**Министерство науки и высшего образования Российской Федерации**
**Федеральное государственное бюджетное образовательное учреждение**
**высшего образования**
**«Московский государственный технический университет**
**имени Н.Э. Баумана**
**(национальный исследовательский университет)»**
**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

# Лабораторная работа № 6

## По дисциплине «Функциональное и логическое программирование»

**Студент: Тимонин А. С.**

**Группа ИУ7-62б**

**Преподаватель Толпинская Н. Б.**

Москва.
2020 г.

# Практическая часть

**Задание 9.**

Переписать функцию how-alike, приведенную в лекции и не пользующую COND, используя конструкция IF, AND/OR.

```
(defun how-alike (x y)
    (if (or ( = x y) (equal x y))
            'sameNumbers
            (if (and (oddp x) (oddp y))
                'oddpNumbers
                (if (and (evenp x) (evenp y))
                    'evenpNumbers
                    'differenceNumbers
                )
            )

    )
)
```

**Задание 10.**

Дано два списка: название стран (4 шт) и список столиц.

Создать а) список из двухэлементных списков

б) список точечных пар.

По созданным спискам: 1 - по столице найти страну 2 - по стране найти столицу (было дано дополнительно)

```
(setq lst1 '(Rus Ukr Blr Kaz))

(setq lst2 '(Moscow Kiev Minsk Astana))


; Создание списка содержащим список страна-столица
(defun createList(lst1 lst2)
    (list (list (car lst1)       (car lst2))
          (list (cadr lst1)     (cadr lst2))
          (list (caddr lst1)   (caddr lst2))
          (list (cadddr lst1) (cadddr lst2))
    )
)
```

```
; Создание точечных пар страна-столица
(defun createPairs(lst1 lst2)
    (list (cons (car lst1)       (car lst2))
          (cons (cadr lst1)     (cadr lst2))
          (cons (caddr lst1)   (caddr lst2))
          (cons (cadddr lst1) (cadddr lst2))
    )
)

defun check (pair key) (cond
((equal (car pair) key) (cdr pair))
((equal (cdr pair) key) (car pair)) ))


(defun find (table key)
(find-if #'(lambda (x) (not (= x Nil)))
(mapcar #'(lambda (pair) (check pair key) key) table) ))
```

**Из лекции how-alike.**

```
(defun how-alike (x y)
    (cond ((or ( = x y) (equal x y))
          (and (oddp x) (oddp y))
          (and (evenp x) (evenp y))
          (T 'difference))))
```