

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



УТВЕРЖДАЮ

Директор
Елабужского института КФУ
 Е.Е. Мерзон

« 23 » 05 2024 г.
МП

Программа дисциплины (модуля)

Компьютерное моделирование и искусственный интеллект

Направление подготовки/специальность: 44.03.01 Педагогическое образование

Направленность (профиль) подготовки: Технология и робототехника

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2024

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

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
ОПК-9	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности
ОПК-9.1	Знать принципы работы современных информационных технологий и способы их использования для решения задач профессиональной деятельности
ОПК-9.2	Уметь применять принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности
ОПК-9.3	Владеть пониманием принципов работы современных информационных технологий и навыками их использования для решения задач профессиональной деятельности

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

основные принципы работы современных информационных технологий, методы искусственного интеллекта для решения стандартных задач по компьютерному моделированию.

Должен уметь:

применять основные принципы работы современных информационных технологий, методы искусственного интеллекта для решения стандартных задач по компьютерному моделированию.

Должен владеть:

пониманием основных принципов работы современных информационных технологий, методами искусственного интеллекта для решения стандартных задач по компьютерному моделированию.

2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в раздел "Б1.О.08.06 Дисциплины (модули)" основной профессиональной образовательной программы 44.03.01 "Педагогическое образование (Технология и робототехника)" и относится к обязательным дисциплинам.

Осваивается на 4 курсе в 8 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 - 18 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 36 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 18 часа(ов).

Контроль (зачёт / экзамен) - 36 часа(ов).

Форма промежуточного контроля дисциплины: экзамен в 8 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	Се м е ст р	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекци и	Практич еские занятия	Лаборат орные работы	
1.	Тема 1. Некоторые известные модели математической физики	8	4	0	4	2
2.	Тема 2. Математическая теория борьбы за существование	8	2	0	6	2
3.	Тема 3. Моделирование процессов диффузии и переноса	8	2	0	6	2
4.	Тема 4. Введение в проектирование интеллектуальных систем	8	2	0	0	2
5.	Тема 5. Модели представления знаний в интеллектуальных системах	8	2	0	2	2
6.	Тема 6. Язык логического программирования Пролог	8	2	0	6	4
7.	Тема 7. Экспертные системы	8	2	0	6	2
8.	Тема 8. Нейронные сети	8	2	0	6	4
	Итого: 108 часов (из них 36 часов контроль)		18	0	36	18

4.2 Содержание дисциплины (модуля)

Тема 1. Некоторые известные модели математической физики

Колебания камертона. Задача Стефана. Задача Стефана представляет собой особый вид краевой задачи для дифференциального уравнения в частных производных, описывающая изменение фазового состояния вещества, при котором положение границы раздела фаз изменяется со временем. Наличие границ раздела между фазами, которые не задаются явно и могут смещаться со временем, является характерной особенностью таких задач. Скорость смещения межфазных границ определяется дополнительным условием на границе раздела фаз, что приводит задачу к нелинейному виду.

Тема 2. Математическая теория борьбы за существование

Система "хищник - жертва" - сложная экосистема, для которой реализованы долговременные отношения между видами хищника и жертвы, типичный пример коэволюции. Отношения между хищниками и их жертвами развиваются циклически, являясь иллюстрацией нейтрального равновесия. Модель совместного существования двух биологических видов (популяций) типа "хищник - жертва" называется также моделью Вольтерры - Лотки.

Тема 3. Моделирование процессов диффузии и переноса

Математическое моделирование экосистем является научным направлением, которое становится действенным аппаратом познания экологических процессов, приближает к осуществлению практики управления ими. Подобные модели содержат в себе информацию как априорную, заключенную в структуре математической модели (тип дифференциального, интегрального, разностного, балансового или другого уравнения), так и информацию, содержащуюся в параметрах (коэффициентах) модели, которые определяются из опытных данных.

Тема 4. 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 5. 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 6. 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 7. 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 8. 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 перцептрона. Алгоритм обучения перцептрона. 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 (утвержденный приказом Министерства науки и высшего образования Российской Федерации от 6 апреля 2021 года № 245)

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 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)

Введение в математическое моделирование - <https://www.intuit.ru/studies/courses/2260/156/info>

Компьютерное моделирование - <https://www.intuit.ru/studies/courses/643/499/INFO>

Основы математического моделирования - <https://www.intuit.ru/studies/courses/66/66/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>

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

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 способом 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.
самостоятельная 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	При подготовке к экзамену необходимо просмотреть программу курса, с целью выявления наиболее проблемных тем, вопросов, которые могут вызвать трудности при подготовке к зачету. Про решать тестовые задания, предложенные в учебно-методическом комплексе. При этом для эффективного закрепления информации первый раз без использования учебных материалов, второй раз с их использованием.

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, помещения 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 полки 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 44.03.01 "Педагогическое образование" и профилю подготовки "Технология и робототехника".

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.О.08.06 Компьютерное моделирование и искусственный интеллект

Направление подготовки: 44.03.01 - Педагогическое образование

Профиль подготовки: Технология и робототехника

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2024

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. Тестирование
 - 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
Экзамен
 - 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 компетенции для данной дисциплины	Оценочные средства текущего контроля и промежуточной аттестации
ОПК-9 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	<p>Знать основные принципы работы современных информационных технологий, методы искусственного интеллекта для решения стандартных задач по компьютерному моделированию.</p> <p>Уметь применять основные принципы работы современных информационных технологий, методы искусственного интеллекта для решения стандартных задач по компьютерному моделированию.</p> <p>Владеть пониманием основных принципов работы современных информационных технологий, методами искусственного интеллекта для решения стандартных задач по компьютерному моделированию.</p>	<p>Текущий контроль: <i>Лабораторные работы</i> по темам Тема 1. Некоторые известные модели математической физики Тема 3. Математическая теория борьбы за существование Тема 4. Введение в проектирование интеллектуальных систем Тема 5. Модели представления знаний в интеллектуальных системах Тема 6. Язык логического программирования Пролог Тема 7. Экспертные системы Тема 8. Нейронные сети <i>Тестирование</i> по темам Тема 1. Некоторые известные модели математической физики Тема 3. Математическая теория борьбы за существование</p> <p>Промежуточная аттестация: <i>Экзамен</i></p>

2. Критерии оценивания сформированности компетенций

Компетенция	Зачтено			Не зачтено
	Высокий уровень (отлично) (86-100 баллов)	Средний уровень (хорошо) (71-85 баллов)	Низкий уровень (удовлетворительно) (56-70 баллов)	Ниже порогового уровня (неудовлетворительно) (0-55 баллов)
ОПК-9	Знает основные принципы работы современных информационных технологий, методы искусственного интеллекта для решения стандартных задач по компьютерному моделированию	Знает основные принципы работы современных информационных технологий, основные методы искусственного интеллекта для решения стандартных задач по компьютерному моделированию, имеющихся, допуская незначительные ошибки при ее решении	Знает основные принципы работы современных информационных технологий, отдельные методы искусственного интеллекта, допуская серьезные ошибки при ее решении	Не знает основные принципы работы современных информационных технологий, методы искусственного интеллекта для решения стандартных задач по компьютерному моделированию
	Умеет применять основные принципы работы современных информационных технологий, методы искусственного интеллекта для решения стандартных задач по компьютерному моделированию	Умеет применять основные принципы работы современных информационных технологий, основные методы искусственного интеллекта для решения стандартных задач по компьютерному моделированию, допуская незначительные ошибки при ее решении	Умеет применять основные принципы работы современных информационных технологий, отдельные методы искусственного интеллекта, допуская серьезные ошибки при ее решении	Не умеет применять основные принципы работы современных информационных технологий, методы искусственного интеллекта для решения стандартных задач по компьютерному моделированию
	Владеет пониманием основных принципов работы современных информационных технологий, методами	Владеет пониманием основных принципов работы современных информационных технологий, основными	Владеет пониманием основных принципов работы современных информационных технологий, отдельными	Не владеет пониманием основных принципов работы современных информационных технологий

	искусственного интеллекта для решения стандартных задач по компьютерному моделированию	методами искусственного интеллекта для решения стандартных задач по компьютерному моделированию, допуская незначительные ошибки при ее решении	методами искусственного интеллекта, допуская серьезные ошибки при ее решении	технологий, методами искусственного интеллекта для решения стандартных задач по компьютерному моделированию
--	--	--	--	---

3. Распределение оценок за формы текущего контроля и промежуточную аттестацию

8 семестр:

Текущий контроль:

Лабораторные работы по темам

Тема 1. Некоторые известные модели математической физики

Тема 3. Математическая теория борьбы за существование

Тема 4. Введение в проектирование интеллектуальных систем

Тема 5. Модели представления знаний в интеллектуальных системах

Тема 6. Язык логического программирования Пролог

Тема 7. Экспертные системы

Тема 8. Нейронные сети

Максимальное количество баллов по БРС - 30.

Тестирование

Тема 1. Некоторые известные модели математической физики

Тема 3. Математическая теория борьбы за существование

Максимальное количество баллов по БРС - 20.

Итого $30+20=50$ баллов

Промежуточная аттестация - экзамен – 50 баллов.

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 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 практическое задание.

Устный или письменный ответ – 20 баллов.

Практическое задание – 30 баллов.

Итого $20+30=50$ баллов.

Общее количество баллов по дисциплине за текущий контроль и промежуточную аттестацию: $50+50=100$ баллов.

Соответствие баллов и оценок:

Для экзамена

86-100 – отлично.

71-85 – хорошо.

56-70 – удовлетворительно.

0-55 – неудовлетворительно.

4. Оценочные средства, порядок их применения и критерии оценивания

4.1. Оценочные средства текущего контроля

4.1.1. Лабораторные работы

Тема 1. Некоторые известные модели математической физики

Тема 3. Математическая теория борьбы за существование

Тема 4. Введение в проектирование интеллектуальных систем

Тема 5. Модели представления знаний в интеллектуальных системах

Тема 6. Язык логического программирования Пролог

Тема 7. Экспертные системы

Тема 8. 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

27-30 баллов ставится, если обучающийся:

Правильно выполнил все задания. Продемонстрировал высокий уровень владения материалом. Проявлены превосходные способности применять знания и умения к выполнению конкретных заданий.

22-26 баллов 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.

18-21 баллов 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.

0-17 баллов 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

Лабораторная работа № 1

"Движение тел в среде с учетом трения"

Задания к лабораторной работе

- 1) Выписать математическую модель, определить состав набора входных параметров и их конкретные числовые значения.
- 2) Спроектировать пользовательский интерфейс программы моделирования, обращая особое внимание на формы представления результатов.
- 3) разработать самостоятельно программу интегрирования системы обыкновенных дифференциальных уравнений с оценкой точности расчетов.
- 4) Произвести отладку и тестирование разработанной программы.
- 5) Выполнить конкретное задание из своего варианта работы.
- 6) Качественно проанализировать результаты моделирования.
- 7) Создать текстовый отчет по лабораторной работе, включающий:
 - титульный лист (указать название работы, исполнителя, номер группы и т.д.);
 - постановку задачи и описание модели;
 - результаты тестирования программы;
 - результаты, полученные в ходе выполнения задания (в различных формах).

Варианты заданий

Вариант 1. Парашютист прыгает с некоторой высоты и летит, не открывая парашюта; на какой высоте (или через какое время) ему следует открыть парашют, чтобы иметь к моменту приземления безопасную скорость (не большую 10 м/с)?

Вариант 2. Изучить, как связана высота прыжка с площадью поперечного сечения парашюта, чтобы скорость приземления была безопасной.

Вариант 3. Промоделировать падение тела с заданными характеристиками (массой, формой) в различных вязких средах. Изучить влияние вязкости среды на характер движения. Скорость движения должна быть столь невелика, чтобы квадратичной составляющей силы сопротивления можно было пренебрегать.

Вариант 4. Промоделировать падение тела с заданными характеристиками (массой, формой) в различных плотных средах. Изучить влияние плотности среды на характер движения. Скорость движения должна быть достаточно большой, чтобы линейной составляющей силы сопротивления можно было пренебрегать (на большей части пути).

Вариант 5. Промоделировать движение исследовательского зонда, «выстреленного» вертикально вверх с уровня земли. В верхней точке траектории над зондом раскрывается парашют, и он плавно спускается в точку старта.

Вариант 6. Промоделировать движение исследовательского зонда, «выстреленного» вертикально вверх с летящего над землей самолета. В верхней точке траектории над зондом раскрывается парашют, и он плавно спускается на землю.

- Вариант 7. Глубинная бомба, установленная на взрыв через заданное время, сбрасывается со стоящего неподвижно противолодочного корабля. Исследовать связь между глубиной, на которой произойдет взрыв, и формой корпуса (сферической, полусферической, каплевидной и т.д.).
- Вариант 8. Глубинная бомба, установленная на взрыв на заданной глубине, сбрасывается со стоящего неподвижно противолодочного корабля. Исследовать связь между временем достижения заданной глубины и формой корпуса (сферической, полусферической, каплевидной и т.д.).
- Вариант 9. Провести моделирование взлета ракеты при значениях параметров $m_0 = 2 \cdot 10^7$ кг, $m_{кон} = 2 \cdot 10^5$ кг, $a_{aaa} = 2 \cdot 10^5$ кг/с, $F_{тяги} = 4 \cdot 10^8$ Н. Ответить на вопрос, достигнет ли ракета при этих значениях параметров первой космической скорости 7,8 км/с?
- Вариант 10. Провести исследование соотношения входных параметров m_0 и $F_{тяги}$, при которых ракета достигнет первой космической скорости (и в соответствующий момент исчерпает горючее). Остальные входные параметры фиксировать произвольно. Построить соответствующую фазовую диаграмму в переменных (m_0 , $F_{тяги}$).
- Вариант 11. Разработать и исследовать усовершенствованную модель взлета ракеты, приняв во внимание, что реальные космические ракеты обычно двухи трехступенчатые и двигатели разных ступеней имеют разную силу тяги.
- Вариант 12. Промоделировать движение исследовательского зонда, снабженного разгонным двигателем небольшой мощности, «выстреленного» вертикально вверх с уровня земли. В верхней точке траектории двигатель выключается, над зондом раскрывается парашют, и он плавно спускается в точку старта.
- Вариант 13. Промоделировать движение исследовательского зонда, снабженного разгонным двигателем небольшой мощности, «выстреленного» вертикально вверх с летящего над землей самолета. В верхней точке траектории над зондом раскрывается парашют, и он плавно спускается на землю.
- Вариант 14. Глубинная бомба-торпеда, снабженная разгонным двигателем, установленная на взрыв через заданное время, сбрасывается со стоящего неподвижно противолодочного корабля. Исследовать связь между глубиной, на которой произойдет взрыв, и формой корпуса (сферической, полусферической, каплевидной и т.д.).
- Вариант 15. Глубинная бомба-торпеда, снабженная разгонным двигателем, установленная на взрыв на заданной глубине, сбрасывается со стоящего неподвижно противолодочного корабля. Исследовать связь между временем достижения заданной глубины и формой корпуса (сферической, полусферической, каплевидной и т.д.).
- Вариант 16. Торпеда, снабженная разгонным двигателем, нацеливается с подводной лодки на стоящий вертикально над ней надводный корабль. Исследовать связь между временем поражения цели и формой корпуса (сферической, полусферической, каплевидной и т.д.).
- Вариант 17. Построить траектории и найти временные зависимости горизонтальной и вертикальной составляющих скорости и перемещения для тела массой 1 кг, брошенного под углом 45° к горизонту с начальной скоростью 10 м/с:
1) в воздухе; 2) в воде. Сравнить результаты с теми, которые получились бы без учета сопротивления среды (последние можно получить либо численно из той же модели, либо аналитически).
- Вариант 18. Найти вид зависимости горизонтальной длины полета тела и максимальной высоты траектории от одного из коэффициентов сопротивления среды, фиксировав все остальные параметры. Представить эту зависимость графически и подобрать подходящую аналитическую формулу, определив ее параметры методом наименьших квадратов.
- Вариант 19. Разработать модель подводной охоты. На расстоянии r под углом α подводный охотник видит неподвижную акулу. На сколько метров выше нее надо целиться, чтобы гарпун попал в цель?
- Вариант 20. Поставить и решить задачу о подводной охоте при дополнительном условии: акула движется.
- Вариант 21. Промоделировать движение исследовательского зонда, «выстреленного» под углом к горизонту. В верхней точке траектории над зондом раскрывается тормозной парашют, затем зонд плавно движется до земли.
- Вариант 22. Глубинная бомба, установленная на взрыв через заданное время, сбрасывается с движущегося противолодочного корабля. Исследовать связь между глубиной, на которой произойдет взрыв, пройденным расстоянием по горизонтали и формой корпуса (сферической, полусферической, каплевидной и т.д.).
- Вариант 23. Глубинная бомба-торпеда, снабженная разгонным двигателем, установленная на взрыв на заданной глубине, сбрасывается с движущегося противолодочного корабля. Исследовать связь между временем достижения заданной глубины, пройденным расстоянием по горизонтали и формой корпуса (сферической, полусферической, каплевидной и т.д.).

Лабораторная работа № 2 МОДЕЛИРОВАНИЕ АВТОВОЛНОВЫХ ПРОЦЕССОВ

1. Задача: Имеется двумерная активная среда, состоящая из элементов, каждый из которых может находиться в трех различных состояниях: покое, возбуждении и рефрактерности. При отсутствии внешнего воздействия, элемент находится в состоянии покоя. В результате воздействия элемент переходит в возбужденное состояние, приобретая способность возбуждать соседние элементы. Через некоторое время после возбуждения элемент переключается в состояние рефрактерности, находясь в котором он не может быть возбужден. Затем элемент сам возвращается в исходное состояние покоя, то есть снова приобретает способность переходить в возбужденное

состояние. Необходимо промоделировать процессы, происходящие в двумерной активной среде при различных параметрах среды и начальном распределении возбужденных элементов.

2. Теория. Рассмотрим обобщенную модель Винера - Розенблюта. Мысленно разобьем экран компьютера на элементы, определяемые индексами i, j и образующими двумерную сеть. Пусть состояние каждого элемента описывается фазой $y_{i,j}(t)$, и концентрацией активатора $u_{ij}(t)$, где t -- дискретный момент времени.

Если элемент находится в покое, то будем считать, что $y_{i,j}(t) = 0$. Если вследствие близости возбужденных элементов концентрация активатора $u_{ij}(t)$ достигает порогового значения h , то элемент возбуждается и переходит в состояние 1. Затем на следующем шаге он переключается в состояние 2, затем -- в состояние 3 и т.д., оставаясь при этом возбужденным. Достигнув состояния r элемент переходит в состояние рефрактерности. Через $(s - r)$ шагов после возбуждения элемент возвращается в состояние покоя.

Будем считать, что при переходе из состояния s в состояние покоя 0 концентрация активатора становится равной 0. При наличии соседнего элемента, находящегося в возбужденном состоянии, она увеличивается на 1. Если p ближайших соседей возбуждены, то на соответствующем шаге к предыдущему значению концентрации активатора прибавляется число возбужденных соседей: $u_{ij}(t + \Delta t) = u_{ij}(t) + p$.

Можно ограничиться учетом ближайших восьми соседних элементов.

3. Алгоритм. Для моделирования автоволновых процессов в активной среде необходимо составить цикл по времени, в котором вычисляются фазы элементов среды в последующие моменты времени и концентрация активатора, стирается предыдущее распределение возбужденных элементов и строится новое. Алгоритм модели представлен ниже.

1. Задают число элементов активной среды, ее параметры s, r, h , начальное распределение возбужденных элементов.

2. Начало цикла по t . Дают приращение по времени: переменной t присваивают значение $t + \Delta t$.

3. Перебирают все элементы активной среды, определяя их фазы $y_{i,j}(t + \Delta t)$ и концентрацию активатора $u_{i,j}(t + \Delta t)$ в момент $t + \Delta t$.

4. Очищают экран и строят возбужденные элементы активной среды.

7. Возвращение к операции 2. Если цикл по t закончился, -- выход из цикла.

4. Компьютерная программа. Ниже представлена программа, моделирующая активную среду и происходящие в ней процессы. В программе заданы начальные значения фазы $y_{i,j}(t + \Delta t)$ всех элементов активной среды, а также имеется цикл по времени, в котором рассчитываются значения $y_{i,j}(t + \Delta t)$ в следующий момент $t + \Delta t$ и осуществляется графический вывод результатов на экран. Параметры среды $r = 6, s = 13, h = 5$, то есть каждый элемент кроме состояния покоя может находиться в 6 возбужденных состояниях и 7 состояниях рефрактерности. Пороговое значение концентрации активатора равно 5. Программа строит однорукавную волну, осциллятор и препятствие.

Program PROGRAMMA6;

uses dos, crt, graph;

Const N=110; M=90; s=13; r=6; h=5;

Var y, yy, u : array [1..N,1..M] of integer;

ii, jj, j, k, Gd, Gm : integer; i : Longint;

Label met;

BEGIN

Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');

If GraphResult <> grOk then Halt(1);

setcolor(8); setbkcolor(15);

(* y[50,50]:=1; { Одиночная волна } *)

For j:=1 to 45 do { Однорукавная волна }

For i:=1 to 13 do y[40+i,j]:=i;

(* For j:=1 to M do { Двuruкавная волна }

For i:=1 to 13 do begin y[40+i,j]:=i;

If j>40 then y[40+i,j]:=14-i; end; *)

Repeat

If k=round(k/20)*20 then y[30,30]:=1; { Осциллятор 1 }

(* If k=round(k/30)*30 then y[20,50]:=1; { Осциллятор 2 } *)

For i:=2 to N-1 do For j:=2 to M-1 do begin

If (y[i,j]>0) and (y[i,j]<s) then yy[i,j]:=y[i,j]+1;

If y[i,j]=s then begin yy[i,j]:=0; u[i,j]:=0; end;

If y[i,j] <> 0 then goto met;

For ii:=i-1 to i+1 do For jj:=j-1 to j+1 do begin

If (y[ii,jj]>0) and (y[ii,jj]<=r) then u[i,j]:=u[i,j]+1;

If u[i,j]>=h then yy[i,j]:=1; end;

met: end; Delay(2000); { Задержка }

cleardevice;

For i:=21 to 70 do begin

yy[i,60]:=0; yy[i,61]:=0; { Препятствие }

circle(6*i-10,500-6*60,3); circle(6*i-10,500-6*61,3); end;

```

For i:=1 to N do For j:=1 to M do
begin y[i,j]:=yy[i,j]; setcolor(12);
If (y[i,j]>=1) and (y[i,j]<=r) then circle(6*i-10,500-6*j,3);
setcolor(8);
If (y[i,j]>6) and (y[i,j]<=s) then circle(6*i-10,500-6*j,2);
end;
until KeyPressed;
CloseGraph;
END.

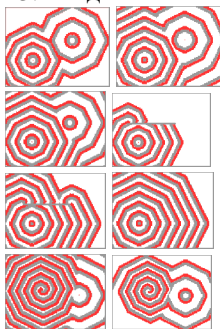
```

5. Задания для самостоятельного решения.

1. Получите модель одиночной волны возбуждения. Для этого достаточно один из элементов активной среды перевести в возбужденное состояние.
2. Промоделируйте серию автоволн. Для этого необходимо, чтобы один из элементов совершал периодические колебания, то есть автоматически через заданное число шагов переходил в возбужденное состояние 1. Такой элемент называется осциллятором. Для получения серии автоволн следует активизировать строку с пометкой "Осциллятор 1".
3. Промоделируйте дифракцию автоволн. Для этого необходимо создать волну, на пути которой расположено препятствие, например, непрозрачный экран, состоящий из невозбуждающихся элементов, расположенных вдоль прямой и всегда находящихся в состоянии 0.
4. Изучите распространение автоволн в двумерной среде, содержащей два параллельно расположенных экрана или экран с отверстием. Пронаблюдайте аннигиляцию автоволн, распространяющихся навстречу друг другу.
5. Промоделируйте эффект синхронизации, состоящий в том, что при наличии двух или более источников автоволн происходит их взаимодействие, в результате которого высокочастотные источники подавляют низкочастотные. В конце концов наступает синхронизация колебаний элементов среды: колебания происходят с частотой, равной частоте высокочастотного источника. Чтобы пронаблюдать это явление на экране компьютера, следует смоделировать два осциллятора, работающих на разных частотах. Для этого необходимо активизировать операторы с пометками "Осциллятор 1" и "Осциллятор 2".



6. Промоделируйте образование однорукавных спиральных волн. Спиральные волны образуются на краях фронта волны, поэтому для моделирования этого процесса необходимо в блоке начальных условий задать плоскую волну, фронт которой обрывается в середине экрана.
7. Промоделируйте образование двухрукавных спиральных волн.
8. Изучите зависимость частоты вращения однорукавной спиральной волны от параметров среды (r , s , h). Повторите этот вычислительный эксперимент для двухрукавной волны.
9. Промоделируйте взаимодействие спиральных автоволн с автоволнами, вырабатываемыми осциллятором, колеблющимся с низкой частотой.
10. Исследуйте распространение и аннигиляцию одиночного импульса в одномерной активной среде.
11. Изучите распространение автоволн в одномерной активной среде при наличии осциллятора.
12. Промоделируйте распространение одиночного импульса в одномерной активной среде, последний элемент которой контактирует с первым.
13. Создайте компьютерную модель распространения автоволн в трехмерной активной среде.



Лабораторная работа 3. 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 делении на 3, вывода 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 $X \geq Y$, 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 истинен*/
```

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 $y = a \cdot x + b$, 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:

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

Лабораторная работа 4. 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 $\sum_{i=1}^N i^i$.
- 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 $N! + (N-1)! + \dots + 2! + 1!$.
- 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 5. 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:

- если к пустому списку [] добавить список P, то в результате получится P;
- 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:

- для разбиения списка на подсписки:

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

Ответ: X=[] Y=[1,2]

X=[1] Y=[2]

X=[1,2] Y=[]

- 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, [_ | 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

Программа 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: $a_1+b_n, a_2+b_{n-1}, \dots, a_{n-1}+b_2, a_n+b_1$.
- 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 6. 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

Цвет \ Фамилия	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 подружки вышли в белом, зеленом и синем платьях и туфлях. Известно, что только у Ани цвета платья и туфель 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 находится 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 Y и меньше T; Y больше Z и больше T; Z больше X и меньше T. В каком порядке расположены эти числа.

Лабораторная работа 7. 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.

Поле для раскраски

a		
b	c	d
e		f

- 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 $n \times n$, элемент $a_{ij} < 0$, 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 O. Закрашенная клетка – стена.

Таблица 11.

		O		

- 9) Быки и коровы. Игрок A выбирает секретный код, представляющий последовательность из N различных десятичных цифр ($N=4$). Игрок пытается угадать задуманный код и спрашивает игрока A 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 8. Обработка файлов. Предикаты для работы с файлами

deletefile(DOS_filename)-удаление файла

save(DOS_filename) –сохранение файла

renamefile(old_DOS_filename, new_DOS_filename)-переименовывание 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) –Переменной 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 переменной File_name.

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

Пример:

file=datafile

file=datafile1;datafile2

openwrite(datafile,filename) - открытие файла для записи или создание, где datafile- введенный 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 fileros.
 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(datafile,"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,
    writedevicе(datafile),
    write(" ",Line),nl,
    writedevicе(screen), /*для вывода данных на экран*/
    fail, write_lines.
  
```

Пример 2. Вывести данные файла на экран.

Программа 28. Чтение данных из 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),
    writedevicе(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 ввода слова "done"*/
readin(_,Cstr) :-
write(Cstr),
readln(Dstr1),
concat(Dstr1, "\13\10",Cstr1),
writedevice(datafile),
readin(Dstr1,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 9. 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 10. 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 соответствии 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 11. 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.

Щелкните 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.

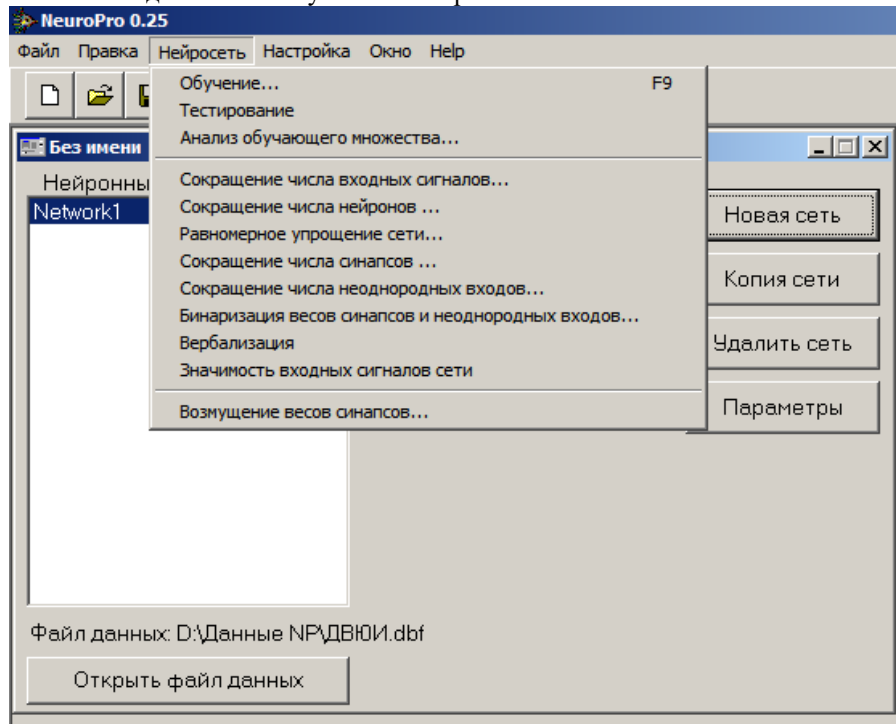
Меню программы содержит следующие пункты, относящиеся к нейронным сетям и работе с ними:

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

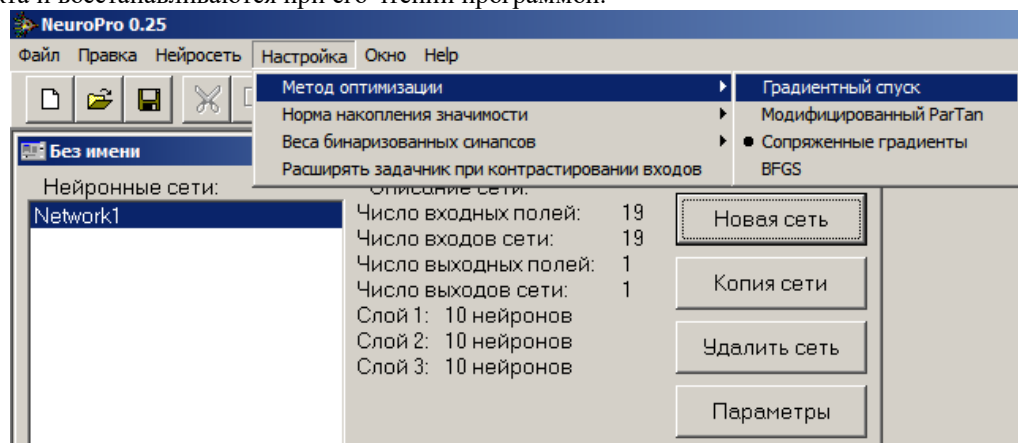
Нейросеть - операции с нейронными сетями. Операция выполняется над активной в данный момент в нейропроекте нейронной сетью и включает в себя следующие опции:

- 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 (ParTap) и метод сопряженных градиентов) при создании нейропроекта автоматически предлагается ParTap.
- **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 (ParTan) и метод сопряженных градиентов) при создании нейропроекта автоматически предлагается ParTan.

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 — неконформизм»
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 «РАБОТА» 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.

ПРОГНОЗИРОВАНИЕ.

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 «ТАВ» 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 отчет.

Лабораторная работа 12. Нейросетевое прогнозирование колебаний котировок валют

Курс доллара США к российскому рублю на сегодня 6 февраля 2015, составил **68.6113**.



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.

- Создайте, обучите, протестируйте, упростите и вербализуйте нейронную сеть.
- По обученной нейронной сети выполните прогноз курса доллара США с горизонтом прогнозирования 3 дня (с 07.02.2015 по 09.02.2015).
- Постройте в табличном редакторе EXCEL линию тренда с уравнением в виде полинома 2 степени и прямой) и выполните прогноз вперед. Проведите корреляционный анализ показателей.
- Сравните результаты прогнозируемых параметров, полученных с использованием нейронных сетей и Excel.
- 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.
- Сделайте выводы и оформите отчет по выполненной лабораторной работе.

Лабораторная работа 13. Нейросетевое распознавание сложных образов (текста)

Часть 1

- Запустите Microsoft Excel.
 - Оставьте первую строку пустой.
 - 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.
 - 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»).
 - Нажмите кнопку «НОВАЯ СЕТЬ» и задайте выходным сигналом столбец с нумерацией строк.
 - Установите точность 0,01.
 - Нажмите кнопку «СОЗДАТЬ».
 - В верхнем меню нажмите «НЕЙРОСЕТЬ» – «ОБУЧЕНИЕ» – после окончания обучения – «ГОТОВО».
 - Щелкните л.к.м. на таблице с данными.
 - Выберите следующие пункты меню: «ТАБЛИЦА» – «ДОБАВИТЬ ЗАПИСЬ».
 - Создайте новый пункт, написав цифрами любое слово с ошибкой.
 - Спрогнозируйте, используя «Нейросеть» – «Тестирование».
- Прогнозируемый параметр должен быть близким к порядковому номеру слова.

Часть 2

- Запустите Microsoft Excel.
- Оставьте первую строку пустую.
- 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. Прогнозируйте, 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. Тестирование

Тема 1. Некоторые известные модели математической физики

Тема 3. Моделирование процессов диффузии и переноса

4.1.2.1. Порядок проведения.

Тестирование проходит в письменной форме или с использованием компьютерных средств. Обучающийся получает определённое количество тестовых заданий. На выполнение выделяется фиксированное время в зависимости от количества заданий. Оценка выставляется в зависимости от процента правильно выполненных заданий. Тестирование проводится по вариантам. В каждом варианте – 10 тестовых заданий. За каждый правильный ответ начисляется 1 балл. Итого за тестирование студент может заработать до 10 баллов.

Ниже приведены примерные задания. Полный банк тестовых заданий хранится на кафедре.

4.1.2.2 Критерии оценивания

17-20 баллов ставится, если обучающийся:

86% правильных ответов и более.

14-16 баллов ставится, если обучающийся:

От 71% до 85 % правильных ответов.

11-13 баллов ставится, если обучающийся:

От 56% до 70% правильных ответов.

0--10 балла ставится, если обучающийся:

55% правильных ответов и менее.

4.1.2.3. Содержание оценочного средства

Вопрос 1. На какие части разделяют описание системы для целей имитационного моделирования?

- 1) Детерминированную и стохастическую;
- 2) Предварительную и уточненную;
- 3) Статическую и динамическую;
- 4) Линейные, динамические, сетевые, массового обслуживания и т.д.;
- 5) Физическую и математическую.

Вопрос 2. Каков первый шаг при создании модели?

- 1) Определение структуры системы;
- 2) Определение ее целевого назначения;
- 3) Выбор типа модели (линейная, динамическая и т.д.);
- 4) Определение входных параметров системы;
- 5) Определение выходных параметров системы.

Вопрос 3. Что такое устойчивая система?

- 1) Отсутствуют реакции на внешнее воздействие;
- 2) Система сама устанавливает цели функционирования и стремится к ним;
- 3) Система стремится перейти в оптимальное состояние, которое в настоящее время отсутствует;
- 4) При изменении внешних условий изменение состояния системы мало;
- 5) Система, стремящаяся сохранить свое состояние при изменении внешних условий.

Вопрос 4. Что такое поисковая система?

- 1) Отсутствуют реакции на внешнее воздействие;
- 2) Система сама устанавливает цели функционирования и стремится к ним;
- 3) Система стремится перейти в оптимальное состояние, которое в настоящее время отсутствует;
- 4) При изменении внешних условий изменение состояния системы мало;

5) Система, стремящаяся сохранить свое состояние при изменении внешних условий.

Вопрос 5. Что такое целеустремленная система?

- 1) Отсутствуют реакции на внешнее воздействие;
- 2) Система сама устанавливает цели функционирования и стремится к ним;
- 3) Система стремится перейти в оптимальное состояние, которое в настоящее время отсутствует;
- 4) При изменении внешних условий изменение состояния системы мало;
- 5) Система, стремящаяся сохранить свое состояние при изменении внешних условий.

Вопрос 6. Что такое фактографические элементы решения?

- 1) Элементы, которые можно подвергнуть проверке с целью определения их истинности;
- 2) Элементы, которые нельзя подвергнуть проверке;
- 3) Элементы, которые будут включены в модель;
- 4) Элементы, исключаемые из модели;
- 5) Элементы, входящие в критерий.

Вопрос 7. Что такое оценочные элементы решения?

- 1) Элементы, которые можно подвергнуть проверке с целью определения их истинности;
- 2) Элементы, которые нельзя подвергнуть проверке;
- 3) Элементы, которые будут включены в модель;
- 4) Элементы, исключаемые из модели;
- 5) Элементы, входящие в критерий.

Вопрос 8. Нужно ли проверять данные, полученные из документов, или интервью с экспертами?

- 1) Ненужно для официальных документов;
- 2) Ненужно, если эксперт признанный специалист;
- 3) Нужно, если компетентность эксперта не подтверждена;
- 4) Нужно, если документы текущие (рабочие), а не официальные;
- 5) Нужно проверять и сопоставлять с другими источниками все данные без исключений.

Вопрос 9. Почему заказчик описывает проблему в неточных и нечетких формулировках?

- 1) Потому, что он не математик;
- 2) Это нам так кажется, так как мы не знакомы с проблемой;
- 3) Если бы он знал точно в чем суть проблемы и как ее решать, он сделал бы это сам;
- 4) У разных специалистов разный взгляд на одно и тоже решение;
- 5) Заказчик не знаком с ЭВМ.

Вопрос 10. Почему при моделировании большое значение придается планированию состава и объемов исследовательских работ и руководству проектом?

- 1) Потому, что заказчик хочет знать стоимость работы;
- 2) Чтобы правильно определить структуру модели;
- 3) Чтобы привлечь к работе специалистов заказчика;
- 4) Потому, что временные и материальные затраты на построение даже простых моделей очень высоки и нужно точно знать, что должно быть сделано, а что делать не нужно;
- 5) Потому, что математические методы моделирования сложны.

Вопрос 11. На чем основан системный подход?

- 1) На признании факта, что из оптимальности элементов следует оптимальность системы;
- 2) На признании факта, что из оптимальности элементов не следует оптимальность системы;
- 3) На факте, что независимо от того как система пришла в данное состояние, если в дальнейшем она будет менять состояния оптимальным способом то и весь процесс будет оптимальным;
- 4) На факте, что если система пришла в данное состояние оптимальным способом, то независимо от того как, в дальнейшем она будет менять состояния весь процесс будет оптимальным;
- 5) На разделении системы на несколько подсистем, поддающихся анализу.

Вопрос 12. В чем заключается системный подход?

- 1) В том, что исследователь пытается изучать систему в целом, а не концентрировать внимание на ее частях;
- 2) В способе деления системы на элементы;
- 3) В том, что исследователь пытается изучать систему по частям;
- 4) Это правила выбора структуры модели;
- 5) Это правила выбора элементов модели.

Вопрос 13. Является ли системный подход научным методом?

- 1) Нет, это подход к исследованию, основанный на здравом смысле;
- 2) Да, это научный метод анализа систем;
- 3) Нет, это метод деления систем на элементы;
- 4) Да, это научный метод построения структурных моделей;
- 5) Да, это методика определения важнейших факторов.

Вопрос 14. Как понимать взаимозависимость?

- 1) Зависимость системы от внешней среды;
- 2) Воздействие системы на внешнюю среду;
- 3) Зависимость системы от внешней среды и одновременно воздействие системы на внешнюю среду;
- 4) Никакие действия в сложной системе невозможно полностью изолировать;
- 5) Воздействие одного фактора системы на другой.

Вопрос 15. Что такое противоинтуитивное поведение?

- 1) Когда недостаточно методов статистического анализа;
- 2) Когда выходные переменные оказывают влияние на входные;
- 3) Когда внесение того или иного корректирующего воздействия может оказаться неэффективным или привести к обратным результатам;
- 4) Когда нельзя без использования математических методов обнаружить связь между факторами;
- 5) Когда зависимость между факторами описывают обратной функцией.

Вопрос 16. Какая задача решается на этапе преобразования знаний о системе в математическую модель?

- 1) Определение назначения модели;
- 2) Определение компонентов, которые должны быть включены в состав модели;
- 3) Определение стоимости исследования;
- 4) Определение параметров и переменных, относящихся к компонентам модели;
- 5) Нет правильного ответа.

Вопрос 17. Какое главное противоречие необходимо преодолевать при конструировании модели?

- 1) Линейность и нелинейность;
- 2) Детерминированность и стохастичность;
- 3) Стоимость и сложность;
- 4) Простота и точность;
- 5) Полезность и общность.

Вопрос 18. Что такое прямая задача?

- 1) Найти отклик системы на заданные входные сигналы;
- 2) Найти входные сигналы, вызвавшие данный отклик;
- 3) Найти математическое описание системы, если заданы входные сигналы и отклик;
- 4) Найти входные сигналы, вызвавшие данный отклик или математическое описание системы, если заданы входные сигналы и отклик;
- 5) Найти структуру системы.

Вопрос 19. Что такое обратная задача?

- 1) Найти отклик системы на заданные входные сигналы;
- 2) Найти входные сигналы, вызвавшие данный отклик;
- 3) Найти математическое описание системы, если заданы входные сигналы и отклик;
- 4) Найти входные сигналы, вызвавшие данный отклик или математическое описание системы, если заданы входные сигналы и отклик;
- 5) Найти структуру системы.

Вопрос 20. Что такое «черный ящик»?

- 1) Нелинейный элемент системы;
- 2) Элемент системы с линейной функцией;
- 3) Элемент системы, функция которого не влияет на поведение модели;
- 4) Элемент системы, исключаемый из рассмотрения;
- 5) Элемент системы, природа которого неизвестна или мало известна.

Задание 21.

Продолжить изучение главы 4.

Выбрать правильный вариант ответа и отметить в карточке ответов.

Вопрос 1. В каких пределах изменяется кумулятивная вероятность?

- 1) $(-1;0)$;
- 2) $(-1;1)$;
- 3) $(0;1)$;
- 4) $(0;1)$;
- 5) $(-1;1)$.

Вопрос 22. Прodelайте сто опытов с моделью «пьяного прохожего». Оцените результат. Выберите среди перечисленных, правильный ответ на вопрос: «Какова вероятность того, что пройдя 10 кварталов, он окажется не далее 2 кварталов от исходной точки?»

- 1) 0,6;
- 2) 0,4;
- 3) 0,7;
- 4) 0,5;
- 5) 0,33.

Вопрос 23. Что является основой для идентификации закона распределения?

- 1) Гистограмма, построенная по теоретическим данным;
- 2) Формула функции распределения;
- 3) Таблица наблюдений;
- 4) График функции распределения;
- 5) Гистограмма, построенная по экспериментальным данным.

Вопрос 24. Является ли несовпадение среднего и дисперсии экспериментальных данных и теоретического распределения достаточным основанием, чтобы отвергнуть гипотезу об идентичности распределений?

- 1) Да, так как расхождение во входных параметрах приведет к еще большей ошибке моделирования;

- 2) Нет, требуются дополнительные оценки;
- 3) Да;
- 4) Нет;
- 5) Да, так как параметры распределений должны совпадать.

Вопрос 25. Является ли совпадение среднего и дисперсии экспериментальных данных и теоретического распределения достаточным основанием, чтобы принять гипотезу об идентичности распределений?

- 1) Да, так как математическое ожидание и дисперсия – главные параметры распределения вероятностей;
- 2) Нет, требуются дополнительные оценки;
- 3) Да;
- 4) Нет;
- 5) Да, так как параметры распределений должны совпадать.

Вопрос 26. Что показывает статистика s^2 ?

- 1) Принадлежность экспериментальных данных к нормальному распределению;
- 2) Расхождение между наблюдаемыми и ожидаемыми значениями частот появления случайной величины;
- 3) Принадлежность экспериментальных данных к распределению Пуассона;
- 4) Принадлежность экспериментальных данных к распределению s^2 ;
- 5) Расхождение между наблюдаемыми и ожидаемыми значениями случайных величин.

Вопрос 27. Какое значение должна принять статистика s^2 , чтобы отбросить гипотезу об отсутствии существенных различий между наблюдаемыми данными и данными, получаемыми на основе теоретического распределения?

- 1) $s^2 > 0$;
- 2) $s^2 > 1$;
- 3) $s^2 < 0$;
- 4) s^2 больше критического;
- 5) $s^2 < 1$.

Вопрос 28. На чем основан критерий Колмогорова – Смирнова?

- 1) На наибольшей абсолютной разнице между экспериментальным и теоретическим интегральными распределениями;
- 2) На статистике s^2 ;
- 3) На наибольшей абсолютной разнице между экспериментальным и теоретическими случайными величинами;
- 4) На наименьшей абсолютной разнице между экспериментальным и теоретическим интегральными распределениями;
- 5) На наибольшей абсолютной разнице между частотами экспериментального и теоретического распределений.

Ответы:

- 1) 1;
- 2) 4;
- 3) 4;
- 4) 4;
- 5) 1;
- 6) 2;
- 7) 3;
- 8) 6;
- 9) 4;
- 10) 5;
- 11) 4;
- 12) 3;
- 13) 2;
- 14) 2;
- 15) 3;
- 16) 3;
- 17) 3;
- 18) 5;
- 19) 3;
- 20) 1; 4;
- 21) 2;
- 22) 4;
- 23) 2;
- 24) 1;
- 25) 1;
- 26) 2;
- 27) 3;
- 28) 4.

4.2. Оценочные средства промежуточной аттестации

По дисциплине предусмотрен экзамен. Экзамен проходит по билетам. В каждом билете один теоретический вопрос и одно практическое задание. 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 ответ

4.2.1.1. Порядок проведения.

Промежуточная аттестация нацелена на комплексную проверку освоения дисциплины. Обучающийся получает вопрос(ы)/задание(я) и время на подготовку. Промежуточная аттестация 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.

4.2.1.2. Критерии оценивания.

18-20 баллов 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.

14-17 баллов 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.

11-13 баллов 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.

0--10 баллов 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. Модели. Моделирование как метод познания. Формализация. Классификация абстрактных моделей. Компьютерное моделирование.
2. Цели и основные этапы компьютерного математического моделирования. Примеры моделей для различных целей моделирования.
3. Этап формализации. Параметры модели. Классификация моделей по свойствам их параметров. Ранжирование параметров. Устойчивость решений. Анализ результатов моделирования.
4. Различные подходы к классификации математических моделей.
5. Основные виды средств компьютерного моделирования. Визуализация в компьютерном моделировании. Алгоритмы построения графиков функций, траекторий движения объектов.
6. Представление скалярных полей с помощью изолиний. Методы условных цветов, условного контрастирования. Примеры использования визуализации в моделировании.
7. Аналитическое моделирование в физике. Примеры. Классификация моделей по общематематическим свойствам: линейные и нелинейные модели. Примеры. Линеаризация. Интегрирование дифференциальных уравнений.
8. Численное моделирование. Развитие физических теорий. Численный эксперимент. Его взаимосвязи с теорией и лабораторным экспериментом. Модель движения тела, брошенного под углом к горизонту.
9. Достоверность численной модели. Ограничения чисел с плавающей точкой как модели действительных чисел. Обусловленность задач. Устойчивость вычислительных алгоритмов. Анализ и интерпретация численных моделей.
10. Детерминированные физические модели. Свободное падение тела с учетом сопротивления среды. Модель взлета ракеты.
11. Классификация задач обработки информации. Искусственный интеллект. "Может ли машина мыслить?"
12. Представление знаний. Классификация знаний. Алгоритмические, фактуальные и концептуальные знания.
13. Логические модели знаний.
14. Лямбда-исчисление.
15. Семантические сети представления знаний.
16. Фреймовые модели знаний.
17. Продукционные модели знаний.
18. Классификация ядер продукции.
19. Стратегии управления выполнением продукций.
20. Экспертные системы.

21. Однослойный перцептрон. Активационные функции (пороговая, сигмоид).
22. Многослойные нейронные сети.
23. Обучение с помощью алгоритма обратного распространения.
24. Функции принадлежности в нечеткой логике.
25. Функции ограничения в нечеткой логике.
26. Операции в нечеткой логике.

4.2.2. Практическое задание

4.2.2.1. Порядок проведения.

Предлагаются задания на решение задач по курсу компьютерное моделирование.

4.2.2.2. Критерии оценивания.

27-30 баллов ставится, если обучающимся:

Задание выполнено полностью и правильно.

22-26 баллов ставится, если обучающимся:

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

18-21 баллов ставится, если обучающимся:

Задание выполнено частично или с фактическими ошибками.

0-17 баллов ставится, если обучающимся:

Задание не выполнено или выполнено с большим количеством фактических ошибок.

4.2.2.3. Оценочные средства.

1. Рассмотрим популяцию, размножающуюся пропорционально количеству особей, продукты ее жизнедеятельности превращаются в среде обитания в яд, истребляющий популяцию. Найти зависимость количества особей от времени.
2. Цилиндрический бак поставлен вертикально и имеет отверстие в дне. Половина воды из полного бака вытекает за 5 минут. За какое время вытечет вся вода?
3. Определить температуру бесконечного круглого цилиндра радиуса R , если его начальная температура T_0 , а на поверхности поддерживается нулевая температура.
4. Определить критический размер куба из активного вещества, если его коэффициент размножения больше нуля, а концентрация вещества на всех гранях поддерживается равной нулю.
5. Сформулировать задачу о плавлении твердой фазы, занимающей полупространство $x > 0$, при наличии теплового потока, поступающего через поверхность $x = 0$.
6. Преобразовать уравнение Кортевега-де Фриза
7. Показать, что модифицированное уравнение Кортевега-де Фриза имеет солитонное решение.
8. Найти автомодельную подстановку для одномерного уравнения теплопроводности на полуоси и решить задачу
9. Найти солитонное решение нелинейного уравнения колебаний струны
10. Найти решение уравнения Д'Аламбера

Перечень литературы, необходимой для освоения дисциплины (модуля)

Направление подготовки: 44.03.01 - Педагогическое образование

Профиль подготовки: Технология и робототехника

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2024

Основная литература:

1. Компьютерное моделирование инновационной деятельности промышленных предприятий: учебное пособие / Ю. А. Антохина, А. Г. Варжапетян, Е. Г. Семенова, М. С. Смирнова. — Санкт-Петербург: ГУАП, 2021. — 189 с. — ISBN 978-5-8088-1598-8. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/216482>
2. Макшанов, А. В. Большие данные. Big Data / А. В. Макшанов, А. Е. Журавлев, Л. Н. Тындыкарь. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 188 с. — ISBN 978-5-8114-9690-7. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/198599>
3. Практикум по информатике: учебное пособие / Н. М. Андреева, Н. Н. Василюк, Н. И. Пак, Е. К. Хеннер. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 248 с. — ISBN 978-5-8114-2961-5. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/205961>
4. Потопахин, В. В. Романтика искусственного интеллекта / Потопахин В. В. - Москва: ДМК Пресс, 2017. - 170 с. - ISBN 978-5-97060-476-2. - Текст: электронный // ЭБС "Консультант студента": [сайт]. - URL : <https://www.studentlibrary.ru/book/ISBN9785970604762.html>
5. Сердюков, Ю. М. Философия виртуальной реальности и искусственного интеллекта: учебное пособие / Ю. М. Сердюков; под редакцией Ю. М. Сердюкова. — Хабаровск: ДВГУПС, 2020. — 169 с. — ISBN 978-5-262-00881-0. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/179385>

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

Направление подготовки: 44.03.01 - Педагогическое образование

Профиль подготовки: Технология и робототехника

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2024

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

1. Microsoft office professional plus 2010
2. Kaspersky Endpoint Security для Windows
3. Электронная библиотечная система «ZNANIUM.COM»
4. Электронная библиотечная система Издательства «Лань»
5. Электронная библиотечная система «Консультант студента»