

Факультет компьютерных технологий и прикладной математики  
Кафедра вычислительных технологий  
02.03.02

Функциональное и логическое программирование  
Список вопросов и типовых задач на экзамен

Лямбда-исчисление

1. Опишите основные принципы функционального программирования и вытекающие из них преимущества и недостатки. Опишите понятия высшая функция, чистая функция, каррирование и ленивые вычисления
2. Опишите математические предположения, которые привели к лямбда исчислению и объясните формат записи лямбда выражений. Дайте определение лямбда терма. Опишите соглашения о возможности опускать скобки, принятые в лямбда выражении.
3. Дайте определения редукции лямбда термов. Опишите стратегии редукции лямбда термов.
4. Дайте определения свободных и связанных переменных в лямбда термах.
5. Дайте понятия подстановки и преобразования. Сформулируйте понятия эквивалентности.
6. Сформулируйте теорему Черча-Россера и докажите два следствия из нее.
7. Сформулируйте и докажите лемму о комбинаторах I, K, S
8. Докажите, что любой терм представим в виде комбинаторов S K
9. Числа Черча. Операция плюс 1. Операция (-1).
10. Кorteжи и Каррирование в лямбда исчислении. Объясните, в чем суть каррирования.
11. Комбинатор неподвижной точки. Приведите два примера, покажите, что они комбинаторы неподвижной точки.
12. Арифметические операции над числами Черча + \* ^
13. Let выражения
14. Булевы константы и оператор if. Реализация булевых операций.
15. Рекурсивные функции (на примере любой функции).
16. Полнота лямбда исчисления по Тьюрингу
17. Реализация списков Черча, head, tail.
18. Покажите набор функций преобразования списков Чёрча.

Типовые формулировки задач

№ 1. Дан терм, представленный в виде аппликации и абстракции лямбда-термов. Определить свободные и связанные переменные, провести редукцию с помощью двух возможных стратегий.

№ 2. Дан терм, представленный в виде аппликации комбинаторов, провести редукцию двумя способами.

№ 3. Дан терм, проверить, является ли он комбинатором неподвижной точки.

№ 4. Составить комбинаторы, позволяющие реализовать указанные функции (числовые или алгебры логики)

№ 5. Провести редукцию арифметических и логических выражений.

## Функциональное программирование на примере языка котлин.

1. Способы написания функций в котлине, разные виды тела функции. Функциональные аргументы и значения. Множество аргументов. Один аргумент.
2. Структура описания класса в котлине. Переменные и значения. Разница и применение в функциональном подходе. Геттеры, сеттеры, свойства. Дата классы.
3. Перечисления и конструкция when. Операторы проверки класса и преобразования к классу.
4. Исключения в Котлин. Структура исключений.
5. Работа с массивами в котлин. Работа со списками в котлин.
6. Работа с множествами в котлин. Работа с ассоциативными массивами в котлин
7. Общая структура классов коллекций в котлин.
8. Наследование в котлин. Делегация в котлин.
9. Понятие функции расширения в котлин. Понятие и особенности локальных функций в котлине. Функции расширения и локальные функции.
10. Регулярные выражения в котлин.
11. Интерфейсы и полиморфизм в котлин.
12. Общий обзор наследования, абстрактных классов, интерфейсов и их реализации в котлин. Набор модификаторов. Сравнение с java
13. Модификаторы доступа в котлин. Запечатанные классы.
14. Способы задания конструкторов в котлине.
15. Структура классов в котлин. Сравнение объектов в котлин.
16. Совместное объявление класса и экземпляра в котлин. Задание фабричных методов в котлин.
17. Лямбда выражения. Передача лямбда выражений. Замыкание. Краткие способы записи лямбда выражений.
18. Методы работы с коллекциями, принимающие функции, как аргумент. Filter, map. Count, sort, max
19. Методы работы с коллекциями, принимающие функции, как аргумент. GroupBy, reduce, fold.
20. Работа с последовательностями в котлин, в java
21. Лямбда выражения с получателями.
22. Поддержка значения null
23. Базовые типы в котлин.
24. Перегрузка арифметических операторов.

### Типовые формулировки задач

Дана таблица с результатами обучения Вашей группы в семестре. Необходимо прочитать все данные о посещаемости и все оценки за КР, ЛР и ЭП. Создать структуру классов для хранения информации.

1. Прочитать суммарные баллы и оценки. Найти лабораторную работу, которую выполнили наибольшее количество человек на любую оценку среди не получивших допуск. Найти КР, которую выполнили наименьшее количество человек хотя бы на допуск среди не получивших допуск.

2. Рассчитать самостоятельно в программе баллы за посещаемость, КР, ЛР и ЭП так же, как в документе. Найти сумму баллов и вывести количество баллов за посещаемость, ЛР, ЭП, КР, ИТОГО для каждого студента.

3. Сформировать рейтинг так же, как и в разделе рейтинг программно. Вывести 5 худших среди получивших допуск, сгруппировав их по группам. Вывести 5 лучших среди не

получивших допуск, сгруппировав их по группам в том же формате, как и в разделе рейтинг, то есть выводить проценты сделанного.

4. Сформировать программно раздел группы так же, как сформировано в таблице и вывести на экран.

5. Найти группу, в которой наименьший средний балл на допуск среди тех, кто получил оценки 3, 4, 5.

#### Логическое программирование на примере языка Swi-Prolog

1. Опишите понятие и структуру фактов в языке Пролог. Раскройте основные возможные типы, опишите понятие атом. Расскажите принцип работы терминальной машины Swi-Prolog, объясните каким образом происходит обработка вопросов.

2. Опишите смысл термина унификация, приведите показательные примеры. Объясните, как задаются предикаты, что такое правила и каким образом происходит работа с ними.

3. На примере числовых алгоритмов объясните смысл рекурсии вверх и рекурсии вниз в прологе.

4. Раскройте на примерах понятие backtracking, оператор отсечения и смысл его применения.

5. Опишите принцип работы со списками Черча в Swi-prolog. Покажите реализации встроенных предикатов работы со списками на основе механизма унификации(append, reverse nth0).

6. Объясните принцип работы со строками. Покажите на примерах основные встроенные предикаты работы со строками.

7. Покажите, каким образом происходит построение стандартных комбинаторных объектов средствами Swi-prolog.

8. Покажите основные принципы реализации переборных алгоритмов на графах средствами пролога.

9. Раскройте понятия статические и динамические факты. Поясните на примерах принцип работы с динамическими фактами.

10. Объясните принцип работы предикатов var, nonvar, atom, atomic, name, functor, arg, repeat.

#### Типовые формулировки задач

1. Построить предикат, позволяющий получить подмножество исходного множества. На его основе решить задачу компоновки рюкзака минимального веса. Дан рюкзак с заданным объемом V. Дан набор объектов [[3,4], [5,6], [7,8], [4,5]], где первое число объем объекта, второе число вес объекта. Сначала решить задачу нахождения подмножества объектов, сумма объемов которых равна объему рюкзака. Далее найти такое подмножество объектов, сумма объемов которых равна объему рюкзака, а сумма весов объекта минимальна возможна.

2. Построить предикат, позволяющий получить сочетание по k элементов из исходного списка. Далее построить предикат, позволяющий построить размещение без повторений по m элементов. На основе построенных предикатов построить все слова длины 10, в которых 3 буквы а, еще одна буква встречается 3 раза, остальные буквы не повторяются. В алфавите 6 букв [abcdef]. Рассчитать количество таких слов, проверить, совпадает ли количество слов в итоговом файле с расчетным.

3. Написать предикат, который по заданному графу возвращает произвольное максимальное паросочетание. Модифицировать предикат так, чтобы он выявлял наибольшее паросочетание.

4. Дан файл, вывести в отдельный файл строки, состоящие из слов, не повторяющихся в исходном файле.

5. Номер 3797 обладает интересным свойством. Будучи простым, можно непрерывно удалять цифры слева направо и оставаться простыми на каждом этапе: 3797, 797, 97 и 7. Аналогично мы можем работать справа налево: 3797, 379, 37 и 3.

Найдите сумму простых чисел, меньших 1000000 которые можно обрезать слева направо и справа налево.

ПРИМЕЧАНИЕ. 2, 3, 5 и 7 не считаются усеченными простыми числами.