МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.16

Работа сданными формата JSON в языке Python.

Выполнил студент группы

ИВТ-б-о-21-1

Богдашов А.В « »_____20__г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил доцент
Кафедры инфокоммуникаций, старший преподаватель
Воронкин Р.А.

_____
(подпись)

Ставрополь 2022

Цель работы: приобретение навыков по работе с данными формата JSON с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

Индивидуальное задание:

Код программы:

```
# !/usr/bin/env python3
import sys
import json
import jsonschema


def add(list_stud):
    """
    Запросить данные о студентах.
    """
    name = input("Фамилия и инициалы? ")
    number = input("Номер группы? ")
    marks = list(map(int, input("Успеваемость").split()))
    # Создать словарь.
    student = {
        "name": name,
        "number": number,
        "marks": marks,
    }
    # Добавить словарь в список.
    list_stud.append(student)
    # Отсортировать список в алфавитном порядке.
    if len(list_stud) > 1:
        list_stud.sort(key=lambda item: item.get("name", ""))
    return list_stud


def list_p():
    """
    Отабразить список работников
    """
    line = "+-{}-+-{}-+-{}-+-{}-+".format("-" * 4, "-" * 30, "-" * 20, "-" * 15)
    print(line)
    print(
        "| {:^4} | {:^30} | {:^20} | {:^15} |".format(
            "№", "Ф.И.О.", "Номер группы", "Успеваемость"
        )
    )
    print(line)
    # Вывести данные о всех студентах.
    for idx, worker in enumerate(students, 1):
        print(
            "| {:>4} | {:<30} | {:<20} | {:>15} |".format(
                idx,
                worker.get("name", ""),
                worker.get("number", ""),
                " ".join([str(x) for x in worker.get("marks", 0)]),
            )
        )
    print(line)
```

```
def select():
    """
    Выбрать студентов с оценками 2
    """
    # Инициализировать счётчик
    count = 0
    # Проверить студентов хотя бы на одну оценку.
    for student in students:
        if 2 in student.get("marks", []):
            count -= 1
            print(
                "{:>4} {}".format("*", student.get("name", "")),
                "{:>1} {}".format("группа №", student.get("number", "")),
            )
            # Если счётчик равен 0, то оценки не найдены.
    if count == 0:
        print("Таких студентов нет")


def save_students(file_name, students):
    with open(file_name, "w", encoding="utf-8") as fout:
        json.dump(students, fout, ensure_ascii=False, indent=4, default=str)


def load_students(file_name):
    schema = {
        "type": "array",
        "items": [
            {
                "type": "object",
                "properties": {
                    "name": {"type": "string"},
                    "number": {"type": "string"},
                    "marks": {
                        "type": "array",
                        "items": [
                            {"type": "integer"},
                            {"type": "integer"},
                            {"type": "integer"},
                            {"type": "integer"},
                            {"type": "integer"},
                        ],
                    },
                },
                "required": ["name", "number", "marks"],
            }
        ],
    }
    with open(file_name, "r", encoding="utf-8") as fin:
        loadfile = json.load(fin)
        validator = jsonschema.Draft7Validator(schema)
        try:
            if not validator.validate(loadfile):
                print("Валидация прошла успешно")
        except jsonschema.exceptions.ValidationError:
            print("Ошибка валидации", file=sys.stderr)
            exit()
        return loadfile


def help_d():
    """
    Отабразить список команд
    """
    print("Список команд:\n")
    print("add - добавить студента;")
    print("list - вывести список студентов;")
```

```
    print("select - вывести список студентов, имеющих оценку 2;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")


if __name__ == "__main__":
    # Список студентов.
    students = []
    # Организовать бесконечный цикл запроса команд.
    while True:
        # Запросить команду из терминала.
        command = input(">>> ").lower()
        # Выполнить действие в соответствие с командой.
        if command == "exit":
            break
        elif command == "add":
            students = add(students)
        elif command == "list":
            list_p()
        elif command == "select":
            select()
        elif command.startswith("save "):
            parts = command.split(maxsplit=1)
            file_name = parts[1]
            save_students(file_name, students)
        elif command.startswith("load "):
            parts = command.split(maxsplit=1)
            file_name = parts[1]
            students = load_students(file_name)
        elif command == "help":
            help_d()
        else:
            print(f"Неизвестная команда {command}", file=sys.stderr)
```

```
>>> add
Фамилия и инициалы? Богдашов Артём Владимирович
Номер группы? ИВТ-6-о-21-1
Успеваемость5 4 3 3 4
>>> save ind.json
>>> load ind.json
Валидация прошла успешно
>>> list
+------+--------------------------------+---------------------+-----------------+
|  №   |            Ф.И.О.              |     Номер группы    |  Успеваемость   |
+------+--------------------------------+---------------------+-----------------+
|    1 | Богдашов Артём Владимирович    | ИВТ-6-о-21-1        |       5 4 3 3 4 |
+------+--------------------------------+---------------------+-----------------+
>>>
```

Рисунок1. Результат

Ответы на контрольные вопросы:

1. Для чего используется JSON?

JSON – текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми.

2. Какие типы значений используются в JSON?

Набор пар ключ: значение. Упорядоченный набор значений.

3. Как организована работа со сложными данными в JSON?

JSON может содержать другие вложенные объекты в JSON, в дополнение к вложенным массивам. Такие объекты и массивы будут передаваться, как значения, назначенные ключам, и будут представлять собой связку ключ-значение.

4. Самостоятельно ознакомьтесь с форматом данных JSON5. В чем отличие этого формата?

Формат обмена данными JSON5 (JSON5) — это надмножество JSON, целью которого является смягчение некоторых ограничений JSON путем расширения его синтаксиса для включения некоторых продуктов из ECMAScript 5.1. Эта библиотека JavaScript является официальной эталонной реализацией библиотек синтаксического анализа и сериализации JSON5. Краткое описание возможностей. Следующие функции ECMAScript 5.1, которые не поддерживаются в JSON, были расширены до JSON5. Объекты. Ключи объекта могут быть идентификатором ECMAScript 5.1

5. Какие средства языка программирования Python могут быть использованы для работы с данными в формате JSON5?

Реализация Python формата данных JSON5. JSON5 расширяет формат обмена данными JSON, чтобы сделать его более удобным для использования в качестве языка конфигурации: Комментарии в стиле JavaScript (как однострочные, так и многострочные) разрешены. Ключи объектов могут быть без кавычек, если они являются допустимыми идентификаторами ECMAScript. Объекты и массивы могут заканчиваться запятыми. Строки могут заключаться в одинарные кавычки, допускаются много строчные строковые литералы. Есть еще несколько более мелких расширений JSON; см. полную информацию на странице выше. Этот проект реализует реализацию чтения и записи для Python; где воз можно, он отражает стандартный пакет Python JSON API для простоты использования. Есть одно заметное отличие от JSON api: методы load () и load () поддерживают опциональную проверку (и отклонение) повторяющихся ключей объекта; pass allow_duplicate_keys = False для этого (по умолчанию разрешены дубликаты). Это ранний выпуск. Это было достаточно хорошо протестировано, но это МЕДЛЕННО. Он может быть в 1000-6000 раз медленнее, чем модуль JSON, оптимизированный для C, и в 200 раз (или более) медленнее, чем модуль JSON на чистом Python

6. Какие средства предоставляет Python для сериализации данных в формате JSON?

json.dump() # конвертировать python объект в json и записать в файл json.dumps() # тоже самое, но в строку.

7. В чем отличие методов json.dump() и json.dumps()?

Dumps записывает в строку, а dump в файл.

8. Какие средства предоставляет Python для десериализации данных JSON?

json.load() # прочитать json из файла и конвертировать в python объект json.loads() # тоже самое, но из строки с json (s на конце от string/строка)

9. Какие средства необходимо использовать для работы с данными формата JSON, содержащими кириллицу?

import codecs json.load(codecs.open('sample.json', 'r', 'utf-8-sig'))

10. Самостоятельно ознакомьтесь со спецификацией JSON Schema? Что такое схема данных?

Схема JSON — это словарь, который позволяет аннотировать и проверять

документы JSON.

Преимущества:

• Описывает ваш существующий формат (ы) данных.

• Предоставляет понятную документацию, читаемую человеком и

машиной.

• Проверяет данные, которые полезны для:

• Автоматизированное тестирование.

• Обеспечение качества предоставленных клиентом данных

Вывод: в ходе работы были приобретены навыки по работе с данными формата JSON с помощью языка программирования Python версии 3.x.