

**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 данными формата JSON в языке Python
по дисциплине «Основы программной инженерии»**

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

Ставрополь, 2022

Пример 1

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import json
import sys
from datetime import date

def get_worker():
    """
    Запросить данные о работнике.
    """
    name = input("Фамилия и инициалы? ")
    post = input("Должность? ")
    year = int(input("Год поступления? "))
    # Создать словарь.
    return {
        'name': name,
        'post': post,
        'year': year,
    }

def display_workers(staff):
    """
    Отобразить список работников.
    """
    # Проверить, что список работников не пуст.
    if staff:
        # Заголовок таблицы.
        line = '+-{}-+-{}-+-{}-+-{}-+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 8
        )
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
                "No",
                "Ф.И.О.",
                "Должность",
                "Год"
            )
        )
        print(line)
        for idx, worker in enumerate(staff, 1):
            print(
                '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
                    idx,
                    worker.get('name', ''),
                    worker.get('post', ''),
                    worker.get('year', 0)
                )
            )
            print(line)
    else:
        print("Список работников пуст.")

def select_workers(staff, period):
    """
    Выбрать работников с заданным стажем.
    """
    # Получить текущую дату.
    today = date.today()
    # Сформировать список работников.
```

```

result = []
for employee in staff:
    if today.year - employee.get('year', today.year) >= period:
        result.append(employee)
# Возвратить список выбранных работников.
return result

def save_workers(file_name, staff):
    """
    Сохранить всех работников в файл JSON.
    """
    # Открыть файл с заданным именем для записи.
    with open(file_name, "w", encoding="utf-8") as fout:
        # Выполнить сериализацию данных в формат JSON.
        # Для поддержки кириллицы установим ensure_ascii=False
        json.dump(staff, fout, ensure_ascii=False, indent=4)
def load_workers(file_name):
    """
    Загрузить всех работников из файла JSON.
    """
    # Открыть файл с заданным именем для чтения.
    with open(file_name, "r", encoding="utf-8") as fin:
        return json.load(fin)
def main():
    """
    Главная функция программы.
    """
    # Список работников.
    workers = []
    # Организовать бесконечный цикл запроса команд.
    while True:
        # Запросить команду из терминала.
        command = input(">>> ").lower()
        # Выполнить действие в соответствии с командой.
        if command == "exit":
            break
        elif command == "add":
            # Запросить данные о работнике.
            worker = get_worker()
            # Добавить словарь в список.
            workers.append(worker)
            # Отсортировать список в случае необходимости.
            if len(workers) > 1:
                workers.sort(key=lambda item: item.get('name', ''))
        elif command == "list":
            # Отобразить всех работников.
            display_workers(workers)
        elif command.startswith("select"):
            # Разбить команду на части для выделения стажа.
            parts = command.split(maxsplit=1)
            # Получить требуемый стаж.
            period = int(parts[1])
            # Выбрать работников с заданным стажем.
            selected = select_workers(workers, period)
            # Отобразить выбранных работников.
            display_workers(selected)
        elif command.startswith("save "):
            parts = command.split(maxsplit=1)
            # Получить имя файла.
            file_name = parts[1]
            # Сохранить данные в файл с заданным именем.
            save_workers(file_name, workers)
        elif command.startswith("load "):
            # Разбить команду на части для выделения имени файла.

```

```

        parts = command.split(maxsplit=1)
        # Получить имя файла.
        file_name = parts[1]
        # Сохранить данные в файл с заданным именем.
        workers = load_workers(file_name)
    elif command == 'help':
        # Вывести справку о работе с программой.
        print("Список команд:\n")
        print("add - добавить работника;")
        print("list - вывести список работников;")
        print("select <стаж> - запросить работников со стажем;")
        print("help - отобразить справку;")
        print("load - загрузить данные из файла;")
        print("save - сохранить данные в файл;")
        print("exit - завершить работу с программой.")
    else:
        print(f"Неизвестная команда {command}", file=sys.stderr)

if __name__ == '__main__':
    main()

```

```

D:\ProgramData\Anaconda3\python.exe C:/Users/admin/Desktop/git/LAB_16/examples/exempl.py
>>> add
Фамилия и инициалы? Сионенко А.С
Должность? Программис
Год поступления? 2022
>>> add
Фамилия и инициалы? тест
Должность? студент
Год поступления? 2023
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |   Год   |
+-----+-----+-----+-----+
|  1 | Сионенко А.С            | Программис          |   2022  |
+-----+-----+-----+-----+
|  2 | тест                    | студент              |   2023  |
+-----+-----+-----+-----+
>>> save list.json
>>> load list.json

```

```

list.json x
1  [
2      {
3          "name": "Сионенко А.С",
4          "post": "Программис",
5          "year": 2022
6      },
7      {
8          "name": "тест",
9          "post": "студент",
10         "year": 2023
11     }
12 ]

```

Сохранение рабочие тип файла JSON

```

>>> load list.json
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |   Год   |
+-----+-----+-----+-----+
|  1 | Сионенко А.С            | Программис          |   2022  |
+-----+-----+-----+-----+
|  2 | тест                    | студент              |   2023  |
+-----+-----+-----+-----+
>>> exit

```

Process finished with exit code 0

Индивидуальные задания 1

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
import json

students = []
count = 0
file_name = "inv_1.json"

def get_command():
    return input(">>> ").lower()

def is_command(is_name_command):
    for command in list_commands:
        if is_name_command == command['name']:
            return True

    return False

```

```

def call_command(in_name_command):
    for command in list_commands:
        if in_name_command == command['name']:
            return command['fun']()

    return False

def student_add():
    name = input("Фамилия и инициалы? ")
    number = input("Номер группы? ")
    z = str(input("Успеваемость? "))

    student = {
        'name': name,
        'number': number,
        'z': z,
    }

    students.append(student)
    if len(students) > 1:
        students.sort(key=lambda item: item.get('name', ''))

def student_print_line():
    print('+-{}-+-{}-+-{}-+-{}-+'.format(
        '-' * 4,
        '-' * 30,
        '-' * 20,
        '-' * 15
    ))

def student_print_list():
    student_print_line()
    print(
        '| {:^4} | {:^30} | {:^20} | {:^15} |'.format(
            "№",
            "Ф.И.О.",
            "Номер группы",
            "Успеваемость"
        )
    )
    student_print_line()
    for idx, worker in enumerate(students, 1):
        print(
            '| {:>4} | {:<30} | {:<20} | {:>15} |'.format(
                idx,
                worker.get('name', ''),
                worker.get('number', ''),
                worker.get('z', 0)
            )
        )

    student_print_line()

def student_print_select():
    cn = 0
    for student in students:
        if "2" in student.get('z', ''):
            cn += 1
            print(
                '{:>4} {}'.format('*', student.get('name', '')),
                '{:>1} {}'.format('группа №', student.get('number', ''))
            )

```

```

    if cn == 0:
        print('Таких студентов нет')

def help_print_list():
    print("Список команд:")
    print("\texit - завершить работу с программой;")
    for command in list_commands:
        print(f"\t{command['name']} - {command['desc']}")

def student_save():
    with open(file_name, "w", encoding="utf-8") as TextIO:
        list_item = []
        for item in students:
            list_item.append(
                {
                    'name': item["name"],
                    'number': item["number"],
                    'z': item["z"]
                }
            )

        json.dump(list_item, TextIO, ensure_ascii=False, indent=4)

def student_load():
    global students

    with open(file_name, "r", encoding="utf-8") as fin:
        res = json.load(fin)
        students = []

        for item in res:
            student = {
                'name': item["name"],
                'number': item["number"],
                'z': item["z"],
            }

            students.append(student)
            if len(students) > 1:
                students.sort(key=lambda item: item.get('name', ''))

list_commands = (
    {'name': 'help', 'fun': help_print_list, 'desc': 'добавить студента'},
    {'name': 'add', 'fun': student_add, 'desc': 'вывести список студентов'},
    {'name': 'select', 'fun': student_print_select, 'desc': 'вывести список студентов, имеющих оценку 2'},
    {'name': 'list', 'fun': student_print_list, 'desc': 'отобразить справку'},
    {'name': 'save', 'fun': student_save, 'desc': 'сохранить'},
    {'name': 'load', 'fun': student_load, 'desc': 'загрузить'}
)

def main():
    while True:
        command = get_command()

        if command == "exit":
            break

        if command:
            if is_command(command):
                call_command(command)

            else:
                print(f"Неизвестная команда {command}", file=sys.stderr)

```

```
if __name__ == '__main__':  
    main()
```

```
D:\ProgramData\Anaconda3\python.exe C:/Users/admin/Desktop/git/LAB_16/Individual/_individual_1_.py  
>>> add  
Фамилия и инициалы? Симоненко А.С  
Номер группы? 44  
Успеваемость? 2  
>>> add  
Фамилия и инициалы? тест  
Номер группы? 38  
Успеваемость? 6  
>>> list  
+-----+-----+-----+-----+  
| № |          Ф.И.О.          | Номер группы | Успеваемость |  
+-----+-----+-----+-----+  
|  1 | Симоненко А.С          | 44          | 2          |  
|  2 | тест                    | 38          | 6          |  
+-----+-----+-----+-----+  
>>> save  
>>> Неизвестная команда save  
  
>>> save  
>>> exit
```

Сохранения файла JSON

```
D:\ProgramData\Anaconda3\python.exe C:/Users/admin/Desktop/git/LAB_16/Individual/_individual_1_.py  
>>> load  
>>> list  
+-----+-----+-----+-----+  
| № |          Ф.И.О.          | Номер группы | Успеваемость |  
+-----+-----+-----+-----+  
|  1 | Симоненко А.С          | 44          | 2          |  
|  2 | тест                    | 38          | 6          |  
+-----+-----+-----+-----+  
>>>
```

Загрузка файла JSON