

## Вариант 4.6.

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:  $y = \frac{\cos(x + 3.1 \cdot z)}{tg(x/r)}$ .

### Часть 2

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в удалении или замене слов. Если слово стоит на четном месте и начинается на букву «е» — оно удаляется; если слово стоит на четном месте и начинается на букву «с», оно замещается на слово, введенное 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 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 root отыскания максимального отрицательного корня уравнения  $f(x) = 0$  с точностью 0.1 В основной программе использовать метод для решения уравнений  $-x^2 + \sin \frac{x}{2} = 0$  и  $x^5 + 3x^3 + x^2 + 1 = 0$ .

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

## ЛР 7

### Часть 1

Организовать программным способом символьный файл **F**. Записать в файл **G** строки файла **F** в обратном порядке, удалив символы e,E,r,t,U. При возникновении непредвиденных ситуаций выдать соответствующие сообщения.

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 — автомобиль, хранящий и умеющий выводить на экран год выпуска и пробег, и также умеющий определять средний пробег в год.

Объект — автомобиль, хранящий свой номер, марку, год выпуска, пробег и фамилию владельца и умеющий выводить их на экран эти параметры и средний пробег.

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:

Простое число называется числом Мерсена, если оно может быть представлено в виде  $2^p - 1$ , где  $p$  — тоже простое число. Написать программу, определяющую количество чисел Мерсена  $\leq$  некоторого  $n$  ( $n$  вводится с клавиатуры). Вывести на печать числа Мерсена и их количество в указанном диапазоне.