

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

УТВЕРЖДЕНО

решением Учёного совета факультета математики,
информационных и авиационных технологий

от «21» июня 2020 г., протокол № 5/20

Председатель

/ М.А. Волков

«21» июня 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина	Интеллектуальные системы и технологии
Факультет	Факультет математики, информационных и авиационных технологий
Кафедра	Телекоммуникационные технологии и сети
Курс	4 (очная), 4 (заочная)

Направление (специальность) 09.03.02 - "Информационные системы и технологии"

код направления (специальности), полное наименование

Направленность (профиль/специализация) Разработка информационных систем

ПОЛНОЕ НАИМЕНОВАНИЕ

Форма обучения очная, заочная

очная, заочная, очно-заочная (указать только те, которые реализуются)

Дата введения в учебный процесс УлГУ:

«1» сентября 2020г.

Программа актуализирована на заседании кафедры: протокол № _____ от 20 ____ г.

Программа актуализирована на заседании кафедры: протокол № _____ от _____ 20____ г.

Программа актуализирована на заседании кафедры: протокол № _____ от 20 ____ г.

Сведения о разработчиках:

ФИО	Кафедра	Должность, ученая степень, звание
Липатова Светлана Валерьевна	Телекоммуникационных технологий и сетей	доцент, к.т.н., доцент

СОГЛАСОВАНО	СОГЛАСОВАНО
Заведующий кафедрой телекоммуникационных технологий и сетей, реализующей дисциплину	Заведующий выпускающей кафедрой телекоммуникационных технологий и сетей
 _____ / Смагин А.А. / Подпись ФИО «21» июня 2020 г.	 _____ / Смагин А.А. / Подпись ФИО «21» июня 2020 г.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ:

Цели освоения дисциплины: формирование общекультурных и профессиональных компетенций, необходимых для реализации информационно-аналитической и научно-исследовательской деятельности

Задачи освоения дисциплины: приобретение в рамках освоения предусмотренного курсом занятий следующих знаний, умений и навыков, характеризующих определённый уровень сформированности компетенций (см. подробнее п.3):

- сформировать системное базовое представление, первичные знания, умения и навыки студентов по основам инженерии знаний и нейроинформатики,
- дать общие представления о прикладных системах искусственного интеллекта,
- дать представление о роли искусственного интеллекта и нейроинформатики в развитии информатики в целом, а также, в научно-техническом прогрессе,
- подготовить студентов к применению концепций интеллектуальных систем при дальнейшем обучении.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП:

Дисциплина «Интеллектуальные системы и технологии» относится к числу дисциплин вариативной части блока Б1.В.ОД, предназначенного для студентов, обучающихся по направлению: 09.03.02 "Информационные системы и технологии".

Для успешного изучения дисциплины необходимы знания и умения, приобретенные в результате освоения курсов «Информатика и программирование»; «Дискретная математика и математическая логика», и полностью или частично сформированные компетенции ОПК-1, УК-1, а именно:

- **знать:** основные понятия, утверждения, а так же методы исследования, методику построения различных дискретных структур, новейшие достижения дискретной математики, основные принципы программирования;
- **уметь:** применять методы дискретной математики на практике, работать в средах программирования;
- **владеть:** методологией и навыками решения научных и практических задач, писать программы на языках высокого уровня.

Основные положения дисциплины используются в дальнейшем при изучении таких дисциплин как: «Экспертные системы», «Системы принятия решений», «Преддипломная практика».

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
ОПК-2 Способен использовать современные информационные	знать: – о истории, целях и задачах исследований в области искусственного интеллекта, – об областях применения интеллектуальных систем, – основные понятия нечетких вычислений,

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	– об основных направлениях в исследованиях новых архитектур компьютеров, – об основных этапах развития робототехники, – понятия инженерии знаний и нейрокибернетики, – прикладных системах искусственного интеллекта, уметь: – свободное использование терминологии как на русском, так и на английском языке (название операторов языка программирования, заимствованной терминологии) – осуществлять анализ предметной области, структурировать и формализовывать знания экспертной и их опыт; владеть: – навыками использования систем разработки интеллектуальных систем
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	знать: – о нечеткости знаний, ее природе и разновидностях, – основные моделей нейронных сетей, методы и алгоритмов их обучения, – проблемах и основных методах представления и обработки знаний, уметь: – ориентироваться в различных типах интеллектуальных систем, – ставить задачу построения экспертной системы для решения задачи выбора вариантов в плохо формализуемой предметной области, владеть: – методами представления и обработки знаний, – навыками формализации знаний экспертов с применением различных методов представления знаний,
ПК-1 Способен проводить предпроектное обследование объекта автоматизации, системный анализ предметной области, анализ научно-технической информации, отечественного и зарубежного опыта по тематике исследования	знать: – о двух подходах к построению интеллектуальных систем – логическом и нейрокибернетическом, эволюционном, – этапы построения экспертных систем, – языках программирования искусственного интеллекта; – о принципах использования генетических алгоритмов, – понятия генетических алгоритмов, – о структуру экспертных систем и их архитектурные особенности в зависимости от особенностей решаемой задачи, – о проблемах и способах построения нейронных сетей, уметь: – ориентироваться в различных методах представления знаний, владеть: – навыками разработки продукционные базы знаний для решения задач задачи выбора вариантов в плохо формализуемой предметной области,

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

	– навыками разработки онтологий; – навыками логического программирования; – навыками использования нейронных сетей, эволюционных методов; – навыками нечеткого моделирования.
--	--

4. ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ

4.1. Объем дисциплины в зачетных единицах (всего) 6

4.2. Объем дисциплины по видам учебной работы (в часах)

Форма обучения очная

Вид учебной работы	Количество часов (форма обучения <u>очная</u>)	
	Всего по плану	В т.ч. по семестрам
		7 семестра
1	2	3
Контактная работа обучающихся с преподавателем	90	90
Аудиторные занятия:	90	90
Лекции	36	36
практические и семинарские занятия	18	18
лабораторные работы (лабораторный практикум)	36	36
Самостоятельная работа	90	90
Текущий контроль	Тестирование, контрольная работа (решение задач)	Тестирование, контрольная работа (решение задач)
Курсовая работа	-	-
Виды промежуточной аттестации (экзамен, зачет)	экзамен (36)	экзамен (36)
Всего часов по дисциплине	216	216

Форма обучения заочная

Вид учебной работы	Количество часов (форма обучения <u>заочная</u>)	
	Всего по плану	В т.ч. по курсам
		7 семестра
1	2	3
Контактная работа обучающихся с преподавателем	22	22
Аудиторные занятия:	22	22
Лекции	10	10
практические и семинарские занятия	8	8
лабораторные работы (лабораторный практикум)	4	4
Самостоятельная работа	185	185
Текущий контроль	Тестирование,	Тестирование,

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

	контрольная работа (решение задач)	контрольная работа (решение задач)
Курсовая работа	-	-
Виды промежуточной аттестации (экзамен, зачет)	экзамен (9)	экзамен (9)
Всего часов по дисциплине	216	216

4.3. Содержание дисциплины (модуля.) Распределение часов по темам и видам учебной работы:
Форма обучения очная

Название разделов и тем	Всего	Виды учебных занятий					Форма текущего контроля знаний
		Аудиторные занятия			Занятия в интерактивной форме	Самостоятельная работа	
		Лекции	Практические занятия, семинары	Лабораторные работы, практикумы			
1	2	3	4	5	6	7	
1. Философские вопросы ИИ	12	2	2	-	-	5	тестирование
2. Подходы и направления исследований в ИИ	14	4	2	-	2	5	тестирование
3. Представление знаний	34	4	2	12	4	10	Проверка решения задач
4. Онтологии и Semantic Web	46	6	2	12	4	20	тестирование
5. Эволюционное моделирование	22	4	2	-	4	10	Проверка решения задач
6. Нечеткие вычисления.	24	6	2	-	4	10	Проверка решения задач
7. Нейронные сети.	64	10	6	12	4	30	Проверка решения задач
Итого	216	36	18	36	22	90	36

Форма обучения заочная

Название разделов и тем	Всего	Виды учебных занятий					Форма текущего контроля знаний
		Аудиторные занятия			Занятия в интерактивной	Самостоятельная работа	
		Лекции	Практические занятия, семинары	Лабораторные работы, практику			

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

				мы	форме		
1	2	3	4	5	6	7	
1. Философские вопросы ИИ	17	1	-	-	-	15	тестирование
2. Подходы и направления исследований в ИИ	17	1	-	-	-	15	тестирование
3. Представление знаний	28	1	1	-	-	25	Проверка решения задач
4. Онтологии и Semantic Web	41	2	1	2	2	35	тестирование
5. Эволюционное моделирование	28	1	1	-	-	25	Проверка решения задач
6. Нечеткие вычисления.	38	1	1	-	-	35	Проверка решения задач
7. Нейронные сети.	47	3	4	2	2	35	Проверка решения задач
Итого	216	10	8	4	4	185	9

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Тема 1. Философские вопросы ИИ.

Определения естественного и искусственного интеллекта. Тест Тьюринга. Мысленный эксперимент «Китайская комната». Теорема Геделя. Технологическая сингулярность. Цели науки ИИ. Научная этика в ИИ.

Тема 2. Подходы и направления исследований в ИИ.

Предыстория, история развития искусственного интеллекта как научного направления. Нейрокибернетика и кибернетика «черного ящика». История развития искусственного интеллекта в России. Основные направления исследований в области искусственного интеллекта. Свойства интеллектуальных информационных систем.

Тема 3. Представление знаний.

Данные и знания, свойства и виды знаний. Модели представления знаний: логическая, фреймовая, продукционная и семантические сети. Их достоинства, недостатки, основные принципы построения и область использования.

Тема 4. Онтологии и Semantic Web.

Определение онтологий. Стек протоколов Semantic Web. Основные элементы онтологий: классы, индивиды, свойства, аксиомы. Запросы к онтологиям. Использование правил в онтологиях. Семантические машины вывода.

Тема 5. Эволюционное моделирование.

Эволюционное моделирование. Определение и основные понятия генетического алгоритма. Операторы кроссовер, мутация, и инверсия. Фито-функция. Методы отбора

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

особей. Виды генетического алгоритма. Задачи, решаемые генетическим алгоритмом.

Тема 6. Нечеткие вычисления.

Теория нечётких множеств. Понятие нечеткого множества. Функция принадлежности. Операции с нечеткими множествами. Нечеткие отношения. Лингвистическая переменная. Нечеткие высказывания. Нечеткая импликация.

Тема 7. Нейронные сети.

Понятие нейрона. Модель математического нейрона. Персептрон Розенблатта. Правила Хебба. Алгоритм обучения по дельта-правилу. Проблема «исключающего или». Обучение многослойной нейронной сети методом обратного распространения ошибки. Классификация нейронных сетей. Задачи, решаемые нейронными сетями. Глубинное обучение. Свёрточные нейронные сети. Рекуррентные нейронные сети. Автокодировщик. Ограниченная машина Больцмана. Инициализация весов НС. Нормализация.

6. ТЕМЫ ПРАКТИЧЕСКИХ И СЕМИНАРСКИХ ЗАНЯТИЙ

Тема 1. Философские вопросы ИИ.

- 1) Какие угрозы видят ученые в развитии технологий ИИ?
- 2) Что такое технологическая сингулярность?
- 3) В чем заключался тест Тьюинга?
- 4) Какие задачи стоят перед современной наукой в области ИИ?
- 5) Как интерпретируют теорему Геделя по отношению к ИИ?
- 6) Как определяют предмет исследований науки об ИИ?
- 7) Какие задачи являются интеллектуальными?

Тема 2. Подходы и направления исследований в ИИ.

- 1) Какие задачи решает компьютерная лингвистика?
- 2) Какие виды анализа выполняют системы при машинном переводе?
- 3) Какие поколения роботов существуют?
- 4) Охарактеризуйте японский проект компьютер 5-го поколения?
- 5) На какие направления исследований делится искусственная жизнь?
- 6) Приведите примеры компьютерных игр с элементами ИИ и какие методы ИИ в них использовались?
- 7) Чем отличается слабый ИИ от сильного?

Тема 3. Представление знаний.

- 1) Достоинства и недостатки продукционной, семантической и фреймовой моделей?
- 2) Что из себя представляет Rete-алгоритм?
- 3) Что такое база знаний?
- 4) Примеры слабоструктурированных предметных областей?
- 5) Какие свойства отличают знания от данных?
- 6) Какого типа знания бывает?
- 7) Виды моделей представления знаний?

Тема 4. Онтологии и Semantic Web.

- 1) Для чего предназначена онтология?
- 2) Как соотносятся протоколы Semantic Web с семиуровневой моделью OSI?
- 3) Чем отличается Data Properties от Object Properties?
- 4) Какие запросы можно выполнить к онтологии?
- 5) Что из себя представляет триплет RDF?
- 6) Как объединяются онтология и правила?
- 7) Что такое слияние и выравнивание онтологий?

Тема 5. Эволюционное моделирование.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

- 1) Достоинства и недостатки эволюционных методов?
- 2) Какие существуют операторы ГА?
- 3) Приведите примеры оператора кроссовера?
- 4) Приведите примеры оператора мутации?
- 5) Что из себя представляет островная модель ГА?
- 6) Как выполняется турнирный отбор?
- 7) Что подразумевает принцип элитизма?

Тема 6. Нечеткие вычисления.

- 1) Как задается нечеткое множество?
- 2) Какие значения может принимать функция принадлежности?
- 3) Приведите пример нечеткой переменной.
- 4) Приведите пример лингвистической переменной?
- 5) Как проверяется полнота нечеткой базы знаний?
- 6) Что такое фаззификация и дефаззификация?
- 7) Какие методы применяются на этапе аккумуляции?

Тема 7. Нейронные сети.

- 1) Какие достоинства и недостатки у НС?
- 2) Как инициализируют синаптические веса НС?
- 3) Какие задачи можно решать с помощью НС?
- 4) Что из себя представляет пакетная нормализация?
- 5) Как выполняется операция свёртка в сверточной НС?
- 6) Как используется градиент в обучении НС?
- 7) Какие нейронные сети относятся к глубинным?

7.ЛАБОРАТОРНЫЕ РАБОТЫ, ПРАКТИКУМЫ

Тема 3. Представление знаний.

Лабораторная работа

Задание 1. Построить базу фактов (не менее 50), используя сервис SWI-Prolog (<https://swish.swi-prolog.org/>) или аналогичный offline приложения согласно варианту. Сформулировать не менее 5 правил (хотя бы одно с использованием рекурсии). Сформулировать не менее 5 запросов с интерпретацией их на естественном языке.

Варианты:

- 1) Родословная Романовых.
- 2) Царства в биологии.
- 3) Географические объекты (страны, столицы, материки, границы государств).
- 4) Состав компьютера.
- 5) Компьютерные сети.
- 6) Организационная структура университета.
- 7) Субъекты РФ.
- 8) Устройство автомобиля.
- 9) Родственные отношения (сваты, зять, свекры и т.д.)
- 10) Свободный вариант (студент предлагает тему и согласует ее с преподавателем).

Задание 2. Решить логическую задачу на прологе согласно варианту.

Варианты:

- 1) Трое, назовем их А, Б и В оказались вместе: один из России, другой из Финляндии, третий из Англии. Один из них увлекается математикой, другой - астрономией, а третий - литературой.
 - а. А живет не в России, Б - не в Финляндии
 - б. Тот, кто из России равнодушен к математике, а финн любит астрономию.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

с. Для Б не интересна литература.

Чем увлекается В и из какой он страны?

2) По древнему поверью, у каждого месяца есть свой камень-талисман. Так, июню, июлю и сентябрю соответствуют камни рубин, сапфир и жемчуг. Эти камни означают мудрость, здоровье и благополучие. У какого месяца какой камень-талисман и что он означает, если известно, что:

- жемчуг и рубин не соответствуют сентябрю;
- в июне и июле мудрости не наблюдается;
- здоровье не соответствует рубину;
- благополучие не относится к июню.

3) Три друга – Петр, Роман и Сергей учатся на математическом, физическом и химическом факультетах университета.

Если Петр математик, то Сергей не физик.

Если Роман не физик, то Петр – математик.

Если Сергей не математик, то Роман – химик.

Определите специальность Сергея.

4) На новогодний праздник три друга – Евгений, Николай, Алексей, выбрали себе костюмы трех богатырей: Ильи Муромца, Алеши Поповича, Добрыни Никитича.

Известно, что:

- Евгений – самый высокий.
- Выбравший костюм Добрыни Никитича меньше ростом, чем выбравший костюм Ильи Муромца.
- Алексею не подошел костюм Добрыни Никитича.
- Ни у одного из друзей имя не совпадает с именем богатырей, выбранных костюмов.

Какой костюм выбрал каждый из друзей?

5) Три друга заняли первое, второе и третье места в соревнованиях универсиады. Друзья — разной национальности, зовут их по-разному и любят они разные виды спорта.

Майкл предпочитает баскетбол и играет лучше чем американец. Израильянин Саймон играет лучше теннисиста. Игрок в крикет занял первое место.

Кто является австралийцем? Каким видом спорта занимается Ричард?

6) Один из пяти братьев разбил окно.

Андрей сказал: Это или Витя, или Коля .

Витя сказал: Это сделал не я и не Юра .

Дима сказал: Нет, один из них сказал правду, а другой неправду .

Юра сказал: Нет, Дима ты не прав .

Их отец, которому, конечно можно доверять, уверен, что не менее трех братьев сказали правду.

Кто разбил окно?

7) В автомобильных гонках три первых места заняли Алеша, Петя и Коля. Какое место занял каждый из них, если Петя занял не второе и не третье место, а Коля - не третье?

8) Витя, Юра и Миша сидели на скамейке. В каком порядке они сидели, если известно, что Миша сидел слева от Юры, а Витя слева от Миши.

9) Трое ребят вышли гулять с собакой, кошкой и хомячком. Известно, что Петя не любит кошек и живет в одном подъезде с хозяйкой хомячка. Лена дружит с Таней, гуляющей с кошкой. Определить, с каким животным гулял каждый из детей.

10) Свободный вариант (студент предлагает тему и согласует ее с преподавателем).

Отчет по лабораторной должен содержать описание программ, запросов и результатов (отчет можно сдавать в электронном виде).

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Информация по языку Пролог: <https://coollib.com/b/186558#nav>.

Теоретические основы

Наиболее известным языком декларативного (логического) программирования, реализующим модифицированную логику предикатов первого порядка, является PROLOG (англ. PROgramming LOGic - логическое программирование).

В 1965 году в работе «A machine oriented logic based on the resolution principle»³, опубликованной в 12 номере журнала «Journal of the ACM», Дж Робинсон представил метод автоматического поиска доказательства теорем в исчислении предикатов первого порядка, получивший название «**принцип резолюции**». На самом деле, идея данного метода была предложена Эрбраном в 1931 году, когда еще не было компьютеров (Herbrand, «Une methode de demonstration», These, Paris, 1931). Робинсон модифицировал этот метод так, что он стал пригоден для автоматического (компьютерного) использования и разработал эффективный алгоритм унификации, составляющий базис его метода.

Идеи использования логики в качестве языка программирования зародилась в начале 1970-х годов. Первыми исследователями, которые занялись разработкой этой идеи, были Роберт Ковальски (Robert Kowalski) из Эдинбурга (теоретические основы, статьи 1971 и 1974 г.), Маартен ван Эмден (Maarten van Emden) из Эдинбурга (экспериментальная демонстрационная система) и Ален Колмероз (Alain Colmerauer) из Марселя (реализация, 1973 г.). В 1973 году «группа искусственного интеллекта» во главе с Аленом Колмероз создала в Марсельском университете программу, предназначенную для доказательства теорем. Эта программа использовалась при построении систем обработки текстов на естественном языке. Программа доказательства теорем получила название Prolog (фр. PROgrammation en LOGique) и послужила прообразом Пролога. Ходят легенды, что автором этого названия была жена Алена Колмероз. Программа была написана на Фортране и работала довольно медленно.

Популяризации Пролога во многом способствовали:

- эффективная реализация (интерпретатор/компилятор) этого языка для ЭВМ DEC-10 Дэвидом Д. Г. Уорреном (David D.H. Warren) из Эдинбурга в 1977 г. Послужила прототипом для многих последующих реализаций Пролога. Что интересно, компилятор был написан на самом Прологе. Эта реализация Пролога, известная как «эдинбургская версия», фактически стала первым и единственным стандартом языка;
- разработка Кларком и Маккейбом (Великобритания) в 1980 году версии для персональных ЭВМ;
- японский проект создания компьютеров V поколения. В конце 1978 г. Министерство внешней торговли и промышленности (МВТП) Японии поручило разработать проект интеллектуальных ЭВМ, специально созданному для этих целей Токийскому институту вычислительной техники (ICOT)⁴. Сердцем этих компьютеров должен был стать не арифметический процессор, а специально оптимизированный для работы с прологоподобными программами.

В 1995 году был опубликован официальный стандарт ISO⁵/IEC⁶ языка Пролог (ISO/IEC 13211-1 «Information technology - Programming languages - Prolog - Part 1: General core» - «Информационные технологии. Языки программирования. Пролог. Часть 1. Общее ядро»).

На сегодня существует довольно много реализаций Пролога. Наиболее известные из них следующие: BinProlog, AMZI-Prolog, Arity Prolog, CProlog, Micro Prolog, МПролог, Prolog-2, Quintus Prolog, SICTUS Prolog, Silogic iis Workbench, Strawberry Prolog, SWI-Prolog, Turbo Prolog (PDC Prolog, Visual Prolog), UNSW Prolog и т. д. В нашей стране были

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

разработаны такие версии Пролога как Пролог-Д (С. Григорьев), Акторный Пролог (А. Морозов), а также Флэнг (А. Манцивода, В. Петухин).

SWI-Prolog является свободно распространяемой (англ. Free Software¹) реализацией (диалектом) языка программирования Пролог, которая практически полностью соответствует стандарту ISO²/IEC³ 13211-1 «Information technology - Programming languages - Prolog - Part 1: General core» (рус. «Информационные технологии. Языки программирования. Пролог. Часть 1. Общее ядро»).

SWI-Prolog развивается с 1987 года. Его создателем и основным разработчиком является Ян Вьелемакер (Jan Wielemaker). Название SWI происходит от Sociaal-Wetenschappelijke Informatica (гол. социально-научная информатика), первоначального названия группы в Амстердамском университете, где работает Вьелемакер.

SWI-Prolog позволяет разрабатывать приложения любой направленности, включая Web-приложения и параллельные вычисления, но основным направлением использования является разработка экспертных систем, программ обработки естественного языка, обучающих программ, интеллектуальных игр и т.п.

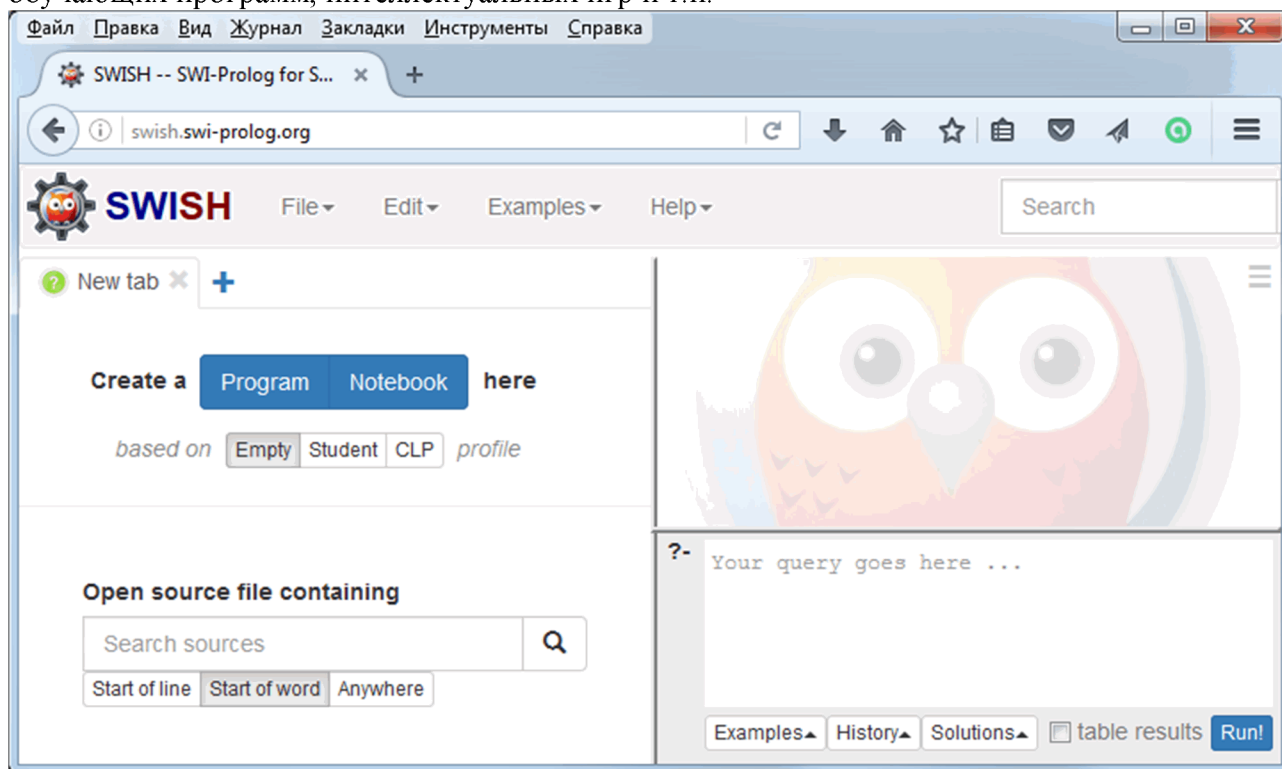


Рис.1. Стандартная online-среда программирования SWI-Prolog

Для перехода в режим редактирования и исполнения программ необходимо нажать на кнопку «Program» (см. рис.1).

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

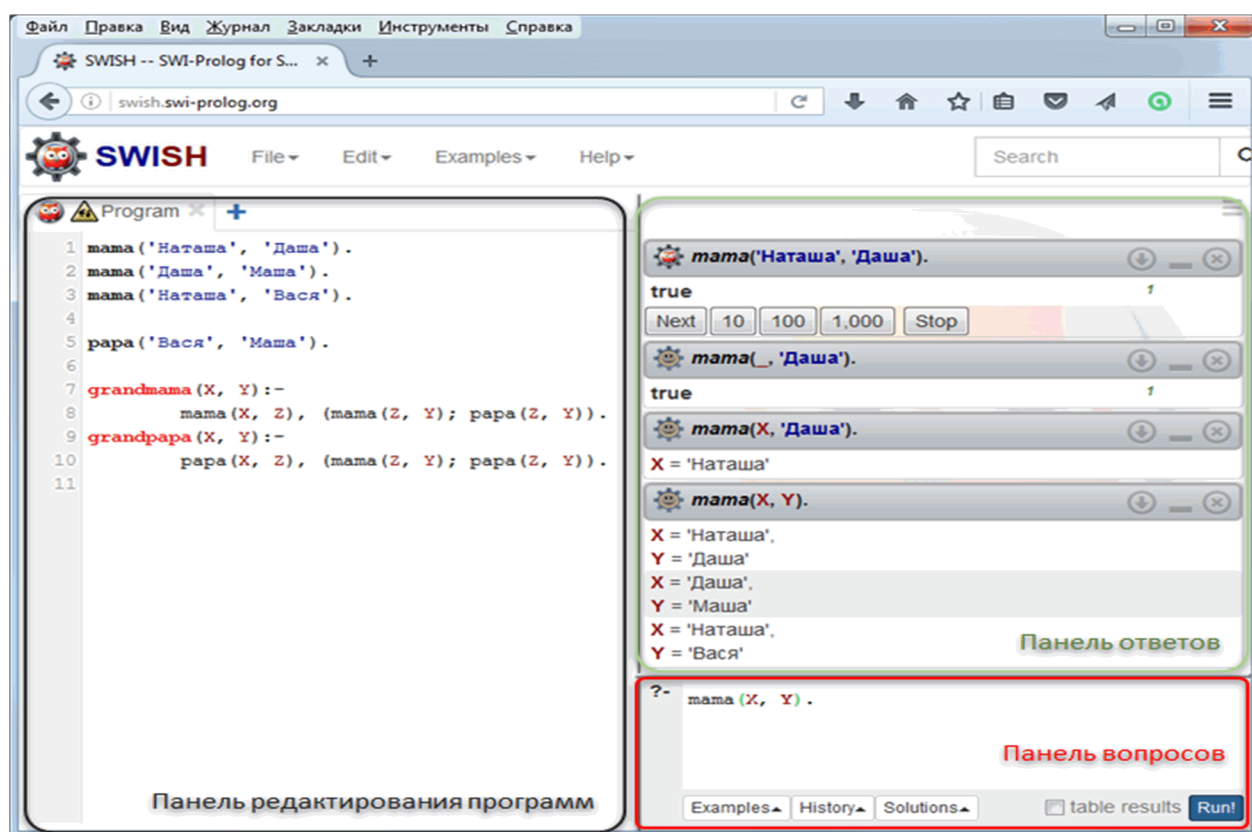


Рис.2. Режим редактирования и исполнения программ

В левой панели осуществляется редактирование программы, содержащей факты и правила.

В правой нижней панели выполняется набор вопросов и запуск их на исполнение с помощью кнопки «Run!».

В правой верхней панели интерпретатор SWI-Prolog выдает ответы на вопросы. В случае если на вопрос может быть получено более одного ответа, с помощью кнопок «Next», «10», «100» и «1,000» можно вывести на панель дополнительные ответы.

Некоторые операции и предикаты SWI-Prolog

Операция / Предикат	Назначение
true	Истина
fail, false	Ложь
=	Для переменной, стоящей слева от операции: - свободной - присваивание без преобразования (вычисления) выражения справа от операции; - связанной - сравнение без преобразования (вычисления) выражения справа от операции.
<, <=, >=, >	Арифметические (только для чисел) операции сравнения
=:=	Арифметическое равенство
=\=	Арифметическое неравенство
is	Для переменной, стоящей слева от операции:

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

	- свободной - присваивание с преобразованием (вычислением) выражения справа от операции; - связанной - сравнение с преобразованием (вычислением) выражения справа от операции.
@<, @=<, @>=, @>	Операции сравнения для констант и переменных любого типа (чисел, строк, списков и т.д.)
==	Равенство констант и переменных любого типа
\==	Неравенство констант и переменных любого типа
not(A)	Отрицание логического выражения A
read(A)	Чтение значения с клавиатуры и присваивание его переменной A
write(A)	Печать A на экран с установкой курсора после последнего напечатанного символа
writeln(A)	Печать A на экран с переводом курсора в начало следующей строки
nl	Перевод курсора в начало следующей строки
repeat	Предикат, выдающий новое истинное значение при возврате. Передоказываемый предикат
!	Предикат (cut, сократить), запрещающий возврат далее той точки, где он стоит
assert(A), assertz(A)	Динамическое добавление факта (правила) в конец списка подобных фактов (правил) базы знаний (программы)
asserta(A)	Динамическое добавление факта (правила) в начало списка подобных фактов (правил) базы знаний
retract(A)	Удаление первого факта (правила) базы знаний
retractall(A)	Удаление всех фактов (правил) базы знаний с именем A

Пример решения логической задачи

Задача:

Как то раз случай свёл в купе астронома, поэта, прозаика и драматурга. Это были Алексеев, Борисов, Константинов и Дмитриев. Оказалось, что каждый из них взял с собой книгу написанную одним из пассажиров этого купе. Алексеев и Борисов углубились в чтение предварительно обменявшись книгами. Поэт читал пьесу, прозаик — очень молодой человек, выпустивший свою книгу, говорил что он никогда и ни чего не читал по астрономии. Борисов купил одно из произведений Дмитриева. Никто из пассажиров не читал свои книги. Что читал каждый из них, кто кем был?

Модификация кода, описанного по ссылке (см. <https://pro-prof.com/archives/1299>)

```
man(alekseev).
man(borisov).
man(konstantinov).
man(dmitriev).
```

```
writebook(astronomy).
writebook(poetry).
writebook(prose).
writebook(piece).
```

```
passenger(_Name, _Read, _Buy, _Write).
```

```
% ДОБАВЛЕНО!!!
unique([]).
unique([X|Xs]) :-
```

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

```

maplist(dif(X),Xs),
unique(Xs).
% -----!!!

check([]):-!.
check([passenger(_, XRead, XBuy, XWrite)|T]):-
!, not(XRead = XWrite), not(XBuy = XWrite), check(T).

solve(Solve):-
Solve = [passenger(X, XRead, XBuy, XWrite), passenger(Y, YRead, YBuy, YWrite),
passenger(Z, ZRead, ZBuy, ZWrite), passenger(W, WRead, WBuy, WWrite)],

% 4 разных пассажира
man(X), man(Y), man(Z), man(W), unique([X, Y, Z, W]),

% каждый написал книгу
writebook(XWrite), writebook(YWrite),
writebook(ZWrite), writebook(WWrite),
unique([XWrite, YWrite, ZWrite, WWrite]),

% каждый купил книгу
writebook(XBuy), writebook(YBuy),
writebook(ZBuy), writebook(WBuy),
unique([XBuy, YBuy, ZBuy, WBuy]),

% каждый читает книгу
writebook(XRead), writebook(YRead),
writebook(ZRead), writebook(WRead),
unique([XRead, YRead, ZRead, WRead]),
% поэт читает пьесу
member(passenger(_, piece, _, poetry), Solve),

% прозаик читает не астрономию
not(member(passenger(_, astronomy, _, prose), Solve)),
% прозаик не покупал астрономию
not(member(passenger(_, _, astronomy, prose), Solve)),

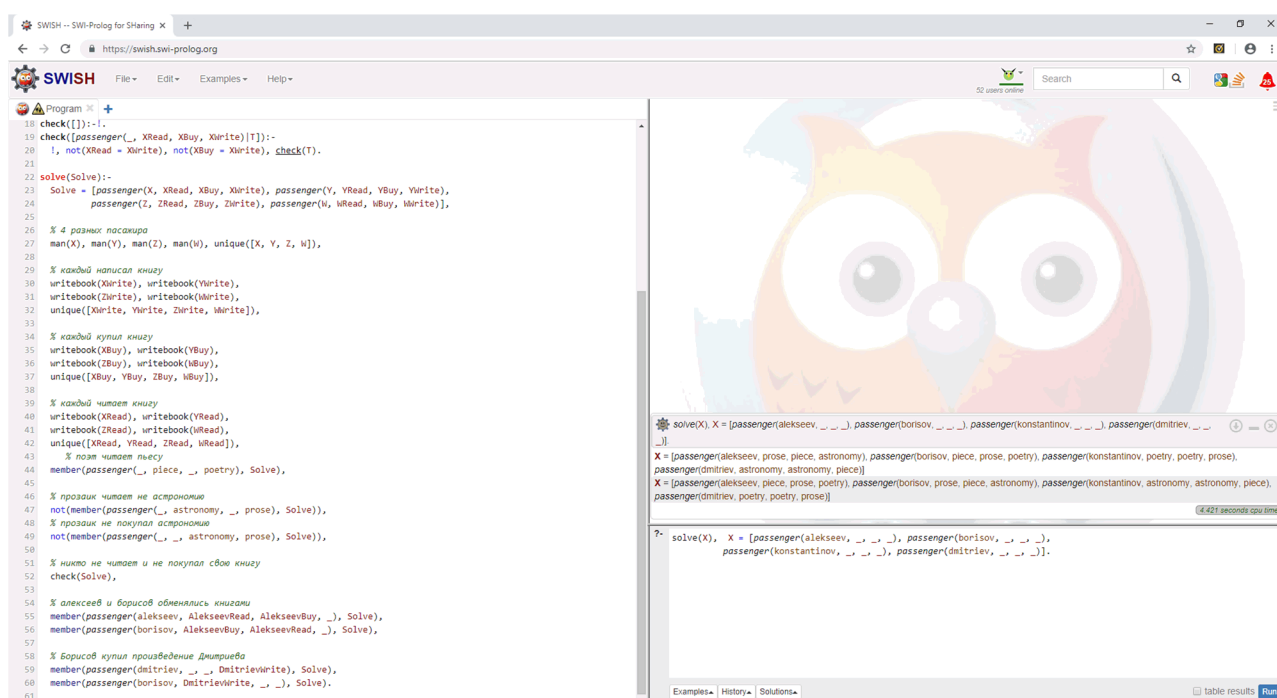
% никто не читает и не покупал свою книгу
check(Solve),

% алексеев и борисов обменялись книгами
member(passenger(alekseev, AlekseevRead, AlekseevBuy, _), Solve),
member(passenger(borisov, AlekseevBuy, AlekseevRead, _), Solve),

% Борисов купил произведение Дмитриева
member(passenger(dmitriev, _, _, DmitrievWrite), Solve),
member(passenger(borisov, DmitrievWrite, _, _), Solve).

```


Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		



Тема 4. Онтологии и Semantic Web.

Цель работы: получение практических навыков построения онтологий.

Задание: Используя программу Protege:

- создать онтологию согласно полученному варианту, онтология должна содержать:
 - иерархию классов (не менее 15);
 - назначенные классам простые свойства Data Properties (не менее 10);
 - назначенные классам свойства-отношения Object Properties (не менее 5) с указанием вида связи между индивидами (функциональная, симметричная и т.д.);
 - индивиды Individuals (не менее 10), с заполненными значениями свойств унаследованного класса;
 - аксиомы, наложенные на свойства и классы в Equivalent to и др. (не менее 5);
 - правила (не менее 5);
- используя плагин OntoGraf (вкладка Window/Tabs/OntoGraf), получить визуальное отображение онтологии в виде графа;
- онтология должна охватывать всю предметную область (требование полноты), и быть достаточно детализированной;
- используя плагин SQWRLTab, построить запросы к онтологии (не менее 3);
- сохранить онтологию;
- открыть созданную онтологию и сохранить во второй файл, затем, используя плагин SWRLTab, построить прогнозируемые аксиомы и дополнить ими онтологию. Провести сравнительный анализ двух онтологий, проверить правильность полученных автоматически элементов онтологии.

Отчет по лабораторной работе должен содержать:

- Фамилию и номер группы учащегося, задание
- Описание онтологий:

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

- a. структуры;
 - b. описание классов;
 - c. свойства;
 - d. индивиды;
3. Графическое отображение онтологий.

Основные теоретические сведения

В последние годы разработка онтологий - формальных явных описаний терминов предметной области и отношений между ними (Gruber 1993) – переходит из мира лабораторий по искусственному интеллекту на рабочие столы экспертов по предметным областям. Во всемирной паутине онтологии стали обычным явлением. Онтологии в сети варьируются от больших таксономий, категоризирующих веб-сайты (как на сайте Yahoo!), до категоризаций продаваемых товаров и их характеристик (как на сайте Amazon.com). Консорциум WWW (W3C) разрабатывает RDF (Resource Description Framework) (Brickley and Guha 1999), язык кодирования знаний на веб-страницах, для того, чтобы сделать их понятными для электронных агентов, которые осуществляют поиск информации. Управление перспективных исследований и разработок министерства обороны США (The Defense Advanced Research Projects Agency, DARPA) в сотрудничестве с W3C разрабатывает Язык Разметки для Агентов DARPA (DARPA Agent Markup Language, DAML), расширяя RDF более выразительными конструкциями, предназначенными для облегчения взаимодействия агентов в сети (Hendler and McGuinness 2000).

Онтология – формальное явное описание понятий в рассматриваемой предметной области (**классов** (иногда их называют **понятиями**)), свойств каждого понятия, описывающих различные свойства и атрибуты понятия (**слотов** (иногда их называют **ролями** или **свойствами**)), и ограничений, наложенных на слоты (**фацетов** (иногда их называют **ограничениями ролей**)).

Онтология вместе с набором индивидуальных **экземпляров** классов образует **базу знаний**. В действительности, трудно определить, где кончается онтология и где начинается база знаний.

Классы описывают понятия предметной области. Например, класс вин представляет все вина. Конкретные вина – экземпляры этого класса. Вино в конкретном бокале – это экземпляр класса вин. Класс может иметь подклассы, которые представляют более конкретные понятия, чем надкласс. Например, мы можем разделить класс всех вин на красные, белые и розовые вина. В качестве альтернативы мы можем разделить класс всех вин на игристые и не игристые вина.

Слоты описывают свойства классов и экземпляров: вино крепкое, оно производится на винном заводе Абрау-Дюрсо. У нас есть два слота, которые описывают вино в этом примере: слот крепость со значением «крепкое» и слот производитель со значением «винный завод Абрау-Дюрсо». Мы можем сказать, что на уровне класса у экземпляров класса Вино есть слоты, которые описывают вкус, крепость, уровень сахара, производителя вина и т.д.

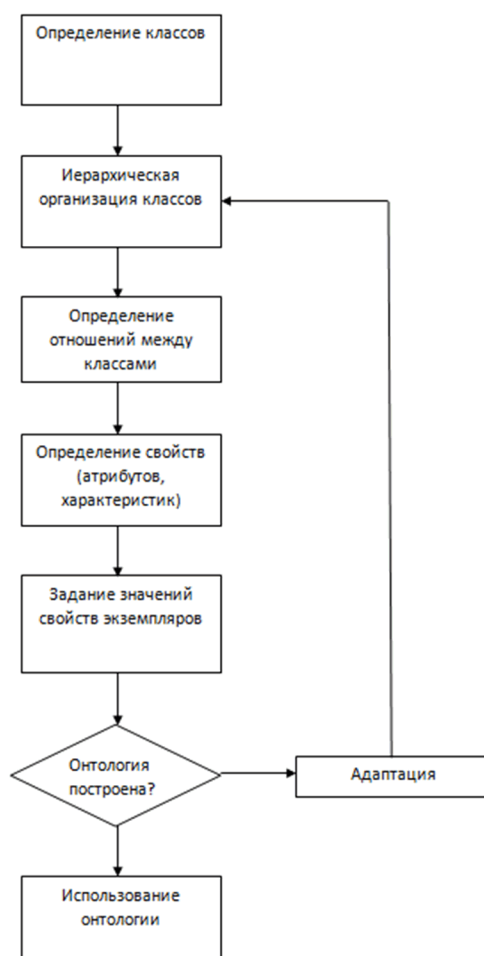
Все экземпляры класса Вино и его подкласс Красное вино имеют слот производитель, значение которого является экземпляром класса Винный завод. Все экземпляры класса Винный завод имеют слот производит, относящийся ко всем винам (экземплярам класса Вино и его подклассов), которые производятся на этом заводе.

Правила построения онтологий.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

- 1) Не существует единственного правильного способа моделирования предметной области – всегда существуют жизнеспособные альтернативы. Лучшее решение почти всегда зависит от предполагаемого приложения и ожидаемых расширений.
- 2) Разработка онтологии – это обязательно итеративный процесс.
- 3) Понятия в онтологии должны быть близки к объектам (физическим или логическим) и отношениям в интересующей вас предметной области. Скорее всего, это существительные (объекты) или глаголы (отношения) в предложениях, которые описывают вашу предметную область.

Схема процесса создания онтологии



Характеристики свойств объектов

Функциональные свойства (Functional) – если свойство является функциональным, то для данного индивида (экземпляра) может существовать не более одного индивида, который имеет отношение к первому индивиду через это свойство.

Обратные функциональные свойства (Inverse functional) – если свойство является обратным функциональному свойству, то это значит, что свойство является обратным функциональным.

Транзитивные свойства (Transitive) – если свойство транзитивное и свойство связывает индивида а и индивида b, а также индивида b связывает с индивидом с, то мы можем вывести, что индивид а связан с индивидом с через это свойство.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Симметричные свойства (Simmetric) – если свойство p симметричное, и свойство связывает индивида a с индивидом b , то индивид b связан также с индивидом a через свойство p .

Асимметричные свойства (Asimmetric) – если свойство p асимметричное, и свойство связывает индивида a с индивидом b , то индивид b не может быть связан с индивидом a через свойство p .

Рефлексивные свойства (Reflexive) – свойство p называется рефлексивным, когда индивид a должен быть связан с собой.

Иррефлексивные свойства (Ireflexive) – если свойство p иррефлексивное, то оно может быть охарактеризовано как свойство, которое связывает индивида a с индивидом b , где индивид a и индивид b обязательно разные.

Список команд для построения аксиом (можно применять в Equivalent to)

Команда	Пример	Значение
some	hasPet some Dog	Things that have a pet that is a Dog This is the most common type of class expression. Also known as, an "SomeValueFrom restriction" or an "Existential Restriction". This kind of class expression consists of a property (in this case hasPet) and a class expression that is known as a filler (in this case the filler is Dog). Individuals that are instances of this class expression have a relationship along the hasPet property to an individual that is an instance of class Dog.
value	hasPet value Tibbs	Things that have a pet that is Tibbs. Here, Tibbs is a specific individual. Intuitively this would describe Tibb's owners. Note that this is a short cut for, and logically equivalent to, (hasPet some {Tibbs}), where the curly brackets describe a class of specific individuals - in this case, a class of one individual that is Tibbs. Also known as a "HasValue restriction"
only	hasPet only Cat	Things that have pets that are only Cats. Note that this does not mean that these things must have a Cat, but if they do have a pet then it will be a Cat. Also known as an "AllValuesFrom restriction" or a "Universal restriction"
min	hasPet min 3 Cat	Things that have at least three pets that are Cats. Things that have at least three pets that are Cats. Also known as a "Min cardinality restriction"
max	hasPet max 5 Dog	Things that have at most five pets that are Dogs. Note that there may be more than five pets in total e.g. three Cats and five Dogs, but the number of Dogs will not be more than five. This class expression also means that there may be no pets at all. Also known as a "Max cardinality restriction"
exactly	hasPet exactly 2 GoldFish	Things that have exactly 2 GoldFish as pets. Note that there may be more than two pets, but two of the pets are GoldFish - no more and no less. Also known as an "Exact cardinality restriction"
and	Person and (hasPet some Cat)	People that have a pet that's a Cat. Here we have a class name and a complex class expression combined with the and keyword. The brackets are optional in this particular case, but have been included for clarity. Also known as an "Intersection" or a "Conjunction". We also say that each class expression in the conjunction is a "conjunct".
or	(hasPet some Cat) or (hasPet some Dog)	Things that have a pet that's a Cat or have a pet that's a Dog Note that the or is not exclusive. That is, this class includes the things that have a Cat as a pet, or have a Dog as a pet, or have both a pet Dot and a pet Cat. Note that we could have also written this example as (hasPet some (Cat or Dog)) Also known as a "Union" or a "Disjunction". We also say that each class expression in the disjunction is a "disjunct".
not	not (hasPet some Dog)	Things that do not have a pet that's a dog. This includes things that either do not have any pet, or if they have a pet then it's not a Dog. Note that this is logically equivalent to (hasPet only (not Dog)) Also known as

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

	а "negation".
--	---------------

Список команд Built-Ins (можно применять в правилах)

Команда	Описание
for Comparisons	
swrlb:equal (from XQuery op:numeric-equal , op:compare , op:boolean-equal , op:yearMonthDuration-equal , op:dayTimeDuration-equal , op:dateTime-equal , op:date-equal , op:time-equal , op:gYearMonth-equal , op:gYear-equal , op:gMonthDay-equal , op:gMonth-equal , op:gDay-equal , op:anyURI-equal)	Satisfied iff the first argument and the second argument are the same.
swrlb:notEqual (from swrlb:equal)	The negation of swrlb:equal.
swrlb:lessThan (from XQuery op:numeric-less-than , op:compare , op:yearMonthDuration-less-than , op:dayTimeDuration-less-than , op:dateTime-less-than , op:date-less-than , op:time-less-than)	Satisfied iff the first argument and the second argument are both in some implemented type and the first argument is less than the second argument according to a type-specific ordering (partial or total), if there is one defined for the type. The ordering function for the type of untyped literals is the partial order defined as string ordering when the language tags are the same (or both missing) and incomparable otherwise.
swrlb:lessThanOrEqual (from swrlb:lessThan, swrlb:equal)	Either less than, as above, or equal, as above.
swrlb:greaterThan (from XQuery op:numeric-greater-than , op:compare , op:yearMonthDuration-greater-than , op:dayTimeDuration-greater-than , op:dateTime-greater-than , op:date-greater-than , op:time-greater-than)	Similarly to swrlb:lessThan
swrlb:greaterThanOrEqual (from swrlb:greaterThan, swrlb:equal)	Similarly to swrlb:lessThanOrEqual.
Math	
swrlb:add (from XQuery op:numeric-add)	Satisfied iff the first argument is equal to the arithmetic sum of the second argument through the last argument.
swrlb:subtract (from XQuery op:numeric-subtract)	Satisfied iff the first argument is equal to the arithmetic difference of the second argument minus the third argument.
swrlb:multiply (from XQuery op:numeric-multiply)	Satisfied iff the first argument is equal to the arithmetic product of the second argument through the last argument.
swrlb:divide (from XQuery op:numeric-divide)	Satisfied iff the first argument is equal to the arithmetic quotient of the second argument divided by the third argument.
swrlb:integerDivide (from XQuery op:numeric-integer-divide)	Satisfied if the first argument is the arithmetic quotient of the second argument idiv the third argument. If the numerator is not evenly divided by the divisor, then the quotient is the xsd:integer value obtained, ignoring any remainder that results from the division (that is, no rounding is performed).
swrlb:mod (from XQuery op:numeric-mod)	Satisfied if the first argument represents the remainder resulting from dividing the second argument, the dividend, by the third argument, the divisor. The operation a mod b for operands that are xsd:integer or xsd:decimal, or types derived from them, produces a

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

	result such that $(a \text{ idiv } b) * b + (a \text{ mod } b)$ is equal to a and the magnitude of the result is always less than the magnitude of b . This identity holds even in the special case that the dividend is the negative integer of largest possible magnitude for its type and the divisor is -1 (the remainder is 0). It follows from this rule that the sign of the result is the sign of the dividend
swrlb:pow	Satisfied iff the first argument is equal to the result of the second argument raised to the third argument power.
swrlb:unaryPlus (from XQuery op:numeric-unary-plus)	Satisfied iff the first argument is equal to the second argument with its sign unchanged.
swrlb:unaryMinus (from XQuery op:numeric-unary-minus)	Satisfied iff the first argument is equal to the second argument with its sign reversed.
swrlb:abs (from XQuery fn:abs)	Satisfied iff the first argument is the absolute value of the second argument.
swrlb:ceiling (from XQuery fn:ceiling)	Satisfied iff the first argument is the smallest number with no fractional part that is greater than or equal to the second argument.
swrlb:floor (from XQuery fn:floor)	Satisfied iff the first argument is the largest number with no fractional part that is less than or equal to the second argument
swrlb:round (from XQuery fn:round)	Satisfied iff the first argument is equal to the nearest number to the second argument with no fractional part.
swrlb:roundHalfToEven (from XQuery fn:round-half-to-even)	Satisfied iff the first argument is equal to the second argument rounded to the given precision. If the fractional part is exactly half, the result is the number whose least significant digit is even.
swrlb:sin	Satisfied iff the first argument is equal to the sine of the radian value the second argument.
swrlb:cos	Satisfied iff the first argument is equal to the cosine of the radian value the second argument.
swrlb:tan	Satisfied iff the first argument is equal to the tangent of the radian value the second argument.
for Boolean Values	
swrlb:booleanNot (from XQuery fn:not)	Satisfied iff the first argument is true and the second argument is false, or vice versa.
for Strings	
swrlb:stringEqualIgnoreCase	Satisfied iff the first argument is the same as the second argument (upper/lower case ignored)
swrlb:stringConcat (from XQuery fn:concat)	Satisfied iff the first argument is equal to the string resulting from the concatenation of the strings the second argument through the last argument.
swrlb:substring (from XQuery fn:substring)	Satisfied iff the first argument is equal to the substring of optional length the fourth argument starting at character offset the third argument in the string the second argument.
swrlb:stringLength (from XQuery fn:string-length)	Satisfied iff the first argument is equal to the length of the second argument.
swrlb:normalizeSpace (from XQuery fn:normalize-space)	Satisfied iff the first argument is equal to the whitespace-normalized value of the second argument.
swrlb:upperCase (from XQuery fn:upper-case)	Satisfied iff the first argument is equal to the upper-cased value of the second argument.
swrlb:lowerCase (from XQuery fn:lower-case)	Satisfied iff the first argument is equal to the lower-cased value of the second argument.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

swrlb:translate (from XQuery fn:translate)	Satisfied iff the first argument is equal to the second argument with occurrences of characters contained in the third argument replaced by the character at the corresponding position in the string the fourth argument.
swrlb:contains (from XQuery fn:contains)	Satisfied iff the first argument contains the second argument (case sensitive).
swrlb:containsIgnoreCase	Satisfied iff the first argument contains the second argument (case ignored).
swrlb:startsWith (from XQuery fn:starts-with)	Satisfied iff the first argument starts with the second argument.
swrlb:endsWith (from XQuery fn:ends-with)	Satisfied iff the first argument ends with the second argument.
swrlb:substringBefore (from XQuery fn:substring-before)	Satisfied iff the first argument is the characters of the second argument that precede the characters of the third argument.
swrlb:substringAfter (from XQuery fn:substring-after)	Satisfied iff the first argument is the characters of the second argument that follow the characters of the third argument.
swrlb:matches (from XQuery fn:matches)	Satisfied iff the first argument matches the regular expression the second argument.
swrlb:replace (from XQuery fn:replace)	Satisfied iff the first argument is equal to the value of the second argument with every substring matched by the regular expression the third argument replaced by the replacement string the fourth argument.
swrlb:tokenize (from XQuery fn:tokenize)	Satisfied iff the first argument is a sequence of one or more strings whose values are substrings of the second argument separated by substrings that match the regular expression the third argument.
for Date, Time and Duration	
swrlb:yearMonthDuration (from XQuery xdt:yearMonthDuration)	Satisfied iff the first argument is the xsd:duration representation consisting of the year the second argument and month the third argument.
swrlb:dayTimeDuration (from XQuery xdt:dayTimeDuration)	Satisfied iff the first argument is the xsd:duration representation consisting of the days the second argument, hours the third argument, minutes the fourth argument, and seconds the fifth argument.
swrlb:dateTime	Satisfied iff the first argument is the xsd:dateTime representation consisting of the year the second argument, month the third argument, day the fourth argument, hours the fifth argument, minutes the sixth argument, seconds the seventh argument, and timezone the eighth argument.
swrlb:date	Satisfied iff the first argument is the xsd:date representation consisting of the year the second argument, month the third argument, day the fourth argument, and timezone the fifth argument.
swrlb:time	Satisfied iff the first argument is the xsd:time representation consisting of the hours the second argument, minutes the third argument, seconds the fourth argument, and timezone the fifth argument.
swrlb:addYearMonthDurations (from XQuery op:add-yearMonthDurations)	Satisfied iff the yearMonthDuration the first argument is equal to the arithmetic sum of the yearMonthDuration the second argument through the yearMonthDuration the last argument.
swrlb:subtractYearMonthDurations (from XQuery op:subtract-yearMonthDurations)	Satisfied iff the yearMonthDuration the first argument is equal to the arithmetic difference of the yearMonthDuration the second argument minus the yearMonthDuration the third argument.
swrlb:multiplyYearMonthDuration (from XQuery op:multiply-yearMonthDuration)	Satisfied iff the yearMonthDuration the first argument is equal to the arithmetic product of the yearMonthDuration the second

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

	argument multiplied by the third argument.
swrlb:divideYearMonthDurations (from XQuery op:divide-yearMonthDuration)	Satisfied iff the yearMonthDuration the first argument is equal to the arithmetic remainder of the yearMonthDuration the second argument divided by the third argument.
swrlb:addDayTimeDurations (from XQuery op:add-dayTimeDurations)	Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic sum of the dayTimeDuration the second argument through the dayTimeDuration the last argument.
swrlb:subtractDayTimeDurations (from XQuery op:subtract-dayTimeDurations)	Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic difference of the dayTimeDuration the second argument minus the dayTimeDuration the third argument.
swrlb:multiplyDayTimeDurations (from XQuery op:multiply-dayTimeDuration)	Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic product of the dayTimeDuration the second argument multiplied by the third argument.
swrlb:divideDayTimeDuration (from XQuery op:divide-dayTimeDuration)	Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic remainder of the dayTimeDuration the second argument divided by the third argument.
swrlb:subtractDates (from XQuery op:subtract-dates)	Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic difference of the xsd:date the second argument minus the xsd:date the third argument.
swrlb:subtractTimes (from XQuery op:subtract-times)	Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic difference of the xsd:time the second argument minus the xsd:time the third argument.
swrlb:addYearMonthDurationToDateTime (from XQuery op:add-yearMonthDuration-to-dateTime)	Satisfied iff the xsd:dateTime the first argument is equal to the arithmetic sum of the xsd:dateTime the second argument plus the yearMonthDuration the third argument.
swrlb:addDayTimeDurationToDateTime (from XQuery op:add-dayTimeDuration-to-dateTime)	Satisfied iff the xsd:dateTime the first argument is equal to the arithmetic sum of the xsd:dateTime the second argument plus the dayTimeDuration the third argument.
swrlb:subtractYearMonthDurationFromDateTime (from XQuery op:subtract-yearMonthDuration-from-dateTime)	Satisfied iff the xsd:dateTime the first argument is equal to the arithmetic difference of the xsd:dateTime the second argument minus the yearMonthDuration the third argument.
swrlb:subtractDayTimeDurationFromDateTime (from XQuery op:subtract-dayTimeDuration-from-dateTime)	Satisfied iff the xsd:dateTime the first argument is equal to the arithmetic difference of the xsd:dateTime the second argument minus the dayTimeDuration the third argument.
swrlb:addYearMonthDurationToDate (from XQuery op:add-yearMonthDuration-to-date)	Satisfied iff the xsd:date the first argument is equal to the arithmetic sum of the xsd:date the second argument plus the yearMonthDuration the third argument.
swrlb:addDayTimeDurationToDate (from XQuery op:add-dayTimeDuration-to-date)	Satisfied iff the xsd:date the first argument is equal to the arithmetic sum of the xsd:date the second argument plus the dayTimeDuration the third argument.
swrlb:subtractYearMonthDurationFromDate (from XQuery op:subtract-yearMonthDuration-from-date)	Satisfied iff the xsd:date the first argument is equal to the arithmetic difference of the xsd:date the second argument minus the yearMonthDuration the third argument.
swrlb:subtractDayTimeDurationFromDate (from XQuery op:subtract-dayTimeDuration-from-date)	Satisfied iff the xsd:date the first argument is equal to the arithmetic difference of the xsd:date the second argument minus the yearMonthDuration the third argument.
swrlb:addDayTimeDurationToTime (from XQuery op:add-dayTimeDuration-to-time)	Satisfied iff the xsd:time the first argument is equal to the arithmetic sum of the xsd:time the second argument plus the dayTimeDuration the third argument.
swrlb:subtractDayTimeDurationFromTime (from XQuery op:subtract-dayTimeDuration-from-time)	Satisfied iff the xsd:time the first argument is equal to the arithmetic difference of the xsd:time the second argument minus the dayTimeDuration the third argument.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

swrlb:subtractDateTimesYieldingYearMonthDuration (from XQuery fn:subtract-dateTimes-yielding-yearMonthDuration)	Satisfied iff the yearMonthDuration the first argument is equal to the arithmetic difference of the xsd:dateTime the second argument minus the xsd:dateTime the third argument.
swrlb:subtractDateTimesYieldingDayTimeDuration (from XQuery fn:subtract-dateTimes-yielding-dayTimeDuration)	Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic difference of the xsd:dateTime the second argument minus the xsd:dateTime the third argument.
for URIs	
swrlb:resolveURI (from XQuery op:resolve-uri)	Satisfied iff the URI reference the first argument is equal to the value of the URI reference the second argument resolved relative to the base URI the third argument.
swrlb:anyURI	Satisfied iff the first argument is a URI reference consisting of the scheme the second argument, host the third argument, port the fourth argument, path the fifth argument, query the sixth argument, and fragment the seventh argument.
for Lists	
swrlb:listConcat (from Common Lisp append)	Satisfied iff the first argument is a list representing the concatenation of the lists the second argument through the last argument.
swrlb:listIntersection	Satisfied iff the first argument is a list containing elements found in both the list the second argument and the list the third argument.
swrlb:listSubtraction	Satisfied iff the first argument is a list containing the elements of the list the second argument that are not members of the list the third argument.
swrlb:member	Satisfied iff the first argument is a member of the list the second argument.
swrlb:length (from Common Lisp list-length)	Satisfied iff the first argument is the length of the list the second argument (number of members of the list).
swrlb:first (from rdf:first)	Satisfied iff the first argument is the first member of the list the second argument.
swrlb:rest (from rdf:rest)	Satisfied iff the first argument is a list containing all members of the list the second argument except the first member (the head).
swrlb:sublist	Satisfied iff the list the first argument contains the list the second argument.
swrlb:empty (from rdf:nil)	Satisfied iff the list the first argument is an empty list.

Варианты заданий

1. Информационные системы
2. Интеллектуальные информационные системы
3. Компьютерные сети
4. Языки программирования
5. Модели представления знаний
6. Классификация нейронных сетей
7. Классификация экспертных систем
8. Классификация систем поддержки принятия решений
9. Электронно-вычислительные машины (компьютеры)
10. Направления искусственного интеллекта
11. Операционные системы
12. Классификация наук
13. Направления исследований в информатике
14. Протоколы Интернет
15. Свободный вариант, студент сам предлагает предметную область (связанна с информатикой) и согласовывает с преподавателем.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Тема 7. Нейронные сети.

Задание: выполните лабораторные работы, согласно приведенным инструкциям в учебно-методическом пособии:

Седов, В. А. Введение в нейронные сети : методические указания к лабораторным работам по дисциплине «Нейроинформатика» для студентов специальности 09.03.02 «Информационные системы и технологии» / В. А. Седов, Н. А. Седова. — Саратов : Ай Пи Эр Медиа, 2018. — 30 с. — ISBN 978-5-4486-0047-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/69319.html> (дата обращения: 09.10.2019). — Режим доступа: для авторизир. Пользователей.

Скачать программу можно по адресу: <http://www.lbai.ru/#;show;install>

8. ТЕМАТИКА КУРСОВЫХ, КОНТРОЛЬНЫХ РАБОТ, РЕФЕРАТОВ

Данный вид работы не предусмотрен УП.

9. ПЕРЕЧЕНЬ ВОПРОСОВ К ЭКЗАМЕНУ

1. Философские вопросы искусственного интеллекта.
2. Системы ИИ. Историческая справка.
3. Основные направления исследований в ИИ.
4. Компьютерная лингвистика: распознавание и синтез речи, машинный перевод.
5. Онтологии. Стандарты Semantic Web.
6. Онтологии: правила и запросы.
7. Знания и данные. Модели представления знаний (логические и продукционные модели).
8. Знания и данные. Модели представления знаний (сетевые и фреймовые модели).
9. Искусственные нейронные сети (основные понятия и определения, виды НС, область применения, отличия архитектуры нейрокомпьютера от архитектуры фон Неймана).
10. Виды функций активаций, инициализация весов, регуляризация нейронных сетей, нормализация в нейронных сетях.
11. Искусственные нейронные сети: алгоритмы обучения (алгоритм обучения по дельта-правилу).
12. Искусственные нейронные сети: алгоритмы обучения (алгоритм обратного распространения ошибки).
13. Свёрточные нейронные сети.
14. Рекуррентные нейронные сети.
15. Теория нечетких множеств (основные понятия и определения, операции над множествами).
16. Теория нечетких множеств (понятие лингвистической переменной, нечеткие высказывания).
17. Генетические алгоритмы (основные понятия и определения, операторы ГА).

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

10. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩИХСЯ

Форма обучения очная

Название разделов и тем	Вид самостоятельной работы (проработка учебного материала, решение задач, реферат, доклад, контрольная работа, подготовка к сдаче зачета, экзамена и др.)	Объем в часах	Форма контроля (проверка решения задач, реферата и др.)
Философские вопросы ИИ.	чтение основной и дополнительной литературы, самостоятельное изучение материала по литературным источникам;	5	тестирование
Подходы и направления исследований в ИИ	чтение основной и дополнительной литературы, самостоятельное изучение материала по литературным источникам;	5	тестирование
Представление знаний	самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты);	10	Проверка решения задач
Онтологии и Semantic Web	чтение основной и дополнительной литературы, самостоятельное изучение материала по литературным источникам;	20	тестирование
Эволюционное моделирование	самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты);	10	Проверка решения задач
Нечеткие вычисления	самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты);	10	Проверка решения задач
Нейронные сети	самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты);	30	Проверка решения задач

Форма обучения заочная

Название разделов и тем	Вид самостоятельной работы (проработка учебного материала,	Объем в часах	Форма контроля
-------------------------	---	---------------	----------------

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

	<i>решение задач, реферат, доклад, контрольная работа, подготовка к сдаче зачета, экзамена и др.)</i>		<i>(проверка решения задач, реферата и др.)</i>
Философские вопросы ИИ.	чтение основной и дополнительной литературы, самостоятельное изучение материала по литературным источникам;	15	тестирование
Подходы и направления исследований в ИИ	чтение основной и дополнительной литературы, самостоятельное изучение материала по литературным источникам;	15	тестирование
Представление знаний	самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты);	25	Проверка решения задач
Онтологии и Semantic Web	чтение основной и дополнительной литературы, самостоятельное изучение материала по литературным источникам;	35	тестирование
Эволюционное моделирование	самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты);	25	Проверка решения задач
Нечеткие вычисления	самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты);	35	Проверка решения задач
Нейронные сети	самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты);	35	Проверка решения задач

11. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) Список рекомендуемой литературы

основная

- 1) Смагин, А. А. Интеллектуальные информационные системы : учеб. пособие для вузов / А. А. Смагин, С. В. Липатова, А. С. Мельниченко ; УлГУ, Фак. математики

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

и информ. технологий, Каф. телекоммуникац. технологий и сетей. - Ульяновск : УлГУ, 2010.- URL: <ftp://10.2.96.134/Text/smagin2.pdf>

- 2) Станкевич, Л. А. Интеллектуальные системы и технологии : учебник и практикум для бакалавриата и магистратуры / Л. А. Станкевич. — Москва : Издательство Юрайт, 2019. — 397 с. — (Бакалавр и магистр. Академический курс). — ISBN 978-5-534-02126-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/433370>

дополнительная

- 3) Каку М., Будущее разума [Электронный ресурс] / Каку М. - М. : Альпина Паблишер, 2016. - 502 с. - ISBN 978-5-91671-369-5 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785916713695.html>
- 4) Седова, Н. А. Теория нечетких множеств : учебное пособие / Н. А. Седова, В. А. Седов. — Саратов : Ай Пи Ар Медиа, 2019. — 421 с. — ISBN 978-5-4497-0196-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/86526.html>
- 5) Павлова, А. И. Информационные технологии: основные положения теории искусственных нейронных сетей : учебное пособие / А. И. Павлова. — Новосибирск : Новосибирский государственный университет экономики и управления «НИНХ», 2017. — 191 с. — ISBN 978-5-7014-0801-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/87110.html>
- 6) Исаев, С. В. Интеллектуальные системы : учебное пособие / С. В. Исаев, О. С. Исаева. — Красноярск : Сибирский федеральный университет, 2017. — 120 с. — ISBN 978-5-7638-3781-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/84365.html>

учебно-методическая

- 7) Липатова С. В. Методические рекомендации для семинарских (практических) занятий, лабораторного практикума и самостоятельной работы по дисциплинам «Интеллектуальные информационные системы» и «Системы искусственного интеллекта» для студентов направлений 09.03.02 «Информационные системы и технологии», 11.03.02 «Инфокоммуникационные технологии и системы», 09.03.03 «Прикладная информатика», 02.03.03 «Математическое обеспечение и администрирование информационных систем» / С. В. Липатова; УлГУ, Фак. математики, информ. и авиац. технологий. - Ульяновск : УлГУ, 2019. - Загл. с экрана; Неопубликованный ресурс. - Электрон. текстовые дан. (1 файл : 2,33 МБ). - Текст : электронный. <http://lib.ulsu.ru/MegaPro/Download/MObject/6218>

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Согласовано:

П. В. Сид. пр. Попова И. И. В. В.

Должность сотрудника научной библиотеки

ФИО

подпись

дата

б) Программное обеспечение

1. Редактор онтологий Protégé (плагины SWRLTab, Pellet).
2. Anaconda (дистрибутив языков программирования Python и R), библиотеки (open source).

в) Профессиональные базы данных, информационно-справочные системы:

1. Электронно-библиотечные системы:

1.1. IPRbooks : электронно-библиотечная система : сайт / группа компаний Ай Пи Ар Медиа. - Саратов, [2020]. – URL: <http://www.iprbookshop.ru>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.2. ЮРАЙТ : электронно-библиотечная система : сайт / ООО Электронное издательство ЮРАЙТ. – Москва, [2020]. – URL: <https://www.biblio-online.ru>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.3. Консультант студента : электронно-библиотечная система : сайт / ООО Политехресурс. – Москва, [2020]. – URL: http://www.studentlibrary.ru/catalogue/switch_kit/x2019-128.html. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.4. Лань : электронно-библиотечная система : сайт / ООО ЭБС Лань. – Санкт-Петербург, [2020]. – URL: <http://www.studentlibrary.ru/pages/catalogue.html> <https://e.lanbook.com>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.5. Znanium.com : электронно-библиотечная система : сайт / ООО Знаниум. – Москва, [2020]. – URL: <http://www.studentlibrary.ru/pages/catalogue.html> <http://znanium.com>. – Режим доступа : для зарегистрир. пользователей. – Текст : электронный.

2. КонсультантПлюс [Электронный ресурс]: справочная правовая система. /ООО «Консультант Плюс» - Электрон. дан. - Москва : КонсультантПлюс, [2020].

3. Базы данных периодических изданий:

3.1. База данных периодических изданий : электронные журналы / ООО ИВИС. – Москва, [2020]. – URL: <https://dlib.eastview.com/browse/udb/12>. – Режим доступа : для авториз. пользователей. – Текст : электронный.

3.2. eLIBRARY.RU: научная электронная библиотека : сайт / ООО Научная Электронная Библиотека. – Москва, [2020]. – URL: <http://elibrary.ru>. – Режим доступа : для авториз. пользователей. – Текст : электронный

3.3. «Grebennikon» : электронная библиотека / ИД Гребенников. – Москва, [2020]. – URL: <https://id2.action-media.ru/Personal/Products>. – Режим доступа : для авториз. пользователей. – Текст : электронный.

4. Национальная электронная библиотека : электронная библиотека : федеральная государственная информационная система : сайт / Министерство культуры РФ ; РГБ. – Москва, [2020]. – URL: <http://www.studentlibrary.ru/pages/catalogue.html> <https://нэб.рф>. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

5. SMART Imagebase // EBSCOhost : [портал]. – URL: <https://ebSCO.smartimagebase.com/?TOKEN=EBSCO-1a2ff8c55aa76d8229047223a7d6dc9c&custid=s6895741>. – Режим доступа : для авториз.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

пользователей. – Изображение : электронные.

6. Федеральные информационно-образовательные порталы:

6.1. Единое окно доступа к образовательным ресурсам : федеральный портал / учредитель ФГАОУ ДПО ЦРГОП и ИТ. – URL: <http://window.edu.ru/>. – Текст : электронный.

6.2. Российское образование : федеральный портал / учредитель ФГАОУ ДПО ЦРГОП и ИТ. – URL: <http://www.edu.ru>. – Текст : электронный.

7. Образовательные ресурсы УлГУ:

7.1. Электронная библиотека УлГУ : модуль АБИС Мега-ПРО / ООО «Дата Экспресс». – URL: <http://lib.ulsu.ru/MegaPro/Web>. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

7.2. Образовательный портал УлГУ. – URL: <http://edu.ulsu.ru>. – Режим доступа : для зарегистрированных пользователей. – Текст : электронный

Согласовано:

Заместитель начальника УИТиТ/ _____ Ключкова А.В. / _____
Должность сотрудника УИТиТ ФИО подпись дата

12. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ:

Аудитории для проведения лекций (лекционные аудитории 3 корпуса УлГУ), семинарских занятий (лекционные аудитории 3 корпуса УлГУ), для выполнения лабораторных работ и практикумов (дисплейные классы 1 корпуса УлГУ), для проведения текущего контроля и промежуточной аттестации (лекционные аудитории 3 корпуса УлГУ).

Аудитории укомплектованы специализированной мебелью, учебной доской. Аудитории для проведения лекций оборудованы мультимедийным оборудованием для предоставления информации большой аудитории. Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде, электронно-библиотечной системе. Перечень оборудования, используемого в учебном процессе, указывается в соответствии со сведениями о материально-техническом обеспечении и оснащённости образовательного процесса, размещёнными на официальном сайте УлГУ в разделе «Сведения об образовательной организации».

13. СПЕЦИАЛЬНЫЕ УСЛОВИЯ ДЛЯ ОБУЧАЮЩИХСЯ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей:

– для лиц с нарушениями зрения: в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации;

– для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации;

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

– для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации;

– для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; индивидуальные задания и консультации.

В случае необходимости использования в учебном процессе частично/исключительно дистанционных образовательных технологий, организация работы ППС с обучающимися с ОВЗ и инвалидами предусматривается в электронной информационно-образовательной среде с учетом их индивидуальных психофизических особенностей.

Разработчик



доцент

С.В.