



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 Управляющие структуры Lisp

Студент Пересторонин П.Г.

Группа ИУ7-63Б

Оценка _____

Преподаватель Толпинская Н. Б.

Оглавление

1	Задания	2
1.1	Написать функцию, которая принимает целое число и возвращает первое четное число, не меньшее аргумента	2
1.2	Написать функцию, которая принимает число и возвращает число того же знака, но с модулем на 1 больше модуля аргумента	2
1.3	Написать функцию, которая принимает 2 число и возвращает список из этих чисел, расположенный по возрастанию	2
1.4	Написать функцию, которая принимает 3 числа и возвращает T только тогда, когда первое число расположено между вторым и третьим	3
1.5	Каков результат вычисления следующих выражений	3
1.6	Написать предикат, который принимает два числа-аргумента и возвращает T, если первое число не меньше второго	4
1.7	Какой из следующих двух вариантов ошибочен и почему? .	4
1.8	Решить задачу 4, используя для ее решения конструкции <code>if</code> , <code>cond</code> , <code>and/or</code>	4
1.9	Переписать функцию <code>how-alike</code> приведенную в лекции и использующую <code>cond</code> , используя конструкции <code>if</code> , <code>and/or</code>	5
2	Ответы на вопросы к лабораторной работе	6
2.1	Классификация функций	6
2.2	Работа функций <code>and</code> , <code>or</code> , <code>if</code> , <code>cond</code>	6
2.3	Способы определения функций	6

1 Задания

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

```
1 (defun first-even-ge (arg)
2   (if (evenp arg) arg (+ arg 1)))
```

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

```
1 (defun module-plus (arg)
2   (+ arg (if (> arg 0) 1 -1)))
```

1.3 Написать функцию, которая принимает 2 число и возвращает список из этих чисел, расположенный по возрастанию

```
1 (defun growing-lst (a b)
2   (if (< a b) (list a b) (list b a)))
```

1.4 Написать функцию, которая принимает 3 числа и возвращает Т только тогда, когда первое число расположено между вторым и третьим

```
1 (defun pred (a b c)
2   (and (> a b) (< a c)))
```

1.5 Каков результат вычисления следующих выражений

```
1 (and 'fee 'fie 'foe)
```

Результат: foe

```
1 (or 'fee 'fie 'foe)
```

Результат: fee

```
1 (and (equal 'abc 'abc) 'foe)
```

Результат: foe

```
1 (or Nil 'fie 'foe)
```

Результат: fie

```
1 (and Nil 'fie 'foe)
```

Результат: Nil

```
1 (or (equal 'abc 'abc) 'foe)
```

Результат: T

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

```
1 (defun predicate-2 (a b)
2   (>= a b))
```

1.7 Какой из следующих двух вариантов ошибочен и почему?

1 вариант:

```
1 (defun pred1 (x)
2   (and (numberp x) (plusp x)))
```

2 вариант:

```
1 (defun pred2 (x)
2   (and (plusp x) (numberp x)))
```

Ошибочен второй вариант, потому что функция `plusp` принимает на вход один аргумент типа `number` и проверять, является ли аргумент числом, после выполнения функции `plusp` не имеет смысла, причем аргументы, не являющиеся числами, будут вызывать ошибку, в то время как 1 вариант будет работать с любым аргументом и возвращать Т для положительных чисел.

1.8 Решить задачу 4, используя для ее решения конструкции if, cond, and/or

Используя if:

```
1 (defun pred (a b c)
2   (if (> a b) (< a c) Nil))
```

Используя cond:

```

1 (defun pred (a b c)
2   (cond
3     (> a b)
4     (cond ((< a c) T) (T Nil))
5     (T Nil)))

```

Используя and/or:

```

1 (defun pred (a b c)
2   (and (> a b) (< a c)))

```

1.9 Переписать функцию how-alike приведенную в лекции и использующую cond, используя конструкции if, and/or

Используя cond:

```

1 (defun how-alike (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 'diff)))

```

Используя if:

```

1 (defun how-alike-if (x y)
2   (if (or (= x y) (equal x y)) 'the_same
3       (if (and (oddp x) (oddp y)) 'both_odd
4           (if (and (evenp x) (evenp y)) 'both_even
5               'diff))))

```

Используя and/or:

```

1 (defun how-alike-and-or (x y)
2   (or (and (or (= x y) (equal x y)) 'the_same)
3       (and (and (oddp x) (oddp y)) 'both_odd)
4       (and (and (evenp x) (evenp y)) 'both_even)
5       'diff))

```

2 Ответы на вопросы к лабораторной работе

2.1 Классификация функций

Функции в Lisp классифицируют следующим образом:

- чистые математические функции;
- рекурсивные функции;
- специальные функции — формы (сегодня 2 аргумента, завтра - 5);
- псевдофункции (создают эффект на внешнем устройстве);
- функции с вариативными значениями, из которых выбирается 1;
- функции высших порядков — функционал: используется для синтаксического управления программ (абстракция языка).

По назначению функции разделяются следующим образом:

1. конструкторы — создают значение (`cons`, например);
2. селекторы — получают доступ по адресу (`car`, `cdr`);
3. предикаты — возвращают `Nil`, `T`.

2.2 Работа функций `and`, `or`, `if`, `cond`

FIXME

2.3 Способы определения функций

FIXME (см. лр 4)