

*Государственное образовательное учреждение высшего профессионального  
образования*

**«Московский государственный технический университет  
имени Н. Э. Баумана»  
(МГТУ им. Н.Э. Баумана)**

---

ФАКУЛЬТЕТ

«Информатика и системы управления»

КАФЕДРА

«Программное обеспечение ЭВМ и информационные технологии»

### **Отчет по лабораторной работе №9**

по курсу ФиЛП

Студент

\_\_\_\_\_  
(Подпись, дата) Якубаускайте М. А.

Преподаватель

\_\_\_\_\_  
(Подпись, дата) Толпинская Н.Б.

Москва 2019

## Практическая часть

### 5.2

Написать предикат `set-equal`, который возвращает `t`, если два его множества-аргумента содержат одни и те же элементы, порядок которых не имеет значения.

```
1 (defun set-equal (x y) (and (subsetp x y) (subsetp y x)))
```

### 5.3

Напишите необходимые функции, которые обрабатывают таблицу из точечных пар (страна.столица), и возвращают по стране - столицу, а по столице - страну.

```
1 (defun check (pair val)
2   (cond ((equal (car pair) val) (cdr pair))
3         ((equal (cdr pair) val) (car pair))
4   )
5 )
6 (defun generate-check (val)
7   (lambda (pair) (check pair val))
8 )
9
10 (defun find-in-table (base val)
11   (find-if #'(lambda (x) (not (eq x Nil)))
12           (mapcar (generate-check val) base)
13   )
14 )
```

### 6.3

Написать функцию, которая возвращает первый аргумент списка-аргумента, который сам является непустым списком.

```
1 (defun get_first_element (lst)
2   (if (and (listp (car lst))
3           (not (null (car lst))))
4       (car lst)
5       (get_first_element (cdr lst)))
6   )
7 )
8 )
```

### 6.4

Написать функцию, которая выбирает из заданного списка только числа между двумя заданными границами.

```

1 (defun select_between_inner (lst left right result)
2   (mapcar #'(lambda (x)
3     (cond ((listp x) (select_between_inner x left right result))
4     ((and (numberp x) (> x left) (< x right))
5       (nconc result (cons x nil)))
6     )
7   )
8   lst
9   )
10  (cdr result)
11 )
12
13 (defun select_between (lst left right)
14   (select_between_inner lst left right (cons nil nil))
15 )

```

## 6.5

Написать функцию, вычисляющую декартово произведение двух своих списков- аргументов.  $A \times B$  - это множество всевозможных пар  $(a, b)$ , где  $a$  принадлежит  $A$ ,  $b$  принадлежит  $B$ .

```

1 (defun decart (a b)
2   (mapcar #'(lambda (x) (mapcar #'(lambda (y) (list x y)) b)) a)
3 )

```

## 6.6

Почему так реализовано reduce, в чем причина?

```

1 (reduce #'+ ()) -> 0
2 (reduce #'* ()) -> 1

```

Сначала функция проверяет список-аргумент. Если он пуст, возвращается значение функции при отсутствии аргументов. Также reduce использует аргумент :initial-value. Этот аргумент определяет значение, к которому будет применена функция при обработке первого элемента списка-аргумента. Если список-аргумент пуст, то будет возвращено значение initial-value.