



**Kolegium Nauk Przyrodniczych  
Uniwersytet Rzeszowski**

**Przedmiot:  
Sieci Komputerowe**

**Dokumentacja projektu:  
Program zliczający pliki o danym  
rozszerzeniu dla podanej ścieżki**

**Wykonał:  
Dawid Pabis 123001**

**Prowadzący: mgr inż. Jarosław Szkoła, prof. UR  
Rzeszów 2023**

## Spis Treści

|     |  |   |
|-----|--|---|
| 1.  | Opis działania kodu (Python).            | 3 |
| 1.1 | Importowanie modułów.                    | 3 |
| 1.2 | Funkcja count_files.                     | 3 |
| 1.3 | Funkcja generate_report.                 | 3 |
| 1.4 | Wywołanie funkcji i generowanie raportu. | 4 |
| 2.  | Wywołanie kodu (Python).                 | 5 |
| 3.  | Opis działania kodu(bat).                | 6 |
| 3.1 | Pobieranie danych.                       | 6 |
| 3.2 | Przetwarzanie plików.                    | 6 |
| 3.3 | Generowanie raportu.                     | 7 |
| 4.  | Wywołanie kodu(bat).                     | 8 |

## 1. Opis działania kodu (Python).

### 1.1 Importowanie modułów.

```
import os
```

Kod importuje moduł `os`, który zapewnia funkcjonalność związane z operacjami na systemie operacyjnym, takimi jak poruszanie się po strukturze katalogów, zarządzanie plikami, etc.

### 1.2 Funkcja `count_files`.

```
1 usage
def count_files(path, extensions):
    count = 0
    for root, dirs, files in os.walk(path):
        for file in files:
            if any(file.endswith(ext) for ext in extensions):
                count += 1
    return count
```

Funkcja `count_files` przyjmuje dwa argumenty: `path` (ścieżka katalogu) i `extensions` (lista rozszerzeń plików). Funkcja ta zlicza pliki o określonych rozszerzeniach w podanym katalogu i zwraca ich liczbę.

### 1.3 Funkcja `generate_report`.

```

def generate_report(count, extensions, output_format, output_directory):
    if output_format == 'txt':
        output_path = os.path.join(output_directory, 'report.txt')
        with open(output_path, 'w') as f:
            f.write(f"Total files: {count}")
            f.write('\n')
            f.write("File count per extension:")
            file_counts = {ext: 0 for ext in extensions}
            for root, dirs, files in os.walk(path):
                for file in files:
                    for ext in extensions:
                        if file.endswith(ext):
                            file_counts[ext] += 1
            for extension, files_count in file_counts.items():
                f.write('\n')
                f.write(f"{extension}: {files_count}")
    elif output_format == 'html':
        output_path = os.path.join(output_directory, 'report.html')
        with open(output_path, 'w') as f:
            f.write('<html><body>')
            f.write(f"<h1>Total files: {count}</h1>")
            f.write(f"<h2>File count per extension:</h2>")
            f.write('<ul>')
            file_counts = {ext: 0 for ext in extensions}
            for root, dirs, files in os.walk(path):
                for file in files:
                    for ext in extensions:
                        if file.endswith(ext):
                            file_counts[ext] += 1
            for extension, files_count in file_counts.items():
                f.write(f"<li>{extension}: {files_count}</li>")
            f.write('</ul>')
            f.write('</body></html>')

```

Funkcja `generate_report` generuje raport dotyczący plików w podanym katalogu na podstawie wcześniej obliczonej liczby plików `count`, listy rozszerzeń `extensions`, wybranego formatu raportu `output_format` i katalogu wyjściowego `output_directory`.

## 1.4 Wywołanie funkcji i generowanie raportu.

```

path = input("Enter the directory path: ")
extensions = input("Enter the file extensions (separated by spaces): ").split()
output_format = input("Enter the output format (txt or html): ")
output_directory = input("Enter the output directory: ")

count = count_files(path, extensions)
generate_report(count, extensions, output_format, output_directory)
print("Report generated successfully.")

```

Kod pobiera od użytkownika ścieżkę do katalogu `path`, listę rozszerzeń plików `extensions`, format raportu `output_format` (txt lub html) i katalog wyjściowy `output_directory`. Następnie wykonuje się funkcja `count_files`, aby obliczyć liczbę plików, a następnie funkcja `generate_report`, aby wygenerować raport w odpowiednim formacie i umieścić go w katalogu wyjściowym. Na koniec wyświetlane jest komunikat o pomyślnym wygenerowaniu raportu.

## 2. Wywołanie kodu (Python).

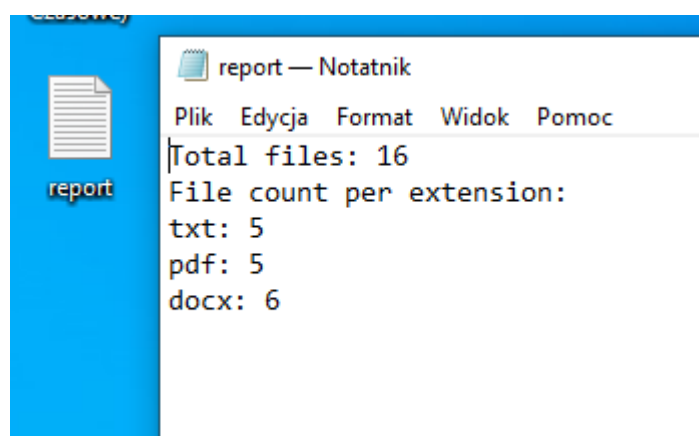
```
C:\Users\HP\PycharmProjects\Analiza\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\Analiza\main.py
Enter the directory path: C:\Users\HP\Desktop\fodder
Enter the file extensions (separated by spaces): txt pdf docx
Enter the output format (txt or html): txt
Enter the output directory: C:\Users\HP\Desktop
Report generated successfully.

Process finished with exit code 0
```

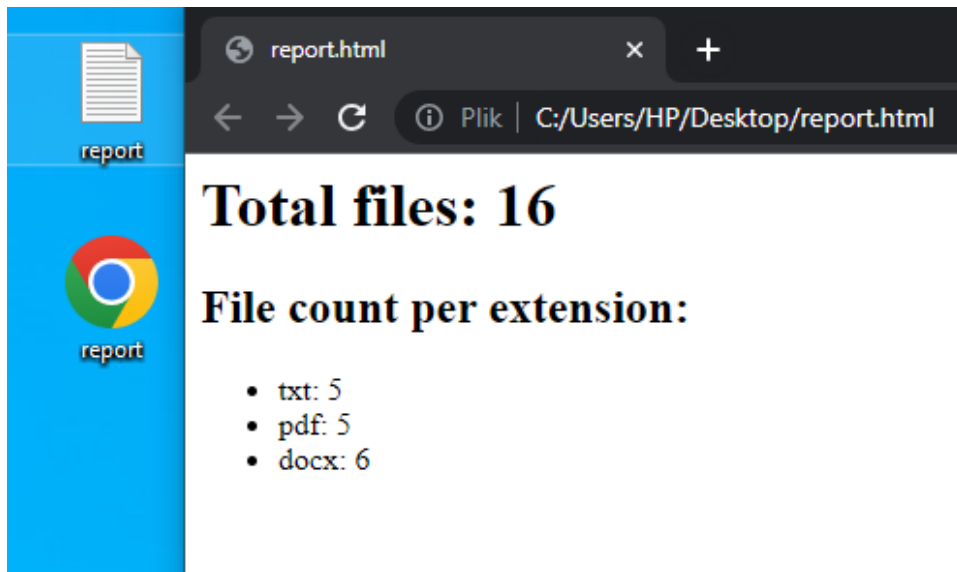
Jako ścieżkę do sprawdzenia został ustawiony losowy folder o nazwie fodder, którego zawartość jest następująca.

| fodder         |                  |                     |          |  |
|----------------|------------------|---------------------|----------|--|
| Nazwa          | Data modyfikacji | Typ                 | Rozmiar  |  |
| if — kopia (2) | 13.06.2023 20:32 | Dokument Office ... | 0 KB     |  |
| if — kopia (3) | 13.06.2023 20:32 | Dokument Office ... | 0 KB     |  |
| if — kopia (4) | 13.06.2023 20:32 | Dokument Office ... | 0 KB     |  |
| if — kopia (5) | 13.06.2023 20:32 | Dokument Office ... | 0 KB     |  |
| if — kopia     | 13.06.2023 20:32 | Dokument Office ... | 0 KB     |  |
| if             | 13.06.2023 20:32 | Dokument Office ... | 0 KB     |  |
| can            | 13.06.2023 20:31 | Dokument tekstowy   | 0 KB     |  |
| code           | 13.06.2023 20:31 | Dokument tekstowy   | 0 KB     |  |
| count          | 13.06.2023 20:31 | Dokument tekstowy   | 0 KB     |  |
| testing        | 13.06.2023 20:31 | Dokument tekstowy   | 0 KB     |  |
| this           | 13.06.2023 20:31 | Dokument tekstowy   | 0 KB     |  |
| Hi — kopia (2) | 06.01.2020 21:56 | Microsoft Edge P... | 1 462 KB |  |
| Hi — kopia (3) | 06.01.2020 21:56 | Microsoft Edge P... | 1 462 KB |  |
| Hi — kopia (4) | 06.01.2020 21:56 | Microsoft Edge P... | 1 462 KB |  |
| Hi — kopia     | 06.01.2020 21:56 | Microsoft Edge P... | 1 462 KB |  |
| Hi             | 06.01.2020 21:56 | Microsoft Edge P... | 1 462 KB |  |

Następnie wybrano by program zliczył pliki typu .txt .pdf i .docx, czyli w tym wypadku wszystkie pliki w tym folderze i ostatecznie wygenerował raport w postaci pliku txt na pulpicie.



Wersja html tego raportu.



### 3. Opis działania kodu(bat).

#### 3.1 Pobieranie danych.

```
@echo off

setlocal enabledelayedexpansion

set /p "path=Enter the directory path: "
set /p "extensions=Enter the file extensions (separated by spaces): "
set /p "output_format=Enter the output format (txt or html): "
set /p "output_directory=Enter the output directory: "
```

Powyższe linie kodu służą do pobierania danych od użytkownika. Składnia `set /p` pozwala na wprowadzenie wartości do zmiennych środowiskowych.

Komenda `@echo off` wyłącza wyświetlanie poleceń w trakcie wykonywania skryptu, co sprawia, że polecenia nie są widoczne dla użytkownika podczas działania skryptu.

#### 3.2 Przetwarzanie plików.

```

set "count=0"
set "report="

for %%E in (%extensions%) do (
    set "file_count=0"
    for /r "%path%" %%F in (*%%E) do (|
        set /a "count+=1"
        set /a "file_count+=1"
    )
    set "report=!report!%%E : !file_count! "
    echo %%E : !file_count!
)

```

Inicjalizowane są dwie zmienne: count (liczba plików) i report (raport).

Powyższy blok kodu przetwarza pliki w podanym katalogu (%path%) o rozszerzeniach wskazanych przez użytkownika (%extensions%). Dla każdego rozszerzenia iteruje się po plikach za pomocą polecenia for /r i zlicza się pliki (file\_count) oraz zwiększa ogólną liczbę plików (count). Dodatkowo, informacja o liczbie plików dla każdego rozszerzenia jest dodawana do zmiennej report i wyświetlana na ekranie.

### 3.3 Generowanie raportu.

```

if "%output_format%"=="txt" (
    echo Total files: %count% > "%output_directory%\report.txt"
    echo File count per extension: >> "%output_directory%\report.txt"
    echo %report% >> "%output_directory%\report.txt"
    echo Report generated successfully. Press any key to exit.
) else if "%output_format%"=="html" (
    echo ^<html^>^<body^> > "%output_directory%\report.html"
    echo ^<h1^>Total files: %count%^</h1^> >> "%output_directory%\report.html"
    echo ^<h2^>File count per extension:^</h2^> >> "%output_directory%\report.html"
    echo ^<ul^> >> "%output_directory%\report.html"
    for %%E in (%extensions%) do (
        echo ^<li^>%%E : !file_count!^</li^> >> "%output_directory%\report.html"
    )
    echo ^</ul^> >> "%output_directory%\report.html"
    echo ^</body^>^</html^> >> "%output_directory%\report.html"
    echo Report generated successfully. Press any key to exit.
) else (
    echo Invalid output format. Please enter txt or html.
)

pause >nul

endlocal

```

Ten fragment kodu generuje raport w zależności od formatu (%output\_format%) wybranego przez użytkownika. Jeśli format to 'txt', raport jest zapisywany do pliku tekstowego report.txt w podanym katalogu wyjściowym (%output\_directory%). Jeśli format to 'html', raport jest zapisywany do pliku HTML report.html. Jeśli format jest nieprawidłowy, wyświetlany jest komunikat o błędzie.

Polecenie pause powoduje zatrzymanie działania skryptu, aż użytkownik naciśnie dowolny klawisz. Następnie, polecenie endlocal kończy działanie skryptu i przywraca pierwotny stan zmiennych środowiskowych.

#### 4. Wywołanie kodu(bat).

```
Enter the directory path: C:\Users\HP\Desktop\folder
Enter the file extensions (separated by spaces): txt pdf docx
Enter the output format (txt or html): html
Enter the output directory: C:\Users\HP\Desktop
txt : 5
pdf : 5
docx : 6
Report generated successfully. Press any key to exit.
```

Do tego wywołania zostały wykorzystane te same pliki co w wypadku pythona.

