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 №20

дисциплины «Основы программной инженерии»

Выполнил:

Рядская Мария Александровна

2 курс, группа ПИЖ-б-о-22-1,

09.03.04 «Программная инженерия»,

направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения

Проверил Воронкин Роман Александрович

Ставрополь, 2024 г.

Тема: Лабораторная работа 2.17. Разработка приложений с интерфейсом

командной строки (CLI) в Python3

Цель работы: приобретение построения приложений с интерфейсом

командной строки с помощью языка программирования Python версии 3.x.

Ход работы.

1. Создание нового репозитория с лицензией MIT.

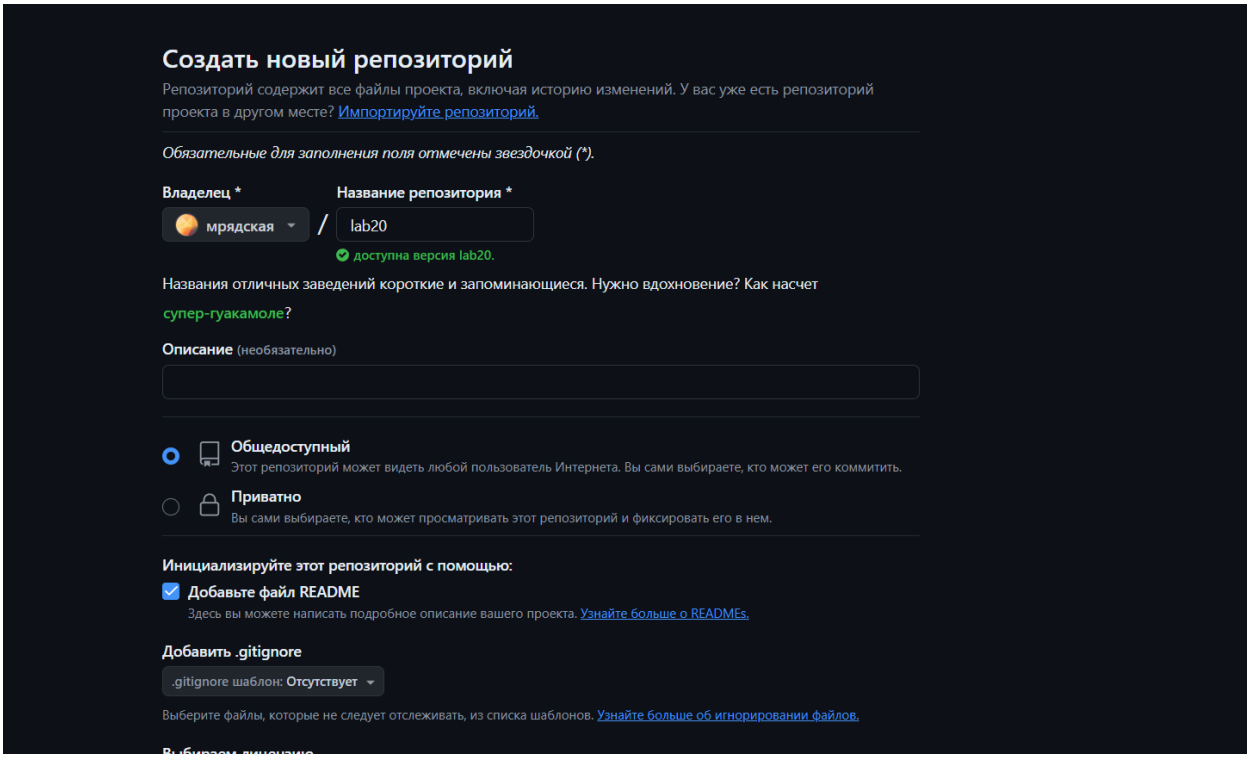


Рисунок 1 – создание репозитория

2. Клонировал репозиторий на рабочий ПК.

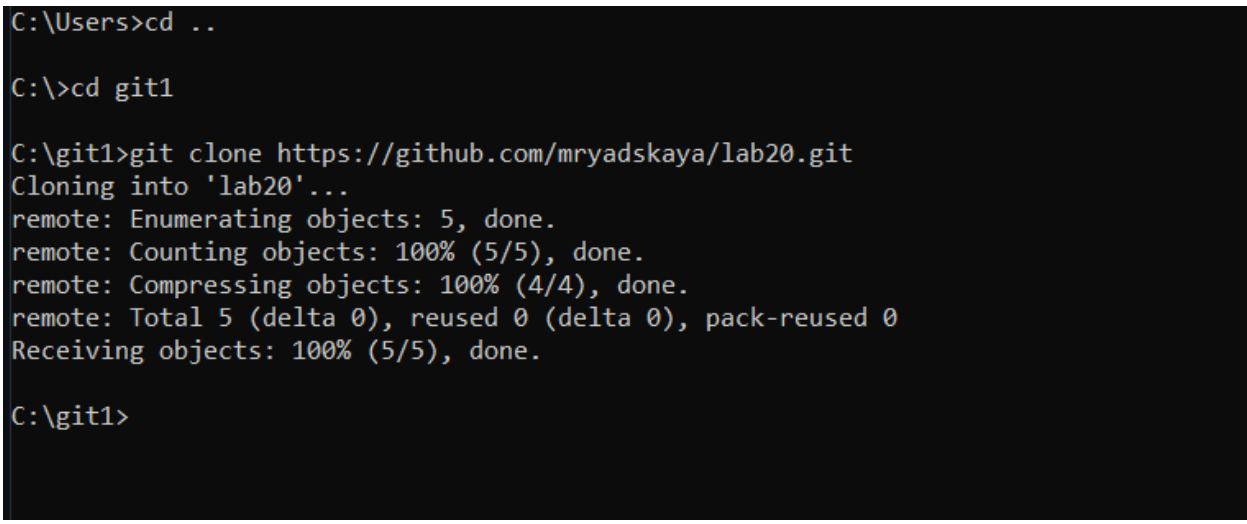


Рисунок 2 – клонирование репозитория

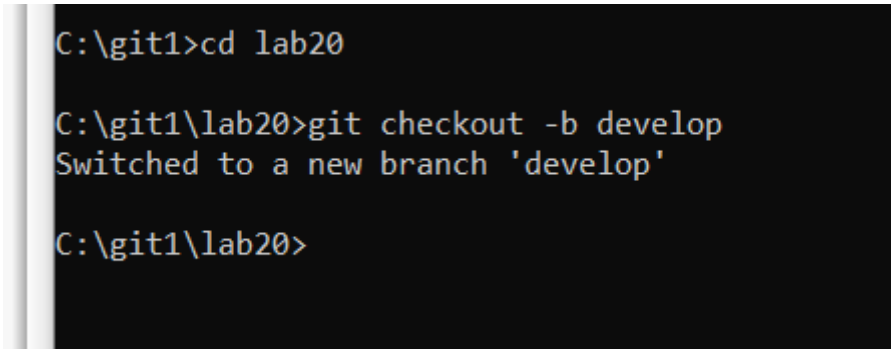


Рисунок 3 – организация ветки

4. Создала виртуальное окружение и установила нужные пакеты.

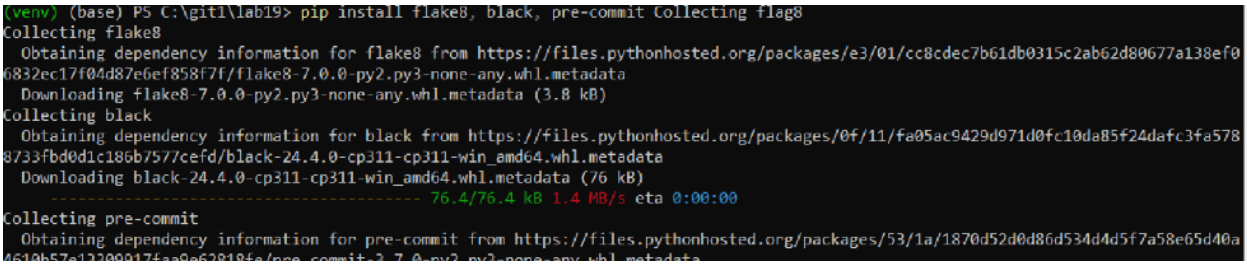
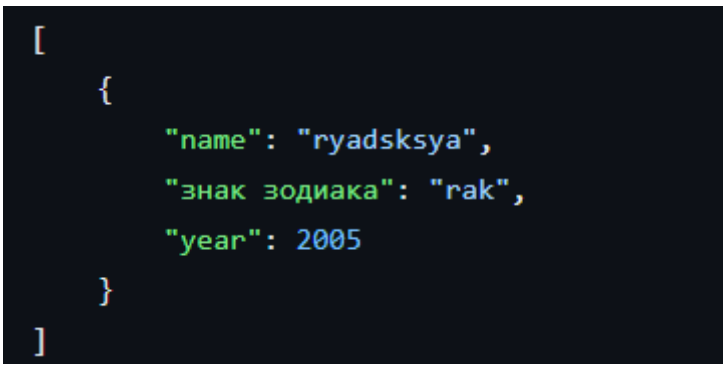
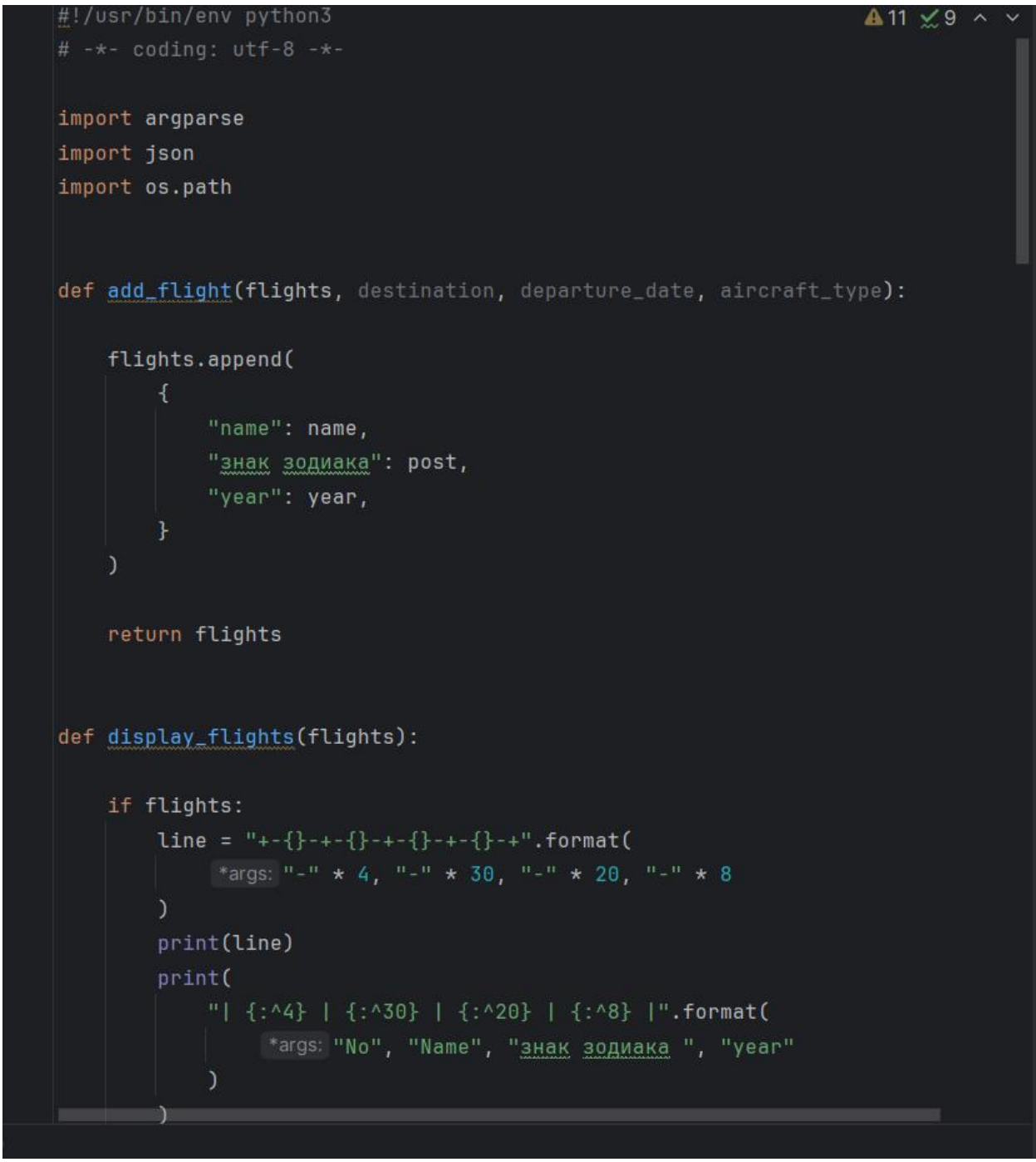


Рисунок 5 - создание виртуального окружения

5.Проработала примеры из лабораторной работы.



6. Для своего варианта лабораторной работы 2.16 необходимо дополнительно реализовать интерфейс командной строки (CLI).



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

import argparse
import json
import os.path


def add_flight(flights, destination, departure_date, aircraft_type):

    flights.append(
        {
            "name": name,
            "знак зодиака": post,
            "year": year,
        }
    )

    return flights


def display_flights(flights):

    if flights:
        line = "+-{}-+-{}-+-{}-+-{}-+".format(
            *args: "-" * 4, "-" * 30, "-" * 20, "-" * 8
        )
        print(line)
        print(
            "| {:^4} | {:^30} | {:^20} | {:^8} |".format(
                *args: "No", "Name", "знак зодиака ", "year"
            )
        )
```

Рисунок 8 – код для реализации задания

9. Самостоятельно изучите работу с пакетом click для построения

интерфейса командной строки (CLI). Для своего варианта лабораторной

работы 2.16 необходимо реализовать интерфейс командной строки с

использованием пакета click .

Рисунок 9 – код для выполнения задания повышенной сложности

Контрольные вопросы:

1.Отличие терминала и консоли:

Терминал (или терминальное окно) - это программа, которая

предоставляет пользователю интерфейс для взаимодействия с операционной

системой через текстовый интерфейс. Терминал обычно предоставляет доступ

к командной строке, где пользователь может вводить команды для

выполнения различных задач.

Консоль - это текстовый интерфейс, предоставляемый операционной

системой для взаимодействия с пользователем или другими программами. Он

может быть встроенным в окружение рабочего стола или виртуальной

машины. Консоль обычно работает поверх терминала и обеспечивает вывод

текстовой информации и ввод команд пользователя.

2.Консольное приложение:

Консольное приложение - это приложение, которое работает в

текстовом режиме и взаимодействует с пользователем через командную

строку или терминал. Оно обычно используется для выполнения задач в

операционной системе или автоматизации определенных процессов. Примеры

консольных приложений включают текстовые редакторы, утилиты командной

строки и скрипты.

3.Средства Python для построения приложений командной строки:

Модуль sys: для доступа к аргументам командной строки и другим

системным параметрам.

Модуль getopt: для парсинга аргументов командной строки в стиле

POSIX.

Модуль argparse: для создания гибких и мощных интерфейсов

командной строки с поддержкой подпрограмм, позиционных и именованных

аргументов, справочной информации и др.

4.Особенности построения CLI с использованием модуля sys:

Модуль sys предоставляет базовые инструменты для работы с

аргументами командной строки, доступом к системным переменным и

другими системными функциями.

С его помощью можно получить доступ к аргументам командной

строки через список sys.argv и осуществить простой парсинг аргументов.

5.Особенности построения CLI с использованием модуля getopt:

Модуль getopt позволяет парсить аргументы командной строки в стиле

POSIX с короткими и длинными опциями.

Он предоставляет более гибкие возможности по обработке аргументов,

чем модуль sys, но требует более сложного кода для использования.

6.Особенности построения CLI с использованием модуля argparse:

Модуль argparse предоставляет мощные средства для создания гибких

интерфейсов командной строки.

Он поддерживает позиционные и именованные аргументы,

подпрограммы, группы аргументов, справочную информацию и многое

другое.

argparse обеспечивает автоматическую проверку типов аргументов,

генерацию справки и сообщений об ошибках.