**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»

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

## По курсу: "Функциональное и Логическое программирование"

Тема Использование управляющих структур, работа со списками.

Группа ИУ7-63Б

Студент Сукочева А.

Преподаватель Толпинская Н.Б.

Преподаватель Строганов Ю. В.

Москва — 2021 г.

# Практическая часть

**Задание 1.** Пусть `(setf lst1 '(a b)) (setf lst2 '(c d))`. Каковы результаты вычисления следующих выражений?

```
(cons lst1 lst2)      ;; ((A B) C D)
(list lst1 lst2)      ;; ((A B) (C D))
(append lst1 lst2)    ;; (A B C D)
```

**Задание 2.** Каковы результаты вычисления следующих выражений?

```
(reverse ())              ;; Nil
```

```
(last ())                 ;; Nil
```

```
(reverse '(a))            ;; (a)
```

```
(last '(a))               ;; (a)
```

```
(reverse '((a b c)))      ;; ((A B C))
```

```
(last '((a b c)))          ;; ((A B C))
```

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

```
(defun f-last-rec (lst)
    (cond ((null (cdr lst)) (car lst))
    (T (f-last-rec (cdr lst)))) )
```

Беремен первый элемент от перевернутого списка.

```
(defun f-last (lst)
    (car (reverse lst)) )
```

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

```
(defun f1 (lst)
    (reverse (cdr (reverse lst))) )
```

```
(defun f1-rec (lst)
    (cond ((null (cdr lst)) ())
    (T (cons (car lst) (f1-rec (cdr lst))))) )
```

**Задание 5.** 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 получает 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.

```
(defvar name-first)
(defvar name-second)

(setf name-first "Alice")
(setf name-second "Pasha")

;; dice = ((1-6 1-6) 1/0)
(defvar dice-first)
(defvar dice-second)
(defvar tmp-dice)

(defun roll-one-dice ()
    (+ (random 6) 1 ) )

(defun roll-two-dice ()
    (list (roll-one-dice) (roll-one-dice)) )

(defun sum (dice)
    (+ (car dice) (cadr dice)) )

(defun is-win (dice)
    (cond ((= (sum dice) 7 ))
        ((= (sum dice) 11)) ) )

(defun repeat-roll (dice)
    (cond ((= (car dice) (cadr dice) 6))
        ((= (car dice) (cadr dice) 1))) )


(defun print-res (name dice)
    (format Nil "~%Win␣~a␣~a␣~%" name  (car dice) (sum (car dice))) )

(defun user-round (name)
    (setf tmp-dice (roll-two-dice))
    (format T "Player␣name:␣~a␣~a␣sum␣=␣~a␣~%" name tmp-dice (sum tmp-dice))
    (cond ((is-win tmp-dice) (list tmp-dice 1))
        ((repeat-roll tmp-dice) (user-round name))
        (T (list tmp-dice 0))) )


(defun play ()
    (setf dice-first (user-round name-first))
    (if (= (cadr dice-first) 1) (print-res name-first dice-first)
    (and (setf dice-second (user-round name-second))
    (cond ((= (cadr dice-second) 1) (print-res name-second dice-second))
        ((> (sum (car dice-first)) (sum (car dice-second))) (print-res
           name-first dice-first))
        ((< (sum (car dice-first)) (sum (car dice-second))) (print-res
           name-second dice-second))
        ((format Nil "Draw"))))))
```

# Теоретическая часть

## Структуроразрушающие и не разрушающие структуру списка функции

Функции для работы со списками делятся на две группы:

1. Не разрушающие структуру. Если сохраняется возможность работать с исходными списками, значит функции не разрушают структуру. (Пример: append, reverse, length, subst ...)

2. Разрушающие структуру. Если после использования какой-то стандартной функции после ее работы теряется возможность работы с теми списками, которые изначально были, значит их структура разрушилась. Чаще всего такие функции начинаются в буквы 'n (Пример: nconc, nreverse, nsubst ...)

# Отличие в работе функции cons, list, append и в их результате