

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 №6
дисциплины «Программирование на языке Python»

Вариант_15_

Выполнила:
Маньшина Дарья Алексеевна
2 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., канд. тех. наук,
доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

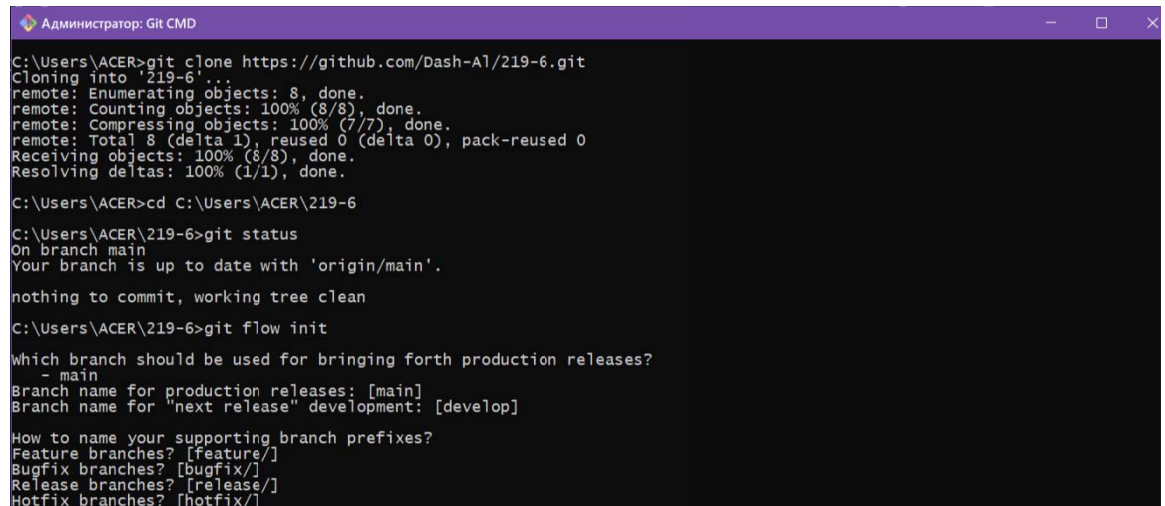
Ставрополь, 2023 г.

Тема: Работа с файловой системой в Python3 с использованием модуля pathlib

Цель: приобретение навыков по работе с файловой системой с помощью библиотеки pathlib языка программирования Python версии 3.x.

Ход работы:

1. Создадим новый репозиторий. Клонировать его и сделаем способ ветвления git-flow. Также установим пакеты: black, isort, flake8



```
Администратор: Git CMD
C:\Users\ACER>git clone https://github.com/Dash-A1/219-6.git
Cloning into '219-6'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (1/1), done.

C:\Users\ACER>cd C:\Users\ACER\219-6

C:\Users\ACER\219-6>git status
On branch main
Your branch is up to date with 'origin/main'.

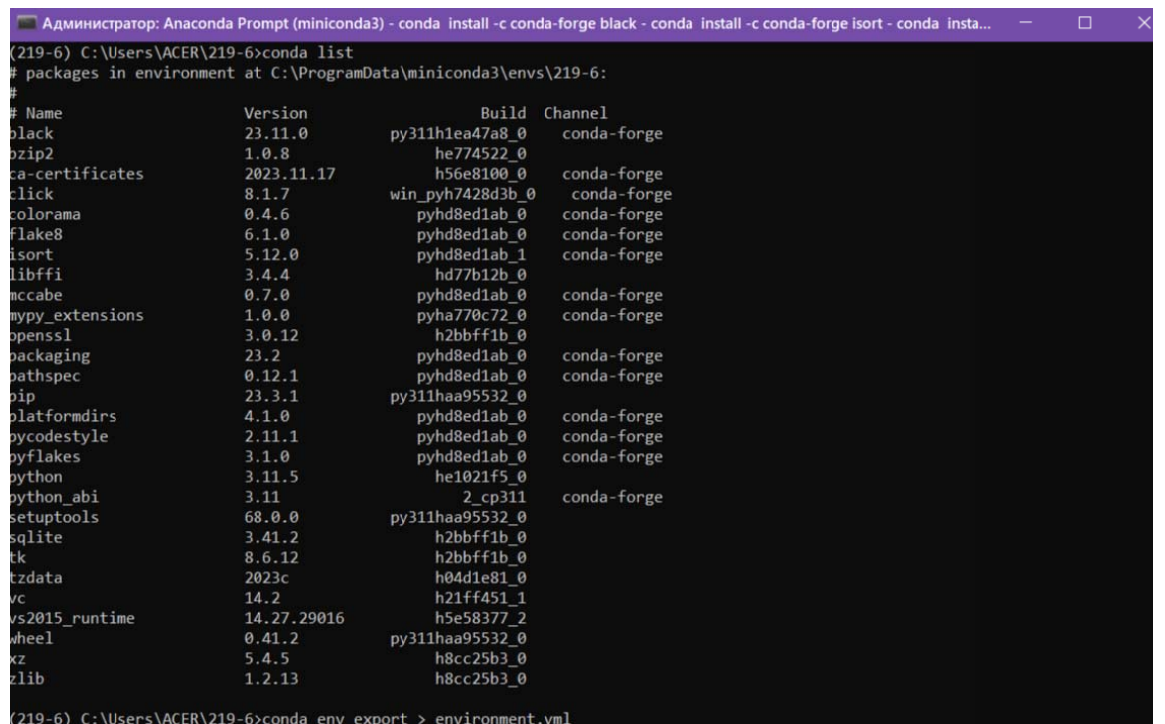
nothing to commit, working tree clean

C:\Users\ACER\219-6>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
```

Рисунок №1. Клонирование репозитория



```
Администратор: Anaconda Prompt (miniconda3) - conda install -c conda-forge black - conda install -c conda-forge isort - conda install -c conda-forge flake8

(219-6) C:\Users\ACER\219-6>conda list
# packages in environment at C:\ProgramData\miniconda3\envs\219-6:
#
# Name                    Version            Build                Channel
black                     23.11.0            py311h1ea47a8_0     conda-forge
bzip2                     1.0.8              he774522_0          conda-forge
ca-certificates           2023.11.17         h56e8100_0          conda-forge
click                     8.1.7              win_pyh7428d3b_0    conda-forge
colorama                  0.4.6              pyhd8ed1ab_0        conda-forge
flake8                    6.1.0              pyhd8ed1ab_0        conda-forge
isort                      5.12.0             pyhd8ed1ab_1        conda-forge
libffi                     3.4.4              hd77b12b_0          conda-forge
mccabe                    0.7.0              pyhd8ed1ab_0        conda-forge
mypy_extensions           1.0.0              pyha770c72_0        conda-forge
openssl                   3.0.12             h2bbff1b_0          conda-forge
packaging                  23.2               pyhd8ed1ab_0        conda-forge
pathspec                   0.12.1             pyhd8ed1ab_0        conda-forge
pip                       23.3.1             py311haa95532_0     conda-forge
platformdirs               4.1.0              pyhd8ed1ab_0        conda-forge
pycodestyle                2.11.1             pyhd8ed1ab_0        conda-forge
pyflakes                   3.1.0              pyhd8ed1ab_0        conda-forge
python                     3.11.5             he1021f5_0          conda-forge
python_abi                 3.11               2_cp311             conda-forge
setuptools                 68.0.0             py311haa95532_0     conda-forge
sqlite                     3.41.2             h2bbff1b_0          conda-forge
tk                          8.6.12             h2bbff1b_0          conda-forge
tzdata                     2023c              h04d1e81_0          conda-forge
vc                          14.2               h21ff451_1          conda-forge
vs2015_runtime             14.27.29016        h5e58377_2          conda-forge
wheel                      0.41.2             py311haa95532_0     conda-forge
xz                          5.4.5              h8cc25b3_0          conda-forge
zlib                       1.2.13             h8cc25b3_0          conda-forge

(219-6) C:\Users\ACER\219-6>conda env export > environment.yml
```

Рисунок №2. Установка пакетов

Пример 1. Написать программу, которая позволяет перемещать все текстовые файлы в один каталог.

Программа:

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

import glob
import os
import shutil

if __name__ == "__main__":
    for file_name in glob.glob("*.txt"):
        new_path = os.path.join("archive", file_name)
        shutil.move(file_name, new_path)
```

Результат:

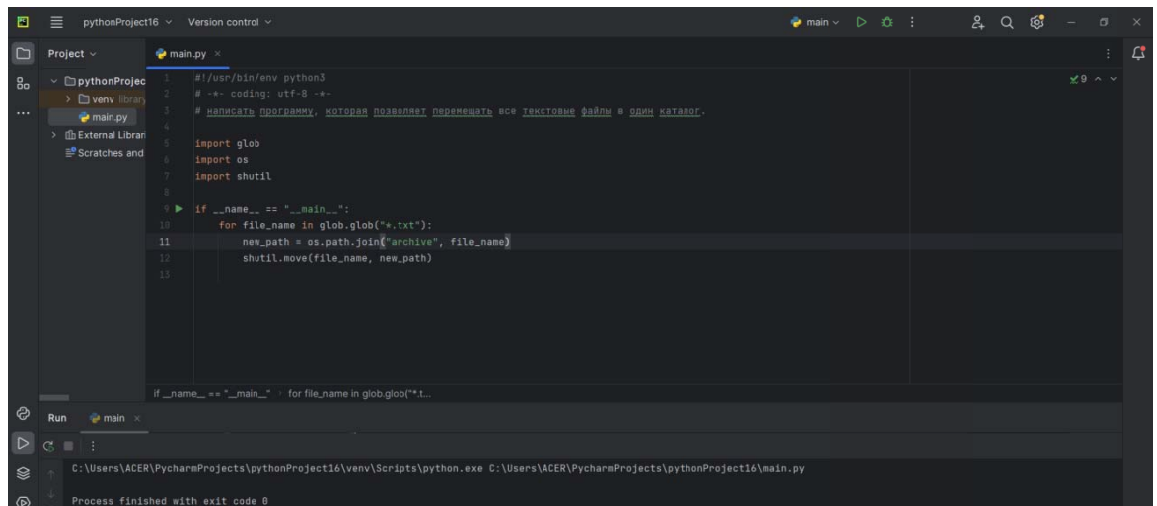


Рисунок №3. Результат работы программы примера 1

Пример 2. Создать уникальное имя файла

Программа:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#Создать уникальное имя файла

import pathlib
def unique_path(directory, name_pattern):
    counter = 0
    while True:
        counter += 1
        path = directory/name_pattern.format(counter)
        if not path.exists():
            return path
    path = unique_path(pathlib.Path.cwd(), 'test{:03d}.txt')
```

Результат:

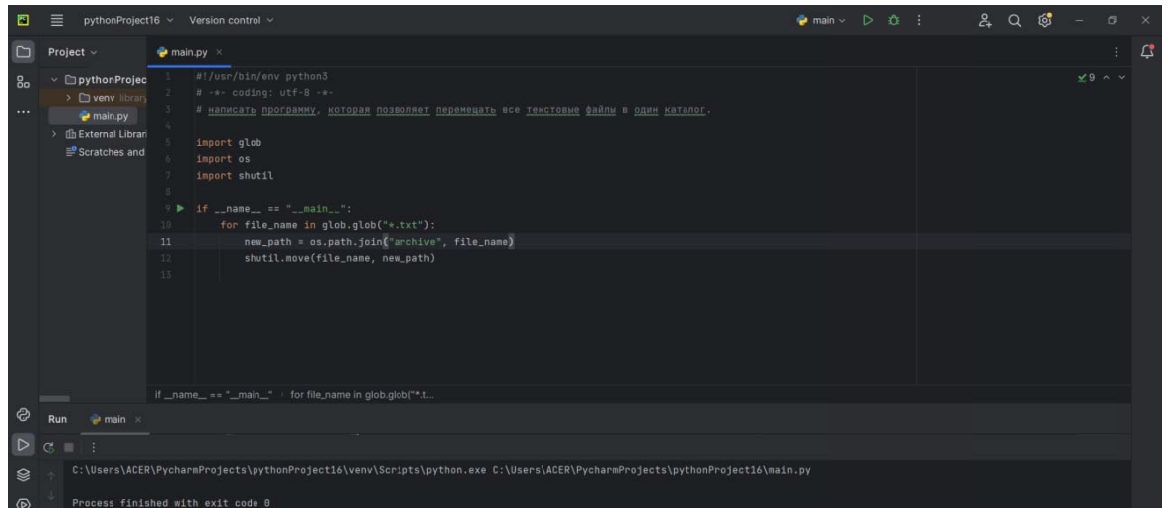


Рисунок №4. Результат программы примера №2

Пример 3. Создать уникальное имя файла

Программа:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#Создать уникальное имя файла
```

```
import pathlib
def tree(directory):
    print(f+ {directory})
    for path in sorted(directory.rglob('*')):
        depth = len(path.relative_to(directory).parts)
        spacer = ' ' * depth
        print(f{spacer}+ {path.name})
```

Результат:

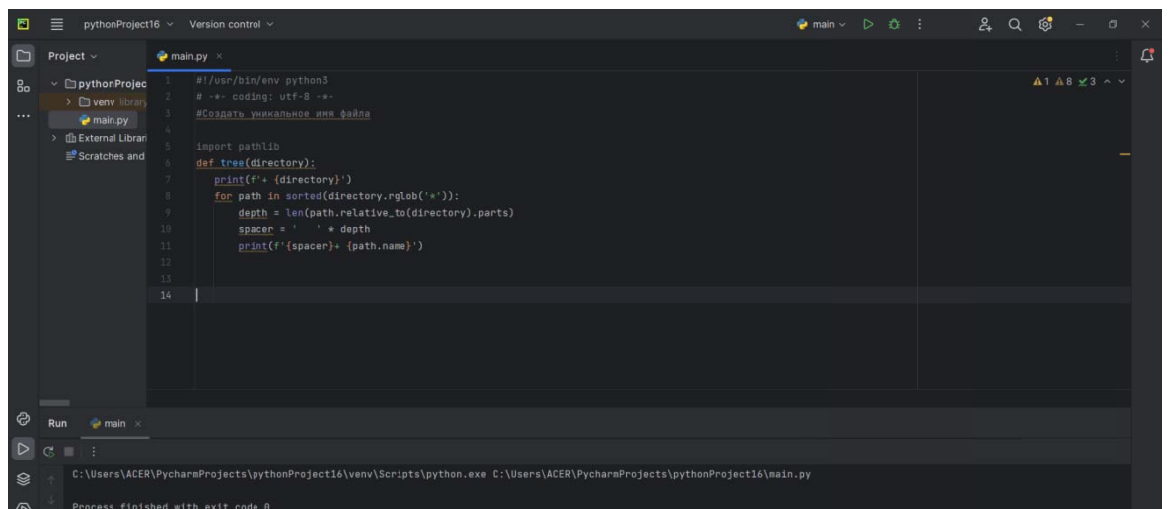


Рисунок №5. Результат программы №3

Выполним индивидуальное задание.

Условие: использовать словарь, содержащий следующие ключи: фамилия, имя; знак Зодиака; дата рождения (список из трёх чисел). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по датам рождения; вывод на экран информацию о людях, родившихся под знаком, название которого введено с клавиатуры; если таких нет, выдать на дисплей соответствующее сообщение. Оформив каждую команду в виде отдельной функции. Добавьте возможность хранения файла данных в домашнем каталоге пользователя. Для выполнения операций с файлами необходимо использовать модуль `pathlib`.

Программа:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#Использовать словарь, содержащий следующие ключи: фамилия, имя; знак Зодиака; дата
рождения (список из трёх чисел).
# Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в
список, состоящий из словарей заданной структуры;
# записи должны быть упорядочены по датам рождения; вывод на экран информацию о
людях, родившихся под знаком, название которого введено с клавиатуры;
# если таких нет, выдать на дисплей соответствующее сообщение. Оформив каждую
команду в виде отдельной функции.
# Добавьте возможность хранения файла данных в домашнем каталоге пользователя.
# Для выполнения операций с файлами необходимо использовать модуль pathlib .

import pathlib
import json
import os

data_file = pathlib.Path.home() / "data.json"

def read_data():
    if data_file.exists():
        with open(data_file, "r") as file:
            data = json.load(file)
        return data
    else:
        return []

def write_data(data):
    with open(data_file, "w") as file:
        json.dump(data, file)
```

```

def input_data():
    data = []
    while True:
        surname = input("Введите фамилию: ")
        name = input("Введите имя: ")
        zodiac = input("Введите знак Зодиака: ")
        birthday = input("Введите дату рождения (через пробел): ").split()
        if len(birthday) != 3:
            print("Неверный формат даты. Повторите ввод.")
            continue
        try:
            birthday = [int(x) for x in birthday]
        except ValueError:
            print("Неверный формат даты. Повторите ввод.")
            continue
        data.append({
            "фамилия": surname,
            "имя": name,
            "знак Зодиака": zodiac,
            "дата рождения": birthday
        })
        if input("Желаете добавить еще запись? (y/n): ") != 'y':
            break
    data.sort(key=lambda x: x["дата рождения"])
    return data

def find_people_by_zodiac(data, zodiac):
    people = []
    for person in data:
        if person["знак Зодиака"] == zodiac:
            people.append(person)
    return people

def print_people(people):
    if len(people) == 0:
        print("Нет людей с таким знаком Зодиака.")
    else:
        for person in people:
            print("Фамилия: {}".format(person["фамилия"]))
            print("Имя: {}".format(person["имя"]))
            print("Знак Зодиака: {}".format(person["знак Зодиака"]))
            print("Дата рождения: {}/{}/{}".format(person["дата рождения"][0], person["дата
рождения"][1],
                                                    person["дата рождения"][2]))
            print()

def main():
    filename = os.environ.get("DATA_FILE")
    if filename:
        with open(filename, "r") as file:
            data = eval(file.read())
    else:
        data = input_data()

```

```
zodiac = input("Введите знак Зодиака для поиска: ")
people = find_people_by_zodiac(data, zodiac)
print_people(people)
```

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

Результат:

```
# Записи должны быть упорядочены по датам рождения! Вывод на экран информации о людях, родившихся под знаком, название которого введено с клавиатуры.
# Если таких нет, выдать на дисплей соответствующее сообщение. Оформить каждую команду в виде отдельной функции.
# Добавить возможность хранения файла данных в домашнем каталоге пользователя.
# Для выполнения операций с файлами необходимо использовать модуль pathlib.

import pathlib
import json
import os

data_file = pathlib.Path.home() / "data.json"

def read_data():
    if data_file.exists():
        with open(data_file, "r") as file:
            data = json.load(file)
        return data
    else:
        return []

data = read_data()
read_data()
```

Run main x

Имя: Михаил
Знак Зодиака: Овен
Дата рождения: 21/3/2000

Process finished with exit code 0

Рисунок №6. Результат программы ИДЗ

Контрольные вопросы

1. Какие существовали средства для работы с файловой системой до Python 3.4?

До Python 3.4 для работы с файловой системой использовались модули `os.path` и `os.walk`.

2. Что регламентирует PEP 428?

PEP 428 регламентирует добавление модуля `pathlib`, который является высокоуровневым интерфейсом для работы с путями файловой системы.

3. Как осуществляется создание путей средствами модуля pathlib?

Создание путей средствами модуля pathlib осуществляется с помощью класса Path, который предоставляет методы для создания путей и манипулирования ими.

4. Как получить путь дочернего элемента файловой системы с помощью модуля pathlib?

Для получения пути дочернего элемента файловой системы нужно использовать метод parent, который возвращает родительский элемент пути.

5. Как получить путь к родительским элементам файловой системы с помощью модуля pathlib?

Получение пути к родительским элементам осуществляется с использованием метода parents, который возвращает список родительских элементов пути.

6. Как выполняются операции с файлами с помощью модуля pathlib?

Операции с файлами выполняются с использованием методов класса Path. Например, метод touch создает или изменяет время доступа и время модификации файла, метод rename переименовывает файл или каталог, метод copy.copy копирует файл или каталог.

7. Как можно выделить компоненты пути файловой системы с помощью модуля pathlib?

Компоненты пути файловой системы можно выделить с помощью методов split и join, которые разбивают путь на компоненты и объединяют компоненты в путь.

8. Как выполнить перемещение и удаление файлов с помощью модуля `pathlib`?

Перемещение и удаление файлов выполняется с использованием методов `rename` и `delete`.

9. Как выполнить подсчет файлов в файловой системе?

Подсчет файлов в файловой системе можно выполнить с использованием функции `path.count()`, которая возвращает количество файлов в указанном каталоге.

10. Как отобразить дерево каталогов файловой системы?

Отобразить дерево каталогов можно с использованием рекурсивного обхода каталога и вывода его содержимого.

11. Как создать уникальное имя файла?

Уникальное имя файла можно создать с использованием функций `uuid` или `generate_uuid`.

12. Каковы отличия в использовании модуля `pathlib` для различных операционных систем?

Отличия в использовании модуля `pathlib` для разных операционных систем могут быть связаны с различиями в синтаксисе путей и в особенностях работы с каталогами.