

Вариант 2.5.

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: $y = (x^2 + 2e^x) \cdot \cos(6\frac{z}{x} - 5)$.

Часть 2

Дана последовательность строк. Строки состоят из слов, разделенных пробелом. Каждая строка содержит существительные единственного числа, оканчивающиеся на «онок» и «енок». Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в преобразовании существительных в единственном числе, во множественное. Для преобразования считать, что множественное число этих существительных образуется на другой основе по образцу:

Цыпленок – цыплята, мышенок – мышата. При этом, «а» перед «т» пишется, если корень оканчивается на шипящую, а в остальных случаях перед «т» будет буква «я».

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. Вычислить значение $\sin 3$ с точностью $\xi = 10^{-3}, 10^{-4}$, используя разложение в ряд:

$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$ Определить, как изменяется число итераций при изменении точности.

Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator.

Часть 3

Составить метод `peibr` проверки принадлежности точки плоскости с координатами (x, y) данной кривой $y = f(x)$. В основной программе использовать метод `peibr` для проверки принадлежности десяти различных точек кривым $y = \cos(x)$ и $y = \sin(x^2)$.

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

ЛР 7

Часть 1

Сформировать программным путем файл **F**, компоненты которого являются целыми числами. Получить файл **G**, образованный из файла **F** исключением повторных вхождений одного и того же числа.

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 — Треугольник, заданный точками на плоскости. Объект умеет выводить на экран значение своих полей и отвечать на запрос об этих значениях и вычислять площадь фигуры.

Объект — Треугольная призма. Объект умеет выводить на экран содержимое своих полей, возвращать по запросу их значения и площадь развертки.

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:

Написать программу, определяющую, сколько из чисел в диапазоне от 0 до n (n вводится с клавиатуры), сами являясь палиндромами, при возведении в квадрат, образуют палиндромы. Например: $22^2 \rightarrow 484$. Вывести на печать числа, квадраты и их количество.