**Лабораторна робота** **№ 3**

## Створення утиліти «DiskInfo»

Мета: використовуючи виклики системних функцій, отримати інформацію про дискову підсистему

**У коді будемо використовувати англійську мову, бо в мене проблеми з реєстрами у консолі і кирилицю воно показує неправильно**

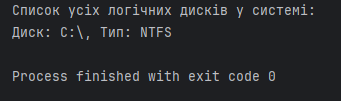
1. Список усіх логічних дисків в системі.

import psutil  
  
def main():  
 partitions = psutil.disk\_partitions()  
 logical\_drives = [partition.device for partition in partitions if partition.fstype != '']  
  
 print("Список усіх логічних дисків у системі:")  
 for drive in logical\_drives:  
 print(drive)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()



1. Отримати тип кожного диску присутнього в системі, та дати пояснення для кожного типу диску.

import psutil  
  
def main():  
 partitions = psutil.disk\_partitions()  
 logical\_drives = [(partition.device, partition.fstype) for partition in partitions if partition.fstype != '']  
  
 print("Список усіх логічних дисків у системі:")  
 for drive, fstype in logical\_drives:  
 print(f"Диск: {drive}, Тип: {fstype}")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()



1. Отримати інформацію про диски в системі та про файлові системи Які Використовують на них.

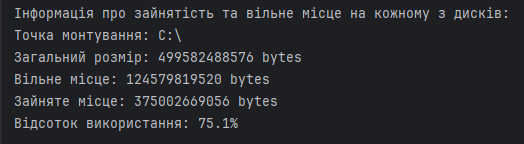
Точно не знаю, що треба було дізнатися, тому просто вивів таку інформацію про диски в системі

import psutil  
  
  
def main():  
 partitions = psutil.disk\_partitions()  
  
 print("Інформація про диски та використовані на них файлові системи:")  
 for partition in partitions:  
 drive = partition.device  
 mountpoint = partition.mountpoint  
 fstype = partition.fstype  
  
 print(f"Диск: {drive}, Точка монтування: {mountpoint}, Файлова система: {fstype}")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()



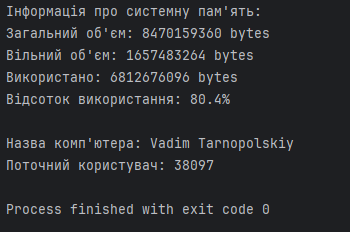
1. Отримати інформацію про зайнятості та вільне місце на кожному з дисків.

import psutil  
  
  
def main():  
 partitions = psutil.disk\_partitions()  
  
 print("Інформація про зайнятість та вільне місце на кожному з дисків:")  
 for partition in partitions:  
 mountpoint = partition.mountpoint  
 usage = psutil.disk\_usage(mountpoint)  
  
 print(f"Точка монтування: {mountpoint}")  
 print(f"Загальний розмір: {usage.total} bytes")  
 print(f"Вільне місце: {usage.free} bytes")  
 print(f"Зайняте місце: {usage.used} bytes")  
 print(f"Відсоток використання: {usage.percent}%\n")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()



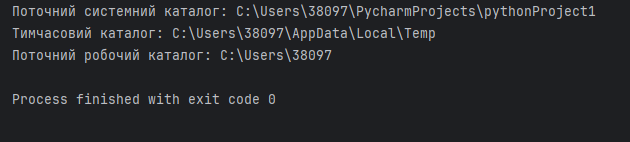
1. Отримати інформацію про системну пам'ять.
2. Отримати інформацію про Назву комп'ютера
3. Отримати Назву поточного користувача

import psutil  
  
def main():  
 # Інформація про системну пам'ять  
 memory = psutil.virtual\_memory()  
 print("Інформація про системну пам'ять:")  
 print(f"Загальний об'єм: {memory.total} bytes")  
 print(f"Вільний об'єм: {memory.available} bytes")  
 print(f"Використано: {memory.used} bytes")  
 print(f"Відсоток використання: {memory.percent}%\n")  
  
 # Інформація про назву комп'ютера  
 hostname = psutil.users()[0].host  
 print("Назва комп'ютера:", hostname)  
  
 # Інформація про поточного користувача  
 current\_user = psutil.users()[0].name  
 print("Поточний користувач:", current\_user)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()



1. Отримати інформацію про поточний системний каталог, Тимчасовий каталог, поточний робочий каталог.

import os  
  
def main():  
 # Поточний системний каталог  
 current\_directory = os.getcwd()  
 print("Поточний системний каталог:", current\_directory)  
  
 # Тимчасовий каталог  
 temp\_directory = os.environ.get('TEMP')  
 print("Тимчасовий каталог:", temp\_directory)  
  
 # Поточний робочий каталог  
 working\_directory = os.path.expanduser('~')  
 print("Поточний робочий каталог:", working\_directory)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()



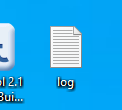
1. Для обраних каталогу на диску, Включити спостереження за змінами, продемонструвати відслідковування більше однієї зміни. Зміни записувати в лог файл.

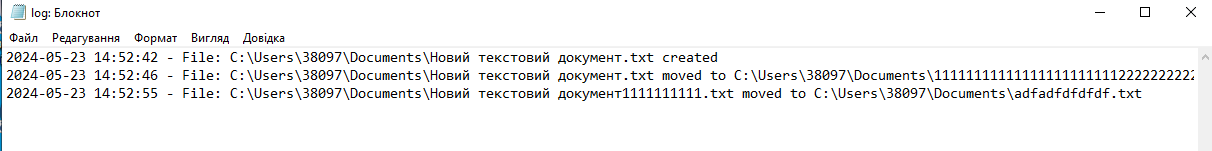
import os  
import time  
  
from watchdog.observers import Observer  
from watchdog.events import FileSystemEventHandler  
  
  
class MyHandler(FileSystemEventHandler):  
 def \_\_init\_\_(self, log\_file):  
 self.log\_file = log\_file  
  
 def on\_created(self, event):  
 self.log(f"File: {event.src\_path} created")  
  
 def on\_deleted(self, event):  
 self.log(f"File: {event.src\_path} deleted")  
  
 def on\_modified(self, event):  
 self.log(f"File: {event.src\_path} modified")  
  
 def on\_moved(self, event):  
 self.log(f"File: {event.src\_path} moved to {event.dest\_path}")  
  
 def log(self, message):  
 with open(self.log\_file, "a") as f:  
 f.write(f"{time.strftime('%Y-%m-%d %H:%M:%S')} - {message}\n")  
  
  
def main():  
 directory\_path = r"C:\Users\38097\Documents"  
 log\_file\_path = r"C:\Users\38097\Desktop\log.txt"  
  
 # Створення каталогу для логу, якщо він не існує  
 if not os.path.exists(directory\_path):  
 os.makedirs(directory\_path)  
  
 event\_handler = MyHandler(log\_file\_path)  
 observer = Observer()  
 observer.schedule(event\_handler, directory\_path, recursive=True)  
 observer.start()  
  
 print(f"Monitoring directory: {directory\_path}")  
 print("Press 'q' to quit.")  
  
 try:  
 while True:  
 time.sleep(1)  
 except KeyboardInterrupt:  
 observer.stop()  
 observer.join()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Тепер при створенні файлу чи редагуванні зміни записуються в лог файл та виводяться в консоль



Log.file:





***Висновок:*** Під час виконання лабораторної роботи, використовуючи виклики системних функцій, отримати інформацію про дискову підсистему