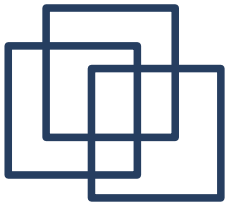
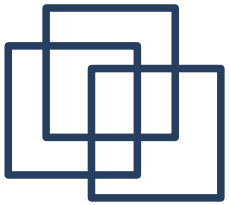


Кафедральный практикум 5 семестр

Часть 2. Язык Scheme (быстрый старт)

http://sp.cmc.msu.ru/~kornykhin/fp/slides/part2-1.pdf



План

Часть 1. Введение.

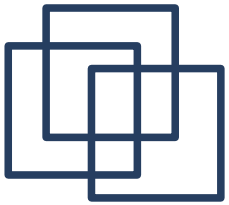
1) Организационные вопросы
2) Функциональный стиль программирования

Часть 2. Язык программирования Scheme.

1) **Быстрый старт**
2) Более внимательный взгляд на Scheme

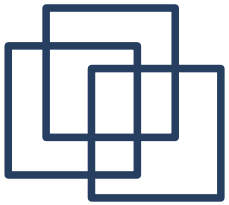
Часть 3. Функции высшего порядка. «Векторное» мышление.

Часть 4. Теоретический фундамент ФП.



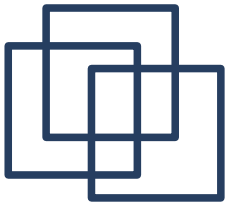
Что где скачать

- http://www.racket-lang.org/download/
- {d} = http://download.racket-lang.org/installers/5.1.3/racket/
- для Windows:
  - x86: {d}/racket-5.1.3-bin-i386-win32.exe
  - x64: {d}/racket/racket-5.1.3-bin-x86_64-win32.exe
- для Debian/Ubuntu:
  - x86: {d}/racket-5.1.3-bin-i386-linux-ubuntu-jaunty.sh
  - x64: {d}/racket/racket-5.1.3-bin-x86_64-linux-debian-squeeze.sh



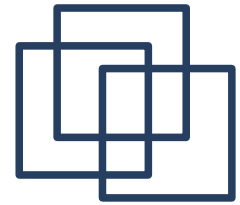
Что запускать

- Чтобы пользоваться всей IDE (и справкой):
  - drracket (DrRacket)
  - drscheme
- Только интерпретатор в интерактивном режиме:
  - mzscheme
- Только интерпретатор без интерактивного режима:
  - mzscheme --script FILE

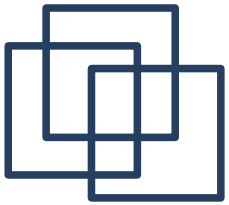


Задача

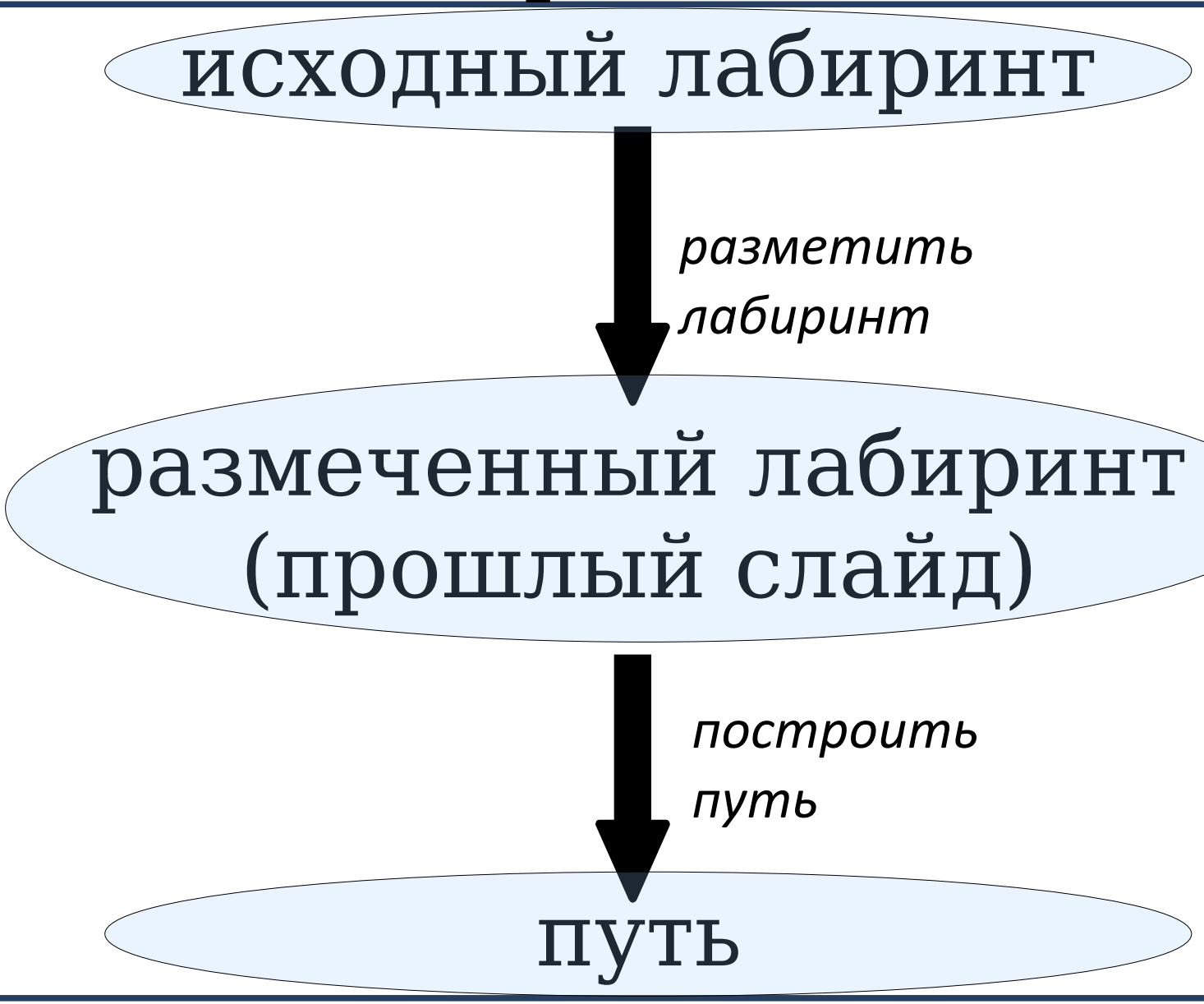
- Есть квадратное поле, в котором выделены
  - клетка-старт
  - клетка-финиш
  - клетки, на которых нельзя находиться ("запрещенные клетки").
- Построить путь из клетки-старт в клетку-финиш, не посещая запрещенные клетки, или сообщить, что такого пути нет.

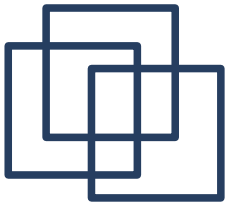


Идея решения



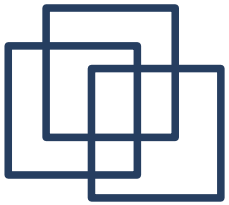
Идея решения





Уравнения

- Функциональная парадигма — определение всё новых данных через уже определенные
- Уравнения — способ задания отношений между «старыми» данными и новыми.
- Выписываем уравнения для «лабиринта»... (см. на доску)



Представление данных

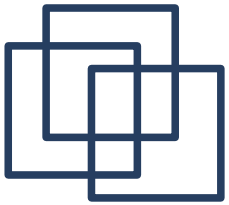
- Как представить "размеченный лабиринт"?

  1) (v11 v12 v13 … v1N v21 v22 …)
     квадрат целиком, вытянутый в список

  2) ((1 2 v12) (4 2 v42) …)
     разреженное представление

  3) ((v11 v12 v13 …) (v21 v22 …) …)
     список списков

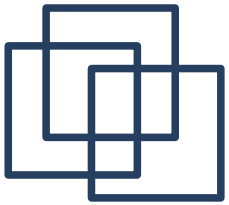
  4) (((1 2) (4 2)) ((3 5)) …)
     список списков (внутренний список — места с одинаковой меткой: у первого списка метка равна 1, у следующего — 2 и т.д.)



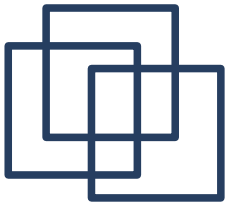
Attention

- Scheme — лишь одна из реализаций некоторого набора идей. Наша цель — ухватить и усвоить эти идеи, а не "зациклиться на изучении языка".

- Эти же идеи реализованы в других языках — а умение пользоваться этими идеями от языка не зависит и его надо развивать!

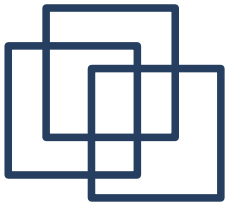


Demo



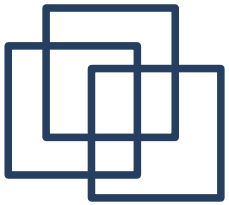
Идентификаторы

- Идентификатор — это практически **любая строчка** без пробелов и скобок!



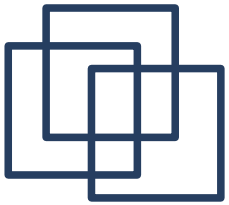
Аналоги fplists

- cons --- create
- car --- first
- cadr --- second
- cdr --- tail
- ( ) --- empty
- null? --- is_empty
- length --- len



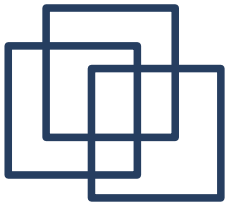
Простой ввод/вывод

- (display выражение)
- (newline)
- (read)



"Expression only"

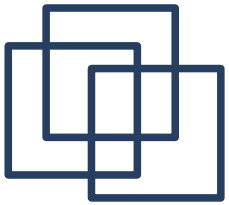
- всюду списочный синтаксис



Домашнее задание

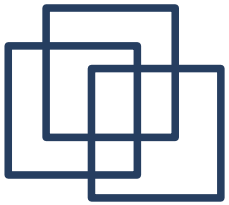
- ejudge : http://earth.ispras.ru

- Берем логин, нажимаем "Забыли пароль" — получаем письмо с паролем. Входим!
- Решения оформлять в виде текстовых файлов и посылать в ejudge
- Есть срок окончания подачи решений! Смотрите внимательно на страницу контеста
- Начисление баллов/штрафов по регламенту, а именно . . .



Баллы за решения

- Посылка полного решения с первого раза даёт полный балл за задачу (50-150 баллов)
- Последующие попытки добавляют по -2 штрафных балла к результату
- Неполное решение получает по 2-3 балла за каждый пройденный тест (минус штраф из предыдущего пункта)



Ваши решения

- В решении запрещается использовать
  - set! и его аналоги
  - векторы
- Исполнение решения на тестах будет ограничено по времени (1-5 секунд) и памяти (64-512 МБ)