

## Вариант 2.38.

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:  $y = \cos \frac{x^2}{(x-2)3x}$ .

### Часть 2

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в удалении лишних пробелов и слов, состоящих из одного символа. Лишними считаются пробелы в начале и конце строки, а также более одного пробела между словами. 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 длину окружности с точностью  $\xi = 10^{-3}, 10^{-4}$  как предел последовательности периметров вписанных правильных многоугольников с удваивающимся числом сторон (начать с  $n = 6$ ). Использовать формулу удвоения стороны  $n$ -угольника:  $a_{2n} = \sqrt{2R^2 - 2R\sqrt{R^2 - a_n^2/4}}$ .

## 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

Сформировать программным путем символьный файл **F**, содержащий слова. Считая, что количество символов в слове не превосходит двадцати определить, сколько в файле **F** имеется слов, состоящих из двух 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 — точка на плоскости, заданная координатами, умеющая выводить их на экран и возвращать в ответ на запрос.

Объект — центр и радиус окружности, умеющий выводить их на экран координаты центра и радиус.

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:

Известно, что произведение двух целых чисел, деленное на их наибольший общий делитель, дает наименьшее общее кратное.

Написать программу, определяющую наибольший общий делитель двух целых чисел  $n$  и  $m$  ( $n, m$  вводятся с клавиатуры), используя известный алгоритм Евклида и наименьшее общее кратное этих же чисел, опираясь на утверждение, сделанное в начале. Вывести на печать все найденные числа и результаты промежуточных итераций. При программировании использовать функции.