

Компонент ОПОП  
Направленность (профиль)

**09.03.02 Информационные системы и технологии**  
**Информационные системы и технологии**  
**искусственного интеллекта**

Б1.В.04

шифр дисциплины

### ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Дисциплины

Функциональное и логическое программирование

Разработчик (и):

Шиманский С.А.

ФИО

доцент

должность

ученая степень, звание

Утверждено на заседании кафедры

наименование кафедры

протокол № 6 от 17.02.2025 г.

Заведующий кафедрой ИТ

Ляш О.И.



подпись

ФИО

### 1. Критерии и средства оценивания компетенций и индикаторов их достижения, формируемых дисциплиной

Код и наименование компетенции	Код и наименование индикатора (-ов) достижения компетенции	Результаты обучения по дисциплине			Оценочные средства текущего контроля	Оценочные средства промежуточной аттестации
		Знать	Уметь	Владеть		
ПК-5 Способен разрабатывать и тестировать программные компоненты решения задач в системах искусственного интеллекта	ИД-1 <sub>ПК-5</sub> Настраивает программное обеспечение и участвует в разработке программных компонентов систем искусственного интеллекта ИД-2 <sub>ПК-5</sub> Разрабатывает приложения систем искусственного интеллекта ИД-3 <sub>ПК-5</sub> Проводит тестирование систем искусственного интеллекта	теоретические основы функционального и логического программирования синтаксис языков программирования, изучаемых в курсе; возможности интегрированных сред программирования	использовать основные средства одного из языков функционального и логического программирования; разрабатывать прикладные программы с помощью интегрированных сред программирования; отлаживать и тестировать создаваемые программы, используя диагностические возможности среды разработки	языками логического и функционального программирования для реализации знаний и умений	- комплект заданий для выполнения лабораторных работ; - тестовые задания; - типовые задания по вариантам для выполнения расчетно-графической работы	Экзаменационные билеты Результаты текущего контроля

## 2. Оценка уровня сформированности компетенций (индикаторов их достижения)

Показатели оценивания компетенций (индикаторов их достижения)	Шкала и критерии оценки уровня сформированности компетенций (индикаторов их достижения)			
	Ниже порогового («неудовлетворительно»)	Пороговый («удовлетворительно»)	Продвинутый («хорошо»)	Высокий («отлично»)
Полнота знаний	Уровень знаний ниже минимальных требований. Имели место грубые ошибки.	Минимально допустимый уровень знаний. Допущены не грубые ошибки.	Уровень знаний в объёме, соответствующем программе подготовки. Допущены некоторые погрешности.	Уровень знаний в объёме, соответствующем программе подготовки.
Наличие умений	При выполнении стандартных заданий не продемонстрированы основные умения. Имели место грубые ошибки.	Продemonстрированы основные умения. Выполнены типовые задания с негрубыми ошибками. Выполнены все задания, но не в полном объёме (отсутствуют пояснения, неполные выводы)	Продemonстрированы все основные умения. Выполнены все основные задания с некоторыми погрешностями. Выполнены все задания в полном объёме, но некоторые с недочётами.	Продemonстрированы все основные умения. Выполнены все основные и дополнительные задания без ошибок и погрешностей. Задания выполнены в полном объёме без недочётов.
Наличие навыков (владение опытом)	При выполнении стандартных заданий не продемонстрированы базовые навыки. Имели место грубые ошибки.	Имеется минимальный набор навыков для выполнения стандартных заданий с некоторыми недочётами.	Продemonстрированы базовые навыки при выполнении стандартных заданий с некоторыми недочётами.	Продemonстрированы все основные умения. Выполнены все основные и дополнительные задания без ошибок и погрешностей. Продemonстрирован творческий подход к решению нестандартных задач.
Характеристика сформированности компетенции	Компетенции фактически не сформированы. Имеющихся знаний, умений, навыков недостаточно для решения практических (профессиональных) задач.  ИЛИ Зачетное количество баллов не набрано согласно установленному диапазону	Сформированность компетенций соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач.  ИЛИ Набрано зачетное количество баллов согласно установленному диапазону	Сформированность компетенций в целом соответствует требованиям. Имеющихся знаний, умений, навыков достаточно для решения стандартных профессиональных задач.  ИЛИ Набрано зачетное количество баллов согласно установленному диапазону	Сформированность компетенций полностью соответствует требованиям. Имеющихся знаний, умений, навыков в полной мере достаточно для решения сложных, в том числе нестандартных, профессиональных задач.  ИЛИ Набрано зачетное количество баллов согласно установленному диапазону

### 3. Критерии и шкала оценивания заданий текущего контроля

#### 3.1. Критерии и шкала оценивания лабораторных работ

Перечень лабораторных работ, описание порядка выполнения и защиты работы, требования к результатам работы, структуре и содержанию отчета и т.п. представлены в электронном курсе в ЭИОС МАУ.

Оценка/баллы	Критерии оценивания
<i>Отлично</i>	Задание выполнено полностью и правильно. Отчёт по лабораторной работе подготовлен качественно в соответствии с требованиями. Полнота ответов на вопросы преподавателя при защите работы
<i>Хорошо</i>	Задание выполнено полностью, но нет достаточного обоснования или при верном решении допущена незначительная ошибка, не влияющая на правильную последовательность рассуждений. Все требования, предъявляемые к работе, выполнены
<i>Удовлетворительно</i>	Задания выполнены частично с ошибками. Демонстрирует средний уровень выполнения задания на лабораторную работу. Большинство требований, предъявляемых к заданию, выполнены
<i>Неудовлетворительно</i>	Задание выполнено со значительным количеством ошибок на низком уровне. Многие требования, предъявляемые к заданию, не выполнены. ИЛИ Задание не выполнено

#### 3.2. Критерии и шкала оценивания расчетно-графической работы

Перечень заданий, рекомендации по выполнению представлены в методических материалах по освоению дисциплины и в электронном курсе в ЭИОС МАУ.

Предусмотрена 1 РГР, направленная на проверку знаний/умений/навыков по теме «Решение задач на языках логического и функционального программирования».

В ФОС включён типовой вариант РГР.

*Типовой вариант РГР*

1. Построить рекурсивную функцию для вычисления  $n$ -го члена последовательности, в которой каждый следующий член равен произведению двух предыдущих, а первые 2 члена равны соответственно 1 и 2. (Например: 1 2 2 4 8 32 ...)

2. Написать программу, определяющую самую часто встречающуюся букву в заданном английском тексте. Считать, что строчные и прописные буквы одинаковы ( $a=A$ ,  $b=B$ ,  $c=C$  и т. д.)

3. Выделить в дереве все поддеревья с указанной меткой

4. Дано действительное  $b > 0$ . Последовательность  $a_1, a_2, \dots$  образована по следующему закону  $a_1 = b$ ,  $a_i = a_{i-1} - 1/\sqrt{i}$ ,  $i = 2, 3, \dots$ . Найти первые  $k$  отрицательных членов последовательности  $a_1, a_2, \dots$

5. Определить последовательность, в которой каждый элемент – список всех простых делителей соответствующего целого числа.

6. Приведите пример нетривиальных выражений, принадлежащих типу согласно варианту:

- ((Char, Integer), String, [Double])
- [(Double, Bool, (String, Integer))]
- ([Integer], [Double], [(Bool, Char)])
- [[[(Integer, Bool)]]]
- (((Char, Char), Char), [String])
- ([[Double], [Bool]], [Integer])
- [(Integer, (Integer, [Bool]))]
- (Bool, ([Bool], [Integer]))
- [[(Bool], [Double])]
- [[(Integer], [Char])]

7. Напишите функцию `strPosMy :: Eq a => a -> [a] -> [int]` которая находит все вхождения первого списка во второй и выводит список номеров элементов, с которого это вхождение начинается; с помощью функций `length`, `(=)`, `taken` `drop`.

8. Определить функцию `sort3`, по трём целым возвращающую отсортированный по возрастанию кортеж из этих значений.

9. С помощью рекурсии опишите функцию, которая для данного  $n$  вычисляет  $n + 10(n - 1 + 10(n - 2 + \dots + 10(3 + 10(2 + 10 \cdot 1)) \dots))$ .

10. Пусть производится работа со списками чисел, причём известно, что все числа в списке не меньше  $-100$  и не больше  $100$ .

- с помощью рекурсии опишите функцию, которая ищет наибольшее число в списке;
- опишите эту же функцию, не используя рекурсию, с помощью стандартной функции `foldr` (или `foldl`).

### Критерии и шкала оценивания РГР

№ п/п	Критерий	Характеристики содержания и результатов работы	Оценка
1	2	3	4
1	Качество программной и алгоритмической реализации заданий	Выполнены все задачи. Алгоритмическая и программная реализации полностью отвечают требованиям. Программа демонстрирует устойчивую работу на тестовых наборах исходных данных, подготовленных обучающимся, обрабатывает все исключительные ситуации.	5
		Выполнены все задачи. Алгоритмическая и программная реализации в целом отвечают требованиям, изложенным в методических указаниях. Имеются одна-две несущественные ошибки в блок-схемах. Программа демонстрирует устойчивую работу на тестовых наборах исходных данных, подготовленных обучающимся, но обрабатывает не все исключительные ситуации.	4
		Выполнены все задачи. Имеются две-три существенные ошибки в блок-схемах. Программа работает неустойчиво, при выполнении некоторых задач (или не решена одна из четырех задач), не обрабатывает исключительные ситуации, тестовые наборы исходных данных неполные.	3
		Большое количество существенных ошибок по сути работы и программа находится практически в нерабочем состоянии.	2
		Программа не разработана ИЛИ представлен чужой вариант задания.	0
2	Содержание пояснительной записки	В пояснительной записке включены все требуемые разделы. Во «Введении» раскрыта суть технологии нисходящего структурного программирования и приведены оригинальные примеры задач, при решении которых возникает необходимость обработки массивов данных. Даны ссылки на все заимствованные материалы.	5
		В пояснительной записке включены все требуемые разделы. Во «Введении» в основном раскрыта суть технологии нисходящего структурного программирования и приведены типовые примеры задач, при решении которых возникает необходимость обработки массивов данных. Даны ссылки не на все заимствованные материалы	4
2	Содержание пояснительной записки	Содержание пояснительной записки в целом соответствует заданию. Есть несущественные замечания по содержанию разделов пояснительной записки. Во «Введении» недостаточно полно раскрыта суть технологии нисходящего структурного программирования и не приведены примеры задач, при решении которых возникает необходимость обработки массивов данных. Даны ссылки не на все заимствованные материалы.	3
		Содержание работы в целом не соответствует заданию. Отсутствуют некоторые разделы пояснительной записки. Имеются ошибки в использовании терминов. Отсутствуют ссылки на все заимствованные материалы.	0

1	2	3	4
3	Оформление	Соответствует в целом предъявленным требованиям.	3
		Имеются одна-две однотипные незначительные ошибки в оформлении, например, пробел после скобок или кавычек, не оформлено продолжение таблицы, подрисовочная запись не центрирована.	2
		Имеется большое количество однотипных незначительных ошибок в оформлении.	1
		Записка небрежно оформлена и содержит большое количество грамматических ошибок. Не выполнены требования по оформлению заголовков разделов, рисунков, таблиц, списка литературы, ссылок на рисунки, таблицы, источники информации.	0
4	Сроки выполнения	РГР выполнена, представлена на проверку и на защиту в установленные сроки.	3
		Работа выполнена с нарушениями сроков сдачи на проверку на 1-2 дня. Защита выполняется в установленные сроки.	2
		РГР представлена на защиту без предварительной проверки преподавателем.	1
		РГР не представлена на защиту в установленные сроки.	0
		Дополнительные баллы за систематическую работу в течение семестра и сдачу 1–5 этапов работы в сроки, указанные в табл. 2. При этом содержание разделов, алгоритмическая и программная реализации в основном отвечают требованиям.	Максимум 5 баллов
5	Защита	При защите работы обучающийся правильно и уверенно отвечает на все заданные ему вопросы, демонстрирует глубокое знание теоретического материала, способен аргументированно объяснить действия, принятые при выполнении заданий, не испытывает трудностей в ответах на предложения незначительно изменить постановку задачи.	5
5	Защита	При защите работы обучающийся правильно и уверенно отвечает на большинство заданных ему вопросов, демонстрирует хорошее знание теоретического материала, но не всегда аргументированно может объяснить действия принятые при выполнении заданий. Испытывает трудности в ответах на предложения незначительно изменить постановку задачи. При наводящих вопросах преподавателя исправляет ошибки в ответе.	4
		При защите работы обучающийся допускает грубые ошибки при ответах на заданные вопросы и /или не дал ответ более чем на 30% вопросов.	3
		Обучающийся плохо ориентируется в представленных на защиту алгоритмах и программном комплексе. При защите РГР обучающийся демонстрирует слабое понимание материала изучаемой дисциплины. Правильно отвечает менее чем на 50 % заданных вопросов.	2

Перевод рейтинговой оценки в традиционную осуществляется на основании следующих соотношений:

- 21–26 – «отлично»,
- 16–20 – «хорошо»,
- 11–15 – «удовлетворительно»,
- < 11 – «неудовлетворительно».

#### 4. Критерии и шкала оценивания результатов обучения по дисциплине при проведении промежуточной аттестации

##### 4.1. Критерии и шкала оценивания результатов освоения дисциплины с экзаменом

Результат промежуточной аттестации складывается из баллов, набранных в ходе текущего контроля и при проведении экзамена:

В ФОС включён список вопросов и заданий к экзамену и типовой вариант экзаменационного билета:

*Вопросы к экзамену по дисциплине «Логическое и функциональное программирование», 6 семестр.*

1. Понятие декларативного программирования. Особенности логического программирования, его отличие от императивного и функционального программирования.
2. Язык Prolog: назначение, особенности, структура программы.
3. Основные типы утверждений в языке Prolog: факты, вопросы, правила.
4. Согласование целевых утверждений: процедура доказательства, понятия связывания и унификации.
5. Механизм отката: назначение, последовательность действий.
6. Списки в языке Prolog: доступ к элементам, методы обработки, примеры программ, выполняющих типовые операции по обработке списков: замена первого элемента списка (`replace_first`), определения вхождения в список (`member`), определения длины списка (`length`).
7. Списки в языке Prolog. Примеры отношений обработки списков: конкатенации списков (`append`), удаление элемента (`delete`), обращение списка (`reverse`).
8. Отсечения: механизм обработки, преимущества, недостатки. Зелёное и красное отсечения. Можно на примере (например, добавления в список отсутствующего элемента).
9. Реализация вычислений на языке Prolog. Привести решение одной (любой из этих задач):
  - реализация рекурсивных функций на примере вычисления факториала;
  - или
10. реализация рекурсивных функций на примере вычисления  $n$ -го числа Фибоначчи.
11. Применение логического программирования для решения задачи о волке, козе и капусте.
12. Общие свойства функциональных языков: краткость, модульность, строгая типизация, чистота, ленивые вычисления.
13. Хвостовая рекурсия. Накапливающие параметры. Пример их использования.
14. Смысл конструкции `let` или `where`. Пример использования. Двумерный синтаксис (`off-side rule`). Его правила.
15. Тип, параметризованный другим типом. Определите тип `Tree` (дерево), который для любого типа определяет дерево, содержащее значения этого типа.
16. Лямбда-выражение. Приведите пример его использования. Замыкание (`closure`). Нелокальные переменные в определении функции.
17. Карринг / каррирование (`currying`). Сечение. Приведите пример использования. Суперпозиция функций. Функции `foldr` и `foldl`.
18. List comprehension. Приведите примеры использования. Ленивый порядок вычислений (`lazy evaluation`). Покажите на каком-нибудь примере, чем порядок вычислений при `lazy evaluation` отличается от обычного. Какие конструкции и понятия из обычного (не функционального) программирования можно, в каком-то смысле, назвать аналогами ленивого выполнения в Haskell'e? Привести любые два примера. В качестве одного из них желательно (но не обязательно) объяснить, что такое идиома `Copy On Write`.
19. Бесконечные структуры данных. «Завязывание узлов» (`tying the knots`). `Deriving`. Приведите примеры их использования.
20. Автоматический вывод типа функции (алгоритм Хиндли-Милнера) на примере какой-нибудь функции. Приём «Представление множеств с помощью функций» Примеры.
21. `Failure continuation`. `Continuation-passing style`. Приведите пример использования.
22. Функция `>>=` для списков. Приведите какой-нибудь пример её использования. Монады. Приведите какие-нибудь два примера монад.
23. Как в чистом лямбда-исчислении можно моделировать целые числа (числа Чёрча)? Приведите пример определения какой-нибудь арифметической операции. Как по

числу Чёрча получить соответствующее ему целое число? Бета-редукция. Сложные случаи при строгом определении бета-редукции.

24. Нормальная форма лямбда-выражений. Нормальный и аппликативный порядки применения редукций. Свойства нормального порядка.

25. Конфлюентность. Теорема о кофлюентности и единственности нормальной формы. Комбинатор неподвижной точки. Пример его использования. Y-комбинатор.

26. Изоморфизм Карри-Ховарда. «Бесплатные теоремы».

#### Темы практических заданий

1. Реализовать функцию `reverse` (обращение списка. Например, `reverse [1, 2, 3] = [3, 2, 1]`). Функция должна работать за линейное время.

2. Определить функцию, аналогичную `foldr`, для деревьев.

3. С помощью катаморфизма для деревьев описать функцию, вычисляющую высоту данного дерева.

4. Перечислите конструкции, которые можно использовать в `list comprehension` (подсказка: их 3 шт.).

5. Пусть дано представление матриц с помощью списков списков. Описать функцию, которая для данного  $n$  возвращает матрицу, изображающую таблицу умножения  $n$  на  $n$ , т. е. матрицу, у которой на пересечении строки  $i$  и столбца  $j$  содержится число  $ij$ .

6. С помощью `list comprehension` решите задачу: «Для данного  $n$  создать список из всех троек  $(i, j, k)$  таких, что  $5i+3j+2k=n$ . Желательно при этом не перебирать лишние варианты.»

7. Определите сортировку списка на основе алгоритма `quicksort` с помощью `list comprehension`.

8. Используя «завязывание в узел», опишите бесконечный список из чисел, состоящих только из цифр 4,5,7. Числа должны идти в порядке возрастания.

9. Какие типы имеют функции `length`, `zip`, `map`, `foldr`?

10. Приведите какой-нибудь пример описания класса в Haskell и какой-нибудь пример того, как объявить `instance` класса.

В билет входят 2 вопроса и практическое задание.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«МУРМАНСКИЙ АРКТИЧЕСКИЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ И ЦИФРОВЫХ ТЕХНОЛОГИЙ

Кафедра информационных технологий

направление подготовки 09.03.02 «Информационные системы и технологии»  
профиль «Информационные системы и технологии искусственного интеллекта»

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

по учебной дисциплине «Функциональное и логическое программирование», 6 семестр

1. Понятие декларативного программирования. Особенности логического программирования, его отличие от императивного и функционального программирования
2. Общие свойства функциональных языков: краткость, модульность, строгая типизация, чистота, ленивые вычисления
3. Практическое задание.

Заведующий кафедрой \_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

Ответы на экзаменационные вопросы и задания оцениваются по критериям и шкале, представленным в таблице:

Оценка	Баллы	Критерии оценки ответа на экзамене
<i>Отлично</i>	20	Обучающийся глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, не затрудняется с ответом при видоизменении вопроса, без ошибок выполнил практическое задание. Владеет специальной терминологией, демонстрирует общую эрудицию в области теории и практики баз данных, использует при ответе ссылки на материал специализированных источников.
<i>Хорошо</i>	15	Обучающийся твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, владеет специальной терминологией на достаточном уровне; могут возникнуть затруднения при ответе на уточняющие вопросы по рассматриваемой теме; в целом демонстрирует общую эрудицию в предметной области. Выполнил практическое задание без грубых ошибок.
<i>Удовлетворительно</i>	10	Обучающийся имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, плохо владеет специальной терминологией, допускает существенные ошибки при ответе. Выполнил практическое задание с грубыми ошибками.
<i>Неудовлетворительно</i>	< 10	Обучающийся не знает значительной части программного материала, допускает существенные ошибки, нарушения логической последовательности в изложении программного материала, не владеет специальной терминологией. Нет ответа на поставленный вопрос. Не выполнил практическое задание.

Оценка, полученная на экзамене, переводится в баллы («5» – 20 баллов, «4» – 15 баллов, «3» – 10 баллов) и суммируется с баллами, набранными в ходе текущего контроля:

Сформированность компетенций ПК-5	Итоговая оценка по дисциплине	Суммарные баллы по дисциплине, в том числе	Критерии оценивания
Высокий	<i>Отлично</i>	91–100	Выполнены все контрольные точки текущего контроля на высоком уровне. Экзамен сдан
Продвинутый	<i>Хорошо</i>	81–90	Выполнены все контрольные точки текущего контроля. Экзамен сдан
Пороговый	<i>Удовлетворительно</i>	70–80	Контрольные точки выполнены в неполном объеме. Экзамен сдан.
Ниже порогового	<i>Неудовлетворительно</i>	69 и менее	Контрольные точки не выполнены или не сдан экзамен

5. Задания диагностической работы для оценки результатов обучения по дисциплине в рамках внутренней независимой оценки качества образования

ФОС содержит задания для оценивания знаний, умений и навыков, демонстрирующих уровень сформированности компетенций и индикаторов их достижения в процессе освоения дисциплины.

Комплект заданий разработан таким образом, чтобы осуществить процедуру оценки каждой компетенции, формируемых дисциплиной, у обучающегося в письменной форме.

Содержание комплекта заданий включает: тестовые задания, расчетные задачи, мини-кейсы, ситуационные задания, практико-ориентированные задания.

### Комплект заданий диагностической работы

Оценочные материалы содержат задания для оценивания знаний, умений и навыков, демонстрирующие уровень сформированности компетенций.

Контрольные задания соответствуют принципам валидности, однозначности, надежности и позволяют объективно оценить результаты обучения и уровни сформированности части компетенций ПК-5.

5.1. Комплекс заданий сформирован таким образом, чтобы осуществить процедуру проверки одной компетенции (части компетенции) у обучающегося в течение 5–10 минут в письменной или устной формах.

Содержание комплекса заданий (вариант создаётся выборкой 20 случайных вопросов из списка).

1 Анонимная переменная на языке Prolog обозначается символом:

\$

#

–

2 Как называется механизм логического вывода в экспертных системах, в котором на основании начальных фактов строится заключение?

комбинированный вывод

прямой вывод

обратный вывод

3 Какой предикат языка программирования Prolog позволяет удалить указанное правило из базы данных программы в процессе её выполнения:

asserta

retractall

append

4 Укажите обозначение логической связки отрицания:

¬

∨

∧

5 В продукционных экспертных системах листья дерева представления знаний соответствуют: элементарным фактам, которые являются исходными для решения задачи

конфликтным множествам

продукционным правилам

6 Какой алгоритм поиска решения устанавливает при каждом итерационном заглублиении определённый диапазон возможных длин пути?

алгоритм градиентного спуска

алгоритм поиска решения ID

алгоритм поиска решения A

алгоритм поиска решения A\*

алгоритм поиска решения IDA\*

7 Укажите префиксный оператор языка Prolog:

not x

x + y

x!

8 Формула  $F$  называется общезначимой, если:

она ложна во всех интерпретациях

она истинна во всех интерпретациях

она истинна хотя бы в одной интерпретации

9 Двойное отрицание с использованием предиката not в Prolog:

всегда генерирует значения

может генерировать значения  
не генерирует значений

10 Укажите бестиповой язык программирования:

C#  
C++  
Prolog  
Haskell  
Pascal  
Basic  
Barsic  
Brainfuck

11 Укажите метод моделирования искусственного интеллекта, основанный на имитации рассуждений и логики человека:

нейрокибернетический  
семиотический  
эмерджентный  
генетический

12 Укажите определение упорядоченной стратегии резолюции:

стратегия, при которой значение имеет порядок литер в дизъюнктах, который сохраняется при применении правила вывода  
стратегия, при которой литера для применения резолюции выбирается в соответствии с некоторым правилом  
стратегия, при которой на каждом шаге резолюции в качестве одной из посылок используется формула, полученная на предыдущем

13 Укажите синтаксис Mercury для объявления функционального типа:

```
:-type t == T1 * T2  
:-type t == T1 -> T2  
:-type t == int  
:-type t -> op1(T1);op2(T2)
```

14 Как называется грамматика в классификации по Хомскому, если все правила грамматики имеют вид  $\alpha \rightarrow a$  либо  $\alpha \rightarrow \beta$ , где  $\alpha, \beta \in N$ ,  $a \in A$ ?

регулярная грамматика  
контекстно-зависимая грамматика  
контекстно-свободная грамматика  
рекурсивно-перечислимая грамматика

15 Абстракция - это ...

применение функции высшего порядка к аргументу  
преобразование функции от пары аргументов в функцию, берущую свои аргументы по одному  
рассмотрение выражения как функции от некоторого аргумента

16 Укажите, какая из представленных ниже записей на языке Prolog будет являться правилом:

```
studied_technical(X):-studied(X,mathematics)  
studied(petya,mathematics)  
studied(vasya,literature)
```

17 Какой из языков программирования является функциональным?

qlisp  
lisp  
pascal

python

18 SLDNF-резолуция ...

всегда заканчивается успехом

не обладает свойством полноты

обладает свойством полноты

19 Укажите, какая из представленных ниже записей на языке Prolog, будет являться фактом:

`studied_technical(X):-studied(X,mathematics)`

`studied(petya,mathematics)`

`speciality(X,programmer):-studied(X,mathematics),studied(X,compscience)`

20 Целевое утверждение, которое приводит к вызову правила, содержащего отсечение, называется:

родительским целевым утверждением

фактом

дочерним целевым утверждением

21 Если в весовой функции алгоритма  $A$  принять  $h(x) = 0$ , то:

получим алгоритм поиска с весовой функцией UCS

получим алгоритм IDA\*

получим алгоритм A\*

22 Укажите особенность логических языков программирования:

применение функции к аргументам изменяет данные

программа задаёт множество возможных переходов в пространстве поиска

выполнение операторов изменяет состояние памяти

программа не содержит алгоритма

23 Термы в Prolog бывают:

простые

сдвоенные

структурные

24 Формула представлена в конъюнктивной нормальной форме, если:

она представлена в виде  $(Q_1x_1) \dots (Q_nx_n)(A)$ , где  $Q_i$  - это квантор существования или общности,

а формула  $A$  не содержит кванторов

она находится в предварённой нормальной форме и не содержит кванторов существования

это конъюнкция конечного числа дизъюнктов

25 Онтологический язык OWL Lite соответствует логике:

shoin(d)

shif(d)

shiq(d)

26 Укажите формат записи и передачи данных RDF:

prolog

html

rdf/xml

27 Аналогом какого предиката является в данном фрагменте программного кода на языке Prolog

предикат  $x ? x(P) :- P, !, fail.x(\_)$ .

repeat

member

not

28 Некоторое множество цепочек  $L \subseteq A^*$  в алфавите  $A$   
грамматикой  
кортежем  
языком  
подстрокой

29 На языке программирования Prolog константы начинаются:  
с символа \$  
со строчной буквы  
с заглавной буквы

30 Грамматика называется контекстно-свободной, если  
для всех правил  $\alpha \rightarrow \beta$  выполняется  $|\alpha| \leq |\beta|$   
все правила грамматики имеют вид  $\alpha \rightarrow a$  либо  $\alpha \rightarrow \beta$ , где  $\alpha, \beta \in N$ ,  $a \in A$   
она задается правилами общего вида  
все правила имеют вид  $\alpha \rightarrow \theta$ , где  $\alpha \in N$ ,  $\theta \in (A \cup N)^*$

31 Парадигма программирования, которая описывает процесс вычисления в виде инструкций, изменяющих состояние программы, называется:  
функциональное программирование  
императивное программирование  
логическое программирование

32 Программа является декларативной, если она:  
является императивной  
описывает как получить результат  
описывает каков должен быть результат

33 Парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании последних, называется:  
логическое программирование  
функциональное программирование  
императивное программирование

34 Императивными языками программирования не являются:  
только функциональные языки  
только логические языки  
функциональные и логические языки  
объектно-ориентированные языки  
процедурные языки  
языки высокого уровня  
языки поколения 4GL

35 Логические языки программирования оперируют:  
данными  
состоянием памяти  
пространством поиска решений

36 Определите, чем является указанное ниже предложение на языке Prolog:  
`studied(petya,english)`  
внешняя цель  
факт  
правило

37 Какой из языков программирования является императивным?

pascal  
lisp  
qlisp

38 Язык программирования F# является:

логическим  
императивным  
функциональным

39 Определите, чем является указанное ниже предложение на языке Prolog: `speciality(X,lit_translator) :- studied_languages(X),studied(X,literature)`

правило  
факт  
внешняя цель

40 Язык программирования Prolog является:

императивным  
функциональным  
логическим

41 Базы данных, снабженные средствами логического программирования для вывода дополнительных фактов:

индуктивные базы данных  
реляционные базы данных  
дедуктивные базы данных

42 Укажите условное обозначение общезначимой формулы:

$A \vdash B$   
 $A \vdash B$   
 $\vdash A$

43 Если формула  $A$  истинна хотя бы в одной интерпретации, то она называется:

общезначимой  
достоверной  
выполнимой

44 Укажите условное обозначение утверждения, что формула  $B$  следует из формулы  $A$ .

$A \rightarrow B$   
 $A \vdash B$   
 $A \vdash B$

45 Исчисление называется формально непротиворечивым, если:

любая общезначимая формула  $A$  выводима  
если любая выводимая формула является общезначимой  
не существует такой формулы  $A$ , что  $\vdash A$  и  $\vdash \neg A$

46 Укажите теорему Гёделя о неполноте:

чистое исчисление предикатов первого порядка полно, то есть любая общезначимая формула выводима  
любая прикладная теория первого порядка, содержащая формальную арифметику, не является полной теорией  
не существует алгоритма установления общезначимости произвольной формулы логики предикатов

47 Формулы произвольной формальной системы - это ...

некоторое подмножество всех слов, которые можно образовать из символов, входящих в алфавит  
некоторое подмножество всех слов, которые можно образовать из правил вывода системы  
бесконечное множество всех слов, которые можно образовать из символов, входящих в алфавит

48 Исчисление разрешимо, если:

это конъюнкция конечного числа дизъюнктов

существует алгоритм определения того, является ли заданная формула общезначимой или нет

не существует алгоритма определения того, является ли заданная формула общезначимой или нет

49 Резолюция для логики предикатов обладает свойством полноты и непротиворечивости только для:

кванторов Хорна

дизъюнктов Гегеля

дизъюнктов Хорна

50 Подстановка называется унификатором для формул или термов  $F$  и  $G$ , если:

$F: F = (F)$

$F = G$

$F \neg$

51 Формула находится в сколемовской нормальной форме, если:

это конъюнкция конечного числа дизъюнктов

она представлена в виде  $Q_1x_1, \dots, Q_nx_nA$ , где  $Q_i$  -- квантор существования или общности, а формула  $A$  не содержит кванторов

она находится в предваренной нормальной форме и не содержит кванторов существования

52 Константы в языке Prolog подразделяются на:

числа

атомы

переменные

53 На языке программирования Prolog переменные начинаются:

со строчной буквы

с символа \$

54 В программе Prolog переменная может изменить свое значение:

в процессе отката и во время повторной унификации

в процессе отката

во время повторной унификации

55 Свободная переменная в программе на языке Prolog:

унифицируется только как значение, с которым она связана

унифицируется без ограничений

не унифицируется

56 Какой встроенный предикат языка Prolog всегда завершается неуспешно?

false

fail

true

57 Укажите предикат языка Prolog, который используется для определения принадлежности элемента списку:

members

list\_members

member

58 Укажите предикат языка Prolog, который используется для конкатенации списков:

append  
permute  
member

59 С точки зрения быстродействия случайного доступа к элементу:

более эффективно использование массива  
у списка и массива одинаковая эффективность  
более эффективно использование списка

60 Какой встроенный предикат на языке Prolog открывает файл на вывод?

telling  
tell  
see  
seeing

61 С точки зрения быстродействия удаления элемента из середины:

более эффективно использование списка  
более эффективно использование массива  
у списка и массива одинаковая эффективность

62 Укажите предикат языка Prolog, который используется для отображения списка:

map  
fold  
filter

63 Предикат `not(P)` в языке программирования Prolog завершается успехом, если:

удается найти все возможные доказательства  $P$   
удается найти хотя бы одно доказательство  $P$   
доказательство  $P$  неуспешно

64 Предикат `repeat` языка программирования Prolog:

всегда заканчивается успехом  
может заканчиваться как успехом, так и неуспехом  
всегда заканчивается неуспехом

65 Процедурная семантическая модель программы на языке Prolog:

рассматривает правила как последовательность шагов, которые необходимо успешно выполнить для того, чтобы соблюдалось отношение, приведенное в заголовке правила  
рассматривает отношения, определенные в программе, при этом порядок следования предложений в программе и условий в правиле не важен  
рассматривает отношения, не определенные в программе

66 Какое из утверждений соответствует дополнительному правилу SLDNF-резолюции:

$\neg A$  заканчивается неуспехом тогда и только тогда, когда  $A$  не имеет конечного успешного дерева вывода  
 $\neg A$  заканчивается неуспехом тогда и только тогда, когда  $A$  имеет конечное успешное дерево вывода  
 $\neg A$  заканчивается успехом тогда и только тогда, когда  $A$  имеет конечное успешное дерево вывода

67 Предикат отрицания `not` в Prolog:

может генерировать значения  
не генерирует значений  
всегда генерирует значения

68 Укажите результат запроса `?- sweet(X),white(X)`. для представленного ниже набора фактов на языке Prolog: `sweet(sugar).white(sugar).white(paper)`.

yes  
x = sugar  
no  
x = paper  
x = paper

69 Как называются отсечения, при отбрасывании которых программа продолжает выдавать те же решения, что и при наличии отсечения?

синие  
зелёными  
Верный ответ)  
красными  
белыми  
чёрными  
бесцветными  
жёлтыми

70 Укажите, какой вид отсечения используется в приведенном ниже фрагменте кода на Prolog:

`max(X, Y, X) :- X > Y, !.max(X, Y, Y) :- X <= Y. ?`

красное отсечение  
зелёное отсечение  
в данном фрагменте нет отсечения  
белое отсечение  
чёрное отсечение  
бесцветное отсечение  
синее отсечение  
жёлтое отсечение

71 Встроенный предикат отсечения на языке Prolog обозначается символом:

#  
\_  
!  
\$

72 Аналогом какого оператора императивного языка является приведенный ниже код на языке программирования Prolog: `W :- A, P, W.W :- !. ?`

while  
if  
for

73 Укажите ответ на запрос `?- num_legs(centipede,X)`. для приведенной ниже программы Prolog: `num_legs(centipede,40):-!.num_legs(human,2):-!.num_legs(_,4)`.

X=4  
no  
X=2  
X=40  
yes

74 Выберите верные утверждения для языков программирования с нестрогой типизацией данных.  
проверяется строгое соответствие типов  
переменные могут принимать значения различной природы  
тип данных нужен в основном для определения объема памяти

75 На каком этапе происходит динамический контроль типов данных в программе?  
на этапе выполнения

на этапе компиляции  
на этапе компиляции и выполнения

76 Укажите особенность динамического контроля типов данных в программе.  
возможность моделирования на бестиповых языках  
в традиционных языках программирования требует избыточного описания типов в программе  
требует более жесткой системы типизации

77 Какой объем оперативной памяти ориентировочно занимает алгоритм поиска в ширину?  
соответствующий среднему ветвлению на каждом шаге алгоритма в степени средней длины пути  
алгоритма поиска  
соответствующий максимальной длине пути алгоритма поиска  
соответствующий среднему ветвлению на каждом шаге алгоритма поиска

78 Укажите, какие высказывания верны при алгоритме поиска решения задачи в ширину.  
первым находится необязательно кратчайший путь  
первым находится кратчайший путь  
можно найти только пути решения без петель

79 Определите детерминизм предиката Prolog, у которого максимальное количество решений больше единицы, а отказ невозможен.  
det  
semidet  
multi  
failure

80 Какой режим доказательства предиката `add(X, Y, Z)` при конкретизированных  $X, Z$ ?  
построение всех комбинаций  
проверка верности суммы  
вычитание  
сложение

81 Укажите синтаксис Mercury для объявления типа `string`.  
`:-type t == str`  
`:-type t == char`  
`:-type t == string`

82 Укажите синтаксис Mercury для объявления декартова произведения.  
`:-type t == int`  
`:-type t == T1 * T2`  
`:-type t --> op1(T1); op2(T2)`  
`:-type t == T1 -> T2`

83 Определите детерминизм предиката Prolog, у которого максимальное количество решений равно единице, а отказ невозможен:  
det  
semidet  
multi  
failure

84 Если алгоритм поиска  $A$  является допустимым, он называется:  
алгоритм  $A\#$   
алгоритм  $A^*$   
алгоритм  $A\_$

85 Какой предикат языка программирования Prolog позволяет добавить указанное правило в базу данных программы в процессе ее выполнения?

retractall  
asserta  
append

86 Укажите особенность жадного алгоритма поиска.

не учитывает оптимальность пути с точки зрения эвристики и не учитывает вес пути  
учитывает оптимальность пути с точки зрения эвристики, но не учитывает вес пути  
не учитывает оптимальность пути с точки зрения эвристики, но учитывает вес пути

87 В каком алгоритме поиска в качестве весовой функции при выборе направления поиска принимается функция  $f^*(X) = g^*(X) + h^*(X)$ ?

алгоритм A  
алгоритм A\*  
жадный алгоритм

88 В весовой функции алгоритма A переменная  $g(X)$  обозначает:

длина текущего пути от начальной вершины до X  
длина кратчайшего пути из A в X  
эвристическая функция  
длина кратчайшего пути из X в Z

89 В каком случае эвристическая функция  $h(x)$  может быть монотонной?

для любых вершин  $u, v$

если для любых вершин  $u, v$  лежащих на одном пути,  $|h(u) - h(v)|$  меньше либо равен стоимости пути от  $u$  до  $v$

если для любых вершин  $u, v$  лежащих на одном пути,  $|h(u) - h(v)|$  больше стоимости пути от  $u$  до  $v$

90 Укажите оператор вычисления значения выражения в Prolog.

vector\_is  
vis  
is

91 Укажите весовую функцию выбора направления поиска для алгоритма A :

$$f(X) = g(X)$$

$$f(X) = g(X) + h(X)$$

$$f^*(X) = g^*(X) + h^*(X)$$

92 Какой встроенный предикат Prolog преобразовывает структурные термы в списки?

==  
=..  
:-

93 Что устанавливает алгоритм поиска решения IDA\* при каждом итерационном заглублиении?

диапазон возможных длин пути  
целочисленное значение длины пути  
вещественное значение длины пути

94 Что рассматривает алгоритм градиентного спуска?

все возможные направления движения  
только одно оптимальное направление движения  
несколько наиболее оптимальных направлений движения

95 Укажите, используемое в лекциях, название расширения для векторов оператора вычисления выражений в Prolog:

is  
vector\_is  
vis

96 Укажите определение языка над некоторым алфавитом  $A$  :  
некоторое множество цепочек  $L \subseteq A^*$  в этом алфавите  
это кортеж, включающий в себя выделенный начальный символ  
два непересекающихся множества нетерминальных  $N$  и терминальных  $A$  символов

97 Кортеж, включающий в себя множество  $R$  правил вида  $a \rightarrow b$ , где  $a, b \in (N \cup A)^*$  :  
алфавит  
грамматика  
язык

98 Укажите верное утверждение:  
грамматика  $G = (N, A, R, S)$  — это некоторое множество цепочек  $L \subseteq A^*$  в алфавите  
три пересекающихся множества нетерминальных  $N$  и терминальных  $A$  символов  
грамматика  $G = (N, A, R, S)$  — это кортеж, включающий в себя выделенный начальный символ  
 $s \in N$

99 Какой из перечисленных ниже языков программирования описывает формальную аксиоматическую систему со свойствами инкапсуляции, наследования и полиморфизма на основе расширения логики первого порядка?

prolog  
f-logic  
description logics

100 Семейство языков представления знаний, позволяющих описывать понятия предметной области в формализованном виде:

html  
f-logic  
description logics

101 Для реализации на логических языках программирования наиболее хорошо подходит...

контекстно-свободная грамматика  
контекстно-зависимая грамматика  
регулярная грамматика

102 Дизъюнкция в Description logics обозначается:

$C \cap D$   
 $C \cup D$   
 $\neg C$

103 Как называется грамматика в классификации по Хомскому, если она задается правилами общего вида?

контекстно-свободная грамматика  
контекстно-зависимая грамматика  
рекурсивно-перечислимая грамматика

регулярная грамматика

104 Искусственный интеллект – это ...

раздел информатики, изучающий человеческие способы рассуждения и решения задач  
парадигма программирования, в которой основными концепциями являются понятия объектов и классов

парадигма программирования, основанная на автоматическом доказательстве теорем

105 Нейрокибернетический метод моделирования искусственного интеллекта - это ...

метод, основанный на имитации рассуждений и логики человека

метод, основанный на создании "самообучающихся" систем

метод, основанный на создании множества простых объектов со сложным взаимодействием

метод, основанный на создании биологической структуры человека и особенностей ее функционирования

106 Семиотический метод моделирования искусственного интеллекта - это ...

метод, основанный на создании множества простых объектов со сложным взаимодействием

метод, основанный на создании биологической структуры человека и особенностей ее функционирования

метод, основанный на создании "самообучающихся" систем

метод, основанный на имитации рассуждений и логики человека

107 Механизм прямого логического вывода в экспертных системах является:

управляемым фактами

управляемым целями

управляемым логикой

108 К какой категории представления знаний относятся семантические сети?

процедурное представление знаний

иерархическое представление знаний

логическое представление знаний

сетевое представление знаний

109 В классической продукционной экспертной системе рабочая память представляет собой:

множество пар "атрибут-значение", описывающих состояние решаемой задачи

множество правил "если-то"

поиск в пространстве состояний множества комбинаций рабочей памяти, переходы в котором задаются правилами

110 Как определяются значения логического типа при формальном построении языка функционального программирования?

$true = \lambda x.\lambda y.x; false = \lambda x.\lambda y.y$

$true = \lambda x.\lambda y.xy; false = \lambda x.\lambda y.yx$

$true = 1; false = 0$

как атомарные константы T и F

111 Какая алгоритмическая модель лежит в основе функционального программирования?

логика высших порядков

машина Тьюринга

логика предикатов 1-го порядка

$\lambda$ -исчисление

112 Какие ошибки содержатся в приведенном фрагменте кода?

```
1 let ProcessImageAsync () =
2   async { let inStream = File.OpenRead(sprintf "Image%d.tmp") }
3   let pixels = inStream.ReadAsync(numPixels)
4   let pixels' = TransformImage(pixels, i)
5   let outStream = File.OpenWrite(sprintf "Image%d.done" i)
6   do outStream.WriteAsync(pixels')
```

в строке 2 отсутствует восклицательный знак после `let`

в строке 3 отсутствует восклицательный знак после `let`

в строке 4 отсутствует восклицательный знак после `let`

в строке 5 отсутствует восклицательный знак после `let`

в строке 6 отсутствует восклицательный знак после `do`

113 Как реализуются повторяющиеся действия в функциональных языках?

с помощью конструкций циклов

с помощью рекурсии

с помощью оператора перехода

с помощью функциональной абстракции

114 Какой список будет порожден конструкцией `[1..2..10]`?

конструкция недопустима

`[1;3;5;7;9]`

`[1;2;3;4;5;6;7;8;9;10]`

`[2;4;6;8;10]`

115 Какой тип имеет функция `apply` в `Eval/Apply`-интерпретаторе?

`apply: expr > ?expr > ?expr`

`apply: env > ?expr > ?env > ?expr > ?expr`

`apply: expr > ?expr`

`apply: env > ?expr > ?expr > ?expr`

116 Сколько памяти выделяется в стеке при вычислении длины списка  $n$  с помощью хвостовой рекурсии?

$O(n^2)$

$O(\log n)$

$O(1)$

$O(n)$

117 Что означает запись `< @ fun x > ? x * 2 @ > ?`

создается дерево выражения, описывающего функцию удвоения

в данном случае запись эквивалентна `fun x > ? x * 2`

создается отложенное вычисление для вычисления функции  $x * 2$

118 В чем разница между конструкциями `fun` и `function` в F#?

`fun` служит для задания функциональной константы, `function` -- для описания функции

`fun` -- это более короткий способ записи `function`

`function` служит для задания функциональной константы, `fun` -- для описания функции

в `function` может использоваться сопоставление с образцом, в `fun` -- нет

119 Какой процесс происходит при нормальном порядке редукции?  
редуцируется самый правый из самых внутренних редексов  
редуцируется самый левый из самых внешних редексов  
редуцируется самый правый из самых внешних редексов  
редуцируется самый левый из самых внутренних редексов

120 Какие из преобразований надо применить, чтобы редуцировать выражение  $(\lambda x. \sin x) 0 \rightarrow 0$ ?  
альфа-редукцию  
бета-редукцию  
гамма-редукцию  
дельта-редукцию

121 Возможно ли в F# использовать ленивые вычисления?  
нет, F# -- энергичный язык  
да, используя конструкции Lazy/Force  
да, используя последовательности seq  
да, используя ключи компилятора

122 Какое множество значений у прямой суммы  $T_1 + T_2$ ?  
множество пар  $(x, y) : x \in T_1, y \in T_2$   
множество  $T_1 \cup T_2 \setminus (T_1 \cap T_2)$   
помеченное объединение  $\{(0, x) \mid x \in T_1\} \cup \{(1, y) \mid y \in T_2\}$   
объединение множеств  $T_1 \cup T_2$

123 Как можно свести нелинейно-рекурсивную функцию к хвостовой рекурсии?  
используя продолжения для разворачивания рекурсии в цепочку функций  
используя стек для хранения оставшихся заданий на обработку  
используя дерево для представления цепочки вычислений

124 Что означает термин "неразрешимость проблемы останова"?  
невозможность построения программы, которая будет останавливаться на определенных входных данных и заикливаться на других  
невозможность построения программы, которая по произвольному алгоритму и входным данным определит, остановится ли он или нет  
невозможность построения программы, которая по некоторой программе будет определять возможные случаи заикливания

125 Какие комбинаторы образуют наименьший базис?  
I, K, S  
S, K  
Y, I, K, S  
Y

126 В каком представлении эффективнее хранить разреженные матрицы?  
в виде списка списков  
в виде массива переменной длины  
в порядковом представлении  
в виде двумерного массива

127 Что помещается в стек возвратов D SECD-машины?  
окружение, в котором вычислялась вызываемая функция  
состояние стеков S, E и C до вызова функции  
номер инструкции, на который необходимо вернуться после вычисления функции  
состояние стека вычислений до вызова функции

128 Какой тип имеет функция `map`?

`('a->bool)->'a list->'a list`  
`'a->'a->'a list->'a list`  
`('a->'b)->'a list->'b list`  
`('a->'a)->'a list->'a list`

129 Какие языки программирования являются преимущественно функциональными?

C#  
Java  
C++  
F#  
Haskell  
FORTH  
Objective C  
OCaml

130 Как правильно описать дерево общего вида на F#?

`type 'T tree = Leaf of 'T | Node of 'T*'T`  
`type 'T tree = Leaf of 'T | Node of 'T*('T tree list)`  
`type 'T tree = Nil | Leaf of 'T | Node of 'T*'T*'T ('T tree)`  
`type 'T tree = Nil | Leaf of 'T | Node of 'T -> ('T tree list)`

131 Какой будет результат выполнения `let x :: y = [1; 2; 3; 4] ?`

`x = 1, y = 2`  
`x = [1; 2; 3], y = 4`  
Ошибка  
`x = 1, y = [2; 3; 4]`  
`x = [1; 2; 3; 4], y = []`

132 Какова сложность проверки вхождения элемента в список длины  $n$  ?

$O(n)$   
 $O(1)$   
 $O(\log n)$   
 $O(n^2)$

133 Какой порядок редукции соответствует передаче параметров по значению?

апликативный  
нормальный  
передача параметров не связана с порядком редукции  
нормальный с мемоизацией

134 Укажите правильное монадическое свойство.

`return x >>= f ∈ x f`  
`(return x >>= f) ∈ return (x >>= f)`  
`return (f x) >>= g ∈ f (g x)`  
`return x >>= f ∈ f x`