



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»

**Отчет по лабораторной работе №3
по дисциплине «Функциональное и логическое
программирование»**

Тема Работа интерпретатора Lisp

Студент Криков А.В.

Группа ИУ7-63Б

Оценка (баллы) _____

Преподаватель Толпинская Н.Б., Строганов Ю.В.

Практические задания

Задание 1

Постановка задачи

Написать функцию, которая принимает целое число и возвращает первое четное число, не меньшее аргумента.

Решение

Листинг 1: Решение задания №1

```
1 (defun f1 (x) (if (= (mod x 2) 1) (+ x 1) x))
```

Задание №2

Постановка задачи

Написать функцию, которая принимает число и возвращает число того же знака, но с модулем на 1 больше модуля аргумента.

Решение

Листинг 2: Решение задания №2

```
1 (defun abs-plus-one (x)  
2   (if (> x 0) (+ x 1) (- x 1)))
```

Задание №3

Постановка задачи

Написать функцию, которая принимает два числа и возвращает список из этих чисел, расположенный по возрастанию.

Решение

Листинг 3: Решение задания №3

```
1 (defun sorted-pair-list (fst snd)
2   (if (> fst snd) (list fst snd) (list snd fst)))
```

Задание №4

Постановка задачи

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

Решение

Листинг 4: Решение задания №4

```
1 (defun between (a b c) (or (and (> a b) (< a c)) (and (> a c) (< a b))))
```

Задание №5

Постановка задачи

Каков результат вычисления следующих выражений?

Решение

Листинг 5: Решение задания №5

```
1 (and 'fee 'fie 'foe) → FOE
2 (or nil 'fie 'foe) → FIE
3 (and (equal 'abc 'abc) 'yes) → YES
4 (or 'fee 'fie 'foe) → FEE
5 (and nil 'fie 'foe) → NIL
6 (or (equal 'abc 'abc) 'yes) → T
```

Задание №6

Постановка задачи

Написать предикат, который принимает два числа аргумента и возвращает Т, если первое число не меньше второго.

Решение

Листинг 6: Решение задания №6

```
1 (defun ge (a b)
2   (if (>= a b) T Nil))
```

Задание №7

Постановка задачи

Какой из следующих двух вариантов предиката ошибочен и почему?

Решение

Листинг 7: Решение задания №7

```
1 (defun pred1 (x)
2   (and (numberp x) (plusp x))); OK
3
4 (defun pred2 (x)
5   (and (plusp x) (numberp x))); RUNTIME ERROR
```

Второй вариант ошибочен, т.к. если в функцию будет передано не число и на него будет применена функция **plusp** (которая работает только с числовыми значениями), интерпретатор выдаст ошибку.

Задание №8

Постановка задачи

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

Решение

Листинг 8: Решение задания №8

```
1 (defun is-between-if (f s th)
2   (if (> f s)
3     (if (< f th)
4       t
5       (if (< f s)
6         (if (> f th)
7           t
8           nil)
9         nil))
10  (if (> f th)
11    t)))
```

```

12
13 (defun is-between-cond (f s th)
14   (cond ((> f s) (cond ((< f th))
15                        (t nil)))
16         ((> f th) t)))
17
18 (defun beetween (a b c) (or (and (> a b) (< a c)) (and (> a c) (< a b))))

```

Задание №9

Постановка задачи

Переписать функцию how-alike, приведенную в лекции и использующую COND, используя конструкции IF, AND/OR.

Решение

Листинг 9: Решение задания №9

```

1 (defun how-alike-cond (x y)
2   (cond ((or (= x y) (equal x y)) 'the_same)
3         ((and (oddp x) (oddp y)) 'both_odd)
4         ((and (evenp x) (evenp y)) 'both_even)
5         (t 'difference)))
6
7 (defun how-alike-if (x y)
8   (if (if (= x y) (equal x y)) 'the_same (
9       if (if (oddp x) (oddp y)) 'both_odd (
10         if (if (evenp x) (evenp y)) 'both_even 'difference))))
11
12 (defun how-alike-andor (x y)
13   (or (and (or (= x y) (equal x y)) 'the_same)
14       (and (oddp x) (oddp y) 'both_odd)
15       (and (evenp x) (evenp y) 'both_even)
16       'difference))

```

Контрольные вопросы

1. *Базис Lisp.*

- (a) атомы и структуры (представляющиеся бинарными узлами);
- (b) несколько базовых функций и функционалов: встроенные — примитивные функции (atom, eq, cons, car, cdr); специальные функции и функционалы (quote, cond, lambda, eval, apply, funcall).

2. *Классификация функций.*

- (a) чистые (математические) функции: имеют фиксированное количество аргументов и в качестве возврата единственное значение;
- (b) рекурсивные функции;
- (c) специальные функции (формы): имеют произвольное количество аргументов, либо эти аргументы обрабатываются не все одинаково;
- (d) псевдофункции: функции, эффект которых виден на внешних устройствах;
- (e) функции с вариантными значениями, из которых выбирается одно;
- (f) функции высших порядков (функционалы) используются для построения синтаксически-управляемых программ, в качестве одного из аргументов принимают описание функции.

3. *Способы создания функций.*

Обычно функции определяются при помощи макроса DEFUN. В качестве имени может использоваться любой символ. Как правило, имена функций содержат только буквы, цифры и знак минус. Список параметров функции определяет переменные, которые будут использоваться для хранения аргументов, переданных при вызове функции. Тело DEFUN состоит из любого числа выражений Lisp.

4. *Работа функций and, or, if, cond.*

Сигнатура функции **cond**:

```
(cond (предикат-1 результат-1))  
(предикат-2 результат-2)  
...  
(предикат-n результат-n)
```

Работа функции **cond**:

сначала просматриваются все предикаты в порядке следования, и если хоть один из них истинный, то `cond` возвращает результат, связанный с этим предикатом. Если ни один предикат не был истинным, то она вернет `Nil`.

Сигнатура функции **if**:

(if условие выражение-1 выражение-2)

Работа функции **if**:

если условие истинно (T), то выполняется выражение-1, иначе (Nil) – выражение-2

Сигнатура функции **and**:

(and выражение-1 выражение-2 ... выражение-n)

Работа функции **and**:

результат функции будет истинным, если все ее выражения истинны. В таком случае в качестве результата вернется значение выражения-n. В случае, если хотя бы одно выражение ложно (Nil), вычисление последующих выражений не производится и результатом функции является Nil.

Сигнатура функции **or**:

(or выражение-1 выражение-2 ... выражение-n)

Работа функции **or**:

результат функции будет ложным (Nil), если все ее выражения ложны. В случае, если хотя бы одно выражение истинно, вычисление последующих выражений не производится и результатом функции является значения выражения, которое первым в списке аргументов дало в результате истину.