Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Умаров Марат Файзуллаевич
Должность: Директор
Дата подписания: 28.09.2026 15:03:52
Уникальный программный ключ:
48505f11ec15acaa386f5219d3113d727fefda78

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Елабужский институт (филиал) КФУ

УТВЕРЖДАЮ
Заместитель директора по
образовательной деятельности
_____ С.Ю. Бахвалов
«__» _____ 2026 г.
МП

Программа дисциплины (модуля)

Документооборот и делопроизводство

Направление подготовки/специальность: 23.03.01 Технология транспортных процессов

Направленность (профиль) подготовки: Эксплуатация транспортных средств

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2026

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

2. Место дисциплины (модуля) в структуре ОПОП ВО

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

4.2. Содержание дисциплины (модуля)

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

6. Фонд оценочных средств по дисциплине (модулю)

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

9. Методические указания для обучающихся по освоению дисциплины (модуля)

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

12. Средства адаптации преподавания дисциплины (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

13. Приложение №1. Фонд оценочных средств

14. Приложение №2. Перечень литературы, необходимой для освоения дисциплины (модуля)

15. Приложение №3. Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Программу дисциплины разработал(а)(и) старший преподаватель, к.н. Анисимова Э.С. (Кафедра математики и прикладной информатики)

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-3	Способен проектировать и управлять IT-проектами, осуществлять тестирование компонентов информационных систем, в том числе интеллектуальных
ПК-3.1.	Знать способы проектирования и управления IT-проектами, технологии тестирования компонентов информационных систем, в том числе интеллектуальных
ПК-3.2.	Уметь проектировать и управлять IT-проектами, осуществлять тестирование компонентов информационных систем, в том числе интеллектуальных
ПК-3.3.	Владеть способностью проектировать и управлять IT-проектами, осуществлять тестирование компонентов информационных систем, в том числе интеллектуальных

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

технологии тестирования компонентов информационных систем, методы решения задач из области искусственного интеллекта.

Должен уметь:

осуществлять тестирование компонентов информационных систем решать задачи из области искусственного интеллекта.

Должен владеть:

навыками тестирования компонентов информационных систем, решения задач из области искусственного интеллекта.

2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в раздел " Б1.В.02.01 Дисциплины (модули)" основной профессиональной образовательной программы 23.03.01 Технология транспортных процессов и относится к части, формируемой участниками образовательных отношений. Осваивается на 4 курсе в 7 семестре.

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 3 зачетных(ые) единиц(ы) на 108 часа(ов).

Контактная работа - 54 часа(ов), в том числе лекции - 6 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 8 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 85 часа(ов).

Контроль (зачёт / экзамен) - 9 часа(ов).

Форма промежуточного контроля дисциплины: экзамен в 7 семестре.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Введение в проектирование интеллектуальных систем	7	1	0	0	17
2.	Тема 2. Модели представления знаний в интеллектуальных системах	7	1	0	0	17
3.	Тема 3. Язык логического программирования Пролог	7	2	0	4	17
4.	Тема 4. Экспертные системы	7	1	0	0	17
5.	Тема 5. Нейронные сети	7	1	0	4	17
	Итого		6	0	8	85

4.2 Содержание дисциплины (модуля)

Тема 1. Введение в проектирование интеллектуальных систем

Мышление и интеллект, философские аспекты проблемы мышления. Интеллектуальные задачи. Искусственный интеллект. Основные направления исследований в области искусственного интеллекта. План имитации мышления Тьюринга. Японский проект ЭВМ пятого поколения. Этапы создания интеллектуальных компьютеров. Внутренняя и внешняя интеллектуализация.

Тема 2. Модели представления знаний в интеллектуальных системах

Признаки интеллектуальных информационных систем (ИИС), классификация ИИС. Данные и знания. Организация базы знаний. Предметное (фактуальное) и проблемное (операционное) знания. Декларативная и процедурная формы представление знаний. Модели представления знаний: продукционная модель, семантические сети, простые и сложные фреймы, формальные логические модели.

Тема 3. Язык логического программирования Пролог

Императивные и декларативные языки программирования. История возникновения и развития Пролога. Области использования, преимущества и недостатки языка Пролог. Логические основы Пролога. Основные понятия Пролога. Управление выполнением программ на Прологе. Операции над списками, множествами. Обработка строк и файлов. Работа с внутренними (динамическими) базами данных.

Тема 4. Экспертные системы

Понятие об экспертной системе (ЭС). Общая характеристика ЭС. Виды ЭС и типы решаемых задач. Структура и режимы использования ЭС. Этапы проектирования и разработки ЭС: идентификация, концептуализация, формализация, реализация, тестирование, опытная эксплуатация. Участники процесса проектирования: эксперты, инженеры по знаниям, конечные пользователи.

Тема 5. Нейронные сети

Нейрон и межнейронное взаимодействие. Модель технического нейрона. Искусственные нейронные сети и их архитектура. Многослойный перцептрон. Типичные задачи, решаемые с помощью нейронных сетей. Ограничения применения нейронных сетей. Обучение персептрона. Алгоритм обучения персептрона. Процедура обратного распространения. Обучающий алгоритм обратного распространения. Пример обучения. Область применения алгоритма и ограничения по использованию.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному

контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства образования и науки Российской Федерации от 5 апреля 2017 года №301)

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений"

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств по дисциплине (модулю) включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений и навыков. Фонд оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

В фонде оценочных средств содержится следующая информация:

- соответствие компетенций планируемым результатам обучения по дисциплине (модулю);

- критерии оценивания сформированности компетенций;

- механизм формирования оценки по дисциплине (модулю);

- описание порядка применения и процедуры оценивания для каждого оценочного средства;

- критерии оценивания для каждого оценочного средства;

- содержание оценочных средств, включая требования, предъявляемые к действиям обучающихся, демонстрируемым результатам, задания различных типов.

Фонд оценочных средств по дисциплине находится в Приложении 1 к программе дисциплины (модулю).

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

Освоение дисциплины (модуля) предполагает изучение учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде - через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями;

- в печатном виде - в Научной библиотеке Елабужского института КФУ. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования Научной библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе "Электронный университет". При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,25 экземпляра на каждого обучающегося из числа лиц, одновременно осваивающих данную дисциплину.

Перечень учебной литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины. Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов Научной библиотеки Елабужского института КФУ.

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Основы программирования на языке Пролог - https://www.intuit.ru/studies/courses/44/44/info

Основы теории нейронных сетей - https://www.intuit.ru/studies/courses/88/88/info

Проектирование систем искусственного интеллекта - https://www.intuit.ru/studies/courses/1122/167/info

Исследование операций Примеры и задачи - http://math.nsc.ru/LBRT/k4/or/

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид работ	Методические рекомендации

Вид работ	Методические рекомендации
лекции	Лекционные занятия проводятся с использованием интерактивных технологий и предполагают активное участие студентов. Для подготовки к занятиям рекомендуется выделять в материале проблемные вопросы, затрагиваемые преподавателем в лекции, и группировать информацию вокруг них. Желательно выделять в используемой литературе постановки вопросов, на которые разными авторам могут быть даны различные ответы. На основании постановки таких вопросов следует собирать аргументы в пользу различных вариантов решения поставленных проблем.
лабораторные работы	Лабораторные занятия - это одна из разновидностей практического занятия, являющаяся эффективной формой учебных занятий в организации высшего образования. Лабораторные занятия имеют выраженную специфику в зависимости от учебной дисциплины, углубляют и закрепляют теоретические знания. На этих занятиях студенты осваивают конкретные методы изучения дисциплины, обучаются экспериментальным способам анализа, умению работать с приборами и современным оборудованием. Лабораторные занятия дают наглядное представление об изучаемых явлениях и процессах, студенты осваивают постановку и ведение эксперимента, учатся умению наблюдать, оценивать полученные результаты, делать выводы и обобщения. Отчёт по итогам выполненных лабораторных работ выполняется на листах белой бумаги формата А4 в печатном или рукописном виде. При оформлении отчёта используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страницы ставятся по центру вверху. При оформлении отчёта в печатном виде желательно соблюдать следующие требования. Для заголовков: полужирный шрифт, 14 пт, центрированный. Для основного текста: нежирный шрифт, 14 пт, выравнивание по ширине. Во всех случаях тип шрифта - Times New Roman, отступ абзаца 1.25 см, полуторный междустрочный интервал. Поля: левое - 3 см, правое - 1 см, верхнее и нижнее - 2 см. Отчет должен содержать следующие элементы: 1) Титульный лист с обязательным указанием варианта; 2) Цель работы; 3) Задание; 4) Основная часть; 5) Вывод.
самостоя-тельная работа	Самостоятельная работа студентов по дидактической сути представляет собой комплекс условий обучения, организуемых преподавателем и направленных на самоподготовку учащихся. Учебная деятельность протекает без непосредственного участия преподавателя и заключается в проработке лекционного материала, подготовке к лабораторным занятиям; изучении учебной литературы из основного и дополнительного списка.
экзамен	Экзамен может проводиться в форме устного опроса по билетам (вопросам) или без билетов, с предварительной подготовкой или без подготовки, по усмотрению кафедры. Преподаватель может проставить экзамен без опроса или собеседования тем студентам, которые активно участвовали на лабораторных занятиях.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем, представлен в Приложении 3 к рабочей программе дисциплины (модуля).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория №60 (423600, Республика Татарстан, г. Елабуга, ул. Казанская, д. 89) для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации,помещение для самостоятельной работы. Комплект мебели (посадочных мест) 29 шт. Комплект мебели (посадочных мест) для преподавателя 1 шт. Компьютерный класс: Компьютеры intel core i5 15 шт. Мониторы ViewSonic 22d 15 шт. Проектор EPSON EB-535W 1 шт. Интерактивная доска IQBoard DVT TN082 1 шт. Трибуна 1 шт. Кондиционер 1 шт. Настенные пол ки 6 шт. Шкаф двухстворчатый с полками 1 шт. Веб-камера 1 шт. Выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Набор учебно-наглядных пособий: комплект презентаций в электронном формате по преподаваемой дисциплине 3-5 шт.

12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с

ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;

- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;

- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;

- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;

- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;

- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;

- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:

- продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;

- продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;

- продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 23.03.01 "Технология транспортных процессов" и профилю подготовки " Проектирование и управление интеллектуальными транспортными системами".

Приложение №1
к рабочей программе дисциплины (модуля)
Б1.В.02.01 Основы искусственного интеллекта

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Елабужский институт (филиал)

Фонд оценочных средств по дисциплине (модулю)
Б1.В.02.01 Документооборот делопроизводство

Направление подготовки: 23.03.01 Технология транспортных процессов
Профиль подготовки: Проектирование и управление интеллектуальными транспортными системами
Квалификация выпускника: бакалавр
Форма обучения: заочное
Язык обучения: русский
Год начала обучения по образовательной программе: 2026

СОДЕРЖАНИЕ

1. Соответствие компетенций планируемым результатам обучения по дисциплине (модулю)
2. Критерии оценивания сформированности компетенций
3. Распределение оценок за формы текущего контроля и промежуточную аттестацию
4. Оценочные средства, порядок их применения и критерии оценивания
    4.1. Оценочные средства текущего контроля
        4.1.1. Лабораторные работы
            4.1.1.1. Порядок проведения.
            4.1.1.2 Критерии оценивания
            4.1.1.3. Содержание оценочного средства
        4.1.2. Эссе
            4.1.2.1. Порядок проведения.
            4.1.2.2 Критерии оценивания
            4.1.2.3. Содержание оценочного средства
    4.2. Оценочные средства промежуточной аттестации
        4.2.1. Устный или письменный ответ на вопрос
            4.2.1.1. Порядок проведения.
            4.2.1.2. Критерии оценивания.
            4.2.1.3. Оценочные средства.
        4.2.2. Практическое задание
            4.2.2.1. Порядок проведения.
            4.2.2.2. Критерии оценивания.
            4.2.2.3. Оценочные средства.

1. Соответствие компетенций планируемым результатам обучения по дисциплине (модулю)

Код и наименование компетенции	Индикаторы достижения компетенций для данной дисциплины	Оценочные средства текущего контроля и промежуточной аттестации
ПК-3 - Способен проектировать и управлять IT-проектами, осуществлять тестирование компонентов информационных систем, в том числе интеллектуальных	Знать технологии тестирования компонентов информационных систем, методы решения задач из области искусственного интеллекта. Уметь осуществлять тестирование компонентов информационных систем решать задачи из области искусственного интеллекта; Владеть навыками тестирования компонентов информационных систем, решения задач из области искусственного интеллекта.	**Текущий контроль:** Лабораторные работы по темам Тема 1. Введение в проектирование интеллектуальных систем Тема 2. Модели представления знаний в интеллектуальных системах Тема 3. Язык логического программирования Пролог Тема 4. Экспертные системы Тема 5. Нейронные сети Эссе по темам Тема 1. Введение в проектирование интеллектуальных систем Тема 2. Модели представления знаний в интеллектуальных системах Тема 3. Язык логического программирования Пролог Тема 4. Экспертные системы Тема 5. Нейронные сети **Промежуточная аттестация:** *Экзамен*

2. Критерии оценивания сформированности компетенций

Компетенция	Зачтено			Не зачтено
	Высокий уровень (отлично)	**Средний уровень (хорошо)**	**Низкий уровень (удовлетворительно)**	**Ниже порогового уровня (неудовлетворительно)**
ПК-3	Демонстрирует высокий уровень знаний технологий тестирования компонентов информационных систем, методов решения задач из области искусственного интеллекта	Знает основы технологий тестирования компонентов информационных систем, методов решения задач из области искусственного интеллекта в базовом (стандартном) объеме	Демонстрирует частичные знания технологий тестирования компонентов информационных систем, методов решения задач из области искусственного интеллекта без грубых ошибок.	Демонстрирует частичные знания технологий тестирования компонентов информационных систем, методов решения задач из области искусственного интеллекта, допуская грубые ошибки
	Демонстрирует высокий уровень умений осуществлять тестирование компонентов информационных систем решать задачи из области искусственного интеллекта	Умеет осуществлять тестирование компонентов информационных систем решать задачи из области искусственного интеллекта в базовом (стандартном) объеме	Демонстрирует частичные умения осуществлять тестирование компонентов информационных систем решать задачи из области искусственного интеллекта без грубых ошибок	Демонстрирует частичные умения осуществлять тестирование компонентов информационных систем решать задачи из области искусственного интеллекта, допуская грубые ошибки
	Демонстрирует владение	Владеет навыками тестирования	Демонстрирует частичные владения	Демонстрирует низкий уровень

	навыками тестирования компонентов информационных систем, решения задач из области искусственного интеллекта на высоком уровне	компонентов информационных систем, решения задач из области искусственного интеллекта в базовом (стандартном) объеме	навыками тестирования компонентов информационных систем, решения задач из области искусственного интеллекта без грубых ошибок	владения навыками тестирования компонентов информационных систем, решения задач из области искусственного интеллекта, допуская грубые ошибки

3. Распределение оценок за формы текущего контроля и промежуточную аттестацию

7 семестр:
Текущий контроль:
Лабораторные работы
Тема 1. Введение в проектирование интеллектуальных систем
Тема 2. Модели представления знаний в интеллектуальных системах
Тема 3. Язык логического программирования Пролог
Тема 4. Экспертные системы Тема 5. Нейронные сети

Эссе
Тема 1. Введение в проектирование интеллектуальных систем
Тема 2. Модели представления знаний в интеллектуальных системах
Тема 3. Язык логического программирования Пролог
Тема 4. Экспертные системы Тема 5. Нейронные сети

Выполнение каждого оценочного средства оценивается по шкале: отлично, хорошо, удовлетворительно, неудовлетворительно.
Общая оценка за текущий контроль представляет собой среднее значение между полученными оценками за все оценочные средства.
Промежуточная аттестация – экзамен.
Промежуточная аттестация проводится после завершения изучения дисциплины или ее части в форме, определяемой учебным планом образовательной программы с целью оценить работу обучающегося, степень усвоения теоретических знаний, уровень сформированности компетенций.
Преподаватель, принимающий экзамен, обеспечивает случайное распределение вариантов экзаменационных заданий между обучающимися с помощью билетов и/или с применением компьютерных технологий; вправе задавать обучающемуся дополнительные вопросы и давать дополнительные задания помимо тех, которые указаны в билете.
Экзамен проводится по билетам. В каждом билете два оценочных средства: устный или письменный ответ на вопрос и решение задачи.

Выполнение каждого задания за промежуточную аттестацию оценивается по шкале: отлично, хорошо, удовлетворительно, неудовлетворительно.
Общая оценка за промежуточную аттестацию представляет собой среднее значение между полученными оценками за все оценочные средства промежуточной аттестации.
В случае невозможности установления среднего значения оценки за промежуточную аттестацию (например, «хорошо» или «отлично»), итоговая оценка выставляется экзаменатором, исходя из принципа справедливости и беспристрастности на основании общего впечатления о качестве и добросовестности освоения обучающимся дисциплины (модуля).

Виды оценок:

Для экзамена:
Отлично Хорошо
Удовлетворительно
Неудовлетворительно

4. Оценочные средства, порядок их применения и критерии оценивания
4.1. Оценочные средства текущего контроля

4.1.1. Лабораторные работы

Тема 1. Введение в проектирование интеллектуальных систем
Тема 2. Модели представления знаний в интеллектуальных системах
Тема 3. Язык логического программирования Пролог
Тема 4. Экспертные системы Тема 5. Нейронные сети

4.1.1.1. Порядок проведения.

Лабораторные работы проводятся в часы аудиторной работы.

Перед выполнением каждой работы студенты-бакалавры должны проработать соответствующий материал, используя конспекты теоретических занятий, периодические издания, учебно-методические пособия и учебники.

По окончании занятий студенты оформляют отчет по каждой работе, соблюдая следующую форму:

- Наименование темы;
- Цель работы;
- Задание и содержание выполненной работы,
- Письменные ответы на контрольные вопросы.
- Выводы по проделанной работе.
- Список использованных источников.

4.1.1.2 Критерии оценивания

Оценка «отлично» ставится, если обучающийся:
Правильно выполнил все задания. Продемонстрировал высокий уровень владения материалом. Проявлены превосходные способности применять знания и умения к выполнению конкретных заданий.
Оценка «хорошо» ставится, если обучающийся:
Правильно выполнил большую часть заданий. Присутствуют незначительные ошибки. Продемонстрирован хороший уровень владения материалом. Проявлены средние способности применять знания и умения к выполнению конкретных заданий.
Оценка «удовлетворительно» ставится, если обучающийся:
Задания выполнил более чем наполовину. Присутствуют серьёзные ошибки. Продемонстрирован удовлетворительный уровень владения материалом. Проявлены низкие способности применять знания и умения к выполнению конкретных заданий.
Оценка «неудовлетворительно» ставится, если обучающийся:
Задания выполнил менее чем наполовину. Продемонстрирован неудовлетворительный уровень владения материалом. Проявлены недостаточные способности применять знания и умения к выполнению конкретных заданий.

4.1.1.3. Содержание оценочного средства

Темы 1-5

Лабораторная работа 1. Общие сведения об языке логического программирования

Пример 1. Наша первая Пролог программа будет содержать информацию о военнослужащих некоторого воинского подразделения и их званиях: “Павлов генерал”, “Сабо полковник”, “Денисов капитан”, “Матвеев капитан”, “Кулёмин сержант”, “Николаев сержант”. Сформулировать на Прологе следующие вопросы: 1) Павлов генерал? 2) Кто является полковником? 3)Кем является Денисов? 4)В подразделение есть военный в звание сержанта? 5) В подразделение есть военный в звание подполковника? 6) Вывести военных, имеющих одинаковые звания.

В программе каждого военного мы представим предикатом military размерности 2, каждый компонент- атом, первый представляет фамилию, а второй – его звание.

Программа 3. База данных «Военная часть»

```
Domains
    s=symbol
Predicates
    military(s,s)
Clauses
    military(pavlov, general).
    military(sabo, polkovnik).
    military(denisov, kapitan).
    military(matveev, kapitan).
    military(kulemin, serzhant).
    military(nikolaev, serzhant).
```

Сформулируем запросы:

1) ? military(pavlov, general)
   Ответ: yes
2) ? military(X, polkovnik)
   Ответ: X= sabo
3) ? military(denisov, X)
   Ответ: X= kapitan
4) ? military(_, serzhant)
   Ответ: yes
5)? military(_, podpolkovnik)
   Ответ: no
6) ? military(X,Y), military(Z, Y), X<>Z
   Ответ: X= denisov Z= matveev Y= kapitan
   X= kulemin Z= nikolaev Y= serzhant

Пример 2. Данные о крупных реках России сведены в таблицу:

Таблица 5.

Данные о крупных реках России

Название реки	**Длина, км**	**Годовой сток, км^3**	**Площадь бассейна, тыс. км^2**	**Истоки**	**Куда впадает**
Амур	4416	350	1855	Яблоневый хребет	Татарский пролив
Лена	4400	488	2490	Байкальский хребет	Море Лаптевых
Обь	4070	400	2990	Предгорья Алтая	Карское море
Иртыш	4248	323	1643	Китай	Обь
Енисей	3487	600	2580	Восточный Саян	Карское море
Волга	3530	255	1360	Валдайская возвы-шенность	Каспийское море
Колыма	2129	44	643	Хребет Черского	Восточносибирское море
Урал	2428	54	231	Южный Урал	Каспийское море
Дон	2200	45	504	Среднерусская возвышенность	Азовское море
Кама	1805	130	507	Верхне — Камская возвышенность	Волга
Печора	1809	130	322	Северный Урал	Баренцево море
Ангара	1779	62	1039	Байкал	Енисей
Селенга	1024	14	447	Монголия	Байкал
Кубань	870	11	58	Кавказ	Азовское море

Составить базу данных и ответить на следующие вопросы:

1) Определить реки, впадающие в Азовское море.
2) Определить реки, исток которых находится на Валдайской возвышенности?
3) Какие реки короче Камы?
4) Какие реки длиннее Иртыша?
5) Как задать вопрос, определяющий все данные о реке Кама?

Программа 4. База данных «Реки России»

```
Domains
    S=symbol
    N=integer
Predicates
    reka(S,N,N,N,S,S)
Clauses
    reka(amur, 4416, 350, 1855,yablonevi_hrebet,tatar_proliv).
    reka(lena, 4400, 488, 2490, baikal_hrebet, more_laptevih).
```

reka(ob, 4070, 400, 2990, altai, more_karskoe).
reka(irtish, 4248, 323, 1643, kitai, ob).
reka(enisei, 3487, 600, 2580, vost_cain, more_karskoe).

reka(volga, 3530, 255, 1360, valdais_vozvishennost, more_kaspi).
reka(kolima, 2129, 44, 643, hrebet_cherskogo, vost_sibir_more).
reka(ural, 2428, 54, 231, yuzhni_ural, more_kaspi).
reka(don, 2200, 45, 504, sredn_rus_vozvvishennost, more_azov).
reka(kama, 1805, 130, 507, verhne_kamsk_ vozvvishennost,
volga).
reka(pechora, 1809, 130, 322, sever_ural, barenzevo_more).
reka(angara, 1779, 62, 1039, baikal, enisei). reka(selenga, 1024,
14, 447, mongolia, baikal). reka(kuban, 870, 11, 58, kavkaz,
more_azov).

Запросы:
reka(X, _, _, _, _, more_azov)
reka(X, _, _, _, valdais_vozvishennost,_)
reka(X, Y, _, _, _, _), reka(lena, Z, _, _, _, _), Y<Z
reka(X, Y, _, _, _, _), reka(irtish, Z, _, _, _, _), Y>Z
reka(kama, A,B,C,D,E)

Пример 3. Известно, что Лене нравится теннис, Денису нравится футбол, Борису – бейсбол, Эдику – плавание, Марку нравится теннис, а Фёдору то, что нравится Борису. Записать факты на Прологе и ответить на вопросы: 1)Кому нравится теннис? 2) Что нравится Федору? 3)Кто занимается одинаковыми видами спорта?

Программа 5. База знаний «Предпочтения»

```
Predicates
    likes(symbol,symbol)
Clauses
    likes(lena, tennis).
    likes(denis, football).
    likes(boris, baseball).
    likes(edic, swimming).
    likes(mark, tennis).
    likes(fedor, Activity):- likes(boris, Activity).
/* Activity играет роль переменной*/
```

Запросы
1) ? likes(X, tennis)
Ответ:
X= lena
X= mark
2) ? likes(fedor, X)
Ответ:
X= baseball
3) ? likes(X, T), likes(Y, T)
Ответ:
X= lena Y= mark T= tennis
X= mark Y= lena T= tennis
X= boris Y= fedor T= baseball
X= fedor Y= boris T= baseball

Пример 4. Лена, Анна, Денис и Борис-люди, лада и нисан - автомобили, Лене нравится лада, Анне - пицца, Денису - футбол, а Борис - Мерседес, Ваське - рыбка. Пицца, лада, мерседес продаются. Человек может купить машину, если она продается, и она ему нравится. Сформулировать на прологе вопросы: 1)Какую машину может купить Лена? 2) Кто-нибудь может купить мерседес? 3) какие машины продаются, и ответить на них.

Программа 6. База знаний «Предпочтения и возможности»

```
Domains
    s=symbol
Predicates
    human(s)
    car(s)
likes(s,s)
can_by(s,s)
cells(s)
Clauses
    human( lena ).
    human( anna ).
    human( denis ).
```

```
human( boris ). car(
lada ). car( nissan ).
likes( lena, lada ).
likes( anna, pizza ).
likes( denis,
football ).
likes( boris,
mersedes ).
likes( vasya, ribka
).
cells(pizza).
cells(lada).
cells(mersedes).
can_by(X,Y):-human(X), car(Y), likes(X,Y), cells(Y).
```

Запросы:

1) can_by(lena, X)
   X=lada
2) can_by(_,mersedes)
   no
3) car(X),cells(X)
   X=lada

Пример 5. Программа иллюстрирует различные способы ввода данных.

Программа 7. Ввод данных

```
Predicates
    vvod_int
    vvod_ch
    vvod_s
Clauses
    vvod_int:- readint(N1), readint(N2), N=N1+N2, write(N).
    vvod_ch:-  readchar(N1), readchar(N2), N=N1+N2, write(N).
    vvod_s:-   readln(N1), readln(N2), concat(N1,N2,N), write(N).
```

Пример 6. Вывести в каждой строке сообщения: Леонард отец Катерины, Карл отец Джейсона, Карл отец Марины.

Программа 8. База знаний «Семья»

```
Domains
    name = symbol
Predicates
    father(name, name)
    everybody
clauses
    father(leonard, katherine).
    father(carl, jason).
    father(carl, marina).
    everybody :-
        father(X, Y),
        write(X, " is ", Y, "s father\n"),
        fail.
```

В некоторых случаях может быть необходимым продолжение поиска дополнительных решений, для этого можно использовать встроенный предикат **fail**. Он не имеет аргументов, всегда считается ложным.

Проверьте работу программы с и без использования предиката fail.

Отрицание задается с помощью предиката **not**.

Пример 7. У нас есть информация о странах-партнерах Европы, имеющих общую границу. Предположим, нас интересуют какие страны-партнеры не имеют общей границы.

Программа 9. База знаний «Страны Европы»

```
Domains
    country=symbol
Predicates
    euro_pair(country, country)
    border(country, country)
    find_non_border_pair
Clauses
    euro_pair(”France”, ”Germany”).
    euro_pair(”France”, ”Spain”).
```

```
    euro_pair(”France”, ”Italy”).
    euro_pair(”Germany”, ”Spain”).
    euro_pair(”Germany”, ”Italy”).
    euro_pair(”Spain”, ”Italy”).
    border(”France”, ”Germany”).
    border(”France”, ”Spain”).
    border (”France”, ”Italy”).
    find_non_border_pair:-
            euro_pair(X,Y),
            not(border(X,Y)),
            write(X,”  – “,Y),fail,nl.
Goal
            find_non_border_pair
```

Результатом запроса будет ответ
Germany - Spain. Germany - Italy. Spain - Italy.
Пример 8. Использование составных объектов

Программа 10. База данных «Коллекция»(Вариант 1)

```
Domains
    personal_library=book(title, author, publisher, year)
    collector, title, author, publisher=symbol
    year=integer
Predicates
    collection(collector, personal_library)
Clauses
    collection(ivanov, book(”Zolushka”, ”Denis Tarakanov”, ”Dinamo”, 2003)).
    collection(petrov, book(”Maslenniza”, ”Anna Zimina”, ”Zenit”, 2005)).
    collection( petrov, book(”Repka”,”Irina Larina”, ”Mir”, 1999)).
```

/*Демонстрация двухуровневого составного объекта*/

Программа 11. База данных «Коллекция»(Вариант 2)

```
Domains
    personal_library=book(title, author, publication)
    publication= publication (publisher, year)
    collector, title, author, publisher =symbol
    year=integer
Predicates
    collection(collector, personal_library)
Clauses
collection(ivanov,book(”Zolushka”,”Den Taran”, publication (”Dina”, 2003))).
collection(petrov,book(”Maslenniza”,”Anna Zima”,publication(”Zenit”,2005))).
collection( petrov, book(”Repka”,”Irina Larina”,publication (”Mir”, 1999))).
```

/*Демонстрация использования конструкций альтернативных доменов*/

Программа 12. База знаний «Клуб по интересам»

```
Domains
    thing=misc_ thing (whatever);
            book(author, title);
            record(artist, album, type)
    person, whatever, title, author,artist, album, type =symbol
Predicates
    owns(person, thing)
    show_misc_things
    show_books
    show_records
Clauses
    owns (“Ivanov“, misc_things(“sports car“)).
    owns (“Petrov“, misc_things(“motor cycle“)).
    owns (“Smirnov“, misc_things(“piano“)).
    owns (“Ivanov”, book (“James A.Mishener“,“Space“)).
    owns (“Petrov”, book (“Frank Herbert“, “Dune“)).
    owns (“Sidorov”, book (“J.R.R. Tolkein“,“Return of the Ring“)).
```

owns (“Ivanov”, record (“Elton John”, “Ice Fair”, “popular“)).
owns (“Petrov”, record(“Michael Jackson“, “We are the World“, “popular“)).
owns (“Sidorov”, record (“Madonna“,“Madonna“, “popular“)).

show_misc_things:- owns (X, misc_things(Y)),write(X, “ ”,Y), nl, fail.
show_books:- owns (X, book (_,Y)),write(X, “ ”,Y), nl, fail.
show_records:- owns (X, record (_,Y,_)),write(X, “ ”,Y), nl, fail.

Задания для самостоятельной работы

1) Дана база данных “Родители и дети”:

родитель(полина, борис), родитель(анатолий, борис), родитель(анатолий, лиза), родитель(борис, катя), родитель(борис, валентина), родитель(полина, евгений).
Сформулировать вопросы на Прологе: Кто является родителем Кати? Есть ли у Лизы ребенок? Кто дети Бориса? Кто чей родитель?

2) Дана база данных “Теремок”:

живет (муха, горюха), живет(комар, пискун), живет(мышка, погрызуха), живет(лягушка, квакушка), живет(заюнок, кривоног),
живет(лиса, краса), живет(волк, хватыш), не_живет(медведь, пригнетыш).
Указать ответы на следующие вопросы:
?-живет(мышка, погрызуха). ? -живет(волк, X).
?-живет(X, кривоног). ?-не_живет(M,P).
Сформулировать вопросы на Прологе:_Живет ли лягушка в теремке? Какое прозвище у лисы? Кто имеет прозвище горюха? Какой следует задать вопрос, чтобы узнать обитателей теремка (без прозвищ)?

3) База данных “Рождение и хобби друзей”:
рождение(иванова, лена, 22, июнь, 1971), рождение(петров, сергей, 25, октябрь, 1973), рождение(сидорова, оля, 1, декабрь, 1974), любит(иванова, лена, книги), любит(иванова, лена, танцы), любит(петров, сергей, видео), любит(сидорова, оля, кино).
Сформулировать вопросы на Прологе:_Кто родился в 1971 году? Кто родился в октябре? Кто любит книги? Кто любит и книги и танцы?

4) База данных “Колобок”: ушел(колобок,дедушка), ушел(колобок, бабушка), ушел(колобок, заяц), ушел(колобок, волк), ушел(колобок, медведь), не_ушел(колобок, лиса).
Сформулировать вопросы на Прологе:_Кто ушел от волка? Кто не ушел от лисы? Кто ушел от волка и от бабушки? Какой следует задать вопрос, чтобы узнать всех персонажей сказки?

5) База данных “Распорядок дня”:
занятие(0, 7, сон), занятие(7, 8, завтрак), занятие(8, 13, школа), занятие(13, 14,обед), занятие(14, 19, свобода), занятие(19, 20, ужин), занятие(20, 23, отдых), занятие(23, 24, сон).

Сформулировать вопросы на Прологе: Когда бывает обед? Что бывает между 14 и 19 часами? Когда бывает сон? (сколько будет решений?)

6) Построить базу данных “Важнейшие события Древнего Мира” на основе установленных фактов, произошедших с 31 по 6 век до нашей эры.

Каждый факт приводить в виде событие(X,Y,Z), где X — название государства, где произошло событие, Y - в каком веке произошло событие, Z — какое произошло событие.

В 31-м веке до нашей эры возникли первые города-государства. Единое государство в Египте образовалось в 30 веке до нашей эры. В 27 веке до нашей эры в Индии появились первые древнейшие города, а в Египте построена пирамида Хеопса. Первые греческие государства появились в 18 веке до нашей эры. В этом же веке в Египте произошло крупное восстание бедняков и рабов. В 15 веке до нашей эры появились первые государства в Китае. Тутмос III правил в Египте в 15 веке до нашей эры. Греция вела троянскую войну в 13 веке до нашей эры. Вторжение борийских племен в Грецию произошло в 11 веке до нашей эры. В 8 веке до нашей эры был основан город Рим. Олимпийские игры стали проводиться в Греции в 8 веке до нашей эры. В 6 веке до нашей эры в Риме была установлена республика, а в Греции произошли реформы Солона. В этом же веке персы взяли Вавилон в Междуречье и завоевали Египет.

1. Составить 3 запроса к этой базе данных.
2. Какие события произошли в период с 15 до 7 в. до н.э.

7) В таблице даны некоторые характеристики движения планет Солнечной системы(числовые величины округлены):

Таблица 6.

Характеристики движения планет солнечной системы

Планета	Расстояние до Солнца (у.е.)	Период обращения	Средние солнечные сутки
Меркурий	39	88 суток	176 суток
Венера	72	225 суток	117 суток
Земля	100	365 суток	24 часа
Марс	152	687 суток	25 часов
Юпитер	520	12 лет	10 часов
Сатурн	954	29 лет	10 часов
Уран	1920	84 года	24 часа
Нептун	3010	165 лет	22 часа
Плутон	3950	247 лет	6 суток

Составить базу данных, учитывая измерение по некоторым параметрам в разных еденицах.

Ответить на вопросы: Какие планеты ближе к Солнцу, чем Земля? Какие планеты дальше от Солнца, чем Земля? На каких планетах солнечные сутки меньше, чем земные? На каких планетах период обращения измеряется в годах?

8) Построить базу знаний “Рабочая смена”:

Мария работает в дневную смену. Сергей работает в вечернюю смену. Борис работает в вечернюю смену. Валентина работает в вечернюю смену. Два служащих знают друг друга, если они работают в одну смену. Определить: Знает ли Сергей Бориса? Кого знает Валентина? Кого знает Мария?

9) Даны результаты сдачи экзаменов для группы из пяти учеников:

Таблица 7.

	Успеваемость		
Фамилия	Алгебра	Геометрия	История
Бобров	5	3	2
Вяткин	5	5	5
Кротов	2	3	3
Соснин	4	4	4
Вавилов	4	2	1

Построить базу знаний о результатах экзаменов, определив в ней следующие правила:

отличник - человек, у которого по всем предметам пятерки;

двоечник - есть хотя бы одна двойка;

математик - по алгебре и по геометрии учится на 4 и 5;

введите предикаты алгебра, геометрия, история для определения оценки Y для ученика X.

Получить ответы на следующие вопросы: Является ли Вяткин отличником? Определить всех отличников. Является ли Соснин математиком? Определить всех неуспевающих по истории.

10) Сформировать базу знаний “Квартет” из следующих фактов и правил:

Мартышка играет на скрипке. Осел играет на альте. Козел играет на виолончели. Мишка играет на контрабасе. Четверо музыкантов X,Y,Z и W могут образовать квартет, если один из них играет на скрипке, другой — на альте, третий — на виолончели и четвертый — на контрабасе.

Ответить на вопросы: Кто играет на альте? На чем играет мартышка? Образуют ли квартет Мартышка, Осел, Козел и Мишка? Кто из музыкантов данной базы знаний может образовать квартет?

11) База знаний “Воинская служба”:

возраст(борис ,18), возраст(андрей, 17), возраст(михаил,18), возраст(анна,18), возраст(юлия ,17), мужчина(андрей), мужчина(борис), мужчина(михаил), женщина(анна), женщина(юлия). Определить правило подлежит призыву, не_подлежит_призыву.

Сформулировать вопросы: Кто подлежит призыву? Подлежит ли призыву Анна?

12) База знаний "Семья”:

мать(екатерина, юлия), мать(екатерина, мария), мать(анна, екатерина), отец(петр, юлия), отец(виктор,петр), отец(андрей, екатерина).

Дополните базу данных предикатом мужчина, женщина. Определите правила дед, бабушка, внук, внучка, тетя, дядя.

Сформулировать вопросы на Прологе:

Кто является ребенком Екатерины и Петра? Кто является дедом Юлии? Кто является бабкой Юлии?

13) Построить базу знаний и сформулировать к ней вопросы, основываясь на следующих утверждениях: Резвый - это собака. Рекс-это собака. Белка – это кошка. Быстрая - это лошадь. Резвый - черная. Белка - белая. Рекс -рыжая. Быстрая - белая. Домашние животные - это собака или кошка. Животные – это либо лошадь, либо домашние животные. Том владеет тем, кто собака и не черного цвета. Кейт владеет тем, кто либо черного цвета, либо лошадь.

14) Построить базу знаний. Муська — коричневая кошка, Стрелка — черная кошка, Мурка — рыжая кошка. Рекс, Дружок и Мухтар — собаки. Дружок — рыжая, Мухтар — белая. Все животные, которыми владеют

Анатолий и Николай, имееют родословные. Анатолий владеет всеми черными и коричневыми животными, а Николай владеет всеми собаками небелого цвета, которые не являются собственностью Анатолия. Иван владеет Муркой, если Николай не владеет Муськой и если Мухгар не имеет родословной. Рекс — пятнистая собака. Определить, какие животные не имеют хозяев.

15) Составить базу знаний “Знакомства” из следующих фактов и правил:

Мери прелестна. Джон добрый. Джон мужественный. Джон сильный. Некто счастлив, если богатый или нравится женщинам. Мужчина нравится женщине, если женщина нравится мужчине и он добрый, либо мужчина добрый и сильный. Мужчине нравится женщина, если она прелестна.

Сформулировать вопрос: счастлив ли Джон? Найти мужчин, которые могут нравиться женщинам.

16) Ответьте на вопрос “Что ест моя кошка?“, если:

Птицам нравятся червяки. Кошкам нравятся рыбы. Друзья нравятся друг другу. Моя кошка мой друг. Моя кошка ест все, что ей нравится.

17) Запрограммируйте утверждения.

Число четное. Число не четное. Ни одно число не является четным и нечетным одновременно. Число не четное, если следующее за ним четное. Число, следующее за данным числом нечетное, если данное число четное, число, следующее за данным числом четное, если данное число нечетное.

Лабораторная работа 2. Арифметика. Управление логическим выводом в программах

Пример 1. Описать предикаты для вычисления суммы, разности, произведения, частного двух чисел, возведения числа в квадрат, вывода остатка при деление на 3, вывод случайного числа из интервала [1,100].

Программа 13. Арифметика

```
Domains
    N=integer
    R=real
Predicates
    add(N,N)
    sub(N,N,N)
    multi(N,N,N)
    division(N,N,R)
    kvadrat(N,N)
    ostat(N,N)
    vivod(N)
Clauses
    add(X,Y):-S=X+Y, write(“Sum=  ”,S),nl.
    sub(X,Y,S):-S=X-Y.
    multi(X,Y,P):-P=X*Y.
    division(X,Y,R):-Y<>0, R=X/Y.
    kvadrat(X,N):-N=X*X.
    ostat(X,N):-N=X mod 3.
    vivod(N):-random(Y), N=1+Y*100.
```

Пример 2. Программирование взаимоисключающих утверждений. Процедуру нахождения наибольшего из двух чисел можно записать в виде отношения

```
max(X,Y,X):-X>=Y. max(X,Y,Y):-X<Y.
```

Эти правила являются взаимоисключающими. Возможна более экономная формулировка: если X>=Y, то максимум=X, иначе =Y. На Прологе это запишется следующим образом:

```
max(X,Y,X):-X>=Y, !.
max(_,Y,Y).
```

Программа 14. Максимум

```
Domains
    N=integer
Predicates
    max(N,N,N)
Clauses
    max(X, Y, X):-X>Y,!.
    max(_,Y,Y).
```

Пример 3. Рассмотрим различные способы записи предиката different, определяющего различны ли числа, использующие сочетание встроенных предикатов ! и fail.

```
different(X,X):-!,fail.
different(_,_).
```

или
different(X,Y):-X=Y,!,fail.
different(_,_).
или
different(X,Y):-X=Y,!,fail; true.
/* true –встроенный предикат, который всегда истиннен*/
или
different(X,Y):-not(X=Y).

Задания для самостоятельной работы

№1. Описать предикаты:

1) Найти квадрат числа X; куб числа X.
2) Найти значение функций y = a*x + b, где a, b и x — целые числа.
3) Найти периметр треугольника, если все его стороны известны.
4) Найти площадь прямоугольного треугольника по двум его катетам.
5) Найти площадь трапеции с основаниями A и B и высотой H.
6) Найти квадрат гипотенузы в прямоугольном треугольнике по двум его катетам.
7) Найти объем прямоугольного параллелепипеда со сторонами A, B и C.
8) Зная скорость V и время T, определите путь.
9) Найти последнюю цифру в записи натурального числа.
10) Найти цифры в десятичной записи двузначного натурального числа.
11) Найти первую цифру в десятичной записи трехзначного натурального числа.
12) Найти сумму цифр в десятичной записи трехзначного натурального числа.

№2.

1) Найти

А. Наименьшее значение из двух чисел;
Б. Наименьшее значение из трех чисел на основе первой задачи;
В. Наименьшее значение из шести чисел на основе второй задачи.

2) Определить, удовлетворяют ли длины трех отрезков условию прямоугольного треугольника.
3) Определить, удовлетворяют ли длины трех отрезков условию треугольника.
4) Найти модуль числа X.
5) Описать предикат для вычисления функции, заданной соотношением:

$$f(x)=\begin{cases} x^2, & x<1 \\ x-1, & 1<x<1 \\ x, & x>1 \end{cases}$$

Лабораторная работа 3. Повторение и рекурсия

Пример 1. Выводить на экран квадрат числа, вводимого пользователем, пока не будет введен 0.

Программа 15. Вывод квадрата числа

```
Domains
    x=integer
Predicates
    go
    repeat
    check(x)
Clauses
    repeat.
    repeat:- repeat.
    go:-  repeat, write (“Введите число пожалуйста или 0-для выхода   ”),
```

```
            readint(A), check(A).
    check(0):-nl,write("ok"),!.
    check(A):-B=A*A, write(B),nl, fail.
```
Или
```
go:-repeat, write ("Введите число   "), readint(A),
            B=A*A, write(A, "^2= " ,B),nl,
            write("прервать да/нет (y/n) "), readchar(C), C= "n",!.
```

Пример 2. Организовать меню для выбора арифметической операции.

Программа 16. Меню

```
Domains
    x=integer
Predicates
    menu
    actions(x)
    repeat
Clauses
    repeat.
    repeat:- repeat.
    menu:- repeat,
            write("Введите 1 для +, 2 для *, 3-для -, 0-для выхода\n"),
            readint(N), actions(N).
    actions(0):-!.
    actions(1):-write("Введите первое число"), nl, readint(A),
                write("Введите второе число") nl, readint(B),
                C=A+B,write(C),nl,fail.
    actions(2):- write("Введите первое число"), nl, readint(A),
                write("Введите второе число") nl, readint(B),
                C=A*B,write(C),nl,fail.
    actions(3):- write("Введите первое число"), nl, readint(A),
                write("Введите второе число") nl, readint(B),
                C=A-B,write(C),nl,fail.
```

Пример 3. Вычислить n-ый член последовательности Фибоначчи. N-ый член последовательности Фибоначчи, начиная с третьего, определяется суммой 2-х предыдущих, а 1-ый и 2-ой члены равняются единице.

Введем двуместный предикат fib, первый аргумент будет определять порядковый номер члена, а второй будет записываться для записи ответа. Введем два факта, первый - первый член последовательности Фибоначчи равен 1, второй - второй член последовательности равен 1, а для определения n-го члена запишем правило. Действительно, чтобы определить n-ый член, мы должны определить значения двух предыдущих и сложить их.

```
fib(1,1).
fib(2,1).
fib(N,F):- N1=N-1, fib(N1,F1), N2=N-2, fib(N2,F2), F=F1+F2.
? fib(1,F)
```

Ответом будет F=1, и Пролог сделает попытку сопоставить с запросом второй факт и потерпит неудачу. Однако сопоставление головы третьего утверждения с запросом происходит успешно и осуществляется попытка доказать цель fib(-1,F1), что в свою очередь, приводит к цели fib(-2, …)и так далее, т.е. образуется бесконечный цикл. Эту ситуацию можно устранить, используя отсечение и тем самым, указывая Прологу, что не существует других решений в случае успешного согласования граничного условия.

Программа 17. Последовательность Фибоначчи

```
Domains
    x=integer
Predicates
    fib(x,x)
Clauses
    fib(1,1):-!.
    fib(2,1):-!.
    fib(N,F):- N1=N-1, fib(N1,F1), N2=N-2, fib(N2,F2), F=F1+F2.
```

Задания для самостоятельной работы

1) Вычислить N!.
2) Вычислить n-ый член последовательности Фибоначчи.
3) Вывести все числа от n до 1. 4) Вывести все числа от 1 до n.
5) Вычислить сумму чисел от 1 до n.
6) Определите x^n, n>0. 7) Определите 2^n, n>0. 8) Определите N^5, n>0.
9) Вычислите сумму четных чисел от 1 до n.
10) Вычислите сумму квадратов нечетных чисел от 1 до n.
11) Вычислите сумму a_k, где $a_k=1/(1+k)$.

12) Вычислить $\sum_{i=1}^{N} x_i$.

13) Вычислить $\sum_{i=1}^{N} i$.

14) Определите корень уравнения методом половинного деления.
15) Найти наибольший общий делитель двух чисел, трех чисел.
16) Определить число сочетаний C_n^m.
17) Вычислить N!+(N-1)!+...+2!+1!.
18) Вычислить количество четных элементов на заданном интервале.
19) Перевести число из десятичной системы счисления в систему с основанием N, где N<10, N>1.

Лабораторная работа 4. Применение рекурсии для обработки списков

Пример 1. Определить количество элементов в списке.

Количество элементов пустого списка равно 0, в противном случае, нужно определить количество элементов в хвосте и найденное значение увеличить на единицу.

Программа 18. Определение длины списка

```
Domains
    list = integer*
Predicates
    length_of(list, integer)
Clauses
    length_of([], 0):-!.
    length_of([_|T], N) :- length_of(T, N1),N = N1 + 1.

?- length ([2, 12, 45]), X).
```

Ответ: X=3.

Пример 2. Найти сумму элементов в списке.

Сумма элементов пустого списка равна 0, сумма элементов непустого списка определяется сложением головы списка с суммой элементов хвоста.

```
    sum ([], 0):-!.
    sum ([H|T], S) :- sum (T, S1), S = S1+H.
    ?-sum ([2, 12, 45]), X).
```

Ответ: X=59

Пример 3. Определить принадлежность элемента списку.

Программа 19. Определение принадлежности элемента списку

```
Domains
    type=integer
    list=type*
Predicates
    member(type,list)
Clauses
    member(H,[H|_]):-!.
    member(H,[_|T]):-member(H,T).

Запрос:
member (4,[1,3,4,9]).
```

Ответ: Yes.

Данная программа имеет очень простой декларативный смысл: элемент принадлежит списку, если он является его головой или принадлежит хвосту списка.

Пример 4. Объединить два списка.

Эту задачу можно описать с помощью следующих предикатов:
а) если к пустому списку [] добавить список P, то в результате получится P;

б) чтобы добавить список P к концу списка [X|Y], нужно P добавить к хвосту Y и затем присоединить к голове X, при этом получается список [X|T].

Программа 20. Объединение двух списков

```
Domains
    type=integer
    list=type*
Predicates
    append(list,list,list)
Clauses
    append([],L,L):-!.
    append([H|T],P,[H|Y]):-append(T,P,Y).

? append ([3, 5, 7], [12,5],K).
```

Ответ: K=[3,5,7,12,5]

Процедуру append можно использовать:

1) для разбиения списка на подсписки:

```
? append (X,Y,[1,2])
```

Ответ: X=[ ]Y=[l,2]

X=[l] Y=[2]

X=[l,2] Y=[]

2) Для вывода последнего элемента списка:

```
last(X,L):-append(_,[X],L)
```

Пример 5. Удаление элементов из списка.

Первый аргумент указывает удаляемый элемент, второй аргумент - исходный список и третий - список-результат.

Программа 21. Удаление элементов из списка

```
Domains
    type=integer
    list=type*
Predicates
    delete(type,list,list)
Clauses
    delete(X,[X|T],T):-!.
    delete(X,[Y|T],[Y|T1]):-delete(X,T,T1).
```

Смысл: результат удаления произвольного элемента из пустого списка, есть пустой список, если удаляемый элемент совпадает с головой списка, то результатом программы является хвост списка, иначе удаление производится из хвоста списка.

Данная программа удаляет первое вхождение элемента в список. Знак отсечения "!" в конце правила предотвращает последующий поиск и вывод лишних вариантов ответов после выполнения ограничительного факта.

Для удаления всех вхождений элемента X программу надо дополнить:

```
delete(_, [], []) :-!.
delete(X, [X | T], W) :- delete(X, T, W),!.
delete(X, [Y | T], [Y|W]) :- delete(X, T, W) .
```

Смысл программы таков: пока список не пуст, удалить элемент, если он совпадает с головой списка, а затем удалять его из хвоста, иначе надо сразу удалять элемент из хвоста.

Пример 6. Индексация элементов списка.

Определить номер элемента X.

```
nomer([X |_], 1, X):-!.
nomer ([_| T], N, X) :- nomer(T, N1, X), N=N1+1.
```

Пример 7. Вывести максимальный элемент.

```
max ( [X] , X) .
max ([X | T], X) :- max (T, M), X>M, !.
max ( [_| T] , M) :- max (T, M) .
```

Смысл программы: если в списке один элемент - он и является максимальным, если более одного, то это голова списка, если она больше максимального элемента хвоста, или максимальный элемент хвоста.

Пример 8. Обращение списка.

Обратить список из одного элемента - означает оставить список без изменения. Обратить более длинный список - обратить его хвост, а потом справа добавить к нему голову исходного списка.

```
reverse([X], [X]):-!.
reverse ([X | T], Z) :- reverse (T, W), append(W, [X], Z).
```

Пример 9. Иногда бывает полезно представить в виде списка информацию, содержащуюся в известных фактах. В Прологе для этой цели служит предикат **findall**. Полный формат предиката: findall(Переменная, Атом, Список), здесь Атом- отношения, которое определяет список элементов, конкретизирующих переменную в качестве аргумента предиката атом. Рассмотрим предикат на примере вычисления общей суммы окладов служащих

Прорамма 22. Сумма окладов служащих

```
Domains
    i=integer
    sp=i*
    fam,im,dolgnost=symbol
Predicates
    sum(sp,i)
    go
    spisok(fam,im,dolgnost,i)
Clauses
    sum([],0):-!.
    sum([H|T],S):-sum(T,S1),S=S1+H.
    spisok(ivanov, ivan, slesar,400).
    spisok(petrov, petr,tokar,200).
    spisok(sidorov, denis,plotnik,100).
    go:-findall(X,spisok(_,_,_,X),L), sum(L,A).
```

Задания для самостоятельной работы

1) Определите количество элементов в списке.
2) Определите сумму элементов списка
3) Определите количество нечетных элементов в списке.
4) Определите, принадлежит ли заданный элемент списку.
5) Определите, сколько раз заданный элемент входит в список.
6) Выведите максимальный элемент.
7) Выведите голову списка.
8) Выведите последний элемент.
9) Замените голову списка.
10) Определите номер элемента X.
11) Выведите элемент под номером N.
12) Удалите из списка все вхождения заданного элемента.
13) Объедините два списка.
14) Перепишите список в обратном порядке.
15) Объедините два списка без дублирования элементов.
16) Удалите первое вхождение заданного элемента.
17) Сложить поэлементно 2 списка.
18) Сложить два списка следующим образом: a_1+b_n, a_2+b_{n-1}, ...,$a_{n-1}+b_2$, a_n+b_1.
19) Найти количество элементов, предшествующих первому(последнему) максимальному.
20) Переместите голову списка в конец списка.
21) Найти сумму максимального и минимального элементов списка.
22) Поменяйте местами элементы с нечетными индексами с элементами с четными индексами.
23) Составить список из цифр заданного числа в обратном порядке. Например, 127645: [5,4,6,7,2,1].
24) Выполните N последовательных перестановок головы в конец списка.
25) Увеличьте каждый элемент списка на заданный элемент.
26) Увеличьте элемент с заданным номером на заданное число.
27) Все вхождения заданного элемента уменьшите на заданное число.
28) Удалите элемент с заданным номером N.
29) Создать список, элементами которого являются 2^n, n!, члены последовательности Фибоначчи.
30) Определите среднее элементов списка. 31) Замените четные элементы списка нулем.
32) Определите сумму элементов, больше заданного N.
33) Отсортируйте список методом пузырька. 34)

Отсортируйте список методом вставками.

35) Отсортируйте список быстрым методом сортировки.

36) Используя предикат findall, решите следующие задачи:

1. Вывести самых молодых жильцов дома и номера квартир, в которых они живут.
2. Вывести фамилии студентов и их возраст с максимальным размером стипендии.
3. Вывести фамилии сотрудников предприятия и их оклады, оклады которых меньше среднего.
4. Вывести студентов с заданной фамилией и посчитать их количество.

Лабораторная работа 5. Решение логических задач.

Пример 1. Беседует трое друзей: Белокуров, Рыжов, Чернов. Брюнет сказал Белокурову: “Любопытно, что один из нас блондин, другой брюнет, третий - рыжий, но ни у кого цвет волос не соответствует фамилии”. Какой цвет волос у каждого из друзей?

Для решения построим вспомогательную таблицу:

Таблица 8.

Вспомогательная таблица соответствия

Фамилия / Цвет	Белокуров	Рыжов	Чернов
блондин	-		
рыжий		-	
брюнет	-		-

Выводы:

1) Белокуров не брюнет и не блондин;
2) Чернов не черный, цвет волос Чернова и Белокурова не совпадают;
3) Рыжов не рыжий, цвет волос у Рыжова и Белокурова, Рыжова и Чернова не совпадают.

Программа 23. Логическая задача на соответствие

```
Predicates
    fam(symbol)
    color(symbol)
    cootvet(symbol, symbol)
Clauses
    fam(belokurov).
    fam(ryzov).
    fam(chernov).
    color(ryziy).
    color(cherniy).
    color(beliy).
    cootvet(X,Y):- fam(X), color(Y), X=belokurov,
                   not(Y=cherniy), not(Y=beliy).
    cootvet(X,Y):- fam(X), color(Y), X= chernov,
                   not(Y=cherniy),not(cootvet(belokurov,Y).
    cootvet(X,Y):- fam(X), color(Y), X= ryzov,
                   not(cootvet(belokurov, Y), not(cootvet(chernov, Y).
```

Пример 2. На скамейке сидели Петя, Боря, Коля. Петя справа от Бори, Боря справа от Коли. Кто сидел посередине? Кто сидел с правого(левого) края? Кто сидел между указанными объектами(увеличьте число объектов)?

Программа 24. Логическая задача на выяснение порядка

```
Predicates
    rayd(symbol, symbol, symbol)
    sprava(symbol, symbol)
    seredina(symbol
    kr_cl(symbol) kr_cpr(symbol)
Clauses
    sprava(kolya, boray).      /*Справа от Коли Боря*/
    sprava(boray, petay).
    rayd(X,Y,Z):-     sprava(X,Y), sprava(Y, Z).
    seredina(X):-  rayd(_,X,_). kr_cl(X):-
    rayd(X,_,_). kr_cpr(X):- rayd(_,_,X).
```

Задания для самостоятельной работы

1) Коля и Саша носят фамилии Шилов и Гвоздев. Какую фамилию носит каждый из них, если Саша с Шиловым живут в разных домах.

2) В соревнованиях по бегу Юра, Гриша и Толя заняли три первых места. Какое место занял каждый ребенок, если Гриша занял не второе и не третье место, а Толя не третье?

3) Три подруги вышли в белом, зеленом и синем платьях и туфлях. Известно, что только у Ани цвета платья и туфлей совпадали. Ни туфли, ни платье Вали не были белыми. Наташа была в зеленых туфлях. Определить цвета платья и туфель на каждой из подруг.

4) На заводе работали три друга: слесарь, токарь и сварщик. Их фамилии Борисов, Иванов и Семенов. У слесаря нет ни братьев, ни сестер. Он самый младший из друзей. Семенов, женатый на сестре Борисова, старше токаря. Назвать фамилии слесаря, токаря и сварщика.

5) В бутылке, стакане, кувшине и банке находятся молоко, лимонад, квас и вода. Известно, что вода и молоко не в бутылке, сосуд с лимонадом находится между кувшином и сосудом с квасом, в банке - не лимонад и не вода. Стакан находится около банки и сосуда с молоком. Как распределены эти жидкости по сосудам.

6) Воронов, Павлов, Левицкий и Сахаров – четыре талантливых молодых человека. Один из них танцор, другой художник, третий-певец, а четвертый-писатель. О них известно следующее: Воронов и Левицкий сидели в зале консерватории в тот вечер, когда певец дебютировал в сольном концерте. Павлов и писатель вместе позировали художнику. Писатель написал биографическую повесть о Сахарове и собирается написать о Воронове. Воронов никогда не слышал о Левицком. Кто чем занимается?

7) Три друга заняли первое, второе, третье места в соревнованиях универсиады. Друзья разной национальности, зовут их по-разному, и любят они разные виды спорта. Майкл предпочитает баскетбол и играет лучше, чем американец. Израильтянин Саймон играет лучше теннисиста. Игрок в крикет занял первое место. Кто является австралийцем? Каким спортом увлекается Ричард?

8) Три девочки Маша, Рита, Лена пошли гулять. На улице было жарко, и они купили мороженое «Белка», «Стрелка», «Гагара». Какое мороженое купила каждая из девочек, если Лена купила не «Белку» и не «Гагару», а Рита – не «Гагару».

9) В комнате находятся Коля, Света, Оля. Каждый из них сидит на отдельной мебели (кровать, стул, диван). Известно, что Коля сидит не на стуле и не на кровати. Света не сидит на стуле. Кто где сидит?

10) На столе лежат ручка, карандаш, фломастер, красного, синего и зеленого цвета. Известно, что ручка лежит между предметом красного и зеленого цвета. Карандаш либо зеленый, либо синий.

11) Однажды в Артеке за круглым столом оказалось пятеро ребят из Москвы, Санкт-Петербурга, Новгорода, Перми, Томска: Денис, Игорь, Иван, Алеша, Сергей. Москвич сидел между томичем и Сергеем, санкт-петербуржец - между Денисом и Игорем, а напротив него сидел пермяк и Иван. Алеша ни разу не был в Санкт-Петербурге, а Денис не бывал в Москве и Томске, а томич с Игорем регулярно переписываются. Определите, кто в каком городе живет каждый из ребят.

12) На улице, встав в кружок, беседует четыре девочки: Аня, Валя, Надя, Галя. Девочка в зеленом платье – не Аня и не Валя - стоит между девочкой в голубом платье и Галей. Девочка в белом платье стоит между девочкой в розовом платье и Валей. Какого цвета платье у каждой из девочек?

13) Трое юношей: Коля, Дима и Юра влюблены в трех девушек: Аню, Лену, Вику. Но эта любовь без взаимности. Коля любит девушку, влюбленную в юношу, который любит Лену. Дима любит девушку, влюбленную в юношу, который любит Вику. Лена не любит Юру.

14) Составить базу знаний по сказке “Репка”. Фактами в этой базе должны быть утверждения типа тянет(X,Y). Составить правила, определяющие: кто первый тянет репку, кто последний тянет реку, кто тянет после бабки, кто тянет на четвертом месте.

15) Даны числа X, Y, Z, T. X меньше Yи меньше T; Y больше Z и больше T; Z больше X и меньше T. В каком порядке расположены эти числа.

Лабораторная работа 6. Головоломки. Игровые программы.

Пример 1. Нарисовать конверт, не отрывая карандаша от бумаги и не проводя два раза по одной и той же линии. Введем обозначения, ребра – латинскими буквами, вершины - цифрами.

Знания о структуре графа можно представить следующим образом: rebro(a,1,2), rebro(a,2,1), rebro(b,2,3), rebro(b,3,2), rebro(c,1,3), rebro(c,1,2), rebro(c,3,1), rebro(d,2,4), rebro(d,4,2), rebro(e,2,5), rebro(e,5,2), rebro(f,3,4), rebro(f,4,3), rebro(g,3,5), rebro(g,5,3), rebro(h,4,5), rebro(h,5,4).

Решением задачи должен явиться список пройденных ребер графа, причем длина его должна быть равна 8, и в нем не должно быть повторяющихся ребер, что можно описать так:

```
prohod(T,P):-length(P,8),write_list(P),!.
prohod(T,P):-rebro(R,T,H), not(member(R,P), prohod(H,[R|P).
```

T-текущая вершина графа, а P-список пройденных ребер. Первое правило означает, что если длина списка P пройденных вершин становится равной 8, то список P выводится на печать. Это правило ограничивает рекурсивный перебор вершин и ребер, проводимый вторым правилом. Второе правило является генератором перебора, оно перебирает предикаты “rebro()” и находит такое ребро R, из текущей вершины T в новую H, чтобы оно не принадлежало списку P, зато это ребро добавляется в качестве головы к списку P, и поиск дальнейшего пути производится уже из новой вершины H.

Пример 2. Программа “Угадай число”.

Программа 25. Логическая игра «Угадай число»

```
Predicates
        play_the_game give_info
        play_it
        generate_rand(integer)
        play_it_sam(integer)
        test_and_tell(integer,integer)
        say_you_got_it_right(integer
        )
        say_too_big say_too_small
Clauses
        play_the_game:-give_info, play_it.
                /*информация для пользователя*/
        give_info:-makewindow(1,7,7,"",0,0,25,80),
                makewindow(2,7,7,"Угадай число",
        2,20,22,45),
                nl, write("Это игра Угадай число"),
        nl, write("Я загадаю число из"), nl,
        write("интервала [1,100]"),
                nl, write("Я буду подсказывать больше"),
                nl, write("или меньше задуманное
                число"),nl,
                nl, write("Кода будешь готов- нажми
                space bar"),
                nl,readchar(_), clearwindow.
                        /*выполнение игры*/
        play_it:-generate_rand(A),
                play_it_sam(A).
                        /*генерация случайного числа*/
        generate_rand(X):-
                random(R),
                X=1+R*100,
                nl,write("Я загадал число"),
                nl,write("Теперь дело за Вами"),nl.
                        /*запрос к пользователю */
        play_it_sam(X):-
                nl, write ("Введите свой вариант"),
                nl, readint(G), test_and_tell(X,G),
                play_it_sam(X).
                        /* проверка и выдача результата*/
        test_and_tell(X,X):-  say_you_got_it_right(X),!.
        test_and_tell(X,Y):-  X>Y, say_too_big,!.
        test_and_tell(_,_):- say_too_small.
                        /*выдача сообщений*/ say_too_big:-nl,
        write("Загаданное число больше").
        say_too_small:-nl, write("Загаданное число меньше").
        say_you_got_it_right(X):-nl, write("Вы правы"),
                                        nl,write("Я загадал ",X),
                                        nl, write("До новых встреч!"),
                                        nl, write("Нажмите space bar"),
                                        nl,readchar(_),
                                        exit.
Goal
        play_the_game.
```

Пример 3. Игра Баше – 23 спички. На столе 23 спички, 2 игрока по очереди берут от 1 до 3 спичек, проигравшим считается тот, кто возьмет последнюю спичку.

Программа 26. Логическая игра «Игра Баше – 23 спички»

```
Predicates
        play_the_game
        do_windows
        play(integer,integer,integer)
```

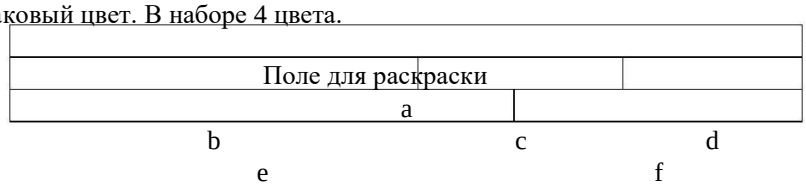
```
    play_it_again(integer,integer,integer)
    real_int(real,integer)

Clauses
    play_the_game:-do_windows,play(23,0,0).
        /*правило для образования окон*/
    do_windows:-makewindow(1,7,7,"",0,0,25,80),
        makewindow(2,7,7,"Игра 23 спички",
    1,5,22,40),
        nl, write("Добро пожаловать!"),
    M=23, H=0, C=0,
        nl, write("Всего ",M," спички" ),
        nl, write("По очереди мы будем перемещать " ),
        nl, write("спички в свои кучки"),
        nl, write("За раз можно взять 1, 2, 3 спички"),nl,
        nl, write("Проиграет, тот кто заберет последнюю спичку"),
        nl, write("Итак, начнем, всего ", M, "сп."),nl,
        nl, write("я взял ",C," сп. Вы взяли ",H, " сп."),nl.
    play(M,H,C):-play_it_again(M,H,C).
        /*правило определения победителя*/
    play_it_again(M,_,_):-
        M<=0,
        nl, write("Вы выиграли!"),
        nl, write("Нажмите любую клавишу для выхода"),
        readchar(_), clearwindow,!,exit.
    play_it_again(1,_,_):-
        nl, write("Я выиграл!"),
        nl, write("Нажмите любую клавишу для выхода"),
        readchar(_), clearwindow,!,exit.
    play_it_again(M,_,_):-
        nl, write("Ваш ход"),
        nl, write("Сколько спичек вы хотите взять?"),
        readint(Hn),
        M2=M-Hn, H2=Hn,
        write("Осталось ",M2,"сп."),
        nl,write("Мой ход"),
        random(F), Rea=1+3*F,
        real_int(Rea,Rint), M3=M2-
        Rint,
        nl, write("Я беру ",Rint," сп."),
        nl, write("Осталось ",M3, "
        сп."),nl,
        M7=M3,H7=H2,C7=Rint,
        play_it_again(M7,H7,C7).
        /*вспомогательное правило*/
    real_int(Re,In):-In=trunc(Re).
Goal
    play_the_game
```

Задания для самостоятельной работы

1) Отгадать число, загаданное компьютером, максимум за 3 попытки.
2) 23 спички. Разработать выигрышную стратегию для компьютера.
3) Имеется два сосуда – на 3 и 5 литров. Как отмерить с их помощью 4 литра воды?
4) Решите задачу о перевозке фермером через реку волка, козы, капусты.
5) Раскрасить плоскую карту так, чтобы никакие две смежные области на ней не были раскрашены в одинаковый цвет. В наборе 4 цвета.

Таблица 9.



6) Лабиринт задан матрицей соединений, в которой для каждой пары комнат указано, соединены ли они коридором. Построить путь перехода из комнаты “a” в комнату “g”:

Таблица 10.

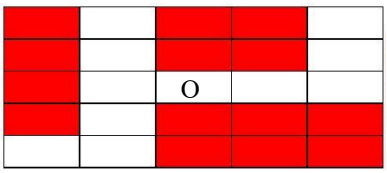
Поле для лабиринта

	A	B	C	D	E	F	G
A	-	1	0	0	0	0	0
B	1	-	1	0	1	0	0
C	0	1	-	1	0	0	0
D	0	0	1	-	1	0	0
E	0	1	0	1	-	0	0
F	0	0	0	0	1	0	0
G	0	0	0	0	1	0	-

7) Имеется n городов. Некоторые из них соединены дорогами известной длины. Вся система дорог задана квадратной матрицей n*n, элемент aij<0, если между городами дороги нет, в противном случае равен расстоянию между городами.

8) Задан лабиринт. Напишите программу для нахождения маршрутов для выхода из этого лабиринта. Начало маршрута точка О. Закрашенная клетка – стена.

Таблица 11.



9) Быки и коровы. Игрок А выбирает секретный код, представляющий последовательность из N различных десятичных цифр (N=4). Игрок пытается угадать задуманный код и спрашивает игрока А о числе “быков” (“быки” - количество совпадающих цифр в одинаковых позициях предполагаемого и задуманного кодов; число “коров”- количество совпадающих цифр, входящих в предполагаемый и задуманный код, но находящиеся на разных позициях).

Лабораторная работа 7. Обработка файлов. Предикаты для работы с файлами

deletefile(DOS_filename)-удаление файла save(DOS_filename) –сохранение файла
renamefile(old_DOS_filename, new_DOS_filename)-переименовыфвание файла
existfile(DOS_filename)-проверка на наличие файла с таким именем
flush(file_domain)-сброс данных из внутреннего буфера, отведенного для данного устройства записи
disk(Path) –выбор дисковода и пути доступа
dir(Path, File_spec, File_name) –Переменой Path должен быть присвоен корректный путь доступа, переменная File_spec задает расширение представляющей интерес группы файлов. Данный предикат выдает каталог имен файлов с заданным расширением; вы можете выбрать среди них нужный и нажать Enter. Имя файла будет присвоено переменой File_name.

При описании файловых доменов тип домена записывается по левую сторону от знака равенства, а имя доомена по правую.

Пример:

file=datafile
file=datafile1;datafile2

openwrite(datafile,filename) - открытие файла для записи или создание, где datafile- введеннный пользователем файловый домен, filename-имя файла в DOS, теперь ссылки на datafile будут означать обращение к filename.

writedevice(datafile) -назначение файла в качестве устройства записи

openread(datafile,filename) - открытие файла для чтения.

readdevice(datafile) -назначение файла устройством чтения

openmodify(datafile,filename) - открытие файла для редактирования, указатель помещается в начало файла, сместить указатель можно при помощи предиката filepos.

openappend(datafile,filename) - открытие файла для добавления данных в конец файла.

closefile(datafile) -закрытие файла

Рассмотрим примеры работы с файлами.

Пример 1. Вывести информацию на экран дисплея и в файл на диске.

Программа 27. Запись данных в файл

```
Domains
    str = string
    file = datafile
Predicates
    data(str)
    write_lines
```

```
Goal
    openwrite(datafle,"file1.dat"),
    write_lines,
    closefile(datafile).
Clauses
    data("Старому году оставьте печали,!").
    data("Забудьте обиду, беду.").
    data("Только успехов, здоровья и счастья,").
    data("Мы Вам желаем в Новом году!").
    write_lines:-
            data(Line),
            write("  ",Line),nl,
            writedevice(datafile),
            write("  ",Line),nl,
            writedevice(screen), /*для вывода данных на экран*/
            fail, write_lines.
```

Пример 2. Вывести данные файла на экран.

Программа 28. Чтение данныхиз файла

```
Domains
    str = string
    file = datafile
Predicates
    read_write_lines
Goal
    openread(datafile,"file1.dat"),
    read_write_lines,
    closefile(datafile).
Clauses
    read_write_lines :-
            readdevice(datafile),
            not(eof(datafile)),
            readln(Line),
            writedevice(screen),
            write("  ",Line),nl,
            read_write_lines.
```

Пример 3. Записать в файл данные, вводимые с клавиатуры

Программа 29. Запись в файл данных, вводимых с клавиатуры

```
Domains
    file = datafile
    dstring, cstring = string
Predicates
    readin(dstring,cstring)
    create_a_file
Goal
    create_a_file.
Clauses
    create_a_file :-
            nl,nl,
            write("Пожалуйста, введите полное имя файла."),
            nl,nl,
            readln(Filename), openwrite(datafile,Filename),
            writedevice(datafile), readln(Dstr),
            concat(Dstr,"\13\10",Cstr),
            readin(Dstr,Cstr),
            closefile(datafile).
    readin("done",_) :- !.
/*ввод данных завершится после вода слова "done"*/
    readin(_,Cstr) :-
            write(Cstr),
            readln(Dstrl),
            concat(Dstrl, "\13\10",Cstr1),
```

```
        writedevice(datafile),
        readin(Dstrl,Cstr1).
```

Предикаты для работы с файлами прямого доступа
Openmodify(fn,filename)-связывает логическое имя файла fn с именем файла
Filepos(fn, pos, mode)-устанавливает указатель файла в заданную позицию

Таблица 12.

Действие системы при операциях с файлами прямого доступа

Режим mode	Действия системы
0	Смещение берется относительно начала файла
1	Смещение берется относительно текущей позиции
2	Смещение берется относительно конца файла

Пример 4. Данные, вводимые с клавиатуры, записать в файл прямого доступа.

Программа 30. Запись данных в файл прямого доступа

```
Domains
    file = datafile
Predicates
    create_a_random_access_file
    write_read_more(real, string)
    pad_str (strIng,string,integer)
Goal
    create_a_random_access_file.
Clauses
    create_a_random_access_file :-
        write("Please enter filename:"),nl,
        readln(Filename),
        openwrite(datafile,Filename),
        closefile(datafile),
        openmodify(datafile,Filename),
        write("Введите строку"),nl,
        readln(Dstr),
        write_read_more(0,Dstr),
        closefile(datafile).
    write_read_more(_,"done") :-
        nl, write(" Press the space bar."),
        readchar(_),exit.
    write_read_more(Index,Dstr) :-
        writedevice(datafile),
        filepos(datafile,Index,0),
        pad_str(Dstr,Padstr,38),
        concat(Padstr, "\10\13",
        Cstr),
        write(Cstr),
        writedevice(screen),
        write("Введите строку"),nl,
        readln(Dstr1), Index1 =
        Index + 40,
        write_read_more (Index1, Dstr1).
    pad_str (Instr,Instr,Length) :-
        str_len(Instr,Testlength),
        Testlength >= Length,!. pad_str
        (Instr,Padstr,Length) :-
        concat(Instr,"-",Newstr),
        pad_str(Newstr,Padstr,Length).
```

Пример 5. Вывести на экран заданную строку файла прямого доступа и выдача их на экран

Программа 31. Вывод данных из файла прямого доступа

```
Domains
    file = datafile
Predicates
    read_a_random_access_file
Goal
    read_a_random_access_file.
Clauses
```

```
read_a_random_access_file:-
        write("Please enter filename:"), nl, readln(Filename),
        openread(datafile,Filename), write("Type 1n record
        number: "),nl,
        readreal(Record),
        Index = (Record - 1) * 40,
        readdevice(datafile),
        filepos(datafile,Index,0),
        readln(Cstr), write(Cstr),
        nl,nl,
        write("Press the space bar."),nl,
        readdevice(keyboard),
        readchar(_), closefile(datafile),
        exit.
```

Задание для самостоятельной работы

1) В файле задана последовательность целых чисел, найти сумму чисел, предшествующих первому отрицательному.
2) Переписать данные файла в обратном порядке.
3) Вывести на экран данные файла в неубывающем порядке.
4) Записать результат сложения чисел, содержащихся в файлах в третий.
5) Переписать числа из файла в другой, дописав за каждым его квадрат.

Лабораторная работа 8. Создание динамической базы данных. Предикаты для работы с базой данных

Для описания предикатов динамической базой данных предназначен раздел **database**. База данных (БД) называется динамической потому, что во время работы можно добавлять, удалять содержащиеся в них утверждения. Другая важная особенность динамических баз данных состоит в том, что она может быть записана на диск и считана с диска в оперативную память.

Иногда предпочтительно иметь часть информации БД в виде утверждений статической БД. Эти данные заносятся в динамическую сразу после активизации программы. В общем случае, предикаты статической БД имеют другое имя, но ту же самую форму представления данных, что и предикаты динамической.

Предикаты для работы с динамической БД:

Asserta заносит новый факт в БД, новый факт помещается перед всеми уже внесенными утверждениями.

Assertz помещает новый факт в БД за всеми имеющимися утверждениями.

Retract удаляет утверждение из БД.

Save сохраняет находящуюся в оперативной памяти БД в текстовом файле. Синтаксис этого предиката таков Save(DOS_file_name)

consult(DOS_file_name) считывает в память файл БД

readterm(Domain,Term) используется для чтения из файла объектов, относящихся к определенному в программе домену.

Предикат findall позволяет собрать все имеющиеся в базе данные в список, который может быть полезен при дальнейшей работе.

В качестве примера рассмотрим БД по игрокам футбольных команд, БД допускает следующие операции: добавление, удаление и просмотр данных. Эта программа создает БД и содержит её в оперативной памяти. Для работы с ней использовался предикат player с аргументами *p_name*-имя игрока, *k_name*-название клуба, *p_number*-номер игрока, *pos*-позиция игрока, *height* -рост, *weight*-вес, *nfl_exp*-стаж выступлений, *college*-учебное заведение)

Программа 32. Динамическая база данных «Футбольная команда»

```
Domains
    p_name,k_name, pos, college = string
    p_number, height, weight, nfl_exp = integer
Database
    dplayer(p_name, k_name,p_number,pos, height,weight,nfl_exp, college)
Predicates
    repeat
    do_mbase
    assert_database
    menu
    process(integer)
    clear_database
    player(p_name, k_name,p_number,pos, height,weight,nfl_exp, college)
    error
```

```
Goal
    do_mbase.
Clauses
    repeat.
    repeat:-repeat.
                    /*База данных футбол*/
    player("Иванов Иван","Спартак",13,"з", 205,90,3, "ГГПИ").
    player("Петров Петр","Динамо",96,"пз", 185,78,4, "ГТК").
    player("Сидоров Денис","Локомотив",69,"в", 190,88,2, "ГТУ").
    player("Васечкин Илья","Торпедо",5,"в", 195,80,5, "ГГПИ").
    player("Алексеев Дима","ЦСКА",1,"н", 165,67,2, "ГТК").
    player("Карпов Павел","Зенит",12,"н",170,74,1, "ГКК").
                    /*конец начальных данных*/
    assert_database:-
    player(P_name,K_name,P_number,Pos,Height,Weight,Nfl_exp,College),
    assertz(dplayer(P_name,K_name,P_number,Pos,Height,Weight,Nfl_exp,
                    College)),
    fail.
    assert_database:-!.

    clear_database:-
            retract(dplayer(_,_,_,_,_,_,_,_)),
            fail.
    clear_database:-!.
    do_mbase :-
            assert_database,
            makewindow(1,7,7," FOOTBALL DATABASE ",0,0,25,80),
            menu, clear_database.
    menu :-
            repeat, clearwindow,
            nl,
            write(" ************************************* "),nl,
            write(" * 1. Добавление нового игрока в БД  * "),nl,
            write(" * 2. Удаление игрока из БД          * "),nl,
            write(" * 3. Просмотр данных                * "),nl, write(" *
            4. Выход из программы             * "),nl,
            write(" ************************************* "),nl,
            write(" Пожалуйста, сделайте свой выбор 1, 2, 3 or 4 : "),
            readint(Vibor),nl,process(Vibor),Vibor = 4,!.
            /* Добавление информации об игроке в БД */
    process(1) :-
            makewindow(2,7,7,"Добавление данных",2,20,18,58),shiftwindow(2),
            write("Введите, пожалуйста:"),nl, write("Имя игрока: "),
            readln(P_name), write("Название клуба: "), readln(K_name),
            write("Номер игрока: "), readint(P_num),
            write("Позицию: "), readln(Pos), write("Рост: "), readint(Ht),
            write("Вес: "), readint(Wt),
            write("Стаж выступлений: "), readint(Exp),
            write("Название учебного заведения: "), readln(College),
            assertz(dplayer(P_name, K_name, P_num,Pos,Ht,Wt,Exp,
            College)),
            write(P_name," добавлен в БД"), nl,!, write("Press space
            bar. "), readchar(_), removewindow, shiftwindow(1).
                    /* Удаление */
    process(2) :-
            makewindow(3,7,7,"Удаление данных",10,30,7,40),shiftwindow(3),
            write("Введите имя удаляемого игрока: "), readln(P_name),
            retract(dplayer(P_name,_,_,_,_,_,_,_)), write(P_name," удален из БД
            "), nl, !,
            write("Press space bar."), readchar(_), removewindow,
```

```
        shiftwindow(1).
                /* Просмотр данных об игроке*/
process(3) :-
        makewindow(4,7,7," Просмотр ", 7,30,16,47),  shiftwindow(4),
        write("Введите имя для просмотра: "), readln(P_name),
        dplayer(P_name,T_name,P_number,Pos,Ht,Wt,Exp,College),nl,
        write(" Имя игрока        : ",P_name),nl, write(" Название
        клуба     : ",T_name),nl, write(" Номер игрока       :
        ",P_number),nl,
        write(" Позиция           : ",Pos),nl, write(" Рост              : ",Ht),
        nl, write(" Вес               : ",Wt),nl,
        write(" Стаж выступлений   : ",Exp), nl,
        write(" Учебное заведение  : ",College),nl, nl,!,
        write("Press space bar"), readchar(_),
        removewindow, shiftwindow(1).
process(3) :-
        makewindow(5,7,7," Неудача ",14,7,5,60), shiftwindow(5),
        write("К сожалению, данных нет."),nl,
        write("Извините, пока!"),nl,!, write("Press space
        bar."),readchar(_),
        removewindow,shiftwindow(1).
                /* Выход */
process(4) :-
        write("До новых встреч! "),readchar(_),exit.
                /*Обработка ошибки*/
process(Vibor):-
        Vibor<1, error; Vibor>5, error.
error:-
        write("Пожалуйста выберите число от 1 до 4"),
        write("(Press the spase bar to continue)"),readchar(_).
```

Задание для самостоятельной работы

Модифицируйте программу, добавив в меню пункты:

1) Вывод списка игроков.
2) Сохранение данных в файл.
3) Выборка данных по 1 из трех критериев.

Лабораторная работа 9. Применение языка для решения задач ИИ. Создание экспертных систем

Пример 1.Рассмотрим пример ЭС для идентификации породы собак. Она помогает потенциальному хозяину выбрать породу собаки в соответствие с определенными критериями.

В данной ЭС используются следующие характеристики:

1. Короткая шерсть;
2. Длинная шерсть;
3. Рост меньше 30 дюймов;
4. Рост меньше 22 дюймов;
5. Низкопосаженный хвост;
6. Длинные уши;
7. Хороший характер
8. Вес больше 100 фунтов.

Каждая характеристика для конкретной породы либо верна, либо не верна. Для каждой породы справедливы следующие характеристики:

Таблица 13.

Характеристики собак

Порода	Характеристики
Английский бульдог	1,4,5,7
Гончая	1,4,6,7
Дог	1,3,6,7,8
Амер.гончая	1,5,6,7
Кокер-спаниэль	2,4,5,6,7
Ирландский сеттер	2,3,6
Колли	2,3,5,7
Сенбернар	2,5,7,8

Программа 33. «Эксперт по породам собак»

```
*Эксперт по породам собак*/ /*Назначение: Демонстрация работы  ЭС*/
Domains
    n=integer
    list=n*
    dog=symbol
Predicates
    rule(n,dog,list)
    cond(n,string)
    do_expert
    show_menu
    do_consulting
    process(n)
    test(n,list)
    topic repeat
Goal
    do_expert.
Clauses
    rule(1,"английский бульдог",[1,4,5,7]).
    rule(2,"гончая",[1,4,6,7]).
    rule(3,"дог",[1,3,6,7,8]).
    rule(4,"американская гончая",[1,5,6,7]).
    rule(5,"коккер-спаниель",[2,4,5,6,7]).
    rule(6,"ирландский сеттер",[2,3,6]).
    rule(7,"колли",[2,3,5,7]).
    rule(8,"сенбернар",[2,5,7,8]).

            /*Характеристики*/
    cond(1,"короткошерстная").
    cond(2,"длинношерстная").
    cond(3,"рост ниже 30 дюймов").
    cond(4,"рост ниже 22 дюймов").
    cond(5,"низкопосаженный хвост").
```

```
cond(6,"большие уши").
cond(7,"хороший характер").
cond(8,"вес более 100 фунтов").

do_expert:-
        makewindow(1,7,5 ,"ЭКСПЕРТНАЯ СИСТЕМА",0,0,25,80),
        show_menu.

repeat.
repeat:-repeat.

                /*Вывод меню*/
show_menu:-
        repeat,
        write("*****************************"),nl,
        write("*****Добро пожаловать!*****"),nl,
        write("*                          *"),nl,
        write("*****1-консультация********"),nl,
        write("*****2-список**************"),nl,
        write("*****3-выход***************"),nl,
        write("*                          *"),nl,
        write("****Сделайте свой выбор****"),nl,
        readint(X), process(X),fail.

                /*Обработка 1 пункта меню “Консультация”*/
process(1):-
        do_consulting,
        readchar(_),
        shiftwindow(1),
        clearwindow.
                /*Обработка 2 пункта меню “Вывод списка”*/
process(2):-
        makewindow(2,7,7,"",5,20,12,25),
        topic, readchar(_), shiftwindow(1),
        clearwindow.
                /* Обработка 3 пункта меню “Выход”*/
process(3):-
        removewindow,
        exit.
                /*Вывод пород собак*/
topic:-
        rule(X,Y,_),
        write(X,".  ",Y),
        nl,fail.
topic.
                /*Консультация*/
do_consulting:-
        test(1,List),
        rule(_,X,List),
        write("Ваш выбор:"  ,X),!.
do_consulting:-
        write("Мне жаль, что не смог Вам помочь.").
                /*Тестирование*/
test(9,[]):-!. test(1,[N|List]):-
        cond(N,Text),
        makewindow(2,7,7,"",5,20,10,35),
        write("Вопрос:-",Text,"?"),nl,
        write("1-да"),nl, write("0-нет"),nl,
```

```
        readint(R),R=1,!,test(3,List).
test(1,List):- test(2,List),!.
test(N,[N|List]):-
        cond(N,Text),
        makewindow(2,7,7,"",5,20,10,35),
        write("Вопрос:-",Text,"?"),nl,
        write("1-да"), nl, write("0-нет"),
        nl, readint(R),M=N+1,
        R=1,!,test(M,List).
        test(N,List):-M=N+1,test(M,List).
```

Задания для самостоятельной работы

Разработать экспертную систему, тему выбрать самостоятельно. Отчет должен содержать следующие пункты:

1. Тема ЭС.
2. Назначение, возможности программы.
3. Разработать структурно-функциональную схему.
4. Определить базу знаний, разработать механизм вывода, интерфейс программы.
5. По каким параметрам программу можно отнести к классу ЭС.

Лабораторная работа 10. Применение нейросетевого прогнозирования при подборе кадров

Основные возможности программы NeuroPro

- Создание нейропроекта;
- Подключение к нейропроекту файла (базы) данных;
- Редактирование файла данных;
- Добавление в нейропроект нейронной сети слоистой архитектуры с числом слоев нейронов от 1 до 10, числом нейронов в слое – до 100 (число нейронов для каждого слоя сети может задаваться отдельно);
- Выбор алгоритма обучения, назначение требуемой точности прогноза, настройка параметров нейронной сети;
- Обучение нейронной сети решению задачи предсказания или классификации;
- Тестирование нейронной сети на файле данных;
- Вычисление показателей значимости каждого из входных сигналов для решения задачи, сохранение вычисленных показателей значимости в файле на диске;
- Упрощение нейронной сети;
- Генерация и визуализация вербального описания нейронной сети. Редактирование и сохранение вербального описания нейронной сети в файле на диске;
- Сохранение нейропроекта.

Форматы файлов

В качестве файлов данных (содержащих обучающую выборку для нейронных сетей) используются файлы форматов DBF(форматы пакетов Dbase, FoxBase, FoxPro, Clipper) и DB (Paradox). Возможно чтение и редактирование этих файлов и сохранение измененных файлов на диске. Программа не накладывает ограничений на число записей (строк) в файле данных.

Интерфейс программы

Для того, чтобы сделать активными пункты меню программы, необходимо открыть файл, содержащий входную информацию для создания нейросети. Таким файлом является файл формата dbf. Его можно создать самостоятельно в программе Excel 2003 или открыть уже созданный в формате dbf файл.

Щекните по кнопке Создать, в новом окне нажмите Открыть файл данных. Откройте файл.

Открытый файл отображается в собственном окне, где предоставляется возможность его редактирования. При подключенном файле данных можно проводить операции создания новых сетей, их обучения, тестирования и упрощения.

Для создания новой нейронной сети необходимо нажать кнопку “Новая сеть” в окне нейропроекта и заполнить окно для создания нейронной сети: подтвердите статусы входных полей, выходного поля, установите Тип поля - количественный, на вкладке Структура сети укажите число слоев нейронов и число нейронов в слое. Нажмите кнопку Создать.

Далее рассмотрим ставшие активными пункты меню программы.

Меню программы содержит следующие пункты, относящиеся к нейронным сетям и работе с ними:

Файл - базовые операции с файлами, включая стандартные опции: создать, открыть, сохранить, сохранить как, выход.

Нейросеть - операции с нейронными сетями. Операция выполняется над активной в данный момент в нейропроекте нейронной сетью и включает в себя следующие опции:

• Обучение - обучение нейронной сети;

• Тестирование - тестирование нейронной сети;

• Сокращение числа входных сигналов - удаление наименее значимых входных сигналов;

• Сокращение числа синапсов - удаление наименее значимых синапсов сети;

• Сокращение числа неоднородных входов - удаление наименее значимых неоднородных входов нейронов сети;

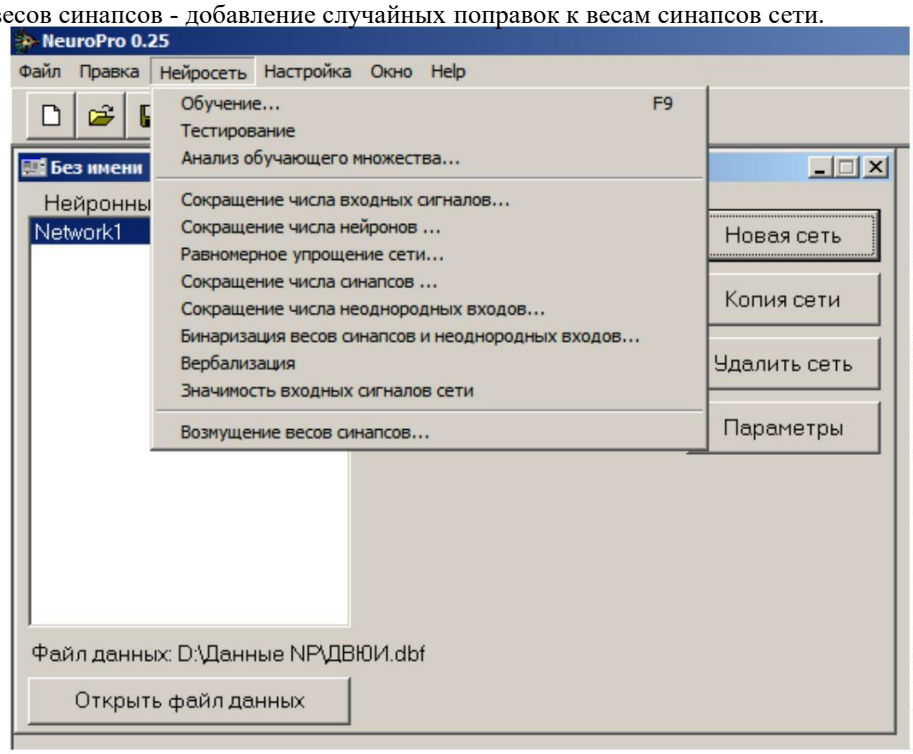
• Равномерное упрощение сети - сокращение числа приходящих на нейроны сети сигналов до заданного пользователем;

• Бинаризация синапсов сети - приведение значений весов синапсов и неоднородных входов нейронов к значениям -1 и 1;

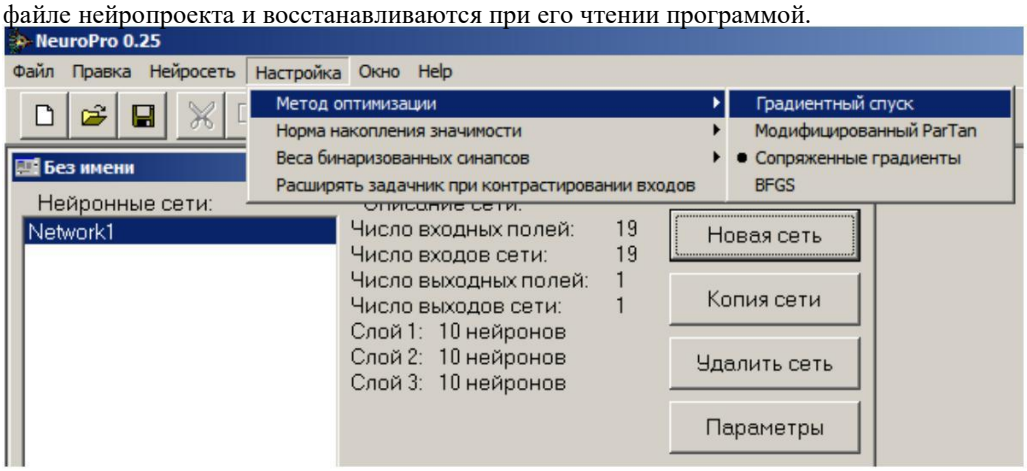
• Вербализация - генерация вербального описания нейронной сети;

• Значимость входов - подсчет и отображение значимости входных сигналов нейронной сети;

• Возмущение весов синапсов - добавление случайных поправок к весам синапсов сети.



Настройка - операции по настройке нейронной сети. Настройки действуют в пределах нейропроекта, сохраняются в файле нейропроекта и восстанавливаются при его чтении программой.



Опции данного пункта меню:

• Метод оптимизации - выбор метода оптимизации для обучения сети. Из трех реализованных в настоящее время в программе методов (градиентный спуск, модифицированный партан-метод (РагТап) и метод сопряженных градиентов) при создании нейропроекта автоматически предлагается РагТап.

• Норма накопления значимости - выбор нормы накопления градиента при подсчете показателей значимости, иначе говоря, показатель совокупной ошибки сети. При создании нейропроекта автоматически выбирается норма в виде суммы модулей.

Сокращение числа неоднородных входов - удаление наименее значимых неоднородных входов нейронов сети;

Равномерное упрощение сети - сокращение числа приходящих на нейроны сети сигналов до заданного пользователем;

Бинаризация синапсов сети - приведение значений весов синапсов и неоднородных входов нейронов к значениям -1 и 1;

Вербализация - генерация вербального описания нейронной сети;

Значимость входов - подсчет и отображение значимости входных сигналов нейронной сети;

Возмущение весов синапсов - добавление случайных поправок к весам синапсов сети.

Настройка - операции по настройке нейронной сети. Настройки действуют в пределах нейропроекта, сохраняются в файле нейропроекта и восстанавливаются при его чтении программой.

Опции данного пункта меню:

Метод оптимизации - выбор метода оптимизации для обучения сети. Из трех реализованных в настоящее время в программе методов (градиентный спуск, модифицированный партан-метод (РагТап) и метод сопряженных градиентов) при создании нейропроекта автоматически предлагается РагТап.

Норма накопления значимости - выбор нормы накопления градиента при подсчете показателей значимости, иначе говоря, показатель совокупной ошибки сети. При создании нейропроекта автоматически выбирается норма в виде суммы модулей.

Постановка задачи

При приеме на службу в органы внутренних дел (ОВД), поступлении в высшее юридическое учебное заведение МВД РФ каждый кандидат проходит обследование с помощью различных методик, направленных на оперативное и удобное обследование психологической сферы индивида (мотивы, познавательные процессы, самооценка, личностные особенности, социально-психологические качества и др.).

Однако данные, полученные в результате обследования, в дальнейшем используются неэффективно. В этой связи возникает необходимость разработки методики, применение которой давало бы возможность сопоставлять результаты тестирования слушателей высших учебных заведений МВД РФ с состоянием их успеваемости, дисциплины, а также эффективностью дальнейшей работы в ОВД и осуществлять поддержку обоснованных решений при приеме лиц на службу в ОВД или на учебу в высшие учебные заведения МВД РФ. Именно такая методика позволит более качественно осуществлять подбор сотрудников в ОВД и слушателей для высших юридических и пожарно-технических учебных заведений МВД РФ.

Для построения нейросетевой модели и ее практического применения использовались результаты тестирования лиц, поступающих на учебу в Дальневосточный юридический институт МВД РФ. Обучающая выборка состоит из результатов тестирования 270 абитуриентов ДВЮИ МВД РФ личностным опросником Кеттела и оценок состояния их дисциплины, успеваемости, а также оценок эффективности их последующей работы в ОВД. Расшифровка оценочных шкал Кеттела приведена в табл. 1.

Таблица 1 – Интерпретация оценочных факторов Кеттела

Факторы	Интерпретация факторов
A	«замкнутость — общительность»
B	«интеллект»
C	«эмоциональная неустойчивость — эмоциональная устойчивость»
E	«подчиненность — доминантность»
F	«сдержанность — экспрессивность»
G	«подверженность чувствам — высокая нормативность поведения»
H	«робость — смелость»
I	«жесткость — чувствительность»
L	«доверчивость — подозрительность»
M	«практичность — развитое воображение»
N	«прямолинейность — дипломатичность»
O	«уверенность в себе — тревожность»
Q1	«консерватизм — радикализм»
Q2	«конформизм — нонконформизм»
Q3	«низкий самоконтроль — высокий самоконтроль»
Q4	«расслабленность — напряженность»

Обученные соответствующим образом искусственные нейронные сети (НС) смогут прогнозировать для поступающего в ДВЮИ МВД РФ оценку состояния дисциплины и успеваемости с горизонтом прогнозирования 4 года и эффективность его последующей работы с горизонтом прогнозирования 7 лет. Обучение НС целесообразно производить на нейроимитаторе Neuropro 0.25, который успешно применяется для решения различных нерегулярных задач.

Последовательность выполнения работы

1. Запуск программного нейроимитатора «NEUROPRO 0.25».

«ПУСК» – «ПРОГРАММЫ» – «NEUROPRO» – «NEUROPRO 0.25» (Или ярлык на рабочем столе)

2. Начало работы в нейроимитаторе.

«ФАЙЛ» – «СОЗДАТЬ» – «ОТКРЫТЬ ФАЙЛ ДАННЫХ»

Открыть файл данных ДВЮИ.dbf, содержащий результаты тестирования и экспертные оценки прогнозируемых параметров, необходимые для обучения сети.

Щелкнуть левой кнопкой мыши (л.к.м.) на предыдущем окне (без имени)

Нажать кнопку «НОВАЯ СЕТЬ». Установить входы и выходы сети: поле

«NOMER» не используется; поле «DISCIP» не используется; поле

«USPEVA» не используется;

поле «RABOTA» выходное, тип поля количественный, точность 0,1.

3. Задать структуру сети:

число слоев – 1; число нейронов – 16;

характеристика – 0,1. Нажать кнопку

«СОЗДАТЬ».

В верхнем меню нажать «НЕЙРОСЕТЬ» – «ОБУЧЕНИЕ» – после окончания обучения – «ГОТОВО».

Определить минимальное количество нейронов для решения данной задачи:

4. Нажать «КОПИЯ СЕТИ».
5. В «СТРУКТУРЕ СЕТИ» изменить количество нейронов.

Далее с п. 3 по п.4 включительно. П Р О Г Н О З И Р О В А

Н И Е.

6. Щелкнуть л.к.м. на таблице с данными.
7. Выбрать следующие пункты меню: «ТАБЛИЦА» – «ДОБАВИТЬ ЗАПИСЬ».

Создать новый пункт (№ 271) и заполнить его произвольными числами, в диапазоне от 1 до 10 (перемещение между столбцами кнопкой «TAB» или «стрелка вправо»), кроме колонки прогнозируемого параметра (РАБОТА).

Щелкнуть л.к.м. на предыдущем окне (Без имени).

Из верхнего меню выбрать «НЕЙРОСЕТЬ» – «ТЕСТИРОВАНИЕ».

Переместиться в конец таблицы и посмотреть значение прогнозируемого параметра.

Закрыть окно тестирования.

8. Оценка сетью значимости входных сигналов (в нашем случае оценочных шкал Кеттела, за каждой из которых скрывается черта характера).

«НЕЙРОСЕТЬ» – «ЗНАЧИМОСТЬ ВХОДНЫХ СИГНАЛОВ».

Выпишите самые важные, с точки зрения сети, оценочные шкалы и закройте окно.

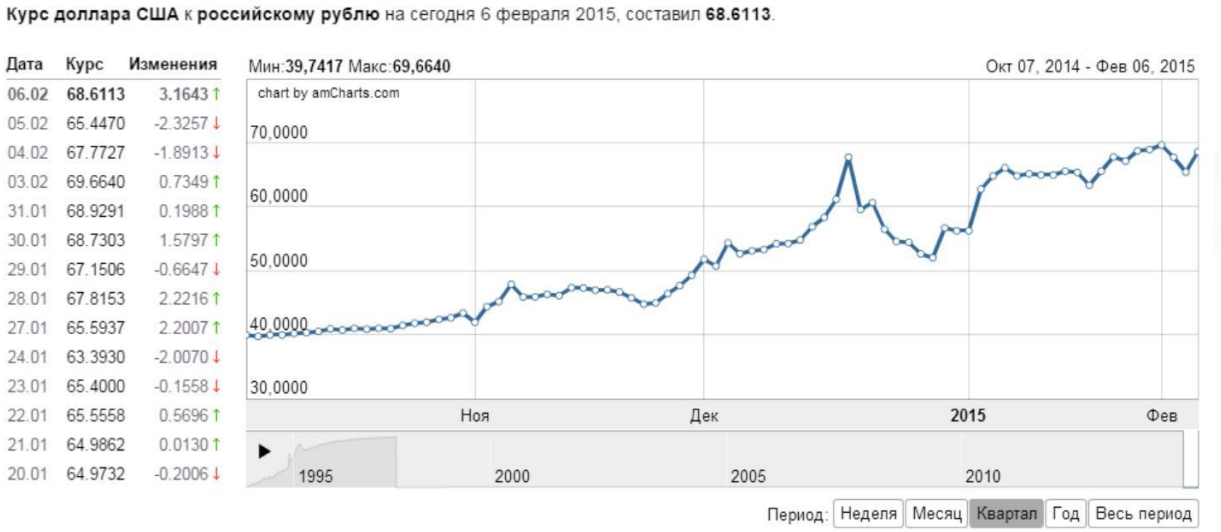
9. Повторить с п.2, используя в качестве прогнозируемого параметра «ДИСЦИПЛИНУ» и «УСПЕВАЕМОСТЬ» (по очереди).

Поэкспериментировать с упрощением сети «НЕЙРОСЕТЬ» – «УПРОЩЕНИЕ».

10. Пользуясь полученными диаграммами значимости входных сигналов сформулируйте в качестве вывода, какие качества личности, по Кеттелу, влияют на дисциплину, успеваемость и эффективность работы выпускников высших учебных заведений МВД РФ.

11. Оформить отчёт.

Лабораторная работа 11. Нейросетевое прогнозирование колебаний котировок валют



Постановка задачи

Прогнозирование каких-либо нестабильных экономических показателей сопряжено с множеством трудностей, обусловленных различными факторами. Однако краткосрочный прогноз возможен и эффективно применяется в аналитической работе различных организаций.

1. Составьте в MS Excel таблицу, содержащую котировки валют с 01.01.2015 г. по 06.02.2015 г. (названия столбцов пишите английскими буквами).

Таблица 2 – Официальные Курсы ЦБ РФ (USD,EUR) на январь, февраль 2015 г.

Data	EUR/RUR	USD/RUR
01.01.2015	68,3681	56,2376
02.01.2015	68,3681	56,2376
03.01.2015	68,3681	56,2376
04.01.2015	68,3681	56,2376
05.01.2015	68,3681	56,2376
06.01.2015	68,3681	56,2376
07.01.2015	68,3681	56,2376
08.01.2015	68,3681	56,2376
09.01.2015	68,3681	56,2376
10.01.2015	68,3681	56,2376
11.01.2015	68,3681	56,2376
12.01.2015	68,3681	56,2376
13.01.2015	74,3551	62,7363
14.01.2015	76,7735	64,8425
15.01.2015	77,9629	66,0983
16.01.2015	76,3352	64,8337
17.01.2015	75,8623	65,1738
18.01.2015	75,8623	65,1738
19.01.2015	75,8623	65,1738
20.01.2015	75,174	64,9732
21.01.2015	75,2735	64,9862
22.01.2015	75,8218	65,5558
23.01.2015	75,7724	65,4
24.01.2015	71,9067	63,393
25.01.2015	71,9067	63,393
26.01.2015	71,9067	63,393
27.01.2015	73,5633	65,5937
28.01.2015	76,2922	67,8153

29.01.2015	76,2629	67,1506
30.01.2015	77,569	68,7303
31.01.2015	78,1105	68,9291
01.02.2015	78,1105	68,9291
02.02.2015	78,1105	68,9291
03.02.2015	78,79	69,664
04.02.2015	76,8271	67,7727
05.02.2015	75,0415	65,447
06.02.2015	77,9356	68,6113

Сохраните таблицу в формате Excel (для дальнейшего редактирования) и в формате DBF4 (для работы в нейроимитаторе). Сохранение в формате DBF4 возможно только в Excel 2003.

2. Создайте, обучите, протестируйте, упростите и вербализуйте нейронную сеть.
3. По обученной нейронной сети выполните прогноз курса доллара США с горизонтом прогнозирования 3 дня (с 07.02.2015 по 09.02.2015).
4. Постройте в табличном редакторе EXCEL линию тренда с уравнением в виде полинома 2 степени и прямой) и выполните прогноз вперед. Проведите корреляционный анализ показателей.
5. Сравните результаты прогнозируемых параметров, полученных с использованием нейронных сетей и Excel.
6. Дополните самостоятельно и отредактируйте в MS Excel таблицу, котировками других валют (не менее 5). Курсы валют приведены в приложении. Например, австралийский доллар AUD/RUR, английский фунт стерлингов GBP/RUR, датские кроны 10 DKK/RUR, шведские кроны 10 SEK/RUR, украинские гривны 10 UAN/RUR, турецкие лиры TRY/RUR, швейцарские франки CHF/RUR, японские йены 100 JPY/RUR (23,2498) и др. Последняя колонка – доллар. Сохраните таблицу в формате DBF 4, который поддерживает программа нейроимитатор. Выполните прогноз котировок валют с горизонтом прогнозирования 3 дня.
7. Сделайте выводы и оформите отчет по выполненной лабораторной работе.

Лабораторная работа 12. Нейросетевое распознавание сложных образов (текста)

Часть 1

1. Запустите Microsoft Excel.
2. Оставьте первую строку пустой.
3. Используя цифровую кодировку букв (А-1, Б-2, В-3, Г-4…), запишите друг под другом слева направо три слова САЛЬДО, ДЕБЕТ, КРЕДИТ (код каждой буквы в отдельной ячейке), пронумеровав строки в последнем столбце.
4. Выделите полученную таблицу (в т.ч. и оставленную пустой первую строку) и сохраните в формате DBF 4. Откройте файл через программу NEURO PRO 0.25 («ФАЙЛ» – «СОЗДАТЬ» – «ОТКРЫТЬ ФАЙЛ ДАННЫХ»).
5. Нажмите кнопку «НОВАЯ СЕТЬ» и задайте выходным сигналом столбец с нумерацией строк.
6. Установите точность 0,01.
7. Нажмите кнопку «СОЗДАТЬ».
8. В верхнем меню нажмите «НЕЙРОСЕТЬ» – «ОБУЧЕНИЕ» – после окончания обучения – «ГОТОВО».
9. Щелкните л.к.м. на таблице с данными.
10. Выберите следующие пункты меню: «ТАБЛИЦА» – «ДОБАВИТЬ ЗАПИСЬ».
11. Создайте новый пункт, написав цифрами любое слово с ошибкой.
12. Спрогнозируйте, используя «Нейросеть» – «Тестирование».

Прогнозируемый параметр должен быть близким к порядковому номеру слова.

Часть 2

1. Запустите Microsoft Excel.
2. Оставьте первую строку пустую.
3. Используя цифровую кодировку букв (А-1, Б-2, В-3, Г-4…), напишите в строку три определения (например, пороговая, линейная, сигмоидальная). В следующей строке напишите другую вариацию понятий, перебрав, таким образом, все возможные варианты (должно получиться 6 строк с разными сочетаниями цифр). Пронумеруйте полученные строки цифрами от 1 до 6.

Таблица 3. – Кодировка алфавита по порядковому номеру букв

А	Б	В	Г	Д	Е	Ё	Ж	З	И
1	2	3	4	5	6	7	8	9	10
Й	К	Л	М	Н	О	П	Р	С	Т
11	12	13	14	15	16	17	18	19	20
У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь
21	22	23	24	25	26	27	28	29	30
Э	Ю	Я 31	32	33					

4. Выделите полученную таблицу (в т.ч. и оставленную пустой первую строку) и сохраните в формате DBF 4.Откройте файл через программу NEURO PRO 0.25 («ФАЙЛ» – «СОЗДАТЬ» – «ОТКРЫТЬ ФАЙЛ ДАННЫХ»).
5. Нажмите кнопку «НОВАЯ СЕТЬ» и задайте выходным сигналом столбец с нумерацией строк.
6. Установите точность 0,01.
7. Нажмите кнопку «СОЗДАТЬ».
8. В верхнем меню нажмите «НЕЙРОСЕТЬ» – «ОБУЧЕНИЕ» – после окончания обучения – «ГОТОВО».
9. Щелкните л.к.м. на таблице с данными.
10. Выберите следующие пункты меню: «ТАБЛИЦА» – «ДОБАВИТЬ ЗАПИСЬ».
11. Добавьте запись, написав три определения, в произвольном порядке допустив не более 5 ошибок (можно просто вставить 5 нулей). Оставьте последнюю колонку пустой.
12. Спрогнозируйте, используя «Нейросеть/Тестирование».
13. Прогнозируемый параметр должен быть близким к порядковому номеру записи, вариацию которой вы выбрали.
14. Сформулируйте выводы и оформите ход работы на ПК.

4.1.2. Эссе

Тема 1. Введение в проектирование интеллектуальных систем
Тема 2. Модели представления знаний в интеллектуальных системах
Тема 3. Язык логического программирования Пролог
Тема 4. Экспертные системы Тема 5. Нейронные сети

4.1.2.1. Порядок проведения.

Обучающиеся пишут на заданную тему сочинение, выражающее размышления и индивидуальную позицию автора по определённому вопросу, допускающему неоднозначное толкование. Оцениваются эрудиция автора по теме работы, логичность, обоснованность, оригинальность выводов.

4.1.2.2 Критерии оценивания

Оценка «отлично» ставится, если обучающийся:

Тему полностью раскрыл. Превосходное владение материалом. Высокий уровень самостоятельности, логичности, аргументированности. Превосходный стиль изложения

Оценка «хорошо» ставится, если обучающийся:

Тему в основном раскрыл. Хорошее владение материалом. Средний уровень самостоятельности, логичности, аргументированности. Хороший стиль изложения.

Оценка «удовлетворительно» ставится, если обучающийся:

Тему частично раскрыл. Удовлетворительное владение материалом. Низкий уровень самостоятельности, логичности, аргументированности. Удовлетворительный стиль изложения.

Оценка «неудовлетворительно» ставится, если обучающийся:

Тему не раскрыл. Неудовлетворительное владение материалом. Недостаточный уровень самостоятельности, логичности, аргументированности. Неудовлетворительный стиль изложения.

4.1.2.3. Содержание оценочного средства

Темы 1-5

1. Искусственный интеллект или что ждет нас в будущем
2. Искусственный интеллект - возможности и ограничения, перспективы и последствия.
3. Искусственный интеллект - фундаментальный угроза для всего человечества.
4. Искусственный интеллект - тот случай, когда нужно быть достаточно дальновидными в вопросах регулирования, иначе может оказаться слишком поздно.

5. Создание искусственного интеллекта может стать последним технологическим достижением человечества, если мы не научимся контролировать риски.
6. Возможно ли создание мыслящего компьютера

4.2. Оценочные средства промежуточной аттестации

По дисциплине предусмотрен экзамен. Экзамен проходит по билетам. В каждом билете один теоретический вопрос и одна задача. Экзамен проводится в устной, письменной или компьютерной форме. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций и решении практических заданий.

4.2.1. Устный или письменный ответ на вопрос

4.2.1.1. Порядок проведения.

Устный или письменный ответ на вопрос направлен на проверку знаний основных разделов по дисциплине «Основы искусственного интеллекта».

4.2.1.2. Критерии оценивания.

Оценка «отлично» ставится, если обучающийся:

В ответе качественно раскрыл содержание темы. Ответ хорошо структурирован. Прекрасно освоен понятийный аппарат. Продемонстрирован высокий уровень понимания материала. Превосходное умение формулировать свои мысли, обсуждать дискуссионные положения. **Оценка «хорошо» ставится, если обучающийся:**

Основные вопросы темы раскрыл. Структура ответа в целом адекватна теме. Хорошо освоен понятийный аппарат. Продемонстрирован хороший уровень понимания материала. Хорошее умение формулировать свои мысли, обсуждать дискуссионные положения.

Оценка «удовлетворительно» ставится, если обучающийся:

Тему частично раскрыл. Ответ слабо структурирован. Понятийный аппарат освоен частично. Понимание отдельных положений из материала по теме. Удовлетворительное умение формулировать свои мысли, обсуждать дискуссионные положения.

Оценка «неудовлетворительно» ставится, если обучающийся:

Тему не раскрыл. Понятийный аппарат освоен неудовлетворительно. Понимание материала фрагментарное или отсутствует. Неумение формулировать свои мысли, обсуждать дискуссионные положения.

4.2.1.3. Оценочные средства.

Вопросы для устного или письменного ответа

1. Подходы к определению искусственного интеллекта. Информационный, бионический и эволюционный подходы.
2. Интеллектуальные системы.
3. Этапы развития и основные направления искусственного интеллекта.
4. Классификация интеллектуальных систем.
5. Основные (базовые) свойства и возможности.
6. Обобщённая типология знаний.
7. Данные и знания.
8. Свойства, характеристики знаний.
9. Процедурные и декларативные знания.
10. Классификация знаний по глубине, по жесткости.
11. Формализация знаний.
12. Формальные языки.
13. Языки (модели) представления знаний.
14. Классификация моделей знаний и данных.
15. Формально-логические модели.
16. Логика высказываний.
17. Алфавит, аксиомы, теоремы, логические переменные, логический вывод.
18. Основные законы и правила вывода логики высказываний.
19. Логика предикатов.
20. Элементы языка логики предикатов.
21. Термы, кванторы всеобщности и общезначимости.
22. Нейронные сети и их применение.
23. Биологический прототип и искусственный нейрон.
24. Математические модели нейронов.
25. Однослойные искусственные нейронные сети.
26. Персептроны и зарождение искусственных нейронных сетей.
27. Алгоритм обучения персептрона.

28 Процедура обратного распространения.

4.2.2. Практическое задание

4.2.2.1. Порядок проведения.

Предлагаются практические задания по дисциплине «Основы искусственного интеллекта».

4.2.2.2. Критерии оценивания.

Оценка «отлично» ставится, если обучающимся:

Задание выполнено полностью и правильно.

Оценка «хорошо» ставится, если обучающимся:

Задание выполнено полностью, но нет достаточного обоснования. Или при верном решении допущена вычислительная ошибка или недочет, не влияющий на правильную последовательность рассуждений.

Оценка «удовлетворительно» ставится, если обучающимся:

Задание выполнено частично или с фактическими и вычислительными ошибками.

Оценка «неудовлетворительно» ставится, если обучающимся:

Задание не выполнено или выполнено с большим количеством фактических и вычислительных ошибок.

4.2.2.3. Оценочные средства.

Решить логическую задачу на языке Prolog

1. Трое ребят вышли гулять с собакой, кошкой и хомячком. Известно, что Петя не любит кошек и живет в одном подъезде с хозяйкой хомячка. Лена дружит с Таней, гуляющей с кошкой. Определить, с каким животным гулял каждый из детей.
2. Беседуют трое друзей: Белокуров, Рыжов и Чернов. Брюнет сказал Белокурову: «Любопытно, что один из нас блондин, другой – брюнет, а
третий – рыжий, но ни у кого цвет волос не соответствует фамилии». Какой цвет волос у каждого из друзей?
3. Витя, Юра, Миша и Дима сидели на скамейке. В каком порядке они сидели, если известно, что Юра сидел справа от Димы, Миша справа от Вити, а Витя справа от Юры.
4. Известно, что Волга длиннее Амударьи, а Днепр короче Амударьи. Лена длиннее Волги. Определить вторую по протяженности реку.

Приложение 2
к рабочей программе дисциплины (модуля)
Б1.В.02.01 Основы искусственного интеллекта

Перечень литературы, необходимой для освоения дисциплины (модуля)

Направление подготовки: 23.03.01 Технология транспортных процессов

Профиль подготовки: Проектирование и управление интеллектуальными транспортными системами

Квалификация выпускника: бакалавр

Форма обучения: заочное

Язык обучения: русский

Год начала обучения по образовательной программе: 2025

Основная литература:

1. Пылов, П. А. Изучение искусственного интеллекта на основе принципа интенсификации обучения : монография / П. А. Пылов, Р. В. Майтак, А. В. Дягилева. - Москва ; Вологда : Инфра-Инженерия, 2024. - 172 с. - ISBN 978-5-9729-1594-1. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2169703
2. Бутл, Р. Искусственный интеллект и экономика : Работа, богатство и благополучие в эпоху мыслящих машин : научно-популярное издание / Р. Бутл. - Москва : Альпина ПРО, 2023. - 424 с. - ISBN 978-5-206-00065-8. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2141006 .
3. Тополь, Э. Искусственный интеллект в медицине: Как умные технологии меняют подход к лечению : научно-популярное издание / Э. Тополь ; пер. с англ. А. Анваер. - Москва : Альпина Паблишер, 2022. - 398 с. - ISBN 978-5-9614-2920-6. - Текст : электронный. - URL: https://znanium.ru/catalog/product/1905828

Дополнительная итература:

4. Булгакова, И. А. Искусственный интеллект и авторское право в сфере культуры : учебное пособие / И. А. Булгакова, И. Ю. Никодимов, М. Ю. Новиков ; под общ. ред. Е. А. Пахомовой ; Российская государственная специализированная академия искусств. – Москва : Издательско-торговая корпорация «Дашков и К°», 2025. - 154 с. – ISBN 978-5-394-06206-3. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2198426
5. Искусственный интеллект в образовании: возможности, методы и рекомендации для педагогов : учебно-практическое пособие / под ред. С. О. Крамарова. — Москва : РИОР : ИНФРА-М, 2025. — 99 с. — (Наука и практика). - ISBN 978-5-369-01968-9. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2173231.

Приложение 3
к рабочей программе дисциплины (модуля)
Б1.В.02.01 Основы искусственного интеллекта

Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Направление подготовки: 23.03.01 Технология транспортных процессов

Профиль подготовки: Проектирование и управление интеллектуальными транспортными системами

Квалификация выпускника: бакалавр

Форма обучения: заочное

Язык обучения: русский

Год начала обучения по образовательной программе: 2025

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

1. Операционная система Microsoft office professional plus 2010, или Microsoft Windows 7 Профессиональная, или Windows XP (Volume License)
2. Пакет офисного программного обеспечения Microsoft Office 365, или Microsoft office professional plus 2010
3. Adobe Reader XI или Adobe Acrobat Reader DC
4. Браузер Mozilla Firefox 5. Браузер Google Chrome
6. Kaspersky Endpoint Security для Windows
7. Программная система для обнаружения текстовых заимствований в учебных и научных работах. АО «Антиплагиат»
8. Электронная библиотечная система «ZNANIUM.COM»
9. Электронная библиотечная система Издательства «Лань»
10. Электронная библиотечная система «Консультант студента»