

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет о лабораторной работе №5
по дисциплине основы программной инженерии

Выполнил: Духно Михаил
Александрович,

2 курс, группа ПИЖ-б-о-20-1,

Проверил: Доцент кафедры
инфокоммуникаций, Воронкин Р.А.

Ставрополь, 2022 г

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

import pandas as pd
import json
from jsonschema import validate
from pathlib import Path
import click

FILE_NAME = 'json_file.json'
SETTINGS_FILE = 'settings.json'

@click.group()
def main():
    pass

@main.command()
def add_element():
    name = input('Конечный пункт: ')
    num = input('Номер поезда: ')
    tm = input('Время отправления: ')
    trains = {}
    trains['name'] = name
    trains['num'] = int(num)
    trains['tm'] = tm
    schema = """
with open('schema.json', 'r') as f:
    schema = json.loads(f.read())
validate(instance=trains, schema=schema)
with open(FILE_NAME, 'a') as f:
    f.write(json.dumps(trains) + '\n')

@main.command()
@click.argument('num')
def find_train(num):
    with open('json_file.json', 'r') as f:
        trains = f.readlines()
        for dcts in trains:
```

```

dcts = json.loads(dcts)
if dcts['num'] == int(num):
    click.echo(
        f'Конечный пункт: {dcts["name"]} \n'
        f'Номер поезда: {dcts["num"]} \n'
        f'Время отправления: {(dcts["tm"])}'
    )
    return
click.echo('Поезда с таким номером нет')

if __name__ == '__main__':
    print('LOADING...')
    with open(SETTINGS_FILE, 'r') as f:
        settings = json.loads(f.read())
        if settings['gitignore'] == False:
            path = Path(__file__).resolve()
            print(path.parents[1])
            par_path = path.parents[1]
            with open(str(par_path) + "\\HYPERLINK \"file://.gitignore/\".gitignore', 'a') as gig:
                gig.write('\n' + '*.json' + '\n')
    with open(SETTINGS_FILE, 'w') as f:
        f.write(json.dumps({'gitignore': True}))

    print('Hello!')
    flag = True
    while flag:
        main()
        """
        print('1. Добавить новый поезд')
        print('2. Вывести информацию о поезде')
        print('3. Выход из программы')
        com = int(input('введите номер команды: '))
        if com == 1:
            add_element()
        elif com == 2:
            train_num = input('Введите номер поезда: ')
            find_train(train_num)
        elif com == 3:
            flag = False
        """

```

Контрольные вопросы:

1. В чем отличие терминала и консоли?

консоль это программа, которая принимает от пользователя кнопки, а от соединения с компьютером — буквы на экран и спец. команды по рисованию картинки на экране. Терминал — обобщённое название как программы, так и специальной железки

2. Что такое консольное приложение?

Консольное приложение - это компьютерная программа, предназначенная для использования через текстовый компьютерный интерфейс, такой как текстовый терминал, интерфейс командной строки некоторых операционных систем или текстовый интерфейс, включенный в большинство операционных систем с графическим пользовательским интерфейсом

3. Какие существуют средства языка программирования Python для построения приложений

командной строки?

`sys`, `getopt`, `argparse`

4. Какие особенности построение CLI с использованием модуля `sys` ?

Он использует подход,

очень похожий на библиотеку C, с использованием `argc` и `argv` для доступа к аргументам.

Модуль `sys` реализует аргументы командной строки в простой структуре списка с именем

`sys.argv` . Каждый элемент списка представляет собой единственный аргумент.

5. Какие особенности построение CLI с использованием модуля `getopt` ?

Модуль `getopt` в Python идет немного дальше и расширяет разделение

входной строки проверкой параметров. Основанный на функции C `getopt` , он позволяет

использовать как короткие, так и длинные варианты, включая присвоение значений.

На практике для правильной обработки входных данных требуется модуль `sys` . Для этого

необходимо заранее загрузить как модуль `sys` , так и модуль `getopt` . Затем из списка входных параметров мы удаляем первый элемент списка (см. код ниже) и сохраняем оставшийся список аргументов командной строки в переменной с именем `arguments_list` .

6. Какие особенности построение CLI с использованием модуля `argparse` ?

Для начала работы с `argparse` необходимо задать парсер:

Далее, парсеру стоит указать, какие объекты Вы от него ждете. В частном случае, это может выглядеть так:

Если действие (action) для данного аргумента не задано, то по умолчанию он будет сохраняться (store) в namespace, причем мы также можем указать тип этого аргумента (int, boolean и тд).

Если имя возвращаемого аргумента (dest) задано, его значение будет сохранено в соответствующем атрибуте namespace.