

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

Студент группы 8О-307Б-18 МАИ *Тояков Артем*, №22 по списку
Контакты: `temathesuper@mail.ru`
Работа выполнена: 09.04.2021

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806
Отчет сдан:
Итоговая оценка:
Подпись преподавателя:

1. Тема работы

Простейшие функции работы со списками common Lisp.

2. Цель работы

Изучить простейшие функции работы со списками common Lisp.

3. Задание (вариант № 2.17)

Запрограммируйте на языке Коммон Лисп функцию, которая принимает в качестве аргументов натуральное число i и список `lst` и выдает элемент списка, имеющий индекс (номер) i , индексация с 0. Если i больше или равно длине `lst`, то вернуть `NIL`.

4. Оборудование студента

Процессор: Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz, память: 3,8 Gb, разрядность системы: 64.

5. Программное обеспечение

UBUNTU 18.04.5 LTS, компилятор `sbcl`

6. Идея, метод, алгоритм

Идея в том, чтобы сравнивать индекс с 0 и уменьшать его на 1, пока он не станет 0. Тогда мы выведем требуемый элемент списка.

- `(nth-element (i list))` - вызов функции `find-el`, с предварительной проверкой корректности индекса с помощью условного оператора.

- (find-el (i list)) - рекурсивная функция: с помощью предиката cond мы проверяем условие с предикатом zerop, который, в свою очередь проверяет является ли нулевым наш индекс и если да, то мы с помощью предиката car выводим нужный элемент. Если же нет функция вызывается рекурсивно от индекса уменьшенного на 1 и списка, у которого удаляется текущий первый элемент с помощью предиката cdr.

7. Сценарий выполнения работы

- Анализ возможных реализаций поставленной задачи на common Lisp
- Изучение синтаксиса и основных функций работы со списками common Lisp
- Реализация поставленной задачи на common Lisp
- Сравнение результатов работы написанной программы и стандартного предиката nth

8. Распечатка программы и её результаты

8.1. Исходный код

```
(defun nth-element (i list)
  (if (< i (list-length list))
      (find-el i list)
  )
)

(defun find-el (index list)
  (cond
    ((zerop index) (car list))
    ((atom 1) (find-el (- index 1) (cdr list)))
  )
)
```

8.2. Результаты работы

```
* (nth-element 3 '(a0 b1 c2 d3 e4))
```

```
D3
```

```
* (nth-element 5 '(a0 b1 c2 d3 e4))
```

```
NIL
```

```
* (nth-element 4 '(ubefbew ejfegn fenfie ijijao fejfefeihf nennn))
```

```
FEJFEFEIHF
```

```
* (nth-element 0 '(1 2 3 2343 3242))
```

1

```
* (nth-element 6 '(7df7 s7fy ew8 wbf3 23jb2 uhf9 we2))
```

WE2

```
* (nth 3 '(a0 b1 c2 d3 e4))
```

D3

```
* (nth 5 '(a0 b1 c2 d3 e4))
```

NIL

```
* (nth 4 '(ubefbew ejfegn fenfie ijijao fejfefeihf nennn))
```

FEJFEFEIHF

```
* (nth 0 '(1 2 3 2343 3242))
```

1

```
* (nth 6 '(7df7 s7fy ew8 wbf3 23jb2 uhf9 we2))
```

WE2

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание

10. Замечания автора по существу работы

Замечаний нет.

11. Выводы

В ходе данной работы я познакомился с представлением и основными особенностями списков в common Lisp. Список в common Lisp представляет собой S-выражение вида: атом | список(хвост). В конце работы я сравнил результат работы моей программы и стандартного предиката nth и после того, как они вывели идентичные значения для одинаковых входных данных, я убедился в том, что алгоритм реализован корректно.