

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

Студент группы 80-306 Пивницкий Даниэль Сергеевич, № по списку 13.

Контакты: pivnitskiydaniel@gmail.com

Работа выполнена: 06.06.2022

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

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 № 2.37)

Запрограммируйте рекурсивно на языке Коммон Лисп функционал $\text{map-set}(f\ X)$, аргументами которого являются функция одного аргумента f и список X , рассматриваемый как множество. Результатом вызова должно быть множество из результатов применения f к каждому из элементов X . В списки, представляющие множества, нет повторений, а порядок элементов не имеет значения.

4. Оборудование студента

Ноутбук Lenovo Legion 5 Pro, процессор Ryzen 5 5600h, память 16ГБ, 64-разрядная система.

5. Программное обеспечение

ОС Windows 10, программа LispWorks Personal Edition 7.1.2

6. Идея, метод, алгоритм

Рекурсивная обработка всех элементов списка и добавление уникальных результатов в итоговое множество.

7. Сценарий выполнения работы

1. Изучение функций для работы со списками
2. Программирование заданного функционала

8. Распечатка программы и её результаты

Программа

```
(defun map-set (f X)
  (if (null X)
      ()
      (map-cons (funcall f (first X))
                 (map-set f (rest X)))))

(defun map-cons (X Y)
  (if (null (member X Y))
      (cons X Y)
      Y))
```

Результаты

```
CL-USER 1 > (defun map-cons (X Y)
  (if (null (member X Y))
      (cons X Y)
      Y))
MAP-CONS
```

```
CL-USER 2 >
(defun map-set (f X)
  (if (null X)
      ()
      (map-cons (funcall f (first X))
                 (map-set f (rest X)))))
MAP-SET
```

```
CL-USER 3 > (map-set #'abs '(1 2 -3 -2))
(1 3 2)
```

```
CL-USER 4 > (map-set #'identity (list 20 20 30 30 40 40))
(20 30 40)
```

```
CL-USER 5 > (map-set #'abs (list 1 1 1 1 -1 -1 -1 -1))
(1)
```

```
CL-USER 6 > (map-set #'abs (list ))
NIL
```

```
CL-USER 7 > (map-set #'sqrt (list 1 4 9 10 25 25))
(1.0 2.0 3.0 3.1622777 5.0)
```

```
CL-USER 8 > (map-set (lambda (x) (+ x 10)) (list 1 2 3))
(11 12 13)
```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1				

10. Замечания автора по существу работы

Задача показалась мне не очень сложной, так как у меня есть опыт программирования на логическом языке Пролог, в котором работа со списками осуществляется схожим образом (выделение первых N элементов и оставшейся части списка).

11. Выводы

В данной лабораторной работе я познакомился со списками в языке Коммон Лисп и изучил функции для работы с ними. Я запрограммировал функционал, который применяет заданную функцию ко всем элементам списка и возвращает множество результатов.

Программа работает по следующему алгоритму: рассматриваем первый элемент списка, применяем к нему заданную функцию. Если полученного результата нет в множестве, то добавляем его туда, иначе ничего не делаем. Затем рекурсивно вызываем функцию для оставшейся части списка.

Программа протестирована на достаточном количестве примеров, покрывающих крайние случаи.