МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«ИВАНОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Шуйский филиал ИвГУ

Кафедра математики, информатики и методики обучения

ОДОБРЕНО:
Руководитель ОП
_____ С.А. Зайцева
(подпись)
«21» июня 2023 г.

Рабочая программа дисциплины
ОСНОВЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Уровень высшего образования:	бакалавриат
Квалификация выпускника:	бакалавр
Направление подготовки:	44.03.05 Педагогическое образование (с двумя профилями подготовки)
Направленность (профиль) образовательной программы:	Математика; Информатика

Шуя

1. Цели освоения дисциплины: содействовать становлению профессиональной компетентности педагога через формирование целостного представления об основах искусственного интеллекта и их использовании в различных сферах, а также развитие у студентов практических навыков работы со средствами языков Lisp и Visual Prolog.

2. Место дисциплины в структуре ОП

Дисциплина «Основы искусственного интеллекта» является дисциплиной по выбору, относится к формируемой участниками образовательных отношений части блока дисциплин Б1, входит в предметно-методический модуль.

Успешное освоение данной дисциплины будет способствовать готовности обучающихся к освоению дисциплин: «История информатики и методики обучения», «Основы робототехники»; успешному прохождению следующих практик: «Производственная практика, научно-исследовательская работа, по предметно-методическому модулю», «Производственная практика, педагогическая (комплексная) практика, по предметно-методическому модулю», подготовке к сдаче и сдаче государственного экзамена, выполнению и защите выпускной квалификационной работы.

Студент, приступающий к изучению данной дисциплины, должен:

Знать:

- особенности системного и критического мышления;
- основы современных технологий сбора, обработки и представления информации;
- основные понятия информатики, их системно-структурные связи в контексте развития информационно-коммуникационных технологий;
- теоретические сведения об архитектуре современного компьютера;
- основные понятия и термины, связанные с алгоритмизацией и программированием.

Уметь:

- анализировать источники информации с точки зрения временных и пространственных условий их возникновения;
- получать, обрабатывать и анализировать информацию из различных источников;
- использовать современные информационно-коммуникационные технологии (включая пакеты прикладных программ, локальные и глобальные компьютерные сети) для сбора, обработки и анализа информации;
- работать с математической и технической литературой;
- методами математического доказательства гипотез;
- использовать программную поддержку курса и оценивать ее методическую целесообразность.

Владеть:

- навыками сопоставления разных источников информации с целью выявления их противоречий и поиска достоверных суждений;
- основными методами математической обработки информации;
- базовыми понятиями и терминами в области математики и информатики;

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

3.1. Компетенции, формированию которых способствует данная дисциплина

При реализации дисциплины формируются следующие компетенции в соответствии с ФГОС ВО по данному направлению подготовки:

а) универсальные компетенции (УК) и индикаторы их достижения:

Наименование категории (группы) универсальных компетенций	Код и наименование универсальной компетенции выпускника	Код и наименование индикатора достижения универсальной компетенции
Системное и критическое мышление	УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1. Демонстрирует знание особенностей системного и критического мышления и готовность к нему. УК-1.2. Применяет логические формы и процедуры, способен к рефлексии по поводу собственной и чужой мыслительной деятельности. УК-1.3. Анализирует источник информации с точки зрения временных и пространственных условий его возникновения. УК-1.4. Анализирует ранее сложившиеся в науке оценки информации. УК-1.5. Сопоставляет разные источники информации с целью выявления их противоречий и поиска достоверных суждений. УК-1.6. Аргументировано формирует собственное суждение и оценку информации, принимает обоснованное решение. УК-1.7. Определяет практические последствия предложенного решения задачи.

б) общепрофессиональные компетенции (ОПК) и индикаторы их достижения:

–

в) профессиональные компетенции (ПК) и индикаторы их достижения:

Код и наименование профессиональной компетенции выпускника	Код и наименование индикатора достижения профессиональной компетенции	Основание (ПС, анализ опыта)
Тип задач профессиональной деятельности: педагогический		
ПК-6. Способен выделять структурные элементы, входящие в систему познания предметной области (в соответствии с профилем и уровнем обучения), анализировать их в единстве содержания, формы и выполняемых функций	ПК-6.1. Выделяет и анализирует единицы различных уровней знаковой математической структуры в единстве их содержания, формы и функций. ПК-6.2. Выделяет и анализирует закономерности и явления информатики как науки в их структурном и функциональном единстве с информационно-коммуникационными технологиями. ПК-6.3. Знает и умеет анализировать аксиомы, определения и математические доказательства в единстве с прикладными аспектами использования.	Профессиональный стандарт 01.001 «Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель)»
ПК-7. Способен определять собственную позицию относительно дискуссионных проблем предметной области (в соответствии с профилем и уровнем обучения)	ПК-7.1. Проявляет мировоззренческую рефлексию при анализе проблем и тенденций в области Математики и Информатики. ПК-7.2. Проявляет способность аргументированно, логические верно и ясно выражать свою позицию по обсуждаемым дискуссионным проблемам в сочетании с готовностью к конструктивному диалогу и толерантному восприятию иных точек зрения.	Профессиональный стандарт 01.001 «Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего,

	ПК-7.3. Выделяет концептуальную основу дискуссий в области Математики и Информатики, соотносит с нею свои мировоззренческие установки, гражданскую позицию и социальную мотивацию.	среднего общего образования) (воспитатель, учитель)»

3.2. Планируемые результаты обучения по дисциплине, соотнесенные с индикаторами достижения формируемых компетенций

В результате освоения дисциплины обучающийся должен:

Знать:

– логические формы и процедуры (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– современные пакеты прикладных программ статистической обработки данных (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– приемы тестирования, анализа и настройки системных и прикладных программ (ПК-6 (индикаторы – 1, 2, 3).

Уметь:

– использовать современные информационно-коммуникационные технологии (включая пакеты прикладных программ, локальные и глобальные компьютерные сети) для сбора, обработки и анализа информации (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– использовать программную поддержку курса и оценивать ее методическую целесообразность (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– использовать различного рода прикладные программы и языки программирования для решения учебных задач (ПК-6 (индикаторы – 1, 2, 3);

– опознавать изученные математические понятия и структуры, определять их характеристики с точки зрения области применения и функционирования (ПК-6 (индикаторы – 1, 2, 3);

– анализировать и характеризовать готовый программный продукт в контексте решения учебных задач (ПК-6 (индикаторы – 1, 2, 3);

– выявлять в конкретной решенной задаче ошибки и недочеты, определять их природу (ПК-6 (индикаторы – 1, 2, 3).

Владеть:

– навыками сопоставления разных источников информации с целью выявления их противоречий и поиска достоверных суждений (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– технологиями алгоритмизации и программирования (ПК-6 (индикаторы – 1, 2, 3);

– механизмами оценки применимости математических положений в прикладных областях (ПК-7 (индикаторы – 1, 2, 3);

– спорными и противоречивыми проблемами в области построения научных теорий математики и информатики (ПК-7 (индикаторы – 1, 2, 3).

4. Объём и содержание дисциплины

Очная форма обучения

Объём дисциплины составляет 4 зачётных единицы (108 час.)

Контактная работа 46,83 час.

Самостоятельная работа 61,17 час.

Виды учебных занятий и формы промежуточной аттестации	8 сем.	
	Контактная работа с преподавателем	Самостоятельная работа обучающегося
Лекции	12	43,5
Лабораторные работы	12	
Практические занятия	13,5	
Групповые консультации перед экзаменом	1,5	-
Экзамен	0,33	25,17
Всего	39,33	68,67

4.1. Содержание дисциплины по разделам (темам), соотнесенное с видами и объемом учебных занятий

Очная форма обучения

№ п/п	Раздел (тема) дисциплины	Семестр	Виды учебных занятий, включая самостоятельную работу обучающихся и объем (в часах)				Формы текущего контроля успеваемости Формы промежуточной аттестации
			Занятия лекционного типа	Занятия семинарского типа		Самостоятельная работа обучающегося	
				Лаб.	Пр.		
1.	Введение в теорию искусственного интеллекта	8	2			7,5	доклад
2.	Компьютерные средства разработки ИИ	8	2	3	3	6	практические и лабораторные задания на компьютере
3.	Языки программирования ИИ	8	2	6	7,5	9	практические и лабораторные задания на компьютере
4.	Основы теории экспертных систем. Основы теории представления знаний	8	2	1,5	1,5	9	практические и лабораторные задания на компьютере
5.	Проблематика и технологии экспертных систем	8	2	1,5	1,5	6	практические и лабораторные задания на компьютере
6.	Искусственный интеллект в системе образования	8	2			6	доклад
ИТОГО			12	12	13,5	43,5	Экзамен

4.2. Развернутое описание содержания учебного материала по разделам (темам)

Тема 1. Введение в теорию искусственного интеллекта.

Основные понятия теории искусственного интеллекта. Основные направления исследований в области искусственного интеллекта (ИИ). Задачи распознавания изображений, логического вывода, моделирования знаний, перевода, семантического анализа конструкций языка. Генетические алгоритмы. Структура генетического алгоритма. Моделирование кроссовера и мутации. Применение генетических алгоритмов.

Тема 2. Компьютерные средства разработки ИИ.

Инструментальные компьютерные средства разработки систем ИИ. Представление о логическом и функциональном программировании. Язык Лисп. Основные структуры языка Лисп — списки, атомы, типы данных. Роль программирования в развитии методов представления знаний. Понятие агента и свойства агентов. Агентные и многоагентные системы. Язык логического программирования Пролог. Диалекты языка Пролог. Структура программы, режимы работы.

Тема 3. Языки программирования ИИ.

Представление знаний о предметной области в виде фактов и правил базы знаний Пролога. Дескриптивный, процедурный и машинный смысл программы на Прологе. Создание консольных приложений в Visual Prolog. Общие понятия, связанные с логическим программированием. Сложные термы. Условные выражения. Факт-переменная. Цикл fail. Цикл foreach. Цикл repeat. Отсечение. Отладчик. Трассировка в PIE. Разработка интерфейса программы и структуры предикатов в Прологе. Рекурсия и работа со списками. Примеры разработки программ.

Тема 4. Основы теории экспертных систем. Основы теории представления знаний.

Моделирование и представление знаний. Система знаний. Модели представления знаний: логическая, сетевая, фреймовая, продукционная. Понятие нечеткой логики. Нечеткое множество, алгебра, нечеткое управление. Получение знаний и обучение. Классификация методов получения знаний. Активные и пассивные методы получения знаний. Методы инженерии знаний. Метод экспертных оценок Делфи. Метод мозгового штурма.

Тема 5. Проблематика и технологии экспертных систем.

Основы теории нейронных и случайных сетей. Нейронные сети и их моделирование. Основные модели нейронов. Виды нейронных сетей и их использование в системах ИИ. Методы обучения сетей. Байсовские сети и сети Петри. Понятие о экспертной системе (ЭС). Общая характеристика ЭС. Виды ЭС и типы решаемых задач. Структура и режимы использования ЭС. Классификация инструментальных средств ЭС и организация знаний в ЭС. Инженерия знаний. Основные задачи инженера знаний. Интеллектуальные информационные ЭС. Задача распознавания образов. Кластерный анализ данных. Основные методы и их классификация. Интеллектуальный анализ данных. Основные методы и их классификация. Применение ЗРО в ИИ.

Тема 6. Искусственный интеллект в системе образования.

Технология прокторинга и автоматизированная система идентификации личности сдающего экзамен. Автоматизированные учебные курсы как база новых технологий подготовки специалистов. Принцип "интеллектуализации объекта и средств обучения". Алгоритм стабилизации меры трудности учебных заданий. Математические модели ИОС и основные требования, предъявляемым к ним. Алгоритмы адаптивного управления процессом обучения. Характеристики и особенности обучающих адаптивных систем (ОАС): ELM-ART, CALAT, WITS и Belvedere.

5. Образовательные технологии, в том числе перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

В соответствии с требованиями ФГОС ВО в программе данного курса рекомендуется использовать следующие образовательные технологии: лекции, лабораторно-практические занятия, самостоятельная работа студентов.

В преподавании дисциплины существуют темы, изучение которых должно обязательно начинаться с теоретической подготовки. Чтобы повысить результативность изучения теоретического материала, применяются следующие приемы:

- лекции сопровождаются компьютерными презентациями;
- применяются методики «мозгового штурма», разбора практических ситуаций;
- используются краткие сообщения студентов по новейшим разработкам (по материалам Интернет) с использованием демонстрационных материалов.

Основной и самой результативной формой обучения являются лабораторно-практические занятия. Для преподавания дисциплины создан учебно-методический комплекс, обеспечивающий предоставление информации (электронные учебники и пособия) и управление деятельностью обучаемого.

Разработанные лабораторные работы включают в себя обучающие тексты, набор пошаговых инструкций, учебных задач и заданий, демонстрационный материал и тестовые задания.

Такой подход к организации проведения лабораторного практикума стимулирует и организует самостоятельную работу студента. Достигаются не только учебные, но и развивающие цели, т.к. формируется умения по самоорганизации учебной деятельности.

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине: мультимедиа технологии; технологии смешанного обучения.

6. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся по дисциплине осуществляется в следующих формах:

- решение задач, связанных с написанием программ на языке логического программирования;
- подготовка доклада по системам искусственного интеллекта с использованием информации, расположенной в литературных и Интернет-источниках;
- изучение и анализ предложенных электронных образовательных ресурсов, содержащих информацию по системам искусственного интеллекта;
- повторение материала по конспектам.

Полностью весь методический материал по обеспечению самостоятельной работы студентов приводится в приложении 1 к настоящей программе. Доступ к методическим материалам осуществляется через сайт Шуйского филиала ИвГУ, в том числе через ЭБС Шуйского филиала ИвГУ на платформе «Руконт».

7. Характеристика оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине

Текущая аттестация осуществляется с использованием накопительного рейтингового оценивания. Критерии оценки деятельности обучающихся разрабатываются в зависимости от уровня их информационно технологической и дидактической подготовки обучающихся.

Для текущего контроля успеваемости применяются:

- доклад по предложенной теме (15 баллов);
- практические задания: 8 заданий по 5 баллов (40 баллов);
- лабораторные работы: 5 работ по 2 балла и 3 работы по 5 баллов (25 баллов);

Помимо этого учитывается:

- степень участия обучающегося в коллективных обсуждениях – 10 баллов,
- посещаемость (отсутствие пропусков занятий) – 10 баллов.

Промежуточная аттестация по дисциплине (экзамен) проводится с использованием рейтинговой шкалы оценки. На экзамене по дисциплине «Основы искусственного интеллекта» контролируемые знания, умения и владения, характеризующие этапы формирования компетенций в процессе освоения образовательной программы. Экзамен проводится в форме устного собеседования по темам (10 баллов), представленным в приложении, а также выполнения практических заданий (20 баллов).

Общая накопительная оценка складывается из баллов текущей успеваемости в семестре по предмету и баллов, полученных студентами на экзамене.

Шкала оценки результатов по дисциплине «Основы искусственного интеллекта» при рейтинговой системе оценивания

Повышенный уровень	Базовый уровень	Пороговый уровень	Ниже порогового уровня
5 (отлично)	4 (хорошо)	3 (удовлетворительно)	2 (неудовлетворительно)
85-100 баллов	71-84 балла	61-70 баллов	60 и менее баллов

8. Учебно-методическое и информационное обеспечение дисциплины

Учебная и учебно-методическая литература:

- ***основная литература:***

1. Ефимова, Е.А. Основы программирования на языке Visual Prolog / Е.А. Ефимова. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 266 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428996
2. Интеллектуальные информационные системы и технологии : учебное пособие / Ю.Ю. Громов, О.Г. Иванова, В.В. Алексеев и др. ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». – Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2013. – 244 с. : ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=277713
3. Серегин, М.Ю. Интеллектуальные информационные системы : учебное пособие / М.Ю. Серегин, М.А. Ивановский, А.В. Яковлев ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». – Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2012. – 205 с. : ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=277790

- ***дополнительная литература:***

1. Калентьев, А. А. Новые технологии в программировании : учебное пособие / А. А. Калентьев, Д. В. Гарайс, А. Е. Горяинов ; Министерство образования и науки

Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : Эль Контент, 2014. - 176 с. : схем., ил. - Библиогр.: с. 166-169. - ISBN 978-5-4332-0185-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=480503

2. Зыков, С.В. Введение в теорию программирования. Объектно-ориентированный подход / С.В. Зыков. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 189 с. : схем. - (Основы информационных технологий). - Библиогр. в кн. - ISBN 5-9556-0009-4 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429073
3. Зыков, С.В. Введение в теорию программирования. Функциональный подход / С.В. Зыков. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 153 с. : ил. - (Основы информационных технологий). - Библиогр. в кн. - ISBN 5-9556-0009-4 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429119
4. Иванова, Г. С. Объектно-ориентированное программирование : учебник для вузов / Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев ; под ред. Г. С. Ивановой. - 2-е изд. ; перераб. и доп. - Москва : Изд-во МГТУ им. Н. Э. Баумана, 2003. - 368 с.
5. Мейер, Б. Объектно-ориентированное программирование и программная инженерия / Б. Мейер. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 286 с. : ил. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429034
6. Прыкина, Е.Н. Основы логического программирования в среде Турбо Пролог : учебное пособие / Е.Н. Прыкина ; Федеральное агентство по культуре и кинематографии, Кемеровский государственный университет культуры и искусств, Кафедра технологии автоматизированной обработки информации. - Кемерово : КемГУКИ, 2006. - 68 с. - ISBN 5-8154-0130-7 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=227891
7. Рублев, В.С. Языки логического программирования : учебное пособие / В.С. Рублев. - Москва : Интернет-Университет Информационных Технологий, 2008. - 115 с. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=234653
8. Сергеев, Н.Е. Системы искусственного интеллекта / Н.Е. Сергеев ; Министерство образования и науки РФ, Южный федеральный университет, Инженерно-технологическая академия. – Таганрог : Издательство Южного федерального университета, 2016. – Ч. 1. – 123 с. : схем., ил., табл. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=493307
9. Сергиевский, Г. М. Функциональное и логическое программирование : учебное пособие для студентов вузов / Г. М. Сергиевский, Н. Г. Волченков. - Москва : Академия, 2010. - 320 с.

Интернет-ресурсы:

1. Единая коллекция цифровых образовательных ресурсов http://school-collection.edu.ru/
2. Единое окно доступа к образовательным ресурсам. http://window.edu.ru/
3. Образовательные сайты и порталы. http://school2100.com/
4. Сайты учителей. http://rus.1september.ru/
5. Профессиональный сайт учителей «Методисты». http://nsportal.ru/
6. Социальная сеть работников образования. http://nsportal.ru/shkola

Современные профессиональные базы данных и информационно-справочные системы:

ЭБС «Университетская библиотека онлайн»;

Коллекция Шуйского филиала ИвГУ на национальном цифровом ресурсе «Руконт».

Программное обеспечение:

В процессе изучения дисциплины используются офисные пакеты Microsoft Office, Open Office, программа для просмотра и чтения файлов PDF Adobe Acrobat Reader DC, программа для воспроизведения флэш-анимации в браузерах Adobe Flash Player, веб-обозреватели Google Chrome, Opera, Антивирус Касперского, система программирования Visual Prolog 5.2, 1С Университет.

9. Материально-техническое обеспечение дисциплины

1. Учебная аудитория для проведения занятий лекционного типа, укомплектованная специализированной мебелью и техническими средствами обучения, служащими для предоставления учебной информации большой аудитории.
2. Учебная аудитория для проведения занятий семинарского типа, укомплектованная специализированной мебелью и техническими средствами обучения.
3. Лаборатория, оснащенная лабораторным оборудованием, укомплектованная специализированной мебелью и техническими средствами обучения.
4. Учебная аудитория для проведения групповых и индивидуальных консультаций, укомплектованная специализированной мебелью и техническими средствами обучения.
5. Учебная аудитория для проведения текущего контроля и промежуточной аттестации, укомплектованная специализированной мебелью и техническими средствами обучения.
6. Помещение для самостоятельной работы, оснащенное специализированной мебелью, компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.
7. Учебно-наглядные пособия для занятий лекционного типа, обеспечивающие тематические иллюстрации: презентации по теоретическим основам искусственного интеллекта, программированию на языке Visual Prolog, сферам применения нейронных сетей.

Автор рабочей программы дисциплины: А.В. Ермошин, доцент кафедры математики, информатики и методики обучения, кандидат технических наук, доцент.

Программа рассмотрена на заседании кафедры математики, информатики и методики обучения «15» мая 2019 г., протокол № 8

Программа обновлена
протокол заседания кафедры № 5 от «21» февраля 2020 г.
Согласовано:
Руководитель ОП
_____ С.А. Зайцева
(подпись)

Программа обновлена
протокол заседания кафедры № 10 от «07» июня 2021 г.
Согласовано:
Руководитель ОП
_____ С.А. Зайцева
(подпись)

Программа обновлена
протокол заседания кафедры № 7 от «18» февраля 2022 г.
Согласовано:
Руководитель ОП
_____ С.А. Зайцева
(подпись)

Программа обновлена
протокол заседания кафедры № 11 от «21» июня 2023 г.
Согласовано:
Руководитель ОП
_____ С.А. Зайцева
(подпись)

Приложение 1

Методические указания для обучающихся по освоению дисциплины

На самостоятельную работу студентов по курсу «Основы искусственного интеллекта» отводится более 60% учебного времени от общей трудоемкости курса. Сопровождение самостоятельной работы студентов по данному курсу может быть организовано в следующих формах:

– согласование индивидуальных заданий (виды и темы заданий, сроки представления результатов) самостоятельной работы студента в пределах часов, отведенных на самостоятельную работу;

– консультации (индивидуальные и групповые), в том числе с применением дистанционных технологий взаимодействия;

– промежуточный контроль хода выполнения заданий, строящихся на основе различных способов самостоятельной информационной деятельности в открытой информационной среде и отражающегося в процессе формирования электронного портфолио студента.

Преподаватель определяет содержание самостоятельной работы, график её выполнения; создает сетевую информационную и коммуникационную среду для организации самостоятельной работы.

В процессе организации самостоятельной работы особое внимание уделяется не только развитию информационной культуры, но и формированию готовности к кооперации, к работе в коллективе в сетевом пространстве, для чего целесообразно стимулировать студентов к совместной деятельности в малых группах.

Одной из важнейших особенностей заданий для самостоятельной работы по курсу «Основы искусственного интеллекта» является их ориентация на формирование способности оценивать приоритеты и ограничения при выборе средств информационных технологий для решения управленческих задач, а также на мотивирование самообразовательной деятельности с использованием ИКТ.

К типовым заданиям для самостоятельной работы студентов относятся:

– Конспектирование и реферирование литературных источников по тематике учебного курса.

– Решение задач, связанных с написанием программ на языке логического программирования;

– Изучение и анализ предложенных электронных образовательных ресурсов, содержащих информацию по системам искусственного интеллекта;

– Подготовка доклада по системам искусственного интеллекта с использованием информации, расположенной в литературных и Интернет-источниках;

– Повторение материала с использованием конспектов.

Задания для организации практических занятий:

Задание 1. Язык программирования Lisp: базовые функции, работа с числами

1.1. Получить все делители натурального числа N.
1.2. Подсчитать количество и сумму цифр целого числа N.
1.3. Найти наибольший общий делитель чисел из заданного списка.
1.4. Из простого списка удалить все нулевые элементы.

Критерии оценки:
Всего баллов – 5 (Максимально)
По 1 баллу за каждое правильное самостоятельно решённое задание.
Составление отчёта по проделанной работе – 1 балл.

Задание 2. Язык программирования Lisp: работа со списками

2.1. Из произвольного списка удалить все отрицательные элементы.
2.2. Из произвольного списка удалить все латинские буквы, расположенные между буквами d и k.
2.3. Найти произведение всех ненулевых чисел из заданного сложного списка.
2.4. Разработать программу преобразования математических выражений в префиксную форму.

Критерии оценки:
Всего баллов – 5 (Максимально)
По 1 баллу за каждое правильное самостоятельно решённое задание.
Составление отчёта по проделанной работе – 1 балл.

Задание 3. Определение отношений в программе Visual Prolog

3.1. Определите в программе о родственных отношениях следующие бинарные отношения:

- внук, внучка, дедушка и бабушка;
- ребенок как обратное к отношению родитель;
- сын, дочь, сестра и брат;
- супруги как симметричное замыкание отношения супруг.

3.2. Напишите программу, которая с помощью бинарных отношений владелец, животное и цвет, хранящих пары хозяин – кличка, кличка – животное и кличка – окрас, соответственно, описывает следующие сведения. Майкл владеет рыжим котом, Сьюзен шоколадным, Дэн и Пит владеют серыми котами. Билл имеет собаку серого окраса, а Бетти шоколадного. Все животные имеют уникальные клички (придумайте их самостоятельно). Составьте запросы к программе:

- найти владельцев серых котов;
- найти животного Билла и цвет этого животного;
- найти животных, которыми владеют Бетти и Майкл;
- найти владельцев животных шоколадного окраса.

3.3. Найдите декларативное значение программы "Птицы".

```
летает("синица").              % факты
летает("лебедь").
летает("аэроплан").

животное("лебедь").
животное("синица").
животное("тигр").

птица("пингвин").
птица("страус").
```

```
птица(X):- животное(X), летает(X).      % правило

?- птица(A).              % запрос
```

3.4. Постройте для логической программы

```
add(o, X, X).
add(s(X), Y, s(Z)):- add(X, Y, Z).
```

- эрбранов универсум;
- эрбранов базис;
- минимальную модель.

Критерии оценки:
Всего баллов – 5 (Максимально)
По 1 баллу за каждое правильное самостоятельно решённое задание.
Составление отчёта по проделанной работе – 1 балл.

Задание 4. Машина вывода Пролога

4.1. Найдите с помощью программы "Библиотека" ответы на запросы:

- Какие журналы за позапрошлый год имеются в библиотеке? (Текущий год определяется с помощью предиката getDate/3).
- Найдите самую старую по году издания литературу.

Листинг программы «Библиотека»:

```
        publication = book(author, string Название, edition
Издание);
            magazine(string Название, integer Номер, integer Год).
        author = author(string Фамилия, string Имя, string
Отчество).
        edition = edition(string Место, string Издательство,
integer Год).


    class facts
        library: (publication).
    clauses
        library(magazine("Компьютерра", 2, 2009)).
        library(magazine("Наука и жизнь", 11, 2012)).
        library(book(author("Чехов", "Антон", "Павлович"),
            "Избранное", edition("Москва", "АСТ, Астрель", 2003))).
        library(book(author("Великова", "Людмила", "Викторовна"),
            "Русский язык", edition("Москва", "МЦНМО", 2003))).
```

```
run():-
    % Что есть в библиотеке?
    library(X),
        write(X), nl,
    fail;
    % Названия книг, изданных в 2003 году
    library(book(_, Title, edition(_, _, 2003))),
        write(Title), nl,
    fail;
    _ = readLine().
```

4.2. Найдите ответ на вопросы с помощью программы "Иностранные языки":

- кто владеет только английским и немецким языками;
- кто владеет ровно одним иностранным языком?

Дополните программу новыми фактами.

Листинг программы «Иностранные языки»:

```
class facts
    knows: (string, string*).
clauses
    knows("Даша", ["английский", "испанский", "французский"]).
    knows("Маша", ["немецкий", "английский"]).
    knows("Глаша", ["английский", "немецкий"]).
    knows("Паша", ["английский"]).

    run():-
        knows(X, Y),
            write(X, " - ", Y), nl,
        fail;
        _ = readLine().
```

4.3. Напишите программу, которая выводит названия месяцев

- с начала года, предшествующие заданному месяцу;
- до конца года, следующие за данным месяцем.

4.4. Постройте дерево SLD-резолютивных вычислений для запроса ?- родитель(X, Y), родитель(Y, Z) к программе ниже.

```
родитель("Иван", "Мария").
родитель("Анна", "Мария").
родитель("Мария", "Павел").
родитель("Мария", "Петр").

женщина("Мария").
женщина("Анна").

мать(X, Y):- женщина(X), родитель(X, Y).
```

Критерии оценки:
Всего баллов – 5 (Максимально)
По 1 баллу за каждое правильное самостоятельно решённое задание.
Составление отчёта по проделанной работе – 1 балл.

Задание 5. Управление перебором. Отсечение в Visual Prolog

5.1. Как-то раз сестры Маша, Даша и Глаша испекли пирог. Одна из них месила тесто, другая готовила начинку, а третья выпекала пирог в духовке. Известно, что каждое из следующих высказываний истинно:

- если Глаша месила тесто, то Даша готовила начинку;
- если Маша выпекала пирог, то месила тесто Даша;
- если Глаша готовила начинку, то Маша выпекала пирог;
- если Даша месила тесто, то Маша готовила начинку;
- если Глаша выпекала пирог, то Маша месила тесто.

Кто из сестер месил тесто, кто готовил начинку, а кто выпекал пирог?
5.2. Жители острова A, B и C, один из которых всегда говорил правду, другой всегда лгал, а третий был хитрецом — иногда говорил правду, а иногда лгал, сообщили о себе следующее:
A: "Я хитрец";
B: "Да, A хитрец";
C: "Я не хитрец".
Определите, кто из них кем был на самом деле.
5.3. Определите предикаты, результатом действия которых является упорядочивание пары чисел или тройки чисел:

```
упорядочение(X, Y, Min, Max);
упорядочение(X, Y, Z, Min, Ave, Max).
```

5.4. Сгенерируйте с помощью конструкции `[…||…]` список всех белых и список всех черных клеток шахматной доски (индексы полей принимают значения от 1 до 8).

Критерии оценки:
Всего баллов – 5 (Максимально)
По 1 баллу за каждое правильное самостоятельно решённое задание.
Составление отчёта по проделанной работе – 1 балл.

Задание 6. Внутренняя база данных в Visual Prolog

6.1. Решите ребус: УФ ○ ФУ = УШУ. В этой записи разным буквам соответствуют разные цифры, причем первая цифра в записи чисел отлична от нуля.
6.2. Создайте базу данных, в которой для каждого сотрудника хранятся фамилия, должность и оклад. Определите средний размер оклада у программистов.
6.3. Найдите количество строк в файле.
6.4. Напишите программу, которая генерирует случайным образом 15 попарно различных точек плоскости с целыми координатами в пределах от 0 до 79 по оси абсцисс и от 0 до 22 по оси ординат. Найдите все тройки точек, которые образуют треугольники максимальной площади. Отобразите точки в окне консоли. Вершины найденных треугольников выделите цветом.

Критерии оценки:
Всего баллов – 5 (Максимально)
По 1 баллу за каждое правильное самостоятельно решённое задание.
Составление отчёта по проделанной работе – 1 балл.

Задание 7. Рекурсия, списки и полиморфизм в Visual Prolog. Часть 1.

7.1. Определите на множестве кубиков отношение "не ниже", которое является рефлексивно-транзитивным замыканием отношения наверху. Пара кубиков принадлежит отношению наверху, если первый из них стоит на втором.
7.2. Сгенерируйте подмножество целых чисел от числа m до числа n включительно с шагом s по возрастанию;
7.3. Сгенерируйте подмножество целых чисел от числа m до числа n включительно с шагом s по убыванию.
7.4. По заданному натуральному числу n найдите два соседних элемента последовательности Фибоначчи с номерами n и n + 1.

Критерии оценки:
Всего баллов – 5 (Максимально)
По 1 баллу за каждое правильное самостоятельно решённое задание.
Составление отчёта по проделанной работе – 1 балл.

Задание 8. Рекурсия, списки и полиморфизм в Visual Prolog. Часть 1.

8.1. Вычислите значение n–й частичной суммы разложения функции y=sin x в ряд по степеням x в заданной точке.
8.2. Вычислите с заданной погрешностью значение функции y=sin x в заданной точке

с помощью разложения в ряд по степеням x.

8.3. По двум точкам, заданным координатами в консоли, постройте соединяющий их "отрезок" в виде набора точек. Отобразите "отрезок" в окне консоли.

8.4. Для представления неотрицательных целых чисел в виде термов o, s(o), s(s(o)), s(s(s(o))), …, определите операции сложения и умножения, а также функцию вычисления факториала.

Критерии оценки:
Всего баллов – 5 (Максимально)
По 1 баллу за каждое правильное самостоятельно решённое задание.
Составление отчёта по проделанной работе – 1 балл.

Задания для организации лабораторных занятий:

Лабораторная работа 1. Система программирования muLisp

1. Цель работы: приобретение практических навыков работы в системе программирования muLisp.

2. Краткие справочные данные.

2.1. Запуск интерпретатора muLisp.

Запуск интерпретатора осуществляется путем инициализации файла mulisp.com (в нашем случае через bat-файл lisp.bat).

После выдачи начальных сообщений система высвечивает приглашение (знак доллара $), которое означает, что muLisp готов к вводу с клавиатуры.

Далее пользователь набирает некоторое выражение и нажимает клавишу Enter. Интерпретатор оценивает это выражение и печатает результирующее значение в начале новой строки, затем снова высвечивается приглашение $ и т.д. Этот цикл взаимодействия интерпретатора с пользователем повторяется до тех пор, пока не будет введена системная команда (SYSTEM), которая завершает работу и возвращает управление операционной системе.

2.2. Механизм работы Лисп-системы.

Так как Лисп является функциональным языком, то любая программа на Лиспе – тоже функция. Программа на Лиспе записывается в виде s-выражения (символьного выражения) как обращение к функции.

В общем случае обращение к функции имеет следующий вид:

```
(F Arg1 Arg2 … ArgN).
```

Здесь F – первый элемент списка, интерпретируется как имя функции, которую необходимо выполнить, взяв в качестве аргументов оставшиеся элементы списка Arg1, Arg2, …, ArgN.

Механизм работы Лисп-системы состоит из трех последовательных шагов (read-eval-print):

- считывание s-выражений (READ);
- интерпретация s-выражений (EVAL);
- печать s-выражения (PRINT).

Интерпретация s-выражений – это единственная и главная задача Лисп-интерпретатора. Ее выполняет функция Лисп-системы EVAL, которая берет одно s-выражение, интерпретирует его, если это возможно, и возвращает другое s-выражение как

результат интерпретации первого s-выражения.

2.3. Прерывания при работе интерпретатора muLisp.

При работе интерпретатора могут возникнуть внутренние и внешние прерывания (остановы). В этом случае на экране дисплея высвечивается подсказка в виде опций:

```
Continue, Break, Abort, Top-level, Restart, System?
```

Затем система ждет, пока пользователь выберет одну из опций по ее имени (C, B, A, T, R или S соответственно).

Действия опций:

Continue (продолжить): возвращает управление программе, которая вызвала прерывание.

Break (останов): временно приостанавливает выполнение программы и выходит на следующий нижний уровень цикла read-eval-print (чтение – вычисление – печать). Выход из останова – команда (RETURN).

Abort (прерывание): прерывает выполнение программы, присваивает формальным параметрам, размещенным в стеке переменных, первоначальные значения и возвращает управление на текущий уровень цикла read-eval-print.

Top-level (верхний уровень): прерывает выполнение программы, присваивает первоначальные значения формальным параметрам, которые размещены в стеке переменных, выводит текущие входные и выходные данные и возвращает управление верхнему уровню цикла read-eval-print.

Restart (повторный старт): перезагрузка системы.

System (система): выход из системы Лисп.

В любое время в ходе выполнения программы пользователь может прервать ее работу, нажав клавишу Esc на клавиатуре.

2.4. Редактор muLisp.

2.4.1. Загрузка редактора и опции главного меню.

Загрузка редактора в существующую среду muLisp выполняется командой:

```
(RDS Edit), а затем (RETURN).
```

Как только редактор начинает работать, он чистит экран, рисует рамку и выдает головное меню: Edit, Print, Screen, Lisp, Quit:.

Затем система ждет, пока пользователь не наберет одну из опций путем ввода первой буквы ее имени (E, P, S, L или Q).

- Edit: выдает на экран подсказку Edit file: и ждет, пока не будет введено имя файла для редактирования. После ввода имени файла (напр., D:\107215\aaa) нажимается Enter. По умолчанию используется расширение .lsp и текущий каталог.
- Print: выдает на экран подсказку Print file: и ждет, пока будет введено имя файла для печати.
- Screen: выдает на экран подсказку Full, Vertical, Horizontal: и ждет выбора одной из опций (F, V или H). Опция Full screen использует для редактирования полный экран. Опция Vertical split screen использует правую половину экрана для редактирования, а левую – для отладки. Опция Horizontal split screen использует верхнюю половину экрана для редактирования, а нижнюю – для отладки.
- Lisp: завершает работу редактора; на экране появляется знак доллара. Из среды muLisp можно вернуться в редактор по команде (RETURN).
- Quit: завершает работу редактора и системы muLisp.

2.4.2. Команды управления курсором.

Основные команды перемещения курсора:

Ctrl-Q, D – в правый конец текущей строки,

Ctrl-Q, S – в левый конец текущей строки,
Ctrl-Q, X – в нижний конец экрана,
Ctrl-Q, E – в верхний конец экрана,
Ctrl-Q, I – влево в предыдущую позицию табуляции.

2.4.3. Команды скроллинга экрана.

Ctrl-Z – сдвигает текст вверх на одну строку,
Ctrl-W – сдвигает текст вниз на одну строку,
Ctrl-R – сдвигает текст на высоту экрана вниз,
Ctrl-C - сдвигает текст на высоту экрана вверх,
Ctrl-Q, C – перемещает курсор и окно в конец файла,
Ctrl-Q, R - перемещает курсор и окно в начало файла.

2.4.4. Изменение режима ввода текста.

Для переключения из режима вставки (Insert) в режим замещения (Replace) и наоборот используется команда Ctrl-V.

2.4.5. Команды удаления и восстановления текста.

Ctrl-Y – удаляет строку, на которой располагается курсор,
Ctrl-Q, Y – удаляет правый конец строки, начиная с курсора,
Ctrl-U – восстанавливает текст, уничтоженный последней командой удаления.

2.4.6. Команды блочных операций.

Ctrl-O, B – отмечает начало блока,
Ctrl-O, E – отмечает конец блока,
Ctrl-O, C – копирует отмеченный блок в место, указанное курсором,
Ctrl-O, M – перемещает отмеченный блок в положение курсора,
Ctrl-O, R – читает текст из указанного файла и вставляет в место, указанное курсором,
Ctrl-O, W – запись отмеченного блока в указанный файл.

2.4.7. Команды списковых структур.

Команды списковых структур позволяют перемещать, копировать, уничтожать, считывать из файла и записывать в файл s-выражения. Двухсимвольные команды, начинающиеся с Esc, инициируют команды списковых структур.

Esc-D – перемещает курсор вперед на одно s-выражение,
Esc-S – перемещает курсор назад на одно s-выражение,
Esc-F – перемещает курсор вперед, устанавливая его после конца списка,
Esc-A - перемещает курсор назад, устанавливая его перед началом списка,
Esc-T – уничтожает s-выражение под курсором и справа от курсора,
Esc-Y – уничтожает определение, где располагается курсор,
Esc-L – закрывает редактор и передает управление в окно вычисления Lisp. Возврат в редактор – команда (RETURN).

Esc-! – выдает в окно вычисления Lisp результат вычисления s-выражения под курсором и справа от него. Данная команда дает возможность выборочно переопределять определения редактируемых функций; это плезно при их тестировании.

2.4.8. Команды сохранения файлов (начинаются с Ctrl-K).

Через несколько секунд после ввода Ctrl-K выдается подсказка, если пользователь не успел ввести второй символ команды.

Ctrl-K, D – сохраняет текст в виде файла и возвращает управление в головное меню редактора,

Ctrl-K, S – сохраняет текст в виде файла и заново открывает файл для редактирования.

Если в текст вносились изменения, Ctrl-K, A и Ctrl-K, Q высвечивают подсказку:

```
Abandon <filename>? (Y/N),
```

где <filename> - имя отредактированного файла. При вводе Y отредактированный текст удаляется, а управление возвращается в головное меню редактора. В противном случае никаких действий не происходит. Если изменения не вносились, текст просто удаляется.

2.4.9. Команды окна.

Ctrl-Q, W – позволяет изменить размер окна и его форму на время редактирования файла. Команда выдает подсказку

Full, Vertical, Horizontal

и ждет выбора одной из опций (F, V или H) (см. пункт 2.4.1).

2.5. Трассировка программы.

Чтобы выполнить трассировку программы, нужно загрузить отладочный пакет muLisp'a с помощью команды

(RDS 'DEBUG)

Трассировка функций осуществляется с помощью функции TRACE. Она вызывается с одной или более функциями для трассировки в качестве своих аргументов. Например, команда

(TRACE 'APP 'REV)

трассирует функции APP и REV.

Если после этого вызвать функцию APP (объединение двух списков) следующим образом

(APP '(A B) '(C D E)),

то будет выведена трасса для этой функции.

Если вывод трассы осуществляется слишком быстро, то для временной остановки вывода можно набрать на клавиатуре Ctrl-S.

Трасса также может быть выведена командой (HISTORY).

Для запрещения трассировки используется команда CLEAR. Например, команда (CLEAR 'APP 'REV) запрещает трассировку функций APP и REV.

2.6. Примитивы Лиспа:

- функция (**car L**) – возвращает голову списка L, напр., (car '((a b) c d)) возвращает (A B);
- функция (**cdr L**) – возвращает хвост списка L, напр., (cdr '((a b) c d)) возвращает (C D);
- функция (**cons S1 S2**) – включает элемент S1 в начало списка S2, напр., (cons 'a '(b c)) возвращает (A B C);
- предикат (**atom S**) – проверяет, является ли аргумент S атомом, напр., (atom '()) возвратит T, а (atom '(nil)) возвратит NIL;
- предикат (**eq S1 S2**) – проверяет тождественность двух символов S1 и S2, напр., (eq 'a (car '(a b c))) возвратит T, а (eq 'a '(a)) возвратит NIL;
- функция (**quote S**) – блокирует вычисление выражения S, значением функции является символьное выражение S, представленное в вызове, напр., (quote (+ 2 3)) возвращает (+ 2 3). Синонимом для (quote (+ 2 3)) является запись '(+ 2 3).

3. Содержание задания по лабораторной работе.

3.1. Загрузить muLisp.

3.2. Ознакомиться с возможностями редактора muLisp.

3.3. Проделать следующие вычисления с помощью интерпретатора Лиспа:

a) 3.234*(4.56+21.45)
b) 5.67+6.342+12.97
c) (454-214-657)/(456+678*3)

3.4. Написать выражение, вычисляющее среднее арифметическое чисел 23, 5, 43 и 17.

3.5.Определить значение следующих выражений:

a) ‘(+ 2 (* 3 4))
b) (+ 2 ‘(* 3 4))
c) (+ 2 (‘* 3 4))
d) (+ 2 (* 3 ‘4))
e) (quote ‘quote)
f) (quote 2)
g) ‘(quote nil)

3.6.Записать последовательность вызовов CAR и CDR, выделяющие из приведенных ниже списков символ «цель». Упростить эти вызовы с помощью функций C…R.

a) (1 2 цель 3 4)
b) ((1) (2 цель) (3 (4)))
c) ((1 (2 (3 4 цель))))

3.7.Вычислить значения следующих выражений. Проверить результат на компьютере.

a) (cons nil ‘(суть пустой список))
b) (cons nil nil)
c) (cons ‘(nil) ‘(nil))
d) (cons (car ‘(a b)) (cdr ‘(a b)))
e) (car ‘(car (a b c)))
f) (cdr (car (cdr ‘(a b c))))

3.8.Проверить, какие из следующих вызовов возвращают значение T?

a) (atom ‘(cdr nil))
b) (atom (* 2 (+ 2 3)))
c) (atom (atom (+ 2 3)))
d) (eq nil ( ))
e) (eq t (atom (car ‘(мышь))))

3.9.Выполнить трассировку функции APP, объединяющей два списка.

Определение функции:

```
( defun app ( lst1 lst2 )
  (( null lst1 ) lst2 )
  ( cons ( car lst1 )
  ( app ( cdr lst1 ) lst2 )))
```

4. Содержание отчета: цель работы, постановка задачи, диалог пользователя с интерпретатором muLisp, выводы. (Для формирования отчета использовать редактор muLisp).

Контрольные вопросы:

1. Инфиксная и префиксная формы записи выражений; какая из этих форм принята в Lisp ?

2. Для чего используется функция QUOTE, чем ее можно заменить ?
3. Что означает символ $ в начале строки при работе с интерпретатором muLisp ?
4. Как производится выход из системы muLisp ?
5. Как будет интерпретироваться системой muLisp s-выражение (a b c d e) ?
6. Как загружается редактор в среду muLisp ?
7. Как можно запустить на выполнение функцию, набранную в редакторе muLisp?
8. По какой команде можно перейти из редактора в окно вычислений muLisp ?
9. По какой команде можно вернуться из окна вычислений в редактор ?
10. Как можно сохранить отредактированный в редакторе текст в виде файла ?
11. Как производится трассировка программы в среде muLisp ?

Критерии оценки:
Всего баллов – 5 (Максимально)
5 баллов выставляется все шаги выполнены полностью правильно и самостоятельно, а студент способен уверенно отвечать на вопросы преподавателя о проделанной работе;
3-4 балла выставляется, если работа выполнена в целом корректна, но имеются неточности в ответах на вопросе преподавателя или некоторые действия выполнены неоптимально;
1-2 балла выставляется, когда работа не может быть завершена студентом самостоятельно, без помощи преподавателя.
0 баллов выставляется, если работа не была выполнена.

Лабораторная работа 2. Основы работы с Visual Prolog в режиме Test Goal

1. Цель работы: ознакомление с режимом Test Goal системы программирования Visual Prolog.

2. Краткие справочные данные. При использовании утилиты Test Goal требуется определить некоторые (не предопределенные) опции компилятора Visual Prolog. Для этого необходимо выполнить следующие действия:

2.1. Запустить среду визуальной разработки Visual Prolog двойным щелчком мыши по пиктограмме.

2.2. Создать новый проект (выбрать команду Project – New Project, активизируется диалоговое окно Application Expert).

2.3.Определить имя проекта и рабочий каталог.

Имя проекта набирается в поле Project Name. После щелчка в поле Name of .VPR File появляется имя проекта с расширением .vpr.

На вкладке Target выбрать параметры: Windows32 в списке Platform, Easywin – в списке UI Strategy, exe – в списке Target Type, Prolog – в списке Main Program.

В поле Base Directory на вкладке General указать директорий для проекта (можно создать новый директорий, используя кнопку Browse…).

Нажать кнопку Create для того, чтобы создать файлы проекта по умолчанию.

2.4. Открыть окно редактора (использовать команду меню File –New). Появится новое окно редактирования с именем noname.pro. Редактор среды визуальной разработки – стандартный текстовый редактор. Можно использовать клавиши управления курсором и мышь так же, как и в других редакторах. Он поддерживает команды Cut, Copy, Delete, Undo и Redo, которые находятся в меню Edit. В меню Edit также показаны комбинации «горячих» клавиш для этих действий.

2.5. Набрать текст программы (см. в п.4).

2.6. Выполнить программу (команда Project – Test Goal или комбинация

клавиш Ctrl-G).

3. **Обработка ошибок**. При наличии ошибок в программе отобразится окно Errors (Warnings), которое будет содержать список обнаруженных ошибок. Дважды щелкнув на одной из этих ошибок, попадаем на место ошибки в исходном тексте. Можно воспользоваться клавишей F1 для вывода на экран интерактивной справочной системы Visual Prolog. Когда окно помощи откроется, следует щелкнуть по кнопке Search, набрать номер ошибки, и на экране появится соответствующее окно помощи с более полной информацией об ошибке.

4**. Проверка правильности настройки системы**. Для проверки того, что система настроена должным образом, можно набрать в окне редактора, например, следующий текст:

```
GOAL write ("Добро пожаловать"),nl.
```

Этого достаточно для программы, чтобы она могла быть выполнена. Результат выполнения программы

```
Добро пожаловать
yes
```

будет расположен в отдельном окне, которое необходимо закрыть перед тем, как выполнять другую программу.

5. **Установка шрифта**. Если имеются проблемы с использованием в программе кириллицы, следует щелкнуть правой клавишей мыши в любом месте поля редактора, в появившемся контекстном меню выбрать команду Font…, затем в открывшемся диалоговом окне выбрать шрифт System или Tahoma.

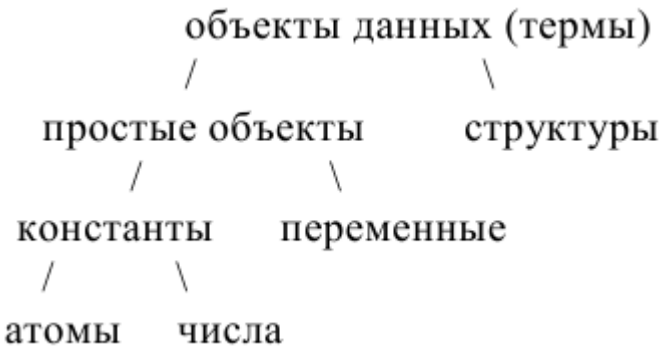
Критерии оценки:
Всего баллов – 2 (Максимально)
2 балла выставляется все шаги выполнены полностью правильно и самостоятельно, а студент способен уверенно отвечать на вопросы преподавателя о проделанной работе;
1 балл выставляется, если работа выполнена в целом корректна, но имеются неточности в ответах на вопросе преподавателя или некоторые действия выполнены неоптимально;
0,5 балл выставляется, когда работа не может быть завершена студентом самостоятельно, без помощи преподавателя.
0 баллов выставляется, если работа не была выполнена.

Лабораторная работа 3. Составление простейшей Пролог-программы

1. Цель работы: приобретение практических навыков составления, отладки и выполнения простейшей программы в системе программирования Visual Prolog (режим Test Goal).

2. Краткие справочные сведения.

Объекты данных в Прологе называются термами. Терм может быть: константой, переменной, структурой (составной терм). Классификация объектов данных приведена на схеме:



Атом - это синтаксически неделимый терм. В качестве атома могут быть: имя Visual Prolog (имя – это начинающаяся с буквы последовательность букв, цифр и символа подчеркивания), строчное представление числа (например, "1048", "25.6"), отдельный символ (кроме пробела).

Числа в Visual Prolog записываются точно так же, как и в других языках программирования.

Константы относятся к одному из стандартных доменов (типов), приведенных в таблице (кроме приведенных в таблице Visual Prolog имеет и другие стандартные домены: byte, short, ushort, word, unsigned, long, ulong, dword).

Тип	Ключевое слово	Диапазон значений	Примеры
символы	char	всевозможные одиночные символы	'a', 'B', '?'
целые числа	integer	-2147483648 … 2147483647	-15, 1235, 9
действительные числа	real	1E-307 … 1E308	4.8, 2.45E-8
строки	string	последовательность символов (до 250)	"Минск", "a_min"
символические имена	symbol	1. Последовательность букв, цифр, знака подчеркивания (первый символ – строчная буква)	read_data1
		2. Последовательность заключенных в кавычки символов	"Delete f1", "Турбо-Си"
файлы	file	допустимое в Windows имя файла	a.txt progr.pro

Переменная - имя, начинающееся с большой (прописной) буквы или знака подчеркивания. Когда значение переменной неважно, то в качестве имени переменной используется знак подчеркивания. Такая переменная называется анонимной.

Структуры (сложные термы) - это объекты, которые состоят из нескольких компонент. Структура записывается с помощью указания ее **функтора** и **компонент**. Компоненты заключаются в круглые скобки и разделяются запятыми. Число компонент в структуре называется **арностью** структуры.

Пример структуры: data_r(12,mart,1962). Здесь data_r – функтор, 12, mart, 1962 – компоненты. Арность приведенной структуры равна трем.

Программа на Visual Prolog состоит из нескольких разделов, каждому из которых предшествует ключевое слово. Типичная простейшая структура программы

представлена ниже:

```
/*      комментарии        */
domains
    <описание доменов (типов данных)>
predicates
    <описание предикатов>
clauses
    <предложения или утверждения>
goal
    <целевое утверждение>
```

В разделе clauses размещаются предложения (утверждения). Предложение представляет собой факт или правило, соответствующее одному из объявленных предикатов.

Факт - простейший вид утверждения, которое устанавливает отношение между объектами. Пример факта:

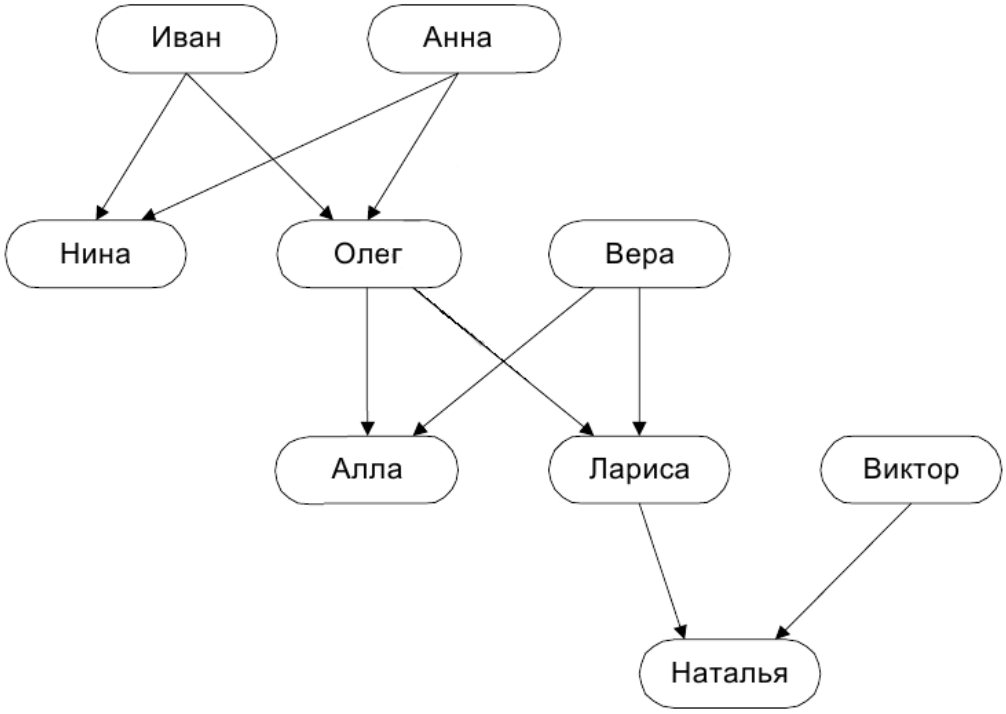
```
anketa("Иванов",5,august,1950).
```

Этот факт содержит атом anketa, который является именем предиката, и в скобках после него - список термов, соответствующих компонентам этого предиката. **Факт всегда заканчивается точкой**. Факты содержат утверждения, которые всегда являются безусловно верными. Компонентами факта могут быть только константы.

В разделе domains объявляются любые нестандартные домены, используемые для аргументов предикатов. Домены на Прологе являются аналогами типов в других языках.

В разделе goal записывается третий тип предложения – **вопрос (цель)**, состоящий из одного или нескольких целевых утверждений (подцелей), разделенных запятыми и оканчивающихся точкой. Пролог-система рассматривает вопрос как цель, к достижению которой нужно стремиться. При запуске программы Visual Prolog вызывает goal и пытается разрешить все подцели вопроса. Если все подцели в разделе goal истинны, то программа завершается успешно и будут выведены все возможные решения, а также значения всех переменных, используемых в goal. Если же какая-то подцель ложна, то считается, что программа завершается неуспешно.

3. Содержание задания по лабораторной работе.



Описать средствами Visual Prolog (с помощью фактов) дерево родственных отношений (рис. 1), используя предикат roditel с двумя параметрами: имя родителя и имя ребенка.

В окне диалога сформировать следующие вопросы:

1. Является ли Иван родителем Нины?
2. Является ли Иван родителем Аллы?
3. Кто родители Ларисы?
4. Как зовут детей Олега?
5. Кто родитель родителя Натальи?
6. Кто чей родитель?
7. Есть ли у Нины и Олега общий родитель?
8. Как зовут жену Ивана?
9. Кто у Анны внуки?
10. Есть ли у Ларисы брат или сестра?

4. Содержание отчета.

Отчет по лабораторной работе должен содержать: цель работы, постановка задачи, текст Пролог-программы, вопросы к программе и ответы системы, выводы.

Контрольные вопросы:

1. Где был разработан Пролог?
2. К какой группе языков относится Пролог?
3. Что является теоретической основой Пролога?
4. Что может быть в качестве атома?
5. Как записываются в Visual Prolog числа?
6. Чем отличаются записи констант и переменных?
7. Что такое анонимная переменная, когда она используется?
8. Форма записи структур (сложных термов).

9. Может ли быть в программе Visual Prolog несколько одинаковых разделов?
10. С какого раздела начинается выполнение программы Visual Prolog?
11. В каком разделе программы описываются пользовательские (нестандартные) типы данных?
12. Что такое предикат и в каких разделах программы его имя может использоваться?
13. Что является аналогом предиката в процедурных языках?
14. Какие объекты могут быть в качестве компонент факта?
15. Какой тип предложения записывается в разделе goal?
16. Способы записи комментариев в программе Visual Prolog.

Критерии оценки:
Всего баллов – 2 (Максимально)
2 балла выставляется все шаги выполнены полностью правильно и самостоятельно, а студент способен уверенно отвечать на вопросы преподавателя о проделанной работе;
1 балл выставляется, если работа выполнена в целом корректна, но имеются неточности в ответах на вопросе преподавателя или некоторые действия выполнены неоптимально;
0,5 балл выставляется, когда работа не может быть завершена студентом самостоятельно, без помощи преподавателя.
0 баллов выставляется, если работа не была выполнена.

Лабораторная работа 4. Формирование правил

1. **Цель работы**: приобретение практических навыков формирования правил и использования их в Пролог-программах.

2. **Краткие справочные данные**.

Правило - это утверждение, истинность которого зависит от некоторых условий. Правило состоит из **заголовка** и **тела**, соединенных символом :-, который читается “если”. Правило, как и факт, заканчивается точкой.

Заголовок является одним из ранее описанных предикатов, в котором в качестве компонентов могут быть переменные. Заголовок правила описывает факт, для определения которого предназначено это правило.

Тело правила описывает подцели, которые должны быть последовательно согласованы с фактами, для того чтобы заголовок правила был истинным. Тело содержит список термов, разделенных запятыми или точкой с запятой (;). При этом:

вместо :- можно использовать if;
вместо , может использоваться and;
вместо ; может использоваться or.

Пример правила:

```
внук(X,Y):- отец(Y,Z),отец(Z,X),мужчина(X).
```

Это правило можно прочитать следующим образом: X является внуком Y, если Y является отцом Z и этот Z является отцом X и X - мужского пола.

Важно отметить, что переменная обозначает один и тот же объект только в пределах одного предложения.

3. Содержание задания по лабораторной работе.

Программу предыдущей лабораторной работы дополнить новыми фактами, позволяющими построить правила для определения следующих целей: отец, мать, сын, дочь, брат, сестра, дядя, тетя, дедушка, бабушка, внук, внучка. Предикат для каждого правила содержит две компоненты, смысл которых аналогичен смыслу компонент правила

из примера п.2.

4. Содержание отчета.

В отчете по лабораторной работе должны быть отражены следующие пункты: цель работы, постановка задачи, текст Пролог-программы, вопросы к программе и ответы системы, выводы.

Контрольные вопросы:

1. В каком разделе программы реализуются правила?
2. Обязательно ли описание предиката, реализованного в программе, и где это описание производится?
3. В каких разделах программы могут быть переменные?
4. Область действия переменной в программе Visual Prolog.
5. Какой символ должен быть в качестве первого в имени переменной?
6. С какого символа может начинаться имя предиката?
7. Что означают символы ":-", "," и ";" в программе Visual Prolog?
8. Может ли имя константы в программе Visual Prolog начинаться с прописной буквы?
10. Различия между свободной и связанной переменными.
11. Можно ли изменить значение связанной переменной?
12. Есть ли в Прологе оператор присваивания?
13. Можно ли между предложениями с одним и тем же предикатом вставить новое предложение с другим предикатом?

Критерии оценки:
Всего баллов – 2 (Максимально)
2 балла выставляется все шаги выполнены полностью правильно и самостоятельно, а студент способен уверенно отвечать на вопросы преподавателя о проделанной работе;
1 балл выставляется, если работа выполнена в целом корректна, но имеются неточности в ответах на вопросе преподавателя или некоторые действия выполнены неоптимально;
0,5 балл выставляется, когда работа не может быть завершена студентом самостоятельно, без помощи преподавателя.
0 баллов выставляется, если работа не была выполнена.

Лабораторная работа 5. Рекурсия

1. Цель работы: приобретение практических навыков составления и отладки программ с использованием рекурсии.

2. Краткие справочные данные.

Мощным средством программирования в Прологе является рекурсия. **Рекурсия** - это определение некоторого отношения через самого себя. Так как в Прологе отсутствуют операторы цикла, то рекурсия служит основным средством программирования циклических процессов.

Классическим примером рекурсии является вычисление факториала F=N! = 1*2*3*...*(N-1)*N. Это выражение можно записать так: N!=(N-1)!*N. Последнее выражение - пример рекурсивных вычислений, т.е. каждый последующий результат получается путем использования предыдущего результата. Для того, чтобы рекурсивный метод решения был результативным, он должен в конце концов привести к задаче, решаемой непосредственно. Решить ее позволяют утверждения, называемые граничными условиями (для нашего случая граничным условием является 0!=1). На Прологе процедура вычисления факториала может быть описана следующим образом:

```
/*  Граничное условие  */
factorial(0,1).   % 0!=1
/*  Рекурсивное правило вычисления F=N!  */
factorial(N,F):-                              % F=N!
                        N>0,N1=N-1,
                        factorial(N1,F1),     % F1=(N-1)!
                        F=F1*N.               % N!=(N-1)!*N
```

При использовании стандартных предикатов ввода-вывода, позволяющих запрограммировать удобный интерфейс пользователя с программой, программа вычисления факториала на Visual Prolog может быть записана в следующем виде:

```
/*  Вычисление факториала F=N!  */
domains
  n=integer
  f=real
predicates
  factorial(n,f)
  result
clauses
  factorial(0,1).
  factorial(N,F):-N>0,N1=N-1, factorial(N1,F1),F=F1*N.
  result:-
    write("Введите число N"),nl,
    write("N="),readint(N),
    factorial(N,F),
    write(N,"!=",F),nl.
goal
  result.
```

В приведенной процедуре вычисления факториала тело правила начинается с рекурсивного вызова определяемого предиката. Такая рекурсия называется **левосторонней (нисходящей)**.

Следующая программа демонстрирует другой вариант определения факториала. Здесь рекурсивный вызов заменен на **итеративный (восходящая рекурсия)**. Итеративной процедурой называется такая процедура, которая вызывает себя только один раз, причем этот вызов расположен в самом конце процедуры. Это так называемая **правосторонняя рекурсия**.

```
/* Программа итеративного вычисления факториала */
predicates
  factorial (integer, real)
  factorial_aux(integer, real, integer, real)
clauses
  factorial(N,F):-factorial_aux(N,F,0,1).
  factorial_aux(N,F,I,P):-
     I<N, NewI=I+1, NewP=P*NewI,
     factorial_aux(N,F,NewI,NewP).
  factorial_aux(N,F,I,F):-I>=N.

goal
  write("N="), readinf(N),
  factorial(N,F),
```

```
write("F=", F),nl.
```

Для обеспечения итеративного характера процедуры возникла необходимость определить вспомогательный предикат factorial_aux с двумя дополнительными параметрами: первый из них служит для проверки условия окончания цикла, второй – для возврата вычисленного значения через всю порожденную итеративную последовательность вызовов.

3. Содержание задания по лабораторной работе.

Возвести вещественное число a в целую степень n (степень n может быть положительной, нулевой или отрицательной). Составить два варианта программы:

а) используя рекурсивное выражение a^n=a^(n-1)*a;

б) используя проверку четности степени: для нечетной степени используется рекурсивное выражение варианта а), для четной степени - a^n = a^(n/2)*a^(n/2).

5. Содержание отчета.

В отчете должны быть следующие пункты: цель работы, постановка задачи, исходный текст программы, результаты тестирования, выводы.

Контрольные вопросы:

1. Что такое рекурсия?
2. Назначение граничного условия в рекурсивном правиле.
3. Принцип построения граничного условия.
4. Может ли быть в рекурсивном правиле более одного граничного условия?
5. Как программируются разветвления в программах Visual Prolog?
6. Принципиальная разница между рекурсивной и итеративной процедурами.
7. Правила сопоставимости двух термов.
8. Поиск с возвратом в программах Visual Prolog.
9. Поточный шаблон предиката.
10. Стандартные предикаты ввода-вывода Visual Prolog.

Критерии оценки:
Всего баллов – 2 (Максимально)
2 балла выставляется все шаги выполнены полностью правильно и самостоятельно, а студент способен уверенно отвечать на вопросы преподавателя о проделанной работе;
1 балл выставляется, если работа выполнена в целом корректна, но имеются неточности в ответах на вопросе преподавателя или некоторые действия выполнены неоптимально;
0,5 балл выставляется, когда работа не может быть завершена студентом самостоятельно, без помощи преподавателя.
0 баллов выставляется, если работа не была выполнена.

Лабораторная работа 6. Использование отсечения в Пролог-программах

1. Цель работы: приобретение практических навыков использования отсечения в программах на Прологе.

2. Краткие справочные данные.

В процессе достижения цели Пролог-система осуществляет автоматический перебор вариантов, делая возврат при неуспехе какого-либо из них. Такой перебор – полезный программный механизм, поскольку он освобождает пользователя от необходимости программировать этот перебор самому. С другой стороны, ничем не

ограниченный перебор может быть источником неэффективности программы. Поэтому иногда требуется его ограничить или исключить вовсе. Для этого в Прологе предусмотрено специальное целевое утверждение !, называемое **отсечением** (cut).

Отсечение реализуется следующим образом: после согласования целевого утверждения, стоящего перед отсечением, все предложения с тем же предикатом, расположенные после отсечения, не рассматриваются.

Можно выделить три основные случая использования отсечения:

1) для устранения бесконечного цикла;

Пример: вычисление суммы 1 + 2 + … + N.

```
сумма( 1, 1 ) :- !.
сумма( N, R ) :- N1=N-1, сумма( N1, R1 ), R = R1 + N.
```

Если граничное условие будет записано в виде

```
сумма( 1, 1 ).
```

без отсечения, то сопоставление головы правила с запросом в разделе goal будет происходить успешно и при N <= 0, т.е. делается попытка доказать цель сумма(0, R), что в свою очередь приводит к цели сумма(-1, R) и т.д., т.е. образуется бесконечный цикл.

2) при программировании взаимоисключающих утверждений;

Пример: нахождение большего числа среди двух чисел X и Y можно запрограммировать в виде отношения

```
max( X, Y, Max ), где Max = X, если X >= Y и Max = Y, если Y > X.
```

Это соответствует двум предложениям программы:

```
max1( X, Y, X ) :- X >= Y.
max1( X, Y, Y ) :- X < Y.
```

Эти предложения являются взаимоисключающими, т.е. если выполняется первое, второе обязательно терпит неудачу, и наоборот. Поэтому возможна более экономная запись при использовании отсечения:

```
max2( X, Y, X ) :- X >= Y, !.
max2( _, Y, Y ).
```

3) при необходимости неудачного завершения доказательства цели;

Пример: присвоение какому-либо объекту категории в зависимости от величины заданного критерия в соответствии с таблицей.

Критерий	Категория
свыше 80 до 100	A
свыше 40 до 80	B
от 0 до 40	C

Возможный вариант программы:

```
категория( Кр, _ ) :- Кр > 100, !, fail; Кр < 0, !, fail.
категория( Кр, 'A') :- Кр > 80, !.
категория( Кр, 'B') :- Кр > 40, !.
категория( _, 'C').
```

При запросе категория(200, X) произойдет сопоставление запроса с головой первого утверждения. Цель Кр > 100 будет достигнута. Затем будет доказана цель-отсечение. Но когда встретится предикат fail, который всегда вызывает состояние неудачи, то стоящий перед ним предикат отсечения остановит работу механизма возврата и в результате ответом на запрос будет No Solution (Нет решения).

Комбинация !, fail вводит возникновение неудачи.

3. Содержание задания по лабораторной работе.

Определить возрастной статус человека по известному году рождения в соответствии с таблицей 3. Разработать два варианта программы: без отсечения и с использованием отсечения.

Возраст, лет	Статус
< 2	младенец
2 – 12	ребенок
12 – 16	подросток
16 – 25	юноша
25 – 70	мужчина
70 – 100	старик
> 100	долгожитель

4. Содержание отчета.

В отчете по лабораторной работе должны быть отражены следующие пункты: цель работы, постановка задачи, тексты Пролог-программ, результаты тестирования программ, выводы.

Контрольные вопросы:

1. Механизм работы отсечения.
2. «Красное» и «зеленое» отсечения.
3. Какое отсечение, «красное» или «зеленое» , в процедуре max2 ?
4. Повлияет ли на результат удаление отсечения в процедуре max2 ?
5. Что происходит после выполнения комбинации !, fail?
6. Могут ли располагаться подцели после предиката fail ?
7. Декларативный и процедурный смысл Пролог-программы.
8. Важен ли порядок следования предложений в процедурах max1 и max2 ?

Критерии оценки:
Всего баллов – 2 (Максимально)
2 балла выставляется все шаги выполнены полностью правильно и самостоятельно, а студент способен уверенно отвечать на вопросы преподавателя о проделанной работе;
1 балл выставляется, если работа выполнена в целом корректна, но имеются неточности в ответах на вопросе преподавателя или некоторые действия выполнены неоптимально;
0,5 балл выставляется, когда работа не может быть завершена студентом самостоятельно, без помощи преподавателя.
0 баллов выставляется, если работа не была выполнена.

Лабораторная работа 7. Классификация знаний. Исследование предметной области

Цель занятия. Изучить заданную предметную область и построить модель знаний в виде графа.

Методические указания. Для построения модели представления знаний в виде графа необходимо выполнить следующие шаги:

1) Определить целевые действия задачи (являющиеся решениями).

2) Определить промежуточные действия или цепочку действий, между начальным состоянием и конечным (между тем, что имеется, и целевым действием).

3) Опередить условия для каждого действия, при котором его целесообразно и возможно выполнить. Определить порядок выполнения действий.

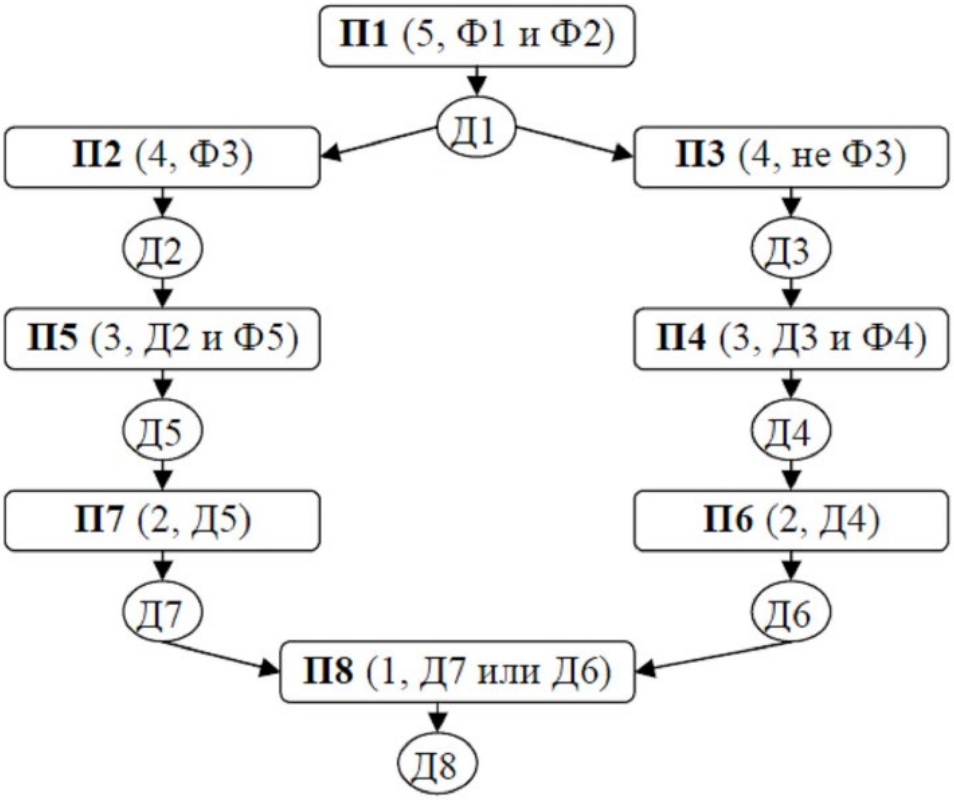
4) Добавить конкретные факты , исходя из поставленной задачи.

5) Преобразовать полученный порядок действий и соответствующие им факты, условия и действия.

6) Для проверки правильности построения записать цепочки, явно проследив связи между ними. Этот набор шагов предполагает движение при построении модели от результата к начальному состоянию, но возможно и движение от начального состояния к результату (шаги 1 и 2).

7) Присвоить обозначения фактам Ф, правилам П, действиям Д.

8) Построить граф предметной области.



Варианты заданий:

1. Построить модель представления знаний в предметной области «Железная дорога» (продажа билетов).

2. Построить модель представления знаний в предметной области «Торговый центр» (организация).

3. Построить модель представления знаний в предметной области «Автозаправка» (обслуживание клиентов).

4. Построить модель представления знаний в предметной области «Компьютерные сети» (организация).

5. Построить модель представления знаний в предметной области «Компьютерная безопасность» (средства и способы ее обеспечения).

6. Построить модель представления знаний в предметной области «Компьютерная безопасность» (угрозы).

7. Построить модель представления знаний в предметной области «Интернет-кафе» (организация и обслуживание).

8. Построить модель представления знаний в предметной области «Разработка информационных систем» (ведение информационного проекта).

9. Построить модель представления знаний в предметной области «Туристическое агентство» (работа с клиентами).

10. Построить модель представления знаний в предметной области «Кухня» (приготовление пищи).

11. Построить модель представления знаний в предметной области «Больница» (прием больных).

12. Построить модель представления знаний в предметной области «Кинопрокат» (ассортимент и работа с клиентами).

13. Построить модель представления знаний в предметной области «Прокат автомобилей» (ассортимент и работа с клиентами).

14. Построить модель представления знаний в предметной области «Операционные системы» (функционирование).

15. Построить модель представления знаний в предметной области «Информационные системы» (виды и функционирование).

16. Построить модель представления знаний в предметной области «Предприятие» (структура и функционирование).

Критерии оценки:
Студент строит модель по одному из выданных преподавателем вариантов.
Всего баллов – 5 (Максимально)
5 баллов выставляется модель построена верно и самостоятельно, а студент способен уверенно отвечать на вопросы преподавателя о проделанной работе;
3-4 балла выставляется, если модель выполнена в целом корректна, но имеются неточности в построении, которые студент замечает при наводящих вопросах преподавателя;
1-2 балла выставляется, когда модель выполнена не полностью и студент не может её закончить самостоятельно.
0 баллов выставляется, если модель не построена.

Лабораторная работа 8. Выявление знаний в системах искусственного интеллекта. Нечеткая логика. Формирование функций принадлежности в программной среде Fuzzy Logic Toolbox

Цель занятия: ознакомится со способами и средствами описания нечётких множеств и продукций в системе нечёткого вывода в интерактивном режиме использования графических средств пакета Fuzzy Logic Toolbox.

2.1. Общие сведения о пакете Fuzzy Logic Toolbox

Для рассмотрения результатов разработки и функционирования систем нечёткой логики будем использовать графические средства пакета **Fuzzy Logic Toolbox**. Эти же средства используются и при разработке систем нечёткого вывода как графический объектно-ориентированный язык автоматического программирования.

В состав этих средств входят:

— редактор систем нечёткого вывода FIS Editor (FIS);

— редактор функций принадлежности систем нечёткого вывода Membership Function

Editor (MFE);

— редактор правил систем нечёткого вывода Rule Editor;

— программа просмотра правил системы нечёткого вывода Rule Viewer;

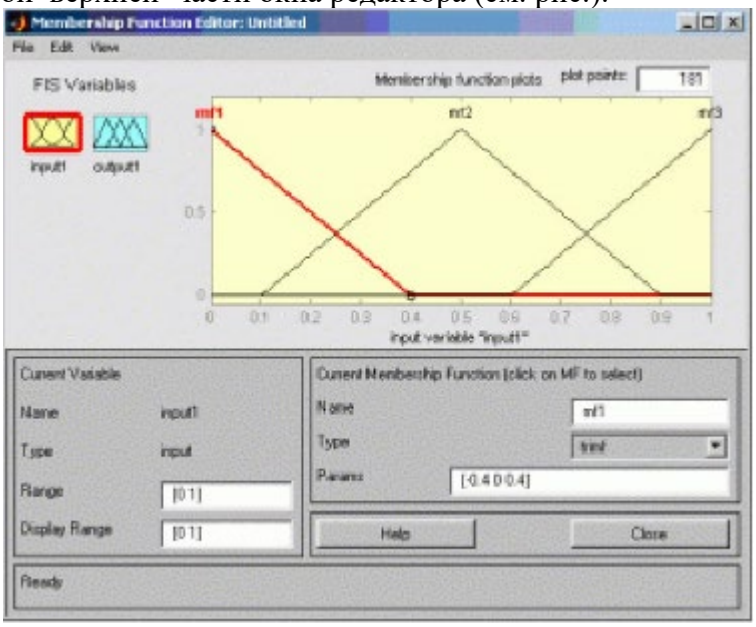
— программа просмотра поверхности нечёткого вывода Sur-face Viewer.

Для описания нечётких высказываний используются нечёткие лингвистические переменные (ЛП). ЛП — это именованная переменная, которая принимает свои значения из множества лингвистических термов, т.е. символьных величин. Для нечёткой ЛП терм-множество задаётся как нечёткое множество. Этот процесс называется фаззификацией. Фаззификация является одной из проблемных задач описания нечёткого 8 вывода и отражает индивидуальные эмпирические знания автора. Нечёткие высказывания в условной части нечёткой продукции могут быть составными, соединёнными связками “И” и/или “ИЛИ”. Эти связки при исчислении высказываний реализуются логическими или арифметическими операциями пересечения или объединения, соответственно.

При получении результата по каждому правилу необходимо дать оценку степени его истинности. Эта оценка зависит от степени истинности высказываний условной части правила, степени истинности отношения, положенного в основу правила, между исходными утверждениями (посылкой) и заключением, т.е. степени истинности импликации, и степени истинности высказывания относительно значения из терм-множества возможных результатов, приведенного в правиле. Получение оценки степени истинности заключения, полученного по правилу, называют активизацией. В случае необходимости получения чёткого количественного значения результата оно может быть получено на основании функции принадлежности терм-множества результата различными способами по алгоритмам, названым по именам их авторов (Мамдани, Сугено, Цукамото и т.д.), что определяет тип системы нечёткого вывода. Эта операция называется дефаззификацией.

2.1.3. Редактор функций принадлежности (MFE)

Редактор функций принадлежности в графическом режиме обеспечивает задание и изменение функции принадлежности любых термов ЛП СНВ. Для фаззификации лингвистической переменной СНВ следует выделить ее изображение – именованный прямоугольник в левой верхней части окна редактора (см. рис.).



В окне редактора выводятся графики функций принадлежности для всех значений выделенной ЛП (по умолчанию для трёх значений).

Для описания функции принадлежности каждого значения ЛП используются три поля: Name, Type и Params. Описываемая функция выделяется щелчком по её графику. В поле Name устанавливается значение ЛП. В поле Type, выбором элемента меню, устанавливается имя нужной функции принадлежности (одной из 11-ти встроенных). В поле ввода Params указываются необходимые параметры функции принадлежности, которые определяют положение ее модальных значений на числовой шкале, диапазон изменения которой указывается в полях ввода Range и Display range.

Эти операции выполняются над всеми значениями из терм-множеств лингвистических переменных СНВ.

Добавление нового значения ЛП со встроенной функцией принадлежности производится по команде основного меню Edit > Add MF.

Удаление ненужного значения ЛП производится нажатием клавиши Delete, после выделения графика функции принадлежности этого значения.

Задание. Для полученного варианта и разработанного графа ПрО (из прошлой лабораторной) сформировать лингвистические переменные и получить для них функции принадлежности.

Критерии оценки:
Всего баллов – 5 (Максимально)
5 баллов выставляется все шаги выполнены полностью правильно и самостоятельно, а студент способен уверенно отвечать на вопросы преподавателя о проделанной работе;
3-4 балла выставляется, если работа выполнена в целом корректна, но имеются неточности в ответах на вопросе преподавателя или некоторые действия выполнены неоптимально;
1-2 балла выставляется, когда работа не может быть завершена студентом самостоятельно, без помощи преподавателя.
0 баллов выставляется, если работа не была выполнена.

Примерная тематика докладов:

1. История развития науки об искусственном интеллекте;
2. Основные стратегии создания интеллектуальных систем;
3. Основные сферы применения искусственного интеллекта;
4. Современные проблемы искусственного интеллекта;
5. Успехи искусственного интеллекта;
6. Перспективы развития и будущее искусственного интеллекта;
7. Основные этапы создания экспертных систем;
8. Распознавание букв, слов и текста;
9. Понятия нейроконтроллера, нейрокомпьютера и нейрочипов;
10. Свойства нейронных сетей, унаследованные от мозга;
11. Методы обучения нейронных сетей;
12. Краткое описание неклассических нейронных сетей;
13. Современные технологии искусственного интеллекта;
14. Опасность искусственного интеллекта;
15. Теория игр в искусственном интеллекте;
16. Теория хаоса в искусственном интеллекте;
17. Обзор систем и языков для логического программирования;
18. Обзор систем и языков для функционального программирования.

Доклад

Задание:

Подготовить доклад по предложенной тематике.

Критерии оценки

Оценки в баллах	Критерии
1	Соответствие содержания работы выбранной теме
3	Самостоятельность в подготовке материала
3	Аргументированность и полнота материала
3	Использование современной и актуальной литературы
3	Качество презентации и изложения материала
2	Оформление работы
15	**Итого**