

Вариант 1.9.

Все консольные приложения Ruby следует реализовывать в виде трех отдельных файлов:

1. основная программа;
2. программа для взаимодействия с пользователем через консоль;
3. программа для автоматического тестирования на основе MiniTest::Unit.
Везде, где это возможно, данные для проверки должны формироваться автоматически по правилам, указанным в задании.

Все тексты программ должны быть проверены на соответствие стилю программирования Ruby при помощи *rubocop* и *reek*.

ЛР 5

Часть 1

Вычислить: $y = \lg \frac{x^2}{x-2} \cdot e^x$.

Часть 2

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в следующем. Изменить порядок букв на обратный в каждом слове, стоящем на нечетной позиции (нумерация слов с нуля). Вывести на печать исходную и скорректированную последовательности строк.

Автоматический тест программы обязательно должен генерировать случайные строки в соответствии с правилами, перечисленными в задании.

ЛР 6

Часть 1

Решить задачу, организовав итерационный цикл. Вычислять сумму бесконечно убывающей геометрической прогрессии: $1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots$ до тех пор, пока очередной член ряда не будет меньше $\xi = 10^{-4}, 10^{-5}$

Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator.

Часть 3

Составить метод `intprg` вычисления определенного интеграла по формуле прямоугольников: $S = \frac{b-a}{n} \sum_{i=1}^n f(x_i)$, где n – количество отрезков разбиения. В основной программе использовать метод `intprg` для вычисления интегралов: $\int_0^1 \frac{e^x}{x+1} dx$ и $\int_0^2 x(x-1) dx$.

Реализовать вызов метода двумя способами: в виде передаваемого `lambda`-выражения и в виде блока.

ЛР 7

Часть 1

Организовать программным способом символьные файлы **F** и **G**. Определить совпадают ли компоненты этих файлов. Если нет, то получить номер первой компоненты, с которой начинаются различия. В случае, когда один из файлов имеет N компонент $N \geq 0$ и повторяет начало другого (более длинного) файла, ответом должно быть число $N + 1$.

Автоматический тест программы обязательно должен проверять работу с файлами.

Часть 2

Разработать и реализовать иерархию классов для описанных объектов предметной области, используя механизмы наследования. Проверить ее на тестовом примере, с демонстрацией всех возможностей разработанных классов на конкретных данных.

Объект — точка на плоскости, заданная координатами, умеющая выводить их на экран и возвращать в ответ на запрос.

Объект — центр и радиус окружности, умеющий выводить их на экран координаты центра и радиус.

В тестирующей программе обеспечить автоматическую проверку того, что созданные объекты действительно соответствуют заданной иерархии классов.

ЛР 8. Ruby on Rails

Разработать веб-приложение, имеющее HTML-страницу с формой ввода данных и HTML-страницу для представления результатов. Результат расчёта должен быть представлен в форме таблицы, оформленной с помощью элемента table или отдельными ячейками div и имеющей не менее двух колонок. Если по условию задания результат может быть представлен только в виде одной строки таблицы, необходимо реализовать вывод промежуточных результатов расчёта в качестве дополнительных строк. В этом случае первой колонкой таблицы будет порядковый номер итерации.

Под вводом с клавиатуры в тексте заданий следует понимать ввод в поле ввода данных формы на HTML-странице.

Текст задания:

Метод Ньютона – Рафсона основывается на утверждении, что квадратный корень числа A можно найти с помощью рекуррентной формулы:

$$x_{i+1} = \frac{1}{2} \left(x_i + \frac{A}{x_i} \right), \text{ где } A — \text{положительное число, } x_i — \text{текущее при-}$$

ближение квадратного корня, x_{i+1} — очередное приближенное значение квадратного корня из числа A . Написать программу, определяющую квадратный корень заданных чисел, используя приведенную формулу. Пре-

кращение процесса вычисления предусмотреть при выполнении условия

$$\frac{x_{i+1}^2 - A}{A} < 0.001.$$

Вывести промежуточные итерации и полученный результат.