

**Московский Авиационный Институт
(Национальный Исследовательский Университет)**

Факультет: “Информационные технологии и прикладная математика”
Кафедра: 806 “Вычислительная математика и программирование”

**Отчет по лабораторной работе №2
по курсу «Функциональное программирование»**

Студент: Полей-Добронравова Амелия Вадимовна

Группа: М8О-307Б, № по списку 16.

Преподаватель: Иванов Д. А., доц. каф. 806

Дата: 28.04.2021

Итоговая оценка:

Подпись преподавателя:

Москва, 2021

1. Тема работы

Простейшие функции работы со списками Коммон Лисп.

2. Цель работы

Научиться конструировать списки, находить элемент в списке, использовать схему линейной и древовидной рекурсии для обхода и реконструкции плоских списков и деревьев.

3. Задание (вариант № 2.21)

Запрограммируйте рекурсивно на языке Коммон Лисп функцию, которая чередует элементы двух списков-аргументов. Если один из аргументов длинней другого, его хвост дописывается в конец результирующего списка как есть.

4. Оборудование студента

MacBook Pro (15-inch, Mid 2012), процессор 2,3 GHz Intel Core i7, память 8 ГБ.

5. Программное обеспечение

macOS Mojave 10.14.6

6. Идея, метод, алгоритм

В языке Common Lisp есть селекторы `first`, возвращающий голову списка, и `rest`, возвращающий хвост. Проще всего реализовать задачу, образуя новый список из головы первого и рекурсивного вызова моей функции, которая возьмет голову второго, а хвост передаст вторым аргументом. Поставив два условных оператора, проверяющих, пустые ли списки (предикат `null`), можно отследить момент, когда нужно остановить вызов рекурсии и вернуть один из оставшихся списков целиком.

7. Сценарий выполнения работы

Изучить методы работы со списками в Common Lisp и написать программу. Протестировать её и отладить.

8. Распечатка программы и её результаты

Программа

```
(defun interleave (list1 list2)
  (if (null list2)
```

```

list1

(if (null list1)

    list2

    (cons (first list1) (interleave list2 (rest
list1))))))

(interleave '(a (b c) e f g) '(1 2 (3 4)))

(interleave '(a a a) '())

```

Результаты

```

CL-USER 3 : 1 > (defun interleave (list1 list2)
  (if (null list2)
      list1
      (if (null list1)
          list2
          (cons (first list1) (interleave list2 (rest list1))))))
INTERLEAVE

CL-USER 4 : 1 >
(interleave '(a (b c) e f g) '(1 2 (3 4)))
(A 1 (B C) 2 E (3 4) F G)

CL-USER 5 : 1 >
(interleave '(a) '())
(A)

CL-USER 6 : 1 > (interleave '(a a a) '())
(A A A)

```

9. Дневник отладки

№ Дата, время Событие Действие по исправлению Примечание

10. Замечания автора по существу работы

Задача довольно простая, принцип реализации лежит в простом вызове рекурсии. Единственное, что нужно использовать из возможностей работы со списками в Common Lisp - селекторы и оператор проверки на пустоту списка null.

11. Выводы

Списки были самой важной и основной составной структурой данных в традиционном LISP. На сегодняшний день Common LISP предоставляет другие структуры данных, такие как вектор, хеш-таблица, классы или структуры.

Списки — это отдельные связанные списки. В LISP списки строятся в виде цепочки простой структуры записей с именем **cons**, связанной вместе.

Списки остаются в языке потому, что они являются прекрасным решением для определенного рода проблем. Одна из них - проблема представления кода, как данных для трансформации и генерации кода в макросах - является специфичной для языка Lisp, и это объясняет то, как другие языки обходятся без списков в стиле Lisp. Вообще говоря, списки - прекрасная структура данных для представления любых неоднородных и/или иерархических данных. Кроме того они достаточно легковесны и поддерживают функциональный стиль программирования - ещё одна важная часть наследия Lisp.