



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 Егорова П.А.

Группа ИУ7-64Б

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

Москва — 2023 г.

Задание 1

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

Решение

Листинг 1: Решение задания №1

```
1 (defun f (x) (if (evenp x) x (+ x 1)))
```

Задание №2

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 f (x) (if (> x 0) (+ x 1) (- x 1)))
```

Задание №3

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

Решение

Листинг 3: Решение задания №3

```
1 ( ( defun sort_two (f s) (if (> s f) (list f s) (list s f) ) )
```

Задание №4

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 f ( a b c) (or (and (> b a) (< a c)) (and (< b a) (> a c)) ) )
```

Задание №5

Каков результат вычисления следующих выражений?

Решение

Листинг 5: Решение задания №5

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

Задание №6

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

Решение

Листинг 6: Решение задания №6

```
1 (defun f (a b) (not (< a b)))
```

Задание №7

Какой из следующих двух вариантов предиката ошибочен и почему?

Решение

Листинг 7: Решение задания №7

```
1 (defun pred1 (x) (and (numberp x) (plusp x))); (1)
2 (defun pred2 (x) (and (plusp x) (numberp x))); (2)
```

Второй вариант ошибочен, так как перед применением функции `plusp` к аргументу необходимо убедиться в том, что аргумент является числом (`and` вычисляет аргументы до первого `NIL`, поэтому если в функцию будет передано не число, в первом варианте ошибки не произойдёт, а во втором произойдёт).

Задание №8

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

Решение

Листинг 8: Решение задания №8

```
1 (defun f (a b c)
2   (if (> b a) (> a c) (> c a))
3 )
4
5 (defun f (a b c)
6   (cond ( (> b a) (> a c))
7     ( T (> c a))
8 ))
9
10 (defun f (a b c)
11   (or (and (> b a) (< a c)) (and (< b a) (> a c)) )
12 )
```

Задание №9

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

Решение

Листинг 9: Решение задания №9

```
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 '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

Вопрос 1. Базис языка Lisp.

Ответ. Базис языка образуют атомы, структуры (точечные пары и списки), базовые

функции, базовые функционалы (функции, аргументами и значением которых являются функции).

Вопрос 2. Классификация функций языка Lisp.

Ответ.

- чистые (с фиксированным количеством аргументов) математические функции;
- рекурсивные функции;
- специальные функции – формы (принимают произвольное количество аргументов);
- псевдофункции (создающие «эффект», например, на экране);
- функционалы.

Вопрос 3. Способы создания функций.

Ответ. Функцию можно определить с помощью **defun** или **lambda**. (defun имя_функции (список_аргументов) тело_функции).

Вопрос 4. Работа функций Cond, if, and/or.

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, 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.