

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

**Отчет по лабораторной работе №16
«Работа с файлами в языке Python»**

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-20-1
Бокань И.П. « » _____ 2022г.
Подпись студента _____
Работа защищена « » _____ 2022г.
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

1. Вывод (Примеры)

```
>>> help
Список команд:

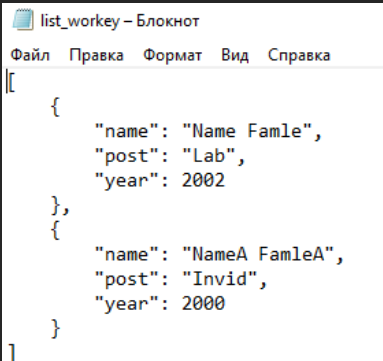
add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
load - загрузить данные из файла;
save - сохранить данные в файл;
exit - завершить работу с программой.
>>> list
Список работников пуст.
>>> add Name Famle
>>> Неизвестная команда add name famle

>>> Неизвестная команда
add
Фамилия и инициалы? Name Famle
Должность? Lab
Год поступления? 2002
>>> add
Фамилия и инициалы? NameA FamleA
Должность? Invid
Год поступления? 2000
>>> list
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |   Год   |
+-----+-----+-----+-----+
| 1 |      Name Famle      |      Lab      |   2002   |
| 2 |      NameA FamleA    |      Invid     |   2000   |
+-----+-----+-----+-----+
>>> |

>>> select 210
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |   Год   |
+-----+-----+-----+-----+
| 1 |      NameA FamleA    |      Invid     |   2000   |
+-----+-----+-----+-----+
```

Рисунок 1.1 - Результаты добавление список рабочих

```
>>> save list_workkey.txt
```



```
[
  {
    "name": "Name Famle",
    "post": "Lab",
    "year": 2002
  },
  {
    "name": "NameA FamleA",
    "post": "Invid",
    "year": 2000
  }
]
```

Рисунок 1.2 - Сохранение рабочие тип файла JSON

```
>>> list
Список работников пуст.
>>> load list_workey.txt
>>> list
```

№	Ф.И.О.	Должность	Год
1	Name Famle	Lab	2002
2	NameA FamleA	Invid	2000

```
>>>
```

Рисунок 1.3 - Загружает рабочие тип файла JSON

2. Индивидуальные задание 1

```
>>> add
Фамилия и инициалы? UserA
Номер группы? G1
Успеваемость? 3
>>> add
Фамилия и инициалы? UserB
Номер группы? G2
Успеваемость? 7
>>> list
```

№	Ф.И.О.	Номер группы	Успеваемость
1	UserA	G1	3
2	UserB	G2	7

```
>>> save
>>> exit
```

```
{
  "name": "UserA",
  "number": "G1",
  "z": "3"
},
{
  "name": "UserB",
  "number": "G2",
  "z": "7"
}
```

Рисунок 2.1 - Сохранения файла JSON

```
>>> list
```

№	Ф.И.О.	Номер группы	Успеваемость
1	UserA	G1	3
2	UserB	G2	7

```
>>> load
>>> list
```

№	Ф.И.О.	Номер группы	Успеваемость
1	UserA	G1	3
2	UserB	G2	7

Рисунок 2.2 - Загрузка файла JSON

3. Индивидуальные задание 2

```
schema = {  
    "type": "object",  
    "properties": {  
        "name": { "type": "string" },  
        "number": { "type": "number" },  
        "z": { "type": "number" }  
    }  
}
```

Рисунок 3.1 - Схема JSON

```
>>> list  
+-----+-----+-----+-----+  
| № |          Ф.И.О.          | Номер группы | Успеваемость |  
+-----+-----+-----+-----+  
  
>>> load  
  
>>> list  
+-----+-----+-----+-----+  
| № |          Ф.И.О.          | Номер группы | Успеваемость |  
+-----+-----+-----+-----+  
| 1 | TEST                    | 50          | 3            |  
+-----+-----+-----+-----+
```

```
{  
    "name": "TEST",  
    "number": 50,  
    "z": 3  
}
```

Рисунок 3.2 - Загрузка файла правильный схемы JSON

```
>>> list  
+-----+-----+-----+-----+  
| № |          Ф.И.О.          | Номер группы | Успеваемость |  
+-----+-----+-----+-----+  
  
>>> load  
Traceback (most recent call last):  
  
  File "C:\Users\...\Desktop\OIM\lab-16\ Individual_task\individual_2.py", line  
182, in <module>  
    main()  
  
  File "C:\Users\...\Desktop\OIM\lab-16\ Individual_task\individual_2.py", line  
178, in main  
    call_command(command)  
  
  File "C:\Users\...\Desktop\OIM\lab-16\ Individual_task\individual_2.py", line  
44, in call_command  
    return command['fun']()  
  
  File "C:\Users\...\Desktop\OIM\lab-16\ Individual_task\individual_2.py", line  
151, in student_load  
    jsonschema.validate(student, schema=schema)  
  
  File "C:\Users\...\anaconda3\lib\site-packages\jsonschema\validators.py", line  
934, in validate  
    raise error  
  
ValidationError: '50' is not of type 'number'  
  
Failed validating 'type' in schema['properties']['number']:  
    {'type': 'number'}  
  
On instance['number']:
```

```
{  
    "name": "TEST",  
    "number": "50",  
    "z": "3"  
}
```

Рисунок 3.3 - Загрузка файла не правильный схемы JSON