



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 №5

Название: Использование управляющих структур, работа со списками.

Дисциплина: Функциональное и логическое программирование

Студент ИУ7-64Б
(Группа)

(Подпись, дата)

Л.Е.Тартыков
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Н.Б.Толпинская
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Ю.В.Строганов
(И.О. Фамилия)

Москва, 2022

1 Теоретические вопросы

1.1 Структуроразрушающие и не разрушающие структуру списка функции.

Не разрушающие структуру списка функции – функции, которые не изменяют исходный список. Примеры таких функций:

1. **append** – имеет переменной число параметров; выполняет объединение списков в один; возвращает список в качестве значения; выполняет копирование всех элементов списка, кроме последнего.
2. **remove** – имеет два аргумента; удаляет все вхождения элемента *el* из списка *lst* (формула 1.1).

$$(remove\ el\ lst) \quad (1.1)$$

3. **reverse** – имеет один аргумент и меняет порядок элементов аргумента на противоположный.
4. **substitute** – имеет три аргумента; заменяет все элементы списка, которые равны второму аргументу на значение первого.
5. **member** – имеет два аргумента; возвращает хвост списка, начиная со списковой ячейки, удовлетворяющей первому аргументу; иначе возвращает *nil*
6. **nthcdr** – имеет два аргумента (номер элемента (*n*) и список) ; возвращает хвост списка, начиная с *n*-ого элемента.
7. **nth** – имеет два аргумента (номер элемента (*n*), список); возвращает *car*-указатель на *n*-ый элемент списка.
8. **length** – имеет один аргумент (последовательность – строки или список); возвращает число элементов второго аргумента.

Структуроразрушающие функции – функции, которые изменяют указатели списка. Таким образом, не происходит копирования элементов списка. Такие функции обычно начинаются с символа *n* (*nreverse*, *nconc*, ...). Примеры таких функций:

1. **nconc** – аналогично функции *append*, но копирования элементов не происходит.
2. **delete** – аналогично функции *remove*; не создает копии исходного списка.
3. **nreverse** – аналогично функции *reverse*; не создает копии исходного списка.
4. **nsubstitute** – аналогично функции *substitute*; не создает копии исходного списка.

1.2 Отличие в работе функций *cons*, *list*, *append*, *nconc* и в их результате.

cons – имеет два аргумента и возвращает бинарный узел. Если вторым аргументом является атом, то возвращается точечная пара; если список – список.

list – имеет произвольное число аргументов и возвращает список.

append – имеет произвольное число аргументов; важным свойством является то, что создается копия всех аргументов, кроме последнего, и дальнейшая работа ведется с ней; при этом сохраняется возможность работать с исходным списком. Таким образом, при изменении. В результате функции возвращается список.

nconc – функция, аналогичная *append* за исключением того, что исходный список не копируется (структуроразрушающая функция).

2 Практические задания

2.1 Написать функцию, которая по своему списку-аргументу `lst` определяет, является ли он палиндромом.

```
1 (defun is_palindrom (lst)
2   (equal lst (reverse lst)))
```

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

```
;lkfe
```

2.3 Напишите свои необходимые функции, которые обрабатывают таблицу из 4-х точечных пар: (страна . столица), и возвращают по стране - столицу, а по столице — страну.

```
ke
```

- 2.4 Напишите функцию `swap-first-last`, которая переставляет в списке-аргументе первый и последний элементы.
- 2.5 Напишите функцию `swap-two-element`, которая переставляет в списке- аргументе два указанных своими порядковыми номерами элемента в этом списке.
- 2.6 Напишите две функции, `swap-to-left` и `swap-to-right`, которые производят одну круговую перестановку в списке-аргументе влево и вправо, соответственно

- 2.7 Напишите функцию, которая добавляет к множеству двухэлементных списков новый двухэлементный список, если его там нет.
- 2.8 Напишите функцию, которая умножает на заданное число-аргумент первый числовой элемент списка из заданного 3-х элементного списка-аргумента, когда а) все элементы списка — числа, б) элементы списка — любые объекты.
- 2.9 Напишите функцию, `select-between`, которая из списка-аргумента из 5 чисел выбирает только те, которые расположены между двумя указанными границами-аргументами и возвращает их в виде списка (упорядоченного по возрастанию списка чисел).