

Вариант 3.26.

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` или `RSpec`. Везде, где это возможно, данные для проверки должны формироваться автоматически по правилам, указанным в задании.

При реализации программ везде, где это возможно, следует избегать использования циклов `for`, `do`, `while`. Вместо них используйте методы из примеси `Enumerable`.

Все тексты программ должны быть проверены на соответствие стилю программирования Ruby при помощи `rubocop .ru` и `reek`.

ЛР 5

Часть 1

Вычислить: $y = tg(x^3 + x^2 - x) - \sin(|x^3 + x^2 - x|) + (x^3 + x^2 - x)^2$.

Часть 2

Дана строка, состоящая из символов латиницы. Необходимо проверить, образуют ли прописные символы и числа из этой строки палиндром.

Часть 3

Дана последовательность строк, содержащая данные о сотрудниках некоторого учреждения, записанные по образцу: Фамилия Имя Отчество. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в записи данных по образцу

- Имя Отчество Фамилия
- Фамилия И.О.

Вывести на печать исходную и скорректированную последовательности строк. Предусмотреть случай отсутствия отчества и случай двойных имён.

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 $\xi = 10^{-4}, 10^{-5}$.
Вычислить сумму ряда:

$$S = 1 + \sum_{k=1}^{\infty} \frac{(-1)^k}{k!}, \text{ точное значение равно } \frac{1}{e}.$$

Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator.

Часть 3

Составить метод `neibg` проверки принадлежности точки плоскости с координатами (x, y) данной кривой $y = f(x)$. В основной программе использовать метод `neibg` для проверки принадлежности десяти различных точек кривым $y = \cos(x)$ и $y = \sin(x^2)$.

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

ЛР 7

Часть 1

Организовать программным способом файл **F**, компонентами которого являются действительные числа. Найти:

1. наименьшее из значений компонент с нечетными номерами;
2. сумму наименьшей и наибольшей компонент;

3. разность первой и последней компонент;
4. при наличии в файле более 10 отрицательных значений выдать на экран сообщение.

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.