

Вариант 2.31.

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: $y = \frac{\sin(x + \pi \cdot z)}{\cos(x/r)}$.

Часть 2

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их обработку. Обработка заключается в следующем. Для каждой строки посчитать количество присутствующих во всех ее словах символов. Вывести строки в порядке уменьшения суммарного количества всех найденных символов в строке. Вывод организовать в виде «строка - символ 1: количество, ..., символ N : количество».

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

ЛР 6

Часть 1

Решить задачу, организовав итерационный цикл. Вычислить значение непрерывной дроби точностью $\xi = 10^{-3}, 10^{-4}$: $q = 1 + \frac{1}{1 + \frac{1}{1 + \dots}}$ Точное значение - 1,618033989....

Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator.

Часть 3

Составить метод `intg` вычисления определенного интеграла по формуле прямоугольников: $S = \frac{b-a}{n} \sum_{i=1}^n f(x_i)$, где n – количество отрезков разбиения, $[a, b]$ – отрезок интегрирования. В основной программе использовать метод `intg` для вычисления интегралов: $\int_{0,1}^1 \frac{\ln(x)}{x} dx$ и $\int_0^2 \sin x \cos x dx$

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

ЛР 7

Часть 1

Дан текстовый файл **F**. Переписать строки файла **F** в файл **G**, вставляя в начало каждой строки по одному пробелу. Порядок строк должен быть сохранен.

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 символ, умеющий выводить на экран свое значение и отвечать на запрос о значении своего символьного поля и числовом коде символа.

Объект запись, состоящая из полей: символ и целое число — количество повторений символа при выводе. Объект умеет выводить на экран строку из указанного количества символов, возвращать по запросу содержимое своих полей и код символа.

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:

Число из n цифр является числом Армстронга, если сумма цифр этого числа, возведенных в степень n равна самому этому числу. Например: $153 = 1^3 + 5^3 + 3^3$. Написать программу, определяющую все числа Армстронга, состоящие из n цифр (n вводится с клавиатуры). Вывести на печать сами числа их количество.