Лабораторна робота №6
Робота з файловою системою операційної системи.

Мета роботи -вивчити основні прийоми програмування простих системних утиліт роботи з файловою системою операційної системи.

Завдання:

1. Розробити програму (скрипт) мовою Пітон, з використанням відповідних пакетів, для отримання статистичних даних про файли та папки, що знаходяться на зовнішніх носіях.
2. Модифікувати допоміжний скрипт (див. нижче) для виконання тестування розробленої програми.

Вимоги до програми:

- отримати статистичні дані про файли та підпапки, розташовані в заданій папці;

- у отримані дані включити сумарну інформацію по всіх файлах та підпапках, що знаходяться в заданій папці:

Таблицю №1:

1) найменування папки (підпапки),
2) кількість підпапок у цій папці та їх сумарний розмір,
3) кількість файлів у цій папці та їх сумарний розмір

Таблицю № 2:

1) Ім'я типу файлу та сумарний розмір (і % від усієї кількості) всіх файлів цього типу, виявлених у всіх заданих папках та підпапках;

- Вихідні дані до роботи необхідно отримувати, використовуючи параметри командного рядка;

- В обов'язковому порядку передбачити виведення опису правил використання програми на консоль, використовуючи опцію /? або --help;

Для перевірки працездатності програми рекомендується побудувати набір тестової структури досліджуваного дерева каталогів та файлів у них.

Для цього можна, наприклад, скористатися таким скриптом:

«Допоміжний скрипт побудови тестових директорій»:

```
import pathlib, itertools, glob, shutil
import random
# root dir
path = 'D://tx//'
# type file & max size (Kb)
ftype = {'.py' : 10, '.txt' : 100, '.xyz' : 5, '.bak' : 1}
# sub dir name
tdir = {'script', 'text', 'file', 'ddd'}
# combinations of name of file
combin=[1, 2, 3, 4, 5]
for d in tdir:
    comb = itertools.combinations(combin, r=2)
    for a, b in comb:
        pathlib.Path(path, d).mkdir(parents=True, exist_ok=True)
        for ext, size in ftype.items():
            name = f'{d}{a}{b}{ext}'
            pathlib.Path(path, d, name).touch(exist_ok=True)
            full_name=pathlib.Path(path, d, name)
            f=open(full_name,"wt")
            n=size*int(1024*(random.random()+1))
            buf=n*ext[1:]
            f.write(buf)
            f.close()
```

Успіхів у роботі.

Приклад

Текст програми:

```
import os
import pathlib
import tabulate
import argparse

def list_to_string(s):
    str1 = ""
    for ele in s:
        str1 += ele
        str1 += " "

    return str1

def get_size(start_path='.'):
    total_size = 0
    for dirpath, dirnames, filenames в os.walk(start_path):
        for f in filenames:
            fp = os.path.join(dirpath, f)
            if not os.path.islink(fp):
                total_size += os.path.getsize(fp)

    return total_size

argparse = argparse.ArgumentParser()
argparse.add_argument("--dir",          default="C://Users//Liza//tx",
type=str)
args = argparse.parse_args()
os.chdir(args.dir)

d = dict (); d2 = dict (); d3 = dict (); files_dict = dict()

column_names = []; rows = []; row1 = []
row2 = []; row3 = []; row4 = []; row5 = []

file_ext_table = []
table2_headers  =  ["тип  файлу",  "суммарний  розмір",  "%  від  усієї
кількості"]
table1_headers = ["назва папки", "кількість підпапок",
                  "сумарний розмір підпапок", "кількість файлів",
                  "сумарний розмір файлів"]

num_files = 0; total_size = 0

##
for dirpath, dirnames, filenames в os.walk("."):
    file_count = 0
    total_size_in_dir = 0
    path = os.path.basename(dirpath)
    column_names.append(f"{path}/")
    rows.append(len(dirnames))

    if len (dirnames)! = 0:
        size = get_size()
    else:
```

```
            size = 0
        row1.append(size)
        row2.append(len(os.listdir(dirpath)) - len(dirnames))

        for i in filenames:
            fp = os.path.join(dirpath, i)
            total_size_in_dir += os.path.getsize(fp)
            fileExt = pathlib.Path(list_to_string(
                filenames[file_count])).suffix.replace(" ", "")

            filePath = os.path.join(dirpath, filenames[file_count])
            fileSize = os.path.getsize(filePath)
            if fileExt in files_dict:
                files_dict[fileExt]["sum"] += fileSize
                files_dict[fileExt]["num"] += 1

            else:
                files_dict[fileExt] = dict()
                files_dict[fileExt]["sum"] = fileSize
                files_dict[fileExt]["num"] = 1

            file_count += 1

        row3.append(total_size_in_dir)
        total_size += total_size_in_dir
        num_files += file_count

for file_ext in files_dict:
        sum = files_dict[file_ext]["sum"]
        num = (files_dict[file_ext]["num"] / num_files) * 100
        row = [file_ext, sum, num]
        file_ext_table.append(row)

d["column_names"] = column_names; d["rows"] = rows
d["row"] = row1; d["row2"] = row2; d["row3"] = row3
d2["rows4"] = row4; d2["rows5"] = row5

print(tabulate.tabulate(d,headers=table1_headers, tablefmt="pipe"))
print()
print(tabulate.tabulate(file_ext_table, headers=table2_headers,
tablefmt="pipe"))
```

Результат роботи:

название папки	кол-во подпапок	суммарный размер подпапок	кол-во файлов	суммарный размер файлов
./	3	4551501	0	0
AppInfo/	0	0	6	58946
DefaultData/	1	4551501	0	0
settings/	0	0	1	10240
DOSBox/	2	4551501	9	4263576
Documentation/	0	0	6	118880
Video Codec/	0	0	3	99859

тип файла	суммарный размер	% от всего количества
.ico	22486	4
.png	35801	12
.ini	659	8
.conf	10240	4
.txt	194973	36
.bat	188	16
.exe	3727360	4
.dll	555751	12
.inf	4043	4