b>	<b>WebOnto</b>
<i>Общая информация</i>							
Разработчик	IMG, University of Manchester	Ontoprize	KSL, Stanford University	ISI, University of Southern California	SMI, Stanford University	Ontology Group, Polytechnic University of Madrid	KMI, Open University
Версия	3.5.5 Oct2003	2.6.6 Mar2004	1.0.650 Oct2002	1.9 Mar2002	2.1.1 June2004	2.1 Mar2003	2.3 May2001
Доступность	Открытый кол	Свободная лицензия	Свободный доступ	Открытый код, свободный	Открытый код	Свободный доступ	Свободный доступ
Поддержка методологией	-	On-To-Knowledge	-	-	-	METH-ONTOLOGY	-
<i>Архитектура программного обеспечения</i>							
Архитектура приложения	3-х уровневая	3-х уровневая	Клиент/сервер	Клиент/сервер	3-х уровневая	n-уровневая	Клиент/сервер
Расширяемость		Плагины			Плагины	Сервер приложения	
Хранение онтологий	файлы	файлы	файлы	файлы	файлы, СУБД	СУБД	файлы
Язык ПО	Java	Java	Lisp	Lisp	Java	Java	Java+ Lisp
<i>Модель знания</i>							
Формализм	DL	Фреймы + FOL	Фреймы + FOL	DL	Фреймы + FOL	Фреймы + FOL	Фреймы + FOL
Основной язык представления знания	DAML+OIL	OXML	Ontolingua	LOOM	OKBC	-	OCML
Формальный язык аксиом	-	FLogic	KIF	LOOM	PAL	WAB	OCML
<i>Редакторы онтологий</i>							
Интерфейс пользователя	Локальное приложение	Локальное приложение	HTML	HTML	Локальное приложение	HTML и апплеты	Апплеты
Графическое редактирование таксономии концептов	-	+	-	-	+	+	+
Редактор формальных аксиом	+	-	-	-	+	+	-
Совместная разработка онтологий	-	+	+	+	-	+	+
Машина вывода	FaCT (встроенная) RACER DIG	OntoBroker	JTP	классификатор LOOM	PAL (встр.) Jess, FaCT, Prolog, FLORA Algermon	Prolog (встроенная) Jess	Система представления знаний OCML
Проверка непротиворечивости	+	+	-	+	+	+	+
<i>Интероперабельность</i>							
С другими инструментами	-	OntoAnnotate OntoMat Semantic-Miner	Chimaera OKBC	OKBC	Prompt OKBC ArgoUML	ODE-KM ODE-SeW ODE-SWS ODE-Annotate Protégé	MnM
Импорт	RDF(S) OIL DAML+OIL SHIQ	OXML RDF(S) DAML+OIL FLogic	Ontolingua KIF CML IDL	LOOM PowerLOOM Stella IDL	XML RDF(S) XML Schema XMI	XML RDF(S) DAML+OIL OWL CARIN	OXML RDF(S)
Экспорт	RDF(S) OIL DAML+OIL OWL SHIQ DIG	OXML RDF(S) DAML+OIL FLogic	Ontolingua KIF LOOM CLIPS CML Epikit Prolog IDL	LOOM PowerLOOM KIF Ontolingua Stella IDL C++	XML RDF(S) XML Schema FLogic CLIPS Java XMI	XML RDF(S) OIL DAML+OIL OWL CARIN FLogic Prolog, Jess, Java	OXML Ontolingua RDF(S)

Для выполнения данной работы будет использоваться программа Protege версии 4.3, т.к. она обладает рядом преимуществ, таких как поддержка многих языков создания онтологий, возможность осуществления программной надстройки, развитый функционал по работе с данными. Более подробное описание представлено ниже.

**Protege** – локальная Java-программа, разработанная группой медицинской информатики Стенфордского университета. • Protege включает редактор онтологий. Структура онтологии сделана аналогично иерархической структуре каталога. На основе сформированной онтологии Protégé может генерировать формы получения знаний для введения экземпляров классов и подклассов.

С момента его создания Protege многие годы использовался экспертами в основном для концептуального моделирования в области медицины. В последнее время его стали применять в других предметных областях - в частности, при создании онтологий для Semantic Web.

### **Построение деревьев решений**

Существует множество реализаций алгоритмов построения деревьев регрессии и классификации. Каждый из них обладает своими преимуществами и недостатками. В идеале такое ПО должно реализовывать быстрый поиск закономерностей, высокую масштабируемость и точность классификации или регрессии, обработку пропущенных значений, иметь удобный интерфейс и управление параметрами конструирования деревьев, обеспечивать реализацию нечеткого логического вывода и так далее. И если большинство перечисленных характеристик в той или иной степени уже внедрены, то использование нечеткой логики в качестве модернизации алгоритмов для повышения их точности и доступности результатов происходит далеко не во всех доступных приложениях.

В итоге для реализации классификации при помощи нечетких деревьев решений была выбрана программа FID – Fuzzy Inference Decision, основанная на нечеткой модификации алгоритма CART. С ее помощью, возможно проводить классификацию объектов по множеству их свойств. Такие свойства можно задавать как в четком, так и в нечетком и номинальном виде. В качестве функций принадлежности используются трапециевидные функции, отвечающие за отображения термов лингвистических переменных. Также возможно регулировать параметры алгоритма конструирования дерева, такие как критерии расщепления и останова, параметры алгоритма кросс-валидации и др. Результат выдается в виде построенного дерева решения с степенью принадлежности каждого листа к каждому классу. Также выдаются результаты для оценки алгоритма, такие как % правильной классификации для примеров из обучающей и тестовой выборки, параметры алгоритма на момент останова. Результат сохраняется в текстовом файле и файле со специальным расширением, который можно использовать для дальнейшей классификации объектов из новой выборки.

Для решения задачи регрессии используется алгоритм, спроектированный для математического пакета Matlab и использующий его встроенные модули для организации конструирования деревьев и применения логического вывода. В данном алгоритме так же допускается настройка параметров алгоритма и выбора функций принадлежности. Недостатком его является невозможность применения для данных с пропущенными значениями. Результат сохраняется в структуры в рабочей области пакета и содержит построенное дерево и правила нечеткого вывода.

### **Выбор СУБД**

В идеальном случае СУБД должна обеспечивать реализацию всех возлагаемых на БД задач и укладываться в ограничения, накладываемые техническими средствами. Для окончательного выбора СУБД необходимо

проанализировать свойства СУБД, ОС, в которой может работать СУБД, возможные структуры данных, типы запросов, возможность создания метаданных, интеграции данных, средств экспорта-импорта, возможность использования ранее созданных приложений. Сведем наиболее используемые СУБД и их основные характеристики в одну таблицу.

**Таблица 3.2 – Характеристики СУБД**

Компании	СУБД	Краткая характеристика
<i>Зарубежные продукты</i>		
Borland	InterBase	SQL-совместимая реляционная, ОС - Windows, Linux и Solaris.
IBM	DB2 Universal Database, Informix	Мультимедийная, Web, ОС - Unix, Linux и Windows, аппаратные платформы - zSeries, iSeries, VSE и VM. СУБД для систем масштаба предприятия и рабочей группы, обеспечивает работу с очень крупными БД в условиях дефицита ресурсов. Используемые языки - Java, SQL 2000
InterSystems	Cache <sup>7</sup>	Постреляционная СУБД, включающая сервер многомерных данных и средство обработки запросов на языке SQL. ОС - Unix, Mac OS, Linux, Windows (32- и 64-разрядные версии).
Microsoft	SQL Server	Реляционная СУБД для управления данными в масштабе предприятия, поддерживает технологии XML и Интернет, обладает встроенным средством анализа и извлечения данных, интегрированным с MS Office, ОС - Windows. Используемый язык Transact-SQL, XML
Oracle	Oracle	СУБД для масштабной обработки транзакций (OLTP), хранилищ данных с высокой интенсивностью потока запросов и ресурсоемких Интернет-приложений. ОС - Unix, Windows и Linux. Последняя версия поддерживает Grid-вычисления. Используемые языки Java, Delphi PL/SQL, XML
Microsoft	dBASE III Plus	ОС Windows. Создаваемые ею файлы импортируются СУБД FoxPro, Paradox, MS Access, а также пакетами прикладных программ MS Excel, Surfer, Grapher и др.
<i>Отечественные продукты</i>		
Рэлэкс	Линтер	Реляционная СУБД, имеющая сертификат Гостехкомиссии при Президенте РФ на соответствие 2 классу защиты информации от несанкционированного доступа, ОС - Unix, Linux, QNX, VAX/VMS, OpenVMS, DOS, Windows, NetWare, OS/2
ВНИИНС	Паллада	Объектно-ориентированная СУБД, предназначенная для АСУ вооруженных сил, функционирует в среде ОС МСВС и ОЛИВИЯ



Компании	СУБД	Краткая характеристика
<i>СУБД с открытым исходным кодом</i>		
MySQL AB	MySQL	Компактная, быстродействующая реляционная СУБД для малых и средних предприятий, ОС - Linux, Mac OS X, Unix и Windows
Сообщество PostgreSQL	PostgreSQL	Реляционная СУБД, имеет многие возможности, которые реализованы в крупных коммерческих продуктах, ОС - Unix, Windows и NetWare

Так как приложение ориентировано на интернет технологии, для разработки базы данных выбрана СУБД oracle. Она обладает еще рядом преимуществ, таких как высочайшая надежность, возможность разбиения крупных баз данных на разделы (large-database partition), что дает возможность эффективно управлять гигантскими гигабайтными базами; наличие универсальных средств защиты информации; эффективные методы максимального повышения скорости обработки запросов; индексация по битовому отображению; свободные таблицы (в других СУБД все таблицы заполняются сразу при создании); распараллеливание операций в запросе, а также наличие широкого спектра средств разработки, мониторинга и администрирования.

## 4. Математическое и программное обеспечение ИСППР

### 4.1. Построение онтологии предметной области

На этапе разработке метода решения сформулированной математической задачи будет создана онтология для получения качественного отображения предметной области, всех доступных в ней объектов и определения связей и характеристик между ними.

Сначала в виде онтологической структуры строится модель классовой иерархии специальностей, студентов и соответствующих характеристик, структуры университета. Затем создаются связи между всеми этими компонентами: прямые, обратные, один-к-одному, один-ко-многим и т.д.

Необходимо занести имеющиеся данные в программу. Для удобного поиска и представления данных, они будут отображены в виде иерархии классов (рис.4.1).

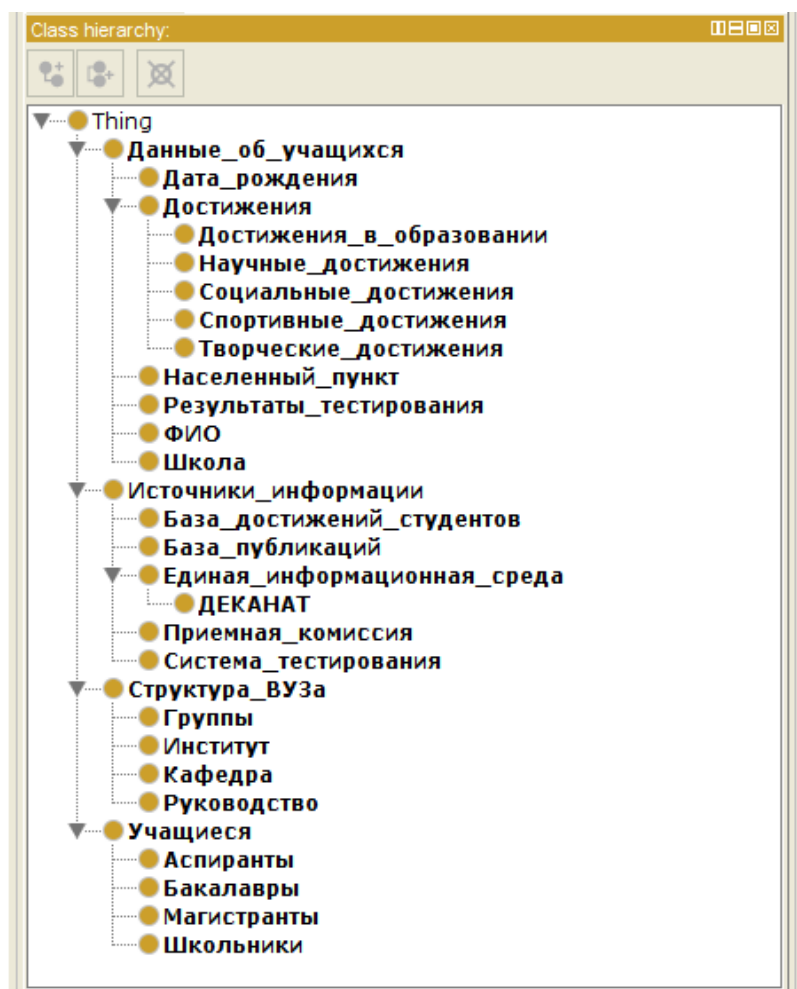


Рис. 4.1. Иерархия классов



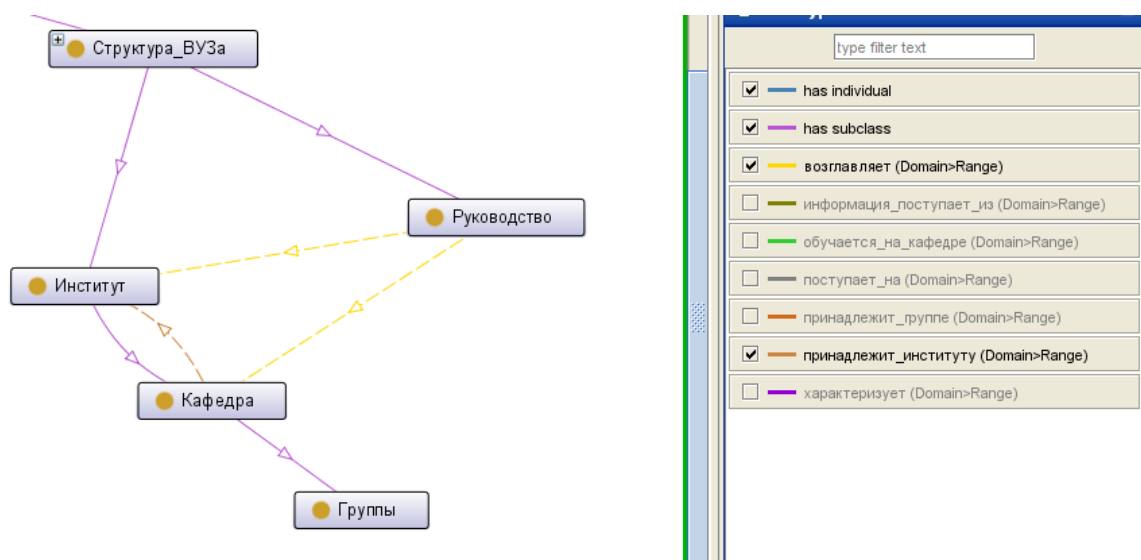


Рис. 4.3. Упрощенный онтограф. Структура ВУЗа со связями между элементами

Здесь легко определить логику и зависимость между элементами предметной области: Структура ВУЗа состоит как из управляемых (институты, кафедры), так и управляющих (руководство) частей. Соответственно эти части и связаны между собой связями типа «возглавляет». Внутри одной ветви также существуют связи подчинения и принадлежности. Группа «принадлежит» кафедре, кафедра – институту, но также и институт «обладает» кафедрой, и т.д.

Аналогично можно отобразить все остальные связи между объектами. Например, Бакалавры являются подклассом Учащихся (рис. 4.4) и имеют свои экземпляры (самих студентов). Их также характеризуют и определяют данные об учащихся. Бакалавры, безусловно прикреплены и обучаются на какой-то кафедре университета, которая принадлежит одному из институтов, что возвращает нас к предыдущей диаграмме и доказывает наличие связей, как прямых, так и косвенных между всеми объектами предметной области.

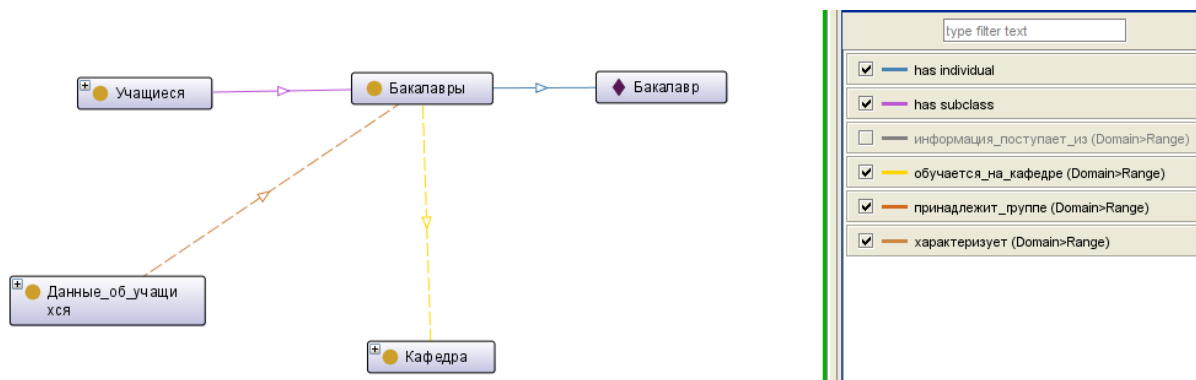


Рис. 4.4. Часть онтографа. Бакалавры

Данные о всех студентах (рис. 4.5) загружаются из источников информации (рис. 4.6).

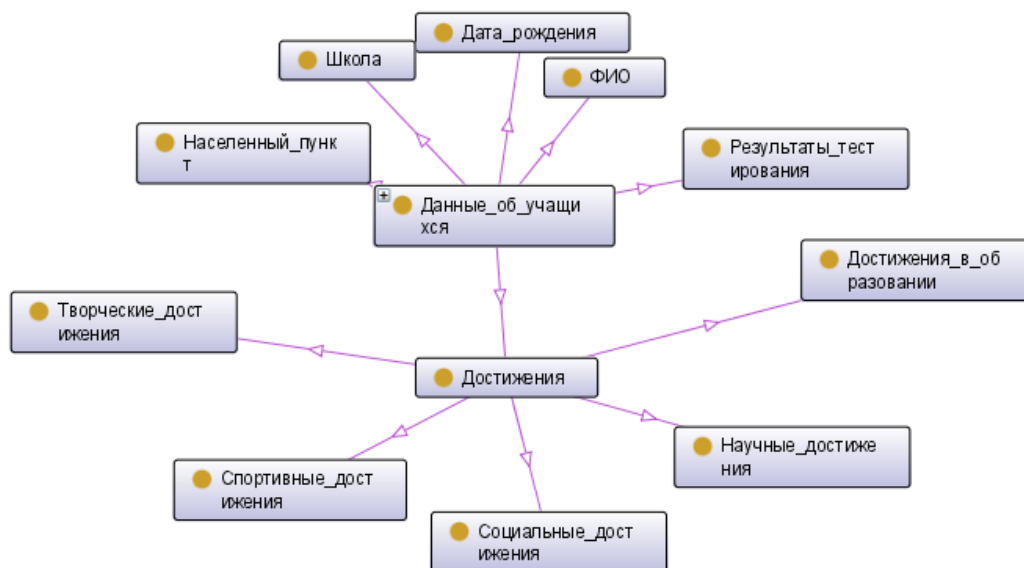


Рис. 4.5. Граф информации об учащихся

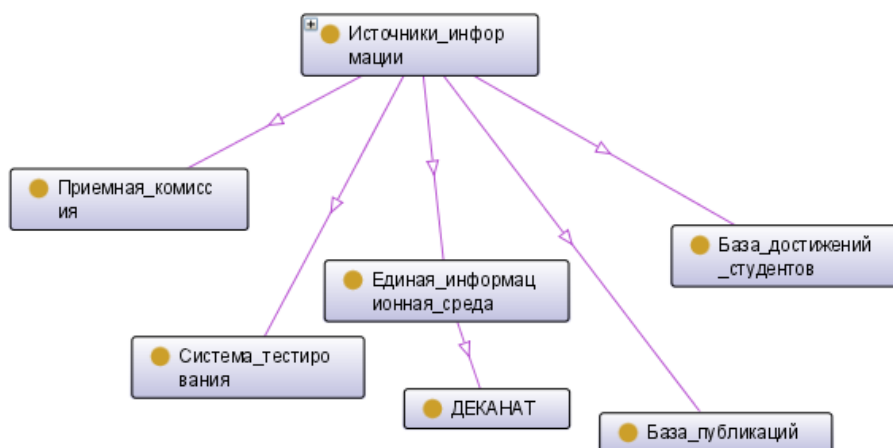
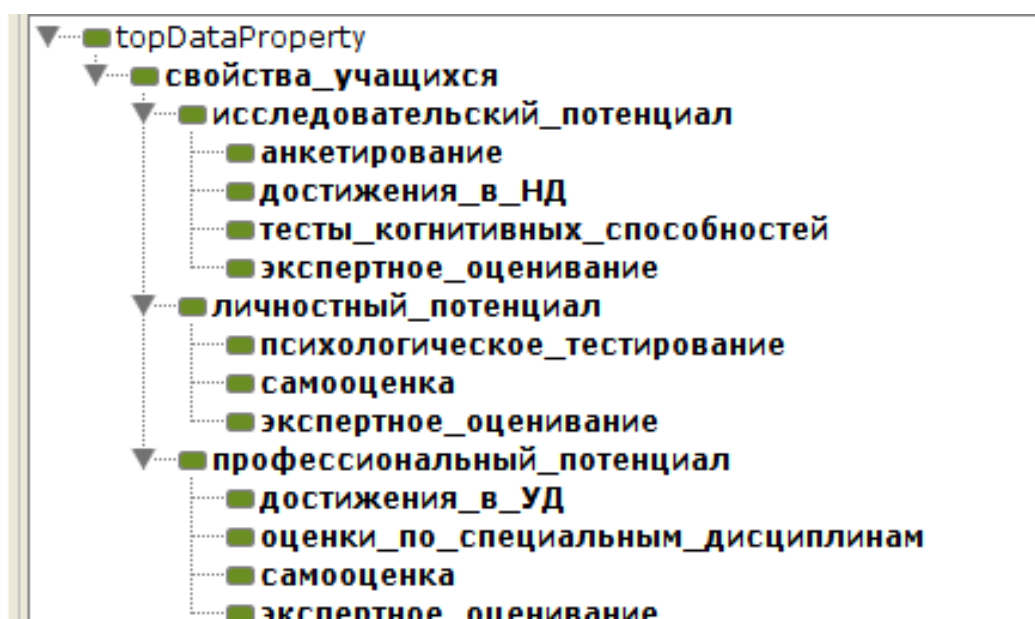


Рис. 4.6. Граф источников информации об учащихся

Для каждого из конечных элементов создан экземпляр класса. В программе Protégé класс обозначается наличием иконки круга слева от названия, а экземпляр класса – ромбом. Каждому созданному представителю специальностей соответствует набор значений личностных и психофизиологических характеристик, исследовательский и учебный потенциал и прочее. Они заносятся в онтологию в качестве свойств данных, например:



Здесь представленные свойства обозначаются в программе «data properties» и задаются на вкладке с аналогичным названием. Такие свойства являются характеристиками экземпляров класса и, фактически, определяют студентов.

Далее для обеспечения корректной работы программы и во избежание ошибок предоставления неполной информации необходимо прописать ограничение к каждому классу об обязательном наличии значений для всех упомянутых свойств личности. Для этого в окне базового класса «Thing» отметим все свойства, поставим метку Some (existential), обозначающую обязательное присутствие данного свойства в классе, и зададим тип значения – целое (integer) (рис. 4.7). На рис. показан результат присвоения свойств классам.

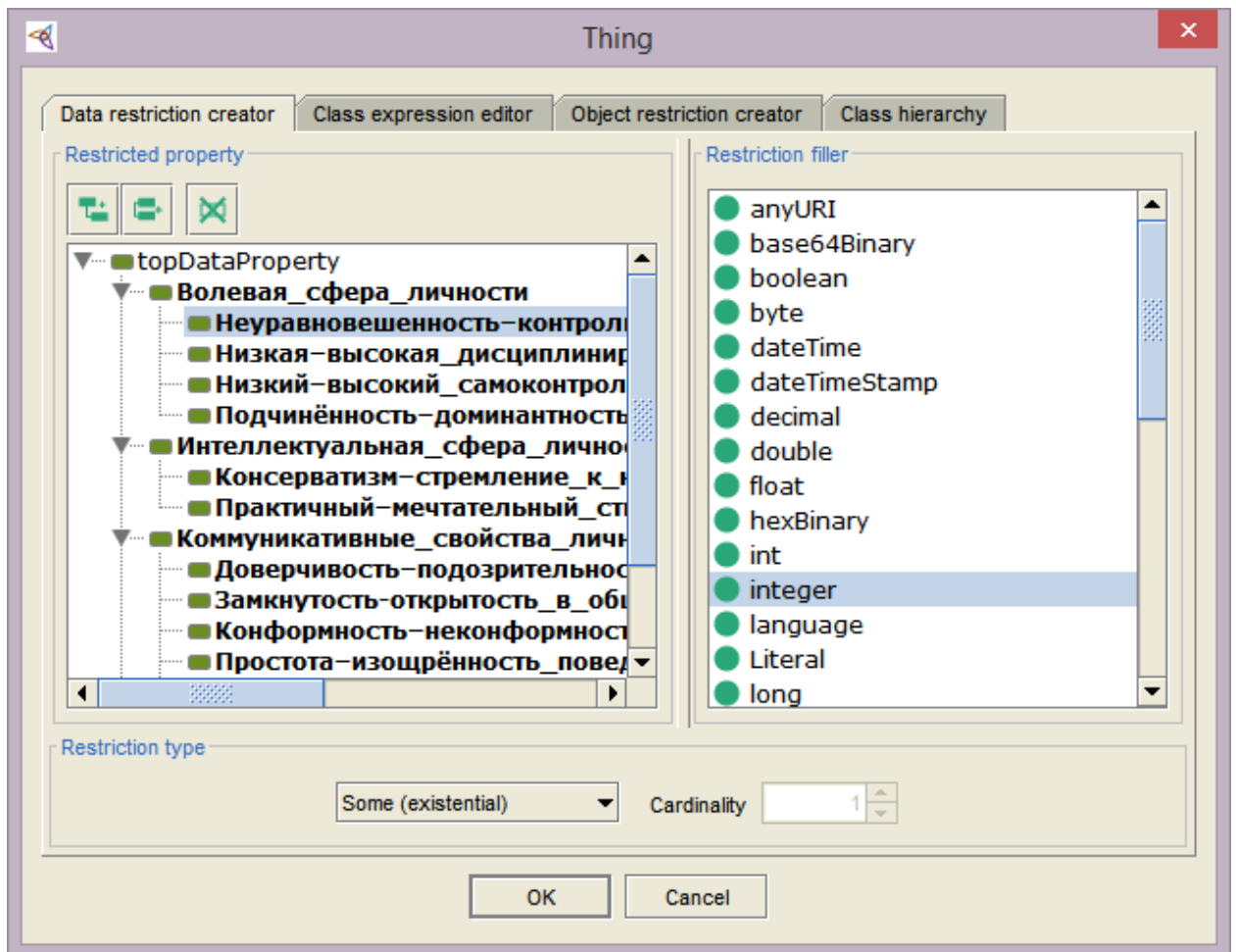


Рис. 4.7. Присвоение свойств классу Учащийся

Здесь свойства были присвоены только базовому классу Thing, но не были присвоены остальным представителям специальностей, как предполагалось сделать изначально. Однако это действие является достаточным, т.к. механизм наследования позволяет передавать свойства родительского класса дочерним. Поэтому, выбрав любой производный класс, можно убедиться в работоспособности указанного принципа.

В результате получаем модель предметной области в виде онтологии. С ее помощью наглядно отображаются все объекты, их свойства и связи, недоступные при первом рассмотрении, что приведет к улучшению качества анализа для построения информационной системы поддержки принятия решений.

## 4.2. Выделение основных факторов и характеристик

Описанные в предыдущем разделе свойства и связи, безусловно, имеют место быть в реальном мире и должны быть отображены в предметной области, однако не все они являются важными для исходной задачи. Поэтому следует выделить лишь главные из них, и возможно оставить те, что представляют интерес для анализа.

### 4.2.1. Определение функциональности системы

Создание информационной системы следует начинать с определения функций и задач, которые она должна уметь выполнять. Опираясь на построенную онтологию как на структуру предметной области, выделим основные задачи будущего приложения.

Программа по запросу пользователя должна:

- Предоставлять, изменять, удалять **Данные\*** об учащихя:
  - Личные данные
    - ФИО
    - Дата рождения
    - Населенный пункт, откуда приехал поступать
  - Данные студента
    - Группа
    - Институт
    - Кафедра
    - Направление
    - Уровень обучения
    - Курс
    - Дата поступления и отчисления
    - Является ли старостой или профоргом
  - Достижения в
    - Науке
      - Научный потенциал (дипломы, публикации, конференции)
      - Участие в грантах, научные стипендии, патенты и т.д.
    - Образовании
      - Средний балл



- Оценки по всем предыдущим семестрам и предметам в циклах
  - цикл ГСЭ - общие гуманитарные и социально-экономические дисциплины;
  - цикл ЕН - общие математические и естественно-научные дисциплины;
  - цикл ОПД - общепрофессиональные дисциплины;
  - цикл СД – специальные дисциплины;
  - ФТД – факультативы,
- Успевал/не успевал вовремя закрыть сессию
- Иностранный язык
- Академические обмены
- Олимпиады, конкурсы
- Уровень достижений в
  - Спорте
  - Творчестве
  - Общественной деятельности
- Личностные характеристики
- Психологические характеристики
- Производить выборку по критериям:
  - Данным\*
  - Периоду времени
- Производить выборку при помощи интеллектуальных методов ранжирования и кластеризации для:
  - Составления рейтинга студентов по группам, направлению, кафедре
  - Классификации студентов для выбора дальнейшей траектории обучения
- Отображать
  - Данные\*
  - Графики
    - Соответствия
    - Прогресса
  - Численные показатели
  - Советы и объяснения выводам
- Осуществление авторизации /выхода
- Настройка авторизованного пользователя



качественные переменные, причем качественные аспекты проблемы имеют тенденцию доминировать.

Неструктурированные проблемы имеют лишь качественное описание. Тем не менее, такие неструктурированные данные все же возможно описать при помощи разных типов шкал. Выделим из таких данных необходимые переменные/частные показатели на основе онтологии и вариантов использования.

Успешность в образовании, прежде всего, складывается из успеваемости и участия в олимпиадах и конкурсах, позволяющих выявить учебные достижения студента.

Средний балл складывается из балла за семестр и балла на экзамене или зачете, определяющихся по стандартной 100-бальной шкале. Оценка высчитывается из данного суммарного балла по ранговой шкале.

Необходимо отдельно анализировать успешность обучающегося при изучении предметов из разных циклов – это позволит определить его интересы и направленность на какую-то определенную сферу. Например, если он(а) сдает специальные дисциплины на высокий балл, а с предметами из общего профиля еле справляется, очевидно стремление к специализации. Здесь определим весовые значимость предметов из каждого ГСЭ:ЕН:ОПД:СД:ФТД цикла в соотношении 1:2:2:4:1, соответственно.

Аналогично в отдельную позицию для оценки следует выделить уровень знания английского языка. Значение данной переменной складывается из среднего балла за предмет, наличия сертификатов и опыта академического обмена, прохождения повышающих курсов, написанных на англ языке статей. Итого определяется по формуле:

$$\text{English} = \text{level} * \text{cert} + 10^{-3} \sum E_{\text{publ}},$$

где level – соответствующая уровням знания английского шкала: {A1 A2 B1 B2 C1 C2; 0,5 1 1,5 2 2,5 3};

cert – значимость сертификата, например, отсутствует -1; ТПУ – 1,5; прочие региональные сертификаты – 2, международные (CEF, TOEFL и др.) – 3;

Epubl – баллы за публикации на английском языке.

Еще одним показателем успешности в учебной деятельности является участие и победа в олимпиадах и конкурсах на различных уровнях. В оценке учитывается уровень мероприятия и занятое место. Баллы за достижения, согласно уже существующим, складываются друг с другом и нормируются на максимальное значение в 10000:

$$AcadAch=10^{-4}*\sum sc_i*p_i*k_i,$$

где  $sc_i$  – балл за соответствующее достижение,

$p_i$  – коэффициент, зависящий от занятого места,

$k_i$  – коэффициент участия.

Далее оценим достижения в науке. В первую очередь свидетельствами активной научной деятельности являются статьи, участие в грантах, научные стипендии, конкурсы и прочее. Для многих проектов и конкурсов допускаются к участию более одного человека, в данном случае общий балл за достижение необходимо домножить на соответствующий коэффициент участия.

Все упомянутые показатели вычисляются аналогично предыдущему.

Чтобы провести всесторонний анализ студента, необходимо учитывать не только его достижения в учебе и науке, а также знать его личностные качества.

Одним из таких ярких показателей будет ответственность. Ее можно определить, например, используя информацию о количестве пропусков (miss) или несданных вовремя предметов (intime), продленных сессиях.

Примем 1 за вовремя сданный предмет, а 0 – закрытый с опозданием.

Общей оценкой будет среднее по всем предметам, т.е. получаем число в интервале [0,1]. Аналогично определяется кол-во пропусков по предметам.

Таким образом, можно получить интегральную оценку, определяющую степень ответственности, суммируя вышеописанные переменные с весовыми коэффициентами:

$$\text{Resp}=10*\text{intime}-\text{miss}.$$

На успешность влияют и психологические параметры личности, такие как Логический интеллект, вербальный интеллект, коммуникабельность, консерватизм, тревожность, мотивация и др.

Они определяются по ранговой шкале исходя из результатов тестирований. Такие данные доступны не для каждого учащегося.

Также интересно посмотреть, существует ли зависимость успехов и достижений студента от типа населенного пункта, в котором он обучался ранее. Поэтому введем номинальную шкалу для описания данного показателя, где 1 – город, 2 – город-миллионник, 3- село и т.д.

В оценке стоит учесть и то, является ли студент какой-либо группы ее старостой или профоргом.

Теперь сведем все необходимые элементы с их допустимыми значениями (табл. 4.1)

Таблица 4.1 – Характеристики объекта

Достижение	Тип шкалы	Значения	Весовой коэффициент	Обозначение
<b>Входные переменные</b>				
Тип нас пункта	Номинальная	1,2,3,4	0,25	locality
Институт			0,25	Inst
Уровень обучения			0,25	Dgree
Курс			0,25	year
<b>Наука</b>				
Научный потенциал	интервальная	[0,10000]	4	publications
Участие в грантах, научные стипендии, патенты	интервальная	[0,10000]	6	sciach
<b>Образование</b>				
Средний балл	отношений	[0,100]		avgScore
оценка	ранговая	2,3,4,5		mark
<b>Оценки по всем предыдущим семестрам и предметам в циклах</b>				
цикл ГСЭ	отношений	[0,100]	1	HSE
цикл ЕН	отношений	[0,100]	2	SCI
цикл ОПД	отношений	[0,100]	2	GP
цикл СД	отношений	[0,100]	3	SP
цикл ФТД	отношений	[0,100]	1	Fac
Успевал/не успевал вовремя закрыть сессию	интервальная	[0,1]	0,5	intime
Пропуски	интервальная	[0,1]	0,5	miss
Олимпиады, конкурсы	интервальная	[0,10000]	5	olympiad
Спорт	ранговая	0, 0.5, 1	1	sport
Творческая деятельность	Ранговая	0, 0.5, 1	1	art
Общественная деятельность	Ранговая	0, 0.5, 1	1	socact
Актив группы	Номинальная	0,1,2	1	lead
<b>Личностные и психологические характеристики</b>				
Креативность	Ранговая	[0,12]	1	cr
Логический интеллект	Ранговая	[0,12]	1	log
Вербальный интеллект	Ранговая	[0,12]	1	spint
Коммуникабельность	Ранговая	[0,12]	1	com
Консерватизм	Ранговая	[0,12]	1	cons
Тревожность	Ранговая	[30,60]	1	wor
Мотивация	Ранговая	[-25,55]	1	mot
Ответственность	интервальная	[0,1]	1	resp
Уровень английского	интервальная	[0,10000]	3	english
<b>Интегральные показатели</b>				
Общий	интервальная	[0,1]	-	Index
Для учебы	интервальная	[0,1]	-	academic
Для науки	интервальная	[0,1]	-	science

В качестве итоговых переменных, определим три интегральных показателя: с акцентом на научную деятельность; определяющий учебный потенциал; и общий, который будет учитывать оба этих пункта.

Таким образом, исходя из анализа предметной области, были выделены наиболее важные параметры рассматриваемого объекта – студента – с точки зрения учебного и научного аспектов. А также определены целевые переменные для построения рейтинга студентов.

### **4.3. Построение дерева решений и базы знаний**

В связи с наличием слабоструктурированных данных, использование нечетких деревьев решения для построения ИСППР оправдано. Главной идеей в таком подходе является сочетание возможностей деревьев решений и нечеткой логики. Отличительной чертой нечетких деревьев решений является то, что в каждом примере для конкретного узла нечеткое дерево решений группирует их степень принадлежности.

Однако для получения адекватных результатов необходим репрезентативный набор обучающих примеров, в противном случае сгенерированное алгоритмом дерево решений будет слабо отражать действительность и, как следствие, выдавать ошибочные результаты.

Для каждой из переменных создается лингвистическая переменная, определяются функции принадлежности и выводятся правила в базе правил при построении нечеткого дерева решений.

В силу специфики рассматриваемой задачи при построении нечетких систем принятия решений, будут применяться групповые прямые методы с привлечением экспертных оценок.

#### **4.3.1. Определение лингвистических переменных**

Определение лингвистических переменных происходит следующим образом:

Пусть определяется научный потенциал студента с помощью понятий «Низкий», «Средний» и «Высокий», при этом минимальная оценка равна 0, а максимальная – 1000. Формализация такого описания может быть проведена с помощью следующей ЛП

$$\text{publications} = (\beta, T, X, G, M),$$

где  $\beta$  = научный потенциал студента;

$$T = \{\text{«Низкий»}, \text{«Средний»}, \text{«Высокий»}\};$$



$$X = [0, 1000];$$

Для каждого терма из  $T$  создается функция принадлежности:

$G$  – процедура образования новых термов с помощью связок «и» «или» и модификаторов типа «очень», «слегка», «не», «почти», «не совсем», и т.п.

$M$  – процедура задания на  $X = [0, 1000]$  нечетких подмножеств:  $A1 =$  «Низкий»,  $A2 =$  «Средний» и  $A3 =$  «Высокий», а также нечетких множеств для термов из  $G(T)$  в соответствии с правилами трансляции нечетких связок и модификаторов.

Функции принадлежности  $\mu_1(x)$ ,  $\mu_2(x)$  и  $\mu_3(x)$ , определяющие « $A1 =$  «Низкий»,  $A2 =$  «Средний» и  $A3 =$  «Высокий», соответственно, представлены на рис. 4.9 и задаются следующим образом:

Функции принадлежности для показателя «научный потенциал студента»:

$$\mu_1(x) = \begin{cases} 1, & 0 \leq x < 50 \\ 2 - 0.02x, & 50 \leq x < 100 \\ 0, & x \geq 100 \end{cases}$$

$$\mu_2(x) = \begin{cases} 0, & 0 \leq x < 50, x \geq 500 \\ 0.02x - 1, & 50 \leq x < 100 \\ 1, & 100 \leq x < 250 \\ 2 - 0.004x, & 250 \leq x < 500 \end{cases}$$

$$\mu_3(x) = \begin{cases} 0, & 0 \leq x < 300 \\ 0.01x - 3, & 300 \leq x < 400 \\ 1, & x \geq 400 \end{cases}$$

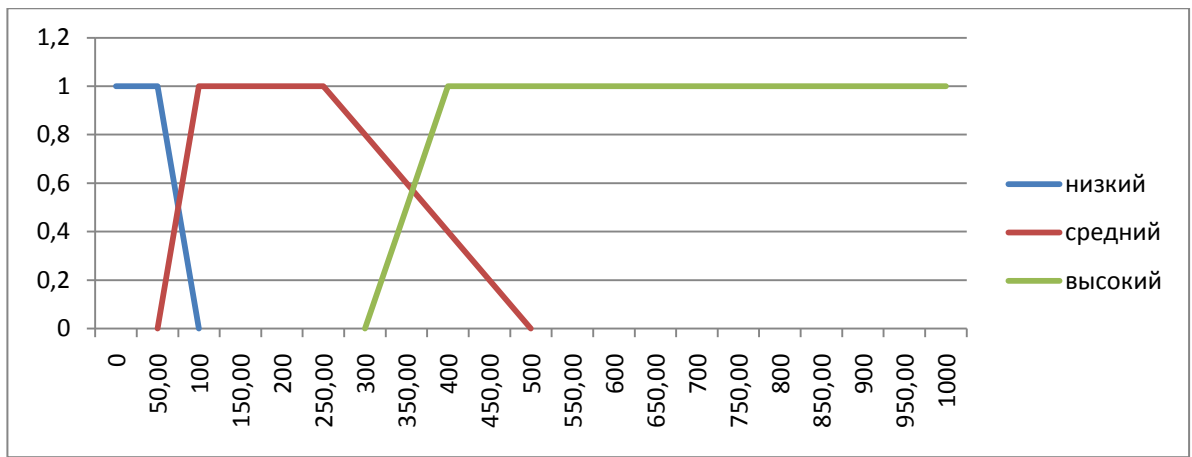


Рис. 4.9. Функции принадлежности для термов ЛП «научный потенциал студента»

Процедура образования новых термов G с помощью связок «и» «или» и модификаторов типа «очень», «слегка», «не», «почти», «не совсем», можно представить следующим образом:

Связки «и» и «или» порождают новую функцию принадлежности из двух используемых при помощи их произведения или среднего значения, соответственно.

Модификатор «очень» сдвигает соответствующую функцию принадлежности на вправо. Похожее действие производят и прочие модификаторы.

Аналогично определяются лингвистические переменные для остальных показателей (см приложение Г).

Таким образом, созданы ЛП для всех интервальных, ранговых параметров и характеристик, описываемых шкалой отношений. Для параметров, принимаемых значения по номинальной шкале, лингвистические переменные не определяется. В таком случае для классификации, каждое значение используется в его исходном виде без фаззификации.

Например, активность студента в группе может принимать следующие значения: «староста», «профорг», и «профорг», и переменную *lead* можно

определить как набор значений и соответствующих им значений числового кода.

$$lead = \{\text{«староста»}, \text{«профорг»}, \text{«профорг»}; 1, 2, 3\}$$

ЛП для классификации является Уровень соответствия студента одному из классов: «Идеальный студент», «Хорошист», «Средний», «Проблемный»

$$(\beta, T, X, G, M),$$

где  $\beta$  = успешность студента;

$$T = \{\text{«Идеальный студент»}, \text{«Хорошист»}, \text{«Средний»}, \text{«Проблемный»}\};$$

$$X = [0, 1];$$

G – процедура образования новых термов с помощью и модификаторов.

M – процедура задания на  $X = [0, 100]$  нечетких подмножеств:  $A_1 =$  «Идеальный студент»,  $A_2 =$  «Хорошист»,  $A_3 =$  «Средний» и  $A_4 =$  «Проблемный», а также нечетких множеств для термов из  $G(T)$  в соответствии с правилами трансляции нечетких связок и модификаторов.

#### 4.3.2. Классификация

Затем данные из базы данных используются для построения нечеткого дерева решений. Внутренние узлы дерева являются атрибутами базы данных.

Формирование базы правил основано на пошаговом режиме обучения, в результате которого получают нечеткое дерево решений. В каждом узле дерева при обучении корректируются значения и интерполируют функции принадлежности с целью последующей фаззификации входных переменных. На этапах агрегирования получаем оптимизацию количества нечетких логических правил. Аккумуляция заключений и дефаззификация происходят пошагово в каждом узле дерева решений. Тестирования построенного нечеткого дерева решений происходит на примерах реальных данных. Далее проверяется критерий качества обучения системы, путем его

сравнивая с целевым значением. При достижении целевого значения считается, что построение нечеткого дерева решений завершено и создана база нечетких правил.

Построим дерево по набору минимальных характеристик: успеваемости в разных предметных циклах и достижений в науке и учебе. Пример использованных данных для построения и тестирования дерева представлены в таблице 4.2.

Таблица 4.2 – Данные для обучения и тестирования дерева решений

Научный потенциал	ГСЭ	ЕН	ОПД	СД	ФТД	Класс
-1	0,6	0,8	0,86	0,94	-1	<b>В</b>
-1	0,8	0,8	0,88	0,95	-1	<b>В</b>
-1	0,7	1	0,95	0,92	-1	<b>В</b>
0,018	1	1	1	0,96	1	<b>А</b>
0,023	1	1	1	0,95	-1	<b>А</b>
-1	0,8	0,9	0,92	0,94	-1	<b>В</b>
-1	1	1	0,86	0,8	0,8	<b>В</b>
0,196	1	1	1	1	-1	<b>А</b>
-1	0,85	0,74	0,8	0,74	-1	<b>С</b>
-1	-1	0,74	0,96	0,9	-1	<b>В</b>
-1	-1	0,9	0,92	0,98	0,975	<b>В</b>
-1	-1	1	1	1	0,975	<b>А</b>
-1	-1	0,74	0,74	0,76	0,8	<b>С</b>
-1	-1	0,8	0,88	0,92	0,925	<b>В</b>
-1	-1	0,8	0,88	0,96	1	<b>В</b>

Алгоритм построения дерева решений допускает существование пропущенных значений. Поэтому в приведенной таблице ячейки, значения которых неизвестны, отмечены числом «-1».

В итоге получаем следующую структуру (рис. 4.10)

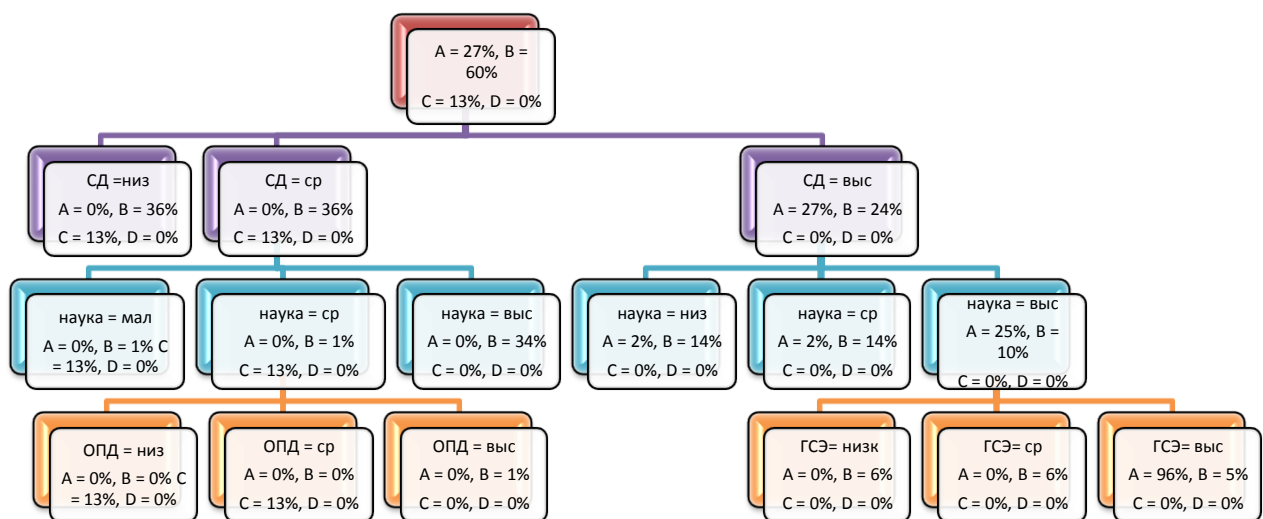


Рис.4.10. Дерево решений

Здесь при формировании правил происходит расщепление по разным показателям на разных уровнях. Результат расщепления в листах принадлежит не одному конкретному классу, а отображает их степень принадлежности, что дает более полную и качественную оценку результатов классификации.

### 4.3.3. Регрессия

По тем же исходным данным, взятым из базы данных, были построены деревья регрессии для формирования интегрального показателя. Пример дерева для построения рейтинга студентов по учебным достижениям представлен на рис. 4.11. Такое дерево формирует значения выходной переменной для новых данных.

Правила вывода для данного дерева представлены ниже:

- 1 если ОПД < 4.9 тогда перейти к узлу 2 иначе если ОПД ≥ 4.9 тогда перейти к узлу 3 иначе 0.15748
- 2 если СД < 3.9 тогда перейти к узлу 4 иначе если СД ≥ 3.9 тогда перейти к узлу 5 иначе 0.148936
- 3 если Актив < 0.5 тогда перейти к узлу 6 иначе если Актив ≥ 0.5 тогда перейти к узлу 7 иначе 0.180978
- 4 если ОПД < 3.85 тогда перейти к узлу 8 иначе если ОПД ≥ 3.85 тогда перейти к узлу 9 иначе 0.128537
- 5 если ЕН < 4.25 тогда перейти к узлу 10 иначе если ЕН ≥ 4.25 тогда перейти к узлу 11 иначе 0.153469

- 6 если научный потенциал < 109.65 тогда перейти к узлу 12 иначе если научный потенциал  $\geq 109.65$  тогда перейти к узлу 13 иначе 0.176644
- 7 интегральный показатель = 0.19398
- 8 интегральный показатель = 0.127805
- 9 интегральный показатель = 0.129268
- 10 если СД < 4.725 тогда перейти к узлу 14 иначе если СД  $\geq 4.725$  тогда перейти к узлу 15 иначе 0.150293
- 11 если ОПД < 4.45 тогда перейти к узлу 16 иначе если ОПД  $\geq 4.45$  тогда перейти к узлу 17 иначе 0.157439
- 12 если научный потенциал < 11.65 тогда перейти к узлу 18 иначе если научный потенциал  $\geq 11.65$  тогда перейти к узлу 19 иначе 0.170039
- 13 интегральный показатель = 0.189854
- 14 если СД < 4.65 тогда перейти к узлу 20 иначе если СД  $\geq 4.65$  тогда перейти к узлу 21 иначе 0.149268
- 15 если СД < 4.775 тогда перейти к узлу 22 иначе если СД  $\geq 4.775$  тогда перейти к узлу 23 иначе 0.151829
- 16 интегральный показатель = 0.149268
- 17 если ЕН < 4.75 тогда перейти к узлу 24 иначе если ЕН  $\geq 4.75$  тогда перейти к узлу 25 иначе 0.160163
- 18 интегральный показатель = 0.170732
- 19 интегральный показатель = 0.169346
- 20 если ЕН < 3.85 тогда перейти к узлу 26 иначе если ЕН  $\geq 3.85$  тогда перейти к узлу 27 иначе 0.149024
- 21 интегральный показатель = 0.149756
- 22 интегральный показатель = 0.151463
- 23 интегральный показатель = 0.152195
- 24 если СД < 4.8 тогда перейти к узлу 28 иначе если СД  $\geq 4.8$  тогда перейти к узлу 29 иначе 0.159024
- 25 интегральный показатель = 0.162439
- 26 интегральный показатель = 0.14878
- 27 интегральный показатель = 0.149268
- 28 интегральный показатель = 0.157561
- 29 интегральный показатель = 0.160488

Здесь  $x_1$  – научный потенциал,  $x_2$  – средний балл в цикле ЕН,  $x_3$  – в цикле ОПД,  $x_4$  – в цикле СД,  $x_5$  – актив группы.

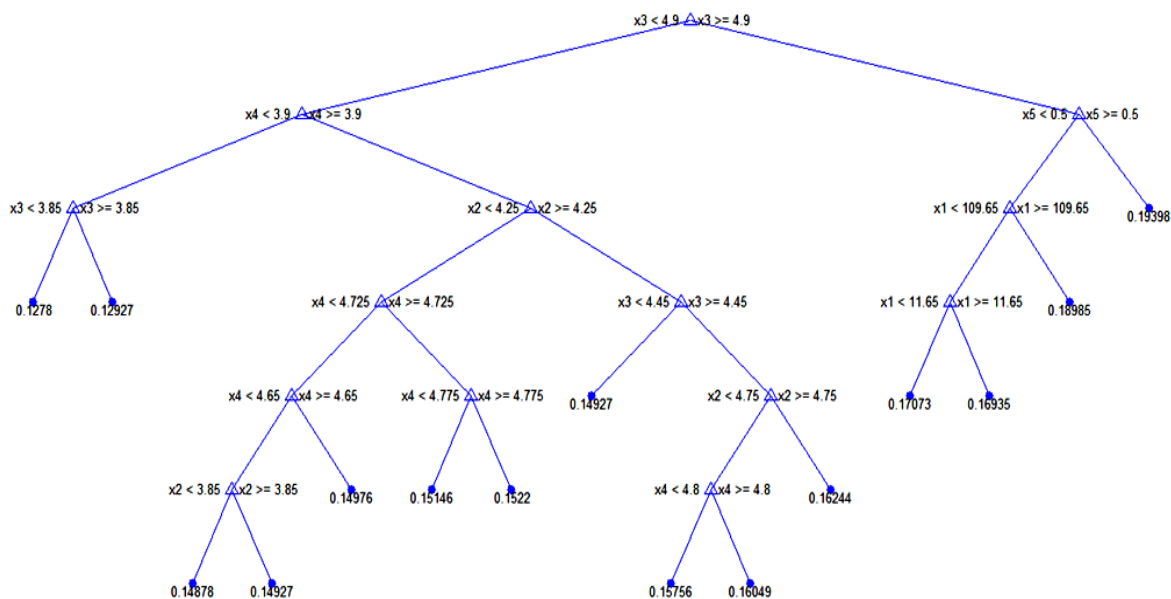


Рис.4.11. Дерево решений для регрессии

Полученные результаты продемонстрировали адекватность/работоспособность выбранных методов для решения задач классификации и формирования рейтинга анализируемых объектов. Построенные деревья решений отображают структуру связей и закономерностей между свойствами объектов и целевыми значениями. В дальнейшем они используются в качестве базы правил для классификации и построения рейтинга для новых объектов (студентов).

## 4.4. Построение базы данных

### 4.4.1. Выбор базовой структуры организации БД

Создание базы начиналось с определения данных, которые уже используются в базе университета, и которые необходимо создавать самостоятельно. Ключевыми из имеющихся данных являются значения `id_person` и `id_student`, с которыми в базе ТПУ ассоциируются все остальные данные, необходимые для оценки. Наличие двух идентификаторов в данном случае необходимо, т.к. один человек может, как продолжать обучение в магистратуре, так и одновременно обучаться на более чем одном направлении, т.е. нельзя однозначно определять личность, например, как студента определенной кафедры. На основе вышесказанного было выбрано три способа организации «ядра» БД – таблиц, описывающих студента и его успеваемости.

Первый способ (рис. 4.12) позволяет использовать полученные данные так как они есть, а также разделять данные, присущие только обучающемуся как личности или только как студенту (например, психологические характеристики относятся к личности и не зависят от того, где он учится, а группа – только к студенту, не определяя данные личности). Но у такого способа также есть и недостаток: при такой организации в каждой таблице достижений будет находиться как минимум два поля в первичном ключе: от таблиц Студент и Личность (`person`), что нежелательно, т.к. усложняет ввод данных и способствует созданию ошибок.

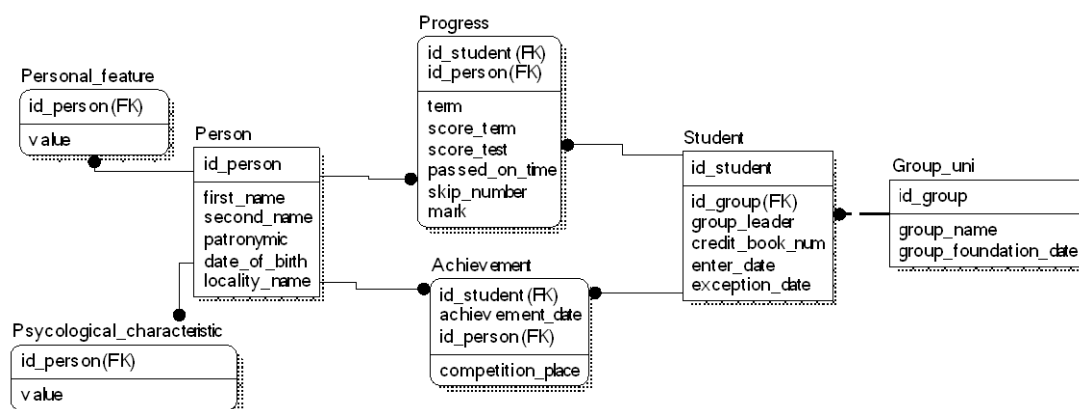


Рис. 4.12. Организация БД. Первый способ



Во втором способе (рис. 4.13) при построении таблиц достижений передается лишь один идентификатор, созданный на основе двух уже существующих при помощи объединения их в одну таблицу Lerner. В данной таблице также собраны все личные данные и сведения как о студенте. Это с одной стороны дает быстрый доступ ко всей информации об обучающемся, а с другой может привести к неполноте информации или ее дублированию в тех случаях, когда человек учится на 2х специальностях, или продолжает обучение в магистратуре. Такая ситуация может создать путаницу в данных и усложнит их поиск и анализ.

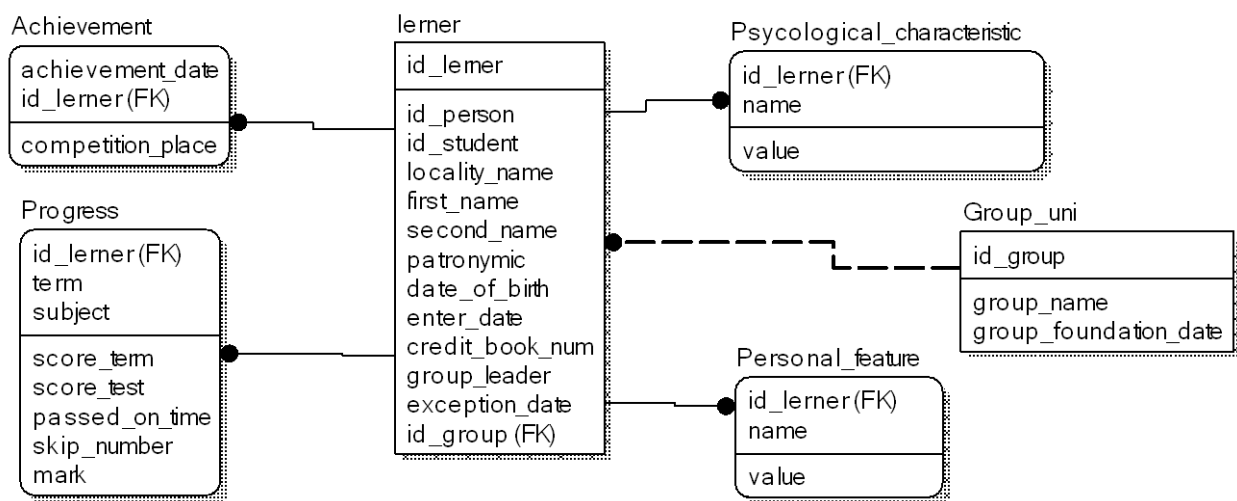


Рис. 4.13. Организация БД. Второй способ

Третий способ, также как и предыдущий, предполагает передачу лишь одного атрибута из первичного ключа таблиц Личность и Студент (рис. 4.14). Он также позволяет хранить соответствующую информацию отдельно друг от друга. Это достигается за счет связывания двух таблиц путем введения идентификатора из Личности к Студенту в качестве внешнего ключа. Таким образом, решается вопрос об отношении «двух» студентов к «одной» личности. Сложностью при таком подходе будет выборка параметров, находящихся на «противоположных» концах базы данных. Это потребует создание крупных запросов с множеством внутренних объединений.

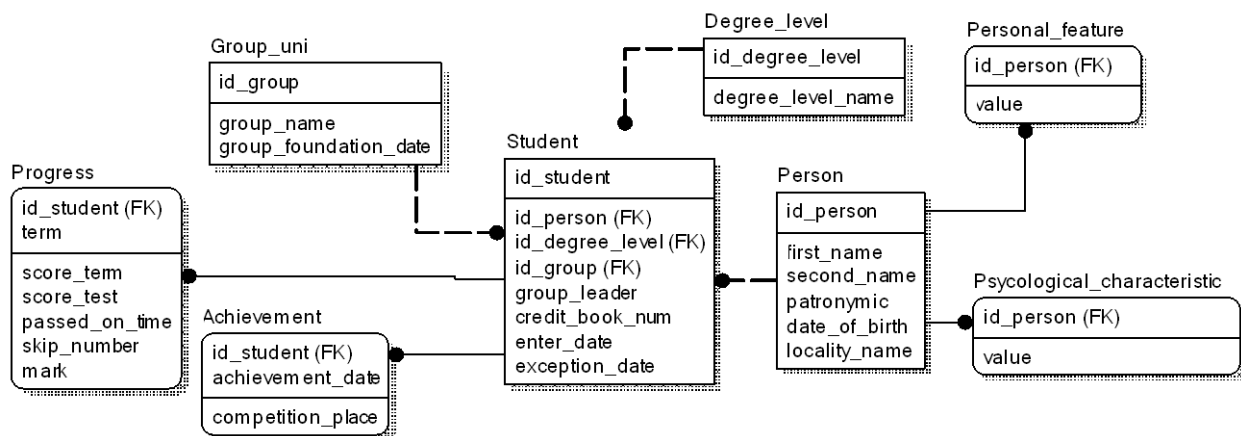


Рис. 4.14. Организация БД. Третий способ

Выбор остановили на третьем варианте, т.к. для анализа и классификации очень важно прозрачность и полнота данных, что исключает второй вариант, а также удобство манипуляции ими, что проще при передаче лишь одного идентификатора. Таким удобством обращения к таблицам не обладает первая организация БД. Остается последняя проблема, касающаяся сложности запросов. Ее можно смягчить, используя выборку на основе представлений в базе данных.

#### 4.4.2. Построение полной схемы БД

Далее были добавлены таблицы, определяющие учебную, научную, общественно-творческую и спортивную деятельности студента, знание иностранных языков и его личностные и психологические характеристики.

Для многих полей потребовалось создать таблицы-кодификаторы, в которых хранятся все возможные значения соответствующих полей. Такие кодификаторы состоят лишь из идентификатора поля, в качестве первичного ключа, и его наименования и имеют ограниченное количество кортежей (записей).

Для каждой таблицы, имеющей свой собственный идентификатор, создавалась последовательность и триггер, устанавливающий значение уникального первичного ключа.

Например, таблица-кодификатор Institute отражает все существующие институты в ТПУ и выглядит следующим образом:

Таблица 4.3 – Таблица-кодификатор об институтах ТПУ

ID_INSTITUTE	INSTITUTE_NAME
1	Институт кибернетики (ИК)
21	Институт природных ресурсов (ИПР)
22	Энергетический институт (ЭНИН)
23	Институт физики высоких технологий (ИФВТ)
24	Физико-технический институт (ФТИ)
25	Институт неразрушающего контроля (ИНК)
26	Институт социально-гуманитарных технологий (ИСГТ)
27	Юргинский технологический институт ТПУ (ЮТИ)
28	Институт электронного обучения (ИнЭО)

Также для обработки сессий приложения необходимо создать таблицы, которые будут хранить данные о пользователях, такие как связки логин-пароль, электронный адрес, уровень доступа к приложению, параметры пользовательских настроек и т.д.

В итоге получаем следующую БД, которая разделена на две независимые части: данные о студентах (рис. 4.16) и данные о пользователях приложения (рис. 4.15).

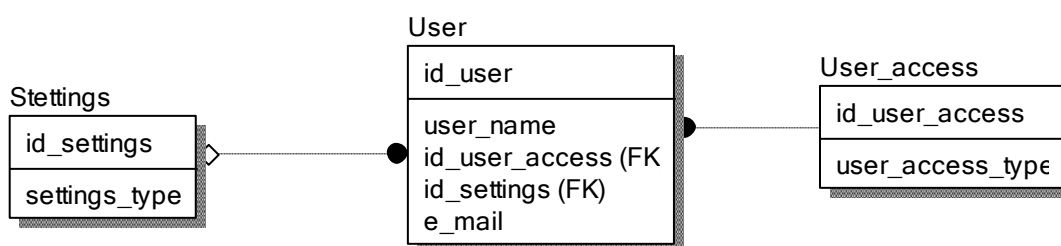


Рис. 4.15. БД по пользователям



участие в общественной деятельности группы (например, является ли старостой или профоргом). Также к параметрам студента относится таблица, содержащая интегральный показатель, определяющийся методами data mining, описанными ранее. Он, как и многие другие данные в базе имеет привязку ко времени, что позволяет провести временной анализ деятельности студента. На основании такого анализа можно принимать решения о продвижении студента в магистратуру, рекомендации ему окончить обучение на текущем уровне, или даже об отчислении.

Таблицы, содержащие сведения о достижениях в научной и учебной деятельности имеют ограничение, состоящее в невозможности присвоения данного достижения более чем одному человеку. А так как в реальности множество конкурсов и проектов осуществляется более чем одним человеком, то для внедрения подобной функциональности, необходимо введение промежуточной таблицы между таблицами Student и Science\_achievement. Такая организация дает возможность присвоить достижение более чем одному человеку, а также указывает коэффициент его участия в этой деятельности.

Многие атрибуты и связи в представленной базе не являются обязательными и могут оставаться пустыми (иметь значение NULL). С одной стороны, это плохо, т.к. не позволит провести полноценный анализ всех студентов по всем характеристикам, что может привести к погрешности в итоговом рейтинге. Однако, также очевидно, что нельзя гарантировать наличие всех данных об обучающихся. С другой стороны, можно было бы сократить количество параметров до минимального, но гарантировать их полный набор. Но в таком случае, можно лишиться более точной характеристики при классификации хотя бы тех студентов, для которых эта информация имеется.

Получившаяся база данных содержит все необходимые поля и связи для описания студентов, их характеристик и деятельности при помощи определенных ранее переменных.

## 4.5. Проектирование программного интерфейса

### 4.5.1. Определение структуры приложения

Информационная модель должна иметь веб-интерфейс, связываться с базой данных и управлять выборкой данных согласно запросам пользователя. Одним из шаблонов проектирования подобных систем является паттерн MVC.

В APEX каждая страница определяется декларативно с использованием метаданных для выбора нужных в приложении шаблонов. APEX динамически отображает HTML страницу, используя метаданные, соединяя соответствующие шаблоны и вставляя динамические данные в шаблоны. HTML страница отображается при ее запросе через веббраузер. При отправке страницы обработчик APEX еще раз выполняет ее обработку, используя декларативно определенные метаданные для выполнения вычислений, валидации, и прочих процессов.

Такой тип обработки отражает шаблон MVC, где созданный с использованием шаблонов приложений HTML является видом (view), обработчик APEX – контроллером, который принимает на вход запросы типа GET или POST и решает, что с ними делать, передавая объектам домена. Модель объектов домена помещена при определении страницы и содержит бизнес-правила и функциональность для выполнения конкретных задач.

Шаблон MVC также реализует другой хороший принцип проектирования, основанный на разделении задач. APEX был разработан таким образом, чтобы его обработчик, шаблоны, и функциональные возможности приложения могли быть оптимизированы независимо друг от друга. Очевидно, что обработка сборки и упорядочивания шагов необходима для визуализации страницы, а обработка страницы важна для общего решения. Разделив эти компоненты и позволив Oracle разбираться со сложностями при помощи APEX, разработчики могут сосредоточиться на предоставлении бизнес-функций. Аналогично, отделив шаблоны представления от бизнес-логики, возможно осуществлять поддержку каждого аспекта отдельно друг от друга. Это

обеспечивает ряд преимуществ, в том числе простоту внесения изменений в структуру, что позволяет изменять шаблоны в разное время либо разными людьми, не нарушая приложения.

Ярким примером этому может служить стандартные темы, представленные в АРЕХ, созданные полностью взаимозаменяемыми. Чтобы переключить стандартные темы достаточно просто загрузить ее из репозитория, и сделать активной. Затем АРЕХ перераспределяет компоненты новой активной темы с помощью идентификаторов шаблона класса.

#### 4.5.2. Схема переходов

Распишем работу приложения. Схема возможных переходов показана на рис. 4.17.

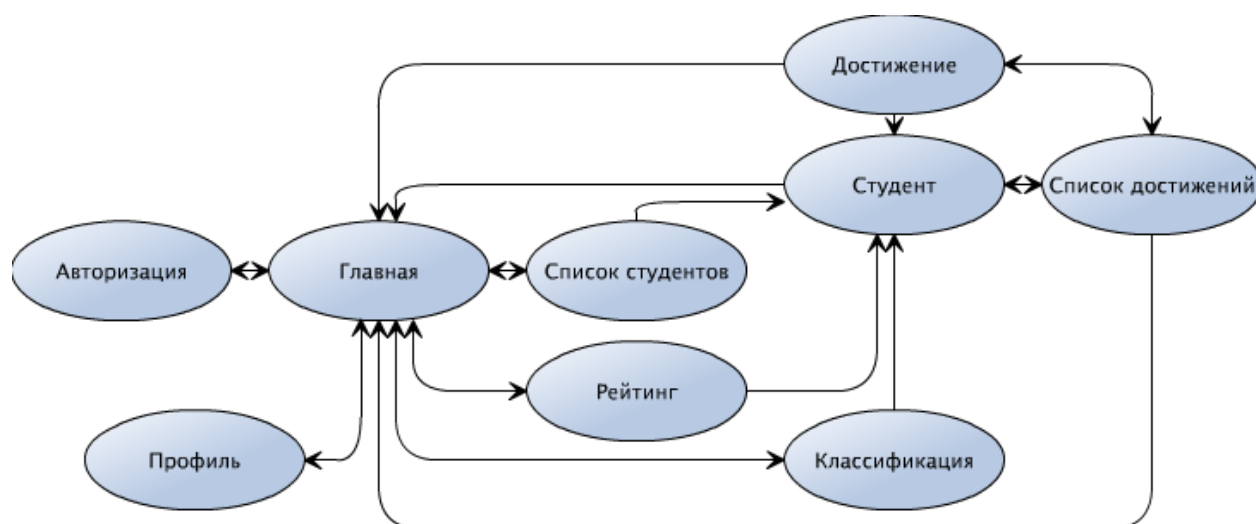


Рис. 4.17. Схема переходов в приложении.

Первая и единственная доступная на момент входа страница – страница авторизации (рис. 4.18). Чтобы воспользоваться функциональностью приложения необходимо ввести пару логин-пароль, содержащиеся в БД для авторизации. При неверной комбинации появится соответствующее сообщение, и доступ получен не будет.

Авторизация

Логин

Пароль

Войти

Рис. 4.18. Страница 101 авторизации

В случае успешной авторизации, пользователь попадает на главную страницу приложения (рис. 4.19), где слева расположен блок отчета со всеми группами с количеством в них учащихся, сгруппированные по соответствующим направлениям и кафедрам. Справа находится блок оценки, с помощью которого можно отправить запрос на формирование рейтинга студентов.

Welcome: DSSAPP Logout

Главная

Search  Display 15 Go Reset

Кафедра	Направление	Группа	Количество человек в группе
Прикладная математика	демо	000	2
Прикладная математика	Прикладная математика и информатика	8БМ41	2
Оптимизация систем управления	Прикладная информатика	8КМ41	8
Оптимизация систем управления	ПИ (8к00)	8592	1
Оптимизация систем управления	ПИ (8к00)	8К00	6
Оптимизация систем управления	ПИ (8к00)	8К10	1
Оптимизация систем	Информатика и вычислительная техника Сети		

Оценка

Кафедра

Направление Прикладная информатика

Группа

Студент

Классификация Рейтинг

Рис. 4.19. Страница 1. Главная

При нажатии на любое значение направления, кафедры или группы, приложение осуществляет переход на страницу 3 – «Список студентов» (рис. 4.20) и передает соответствующее значение в скрытые поля на ней.



Главная > Список студентов

Номер группы **8КМ41**  
 Дата создания **01-09-2014**  
 Институт **Институт кибернетики (ИК)**  
 Номер специальности **90403**  
 Специальность **Прикладная информатика**  
 Кафедра **Оптимизация систем управления**  
 Количество человек в группе **8**

Go Create

	Фамилия	Имя	Очество
	Абраамян	Манас	Абраамович
	Алексеева	Алина	Александровна
	Баранова	Анна	Владимировна
	Верхотурова	Ариадна	Эдуардовна
	Герасимова	Надежда	Игоревна
	Рамазанов	Рахим	Нарудинович
	Садковская	Светлана	Юрьевна
	Шагарова	Мария	Дмитриевна

Рис. 4.20. Страница 3. Список студентов

На третьей странице согласно переданному с главной значению отображаются все студенты выбранного подразделения. Если это была группа – то еще выдается информация о самой группе: название, дата создания, институт, кафедра, направление и количество человек в группе. С данной страницы можно либо вернуться на главную (нажав на соответствующий путь в иерархической навигации сверху страницы или кнопку назад в браузере), или же перейти на страницу редактирования и подробного отчета о студенте – 4 (рис. 4.21), нажав на значок редактирования слева от выбранного имени.

Главная > Студент

Данные о студенте

\*Фамилия   
 \*Имя   
 Очество   
 \*Дата рождения   
 Откуда   
 \*Тип населенного пункта   
 Творчество   
 Спорт   
 Общественная деят-ль

Удалить Отмена Сохранить

Показать достижения

Рис. 4.21. Страница 4. Студент

Попав на четвертую страницу, сначала будет отображена только форма о личных данных студента, которые можно поменять при желании, а также выпадающий список активности студента ниже этой формы. Значение по умолчанию в нем равно «не выбрано». При смене значения появляется соответствующая информация о студенте с графиком прогресса в течение времени. Например, выбрав значение «успеваемость» страница отчета примет следующий вид (рис. 4.22). Здесь, в центральном блоке даны краткие и общие сведения по выбранному параметру – успеваемости – такие как средний балл за дисциплины в различных циклах, количество пропусков, количество несданных вовремя предметов и т.д. Аналогичным видом и функциональностью обладают и прочие значения из выпадающего списка: «наука», «успеваемость» и «личностные и психологические характеристики».

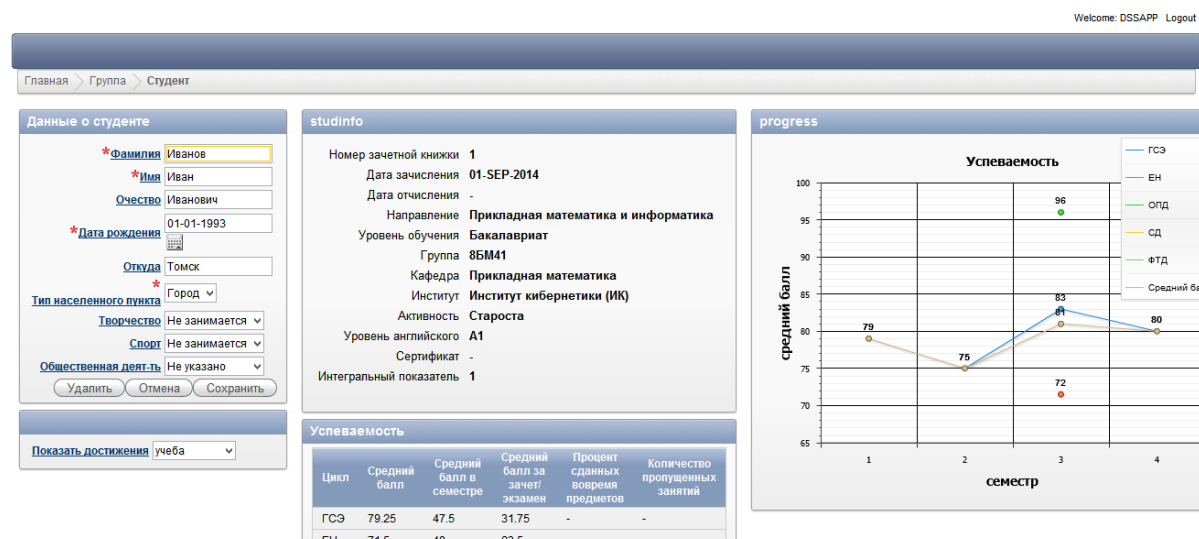


Рис. 4.22. Страница 4. Студент, с выбранным значением «учеба» в выпадающем списке

Ниже центрального списка находится кнопка «Подробнее», при нажатии на которую произойдет переход на страницу 2 «Список достижений» (рис. 4.23.), где уже будет отображен полный список предметов с баллами и пропусками по каждому, сгруппированные и отсортированные по циклам и семестрам.

Главная > Группа > Студент > Report on PROGRESS

Report on PROGRESS Create

Edit	Id Student	Term	Id Subject	Type Academic	Score Term	Score Test	Passed On Time	Skip Number	Mark
	6	1	1		56	23			4
	6	2	1		40	35			
	6	3	1		45	40			
	6	4	1		50	30			
	6	3	21		50	22			
	6	3	21		46	25			
	6	3	22		58	38			

Рис.4.23. Страница 2. Список достижений

На этой странице в первом столбце таблицы также располагаются кнопки редактирования данных, при нажатии на которых, осуществляется переход на страницу 5 «Достижение» (рис. 4.24.) с формой для этих данных.

Главная > Студент > Список достижений > Достижение

Семестр

\*Цикл предмета

Балл в семестре

Балл за экзамен

Своевременность сдачи

Количество пропусков

Оценка

Рис. 4.24. Страница 5. Достижение

С данной страницы можно вернуться на любую предыдущую страницу, воспользовавшись путем вверх (рис. 4.25)

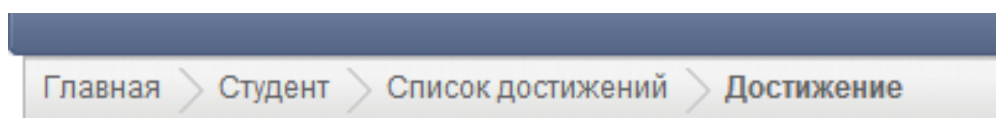


Рис. 4.25. Путь на странице 5

Аналогичным образом, меняя значения в выпадающем списке на странице 3 «Список студентов», пользователю выдается информация, соответствующая запросу.

Теперь рассмотрим подробнее функциональность второго блока главной страницы – блока «Оценка» (рис. 4.26) . Как уже было упомянуто ранее, в этом блоке определяются параметры, по которым формируется рейтинг студентов или классификация в соответствующем подразделении, определяемым в выпадающих списках этого блока.

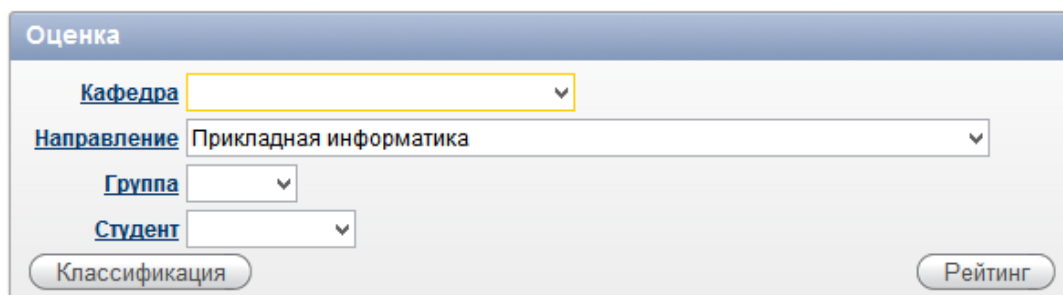
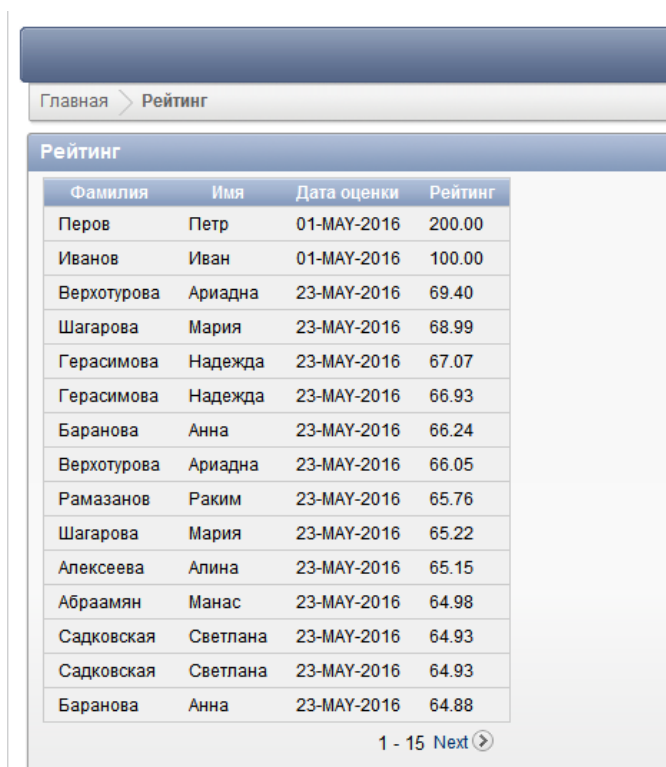


Рис. 4.26. Блок оценки

При нажатии на кнопку «рейтинг» внизу, приложение осуществляет переход на страницу 6 – «Рейтинг» (рис. 4.27). На данной странице отображается список студентов, сортированный по интегральному показателю. Интегральный показатель определяется при помощи регрессионного дерева, описанного в главе 4.3.

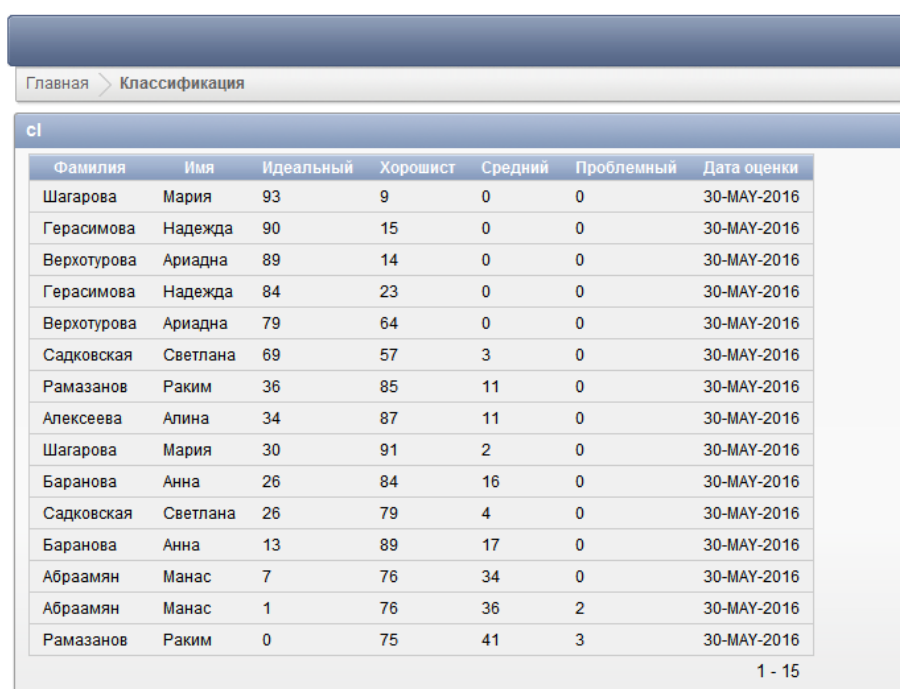


Фамилия	Имя	Дата оценки	Рейтинг
Перов	Петр	01-MAY-2016	200.00
Иванов	Иван	01-MAY-2016	100.00
Верхотурова	Ариадна	23-MAY-2016	69.40
Шагарова	Мария	23-MAY-2016	68.99
Герасимова	Надежда	23-MAY-2016	67.07
Герасимова	Надежда	23-MAY-2016	66.93
Баранова	Анна	23-MAY-2016	66.24
Верхотурова	Ариадна	23-MAY-2016	66.05
Рамазанов	Рахим	23-MAY-2016	65.76
Шагарова	Мария	23-MAY-2016	65.22
Алексеева	Алина	23-MAY-2016	65.15
Абраамян	Манас	23-MAY-2016	64.98
Садковская	Светлана	23-MAY-2016	64.93
Садковская	Светлана	23-MAY-2016	64.93
Баранова	Анна	23-MAY-2016	64.88

Рис. 4.27. Страница 6. Рейтинг

Со страницы 6 возможно как вернуться на главную, так и перейти на страницу просмотра студента 4, нажав на иконку редактирования слева.

При нажатии на кнопку «Классификация» внизу блока оценки на главной странице, попадаем на страницу 7 – «Классификация» (рис. 4.28). На данной странице выводятся все студенты указанного подразделения, а так же их классовое разбиение с датой оценки. Сортировка нисходящая по степени принадлежности к классам, начиная с лучшего.



The screenshot shows a web interface with a breadcrumb trail 'Главная > Классификация'. Below it is a table with the following data:

Фамилия	Имя	Идеальный	Хорошист	Средний	Проблемный	Дата оценки
Шагарова	Мария	93	9	0	0	30-MAY-2016
Герасимова	Надежда	90	15	0	0	30-MAY-2016
Верхотурова	Ариадна	89	14	0	0	30-MAY-2016
Герасимова	Надежда	84	23	0	0	30-MAY-2016
Верхотурова	Ариадна	79	64	0	0	30-MAY-2016
Садковская	Светлана	69	57	3	0	30-MAY-2016
Рамазанов	Рахим	36	85	11	0	30-MAY-2016
Алексеева	Алина	34	87	11	0	30-MAY-2016
Шагарова	Мария	30	91	2	0	30-MAY-2016
Баранова	Анна	26	84	16	0	30-MAY-2016
Садковская	Светлана	26	79	4	0	30-MAY-2016
Баранова	Анна	13	89	17	0	30-MAY-2016
Абраамян	Манас	7	76	34	0	30-MAY-2016
Абраамян	Манас	1	76	36	2	30-MAY-2016
Рамазанов	Рахим	0	75	41	3	30-MAY-2016

1 - 15

Рис. 4.28. Страница 7. Классификация

#### 4.6. Обеспечение функциональности поддержки принятия решений

Получив деревья решений необходимо иметь возможность их дальнейшего использования для анализа новых данных и представления их через интерфейс программы. Для этого создадим функции в базе данных, отвечающие за выборку и сортировку объектов согласно логике полученной базы знаний.

Реализация процедуры классификации студентов будет зависеть от их характеристик (входные переменные). В качестве результата будут выдаваться соответствующие каждому классу степени принадлежностей (выходная переменная). Также в процессе анализа новых данных, для каждого из студентов в файл будет вестись запись прохождения по базе знаний и отмечать принятые решения. В результате получаем словесное описание результатов классификации для каждого объекта.

Однако такой запрос описанной процедуры к базе данных для выполнения данного алгоритма будет слишком сложным и объемным. Для упрощения можно создать промежуточное представление в базе данных (см. приложение 3), в котором будут собраны все необходимые показатели и характеристики определенные в главе (табл. 4.1). Часть результата, выводимого таким представлением отображена в табл. 4.4.

Таблица 4.4 – Представление данных их БД для классификации и регрессии

N студента	сдан о в о в р е м я	нау ка	ГСЭ	ЕН	ОП Д	СД	ФТД	ак ти в гр у п пы	Олим п.	сп ор т	об щ. д- ть	тв ор ч е с т в о	ур ов е н ь а н г л.	Н а с. П у н к т
41	1		3,00	4,00	4,30	4,70		нет					1	1
42	0,96		4,00	4,00	4,40	4,75		нет		0,5			1	1
43	1		3,50	5,00	4,75	4,60		нет	12,80		1		0,5	1
44	1	18,30	5,00	5,00	5,00	4,80	5,00	ста рос та		1	0,5	0	1	1

45	1	23,30	5,00	5,00	5,00	4,75		нет		1			1	2
46	0,99		4,00	4,50	4,60	4,70		нет		0			2	2
47	1		5,00	5,00	4,30	4,00	4,00	нет	49,70			1	1,5	1
48	1	196,00	5,00	5,00	5,00	5,00		нет	85,40	0,5		0,5	2	1
58	0,7		4,25	3,70	4,00	3,70		пр оф орг		0,5			1,5	1
60	1			3,70	4,80	4,50		нет	2,75		1		1	2
61	1			4,50	4,60	4,90	4,88	нет		0	1		2	1
62	1			5,00	5,00	5,00	4,88	нет					1,5	2
63	0,82			3,70	3,70	3,80	4,00	нет				0,5	0,5	1
64	0,98			4,00	4,40	4,60	4,63	нет		0			1	2
65	1			4,00	4,40	4,80	5,00	нет	10,30				1	1

Теперь определяется сама процедура классификации с одним входным параметром – номером студента. Далее согласно переданному идентификатору, выбираются значения необходимых для использования дерева характеристик из созданного представления и присваиваются соответствующим переменным внутри процедуры. Далее эти значения используются при прохождении построенного дерева (при помощи конструкций «если – то – иначе»), где определяются степени принадлежности к четырем обозначенным классам. После прохождения дерева, полученные для всех классов значения заносятся в таблицу Classification в базе данных вместе с номером студента и датой.

```
CREATE or replace PROCEDURE Classify
(id IN number)
COMMENT 'процедура классификации'
```

```
BEGIN
```

```
    DECLARE c1A NUMBER DEFAULT 0, c1B NUMBER DEFAULT 0, c1C
NUMBER DEFAULT 0, c1D NUMBER DEFAULT 0, sci NUMBER, sd NUMBER,
hse NUMBER, gp NUMBER, sp NUMBER
```

```
    SELECT STUDENTPROGRESSDEMO.AVG_MARK INTO sci
    FROM STUDENTPROGRESSDEMO
    WHERE STUDENTPROGRESSDEMO.ID_STUDENT = id
    AND
    STUDENTPROGRESSDEMO.SUBJECT_TYPE_ACADEMIC_NAME = 'EH'
    SELECT STUDENTPROGRESSDEMO.AVG_MARK INTO sd
```

```

    FROM STUDENTPROGRESSDEMO
    WHERE STUDENTPROGRESSDEMO.ID_STUDENT = id
    AND
    STUDENTPROGRESSDEMO.SUBJECT_TYPE_ACADEMIC_NAME = 'СД'
    SELECT STUDENTPROGRESSDEMO.AVG_MARK INTO hse
    FROM STUDENTPROGRESSDEMO
    WHERE STUDENTPROGRESSDEMO.ID_STUDENT = id
    AND
    STUDENTPROGRESSDEMO.SUBJECT_TYPE_ACADEMIC_NAME =
'ТСЭ'
    SELECT STUDENTPROGRESSDEMO.AVG_MARK INTO gp
    FROM STUDENTPROGRESSDEMO
    WHERE STUDENTPROGRESSDEMO.ID_STUDENT = id
    AND
    STUDENTPROGRESSDEMO.SUBJECT_TYPE_ACADEMIC_NAME =
'ОПД'
    SELECT STUDENTPROGRESSDEMO.SCIACH INTO sp
    FROM STUDENTPROGRESSDEMO
    WHERE STUDENTPROGRESSDEMO.ID_STUDENT = id

IF sp<4 THEN
    IF sci<4 THEN
        clA = 0;
        IF GP<4 THEN
            clC = 2;
        ELSE    clB = 0.25;
        ENDIF
        IF GP<3 THEN
            clA = 0;
            clB = 0;
            clC = 0.2;
            clD = 0;
        ENDIF
    ELSE    clB = 1;
    ENDIF
    IF sci < 3 THEN
        clA = 0;
        clB = 0.05;
        clC = 0.4;
        clD = 0;
    ENDIF
ELSE    IF sci < 4 THEN
        clB = 5.33;
    ELSE    IF HSE < 4 THEN
            clA = 0.35;

```



```

        c1B = 2.13;
    ELSE      c1A = 3.58;
            c1B = 0.48;
    ENDIF
    IF HSE < 3 THEN
        c1A = 0.06;
        c1B = 0.39;
        c1C = 0;
        c1D = 0;
    ENDIF
    ENDIF
ENDIF
IF sp<3then
    c1A = 0;
    c1B = 0;
    c1C = 0.33;
    c1D = 0;
ENDIF

INSERT into Classification (*) value (id, date, c1A, c1B, c1C, c1D);
END;

```

После обработки функции приложение отображает полученные результаты пользователю. А также формирует информационное сообщение, основанное на виде дерева решений, в котором описываются основания для полученной классификации для каждого студента. Пример такого сообщения представлен ниже:

-----

Студент: Иванов Иван Иванович

Группа: 80M40

Результаты классификации:

A (Идеальный студент)	0,96
B (Хорошист)	0,07
C (Средний)	0
D (Проблемный)	0

Входные данные:

Научные успехи = 19,7

ГСЭ = 4

ЕН = 4,5

ОПД = 4,6

СД = 4,7

ФТД = 4,2

Т.к. средний балл по специальным дисциплинам – высокий,  
средний балл по естественнонаучным дисциплинам – высокий,  
средний балл по естественнонаучным дисциплинам – средний,  
то:

- принадлежность классу «Идеальный студент» = 0,96
- принадлежность классу «Хорошист» = 0,07
- принадлежность классу «Средний» = 0
- принадлежность классу «Проблемный» = 0

-----

Данное сообщение носит исключительно информационный характер для лица, принимающего решения и эти данные не записываются в базу, однако при необходимости его можно сохранить как текстовый документ, иначе информация будет потеряны.

Обеспечение функциональности блока «рейтинг» происходит аналогично с внедрением функции, обеспечивающей прохождение по регрессионному дереву (глава 4.3). Единственное отличие в том, что процедура вычисляет единственный показатель (а не 4) и записывает его в таблицу Integral Index вместе с датой оценки и номером студента.

Таким образом в приложении полностью реализованы заявленные характеристики по манипуляциям с данными и применению математических методов для построения рейтинга и классификации студентов согласно из показателям.

## 5. Результаты проведенного исследования

В ходе выполнения работ в рамках данной работы было проделано следующее:

- На основе проведенного обзора литературы по предметной области выбраны следующие методы для решения поставленных задач:
  - построение онтологии предметной области для отображения предметной области;
  - построение базы знаний на основе нечетких деревьев решений
  - создание web-приложения на основе MVC паттерна проектирования для диалога системы с пользователем базой данных
- Проведен сравнительный обзор существующих программных средств и выбраны конкретные приложения
- Построена модель, отражающая отношения между исходными объектами и их свойствами.
- Спроектирована база данных, отражающая все необходимые для решения задачи характеристики объектов
- Выделены задачи ИС, отображены на диаграмме вариантов использования.
- Определена структурная схема ИС с применением паттерна проектирования MVC. Разделение информационных потоков на 3 отдельных компонента позволит отделить логику представления от логики приложения и уменьшить сложность конструкции.
- Проведен сравнительный анализ методов data mining и в качестве модели базы знаний выбрана структура нечетких деревьев решений, так как они способны решать поставленные задачи классификации и регрессии, имеют высокую скорость обработки данных и предоставляют заключение в доступном для понимания любому пользователю виде. А учитывая наличие слабо структурированных данных, введение нечеткой логики необходимо для повышения точности классификации.
- Определены и описаны переменные, в том числе лингвистические, использованные в задаче классификации, их диапазоны.
- Построены деревья решений для классификации и регрессии на обучающей выборке и проверены на тестовой выборке
- Проведена классификация остальных объектов из базы данных
- Спроектирована web-оболочка для СППР, при помощи которой, авторизованный пользователь может управлять данными и получать результаты классификации и рейтинг студентов.

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8БМ41	Пискуновой Татьяне Александровне

<b>Институт</b>	<b>кибернетики</b>	<b>Кафедра</b>	<b>ПМ</b>
<b>Уровень образования</b>	магистр	<b>Направление</b>	Прикладная математика и информатика

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	...
2. <i>Нормы и нормативы расходования ресурсов</i>	...
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	...

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. <i>Оценка коммерческого и инновационного потенциала НТИ</i>	...
2. <i>Разработка устава научно-технического проекта</i>	...
3. <i>Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок</i>	...
4. <i>Определение ресурсной, финансовой, экономической эффективности</i>	...

**Перечень графического материала (с точным указанием обязательных чертежей):**

1. <i>«Портрет» потребителя результатов НТИ</i>
2. <i>Сегментирование рынка</i>
3. <i>Оценка конкурентоспособности технических решений</i>
4. <i>Диаграмма FAST</i>
5. <i>Матрица SWOT</i>
6. <i>График проведения и бюджет НТИ</i>
7. <i>Оценка ресурсной, финансовой и экономической эффективности НТИ</i>
8. <i>Потенциальные риски</i>

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
доцент	Конотопский В.Ю.	К.Э.Н.		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8БМ41	Пискунова Татьяна Александровна		

## 6. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

### 6.1. Организация и планирование работ

При организации процесса реализации конкретного проекта необходимо рационально планировать занятость каждого из его участников и сроки проведения отдельных работ. Это можно сделать при помощи линейного графика работ. Для его построения сначала определим полный перечень проводимых работ, их продолжительность и исполнителей. Полученные данные сведены в таблице 6.1.

Таблица 6.1 – Перечень работ и продолжительность их выполнения

Этапы работы	Исполнители	Загрузка исполнителя НР, %	Загрузка исполнителя И, %
Постановка целей и задач, получение исходных данных	НР	100	
Составление и утверждение ТЗ	НР, И	100	10
Подбор и изучение материалов по тематике	НР, И	25	100
Разработка календарного плана	НР, И	100	10
Обсуждение литературы	НР, И	30	100
Выбор структурной схемы системы и используемых математических методов	НР, И	70	100
Выбор методов и программных средств реализации	НР, И	20	100
Проведение расчетов и валидация модели	И		100
Создание программного продукта	И		100
Оформление расчетно-пояснительной записки	И		100
Подведение итогов	НР, И	45	100

### 6.2. Продолжительность этапов работ

Расчет продолжительности этапов работ определены опытно-статистическим экспертным методом.

Определим ожидаемое время проведения работ, длительность этапов в рабочих и календарных днях, по формулам:

$$t_{ож} = \frac{3 \cdot t_{\min} + 2 \cdot t_{\max}}{5}$$

где  $t_{\min}$  – минимальная продолжительность работы, дн.;

$t_{\max}$  – максимальная продолжительность работы, дн.;

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} \cdot K_{Д}$$

где  $t_{ож}$  – продолжительность работы, дн.;

$K_{ВН}$  – коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определенных длительностей, в частности, возможно  $K_{ВН} = 1$ ;

$K_{Д}$  – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ ( $K_{Д} = 1-1,2$ ).

Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{КД} = T_{РД} \cdot T_{К}$$

где  $T_{КД}$  – продолжительность выполнения этапа в календарных днях;

Возьмем  $K_{Д} = 1,1$ .

$T_{К}$  – коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях, и рассчитываемый по формуле

$$T_{К6} = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}} = \frac{365}{365 - 52 - 10} = 1,205$$

где  $T_{КАЛ}$  – календарные дни ( $T_{КАЛ} = 365$ );

$T_{ВД}$  – выходные дни ( $T_{ВД} = 52$ );

$T_{ПД}$  – праздничные дни ( $T_{ПД} = 10$ ).

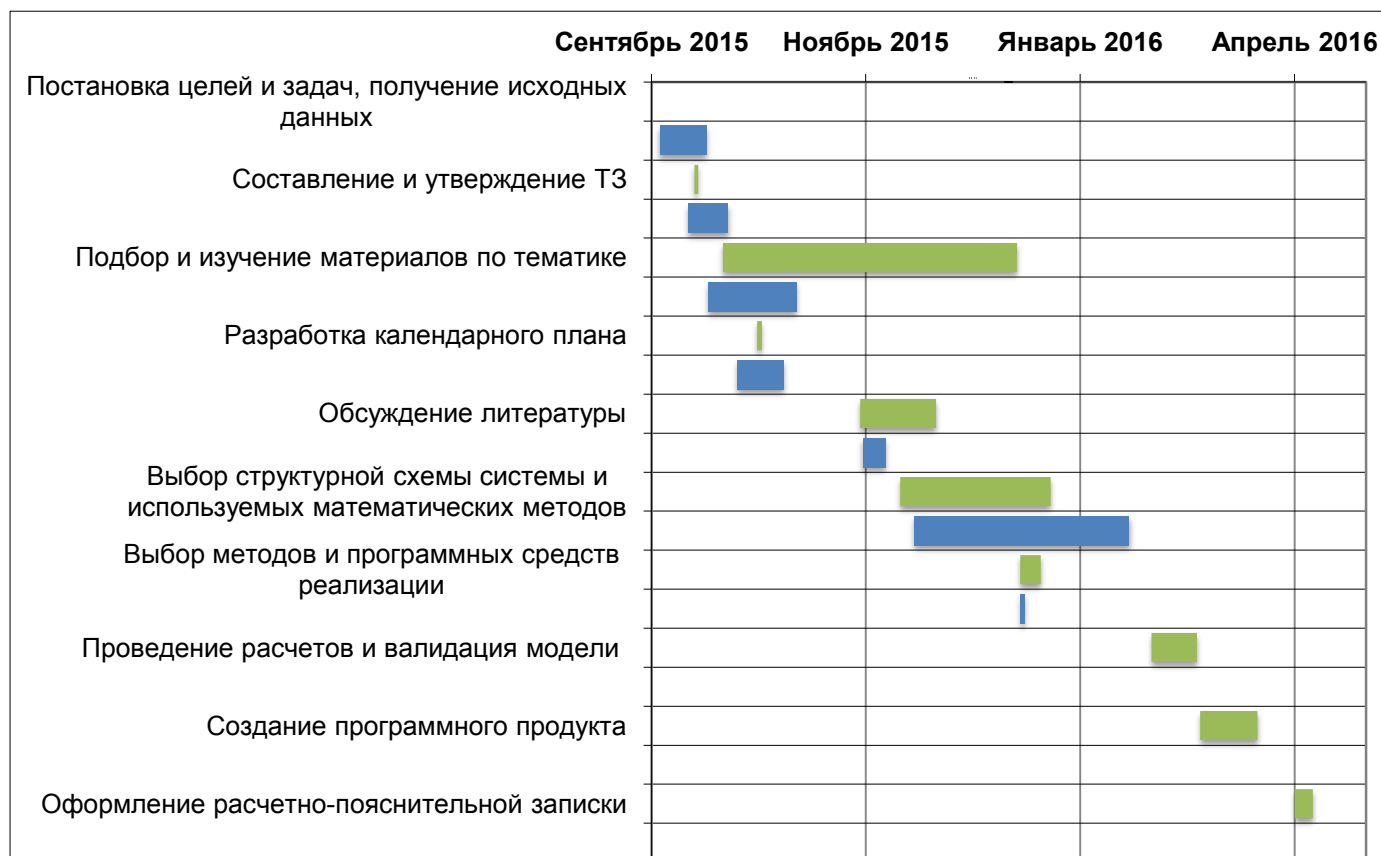
В таблице 6.2 приведены продолжительности этапов работ и их трудоемкости по исполнителям, занятым на каждом этапе.

Таблица 6.2 – Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.- дн.			
		t <sub>min</sub>	t <sub>max</sub>	t <sub>ож</sub>	Т <sub>рд</sub>		Т <sub>кд</sub>	
					НР	И	НР	И
1	2	3	4	5	6	7	8	9
Постановка целей и задач, получение исходных данных	НР	2	3	2,4	2,64	0,00	3,18	0,00
Составление и утверждение ТЗ	НР, И	1	3	1,8	1,98	0,20	2,39	0,24
Подбор и изучение материалов по тематике	НР, И	15	22	17,8	4,90	19,58	5,90	23,59
Разработка календарного плана	НР, И	2	3	2,4	2,64	0,26	3,18	0,32
Обсуждение литературы	НР, И	3	6	4,2	1,39	4,62	1,67	5,57
Выбор структурной схемы системы и используемых математических методов	НР, И	8	15	10,8	8,32	11,88	10,02	14,32
Выбор методов и программных средств реализации	НР, И	5	9	6,6	1,45	7,26	1,75	8,75
Проведение расчетов и валидация модели	И	12	18	14,4	0,00	15,84	0,00	19,09
Создание программного продукта	И	15	23	18,2	0,00	20,02	0,00	24,12
Оформление расчетно-пояснительной записки	И	5	7	5,8	0,00	6,38	0,00	7,69
Подведение итогов	НР, И	3	6	4,2	2,08	4,62	2,51	5,57
<b>Итого:</b>				<b>88,6</b>	<b>25,39</b>	<b>90,66</b>	<b>30,59</b>	<b>109,25</b>

Величины трудоемкости этапов по исполнителям Т<sub>кд</sub> позволяют построить линейный график осуществления проекта (табл. 6.3).

Таблица 6.3 – Линейный график работ



### 6.1.2 Расчет накопления готовности проекта

Оценка текущих состояний (результатов) работы над проектом определяется согласно формулам ниже и отражена в табл. 6.4.

- $TP_{\text{общ}}$  – общая трудоемкость проекта;
- $TP_i$  ( $TP_k$ ) – трудоемкость  $i$ -го ( $k$ -го) этапа проекта,  $i = \overline{1, I}$ ;
- $TP_i^H$  – накопленная трудоемкость  $i$ -го этапа проекта по его завершении;
- $TP_{ij}$  ( $TP_{kj}$ ) – трудоемкость работ, выполняемых  $j$ -м участником на  $i$ -м этапе, здесь  $j = \overline{1, m}$  – индекс исполнителя, Степень готовности определяется формулой:

$$CG_i = \frac{TP_i^H}{TP_{\text{общ.}}} = \frac{\sum_{k=1}^i TP_k}{TP_{\text{общ.}}} = \frac{\sum_{k=1}^i \sum_{j=1}^m TP_{km}}{\sum_{k=1}^I \sum_{j=1}^m TP_{km}}$$



Таблица 6.4 – Нарастание технической готовности работы и удельный вес каждого этапа

Этап	ТР <sub>i</sub> , %	СГ <sub>i</sub> , %
Постановка целей и задач, получение исходных данных	2,83	2,83
Составление и утверждение ТЗ	2,74	5,57
Подбор и изучение материалов по тематике	19,01	24,58
Разработка календарного плана	3,02	27,60
Обсуждение литературы	5,63	33,23
Выбор структурной схемы системы и используемых математических методов	12,53	45,76
Выбор методов и программных средств реализации	10,17	55,93
Проведение расчетов и валидация модели	10,60	66,53
Создание программного продукта	17,69	84,22
Оформление расчетно-пояснительной записки	5,78	90,00
Подведение итогов	10,00	100,00

## 6.2 Расчет сметы затрат на выполнение проекта

Затраты на создание проекта включают все расходы, необходимые для реализации комплекса работ, составляющих содержание данной разработки. Расчет сметной стоимости ее выполнения производится по следующим статьям затрат: материалы и покупные изделия; заработная плата; социальный налог; расходы на электроэнергию (без освещения); амортизационные отчисления; командировочные расходы; оплата услуг связи; арендная плата за пользование имуществом; прочие услуги (сторонних организаций); прочие (накладные расходы) расходы.

### 6.2.1 Расчет затрат на материалы

Так как для написания ВКР не требовалась покупка какого-либо материального оборудования и лицензий на ПО (использовались оборудование и лицензии университета), то к данной статье расходов можно отнести только расходы на распечатку материалов (прим 300 листов за время выполнения проекта), ручки, блокноты и ТЗР, см. табл. 6.5.

Таблица 6.5 – Расчет затрат на материалы

Наименование материалов	Цена за ед., руб.	Кол-во	ед	Сумма, руб.
Распечатка листов А4	2,5	300	шт	750
Ручка	17,5	10	шт	175
блокнот	33,8	5	шт	169
ТЗР		10	%	109,4
<b>Итого:</b>				<b>1203,4</b>

### 6.2.2 Расчет заработной платы

Данная статья расходов включает заработную плату научного руководителя и инженера (в его роли выступает исполнитель проекта), а также премии, входящие в фонд заработной платы. Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя. Величины месячных окладов по нормам ТПУ для научного руководителя принимается равным 33 162,87р., а для студента-исполнителя – 14 874,45р.

Среднедневная тарифная заработная плата ( $ЗП_{дн-т}$ ) рассчитывается по формуле, учитывающей, что в году 298 рабочих дней и, следовательно, в месяце в среднем 24,83 рабочих дня при шестидневной рабочей неделе:

$$ЗП_{дн-т} = MO/24,83$$

Расчеты полной заработной платы для обоих участников проекта, с учетом ряда коэффициентов ( $K_{ГР} = 1,1$ ;  $K_{доп.ЗП} = 1,188$ ;  $K_p = 1,3$ ), приведены в таблице 6.6.

Таблица 6.6 – Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./раб.день	Затраты времени, раб.дни	Коэффициент	Фонд з/платы, руб.
НР	33 162,87	1617,70	26,00	1,59	71 453,59
И	14 874	599,05	91,00	1,70	92 610,04
<b>Итого:</b>					<b>164 063,63</b>

### 6.2.3 Расчет затрат на социальный налог

Затраты на единый социальный налог (ЕСН) включают отчисления в пенсионный фонд, социальное и медицинское страхование, и составляют 30 % от полной заработной платы по проекту (табл. 6.7):

Таблица 6.7 – Затраты на ЕСН

Исполнитель	ЕСН
НР	21 436,08
И	27 783,01
<b>Итого:</b>	<b>49 219,09</b>

### 6.2.4 Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию, потраченную в ходе выполнения проекта на работу используемого оборудования, рассчитываемые по формуле:

$$C_{\text{эл.об.}} = P_{\text{об}} \cdot t_{\text{об}} \cdot Ц_{\text{Э}}$$

где  $P_{\text{об}}$  – мощность, потребляемая оборудованием, кВт;

$Ц_{\text{Э}}$  – тариф на 1 кВт·час; Для ТПУ  $Ц_{\text{Э}} = 5,257$  руб./кВт·час (с НДС).

$t_{\text{об}}$  – время работы оборудования, час.

$$t_{\text{об}} = T_{\text{рд}} * K_t,$$

где  $K_t \leq 1$  – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к  $T_{\text{рд}}$ , определяется исполнителем самостоятельно. В ряде случаев возможно определение  $t_{\text{об}}$  путем прямого учета, особенно при ограниченном использовании соответствующего оборудования.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{\text{об}} = P_{\text{ном.}} * K_C$$

где  $P_{\text{ном.}}$  – номинальная мощность оборудования, кВт;

$K_C \leq 1$  – коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности  $K_C = 1$ .

Пример расчета затраты на электроэнергию для технологических целей приведен в таблице 6.8.

Таблица 6.8 – Затраты на электроэнергию технологическую

Наименование оборудования	Кт	Время работы оборудования $t_{OB}$ , час	Потребляемая мощность $P_{OB}$ , кВт	Затраты $\Delta_{OB}$ , руб.
Персональный компьютер	0,9	655,2	0,3	1033,32
принтер	0,01	7,28	0,1	3,83
<b>Итого:</b>				<b>1037,14</b>

### 6.2.5 Расчет амортизационных расходов

Для расчета амортизации используемого оборудования используется формула:

$$C_{AM} = \frac{N_A * C_{OB} * t_{рф} * n}{F_d},$$

где  $N_A$  – годовая норма амортизации единицы оборудования;

$C_{OB}$  – балансовая стоимость единицы оборудования с учетом ТЗР.

$F_d$  – действительный годовой фонд времени работы соответствующего оборудования

$t_{рф}$  – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта;

$n$  – число задействованных однотипных единиц оборудования.

Таблица 6.9 – Амортизационные затраты

Наименование оборудования	год фонд врем $F_d$	Факт Время работы оборудования $t_{рф}$ , час	$N_a$	$C_{об}$	$C_{ам}$
Персональный компьютер	2384	655,2	0,33	44000,00	4030,87
МФУ	2384	7,28	0,40	9350,00	11,42
<b>Итого:</b>					<b>4042,29</b>

### 6.2.6 Расчет расходов, учитываемых непосредственно на основе платежных (расчетных) документов (кроме суточных)

Непосредственно учитываемые расходы отсутствуют.

### 6.2.7 Расчет прочих расходов

Здесь, неучтенные в предыдущих статьях расходы на выполнение проекта, принимаем равными 10% от суммы всех предыдущих расходов, т.е.

$$C_{\text{проч.}} = (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{ам}} + C_{\text{нп}}) \cdot 0,1$$

$$C_{\text{проч}} \quad \quad \quad 28537,989$$

### 6.2.8 Расчет общей себестоимости разработки

Определим общую себестоимость. Табл 6.10

Таблица 6.10 – Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	1203,4
Основная заработная плата	$C_{\text{зп}}$	164 063,63
Отчисления в социальные фонды	$C_{\text{соц}}$	49 219,09
Расходы на электроэнергию	$C_{\text{эл.}}$	1037,14
Амортизационные отчисления	$C_{\text{ам}}$	4042,29
Непосредственно учитываемые расходы	$C_{\text{нр}}$	0
Прочие расходы	$C_{\text{проч}}$	21956,556
<b>Итого:</b>		<b>241 522,11</b>

### 6.2.9 Расчет прибыли

Прибыль от реализации проекта принимается в размере 15 % от полной себестоимости проекта и равна 24152,21.

### 6.2.10 Расчет НДС

НДС составляет 18% от суммы затрат на разработку и прибыли, а именно 47821,38.

### 6.2.11 Цена разработки НИР

Цена равна сумме полной себестоимости, прибыли и НДС: 313495,70.

### **6.3 Оценка экономической эффективности проекта**

В связи с отсутствием достоверных данных и сложностью их получения, нецелесообразно проводить количественную оценку экономической эффективности проекта. Однако отразим его значимость при помощи качественных показателей, отражающихся в экономической и социальный эффект.

В ходе выполнения ВКР разрабатывается информационная система поддержки принятия решений в виде веб-приложения с математическим аппаратом обработки данных. Данная ИСППР будет применяться для анализа и обработки данных об учебной, научной, творческой, спортивной и общественной деятельности обучающихся университета, вместе с оценкой их личностных и психологических характеристик.

Комплексная информация о студентах предоставит возможность проводить качественный и количественный анализ их успеваемости, научного потенциала и общей мотивации к обучению. Это позволит увидеть тенденции развития/деградации отдельно взятых студентов, а также общую динамику по подразделениям; определять дальнейшую индивидуальную траекторию для студентов согласно их текущей деятельности (например, стоит ли данному обучающемуся поступать в магистратуру, т.к. он стремится заниматься наукой и теоретическими исследованиями, или же он более подходит для профессиональной деятельности на предприятиях города). Такая информационная поддержка позволит выпускникам лучше адаптироваться к дальнейшей деятельности (и не терять еще 2 года в магистратуре), а также увеличить качество образования (за счет улучшенного отбора).

Повышение качества принятия решения приведет уменьшение затрат на обучение студентов, ошибочно принятых в магистратуру или аспирантуру, и в итоге не закончивших обучение. Также, так как сократится нагрузка на учебно-вспомогательный персонал, возможно будет сократить расходы на выполнение функции сбора и обработки данных для принятия решений об отчислении, принятии для последующего обучения или на работу.

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8БМ41	Пискуновой Татьяне Александровне

<b>Институт</b>	<b>ИК</b>	<b>Кафедра</b>	<b>ПМ</b>
Уровень образования	магистратура	Направление/специальность	Прикладная математика и информатика

**Исходные данные к разделу «Социальная ответственность»:**

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является учебная и научная деятельность студентов в процессе обучения в университете, а также из личностные и психологические характеристики.
--	---

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

<b>1. Производственная безопасность</b> 1.1. Анализ выявленных вредных факторов при разработке и эксплуатации ИСППР.	Отклонение параметров микроклимата повышенный уровень электромагнитных излучений Отклонение параметров освещенности Нервно-психологическое напряжение
1.2. Анализ выявленных опасных факторов при разработке и эксплуатации ИСППР.	Повышенное значение напряжения в электрической цепи, статического электричества
<b>2. Экологическая безопасность:</b>	Анализ воздействия объекта на литосферу, утилизация
<b>3. Безопасность в чрезвычайных ситуациях:</b>	Возникновение пожара
<b>4. Правовые и организационные вопросы обеспечения безопасности:</b>	Организация рабочего места

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
доцент	Анищенко Ю.В.	к.т.н.		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8БМ41	Пискунова Татьяна Александровна		

## 7. Социальная ответственность

### 7.1. Введение

В ходе выполнения ВКР разрабатывается информационная система поддержки принятия решений в виде веб-приложения с математическим аппаратом обработки данных. Данная ИСППР будет применяться для анализа и обработки данных об учебной, научной, творческой, спортивной и общественной деятельности обучающихся университета, вместе с оценкой их личностных и психологических характеристик. Пользователями системы являются работники университета. Пользователи не обязаны обладать специальными знаниями, навыками или разрешениями для работы с данным ПО. Разработка и эксплуатация осуществляется на ЭВМ.

### 7.2. Производственная безопасность

В процессе разработки и эксплуатации ИСППР при выполнении работ на персональном компьютере (ПК) согласно ГОСТу 12.0.003-74 “ССБТ. Опасные и вредные производственные факторы. Классификация” могут иметь место следующие факторы (Табл. 7.1)

Таблица 7.1 – Опасные и вредные факторы при проектировании ИСППР

Источник фактора, наименование видов работ	Факторы (по ГОСТ 12.0.003-74.)		Нормативные документы
	Вредные	Опасные	
Работа за ПЭВМ	Повышенный уровень электромагнитных излучений  Отклонение параметров освещенности	Повышенное значение напряжения в электрической цепи, замыкание;	ГОСТ 12.1.038–82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов.  ГОСТ Р 12.1.019-2009 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты  СанПиН 2.2.2/2.4.1340–03. Санитарно-эпидемиологические правила и нормативы «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».



			ГОСТ Р 50948-2001 Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности
--	--	--	--

### **7.7.1. Повышенное значение напряжения в электрической цепи, замыкание;**

Степень опасного и вредного воздействия на человека электрического тока, электрической дуги и электромагнитных полей зависит от:

- рода и величины напряжения и тока;
- частоты электрического тока;
- пути тока через тело человека;
- продолжительности воздействия электрического тока или электромагнитного поля на организм человека;
- условий внешней среды.

Нормы на допустимые токи и напряжения прикосновения в электроустановках должны устанавливаться в соответствии с предельно допустимыми уровнями воздействия на человека токов и напряжений прикосновения и утверждаться в установленном порядке.

Электробезопасность обеспечивается конструкцией ПЭВМ. Требования безопасности при пользовании электроустановками бытового назначения содержатся в прилагаемых к ним инструкциях по эксплуатации предприятий-изготовителей.

Для обеспечения защиты от случайного прикосновения к токоведущим частям применяются следующие способы и средства:

- защитные оболочки;
- безопасное расположение токоведущих частей.
- изоляция токоведущих частей (основная, дополнительная, усиленная, двойная);

- малое напряжение;
- защитное отключение;

### **7.7.2. Отклонение параметров освещенности**

Помещения для эксплуатации ПЭВМ должны иметь естественное и искусственное освещение. Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Оконные проемы должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.

Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк.

Степень несведения цветов в любом месте многоцветного экрана для дисплеев должна быть не более 3,4' при проектном расстоянии наблюдения. Проектное расстояние наблюдения принимают равным 50 см для дисплеев с размером экрана по диагонали 14"-17" и 75 см - для экранов 19"-21".

Временная нестабильность изображения (мелькания) для дисплеев плоских дискретных экранах не должна быть зафиксирована, частота обновления изображения должна быть не менее 60. Амплитуда смещения

изображения (пространственная нестабильность изображения - дрожание) должна быть не более  $2 \cdot 10^{-4}l$ , где  $l$ - проектное расстояние наблюдения, мм.

Изменение размеров однотипных знаков по рабочему полю должно быть в пределах  $\pm 5\%$  высоты знака. Максимальная разность длин строк текста на рабочем поле должна быть не более 2% средней длины строки. Максимальная разность длин столбцов текста на рабочем поле должна быть не более 2% средней длины столбца.

При необходимости распознавания или идентификации цветовых параметров прикладная программа должна предлагать устанавливаемый по умолчанию набор цветов (4.2.9-4.2.11 ГОСТ Р 50948-2001)., высота символа должна быть не менее 20' при проектном расстоянии наблюдения, угловой размер изображения должен быть не менее 30' при проектном расстоянии наблюдения (предпочтительно - 40'). Следует избегать применения насыщенного синего цвета для изображений, имеющих угловой размер менее 2°.

Яркость знака должна быть не менее 20 кд/м. Неравномерность яркости рабочего поля экрана должна быть не более 20%. Неравномерность яркости элементов знака должна быть не более 20%. Яркостный контраст изображения должен быть не менее 3:1 при угле наблюдения от минус 40° до плюс 40°. Яркостный контраст внутри знака и между знаками должен быть не менее 3:1.

Ширина контура знака должна быть от 0,25 до 0,5 мм.

### **7.7.3. Повышенный уровень электромагнитных излучений**

Напряженность электрической составляющей переменного электромагнитного поля дисплея должна быть не более:

25 В/м - в диапазоне частот от 5 Гц до 2 кГц;

2,5 В/м - в диапазоне частот от 2 до 400 кГц в точках, имеющих координаты 0°, 90°, 180°, 270° на расстоянии 0,4 м от центра клавиатуры портативного компьютера.

Плотность магнитного потока должна быть не более:

250 нТл - в диапазоне частот от 5 Гц до 2 кГц;

25 нТл - в диапазоне частот от 2 до 400 кГц.

Плотность магнитного потока переменного электромагнитного поля дисплея портативного компьютера устанавливаются для обоих диапазонов частот в 48 точках на расстоянии + 0,4 м от центра клавиатуры портативного компьютера.

### **7.3. Экологическая безопасность**

При разработке и эксплуатации ИСППР не происходит вредных выбросов в атмосферу, сбросов в гидросферу и отходов. Однако работа происходит на ЭВМ, некоторые компоненты которых нуждаются в утилизации и переработке после прекращения эксплуатации.

Компьютеры относятся к отходам 3 класса опасности – их неправильная утилизация наносит вред экологии. Подобные действия могут привести к определенным проблемам, ведь это запрещено законодательством. Федеральный закон от 24.06.1998 № 89-ФЗ «Об отходах производства и потребления».

Компьютерная и офисная техника является сырьем строгой отчетности, их учет и переработка осуществляются в соответствии с Федеральным законом от 26.03.1998 № 41-ФЗ «О драгоценных металлах и драгоценных камнях» и Инструкцией Министерства финансов Российской Федерации «О порядке учета и хранения драгоценных металлов, драгоценных камней, продукции из них и ведения отчетности при их производстве, использовании и обращении».

Объекты, числящиеся за предприятием, должны списываться и утилизироваться согласно определенным правилам.

Утилизация включает в себя демонтаж, разборку технических узлов, разделение и сортировку конструкционных материалов, сбор и сортировку электронного лома согласно действующей нормативно-технической документации по переработке драгосодержащих материалов и изделий. Полученные в результате демонтажа материалы проходят физико-химическую переработку до принятых стандартов конкретно по каждому виду вторсырья и далее отправляются на заводы переработчики. Непосредственно электронный лом после специальной подготовки направляется на извлечение цветных и драгоценных металлов. Последние после аффинажа сдаются в Госфонд.

#### **7.4.Безопасность в чрезвычайных ситуациях.**

К ЧС при работе с ИСППР на ПК можно отнести возможность возникновения и распространение пожара, так как в современных ЭВМ очень высокая плотность размещения элементов электронных схем, в непосредственной близости друг от друга располагаются соединительные провода и кабели, при протекании по ним электрического тока выделяется значительное количество теплоты, при этом возможно оплавление изоляции и возникновение возгорания. Возникновение других видов ЧС – маловероятно.

Для предотвращения возникновения данного ЧС необходимо следовать технике безопасности при работе с ЭВМ.

В случае возникновения пожара люди должны покинуть помещение в течение минимального времени согласно плану эвакуации.

В помещениях с компьютерной техникой, недопустимо применение воды и пены ввиду опасности повреждения или полного выхода из строя дорогостоящего электронного оборудования.

Для тушения пожаров необходимо применять углекислотные и порошковые огнетушители, которые обладают высокой скоростью тушения, большим временем действия, возможностью тушения электроустановок, высокой эффективностью борьбы с огнем. Воду разрешено применять только во вспомогательных помещениях.

Помещение должно быть оборудовано пожарными извещателями, которые позволяют оповестить дежурный персонал о пожаре.

При обнаружении пожара или признаков горения (задымление, запах гари, повышение температуры и т.п.) сотрудники, лица профессорско-преподавательского состава, обучающиеся, арендаторы, обязаны:

- незамедлительно сообщить об этом по телефону 01 в пожарную охрану (при этом необходимо назвать адрес объекта, место возникновения пожара, а также сообщить свою фамилию), а также оперативному дежурному, или на ближайший пост охраны;
- принять по возможности меры по эвакуации людей, тушению пожара и сохранности материальных ценностей.

#### **7.5. Правовые и организационные вопросы обеспечения безопасности.**

СанПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы" регулирует нормы организации рабочего места при работе с ПЭВМ.

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 - 2,0 м.

Экран монитора должен находиться от глаз пользователя на расстоянии 600 - 700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Поверхность рабочего стола должна иметь коэффициент отражения 0,5 - 0,7.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ.

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

Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

## Заключение

В заключении следует отметить, что в процессе создании проектируемого решения все поставленные цели и задачи были выполнены в полном объеме. В качестве математического метода, положенного в основу информационной системы поддержки принятия решений был выбран метод, являющийся сочетанием деревьев решений и нечеткого логического вывода, т.к. это предоставляет возможность решать задачи со слабоструктурированными данными. Вместо количества примеров конкретного узла нечеткое дерево решений группирует их степень принадлежности, и алгоритм выдает для нового объекта не только класс, но и степень принадлежности к нему. Метод характеризуется высокой точностью классификации, быстрым процессом обучения, а результат прост для интерпретации.

Получено формализованное описание предметной области в виде онтологической модели данных.

Для классов, определяющих студентов, структуру университета и руководство были определены их иерархии и связи между объектами. Созданные экземпляры классов обладают соответствующими им свойствами и характеристиками. Построенный граф, отображающий все структурные связи в онтологии между классами и их экземплярами, использован в качестве модели для выделения основных характеристик – состава информационной базы для оценки учебного и научного потенциала студентов ВУЗа. К таким характеристикам отнесены успеваемость по предметам в разных циклах, научные и учебные достижения. Также для анализа влияния были отобраны менее очевидные факторы, такие как участие в спортивной, творческой и общественной деятельности, тип населенного пункта, личностные и психологические характеристики студентов.

По выделенным критериям спроектирована база данных, учитывающая не только связи между объектами и их свойствами, но и возможность



отсутствия определенных данных. В качестве ядра базы данных была выбрана организация двух главных таблиц, хранящих данные о личности и студенте, обеспечивающая полноту, прозрачность и доступность всех хранимых данных. Нечеткие деревья решений обеспечивают реализацию компонента, отвечающего за классификацию и регрессию, и на их основе предоставляются решения относительно потенциала конкретного студента.

Все полученные результаты доступны при использовании программного web-интерфейса, реализующего концепт проектирования MVC и отвечающего за удобство взаимодействия пользователя с базой данных и математическим аппаратом согласно уровню доступа самого пользователя.

Внедрение разработанной системы позволит облегчить процесс обработки большого объема информации и нахождения закономерностей в них для предоставления рекомендаций пользователю. Дальнейший выбор следовать ли этим рекомендациям или нет – остается за лицом, принимающим решение. С экономической точки зрения, такая СППР позволит сократить расходы на сбор и обработку информации, а также улучшить качества отбора студентов, имея возможность анализа его текущей деятельности и прогноза его будущей успешности по настоящему учебно-научному потенциалу.

## Список публикаций

- Пискунова Т.А., Применение интеллектуального анализа данных для создания системы решающих правил, XIII Всероссийская научно-практическая конференция студентов, аспирантов и молодых ученых «Технологии Microsoft в теории и практике программирования», г. Томск, 2016
- Пискунова Т.А., Проектирование онтологий в социальной сфере и медицине, II Международная научная конференция, Информационные технологии в науке, управлении, социальной сфере и медицине, г. Томск, 2015
- Пискунова Т.А., Применение нечетких деревьев решений для оценки потенциала студентов, III Международная научная конференция, Информационные технологии в науке, управлении, социальной сфере и медицине, г. Томск, 2016
- Пискунова Т.А., Мокина Е.Е., Построение интеллектуальной системы для оценки потенциала студентов, III Международная научная конференция, Информационные технологии в науке, управлении, социальной сфере и медицине, г. Томск, 2016

## Список литературы

1. Бразовская Н.В., Бразовская О.В. Математические методы принятия управленческих решений: Учебное пособие/Алт. Госуд. Технич. Университет им. И.И. Ползунова.- Барнаул: изд. АлтГТУ, 2004. – 153 с.
2. Ивасенко А.Г. Разработка управленческих решений: учебное пособие – 2-е изд., перераб. И доп. – М.: КНОРУС, 2008 – 168 с.
3. Михненко П.А. Секреты эффективных бизнес - решений. Настольная книга для руководителей. – М.: НТ Пресс, 2007 – 288 с.
4. Парахина В.Н., Максименко Л.С., Панасенко С.В. Стратегический менеджмент: учебник. М: КНОРУС, 2006 – 234 с.
5. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес. Приёмы объектно-ориентированного проектирования. Паттерны проектирования 2001
6. Hand D., Mannila H., Smyth P. - Principles of data mining, MIT, 2001
7. В. Г. Чернов. Нечеткие деревья решений (нечеткие позиционные игры), Информационно-управляющие системы №5, 2010
8. Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д. А. Поспелова. — М.: Наука, 1986. — 312 с.
9. A fuzzy random forest, Piero Bonissone, Jos M. Cadenas, M. Carmen Garrido, R. Andrus Dhaz-Valladares, International Journal of Approximate Reasoning 51 (2010) 729–747
10. *Информатика и системы управления*, 2014., №3((41)), *Интеллектуальные системы* Ю.Н. Кульчин, А.Ю. Ким, Б.С. Ноткин А.Б. Люхтер, ПОСТРОЕНИЕ АЛГОРИТМА НЕЧЕТКОГО ДЕРЕВА РЕШЕНИЙ НА ОСНОВЕ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ ПРИ ОБРАБОТКЕ СИГНАЛОВ РВОИС
11. Решение задач многокритериального альтернативного выбора на основе геометрической проекции нечетких множеств В. Г. Чернов, Информационноуправляющие системы системный анализ 50 № 1, 2007
12. Yager R. R. Multiple — objective decision — making using fuzzy sets // Intern. J. Man – machine Studies. 1977. Vol. 9. N 4. P. 375–382.
13. Борисов А. Н., Крумберг О. А., Федоров И. П. Принятие решений на основе нечетких моделей: Примеры использования. Рига: Зинатне, 1990. 184 с.
14. Нечеткие деревья решений (нечеткие позиционные игры) В. Г. Чернов, ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ № 5, 2010
15. A new Approach for Selecting Best Resources Nodes by Using Fuzzy Decision Tree in Grid Resource Broker, Asgarali Bouyer, Mohammadbager Karimi, Mansour Jalali, Mohd Noor MD SAP International Journal of Grid and Distributed Computing
16. И.и. Мацко, о.с. Логунова, в.в. Павлов, о.с. Мацко, адаптивное нечеткое дерево принятия решений с динамической структурой для автоматизированной системы управления производством непрерывнолитой заготовки1 ISSN 2079-0031 Вестник НТУ "ХПИ", 2012, № 62 (968)
17. Бобровский Д. Введение в теорию динамических систем с дискретным временем / Д. Бобровский. – СПб.: Регулярная и хаотическая динамика, 2006. – 360 с.
18. Львовский Е.Н. Статистические методы построения эмпирических формул: Учеб. пособие для вузов / Е.Н. Львовский. – 2-е изд., перераб. и доп. – М.: Высшая школа, 1998. – 239 с.

19. *Hastie T.* The Elements of Statistical Learning: Data Mining, Inference and Prediction / *T. Hastie, R. Tibshirani, J. Friedman.* – New York: Springer Science + Business Media, 2009. – 746 с.
20. *Шахиду А.* Деревья решений – общие принципы работы / *А. Шахиду* // Режим доступа: <http://www.basegroup.ru/library/analysis/tree/description/>
21. *Quinlan J.R.* C4.5: Programs for Machine Learning / *J.R. Quinlan* // San Mateo: Morgan Kaufmann Publishers Inc., 1993. – 302 с.
22. *Janikow C.Z.* Fuzzy Decision Trees: Issues and Methods / *C.Z. Janikow* – IEEE Trans Syst Man Cybern, 1998. – P. 1 – 14.
23. *Берестнева О.Г.* Построение логических моделей с использованием деревьев решений / *О.Г. Берестнева, Е.А. Муратова* // Известия Томского политехнического университета, 2004. – Т. 307. – № 2. – С. 154 – 160.
24. *Тененев, В. А.* Решение задач классификации и аппроксимации с применением нечетких деревьев решений / *В. А. Тененев, В. И. Ворончак* // Интеллектуальные системы в про- изводстве. – 2005. – № 2. – С. 46–69.
25. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 31, NO. 2, APRIL 2001 215, A Comparative Study on Heuristic Algorithms for Generating Fuzzy Decision Trees, X.-Z. Wang, D. S. Yeung, E. C. C. Tsang,
26. *Царьков С.* Нечеткие деревья решений. Технологии анализа данных BaseGroup
27. Labs [Электрон. ресурс]. Режим доступа: [http://www.basegroup.ru/library/analysis/fuzzylogic/fuzzy\\_dtrees/](http://www.basegroup.ru/library/analysis/fuzzylogic/fuzzy_dtrees/).
28. Construction of Fuzzy Decision Tree using Expectation Maximization Algorithm, Daveedu Raju Adidela, Jaya Suma, Lavanya Devi. International Journal of Computer Science and Management Research Vol 1 Issue 3 October 2012 ISSN 2278-733X
29. Analyzing Gene Expression Data: Fuzzy Decision Tree Algorithm applied to the Classification of Cancer Data, Simone A. Ludwig, Domagoj Jakobovic and Stjepan Picek, *I.J. Modern Education and Computer Science*, 2011, 5, 18-25 Published Online August 2011 in MECS (<http://www.mecspress.org/>)
30. Андрейчиков А.В., Андрейчикова О.Н. Анализ, синтез, планирование решений в экономике. – М.: Финансы и статистика, 2002. – 368 с.: ил.
31. Жданова, Е.И. Методические указания к выполнению лабораторных работ по дисциплине «Проектирование баз данных и баз знаний» / Е.И. Жданова, Ю.В. Трошин, Р.Р. Халимов. – ПГУТИ, 2011.
32. Жиров В.Г. Графическое представление и анализ нечеткой модели логического вывода в базе знаний информационной системы. – Самара, 2010.
33. Матвеев М.Г., Свиридов А.С., Алейникова Н.А. Модели и методы искусственного интеллекта. Применение в экономике. – М.: Инфра-М, 2008. – 448 с.: ил.
34. Пономарев, А.С. Нечёткие множества в задачах автоматизированного управления и принятия решений: Учебное пособие. – Харьков, 2005.
35. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечёткие системы/ Д. Рутковская, М. Пилиньский, Л. Рутковский; пер. с польск. И. Д. Рудинского.- М.: Горячая линия-Телеком, 2004.- 452 с.: ил.

36. Штовба, С.Д. Введение в теорию нечетких множеств и нечеткую логику. Сообщество пользователей Matlab и Simulink. Режим доступа: <http://matlab.exponenta.ru>, свободный.
37. Ярушкина, Н. Г. Основы теории нечётких и гибридных систем: учеб. пособие для вузов/ Н. Г. Ярушкина.- М.: Финансы и статистика, 2004.- 320 с.: ил.
38. Яхьяева, Г. Э. Нечёткие множества и нейронные сети: учеб. пособие/ Г.Э.Яхьяева. – М.: Интернет Ун-т Информ. Технологий: БИНОМ. Лаб. знаний, 2006.- 316 с.: ил.
39. A complete fuzzy decision tree technique Cristina Olaru\*, Louis Wehenkel, Fuzzy Sets and Systems 138 (2003) 221–254
40. Литвак, Б.Г. Разработка управленческого решения: Учебник./ Б.Г. Литвак – М.: Дело, 2000.
41. Шикин, Е.В. Математические методы и модели в управлении: Учеб. пособие. / Е.В. Шикин, А.Г. Чхартишвили.–М.: Дело, 2002,–440 с.
42. Гасанов, А.З. Разработка управленческого решения. Учеб. пособие, / А.З. Гасанов.– 2004.
43. Сулицкий, В.Н. Методы статистического анализа в управлении: Учеб. пособие. / В.Н. Сулицкий. – М.: Дело, 2002. –520 с.
44. Фомин, Г.П. Методы и модели линейного программирования коммерческой деятельности: Учеб. пособие./ Г.П. Фомин– М.: Финансы и статистика, 2000.
45. Елизарова, Н.Н. Информационное обеспечение стратегического менеджмента Учебное пособие/ Н.Н. Елизарова, Б.А. Баллод.– Иван. гос. энерг. ун-т. Иваново, 2005. – 124 с.
46. Мастяева, И.Н. Математические методы и модели в логистике: Учеб. пособие/ Московский государственный университет экономики, статистики и информатики. – М.: МЭСИ, 2004.- 50 с.
47. Елманова, Н. Введение в DataMining. /Н. Елманова – М: КомпьютерПресс . 2003, № 3.
48. Деревья решений. Методы классификации и прогнозирования. <http://www.intuit.ru/department/database/datamining/9/>.
49. Классификация наблюдений с использованием иерархических деревьев решений. <http://www.ievbran.ru/kiril/Library/Book1/content391/content391.htm>
50. Вячеслав Дюк Data mining - интеллектуальный анализ данных
51. Parsaye K A Characterization of Data Mining Technologies and Processes, The Journal of Data Warehousing. 1998. № 1
52. Иванов О.В. Статистика / Учебный курс для социологов и менеджеров. Часть 1. Описательная статистика. Теоретико-вероятностные основания статистического вывода, М. 2005. 187 с
53. Мусаев А.А. Алгоритмы аналитического управления производственными процессами
54. Автоматизация в промышленности. 2004, №1, с. 30-35
55. Дюк В.А., Самойленко А.П. Data Mining: учебный курс, СПб.: Питер, 2001
56. Data Mining: Concepts, Algorithms, and Applications, Wei Wang Fall 2004, The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL
57. Беднарц Энн Новое слово в аналитике, Computerworld, 2005, №16

58. Дюк В.А., Эмануэль В.Л. Информационные технологии в медико-биологических исследованиях, Питер, 2003, 525 с
59. V. Scholkopf, G. Ratsch, K. Muller, K. Tsuda, S. Mika An Introduction to Kernel-Based Learning Algorithms, IEEE Neural Networks, 12(2):181-201, May 2001
60. Медведев В.С., Потемкин В.Г. Нейронные сети. Matlab 6, Диалог-МИФИ. 2002, 496 стр
61. Информационные технологии и программирование: Межвузовский сборник статей. Вып. 2 (7), М.: МГИУ, 2003. 62 с
62. Венкатеш Ганти, Сураджит Чаудхури, Умешвар Дайал Технология баз данных в системах поддержки принятия решений, Открытые системы, 2002, №1
63. Резник Н.И., Берестнева О.Г., Вопросы образования. Инвариантный подход. Компетентностный подход.- Томск.: Издательство ТПУ, 2009
64. Ломазов В.А., Прокушев Я.Е. Решение задачи экономического многокритериального выбора на основе метода анализа иерархий, Научные ведомости 2010 №7 выпуск 14/1
65. Азарнова Т. В., Степин В. В., Щепина И. Н. Повышение эффективности методов управления развитием персонала на основе нейросетевых моделей и нечетких экспертных технологий - Вестник ВГУ. серия: экономика и управление. 2014. № 3 2014 г.
66. Берестнева О.Г. Шевелев Г.Е.<sup>1</sup>, Фисоченко О.Н.<sup>3</sup> Алгоритмы принятия решений о компетентности студентов и молодых специалистов -Современные проблемы науки и образования №6, 2012
67. Никитина Н. Ш., Бурмистрова Е. В. Методика отбора персонала на вакансию на основе нечетких показателей - Университетское управление. 2004. № 3(31). С. 98-103.
68. Фрумкин А.А. Психологический отбор в профессиональной и образовательной деятельности.- СПб.: Речь, 2004
69. Ажмухамедов И. М. Нечеткая когнитивная модель оценки компетенций специалиста - Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика № 2 / 2011
70. Микони С.В. Теория и практика рационального выбора. – М. : Маршрут, 2004. – 462 с.
71. Кнышова Е. Н. Экономика организации : учебник / Е. Н. Кнышова, Е. Е. Панфилова. – Москва: Форум Инфра-М, 2012. – 334 с.: ил. – Профессиональное образование.
72. Бочаров В. В. Инвестиции : учебник для вузов / В. В. Бочаров. – 2-е изд. – СПб: Питер, 2009. – 381 с. – Учебник для вузов.
73. Староверова Г. С. Экономическая оценка инвестиций : учебное пособие / Г. С. Староверова, А. Ю. Медведев, И. В. Сорокина. – 2-е изд., стер. – Москва: КноРус, 2009. – 312 с
74. Несветаев Ю. А. Экономическая оценка инвестиций: учебное пособие / Ю. А. Несветаев; Московский Государственный индустриальный университет; Институт дистанционного образования. – 3-е изд., стер. – Москва: Изд-во МГИУ, 2006. – 162 с.

75. Шульмин В. А. Экономическое обоснование в дипломных проектах : учебное пособие для вузов / В. А. Шульмин, Т. С. Усынина. – Старый Оскол: ТНТ, 2012. – 192 с.
76. Мигуренко Р. А. Научно-исследовательская работа: учебно-методическое пособие / Р. А. Мигуренко; Национальный исследовательский Томский политехнический университет (ТПУ), Институт дистанционного образования (ИДО). – 2-е изд., стер. – Томск: Изд-во ТПУ, 2010. – 184 с.
77. Белов, Сергей Викторович. Безопасность жизнедеятельности и защита окружающей среды (техносферная безопасность) [Электронный ресурс] : учебник для бакалавров / С. В. Белов. — 4-е изд. Схема доступа: <http://www.lib.tpu.ru/fulltext2/m/2013/FN/fn-2440.pdf>
78. Безопасность жизнедеятельности. Безопасность технологических процессов и производств. Охрана труда : учебное пособие для вузов / П. П. Кукин [и др.]. — 5-е изд., стер. — Москва: Высшая школа, 2009. — 335 с.: ил. — Для высших учебных заведений. — Безопасность жизнедеятельности. — Библиогр.: с. 333.
79. Авраамов, Ю. С. Защита человека от электромагнитных воздействий / Ю. С. Авраамов, Н. Н. Грачев, А. Д. Шляпин. — Москва: Изд-во МГИУ, 2002. — 232 с.: ил. — Это важно знать!. — Библиогр.: с. 227-231.
80. ГОСТ 12.0.003-74. ССБТ. Опасные и вредные производственные факторы. Классификация
81. ГОСТ 12.1.003-83 ССБТ. Шум. Общие требования безопасности.
82. ГОСТ 12.1.005-88. ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны
83. ГОСТ 12.1.038-82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов.
84. СанПиН 2.2.1/2.1.1.1278-03. Гигиенические требования к естественному, искусственному и совмещённому освещению жилых и общественных зданий.
85. СанПиН 2.2.2/2.4.1340-03. Санитарно-эпидемиологические правила и нормативы «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».
86. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений.
87. ГОСТ Р 22.0.01-94. Безопасность в ЧС. Основные положения.
88. ГОСТ Р 22.3.03-94. Безопасность в ЧС. Защита населения. Основные положения.
89. ГОСТ Р 22.0.07-95. Безопасность в чрезвычайных ситуациях. Источники техногенных чрезвычайных ситуаций. Классификация и номенклатура поражающих факторов и их параметров.
90. ГОСТ 12.2.049-80 ССБТ. Оборудование производственное. Общие эргономические требования.
91. ГОСТ 12.2.032-78. Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования.

## Приложение А.

### Раздел 3 Английский язык

Студент

8БМ41	Пискунова Татьяна Александровна		
Группа	ФИО	Подпись	Дата

Консультант кафедры Прикладной математики (ПМ):

доцент	Вылегжанин	К.Х.Н	
Должность	ФИО	Ученая степень, звание	Подпись
			Дата

Консультант – лингвист кафедры Английского языка (ИЯИК):

Зав. каф	Сидоренко Т.В.	К.П.Н	
Должность	ФИО	Ученая степень, звание	Подпись
			Дата



## **7. Decision tree and knowledge base construction**

Due to the semi-structured data, the use of fuzzy decision trees to build IDSS is justified. The main idea of this approach is the combination of decision trees and fuzzy logic. A distinctive feature of fuzzy decision trees is that they group the degree of membership of each instance in a particular node.

However, a representative set of training examples is necessary to obtain adequate results; otherwise the generated decision tree algorithm reflects the reality poorly and, as a consequence, gives fault results.

For each variable a linguistic variable is created, a membership function is determined and the rules obtained in the rule base when constructing a fuzzy decision tree are made out.

Due to the nature of the problem, the group direct methods with the assistance of expert assessments will be applied to the construction of fuzzy decision-making systems.

### **7.1. The linguistic variables definition**

The definition of linguistic variables is as follows:

The scientific potential of the a student is determined by the terms "Low", "Medium" and "High". The minimum possible value is 0 and the maximum – 1000. The formalization of such a description can be carried out using the following linguistic value

$$\text{publications} = (\beta, T, X, G, M)$$

where  $\beta$  = the scientific potential of the student;

$$T = \{\text{"Low"}, \text{"Medium"}, \text{"High"}\};$$

$$X = [0, 1000];$$

G - procedure for the formation of new terms by connectives "and" "or" and modifiers such as "very", "slight", "no", "almost", "not really", etc.

M – procedure that defines fuzzy subsets: A1 = “Low” , A2 =”Medium” and A3 =”High” in  $X = [0, 1000]$  and fuzzy sets for terms of G (T) in respect with the rules of broadcasting fuzzy connectives and modifiers.

The membership function is created for each term from T

The Membership functions describing A1 = “Low”, A2 = “Medium” and A3 =”High” (Figure 7.1.) are defined as follows:

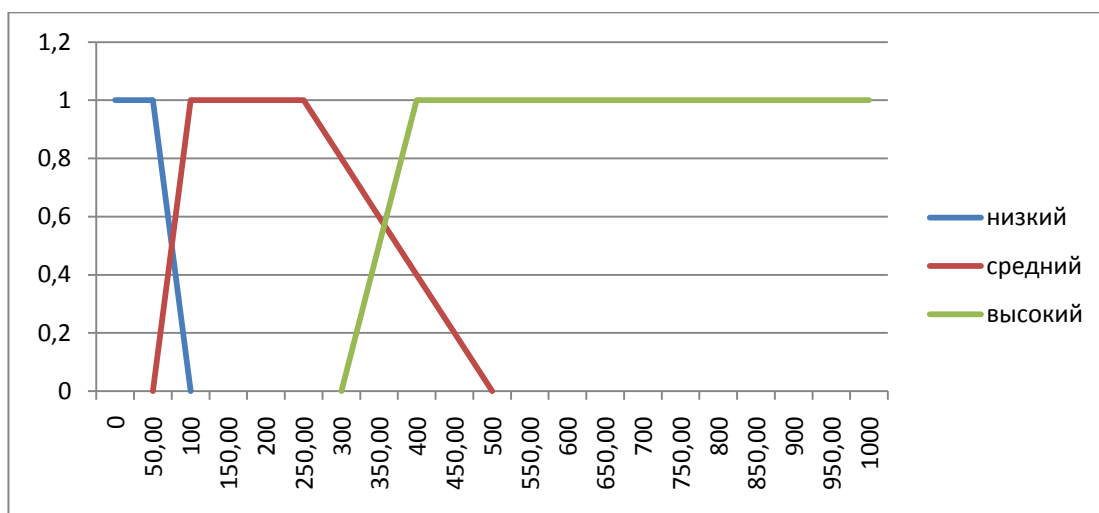


Fig. 7.1. Membership functions for "scientific potential of the student" LV terms

Similarly we define the linguistic variables for other factors (in the appendix).

Thus, the linguistic values for all the interval, ranked and interval parameters and characteristics are described. For values of nominal scale, linguistic variables are not to define. In this case, for classification the original values without fuzzification are used.

For example, a student activity in a group can take the following values: "group leader", "trade union leader" and "none" and the variable “lead” can be defined as a set of values and the corresponding numerical code values.

lead = {"group leader", "trade union leader", "none"; 1, 2, 3}

LV for the classification is a level of matching one of the following classes:  
"Ideal student", "Good", "Medium", "Problem"

( $\beta$ , T, X, G, M)

where  $\beta$  = student success;

T = {"Ideal student", "Good", "Medium", "Problem"};

X = [0,1];

### 7.1. Classification

Then the data from the database are used to construct a fuzzy decision tree. The internal nodes of the tree are the attributes of the database.

Formation of the rule base is based on a step by step learning mode, and the fuzzy decision tree is obtained as a result. During training values are adjusted in each node of the tree and the membership functions are interpolated for subsequent fuzzification of input variables. On aggregation step the number of fuzzy logic rules is optimized. Inference accumulation and defuzzification are performed steps-by-step at each node of the decision tree. Testing of a built fuzzy decision tree occurs on real data examples. Further the quality criterion of the system training is tested by comparing with the goal value. Reaching the target value means that the construction of a fuzzy decision tree is completed and the base of fuzzy rules is established.

Let's construct a tree on a set of minimum features: progress in different subject series and achievements in science and education. An example of the data used to build and test the tree are shown in Table 7.1.

Table 7.1. Data for classification

The scientific	HSE	SCI	GP	SD	FD	Class
----------------	-----	-----	----	----	----	-------

potential						
-1	0,6	0,8	0,86	0,94	-1	<b>B</b>
-1	0,8	0,8	0,88	0,95	-1	<b>B</b>
-1	0,7	1	0,95	0,92	-1	<b>B</b>
0,018	1	1	1	0,96	1	<b>A</b>
0,023	1	1	1	0,95	-1	<b>A</b>
-1	0,8	0,9	0,92	0,94	-1	<b>B</b>
-1	1	1	0,86	0,8	0,8	<b>B</b>
0,196	1	1	1	1	-1	<b>A</b>
-1	0,85	0,74	0,8	0,74	-1	<b>C</b>
-1	-1	0,74	0,96	0,9	-1	<b>B</b>
-1	-1	0,9	0,92	0,98	0,975	<b>B</b>
-1	-1	1	1	1	0,975	<b>A</b>
-1	-1	0,74	0,74	0,76	0,8	<b>C</b>
-1	-1	0,8	0,88	0,92	0,925	<b>B</b>
-1	-1	0,8	0,88	0,96	1	<b>B</b>

An algorithm for constructing a decision tree allows missing values. Therefore, in the following table cells unknown values are marked with number "-1".

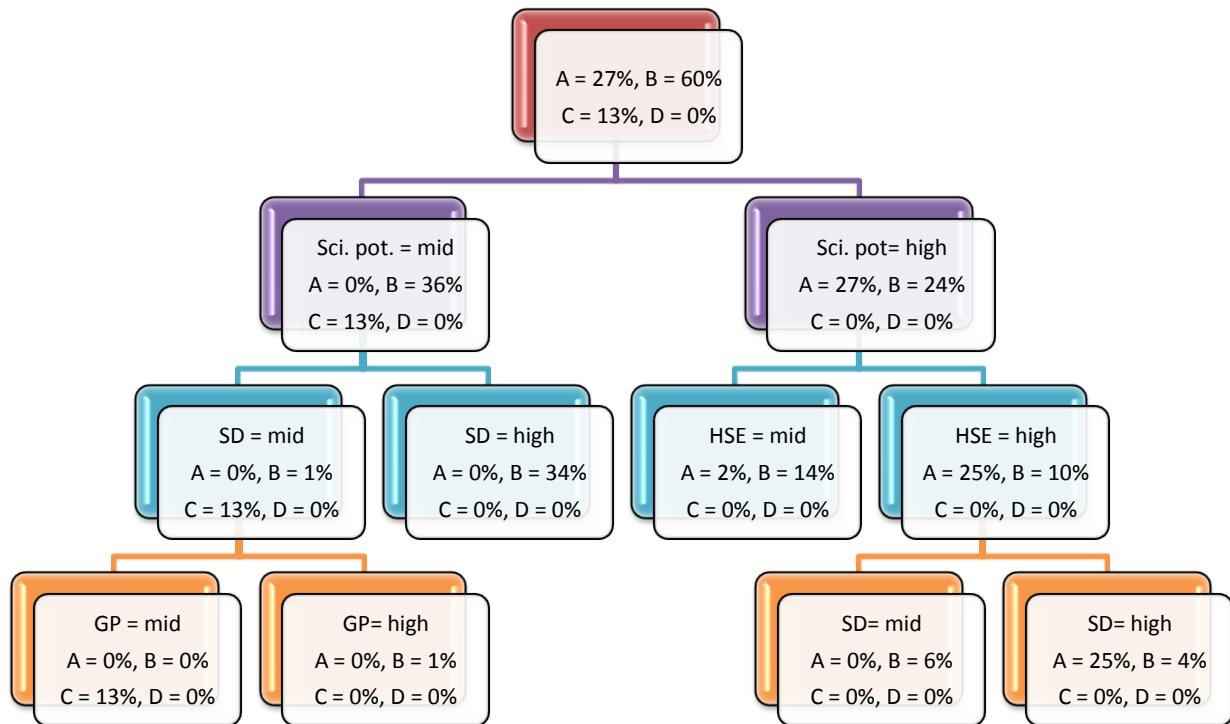


Figure 7.2. A decision tree

Here, when forming the rules the tree is split into various features at the different levels. The result of the splitting into leaves does not belong to one particular class but displays their degree of membership. That provides more complete and qualitative assessment of the classification results.

## 8. The database building

### 8.1 The basic structure of the database selection

A database creating starts with defining the data that are already used in the university DB, and those which is necessary to create. The key data out of the given ones are the values of `id_person` and `id_student`, which are associated with all other data required for the assessment in the database of TPU. In this case two identifiers are essential, because one person may continue his/her education after graduation (and get second (higher) degree) as well as study more than one specialty. It means that one cannot uniquely identify a person as a student of the particular department. Based on the given above three ways to organize the database "core" – tables that describe the student and his/her academic performance – have been selected.

The first method (Fig. 8.1) allows to use the data as they are, and divide the data associated with a learner only as an individual, or as a student (e.g., psychological characteristics are relevant to an individual and does not depend on his/her study; and the data about the group belongs only to a student and doesn't specify the individual data). But this method also has a drawback: such organization of the DB leads to the situation when each table with achievements has at least two fields in the primary key: from tables "Student" and "Person". This is undesirable, because the data input becomes more complicated and it increases the number of errors.

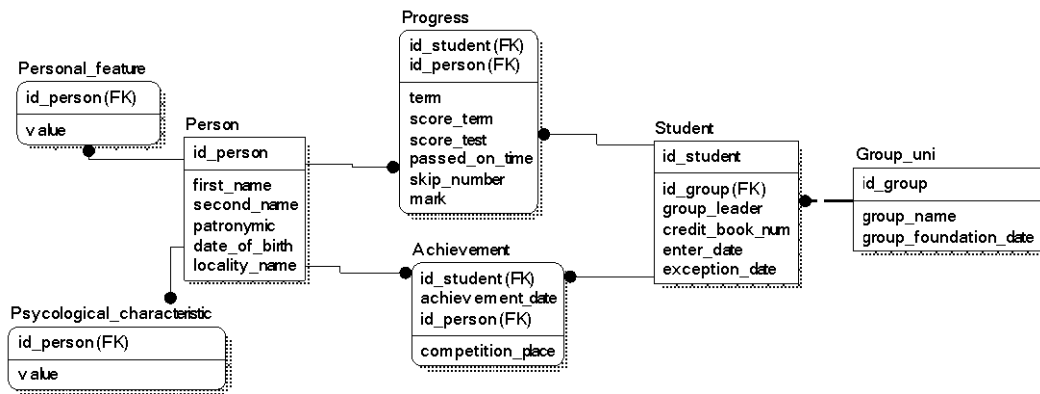


Fig. 8.1. The first type of the database organization.

Fig. 8.1. The first type of the database organization.

The second type (Fig. 8.2) allows to transmit only one ID that is created on the base of two already existing by combining them into one table Lerner. This table also contains all the personal data and information about any student. On the one hand, it gives you a fast access to all the information about students and on the other hand, it can lead to the lack of information or its duplication in cases when a person studies in two specialties, or continues the training in magistracy. This situation can create the confusion in the data and complicate a data search and analysis.

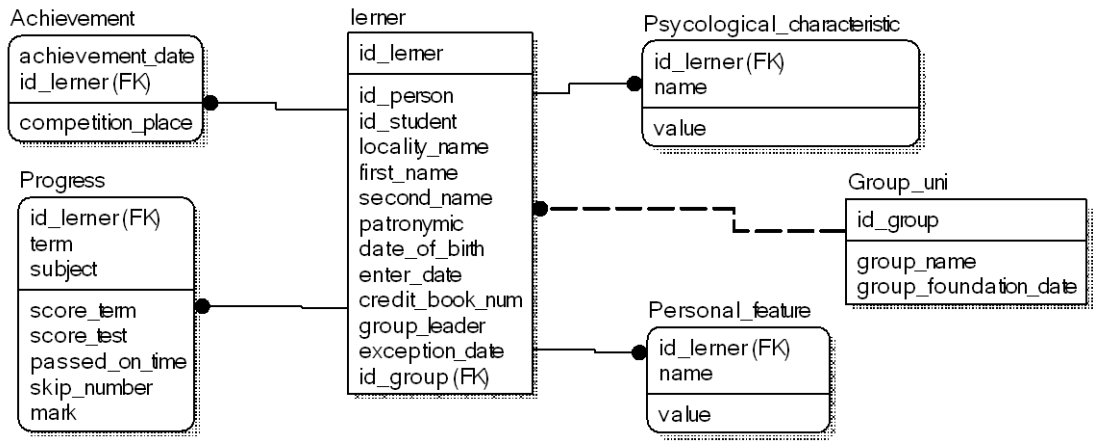


Fig. 8.2. The second type of the database organization

The third method, like the previous one, involves the transfer of only one attribute from the primary key of tables Person and Student (Fig. 8.3). It also allows you to store relevant information separately. This is accomplished by binding two tables: the identification from the “Person” table is put in the “Student” as a foreign key. Thus, the problem about the relation of "two" students to "one" person is solved. But in this approach it will be difficult to construct in terms of selecting some parameters which are on "opposite" database sides. This will require the creation of large queries with multiple inner joints.

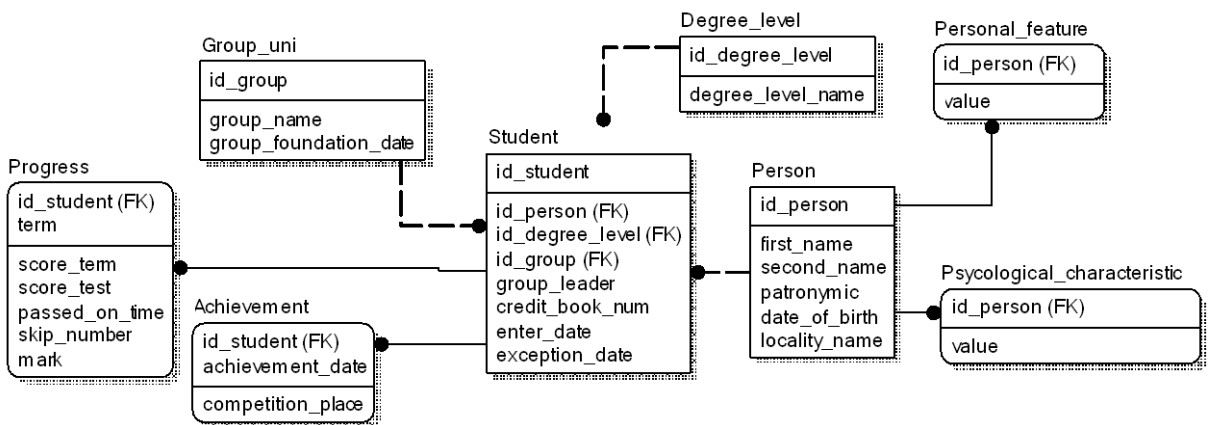


Fig. 8.3. The third type of the database organization

The third type was chosen, since for the analysis and classification the transparency and completeness of the data is very important. That eliminates the second option. Also the ease of the data manipulation is required, and the transfer of only one identifier can provide that. The first type of the database organization is not

so convenient in handling the tables. Now there still is the last problem concerning the complexity of the queries. It can be decreased by using samples based on the views of the database.

## 8.2 The complex database schema

Next several more tables were added to the DB. They define the educational, scientific, social, artistic and sports activities of the student, knowledge of foreign languages, personal and psychological characteristics.

Many fields required the creating codifiers-tables, which contain all the possible values of the corresponding fields. Such codifiers have a limited number of records and consist only of the two fields: identifier as the primary key and the characteristic's name.

For each table having its own identifier, the sequence and trigger were created. Triggers set the value of a unique primary key.

For example, a codicator-table "Institute" shows all the existing institutions in the TPU:

ID_INSTITUTE	INSTITUTE_NAME
1	Institute of Cybernetics (IC)
21	Institute of Natural Resources (INR)
22	Institute of Power Engineering (IPE)
23	Institute of High Technology Physics (IHTP)
24	Institute of Physics and Technology (IPT)
25	Institute of Non-Destructive Testing (INDT)
26	Institute of Humanities, Social Sciences and Technologies (IHSST)



27	Institute of International Education and Language Communication (IIELC)
28	Yurga Institute of Technology (YIT)
29	E-learning Institute (EI)

Also the application session processing needs creating tables for storing users' information such as login-password, email address, access level, user settings, etc.

As a result, the following database is obtained. It is divided into two independent parts: the student data (see Figure 8.4) and the application user data (see Figure 8.5).

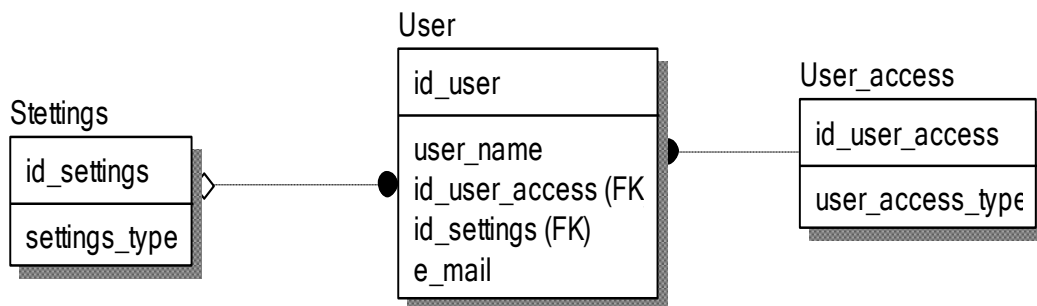


Fig. 8.4. Database about users

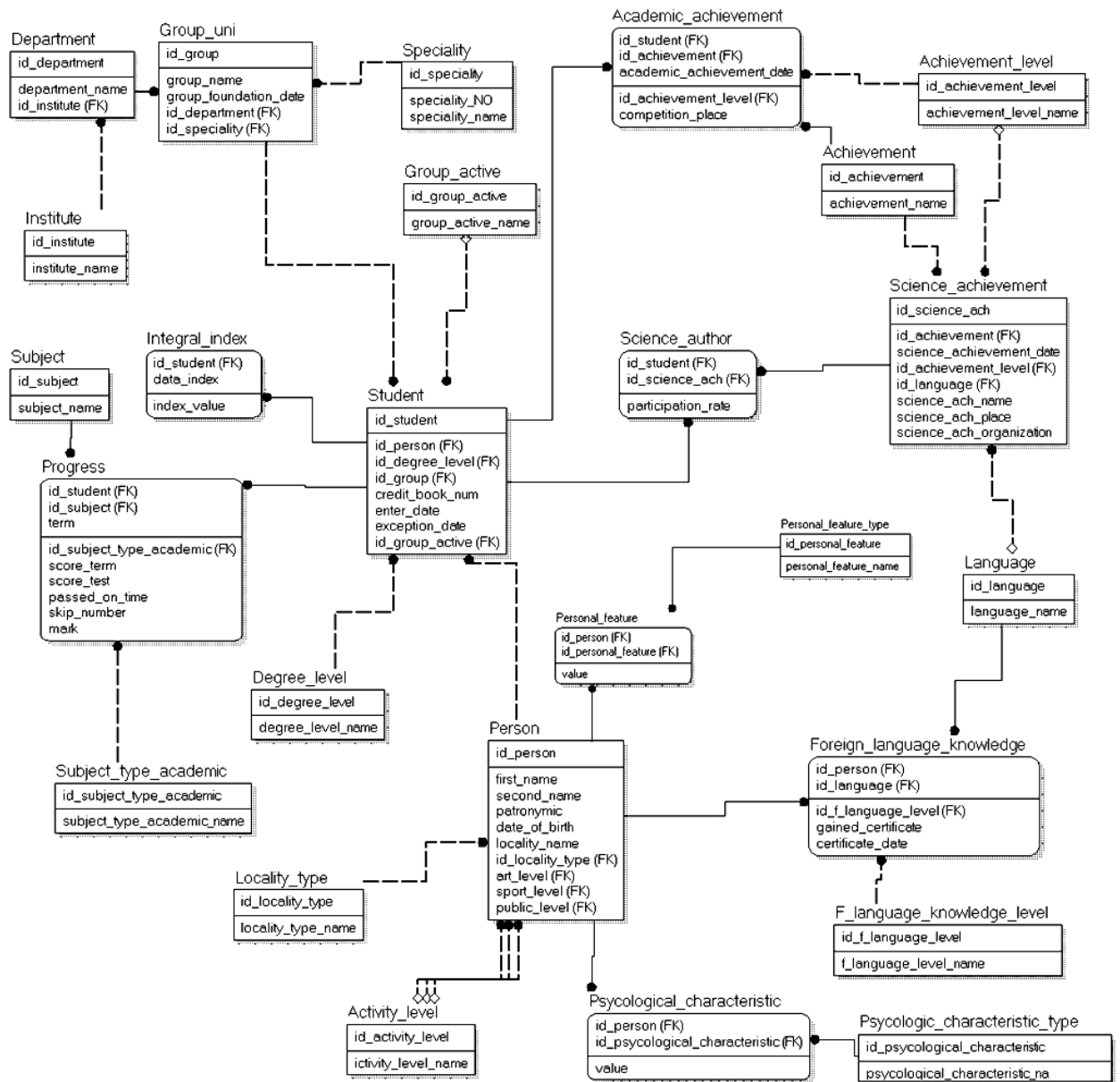


Fig. 8.5. The database scheme with student information

The database scheme that defines the student data can also be divided (only conventionally) into two parts. Again, according to the original data, the first part is correlated with a student personality. It is associated with a table, which stores the personal data and the information about personal and psychological characteristics, social, artistic and sports activities, language skills. The second part contains the data about activities in the university, ie data on a student's department, grades and other academic achievements, publications and science success, social activity in a group (for example, whether a student is a group leader or a member of a trade-union). Also the parameters of a student are described by the table that contains the integral index,

which was determined using data mining methods, as it is described previously. The index, like other data in the database, depends on the time. That allows spending the time analysis of a student's activities, as well as to build further forecast. Based on this analysis, we can make decisions about a student's promotion to the magistracy; give the recommendation to finish his education at the current level, or even the insist on his dismissal.

There are restrictions on the tables containing the information on the achievements in scientific and educational activities: it is not possible to associate an achievement with more than one person. The intermediate table between the tables "Student" and "Science\_achievement" is necessary to be introduced to implement this functionality since there are a lot of competitions and projects that let more than one person to participate. This arrangement allows assigning the achievement to more than one person, and indicates his participation rate.

Many attributes and relationships in the database are optional and can be empty (NULL). On the one hand, it is not good, because it makes impossible to perform the complex analysis all the students' characteristics and as a result it can lead to some errors in the final ranking. However, it is also obvious that it is impossible to guarantee that all the data about students will be obtained. On the other hand, we could reduce the number of parameters and ensure their availability. But in this case, it is possible during the classification to lose more precise characteristics of those students for whom this information is available.

## **The application interface building**

### **The application structure**

The application should have a web interface, be able to connect to the database and control the data sampling according to a user query. One of the design patterns that provide the functionality for such systems is MVC.

Each page is declaratively defined using metadata to select templates to be used for the presentation. The APEX engine dynamically renders an HTML page

using the metadata, assembling relevant templates and injecting dynamic data into the templates. The HTML page is viewed when you request a page through a web browser. When you submit a page, the APEX engine performs page processing, once again using declaratively defined metadata to perform computations, validations, etc.

This type of processing is a typical Model-View-Controller (MVC) pattern, where the view is the HTML generated using the application templates. The APEX engine is the controller and receives the GET or POST input and decides what to do with it, handing over to domain objects. The domain objects model is encapsulated in the `page_definition` and contains the business rules and functionality to carry out the specific tasks.

The MVC pattern also promotes another good design principle-separation of concerns. APEX has been designed so that the APEX engine, templates, and the application functionality can be optimized independently from each other. Clearly, the process of assembling and sequencing the steps necessary to render a page, and process a page are important to the overall solution. By separating this out and letting Oracle deal with the complexities of this through the APEX engine, it allows programmers to be concentrated on providing a business functionality. Equally, by separating presentation templates from the business logic, it allows each aspect to be maintained separately. This provides a number of advantages including ease of design change, allowing templates to be modified either by different people or at different times to enhance the interface without breaking the application.

An excellent example of this is the standard themes provided in APEX, which have been designed to be completely interchangeable. Switching standard themes is simply a matter of loading a theme from the repository and then switching the active theme. APEX then remaps components to the new active theme using the template class identifiers.

## Page changing scheme

Let's describe the application logic. The page changing scheme is shown in Fig.9.2.

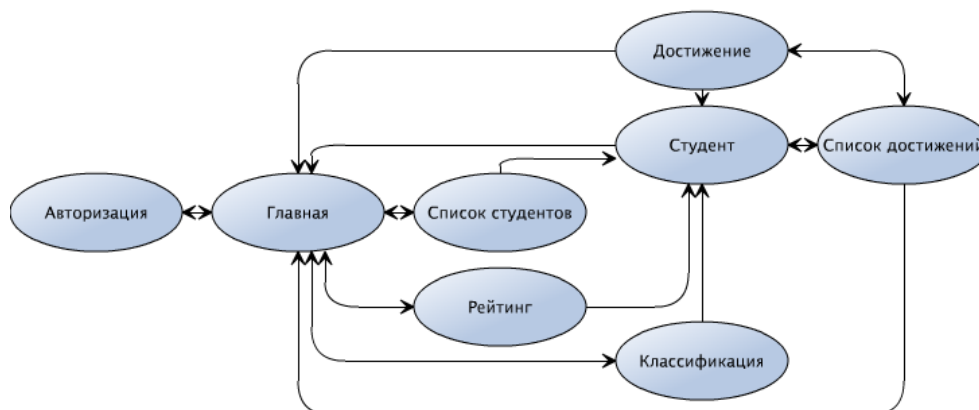


Fig.9.2. Page changing scheme

The first and only available page at the entry - login page (Figure 9.3.). One must enter the login and password pair that is contained in the authorization database to be able to use the functionality of the application. If the wrong combination was entered an appropriate message will be shown and access will be denied.

Авторизация

Логин

Пароль  Войти

Fig. 9.3. Page 101. Authorization

In the case of successful authentication, the user sees the home page of the application (Fig. 9.4). On the left the report block is located. It represents all groups with the number of students in them, grouped in the relevant specialties and departments. On the right there is the evaluation unit, which user can use to send a request to form the rating of students.



Fig. 9.4. Page 1. Home Page

When you click on any specialty, department or group, the app navigates to page 3 "List of students" (Figure 9.5), and transmits there the corresponding value in a hidden field.

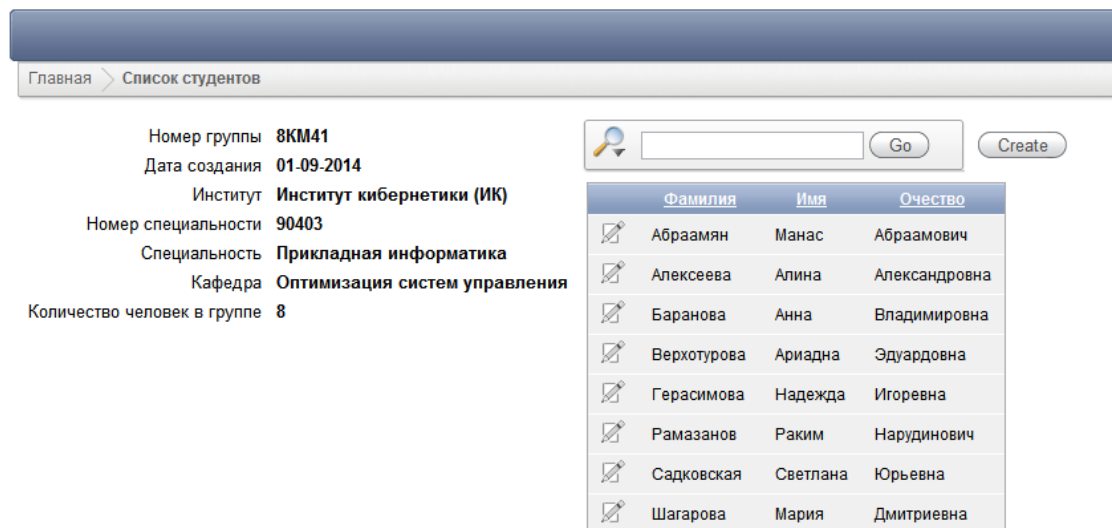


Fig. 9.5. Page 3. List of students

On the third page according to the transmitted from the home page value all the students of the selected value are displayed. If the value is a group than the information about the group is given as well: name, date of creation, the institution, department, specialty, and number of people in the group. From this page you can either return to the home page (by clicking on the appropriate icon in breadcrumbs or back button in a browser), or go to the edit page and a detailed report about the

student – page 4 (Figure 9.6) – by clicking on the edit icon on the left from the selected name.

Главная > Студент

Данные о студенте

\*Фамилия

\*Имя

Очество

\*Дата рождения

Откуда

\*Тип населенного пункта

Творчество

Спорт

Общественная деят-ть

Показать достижения

Fig. 9.6. Page 4. Student

Initially the fourth page display only the form on a student personal data that can be changed if desired, and a drop-down list of the student's activity is shown as well below this form. Its default value is "not selected." If user changes the value the relevant information about the student and a progress graph appear. For example, when selecting "progress", the report page looks as shown in Fig. 9.7. Here, the central unit provides a brief and general information on the selected option - progress - such as the average score for a discipline in different study series, the number of skipped lessons, the number of subjects passed on time, etc. Other values from select list: "science", "performance" and "personal and psychological characteristics" have similar look and functionality.

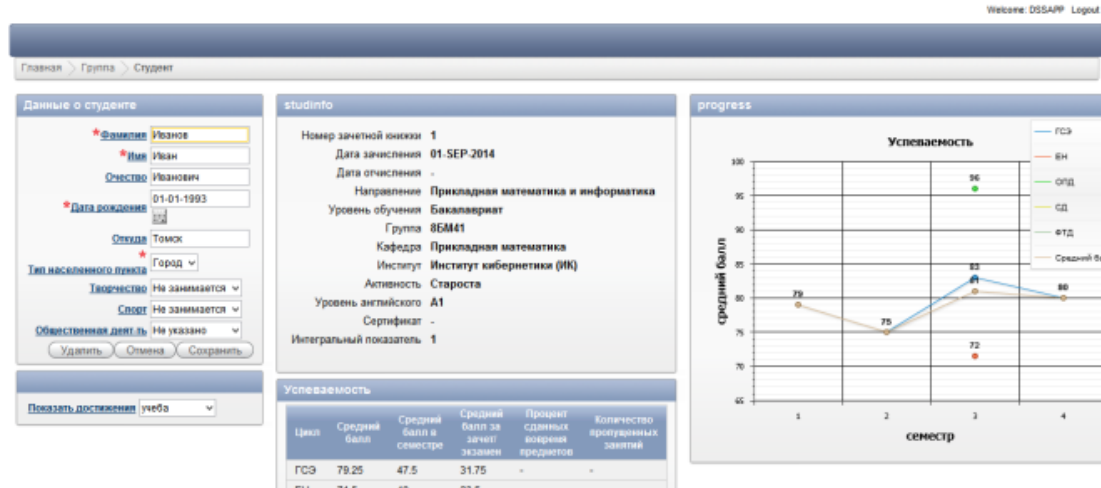


Fig. 9.7. Page 4. Student on a value from select list equal to “progress”

Below the central list there is a button "details". When clicked, it send the user to page 2 "The list of achievements” (Fig. 9.8.), where the list of subjects with scores and passes, grouped and sorted according to the cycles and semesters are displayed.



Fig.9.8. Page 2 List of achievements

On this page, the first column of the table also contains the data edit buttons, which being clicked, proceed the user to page 5 "Achievements" (Fig. 9.9.) which have a form for editing the data.



Главная > Студент > Список достижений > Достижение

Семестр

\*Цикл предмета

Балл в семестре

Балл за экзамен

Своевременность сдачи

Количество пропусков

Оценка

Fig. 9.9. Page 5. Achievement

On this page, you can return to any previous page by using the breadcrumbs (Fig. 9.10)

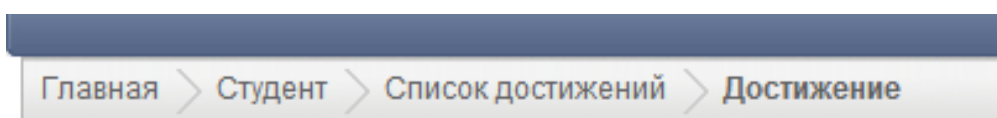


Fig. 9.10. Breadcrumbs on the page 5

Similarly, changing the values in the select list on page 3 "List of students," the user gets the information according to the request.

Now consider the functionality of the second block "Evaluation" on the home page more detailed (Fig 9.11.). As it was mentioned earlier this section defines the parameters, which formed the rating or classification of students in the unit corresponding to the values of the select lists.

Оценка

Кафедра

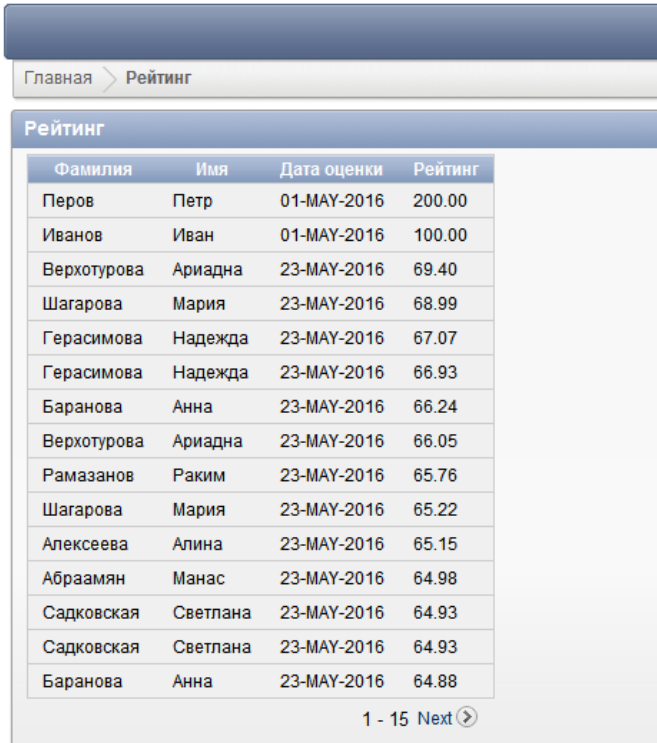
Направление

Группа

Студент

Fig. 9.11. The evaluation unit

By clicking the "rating" button below the app navigates to a page 6 "Rating" (Fig 9.12.). This page displays a list of students, sorted by an integral index. The integral index is determined using a regression tree, described in Chapter 7.



Фамилия	Имя	Дата оценки	Рейтинг
Перов	Петр	01-MAY-2016	200.00
Иванов	Иван	01-MAY-2016	100.00
Верхотурова	Ариадна	23-MAY-2016	69.40
Шагарова	Мария	23-MAY-2016	68.99
Герасимова	Надежда	23-MAY-2016	67.07
Герасимова	Надежда	23-MAY-2016	66.93
Баранова	Анна	23-MAY-2016	66.24
Верхотурова	Ариадна	23-MAY-2016	66.05
Рамазанов	Рахим	23-MAY-2016	65.76
Шагарова	Мария	23-MAY-2016	65.22
Алексеева	Алина	23-MAY-2016	65.15
Абрамян	Манас	23-MAY-2016	64.98
Садковская	Светлана	23-MAY-2016	64.93
Садковская	Светлана	23-MAY-2016	64.93
Баранова	Анна	23-MAY-2016	64.88

Fig. 9.12. Page 6. Ranking

From the page 6 one can return to the home page or by clicking on the edit icon on the left go to the page 4 with the student info.

When user clicks on "Classification" button at the bottom of the evaluation unit on the main page, the page 7 "Classification" appear (Figure 9.13).

Главная > Классификация

с1

Фамилия	Имя	Идеальный	Хорошист	Средний	Проблемный	Дата оценки
Шагарова	Мария	93	9	0	0	30-MAY-2016
Герасимова	Надежда	90	15	0	0	30-MAY-2016
Верхотурова	Ариадна	89	14	0	0	30-MAY-2016
Герасимова	Надежда	84	23	0	0	30-MAY-2016
Верхотурова	Ариадна	79	64	0	0	30-MAY-2016
Садковская	Светлана	69	57	3	0	30-MAY-2016
Рамазанов	Рахим	36	85	11	0	30-MAY-2016
Алексеева	Алина	34	87	11	0	30-MAY-2016
Шагарова	Мария	30	91	2	0	30-MAY-2016
Баранова	Анна	26	84	16	0	30-MAY-2016
Садковская	Светлана	26	79	4	0	30-MAY-2016
Баранова	Анна	13	89	17	0	30-MAY-2016
Абраамян	Манас	7	76	34	0	30-MAY-2016
Абраамян	Манас	1	76	36	2	30-MAY-2016
Рамазанов	Рахим	0	75	41	3	30-MAY-2016

1 - 15

Fig. 9.12. Page 7 Classification

## **Приложение Б. Баллы за достижения в научной и учебной деятельности**

### **Наука**

#### **Интеллектуальная собственность**

Патент на изобретение

+ 200 баллов

Патент на полезную модель

+ 100 баллов

Свидетельство о регистрации программы для ЭВМ, базы данных, интегральных схем

и т.п.

+ 80 баллов

Акты о внедрении

+ 50 баллов

#### **Участие в конференциях**

Диплом на международной конференции

+ 60 баллов

Диплом на российской конференции

+ 50 баллов

Диплом на региональной/областной/городской/университетской/корпоративной (организованной промышленным предприятием) конференции

+ 40 баллов

Очное участие в конференции за пределами РФ

+ 80 баллов

#### **Конкурсы НИР, выставки**

Работа, отмеченная медалью РАН

+ 300 баллов

Работа, отмеченная медалью или дипломом на международном конкурсе

+ 150 баллов

Работа, отмеченная дипломом на российском конкурсе

+ 100 баллов

Работа, отмеченная дипломом на региональном/областном/университетском/корпоративном (организованном промышленным предприятием) конкурсе

+ 80 баллов

#### **Премии, звания, стипендии**

Международные

+ 150 баллов

Российские (учебный год)

+ 100 баллов

Региональные, областные, городские (учебный год)

+ 80 баллов

Университетские (учебный год)

+ 30 баллов

Российские (семестр)

+ 50 баллов

Региональные, областные, городские (семестр)

+ 40 баллов

Университетские (семестр)

+ 15 баллов

**Научные стажировки**

Из средств Международных фондов

+ 300 баллов

Из средств Российских фондов

+ 200 баллов

Из средств ТПУ

+ 40 баллов

**Участие в выполнении исследований по программам и грантам**

Международным

+ 100 баллов

Российским

+ 80 баллов

Областным, городским

+ 60 баллов

Университетским

+ 40 баллов

Трудоустройство в ТПУ на ставку/долю ставки

+ 20 баллов

Сертификаты об участии в конкурсах НИР, конференциях, выставках, грантах

Сертификаты/дипломы/грамоты об участии в конкурсах НИР, конференциях, выставках, грантах

+ 10 баллов

**Публикации**

Статьи в научной периодике, индексируемой международными базами данных (Web of Science, Scopus и др.)

+ 300 баллов У Вас пока нет записей в этой категории.

Статьи в рецензируемом журнале из списка ВАК, входящем в российскую индексируемую базу РИНЦ

+ 150 баллов У Вас пока нет записей в этой категории.

Статья в российских и зарубежных изданиях, не входящих в вышеперечисленные базы данных

+ 60 баллов У Вас пока нет записей в этой категории.

Доклады на международной конференции

+ 30 баллов У Вас пока нет записей в этой категории.

Доклады на российской конференции

+ 20 баллов У Вас пока нет записей в этой категории.

Доклады на региональной конференции

+ 15 баллов У Вас пока нет записей в этой категории.

Доклады на университетской конференции

+ 10 баллов У Вас пока нет записей в этой категории.

Издания (Учебные пособия, монографии)

+ 100 баллов У Вас пока нет записей в этой категории.

Тезисы доклада

+ 10 баллов У Вас пока нет записей в этой категории.

## **Учеба**

### **Победитель или призер ОЛИМПИАДЫ (за один диплом)**

Международная - I место

+ 12 баллов

Международная - II место

+ 11 баллов

Международная - III место

+ 10 баллов

Международная - Диплом лауреата (без указания степени)

+ 9 баллов

Всероссийская (отраслевая) - I место

+ 10 баллов

Всероссийская (отраслевая) - II место

+ 9 баллов

Всероссийская (отраслевая) - III место

+ 8 баллов

Всероссийская (отраслевая) - Диплом лауреата (без указания степени)

+ 7 баллов

Региональная - I место

+ 8 баллов

Региональная - II место

+ 7 баллов

Региональная - III место

+ 6 баллов

Региональная - Диплом лауреата (без указания степени)

+ 5 баллов

Межвузовская (городская, муниципальная) - I место

+ 6 баллов

Межвузовская (городская, муниципальная) - II место

+ 5 баллов

Межвузовская (городская, муниципальная) - III место

+ 4 балла

Межвузовская (городская, муниципальная) - Диплом лауреата (без указания степени)

+ 3 балла

Университетская - I место

+ 4 балла

Университетская - II место

+ 3 балла

Университетская - III место

+ 2 балла

Университетская - Диплом лауреата (без указания степени)

+ 1 балл

### **Победитель или призер КОНКУРСА, соревнования, состязания, направленных на выявление учебных достижений студентов (за один диплом)**

Международная - I место

+ 10 баллов

Международная - II место  
+ 9 баллов

Международная - III место  
+ 8 баллов

Международная - Диплом лауреата (без указания степени)  
+ 7 баллов

Всероссийская (отраслевая) - I место  
+ 8 баллов

Всероссийская (отраслевая) - II место  
+ 7 баллов

Всероссийская (отраслевая) - III место  
+ 6 баллов

Всероссийская (отраслевая) - Диплом лауреата (без указания степени)  
+ 5 баллов

Региональная - I место  
+ 6 баллов

Региональная - II место  
+ 5 баллов

Региональная - III место  
+ 4 балла

Региональная - Диплом лауреата (без указания степени)  
+ 3 балла

Межвузовская (городская, муниципальная) - I место  
+ 4 балла

Межвузовская (городская, муниципальная) - II место  
+ 3 балла

Межвузовская (городская, муниципальная) - III место  
+ 2 балла

Межвузовская (городская, муниципальная) - Диплом лауреата (без указания степени)  
+ 1 балл

Университетская - I место  
+ 2 балла

Университетская - II место  
+ 1 балл

Университетская - III место  
+ .5 баллов

Университетская - Диплом лауреата (без указания степени)  
+ .25 баллов

**Наличие сертификатов на ЗНАНИЕ ИН. ЯЗЫКА с указанием степени владения**

ТПУ  
+ 2 балла

Иностр.  
+ 4 балла

**Результаты академического обмена**

Результаты академического обмена  
+ 4 балла

## Приложение В. SQL код базы данных

### Представления:

Представление для классификации

```
CREATE OR REPLACE FORCE VIEW "DSS"."STUDENTPROGRESSDEMO"
("ID_STUDENT", "AVG_SCORE_TERM", "AVG_SCORE_TEST",
"AVG_PASSED_ON_TIME", "SUM_SKIP_NUMBER", "AVG_MARK", "COLUMN1",
"SUBJECT_TYPE_ACADEMIC_NAME", "ID_GROUP_ACTIVE", "ID_DEGREE_LEVEL") AS
SELECT STUDENT.ID_STUDENT,
AVG(PROGRESS.SCORE_TERM)           AS Avg_SCORE_TERM,
AVG(PROGRESS.SCORE_TEST)          AS Avg_SCORE_TEST,
AVG(PROGRESS.PASSED_ON_TIME)      AS Avg_PASSED_ON_TIME,
SUM(PROGRESS.SKIP_NUMBER)         AS Sum_SKIP_NUMBER,
AVG(PROGRESS.MARK)                AS Avg_MARK,
AVG(PROGRESS.SCORE_TERM + PROGRESS.SCORE_TEST) AS Column1,
SUBJECT_TYPE_ACADEMIC.SUBJECT_TYPE_ACADEMIC_NAME,
STUDENT.ID_GROUP_ACTIVE,
STUDENT.ID_DEGREE_LEVEL
FROM STUDENT
INNER JOIN PROGRESS
ON STUDENT.ID_STUDENT = PROGRESS.ID_STUDENT
INNER JOIN SUBJECT
ON SUBJECT.ID_SUBJECT = PROGRESS.ID_SUBJECT
INNER JOIN SUBJECT_TYPE_ACADEMIC
ON SUBJECT_TYPE_ACADEMIC.ID_SUBJECT_TYPE_ACADEMIC =
PROGRESS.ID_SUBJECT_TYPE_ACADEMIC
GROUP BY STUDENT.ID_STUDENT,
SUBJECT_TYPE_ACADEMIC.SUBJECT_TYPE_ACADEMIC_NAME,
STUDENT.ID_GROUP_ACTIVE,
STUDENT.ID_DEGREE_LEVEL;
```

### Последовательности и триггеры:

```
create sequence achSeq;
create or replace TRIGGER "ach_trig"
BEFORE INSERT ON Achievement
FOR EACH ROW
DECLARE
ach_id number;
```



```

BEGIN
  SELECT achSeq.nextval
    INTO ach_id
    FROM dual;
  :new.id_achievement := ach_id;
END;

CREATE SEQUENCE achLevSeq;

create or replace TRIGGER "ach_lev_trig"
BEFORE INSERT ON Achievement_level
FOR EACH ROW
DECLARE
  ach_lev_id number;
BEGIN
  SELECT achLevSeq.nextval
    INTO ach_lev_id
    FROM dual;
  :new.id_achievement_level := ach_lev_id;
END;

```

Последовательности и триггеры для остальных 15 таблиц созданы аналогично представленным

### **Запросы из приложения:**

Для главной:

```

select * from (
select  *
  from  "MAIN_PAGE" "MAIN_PAGE")
where (
instr(upper("GROUP_NAME"),upper(nvl(:P1_SEARCH1,"GROUP_NAME"))) > 0 or
instr(upper("SPECIALITY_NAME"),upper(nvl(:P1_SEARCH1,"SPECIALITY_NAME"))) > 0
or
instr(upper("DEPARTMENT_NAME"),upper(nvl(:P1_SEARCH1,"DEPARTMENT_NAME"))) >
0
)

```

Страница 2:

```

select "ID_SUBJECT",
"ID_STUDENT",
"TERM",
"ID_SUBJECT_TYPE_ACADEMIC",
"SCORE_TERM",
"SCORE_TEST",
"PASSED_ON_TIME",
"SKIP_NUMBER",

```

```
"MARK"  
from "#OWNER#". "PROGRESS"  
WHERE PROGRESS.ID_STUDENT = :P2_X
```

Страница 3:

```
SELECT GROUP_UNI.GROUP_NAME,  
       GROUP_UNI.GROUP_FOUNDATION_DATE,  
       INSTITUTE.INSTITUTE_NAME,  
       SPECIALITY.SPECIALITY_NO,  
       SPECIALITY.SPECIALITY_NAME,  
       DEPARTMENT.DEPARTMENT_NAME,  
       GROUP_UNI.ID_GROUP,  
       SPECIALITY.ID_SPECIALITY,  
       DEPARTMENT.ID_DEPARTMENT,  
       INSTITUTE.ID_INSTITUTE,  
       COUNT(STUDENT.ID_STUDENT) AS Count_ID_STUDENT  
FROM GROUP_UNI  
INNER JOIN SPECIALITY  
ON SPECIALITY.ID_SPECIALITY = GROUP_UNI.ID_SPECIALITY  
INNER JOIN DEPARTMENT  
ON DEPARTMENT.ID_DEPARTMENT = GROUP_UNI.ID_DEPARTMENT  
INNER JOIN INSTITUTE  
ON INSTITUTE.ID_INSTITUTE = DEPARTMENT.ID_INSTITUTE  
INNER JOIN STUDENT  
ON GROUP_UNI.ID_GROUP = STUDENT.ID_GROUP  
WHERE GROUP_UNI.ID_GROUP = :P3_X  
GROUP BY GROUP_UNI.GROUP_NAME,  
       GROUP_UNI.GROUP_FOUNDATION_DATE,  
       INSTITUTE.INSTITUTE_NAME,  
       SPECIALITY.SPECIALITY_NO,  
       SPECIALITY.SPECIALITY_NAME,  
       DEPARTMENT.DEPARTMENT_NAME,  
       GROUP_UNI.ID_GROUP,  
       SPECIALITY.ID_SPECIALITY,  
       DEPARTMENT.ID_DEPARTMENT,  
       INSTITUTE.ID_INSTITUTE  
  
SELECT DEPARTMENT.DEPARTMENT_NAME,  
       GROUP_UNI.ID_GROUP,  
       GROUP_UNI.GROUP_NAME,  
       SPECIALITY.ID_SPECIALITY,  
       DEPARTMENT.ID_DEPARTMENT,  
       STUDENT.ID_STUDENT,  
       STUDENT.ID_PERSON,  
       PERSON.FIRST_NAME,  
       PERSON.SECOND_NAME,
```

```

PERSON.PATRONYMIC
FROM GROUP_UNI
INNER JOIN SPECIALITY
ON SPECIALITY.ID_SPECIALITY = GROUP_UNI.ID_SPECIALITY
INNER JOIN DEPARTMENT
ON DEPARTMENT.ID_DEPARTMENT = GROUP_UNI.ID_DEPARTMENT
INNER JOIN STUDENT
ON GROUP_UNI.ID_GROUP = STUDENT.ID_GROUP
INNER JOIN PERSON
ON PERSON.ID_PERSON = STUDENT.ID_PERSON
WHERE (nvl(GROUP_UNI.ID_GROUP , '0') = :P3_X)
OR (nvl(SPECIALITY.ID_SPECIALITY , '0') = :P3_XS)
OR (nvl(DEPARTMENT.ID_DEPARTMENT , '0') = :P3_XD)

```

Страница 4:

```

select  "STUDENTINFO"."ID_PERSON" as "ID_PERSON",
        "STUDENTINFO"."DEPARTMENT_NAME1" as "DEPARTMENT_NAME1",
        "STUDENTINFO"."F_LANGUAGE_LEVEL_NAME" as
"F_LANGUAGE_LEVEL_NAME",
        "STUDENTINFO"."GAINED_CERTIFICATE" as "GAINED_CERTIFICATE",
        "STUDENTINFO"."GROUP_ACTIVE_NAME1" as "GROUP_ACTIVE_NAME1",
        "STUDENTINFO"."DEGREE_LEVEL_NAME" as "DEGREE_LEVEL_NAME",
        "STUDENTINFO"."GROUP_NAME" as "GROUP_NAME",
        "STUDENTINFO"."SPECIALITY_NAME" as "SPECIALITY_NAME",
        "STUDENTINFO"."INSTITUTE_NAME" as "INSTITUTE_NAME",
        "STUDENTINFO"."INDEX_VALUE" as "INDEX_VALUE",
        "STUDENTINFO"."EXCEPTION_DATE" as "EXCEPTION_DATE",
        "STUDENTINFO"."ENTER_DATE" as "ENTER_DATE",
        "STUDENTINFO"."CREDIT_BOOK_NUM" as "CREDIT_BOOK_NUM",
        "STUDENTINFO"."ID_STUDENT" as "ID_STUDENT"
from    "STUDENTINFO" "STUDENTINFO"
WHERE  STUDENTINFO.ID_PERSON = :P4_ID_PERSON

```

## Приложение Г. Лингвистические переменные и функции принадлежности

Средний балл  $\text{avgScore} = (\beta, T, X, G, M)$ ,

где  $\beta$  = средний балл студента;

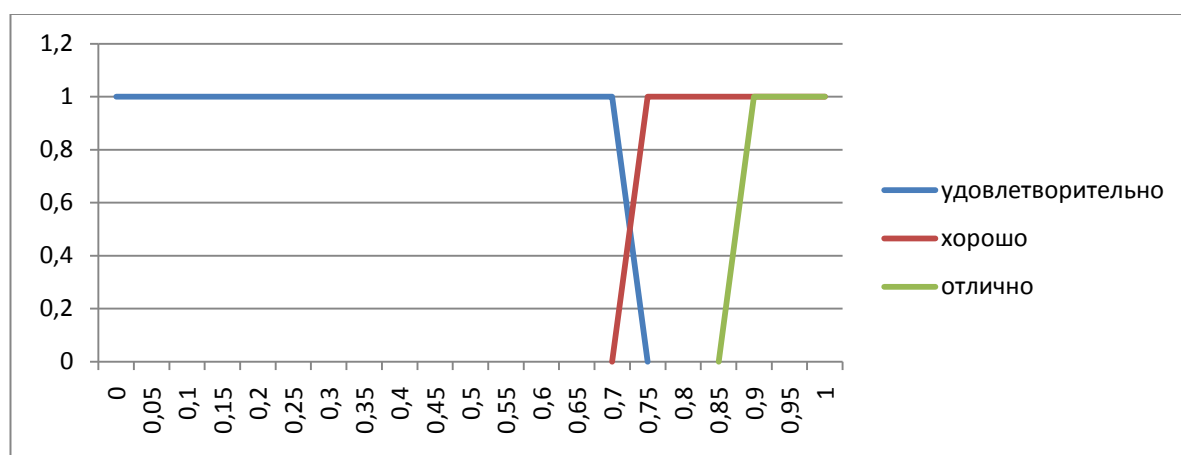
$T = \{\text{«Низкий»}, \text{«Средний»}, \text{«Высокий»}\}$ ;

$X = [0,100]$ ;

$G$  – процедура образования новых термов с помощью связок «и» «или» и модификаторов типа «очень», «слегка», «не», «почти», «не совсем», и т.п.

$M$  – процедура задания на  $X = [0, 100]$  нечетких подмножеств:  $A1 = \text{«Низкий»}$ ,  $A2 = \text{«Средний»}$  и  $A3 = \text{«Высокий»}$ , а также нечетких множеств для термов из  $G(T)$  в соответствии с правилами трансляции нечетких связок и модификаторов и других операций над нечеткими множествами вида:  $A \text{ } \zeta \text{ } B$ ,  $A \text{ } \dot{\text{E}} \text{ } B$ .

Функции принадлежности, определяющие  $A1 = \text{«Низкий»}$ ,  $A2 = \text{«Средний»}$  и  $A3 = \text{«Высокий»}$  (рис.), представлены ниже:



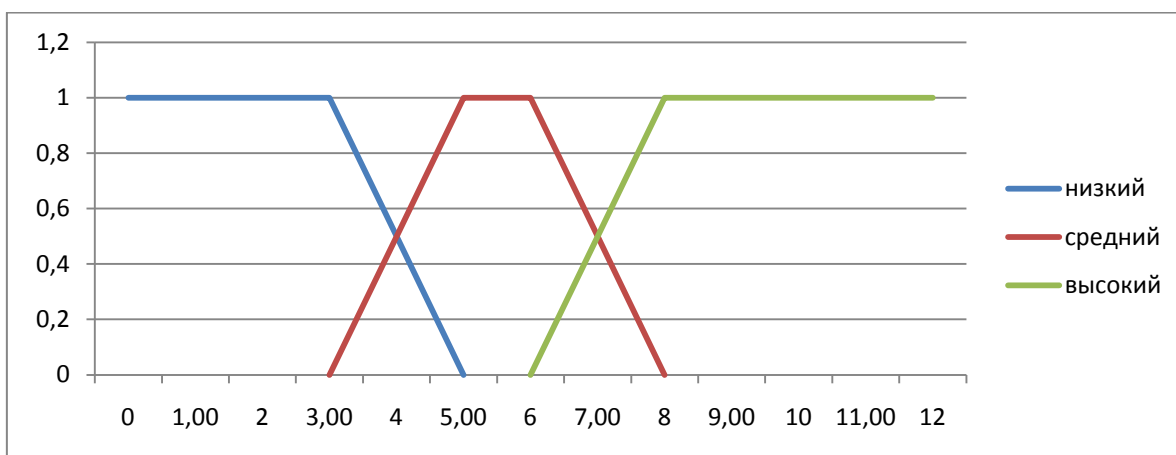
ЛП для психологических оценок из опросника Кеттела:

$T = \{\text{«Низкий»}, \text{«Средний»}, \text{«Высокий»}\}$ ;

$X = [0,12]$ ;

G – процедура образования новых термов с помощью связок «и» «или» и модификаторов типа «очень», «слегка», «не», «почти», «не совсем», и т.п.

M – процедура задания на  $X = [0, 100]$  нечетких подмножеств:  $A_1 =$  «Низкий»,  $A_2 =$  «Средний» и  $A_3 =$  «Высокий», а также нечетких множеств для термов из  $G(T)$  в соответствии с правилами трансляции нечетких связок и модификаторов и других операций над нечеткими множествами вида:  $A \text{ } \zeta \text{ } B$ ,  $A \text{ } \grave{E} \text{ } B$ .



Тип населенного пункта возможны значения «город», «село», и т.д., и переменную *locality* можно определить как набор значений и соответствующих им значений числового кода.

$$\text{Locality} = \{\text{«город»}, \text{«село»} \dots; 1, 2, \dots\}$$