**Пример экзаменационного билета**

Южно-Уральский государственный университет (национальный исследовательский университет)

Институт естественных и точных наук

Кафедра прикладной математики и программирования

Дисциплина Функциональное и логическое программирование

Код и направление подготовки 01.03.02 Прикладная математика и информатика

Профиль Математическое и программное обеспечение вычислительных машин и систем

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

1. Где используется логическое программирование?

Какие особенности Prolog позволяют применять его в этих областях?

2. Назначение и примеры использования функции car

3. Написать предикат, который находит индекс первого вхождения минимального элемента в список:

?-minind([6,4,1,7],X).

X=3

Зав. кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ведущий дисциплину лектор (Экзаменатор)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Теоретические вопросы:*  
1. Управляющие конструкции в ФП: композиция вызовов, условное предложение.   
Что с переменными, циклами, присваиванием, последовательностью действий в ФП?  
  
2. Какие возможности ФП упрощают написание кода.  
  
3. Роль списков в ФП. S-выражения.   
Сколько базовых функций необходимо для обработки S-выражений?  
  
4. Правила определения новых функций для обработки S-выражений?  
Пример функции (длина списка).  
  
5. Дополнительные параметры при реализации аналога циклов в ФП.  
  
6. Блоки локальных определений в ФП: синтаксис, назначение.  
  
7. Что такое функции высших порядков? Синтаксис лямбда-выражения в HFL, Kotlin.  
  
8. Рекурсивные формы блоков локальных определений в ФП: синтаксис, назначение.  
  
9. Определение контекста в ФП. Замыкание для функций и лямбда-выражений.  
  
10. Способы вычислений в ФП: энергичный, ленивый.  
  
11. Карринг и частичное применение в ФП. Плюсы при написании программ.  
  
12. Конвейер в ФП. Плюсы при написании программ.  
  
13. Запоминание в ФП. Почему это возможно в ФП, плюсы при написании программ.  
  
14. Обозначение типа функции в ФП (по лекции или на примере Kotlin).  
  
15. Чистое лямда-исчисление (определение, синтаксис, назначение). Для чего нужен Y-комбинатор.  
  
16. Комбинаторная логика (назначение, синтаксис выражений, комбинаторы для одного из базисов).  
  
17. Где используется логическое программирование?   
Какие особенности Prolog позволяют применять его в этих областях?  
  
18. Атомы и переменные в Prolog. Анонимные переменные.  
  
19. Структуры в Prolog (синтаксис, назначение, функтор, арность).  
Списки (синтаксис, представление в виде структуры).  
  
20. Сопоставление в ЛП (явное и неявное, назначение).  
  
21. Правила определения предикатов для обработки списков в ЛП.  
Пример предиката (длина списка).  
  
22. Отсечение в Prolog (обозначение, что делает).  
  
23. Использование отсечения в ЛП для подтверждение правильности выбора правила. Красное и зеленое отсечение.  
  
24. Использование отсечения в ЛП для прекращения доказательства цели.  
Проблемы с отрицательными предикатами, рекомендации по их решению.  
  
25. Использование отсечения в ЛП для завершения последовательности порождения  
и проверки вариантов.  
  
26. Исчисление предикатов (назначение, используемые обозначения и термины).  
  
27. Принцип резолюции. Что такое СКНФ? Резольвента. Что означает пустая резольвента?  
  
28. Определения: верификация, спецификация, частичная правильность и  
  полнота, полная правильность программы, разрешимость.  
  
29. Определения: логический алгоритм, вычисление, производимое вычисление, критерий для верификации программы.  
  
30. Отличие знаний от данных.  
  
31. Представление знаний и вывод в продукционных системах.  
  
32. Представление знаний и вывод в семантических сетях.  
  
33. Представление знаний и вывод во фреймовых системах.  
  
34. Экспертные системы (определение, особенности). Где сейчас используются ЭС?  
  
35. Пространство состояний, эвристический поиск в пространстве состояний.

*Вопросы по функциям и предикатам:*  
1. Назначение и пример использования функции car  
  
2. Назначение и пример использования функции cdr  
  
3. Назначение и пример использования функции cons  
  
4. Назначение и пример использования функции eq  
  
5. Назначение и пример использования функции atom  
  
6. Назначение и пример использования операции "задержать" (delay)  
  
7. Назначение и пример использования операции "возобновить" (force)  
  
  
8. Назначение и пример использования предиката var  
  
9. Назначение и пример использования предиката arg  
  
10. Назначение и пример использования предиката number  
  
11. Назначение и пример использования предиката atom  
  
12. Назначение и пример использования предиката is  
  
13. Назначение и пример использования предикатов для арифметических сравнений (=:=, =\=, < и т.д.)  
  
14. Назначение и пример использования предиката listing  
  
15. Назначение и пример использования предиката write  
  
16. Назначение и пример использования предиката functor  
  
17. Назначение и пример использования предиката =..  
  
18. Назначение и пример использования предикатов bagof/setof  
  
19. Назначение и пример использования предиката findall  
  
20. Назначение и пример использования предиката ==  
  
21. Назначение и пример использования предиката atom\_codes  
  
22. Назначение и пример использования предиката retract  
  
23. Назначение и пример использования предикатов asserta/assertz  
  
24. Назначение и пример использования предиката clause  
  
25. Назначение и пример использования предиката member  
  
26. Назначение и пример использования предиката append  
  
27. Назначение и пример использования предиката reverse  
  
28. Назначение и пример использования предиката nl  
  
29. Назначение команды :-dynamic  
  
30. Назначение команды :-op

*Задачи*

1. Написать предикат, который находит индекс первого вхождения минимального элемента в список:

?-minind([6,4,1,7],X).

X=3

2. Определите предикат для печати арифметического выражения в обратной польской записи

?-печать(a+b\*c/2-3).

a b c \* 2 / + 3 -

3. Определите предикат, который заменяет все положительных значений на *X* в списке чисел.

?-replpol([1,2,-3,4],7,R).

R=[7,7,-3,7]

4. Определите предикат, который удаляет все положительных значения в списке чисел.

?-delpol([1,2,-3,4,0],R).

R=[-3,0]

5. Определите предикат, который выполняет слияние двух упорядоченных по возрастанию списков

?-merge([1,5,7],[2,3,6],R).

R=[1,2,3,5,6,7]

6. Напишите функцию возр(s), которая проверяет, что числовой список s является упорядоченным по возрастанию

7. Определите функцию слияние(s1,s2), которая выполняет слияние двух упорядоченных по возрастанию списков s1 и s2

8. Определите функцию найти(f,s), которая первый элемент списка s, для которого функция f возвращает #T

9. Определите функцию найти(f,s), которая первый индекс элемента списка s, для которого функция f возвращает #T

10. Определите функцию каждый(f,s), которая проверяет, что для всех элементов списка s функция f возвращает #T

11. Написать предикат, который находит индекс последнего вхождения терма в список:

?-lastind(2,[6,4,2,3,2,1],X).

X=5

12. Определить предикат arg1 (аналог arg) через предикаты =.. и functor

13. Написать функцию НОК(x,y) для нахождения наименьшего общего кратного двух целых чисел. Для нахождения НОК необходимо произведение чисел поделить на их НОД. НОД можно найти по алгоритму Эвклида.

14. Написать функцию isort(L) для сортировки числового списка методом вставок. Определить дополнительную функцию для вставки числа в отсортированный список

15. Написать функцию индпосл(L,X), которая находит индекс последнего вхождения элемента в список.

16. Написать функцию срзнач(L), которая находит среднее значение элементов числового списка.

17. Написать функцию часто(L), которая находит самый часто встречающийся элемент числового списка.

18. Написать функцию повтор(L), которая возвращает список элементов, встречающихся в списке L не менее двух раз. После обнаружения повторяющегося элемента не забудьте удалить из списка все его вхождения.

19. Написать предикат, который длину и индекс начала самого длинного подсписка из одинаковых элементов:

?-maxlen([1,7,3,1,1,4,4,4,8,7,4],N,L).

N=6, L=3 % [4,4,4] начиная с 6 элемента списка

20. Написать предикат, который удаляет из списка те элементы, которые встречаются в списке более одного раза:

?-delrep([7,7,3,1,4,8,7,4],R).

R=[3,1,8]

22. Написать предикат, который удаляет из числового списка те

элементы, которые не входят в заданный диапазон от A до B.

?-delab([7,3,1,4,8,4],2,5,R).

R=[3,4,4]

23. Написать предикат, который удаляет *K*-ый элемент из списка.

?-delk(3,[a,b,c,d],R).

R=[a,b,d]

24. Написать предикат, который выполняет быструю сортировку числового списка. Использовать вспомогательный предикат, который делит список на 3 части: в одну входят элементы, меньшие заданного, в другую - равные, в третью - большие. Предикат сортировки делит список на три части по первому элементу списка, рекурсивно сортирует 1 и 3 части и соединяет результаты с помощью append.

?-qsort([2,7,4,1],R).

R=[1,2,4,7]

25. Написать предикат, который применяет к каждому элементу списка некоторый предикат от двух аргументов (первый аргумент - вход - элемент исходного списка, второй аргумент - выход - элемент списка-результата).

inc(X,Y) :- Y is X+1.

?-make(inc,[1,2,5],R).

R=[2,3,6]

26. Написать предикат, который оставляет в списке те элементы, которые удовлетворяют заданному предикату:

odd(X) :- X mod 2 =:= 1.

?-filtr(odd,[2,3,7,4],R).

R=[3,7]

27. Написать предикат, который заменяет i-й аргумент в структуре на новый терм:

?-argrpl(t(a(1),z,1),2,b,R).

R=t(a(1),b,1).

28. Написать предикат, который находит разложение заданного целого числа>2 на простые множители:

?-decay(28,X). % результат X - это структура

X=2\*2\*7

29. Написать предикат, который удаляет i-е вхождение заданного терма из списка:

?-idel(2,a,[a,f,g,a,e,a],R).

R=[a,f,g,e,a]

30. Написать предикат, который исключает все вхождения заданного терма из списка:

?-delall(a,[a,f,g,a,e],R).

R=[f,g,e]

31. Написать предикат, который находит высоту дерева,

описываемой структурой вида t(ЛеваяВетвь, ПраваяВетвь).

?-height(t(t(1,t(a,c)),t(b,f)),H).

H=3

32. Определите предикат входит(A1,A2), проверяющий, что все буквы атома A1 входят в набор букв атома A2.

?-входит(автор,авиатор).

yes

33. Определите функцию слияние(a,b), объединяющую два упорядоченных по возрастанию числовых списка a и b в новый упорядоченный список.

34. Определите функцию подстрока(a,b), определяющую, содержится ли последовательность a (в указанном порядке, целиком и непрерывно) в последовательности b.

35. Написать предикат replace(List,K,Val,NewList), который заменяет

K-ый элемент списка на новое значение Val:

?-replace([a,b,c,d],2,z,R).

R=[a,z,c,d]