

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3
дисциплины «Основы программной инженерии»

Выполнил:
Магомедов Имран Борисович
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., кандидат технических
наук, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

ТЕМА: Работа с файлами в языке Python

Цель работы – приобретение навыков по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучение основных методов модуля os для работы с файловой системой, получение аргументов командной строки.

МЕТОДИКА И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk ().*

Owner *

Repository name *


 m4g0med0v

/ BSE_Python_LR_2.15


 BSE_Python_LR_2.15 is available.

Great repository names are short and memorable. Need inspiration? How about [scaling-octo-goggles](#) ?

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: Python

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: MIT License

3. Выполните клонирование созданного репозитория.

```
~/work git clone git@github.com:m4g0med0v/BSE_Python_LR_2.15.git ✓
Клонирование в «BSE_Python_LR_2.15»...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Получение объектов: 100% (5/5), готово.
```

4. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```
~/w/BSE_Python_LR_2.15 / main git checkout -bdevelop 1 ✗
Переключились на новую ветку «develop»
```

5. Проработать примеры лабораторной работы.

```
example_17.py X
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import os
5 import secrets
6 import string
7 import sys
8
9
10 if __name__ == "__main__":
11     if len(sys.argv) != 2:
12         print("The password length is not given!", file=sys.stderr)
13         sys.exit(1)
14
15     chars = string.ascii_letters + string.punctuation + string.digits
16     length_pwd = int(sys.argv[1])
17
18     result = []
19     for _ in range(length_pwd):
20         idx = secrets.SystemRandom().randrange(len(chars))
21         result.append(chars[idx])
22
23     print(f"Secret Password: {''.join(result)}")
```

6. Выполнить индивидуальные задания.

Индивидуальное задание №1

Написать программу, которая считывает английский текст из файла и выводит его на экран, заменив каждую первую букву слов, начинающихся с гласной буквы, на прописную.

Код программы:

```
import sys

def capitalize_vowels(text):
    words = text.split()
    for i in range(len(words)):
        if words[i] and (words[i][0] in 'aeiouAEIOU'):
            words[i] = words[i].capitalize()
    return ' '.join(words)

def main():
    try:
        with open(sys.argv[1], 'r') as file:
            text = file.read()
            modified_text = capitalize_vowels(text)
            print(modified_text)
    except FileNotFoundError:
        print(f"Ошибка: Файл '{sys.argv[1]}' не найден.")

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

Файл, который будет дан на вход для программы:

```
this is a sample text to demonstrate the functionality of the program.
it will capitalize the first letter of words starting with vowels.
```

Вывод программы:

```
~/w/BSE_Python_LR_2.15 / git develop ?6 python3 individual_task_1.py te
xt_to_capitalize.txt
this Is A sample text to demonstrate the functionality Of the program. It will c
apitalize the first letter Of words starting with vowels.
```

Индивидуальное задание №2

Автоматическая проверка орфографии не помешала бы многим из нас. В данном упражнении мы напишем простую программу, сверяющую слова из текстового файла со словарем. Неправильно написанными будем считать все слова, которых не нашлось в словаре. Имя файла, в котором требуется выполнить орфографическую проверку, пользователь должен передать при помощи аргумента командной строки. В случае отсутствия аргумента должна выдаваться соответствующая ошибка. Сообщение об ошибке также должно появляться, если не удастся открыть указанный пользователем файл. Также Вам следует игнорировать регистр символов при выполнении проверки.

Код программы:

```
import sys

def load_dictionary(file_name):
    dictionary = set()
    try:
        with open(file_name, 'r', encoding='windows-1251') as file:
            for line in file:
                dictionary.add(line.strip().lower())
    except FileNotFoundError:
        print(f"Ошибка: Файл '{file_name}' не найден.")
        sys.exit(1)
    return dictionary

def spell_check(file_name, dictionary):
    try:
        with open(file_name, 'r', encoding='utf-8') as file:
            for line_number, line in enumerate(file, start=1):
                words = line.strip().split()
                for word in words:
                    if word.lower() not in dictionary:
                        print(f"Ошибка в строке {line_number}: слово '{word}' не найдено в словаре.")
    except FileNotFoundError:
        print(f"Ошибка: Файл '{file_name}' не найден.")
        sys.exit(1)

def main():
    if len(sys.argv) != 2:
        print("Использование: python spell_checker.py <имя_файла>")
        sys.exit(1)

    file_name = sys.argv[1]

    dictionary = load_dictionary("dictionary.txt")

    spell_check(file_name, dictionary)

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

Файл, который будем проверять на орфографию:

Это пример текста, который содержит некоторые слова с неправильной орфографией. Я специально включил некоторые ошибки, чтобы проверить нью программу.

Вывод программы:

```
~/work/BSE_Python_LR_2.15 / git develop ?4 python3 individual_task_1.py text_to_check.txt
Ошибка в строке 1: слово 'текста,' не найдено в словаре.
Ошибка в строке 1: слово 'орфографией.' не найдено в словаре.
Ошибка в строке 2: слово 'ошибки,' не найдено в словаре.
Ошибка в строке 2: слово 'проверить' не найдено в словаре.
Ошибка в строке 2: слово 'нью' не найдено в словаре.
Ошибка в строке 2: слово 'программу.' не найдено в словаре.
```

7. Зафиксируйте изменения в репозитории.
8. Самостоятельно подберите или придумайте задачу для работы с изученными функциями модуля `os`. Приведите решение этой задачи.

Написать программу, которая сканирует заданную директорию и выводит список всех файлов и папок в этой директории, а также их размеры. Затем программа должна определить общий размер всех файлов в директории.

Код программы:

```

import os, sys

def get_directory_contents(directory):
    contents = []
    for item in os.listdir(directory):
        full_path = os.path.join(directory, item)
        if os.path.isfile(full_path):
            size = os.path.getsize(full_path)
            contents.append((item, "File", size))
        elif os.path.isdir(full_path):
            contents.append((item, "Directory", "-"))
    return contents

def calculate_total_size(directory):
    total_size = 0
    for item in os.listdir(directory):
        full_path = os.path.join(directory, item)
        if os.path.isfile(full_path):
            total_size += os.path.getsize(full_path)
    return total_size

def main():
    directory = sys.argv[1]

    if not os.path.exists(directory):
        print(f"Ошибка: Директория '{directory}' не существует.")
        return

    contents = get_directory_contents(directory)

    print("Содержимое директории:")
    print("{:<30} {:<10} {:<10}".format("Имя", "Тип", "Размер"))
    for item in contents:
        print("{:<30} {:<10} {:<10}".format(item[0], item[1], item[2]))

    total_size = calculate_total_size(directory)
    print(f"\nОбщий размер всех файлов в директории: {total_size} байт")

if __name__ == "__main__":
    main()

```

Вывод программы:

```
~/w/BSE_Python_LR_2.15 / develop ?1 python task_3.py /bin ✓
Содержимое директории:
Имя                Тип      Размер
kdestroy           File     14280
as                 File     798784
conda-env          File     142
alsoft-config      File     211136
newuidmap          File     37456
rav1e              File     4488312
ss                 File     127704
sndfile-metadata-get File     30752
ModemManager       File     2259520
defrag.f2fs        File     227888
lz4c               File     1401944
modinfo            File     166008
e2undo             File     22472
teamdctl           File     32520
pldd               File     22768
fix-qdf            File     179320

modeprint          File     18448
nl-list-caches     File     14288
llvm-cxxdump       File     83992
systemd-hwdb       File     14336
nice               File     34968
ffmpeg            File     321640
pps                File     18800
gtk4-query-settings File     14288
rapper             File     26640
electron27         File     403
djvuserve          File     43136
kernel-install     File     55528
showmount          File     14552
smb2-quota         File     6458
transmission-remote File     412008
sudo_logsrvd       File     266400
llvm-modextract    File     26640

Общий размер всех файлов в директории: 843603691 байт
```

9. Зафиксируйте изменения в репозитории.
10. Добавьте отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксируйте изменения.
11. Выполните слияние ветки для разработки с веткой master/main.


```
~/work/BSE_Python_LR_2.15 / main git merge develop
Обновление a990cf0..d131854
Fast-forward
dictionary.txt | 1532629 ++++++
examples/example_1.py | 7 +
examples/example_10.py | 4 +
examples/example_11.py | 4 +
examples/example_12.py | 4 +
examples/example_13.py | 6 +
examples/example_14.py | 4 +
examples/example_15.py | 9 +
examples/example_16.py | 10 +
examples/example_17.py | 23 +
examples/example_2.py | 4 +
examples/example_3.py | 9 +
examples/example_4.py | 6 +
examples/example_5.py | 6 +
examples/example_6.py | 19 +
examples/example_7.py | 11 +
examples/example_8.py | 10 +
examples/example_9.py | 4 +
individual_task_1.py | 30 +
individual_task_2.py | 54 +
task_3.py | 47 +
text_to_check_spell.txt | 2 +
text_to_capitalize.txt | 2 +
23 files changed, 1532904 insertions(+)
create mode 100644 dictionary.txt
create mode 100644 examples/example_1.py
create mode 100644 examples/example_10.py
create mode 100644 examples/example_11.py
```

12. Отправьте сделанные изменения на сервер GitHub.
13. Отправьте адрес репозитория GitHub на электронный адрес преподавателя.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как открыть файл в языке Python только для чтения?

Для открытия файла только для чтения в Python используется функция `open()` с параметром `r`. Например:

```
with open('file.txt', 'r') as f:
```

2. Как открыть файл в языке Python только для записи?

Для открытия файла только для записи в Python используется функция `open()` с параметром `w`. Например:

```
with open('file.txt', 'w') as f:
```

3. Как прочитать данные из файла в языке Python?

Для чтения данных из файла в Python можно использовать метод `read()` объекта файла. Например:

```
with open('file.txt', 'r') as f:
    data = f.read()
```

```
print(data)
```

4. Как записать данные в файл в языке Python?

Для записи данных в файл в Python можно использовать метод `write()` объекта файла. Например:

```
with open('file.txt', 'w') as f:  
    f.write('Hello, world!')
```

5. Как закрыть файл в языке Python?

Файл автоматически закрывается при выходе из блока `with`, но если вы работаете без контекстного менеджера, то файл можно закрыть вручную с помощью метода `close()`. Например:

```
f = open('file.txt', 'r')  
f.close()
```

6. Изучите самостоятельно работу конструкции `with ... as`. Каково ее назначение в языке Python? Где она может быть использована еще, помимо работы с файлами?

Конструкция `with ... as` используется для автоматического управления ресурсами, таких как файлы или сетевые соединения. Она гарантирует, что ресурс будет корректно закрыт после завершения работы с ним, даже в случае возникновения исключений. Помимо работы с файлами, `with ... as` может быть использована, например, для работы с сетевыми соединениями, базами данных и т. д.

7. Изучите самостоятельно документацию Python по работе с файлами. Какие помимо рассмотренных существуют методы записи/чтения информации из файла?

Помимо методов `read()` и `write()`, существуют другие методы для работы с файлами в Python. Например:

- `readline()`: чтение одной строки из файла.

- `readlines()`: чтение всех строк файла в список.
- `writelines(lines)`: запись списка строк в файл.
- `seek(offset, whence)`: перемещение указателя файла.
- `tell()`: получение текущей позиции указателя в файле.

8. Какие существуют, помимо рассмотренных, функции модуля `os` для работы с файловой системой?

Помимо функций `open()` и `close()`, модуль `os` предоставляет другие функции для работы с файловой системой. Некоторые из них:

- `os.rename(src, dst)`: переименование файла или директории.
- `os.remove(path)`: удаление файла.
- `os.mkdir(path)`: создание директории.
- `os.rmdir(path)`: удаление директории.
- `os.path.exists(path)`: проверка существования файла или директории.
- `os.path.isfile(path)`: проверка, является ли путь файлом.
- `os.path.isdir(path)`: проверка, является ли путь директорией.