# Лекция 5 (16.03.2022)

О чём думать при написании рекурсии:

1. Остановить рекурсию

Хвостовая рекурсия. (не остаётся не вычисленных результатов)

Один вариант выхода и одно условие рекурсии:

**(defun** fun**(**x**)**

**(cond** **(**end\_test end\_value**)**

**(**T **(**fun changed\_x**))))**

Хочу написать функцию, которая ищет первый атом на всех уровнях:

**(**a b c**)**

**(((**a b**))** c**)**

**(defun** first-a**(**lst**)**

**(cond** **((atom** lst**)** lst**)**

**(**T **(**first-a **(car** lst**)))))**

Найти первое нечетное:  
**((**a 2**)** **(**3 c**)** d**)**

**(defun** find-first-odd**(**lst**)**

**(cond** **((null** lst**)** NIL**)**

**((oddp** **(**first-a lst**))** **(**first-a lst**))**

**(**T **(**find-first-odd **(cdr** lst**)))))**

Функция my-nth:

**(defun** my-nth**(**lst n**)**

**(cond** **((null** lst**)** **nil)**

**((=** n 0**)** **(car** lst**))**

**(**T **(**my-nth **(cdr** lst**)** **(-** n 1**)))))**

Дополняемая рекурсия:

**(defun** funс **(**x**)**

**(cond** **(**end\_test end\_value**)**

**(**T **(**add\_function add\_value

**(**func change\_x**)))))**

Пример: длина на верхнем уровне

**(defun** my-length**(**lst**)**

**(cond** **((null** lst**)** 0**)**

**(**T **(+** 1 **(**my-length **(cdr** lst**))))))**

Не очень эффективный метод работы (append):

**(defun** into-one-level**(**lst**)**

**(cond** **((null** lst**)** **nil)**

**((atom** lst**)** **(cons** lst NIL**))**

**(**T **(append** **(**into-one-level **(car** lst**))**

**(**into-one-level **(cdr** lst**))))))**

**(setf** lst **'((**a**)** **(**b c**)** **(**d e f**)))**

**(**into-one-level lst**)**

Сортировка:

**(defun** insert-help**(**x lst**)**

**(cond** **((null** lst**)** **(list** x**))**

**((<=** x **(car** lst**))** **(cons** x lst**))**

**(**T **(cons** **(car** lst**)** **(**insert-help x **(cdr** lst**))))))**

**(defun** sort-help**(**lst1 lst2**)**

**(cond** **((null** lst1**)** lst2**)**

**(**T **(**sort-help **(cdr** lst1**)** **(**insert-help **(car** lst1**)** lst2**)))))**

**(defun** sort-ins**(**lst**)**

**(**sort-help lst **()))**

**(setf** lst **'(**1 4 3 2 5**))**

**(**sort-ins lst**)**