



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 №5 по курсу «Функциональное и логическое программирование»

Тема Функции cond, or, and, if

Студент Кононенко С.С.

Группа ИУ7-63Б

Оценка (баллы) _____

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

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 make-first-even (num)
2   (if (evenp num) num (+ num 1)))
```

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 make-abs-greater-num (num)
2   (if (> num 0) (+ num 1) (- num 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 make-asc-pair (a b)
2   (if (> a b) (list a b) (list b a)))
```

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 is-between (a b c)
2   (if (and (> a b) (< a c)) T Nil))
```

Задание 5

Постановка задачи. Каков результат вычисления следующих выражений?

Решение.

Листинг 5 – Решение задания 5

```
1 (and 'fee 'fie 'foe) ; FOE
2 (or 'fee 'fie 'foe) ; FEE
3 (and (equal 'abc 'abc) 'yes) ; YES
4 (or nil 'fie 'foe) ; FIE
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 is-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))) ; Good
3
4 (defun pred2 (x)
5   (and (plusp x) (numberp x))) ; Error
```

Второй предикат ошибочен, потому что в случае, если на вход функции будет подан аргумент, не являющийся числом, к нему будет применена функция `plusp`, определяющая, является ли число положительным. Она определена только для чисел, поэтому в случае нечислового аргумента 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 ;; IF variant
2 (defun is-between-if (a b c)
3   (if (and (> a b) (< a c)) T Nil))
4 ;; COND variant
5 (defun is-between-cond (a b c)
6   (cond ((and (> a b) (< a c)) T)
7         (T Nil)))
8 ;; AND/OR variant
9 (defun is-between-andor (a b c)
10  (and (> a b) (< a c)))
```

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 ;; COND variant
2 (defun how-alike-cond (x y)
3   (cond ((or (= x y) (equal x y)) 'the_same)
4         ((and (oddp x) (oddp y)) 'both_odd)
5         ((and (evenp x) (evenp y)) 'both_even)
6         (T 'difference)))
7 ;; IF variant
8 (defun how-alike-if (x y)
9   (if (if (= x y) (equal x y)) 'the_same (
10      if (if (oddp x) (oddp y)) 'both_odd (
11          if (if (evenp x) (evenp y)) 'both_even 'difference))))
12 ;; AND/OR variant
13 (defun how-alike-andor (x y)
14   (or (and (or (= x y) (equal x y)) 'the_same)
15       (and (oddp x) (oddp y) 'both_odd)
16       (and (evenp x) (evenp y) 'both_even)
17       'difference))
```

Ответы на контрольные вопросы

Вопрос 1. Классификация функций.

Ответ. Функции в языке Lisp:

- чистые (с фиксированным количеством аргументов) математические функции;
- рекурсивные функции;
- специальные функции – формы (принимают произвольное количество аргументов или по разному обрабатывают аргументы);
- псевдофункции (создающие «эффект» – отображающие на экране процесс обработки данных и т.п.);

- функции с вариативными значениями, выбирающие одно значение;
- функции высших порядков – функционалы (используются для построения синтаксически управляемых программ);

Вопрос 2. Работа функций `and`, `or`, `if`, `cond`.

Ответ.

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 (`T`), если все ее выражения истинны. 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 (`T`), вычисление последующих выражений не производится и результатом функции является `T`.

Вопрос 3. Способы определения функций.

Ответ. Функцию можно задать через функцию `defun` или `lambda`.

(`defun` имя_функции (список_аргументов) тело_функции)

Например, `(defun sum (x y) (+ x y))`

Вызов функции: `(sum 2 3) -> 5`

`(lambda (список_аргументов) тело_функции)`

Например, `(lambda (x y) (+ x y))`

Вызов функции: `(lambda (x y) (+ x y) 2 3) -> 5`