

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и
информатики»

Кафедра прикладной математики и кибернетики

Лабораторная работа № 3
«Предикаты и функционалы»
по дисциплине «Функциональное и логическое программирование»

Бригада № 6

Выполнил:
студент 3 курса
группы ИП-811
Мироненко К. А

Проверил:
доцент кафедры ПМиК
Галкина М.Ю.

Оглавление

1. Постановка задачи.....	3
2. Примеры работы программы	5
<i>Приложение</i> Листинг.....	6

1. Постановка задачи

Каждая бригада должна выполнить 3 задачи (номера выбираются из таблицы). В теле функции использование операторов set, let и setq не допускается! Все функции должны быть рекурсивными.

Номер бригады	1	2	3	4	5	6	7	8	9	10
Номера задач	1	2	3	2	1	3	2	3	1	2
	4	5	6	7	8	5	6	8	7	4
	9	10	11	12	13	12	9	11	13	10

Определите предикат, проверяющий:

- 1) Равенство двух множеств.
- 2) Является ли одно множество подмножеством другого.
- 3) Являются ли два множества пересекающимися.

Определите функцию:

- 4) Возвращающую пересечение двух множеств.
- 5) Возвращающую объединение двух множеств.
- 6) Возвращающую разность двух множеств, т.е. множество из элементов первого множества, не входящих во второе.
- 7) Возвращающую симметрическую разность двух множеств, т.е. множество из элементов, не входящих в оба множества.
- 8) Транспонирующую матрицу, представленную в виде двухуровневого списка.

Например, для матрицы размера 3×4 ((1 2 3 4) (5 6 7 8) (9 10 11 12)) \rightarrow ((1 5 9) (2 6 10) (3 7 11) (4 8 12))

Определите функционал:

- 9) Вставляющий перед каждым элементом списка, обладающим определенным свойством, символ *. Проверьте работу функционала для предикатов:
 - неотрицательное число (функциональный аргумент – лямбда выражение);
 - четное число (функциональный аргумент – имя встроенного предиката evenp).
- 10) Заменяющий все элементы списка, не обладающие определенным свойством, на символ *. Проверьте работу функционала для предикатов:

- число (функциональный аргумент – имя встроенного предиката `numberp`);
- отрицательное число (функциональный аргумент – лямбда выражение).

11) Выполняющий определенную операцию над соответствующими элементами двух списков (Используйте применяющий функционал `FUNCALL`). Проверьте работу функционала для операций:

- выбор максимального элемента (функциональный аргумент – лямбда выражение);
- деление (функциональный аргумент – имя встроенной функции `/`).

12) Аналогичный предикату `MAPLIST` для одноуровневого списка. (Используйте применяющий функционал `FUNCALL`).

13) Возвращающий истину, если определенным свойством обладает хотя бы один элемент списка. (Используйте применяющий функционал `FUNCALL`). Проверьте работу функционала для предикатов:

- неположительное число (функциональный аргумент – лямбда выражение);
- символ (функциональный аргумент – имя встроенного предиката `symbolp`).

2. Примеры работы программы

```
PS D:\STUDY\Functional&LogicalProgramming\Labs> clisp .\lab3.lisp
```

```
#####  
#           Бригада #6           #  
#####
```

Определите предикат, проверяющий:

3) Являются ли два множества пересекающимися.

Результат:

A=(1 2), B=(3 1 4) -> T

Определите функцию:

5) Возвращающую объединение двух множеств.

Результат:

A=(1 2), B=(3 1 4) -> (2 3 1 4)

A=(1 2 3 4 5), B=(0 1 2 3 4 5 A) -> (0 1 2 3 4 5 A)

Определите функционал:

12) Аналогичный предикату MAPLIST для одноуровневого списка. (Используйте применяющий функционал FUNCALL).

Результат:

A=(1 2 3 4 5), fun=#<SYSTEM-FUNCTION REVERSE> -> ((5 4 3 2 1) (5 4 3 2) (5 4 3) (5 4) (5))

A=(A B C), fun=#<FUNCTION LAMBDA (X) (CONS 0 X)> -> ((0 A B C) (0 B C) (0 C))

```
PS D:\STUDY\Functional&LogicalProgramming\Labs>
```

Приложение Листинг

```
;;; Каждая бригада должна выполнить 3 задачи (номера выбираются из таблицы). В теле
;;; функции использование операторов set, let иsetq не допускается!
;;; Все функции должны быть рекурсивными
;;; Бригада №6: 3, 5, 12
```

```
(format t "~%#####~%#
#~%#####~%~%~%")
```

Бригада #6

```

;;; Определите предикат, проверяющий:
;;; 3) Являются ли два множества пересекающимися.
(format t "Определите предикат, проверяющий:~%3) Являются ли два множества
пересекающимися.~%Результат:~%")
(defun intersectingSets (A B)
  (cond
    ((null A) NIL)
    ((member (CAR A) B) T)
    (T (intersectingSets (CDR A) B)))
  )
)
(setq A '(1 2))
(setq B '(3 1 4))
(format t " A=~A, B=~A -> ~A~%" A B (intersectingSets A B))
(format t " ~%~%")

```

```

;;; Определите функцию:
;;; 5) Возвращающую объединение двух множеств.
(format t "Определите функцию:~%5) Возвращающую объединение двух множеств.~%Результат:~%" )
(defun unionSets (A B)
  (cond
    ((null A) B)
    ((null B) A)
    ((member (CAR A) B) (unionSets (CDR A) B))
    (T (cons (CAR A) (unionSets (CDR A) B))))
  )
)
(setq A '(1 2))
(setq B '(3 1 4))
(format t "  A=~A, B=~A -> ~A~%" A B (unionSets A B))
(setq A '(1 2 3 4 5))
(setq B '(0 1 2 3 4 5 a))
(format t "  A=~A, B=~A -> ~A~%" A B (unionSets A B))
(format t " ~%~%" )

```

```
;; Определите функционал:
;; 12) Аналогичный предикату MAPLIST для одноуровневого списка.
;; (Используйте применяющий функционал FUNCALL).
(format t "Определите функционал:~%12) Аналогичный предикату MAPLIST для одноуровневого
списка.(Используйте применяющий функционал FUNCALL).~%Результат:~%")
(defun maplst (fun A)
  (cond
    ((null A) NIL)
    (T (cons (funcall fun A) (maplst fun (CDR A))))
  )
)
```

```
)  
(setq A '(1 2 3 4 5))  
(setq fun (function reverse))  
(format t "  A=~A, fun=~A -> ~A~%" A fun (maplst fun A))  
(setq A '(a b c))  
(setq fun (function (lambda (x) (cons 0 x))))  
(format t "  A=~A, fun=~A -> ~A~%" A fun (maplst fun A))  
(format t "~%~%")
```