import os

import datetime

# Hardcoded folder location

folder\_path = "/path/to/your/folder"

# Function to get general information about a file

def get\_file\_info(file\_path):

    file\_name = os.path.basename(file\_path)

    file\_extension = os.path.splitext(file\_name)[1]

    created\_time = datetime.datetime.fromtimestamp(os.path.getctime(file\_path)).strftime('%Y-%m-%d %H:%M:%S')

    updated\_time = datetime.datetime.fromtimestamp(os.path.getmtime(file\_path)).strftime('%Y-%m-%d %H:%M:%S')

    return file\_name, file\_extension, created\_time, updated\_time

# Function to get information about image files

def get\_image\_info(file\_path):

    import PIL.Image

    with PIL.Image.open(file\_path) as img:

        width, height = img.size

    return f"{width}x{height}"

# Function to get information about text files

def get\_text\_info(file\_path):

    with open(file\_path, 'r') as file:

        lines = file.readlines()

        line\_count = len(lines)

        word\_count = sum(len(line.split()) for line in lines)

        char\_count = sum(len(line) for line in lines)

    return line\_count, word\_count, char\_count

# Function to get information about program files

def get\_program\_info(file\_path):

    with open(file\_path, 'r') as file:

        lines = file.readlines()

        line\_count = len(lines)

        class\_count = sum(1 for line in lines if line.strip().startswith("class "))

        method\_count = sum(1 for line in lines if line.strip().startswith("def "))

    return line\_count, class\_count, method\_count

# Main function

def main():

    while True:

        action = input("Enter action (commit/info <filename>/status): ").split()

        if action[0] == "commit":

            snapshot\_time = datetime.datetime.now()

            print(f"Snapshot time updated to: {snapshot\_time}")

        elif action[0] == "info":

            if len(action) != 2:

                print("Invalid command. Please provide filename.")

                continue

            file\_name = action[1]

            file\_path = os.path.join(folder\_path, file\_name)

            if not os.path.exists(file\_path):

                print("File not found.")

                continue

            file\_info = get\_file\_info(file\_path)

            file\_extension = file\_info[1]

            if file\_extension in ['.png', '.jpg']:

                print(f"Image Size: {get\_image\_info(file\_path)}")

            elif file\_extension == '.txt':

                text\_info = get\_text\_info(file\_path)

                print(f"Line count: {text\_info[0]}")

                print(f"Word count: {text\_info[1]}")

                print(f"Character count: {text\_info[2]}")

            elif file\_extension in ['.py', '.java']:

                program\_info = get\_program\_info(file\_path)

                print(f"Line count: {program\_info[0]}")

                print(f"Class count: {program\_info[1]}")

                print(f"Method count: {program\_info[2]}")

            else:

                print("Unknown file type.")

        elif action[0] == "status":

            snapshot\_time = datetime.datetime.now()

            print(f"Snapshot time: {snapshot\_time}")

            for filename in os.listdir(folder\_path):

                file\_path = os.path.join(folder\_path, filename)

                if os.path.isfile(file\_path):

                    file\_info = get\_file\_info(file\_path)

                    if file\_info[3] > snapshot\_time:

                        print(f"{file\_info[0]} has been changed since snapshot time.")

        else:

            print("Invalid action.")

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

    main()