**Хамикоева Ольга, ПИ19-3, Файловый менеджер**

Я постаралась максимально часто оставлять комментарии по ходу программы, поэтому каждая функция подписана. Программа разделена на 3 файла (модуля):

* **main.py (здесь запускается программа, создаются экземпляр обработчика файлов, а также завершается программа) :**

import os

import shutil

import pathlib

from typing import Dict

import pathStorage

import fileProcessing

class Colors:

"""Хранение цветов"""

HEADER = "\033[95m"

OKBLUE = "\033[94m"

OKCYAN = "\033[96m"

OKGREEN = "\033[92m"

WARNING = "\033[93m"

FAIL = "\033[91m"

ENDC = "\033[0m"

BOLD = "\033[1m"

UNDERLINE = "\033[4m"

print(os.sep)

def main():

file\_processing = fileProcessing.FileProcessing() # обработчик файлов

while True:

command = input("\nВведите команду -> ").split(" ")

if command[0] == "exit": #чтоб мы могли красиииивооо уйти из программы (отсылка к песне Меладзе)

break

# Проверяем, существует ли такая команда

result = file\_processing.router(command[0])

# Если ДА

if result:

try:

result(\*command[1:])

except TypeError:

print(f"Команда {command[0]} была вызвана с некорректными аргументами")

else: #иначе

commands\_str = "\n".join(

[

f"{Colors.OKGREEN}{key}{Colors.ENDC} - {value}"

for (key, value) in fileProcessing.FileProcessing.get\_commands().items()

]

)

print(f"Команда {command[0]} не найдена! Список команд:\n{commands\_str}")

print("До новых встреч!")

try:

if \_\_name\_\_ == "\_\_main\_\_":

main()

except:

print("Что-то пошло не так... :(" )

* **pathStorage.py (в этом модуле ведется работа с путями, обрабатывается и хранится информация о путях) :**

import os

import shutil

import pathlib

from typing import Dict

class Colors:

"""Хранение цветов"""

HEADER = "\033[95m"

OKBLUE = "\033[94m"

OKCYAN = "\033[96m"

OKGREEN = "\033[92m"

WARNING = "\033[93m"

FAIL = "\033[91m"

ENDC = "\033[0m"

BOLD = "\033[1m"

UNDERLINE = "\033[4m"

class PathStorage:

"""Здесь ведется работа с путями (хранится информация)"""

def \_\_init\_\_(self) -> None:

self.\_\_storage = ["storage"]

def add\_path(self, path: str) -> None:

"""Добавляет файл в иерархию каталогов"""

# Если хотим выйти на уровень выше

if ".." in path and len(self.\_\_storage) != 1:

self.\_\_storage.pop(-1)

# Если хотим выйти за пределы Вселенной

elif ".." in path:

print("Вы хотите выйти за пределы этой Вселенной!")

# Значит хотим перейти на уровень ниже

else:

self.\_\_storage.append(path)

def file2path(self, file\_name: str) -> str:

"""Возвращает указанный файл в текущей иерархии каталогов"""

locale\_storage = self.\_\_storage.copy()

locale\_storage.append(file\_name)

abs\_path = pathlib.Path(\_\_file\_\_).parent.absolute()

return str(abs\_path) + "/" + "/".join(locale\_storage)

@property

def path(self):

"""Возвращает текущую иерархию каталогов"""

abs\_path = pathlib.Path(\_\_file\_\_).parent.absolute()

return str(abs\_path) + "/" + "/".join(self.\_\_storage)

@property

def upper\_path(self):

"""Возвращает директорию выше текущей"""

abs\_path = pathlib.Path(\_\_file\_\_).parent.absolute()

print(self.\_\_storage[1:])

return str(abs\_path) + "/" + "/".join(self.\_\_storage[:1])

* **fileProcessing.py (в данном модуле как раз и происходит работа с файлами, директориями, описаны команды, присущие файловому менеджеру) :**

import os

import shutil

import pathlib

from typing import Dict

import pathStorage

class Colors:

"""Хранение цветов"""

HEADER = "\033[95m"

OKBLUE = "\033[94m"

OKCYAN = "\033[96m"

OKGREEN = "\033[92m"

WARNING = "\033[93m"

FAIL = "\033[91m"

ENDC = "\033[0m"

BOLD = "\033[1m"

UNDERLINE = "\033[4m"

class FileProcessing:

"""Тут происходит работа, связанная с файлами"""

@staticmethod

def get\_commands() -> Dict[str, str]:

"""Выводится список возможных команд"""

commands\_dict = {

"travel": "Перемещение между папками",

"content": "Вывод содержимого текущей папки на экран",

"mkfold": "Создание папки",

"rmfold": "Удаление папки",

"create": "Создание файла",

"new\_name": "Переименование файла/папки",

"reading": "Чтение файла",

"destroy": "Удаление файла",

"replicate": "Копирование файла/папки",

"moving": "Перемещение файла/папки",

"compose": "Запись в файл",

}

return commands\_dict

def \_\_init\_\_(self) -> None:

self.storage = pathStorage.PathStorage()

def mkfold(self, filename: str):

"""Создание папки (с указанием имени)"""

current\_path = self.storage.file2path(filename)

try:

os.mkdir(current\_path)

except FileNotFoundError:

os.makedirs(current\_path)

except FileExistsError:

print(f"Директория {filename} уже существует")

def rmfold(self, filename: str):

"""Удаление папки по имени"""

current\_path = self.storage.file2path(filename)

try:

os.rmdir(current\_path)

except OSError:

try:

shutil.rmtree(current\_path, ignore\_errors=False, onerror=None)

except FileNotFoundError:

print(f"Директория {filename} не найдена")

except NotADirectoryError:

print(f"Файл {filename} не является директорией")

except FileNotFoundError:

print(f"Директория {filename} не найдена")

except NotADirectoryError:

print(f"Файл {filename} не является директорией")

def travel(self, filename: str):

"""

Перемещение между папками может осуществляться на уровень выше, заходя по имени папки

и в пределах текущей директории

"""

self.storage.add\_path(filename)

current\_path = self.storage.path

try:

os.chdir(current\_path)

except FileNotFoundError:

self.storage.add\_path("../")

print(f"Директория {filename} не найдена")

except NotADirectoryError:

self.storage.add\_path("../")

print(f"Файл {filename} не является директорией")

def content(self):

"""

Показать содержимое папки

"""

current\_path = self.storage.path

filelist = os.listdir(current\_path)

for i in range(len(filelist)):

if os.path.isdir(self.storage.file2path(filelist[i])):

filelist[i] = f"[dir] {Colors.OKCYAN}{filelist[i]}{Colors.ENDC}"

elif os.path.isfile(self.storage.file2path(filelist[i])):

filelist[i] = f"[file] {Colors.WARNING}{filelist[i]}{Colors.ENDC}"

r = "\n".join(filelist)

print(f"Содержимое {current\_path}:\n{r}")

def touch(self, filename: str):

"""Создание новых файлов, в качестве параметра - имя файла"""

current\_path = self.storage.file2path(filename)

try:

open(current\_path, "a").close()

except IsADirectoryError:

print(f"Директория с именем {filename} уже существует. Придумайте другое название")

def cat(self, filename: str) -> str:

"""Чтение файла"""

current\_path = self.storage.file2path(filename)

try:

with open(current\_path, "r") as file:

print(file.read())

except FileNotFoundError:

print(f"Файл {filename} не найден")

except IsADirectoryError:

print(f"Файл {filename} является директорией")

def new\_name(self, filename\_old: str, filename\_new: str):

"""Переименование файла"""

path\_old = self.storage.file2path(filename\_old)

path\_new = self.storage.file2path(filename\_new)

# На случай, если файл с именем уже существует

try:

if not os.path.isfile(path\_new):

os.rename(path\_old, path\_new)

else:

raise IsADirectoryError

except FileNotFoundError:

print(f"Указанного файла {filename\_old} не существует")

except IsADirectoryError:

print(f"Файл с названием {filename\_new} уже существует")

def rm(self, filename: str):

"""Удаление, в качестве параметра - имя файла"""

path = self.storage.file2path(filename)

if os.path.isfile(path):

os.remove(path)

else:

print(f"Файла {filename} не существует")

def cp(self, filename: str, path: str):

"""Копирование файла из одной папки в другую"""

path\_old = self.storage.file2path(filename)

# Копирование на уровень выше

if ".." in path:

path\_new = self.storage.upper\_path + "/" + filename

else:

# Проверяем на то, что это за тип файла

buff = self.storage.file2path(path)

# Если конечный путь папка - закидываем туда файл

if os.path.isdir(buff):

path\_new = self.storage.file2path(path + "/" + filename)

else:

# Значит это копирование на одном уровне

path\_new = self.storage.file2path(path)

try:

shutil.copyfile(path\_old, path\_new)

# Если это директория - копируем директорию

except IsADirectoryError:

shutil.copytree(path\_old, path\_new)

except FileNotFoundError:

print(f"Файл {filename} не найден")

def mv(self, filename: str, path: str):

"""Перемещение файла"""

path\_old = self.storage.file2path(filename)

if ".." in path:

path\_new = self.storage.upper\_path + "/" + filename

else:

# Проверяем на то, что это за тип файла

buff = self.storage.file2path(path)

# Если директория - закидываем туда файл

if os.path.isdir(buff):

path\_new = self.storage.file2path(path + "/" + filename)

else:

# Значит это перемещение на одном уровне

path\_new = self.storage.file2path(path)

try:

shutil.move(path\_old, path\_new)

except FileNotFoundError:

print(f"Файл {filename} не найден")

def write(self, filename: str, \*data: str):

"""Запись в файл"""

text = " ".join(data)

path = self.storage.file2path(filename)

try:

with open(path, "a") as file:

file.write(text)

except IsADirectoryError:

print(f"Указанный файл {filename} - директория")

def router(self, command: str):

"""Параллель между командами и методами FileProcessing"""

commands = [

self.travel,

self.content,

self.mkfold,

self.rmfold,

self.touch,

self.new\_name,

self.cat,

self.rm,

self.cp,

self.mv,

self.write,

]

item\_dict = dict(zip(FileProcessing.get\_commands().keys(), commands))

return item\_dict.get(command, None)

**Работа с программой:**

Введите команду -> ыва

Команда ыва не найдена! Список команд:

travel - Перемещение между папками

content - Вывод содержимого текущей папки на экран

mkfold - Создание папки

rmfold - Удаление папки

create - Создание файла

new\_name - Переименование файла/папки

reading - Чтение файла

destroy - Удаление файла

replicate - Копирование файла/папки

moving - Перемещение файла/папки

compose - Запись в файл

Введите команду -> content

Содержимое D:\Рабочий стол 2\Учеба 2 курс\_2 семестр\Практикум по программирвоанию\File Manager/storage:

[file] file1.txt

[file] test2

[dir] test3

Введите команду -> travel test1

Директории test1 не существует

Введите команду -> travel test2

Файл test2 не является директорией

Введите команду -> travel test3

Введите команду -> content

Содержимое D:\Рабочий стол 2\Учеба 2 курс\_2 семестр\Практикум по программирвоанию\File Manager/storage/test3:

[dir] test2

Введите команду -> create fileeee.txt

Введите команду -> compose fileeee.txt "hello-hello-hello"

Введите команду -> reading fileeee.txt

"hello-hello-hello"

Введите команду -> new\_name fileeee.txt filee.txt

Введите команду -> content

Содержимое D:\Рабочий стол 2\Учеба 2 курс\_2 семестр\Практикум по программирвоанию\File Manager/storage/test3:

[file] filee.txt

[dir] test2

Введите команду -> destroy filee.txt

Введите команду -> content

Содержимое D:\Рабочий стол 2\Учеба 2 курс\_2 семестр\Практикум по программирвоанию\File Manager/storage/test3:

[dir] test2

Введите команду -> cd ..

Команда cd не найдена! Список команд:

travel - Перемещение между папками

content - Вывод содержимого текущей папки на экран

mkfold - Создание папки

rmfold - Удаление папки

create - Создание файла

new\_name - Переименование файла/папки

reading - Чтение файла

destroy - Удаление файла

replicate - Копирование файла/папки

moving - Перемещение файла/папки

compose - Запись в файл

Введите команду -> cd ../

Команда cd не найдена! Список команд:

travel - Перемещение между папками

content - Вывод содержимого текущей папки на экран

mkfold - Создание папки

rmfold - Удаление папки

create - Создание файла

new\_name - Переименование файла/папки

reading - Чтение файла

destroy - Удаление файла

replicate - Копирование файла/папки

moving - Перемещение файла/папки

compose - Запись в файл

Введите команду -> travel ../

Введите команду -> content

Содержимое D:\Рабочий стол 2\Учеба 2 курс\_2 семестр\Практикум по программирвоанию\File Manager/storage:

[file] file1.txt

[file] test2

[dir] test3

Введите команду -> moving file1.txt test3

Введите команду -> content

Содержимое D:\Рабочий стол 2\Учеба 2 курс\_2 семестр\Практикум по программирвоанию\File Manager/storage:

[file] test2

[dir] test3

Введите команду -> travel test3

Введите команду -> content

Содержимое D:\Рабочий стол 2\Учеба 2 курс\_2 семестр\Практикум по программирвоанию\File Manager/storage/test3:

[file] file1.txt

[dir] test2

Введите команду -> travel ..

Вы хотите выйти за пределы этой Вселенной!

Введите команду -> content

Содержимое D:\Рабочий стол 2\Учеба 2 курс\_2 семестр\Практикум по программирвоанию\File Manager/storage:

[file] test2

[dir] test3

Введите команду -> exit

До новых встреч!