

Вариант 2.15.

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: $a = x(\cos(z) + e^{-(x+3)})$.

Часть 2

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в следующем: если среди слов строки обнаруживаются слова, заключенные в символ `#`, заменить их словами, введенными с клавиатуры. 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 с точностью $\xi = 10^{-4}, 10^{-5}$.

Вычислить сумму ряда $S = \sum_{k=1}^{\infty} \frac{1}{(2k-1)(2k+1)}$, точное значение: 0.5.

Определить, как изменяется число итераций при изменении точности.

Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator.

Часть 3

Составить метод differ для вычисления производных функции $Y(X)$ в некоторых 3 соседних точках, отстоящих на величину шага h . Для вычислений использовать формулы Лагранжа: $y'_0 = \frac{-3y_0 + 4y_1 - y_2}{2h}$; $y'_1 = \frac{-y_0 + y_2}{2h}$; $y'_2 = \frac{y_0 - 4y_1 + 3y_2}{2h}$, где y_0 , y_1 и y_2 — координаты точек. В основной программе использовать метод differ для вычисления производных функций $\sin(x)$ и $tg(x + 1)$ в точках 0.49, 0.5 и 0.51.

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

ЛР 7

Часть 1

Организовать программным способом текстовый файл **F**, где каждая строка содержит несколько натуральных чисел в их символьном представлении. Числа разделяются запятыми или пробелами и определяют вид некоторой геометрической фигуры, ее размеры и положение на экране. Приняты следующие соглашения:

1. для отрезка последовательность состоит из 5 чисел: первое число — 1, четыре остальных задают концы отрезка;
2. для окружности задается четыре числа: первое число — 2, остальные координаты центра и радиус.

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 — Прямоугольник, характеризующийся размерами. Объект умеет выводить на экран значения своих полей и отвечать на запрос о площади.

Объект — Прямоугольный параллелепипед, характеризующийся размерами. Объект умеет выводить на экран содержимое своих полей, возвращать по запросу их содержимое и определять объем параллелепипеда.

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 все меньшие n ($n \leq 10^6$) натуральных числа, которые являются палиндромами как в десятичной, так и в двоичной системе. При программировании использовать функцию. Вывести на печать все найденные числа в десятичной и двоичной системе. Например: 33 — 100001.