Лабораторная работа №4

**Тема:** Тестирование

**Задача 4.1**

**Постановка задачи:** Разработать программу для считывания данных JSON-формата из файла и вывод их в табличном виде на экран. Организовать тестирование работоспособности программы с помощью assert, print.

**Код программы:**

"""

Разработать программу для считывания данных JSON-формата из файла и вывод их в табличном виде на экран.

Организовать тестирование работоспособности программы с помощью assert, print.

"""

import json

def json\_table(data):

table = []

sstr = '| {id:^3} | {first\_name:^10} | {last\_name:^15} | {email:^30} | {gender:^6} | {ip\_address:^16} |'

t\_caption = '| {:^3} | {:^10} | {:^15} | {:^30} | {:^6} | {:^16} |'.format('id','first\_name','last\_name','email','gender','ip\_address')

roof = '-'\*len(t\_caption)

table.append(roof)

table.append(t\_caption)

table.append(roof)

for el in range(len(data)):

temp = data[el]

res = sstr.format(\*\*temp)

table.append(res)

table.append(roof)

return(table)

def test\_func(func,arg):

"""

Эта функция тестирует функцию json\_table

func - тестируемая функция

"""

try:

assert func(arg)[1] == '| id | first\_name | last\_name | email | gender | ip\_address |'

except:

print('Тест провален, шапка не соответствует ожиданиям')

try:

assert type(func(arg)) is tuple

except:

print('Тест на тип провален', type(func(arg)))

try:

assert len(func(arg)) == 100

except:

print('Количество строк не соответствует ожиданиям -',len(func(arg)))

def main():

with open('MOCKDATA.json') as f:

data\_dict = json.load(f)

test\_func(json\_table,data\_dict)

"""

for el in table:

print(el)

"""

main()

**Результат тестирования:**

**Задача 4.2**

**Постановка задачи:** Дополнение программы задания 4.1 (считывание данных JSON-формата) тестами с использованием библиотеки doctest.

**Код программы:**

"""

Разработать программу для считывания данных JSON-формата из файла и вывод их в табличном виде на экран.

Организовать тестирование работоспособности программы с помощью assert, print.

"""

import json

def json\_table():

"""

>>> json\_table()[1]

'| id | first\_name | last\_name | email | gender | ip\_address |'

"""

with open('MOCKDATA.json') as f:

data\_dict = json.load(f)

table = []

sstr = '| {id:^3} | {first\_name:^10} | {last\_name:^15} | {email:^30} | {gender:^6} | {ip\_address:^16} |'

t\_caption = '| {:^3} | {:^10} | {:^15} | {:^30} | {:^6} | {:^16} |'.format('id','first\_name','last\_name','email','gender','ip\_address')

roof = '-'\*len(t\_caption)

table.append(roof)

table.append(t\_caption)

table.append(roof)

for el in range(len(data\_dict)):

temp = data\_dict[el]

res = sstr.format(\*\*temp)

table.append(res)

table.append(roof)

return(table)

def main():

a = json\_table()

"""

for el in table:

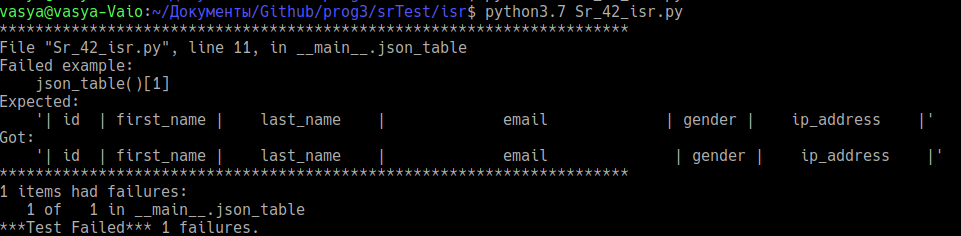
print(el)

"""

if \_\_name\_\_ == "\_\_main\_\_":

import doctest

doctest.testmod()

**Результат тестирования:**

**Задача 4.3**

**Постановка задачи:** Дополнение программы задания 4.1, 4.2 (считывание данных JSON-формата) тестами с использованием пакета py.test.

**Код программы:**

"""

Разработать программу для считывания данных JSON-формата из файла и вывод их в табличном виде на экран.

Организовать тестирование работоспособности программы с помощью assert, print.

"""

import pytest

import json

def json\_table(data):

table = []

sstr = '| {id:^3} | {first\_name:^10} | {last\_name:^15} | {email:^30} | {gender:^6} | {ip\_address:^16} |'

t\_caption = '| {:^3} | {:^10} | {:^15} | {:^30} | {:^6} | {:^16} |'.format('id','first\_name','last\_name','email','gender','ip\_address')

roof = '-'\*len(t\_caption)

table.append(roof)

table.append(t\_caption)

table.append(roof)

for el in range(len(data)):

temp = data[el]

res = sstr.format(\*\*temp)

table.append(res)

table.append(roof)

return(table)

def main():

with open('MOCKDATA.json') as f:

data\_dict = json.load(f)

a = json\_table(data\_dict)

"""

for el in table:

print(el)

"""

def test\_json\_table():

assert json\_table([{"id":1,"first\_name":"Susann","last\_name":"Wyldish","email":"swyldish0@bing.com","gender":"Female","ip\_address":"112.109.35.15"},

{"id":2,"first\_name":"Oliy","last\_name":"Bruton","email":"obruton1@forbes.com","gender":"Female","ip\_address":"166.20.188.54"},

{"id":3,"first\_name":"Ginelle","last\_name":"Inkpen","email":"ginkpen2@tinypic.com","gender":"Female","ip\_address":"208.129.167.239"}]) == ['---------------------------------------------------------------------------------------------------',

'| id | first\_name | last\_name | email | gender | ip\_address |',

'---------------------------------------------------------------------------------------------------',

'| 1 | Susann | Wyldish | swyldish0@bing.com | Female | 112.109.35.15 |',

'| 2 | Oliy | Bruton | obruton1@forbes.com | Female | 166.20.188.54 |',

'| 3 | Ginelle | Inkpen | ginkpen2@tinypic.com | Female | 208.129.167.239 |',

'---------------------------------------------------------------------------------------------------']

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Результат тестирования:**