

## Вариант 4.19.

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: 
$$a = \frac{|x - 1|^{1/2} - |y|^{1/2}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}.$$

### Часть 2

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в следующем. Изменить порядок букв на обратный в каждом слове, стоящем на нечетной позиции (нумерация слов с нуля). 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. Вычислить сумму ряда  $S = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k}$  с точностью  $\xi = 10^{-2}, 10^{-4}$ . Точное значение:  $\ln 2$ . Определить, как изменяется число итераций при изменении точности.

## Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator.

## Часть 3

Составить метод `intprg` вычисления определенного интеграла по формуле прямоугольников:  $S = \frac{b-a}{n} \sum_{i=1}^n f(x_i)$ , где  $n$  – количество отрезков разбиения. В основной программе использовать метод `intprg` для вычисления интегралов:  $\int_0^1 \frac{e^x}{x+1} dx$  и  $\int_0^2 x(x-1) dx$ .

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 **H** с сохранением порядка следования те символы файла **F**, которым в этом файле предшествует буква «а».

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 — «тройка» — строка — два числа операнды и операция между ними, например «2+3». Объект умеет выводить на экран свою строку и результат.

Объект — «тройка» в скобках. Объект умеет выводить на экран строку и результат.

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.