**# 1.** Вивести "Hello, world!"

**# 2.** Написати функцію(метод), яка буде примймати аргумент name і виводити "Hello, [name]", наприклад якщо name - "Оля", то функція повинна вивести "Hello, Оля!". Програма повинна вивести кілька таких "привітів"

**# 3.** Те саме що й попереднє, тільки в аргумента name повинно бути значення по замовчанню. Програма повинна вивести кілька таких "привітів", один з них - за замовчанням

**# 4.** Клас Greeter, в якому повинен бути метод hello, який приймає аргумент name з дефолтним значенням nil, якщо не передано ніякого значення (або nil), то метод повинен виводити Hello, world!, якщо передане значення то Hello, #{name}!. В програмі створити об'єкт класу Greeter і вивести з його допомогою кілька "привітів".

**# 5.** поміняти клас Greeter щоб він використовував тернарний оператор (ternary operator) test ? value1 : value2

**# 6.** поміняти клас Greeter щоб він використовував логічний оператор "or"

5 і 6 метод hello - в один рядок

**# 7**. об'явити змінну "a", присвоїти їй значення "string", вивести її значення і клас (a.class), присвоїти значення 5, вивести значення і клас, те саме для 3.14, для true, false, :name, nil

**# 8**. об'явити масив з назвами 5-ти країн, вивести його з допомогою puts і p, вивести список країн з номерами від 1 до 5, з допомогою змінної і метода each. наприклад:

# 1. Ukraine

# 2. Canada

# .....

**# 9**. об'явити порожній масив array, заповнити його 10-ма значеннями випадкових чисел (з допомогою генератора випадкових чисел), від 0 до 100, вивести масив, знайти мінімальне, максимальне і середнє значення, вивести їх

**# 10**. написати функцію сортування масива бульбашкою (bubble sort), об'явити порожній масив array, заповнити його 100 значеннями випадкових чисел (з допомогою генератора випадкових чисел), відсортувати за допомогою написаної функції. Вивести початковий масив і відсортований масив з допомогою p.

**# 11**. Винести функцію сортування бульбашкою з попереднього завлання в окремий файл. Включити цей файл в програму для цьго завдання (за допомогою require). Об'явити порожній масив array, заповнити його 1000000 значеннями випадкових чисел (з допомогою генератора випадкових чисел), відсортувати за допомогою своєї функції сортування, заміряти час виконання даної операції і вивести цей час, відсортувати за допомогою стандартного sort, заміряти час даної операції і вивести.

**# 12**. Об'явити хеш з ключами :firstname, :lastname, :age і якимись значеннями. вивести його з допомогою функції p, вивести його в красивому вигляді, напр:

# First name: Vasya

# Last name: Pupkin

# Age: 22

# поміняти значення віку, знову вивести

**# 13.** Об'явити порожній масив. Об'явити масив з 5-10 різними іменами, об'явити масив з 5-10 прізвищами. заповнити перший масив 30-ма хешами як в завданні 10, з іменами і прізвищами, взятими рандомно з масивів імен і прізвищ і випадковим віком від 10 до 20. Вивести за допомогою pp. Вивести як нумерований список сортований по алфавіту прізвищ

**# 14**. зробити сортування бульбашкою в окремому файлі (з завдання 11) як mixin для Array, два варіанти buble\_sort! і bubble\_sort

**# 15**. написати функцію обчислення факторіала через рекурсію, кидать ексепшен якщо аргумент менше нуля

**# 16**. написати функцію обчислення факторіала через цикл, кидать ексепшен якщо аргумент менше нуля

**# 17**. згенерувати масив з 10 числами від 0 до 20, замінити числа в масиві на ці самі числа в квадраті

**# 18**. Написати клас Person з конструктором в який передається три параметри: ім'я, прізвище і вік. Значення цих параметрів присвоїти відповідним instance variables. Написати метод to\_s, який вертатиме стрінг, приблизно як в задачі 13. напр: "Пупкін Вася 13 років". Створити пару об'єктів цього класу і вивести їх за допомогою команд p i puts

**# 19**. Додати до класа Person з попередньої задачі методи для отримання імені, прізвища і віку. щоб можна було написати person.name, person.lastname, person.age. Повиводити значення з допомогою цих методів (напр puts person.lastname)

**# 20**. Додати до класа Person з попередньої задачі методи для присвоювання імені, прізвища і віку. щоб можна було написати person.name = "John", person.lastname = "Snow", person.age = 30. Поприсвоювати значення і повиводити об'єкт(и)

**# 21**. замінити код методів отримання і присвоювання на attr\_accessor. Перевірити чи все працює так само як і з методами написаними вручну

**# 22**. написати клас Animal (тварина), з методом move, який просто виводить щось типу "I'm moving...", написати клас Bird, який наслідується (inherit) від класу Animal, і має додатковий метод fly, який виводить щось типу "I'm flying...". Створити об'єкт класу Bird, виконати в нього методи move і fly

**# 23**. Написати клас Runner (бігун), в якого є метод speed, який повертає 8.0 (типу 8 км/год), написати клас Sprinter (спринтер), який повертає швидкість вдвічі більшу від швидкості просто бігуна, написати клас Jogger (бігун підтюпцем (трусцой) ), швидкість такого бігуна має бути 0.75 швидкості типового бігуна

**# 24**. Написати клас Point (точка, в геометричному сенсі), з двома змінними об'єкта (instance variables) - x, y, які доступні для читання і запису, і можуть передаватись в конструктор, якщо в конструктор не передаєм нічого - то 0, 0. Зробити щоб код для тестування не виконувався коли файл включається з іншого файла

# повинно використовуватись так:

p1 = Point.new

=> #<Point:0x007ff0a3922f98 @x=0, @y=0>

p2 = Point.new(10, 20)

=> #<Point:0x007ff0a3913ef8 @x=20, @y=10>

p1.x = 1

p p1

#<Point:0x007ff0a3922f98 @x=1, @y=0>

p2.y

=> 10

**# 25**. Написати клас Shape (фігура, в геометричному сенсі), змінна об'єкта position (координата, використати клас Point з попередньої задачки), методи: area (площа), center (центр), які нічого не роблять (можливо кидати ексепшен "not implemented")

**# 26**. Написати клас Rectangle (прямокутник), наслідується від Shape, в конструктор передаєм 4 параметра - x, y, w, h - координата лівого верхнього кута, ширина, висота. Реалізувати методи area і center - площа і центр прямокутника відповідно

**# 27**. Написати клас Circle (коло), наслідується від Shape, в конструктор передаєм 3 параметра - x, y, r - координата центра, радіус. Реалізувати методи area і center - площа і центр кола відповідно

**# 28**. Додати в класи Shape, Rectangle i Circle метод bound, який повертає найменший описуючий прямокутник (для Shape він порожній з ексепшеном, для прямокутника він сам себе поверає або копію, для кола - ну понятно :) )

**# 29**. Клас Polygon, наслідується від Shape, в конструктор передається масив Point - вершини багатокутника, position - перша вершина. Реалізувати center (як середнє арифметичне відповідних координат), bound (як найменший описуючий прямокутник) і area (площа багатокутника)

**# 30.** Руками скачуєм файл https://upload.wikimedia.org/wikipedia/commons/5/55/Blank\_map\_of\_Ukraine.svg і кладем в папочку з усім цим і в репозиторій. Власне завдання: прочитати цей файл порядково, знайти за допомогою регулярного виразу (regexp) рядок, який починається з "d=" (він там один). Взяти вміст лапок (те що між " і "), розбити на елементи масива (елементи розділяються одним або більше пробілами, тому теж регексп). Вивести масив за допомогою "p"

**# 31**. Для класа Polygon з завдання 29 пишем метод класа polygon\_from\_svg\_path\_elements(array), який бере масив з попереднього завдання (30) і формує з нього полігон (документація по SVG paths: http://www.w3.org/TR/SVG/paths.html#PathDataGeneralInformation)

**# 32**. З коду з попередніх двох завдань робимо для класа Polygon метод класа polygon\_from\_svg\_path(string), який приймає svg path (як стрінг, без лапок), і формує з нього полігон

**# 33**. Модифікувати метод polygon\_from\_svg\_path так щоб йому можна було передати і масив і стрінг

**# 34**. Написати модуль для кодування англійського текста азбукою морзе і назад.