

Вариант 4.28.

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:
$$a = \frac{2 \cdot \cos\left(x - \frac{\pi}{6}\right)}{\frac{1}{2} + \sin^2(y)}.$$

Часть 2

С клавиатуры вводится две строки: в первой — слова, разделенные пробелами, во второй — числа, разделенные пробелами. Написать, программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в следующем. Каждому слову из первой строки ставится в соответствие число из второй, их соответствие определяется порядковым номером (первое слово — первое число, второе слово — второе число и т.д.). Числа представляют желаемую длину слов. Если длина слова больше желаемой, сократить длину слова до желаемой длины, если меньше или равна — оставить без изменений. Вывести скорректированную строку. Например, первая строка «Мама мыла раму», вторая строка «2 2 10», результат «Ма мы раму».

Указание: при выполнении задания рекомендуется использовать метод `zip` класса `Array`.

Автоматический тест программы обязательно должен генерировать случайные строки в соответствии с правилами, перечисленными в задании.

ЛР 6

Часть 1

Решить задачу, организовав итерационный цикл с точностью $\xi = 10^{-3}, 10^{-4}$.
Вычислить значение определенного интеграла методом трапеции:

$\int_0^{\pi} (1 + \sin x) dx$, точное значение: $\frac{\pi}{2} + 1$. 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

Организовать программным способом файл **F**, компоненты которого являются целыми числами. Получить в файле **G** все компоненты файла **F**, делящиеся на 3 и не делящиеся на 7. 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 — треугольник, заданный длинами сторон. Объект умеет выводить на экран значение своих полей и отвечать на запрос о площади фигуры.

Объект — четырехугольник, заданный длинами сторон и диагонали. Объект умеет выводить на экран содержимое своих полей, возвращать по запросу их значения и площадь.

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.