



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»

**Отчёт по лабораторной работе №2
по курсу «Функциональное и логическое
программирование»**

**Тема: Использование стандартных
функций. Определение функций
пользователя**

Студент: Сорокин А. П. ИУ7-66Б

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

2021 г.

Задание 2

В данном задании '(...)' в выражениях – это некоторый список.

- 1) выражение, возвращающее второй элемент заданного списка:
(car(cdr '(...)))
- 2) выражение, возвращающее третий элемент заданного списка:
(car(cdr(cdr '(...))))
- 3) выражение, возвращающее четвёртый элемент заданного списка:
(car(cdr(cdr(cdr '(...)))))

Задание 3

Результаты выражений

Пункт	Выражение	Результат
a	(CAADR '((blue cube) (red pyramid)))	RED
b	(CDAR '((abc) (def) (ghi)))	NIL
c	(CADR '((abc) (def) (ghi)))	(DEF)
d	(CAADR '((abc) (def) (ghi)))	DEF

Задание 4

Результаты выражений

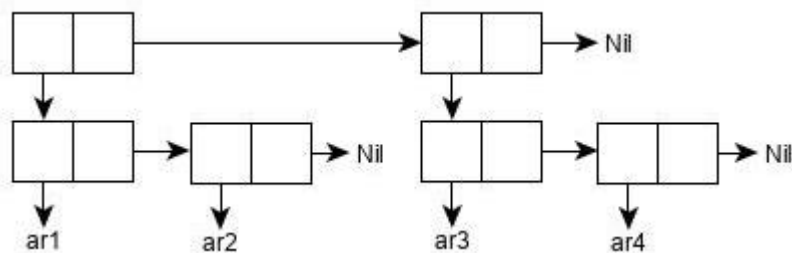
Выражение	Результат
(list 'Fred 'and Wilma)	Ошибка: UNBOUND-VARIABLE
(list 'Fred '(and Wilma))	(FRED (AND WILMA))
(cons Nil Nil)	(NIL)
(cons T Nil)	(T)
(cons Nil T)	(NIL . T)
(list Nil)	(NIL)
(cons (T) Nil)	Ошибка: UNDEFINED-FUNCTION
(cons '(T) Nil)	((T))
(list '(one two) '(free temp))	((ONE TWO) (FREE TEMP))
(cons 'Fred '(and Wilma))	(FRED AND WILMA)
(cons 'Fred '(Wilma))	(FRED WILMA)
(list Nil Nil)	(NIL NIL)
(list T Nil)	(T NIL)
(list Nil T)	(NIL T)
(cons T (list Nil))	(T NIL)
(list (T) Nil)	Ошибка: UNDEFINED-FUNCTION
(list '(T) Nil)	((T) NIL)
(cons '(one two) '(free temp))	((ONE TWO) FREE TEMP)

Задание 5

В задании представлено по два варианта реализации функции: с использованием LIST и с использованием CONS.

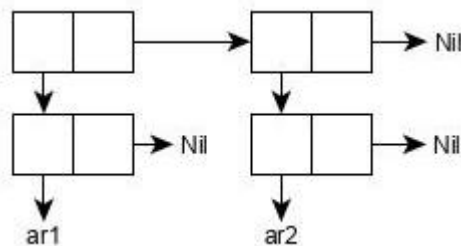
1. Функция ($f\ ar1\ ar2\ ar3\ ar4$), возвращающая список $((ar1\ ar2)\ (ar3\ ar4))$:

```
(defun f (ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4)))  
  
(defun f (ar1 ar2 ar3 ar4)  
  (cons (cons ar1 (cons ar2 nil)) (cons (cons ar3 (cons ar4  
nil)) nil)))
```



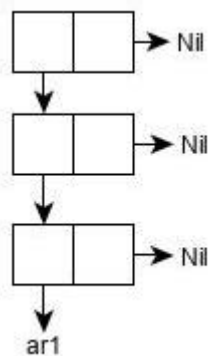
2. Функция ($f\ ar1\ ar2$), возвращающая список $((ar1)\ (ar2))$:

```
(defun f (ar1 ar2) (list (list ar1) (list ar2)))  
  
(defun f (ar1 ar2)  
  (cons (cons ar1 nil) (cons (cons ar2 nil) nil)))
```



3. Функция ($f\ ar1$), возвращающая список $((ar1))$:

```
(defun f (ar1) (list (list (list ar1))))  
  
(defun f (ar1) (cons (cons (cons ar1 nil) nil) nil))
```



Ответы на вопросы

1. Базовые элементы языка

Вся информация в языке Lisp (и данные, и программа) представляются с помощью символьных выражений, или S-выражений. К S-выражениям относятся атомы и точечные пары.

S-выражение ::= <атом> | <точечная пара>

Базовыми элементами языка являются S-выражения и списки.

Атомы – элементарные конструкции языка:

- символы (идентификаторы) – синтаксически, набор литер (букв и цифр), начинающихся с буквы;
- специальные символы – { T, Nil } – используются для обозначения «логических» констант;
- самоопределимые атомы – натуральные числа, дробные числа, вещественные числа, строки (последовательности символов, заключенных в двойные апострофы).

Точечные пары ::= (<S-выражение>.< S-выражение>).

Точечные пары (структуры) строятся с помощью бинарных узлов.

Список – динамическая структура данных, которая может быть пустая или непустая, состоящая из головы и хвоста, который является списком. В Lisp список является частным случаем S-выражения.

2. Классификация функций

- чистые математические: принимают фиксированное число аргументов;
- рекурсивные;
- специальные функции (формы): принимают произвольное число аргументов, которые по-разному обрабатываются;
- псевдофункции: создают эффект на внешних устройствах;
- функции с вариантными значениями, из которых выбирается только одно;
- функции высших порядков (функционалы): используются для синтаксически управляемых программ.

Классификация базисных функций:

- селекторы: car, cdr;
- конструкторы: cons, list;
- предикаты: null, atom, numberp, symbolp и т. д.
- сравнения: eq, eql, equal, = и т. д.

3. Функции *CAR* и *CDR*.

Функции *CAR* и *CDR* являются базисными функциями-селекторами, т. е. они осуществляют доступ к элементам списка. Обе функции – чистые математические: они принимают в качестве аргумента точечную пару или список.

Функция *CAR* переходит по *car*-указателю к первому элементу списка. В случае пустого списка вернёт *Nil*.

Функция *CDR* переходит по *cdr*-указателю к остальному списку. Если в списке меньше двух элементов, то функция вернёт *Nil*.

4. Функции *CONS* и *LIST*.

Функции *CONS* и *LIST* – функции-конструкторы. Обе функции могут использоваться для создания списков.

Функция *CONS* является базисной, чистой математической функцией. *CONS* создаёт бинарный узел и устанавливает его указатели на два принятых аргумента.

Примеры:

```
(cons 'A 'B) → (A . B)
(cons 'A '(B)) → (A B)
(cons 'A NIL) → (A)
```

Функция *LIST* является формой, т. е. принимает переменное число аргументов. *LIST* возвращает список, элементами которого являются аргументы функции: *car*-указатели ссылаются на аргументы, а *cdr*-указатели «сцепляют» списковые ячейки в список.

Примеры:

```
(list 'A 'B) → (A B)
(list 'A '(B)) → (A (B))
(list 'A NIL) → (A NIL)
```

5. Базис языка *Lisp*

Базис – это минимальный набор необходимых конструкций, с помощью которого можно реализовать задачу.

В базис *Lisp* входят:

- атомы и структуры (бинарные узлы)
- базисные функции и функционалы (*atom*, *eq*, *car*, *cdr*, *cons*, *quote*, *cond*, *lambda*, *label*, *eval*).