



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 курсу "Функциональное и логическое программирование"

Тема Работа интерпретатора Lisp

Студент Варин Д.В.

Группа ИУ7-66Б

Оценка (баллы) _____

Преподаватели Толпинская Н.Б.

Москва — 2022 г.

Задание 1

Постановка задачи

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

Листинг 1 – Решение задания №1

Решение

Задание 2

Постановка задачи

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

Решение

Задание 3

Условие задачи

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

Решение

Задание 4

Условие задачи

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

Решение

Задание 5

Условие задачи

Каков результат вычисления следующих выражений?

Решение

Листинг 2 – Решение задания №5

1	(and 'fee 'fie 'foe)	; <i>FOE</i>
2	(or 'fee 'fie 'foe)	; <i>FEE</i>
3	(and (equal 'abc 'abc) 'yes)	; <i>YES</i>
4	(or nil 'fie 'foe)	; <i>FIE</i>
5	(and nil 'fie 'foe)	; <i>NIL</i>
6	(or (equal 'abc 'abc) 'yes)	; <i>T</i>

Задание 6

Условие задачи

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

Решение

Листинг 3 – Решение задания №6

```
1 (cons 3 (list 5 6)) → (356)
2 (cons 3 '(list 5 6)) → (3 LIST 5 6)
3 (list 3 'from 9 'lives (- 9 3)) → 3 FROM 9 LIVES 6
4 (+ (length for 2 too)) (car '(21 22 23)) → error
5 (cdr '(cons is short for ans)) → (IS SHORT FOR ANS)
6 (car (list one two)); VARIABLE ONE IS UNBOUND; (car (list 'one 'two));
  → ONE
7 (car (list 'one 'two)) → ONE
```

Задание 7

Условие задачи

Какой из следующих двух вариантов предиката ошибочен и почему?

Решение

Листинг 4 – 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))) ; Not Ok
```

Задание 8

Условие задачи

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

Решение

Задание 9

Условие задачи

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

Решение

Листинг 5 – Решение задания №9

```
1 (list 'cons t NIL); (CONS T NIL)
2 (eval (list 'cons t NIL)) ; (T)
3 (eval (eval (list 'cons t NIL))) error
4 (apply #cons "(t␣NIL))
5 (eval␣NIL)
6 (list␣'eval␣NIL)
7 (eval␣(list␣'eval␣NIL))
```

Контрольные вопросы

1.Базис языка Lisp

Базис языка представлен:

- структурами, атомами;
- функциями:
atom, eq, cons, car, cdr,
cond, quote, lambda, eval, label.

2. Классификация функций языка Lisp

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).

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**.

3. Способы создания функций

С помощью макро определения **defun** или с использованием Лямбда-нотации (функция без имени).

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

Функция **and**

Синтаксис: (**and** **expression-1** **expression-2** ... **expression-n**)

Функция возвращает первое **expression**, результат вычисления которого = **Nil**. Если все не **Nil**, то возвращается результат вычисления последнего выражения.

Примеры:

1	(and 1 Nil 2) → Nil
2	(and 1 2 3) → 3

Функция **or**

Синтаксис: (**or** **expression-1** **expression-2** ... **expression-n**)

Функция возвращает первое **expression**, результат вычисления которого не **Nil**. Если все **Nil**, то возвращается **Nil**.

Примеры:

```
1 (or Nil Nil 2) -> 2
2 (or 1 2 3) -> 1
```

Функция if

Синтаксис: (if condition t-expression f-expression)

Если вычисленный предикат не Nil, то выполняется t-expression, иначе - f-expression.

Примеры:

```
1 (if Nil 2 3) -> 3
2 (if 0 2 3) -> 2
```

Функция cond

Синтаксис:

```
1 (cond
2   (condition-1 expression-1)
3   (condition-2 expression-2)
4   ...
5   (condition-n expression-n))
```

По порядку вычисляются и проверяются на равенство с Nil предикаты. Для первого предиката, который не равен Nil, вычисляется находящееся с ним в списке выражение и возвращается его значение. Если все предикаты вернут Nil, то и cond вернет Nil.

Примеры:

```
1 (cond (Nil 1) (2 3)) -> 3
2 (cond (Nil 1) (Nil 2)) -> Nil
```