

Вариант 4.16.

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: $y = \frac{1 - (1 + |\sin(x)|)^{1/2}}{2 + \cos^2(x)}$.

Часть 2

Дана последовательность строк. Строки содержат слова, разделенные пробелом. Используя цифровой шифр, например 31206, зашифровать каждую строку по следующей методике: 31206 312063 12 063

Пирог сгорел до тла
Ткток фкррло ер есг

То есть, к каждой букве применяют соответствующую цифру для определения смещения этой буквы, с целью получения буквы шифра. Написать программу, обеспечивающую ввод строк, шифровку и дешифровку. Вывести на печать зашифрованную и подвергнутую дешифровке 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 $S = \sum_{k=1}^{\infty} \frac{1}{k(k+1)(k+2)(k+3)}$ с точностью $\xi = 10^{-2}, 10^{-3}$. Точное значение: $\frac{1}{3 \cdot 3!}$. Определить, как изменяется число итераций при изменении точности.

Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator.

Часть 3

Составить метод trap для вычисления определенного интеграла по формуле трапеций $\int_a^b f(x) dx \approx \left(\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(x_i) \right) \cdot \frac{b-a}{n}$, где $f(x)$ подынтегральная функция, $[a, b]$ - интервал интегрирования, n - число отрезков разбиения. В основной программе использовать метод trap для вычисления интегралов: $\int_{-1}^4 (x + \cos x) dx$ и $\int_1^2 \frac{tg(x+1)}{x+1} dx$.

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

ЛР 7

Часть 1

Дан текстовый файл **F**. Переписать строки файла **F** в файл **G**, вставляя в начало каждой строки по одному пробелу. Порядок строк должен быть сохранен.

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 — окно, хранящее размеры и умеющее выводить на экран площадь.

Объект — окно, хранящее размеры и сведения о наличии жалюзи и умеющее выводить на экран площадь и признак наличия жалюзи.

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 Симона о факториале. Она гласит, что существует 4 факториала, которые представимы в виде произведения трех последовательных чисел. Например: $4! = 2 \cdot 3 \cdot 4$. Написать программу, определяющую эту четверку, подсчитать действительное количество таких факториалов, если это возможно и тем самым доказать неправильность гипотезы. Вывести на печать результаты каждой итерации и финальных расчетов.