

Пример варианта письменной контрольной работы

Указание: для ответа на вопросы сделайте отметки в ☐ рядом с правильными вариантами ответов (верными могут быть один или более вариантов).

1	2	3	Σ

I. Какое подвыражение вычисляется первым при вычислении выражения $(f\ a\ b)$:

- a) ☐ a; c) ☐ f;
b) ☐ b; d) ☐ неизвестно (определяется интерпретатором).

II. Укажите, какие из перечисленных выражений являются литералами:

- a) ☐ #t; d) ☐ (/ 2 3); g) ☐ abc;
b) ☐ "abc"; e) ☐ (quote (/ 2 3)); h) ☐ 'abc
c) ☐ 2/3; f) ☐ '(/ 2 3); i) ☐ (a b c).

III. Укажите, что не так в определении (define cube (x) (* x x x)):

- a) ☐ вместо define нужно писать def; d) ☐ имя cube переопределять нельзя;
b) ☐ всё верно, ошибок нет; e) ☐ имя cube следует писать в скобках
c) ☐ отсутствует оператор возврата; вместе с параметром функции.

IV. Положим, что функция square перемножает свой параметр сам на себя. Укажите, сколько умножений потребуется выполнить при аппликативном порядке вычислений, чтобы узнать значение выражения (square (square 3)):

- a) ☐ 1; b) ☐ 2; c) ☐ 3; d) ☐ 4.

V. Укажите всё, что справедливо для специальной формы cond:

- a) ☐ в cond всегда должна быть else альтернатива;
b) ☐ в cond всегда должно быть две или более альтернатив;
c) ☐ в cond предикат из очередной альтернативы вычисляется, только если все предыдущие были ложны;
d) ☐ если в cond истинны предикаты в нескольких альтернативах, то выполняемая альтернатива выбирается из них случайно.

VI. Укажите всё, что справедливо для рекурсивных функций:

- a) ☐ рекурсивная функция может породить итеративный процесс;
b) ☐ рекурсивная функция может породить рекурсивный процесс;
c) ☐ рекурсивная функция порождает только рекурсивные процессы.

VII. Итеративное вычисление чисел Фибоначчи, реализованное на языке Scheme, требует:

- a) ☐ линейной памяти; e) ☐ линейного времени счёта;
b) ☐ памяти постоянного размера; f) ☐ константного времени счёта;
c) ☐ квадратичной памяти; g) ☐ квадратичного времени счёта;
d) ☐ экспоненциальной памяти; h) ☐ экспоненциального времени счёта.

VIII. Какой процесс порождает определение функции (define f (lambda (x) (f (+ 1 x)))):

- a) ☐ итеративный;
b) ☐ рекурсивный;
c) ☐ никакой, так как в определении синтаксическая ошибка;
d) ☐ зацикливающийся.

IX. Что сделать, чтобы программным способом убедиться, что and в Scheme является специальной формой:

- a) ☐ использовать and внутри тела некоторой функции;
b) ☐ вычислить and, указав среди аргументов вызов некоторой функции;
c) ☐ попытаться переопределить and;
d) ☐ попытаться использовать and внутри and;
e) ☐ никак, в этом нельзя убедиться программным способом.

X. Если упростить выражение (not (or (not (and a b)) (or c (not d)))) то получится:

- a) ☐ (and a b (not c) d);
b) ☐ (or (or (not a) (not b)) (or c (not d)));
c) ☐ (and (or a b) (or c d));
d) ☐ (or a b c d).

Пример варианта письменной контрольной работы

Указание: Эффективность решения задач по времени счёта и памяти учитывается и в случае правильного решения может принести дополнительные баллы.

2.I. (2 балла) функции (`abc lst`) в качестве аргумента подаёт список из чисел. Результатом функции является список из тех же элементов, отсортированный по неубыванию. Реализуйте функцию методом простого выбора.

2.II. (2 балла) функции (`tree-list t`) в качестве аргумента подается дерево, представленное в виде вложенных списков. Корень дерева и промежуточные вершины не содержат данных, листьями являются атомы, указанные в списке, вложенные списки обозначают поддеревья. Функция возвращает список листовых вершин, расположенных справа налево. Пример

`(tree-list '(a b ((c) d))) ==> '(d c b a)`

2.III. (2 балла) Квадродеревом называется следующий способ представления растровых черно-белых изображений:

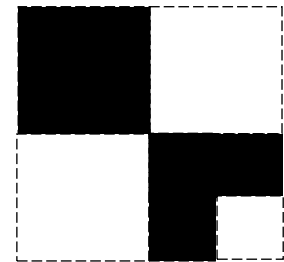
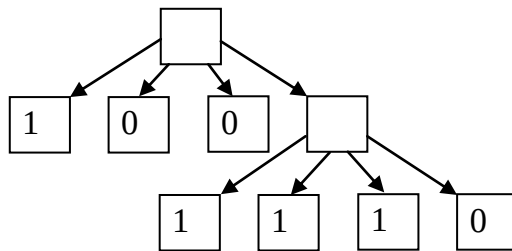
1) Если изображение целиком белое, то оно представляется квадродеревом из одной «белой» вершины. Линейная запись такого квадродерева: 0. [ноль]

2) Если изображение целиком чёрное, то оно представляется квадродеревом из одной «чёрной» вершины. Линейная запись такого квадродерева: 1. [единица]

3) Если на изображении есть и чёрные и белые участки, то оно делится на 4 равные части (верхнюю левую, верхнюю правую, нижнюю левую, нижнюю правую) и представляется квадродеревом, состоящим из корневой вершины и четырёх поддеревьев, которые описывают части изображения. Пусть линейные записи поддеревьев таковы: <верхлевдерево>, <верхправдерево>, <нижлевдерево>, <нижправдерево>; тогда запись всего дерева будет списком из четырёх элементов: (<верхлевдерево> <верхправдерево> <нижлевдерево> <нижправдерево>)

Пример: линейная запись квадродерева: '(1 0 0 (1 1 1 0))

квадродерево в виде графа: описываемое изображение:



Составьте функцию (`black-square t s`), которая находит суммарную площадь чёрных участков изображения, считая, что площадь всего участка, описываемого квадродеревом `t`, равна `s`. Результат для дерева из примера равен 7, если площадь квадрата равна 16.

2.IV. (3 балла) функция (`flatten lst cc`) должна быть написана в стиле передачи остаточных вычислений. В качестве первого аргумента функции подаётся произвольный список. Результатом является список, получающийся перечислением всех атомов, содержащихся в исходном списке и всех вложенных списках, с сохранением порядка.

Пример: `(flatten '(((a b) (a c)) (a b b)) (lambda (x) x)) ==> '(a b a c a b b)`

2.V. (2 балла) с использованием функций метапрограммирования реализуйте функцию (`increment-longest lst`), которая возвращает список, получаемый из исходного списка списков `lst` прибавлением 1 к всем элементам списков, являющихся элементами списка `lst` и имеющих наибольшую длину. Функция `increment-longest` не должна быть рекурсивной. Не допускается определять дополнительные функции. Не используйте встроенные функции, кроме функций метапрограммирования (`filter`, `foldl`, `foldr`, `map`), базовых функций работы со списками (`car`, `cdr`, `cons`), арифметических функций, функций сравнения.

Пример: `(increment-longest '((1) (2 3) (4 5) (6))) ==> '((1) (3 4) (5 6) (6))`

2.VI. (2 балла) функция (`scal v1 v2`) принимает в качестве аргументов два списка чисел одинаковой длины и вычисляет их скалярное произведение. Предложите две реализации функции, порождающие рекурсивный и итеративный процесс.

2.VII. (1 балл) функция (`super f1 f2`) возвращает в качестве результата функцию, являющуюся суперпозицией `f1` и `f2`, так что `((super f1 f2) x) = (f2 (f1 x))`. Реализуйте функцию `super`.

Пример варианта письменной контрольной работы

Текстовые вопросы:

3.I. (2 балла) Расскажите об аппликативном порядке вычислений. В чём он заключается? Каковы его преимущества и недостатки по сравнению с нормальным порядком? Проиллюстрируйте свой ответ примерами.

3.II. (2 балла) Нарисуйте стрелочную диаграмму для выражения
`(list (cons 1 2) (list (cons 3 4)) 5)`

3.III. (2 балла) как работает специальная форма `let`? Приведите пример её осмысленного использования.