

Вариант 4.33.

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 корректировку. Корректировка заключается в следующем. Если слово содержит символы, отличные от букв латинского алфавита и цифр, то удалить его. Если слово состоит из букв латинского алфавита и цифр и начинается с цифры, заменить эту цифру символом «_» подчеркивание. Вести подсчет количества корректировок. Вывести на печать исходные и скорректированные 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 `maxim` для определения максимального расстояния между двумя кривыми $F(x)$ и $G(x)$ в точке $x \in [a, b]$. В основной программе использовать метод `maxim` для функций $y = \frac{\sin(x)}{x}$ и $y = \frac{\operatorname{tg}(x+1)}{x+1}$ в интервале $[0.5, 1]$ с шагом 0,01.

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

ЛР 7

Часть 1

Дан текстовый файл **F**. Записать в перевернутом виде строки файла **F** в файл **G**. Порядок строк в файле **G** должен совпадать с порядком исходных строк в файле **F**.

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, включающий поле — слово. Объект умеет выводить на экран значение своего поля и отвечать на запрос о его значении и количестве букв в слове.

Объект, включающий поля: целое число (длина слова) и слово. Объект умеет выводить на экран содержимое своих полей, возвращать по запросу их содержимое и количество согласных букв слова.

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.