Отчёт по итоговой четвёртой работе   
по Python за 1 семестр

(Python LR4)

Выполнил: Соболь Евгений

Группа: ФПММ-ИСТ-19-1бзу

**Вариант: 1**

**Задание**

Реализовать программный модуль шифрующий стихотворение, записанное в файл. Результат сохранить в отдельном файле. Затем разработать отдельный модуль – дешифровщик. Модуль должен загружать данные из файла с зашифрованным текстом и выводить на экран расшифрованный текст. Примеры алгоритмов есть по ссылке:

<https://tproger.ru/translations/10-codes-and-ciphers/>

Для реализации выбран алгоритм № 4 “Шифр Виженера”

**Решение:**

**Программный код модуля шифрования itog.py:**

import numpy as np

class vij:

"Шифр Виженера модуль шифрования. Принимает на вход 4 параметра 1 - кодовое слово, 2 - имя файла в котором текст 3 -имя файла в который записать шифр 4 - словарь(если не указать берётся по умолчанию)"

# Словарь по умолчанию( Массив с набором используемых символов)

\_\_mas = ['а', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и', 'й',

'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф',

'х', 'ц', 'ч', 'ш', 'щ', 'ь', 'ы', 'ъ', 'э', 'ю', 'я',

'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й',

'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф',

'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ь', 'Ы', 'Ъ', 'Э', 'Ю', 'Я',

'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k',

'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',

'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G',

'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R',

'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '!', '/', '#',

'.', ':', '&', '?', '-', '—', ',', '~', '1', '2', '3',

'4', '5', '6', '7', '8', '9', '0', ' ', ']']

# Функция проверки символов в строке на наличие в словаре

@staticmethod

def \_\_validValue(s, m):

for i in range(len(s)):

if m.count(s[i]) == 0:

print('В введённых данных используется недопустимый символ!')

exit()

# Функция шифрования строки

# 1 - словарь 2-строка 3-кодовое слово

@staticmethod

def \_\_shifr(m, s, k):

n = 0

# Создаю массив для шифрования

mas1 = np.zeros((2, len(s)))

# Проверяю значения шифруемой строки на присутствие в словаре(массиве)

vij.\_\_validValue(s, m)

# Проверяю значения кодового слова на присутствие в словаре(массиве)

vij.\_\_validValue(k, m)

# Заполняю первую строку порядковыми номерами элементов

for i in range(len(s)):

mas1[0][i] = m.index(s[i])

# Заполняю вторую строку порядковыми номерами элементов кодового слова

for i in range(len(mas1[1])):

if n >= len(k): n = 0

mas1[1][i] = m.index(k[n])

n += 1

# Складываем порядковые номера элементов кодового слова и шифруемой строки

for i in range(len(mas1[1])):

mas1[1][i] += mas1[0][i]

# По модулю

if mas1[1][i] > (len(m) - 1): mas1[1][i] -= (len(m) - 1)

s = ''

# Записываем зашифрованные символы в строку

for i in range(len(mas1[1])):

s += m[int(mas1[1][i])]

# Возвращаю строку

return s

# Функция считывания, записи и шифрования

# 1- словарь, 2-кодовое слово, 3- файл с текстом 4 - шифр

@staticmethod

def shifrVij(k, r, w, m = \_\_mas):

# Открываю файл на чтение

f = open(r)

# Открываю файл на запись

w1 = open(w, 'w')

# Считываю построчно из файла

for line in f:

l = line

# Удаляю знак переноса строки

l = l[:-1] if l.find("\n") != -1 else l

# Записываю в файл зашифрованную строку

w1.write(str(vij.\_\_shifr(m, l, k)) + '\n')

# Закрываю файл на запись

w1.close()

# Закрываю файл на чтение

f.close()

print("Успешно!")

**Пример использования модуля шифрования:**

rf = input("Введите название файла, который зашифровать: ")

wf = input("Введите файл в который записать шифр:")

kod = input("Введите кодовое слово: ")

r = vij()

r.shifrVij(kod, rf, wf)

**Программный код модуля дешифрования itog\_2.py:**

import numpy as np

class vijDech:

"Шифр Виженера модуль дешефровки. Принимает на вход 3 параметра 1 - кодовое слово, 2 - имя файла который дешифровать 3 - массив словарь(если не указать берётся по умолчанию)"

# Словарь по умолчанию(Массив с набором используемых символов)

\_\_mas = ['а', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф',

'х', 'ц', 'ч', 'ш', 'щ', 'ь', 'ы', 'ъ', 'э', 'ю', 'я', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й',

'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ь', 'Ы', 'Ъ', 'Э', 'Ю', 'Я',

'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',

'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R',

'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '!', '/', '#', '.', ':', '&', '?', '-', '—', ',', '~', '1', '2', '3',

'4', '5', '6', '7', '8', '9', '0', ' ', "]"]

@ staticmethod

def \_\_validValue(s, m):

# Проверяю присутствует ли символ в словаре

for i in range(len(s)):

if m.count(s[i]) == 0:

print('В введённых данных используется недопустимый символ!')

exit()

# Функция дешифрования строки

# 1 - словарь 2-строка 3-кодовое слово

@staticmethod

def \_\_shifr(m, s, k):

# Счётчик длины кодового слова

n = 0

# Создаю матрицу 2 X длина строки

mas1 = np.zeros((2, len(s)))

# Проверка данных на присутствие в словаре(массиве с символами)

vijDech.\_\_validValue(s, m)

# Проверка данных на присутствие в словаре(массиве с символами)

vijDech.\_\_validValue(k, m)

# Заполняю первую строку матрицы порядковыми номерами элементов

for i in range(len(s)):

mas1[0][i] = m.index(s[i])

# Заполняю вторую строку матрицы порядковыми элементами кодового слова

for i in range(len(mas1[1])):

if n >= len(k): n = 0

mas1[1][i] = m.index(k[n])

n += 1

# Вычитаю из первой строки вторую строку

for i in range(len(mas1[1])):

mas1[0][i] -= (mas1[1][i])

# По модулю

if mas1[0][i] < 0: mas1[0][i] += (len(m) - 1)

s = ''

# Записываю дешефрованные символы в строку

for i in range(len(mas1[0])):

s += m[int(mas1[0][i])]

# Возвращаяю строку

return s

# Функция чтения из файла строк, дешефровка и вывод на экран.

@staticmethod

def shifrVijDech(k, r, m = \_\_mas):

# Открываю файл на чтение

f = open(r)

# Построчно считываю из файла данные и дешефрую

print("--------------------------------")

for line in f:

l = line

# Удаляю символы переноса строки если они есть

l = l[:-1] if l.find("\n") != -1 else l

# Вывод на экран результата дешефровки строки

print(vijDech.\_\_shifr(m, l, k))

# Закрываю файл на чтение

f.close()

print("--------------------------------")

print("Успешно!")

**Пример использования модуля дешифрования:**

rf = input("Введите название файла, расшифровать: ")

kod = input("Введите кодовое слово: ")

p = vijDech()

p.shifrVijDech(kod, rf)

**Результат работы модулей:**

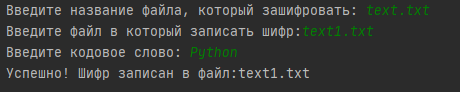


Рисунок - Ввод- Вывод в модуле шифрования

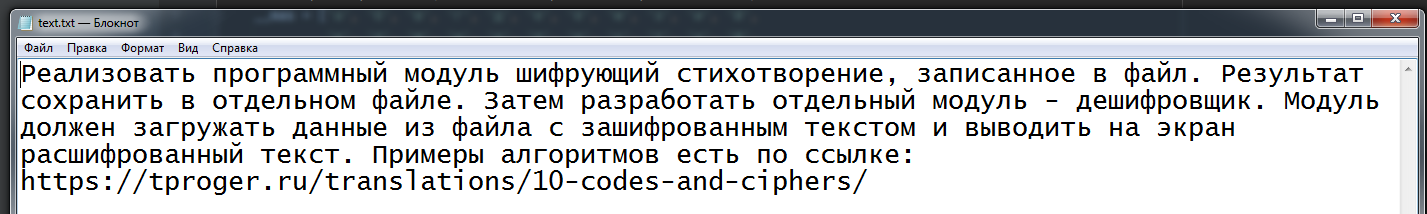


Рисунок - Текст для шифрования записанный в файл text.txt

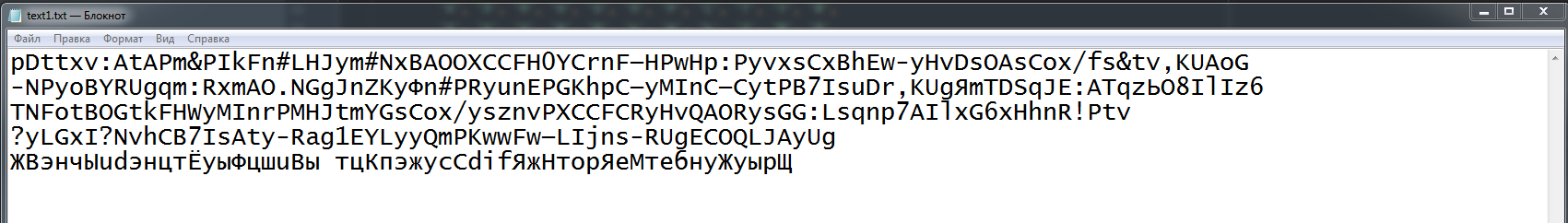


Рисунок - Результат работы модуля шифрования записанный в файл text1.txt

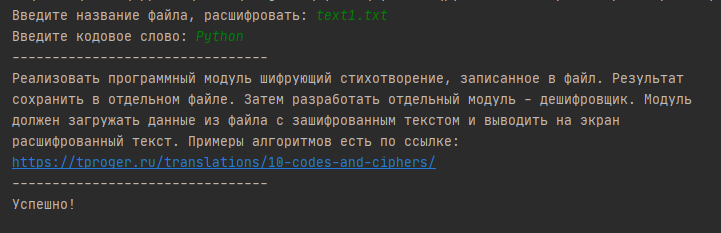


Рисунок - Ввод вывод в модуле дешифрования

**Блок-схемы модуля шифрования:**

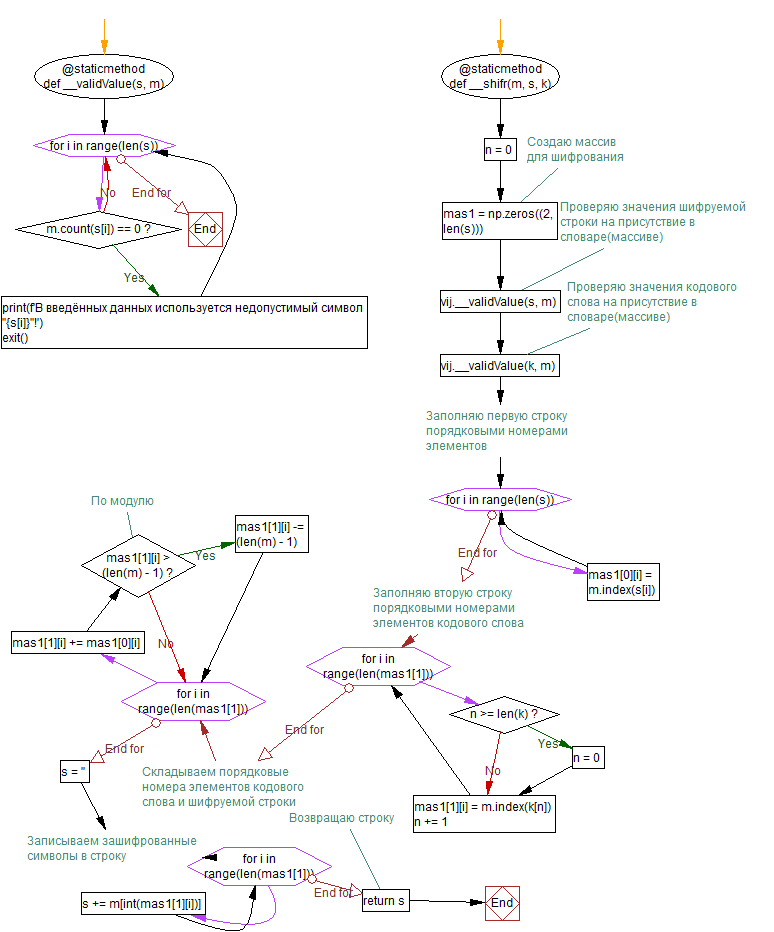


Рисунок -Блок схема методы модуля шифрования

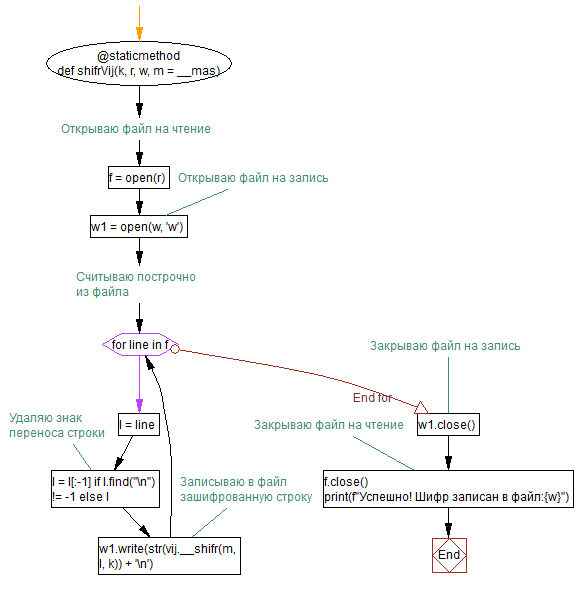


Рисунок -Блок схема методы модуля шифрования (2)

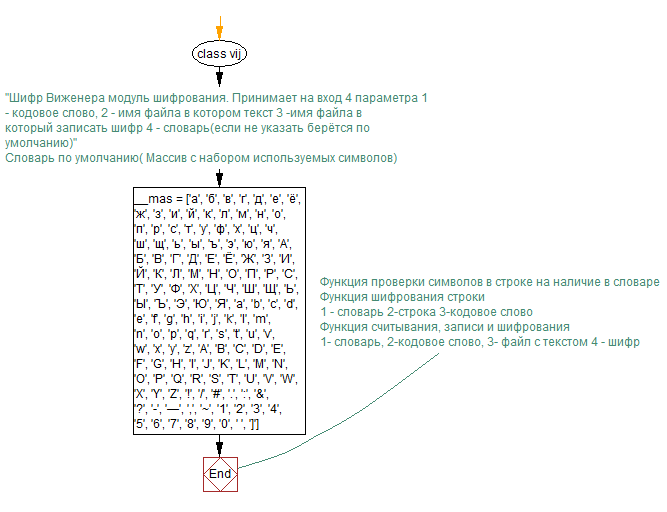


Рисунок 7-Блок-схема класса модуля шифрования

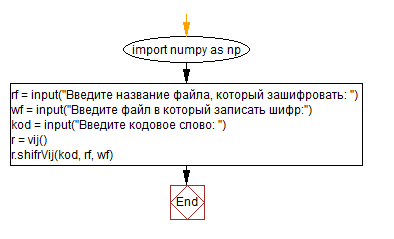


Рисунок 8-Блок схема шифрования с помощью модуля шифрования

**Блок-схемы модуля дешифрования:**

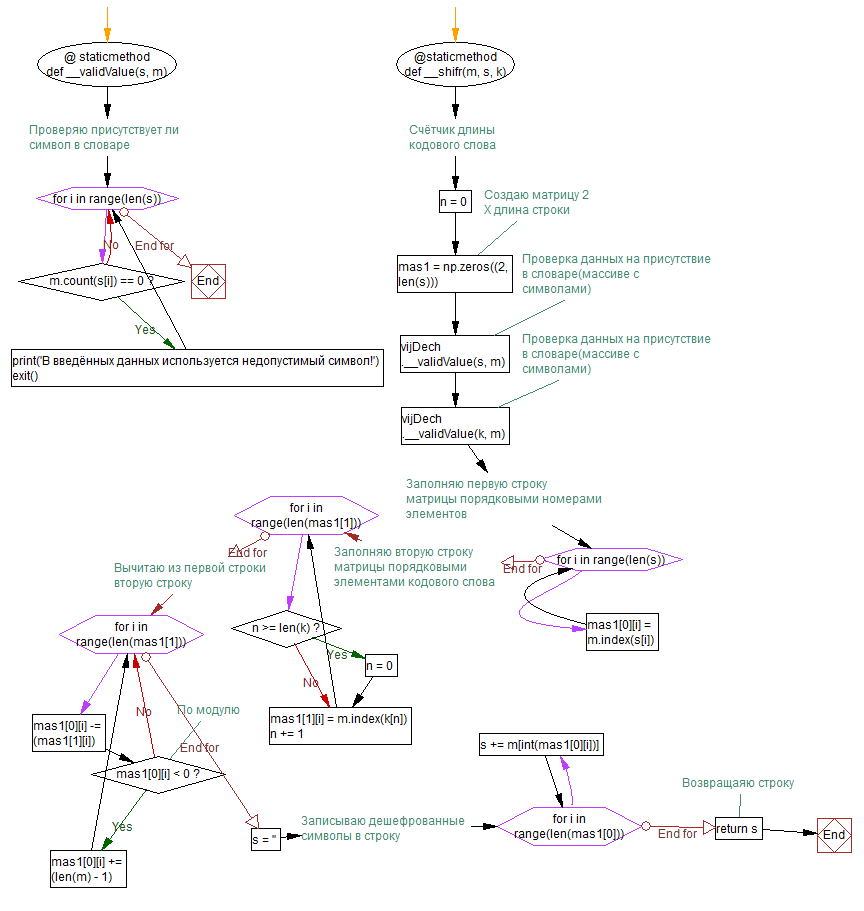
****

Рисунок 9- Методы модуля дешифрования

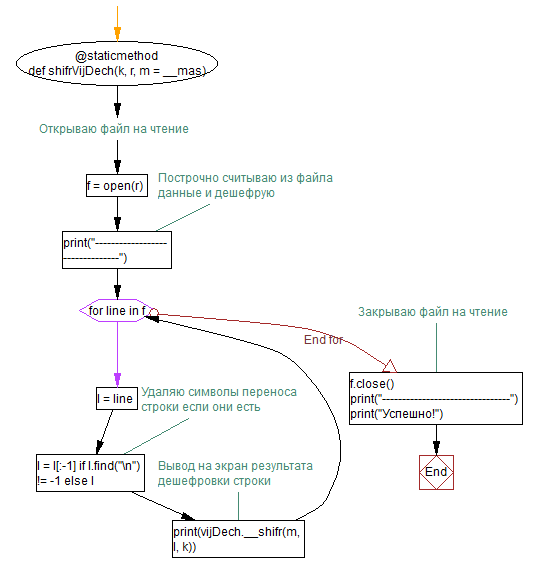


Рисунок 0-Метод модуля дешифрования

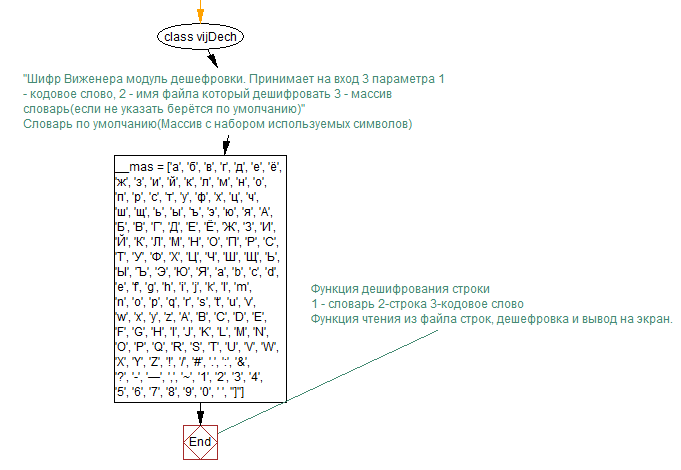


Рисунок 11-Блок схема класса модуля дешифрования

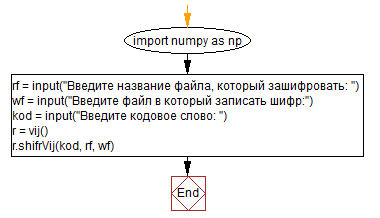


Рисунок 12-Блок-схема использования модуля дешифрования