МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Самарский государственный технический университет»
(ФГБОУ ВО «СамГТУ»)

УТВЕРЖДАЮ:
Проректор по учебной работе

_____
(подпись, ФИО)
«____»_____20__ г.

РАБОЧАЯ ПРОГРАММА МОДУЛЯ

Методы искусственного интеллекта

УГСН	10.00.00 Информационная безопасность
Направление подготовки (специальность)	10.03.01 Информационная безопасность
Квалификация	*Бакалавр*
Форма обучения	*Очная*
Год начала подготовки	*2025*
Объем дисциплины, ч. / з.е. / недели	*180/5/3*
Образовательные технологии	*Традиционные*
Форма контроля (промежуточная аттестация)	*Семестр 1: Зачет с оценкой; Семестр 2: Экзамен, Курсовая работа*
Подразделение – разработчик	*Кафедра "Электронные системы и информационная безопасность"*

Самара 2026

Методы искусственного интеллекта

Разработчик

Разработчик РПМ
Карпова Надежда Евгеньевна
Руководитель подразделения
Карпова Надежда Евгеньевна

СОДЕРЖАНИЕ

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

2. Место модуля в структуре образовательной программы 3

3. Объем модуля в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся 4

4. Содержание модуля, структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий 5

4.1. Содержание лекционных занятий 6

4.2. Содержание лабораторных занятия 7

4.3. Содержание практических занятий 8

4.4. Содержание самостоятельной работы 9

5. Перечень учебной литературы и учебно-методического обеспечения по модулю 9

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем 10

7. Описание материально-технической базы, необходимой для осуществления образовательного процесса по модулю 10

ПРИЛОЖЕНИЕ 11

Фонд оценочных средств 11

1. Перечень планируемых результатов обучения по модулю, соотнесенных с планируемыми результатами освоения образовательной программы

Таблица 1

Наименование категории (группы) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть)
Профессиональные компетенции			
-	ПК-2 Способен проводить разработку и настройку систем защиты информации и их компонентов	ПК-2.1 Знает принципы построения систем защиты информации, включая настройку таких систем и их компонентов	Знать принципы построения систем защиты информации, включая настройку таких систем и их компонентов.
		ПК-2.2 Умеет проводить разработку и настройку систем защиты информации и их компонентов	Уметь проводить разработку и настройку систем защиты информации и их компонентов.
		ПК-2.3 Владеет навыками разработки, построения и настройки систем защиты информации и их компонентов	Владеть навыками разработки, построения и настройки систем защиты информации и их компонентов.

2. Место модуля в структуре образовательной программы

Модуль входит в Вариативную часть.

Входные требования:

Предшествующие осваиваемые модули: Технологии и методы программирования; *Последующие осваиваемые модули:* Технологии и методы программирования.

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

Таблица 3

Вид учебной работы	Всего часов	В том числе часов в электронной форме	В том числе часов в форме практической подготовки	Семестр 1 часов/часов в электронной форме/в форме практической подготовки	Семестр 2 часов/часов в электронной форме/в форме практической подготовки
Аудиторная контактная работа (всего), в том числе:	**100**	**0**	**0**	**64**	**36**
лекционные занятия (ЛЗ)	36	0	0	24	12
лабораторные работы (ЛР)	48	0	0	24	24
практические занятия (ПЗ)	16	0	0	16	0
Внеаудиторная контактная работа, КСР	**5**	**0**	**0**	**2**	**3**
Самостоятельная работа (всего), в том числе:	**75**	**0**	**0**	**6**	**69**
Подготовка к лабораторным работам	*48*	*0*	*0*	*6*	*42*

Вид учебной работы	Всего часов	В том числе часов в электронной форме	В том числе часов в форме практической подготовки	Семестр 1 часов/часов в электронной форме/в форме практической подготовки	Семестр 2 часов/часов в электронной форме/в форме практической подготовки
Выполнение курсовых работ	*27*	*0*	*0*	*0*	*27*
ИТОГО: час.	**180**	**0**	**0**	**72**	**108**
ИТОГО: з.е.	**5.0**	**0.0**	**0.0**	**2.0**	**3.0**

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

Таблица 4

№ раздела	Наименование раздела дисциплины	Виды учебной нагрузки и их трудоемкость, часы				
		ЛЗ	**ЛР**	**ПЗ**	**СРС**	**Всего часов**
1.	Понятие искусственного интеллекта. Основные функции интеллекта.	2	8	8	6	24
2.	Основные функции интеллектуальных структур.	2	8	6	0	16
3.	Система искусственного интеллекта (СИИ).	10	8	2	0	20
4.	Модели представления знаний в СИИ.	10	8	0	42	60
5.	Система распознавания образов.	4	8	0	27	39
6.	Алгоритмы распознавания образов.	8	8	0	0	16
	КСР					*5*
	Итого:	*36*	*48*	*16*	*75*	*180*

4.1. Содержание лекционных занятий

Таблица 4.1

№ ЛЗ	Наименование раздела	Тема лекции	Содержание лекции (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов	Количество часов в электронной форме
Семестр 1					
1.	Понятие искусственного интеллекта. Основные функции интеллекта.	Понятие искусственного интеллекта (ИИ).	1.1. Введение. Предмет, содержание и задачи дисциплины. Методы ее изучения. 1.2. Значение и место дисциплины в подготовке бакалавра по информационной безопасности. 2.1. Современные проблемы искусственного интеллекта. Роль и задачи ИИ в комплексной защите информации.	2	0
2.	Основные функции интеллектуальных структур.	Основные функции интеллектуальных структур.	2.1. Анализ внешней среды и внутреннего состояния системы. 2.2. Функция целеобразования. Функция накопления опыта. 2.3. Функция обучения, планирования и выполнения действий.	2	0
3.	Система искусственного интеллекта	Система искусственного интеллекта.	3.1. Обобщенная структура. СИИ. Подсистема базы данных. Подсистема базы целеобразований.	2	0

№ ЛЗ	Наименование раздела	Тема лекции	Содержание лекции (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов	Количество часов в электронной форме
	(СИИ).				
4.	Система искусственного интеллекта (СИИ).	Система искусственного интеллекта.	Блок алгоритма распознавания. Отличительные особенности СИИ от традиционных сложных технических систем.	2	0
5.	Система искусственного интеллекта (СИИ).	Система искусственного интеллекта.	Системный подход к анализу СИИ. Принципы системного исследования.	2	0
6.	Система искусственного интеллекта (СИИ).	Система искусственного интеллекта.	Функциональное описание сложных систем. Морфологическое описание систем.	2	0
7.	Система искусственного интеллекта (СИИ).	Система искусственного интеллекта.	Информационное описание систем. Тезаурус системы. Методы и средства оценки ситуаций.	2	0
8.	Модели представления знаний в СИИ.	Методы представления знаний в СИИ.	Декларативный метод представления знаний.	2	0
9.	Модели представления знаний в СИИ.	Методы представления знаний в СИИ.	Особенности этого подхода и целесообразность его применения в системах искусственного интеллекта.	2	0
10.	Модели представления знаний в СИИ.	Методы представления знаний в СИИ.	Продукционная модель представления знаний в СИИ. Операторы продукции. Правила переписывания операторов. Положительные и отрицательные стороны данного представления.	2	0
11.	Модели представления знаний в СИИ.	Методы представления знаний в СИИ.	Процедуральные модели представления знаний в СИИ. Представление знаний семантической сетью.	2	0
12.	Модели представления знаний в СИИ.	Методы представления знаний в СИИ.	Лингвистические, логические, теоретико-множественные и квантифицированные отношения в семантической сети.	2	0
			Итого за семестр:	*24*	*0*
			Семестр 2		
1.	Система распознавания образов.	Распознавание образов.	Основные понятия и методология решения. Принципы перечисления объектов распознавания. Принцип общности свойств при автоматической классификации образов. Классификация систем распознавания образов. Одноуровневые системы распознавания. Многоуровневые системы распознавания.	2	0
2.	Система распознавания образов.	Распознавание образов.	Система распознавания образов без обучения. Распознающие системы с обучением. Самообучающиеся системы распознавания образов.	2	0
3.	Алгоритмы распознавания	Алгоритмы распознавания образов.	Кластеризация образов с помощью функций расстояния. Максиминный алгоритм формирования кластеров. Автоматический	2	0

№ ЛЗ	Наименование раздела	Тема лекции	Содержание лекции (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов	Количество часов в электронной форме
	образов.		классификатор распознавания образов на основе Евклидова расстояния. Алгоритм К внутригрупповых средних.		
4.	Алгоритмы распознавания образов.	Алгоритмы распознавания образов.	Распознавание образов с помощью функций правдоподобия – задача теории статистических решений. Байесовский классификатор в случае, когда образы характеризуются нормальным распределением.	2	0
5.	Алгоритмы распознавания образов.	Алгоритмы распознавания образов.	Детерминированная задача распознавания образов. Алгоритм Персептрона – обучаемый классификатор образов. Обучаемый классификатор образов на основе алгоритма наименьшей среднеквадратической ошибки (НСКО).	2	0
6.	Алгоритмы распознавания образов.	Алгоритмы распознавания образов.	Лингвистическое, структурное распознавание образов. Обучение и грамматический вывод.	2	0
			Итого за семестр:	*12*	*0*
			Итого:	**36**	**0**

4.2. Содержание лабораторных занятий

Таблица 4.2

№ ЛабЗ	Наименование раздела	Наименование лабораторной работы	Содержание лабораторной работы (перечень рассматриваемых дидактических единиц: подтем, вопросов)	Количество часов	Количество часов в электронной форме	Количество часов в форме практической подготовки
Семестр 1						
1.	Понятие искусственного интеллекта. Основные функции интеллекта.	Генеалогическое дерево на Prolog.	Построение генеалогического дерева на Prolog.	4	0	*0*
2.	Понятие искусственного интеллекта. Основные функции интеллекта.	Алгоритмы поиска.	Изучение алгоритмов поиска.	4	0	*0*
3.	Основные функции интеллектуальных структур.	Информационная система на базе семантической сети.	Создание информационной системы на базе семантической сети.	4	0	*0*
4.	Основные функции	Нейронные сети.	Работа с нейронными сетями.	4	0	*0*

№ ЛабЗ	Наименование раздела	Наименование лабораторной работы	Содержание лабораторной работы (перечень рассматриваемых дидактических единиц: подтем, вопросов)	Количество часов	Количество часов в электронной форме	Количество часов в форме практической подготовки
	интеллектуальных структур.					
5.	Система искусственного интеллекта (СИИ).	Задача многоклассовой классификации.	Решение задачи многоклассовой классификации.	4	0	*0*
6.	Система искусственного интеллекта (СИИ).	Задача кластеризации.	Решение задачи кластеризации.	4	0	*0*
			Итого за семестр:	*24*	*0*	*0*
			Семестр 2			
1.	Модели представления знаний в СИИ.	Продукционная модель представления знаний.	Алгоритм, реализующий продукционную модель представления знаний. Написание программы, реализующей механизмы прямого и обратного вывода.	4	0	*0*
2.	Модели представления знаний в СИИ.	Фреймовая модель представления знаний. Представление знаний семантической сетью.	Составление алгоритма фреймовой модели. Составление программы вывода на экран фреймов. Разработка алгоритма семантической сети. Составление программы и отчета семантической сети.	4	0	*0*
3.	Система распознавания образов.	Моделирование и создание прототипа экспертной системы.	Построение экспертной оболочки для решения задачи классификации. Программные конструкции, реализующие модули базы данных и базы знаний.	4	0	*0*
4.	Система распознавания образов.	Нечеткие множества. Составление алгоритма.	Формирование нечетких множеств. Составление алгоритма.	4	0	*0*
5.	Алгоритмы распознавания образов.	Алгоритм Персептрона – обучаемый классификатор образов.	Изучение алгоритма Персептрона – обучаемого классификатора образов.	4	0	*0*
6.	Алгоритмы распознавания образов.	Алгоритм наименьшей среднеквадратической ошибки (НСКО).	Изучение алгоритма наименьшей среднеквадратической ошибки (НСКО).	4	0	*0*
			Итого за семестр:	*24*	*0*	*0*
			Итого:	**48**	**0**	***0***

4.3. Содержание практических занятий

Таблица 4.3

№ ПЗ	Наименование раздела	Тема практического занятия	Содержание практического занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов	Количествочасов в электронной форме	Количество часов в форме практической подготовки
Семестр 1						
1.	Понятие искусственного интеллекта. Основные функции интеллекта.	Практическое занятие №1.	Современные проблемы искусственного интеллекта. Проблемы обработки информации.	2	0	*0*
2.	Понятие искусственного интеллекта. Основные функции интеллекта.	Практическое занятие №2.	Современные проблемы искусственного интеллекта. Проблемы обработки информации.	2	0	*0*
3.	Понятие искусственного интеллекта. Основные функции интеллекта.	Практическое занятие №3.	Основные функции интеллектуальных структур. Методология решения и основные концепции в автоматическом распознавании образов.	2	0	*0*
4.	Понятие искусственного интеллекта. Основные функции интеллекта.	Практическое занятие №4.	Основные функции интеллектуальных структур. Методология решения и основные концепции в автоматическом распознавании образов.	2	0	*0*
5.	Основные функции интеллектуальных структур.	Практическое занятие №5.	Система искусственного интеллекта. Обобщенная блок схема СИИ. Блок алгоритма распознавания.	2	0	*0*
6.	Основные функции интеллектуальных структур.	Практическое занятие №6.	Система искусственного интеллекта (окончание). Интеллектуальный банк данных: база данных, база знаний и база целей.	2	0	*0*
7.	Основные функции интеллектуальн	Практическое занятие №7.	Примеры интеллектуальных систем. Структура интелллектного робота. Система восприятия. Система планирования и выполнения	2	0	*0*

№ ПЗ	Наименование раздела	Тема практического занятия	Содержание практического занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов	Количествочасов в электронной форме	Количество часов в форме практической подготовки
	ых структур.		целенаправленных действий.			
8.	Система искусственного интеллекта (СИИ).	Практическое занятие №8.	Методы представления знаний в СИИ. Анализ методов и целесообразность их применения в СИИ. Декларативное представление знаний. Процедуральное представление знаний. Представление знаний на базе фреймов.	2	0	*0*
			Итого за семестр:	*16*	*0*	*0*
			Итого:	**16**	**0**	***0***

4.4. Содержание самостоятельной работы

Таблица 4.4

Наименование раздела	Вид самостоятельной работы	Содержание самостоятельной работы (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов
Семестр 1			
Понятие искусственного интеллекта. Основные функции интеллекта.	Подготовка к лабораторным работам	Современные проблемы искусственного интеллекта. Проблемы обработки информации. Современные проблемы искусственного интеллекта. (окончание). Роль и задачи ИИ в распознавании образов.	6
		Итого за семестр:	*6*
Семестр 2			
Модели представления знаний в СИИ.	Подготовка к лабораторным работам	Тема 4.2. Продукционная модель представления знаний в СИИ. Тема 4.5. Фреймовая модель представления знаний. Тема 4.4. Представление знаний семантической сетью.	42
Система распознавания образов.	Выполнение курсовых работ	Алгоритм Персептрона – обучаемый классификатор образов. Алгоритм наименьшей среднеквадратической ошибки (НСКО).	27
		Итого за семестр:	*69*
		Итого:	**75**

5. Перечень учебной литературы и учебно-методического обеспечения по модулю

Таблица 5

№ п/п	Библиографическое описание	Ресурс НТБ СамГТУ	Ссылка на ресурс
1.	Системы искусственного интеллекта (экспертные системы) : метод.указания к лаб.работам / Самар.гос.техн.ун-т, Вычислительная техника; сост.: В. А. Лукиных, А. А. Цыганов.- Самара, 2014.- 38 с.	Электронная библиотека СамГТУ (ЭБС Политех)	http://platform.samgtu.ru/getinfo?uid=els_samgtu-elib-1043
2.	Методы искусственного интеллекта в обработке данных и изображений; Томский политехнический университет, 2016	ЭБС IPR Books	http://platform.samgtu.ru/getinfo?uid=els_samgtu-iprbooks-84054
3.	Введение в теорию искусственного интеллекта; Ай Пи Ар Медиа, 2021	ЭБС IPR Books	http://platform.samgtu.ru/getinfo?uid=els_samgtu-iprbooks-108282

Доступ обучающихся к ЭР НТБ СамГТУ (elib.samgtu.ru) осуществляется посредством электронной

информационной образовательной среды университета и сайта НТБ СамГТУ по логину и паролю.

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем

Таблица 6

№ п/п	Наименование	Ссылка на веб-страницу ресурса	Режим доступа
1.	*Scopus: база данных рефератов и цитирования [Электронный ресурс]. - Электрон. текстовые, граф., зв. дан., Сайт в сети Интернет*	*http://www.scopus.com*	*Ресурсы ограниченного доступа*
2.	*eLIBRARY.RU (НЭБ - Научная электронная библиотека "Российский индекс научного цитирования") [Электронный ресурс]. - Электрон. текстовые, граф., зв. дан., Сайт в сети Интернет*	*http://www.elibrary.ru/*	*Ресурсы открытого доступа*
3.	*УИС РОССИЯ - Университетская информационная система РОССИЯ [Электронный ресурс]. - Электрон. текстовые, граф., зв. дан., Сайт в сети Интернет*	*http://www.cir.ru/*	*Ресурсы открытого доступа*
4.	*Единое окно доступа к образовательным ресурсам [Электронный ресурс]. - Электрон. текстовые, граф., зв. дан., Сайт в сети Интернет*	*http://window.edu.ru/*	*Ресурсы открытого доступа*
5.	*Словари и энциклопедии on-line [Электронный ресурс]. - Электрон. текстовые, граф., зв. дан., Сайт в сети Интернет*	*http://dic.academic.ru/*	*Ресурсы открытого доступа*

7. Описание материально-технической базы, необходимой для осуществления образовательного процесса по модулю

Лекционные занятия проводятся в:
Учебная аудитория.

Практические занятия проводятся в:
Учебная аудитория.

Самостоятельная работа проводятся в:
Корпус № 8/200 - Читальный зал (Аудитория на 70 посадочных мест оснащена следующими техническими средствами обучения и оборудованием: учебная мебель, столы учебные.).

Лабораторные занятия проводятся в:
Корпус № 8/109 - Учебная аудитория (Аудитория на 20 посадочных мест оснащена следующими техническими средствами обучения и оборудованием: учебная мебель, столы учебные, доска аудиторная. Коммутатор D-Link DES-1016D. 10 ПК P4631\3.0GHz64bit2M в комплекте. На ПК установлено следующее программное обеспечение: операционная система MS Windows 7, офисный пакет MS Office 2007, Auto Cad, ЛИРА, антивирусное ESET nod 32, Auto CAD, PTC Mathcad Prime 3.1, ArchiCAD. Обеспечено проводное подключение ПК к локальной сети и сети Интернет.).

ПРИЛОЖЕНИЕ

Фонд оценочных средств

Фонд оценочных средств для текущего контроля успеваемости, промежуточной аттестации обучающихся по модулю

Настоящий Фонд оценочных средств (ФОС) по модулю Методы искусственного интеллекта является неотъемлемым приложением к рабочей программе модуля. На данный ФОС распространяются все реквизиты утверждения, представленные в рабочей программе модуля.

1. Перечень оценочных средств

Таблица 1

№ п/п	Оценочное средство	Краткая характеристика оценочного средства	Количество	Представление оценочного средства в ФОС
1	Отчёт по лабораторной работе	Оценка способности студента применить полученные ранее знания для проведения анализа, опыта, эксперимента и выполнения последующих расчетов, а также составления выводов.	12	Требования к содержанию и оформлению отчета
2	Контрольная точка в форме тестирования	Инструмент, с помощью которого преподаватель оценивает степень достижения студентом требуемых знаний, умений, навыков в формате рубежного контроля по итогам изучения разделов. Составление теста включает в себя создание выверенной системы вопросов, собственно процедуру проведения тестирования и способ измерения полученных результатов.	2	Спецификация тестов и примеры вариантов тестовых заданий

Раздел 2. Этапы формирования компетенций, соотнесенные с видами оценочных средств

Таблица 2

Контролируемые разделы модуля	Код и формулировка индикатора компетенции	Результат обучения	Оценочные средства	Номер оценочного средства
Понятие искусственного интеллекта. Основные функции интеллекта.	ПК-2.1 Знает принципы построения систем защиты информации, включая настройку таких систем и их компонентов	Знать принципы построения систем защиты информации, включая настройку таких систем и их компонентов.	Отчёт по лабораторной работе	1-2
Понятие искусственного интеллекта. Основные функции интеллекта.	ПК-2.2 Умеет проводить разработку и настройку систем защиты информации и их компонентов	Уметь проводить разработку и настройку систем защиты информации и их компонентов.	Отчёт по лабораторной работе	1-2
Понятие искусственного интеллекта. Основные функции интеллекта.	ПК-2.3 Владеет навыками разработки, построения и настройки систем защиты информации и их компонентов	Владеть навыками разработки, построения и настройки систем защиты информации и их компонентов.	Отчёт по лабораторной работе	1-2
Основные функции интеллектуальных	ПК-2.1 Знает принципы построения	Знать принципы построения систем	Отчёт по лабораторной работе	3-4

Контролируемые разделы модуля	Код и формулировка индикатора компетенции	Результат обучения	Оценочные средства	Номер оценочного средства
структур.	систем защиты информации, включая настройку таких систем и их компонентов	защиты информации, включая настройку таких систем и их компонентов.		
Основные функции интеллектуальных структур.	ПК-2.2 Умеет проводить разработку и настройку систем защиты информации и их компонентов	Уметь проводить разработку и настройку систем защиты информации и их компонентов.	Отчёт по лабораторной работе	3-4
Основные функции интеллектуальных структур.	ПК-2.3 Владеет навыками разработки, построения и настройки систем защиты информации и их компонентов	Владеть навыками разработки, построения и настройки систем защиты информации и их компонентов.	Отчёт по лабораторной работе	3-4
Система искусственного интеллекта (СИИ).	ПК-2.1 Знает принципы построения систем защиты информации, включая настройку таких систем и их компонентов	Знать принципы построения систем защиты информации, включая настройку таких систем и их компонентов.	Отчёт по лабораторной работе	5-6
Система искусственного интеллекта (СИИ).	ПК-2.2 Умеет проводить разработку и настройку систем защиты информации и их компонентов	Уметь проводить разработку и настройку систем защиты информации и их компонентов.	Отчёт по лабораторной работе	5-6
Система искусственного интеллекта (СИИ).	ПК-2.3 Владеет навыками разработки, построения и настройки систем защиты информации и их компонентов	Владеть навыками разработки, построения и настройки систем защиты информации и их компонентов.	Отчёт по лабораторной работе	5-6
Система искусственного интеллекта (СИИ).	ПК-2.1 Знает принципы построения систем защиты информации, включая настройку таких систем и их компонентов	Знать принципы построения систем защиты информации, включая настройку таких систем и их компонентов.	Контрольная точка в форме тестирования	1
Модели представления знаний в СИИ.	ПК-2.1 Знает принципы построения систем защиты информации, включая настройку таких систем и их компонентов	Знать принципы построения систем защиты информации, включая настройку таких систем и их компонентов.	Отчёт по лабораторной работе	7-8
Модели представления знаний в СИИ.	ПК-2.2 Умеет проводить разработку и настройку систем защиты информации и их компонентов	Уметь проводить разработку и настройку систем защиты информации и их компонентов.	Отчёт по лабораторной работе	7-8
Модели представления знаний в СИИ.	ПК-2.3 Владеет навыками разработки, построения и настройки систем защиты информации и их компонентов	Владеть навыками разработки, построения и настройки систем защиты информации и их компонентов.	Отчёт по лабораторной работе	7-8
Система распознавания	ПК-2.1 Знает принципы построения	Знать принципы построения систем	Отчёт по лабораторной работе	9-10

Контролируемые разделы модуля	Код и формулировка индикатора компетенции	Результат обучения	Оценочные средства	Номер оценочного средства
образов.	систем защиты информации, включая настройку таких систем и их компонентов	защиты информации, включая настройку таких систем и их компонентов.		
Система распознавания образов.	ПК-2.2 Умеет проводить разработку и настройку систем защиты информации и их компонентов	Уметь проводить разработку и настройку систем защиты информации и их компонентов.	Отчёт по лабораторной работе	9-10
Система распознавания образов.	ПК-2.3 Владеет навыками разработки, построения и настройки систем защиты информации и их компонентов	Владеть навыками разработки, построения и настройки систем защиты информации и их компонентов.	Отчёт по лабораторной работе	9-10
Алгоритмы распознавания образов.	ПК-2.1 Знает принципы построения систем защиты информации, включая настройку таких систем и их компонентов	Знать принципы построения систем защиты информации, включая настройку таких систем и их компонентов.	Отчёт по лабораторной работе	11-12
Алгоритмы распознавания образов.	ПК-2.2 Умеет проводить разработку и настройку систем защиты информации и их компонентов	Уметь проводить разработку и настройку систем защиты информации и их компонентов.	Отчёт по лабораторной работе	11-12
Алгоритмы распознавания образов.	ПК-2.3 Владеет навыками разработки, построения и настройки систем защиты информации и их компонентов	Владеть навыками разработки, построения и настройки систем защиты информации и их компонентов.	Отчёт по лабораторной работе	11-12
Алгоритмы распознавания образов.	ПК-2.1 Знает принципы построения систем защиты информации, включая настройку таких систем и их компонентов	Знать принципы построения систем защиты информации, включая настройку таких систем и их компонентов.	Контрольная точка в форме тестирования	2

3. Содержание оценочных средств текущего контроля

Оценочное средство 1:
Пример отчета и вопросы по лабораторной работе прилагаются к рабочей программе.
Оценочное средство 2:
Пример тестов прилагается к рабочей программе.

4. Содержание оценочных средств промежуточной аттестации

Семестр 1:

Вид аттестационного испытания 1: Письменная работа.
Форма проведения промежуточной аттестации: Зачет с оценкой.
Типовые задания: Типовые вопросы к экзамену прилагаются к рабочей программе.

Семестр 2:

Вид аттестационного испытания 1: Письменная работа.
Форма проведения промежуточной аттестации: Экзамен.

Типовые задания: Типовые вопросы к экзамену прилагаются к рабочей программе.

Вид аттестационного испытания 2: Письменная работа.

Форма проведения промежуточной аттестации: Курсовая работа.

Типовые задания: Примерные методические указания к выполнению курсовой работы прилагаются к рабочей программе.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций

Модуль состоит из набора разделов. Объем учебного раздела раскрывает отдельную (ые) тему(ы) модуля / этап прохождения практики. Модуль разбивается, как правило, на 2-4 раздела в семестр.

Раздел состоит из учебных элементов (дидактических единиц), сгруппированных по темам в рамках раздела. Объем учебного материала учебного элемента раскрывает отдельный вопрос темы. Путем набора соответствующих тем и учебных элементов формируется раздел, содержание которого соответствует запланированным результатам освоения ОП согласно учебному плану.

По каждому разделу устанавливается перечень обязательных видов работы студента, включающий: ответы на теоретические вопросы на семинаре; решение практических задач и выполнение заданий на практических занятиях; выполнение лабораторных работ; выполнение контрольных работ; написание рефератов; участие в коллоквиумах; тестирование по теме (группе тем); другие виды работ, определяемые преподавателем (в соответствии с установленными требованиями балльно-рейтинговой технологии).

Каждый раздел включает определенные формы контроля для оценки уровня усвоения учебного материала и определения рейтинговой оценки качества усвоения учебного материала.

Рейтинговая оценка знаний студентов определяется по 100-балльной шкале в каждом семестре и включает текущий, рубежный и итоговый контроль.

Накопленный рейтинг (фиксированная величина по окончании семестра) — это фактический рейтинг по всем освоенным к данному моменту разделам дисциплины, включая их текущий, рубежный и итоговый контроль, составляет N баллов (N – индивидуальная величина, зависящая от проделанной работы студента по дисциплине).

Проходной рейтинг (включает текущий контроль, рубежный контроль в семестре и промежуточную аттестацию) — это минимум баллов, набрав который студент будет считаться аттестованным по дисциплине, составляет 60 баллов в семестр по одной дисциплине.

Пороговый рейтинг (включает текущий контроль и рубежный контроль в семестре) — это минимальный фактический рейтинг семестрового контроля, набрав который, студент проходит на итоговый контроль, составляет 50% от суммы баллов, которые студент может набрать за текущий контроль и рубежный контроль в течение семестра.

Текущий контроль служит для получения первичной информации о ходе усвоения отдельных элементов содержания учебной дисциплины. Основное его назначение состоит в осуществлении контроля за всеми видами аудиторной и самостоятельной внеаудиторной деятельности студентов, результаты которого оцениваются в баллах.

Рубежный контроль (контрольные точки текущего контроля) проводится после изучения завершенной в смысловом и логическом плане части учебного материала (раздела, модуля). Этот вид контроля позволяет получить информацию о ходе усвоения некоторой совокупности содержательных элементов учебной дисциплины (раздела, модуля) и оценивается в баллах.

Промежуточный итоговый контроль (контрольные точки промежуточной аттестации) – курсовой проект, курсовая работа и экзамен, зачет, зачет с оценкой. Итоговый контроль по дисциплине в семестре предназначен для аттестации студента по учебной дисциплине: процедура установления соответствия уровня и качества подготовки студентов системе требований, зафиксированной в ФГОС ВО.

Распределение баллов рейтинговой оценки между видами контроля

Таблица 3

Номер семестра	Распределение баллов					Проходной рейтинг
	Текущий и рубежный контроль			Промежуточный итоговый контроль		
	Текущий контроль	Рубежный контроль	Пороговый рейтинг	Курсовая работа/курсовой проект	Зачет/зачет с оценкой/экзамен	
1	40	30	35	0	30	60
2	40	30	35	15	15	

Перевод оценок 100-балльной шкалы в оценки промежуточной аттестации

Таблица 4

100-балльная шкала	85-100	75-84	60-74	0-59
5-балльная шкала	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Бинарная шкала	Зачтено			Не зачтено

Распределение рейтинговых баллов по разделам модуля с указанием применяемых для их начисления оценочных средств представлено в Технологической карте БРТ.

Технологическая карта БРТ

Семестр 1

Раздел учебного модуля	Оценочные средства			Итого
	Отчёт по лабораторной работе	**Контрольная точка в форме тестирования**	**Зачет с оценкой**	
Раздел 1				
ЛР 1.1	6			6
ЛР 1.2	6			6
Итого по разделу 1	12			12
Раздел 2				
ЛР 2.3	7			7
ЛР 2.4	7			7
Итого по разделу 2	14			14
Раздел 3				
ЛР 3.5	7	0		7
ЛР 3.6	7	30		37
Итого по разделу 3	14	30		44
Промежуточная аттестация			30	30
Итого	40	30	30	100

Семестр 2

Раздел учебного модуля	Оценочные средства				Итого
	Отчёт по лабораторной работе	**Контрольная точка в форме тестирования**	**Экзамен**	**Курсовая работа**	
Раздел 1					
ЛР 1.1	6				6
ЛР 1.2	7				7
Итого по разделу 1	13				13
Раздел 2					
ЛР 2.3	7				7
ЛР 2.4	7				7
Итого по разделу 2	14				14
Раздел 3					
ЛР 3.5	7	0			7
ЛР 3.6	6	30			36
Итого по разделу 3	13	30			43
Промежуточная аттестация			15	15	30
Итого	40	30	15	15	100

Методические указания к практической части лабораторных работ

ЛАБОРАТОРНАЯ РАБОТА №1. ПРОДУКЦИОННАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Цель работы – ознакомиться с методами представления знаний, использующих продукционную модель.

Сведения из теории

В продукционной модели (модели правил) знания представлены совокупностью правил вида "ЕСЛИ-ТО". Системы с базами знаний, основанные на этой модели, называются продукционными системами.

Продукционные системы бывают двух диаметрально противоположных типов - с прямыми и обратными выводами. Типичными представителями первого типа являются системы, используемые для решения задач диагностического характера, а типичными представителями систем второго типа - системы, используемые для решения задач проектирования.

Системы продукций с прямыми выводами среди систем, основанных на использовании знаний, имеют наиболее давнюю историю, поэтому они являются в некотором смысле основополагающими. Эти системы включают три компонента: базу правил, состоящую из набора продукций (правил вывода), базу данных, содержащую множество фактов, и интерпретатор для получения логического вывода на основании этих знаний. База правил и база данных образуют базу знаний, а интерпретатор соответствует механизму логического вывода. Вывод выполняется в виде цикла "понимание-выполнение", причем в каждом цикле выполняемая часть выбранного правила обновляет базу данных. В результате содержимое базы данных преобразуется от первоначального к целевому, т.е. целевая система синтезируется в базе данных.

Поясним вышесказанное на примере, приведенном в работе. Пусть данные, хранящиеся в базе данных, представляют собой образцы в виде наборов символов, например, "намерение - отдых", "место отдыха - горы" и т. п. Правила, накапливаемые в базе правил, содержат в условной части либо одиночные образцы, либо несколько условий, соединенных союзом "и", а в заключительной части - образцы, дополнительно помещаемые в базу данных.

Итак, упорядочим сильные и слабые стороны хорошо известных систем продукций. Преимущества:

- •простота создания и понимания отдельных правил;
- •простота пополнения и модификации;
- •простота механизма логического вывода.

К недостаткам можно отнести следующее:

- •неясность взаимных отношений правил;
- •сложность оценки целостного образа знаний;
- •крайне низкая эффективность обработки;
- •отличие от человеческой структуры знаний;
- •отсутствие гибкости в логическом выводе.

Порядок выполнения работы:

1. Составьте алгоритм, реализующий продукционную модель представления знаний в произвольной предметной области. Структурно алгоритм должен состоять из трёх частей:

- Базы данных, в которой хранятся факты, цели и утверждения (15-20 записей);
- Базы правил, это конструкции вида [ЕСЛИ - ТО], оперирующие с информацией из базы данных (10 – 15 правил);
- Механизм вывода, реализующий утверждения и использующий механизмы прямого и обратного вывода.

2. По составленному алгоритму напишите программу, осуществляющую вывод на экран заключения по введённому с клавиатуры пользователем цели или факта.

3. Составьте отчёт, содержащий алгоритм, пояснения к работе программы и листинг

Пример 1:

Правило 1. ЕСЛИ "намерение - отдых" и "дорога ухабистая", ТО "использовать джип"

Правило 2. ЕСЛИ "место отдыха - горы", ТО "дорога ухабистая"

После того как в базу данных заносятся образцы "намерение - отдых" и "место отдыха - горы", рассматривается возможность применения этих правил.

Сначала механизм вывода сопоставляет образцы из условной части правила с образцами, хранящимися в базе данных. Если все образцы имеются в базе данных, то условная часть считается истинной, в противном случае - ложной.

В данном примере образец "намерение -отдых" существует в базе данных, а образец "дорога ухабистая" отсутствует, поэтому условная часть правила 1 считается ложной.

Что касается правила 2, то его условная часть истинна. Поскольку в данном случае существует только одно правило с истинно условной частью, то механизм вывода сразу же выполняет его заключительную часть, и образец "дорога ухабистая" заносится в базу данных.

При попытке вторично применить эти правила получается, что можно применить лишь правило 1, поскольку правило 2 уже было применено и выбыло из числа кандидатов. К этому времени содержимое рабочей памяти было дополнено новым образцом - результатом применения правила 2, поэтому условная часть правила 1 становится истинной, и содержимое базы данных пополняется образцом его заключительной части - "использовать джип".

В системе продукций с обратными выводами механизм логического вывода основан на ином принципе.

Поясним этот принцип на том же примере.

Допустим, что цель - это "использовать джип", и исследуем сначала возможность применения правила 1, подтверждающего этот факт. Поскольку образец "намерение - отдых" из условной части правила 1 уже занесен в базу данных, то для достижения цели достаточно подтвердить факт "дорога ухабистая". Однако если принять образец "дорога ухабистая" за новую цель, то потребуется правило, подтверждающее этот факт.

Поэтому исследуем возможность применения правила 2. Условная часть этого правила в данный момент является истинной, поэтому правило 2 можно сразу же применять. При этом база данных пополнится образцом "дорога ухабистая", и в результате возможности применения правила 1 подтверждается цель "использовать джип".

Пример 2:

Листинг программы

```
(deffunction my-shedule (?question) # определяем функцию
        (printout t ?question " (yes or no) ") # предоставляем пользователю возможность ввести один из двух вариантов ответа
        (bind ?answer (read)) # читаем данный пользователем ответ
        (while (and (neq ?answer yes) (neq ?answer no)) # до тех пор пока пользователь не введет правильный ответ, просить его ввести ответ еще раз
                (printout t ?question " (yes or no) ")
                (bind ?answer (read))) # читаем данный пользователем ответ
        (return ?answer)) # возвращаем значение функции

(defrule rule1 # определяем правило 1
        (not (Vorozey ?)) # Создаем шаблон правила, состоящий из ограничения, предназначенного для сопоставления с полями факта: проверяем отсутствие факта в списке фактов
    =>
        (bind ?Vorozey (my-shedule "Is your first teacher Vorozeykin?")) # задаем пользователю вопрос
        (assert (Vorozey ?Vorozey)) # добавляем факт к списку фактов
    )
```

```
(defrule rule2 # определяем правило 2
    (and (not (Karpova ?)) (Vorozey no)) # Создаем шаблон правила, состоящий из ограничения, предназначенного для сопоставления с полями факта: проверяем отсутствие факта в списке фактов и наличие отрицательного ответа на заданный пользователю вопрос
    =>
        (bind ?Karpova (my-shedule "Is your first teacher Karpova?")) # задаем пользователю вопрос
        (assert (Karpova ?Karpova)) # добавляем факт к списку фактов

    )

    (defrule rule3 # определяем правило 3
        (and (not (Muratova ?)) (Karpova no)) # Создаем шаблон правила, состоящий из ограничения, предназначенного для сопоставления с полями факта: проверяем отсутствие факта в списке фактов и наличие отрицательного ответа на заданный пользователю вопрос
        =>
            (bind ?Muratova (my-shedule "Is your first teacher Muratova?")) # задаем пользователю вопрос
            (assert (Muratova ?Muratova)) # добавляем факт к списку фактов
    )
    (defrule rule4 # определяем правило 4
        (and (not (Chetaev ?)) (Muratova no)) # Создаем шаблон правила, состоящий из ограничения, предназначенного для сопоставления с полями факта: проверяем отсутствие факта в списке фактов и наличие отрицательного ответа на заданный пользователю вопрос
        =>
            (bind ?Chetaev (my-shedule "Is your first teacher Chetaev?")) # задаем пользователю вопрос
            (assert (Chetaev ?Chetaev)) # добавляем факт к списку фактов
    )
    (defrule rule5 # определяем правило 5
        (or (Vorozey yes)(Chetaev yes)) # Создаем шаблон правила, состоящий из ограничения, предназначенного для сопоставления с полями факта: проверяем наличие хотя бы одного положительного ответа на заданные вопросы
        =>
            (printout t "Go to the canteen" crlf) # выводим сообщение на экран
            (assert (have a nice rest)) # добавляем новый факт к списку фактов
    )
    (defrule rule6 # определяем правило 6
        (or (Karpova yes)(Muratova yes)) # Создаем шаблон правила, состоящий из ограничения, предназначенного для сопоставления с полями факта: проверяем наличие хотя бы одного положительного ответа на заданные вопросы
        =>
            (printout t "Study hard and don't be late" crlf) # выводим сообщение на экран
            (assert (hard day)) # добавляем новый факт к списку фактов
    )
    (defrule rule7 # определяем правило 7
        (Chetaev no) # Создаем шаблон правила, состоящий из ограничения, предназначенного для сопоставления с полями факта: проверяем наличие положительного ответа на заданный вопрос
        =>
            (printout t "Stay at home and sleep!" crlf) # выводим сообщение на экран
            (assert (mne ko vtoroy)) # добавляем новый факт к списку фактов
```

Контрольные вопросы

1. Каких типов бывают продукционные системы?

2. На каком принципе основан механизм логического вывода в системе продукций с обратными выводами?
3. Какие преимущества и недостатки систем продукций?

ЛАБОРАТОРНАЯ РАБОТА №2. ФРЕЙМОВАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Цель работы - ознакомление с методами представления знаний, использующих фреймовую модель, и способами представления данной модели с помощью вычислительных средств.

Сведения из теории

Фреймовая модель, или модель представления знаний, основанная на фреймовой теории М. Минского, представляет собой систематизированную в виде единой теории психологическую модель памяти человека и его сознания. Однако из-за отсутствия конкретных идей, позволяющих создать язык представления знаний только на основе этой теории, она носит довольно абстрактный характер. Важным моментом в этой теории является понимание фрейма - структуры данных для представления некоторого концептуального объекта. Каждый фрейм описывает один концептуальный объект, а конкретные свойства этого объекта и факта, относящиеся к нему, описываются в слотах - структурных элементах данного фрейма.

В общем случае фрейм можно представить в виде следующей структуры.

(Имя фрейма:
Имя слота 1 (значение слота 1); Имя слота 2 (значение слота 2);
• • •
Имя слота N (значение слота N)).

Значением слота может быть практически что угодно: числа, формулы, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма или других фреймов. В качестве значения слота может выступать набор слотов более низкого уровня, что позволяет реализовывать во фреймовых представлениях "принцип матрешки".

Например, фрейм служащего выглядит следующим образом:

(Служащий:
Фамилия (Иванов) ;
Год рождения (1974);
Специальность (Программист);
Стаж (3)).

Все фреймы взаимосвязаны и образуют единую фреймовую систему, в которой органически объединены декларативные и процедурные знания. Поскольку концептуальному представлению свойственна иерархичность, целостный образ знаний строится в виде одной фреймовой системы, имеющей иерархическую структуру.

Механизм управления выводом может быть организован следующим образом. Связи между данным фреймом и другими фреймами задаются при помощи специального слота, значением которого является присоединенная процедура - специфичная процедура вывода в этом фрейме. При осуществлении вывода сначала запускается одна из присоединенных процедур некоторого фрейма. Затем оценивается возвращаемое значение, и в зависимости от него последовательно за-пускаются присоединенные процедуры других фреймов. В ходе этого процесса происходит генерация и уничтожение слотов, изменение значений слотов и т. п. Таким образом, происходит постепенное продвижение к получению целевого значения.

Язык представления знаний, основанных на фреймовой модели, особенно эффективен для структурного описания сложных понятий и решения задач, в которых в соответствии с ситуацией желательно применять различные способы вывода.

В то же время на таком языке затрудняется управление завершенностью и постоянством целостного образа. В частности, по этой причине существует большая опасность нарушения присоединенной процедуры. Следует отметить, что фреймовую систему без механизма присоединенных процедур (а, следовательно, и механизма пересылки сообщений) часто используют как базу данных системы продукций.

Теория фреймов послужила толчком к разработке нескольких языков представления знаний, которые благодаря своим широким возможностям и гибкости получили широкое распространение.

Порядок выполнения работы:

1. Составьте алгоритм, реализующий фреймовую модель представления знаний. В качестве предметной области используйте данные об аудитории (рис. 1)
2. Определите имена фреймов, имена слотов и их заполнители;
3. Составьте отчёт, содержащий алгоритм.
4. Составьте программу, осуществляющую вывод на экран фреймов и их содержимого различного уровня иерархии (аудитории, передней стены и т.д.), а также предусмотрите возможность создания структуры более высокой иерархии (например, фрейм этажа).
5. Составьте отчёт, содержащий пояснения к работе программы и листинг программы.

Пример 1:

Пример фреймовой системы, описывающей аудиторию, показан на рис.1

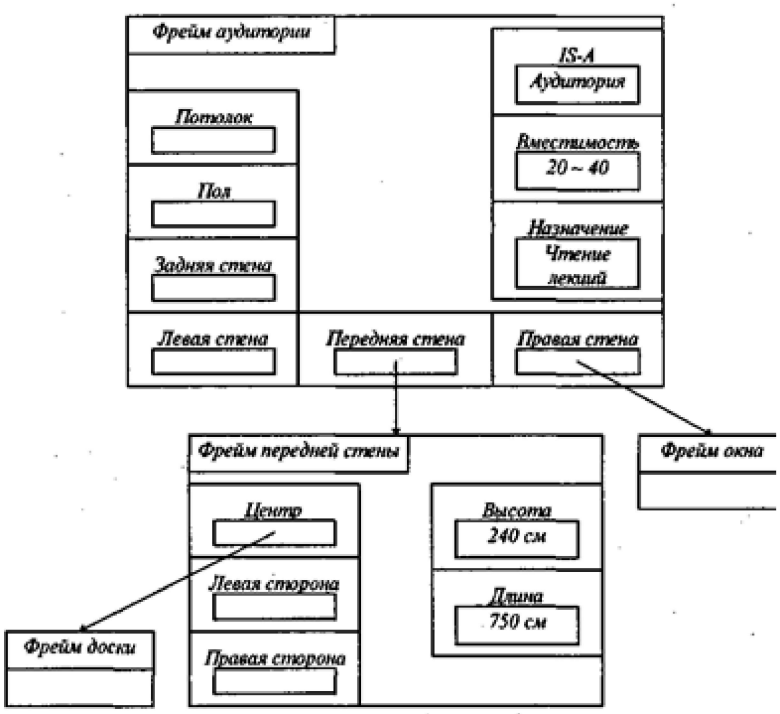


Рис. 1. Пример фреймового представления аудитории

Пример 2:

Программа представляет объектную модель автомата по продаже газированной воды.

Листинг программы

```
(defclass gas-water-automate
(is-a USER)                    # родительский класс
(role concrete)                # класс неабстрактный
(pattern-match reactive)       # класс активный
(slot gas-water (type INTEGER) (create-accessor read-write))          # вода
(slot syrup (type INTEGER) (create-accessor read-write))              # сироп
 (message-handler getwater)                  # предварительно объявляем обработчик сообщений
```

```
)
(definstances automates    # объявляем экземпляр класса
(our-automate of gas-water-automate
(gas-water 2)
(syrup 3)
)
)
 (defmessage-handler gas-water-automate getwater (?money) # обработчик сообщений
(if (= ?money 1) then                    # если 1 монета, то выдаем воду без сиропа
(if (> (dynamic-get gas-water) 0) then          # если воды нет, уведомить об этом
(dynamic-put gas-water (- (dynamic-get gas-water) 1))
(printout t «Your gas-water, please» crlf)
else
(printout t «Sorry, no more gas-water» crlf)
)
else
(if (= ?money 3) then                    # если 3 монеты, то выдаем воду с сиропом
# если воды или сиропа нет, то нужно оповестить об этом
(if (and (> (dynamic-get gas-water) 0) (> (dynamic-get syrup) 0)) then
(dynamic-put gas-water (- (dynamic-get gas-water) 1))
(dynamic-put syrup (- (dynamic-get syrup) 1))
(printout t «Your gas-water with syrup, please» crlf)
else
(printout t «Sorry, no more gas-water or syrup» crlf)
)
else                    # если ввели не 1 и не 3 монеты, то сообщаем об этом
(printout t «Wrong money» crlf)
)
)
)
# правило, которое выполняется когда в базе фактов есть монеты
(defrule drinkwater
?f<-(money ?money)
=>
 (send [our-automate] getwater ?money)      # вызываем обработчик сообщений
 (retract ?f)                    # удаляем факт
(deffacts world    #  чтобы получить воду с сиропом необходимо задать факт
(money 3)
)
```

Была составлена программа при использовании возможностей, предоставляемых расширением COOL.

Контрольные вопросы

1. Что собой представляет фреймовая модель?
2. Опишите как может быть организован механизм управления выводом?
3. Как выглядит общая структура фрейма?
4. Какие значения может принимать слот?

ЛАБОРАТОРНАЯ РАБОТА № 3. ПРЕДСТАВЛЕНИЕ ЗНАНИЙ СЕМАНТИЧЕСКОЙ СЕТЬЮ

Цель работы - ознакомится с методами представления знаний, использующих семантическую сеть, и способами представления семантической сети с помощью вычислительных средств.

Сведения из теории

Семантической сетью называется структура данных, имеющая определенный смысл как сеть. Стандартного определения семантической сети не существует, но обычно под ней подразумевают систему знаний, имеющую определенный смысл в виде целостного образа сети, узлы которой соответствуют понятиям и объектам, а дуги - отношениям между объектами. Следовательно, всевозможные сети можно рассматривать как сети, входящие в состав семантической сети.

Например, если "Петров" и "студент" являются узлами сети, то, установив между этими узлами связь "есть", получим смысловое предложение "Петров есть студент" -. Аналогичное предложение можно получить, если глагольную связку "есть" принять за отдельный узел и установить связь, имеющую грамматический смысл, из другого узла к этим трем узлам.

Особенность семантической сети заключается в целостности системы, выполненной на ее основе, не позволяющей разделить базу знаний и механизм выводов. Обычно интерпретация семантической сети определяется с помощью использующих ее процедур.

Эти процедуры основаны на нескольких способах, но наиболее типичный из них - это способ сопоставления частей сетевой структуры. Он основан на построении подсети, соответствующей вопросу, и сопоставлении ее с базой данных сети.

Порядок выполнения работы:

1. Составьте алгоритм, реализующий семантическую сеть, изображённую на рисунке №1. Определите имена фреймов, имена слотов и их заполнители;
2. Составьте отчёт, содержащий алгоритм семантической сети.
3. Составьте программу, которая бы отвечала на запросы пользователя (типа “КТО ЕСТЬ ПЕТРОВ”,” ЧТО ЕСТЬ МОСКВИЧ” и т.п.) согласно данной семантической сети.
4. Составьте отчёт, содержащий пояснения к работе программы и листинг программы.

Пример 1:

На рис. 1 показан пример семантической сети, иллюстрирующей предложение "Петров на протяжении периода времени с tl по t2 владел автомобилем марки "Святогор". На рисунке показаны дуги s, e, владелец, объект, начало, конец;

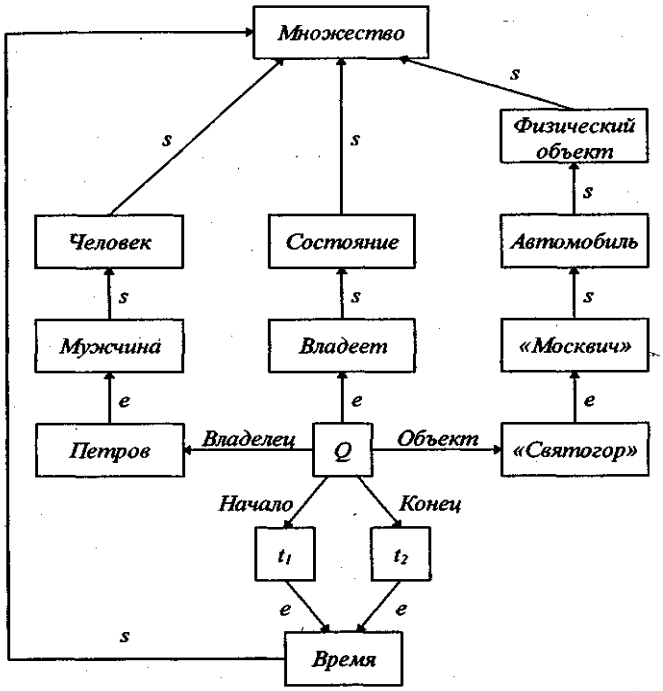


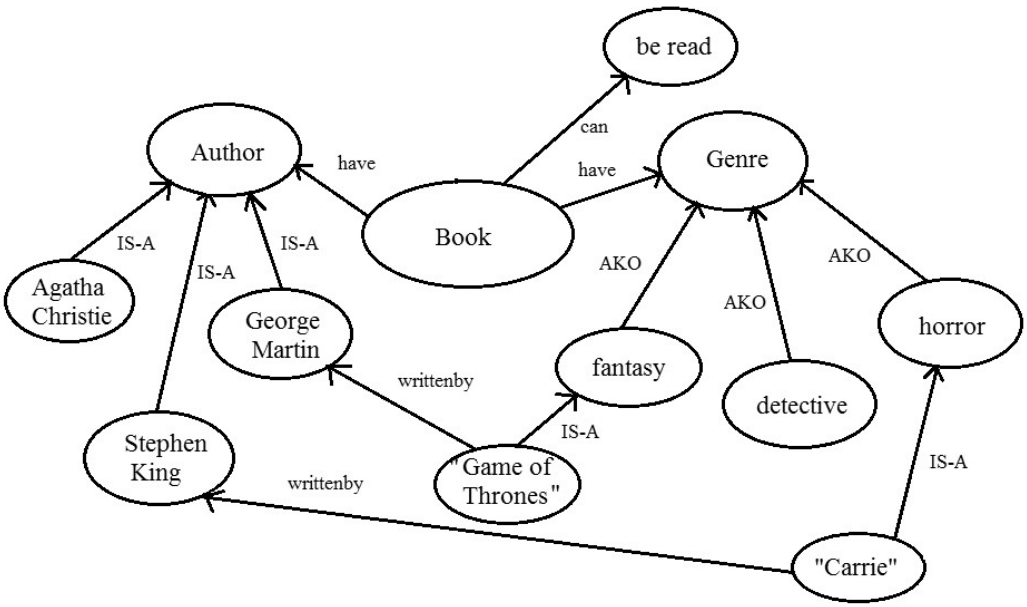
Рис. 1. Пример семантической сети

s и e означают "подмножество" и "элемент" соответственно и показывают иерархические понятия.
Q - обозначает конкретную описываемую ситуацию.

Контрольные вопросы

1. Что означает понятие «семантическая сеть»?
2. В чем заключается особенность семантической сети?

Пример 2:



Листинг программы

```
(deftemplate objects        # с помощью конструкции deftemplate, определяется структура вершин семантической сети
        (slot node)         # слот node отражает характеристику вершины
        (slot ako)          # задаем отношение «является разновидностью»
        (slot isa)# задаем отношение «есть»
        (slot have1)        # задаем отношение «имеет1»
        (slot have2)        # задаем отношение «имеет2»
        (slot can)          # задаем отношение «может»
        (slot writtenby)    # задаем отношение «написана»
)

(deffacts start      #  конструкция deffacts позволяет вводить множество фактов сразу
        (objects (node book) (can "be read") (have1 author) (have2 genre))
        (objects (node "be read"))
        (objects (node author))
        (objects (node genre))

        (objects (node fantasy) (ako genre))
        (objects (node detective) (ako genre))
        (objects (node horror) (ako genre))

        (objects (node "Agatha Christie") (isa author))
        (objects (node "Stephen King") (isa author))
        (objects (node "George Martin") (isa author))

        (objects (node "Game of Thrones") (isa fantasy) (writtenby "George Martin"))
        (objects (node "Carrie") (isa horror) (writtenby "Stephen King"))

)

(defrule rule1
        (declare (salience 1))       # данное ключевое слово позволяет явно задать приоритеты правил
        (objects (node ?x) (isa ?y&~nil) (writtenby ?z&~nil))          # правило rule1 ищет такой факт objects, у которого слот writtenby не равен значению nil. Это реализуется специальным символом & со знаком отрицания ~
=>
        (printout t "The " ?x " is a " ?y " by " ?z crlf)          # вывод на экран такого факта
)

(defrule rule2
        (declare (salience 5))       # данное ключевое слово позволяет явно задать приоритеты правил
        (objects (node ?x) (ako ?y&~nil))  # правило rule2 ищет такой факт objects, у которого слот ako не равен значению nil. Это реализуется специальным символом & со знаком отрицания ~
=>
        (printout t "A " ?x " is a kind of " ?y crlf)               # вывод на экран такого факта

)

(defrule rule3
        (declare (salience 10))      # данное ключевое слово позволяет явно задать приоритеты правил
        (objects (node ?x) (isa ?y&~nil))              # правило rule3 ищет такой факт objects, у которого слот isa не равен значению nil. Это реализуется специальным символом & со знаком отрицания ~
```

```
=>
        (printout t ?x " is an " ?y crlf)        # вывод на экран такого факта
)

(defrule rule4
        (declare (salience 15))      # данное ключевое слово позволяет явно задать приоритеты правил
        (objects (node ?x) (isa ?y&~nil) (can ?z&~nil))      # правило rule4 ищет такие факты objects, у которых слоты isa  и can не равны значению nil. Это реализуется специальным символом & со знаком отрицания ~

=>
        (printout t ?x " is a " ?y " and " ?z crlf)        # вывод на экран таких фактов
)

(defrule rule5
        (declare (salience 20))      # данное ключевое слово позволяет явно задать приоритеты правил
        (objects (node ?x) (have1 ?y&~nil) (can ?z&~nil))  # правило rule5 ищет такие факты objects, у которых слоты have1  и can не равны значению nil. Это реализуется специальным символом & со знаком отрицания ~
=>
        (printout t ?x " have a " ?y " and " ?z crlf)    # вывод на экран таких фактов
)
```

была разработана семантическая сеть, состоящая из вершин и соединяющих их стрелок-отношений, и была составлена программа на языке CLIPS, моделирующая разработанную семантическую сеть.

ЛАБОРАТОРНАЯ РАБОТА №4. МОДЕЛИРОВАНИЕ И СОЗДАНИЕ ПРОТОТИПА ЭКСПЕРТНОЙ СИСТЕМЫ

Построение экспертной оболочки для решения задачи классификации

Цель работы - ознакомление с методами построения экспертных систем, этапами их разработки.

Сведения из теории

Типы экспертных систем

Одним из часто употребляемых оснований классификации экспертных систем является тип решаемых ими задач.

Экспертные системы, применяемые для решения перечисленных типов задач, носят названия интерпретирующих, прогнозирующих и т.п.

Другим основанием классификации может служить тип модели предметной области.

Различают статические и динамические предметные области. Предметную область называют статической, если ее модель остается неизменной за все время решения задачи, т.е. остаются неизменными набор сущностей, их атрибуты, связи между ними и т. д.

Если это условие не выполняется, то предметную область называют динамической. В соответствии с этим экспертные системы делятся на статические и динамические.

Существует мнение, что решение ряда важнейших практических неформализованных задач возможно лишь с использованием динамических экспертных систем.

В настоящее время именно этот класс систем переживает период плодотворного развития.

Иногда, говоря о динамических экспертных системах, их называют системами реального времени. Хотя в общем случае это не совсем корректно, так как данный термин подразумевает использование другого основания классификации - время реакции системы. Так, например, при решении задач управления под системами реального времени понимают системы, обеспечивающие обработку поступающей информации и выдачу управляющих воздействий с задержкой, исключающей переход управляемого объекта в аварийное состояние.

Методы разработки экспертных систем

По опыту известно, что большая часть знаний в конкретной предметной области остается личной собственностью эксперта. И наибольшую проблему при разработке экспертной системы представляет процедура получения знаний у эксперта и занесения их в базу знаний, называемой извлечением знаний. Это происходит не потому, что он не хочет разглашать своих секретов, а потому, что он не в состоянии сделать этого - ведь эксперт знает гораздо больше, чем сам осознает.

Кроме того, обладая большими знаниями и опытом в своей предметной области, эксперт может не быть специалистом в области компьютеров и систем ИИ. Поэтому для выявления знаний эксперта и их формализации на протяжении всего периода разработки системы с ним взаимодействует инженер по знаниям.

В целом процесс разработки экспертной системы носит эволюционный характер.

На этапе идентификации инженер по знаниям и эксперт определяют цели и задачи построения экспертной системы, ее предметную область, необходимые для нее ресурсы (время, вычислительные средства). Они также указывают участников процесса создания системы (например дополнительных экспертов).

В ходе этапа концептуализации эксперт и инженер по знаниям выявляют основные понятия, отношения и характер информационных потоков, необходимые для описания процесса решения задач в данной предметной области.

На этапе формализации инженер по знаниям производит выбор инструментального средства разработки экспертных систем и при помощи эксперта представляет основные понятия и отношения в рамках некоторого формализма, задаваемого выбранным средством разработки.

В ходе этапа реализации эксперт осуществляет наполнение базы знаний, а инженер по знаниям комбинирует и реорганизует формализованное знание. Результатом этого этапа является программа-прототип, которую можно выполнять и подвергать контрольным испытаниям.

Наконец, в ходе испытания проводится оценка работы программы-прототипа. Как правило, эксперт дает оценку работы программы и помогает инженеру по знаниям в последующих ее модификациях.

Иногда к рассмотренным пяти этапам добавляют шестой - этап опытной эксплуатации, в ходе которого проверяется пригодность экспертной системы для конечных пользователей.

Перечисленные этапы создания экспертной системы не являются четко очерченными, детально определенными или даже независимыми между собой. В лучшем случае они грубо описывают сложный процесс извлечения знаний. На каждом из них возможен откат на несколько этапов назад. Таким образом, экспертная система эволюционирует, постепенно усложняя организацию и представление знаний. Время от времени, когда появляется необходимость в новых свойствах, которых нельзя достичь, исходя из возможностей существующей системы, происходит существенная реорганизация и перестройка всей ее архитектуры.

Инструментальные средства разработки экспертных систем

Различают следующие типы инструментальных средств разработки экспертных систем ЭС:

1. Языки программирования;
2. Языки представления знаний (языки инженерии знаний);
3. Средства автоматизации разработки (проектирования);
4. Оболочки ЭС.

Указанные типы инструментальных средств перечислены в порядке убывания эффективности, требуемой квалификации пользователя и трудозатрат, необходимых для создания ЭС. Действительно, при использовании инструментальных средств первого типа в задачу разработчика входит программирование всех компонентов ЭС на языке довольно низкого уровня. Использование инструментальных средств второго типа позволяет значительно

повысить уровень языка, что, как правило, приводит к некоторому снижению эффективности. Инструментальные средства третьего типа позволяют разработчику не программировать все или часть компонентов ЭС, а выбирать их из заранее составленного набора. При применении инструментальных средств четвертого типа разработчик ЭС полностью освобождается от работ по созданию про-грамм, так как берет готовую пустую ЭС.

Поэтому прототипы ЭС создаются с помощью оболочек или средств автоматизации разработки, так как основная цель на этой фазе - минимизация трудозатрат, а промышленные или коммерческие ЭС - на языках представления знаний или языках программирования, обеспечивающих более высокую эффективность.

Оболочки ЭС ориентированы на работу с пользователем-непрофессионалом в области программирования. Основным свойством оболочек является то, что они содержат все компоненты ЭС в готовом виде и их использование не предполагает программирования, а сводится лишь к вводу в оболочку знаний о проблемной области.

Каждая оболочка характеризуется фиксированным способом представления знаний и организации вывода и функционирования компонентов, которые будут использоваться во всех приложениях, где будет применяться оболочка.

Желание предоставить разработчику ЭС разнообразные средства для учета особенностей приложения привело к объединению в рамках одной системы различных методов решения задач, представления и интерпретации знаний.

В их состав могут входить средства модифицирования функционирования оболочки, набор компонентов, позволяющих конструировать собственные оболочки, средства комплексирования компонентов в виде языка высокого уровня, развитый графический интерфейс с пользователем. Подобные средства называют средствами автоматизации разработки (проектирования) ЭС. Они обычно позволяют представлять предметную область в виде сложной структуры, состоящей из произвольного числа понятий. Необходимость выбора средств, анализа их возможностей, модификации их функционирования и так далее требует от пользователя специальных знаний в области ИИ.

При разработке реальных экспертных систем в большинстве случаев используются так называемые языки искусственного интеллекта типа Лисп и Пролог. Однако в последнее время наметился переход к использованию языков традиционного программирования (C, C++ и тому подобное), что позволяет упростить интеграцию создаваемой экспертной системы в эксплуатируемые информационно-аналитические системы и снизить требования к быстродействию и объему памяти.

Порядок выполнения работы:

1. Построить действующий прототип экспертной системы для решения задачи классификации в определённой предметной области, являющейся экспертной оболочкой;

2. Составьте отчёт, содержащий алгоритм.

Пример:

Базовая структура экспертной системы показана на рис. 1. Перечисленные структурные элементы являются наиболее характерными для большинства экспертных систем, хотя в реальных условиях некоторые из них могут отсутствовать.

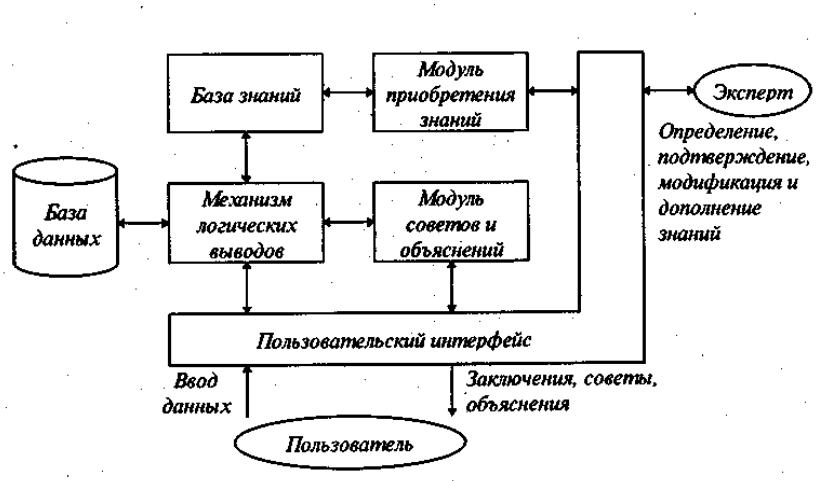


Рис. 1. Пример структуры экспертной системы

Контрольные вопросы

1. Назовите виды предметных областей?
2. Перечислите типы инструментальных средств разработки экспертных систем ЭС?
3. Назовите основное свойство оболочек?

ЛАБОРАТОРНАЯ РАБОТА №5. МОДЕЛИРОВАНИЕ И СОЗДАНИЕ ПРОТОТИПА ЭКСПЕРТНОЙ СИСТЕМЫ

Программные конструкции, реализующие модули базы данных и базы знаний

Цель работы - ознакомление с методами построения экспертных систем, этапами их разработки. Изучить структуру и рассмотреть способы программной реализации экспертных систем

Порядок выполнения работы:

1. Построить действующий прототип экспертной системы для решения задачи классификации в определённой предметной области, являющейся экспертной оболочкой;
Для разработки данной системы необходимо следующее:

a. Создать программные конструкции, реализующие модули, представленные на рис. №1. Возможно использование готовых алгоритмов и программ, выполненных в предыдущих работах;
b. Осуществить возможность наполнения базы данных, базы знаний и реализовать интерфейс данных модулей и эксперта;
c. Проанализировать выбранную предметную область, выделить и структурировать интересующую экспертную информацию. Определить модели для представления знаний;
d. Сформулировать 15-20 фактов и правил;
e. Заполнить экспертную систему знаниями;
f. Протестировать программное обеспечение.

2. Составьте отчёт, содержащий пояснения к работе программы и листинг программы.

Пример:

Листинг программы

```
(deftemplate node
    (slot name)      # слот отображает название узла дерева
    (slot type)      # тип (узел принятия решения decision или узел ответа – answer)
```

```
        (slot question)          # вопрос для узла принятия решения
        (slot yes-node)   # название узла, к которому мы должны перейти при выборе ответа «yes» на вопрос question
        (slot no-node)           # название узла, к которому мы должны перейти при выборе ответа «no» на вопрос question
        (slot any-node)   # название узла, к которому мы должны перейти при выборе ответа «any» на вопрос question
        (slot answer))           # слот отображает ответ, который дается данным деревом решений при осуществлении им классификации

(deffunction ask-yes-no-any (?question)
        (printout t ?question " (yes, no, any) ")
        (bind ?answer (read))
        (while (and (and (neq ?answer yes) (neq ?answer no)) (neq ?answer any))
                (printout t ?question " (yes, no, any) ")
                (bind ?answer (read)))
        (return ?answer))

(deffunction ask-yes-or-no (?question)
        (printout t ?question " (yes or no) ")
        (bind ?answer (read))
        (while (and (neq ?answer yes) (neq ?answer no))
                (printout t ?question " (yes or no) ")
                (bind ?answer (read)))
        (return ?answer))



(defrule initialize
        (not (node (name root)))
        =>
        (assert(node (name root) (type decision)
        (question "Are you lookung for a new book?")
        (yes-node node1) (no-node node2) (any-node node3)))

(assert(node (name node1) (type decision)
        (question "Is that book a fantasy?")
        (yes-node node4) (no-node node5) (any-node node3)))

(assert(node (name node2) (type decision)
        (question "Is that book a classic?")
        (yes-node node6) (no-node node7) (any-node node3)))

(assert(node (name node3) (type answer) (answer Dictionary)))

(assert(node (name node4) (type answer) (answer "Game of Thrones")))

(assert(node (name node5) (type answer) (answer "Purity")))

(assert(node (name node6) (type answer) (answer "Crime and Punishment")))

(assert(node (name node7) (type answer) (answer "Dandellion wine")))
```

```
(assert (current-node root)))


(defrule ask-decision-node-question
        ?node <- (current-node ?name)
                (node (name ?name)
                (type decision)
                (question ?question))
        (not (answer ?))
        =>
        (assert (answer (ask-yes-no-any ?question))))

(defrule proceed-to-yes-branch      # если в списке фактов имеется факт (answer yes)
        ?node <- (current-node ?name)
        (node (name ?name)
                (type decision)
                (yes-node ?yes-branch))
        ?answer <- (answer yes)
        =>
        (retract ?node ?answer)
        (assert (current-node ?yes-branch)))

(defrule proceed-to-no-branch       # в случае факта (answer no)
        ?node <- (current-node ?name)
        (node (name ?name)
                (type decision)
                (no-node ?no-branch))
        ?answer <- (answer no)
        =>
        (retract ?node ?answer)
        (assert (current-node ?no-branch)))

(defrule proceed-to-any-branch      # в случае факта (answer any)
        ?node <- (current-node ?name)
        (node (name ?name)
                (type decision)
                (any-node ?any-branch))
        ?answer <- (answer any)
        =>
        (retract ?node ?answer)
        (assert (current-node ?any-branch)))

(defrule ask-if-answer-is-correct
        ?node <- (current-node ?name)
        (node (name ?name) (type answer)
                (answer ?value))
                (not (answer ?))
        =>
        (printout t "I guess it is a " ?value crlf)
        (assert (answer (ask-yes-or-no "Am I correct?"))))
```

```
(defrule answer-node-guess-is-correct          # если пользователь согласен с предоставленным ответом
        ?node <- (current-node ?name)
        (node (name ?name) (type answer))
        ?answer <- (answer yes)
        =>
        (assert (ask-try-again))
        (retract ?node ?answer))

(defrule ask-try-again
        (ask-try-again)
        (not (answer ?))
        =>
        (assert (answer (ask-yes-or-no "Try again?"))))

(defrule one-more-time
        ?phase <- (ask-try-again)
        ?answer <- (answer yes)
        =>
        (retract ?phase ?answer)
        (assert (current-node root)))

(defrule no-more # если пользователь хочет завершить работу в системе, то происходит выход из CLIPS
        (ask-try-again)
        (answer no)
        =>
        (exit))
(defrule answer-node-guess-is-incorrect       # пользователь не согласен с полученным ответом
        ?node <- (current-node ?name)
        (node (name ?name) (type answer))
        ?answer <- (answer no)
        =>
        (assert (replace-answer-node ?name))          # внесение факта запускает следующее правило
        (retract ?node ?answer))

(defrule replace-answer-node
        ?phase <- (replace-answer-node ?name)
        ?data <- (node (name ?name)
                       (type answer)
                       (answer ?value))
        =>
        (retract ?phase)
        (printout t "What is the book? ")
        (bind ?new-book (read))
        (printout t "What question when answered yes ")
        (printout t "will distinguish " crlf " a ")
        (printout t ?new-book " from a " ?value "? ")
        (bind ?question (readline))
        (printout t "Now I can guess " ?new-book crlf)
        (bind ?newnode1 (gensym*))
        (bind ?newnode2 (gensym*))
        (modify ?data (type decision)
```

```
                (question ?question)
                (yes-node ?newnode1)
                (no-node ?newnode2))
(assert (node (name ?newnode1)
                (type answer)
                (answer ?new-book)))
(assert (node (name ?newnode2)
                (type answer)
                (answer ?value)))
(assert (ask-try-again)))
```

Было разработано дерево решений, в котором вопросы некоторых узлов принятия решений подразумевают ответы «да» или «нет», а вопросы других узлов принятия решений допускают выбор какого-либо ответа из множества возможных.

Была проведена необходимая модификация представленных компонентов программы-прототипа экспертной системы и составлена из этих компонентов программа.

Контрольные вопросы

1. Назовите виды предметных областей?
2. Перечислите типы инструментальных средств разработки экспертных систем ЭС?
3. Назовите основное свойство оболочек?

ЛАБОРАТОРНАЯ РАБОТА №6. ФОРМИРОВАНИЕ НЕЧЕТКИХ МНОЖЕСТВ
Составление алгоритма

Цель работы - получение навыков составления, логического вывода и ряда операций с нечёткими множествами.
Сведения из теории:

Процесс формирования ЛП включает в себя следующие этапы:

1. Определение множества термов ЛП и его упорядочивание.
2. Построение числовой области определения ЛП.
3. Выяснение схемы опроса экспертов и проведение опроса.
4. Построение функций принадлежности для каждого терма ЛП.

Этап 1 предполагает задание экспертом, формирующим ЛП, количества термов ЛП и названий соответствующих им нечетких переменных. Как правило, количество термов выбирается из диапазона п = 7±2. Названия нечетких переменных определяются, исходя из назначения ЛП.

На этапе 2 описывается универсальное множество U. Оно может быть, как числовым, так и нечисловым. Вид универсального множества ЛП зависит от описываемых объектов и определяет способ формирования функций принадлежности термов ЛП. Тем не менее для нечисловой ЛП все равно искусственно вводят числовое универсальное множество U’ÌÂ=(-¥;+¥) с целью использования единых подходов для работы с ЛП различных видов.

Этап 3 является ключевым при формировании ЛП. Существует четыре группы способов опросов экспертов: индивидуальный прямой опрос, индивидуальный косвенный опрос, групповой прямой опрос и групповой косвенный опрос. С точки зрения простоты организационной и программной реализаций предпочтительными являются индивидуальные способы опроса экспертов.

При прямом опросе эксперт непосредственно указывает все параметры функций принадлежности, недостатком здесь является проявление субъективизма в суждениях и необходимость знания экспертом основ нечеткой логики. При косвенном опросе функции принадлежности формируются на основе ответа эксперта на "наводящие" вопросы, при этом повышается объективность оценки и, как правило, не требуется знаний в нечеткой

логике, однако повышается риск несогласованности суждений эксперта. Групповые методы опроса обладают аналогичными свойствами, только результат формируется на основе агрегирования мнений нескольких экспертов, для чего также существуют и описаны в литературе соответствующие методы.

Порядок выполнения работы

1. Составить алгоритм, моделирующий процесс формирования нечёткого множества из вышеупомянутого примера с ростом людей.

2. Составьте отчёт, содержащий алгоритм.

Пример:

На практике наиболее часто используемым является индивидуальный косвенный опрос, основанный на процедуре вычисления собственного вектора матрицы парных сравнений, суть которого рассматривается ниже. Выполнение этого опроса составляет содержание четвертого этапа.

Формирование числовой ЛП можно описать следующими шагами:

1. Эксперт указывает количество термов ЛП -n.

2. Множество U делится на s»3n (до 30) равных отрезков, границы которых образуют упорядоченную последовательность {и1, и2, и3 ... us+l}, где u1=umin, uS+1=umax, U= [umin;umax].

3. Эксперт для каждого терма ЛП формирует матрицу парных сравнений, отвечая на вопросы: "Сравните принадлежность терму Tk значений иi и иj,, k = 1... п; i,j = 1 ... s+1».

4. Полученные для каждого терма векторы приоритетов {p1,p2,...pт+1} преобразуются так, чтобы pmax = 1.

5. Для каждого терма ЛП строится кусочно-линейная функция принадлежности, имеющая следующие точки: mTj(иi) = pi, i=1...s+1; j=1...n.

Если ЛП нечисловая, то эксперт выбирает п эталонных объектов (для каждого базового терма) и ставит им в соответствие числа из искусственного числового множества U', выбранного для ЛП (обычно это интервал [0;1]). В этих точках функции принадлежности соответствующих термов будут принимать единичные значения. Далее аналогично выполняются этапы 3 и 4, эксперт отвечает на вопрос: "Сравните сходство объектов f и g с объектом oi", где oi - эталонный объект терма Ti, i = 1 ... п».

При необходимости полученные кусочно-линейные функции принадлежности могут быть аппроксимированы некоторыми гладкими функциями. Обычно такая задача ставится при необходимости обеспечить компактность представления ЛП или минимизировать вычислительные затраты.

Контрольные вопросы

1.Что такое нечеткая логика?

2. Что лежит в основе понятия лингвистической переменной?

3.Каким условиям должны удовлетворять функции принадлежности нечетких множеств универсального множества, составляющие количественный смысл базовых термов ЛП?

4.Какие этапы включает в себя процесс формирования ЛП?

5.Какие существуют операции над нечеткими множествами?

ЛАБОРАТОРНАЯ РАБОТА №7. ФОРМИРОВАНИЕ НЕЧЕТКИХ МНОЖЕСТВ

Цель работы - получение навыков составления, логического вывода и ряда операций с нечёткими множествами.

Порядок выполнения работы

1. Написать программу, которая бы по введённому числовому значению роста человека его количественную характеристику (Высокий, Низкий, Очень высокий и т.д.)

2. Составьте отчёт, содержащий пояснения к работе программы и листинг программы.

Пример:

Листинг программы.

Данная программа по введенному числовому значению популярности книги выдает числовое значение ее стоимости.

```
(deftemplate Popularity # конструкция для нечеткого факта популярности
0 100
((low (Z 30 0)) # диапазон минимальных значений для популярности
(medium (PI 31 71)) # диапазон средних значений для популярности
(high (S 72 100)))) # диапазон максимальных значений для популярности
(deftemplate Price # конструкция для нечеткого факта стоимости
50 1000
((low (Z 300 50)) # диапазон минимальных значений для стоимости
(medium (PI 301 700)) # диапазон средних значений для стоимости
(high (S 701 1000)))) # диапазон максимальных значений для стоимости
(defrule getPopularity # получаем четкое значение популярности
(declare (salience 100)) # максимально допустимое значение популярности 100 %
=>
(printout t "Enter popularity rate (%): ") # ввод пользователем значения популярности книги, выраженного в процентах
(bind ?t (read))
(assert (popularity ?t)))
(defrule FuzzifyPopularity # определяем нечеткое множество
(popularity ?t)
=>
(bind ?t1 (- ?t 1))
(bind ?t2 (+ ?t 1))
(assert (Popularity (?t1 0) (?t 1) (?t2 0))))
(defrule SetPriceLow
(Popularity low)
=>
(assert (Price low)))
(defrule SetPriceMedium
(Popularity medium)
=>
(assert (Price medium)))
(defrule SetPriceHigh
(Popularity high)
=>
(assert (Price high)))
(defrule ShowPrice # дефаззифицируем полученное нечеткое множество, получая таким образом четкое значение стоимости
(declare (salience -100))
?f <- (Price ?p)
=>
(plot-fuzzy-value t "*" nil nil ?f)
(printout t "I set " (moment-defuzzify ?f) " rub price" crlf))
```

Контрольные вопросы

1.Что такое нечеткая логика?

2.Что лежит в основе понятия лингвистической переменной?

3.Каким условиям должны удовлетворять функции принадлежности нечетких множеств универсального множества, составляющие количественный смысл базовых термов ЛП?

4.Какие этапы включает в себя процесс формирования ЛП?

5.Какие существуют операции над нечеткими множествами?

ЛАБОРАТОРНАЯ РАБОТА №8. АЛГОРИТМ ПЕРСЕПТРОНА – ОБУЧАЕМЫЙ КЛАССИФИКАТОР ОБРАЗОВ

Цель работы – ознакомиться с алгоритмом Персептрона, построить решающую функцию по заданной выборке образов с помощью алгоритма Персептрона.

Сведения из теории

Персептрон (Perceptron) – простейший вид нейронных сетей. В основе лежит математическая модель восприятия информации мозгом, состоящая из сенсоров, ассоциативных и реагирующих элементов.

Решение задачи о разделении на два класса эквивалентно решению системы линейных неравенств. Так, если задано два множества образов ω_1 и ω_2, то решение ищется в виде вектора весов , обладающих следующими свойствами:

$$\overline{w}^T \overline{x} > 0, \text{ если } \overline{x} \in \omega_1 \text{ и } \overline{w}^T \overline{x} < 0, \text{ если } \overline{x} \in \omega_2 .$$

Если образы класса умножить на –1, то условие становится общим для всех образов. Если обозначить через N общее количество пополненных выборочных образов обоих классов, то данную задачу можно свести к нахождению (определению) вектора весов , удовлетворяющих системе неравенств:

$$\overline{x} \cdot \overline{w} > 0 \quad (1)$$

где $\overline{x}^T = [x_1, x_2, \ldots, x_n, x_{n+1}]$, $\overline{w}^T = [w_1, w_2, \ldots, w_n, w_{n+1}]$ и $x_{n+1} = 1$.

Данный алгоритм следует из модели процесса обучения машины для распознавания двух классов, предложенный Розенблаттом и известный в виде устройства с именем $\omega_i \quad (i = \overline{1, M})$ Персептрон.

Эта схема легко распространяется на случай разделения на несколько классов путем увеличения числа входящих в нее первичных элементов, т.е. если имеем , то:

$$R_i = \sum_{k=1}^{n+1} w_k^{(i)} x_k = \overline{w}^{(i)T} \cdot \overline{x} \qquad (i = \overline{1, M})$$

- решающие функции (2)

и $x \in \omega_i$ если $R_i > R_j \quad \forall j \neq i$.

Основная модель Персептрона может быть распространена и на случай нелинейных решающих функций введением соответствующих нелинейных преобразований.

Модель Персептрона вида (2) сводится к следующему алгоритму: Пусть заданы два обучающих множества из классов и ; a - начальный вектор весов, выбранный произвольно.

Тогда k-й шаг обучения будет иметь вид:

Если $\bar{x}(k) \in \omega_i$ и $\bar{w}^T(k)\bar{x}(k) \le 0$, то вектор весов $\bar{w}(k)$ заменяется:

$$\bar{w}(k+1) = \bar{w}(k) + C\bar{x}(k) \quad (3)$$

где: C – корректирующее приращение.

2. Если $\bar{x}(k) \in \omega_2$ и $\bar{w}^T(k)\bar{x}(k) \ge 0$, то $\bar{w}(k)$ заменяется вектором:

$$\bar{w}(k+1) = \bar{w}(k) - C\bar{x}(k) \quad (4)$$

3. В противном случае $\bar{w}(k)$ не изменяется, т.е.:

$$\bar{w}(k+1) = \bar{w}(k) \quad (5)$$

Иначе говоря, алгоритм вносит изменения в вектор весов тогда и только тогда, если образ, предъявленный на k-м шаге обучения был неправильно классифицирован с помощью соответствующего вектора весов.

Очевидно, что алгоритм Персептрона является процедурой типа «подкрепление-наказание», и поощрением является, по сути, отсутствие наказания. Таким образом:

1. если образ классифицирован правильно, то система подкрепляется тем, что в вектор $\bar{w}$ не вносится никаких изменений.

2. если образ классифицирован неправильно и $\bar{w}^T(k)\bar{x}(k) < 0$, когда должно быть > 0, то система «наказывается» увеличением значения весов $\bar{w}(k)$ на величину, пропорциональную $\bar{x}(k)$.

3. если $\bar{w}^T(k)\bar{x}(k) > 0$, когда оно должно быть < 0, то система наказывается противоположным образом.

Сходимость алгоритма наступает при правильной классификации всех образов с помощью некоторого вектора весов.

Изменяя в (3), (4) способ выбора корректирующего приращения C можно получить несколько модификаций алгоритма Персептрона среди которых выделяют:

1.алгоритм фиксированного приращения
2.алгоритм коррекции абсолютной величины
3.алгоритм дробной коррекции

Если воспользоваться эквивалентной формой представления алгоритма Персептрона, для чего следует умножить образы одного из классов на –1 (например, класс ω_2) в виде

$$\bar{w}(k+1) = \begin{cases} \bar{w}(k) & \text{если} \quad \bar{w}^T(k)\bar{x}(k) > 0 \\ \bar{w}(k) + C\bar{x}(k) & \text{если} \quad \bar{w}^T(k)\bar{x}(k) \le 0 \end{cases} \quad (6)$$

то все указанные модификации алгоритма Персептрона можно определить следующим образом:

1.коэффициент C = const >0, т.е. алгоритм фиксированного приращения

2.коэффициент C выбирается достаточно большим, что гарантирует правильную классификацию образов после коррекции коэффициентов весов, т.е. если $\overline{w}^T(k)\overline{x}(k) \le 0$, что соответствует неправильной классификации образов, то коэффициент C выбирается таким образом, чтобы

$$\overline{w}(k+1)\overline{x}(k) = [\overline{w}(k) + C\overline{x}(k)]^T \overline{x}(k) > 0 \quad (7)$$

Одним из способов, обеспечивающих выполнение (7) является выбор C согласно следующему соотношению:

$$C = \text{int}\frac{|\overline{w}(k)\overline{x}(k)|}{\overline{x}^T(k)\overline{x}(k)} \text{ при } \overline{w}(1) \neq 0 \qquad k = \overline{1, N}$$

где:

int(·) – наименьшее целое;

N – число образов в выборке.

Это дает в итоге алгоритм коррекции абсолютной величины.

Коэффициент C выбирается согласно соотношению, что делает алгоритм дробной коррекции:

$$C = \lambda \cdot \frac{|\overline{w}(k)\overline{x}(k)|}{\overline{x}^T(k)\overline{x}(k)}, \text{ где } 1 < \lambda < 2 \quad (8), \text{ где} \quad (8)$$

что обеспечивает правильную классификацию образов после каждой коррекции коэффициентов весов.

Порядок выполнения работы

1.Записать алгоритм обучения Персептрона для случайно выбранной цифры.

2.Составьте отчёт, содержащий алгоритм.

Пример:

Случайным образом выбираем цифру 5 и прогоняем ее через сеть, модифицируя ее веса.

Следовательно, мы должны учесть следующие важные моменты:

•Если наша нейросеть правильно распознала/отвергла цифру 5, то мы ничего не предпринимаем (все ведь замечательно!).

•Если нейросеть ошиблась и распознала неверную цифру как 5, то мы должны ее наказать – мы уменьшаем веса тех связей, через которые прошел сигнал. Другими словами, веса, связанные с возбудившимися входами, уменьшаются.

•Если нейросеть ошиблась и не распознала цифру 5, то мы должны увеличить все веса, через которые прошел сигнал. Таким образом мы как бы говорим сети, что такие связи, а значит и связанные с ними входы – правильные.

Теперь запишем непосредственно алгоритм обучения:

1.Подать на входы нейросети цифру в строковом формате.

2.Если цифра распознана/отвергнута верно, то перейти к шагу 1.

3.Если сеть ошиблась и распознала неверную цифру как 5, то вычесть из всех связей, связанных с возбудившимися элементами единицу.

4.Если сеть ошиблась и отвергла цифру 5, то добавить единицу ко всем связям, связанным с возбудившимися элементами.

Почему в алгоритме мы прибавляем или отнимаем именно 1. На самом деле эту величину можно задать любой. Понятно, что эта величина влияет на эффективность обучения.

Контрольные вопросы

1.Что лежит в основе Персептрона?

2.Какие выделяют модификации алгоритма Персептрона?

3.Когда наступает сходимость алгоритма?

ЛАБОРАТОРНАЯ РАБОТА №9. АЛГОРИТМ НАИМЕНЬШЕЙ СРЕДНЕКВАДРАТИЧЕСКОЙ ОШИБКИ (НСКО)

Цель работы: ознакомиться с процессом работы алгоритма наименьшей среднеквадратической ошибки и использовать полученные знания в определении разделимости классов.

Порядок выполнения работы:

1. Определите, обладают ли классы $\boldsymbol{\omega_1}: \{(\mathbf{0},\mathbf{0})', (\mathbf{1},\mathbf{1})'\}$ и $\boldsymbol{\omega_2}: \{(\mathbf{0},\mathbf{1})', (\mathbf{1},\mathbf{0})'\}$ линейной разделимостью при $\mathrm{c} = 1$ и $b(1) = (1,1,1,1)'$. Приведите решение;

2. Составьте отчёт, содержащий алгоритм НСКО и решение задачи из пункта (1) .

Пример:

Рассмотрим класс $\boldsymbol{\omega_1}$, содержащий образы $\{(\mathbf{0},\mathbf{0})', (\mathbf{0},\mathbf{1})\}$ и класс содержащий образы $\{(\mathbf{1},\mathbf{0})', (\mathbf{1},\mathbf{1})'\}$. Путем пополнения образов и умножения всех образов, входящих в класс $\boldsymbol{\omega_2}$ на —1 получим матрицу

$$\mathbf{X} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ -1 & 0 & -1 \\ -1 & -1 & -1 \end{pmatrix}.$$

Обобщенная обратная матрица $\boldsymbol{X'' = (X'X)^{-1}X'}$ равна

$$\mathbf{X}^{\#} = \frac{1}{2}\begin{pmatrix} -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & -1 \\ \frac{3}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

Положив и применив алгоритм (10), получим

$$\mathbf{w}(1) = \mathbf{X}^{\#}\mathbf{b}(1) = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}.$$

Так как

$$\mathbf{X}\mathbf{w}(1) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

из (1) следует, что вектор $\boldsymbol{w(1)}$ есть решение. Подтверждением этого служит и тот факт, что вектор ошибки $\boldsymbol{e(1)}$ равен нулю. Практически обычно считается, что алгоритм определил вектор решения $\boldsymbol{w(k)}$ если $\boldsymbol{Xw(k) > 0}$. Это условие сходимости алгоритма обычно выполняется раньше, чем достигается равенство (1).

Контрольные вопросы

1. В каких случаях сходятся модификации алгоритма Персептрона?
2. При выполнении, какого условия функция достигает своего минимального значения?

Пример тестирования

1. Какие из перечисленных функций относятся к интеллектуальным структурам?

   a. Структурирование данных.

   b. Распознавание образов.

   c. Манипуляция данными.

   d. Все вышеперечисленное.

2. Какая модель представления знаний использует правила вида "если-то" для описания знаний?

   a. Продукционная модель представления знаний.

   b. Иерархическая модель данных.

   c. Реляционная модель данных.

   d. Объектно-ориентированные модели данных.

3. Какие виды представления знаний существуют?

   a. Декларативное, процедурное и специальное представление знаний.

   b. Линейное, иерархическое и сетевое представление знаний.

   c. Изобразительное и текстовое представление знаний.

   d. Формальное и неформальное представление знаний.

4. Что означает распознавание образов в контексте искусственного интеллекта?

   a. Поиск и извлечение текстовой информации из изображений.

   b. Определение, классификация и анализ объектов на основе их визуальных характеристик.

   c. Определение эмоционального состояния человека на фотографии.

   d. Анализ музыкальных произведений.

5. Какие классификации систем распознавания существуют?

   a. Двухуровневая и многоуровневая системы распознавания.

   b. Реляционные и иерархические системы распознавания.

   c. Статические и динамические системы распознавания.

   d. Визуальные и аудио системы распознавания.

6. Что представляет собой интеллектуальный банк данных в системах искусственного интеллекта?

a. Центральное хранилище фотографий и изображений.

b. Централизованная база данных с информацией о пользовательских предпочтениях.

c. Хранилище структурированных и неструктурированных данных для поддержки интеллектуальных решений.

d. Библиотека искусственных нейронных сетей.

7. Какие особенности характеризуют многоуровневую систему распознавания образов?

a. Использование одного уровня анализа.

b. Использование разных методов на каждом уровне анализа.

c. Отсутствие необходимости в классификации образов.

d. Отсутствие необходимости в обучении системы.

8. Какова структура и функции интеллектуальной системы?

a. Система для сбора и хранения данных.

b. Система для генерации случайных чисел.

c. Система для обработки данных и принятия решений на основе знаний.

d. Программа для проведения видеоконференций.

9. Какое понятие описывает систему искусственного интеллекта (СИИ)?

a. Автоматическую систему обработки данных.

b. Систему для создания искусственных объектов.

c. Систему, способную к самообучению.

d. Систему для обработки текстовой информации.

10. Какие виды распознающих систем существуют?

a. Распознающие системы без обучения, с обучением и с самообучением.

b. Только распознающие системы без обучения.

c. Только распознающие системы с обучением.

d. Только распознающие системы с самообучением.

Правильные ответы:

1. d
2. a
3. a
4. b
5. a
6. c
7. b
8. c
9. c
10. a

Подготовка к курсовой работе

Тематика курсовой работы: «Алгоритм распознавания образов».

Студентам выдается индивидуальное задание на курсовой работу. Цель курсовой работы – получение знаний по основам алгоритмов распознавания образов. Учебным планом предусматривается время на выполнение в объеме 36 часов.

Курсовая работа связана с изучением производственной, научной, учебной, нормативной и другой литературы и с выполнением необходимых расчетов.

Тематика курсовой работы определяется общей направленностью подготовки студента и может касаться объекта исследования будущей ВКР.

Обязательным элементом курсовой работы является расчетно-пояснительная записка. Расчетно-пояснительная записка включает такие структурные части:

- титульный лист;

- задание на курсовую работу;

- содержание;

- введение;

- основная (расчетная) часть;

- обсуждение полученных результатов;

- заключение (выводы);

- список использованной литературы и нормативных источников.

Задание на курсовую работу выдается преподавателем, ведущим данную дисциплину. Материалы курсового проекта могут быть использованы при выполнении выпускной квалификационной работы.

Подробное описание требований к содержанию и оформлению курсового проекта представлено в Методических указаниях к выполнению курсовой работы по дисциплине.

Контроль выполнения данного вида самостоятельной работы осуществляется во время консультаций (контактная внеаудиторная работа) в течение семестра и в форме защиты курсовой работы (промежуточный контроль).

Методические рекомендации по выполнению курсовой работы. Пример технического задания и структуры курсовой работы

Целью курсовой работы (КР) является закрепление знаний, полученных студентами в процессе изучения данной дисциплины;

Основной задачей курсовой работы является подготовка бакалавров к выполнению выпускной квалификационной работы.

Курсовой работа состоит из пояснительной записки с расчетной частью. Пояснительная записка состоит из трех основных частей: структуры алгоритма, математического обеспечения алгоритма и программной реализации алгоритма.

Изложение пояснительной записки должно быть технически грамотным, четким и сжатым и строиться на фактическом материале.

Порядок оформления Курсовой Работы следующий:

- Титульный лист
- Реферат (должен быть на одном листе; нумеруется, но номер страницы не проставляется)
- Содержание
- Введение
- Структура алгоритма
- Математическое обеспечение алгоритма
- Программная реализация алгоритма (на любом языке программирования, которым владеет студент)
- Заключение
- Список используемых источников

Список вопросов для промежуточного контроля (зачет)

1. Основные функции интеллектуальных структур.
2. Продукционная модель представления знаний.
3. Понятие декларативного, процедурального и специального представления знаний.
4. Распознавание образов. Основные понятия и методология решения.
5. Классификация систем распознавания. Одноуровневая система распознавания.
6. Интеллектуальный банк данных в СИИ.
7. Многоуровневая система распознавания образов.
8. Структура и функции интеллектной системы.
9. Понятие системы искусственного интеллекта (СИИ). Структурная схема СИИ.
10. Распознающие системы (РС) без обучения, с обучением.
11. Распознающие системы с самообучением.
12. Алгоритм наименьшей среднеквадратической ошибки (НСКО).
13. Решающие функции для автоматической классификации образов. Пример разбиения, когда каждый класс отделяется от всех остальных классов одной разделяющей поверхностью.
14. Язык PDL для описания изображений.
15. Выбор непроизводных элементов при лингвистическом анализе изображений. Использование грамматик и языков для лингвистического (структурного) описания объектов.

Список вопросов для промежуточного контроля (экзамен)

1. Основные функции интеллектуальных структур.
2. Продукционная модель представления знаний.
3. Понятие декларативного, процедурального и специального представления знаний.
4. Распознавание образов. Основные понятия и методология решения.
5. Классификация систем распознавания. Одноуровневая система распознавания.
6. Интеллектуальный банк данных в СИИ.
7. Многоуровневая система распознавания образов.
8. Структура и функции интеллектной системы.
9. Понятие системы искусственного интеллекта (СИИ). Структурная схема СИИ.
10. Распознающие системы (РС) без обучения, с обучением.
11. Распознающие системы с самообучением.
12. Алгоритм наименьшей среднеквадратической ошибки (НСКО).
13. Решающие функции для автоматической классификации образов. Пример разбиения, когда каждый класс отделяется от всех остальных классов одной разделяющей поверхностью.
14. Язык PDL для описания изображений.
15. Выбор непроизводных элементов при лингвистическом анализе изображений. Использование грамматик и языков для лингвистического (структурного) описания объектов.
16. Решающие функции для автоматической классификации образов. Рассмотреть случай, когда классы попарно разделимы.
17. Кластерный анализ с использованием функций расстояния. Виды расстояния. Пороговый алгоритм выявления кластеров.
18. Структурный (лингвистический) метод распознавания образов. Процедура распознавания.
19. Байесовский классификатор на базе аппроксимации плотностей распределения, используя многочлены Эрмита.
20. Алгоритм максиминного расстояния при автоматической классификации образов.
21. Алгоритм внутригрупповых средних при автоматической классификации образов.
22. Интеллектуальный банк данных в системах искусственного интеллекта (СИИ).
23. Аппроксимация плотностей распределения функциями при вероятностной задаче распознавания.
24. Персептрон Розенблатта. Принцип работы.
25. Обучающий алгоритм Персептрона как пример детерминированной задачи распознавания образов.
26. Байесовский классификатор для образов, характеризующийся нормальным распределением.
27. Общие сведения о моделях представления знаний в СИИ.