

Московский государственный технический университет
имени Н. Э. Баумана



Факультет: Информатика и системы управления

Кафедра: Программное обеспечение ЭВМ и информационные технологии

Дисциплина: Функциональное и логическое программирование

Лабораторная работа №3

Выполнили: Никичкин А. С., Фокеев А. И.
Группа: ИУ7–61

Москва, 2015 г.

1 Написать функцию

Задание 1.1 Функция принимает целое число и возвращает первое чётное число, не меньшее аргумента.

```
1 (defun problem-1-1 (arg)
2   "Return arg if even else arg+1"
3   (if (evenp arg)
4       arg
5       (+ 1 arg)))
```

Задание 1.2 Функция принимает число и возвращает число того же знака, но с модулем на единицу больше модуля аргумента.

```
1 (defun problem-1-2 (arg)
2   (* (signum arg)
3      (+ 1 (abs arg))))
```

Задание 1.3 Функция принимает два числа и возвращает список из этих чисел отсортированный по возрастанию.

```
1 (defun problem-1-3 (arg1 arg2)
2   (sort (list arg1 arg2)
3         #'<))
```

2 Реализовать функцию разными способами

Задание 2.1 Функция принимает три числа и возвращает Т если первое число расположено между вторым и третьим.

- с использованием функции SORT;

```
1 (defun is-first-between1 (arg1 arg2 arg3)
2   (equal arg1
3          (second (sort (list arg1 arg2 arg3)
4                        #'<))))
```

- с помощью специального оператора IF;

```
1 (defun is-first-between2 (arg1 arg2 arg3)
2   (if (< arg2 arg1)
3       (< arg1 arg3)
4       (if (< arg3 arg1)
5           (< arg1 arg2))))
```

- с помощью макросов **AND** и **OR**;

```

1 (defun is-first-between3 (arg1 arg2 arg3)
2   (or (and (< arg2 arg1)
3           (< arg1 arg3))
4       (and (< arg3 arg1)
5           (< arg1 arg2))))

```

- с помощью макроса **COND**.

```

1 (defun is-first-between4 (arg1 arg2 arg3)
2   (cond ((< arg2 arg1)
3         (< arg1 arg3))
4         ((< arg3 arg1)
5         (< arg1 arg2))))

```

Задание 2.2 Функция how-alike

- реализация из лекции

```

1 (defun how-alike-cond (arg1 arg2)
2   (cond ((or (= arg1 arg2) (equal arg1 arg2)) 'the_same)
3         ((and (oddp arg1) (oddp arg2)) 'both_odd)
4         ((and (evenp arg1) (evenp arg2)) 'both_even)
5         (t 'difference)))

```

- с помощью специального оператора **IF**

```

1 (defun how-alike-if (arg1 arg2)
2   (if (or (= arg1 arg2) (equal arg1 arg2)) 'the_same
3       (if (and (oddp arg1) (oddp arg2)) 'both_odd
4           (if (and (evenp arg1) (evenp arg2)) 'both_even
5               'difference))))

```

- с помощью макросов **AND** и **OR**

```

1 (defun how-alike-andor (arg1 arg2)
2   (or (and (or (= arg1 arg2)
3               (equal arg1 arg2))
4           'the_same)
5       (and (and (oddp arg1)
6                 (oddp arg2))
7           'both_odd)
8       (and (and (evenp arg1)
9                 (evenp arg2))
10          'both_even)
11          'difference))

```

3 Вычислить результат выражения

Задание 3.1

```
(and 'fee 'fie 'foe) => FOE
```

Задание 3.2

```
(or 'fee 'fie 'foe) => FEE
```

Задание 3.3

```
(and (equal 'abc 'abc) 'yes) => YES
```

Задание 3.4

```
(or nil 'fie 'foe) => FIE
```

Задание 3.5

```
(and nil 'fie 'foe) => NIL
```

Задание 3.6

```
(or (equal 'abc 'abc) 'yes) => T
```

4 Написать предикат

Задание 4.1 Предикат принимает два числа и возвращает T, если первое число не меньше второго.

```
1 (defun geqp (arg1 arg2)
2   (and (numberp arg1)
3        (numberp arg2)
4        (>= arg1 arg2)))
```

5 Найти ошибку

Задание 5.1 Найти ошибочный предикат, Объяснить почему.

```
1 (defun pred1 (x)
2   (and (numberp x)
3        (plusp x)))
4
5 (defun pred2 (x)
6   (and (plusp x)
7        (numberp x)))
```

Предикат `pred2` является неправильным, так как данная реализация выдаст ошибку, если передать в качестве аргумента не число. Ошибка связана с тем, что в реализации `pred2`, до проверки на то, что аргумент является числом, выполняется функция предназначенная для чисел.

6 Заключение

В данной лабораторной работе было изучено применение специального оператора `IF`, а так же работа макросов `AND`, `OR` и `COND`. Был сделан вывод, что в ряде случаев макрос `COND` гораздо удобнее специального оператора `IF`. Так же были рассмотрены функции: `EVENP`, `SIGNUM`, `ODDP`, `EVENP`, `NUMBERP`, `PLUSP`