

Вариант 5.22.

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 или RSpec. Везде, где это возможно, данные для проверки должны формироваться автоматически по правилам, указанным в задании.

При реализации программ везде, где это возможно, следует избегать использования циклов `for`, `do`, `while`. Вместо них используйте методы из примеси `Enumerable`.

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

ЛР 5

Часть 1

Вычислить: $y = -\frac{\cos^{1/2}\left(\frac{5 \cdot x}{1-6 \cdot x}\right)}{x^2}$.

Часть 2

Дан одномерный массив числовых значений, насчитывающий N элементов. Определить, образуют ли элементы массива, расположенные перед первым отрицательным элементом, возрастающую последовательность.

Часть 3

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, мышонок – 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 с точностью $\xi = 10^{-2}, 10^{-4}$, организовав итерационный цикл. Найти первый член последовательности $y = \frac{n}{n^2 + 2}$, для которого $y < \xi$. 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. Записать в файл **G** все четные числа файла **F**, а в файл **H** – все нечетные. Порядок следования чисел сохраняется.

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 гласных букв в слове.

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:

Существует гипотеза, что для любого натурального числа, если оно не палиндром, можно получить палиндром за конечное число шагов по следующей схеме. Меняем порядок его цифр на обратный и складываем исходное число с получившимся. Если сумма не палиндром, то над ней повторяется то же действие и так до тех пор, пока не получится палиндром. Написать программу, подтверждающую эту гипотезу и выводящую результаты на каждом шаге в формате: номер шага, текущее значение, указание подтверждена ли гипотеза или нет.