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 дисциплине «Функциональное и логическое программирование»

Тема Использование управляющих структур, работа со списками

Студент Романов А.В.

Группа ИУ7-63Б

Оценка (баллы) _____

Преподаватель Толпинская Н.Б., Строганов Ю. В.

Москва — 2021 г.

Задание 1

Постановка задачи

Чем принципиально отличаются функции `cons`, `list`, `append`?
Пусть `(setf lst1 '(a b)) (setf lst2 '(c d))`
Каковы результаты следующих выражений?

```
(cons lst1 lst2)
(list lst1 lst2)
(append lst1 lst2)
```

Решение

1. `((A B) C D)`
2. `((A B) (C D))`
3. `(A B C D)`

Задание №2

Постановка задачи

Каковы результаты вычисления следующих выражений?

```
(reverse ())
(last ())
(reverse '(a))
(last '(a))
(reverse '((a b c)))
(last '((a b c)))
```

Решение

1. `Nil`
2. `Nil`
3. `(a)`
4. `(a)`
5. `((a b c))`
6. `((a b c))`

Задание №3

Постановка задачи

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

Решение

Листинг 1: Решение задания №3

```
(defun last (lst)
  (if (cdr lst)
    (last-elem (cdr lst))
    (car lst))
)

(defun last-reduce (lst)
  (reduce #'(lambda (acc current) current) lst)
)
```

Задание №4

Постановка задачи

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

Решение

Листинг 2: Решение задания №3

```
(defun init-internal (lst acc)
  (if (cdr lst)
    (init-internal (cdr lst) (append acc (cons (car lst) Nil)))
    acc))

(defun init (lst)
  (init-internal lst ()))
```

Задание №5

Постановка задачи

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`.

Решение

Листинг 3: Решение задания №5

```
(defun random-cube-value ()
  (list (random 7) (random 7)))

(defun dices-sum (pair)
  (+ (car pair) (car (cdr pair))))

(defun check-absolute-win (sum)
  (or (= sum 7) (= sum 11)))

(defun check-rerun (pair)
  (let* ((fst (car pair))
         (snd (car (cdr pair))))
  (or (= fst snd 1) (= fst snd 6)))
)

(defun dices ()
  (let* ((fst-dices-pair (random-cube-value))
       (fst-dices-sum (dices-sum fst-dices-pair)))
         (format T "Player one dices: ~a.~%" fst-dices-pair)
         (cond ((check-absolute-win fst-dices-sum)
           (format T "Player one win!~%"))
         ((check-rerun fst-dices-pair)
           (format T "Rerun!~%") (dices))
         (t (let* ((snd-dices-pair (random-cube-value))
                   (snd-dices-sum (dices-sum snd-dices-pair)))
                     (format T "Player two dices: ~a.~%" snd-dices-pair)
                   (cond ((check-absolute-win snd-dices-sum)
                     (format T "Player two win!~%"))
                   ((> fst-dices-sum snd-dices-sum)
                     (format T "Player one win!~%"))
                   (t
                     (format T "Player two win!~%")))))))
)
```

Контрольные вопросы

Вопрос 1. Структуроразрушающие и не разрушающие структуру списка функции

Ответ. Не разрушающие структуру списка функции. Данные функции не меняют сам объект-аргумент, а создают копию. К таким функциям относятся: `append, reverse, last, nth, nthcdr, length, remove, subst` и прочие.

Структуроразрушающие функции. Данные функции меняют сам объект-аргумент, невозможно вернуться к исходному списку. Чаще всего такие функции начинаются с префикса `n-`. К такми функция относятся: `nreverse, nconc, nsubst` и прочие.

Вопрос 2. Отличие в работе функций `cons`, `list`, `append` и в их результате.

Ответ. Функция `cons` — чисто математическая, конструирует списковую ячейку, которая может вовсе и не быть списком. Является списком только в том случае, если вторым аргументом передан список.

Функция `list` — форма, принимает произвольное количество аргументов и конструирует из них список. Результат — всегда список. При нуле аргументов возвращает пустой список.

Функция `append` — форма, принимает на вход произвольное количество аргументов и для всех аргументов, кроме последнего, создает копию, ссылая при этом последний элемент каждого списка-аргумента на первый элемент следующего по порядку списка-аргумента. Копирование для последнего не делается в целях эффективности.