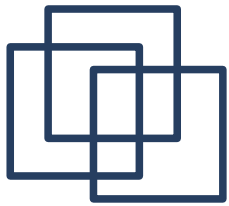




Кафедральный практикум 5 семестр

Часть 3. Функции высшего порядка (списочные комбинаторы)

<http://sp.cmc.msu.ru/~kornyxin/fp/slides/part3-2.pdf>



План

Часть 1. Введение.

Часть 2. Язык программирования Scheme.

Часть 3. Функции высшего порядка.
«Векторное» мышление.

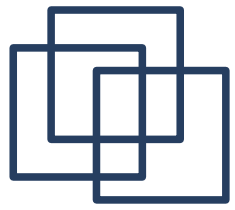
- 1) continuation-passing style
- 2) **списочные комбинаторы**
- 3) замыкания (программирование объектами-функциями)
- 4) свертки над структурами данных
- 5) программы, управляющие вычислениями

Часть 4. Теоретический фундамент ФП.



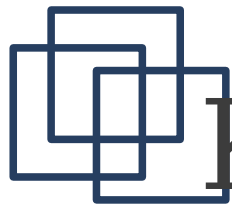
FP for dummies

«*FP is pure programming with lambda*»



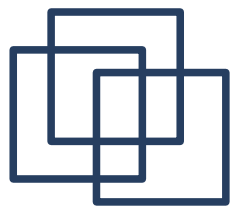
ФП в других языках

- Демо



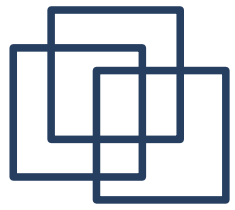
Как надо думать в ФП

- 1) комбинировать выражения
- 2) использовать лямбду для вставки кода в выражения



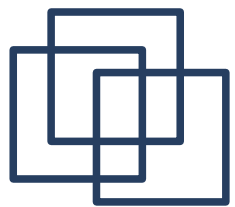
Полезные функции

- High-order functions:
 - map
 - filter
 - foldl
 - foldr
 - unfold (srfi/1)
 - unfold-right (srfi/1)
- First-order functions:
 - reverse
 - flatten
 - sort
 - member
 - remove
 - remove*
 - remove-duplicates
- В ТОМ ЧИСЛЕ И В МНОГОАРНОМ ВИДЕ

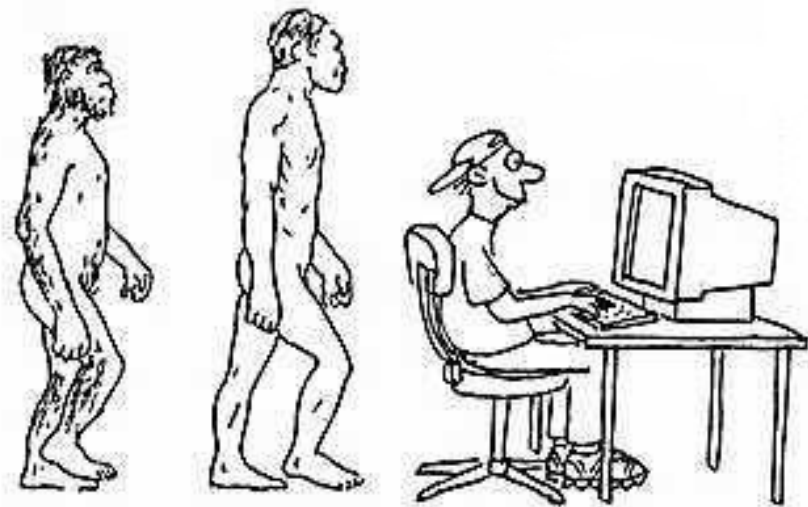
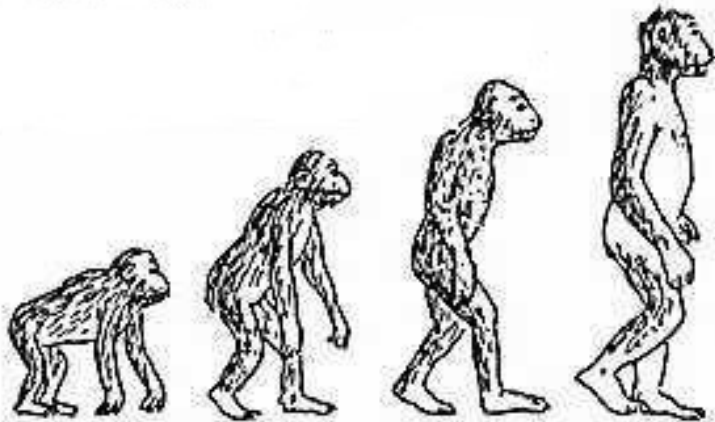


High-order functions

- (map calc-element(x) список-из-иксов)
- (filter calc-condition(x) список-из-иксов)
- (foldl calc-newres(x, res) res-for-empty с-из-х)
- (foldr calc-newres(x, res) res-for-empty с-из-х)
- (unfold stop-condition(i) calc-element(i) calc-next(i))
 - не забыть (require srfi/1)



Генетическое программирование

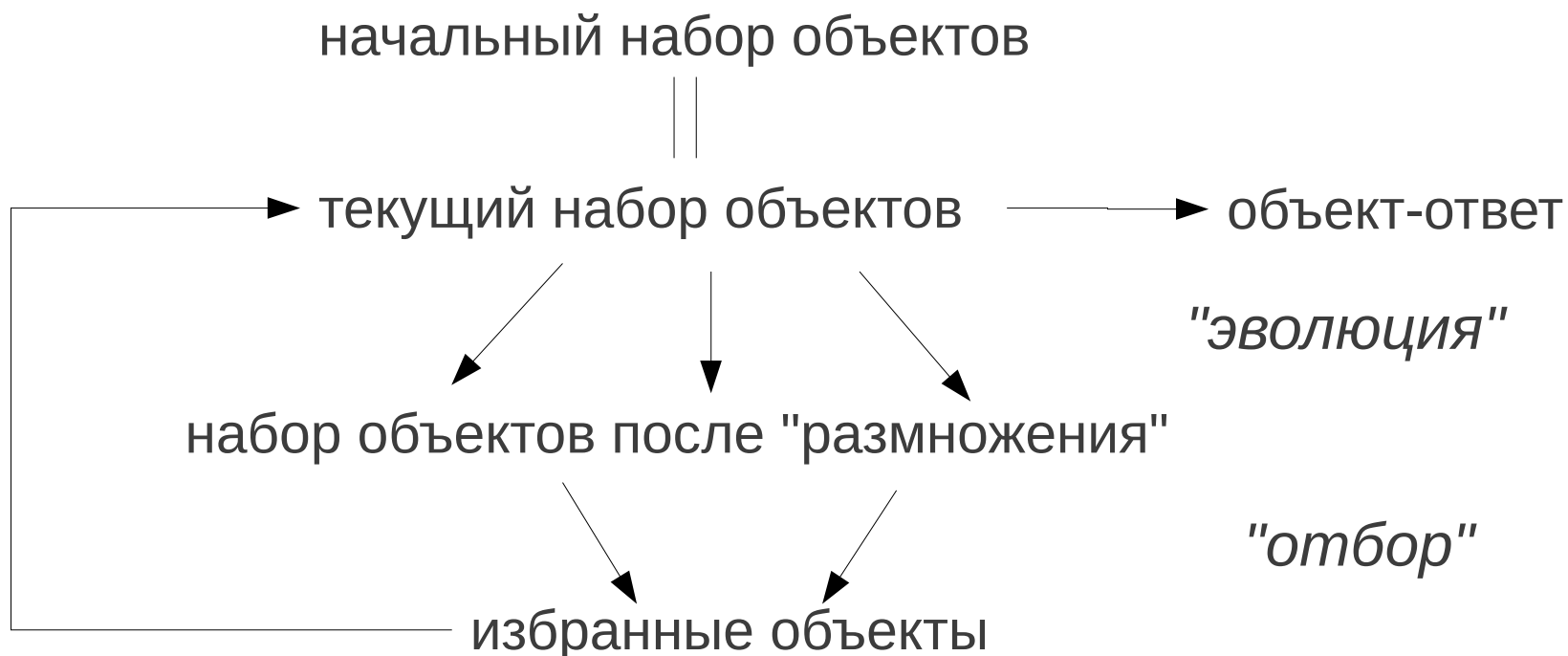


- задачи оптимизации
- поиск "приближенного решения"
- комбинация перебора и жадных алгоритмов



Схема методики

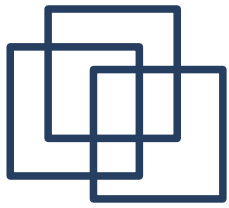
- состояние искомого объекта — кортеж характеристик (кортеж — "хромосома")
- схема эволюционного поиска решения:





ЭВОЛЮЦИЯ И ОТБОР

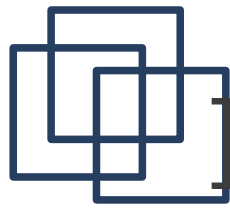
- Мутация — сменить значение одной из характеристик объекта на случайное
- Скрещивание — обменять два объекта их некоторыми характеристиками
- Функция отбора — дать каждому объекту *оценку*, отобрать некоторые из объектов на основе оценки



Пример

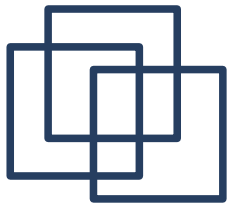
- Есть фургон для перевозки призов для АСМ, вместимость фургона ограничена
- Есть спонсоры, а у них призы (на складе количество призов неограничено):
 - книжка — вес, ценность
 - футболка — вес, ценность
 - кружка — вес, ценность
- Как заполнить фургон, максимизировав суммарную ценность?





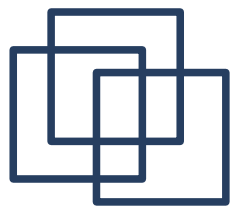
Параметры эволюции

- вероятность мутации
- размер "популяции"
- представительность "популяции"
 - большая — снижает скорость сходимости
 - малая — можно увязнуть в локальном экстремуме



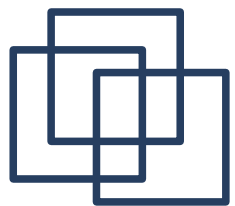
Что надо делать

- разработать решение задачи методом генетического программирования (понять, что такое "хромосома", "функция отбора",...)
- закодить решение на Scheme
- прикрутить визуализацию процесса нахождения решения
- экспериментально подобрать параметры эволюционного процесса решения, дающие лучшую сходимость (из-за этого особенно удобно сразу писать в high-order functions!)



Где еще почитать

- книжка "Прототипирование программ на языке Scheme"
- есть в библиотеке факультета
- есть на сайте кафедры:
<http://sp.cmc.msu.ru/courses/prak3/scheme.pdf>



Домашнее задание

- Делаем практическое задание по генетическому программированию
- Варианты будут присланы через ejudge
- Решение посылаем в ejudge
- Дедлайн — 14 ноября
- Написать на kornevgen@ctc.msu.ru, что интересно запрограммировать еще (по какому спецкурсу)