

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
Факультет прикладної математики  
Кафедра прикладної математики

Звіт  
із лабораторної роботи №7  
із дисципліни «Основи програмування»  
на тему  
РОБОТА З ФАЙЛАМИ

Виконав:  
студент групи КМ-93  
*Пиндиківський Т. Р.*

Керівник:  
*асистент Дрозденко О. М.*

Київ — 2019

## ЗМІСТ

МЕТА РОБОТИ .....	3
ПОСТАНОВКА ЗАДАЧІ .....	4
ОСНОВНА ЧАСТИНА .....	5
ВИСНОВКИ .....	6
ДОДАТОК А .....	7
ДОДАТОК Б .....	10

## **МЕТА РОБОТИ**

Вивчити методи роботи з файлами та здобути практичні навички із створення і обробки текстових файлів.

## ПОСТАНОВКА ЗАДАЧІ

### *Завдання 1:*

Є текстовий файл.

Надрукувати:

- ☐ його перший рядок;
- ☐ його п'ятий рядок;
- ☐ його перші 5 рядків;
- ☐ його рядки з s1-го по s2-ий;
- ☐ весь файл.

## ОСНОВНА ЧАСТИНА

*Завдання 1* : Для виконання програми спочатку імпортуються модулі `re`, `sys`, які використовуються для обробки регулярних виразів та системних команд.

Першою чергою відбувається відкриття файлу у режимі читання, дані з якого будуть опрацьовані. У разі випадку коли такий файл не знайдено, буде виведено повідомлення про відсутність файлу та відбудеться завершення роботи програми. Такий функціонал створений конструкцією `try-except` та командою `sys.exit()`. Після цього відбувається підрахування загальної кількості рядків файлу. Згодом, за допомогою команди `file_lines = [line.strip() for line in file]` з використанням генератора списку відбувається створення ітерованого об'єкта, заповненого рядками файлу, як його елементами. Це дозволяє виконати перше, друге та третє підзавдання простим виведенням відповідних елементів списку. Виконання четвертої частини передбачає введення користувачем номерів рядків початку і кінця тексту, який має бути виведений. У такій ситуації використовуються валідатори для отриманих даних, які мають відповідати чисельному типу та не можуть бути меншими за нуль, або більшими за кількість рядків цілого файлу. Після отримання правильних даних, що стало можливим після застосування валідаторів, з діапазону від початку до кінця виводяться потрібні рядки файлу. На кінець відповідно до завдання, виводиться весь вміст файлу.

## **ВИСНОВКИ**

На цій лабораторній роботі було вивчено методи роботи з файлами та здобуті практичні навички із створення і обробки текстових файлів. Були розроблені програми, які реалізують поставлену задачу відповідно до вимог. Їх можна застосовувати для вирішення однотипних завдань. Є багато можливостей для розширення функціоналу з метою забезпечення виконання складніших завдань.

## ДОДАТОК А

### 1. Програмна реалізація задачі №1

```
#D:\Taras
Pyndyktivskiy\kpi\programming\programming_1\laboratory_works\#7
"""
Є текстовий файл.
Надрукувати:
└ його перший рядок;
└ його п'ятий рядок;
└ його перші 5 рядків;
└ його рядки з s1-го по s2-ий;
└ весь файл.
"""

import sys
import re

pattern=re.compile(r"^[+]?[d+${}")

def validator(pattern, prompt):
    data=input(prompt)
    while not bool(pattern.match(data)):
        data=input(prompt)
    return data

def start_line_validator(pattern, prompt, number_lines):

    number=int(validator(pattern, prompt))
    while number<0 and number>number_lines-1:
        number=int(validator(pattern, prompt))

    return number

def finish_line_validator(pattern, prompt, start_line_to_print,
number_lines):
    number=int(validator(pattern, prompt))
    while number<start_line_to_print and number>number_lines-1:
        number=int(validator(pattern, prompt))

    return number

try:
    file=open("task1.py", 'r', encoding = "utf-8")
except FileNotFoundError:
    print("Вказаного файлу не існує")
    sys.exit()

number_lines=0

while True:
    s = file.readline()
    # print(s)
    number_lines+=1
    if not s:
```

```

        break

file.close()

number_lines-=1
#print(number_lines)

file=open("task1.py", 'r', encoding = "utf-8")
if number_lines<5:
    print("\nУ Вашому файлі менше 5 рядків")
    sys.exit()

file_lines = [line.strip() for line in file]
#file_lines=[line for line in file]
#print(file_lines)

print("\nПерший рядок файлу : ")
print(file_lines[0])

print("\nП'ятий рядок файлу : ")
print(file_lines[4])

number_lines=0
print("\nПерші 5 рядків файлу")
for i in range(5):
    print(file_lines[i])

print('\n')
start_line_to_print=start_line_validator(pattern, "Введіть початковий номер рядка файлу : ", number_lines)
finish_line_to_print=finish_line_validator(pattern, "Введіть кінцевий номер рядка файлу : ", start_line_to_print, number_lines)

print("\nВміст файлу з " + str(start_line_to_print) + " до " + str(finish_line_to_print) + " рядку.")
for i in range(start_line_to_print, finish_line_to_print+1):
    print(file_lines[i])

file.close()

file=open("task1.py", 'r', encoding = "utf-8")
file_lines = [line.strip() for line in file]

print("\nВесь вміст файлу: ")
for i in range(len(file_lines)):
    print(file_lines[i])

file.close()

```



```

D:\Taras Pyndykivskiy\kpi\programming\programming_1\laboratory_works\#7
λ python task1.py

Натисніть Enter, щоб продовжити, будь-що інше, щоб завершити роботу з програм

Перший рядок файлу :
#D:\Taras Pyndykivskiy\kpi\programming\programming_1\laboratory_works\#7

П'ятий рядок файлу :
▣ його перший рядок;

Перші 5 рядків файлу
#D:\Taras Pyndykivskiy\kpi\programming\programming_1\laboratory_works\#7
"""
Є текстовий файл.
Надрукувати:
▣ його перший рядок;

Введіть початковий номер рядка файлу : 2
Введіть кінцевий номер рядка файлу : 10

Вміст файлу з 2 до 10 рядку.
Є текстовий файл.
Надрукувати:
▣ його перший рядок;
▣ його п'ятий рядок;
▣ його перші 5 рядків;
▣ його рядки з s1-го по s2-ий;
▣ весь файл.
"""
import sys

Весь вміст файлу:
#D:\Taras Pyndykivskiy\kpi\programming\programming_1\laboratory_works\#7
"""
Є текстовий файл.

```

Рис. 1 - програмна реалізація задачі №1

## ДОДАТОК Б

### *Питання для самоперевірки*

*1. Що таке «файлова змінна»?*

Змінна, за допомогою якої програма буде здійснювати зв'язок з файлом.

*2. Які основні етапи роботи з файлами в програмі?*

Відкриття, читання, запис, завершення роботи з файлом.

*3. Які можливі режими відкриття файлу?*

Режими відкриття файлу можуть дозволяти робити зчитування даних, запис, або одне і те саме.

*4. Як можна почати читання даних з файлу з самого початку?*

Для читання рядка текстового файлу використовується метод `readline()`.

*5. Що таке «послідовний доступ до даних»?*

Текстовий файл називають «файлом послідовного доступу до даних». Це означає, що для того, щоб прочитати 100-е за рахунком значення з файлу, потрібно спочатку прочитати попередні 99.

*6. Як можна прочитати всі рядки файлу, якщо їх кількість невідома?*

У подібних випадках для читання всіх рядків можна використовувати інструкцію циклу `for` зі змінною `line`.

*7. Як можна рядки файлу записати в список?*

`a = [line.strip() for line in f]`

*8. Як визначити, що дані в файлі закінчилися?*

Після того як викликаний метод `read()` на файловому об'єкті, якщо повторно викликати `read()`, то буде порожній рядок. Це відбувається тому, що після першого прочитання вказівник знаходиться в кінці файлу.

*9. Як прочитати построково файл?*

`a.readline()`

*10. Функції модуля OS для роботи з файлом.*

Ряд можливостей по роботі з каталогами та файлами надає вбудований модуль:

- ☐ `mkdir()`: створює нову папку
- ☐ `rmdir()`: видаляє папку
- ☐ `rename()`: перейменовує файл
- ☐ `remove()`: видаляє файл

*11. Функції модуля OS для роботи з каталогом.*

Ряд можливостей по роботі з каталогами та файлами надає вбудований модуль:

- ☐ `mkdir ()`: створює нову папку
- ☐ `rmdir ()`: видаляє папку
- ☐ `rename ()`: перейменовує файл
- ☐ `remove ()`: видаляє файл