Практическое задание №13.

Настройка функций и модулей. Обработка исключений и работа с файлами.

Цель: научиться использовать современные методы контроля производительности информационно-коммуникационной системы и локализовывать отказ и инициировать корректирующие действия

Задание:

Необходимо создать и редактировать, в соответствии с заданием, скрипты на языке программирования Python.

Инструкция:

Шаг 1: Написать python скрипт анализа размера каждой директории и файла в текущей директории с выводом результатов в терминал, используя функции и циклы;

Шаг 2: Добавить сортировку по уменьшению размера;

Шаг 3: Прикрепить скриншот вывода работы скрипта.

Шаг 4: Прикрепить файл скрипта.

Оформите выполнение в текстовый документ, где распишите выполнение каждого шага и прикрепите скриншоты.

Выполнение

Шаг 1: Написать python скрипт анализа размера каждой директории и файла в текущей директории с выводом результатов в терминал, используя функции и циклы;

```
в терминале: создаем файл скрипта: mcedit size.py
в mc: импортируем os модуль: import os
если запускаемый модуль main, тогда запускаем функцию main: if name == " main ":
  main()
определяем функцию: def main():
получаем текущую директорию: pwd = os.getcwd()
получаем список всех файлов и директорий: items = os.listdir(pwd)
делаем цикл: for item in items:
вывод: print (item)
добавляем размер при помощи функции def get size(path):
добавляем условие если: if os.path.isfile(path):
    size = os.path.getsize(path)
иначе: else:
    for dirpath, dirname, filenames in os.walk(path):
то возвращаем размер: return size
далее цикл:
                  for filename in filenames:
соединяем пути: fp = os.path.join(dirpath, filename)
добавляем условие (если это файл, то суммируем размер): if os.path.isfile(fp):
           size += os.path.getsize(fp)
добавим полный путь и размер к циклу for item in items
full path = os.path.join(pwd, item)
```

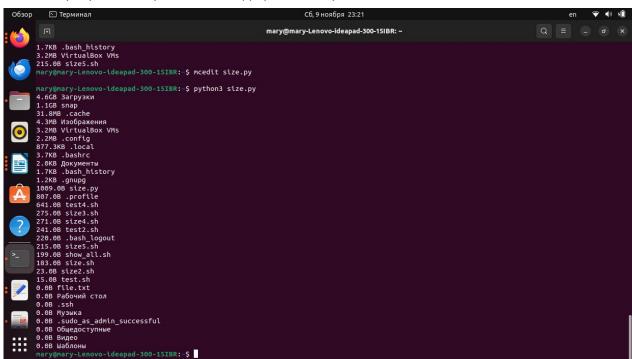
```
сделаем вывод: print("{} {}".format(size, item))
для удобства чтения добавим функцию: def human_readable_size(size):
и цикл для определения единиц измерения for unit in ['B', 'KB', 'MB', 'GB', 'TB']:
добавляем условие если: if size < 1024:
то прервать цикл: break
если больше или равен, то делим на 1024: size /= 1024
возвращаем размер и ед измерения: return "{:.1f}{}".format(size, unit)
```

Шаг 2: Добавить сортировку по уменьшению размера;

size = get size(full path)

```
создадим переменную: size_list = [] и добавим вместо вывода: size_list.append((size, item)) отсортируем по уменьшению размера с помощью функции: size_list.sort(key=lambda x: x[0], reverse=True) цикл: for size, item in size_list: вывод списка директорий и файлов: print("{} {}".format(human_readable_size(size), item))
```

Шаг 3: Прикрепить скриншот вывода работы скрипта.



шаг 4: Прикрепить файл скрипта.

