



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

### **Лабораторная работа № 10**

**По дисциплине «Функциональное и логическое программирование»**

**Студент: Тимонин А. С.**

**Группа ИУ7-626**

**Преподаватель Толпинская Н. Б.**

Москва.  
2020 г.

## Практическая часть

### Задание 7.

Пусть list-of-list список, состоящий из списков. Написать функцию, которая вычисляет сумму длин всех элементов list-of-list, т.е. например для аргумента  $((1\ 2)\ (3\ 4)) \rightarrow 4$ .

### Задание 8.

Написать рекурсивную версию (с именем reg-add) вычисления суммы чисел заданного списка. Например:  $(\text{reg-add}\ (2\ 4\ 6)) \rightarrow 12$

### Задание 9.

Написать рекурсивную версию с именем recnth функции nth.

### Задание 10.

Написать рекурсивную функцию alloddr, которая возвращает t когда все элементы списка нечетные.

### Задание 11.

Написать рекурсивную функцию, относящуюся к хвостовой рекурсии с одним тестом завершения, которая возвращает последний элемент списка - аргументы.

### Задание 12.

Написать рекурсивную функцию, относящуюся к дополняемой рекурсии с одним тестом завершения, которая вычисляет сумму всех чисел от 0 до n-ого аргумента функции.

Вариант:

- 1) от n-аргумента функции до последнего  $\geq 0$ ,
- 2) от n-аргумента функции до t-аргумента с шагом d.

**Задание 13.**

Написать рекурсивную функцию, которая возвращает последнее нечетное число из числового списка, возможно создавая некоторые вспомогательные функции.

**Задание 14.**

Используя cons-дополняемую рекурсию с одним тестом завершения, написать функцию которая получает как аргумент список чисел, а возвращает список квадратов этих чисел в том же порядке.

**Задание 15.**

Написать функцию с именем select-odd, которая из заданного списка выбирает все нечетные числа.

(Вариант 1: select-even,

Вариант 2: вычисляет сумму всех нечетных чисел(sum-all-odd) или сумму всех четных чисел (sum-all-even) из заданного списка. )

# **Теоретическая часть**

## **Способы организации повторных вычислений в Lisp.**

1. Использование функционалов;
2. Использование рекурсии.

## **Что такое рекурсия?**

Рекурсия – это ссылка на определяемый объект во время его определения.

Рекурсия в Lisp - естественный принцип обработки списков.

## **Классификация рекурсивных функций в Lisp.**

1. Простая функция - один рекурсивный вызов в теле;
2. Рекурсия первого порядка - рекурсивный вызов встречается несколько раз;
3. Взаимная рекурсия - используется несколько функций, рекурсивно вызывающих друг друга.

В силу возможной сложности и разнообразия постановок задач, возможны комбинации и усложнения приведенных групп функций.

Функция называется косвенно рекурсивной в том случае, если она содержит обращение к другой функции, содержащей прямой или косвенный вызов определяемой (первой) функции. В этом случае по тексту определения функции ее рекурсивность (косвенная) может быть не видна.

Если в теле функции явно используется вызов этой же функции, то имеет место прямая рекурсия.

Рекурсия называется однократной, если функция вызывает саму себя один раз. Если функция вызывает саму себя два раза, то рекурсия называется двукратной и т.д.

## **Различные способы организации рекурсивных функций и порядок их реализации.**

Существуют типы рекурсивных функций:

1. хвостовая;

2. дополняемая;
3. множественная; взаимная рекурсия;
4. рекурсия более высокого порядка.

### **Способы повышения эффективности реализации рекурсии.**

В целях повышения эффективности рекурсивных функций рекомендуется формировать результат не на выходе из рекурсии, а на входе в рекурсию, все действия выполняя до ухода на следующий шаг рекурсии. Это и есть хвостовая рекурсия.

Для превращения не хвостовой рекурсии в хвостовую и в целях формирования результата (результатирующего списка) на входе в рекурсию рекомендуется использовать дополнительные (рабочие) параметры. При этом становится необходимым создать функцию-оболочку (как в задании 4) для реализации очевидного обращения к функции.