|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 №7**

**по курсу “Функциональное и логическое программирование”**

**по теме “Использование функционалов”**

|  |  |
| --- | --- |
| Студент: | Уласик Е.А. |
| Группа: | ИУ7-61 |
| Преподаватель: | Толпинская Н.Б. |

*2020 г.*

1. **Практическая часть**

Задание 1. Чем принципиально отличаются функции cons, list, append? Пусть (setf lst ‘(a b)) (setf ls2 ‘(c d)) Каковы результаты вычисления следующих выражений?

1. (cons ls1 ls2)

Результат: ((A B) C D)

1. (list ls1 ls2)

Результат: ((A B) (C D))

1. (append ls1 ls2)

Результат (A B C D)

Задание 2. Каковы результаты вычисления следующих выражений?

1. (reverse ())

Результат: Nil

1. (last ())

Результат: Nil

1. (reverse ‘(a))

Результат: (A)

1. (last ‘(a))

Результат: (A)

1. (reverse ‘((a b c)))

Результат: ((a b c))

1. (last ‘((a b c)))

Результат: ((a b c))

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

1. (defun last\_1(lst)(car (last lst)))
2. (defun last\_2(lst)(car (reverse lst)))

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

1. (defun lst\_1(lst)(reverse (cdr (reverse lst))))
2. (defun my\_last(lst)

(cond

((>= (length lst) 2)

(remove (car (last lst)) lst :start (- (length lst) 2) :end (- (length lst) 1)))

(T nil)

)

)

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. Результат игры и значения выпавших костей выводить на экран с помощью функции print.

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

(defun play-sum (res)

(+ (first res) (second res))

)

(defun is-win (res)

(or (equal 7 (play-sum res))

(equal 11 (play-sum res))

)

)

(defun is-lucky (res)

(or (equal res '(1 1))

(equal res '(6 6))

)

)

(defun play ()

(setq player1 (roll-dice))

(format T " First player ~A" player1)

(cond ((equal T (is-win player1))

(print " First player win!"))

(T (cond ((equal T (is-lucky player1))

(play))

(T (last

(list

(setq player2 (roll-dice))

(format T " Second player ~A" player2)

(cond ((= (play-sum player1) (play-sum player2))

(print "Tie!"))

(T (cond ((> (play-sum player1) (play-sum player2))

(print "First player win!"))

(T (print "Second player win!"))

))

)

)

))

)))

)