Вариант 2.10.

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: $y = \dfrac{\cos(x + 3.1 \cdot z)}{tg(x/r)}$.

Часть 2

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в следующем. Если слово содержит символы, отличные от букв латинского алфавита и цифр, то удалить его. Если слово состоит из букв латинского алфавита и цифр и начинается с цифры, заменить эту цифру символом «_» подчеркивание. Вести подсчет количества корректировок. Вывести на печать исходные и скорректированные 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 с точностью $\xi = 10^{-3}, 10^{-4}$. Вычислить длину кривой, определяемой функцией $y = ln\ x$ при $x \in [1, 2]$. Определить, как изменяется число разбиений при изменении точности.

Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator.

Часть 3

Составить метод neibr проверки принадлежности точки плоскости с координатами (x, y) данной кривой $y = f(x)$. В основной программе использовать метод neibr 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**, компоненты которого являются целыми числами. Число компонент файла делится на 5. Записать в файл **G** наибольшее значение первых пяти компонент файла **F**, затем – следующих пяти компонент и т.д.

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 — целое число. Объект умеет выводить на экран значение своего поля и отвечать на запрос о его значении и количестве цифр в числе.

Объект, включающий поля: целое число и вещественное число. Объект умеет выводить на экран содержимое своих полей, возвращать по запросу их содержимое и количество цифр целого числа.

В тестирующей программе обеспечить автоматическую проверку того, что созданные объекты действительно соответствют 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:

Дано натуральное число m. Написать программу, определяющую такое натуральное число n, что двоичная запись числа n получается из двоичной записи числа m изменением порядка цифр на обратный порядок их следования. Например: $6 = 110$, а $3 = 011$. Вывести на печать числа и их двоичное представление.