

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №19
дисциплины «Основы программной инженерии»

Выполнил:
Плугатырев Владислав Алексеевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Работа с файлами в языке Python. Виртуальные окружения.

Цель работы: приобретение навыков при работе с данными в формате JSON при написании программ с помощью языка программирования Python версии 3.x, изучение основных методов модуля os для работы с файловой системой, получение аргументов командой строки.

Ход работы.

1. Создание нового репозитория с лицензией MIT.

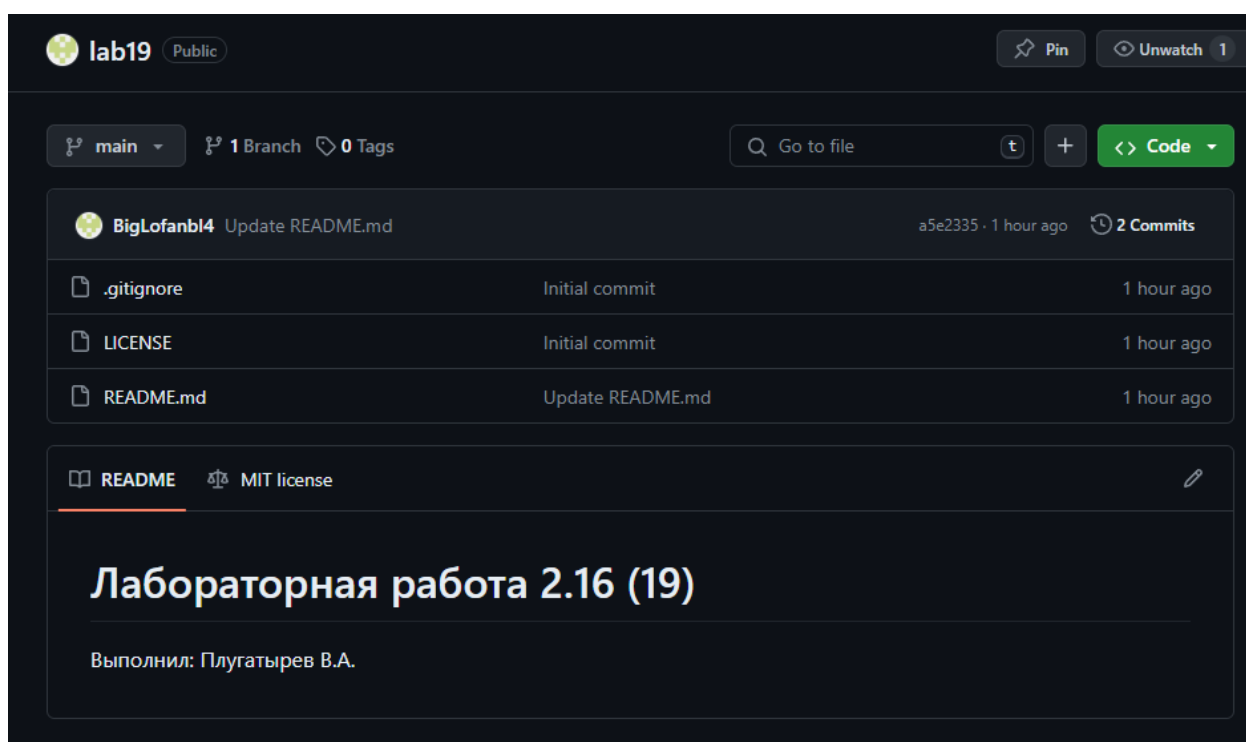


Рисунок 1.1 – Созданный репозиторий

2. Проработал примеры из лабораторной работы.

```
89 def main():
90     """
91     Главная функция программы.
92     """
93     # Список работников.
94     workers = []
95     # Организовать бесконечный цикл запроса команд.
96     while True:
97         # Запросить команду из терминала.
98         command = input(">>> ").lower()
99         # Выполнить действие в соответствие с командой.
100         if command == "exit":
101             break
102         elif command == "add":
103             # Запросить данные о работнике.
104             worker = get_worker()
105             # Добавить словарь в список.
106             workers.append(worker)
107             # Отсортировать список в случае необходимости.
108             if len(workers) > 1:
109                 workers.sort(key=lambda item: item.get("name", ""))
110         elif command == "list":
111             # Отобразить всех работников.
112             display_workers(workers)
113         elif command.startswith("select "):
114             # Разбить команду на части для выделения стажа.
115             parts = command.split(maxsplit=1)
116             # Получить требуемый стаж.
117             period = int(parts[1])
118             # Выбрать работников с заданным стажем.
119             selected = select_workers(workers, period)
120             # Отобразить выбранных работников.
121             display_workers(selected)
122         elif command.startswith("save "):
123             # Разбить команду на части для выделения имени файла.
124             parts = command.split(maxsplit=1)
125             # Получить имя файла.
126             file_name = parts[1]
127             # Сохранить данные в файл с заданным именем.
128             save_workers(file_name, workers)
129         elif command.startswith("load "):
130             # Разбить команду на части для выделения имени файла.
131             parts = command.split(maxsplit=1)
132             # Получить имя файла.
133             file_name = parts[1]
134             # Сохранить данные в файл с заданным именем.
135             workers = load_workers(file_name)
136         elif command == "help":
137             # Вывести справку о работе с программой.
138             print("Список команд:\n")
139             print("add - добавить работника;")
140             print("list - вывести список работников;")
141             print("select <стаж> - запросить работников со стажем;")
142             print("help - отобразить справку;")
143             print("load - загрузить данные из файла;")
144             print("save - сохранить данные в файл;")
145             print("exit - завершить работу с программой.")
146         else:
147             print(f"Неизвестная команда {command}", file=sys.stderr)
148
149
150 if __name__ == "__main__":
151     main()
152
```

Рисунок 2.1 – Код примера 1

```

PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\19\lab19> python example1.py
>>> add
Фамилия и инициалы? Plugatyrev V.A.
Должность? Student
Год поступления? 2022
>>> add
Фамилия и инициалы? Chickodan A.Y.
Должность? student
Год поступления? 2021
>>> save staff.json
>>> exit
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\19\lab19> & C:/Users/vladi/AppData/Local/Programs/Python/
adi/OneDrive/Рабочий стол/Основы программной инженерии/19/lab19/example1.py"
>>> list
Список работников пуст.
>>> load staff.json
>>> list

```

№	Ф.И.О.	Должность	Год
1	Chickodan A.Y.	student	2021
2	Plugatyrev V.A.	Student	2022

Рисунок 2.2 – Вывод программы из примера 1

3. Индивидуальное задание.

```

79 def main():
80     """
81     Главная функция программы.
82     """
83     people = []
84
85     while True:
86         command = input("Введите команду (add, info, list, load, save, exit, help): ").strip().lower().split(maxsplit=1)
87
88         match command[0]:
89             case "exit":
90                 break
91
92             case "load":
93                 file_name = command[1]
94                 people = load_people(file_name)
95
96             case "save":
97                 file_name = command[1]
98                 save_people(file_name, people)
99
100            case "add":
101                person = get_person()
102                people.append(person)
103                people.sort(
104                    key=lambda x: datetime.strptime(".".join(x["birthday"]), "%d.%m.%Y")
105                )
106
107            case "info":
108                surname = input("Введите фамилию: ")
109                selected = select_people(surname, people)
110                display_people(selected)
111
112            case "list":
113                display_people(people)
114
115            case "help":
116                get_instructions()
117
118            case _:
119                print(f"Неизвестная команда {command[0]}", file=sys.stderr)
120
121 if __name__ == "__main__":
122     main()

```

Рисунок 3.1 – Код программы

```

PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\19\lab19> python ex1.py
Введите команду (add, info, list, load, save, exit, help): add
Введите фамилию: Plugatyrev
Введите имя: Vladislav
Введите знак зодиака: Capicorn
Дата рождения (число.месяц.год):12.01.2005
Введите команду (add, info, list, load, save, exit, help): save people.json
Введите команду (add, info, list, load, save, exit, help): add
Введите фамилию: Plugatyrev
Введите имя: Alexey
Введите знак зодиака: Scorpion
Дата рождения (число.месяц.год):26.10.1977
Введите команду (add, info, list, load, save, exit, help): save people.json
Введите команду (add, info, list, load, save, exit, help): exit
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\19\lab19> python ex1.py
Введите команду (add, info, list, load, save, exit, help): list
Список пуст
Введите команду (add, info, list, load, save, exit, help): load people.json
Введите команду (add, info, list, load, save, exit, help): list
+-----+-----+-----+-----+-----+
| № | Фамилия | Имя | Знак зодиака | Дата рождения |
+-----+-----+-----+-----+-----+
| 1 | Plugatyrev | Alexey | Scorpion | 26.10.1977 |
| 2 | Plugatyrev | Vladislav | Capicorn | 12.01.2005 |
+-----+-----+-----+-----+-----+
Введите команду (add, info, list, load, save, exit, help): exit

```

Рисунок 3.2 – Вывод программу

4. Задание повышенной сложности.

```

3
4 import sys
5 import json
6 from jsonschema import validate
7 from jsonschema.exceptions import ValidationError
8 from datetime import datetime
9
10
11 def validation(instance):
12     schema = {
13         "type": "array",
14         "items": {
15             "type": "object",
16             "properties": {
17                 "surname": {"type": "string"},
18                 "name": {"type": "string"},
19                 "zodiac": {"type": "string"},
20                 "birthday": {
21                     "type": "array",
22                     "items": {"type": "string"},
23                     "minitems": 3,
24                 },
25             },
26             "required": ["surname", "name", "zodiac", "birthday"]
27         },
28     }
29     try:
30         validate(instance, schema=schema)
31         return True
32     except ValidationError as err:
33         print(err.message)
34         return False
35
36
37 def load_people(file_name):
38     with open(file_name, "r") as f:
39         people = json.load(f)
40
41     if validation(people):
42         return people
43

```

Рисунок 4.1 – Реализация валидации

```
(lab19) C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\19\lab19>python hard.py
Введите команду (add, info, list, load, save, exit, help): list
Список пуст
Введите команду (add, info, list, load, save, exit, help): load people.json
Введите команду (add, info, list, load, save, exit, help): list
```

%	Фамилия	Имя	Знак зодиака	Дата рождения
1	Chickodan	Alexey	asd	08.08.2003
2	Plugatyrev	Vladislav	Capicorn	12.01.2005

```
Введите команду (add, info, list, load, save, exit, help): exit
```

Рисунок 4.2 – Результат

Ответы на контрольные вопросы

1. Для чего используется JSON?

JSON (JavaScript Object Notation) используется для хранения и передачи данных между сервером и веб-приложением. Благодаря своей легко читаемой форме для людей и одновременно простой интерпретации для машин, JSON стал одним из наиболее популярных форматов для обмена данными, особенно в контексте веб-разработки.

2. Какие типы значений используются в JSON?

В JSON используются следующие типы данных:

- Строки (string)
- Числа (number)
- Объекты (object)
- Массивы (array)
- Булевы значения (true/false)
- Null

3. Как организована работа со сложными данными в JSON?

Сложные данные представляются в JSON в виде вложенных объектов (свойства-значения) и массивов (упорядоченных списков значений). Объекты могут содержать другие объекты, массивы и значения примитивных типов, тем самым создавая многоуровневые структуры данных.

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

JSON5 - это надмножество JSON, созданное для упрощения работы разработчиков. Оно поддерживает некоторые дополнительные функции ECMAScript 5:

- Ключи могут не заключаться в кавычки, если они являются допустимыми идентификаторами ES5.
- Можно использовать одинарные кавычки для строк.
- Строки могут быть многострочными.
- Могут использоваться комментарии.
- Дополнительные средства числовых литералов, например, начальные нули, трейлинг comma и другие.

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

В Python для работы с данными в формате JSON5 можно использовать стороннюю библиотеку ``json5``, которая доступна для установки через ``pip install json5``.

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

Python предоставляет стандартный модуль ``json``, который содержит функции ``json.dump()`` для записи данных в файл и ``json.dumps()`` для сериализации данных в строку JSON.

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

Функция ``json.dump()`` сериализует объект Python и записывает его в файл. В то время как ``json.dumps()`` сериализует объект Python и возвращает строку в формате JSON.

8. Какие средства предоставляет язык Python для десериализации данных из формата JSON?

Стандартный модуль ``json`` также включает функции ``json.load()`` для чтения и десериализации данных из файла и ``json.loads()`` для десериализации данных из строки формата JSON.

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

При сериализации данных в JSON, содержащих кириллицу, важно установить параметр ``ensure_ascii`` в ``False``, таким образом символы кириллицы не будут экранированы. Например: ``json.dumps(data, ensure_ascii=False)``.

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

Схема данных - это описание структуры данных, чаще всего используется для валидации формата JSON. Схема указывает, какие поля должны быть присутствовать, их типы, форматы и другие ограничения.