

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Таныгин Максим Олегович
Должность: и.о. декана факультета фундаментальной и прикладной информатики
Дата подписания: 21.09.2023 12:56:21
Уникальный программный ключ:
65ab2aa0d384efe8480e6a4c688eddbc475e411a

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования

«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Лукичинова

« 24 » 12



ОПИСАНИЕ И ВЫЗОВ ОСНОВНЫХ ФУНКЦИЙ В ЯЗЫКЕ LISP

Методические указания по выполнению лабораторной работы по дисциплине «Функциональное и логическое программирование» для студентов направления подготовки 09.03.04 «Программная инженерия»

Курск 2017

УДК 004.65

Составители: В.Г. Белов, Т.М. Белова

Рецензент

Кандидат технических наук, доцент кафедры программной инженерии ЮЗГУ И.Н. Ефремова

Описание и вызов основных функций в языке LISP: методические указания по выполнению лабораторной работы по дисциплине "Функциональное и логическое программирование" для студентов направления подготовки 09.03.04 "Программная инженерия" / Юго-Зап. гос. ун-т; сост.: В.Г. Белов, Т.М. Белова, – Курск, 2017. – 17 с.

Дано описание и примеры использования основных функций AutoLisp.

Материал предназначен для студентов направления подготовки 09.03.04 «Программная инженерия» специализации «Разработка программно-информационных систем»

Подписано в печать *24.12.17*. Формат 60x84 1/16.
Усл. печ. л. *04*. Уч.-изд. л. *06*. Тираж 100 экз. Заказ *4387*. Бесплатно.
Юго-Западный государственный университет

305040, Курск, ул.50 лет Октября, 94.

Содержание

1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ.....	4
2 ФУНКЦИИ ПРИСВОЕНИЯ ЗНАЧЕНИЯ СИМВОЛУ	5
2.1 Функция SET	5
2.2 Функция SETQ	5
2.3 Выполнение раздела 2 лабораторной работы	6
3 ФУНКЦИИ ВЫБОРА	7
3.1 Функция CAR	7
3.2 Функция CDR	8
3.3 Выполнение раздела 3 лабораторной работы	8
4 ФУНКЦИЯ КОНСТРУКТОРА.....	9
4.1 Функция CONS.....	9
4.2 Выполнение раздела 4 лабораторной работы	10
5 ФУНКЦИЯ ОПРЕДЕЛЕНИЯ ФУНКЦИИ	10
5.1 Функция DEFUN	10
5.2 Выполнение раздела 5 лабораторной работы	11
6 СОСТАВНЫЕ ФУНКЦИИ ВЫБОРА.....	12
6.1 Выполнение раздела 6 лабораторной работы	14
7 ПРОГРАММА, РЕАЛИЗУЮЩАЯ ЛИНЕЙНУЮ ПОСЛЕДОВАТЕЛЬНОСТЬ ДЕЙСТВИЙ.....	14
7.1 Выполнение раздела 7 лабораторной работы	14
8 СОДЕРЖАНИЕ ОТЧЕТА	16
9 КОНТРОЛЬНЫЕ ВОПРОСЫ К ЗАЩИТЕ ЛАБОРАТОРНОЙ РАБОТЫ.....	17

1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ

Целью лабораторной работы является приобретение знаний и умения для составления простейшей программы на языке LISP, реализующую линейную последовательность действий с применением базовых функций языка LISP.

2 ФУНКЦИИ ПРИСВОЕНИЯ ЗНАЧЕНИЯ СИМВОЛУ

Функции присвоения значения используются обычно для задания начальных значений переменным. Они позволяют изменять значения формальных аргументов функций без необходимости вызова функции.

2.1 Функция SET

Функция (SET символ объект) вычисляет значение <символа> и присваивает этому значению вычисленное значение <объекта>. Если вычисленное значение <символа> не является символом, то SET генерирует прерывание по ошибке "Несимвольный аргумент". В качестве значения SET возвращает вычисленное значение <объекта>.

Пример:

```
(SET 'FOO '(A B C)) --> (A B C) ; This is equivalent
```

```
FOO --> (A B C) ; to (SETQ FOO '(A B C))
```

```
(SET 'PET 'DOG) --> DOG
```

```
(SET PET 'ROVER) --> ROVER
```

```
DOG --> ROVER
```

```
PET --> DOG .
```

2.2 Функция SETQ

Функция (SETQ символ форма) в отличие от SET не вычисляет значение <символа> и присваивает вычисленное значение <формы> <символу>. Если <символ> не является символом, то SETQ генерирует прерывание по ошибке "Несимвольный аргумент".

Если SETQ задана более, чем с 2-мя аргументами, то вычисление значений `<форм>` и присвоение вычисленных значений осуществляется последовательно. Если задано нечетное количество аргументов, то последний символ принимается за NIL.

SETQ возвращает новое значение последнего задаваемого символа.

Пример:

```
(SETQ FOO '(D E F)) --> (D E F)
```

```
FOO --> (D E F)
```

```
(SETQ FOO (CDR FOO)) --> (E F)
```

```
FOO --> (E F)
```

```
(SETQ SUM 5) --> 5
```

```
(SETQ SUM (+3 4) SQR (* SUM SUM)) --> 49
```

```
SUM --> 7
```

```
SQR --> 49 .
```

2.3 Выполнение раздела 2 лабораторной работы

2.3.1 Запустить систему AutoLisp.

2.3.2 Выполнить все перечисленные ниже вызовы, фиксируя результат каждого вызова

```
$ X
```

```
$ 'X
```

```
$ (SETQ X 'Y)
```

```
$ Y
```

```
$ 'Y
```

```
$ X
```

\$ 'X

\$ (SET X 'Z)

\$ Y

\$ 'Y

\$ X

\$ 'X

\$ (SETQ SP '(A B C D E F G H I))

\$ SP

\$ (SETQ SPD '((A B C) (D E F) (G H I)))

\$ SPD .

3 ФУНКЦИИ ВЫБОРА

Основной структурой данных в muLISP является бинарное дерево, состоящее из cons-ов (ветвей) и заканчивающееся атомами (листьями). Основные функции выбора CAR и CDR выбирают соответственно car (левую) и cdr (правую) ветви деревьев.

3.1 Функция CAR

Функция (CAR объект) возвращает car-элемент <объекта> (т.е. объект данных, на который указывает car-элемент <объекта>). Корректная интерпретация car-элемента объекта зависит оттого, является ли объект атомом, а если нет, рассматривается ли он как список или бинарное дерево. Если объект - атом, то его car-элемент есть текущее значение атома. Если объект - список, его car-элемент есть первый элемент списка. Если объект - бинарное дерево, его car-элемент есть левая ветвь дерева, или car-ветвь.

Пример:

(CAR '(A B C D E)) --> A

(CAR '(A.B) . C)) --> (A.B)

(SETQ NUM 7) --> 7

(CAR 'NUM) --> 7 .

3.2 Функция CDR

Функция (CDR объект) возвращает cdr-элемент <объекта> (т.е. объект данных, на который указывает cdr-элемент <объекта>). Корректная интерпретация cdr-элемента объекта зависит оттого, является ли объект атомом, списком или бинарным деревом.

Если объект - символ, его cdr-элемент есть список свойств символа. Если объект - число, его cdr-элемент указывает признак и тип числа . Если объект - список, его cdr-элемент есть остаток списка (т.е. все, кроме первого, элементы списка). Если объект - бинарное дерево, его cdr-элемент есть правая ветвь дерева, или cdr-ветвь.

Пример:

(CDR '(A B C D E)) --> (B C D E)

(CDR '((A.B) . C)) --> C

(CDR 'JOE 'SEX 'MALE) --> MALE

(CDR 'JOE) --> ((SEX . MALE)) .

3.3 Выполнение раздела 3 лабораторной работы

Выполнить все перечисленные ниже вызовы, фиксируя результат каждого вызова.

\$ (CAR '(A B C D E F G H I))

\$ (CDR '(A B C D E F G H I))

\$ SP

\$ (CAR SP)

\$ (CDR SP)

\$ SPD

\$ (CAR SPD)

\$ (CDR SPD) .

4 ФУНКЦИЯ КОНСТРУКТОРА

Lisp программы могут автоматически генерировать новые структуры данных, используя функции конструктора. Эти функции могут создавать бинарные деревья или связанные списки, которые моделируют структуры данных практически для любой задачи.

4.1 Функция CONS

Функция (CONS объект1 объект2) возвращает cons, у которого car-элемент указывает на <объект1>, а cdr-элемент - на <объект2>. Корректная интерпретация (CONS объект1 объект2) зависит от того, как рассматривается <объект2>. Если <объект2> - список, CONS создает список, первым элементом которого является <объект1>, а остатком - <объект2>. Если <объект2> - атом или бинарное дерево, CONS создает дерево, у которого левая ветвь, или car-ветвь есть <объект1>, а правая ветвь, или cdr-ветвь - <объект2>.

Пример:

(CONS 'A '(B C D)) --> (A B C D)

(CONS 'A 'B) --> (A . B) .

4.2 Выполнение раздела 4 лабораторной работы

Выполнить все перечисленные ниже вызовы, фиксируя результат каждого вызова.

\$ (CONS 'A 'B)

\$ (CONS 'A NIL)

\$ (CONS 'A '(B C D))

\$ (CONS '(A P R) '(P P P))

\$ (CONS '(A P R) (CONS '(P P P) NIL))

\$ (CONS '(A B C) '((D E F) (G H I))) .

5 ФУНКЦИЯ ОПРЕДЕЛЕНИЯ ФУНКЦИИ

5.1 Функция DEFUN

Для задания определения функции может использоваться

Функция (DEFUN name (arg1 arg2 ...)

task1

task2

...

taskn) ,

где name - имя переменной, с которой связывается определение функции; arg1, arg2 ... - имена формальных аргументов;

task1 , task2 ... - вызовы функций, которые кроме прочих используют формальные аргументы arg1, arg2

Вызов определенной таким образом функции осуществляется при помощи записи

(name farg1 farg2 ...) ,

где name - имя определенной функции;

farg1, farg2 ... - фактические аргументы функции name.

При определении функции через DEFUN следует учитывать, что при вызове таким образом определенной функции сначала произойдет сохранение текущих значений формальных аргументов,

а затем произойдет присвоение им вычисленных значений фактических аргументов.

В качестве значения функция DEFUN возвращает имя функции т.е name.

5.2 Выполнение раздела 5 лабораторной работы

5.2.1 Выполнить все перечисленные ниже вызовы, фиксируя результат вызова

\$ (DEFUN HEAD (LST)

(CAR LST))

\$ (DEFUN TAIL (LST)

(CDR LST))

\$ SP

\$ (HEAD SP)

\$ (TAIL SP)

\$ SPD

\$ (HEAD SPD)

\$ (TAIL SPD)

\$ (DEFUN CHT (LST)

(CONS (HEAD LST) (TAIL LST)))

\$ SP

\$ (CHT SP)

\$ (CHT (TAIL SP))

\$ SPD

\$ (СНТ SPD) .

6 СОСТАВНЫЕ ФУНКЦИИ ВЫБОРА

Разрешается использовать функции, являющиеся комбинациями CAR и CDR. Имя каждой из этих функций начинается с "С" и заканчивается на "R". Между ними располагается последовательность из букв "А" и "D", указывающих, как комбинируются CAR и CDR при выполнении функции. Например:

CAAR (object)

CDAR (object)

CDDR (object)

CAAAR (object)

CDAAR (object)

.....

CAAAAR (object)

.....

CDDDDR (object) .

Типичными определениями для составных функций выбора будут:

(DEFUN CAAR (OBJ)

(CAR (CAR OBJ)))

(DEFUN CADDR (OBJ)

(CAR (CDR (CDR OBJ))))

(DEFUN CDADAR (OBJ)

(CDR (CAR (CDR (CAR OBJ)))))

```

(SETQ TREE '((((A . B) . (C . D)) . ((E . F) . (G . H))) .
  (((I . J) . (K . L)) . ((M . N) . (O . P)))) )
(CAR TREE) --> (((A . B) . (C . D)) . ((E . F) . (G .
H)))
(CDR TREE) --> (((I . J) . (K . L)) . ((M . N) . (O . P)))
(CAAR TREE) --> ((A . B) . (C . D))
(CDAR TREE) --> ((E . F) . (G . H))
(CADR TREE) --> ((I . J) . (K . L))
.....
(CAAAR TREE) --> (A . B)
(CDAAR TREE) --> (C . D)
(CADAR TREE) --> (E . F)
(CDDAR TREE) --> (G . H)
.....
(CAAAAR TREE) --> A
(CDAAAAR TREE) --> B
(CADAAAAR TREE) --> C
.....
(CDDDDR TREE) --> P .

```

Часто более естественным бывает рассматривать объект данных как список, чем как сильно разветвленное бинарное дерево. Следующие селекторы широко используются для извлечения элементов списка:

```

(CAR '(A B C D E)) --> A
(CDR '(A B C D E)) --> (B C D E)
(CADR '(A B C D E)) --> B

```

(CADDR '(A B C D E)) --> C

(CADDDR '(A B C D E)) --> D .

Данным селекторам могут быть присвоены при помощи определения функции другие мнемонические имена: CAR может замениться на FIRST, CDR замениться на REST, CADR замениться на SECOND и т.д.

6.1 Выполнение раздела 6 лабораторной работы

Выполнить все перечисленные ниже вызовы, фиксируя результат вызова.

\$ SP

\$ (CADDDR SP)

\$ SPD

\$ (CAADDR SPD) .

7 ПРОГРАММА, РЕАЛИЗУЮЩАЯ ЛИНЕЙНУЮ ПОСЛЕДОВАТЕЛЬНОСТЬ ДЕЙСТВИЙ

7.1 Выполнение раздела 7 лабораторной работы

7.1.1 Составить программу для вычисления функции OTW.

Составить программу для вычисления функции OTW, аргумент которой LST представляет собой список, содержащий три подсписка, в каждом из которых по три элемента. Например, значением LST может быть список

((A B C) (D E F) (G H I))) .

Значением функции OTW должен быть список, содержащий три подсписка, в каждом из которых по три элемента. При этом в первом подсписке все три элемента являются первыми элементами

подписков LST, во втором подписке все три элемента являются вторыми элементами подписков LST, в третьем подписке все три элемента являются третьими элементами подписков LST. Так для приведенного примера результат будет

((A D G) (B E H) (C F I)) .

Программа должна содержать определения пяти функций, главное из которых есть определение функции OTW. В этом определении должен быть только один единственный вызов функции CONS с соответствующими аргументами, являющимися вызовами двух других функций.

Последовательность действий, которая осуществляется в результате вызовов всех функций должна быть следующей:

- строится первый подпись SP1;
- строится второй подпись SP2;
- строится третий подпись SP3;
- подписки SP2 и SP3 объединяются в список SP23;
- подписание SP1 и список SP23 объединяются в результирующий список.

При составлении определений функций необходимо учитывать, что вызов функции верхнего уровня вложенности вызовов функций осуществляется первым, а результат ее применения формируется последним. Это замечание справедливо для функции любого уровня вложенности вызовов функций.

В определениях функций необходимо использовать только вызовы функций CONS, CA..DR и функций, определенных через вызовы этих функций.

7.1.2 Запустить AutoLisp набрать в нем определение функции OTW.

7.1.3 Загрузить определение функции OTW.

7.1.4 Отладить функцию OTW, используя при трассировке имена всех самостоятельно определенных функций и, если необходимо, имена функций CA..DR.

7.1.5 Результат окончательной трассировки зафиксировать для отчета.

8. СОДЕРЖАНИЕ ОТЧЕТА

Вызовы всех функций подраздела 2.2. с результатами их применения и объяснением смысла применения.

Вызовы всех функций подраздела 3.2. с результатами их применения и объяснением смысла применения.

Вызовы всех функций подраздела 4.2. с результатами их применения и объяснением смысла применения.

Вызовы всех функций подраздела 5.2. с результатами их применения и объяснением смысла применения.

Описание функции OTW.

Результаты трассировки отлаженной функции OTW.

9. КОНТРОЛЬНЫЕ ВОПРОСЫ К ЗАЩИТЕ ЛАБОРАТОРНОЙ РАБОТЫ

Какое значение имеет символ при первом появлении в среде AutoLisp?

Какими способами можно присвоить значение символу?

Что происходит со значением символа, который является формальным параметром функции, при вызове этой функции?

Что происходит со значением символа, который является формальным параметром функции, после применения этой функции?

Как надо определить функцию, выбирающую седьмой элемент списка?

Каким образом при вызове функции, определенной при помощи DEFUN, можно передать не значение фактического параметра, а его имя?

Каким образом используется принцип нисходящего проектирования при составлении программ на языке LISP?