

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Вычислительной и прикладной математики»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ
«Логическое программирование»

Направление подготовки

09.03.04 Программная инженерия

Направленность (профиль) подготовки

«Программная инженерия»

Квалификация выпускника — бакалавр

Форма обучения — очная

1. ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур), предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной профессиональной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям ОПОП.

Основная задача – обеспечить оценку уровня сформированности общекультурных, общепрофессиональных и профессиональных компетенций.

Контроль знаний обучающихся проводится в форме промежуточной аттестации.

Промежуточный контроль по дисциплине осуществляется проведением экзамена.

2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;

2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;

3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых дисциплиной:

а) описание критериев и шкалы оценивания тестирования:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 75 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 60 до 74%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 59%

б) описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя.
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов.
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя.
0 баллов	выставляется студенту, который не смог ответить на вопрос

в) описание критериев и шкалы оценивания практического задания:

Шкала оценивания	Критерий
-------------------------	-----------------

3 балла (эталонный уровень)	Задача решена верно
2 балла (продвинутый уровень)	Задача решена верно, но имеются технические неточности в расчетах
1 балл (пороговый уровень)	Задача решена верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задача не решена

На экзамен выносятся: тестовое задание, 1 практическое задание и 1 теоретический вопрос. Студент может набрать максимум 9 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Шкала оценивания	Критерий	
отлично (эталонный уровень)	8 – 9 баллов	Обязательным условием является выполнение всех предусмотренных в течение семестра заданий
хорошо (продвинутый уровень)	6 – 7 баллов	
удовлетворительно (пороговый уровень)	4 – 5 баллов	
неудовлетворительно	0 – 3 баллов	Студент не выполнил всех предусмотренных в течение семестра текущих заданий

3. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части)	Наименование оценочного мероприятия
1	2	3	4
1	Тема 1. Описание предметной области на языке Пролог	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
2	Тема 2. Общая схема выполнения программы, четыре фазы доказательства предиката	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
3	Тема 3. Общая схема выполнения программы, четыре фазы доказательства предиката	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
4	Тема 4. Арифметические и операторные выражения. Ввод и вывод в Прологе	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
5	Тема 5. Управление выполнением программы на Прологе	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
6	Тема 6. Управление выполнением программы на Прологе. Встроенные предикаты fail, repeat. Организация получения всего множества решений. Работа с динамической базой данных	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен

7	Тема 7. Встроенные предикаты fail, repeat. Организация получения всего множества решений	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
8	Тема 8. Работа с динамической базой данных	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
9	Тема 9. . Рекурсия. Виды рекурсий. Воз-врат и отсечение.	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
10	Тема 10. Теория сложности алгоритмов.	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
11	Тема 11. Списки	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
12	Тема 12. Списки	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
13	Тема 13. Строки	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
14	Тема 14. Строки	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
15	Тема 15. Структуры	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
16	Тема 16. Программирование задач искусственного интеллекта на языке Пролог	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен
17	Тема 17. Дескрипционная логика. Язык OWL. Онтологии и их применение для организации знаний	ПК-1.3, ПК-3.1, ПК-3.2	Экзамен

4. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация (экзамен)

ПК-1: Способен разрабатывать требования, проектировать и выполнять программную реализацию программного обеспечения
ПК-1.3. Проектирует программное обеспечение и выполняет его программную реализацию

а) типовые тестовые вопросы закрытого типа

Вопрос 1: Что понимается под предметной областью при описании ее на языке Пролог?

Шаблоны ответов (знак + - правильный ответ):

- +1: Множество рассматриваемых объектов и отношений: между объектом и его свойствами, между объектами
 - 2: Множество решаемых задач
 - 3: Наиболее общая информация о многих объектах предметной области.
 - 4: Предметная область – это предметы, окружающие нас
- Ответ:-1

Вопрос 2: К какому типу языков относится язык Пролог?

Шаблоны ответов:

- 1: Алгоритмический язык
 - +2: Декларативный язык
 - 3: Процедурный язык
 - 4: Функциональный язык
- Ответ: - 2

Вопрос 3: Какие элементы языка Пролог описывают конкретные объекты предметной области?

Шаблоны ответов:

- 1: Переменные
- 2: Предикаты
- +3: Константы
- 4: Утверждения

Ответ: -3

Вопрос 4: Какие элементы языка Пролог описывают некоторые или любые объекты предметной области?

Шаблоны ответов:

- +1: Переменные
- 2: Предикаты
- 3: Константы
- 4: Утверждения

Ответ: - 1

Вопрос 5: Какие элементы языка Пролог описывают отношения в предметной области?

Шаблоны ответов:

- 1: Переменные
- +2: Предикаты
- 3: Константы
- 4: Утверждения

Ответ: - 2

Вопрос 6: Какие элементы нижеприведенного списка являются правильными символическими именами в языке Visual Prolog?

Шаблоны ответов:

- +1: demon
- 2: «Наполеон»
- 3: 3Кит
- +4: студент_группа
- 5: Werdikt

Ответ: - 1, 4

Вопрос 7: Какие элементы нижеприведенного списка являются переменными в языке Visual Prolog?

Шаблоны ответов:

- 1: лимон
- 2: «Наполеон»
- +3: Student
- +4: _perv
- +5: Вердикт

Ответ: 3,4,5

Вопрос 8: Что такое предикат?

Шаблоны ответов:

- 1: Предикат – это утверждение
- +2: Предикат – это логическая функция n аргументов, имеющая два значения: истина и ложь
- 3: Предикат – это функция n аргументов, имеющая значение объекта предметной области
- 4: Предикат – это всегда истинное утверждение

Ответ: - 2

Вопрос 9

Текст вопроса: Какие виды утверждений существуют в языке Visual Prolog?

Шаблоны ответов:

- +1: Факты
- 2: Условные утверждения
- +3: Правила
- 4: Высказывания
- +5: Вопросы

Ответ:1,3,5

Вопрос 10

Текст вопроса: Какое из нижеприведенных утверждений является фактом?

Шаблоны ответов:

- 1: обед(«картофельный суп», Y,Z)
- 2: лето:- солнце, тепло, зеленая_трава.
- +3: студент(«Соколов»,343, «м»).
- 4: goal студент(Fam, Gruppo, Pol).

Ответ: - 3

б) типовые тестовые вопросы открытого типа

Вопрос 11

Текст вопроса: Запишите правильную структуру правила.

Ответ:<заголовок>:-<тело>

Вопрос 12

Текст вопроса: Какой логической операцией соединены предикаты в теле правила, перечисленные через запятую?

Ответ: конъюнкцией

Вопрос 13

Текст вопроса: Какой логической операцией соединены предикаты заголовка и тела правила?

Ответ: импликацией

Вопрос 14

Текст вопроса: Какой логической операцией соединены предикаты заголовков правил с одинаковым именем и числом аргументов, например, $p:-a,b$. $p:-b,a$.?

Ответ: дизъюнкцией

Вопрос 15

Текст вопроса: С каким квантором «всеобщности» или «существования» входят переменные в **заголовок** правила?

Ответ: «всеобщности»

Вопрос 16

Текст вопроса: С каким квантором «всеобщности» или «существования» входят переменные в **запрос**?

Ответ: «существования»

Вопрос 17

Текст вопроса: Что такое вопрос?

Ответ: Гипотеза, истинность или ложность которой доказывается в процессе выполнения программы

Вопрос 18

Текст вопроса: Что такое факт?

Ответ: Предикат, описывающий истинное в данной предметной области отношение

Вопрос 19

Текст вопроса: Ниже приведен фрагмент программы. Какой будет получен результат в ответ на запрос?

CLAUSES

студент(1,«Иванов»,343).

предмет(2,«Системы искусственного интеллекта»).

студент_предмет_оценка(1, 2, 4).

оценки_студента(Fam,Disc,Ocenka):- студент(IdS,Fam,_)

предмет(IdD,Disc),

студент_предмет_оценка(IdS, IdD,Ocenka).

GOAL

оценки_студента(Fam,Disc,Ocenka), write("Fam=",Fam," Disc=",Disc," Ocenka=",Ocenka),nl.

Ответ: Fam=Иванов Disc=Системы искусственного интеллекта Ocenka=4

Вопрос 20

Текст вопроса: запишите последовательно фазы обработки целевого утверждения?

Ответ:

- 1: Сопоставление с образцом
- 2: Унификация аргументов
- 3: Проверка – факт или правило
- 4: Процесс возврата

Вопрос 21

Текст вопроса: При выполнении фазы «сопоставление с образцом» что принимается за образец?

Ответ:

- 1: Имя и число аргументов предиката вопроса

ПК-3: Способен разрабатывать компоненты системных программных продуктов
ПК-3.1. Разрабатывает системные утилиты программного обеспечения
ПК-3.2. Создает компоненты инструментальных средств программирования

а) типовые тестовые вопросы закрытого типа

Вопрос 22

Текст вопроса: Какое из нижеприведенных утверждений является правилом?

Шаблоны ответов:

- 1: дата(2008, январь ,11)

- +2: обед(X,Y,Z):-закуска(X),блюдо(Y),десерт(Z).
- 3: студент(«Соколов»,343, «М»).
- 4: goal обед(Zak, VL,»лимон”).

Ответ:-2

Вопрос 23

Текст вопроса: Какое из нижеприведенных утверждений является вопросом?

Шаблоны ответов:

- 1: дата(2008, январь ,11)
- 2: обед(X,Y,Z):-закуска(X),блюдо(Y),десерт(Z).
- 3: студент(«Соколов»,343, «М»).
- +4: goal обед(Zak, VL,»лимон”).

Ответ: - 4

Вопрос 24

Текст вопроса: Какая фаза выполняется после успешного завершения фазы «сопоставление с образцом»?

Шаблоны ответов:

- 1: Проверка: факт или правило
- +2: Унификация аргументов
- 3: Процесс возврата
- 4: Конкретизация переменных

Вопрос 25

Текст вопроса: Какая фаза выполняется после успешного завершения фазы «Унификация аргументов»?

Шаблоны ответов:

- +1: Проверка: факт или правило
- 2: Сопоставление с образцом
- 3: Процесс возврата
- 4: Конкретизация переменных

Вопрос 26

Текст вопроса: Какая фаза выполняется после успешного завершения фазы «Проверка: факт или правило»?

Шаблоны ответов:

- 1: Унификация аргументов
- +2: Сопоставление с образцом
- 3: Процесс возврата
- 4: Конкретизация переменных

Вопрос 27

Текст вопроса: В каких случаях начинается фаза возврата?

Шаблоны ответов:

- +1: Сопоставление с образцом закончилось неудачей
- 2: Текущий предикат имеет значение истина
- +3: Текущий предикат имеет значение ложь
- +4: В теле правила встретился предикат *fail*

Вопрос 28

Текст вопроса: В каких случаях унификация двух аргументов заканчивается неудачей?

Шаблоны ответов:

- 1: Когда оба аргумента не конкретизированные переменные
- 2: Когда один аргумент константа, а другой – не конкретизированная переменная
- +3: Когда оба аргумента конкретизированные переменные и не равны друг другу
- 4: Когда оба аргумента конкретизированные переменные и равны друг другу

Вопрос 29

Текст вопроса: В каких случаях унификация двух аргументов заканчивается успехом?

Шаблоны ответов:

- +1: Когда оба аргумента не конкретизированные переменные
- +2: Когда один аргумент константа, а другой – не конкретизированная переменная
- 3: Когда оба аргумента конкретизированные переменные и не равны друг другу
- +4: Когда оба аргумента конкретизированные переменные и равны друг другу

Вопрос 30

Текст вопроса: При возникновении процесса возврата к доказательству какого предиката осуществляется переход?

Шаблоны ответов:

- 1: К следующему за текущим предикатом
- +2: К предикату, доказывавшемуся на предыдущем шаге
- 3: К первому предикату тела правила
- 4: К предикату, стоящему в разделе GOAL

Вопрос 31

Текст вопроса: Какие переменные теряют свое значение(расконкретизируются) при возникновении процесса возврата?

Шаблоны ответов:

- 1: Все переменные
- 2: Переменные, стоящие в заголовке правила
- +3: Переменные, получившие значение на предыдущем шаге
- 4: Переменные, получившие значение на текущем шаге

Вопрос 32

Текст вопроса: Утверждение, с которым успешно согласовался текущий предикат, оказалось фактом. Что из этого следует, и к рассмотрению какого предиката перейдет интерпретатор?

Шаблоны ответов:

- 1: Следует, что нужно перейти к следующему утверждению программы.
- 2: Следует, что для доказательства предиката надо перейти к первому предикату тела правила
- +3: Следует, что текущий предикат истинен, его доказательство закончено и следует перейти к следующему за ним (если он есть) в конъюнкции предикатов.
- 4: Следует, что начнется процесс возврата.

Вопрос 33

Текст вопроса: Утверждение, с которым успешно согласовался текущий предикат, оказалось правилом. Что из этого следует, и к рассмотрению какого предиката перейдет интерпретатор?

Шаблоны ответов:

1: Следует, что нужно перейти к следующему утверждению программы.

+2: Следует, что для доказательства предиката надо перейти к первому предикату тела правила.

3: Следует, что текущий предикат истинен, его доказательство закончено и следует перейти к следующему за ним (если он есть) в конъюнкции предикатов.

4: Следует, что начнется процесс возврата.

б) типовые тестовые вопросы открытого типа

Вопрос 34

Текст вопроса: Дан фрагмент программы. Сколько раз будет напечатана буква «a»?

CLAUSES

p:-a,b.

p:-c.

p:-write("a").

a:- write("a").

a:- write("a").

b:- write("a").

b:- write("a").

c:- write("a").

h:-p,fail.

h.

Goal

h.

Ответ: 8

Вопрос 35

Текст вопроса: Какие три разных смысла заложены в программе на языке Пролог?

Ответ:

1: Декларативный смысл

3: Процедурный смысл

4: Смысл абстрактной машины

Вопрос 36

Текст вопроса: Конъюнкция предикатов в теле правила может рассматриваться как ...?

Ответ: Разбиение сложной задачи на подзадачи, решение каждой из них описывается логической функцией

Вопрос 37

Текст вопроса: Несколько утверждений с одинаковыми заголовками (т.е. у них предикаты в заголовке имеют одинаковое имя и число аргументов) могут рассматриваться как ...?

Ответ: альтернативные варианты одного и того же отношения.

Вопрос 38

Текст вопроса: Какие встроенные предикаты используются для организации повторяющихся процессов (циклов)?

Шаблоны ответов:

repeat, fail

Вопрос 39

Текст вопроса: Каково назначение предиката *fail* при организации повторяющихся процессов?

Ответ: Создать процесс возврата для получения всех ответов на запрос или всевозможных вариантов решения задачи.

Вопрос 40

Текст вопроса: Какое значение имеет предикат *fail*?

Ответ: Ложь

Вопрос 41

Текст вопроса: Чему равно минимальное число утверждений в определении повторяющегося процесса с удачным завершением?

Ответ: 2

Наименование вопроса: Вопрос 42

Текст вопроса: Зачем нужно всегда успешное завершающее утверждение при организации повторяющегося процесса?

Ответ: Чтобы предикат, описывающий повторяющийся процесс, имел значение «истина» после завершения цикла

Вопрос 43

Текст вопроса: При каких условиях можно использовать предикат *fail* для организации повторяющегося процесса?

Ответ: При условии, что в теле правила, содержащего fail, есть хотя бы один предикат, для которого в базе знаний имеется несколько (>1) вариантов согласований

Вопрос 44

Текст вопроса: При каких условиях можно использовать предикат *repeat* для организации повторяющегося процесса?

Ответ: При условии, что в теле правила стоит предикат (условие), зависящий от изменяющихся переменных. Когда этот предикат ложен, возникает возврат и повторение действий, когда истинен, цикл заканчивается

Вопрос 45

Текст вопроса: Запишите правильное определение предиката *repeat*?

repeat.
repeat:-repeat.

Вопрос 46

Текст вопроса: Что представляет собой динамическая база данных?

Ответ: Это факты, объявленные в разделе FACTS, которые можно модифицировать

Вопрос 47

Текст вопроса: Какие встроенные предикаты используются для работы с фактами динамической базы данных?

Ответ: assert, retract

Вопрос 48

Текст вопроса: Какие действия с базой данных позволяет выполнить встроенный предикат asserta(X)?

Ответ: Добавляет факт X в начало базы данных

Вопрос 49

Текст вопроса: Какие действия с базой данных позволяет выполнить встроенный предикат assertz(X)?

Ответ: Добавляет факт X в конец базы данных

Вопрос 50

Текст вопроса: Какие действия с базой данных позволяет выполнить встроенный предикат assert(X)?

Шаблоны ответов: Добавляет факт X в конец базы данных

Вопрос 51

Текст вопроса: Какие действия с базой данных позволяет выполнить встроенный предикат retract(X)?

Ответ: Удаляет факт X из базы данных

Вопрос 52

Текст вопроса: Какой из предикатов пропущен в нижеприведенном фрагменте программы вычисления суммы чисел от 1 до N ?

```
сумма(N,S):- assert(sum(0,0)),  
              repeat,  
              retract(sum(K1,S1)),  
              K2 = K1+1,  
              S2 = S1+K2,  
              ...,  
              K2 = N,  
              retract(sum(_,S)),!.
```

Ответ: asserta(sum(K2,S2))

Вопрос 53

Текст вопроса: Какой из предикатов пропущен в нижеприведенном фрагменте программы вычисления суммы чисел от 1 до N?

```

сумма(N,S):- assert(sum(0,0)),
              repeat,
              retract(sum(K1,S1)),
              K2 = K1+1,
              S2 = S1+K2,
              asserta(sum(K2,S2)),
              K2 = N,
              ...,
              !.

```

Ответ: retract(sum(_,S))

Вопрос 54

Текст вопроса: Какой из предикатов пропущен в нижеприведенном фрагменте программы вычисления $y = a^N$?

```

степень(A,N,St):- assert(st(0,1)),
                  repeat,
                  retract(st(K1,P1)),
                  K2 = K1+1,
                  ...,
                  asserta(st(K2,P2)),
                  K2 = N,
                  retract(st(_,St)),
                  !.

```

Ответ: P2 = P1 * A

Вопрос 55

Текст вопроса: Какой из предикатов пропущен в нижеприведенном фрагменте программы определения числа студентов в институте, если в базе знаний хранятся факты «студент(<фамилия>,<группа>)»?

```

количество(N):- assert(kol(0)),
                 студент(F,G),
                 retract(kol(K1)),
                 K2 = K1+1,
                 write(F),nl,
                 asserta(kol(K2)),
                 ...,
количество(N):-retract(kol(N)),
                 !.

```

Ответ: fail

Вопрос 56

Текст вопроса: Что такое рекурсия?

Ответ: Это способ описания объектов, данных, процессов или функций через самих себя

Вопрос 57

Текст вопроса: Какие виды рекурсии Вы знаете?

Ответ: Нисходящая рекурсия, Параллельная рекурсия, Восходящая рекурсия

Вопрос 58

Текст вопроса: Что такое терминальная ситуация?

Ответ: Это ситуация, в которой рекурсивное обращение предиката к самому себе прекращается

Вопрос 59

Текст вопроса: Как вычисляется результат при нисходящей рекурсии?

Ответ: При нисходящей рекурсии результат вычисляется на обратном ходе и окончательное его значение получается в вершине доказательства

Вопрос 60

Текст вопроса: Какие высказывания из нижеприведенного списка относятся к понятию восходящая рекурсия?

Шаблоны ответов: При восходящей рекурсии результат известен в момент достижения терминальной ситуации; При восходящей рекурсии параметры вычисляются на каждой стадии рекурсии в процессе выполнения прямого хода

Вопросы к экзамену по дисциплине (модулю)

1. Представление знаний с помощью фактов и правил. Структура программы.
2. Утверждения: факты, правила, вопросы.
3. Константы и переменные. Синтаксис и семантика.
4. Общая схема выполнения программы: сопоставление с образцом, унификация аргументов, процесс возврата (бектрекинг).
5. Дерево И/ИЛИ как модель выполнения программы на Прологе. Правила построения дерева.
6. Арифметические и операторные выражения. Ввод и вывод. Работа с файлами.
7. Управление выполнением программы. Организация разветвляющихся процессов.
8. Организация повторяющихся процессов.
9. Встроенные предикаты fail, repeat. Организация получения всего множества решений
10. Преобразования базы знаний с помощью встроенных предикатов.
11. Рекурсия. Виды рекурсий.
12. Восходящая рекурсия.
13. Нисходящая рекурсия.
14. Управление возвратом с помощью отсечения.
15. Списки. Ввод-вывод списков. Голова и хвост списка.
16. Операции со списками.
17. Представление информации с помощью списков.
18. Сортировка списков.
19. Строки. Представление строк в виде списков кодов литер.
20. Преобразование строк.
21. Операции со строками.
22. Ввод и вывод строк на терминал и в файл.
23. Структура. Определение структуры.
24. Обработка информации в структурах.

25. Создание структур и работа с компонентами структур.

Самостоятельная работа бакалавров

Основными видами самостоятельной работы по дисциплине являются:

- составление и отладка программ на языке Пролог в среде Visual Prolog 7;
- доработка конспекта лекции с применением учебника, методической и дополнительной литературы; изучение и конспектирование первоисточников;
- подбор иллюстраций (примеров) к теоретическим положениям, самостоятельное изучение отдельных вопросов и тем курса.

Типовые задания для самостоятельной работы (примеры)

1. Частотный словарь - это перечень слов, встречающихся в данном тексте, записанный в алфавитном порядке с указанием числа вхождений каждого слова в тексте. Написать программу составления частотного словаря по списку слов, представленных в виде строк Пролога.
2. Написать предикат замены всех английских букв русскими в строке и наоборот.
3. Написать предикат разбиения слов по слогам, который описывает следующие правила:
 - i. если в слове встречается последовательность букв $\langle \text{гласная} \rangle, \langle \text{согласная} \rangle, \langle \text{гласная} \rangle$, то слово делится после первой гласной, например: 'а_том', 'бо_рис', 'се_год_ня';
 - ii. если последовательность букв имеет вид: $\langle \text{гласная} \rangle, \langle \text{согласная} \rangle, \langle \text{согласная} \rangle, \langle \text{гласная} \rangle$, то в этом случае слово делится между согласными, например, 'раз_мер', 'пар_та'.
4. Написать предикат 'индекс'(X,Y,K), который имеет значение "истина", если K - это номер самого левого элемента в X; начиная с него Y входит в X в качестве подстроки.
5. Написать предикат сцепления строки X с самой собой N раз 'сцепление'(X,N,Y), Y - результат.

В рамках самостоятельной работы бакалаврам предлагается выполнить письменные работы по предложенным темам.

Темы для рефератов по логическому программированию

1. Парадигма логического программирования — альтернатива процедурному и функциональному программированию.
2. Различные версии языка ПРОЛОГ
3. Логическая природа решения задач на Прологе.
4. Примеры баз знаний на Прологе.
5. Моделирование интеллектуальной деятельности человека и Пролог.
6. Рекурсия — основной метод программирования на Прологе.
7. Управление логическим выводом. Отсечение.
8. Списки и структуры в решении логических задач.
9. Разработка программ с использованием графического интерфейса.
10. Разработать программы сортировки списков на языке Visual Prolog.
11. Представление деревьев на языке Visual Prolog.
12. Символьные преобразования на языке Visual Prolog.
13. Эволюция систем программирования на Прологе.