

## Вариант 4.23.

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:  $b = \frac{1 + \cos(y - 2)}{\frac{x^4}{2} + \sin^2(z)}$ .

### 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, 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 root отыскания минимального корня уравнения  $f(x) = 0$  с точностью 0.1 на отрезке  $[a, b]$ . В основной программе использовать метод для решения уравнений  $x^2 + \sin \frac{x}{2} = 0$  и  $\arctg(x) + x = 1$ .

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

## ЛР 7

### Часть 1

Сформировать программным путем символьный файл **F**, содержащий слова. Определить количество слов в файле **F**, имеющих длину 5 символов.

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 типе: квадрат или нет.

Объект — Прямоугольный параллелепипед, характеризующийся размерами. Объект умеет выводить на экран содержимое своих полей, возвращать по запросу их содержимое и определять тип 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 счастливые шестизначные номера  $n$  из возможных ( $0 \leq n \leq 999999$ ). Число считается счастливым, если сумма его первых трех цифр равна сумме его трех последних цифр. Если в числе меньше шести цифр, то недостающие начальные цифры заменяются нулями. Вывести на печать все полученные числа.