

Лабораторная работа 4.

Методические указания

Использование управляющих структур, работа со списками

Цель работы: приобрести навыки работы с управляющими структурами Lisp.

Задачи работы: изучить работу функций с произвольным количеством аргументов, функций разрушающих и неразрушающих структуру исходных аргументов.

Краткие теоретические сведения

Многие стандартные функции Lisp являются формами и реализуют особый способ работы со своими аргументами. К таким функциям относятся функции, позволяющие работать с произвольным количеством аргументов, или особым образом обрабатывающие свои аргументы, или и то и другое: cond, if, and, or, append, reverse, last, remove и др.

Если на вход функции подается структура данных (список), то возникает вопрос: сохранится ли возможность в дальнейшем работать с исходными структурами, или они изменятся в процессе реализации функции. В Lisp существуют функции, использующие списки в качестве аргументов и разрушающие или не разрушающие структуру исходных аргументов. При этом часть из них позволяет использовать произвольное количество аргументов, а часть нет. Функция append объединяет списки; length, reverse, last — одноаргументные функции, работают по верхнему уровню списковых ячеек; функция (remove el lst) удаляет el из lst, проходя по верхнему уровню списковых ячеек.

Для выполнения лабораторной работы может потребоваться функция, которая символу ставит в соответствие некоторое значение. Для этого может быть использована форма setf, которая первому аргументу ставит в соответствие значение второго аргумента, например: (setf lst1 '(a b))

Во многие реализации Lisp включена форма :

```
( Let ( (x1 a1)
      (x2 a2)
      (xn an) )
      форма1
```

форма2...) , которая устанавливает значения a1...an для лексических параметров x1 ... xn, позволяя использовать эти значения в формах. Форма Let является синтаксическим видоизменением lambda-вызова, в котором формальные и фактические параметры помещены совместно в начале формы: ((lambda (x1 x2 ... xn) форма1 форма2 ...) a1 a2 ... an).

Указания к выполнению работы

При выполнении лабораторной работы следует

- изучить правила и особенности работы функций: cond, if, and, or , append, reverse, last

Отчет по лабораторной сдается в письменной форме по окончании работы.

Задания:

Теор вопросы:

1. синтаксическая форма и хранение программы в памяти,
2. Трактовка элементов списка,
3. Порядок реализации программы,
4. Способы определения функции,
5. Работа со списками

Практические задания (Common Lisp):

1. Чем принципиально отличаются функции `cons`, `list`, `append`?

Пусть (setf lst1 '(a b c))

```
(setf lst2 '( d e)).
```

Каковы результаты вычисления следующих выражений?

```
(cons lst1 lst2)
```

```
(list lst1 lst2)
```

```
(append lst1 lst2)
```

2. Каковы результаты вычисления следующих выражений, и почему?

```
(reverse '(a b c))
```

```
(reverse ())
```

```
(reverse '(a b (c (d))))
```

```
(reverse '((a b c)))
```

(reverse '(a))

(last '(a b c))

(last '(a b (c)))

(last '(a))

(last ())

```
(last '((a b c)))
```

3. Написать, по крайней мере, два варианта функции, которая возвращает последний элемент своего списка-аргумента.

4. Написать, по крайней мере, два варианта функции, которая возвращает свой список аргумент без последнего элемента.

5. Напишите функцию `swap-first-last`, которая переставляет в списке-аргументе первый и последний элементы.

6. Написать простой вариант игры в кости, в котором бросаются две правильные кости. Если сумма выпавших очков равна 7 или 11 — выигрыш, если выпало (1,1) или (6,6) — игрок имеет право снова бросить кости, во всех остальных случаях ход переходит ко второму игроку, но запоминается сумма выпавших очков. Если второй игрок не выигрывает абсолютно, то выигрывает тот игрок, у которого больше

очков. Результат игры и значения выпавших костей выводить на экран с помощью функции print.

7. Написать функцию, которая по своему списку-аргументу lst определяет является ли он палиндромом (то есть равны ли lst и (reverse lst)).
8. Напишите свои необходимые функции, которые обрабатывают таблицу из 4-х точечных пар:
(страна . столица), и возвращают по стране - столицу, а по столице — страну.
9. 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.