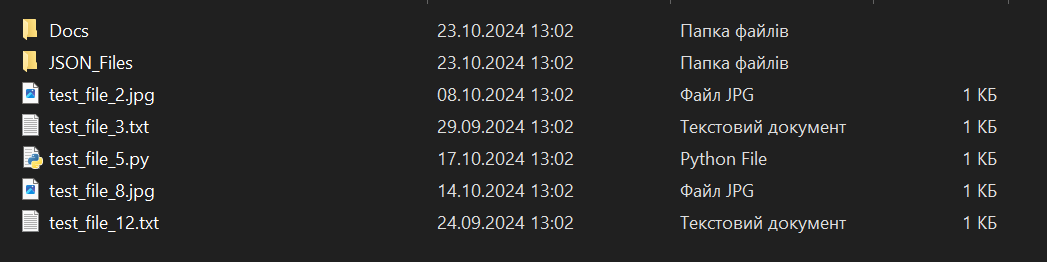
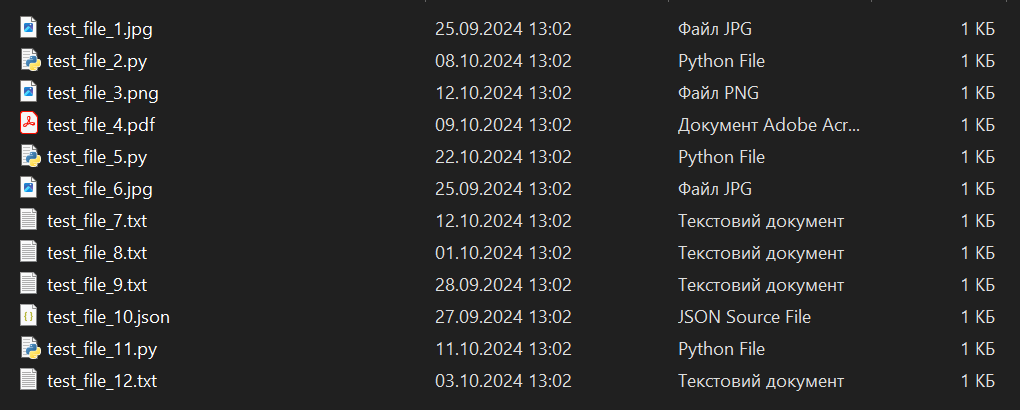
Завдання 1

Написав код для генерації файлів у двох папках:

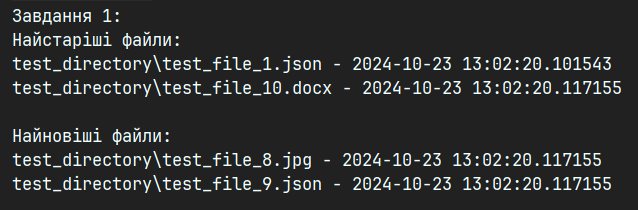
Папка test\_directory:



Папка test\_directory\_2:

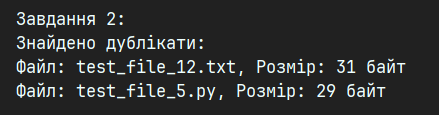


Запустив виконання коду для першого завдання та отримав такий результат:



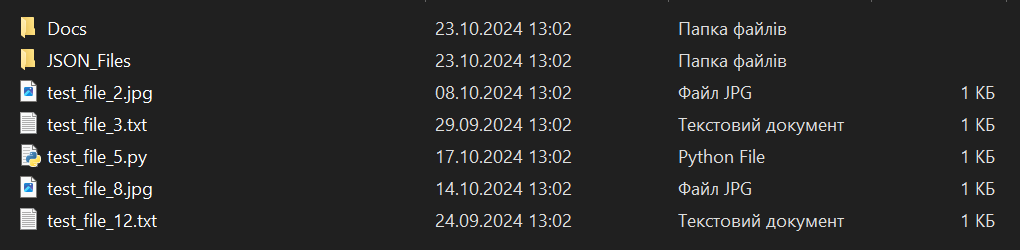
Завдання 2

Знайдені дублікати в папках test\_directory та test\_directory\_2:

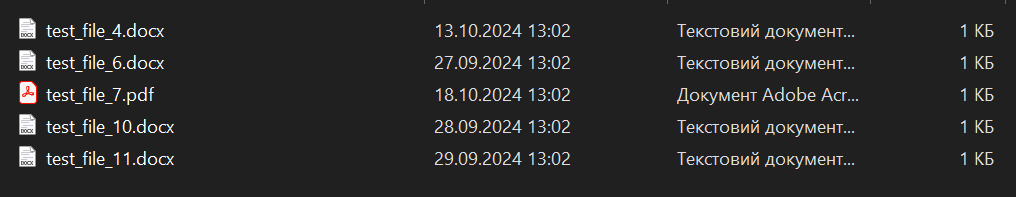


Завдання 3

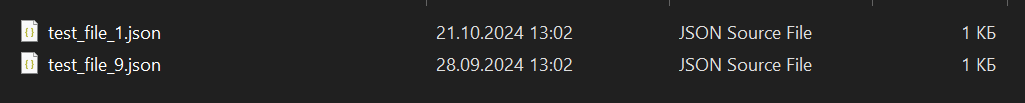
Маю папку test\_directory, в які є інші папки:



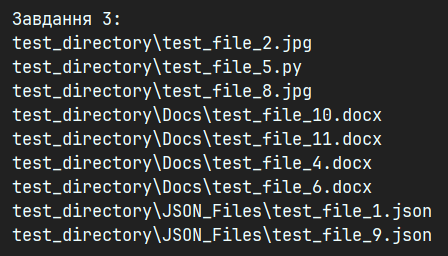
Папка Docs:



Папка JSON\_Files:

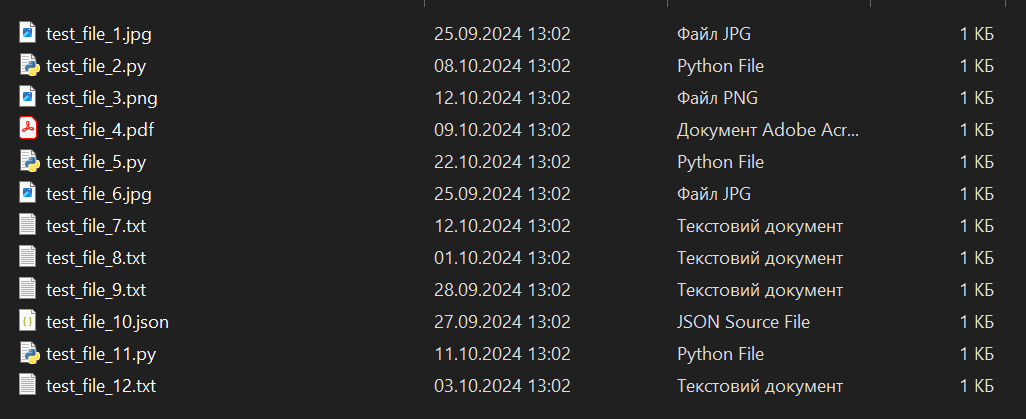


Функція третього завдання проходиться по всіх вкладених папках та виводить файли з вказаними розширеннями:

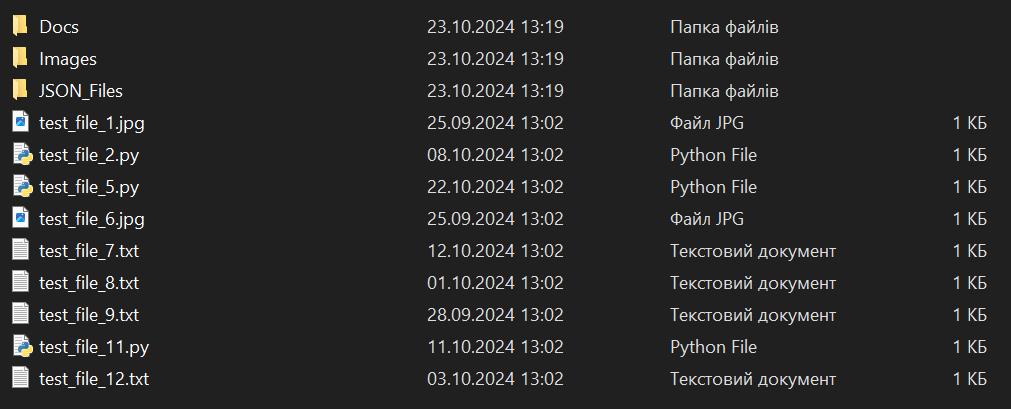


Завдання 4

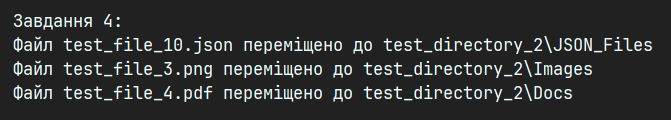
Папка test\_directory\_2 до виконання функції четвертого завдання:



Папка test\_directory\_2 після виконання функції четвертого завдання:



Вивід функції четвертого завдання:



**Код програми**



| *import* os  *import* random  *import* shutil  *from* datetime *import* datetime, timedelta  *def* generate\_test\_files(directory):  *"""*  *Функція для генерації файлів різного типу*  ***:param*** *directory:*  ***:return****:*  *"""*  os.makedirs(directory, exist\_ok=*True*)  extensions = ['txt', 'py', 'jpg', 'json', 'pdf', 'docx', 'png']  *for* i *in* range(1, 13):  ext = random.choice(extensions)  filepath = os.path.join(directory, f"test\_file\_{i}.{ext}")  *with* open(filepath, 'w') *as* f:  f.write(f"Test file {i} with extension {ext}")  mod\_time = datetime.now() - timedelta(days=random.randint(1, 30))  os.utime(filepath, (mod\_time.timestamp(), mod\_time.timestamp()))  *def* find\_oldest\_newest\_files(directory):  *"""*  *Завдання 1: Знаходимо найстаріші та найновіші файли*  ***:param*** *directory:*  ***:return****:*  *"""*  files = [os.path.join(directory, f) *for* f *in* os.listdir(directory) *if* os.path.isfile(os.path.join(directory, f))]  files.sort(key=*lambda* x: os.path.getctime(x))  print("Найстаріші файли:")  *for* file *in* files[:2]:  print(f"{file} - {datetime.fromtimestamp(os.path.getctime(file))}")  print("\nНайновіші файли:")  *for* file *in* files[-2:]:  print(f"{file} - {datetime.fromtimestamp(os.path.getctime(file))}")  *def* find\_duplicates(dir1, dir2):  *"""*  *Завдання 2: Пошук дубльованих файлів у двох папках*  ***:param*** *dir1:*  ***:param*** *dir2:*  ***:return****:*  *"""*  files1 = {f: os.path.getsize(os.path.join(dir1, f)) *for* f *in* os.listdir(dir1) *if*  os.path.isfile(os.path.join(dir1, f))}  files2 = {f: os.path.getsize(os.path.join(dir2, f)) *for* f *in* os.listdir(dir2) *if*  os.path.isfile(os.path.join(dir2, f))}  duplicates = [(f, size) *for* f, size *in* files1.items() *if* f *in* files2 *and* files2[f] == size]  print("Знайдено дублікати:")  *for* f, size *in* duplicates:  print(f"Файл: {f}, Розмір: {size} байт")  *def* filter\_files\_by\_extension(directory, extensions):  *"""*  *Завдання 3: Перевірка типу файлів і виведення загального списку файлів*  ***:param*** *directory:*  ***:param*** *file\_types:*  ***:return****:*  *"""*  *for* root, \_, files *in* os.walk(directory):  *for* file *in* files:  *if* file.split('.')[-1] *in* extensions:  print(os.path.join(root, file))  *def* move\_files\_by\_type(directory):  *"""*  *Завдання 4: Переміщення файлів у окремі папки за типами*  ***:param*** *directory:*  ***:return****:*  *"""*  folder\_mapping = {  'json': 'JSON\_Files',  'png': 'Images',  'pdf': 'Docs',  'docx': 'Docs'  }  *for* file *in* os.listdir(directory):  file\_path = os.path.join(directory, file)  *if* os.path.isfile(file\_path):  ext = file.split('.')[-1]  *if* ext *in* folder\_mapping:  target\_dir = os.path.join(directory, folder\_mapping[ext])  os.makedirs(target\_dir, exist\_ok=*True*)  shutil.move(file\_path, os.path.join(target\_dir, file))  print(f"Файл {file} переміщено до {target\_dir}")  *if* \_\_name\_\_ == '\_\_main\_\_':  *# Генерація файлів та запуск завдань*  base\_dir = "test\_directory"  second\_dir = "test\_directory\_2"  *# Генерація файлів у двох папках*  generate\_test\_files(base\_dir)  generate\_test\_files(second\_dir)  print("Завдання 1:")  find\_oldest\_newest\_files(base\_dir)  print("\nЗавдання 2:")  find\_duplicates(base\_dir, second\_dir)  print("\nЗавдання 3:")  filter\_files\_by\_extension(base\_dir, ['docx', 'py', 'jpg', 'json',])  print("\nЗавдання 4:")  move\_files\_by\_type(base\_dir) |
| --- |