

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:

```
vasya@vasya-Vaio:~/Документы/Github/prog3/srTest/isr$ python3.7 Sr_41_isr.py
Тест на тип провален <class 'list'>
Количество строк не соответствует ожиданиям - 104
```

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:

```

vasya@vasya-Vaio:~/Документы/Github/prog3/srTest/isr$ python3.7 Sr_42_isr.py
*****
File "Sr_42_isr.py", line 11, in __main__.json_table
Failed example:
    json_table()[1]
Expected:
'| id | first_name | last_name | email | gender | ip_address |'
Got:
'| id | first_name | last_name | email | gender | ip_address |'
*****
1 items had failures:
  1 of 1 in __main__.json_table
***Test Failed*** 1 failures.

```

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:

```

vasya@vasya-Vaio:~/Документы/Github/prog3/srTest/lsr$ python3.7 -m pytest -q Sr_43_lsr.py
1 passed in 0.01s

```

[100%]