|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 (ИУ7)**

**ОТЧЕТ**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 2 |

**Название**: Использование базовых функций.

**Дисциплина:** Функциональное и логическое программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-63Б |  |  | В.П. Федоров |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Н.Б. Толпинская |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2021

**Задача №2**

*а) второй элемент*

(car (cdr ‘(lst))

Альтернативный вариант записи:

(CADR ‘(lst))

*b) третий элемент*

(car (cdr (cdr ‘(lst))))

Альтернативный вариант записи:

(CADDR ‘(lst))

*с) четвертый элемент*

(car (cdr (cdr (cdr ‘(lst)))))

Альтернативный вариант записи:

(CADDDR ‘(lst))

**Задача №3**

1. (CAADR ‘((blue cube) (red pyramid))) - RED
2. (CDAR ‘((abc) (def) (ghi))) - Nil
3. (CADR ‘((abc) (def) (ghi))) - (DEF)
4. (CADDR ‘((abc) (def) (ghi))) - (GHI)

**Задача №4**

|  |  |
| --- | --- |
| (list ‘Fred ‘and ‘Wilma) | (FRED AND WILMA) |
| (cons ‘Fred ‘(and Wilma)) | (FRED AND WILMA) |
| (list ‘Fred ‘(and Wilma)) | (FRED (AND WILMA)) |
| (cons ‘Fred ‘(Wilma)) | (FRED WILMA) |
| (cons Nil Nil) | (Nil) |
| (list Nil Nil) | (Nil Nil) |
| (cons T Nil) | (T) |
| (list T Nil) | (T Nil) |
| (cons Nil T) | (Nil . T) |
| (list Nil T) | (Nil T) |
| (list Nil) | (Nil) |
| (cons T (list Nil)) | (T Nil) |
| (cons ‘(T) Nil) | ((T)) |
| (list ‘(T) Nil) | ((T) Nil) |
| (list ‘(one two) ‘(free temp)) | ((one two) (free temp)) |
| (cons ‘(one two) ‘(free temp)) | ((one two) . free temp) |

**Задача №5**

*Написать функцию (f ar1 ar2 ar3 ar4), возвращающую ((ar1 ar2) (ar3 ar4)).*

(defun f(ar1 ar2 ar3 ar4)

(list (list ar1 ar2) (list ar3 ar4))

)

*Написать функцию (f ar1 ar2), возвращающую ((ar1) (ar2)).*

(defun f(ar1 ar2)

(list (list ar1) (list ar2))

)

*Написать функцию (f ar1), возвращающую (((ar1))).*

(defun f(ar1)

(list (list (list ar1)))

)

# 

# Контрольные вопросы.

1. *Классификация функций в LISP*

* по аргументам и поведения
  + Чистые функции - фиксированное количество аргументов, для определенного набора аргументов есть фиксированный результат;
  + Функции формы (специальные функции) - функции, которые принимают произвольное количество аргументов или по разному обрабатывают результат ;
  + Функции высшего порядка (Функционалы) - принимают или возвращают в качестве результата функцию.
  + псевдофункции - создают эффект на экране;
* по именованию:
  + именованные - можно определить через defun;
  + неименованные - определяются через lambda.

1. *Базис языка LISP*

**Базис** - минимальный набор возможностей, (средств) с помощью которых можно решить какую-то задачу.

**Базис Lisp** образуют атомы, структуры, базовые функции и функционалы.

1. *Список: представление и интерпретация*

Список - структура данных, состоящая из элементов, которыми могут быть атомы или другие списки. Список также может не содержать элементов вовсе, такой список называется пустым и обозначается как **Nil** или **()**.

Список является фундаментом языка LISP и может представлять как данные, так и код.

1. *Как выполняются car и cdr? Какие результаты? Примеры.*

Функции car и cdr служат для выделения головы и хвоста списка соответственно. Функции car и cdr можно применять только к списку и точечной паре. Попытка применить car и cdr к атому приведет к ошибке ERRSTATE.

Примеры:

|  |  |
| --- | --- |
| (car ‘(1 2 3)) | 1 |
| (cdr ‘(1 2 3)) | 2 3 |
| (cdr (cdr ‘(1 2 3))) | 3 |
| (cdr (cdr (cdr ‘(1 2 3)))) | Nil |
| (car ‘(a . b)) | a |
| (cdr ‘(a . b)) | b |
| (car 2) | ERRSTATE |
| (cdr 3) | ERRSTATE |

1. *Отличия в выполнении list и cons.*

list можно применить для произвольного количества аргументов, cons работает только для двух. cons объединяет значение двух своих аргументов в точечную пару, lisp формирует список.