

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()
```

```
Список усіх логічних дисків у системі:
C:\
```

2. 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()
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.123.15.000 – Лр.3						
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи №3			Лім.	Арк.	Аркуші	
Розроб.		Тарнопольський									
Перевір.		Власенко О.В								1	9
Реценз.								ФІКТ, гр. КІ-21-1			
Н. Контр.											
Зав.каф.		Єфіменко А.А.									

```
Список усіх логічних дисків у системі:  
Диск: C:\, Тип: NTFS  
  
Process finished with exit code 0
```

3. 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()
```

```
Інформація про диски та використовані на них файлові системи:  
Диск: C:\, Точка монтування: C:\, Файлова система: NTFS
```

4. 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")
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.123.15.000 – Лр.3	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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()

```

```

Інформація про зайнятість та вільне місце на кожному з дисків:
Точка монтування: C:\
Загальний розмір: 499582488576 bytes
Вільне місце: 124579819520 bytes
Зайняте місце: 375002669056 bytes
Відсоток використання: 75.1%

```

5. Отримати інформацію про системну пам'ять.
6. Отримати інформацію про Назву комп'ютера
7. 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()

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.123.15.000 – Лр.3	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Інформація про системну пам'ять:
Загальний об'єм: 8470159360 bytes
Вільний об'єм: 1657483264 bytes
Використано: 6812676096 bytes
Відсоток використання: 80.4%

Назва комп'ютера: Vadim Tarnopolskiy
Поточний користувач: 38097

Process finished with exit code 0

```

8. 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()

```

```

Поточний системний каталог: C:\Users\38097\PycharmProjects\pythonProject1
Тимчасовий каталог: C:\Users\38097\AppData\Local\Temp
Поточний робочий каталог: C:\Users\38097

```

```

Process finished with exit code 0

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.123.15.000 – Лр.3	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

9. 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()
```

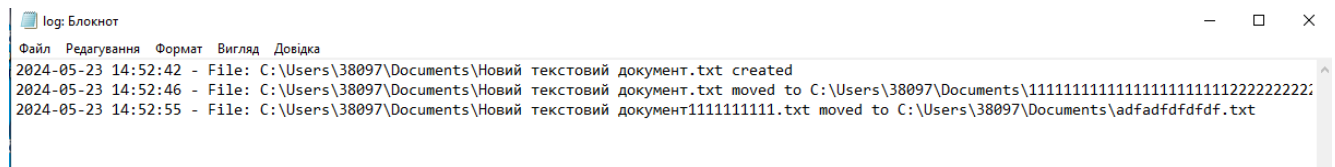
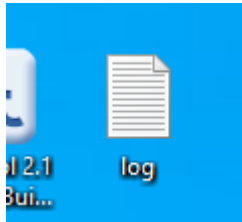
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.123.15.000 – Лр.3	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

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

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

```
Monitoring directory: C:\Users\38097\Documents
```

Log.file:



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

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.123.15.000 – Лр.3	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		