

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ

ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра информационных систем и технологий

Отчет по лабораторной работе №17.

Дисциплина: «Основы программной инженерии»

Выполнил:

Студент группы ПИЖ-б-о-22-1,

направление подготовки: 09.03.04

«Программная инженерия»

ФИО: Рядская Мария Александровна

Проверил:

Воронкин Р. А.

Ставрополь 2024

Тема: Лабораторная работа 2.15 Работа с файлами в языке Python

Цель работы: приобретение навыков по работе с текстовыми файлами

при написании программ с помощью языка программирования Python версии 3.x, изучение основных методов модуля os для работы с файловой системой, получение аргументов командной строки.


Ход работы.

1. Создание нового репозитория с лицензией MIT.

проекта в другом месте: [импортируйте репозитории](#).

Обязательные поля отмечены звездочкой (*).

Владелец *

 мрядская ▾

Название репозитория *

lab18

✓ lab18.

Отличные названия репозитория короткие и запоминающиеся. Нужно вдохновение? Как насчет [animated-doodle?](#)

Описание (необязательно)



Общедоступный

Этот репозиторий может видеть любой пользователь Интернета. Вы сами выбираете, кто может совершать коммиты.



Приватный

Вы сами выбираете, кто может просматривать этот репозиторий и фиксировать его в нем.

Инициализируйте этот репозиторий с помощью:



Добавьте файл README

Здесь вы можете написать подробное описание вашего проекта. [Узнайте больше о README](#).

Добавить файл .gitignore

.шаблон gitignore: Отсутствует ▾

Выберите файлы, которые не следует отслеживать, из списка шаблонов. [Узнайте больше об игнорировании файлов](#).

Выборите лицензию

2. Клонировала репозиторий на рабочий ПК.

```
C:\>cd git1

C:\git1>git clone https://github.com/mryadskaya/lab18.git
Cloning into 'lab18'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

C:\git1>
```

3. Дополнила файл .gitignore необходимыми инструкциями.

```
1      # Byte-compiled / optimized / DLL files
2      __pycache__/
3      *.py[cod]
4      *$py.class
5
6      # C extensions
7      *.so
8
9      # Distribution / packaging
10     .Python
11     build/
12     develop-eggs/
13     dist/
14     downloads/
15     eggs/
16     .eggs/
17     lib/
18     lib64/
19     parts/
20     sdist/
21     .....
```

Рисунок 1 – Файл .gitignore

```
C:\git1>cd lab18.1

C:\git1\lab18.1>git checkout -b develop
Switched to a new branch 'develop'
```

Рисунок 4 – организация ветки

4. Выполнила примеры лабораторной работы.

5. Задание 1. Написать программу, которая считывает текст из файла и выводит на экран сначала вопросительные, а затем восклицательные предложения.

Задание 2. В данном упражнении вы должны написать программу, которая анализирует исходный код функций в указанных файлах и выводит названия функций, не снабженных комментариями.

```
# Открываем файл для чтения
with open('text.txt', 'r') as file:
    text = file.read()

# Разделяем текст на предложения
sentences = text.split('.')

# Инициализируем списки для вопросительных и восклицательных предложений
question_sentences = []
exclamation_sentences = []

# Итерируем по всем предложениям и добавляем их в соответствующие списки
for sentence in sentences:
    if len(sentence) > 0:
        if sentence.strip()[-1] == '?':
            question_sentences.append(sentence.strip() + '.')
        elif sentence.strip()[-1] == '!':
            exclamation_sentences.append(sentence.strip() + '.')

# Выводим вопросительные предложения
print("Вопросительные предложения:")
for question in question_sentences:
    print(question)

# Выводим восклицательные предложения
print("\nВосклицательные предложения:")
for exclamation in exclamation_sentences:
    print(exclamation)
```

Рисунок 5 – код для выполнения задания 1

```
Вопросительные предложения:
What beautiful flowers?.
How can you be such an optimist?.
Look around! What do you see?.
Do you see this beauty! Amazing, isn't it?.

Восклицательные предложения:
What a wonderful morning!.
Why are you so happy!.
Thank you, my friend!.
```

Рисунок 6 – пример выполнения задания

```
import ast

def find_functions_without_comments(file_path):
    try:
        with open(file_path, 'r') as file:
            tree = ast.parse(file.read(), filename=file_path)
    except FileNotFoundError:
        print(f"Файл '{file_path}' не найден.")
        return

    for node in ast.walk(tree):
        if isinstance(node, ast.FunctionDef):
            if not ast.get_docstring(node):
                print(f"Функция '{node.name}' в файле '{file_path}' начиная с строки {node.lineno} не имеет комментариев.")

# Список файлов для анализа
files_to_analyze = ['example1.py', 'example2.py']

for file_path in files_to_analyze:
    find_functions_without_comments(file_path)
```

```
Функция 'multiply_numbers' в файле 'example1.py' начиная с строки 7 не имеет комментариев.
Функция 'greet' в файле 'example2.py' начиная с строки 1 не имеет комментариев.
Функция 'calculate_square' в файле 'example2.py' начиная с строки 4 не имеет комментариев.
Функция 'subtract_numbers' в файле 'example2.py' начиная с строки 7 не имеет комментариев.

Process finished with exit code 0
```

6. Самостоятельно подберите или придумайте задачу для работы с изученными функциями модуля `os`. Приведите решение этой задачи.

Этот код просит пользователя ввести названия городов через запятую. Затем создает для каждого города папку "Photos" с подпапками "Landmarks",

"Restaurants" и "Hotels". Каждая из этих подпапок содержит фотографии соответствующих мест или учреждений в городе.

```
import os

def create_city_folders(cities):
    for city in cities:
        city_folder = os.path.join("Photos", city)
        os.makedirs(city_folder, exist_ok=True)

        for subfolder in ["Landmarks", "Restaurants", "Hotels"]:
            subfolder_path = os.path.join(city_folder, subfolder)
            os.makedirs(subfolder_path, exist_ok=True)

if __name__ == "__main__":
    cities = input("Введите названия городов через запятую: ").split(",")
    create_city_folders([city.strip() for city in cities])
```

Рисунок 7 – код для выполнения программы

льный диск (C:) > git1 > lab11 > PyCharm > задание1 > Photos					Поиск
Имя	Дата изменения	Тип	Разм		
Москва	15.04.2024 17:24	Папка с файлами			
Ставрополь	15.04.2024 17:24	Папка с файлами			

Рисунок 8 – пример выполнения программы

7. Установила black, flake8, pre-commit. Использовал их во время коммита.

Контрольные вопросы:

1. Для открытия файла только для чтения в Python используется функция

open() с режимом "r".

2. Для открытия файла только для записи в Python используется функция

`open()` с режимом "w".

3. Для чтения данных из файла в Python используется метод `read()`

объекта файла.

4. Для записи данных в файл в Python используется метод `write()` объекта

файла.

5. Чтобы закрыть файл в Python и освободить ресурсы, используется

метод `close()` объекта файла.

6. Конструкция `with ... as` в Python предназначена для создания

контекстного менеджера, который автоматически управляет ресурсами.

Это

может быть полезно не только при работе с файлами, но и при работе с

сетевыми соединениями, базами данных и другими ресурсами, требующими

явного освобождения. Например, при работе с файлами она обеспечивает

автоматическое закрытие файла после выхода из контекста, даже если

произошло исключение.

7. Помимо рассмотренных методов чтения/записи данных из файла в

Python существуют также методы:

-`readline()`: для чтения одной строки из файла;

-`readlines()`: для чтения всех строк из файла в список;

-`writelines(lines)`: для записи списка строк в файл.

8. Помимо рассмотренных функций модуля `os` для работы с файловой системой, существуют также функции:

-`os.rename(src, dst)`: для переименования файла или директории;

-`os.remove(path)`: для удаления файла;

-`os.mkdir(path)`: для создания директории;

-`os.rmdir(path)`: для удаления директории.