

**Министерство образования и науки Российской Федерации**  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра автоматизации обработки информации (АОИ)

## **ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ И ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ**

Методические указания к практическим занятиям и организации  
самостоятельной работы  
для студентов направления 38.03.05  
«Бизнес-информатика»  
(уровень бакалавриата)

2018

**Салмина Нина Юрьевна**

Функциональное программирование и интеллектуальные системы: Методические указания к практическим занятиям и организации самостоятельной работы для студентов направления «Бизнес-информатика» (уровень бакалавриата) / Н.Ю.Салмина. – Томск, 2018. – 29 с.

## Оглавление

1 Введение.....	4
2 Методические указания к проведению практических занятий.....	5
2.1 Практическое занятие «Основы языка Лисп» .....	5
2.2 Практическое занятие «Работа с функциями».....	9
2.3 Практическое занятие «Рекурсивные функции».....	12
2.4 Практическое занятие «Разработка функциональных программ» .....	17
2.5 Практическое занятие «Функционалы».....	21
2.6 Практическое занятие «Циклы и блочные функции» .....	23
2.7 Практическое занятие «Создание БЗ» .....	24
3 Методические указания для организации самостоятельной работы .....	27
3.1 Общие положения .....	27
3.2 Проработка лекционного материала .....	27
3.3 Самостоятельное изучение тем теоретической части курса .....	28
3.3.1 Разрушающие функции .....	28
3.3.2 Графы и Деревья .....	29
4 Рекомендуемая литература .....	29

# 1 Введение

Цель дисциплины – ознакомление студентов с основами функционального программирования на примере языка Лисп, принципами структурирования знаний в виде фреймов и семантических сетей, а также формирование у студентов профессиональных знаний и практических навыков по разработке и созданию интеллектуальных систем с помощью языка функционального программирования Лисп.

Изучение дисциплины включает в себя: теоретический раздел (изучение теоретического материала); практический раздел (выполнение практических занятий и контрольных работ); итоговый контроль результата изучения дисциплины. Данное пособие содержит в себе методические указания и варианты заданий для практических занятий и самостоятельной работы.

## 2 Методические указания к проведению практических занятий

### 2.1 Практическое занятие «Основы языка Лисп»

#### Цель занятия

Целью данного практического занятия является знакомство со средой работы в LispIDE и получение первичных навыков работы в этой среде путем написания простейших выражений с использованием базовых функций Лисп.

#### Рекомендации по подготовке к занятию

Для выполнения данного практического занятия и последующих работ прежде всего необходимо ознакомиться с работой в среде LispIDE.

LispIDE является программой, предназначенной для работы с различными реализациями языка Лисп, не имеющими собственных оболочек. При первоначальном запуске программы необходимо произвести следующие действия: зайдите в пункт меню Settings/SetLispPath и выберите ту версию языка Лисп, в которой вы будете работать. Для выполнения практических заданий рекомендуется использовать язык CLisp версия 2.49 (файл clisp-2.49-win32-mingw-big.exe). После установки требуемой версии в нижнем (диалоговом) окне появится надпись, сообщающая о том, что требуемая программа загружена, и вы можете работать.

Нижнее окно предназначено для работы в диалоговом режиме. Этот режим удобен для проверки и выполнения простейших вычислимых выражений, а также для запуска тех функций, которые были уже откомпилированы.

Верхнее окно программы предназначено для работы с файлами: здесь можно открыть существующий файл с программами, создать новый файл, редактировать и сохранять программу. Откомпилировать выбранные функции можно с помощью кнопки «(»)» или команды Edit/Send to Lisp. После компиляции созданные функции можно вызывать в диалоговом окне.

**Внимание!!! Данная среда не понимает пути, в которых используется кириллица!** Если вы создаете новый файл, то ВСЕ промежуточные каталоги должны иметь латинские имена. Если вы хотите открыть существующий файл, то проследите, чтобы он находился в каталогах (на всем пути) с латинскими именами!

На данном практическом занятии вам необходимо выполнить три задания. При выполнении первого задания используйте суперпозиции функций CAR и CDR. Помните, что функция CAR определяет голову списка, а функция CDR выделяет хвост списка.

Внимательно следите за балансом скобок!

Для выполнения второго задания используйте арифметические операции и суперпозиции CAR и CDR.

При выполнении третьего задания используйте функцию проверки условий COND, а также, по необходимости, другие основные функции Лисп.

### Содержание занятия

На данном практическом занятии необходимо выполнить три задания. Каждый студент выполняет задание индивидуально согласно своему варианту.

**Задание 1.** Записать выражение, выбирающее выделенный элемент списка.

**Задание 2.** Записать выражение, выполняющее заданную арифметическую операцию над всеми элементами заданного числового списка (на всех уровнях).

**Задание 3.** Записать условие, которое проверяет список и выдает истину, если выполняется хотя бы одно из заданных условий. Если ни одно из заданных условий не выполняется, выдать NIL.

### Варианты заданий

#### Задание 1

№ варианта	Список
1	(1 2 (3 4) 5 6 7)
2	(((q w e) g f) n)
3	(a (x c) (s v) n q)
4	((1 2 3) (4 5 6))
5	(9 8 (7 6 5) e w)
6	((d) f (h I k) z)
7	(3 (4 5) (j l))
8	(h z ((q) d k l) a)
9	((1) (2) (3) (4) (5))
10	(h ((a s d) l k)
11	(0 (2 g a) 1 5)
12	((b) v (1 2 3) z a)

13	(e ((1) (2)) a s)
14	(r f <b>b</b> c n z)
15	((1 2 3) s (4 5) f)
16	((d) <b>f</b> (h I k) z)
17	(q w ( <b>d</b> (s)) 3 f)

### Задание 2

№ варианта	Арифм. Операция	Список
1	+	(12 34 (5) 6)
2	-	((8) (1) 2)
3	+	((6) 8) 32)
4	*	(3 (12 67) 2)
5	/	(100 (3) 4)
6	*	(4 3 (7))
7	*	((2 8) (4 5))
8	-	((90) 4 (23))
9	+	(2 (1) (5) 9)
10	/	((((30)) 2)
11	+	(10 (45 1))
12	*	(2 (3 4) 6 10)
13	-	(8 4 5 3)
14	/	(23 3 (4))
15	*	((12) (3) (5))
16	/	(120 (2 3) 4)
17	+	(32 (45) (123))

### Задание 3

№ варианта	Условия проверки
1	а) список содержит всего два элемента; или б) список содержит три элемента.
2	а) первый и третий элемента списка одинаковы; или б) второй и третий элементы списка одинаковы
3	Хотя бы один из первых трех элементов является списком
4	Первый элемент списка является атомом или числом
5	Второй или третий элемент списка является числом
6	Первый элемент списка равен второму или Второй элемент списка равен третьему

7	второй или третий элементы списка являются списками
8	Первый элемент списка является числом или Третий элемент списка является списком
9	а) первый элемент списка равен 2; или б) второй элемент списка равен 3; или в) 3-й элемент списка = 4
10	Первый или третий элементы списка являются атомами
11	Список содержит ровно четыре элемента или Список пуст
12	Второй или третий элементы списка являются списками
13	Один из первых трех элементов списка является числом
14	Один из первых трех элементов списка является списком
15	а) первый и третий элементы равны; или б) первый и четвертый элементы равны; или в) второй элемент - список
16	Первые три элемента списка равны между собой
17	Первый элемент списка равен 0 или второй элемент списка равен 1 или третий элемент – равен 2
18	Первый и второй элементы списка – числа или второй элемент – список
19	Первый элемент списка атом или второй элемент списка – число
20	Список имеет больше трех элементов или список пуст
21	Первые два элемента списка являются числами ИЛИ в списке больше двух элементов
22	Первый или третий элемент списка является пустым списком
23	Второй элемент списка равен 2 и третий элемент списка является списком

## 2.2. Практическое занятие «Работа с функциями».

### Цель занятия

Целью данного практического занятия является знакомство с функциями и лямбда-выражениями в языке Лисп и получение первичных навыков написания собственных функций.

### Рекомендации по подготовке к занятию

Все задания можно выполнять в диалоговом режиме либо сначала записывать создаваемые функции в файл.

В случае записи функций в файл, их необходимо компилировать, и только после этого запускать выражения на выполнения.

Вызов созданных функций после компиляции можно осуществлять в диалоговом режиме.

В процессе выполнения практики необходимо написать две функции, описанные в вашем варианте задания.

При выполнении второго пункта задания используйте функцию, описанную в первом пункте задания.

При выполнении обоих пунктов задания можно использовать, по необходимости, встроенные функции Лисп.

### Содержание занятия

- а) записать функцию, определенную вашим вариантом в пункте а).
- б) написать программу с использованием функции созданную вами на первом этапе занятия.

### Варианты заданий

1. а) Напишите функцию, осуществляющую циклическую перестановку элементов в списке: первый элемент списка функция переставляет в конец списка, т.е.  $(a\ s\ d\ f) \Rightarrow (s\ d\ f\ a)$   
б) написать функцию, которая сравнивает первые два элемента списка. Если эти элементы равны, возвращается исходный список, если разные, то первый элемент списка помещается в конец списка.
2. а) Определите функцию  $(f\ a\ b\ c)$ , которая равна истине тогда и только тогда, когда из отрезков  $a$ ,  $b$  и  $c$  можно построить треугольник.  
б) если из отрезков  $a, b$  и  $c$  можно построить треугольник, функция должна проверять, является ли треугольник прямоугольным и

- возвращать T или NIL; если треугольник построить нельзя, функция должна возвращать число, равное сумме трех чисел: a, b и c.
3. а) написать функцию, которая возвращает t, если первый и последний элементы списка равны, иначе - nil  
 б) Определите функцию, которая меняет местами первый и последний элементы списка, оставляя остальные на своих местах, если элементы списка разные, иначе функция возвращает список без первого и последнего элементов.
  4. а) Напишите функцию трех аргументов (list3 x y z), которая бы формировала список из аргументов: (x y z). Функцию list не использовать.  
 б) написать функцию трех аргументов. Если все три аргумента атомы – создать новый список из этих аргументов. Если какой-то аргумент является списком, то в новый список включать только первый элемент списка:  
 $1\ 2\ 3 \Rightarrow (1\ 2\ 3)$   
 $1\ (a\ s)\ (d\ f\ c) \Rightarrow (1\ a\ d)$
  5. а) Напишите функцию, единственный аргумент которой является списком. Функция возвращает t только в том случае, если первые два элемента этого списка являются списками. В остальных случаях функция должна выдавать nil.  
 б) написать функцию с одним аргументом – списком. Если функция из п. а) возвращает t, создать список из четырех элементов: 1 и 2 элемент 1-го списка + 1 и 2 элементы 2-го списка. Если функция возвращает nil, то убрать из списка первые два элемента.
  6. а) Написать функцию, на вход которой подается три целых числа – сторона квадрата, длина и ширина прямоугольника. Функция должна определять, сколько квадратов с заданной стороной поместится в этот прямоугольник (например, при стороне=2, ширине=3 и длине=5, можно поместить 2 квадрата).  
 б) написать функцию, которая вычисляла бы общую площадь квадратов размером c\*c, вписывающихся в прямоугольник размерами a\*b.
  7. а) написать функцию, которая выдает истину, если первые три элемента списка являются числами, и ложь в противном случае.  
 б) создать функцию, на вход которой подается список Y и число N. Если первые три элемента списка Y являются числами, то функция возвращает список этих чисел, увеличенных в N раз. Если нет, то функция возвращает просто число N.

8. а) Напишите функцию двух аргументов ( $f\ q\ w$ ), где аргументы являются списками. Функция возвращает  $t$  только в том случае, если первые два элемента этих списков соответственно равны друг другу. В остальных случаях функция должна выдавать  $nil$ .  
 б) если предыдущая функция возвращает  $t$ , создать список из двух предыдущих путем слияния, исключив из них первые два элемента. Если  $nil$  – список из четырех элементов: 1 и 2 элемент 1-го списка + 1 и 2 элементы 2-го списка.
9. а) написать функцию от трех аргументов – списков. Функция должна возвращать 3, если в каждом списке не менее 3-х элементов, 2 – если в каждом списке не менее 2-х элементов и 1 – если в каждом списке есть по крайней мере один элемент. Если среди исходных списков есть пустой – функция возвращает  $NIL$ .  
 б) если предыдущая функция возвращает  $i$  – создать список из  $i$ -х элементов исходных списков. Если  $NIL$  – вернуть пустой список.
10. а) написать функцию с одним аргументом – числовым списком, длиной не менее трех. Функция должна возвращать  $T$ , если первые три элемента списка – четные и  $NIL$  в противном случае.  
 б) если предыдущая функция возвращает  $T$ , вернуть исходный список, уменьшив его первые три элемента вдвое, иначе вернуть исходный список без первых трех элементов.
11. а) написать функцию с одним аргументом – списком. Функция должна возвращать 2, если первые два элемента списка – списки, 1 – если только первый элемент является списком и 0 – в противном случае.  
 б) создать функцию, которая бы изменяла исходный список, поднимая первые элементы первых двух подсписков на верхний уровень, например:  
 $((a\ s)\ 2\ 3\ 4) \implies (a\ (s)\ 2\ 3\ 4)$   
 $((q\ w\ e)\ (1\ 2\ 3)\ 4\ 5\ (a)) \implies (q\ (w\ e)\ 1\ (2\ 3)\ 4\ 5\ (a))$   
 $(1\ 2\ (a\ 3)) \implies (1\ 2\ (a\ 3))$
12. а) написать функцию, которая меняла бы первые два элемента местами.  
 б) определить функцию, которая меняет местами элементы в исходном списке: 1-й и 2-й, 3-й и 4-й, 5-й и 6-й. Если количество элементов меньше 6, то менять местами 1-2, 3-4 (или только 1-2, если количество элементов меньше 4-х). Для перестановки элементов использовать функцию из пункта а).
13. а) написать функцию с одним аргументом - списком, которая бы меняла местами последние два элемента списка.

- б) определить функцию, на вход которой подается список и число. Если длина списка меньше заданного числа, то число добавляется в список в качестве первого элемента. Если больше, то функция меняет местами два последних элемента исходного списка.
14. а) определить функцию с двумя аргументами  $X$  и  $Y$ , где  $Y$  – список. Функция должна вставлять  $X$  в конец списка  $Y$  (не использовать функцию `append`).
- б) написать функцию, имеющую два аргумента:  $X$  –  $S$ -выражение и  $Y$  – список. Если  $X$  равен первому элементу  $Y$ , то он добавляется в конец списка  $Y$ , если нет – в начало.
15. а) функция имеет три аргумента – действительные числа, которые являются параметрами квадратного уравнения. Функция должна возвращать `nil`, если уравнение не имеет корней и значение дискриминанта, если решение существует. Функция `SQRT` извлекает квадратный корень из аргумента.
- б) определить функцию, которая решает квадратное уравнение, выдавая в качестве результата список из полученных корней.
16. а) написать функцию с двумя аргументами – числами. Функция должна увеличивать первый аргумент в три раза и уменьшать второй аргумент в два раза.
- б) Определить функцию, имеющую один аргумент – числовой список. Функция должна изменять исходный список следующим образом: первый элемент списка увеличивается в три раза, второй элемент уменьшается в два раза, остальные элементы остаются неизменными.

## 2.3 Практическое занятие «Рекурсивные функции»

### Цель занятия

Целью выполнения данного занятия является знакомство с рекурсией и принципами построения рекурсивных функций.

### Рекомендации по подготовке к занятию

В процессе выполнения практики необходимо написать две функции, описанные в вашем варианте задания. При создании функций используйте рекурсию. Для этого вам необходимо сначала определить терминальную ветвь (при необходимости терминальных ветвей может быть несколько), которая определяет правило останова. Правило останова может определять

две ситуации: если мы выполнили все возможные действия и не достигли нужного результата; если мы достигли нужного результата.

На втором этапе спланируйте рекурсивную ветвь (ветви), которая определяет рекурсивный вызов функции с упрощением аргументов. При каждом рекурсивном обращении мы должны приближаться к условию останова!

Формирование результата может быть осуществлено двумя способами: в процессе рекурсивных вызовов (при спуске «вниз»), или в процессе подъема «наверх». Если рекурсия вызывает у вас затруднения, то, как правило, проще формировать результат при спуске «вниз». Для этого введите вспомогательные параметры, в которых и будет формироваться результат. При достижении правила останова сформированный результат и будет результатом работы функции.

***Внимание! Циклы и функцию присвоения не использовать!***

### **Задание на практику**

Написать функции, заданные в вашем варианте с обязательным использованием рекурсии.

### **Варианты заданий**

1. а) напишите функцию, которая проверяет, являются ли все элементы списка положительными числами. Функция возвращает NIL, если хотя бы один элемент списка НЕ является числом, или если в списке есть хотя бы один отрицательный элемент.  
б) определите функцию, аргументом которой является числовой список  $z$ , и которая формирует новый список, удаляя из  $z$  на всех уровнях положительные элементы ( $z$  может быть многоуровневым).
2. а) определите функцию умножения двух целых чисел через сложение и вычитание.  
б) напишите функцию, которая увеличивает все элементы числового списка в два раза (список может быть многоуровневым).
3. а) определите функцию от двух аргументов  $x$  и  $n$ , которая создает список из  $n$  раз повторенных элементов  $x$ .

б) напишите функцию, аналогичную встроенной функции subst замены в списке s выражения x на y, но производящую взаимную замену y на x:  $x \rightarrow y, y \rightarrow x$ .

Например:  $(f\ 2\ a\ (s\ d\ 2\ f\ a\ 4\ 2)) \implies (s\ d\ a\ f\ 2\ 4\ a)$

4. а) Определите функцию (f n) (n кратное 3), вычисляющую сумму:  
 $1*2*3+4*5*6+\dots+(n-2)*(n-1)*n$   
б) определите функцию, на вход которой подается список (он может быть многоуровневым). Функция должна возвращать T, если все атомы списка (на всех уровнях) являются числами, и NIL – в противном случае.
5. а) последовательность чисел Фибоначчи 1,1,2,3,5,8,13... строится по следующему закону: первые два числа – единицы; любое следующее число есть сумма двух предыдущих. Напишите функцию (f n f1 f2) с накапливающимися параметрами f1 и f2, которая вычисляет n-е число Фибоначчи.  
б) напишите функцию, которая считает полное количество атомов (на всех уровнях и не равных NIL) во входном списке.
6. а) определите функцию, аргументом которой является список с четным количеством элементов – атомов. Функция должна менять местами каждую пару элементов списка (1-й элемент со 2-м, 3-й с 4-м и т.д.). Если количество элементов списка является нечетным, функция должна возвращать NIL.  
б) Определите функцию, на вход которой подаются два списка – множества. Функция должна выдавать объединение этих множеств
7. а) напишите функцию, имеющую два аргумента: числовой список и целое число N. Функция должна возвращать список, в котором элементы, стоящие на четных позициях, увеличены в N раз.  
б) Определите функцию, на вход которой подаются два списка – множества. Функция должна искать пересечение этих множеств
8. а) напишите функцию, аргументом которой является целое число, которая бы вычисляла факториал этого числа.  
б) напишите функцию, аргументом которой является список с любыми s-выражениями. Функция должна возвращать линейный список из всех атомов исходного.

Например, список  $(q (s d) (f (x c)) b)$  должен преобразовываться функцией в список  $(q s d f x c b)$ .

9. а) Написать функцию, на вход которой подается список из двух списков чисел. Результатом работы функции должен стать список, элементами которого являются поэлементные суммы этих двух подсписков. Например:  $((1 3 2)(4 6 3)) \implies (5 9 5)$   
б) Определить функцию, которая определяет, является ли данное натуральное число простым
10. а) Написать функцию, на вход которой подается список атомов. Функция должна формировать список из двух подсписков следующим образом: в 1-й подсписок включаются все элементы исходного списка, стоящие на нечетных позициях, во 2-й – на четных. Например:  $(f y v a p r) \implies ((f v p)(y a r))$   
б) Определите функцию  $(f s)$ , вычисляющую знакопеременную сумму  $a_1 - a_2 + a_3 - a_4 + \dots + a_k \cdot (-1)^{(k+1)}$  для списка  $s$ , имеющего вид  $(a_1 a_2 a_3 \dots a_k)$ .
11. а) Определите функцию, аргументом которой является одноуровневый список, элементами которого могут быть как числа, так и символьные атомы. Функция должна формировать новый список, состоящий из двух подсписков: в 1-й подсписок включаются числа, во 2-й – все символьные атомы. Например:  $(v a 1 3 y 5 f y) \implies ((1 3 5)(v a y f y))$   
б) Напишите функцию, аргументом которой является список, удаляющую из этого списка все не числовые атомы на всех уровнях (список может быть многоуровневым).
12. а) Определите функцию, аргументом которой является многоуровневый список, атомы которого могут быть как числа, так и символьные атомы. Функция должна менять все числа на атом NUM. Например:  $(a s 4 (s 1 2) c) \implies (a s num (s num num) c)$   
б) напишите функцию, определяющую глубину первого вхождения элемента  $u$  в список  $w$ .
13. а) определите функцию, аргументом которой является целое число  $n$ , результатом которой является сумма цифр натурального числа  $n$ . (встроенная функция MOD дает остаток от деления).

- б) Определите функцию, аргументом которой является одноуровневый список. Функция должна менять все отрицательные числа на атом MINUS, положительные – на атом PLUS, нули – на атом ZERO и символьные атомы – на атом SMV. Например: (a -4 s 1 0 c) ==> (smv minus smv plus zero smv)
14. а) Напишите функцию, которая делает из списка множество, т.е. удаляет все повторяющиеся элементы.  
 б) Определите функцию (f s n), где s – список, n – число. Функция должна возвращать n-й атом из списка s. Список s может быть многоуровневым.
15. а) функция (f l) имеет единственный аргумент – список, состоящий из четного количества элементов – чисел. Каждая пара чисел в списке задает стороны прямоугольника. Функция должна возвращать список из периметров прямоугольников.  
 б) Напишите функцию, на вход которой подается многоуровневый список. Функция должна подсчитывать количество всех подсписков на всех уровнях.  
 Например: ((2 3) 4 ((3 (4 (2))) 1)) ==> 5
16. а) Определите функцию, реверсирующую список и все его подсписки на любом уровне. Например, (1 (2 3) 4) ==> (4 (3 2) 1).  
 б) Напишите функцию, зависящую от двух аргументов x и y, удаляющую все вхождения x в список y на всех уровнях. X может быть атомом или списком.
17. а) напишите функцию, на вход которой подается список и два числа:  $n < k$ . Функция должна формировать новый список, включая в него только те элементы исходного списка, которые являются числами. При этом выбираемые числа должны быть или меньше n, или больше k.  
 б) на вход функции подается список, элементами которого являются списки из двух атомов. Функция должна менять местами элементы в каждом из подсписков:  
 ((1 2) (ф ы) (3 в)) ==> ((2 1) (ы ф) (в 3)). Функцию reverse не использовать!
18. а) напишите функцию, которая бы подсчитывала количество пустых списков на всех уровнях в единственном аргументе функции – списке.

- б) На вход функции подается числовой список с четным количеством элементов. Определить функцию, которая бы проводила расчеты по следующей схеме:  $(x_1 x_2 x_3 x_4 \dots) \implies x_1 * x_2 + x_3 * x_4 + \dots$   
 Например:  $(1 2 3 4 5 6) \implies 44$
19. а) Определить функцию, которая изменяет входной список, переставляя местами 1-й и 2-й элементы, 3-й и 4-й элементы и т.д.  
 Например:  $(a s d f g h) \implies (s a f d h g)$   
 б) определить функцию, которая находит максимальную сумму двух соседних чисел в заданном списке. Единственный аргумент функции является одноуровневым числовым списком
20. а) Определить функцию, на вход которой подается два числа K, L и числовой список, на выходе формируется список, положительные элементы которого увеличиваются в K раз, а отрицательные элементы увеличиваются в L раз. Например:  
 $(2 3 (-1 2 3 -4 5)) \implies (-3 4 6 -12 10)$   
 б) Написать функцию, на вход которой подается натуральное положительное число. Функция должна возвращать список его делителей.
21. а) На вход функции подается одноуровневый список чисел. Необходимо найти номер первого максимального элемента списка.  
 б) написать функцию, удаляющую из исходного списка все числовые атомы. Исходный список может быть многоуровневым.

## 2.4 Практическое занятие «Разработка функциональных программ»

### Цель занятия

Целью проведения данного занятия является знакомство с принципами построения сложных рекурсивных функций, лямбда-выражений, модульной структуры программ.

### **Рекомендации по подготовке к занятию**

В процессе выполнения практики необходимо написать программу, выполняющую действия, описанные в вашем варианте задания. Количество необходимых функций, их входные и выходные параметры определить самостоятельно, исходя из задания.

При написании программы декомпозируйте задачу: модульная структура программы облегчает как написание, так и понимание рассматриваемых действий. Всегда легче писать более мелкие функции, из которых можно компоновать более сложные задачи.

***Внимание! Циклы и функцию присвоения не использовать!***

### **Задание на практику**

Требуется составить программы на Лиспе, в каждом задании используя вспомогательные функции.

### **Варианты заданий**

1. Определите функцию, зависящую от одного аргумента – многоуровневого списка, которая по данному списку формирует одноуровневый список его элементов, встречающихся в нем более одного раза.
2. Определите функцию  $(f\ a\ n)$ , которая от двух числовых аргументов  $a$  и  $n$  вычисляет величину  $a+a*(a+1)+a*(a+1)*(a+2)+\dots+a*(a+1)*(a+2)*\dots*(a+n)$ .
3. Напишите функцию, на вход которой подается атом  $Y$  и список  $W$ . Функция должна заменять  $Y$  на число, равное глубине вложения  $Y$  в  $W$ .  
Например,  $Y=a$ ,  $W=((a\ b)\ a\ (c\ (a\ (a\ d))))$  ->  $((2\ b)\ 1\ (c\ (3\ (4\ d))))$ .
4. Напишите функцию, которая сортирует список чисел по возрастанию. Список может быть многоуровневым. Тогда вес (значение) любого подсписка для сортировки определяется суммой его элементов. Любой подсписок внутри тоже должен быть отсортирован.  
Например:  $(9\ 2\ (1\ 2)\ 5\ (3\ 6\ 2)\ 12\ (6\ (3\ 1)))$  =>  
 $(2\ (1\ 2)\ 5\ 9\ ((1\ 3)\ 6)\ (2\ 3\ 6)\ 12)$ .

5. Написать программу , которая возвращает список ( $m_1 m_2 m_3$ ), состоящий из трех наибольших элементов исходного числового списка  $s$ :  $m_1 \geq m_2 \geq m_3$ . Исходный список содержит не менее трех элементов.
6. Написать программу, результатом работы которой является список, получающийся из списка списков  $s$  после удаления всех подсписков, содержащих числа.
7. Определите функцию ( $f s$ ), которая в многоуровневом списке  $s$  переставляет все отрицательные элементы в начало списка, положительные – в конец, и делает его одноуровневым. Например,  $(f (4 -2 (3 (-1 -9)) 8)) \rightarrow (-2 -1 -9 4 3 8)$
8. Определить функцию с одним аргументом – одноуровневым списком. Функция должна подсчитывать количество вхождений в список каждого атома и выдавать результат в виде списка списков:  $(a s 3 d s a 3) \Rightarrow ((a 2) (s 3) (3 2) (d 1))$
9. Написать программу, осуществляющую перевод десятичного числа в любую заданную систему счисления (двоичную, восьмеричную и т.п.)
10. Определить функцию с двумя аргументами  $A$  и  $B$  – числовыми списками. Функция должна проверять, выполняется ли следующее соотношение между элементами списков:  $b_i = a_i + 1$ ;  $a_{i+1} = 2 * b_i$
11. Определить функцию трех аргументов:  $a$ ,  $b$  и  $c$ . Первым аргументом является множество – список целых чисел,  $b$  и  $c$  – целые числа. Функция должна формировать список трех подмножеств данного множества, определяемых следующим образом:
  - a. Четные числа множества;
  - b. Числа множества, являющиеся квадратами числа;
  - c. Все числа  $b \leq a_i \leq c$ .
 Например:  $(f (1 2 3 4) 0 2) \Rightarrow ((2 4) (1 4) (1 2))$
12. Написать функцию, которая сортирует список чисел, используя алгоритм простой вставки.

13. На вход функции подается многоуровневый список чисел. Необходимо найти номер первого экстремального (максимального или минимального) элемента списка. Нумерация должна идти по атомам.
14. Напишите программу, формирующую список простых чисел на заданном интервале.
15. Задан числовой список. Написать программу, подсчитывающую среднее значение элементов списка, за исключением максимального и минимального элементов.
16. Написать программу, осуществляющую перевод числа, представленного в любой заданной системе счисления (двоичной, восьмеричной и т.п.), в десятичную систему счисления.
17. Написать функцию, которая сортирует список чисел методом пузырька.
18. Написать программу, определяющую дизъюнктивную сумму между двумя множествами:  $A+B=(A \cup B) - (A \cap B)$
19. Задан многоуровневый список чисел. Написать программу, возвращающую список из двух чисел – атомов исходного с максимальной разницей между ними.
20. Задан список вида  $((A_1 N_1)(A_2 N_2)...(A_K N_K))$ , где  $A_i$  – символьный атом,  $N_i$  – число. Написать программу, возвращающую список из двух символьных атомов исходного списка  $(A_i A_j)$  с максимальным произведением  $N_i * N_j$ .
21. Написать функцию, упорядочивающую список, заданный в качестве ее первого аргумента, переставляя его элементы в той последовательности, в какой они встречаются в списке, являющемся значением второго аргумента.
22. Определите функцию  $(f n)$ , которая для натурального числа  $n$  вычисляет  $(1+0)!+(2+1)!+(3+2)!+\dots+(n+(n-1))!$

## 2.5 Практическое занятие «Функционалы»

### Цель занятия

Целью проведения данного занятия является знакомство с функционалами и принципами их использования при написании новых функций.

### Рекомендации по подготовке к занятию

В процессе выполнения практики необходимо написать функцию, описанную в вашем варианте задания, с использованием функционалов. В случаях, когда обрабатывается многоуровневый список, может потребоваться использование косвенной рекурсии.

Отображающие функционалы работают с функциями, обрабатывающими единственный аргумент –  $s$ -выражение (список или атом). В случае, если вам необходимо использовать дополнительные параметры, опишите требуемую функцию как лямбда-выражение. Лямбда-выражение внутри функции видит все локальные переменные этой функции.

### Задание на практику

Напишите функцию с обязательным использованием функционалов. При необходимости используйте локальные или вспомогательные функции.

### Варианты заданий

1. Напишите функцию ( $\text{exist } p \ x$ ), которая проверяет «Существует ли элемент списка  $x$ , удовлетворяющий предикату  $p$ ?» ( $p$  – функция или функциональное имя).
2. Написать функцию, аргументом которой является числовой список. Функция должна возвращать список сумм: 1-го элемента, 2-х первых элементов, 3-х первых элементов и т.д.  
Например:  $(f \ (1 \ 2 \ 3 \ 4)) \Rightarrow (1 \ 3 \ 6 \ 10)$ .
3. Определите функцию ( $f \ s$ ), которая из списка чисел  $s$  создает новый список, меняя знак у каждого атома. Исходный список не предполагается одноуровневым.

4. Напишите функцию (`sum_list s`), аргументом которой является список арифметических выражений. Функция должна возвращать сумму вычисленных арифметических выражений. Например:  
`(sum_list '((+ 1 2) (* 3 4) (- 8 4))) => 19`
5. Напишите функцию (`all p x`), которая проверяет «Для всех ли элементов списка `x` выполняется предикат `p`?» (`p` – функция или функциональное имя).
6. Напишите функцию (`count p x`), которая подсчитывает, сколько атомов в списке `x` удовлетворяет предикату `p` (`p` – функция или функциональное имя). Список `x` не предполагается одноуровневым.
7. Определите функцию (`f s n`), которая из списка чисел `s` создает новый список, прибавляя к каждому атому число `n`. Исходный список не предполагается одноуровневым.
8. Напишите функцию (`filter p x`), которая создает список из элементов списка `x`, удовлетворяющим предикату `p` (`p` – функция или функциональное имя).
9. Напишите функцию, строящую список всех подмножеств данного множества.
10. Напишите функцию (`break p x`), которая бы формировала список, состоящий из двух подсписков: в 1-й подсписок должны попадать все элементы `x`, удовлетворяющие предикату `p`, во 2-й – не удовлетворяющие.
11. Напишите функцию (`f s n`), которая из многоуровневого списка чисел `s` создает новый список, исключив из него все элементы, которые превышают заданное число `n`.
12. Напишите функцию (`f-n p x n`), которая выдает истину только если ровно для `n` элементов списка `x` выполняется предикат `p` (`p` – функция или функциональное имя).
13. Напишите функцию (`f n l_f x`), где `n` – число, `l_f` – список арифметических операций, `x` – числовой список. Функция должна возвращать число – результат применения `n`-й операции к элементам списка `x`.

14. Напишите функцию, на вход которой подается многоуровневый список. Функция должна преобразовывать этот список в одноуровневый.
15. Напишите функцию, на вход которой подается многоуровневый список и атом. Функция должна удалять из списка все элементы, совпадающие с указанным атомом.
16. Написать функцию, удаляющую из исходного списка все числовые атомы. Исходный список может быть многоуровневым.
17. Определить функцию, которая находит максимальную сумму двух соседних чисел в заданном списке. Единственный аргумент функции является одноуровневым числовым списком.
18. Напишите функцию, которая считает полное количество атомов (на всех уровнях и не равных NIL) во входном списке.
19. Напишите функцию ( $f\ s\ n$ ), которая в многоуровневом списке чисел  $s$  суммирует только те числа, которые превышают заданное число  $n$ .
20. напишите функцию, которая бы подсчитывала количество пустых списков на всех уровнях в единственном аргументе функции – списке.
21. Напишите функцию ( $new\_list\ p\ x$ ), которая создает список из элементов списка  $x$ , НЕ удовлетворяющих предикату  $p$  ( $p$  – функция или функциональное имя).

## 2. 6 Практическое занятие «Циклы и блочные функции»

### Цель занятия

Целью проведения данного занятия является знакомство с циклами и блочными функциями и принципами их использования при написании новых функций.

### Рекомендации по подготовке к занятию

В процессе выполнения практики необходимо написать функцию с использованием циклов и блочных функций, описанную в вашем варианте

задания.

Зачастую рекурсивная обработка списков требует большого объема оперативной памяти для организации стека. В таком случае использование циклов позволяет строить более эффективные программы, минимизирующие затраты ресурсов.

Выбор требуемого цикла или блочной функции осуществлять самостоятельно. На данном занятии возможно использование функции присвоения.

### **Задание на практику**

Напишите функцию с использованием циклов и блочных функций. При необходимости используйте локальные или вспомогательные функции.

***Внимание! Рекурсию не использовать!***

В качестве заданий используйте свои варианты из практического занятия «Рекурсивные функции».

## **2.7 Практическое занятие «Создание БЗ»**

### **Цель занятия**

Целью данного практического занятия является знакомство с моделями представления знаний: фреймами и семантическими сетями, а также получение практических навыков создания Баз Знаний и работы с ними.

### **Рекомендации по подготовке к занятию**

В процессе выполнения практики необходимо создать комплекс программ, позволяющий создавать и корректировать базу знаний, а также выполнять поиск требуемой информации в созданной базе.

Задан список предметных областей для создания Базы Знаний. Выберите предметную область из предложенного списка (можно предложить собственную предметную область и согласовать ее с преподавателем).

Выберите модель представления знаний: фреймы или семантическую сеть.

При выборе фреймов необходимо построить фреймовую структуру в

заданной области: построить иерархическую классификацию и отобразить ее с помощью языка Лисп. Хотя бы один из фреймов должен содержать применяющую процедуру.

При выборе семантической сети необходимо построить ориентированный граф, определить объекты и отношения в сети; реализовать сеть с помощью языка Лисп.

Для построения можно пользоваться фрагментами программ (программами) рассмотренными на лекциях.

Создать интерфейс для ввода и отображения информации.

Реализовать функцию работы с Базой Знаний, определенную преподавателем.

### **Задание на практику**

1. Выбрать предметную область.
2. Выбрать модель представления знаний (фреймы, семантическая сеть).
3. Для фреймовой модели: определить структуру фрейма; построить иерархическую классификацию; определить хотя бы одну применяющую процедуру.
4. Для семантической сети: определить объекты сети и возможные отношения между ними; построить граф сети; определить возможные вопросы, на которые должна отвечать система.
5. Реализовать разработанную модель на языке Лисп.
6. Обязательные функции:
  - a. Добавление фрейма/объекта в БЗ
  - b. Просмотр структуры БЗ
7. Дополнительные функции:
  - a. Удаление фрейма из структуры:
    - удаление имени фрейма из списка дочерних фреймов предка;
    - удаление всех свойств фрейма;
    - удаление всех дочерних фреймов удаляемого фрейма.
  - b. Удаление объекта из сети: удаление имени объекта; удаление всех связей объекта; удаление всех объектов, связанных с удаляемым отношениями IS-A.
  - c. изменение информации о фрейме (любой слот может быть изменен).
  - d. просмотр информации о фрейме: Выводить предка фрейма, список дочерних фреймов, свойства фрейма.

- e. поиск всех фреймов с заданным свойством (в качестве поискового значения может быть задано значение любого свойства фрейма или имя слота).
- f. поиск фрейма с заданными свойствами (не обязательно полный список свойств, свойства могут быть заданы в различном порядке).
- g. Вывод всех свойств заданного фрейма (включая свойства всех предков)
- h. Поиск всех фреймов-экземпляров с заданным свойством (конечных фреймов на дереве иерархии)
- i. добавление информации о фрейме (слот может быть добавлен или удален)
- j. определение всех предков заданного фрейма (отца, деда, прадеда и т.д.)
- k. определение всех потомков заданного фрейма (детей, внуков, и т.д.)
- l. изменение имени заданного фрейма (учесть, что он может быть для других фреймов как отцом, так и потомком)

#### **Список предметных областей:**

1. цветы (ботаника).
2. дисциплины в учебном плане. Примечание: здесь каждый слот для конечного фрейма – конкретной дисциплины (лекции, практики, лаб. раб, курс. раб., самост. раб.) имеет числовое значение (количество часов).
3. книги.
4. деревья.
5. автомобили.
6. музыка.
7. фильмы.
8. птицы.
9. насекомые.
10. инструмент.
11. строения.
12. рыбы (водный мир).
13. дом/жилище.

## **3 Методические указания для организации самостоятельной работы**

### **3.1 Общие положения**

Цель самостоятельной работы по дисциплине – проработка лекционного материала, самостоятельное изучение некоторых разделов курса, подготовка к лабораторным работам, опросам на лекциях и контрольным работам.

Самостоятельная работа студента по дисциплине «Функциональное программирование и интеллектуальные системы» включает следующие виды его активности:

1. проработка лекционного материала;
2. изучение тем теоретической части дисциплины, вынесенных для самостоятельной проработки;
3. подготовка к практическим занятиям;
4. подготовка к контрольным работам;
5. подготовка к экзамену.

### **3.2 Проработка лекционного материала**

При проработке лекционного материала необходимо:

а) отработать прослушанную лекцию (прочитать конспект, прочитать учебное пособие и сопоставить записи с конспектом, просмотреть слайды) и восполнить пробелы, если они имелись (например, если вы что-то не поняли или не успели записать);

б) перед каждой последующей лекцией прочитать предыдущую, дабы не тратилось много времени на восстановление контекста изучения дисциплины при продолжающейся теме.

Данный вид деятельности ориентирован как на закрепление материала, так и на подготовку к тестовым заданиям и контрольным работам.

Содержание лекций:

Концепция и особенности функционального программирования. Свойства функциональных языков. Основные особенности Лиспа, достоинства языка.

Элементарные понятия языка Лисп: атомы и списки. Программа на языке Лисп. Вычисляемые выражения. Понятие функции, префиксная нотация.

Вычисление лямбда-выражений. Порядок редукций и нормальные формы. Определение функций в Лиспе. Базовые функции языка, предикаты.

Понятие рекурсии. Правила записи рекурсивной функции. Терминальная ветвь, рекурсивная ветвь. Прямая и косвенная рекурсия. Рекурсия с несколькими терминальными ветвями, рекурсивными ветвями.

Внутреннее представление списков. Вспомогательные функции над списками.

Глобальные и локальные переменные. Изменение значений переменных. Диалоговый режим работы. Функции ввода-вывода. Циклы и блочные функции. Обработка текстовых данных. Работа с файлами. Массивы.

Функции высших порядков. Различие между данными и функциями. Функционалы.

Обработка и хранение знаний: свойства символов; ассоциативные списки. Представление знаний: фреймы, семантические сети.

### **3.3 Самостоятельное изучение тем теоретической части курса**

#### **3.3.1 Разрушающие функции.**

Рассматриваются функции, изменяющие структуру списков. Обычные функции не изменяют структуру данных в Лиспе: при формировании новых структур выделяется новая область памяти, существующие же списки сохраняются неизменными.

**Важно!** При использовании разрушающих функций мы меняем значения одних переменных, и при этом могут измениться значения других переменных, связанных с теми же структурами.

Применение разрушающих функций удобно для изменения данных в Базах Знаний, когда данные должны быть связаны на глобальном, смысловом уровне.

Для лучшего понимания работы разрушающих функций необходимо разобраться с внутренним представлением списков. Обратите внимание, как изменяется внутреннее представление списков при их формировании базовыми функциями, и при их изменении с помощью разрушающих функций.

### 3.3.2 Графы и Деревья

Рассматриваются возможные способы представления ориентированных и неориентированных графов. Решение различных задач обработки графов напрямую зависит от способа их представления. Так, для поиска пути на графе удобнее использовать вектора смежности. Если же мы решаем задачу построения остовного дерева, граф удобнее представлять в виде списка ребер и множества узлов графа.

Представление деревьев зависит от вида дерева: обычное или бинарное. Основными действиями над деревьями являются:

- поиск элемента в дереве;
- включение элемента в дерево;
- стяжение ветви (склеивание узлов).

## 4 Рекомендуемая литература

Салмина Н. Ю., Функциональное программирование и интеллектуальные системы: учебное пособие [Электронный ресурс] / Салмина Н. Ю. — Томск: ТУСУР, 2016 . — 100 с. — Режим доступа: <https://edu.tusur.ru/publications/63572>).