

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Програмування

Лабораторна робота №7

«Обробка виключень та робота з файлами»

Виконала:
Студентка групи ІО-15
Кушнерик Є.О.
Залікова книжка №1508

Перевірив:
Пономаренко А. М.

Київ – 2021

Лабораторна робота №7

Тема: «Обробка виключень та робота з файлами».

Мета: вивчити основні способи роботи з виключеннями. Виключення користувача. Відкриття файлів, зчитування та запис у файл. Шляхи доступу до файлів. Функції, методи та атрибути для роботи з файлами.

Загальне завдання:

1. Вивчити матеріал лекцій 24, 25, 26 та 27.
2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

Короткі теоретичні відомості по темі лабораторної роботи:

Обробка виключень

Виключення – це повідомлення інтерпретатора, які виникають у випадку виникнення помилки в програмному коді або при настанні якоїнебудь події. Якщо в коді не передбачена обробка виключення, то виконання програми переривається, і виводиться повідомлення про помилку.

Існують три типи помилок у програмі:

Синтаксичні помилки – це помилки в імені оператора або функції, відсутність закриваючих або відкриваючих лапок і т. д., тобто помилки в синтаксисі мови.

Логічні помилки – це помилки в логіці програми, які можна виявити тільки за результатами її роботи.

Помилки часу виконання – це помилки, які виникають під час роботи програми. Причиною є події, не передбачені програмістом.

Інструкція try...except...else...finally

Для обробки виключень призначена інструкція try.

Робота з файлами

Відкриття файлу

Функція відкриття має наступний формат:

```
open(<Шлях до файлу>[, mode='r'] [, buffering=-1) [,
encoding=None) [,errors =None] [, newline=None) [,
closefd=True])
```

Відносний шлях буде автоматично перетворений в абсолютний шлях за допомогою функції `abspath()` з модуля `os.path`. Можливі наступні варіанти:

1. Якщо файл, що відкривається, перебуває в поточному робочому каталозі, то можна вказати тільки назву файлу.
2. Якщо файл, що відкривається, розташований у вкладеній папці, то перед назвою файлу приводять назви вкладених папок через слеш.
3. Якщо папка з файлом розташована вище рівнем, то перед назвою файлу вказують дві крапки й слеш ("`.. \`").
4. Якщо на початку шляху розташований слеш, то шлях відлічується від кореня диска. У цьому випадку місце розташування поточного робочого каталогу не має значення.

В абсолютному й відносному шляхах допускається вказувати як прямі, так і зворотні слеші. Усі вони будуть автоматично перетворені з урахуванням значення атрибута `sep` з модуля `os.path`. Значення цього атрибута залежить від використовуваної операційної системи.

Необов'язковий параметр `mode` у функції `open()` може приймати наступні значення:

`r` – тільки читання (значення за замовчуванням). `r+` – читання й запис. `w` – запис.
`w+` – читання й запис. `a` – запис. `a+` – читання й запис.
`x` – створення файлу для запису.
`x+` – створення файлу для читання й запису.
Після вказівки режиму може слідувати модифікатор:

`b` – файл буде відкритий у бінарному режимі. Файлові методи приймають і повертають об'єкти типу `bytes`;

`t` – файл буде відкритий у текстовому режимі (значення за замовчуванням у Windows).

У параметрі `errors` можна вказати рівень обробки помилок. Можливі значення:

`"strict"` (при помилці виконується виключення `ValueError` – значення за замовчуванням),

`"replace"` (невідомий символ замінюється символом питання або символом з кодом `\ufffd`),

`"ignore"` (невідомі символи ігноруються),

`"xmlcharrefreplace"` (невідомий символ замінюється послідовністю `&#xxxx;`);

`"backslashreplace"` (невідомий символ замінюється послідовністю `\uxxxx`).

Параметр `newline` задає режим обробки символів кінця рядків. Підтримувані ним значення такі:

- `None` (значення за замовчуванням) – виконується стандартна обробка символів кінця рядка. Наприклад, в Windows при читанні символи `\r\n` перетворюються в символ `\n`, а при записі проводиться зворотне перетворення;
- `" "` (порожній рядок) – обробка символів кінця рядка не виконується;
- `"<Спеціальний символ>"` – зазначений спеціальний символ використовується для позначення кінця рядка, і додаткова обробка не виконується. Як спеціальний символ можна вказати лише `\r\n`, `\r` і `\n`.

Методи для роботи з файлами

- `close()` – закриває файл.
- `write (<дані>)` – записує рядок або послідовність байтів у файл.
- `writelines (<Послідовність>)` – записує послідовність у файл.
- `writable()` – повертає `True`, якщо файл підтримує запис, і `False` – якщо ні.
- `read([<Кількість>])` – зчитує дані з файлу.

- `readline([<Кількість>])` – зчитує з файлу один рядок при кожному виклику.

- `readlines()` – зчитує весь вміст файлу в список.

- `flush()` – примусово записує дані з буфера на диск;

- `fileno()` – повертає цілочисельний дескриптор файлу. Значення, що повертається, завжди буде більшим за число 2, оскільки число 0 закріплене за стандартним вводом `stdin`, 1 – за стандартним виводом `stdout`, а 2 – за стандартним виводом повідомлень про помилки `stderr`.

- `truncate([<Кількість>])` – обрізає файл до зазначеної кількості символів (якщо заданий текстовий режим) або байтів (у випадку бінарного режиму).

- `tell()` – повертає позицію покажчика відносно початку файлу у вигляді цілого числа.

- `seek(<Зсув>[, <Позиція>])` – установлює покажчик у позицію, що має зсув `<Зсув>` відносно позиції `<Позиція>`. У параметрі `<Позиція>` можуть бути зазначені наступні атрибути з модуля `io` або відповідні їм значення:

- `io.SEEK_SET` або 0 – початок файлу (значення за замовчуванням);

- `io.SEEK_CUR` або 1 – поточна позиція покажчика. Додатне значення зсуву викликає переміщення до кінця файлу, від'ємне - до його початку;

- `io.SEEK_END` або 2 – кінець файлу.

- `seekable()` – повертає `True`, якщо покажчик файлу можна зсунути в іншу позицію, і `False` – якщо ні.

Крім методів, об'єкти файлів підтримують кілька [атрибутів](#):

`name` – ім'я файлу; `mode` – режим, у якому був відкритий файл; `closed` – повертає `True`, якщо файл був закритий, і `False` – якщо ні. `encoding` – назва кодування, яке буде використовуватися для перетворення рядків перед записом у файл або при читанні.

Стандартний вивід `stdout` також є файловим об'єктом. Атрибут `encoding` цього об'єкта завжди містить кодування пристроїв виводу, тому рядок перетвориться в послідовність байтів у правильному кодуванні.

`buffer` – дозволяє одержати доступ до буфера. Атрибут доступний тільки в текстовому режимі. За допомогою цього об'єкта можна записати послідовність байтів у текстовий потік.

Індивідуальні завдання:

1. Написати програму створення каталогу зі шляхом та назвою: «C:\lab7\»
2. Написати програму створення підкаталогу «C:\lab7\<прізвище>»
3. Завантажити в даний підкаталог файл *.txt, де * – номер Вашого варіанту лабораторної роботи та виконати з ним дії, що описані в номері Вашого варіанту.
4. Зберегти об'єкти з даними, які створені Вами при виконанні лабораторної роботи №5, у файл, користуючись модулем `pickle`. Створений файл перемістити в попередньо створений каталог «C:\lab5». Зчитати файл, доповнити даними та записати в даний каталог з іншим ім'ям.
5. Зберегти об'єкти з даними, які створені вами при виконанні лабораторної роботи №6, у файл, користуючись модулем `shelve`. Файл перемістити в попередньо створений каталог «C:\lab6». Застосувати три відомі вам методи до модифікації файлу. Вивести на друк модифікований файл.
6. Для виконання наступного завдання вибрати файл, який відповідає номеру варіанта з архіву, що розміщений на сайті!!!.

Використання своїх файлів заборонено!!

Варіант 8:

Зчитати файл «8.txt» та перетворити його у файл «81.txt», який складається з відсортованих за довжиною речень, та у файл «82.txt», який містить слова, що починаються з заданої букви. Зберегти цей файл у кодуванні UTF-8.

Код програми **second.py**

```
import pickle
import shelve
import shutil
import os
import re
from lab5 import planet_dict
import lab6

try:
    os.mkdir(r'C:\\lab7')
except FileExistsError:
    print('Папка lab7 вже існує!')

try:
    os.mkdir(r'C:\\lab7\\Kushneryk')
except FileExistsError:
    print('Папка Kushneryk вже існує!!')

try:
    shutil.copyfile(r'C:\\Users\\HP840\\Desktop\\8.txt', r'C:\\lab7\\Kushneryk\\8.txt')
    print('Файл "8.txt" скопійований з Desktop до папки Kushneryk')
except:
    print('Не вдалось скопіювати файл "8.txt"')

try:
    os.mkdir(r'C:\\lab5')
except FileExistsError:
    print('Папка lab5 вже існує!')

with open('lab5.txt', 'wb') as file:
    pickle.dump(planet_dict, file)

try:
    shutil.move('lab5.txt', r'C:\\lab5\\lab5.txt')
except FileNotFoundError:
    print('Файлу не існує!')
else:
    print('Файл скопійований')
```

```

with open(r'C:\\lab5\\lab5.txt', 'rb') as file:
    newp = pickle.load(file)
    newp['Додаткова планета'] = (11.11, 22.22, 33.33)
with open(os.path.abspath('C:\\lab5\\lab_5.txt'), 'wb') as file:
    pickle.dump(newp, file)

try:
    os.mkdir(r'C:\\lab6')
except FileExistsError:
    print('Папка lab6 вже існує!!')

with shelve.open('C:\\lab6\\lab6.txt', flag='n') as file:
    for i in lab6.planets:
        file.update({i:lab6.planets[i]})
    file.popitem()
    file.popitem()
    file.update({"Додаткова планета": (9, 11.11, 22.22, 33.33, 44.44)})
    file.pop("Нептун")
    print("Вміст файлу lab6.txt")
    for i in file:
        print("{:20}{}".format(i, file[i]))

with open(r'C:\\lab7\\Kushneryk\\8.txt', 'r') as file:
    list = file.read()
    list1 = list.replace('\n', '')
    list2 = list1.replace(' ', '')

sen = sorted(re.split(r'(?<!\w\.\w.) (?<![A-Z][a-z]\.)(?<=\.|\|?)\s', list2), key=len)

with open(r'C:\\lab7\\Kushneryk\\81.txt', 'w') as file:
    file.write('\n'.join(sen))

a = input("Введіть букву, з якої повинно починатися слово в тексті: ")
list3 = list2.split(" ")
with open(r'C:\\lab7\\Kushneryk\\82.txt', 'w', encoding='utf-8') as file:
    new_str = str()
    for elem in (list3):
        if len(elem) == 0:
            elem = "Нічого"
        elif a.lower() == elem[0].lower():
            new_str += elem + "\n"
    file.write(new_str)

```

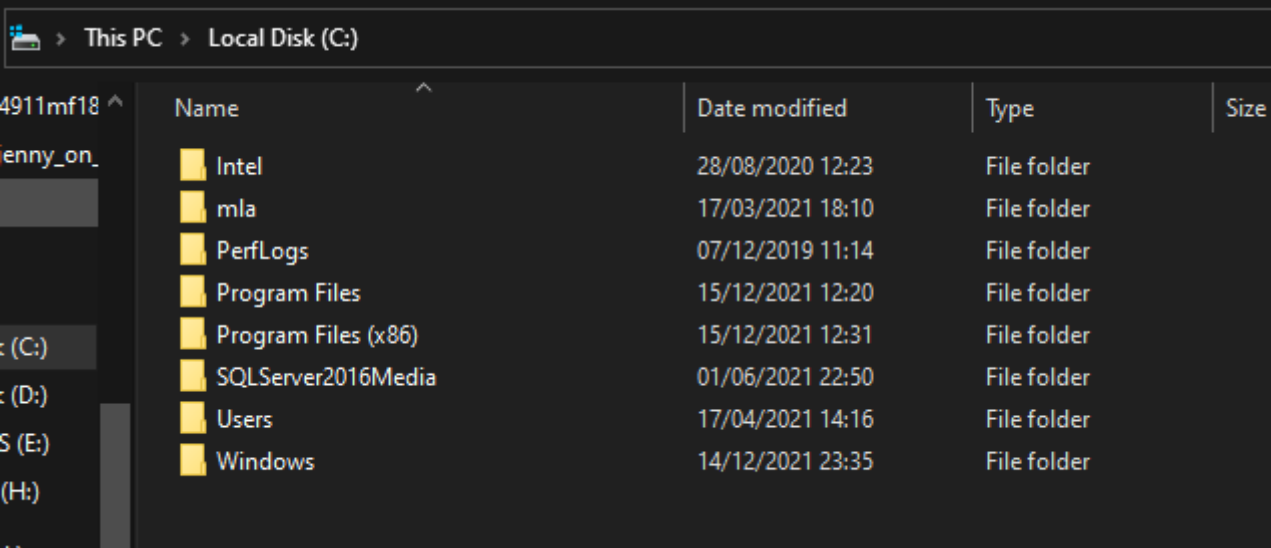

lab5.py

```
planet_dict = { "Меркурій" : (46001200, 6.083e10, 47.87),
                "Венера" : (107476259, 9.38e11, 35.02),
                "Земля" : (147098290, 10.8321e11, 29.783),
                "Марс" : (2.06655e8, 1.6318e11, 24.13),
                "Юпітер" : (7.405736e8, 1.43128e15, 13.07),
                "Сатурн" : (1353572956, 8.2713e14, 9.69),
                "Уран" : (2748938461, 6.833e13, 6.81),
                "Нептун" : (4452940833, 6.254e13, 5.4349) }
```

lab6.py

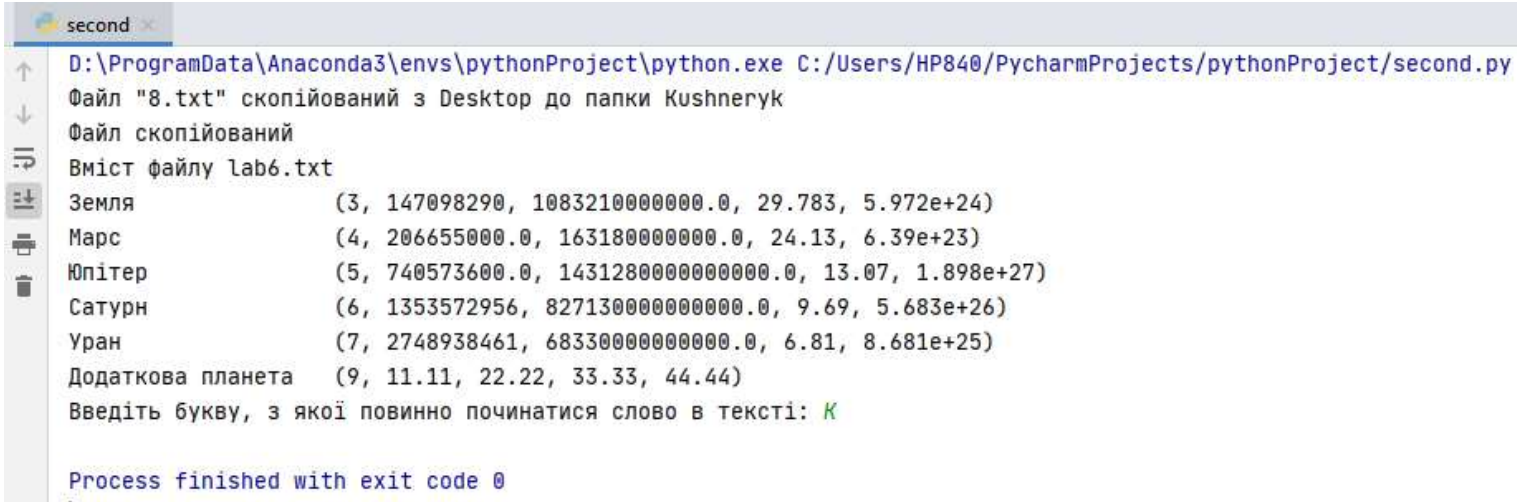
```
class Planet:
    def __init__(self, name, num, distance, volume, speed, mass):
        self.name = name
        self.num = num
        self.volume = volume
        self.mass = mass
        self.speed = speed
        self.distance = distance
    def density(self):
        return self.mass/self.volume
    def ratiomass(self):
        return zem.mass/self.mass
    def ratiospeed(self):
        return zem.speed/self.speed
mer = Planet("Меркурій", 1, 46001200, 6.083e10, 47.87, 3.285e23)
ven = Planet("Венера", 2, 107476259, 9.38e11, 35.02, 4.867e24)
zem = Planet("Земля", 3, 147098290, 10.8321e11, 29.783, 5.972e24)
mar = Planet("Марс", 4, 2.06655e8, 1.6318e11, 24.13, 6.39e23)
yup = Planet("Юпітер", 5, 7.405736e8, 1.43128e15, 13.07, 1.898e27)
sat = Planet("Сатурн", 6, 1353572956, 8.2713e14, 9.69, 5.683e26)
ura = Planet("Уран", 7, 2748938461, 6.833e13, 6.81, 8.681e25)
nep = Planet("Нептун", 8, 4452940833, 6.254e13, 5.4349, 1.024e26)
planets = dict({mer.name:(mer.num, mer.distance, mer.volume, mer.speed,mer.mass),
                ven.name:(ven.num, ven.distance, ven.volume, ven.speed,ven.mass),
                zem.name:(zem.num, zem.distance, zem.volume, zem.speed,zem.mass),
                mar.name:(mar.num, mar.distance, mar.volume, mar.speed,mar.mass),
                yup.name:(yup.num, yup.distance, yup.volume, yup.speed,yup.mass),
                sat.name:(sat.num, sat.distance, sat.volume, sat.speed,sat.mass),
                ura.name:(ura.num, ura.distance, ura.volume, ura.speed,ura.mass),
                nep.name:(nep.num, nep.distance, nep.volume, nep.speed,nep.mass) })
```

Результат програми



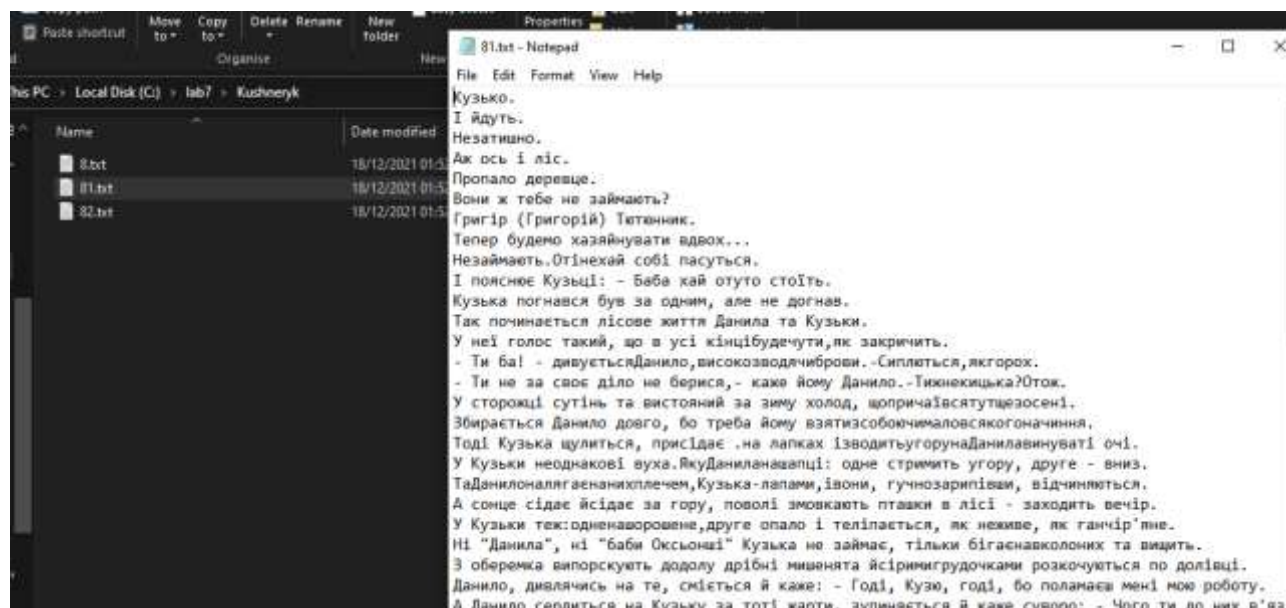
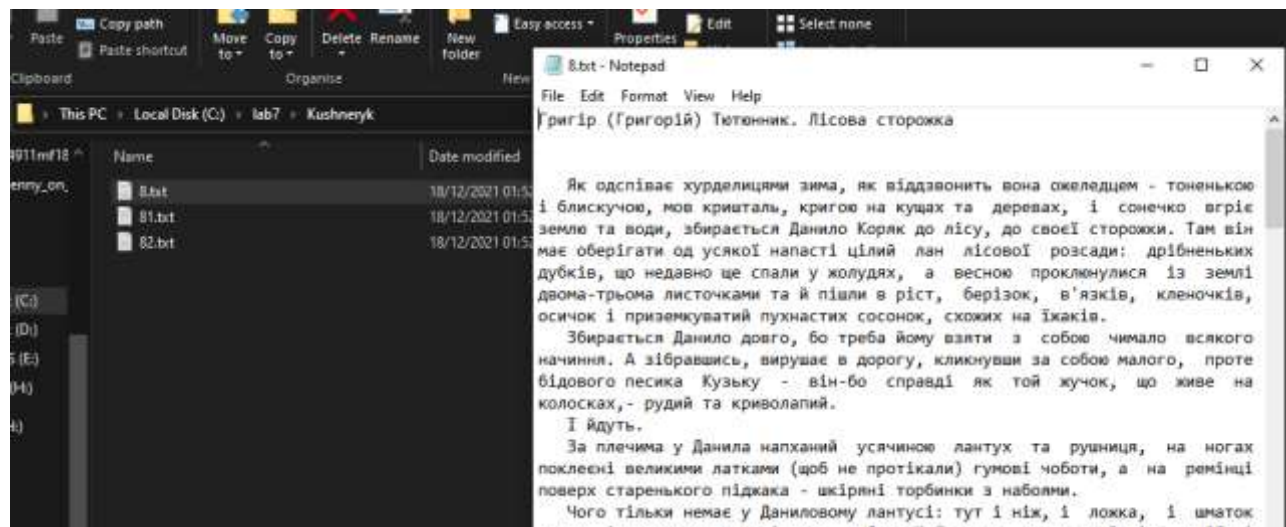
This screenshot shows a Windows File Explorer window with the address bar set to 'This PC > Local Disk (C:)'. The main pane displays a list of folders on the C: drive. The left sidebar shows the navigation pane with 'Local Disk (C:)' selected. The right pane shows a table of folders with columns for Name, Date modified, Type, and Size.

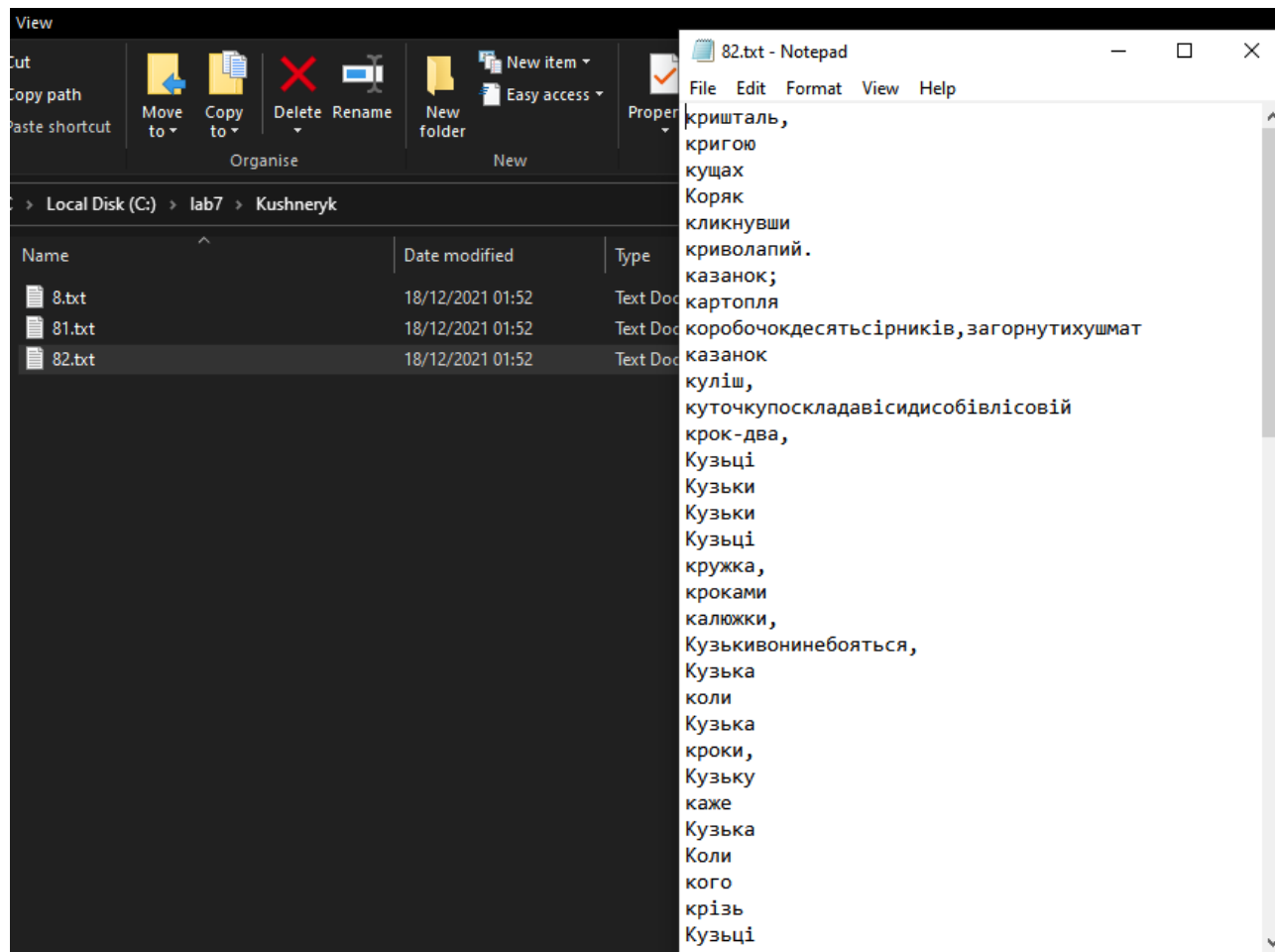
Name	Date modified	Type	Size
Intel	28/08/2020 12:23	File folder	
mla	17/03/2021 18:10	File folder	
PerfLogs	07/12/2019 11:14	File folder	
Program Files	15/12/2021 12:20	File folder	
Program Files (x86)	15/12/2021 12:31	File folder	
SQLServer2016Media	01/06/2021 22:50	File folder	
Users	17/04/2021 14:16	File folder	
Windows	14/12/2021 23:35	File folder	



This screenshot shows a terminal window with a single tab titled 'second'. The terminal output shows the execution of a Python script. The first line is the command to run 'second.py' using the Python interpreter from an Anaconda environment. Subsequent lines show the script's execution, including file copying and the printing of a table of planetary data. The table has two columns: planet names in Ukrainian and their corresponding numerical data. The script ends with a prompt for a letter and a confirmation message. The final line shows the process finished with exit code 0.

```
D:\ProgramData\Anaconda3\envs\pythonProject\python.exe C:/Users/HP840/PycharmProjects/pythonProject/second.py
Файл "8.txt" скопійований з Desktop до папки Kushneryk
Файл скопійований
Вміст файлу lab6.txt
Земля      (3, 147098290, 1083210000000.0, 29.783, 5.972e+24)
Марс      (4, 206655000.0, 163180000000.0, 24.13, 6.39e+23)
Юпітер    (5, 740573600.0, 1431280000000000.0, 13.07, 1.898e+27)
Сатурн    (6, 1353572956, 827130000000000.0, 9.69, 5.683e+26)
Уран      (7, 2748938461, 6833000000000.0, 6.81, 8.681e+25)
Додаткова планета (9, 11.11, 22.22, 33.33, 44.44)
Введіть букву, з якої повинно починатися слово в тексті: K
Process finished with exit code 0
```



Висновки:

Я вивчила основні способи роботи з виключеннями. Виключення користувача. Відкриття файлів, зчитування та запис у файл. Шляхи доступу до файлів. Функції, методи та атрибути для роботи з файлами.