

Вариант 3.28.

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

Часть 2

С клавиатуры вводится целочисленный массив и число К. Найти К наибольших и К наименьших элементов в этом массиве и вывести их.

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

да $S = \sum_{k=1}^{\infty} \frac{1}{k(k+1)(k+2)(k+3)}$ 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

$$\int_a^b f(x) dx \approx \frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(x_i), \text{ где } f(x) \text{ подынтегральная функция, } [a, b] - \text{интервал интегрирования, } n - \text{число отрезков разбиения. В основной программе использовать метод trap для вычисления интегралов: } \int_{-1}^4 (x + \cos x) dx \text{ и } \int_0^2 \frac{tg(x+1)}{x+1} dx.$$

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

ЛР 7

Часть 1

Организовать программным способом символьные файлы **F** и **G**. Переписать в файл **H** все начальные совпадающие компоненты файлов **F** и **G**. При возникновении непредвиденных ситуаций выдать соответствующие сообщения.

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 — больной. Параметры: фамилия, возраст. Методы: инициализирующий, вывода на экран фамилии и возраста.

Объект — больной. Параметры: фамилия, возраст, год последней диспансеризации. Методы: инициализирующий, определения года следую-

щей диспансеризации, исходя из того, что диспансеризация должна проводиться каждые 3 года.

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.