



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»

**Отчёт по лабораторной работе №5
по курсу «Функциональное и логическое
программирование»**

**Тема: Использование управляющих
структур, работа со списками**

Студент: Сорокин А. П. ИУ7-66Б

Преподаватели: Толпинская Н. Б.
Строганов Ю. В.

2021 г.

1. Функция, которая принимает целое число и возвращает первое четное число, не меньшее аргумента.

```
(defun even-near-2 (x) (if (evenp x) x (+ x 1)))
```

2. Функция, которая принимает число и возвращает число того же знака, но с модулем на 1 больше аргумента.

```
(defun mod+1 (x)
  (if (= x 0) 1
      (* (/ x (abs x)) (+ (abs x) 1))))
```

3. Функция, которая принимает два числа и возвращает список из этих чисел, расположенный по возрастанию.

```
(defun two-in-order (x y)
  (if (> x y) (list y x) (list x y)))
```

4. Функция, которая принимает три числа и возвращает Т только тогда, когда первое число расположено между вторым и третьим.

8. Решить задачу 4, используя для ее решения конструкции IF, COND, AND/OR.

```
(defun is-between-or-and (x y z)
  (and (or (and (<= y x) (<= x z)) (and (<= z x) (<=
x y))) T)
)
```

```
(defun is-between-cond (x y z)
  (cond
    ((<= y x) (<= x z))
    ((<= z x) (<= x y))
  )
)
```

```
(defun is-between-if (x y z)
  (if (<= y x) (<= x z)
      (if (<= z x) (<= x y)
          Nil
      )
  )
)
```

5. Результаты выражений с and и or.

| Выражение | Результат |
|------------------------------|-----------|
| (and 'fee 'fie 'foe) | foe |
| (or 'fee 'fie 'foe) | fee |
| (and nil 'fie 'foe) | Nil |
| (or nil 'fie 'foe) | fie |
| (and (equal 'abc 'abc) 'yes) | yes |
| (or (equal 'abc 'abc)) | T |

6. Предикат, который принимает два числа-аргумента и возвращает T, если первое число не меньше второго

```
(defun fun (x y) (>= x y))
```

7. Какой из следующих двух вариантов предиката ошибочен и почему?

```
(defun pred1 (x)
  (and (numberp x) (plusp x))
)
(defun pred2 (x)
  (and (plusp x) (numberp x))
)
```

Функция and выполняется до первого аргумента, равного Nil, который и возвращает. По этой причине предикат pred2 является ошибочным, поскольку если x – не число, то выполняющийся раньше plusp завершится ошибкой, в отличие от предиката pred1, в котором сначала проверяется x на число, и если x – не число, то функция and сразу вернёт Nil.

9. Функция how-alike с использованием конструкций IF, AND/OR.

```
(defun how-alike (x y)
  (if (or (= x y) (equal x y)) 'the_same
      (if (and (oddp x) (oddp y)) 'both_odd
          (if (and (evenp x) (evenp y)) 'both_even
              'difference)
          )
      )
)
```

Ответы на вопросы

1. Классификация функций.

- чистые математические: принимают фиксированное число аргументов;
- рекурсивные;
- специальные функции (формы): принимают произвольное число аргументов, которые по-разному обрабатываются;
- псевдофункции: создают эффект на внешних устройствах;
- функции с вариантными значениями, из которых выбирается только одно;
- функции высших порядков (функционалы): используются для синтаксически управляемых программ.

Классификация базисных функций:

- селекторы: car, cdr;
- конструкторы: cons, list;
- предикаты: null, atom, numberp, symbolp и т. д.
- сравнения: eq, eql, equal, = и т. д.

2. Работа функций and, or, if, cond.

and – функция-форма, выполняющая логическое умножение («И»).

```
(and <arg1> <arg2> ... <argN>)
```

Функция and проверяет последовательно каждый аргумент на Nil до тех пор, пока не встретит такой аргумент, т. к. результат логического умножения становится известным (Nil). Если все аргументы оказались равными не Nil, то возвращается последний аргумент.

Примеры:

```
(and 'A 'B Nil 'C) -> Nil  
(and Nil) -> Nil  
(and Nil 'A 'B) -> Nil  
(and 'A 'B 'C 'D) -> D
```

or – функция-форма, выполняющая логическое сложения («ИЛИ»).

```
(or <arg1> <arg2> ... <argN>)
```

Функция or проверяет последовательно каждый аргумент на Nil до тех пор, пока не встретит аргумент, который не равен Nil, т. к. результат логического сложения становится известным (T). Если все аргументы оказались равными Nil, то возвращается последний аргумент, т. е. Nil.

Примеры:

```
(or 'A 'B Nil 'C) -> A
(or Nil) -> Nil
(or Nil 'A 'B) -> A
(or Nil Nil 'C 'D Nil) -> C
```

if – функция-форма, которая возвращает различные значения в зависимости от результата выражения-условия.

```
(if <условие> <Т-выражение>[ <F-выражение>])
```

Если выражение «условие» соответствует Т (не равно Nil), то вычисляется Т-выражение, в противном случае вычисляется F-выражение. F-выражение является необязательным аргументом: если не подаётся этот аргумент, то функция if при невыполнении «условия» возвращает Nil.

cond – функция-форма, которая возвращает различные значения в зависимости от результатов принимаемых выражений-условий.

```
(cond      (test1 value1)
           (test2 value2)
           ...
           (testN valueN))
```

Функция cond принимает в качестве аргументов списки, которые содержат в качестве первого элемента предикат, а второго – выражение, которое нужно вычислить, если результат предикат не Nil.

cond вычисляет последовательно предикаты до тех пор, пока не встретится предикат, результат которого не равен Nil. В этом случае вычисляется выражение, соответствующее этому предикату, и результат этого выражения будет возвращён как результат функции cond. Если результат всех предикатов равны Nil, то функция cond возвращает Nil.

3. Способы определения функции.

- без имени: (lambda <лямбда-список> (<тело_функции>))
- с именем: (defun <имя> <лямбда-список> <тело_функции>)

<лямбда-список> - список формальных параметров