

Міністерство освіти і науки України
Державний університет „Житомирська політехніка”

Кафедра ІПЗ
Група: ІПЗ-23-1

Програмування мовою Python
Лабораторна робота №6
«РОБОТА З ФАЙЛАМИ»

Виконав:

Риженко Я.В.

Прийняв:

Желізко В. В.

					«Житомирська політехніка».24.121.05.000						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Риженко Я.В.			Звіт з лабораторної роботи			Літ.	Арк.	Аркушів	
Перевір.		Желізко В. В.								1	4
Керівник								ФІКТ, гр. ІПЗ-23-1			
Н. контр.											
Затверд.											

Мета роботи: ознайомитися з засобами роботи з файлами в мові Python, діями над ними.

Хід роботи

Завдання 1. Створіть новий файл numbers.txt у текстовому редакторі і запишіть у нього 10 чисел, кожне з нового рядка. Напишіть програму, яка зчитує ці числа з файла і обчислює їх суму, виводить цю суму на екран і, водночас, записує цю суму у інший файл з назвою sum_numbers.txt.

Листинг програми:

```
with open("numbers.txt", "w") as file:
    for number in [5, 356, 12, -120, -25, -37, 35, 40, 45, 50]:
        file.write(f"{number}\n")

with open("numbers.txt", "r") as file:
    numbers = [int(line.strip()) for line in file]
    total_sum = sum(numbers)

print(f"Сума чисел: {total_sum}")

with open("sum_numbers.txt", "w") as file:
    file.write(f"{total_sum}")
```

Результат виконання:

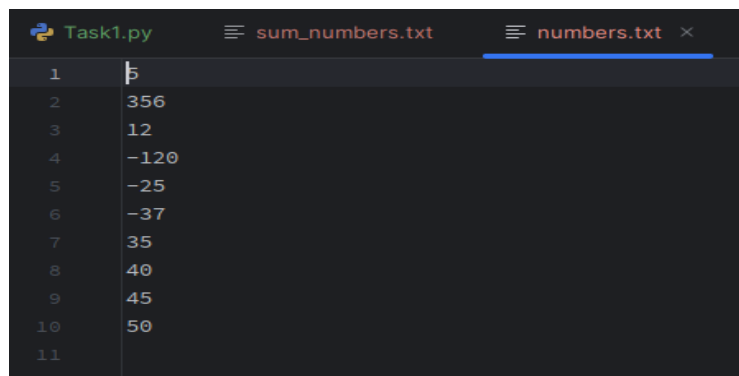


Рис. 1. Числа.

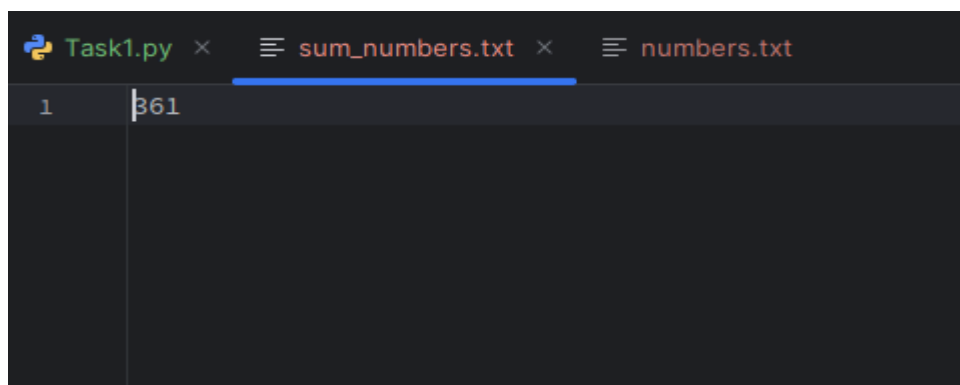


Рис. 2. Текстовий документ, створений після виконання.

					«Житомирська політехніка».24.121.05.000	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.

Реалізуйте програму, яка зчитує довільну кількість цілих чисел, що вводяться з командного рядка, і записує у текстовий файл інформацію, щодо парності або непарності чисел.

Листинг програми:

```
nums = []
print("Введіть довільну кількість цілих чисел (або введіть 'end' для завершення):")

while True:
    user_input = input()
    if user_input.lower() == "end":
        break
    try:
        num = int(user_input)
        nums.append(num)
    except ValueError:
        print("Будь ласка, введіть ціле число або 'end'.")

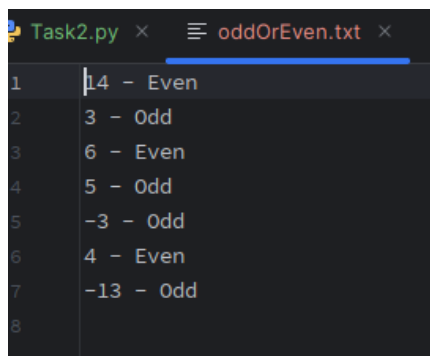
print("Введені числа:", nums)

with open("oddOrEven.txt", "w") as file:
    for num in nums:
        if num % 2 == 0: file.write(str(num)+" - Even" + "\n")
        else: file.write(str(num)+" - Odd" + "\n")
```

Результат виконання:

```
Введіть довільну кількість цілих чисел (або введіть 'end' для завершення):
14
3
6
5
-3
4
-13
end
Введені числа: [14, 3, 6, 5, -3, 4, -13]
```

Рис. 3. Результат.



```
Task2.py x oddOrEven.txt x
1 14 - Even
2 3 - Odd
3 6 - Even
4 5 - Odd
5 -3 - Odd
6 4 - Even
7 -13 - Odd
8
```

Рис. 4. Текстовий документ, створений після виконання.

					«Житомирська політехніка».24.121.05.000	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 3.

Створіть новий файл у текстовому редакторі і напишіть кілька рядків тексту у ньому про можливості Python. Кожен рядок повинен починатися з фрази: «Python можна використати для ...». Збережіть файл з ім'ям learning_python.txt. Напишіть програму, яка зчитує файл і виводить текст з перебором рядків файла і зі збереженням рядків у списку з подальшим сортуванням списку за довжиною рядків в ньому від найбільшого до найменшого.

Листинг програми:

```
with open("learning_python.txt", "w", encoding="utf-8") as file:
    file.write("Python можна використати для аналізу даних.\n")
    file.write("Python можна використати для створення веб-додатків.\n")
    file.write("Python можна використати для машинного навчання.\n")
    file.write("Python можна використати для автоматизації задач.\n")

filename = "learning_python.txt"

try:
    with open(filename, "r", encoding="utf-8") as file:
        lines = [line.strip() for line in file.readlines()]

        sorted_lines = sorted(lines, key=len, reverse=True)

        print("Вміст файлу, відсортований за довжиною рядків:")
        for line in sorted_lines:
            print(line)

except FileNotFoundError:
    print(f"Файл '{filename}' не знайдено. Будь ласка, створіть його спочатку.")

print(f"Файл '{filename}' створено та заповнено.")
```

Результат виконання:

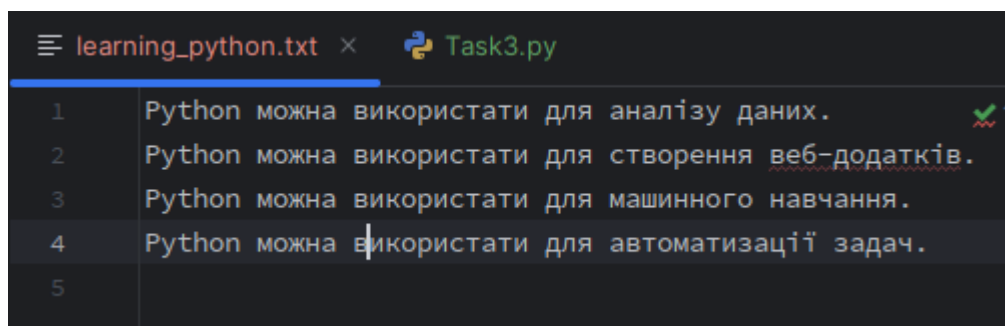


Рис. 5. Текстовий документ, створений після виконання.

					«Житомирська політехніка».24.121.05.000	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 4.

Прочитайте кожен рядок зі створеного у попередньому завданні файла `learning_python.txt` і замініть слово `Python` назвою іншої мови, наприклад `C` при виведенні на екран. Отриманий файл має бути створений в новому каталозі, що розміщується в поточному. Відкрийте файл пострічково і дайте можливість користувачеві визначити які змінені фрази є актуальними, наприклад для мови `C`, а які ні. Всі хибні твердження запишіть в інший файл, а істинні – в поточний.

Листинг програми:

```
import os

original_file = "learning_python.txt"

new_directory = "new_directory"
os.makedirs(new_directory, exist_ok=True)

true_file = os.path.join(new_directory, "true_statements.txt")
false_file = os.path.join(new_directory, "false_statements.txt")

modified_file = os.path.join(new_directory, "learning_c.txt")

with open(original_file, "r", encoding="utf-8") as infile, \
    open(modified_file, "w", encoding="utf-8") as outfile:
    for line in infile:
        modified_line = line.replace("Python", "C")
        outfile.write(modified_line)

print(f"Файл зі зміненими рядками створено: {modified_file}")

with open(modified_file, "r", encoding="utf-8") as infile, \
    open(true_file, "w", encoding="utf-8") as true_outfile, \
    open(false_file, "w", encoding="utf-8") as false_outfile:

    for line in infile:
        print(f"Чи актуальне це твердження? {line.strip()}")
        user_response = input("Введіть 'y' для так або 'n' для ні: ")
        user_response = user_response.strip().lower()

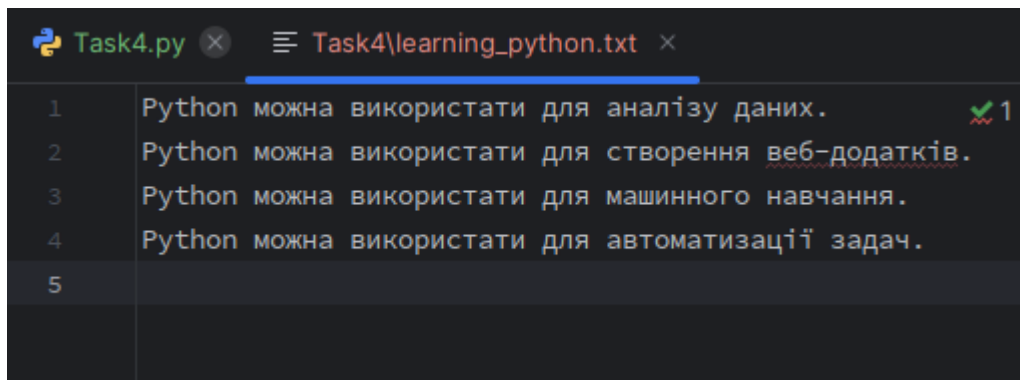
        if user_response == "y":
            true_outfile.write(line)
        elif user_response == "n":
            false_outfile.write(line)
        else:
            print("Некоректна відповідь. Пропускаємо це твердження.")
```

					«Житомирська політехніка».24.121.05.000	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання:

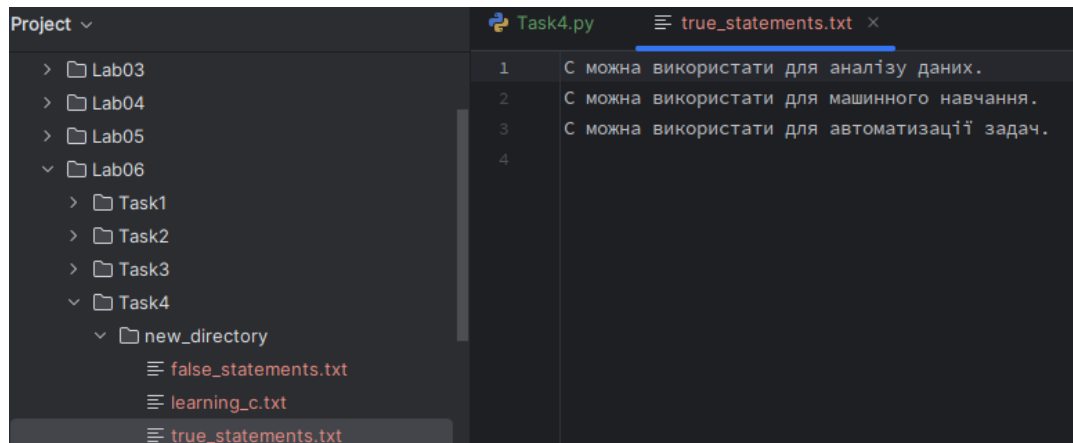
```
Файл зі зміненими рядками створено: new_directory\learning_c.txt
Чи актуальне це твердження? С можна використати для аналізу даних.
Введіть 'у' для так або 'n' для ні: у
Чи актуальне це твердження? С можна використати для створення веб-додатків.
Введіть 'у' для так або 'n' для ні: n
Чи актуальне це твердження? С можна використати для машинного навчання.
Введіть 'у' для так або 'n' для ні: у
Чи актуальне це твердження? С можна використати для автоматизації задач.
Введіть 'у' для так або 'n' для ні: у
```

Рис. 6. Результат.



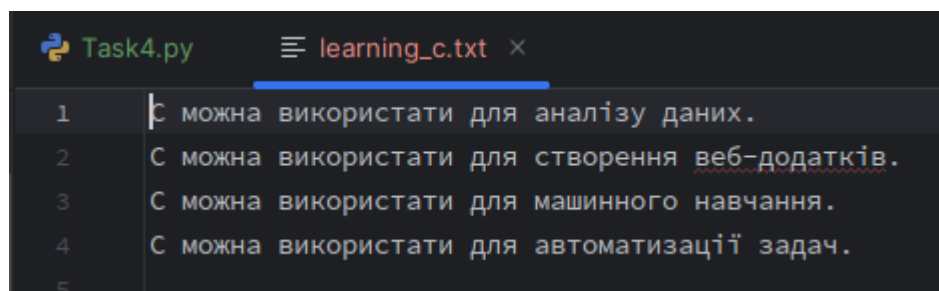
```
Task4.py x Task4\learning_python.txt x
1 Python можна використати для аналізу даних. ✓ 1
2 Python можна використати для створення веб-додатків.
3 Python можна використати для машинного навчання.
4 Python можна використати для автоматизації задач.
5
```

Рис. 7. Текстовий документ, створений після виконання.



```
Project v Task4.py true_statements.txt x
> Lab03
> Lab04
> Lab05
v Lab06
  > Task1
  > Task2
  > Task3
  v Task4
    v new_directory
      false_statements.txt
      learning_c.txt
      true_statements.txt
1 С можна використати для аналізу даних.
2 С можна використати для машинного навчання.
3 С можна використати для автоматизації задач.
4
```

Рис. 8. Текстовий документ, створений після виконання.



```
Task4.py x learning_c.txt x
1 С можна використати для аналізу даних. ✓ 1
2 С можна використати для створення веб-додатків.
3 С можна використати для машинного навчання.
4 С можна використати для автоматизації задач.
5
```

Рис. 9. Текстовий документ, створений після виконання.

					«Житомирська політехніка».24.121.05.000	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

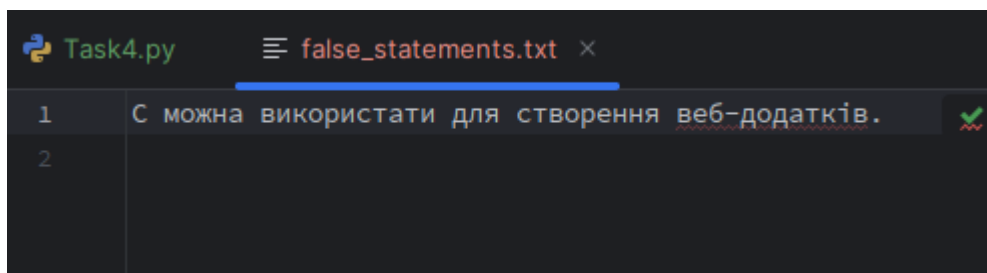


Рис. 10. Текстовий документ, створений після виконання.

Завдання 5.

Створіть порожній файл `guest_book.txt` у текстовому редакторі.

Напишіть програму, яка запитує у користувачів імена. При введенні кожного імені виведіть на екран рядок з вітанням для користувача і запишіть рядок вітання у файл з ім'ям `guest_book.txt`. Простежте за тим, щоб кожне повідомлення розміщувалося в окремому рядку файлу з зазначенням часу внесення цього повідомлення. Передбачте зазначення в файлі часу його створення і вказання в ньому часу останніх внесених змін.

Листинг програми:

```
import time
import os

def add_guest_to_book(guest_name, file_name="guest_book.txt"):
    current_time = time.strftime("%Y-%m-%d %H:%M:%S")
    greeting = f"Привіт, {guest_name}! (Додано: {current_time})"
    print(greeting)
    with open(file_name, "a", encoding="utf-8") as file:
        file.write(f"{greeting}\n")
    update_last_modified_time(file_name)

def update_last_modified_time(file_name):
    if os.path.exists(file_name):
        with open(file_name, "r", encoding="utf-8") as file:
            lines = file.readlines()
            last_modified = f"Останні зміни: {time.strftime('%Y-%m-%d %H:%M:%S')}\n"
            if len(lines) > 0 and "Останні зміни:" in lines[-1]:
                lines[-1] = last_modified
            else:
                lines.append(last_modified)
                lines.append("-" * 50 + "\n")
        with open(file_name, "w", encoding="utf-8") as file:
            file.writelines(lines)

print("Введіть імена гостей. Для завершення введіть 'вихід'.")
while True:
```

```

guest_name = input("Введіть ім'я: ").strip()
if guest_name.lower() == 'вихід':
    print("Роботу завершено.")
    break
elif guest_name:
    add_guest_to_book(guest_name)

```

Результат виконання:

```

Введіть імена гостей. Для завершення введіть 'вихід'.
Введіть ім'я: Ян
Привіт, Ян! (Додано: 2024-11-29 21:33:00)
Введіть ім'я: Олександр
Привіт, Олександр! (Додано: 2024-11-29 21:33:07)
Введіть ім'я: Віктор
Привіт, Віктор! (Додано: 2024-11-29 21:33:10)
Введіть ім'я: вихід
Роботу завершено.

```

Рис. 11. Результат.

```

Task5.py x  guest_book.txt x
1  Привіт, Ян! (Додано: 2024-11-24 18:02:31)
2  Останні зміни: 2024-11-24 18:02:31
3  -----
4  Привіт, Володимир! (Додано: 2024-11-24 18:02:34)
5  Останні зміни: 2024-11-24 18:02:34
6  -----
7  Привіт, Зеленський! (Додано: 2024-11-24 18:02:38)
8  Останні зміни: 2024-11-24 18:02:38
9  -----
10 Привіт, Чед! (Додано: 2024-11-24 18:02:43)
11 Останні зміни: 2024-11-24 18:02:43
12 -----
13 Привіт, Ян! (Додано: 2024-11-29 21:33:00)
14 Останні зміни: 2024-11-29 21:33:00
15 -----
16 Привіт, Олександр! (Додано: 2024-11-29 21:33:07)
17 Останні зміни: 2024-11-29 21:33:07
18 -----
19 Привіт, Віктор! (Додано: 2024-11-29 21:33:10)
20 Останні зміни: 2024-11-29 21:33:10
21 -----
22

```

Рис. 12. Текстовий документ, створений після виконання.

					«Житомирська політехніка».24.121.05.000	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

Завдання 6.

Збережіть в текстовому файлі публікацію про Python на 3000 слів англійською мовою. Напишіть програму, що аналізуватиме частоту з якою в тексті зустрічаються окремі літери чи слова незалежно від їх регістру. Результат роботи програми має виводитись в консоль і зберігатись в окремому файлі з зазначенням часу його створення, часу виконання окремих змін, результатів пошуку і часу, що знадобся на виконання цього пошуку.

Листинг програми:

```
import os
import time
from collections import Counter

input_file = "python.txt" #Публікація на 3000 слів англійською мовою
output_file = f"analysis_results_{time.strftime('%Y%m%d_%H%M%S')}.txt"

def analyze_text(file_path):
    start_time = time.time()

    with open(file_path, "r", encoding="utf-8") as file:
        text = file.read()

    letters = Counter(c.lower() for c in text if c.isalpha())

    words = Counter(word.lower() for word in text.split())

    end_time = time.time()
    elapsed_time = end_time - start_time

    results = [
        f"Analysis performed on: {time.strftime('%Y-%m-%d %H:%M:%S')}",
        f"Time taken for analysis: {elapsed_time:.2f} seconds",
        "\nLetter Frequency:",
        *[f"{letter}: {count}" for letter, count in letters.most_common()],
        "\nWord Frequency:",
        *[f"{word}: {count}" for word, count in words.most_common(20)] # Top 20
    ]

    with open(output_file, "w", encoding="utf-8") as out_file:
        out_file.write("\n".join(results))

    return results, output_file

analysis_results, result_file_path = analyze_text(input_file)

print("\n".join(analysis_results))
```

					«Житомирська політехніка».24.121.05.000	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

```
print(f"\nResults saved to: {result_file_path}")
```

Результат виконання:

```
Analysis performed on: 2024-11-29 21:35:34
Time taken for analysis: 0.01 seconds

Letter Frequency:
e: 687
i: 595
a: 579
n: 553
t: 534
o: 517
s: 477
r: 380
l: 368
p: 247
d: 244
c: 237
h: 213
m: 205
g: 201
y: 174
u: 171
f: 158
b: 101
w: 84
v: 76
k: 62
x: 19
j: 12
z: 5
q: 3
```

```
Word Frequency:
and: 64
python: 35
of: 28
to: 28
the: 25
for: 23
is: 20
a: 20
its: 16
like: 15
in: 14
it: 11
data: 11
libraries: 10
python's: 9
on: 9
language: 8
an: 8
developers: 8
programming: 7

Results saved to: analysis_results_20241129_213534.txt
```

Рис. 13. Результат.

					«Житомирська політехніка».24.121.05.000	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

```
Task6.py x analysis_results_20241129_213534.txt x
1 Analysis performed on: 2024-11-29 21:35:34
2 Time taken for analysis: 0.01 seconds
3
4 Letter Frequency:
5 e: 687
6 i: 595
7 a: 579
8 n: 553
9 t: 534
10 o: 517
11 s: 477
12 r: 380
13 l: 368
14 p: 247
15 d: 244
16 c: 237
17 h: 213
18 m: 205
19 g: 201
20 y: 174
21 u: 171
22 f: 158
23 b: 101
24 w: 84
25 v: 76
26 k: 62
27 x: 19
28 j: 12
29 z: 5
30 q: 3
31
32 Word Frequency:
33 and: 64
34 python: 35
35 of: 28
36 to: 28
37 the: 25
38 for: 23
39 is: 20
40 a: 20
41 its: 16
42 like: 15
```

Рис. 14. Текстовий документ, створений після виконання.

Завдання 7.

Завантажте файл marks.csv і визначте кількість студентів, що проходили тестування. Виведіть інформацію про те яку оцінку набрали відповідна кількість студентів. Виведіть інформацію яку середню оцінку отримували студенти за певний час виконання КМР (крок – 1 хв). Створіть текстовий файл і запишіть в нього статистику по правильним відповідям для кожного окремого питання (який відсоток правильних і неправильних відповідей на питання дали студенти). В цей же файл внесіть інформацію про 5 найкращих оцінок в співвідношенні оцінка/час витрачений складання КМР.

Листинг програми:

```
import csv

def parse_duration(duration_str):
    if "хв" in duration_str and "сек" in duration_str:
        parts = duration_str.split(" ")
        minutes = int(parts[0].replace("хв", ""))
        seconds = int(parts[2].replace("сек", ""))
        return minutes + seconds / 60
    return 0

def safe_float_conversion(value):
    try:
        return float(value.replace(",", "."))
    except ValueError:
        return None

file_name = "marks.csv"
students_data = []
with open(file_name, "r", encoding="utf-8") as file:
    reader = csv.reader(file)
    for row in reader:
        students_data.append(row)

student_ids = {row[0] for row in students_data}
students_count = len(student_ids)

grades_count = {}
for row in students_data:
    grade = safe_float_conversion(row[4])
    if grade is not None:
        grades_count[grade] = grades_count.get(grade, 0) + 1

time_to_grades = {}
for row in students_data:
    duration = parse_duration(row[3])
    grade = safe_float_conversion(row[4])
    if grade is not None:
```

					«Житомирська політехніка».24.121.05.000	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        rounded_time = round(duration)
        if rounded_time not in time_to_grades:
            time_to_grades[rounded_time] = []
        time_to_grades[rounded_time].append(grade)

average_grades_per_time = {time: sum(grades) / len(grades) for time, grades in
time_to_grades.items()}

questions_stats = {}
questions = students_data[0][5:]
for i, question in enumerate(questions):
    correct = sum(1 for row in students_data if safe_float_conversion(row[5 + i])
and safe_float_conversion(row[5 + i]) > 0)
    total = len(students_data)
    questions_stats[f"Question {i + 1}"] = {
        "correct_percent": (correct / total) * 100,
        "incorrect_percent": 100 - (correct / total) * 100,
    }

ratios = []
for row in students_data:
    grade = safe_float_conversion(row[4])
    duration = parse_duration(row[3])
    if grade is not None and duration > 0:
        ratio = grade / duration
        ratios.append((ratio, row[0]))

top_5_ratios = sorted(ratios, reverse=True)[:5]

output_file = "statistics.txt"
with open(output_file, "w", encoding="utf-8") as file:
    file.write(f"Кількість студентів: {students_count}\n")
    file.write("Статистика за оцінками:\n")
    for grade, count in grades_count.items():
        file.write(f"Оцінка {grade}: {count} студентів\n")
    file.write("Середня оцінка за час виконання:\n")
    for time, avg_grade in sorted(average_grades_per_time.items()):
        file.write(f"{time} хв: {avg_grade:.2f}\n")
    file.write("Аналіз правильності відповідей:\n")
    for question, stats in questions_stats.items():
        file.write(f"{question}: Правильно: {stats['correct_percent']:.2f}%,
Неправильно: {stats['incorrect_percent']:.2f}%\n")
    file.write("Топ-5 співвідношень оцінка/час:\n")
    for ratio, student_id in top_5_ratios:
        file.write(f"ID {student_id}: {ratio:.2f}\n")

```

					«Житомирська політехніка».24.121.05.000	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання:

```
Task7.py  statistics.txt ×
1  Кількість студентів: 170
2  Статистика за оцінками:
3  Оцінка 8.0: 18 студентів
4  Оцінка 7.0: 7 студентів
5  Оцінка 4.0: 3 студентів
6  Оцінка 9.5: 31 студентів
7  Оцінка 10.0: 44 студентів
8  Оцінка 5.5: 4 студентів
9  Оцінка 7.5: 10 студентів
10 Оцінка 6.0: 5 студентів
11 Оцінка 8.5: 16 студентів
12 Оцінка 9.0: 25 студентів
13 Оцінка 0.5: 1 студентів
14 Оцінка 5.0: 1 студентів
15 Оцінка 2.0: 1 студентів
16 Оцінка 4.5: 1 студентів
17 Оцінка 3.0: 1 студентів
18 Оцінка 3.5: 1 студентів
19 Оцінка 2.5: 1 студентів
20 Середня оцінка за час виконання:
21 0 хв: 8.18
22 5 хв: 9.00
23 6 хв: 9.12
24 7 хв: 8.60
25 8 хв: 9.25
26 9 хв: 9.12
27 10 хв: 8.21
28 11 хв: 8.41
29 12 хв: 7.80
30 13 хв: 8.75
31 14 хв: 9.00
32 15 хв: 9.04
33 16 хв: 9.11
34 17 хв: 8.69
35 18 хв: 8.00
36 19 хв: 9.10
37 20 хв: 7.57
```

```

38  Аналіз правильності відповідей:
39  Question 1: Правильно: 85.88%, Неправильно: 14.12%
40  Question 2: Правильно: 97.06%, Неправильно: 2.94%
41  Question 3: Правильно: 92.35%, Неправильно: 7.65%
42  Question 4: Правильно: 93.53%, Неправильно: 6.47%
43  Question 5: Правильно: 89.41%, Неправильно: 10.59%
44  Question 6: Правильно: 86.47%, Неправильно: 13.53%
45  Question 7: Правильно: 59.41%, Неправильно: 40.59%
46  Question 8: Правильно: 80.59%, Неправильно: 19.41%
47  Question 9: Правильно: 87.65%, Неправильно: 12.35%
48  Question 10: Правильно: 96.47%, Неправильно: 3.53%
49  Question 11: Правильно: 92.94%, Неправильно: 7.06%
50  Question 12: Правильно: 90.59%, Неправильно: 9.41%
51  Question 13: Правильно: 96.47%, Неправильно: 3.53%
52  Question 14: Правильно: 97.06%, Неправильно: 2.94%
53  Question 15: Правильно: 82.94%, Неправильно: 17.06%
54  Question 16: Правильно: 92.35%, Неправильно: 7.65%
55  Question 17: Правильно: 72.94%, Неправильно: 27.06%
56  Question 18: Правильно: 67.06%, Неправильно: 32.94%
57  Question 19: Правильно: 68.24%, Неправильно: 31.76%
58  Question 20: Правильно: 74.71%, Неправильно: 25.29%
59  Топ-5 співвідношень оцінка/час:
60  ID cle4e81f089568f9b39e1d9814224daf: 1.99
61  ID f6662d314e7d42c4cbdf4b1c8c8f4f0a: 1.68
62  ID fa266e9b0ab02aada4c3943cb37a3189: 1.54
63  ID 0abbca5c0d2cc3c8a869a8e02521ad3f: 1.51
64  ID 5dd0cc082f1fae8b18cdb603fb2f2e3d: 1.42
65

```

Рис. 15. Текстовий документ, створений після виконання.

Висновок: Я навчився працювати з файлами в Python: відкривати їх у різних режимах, читати та записувати дані, використовувати контекстний менеджер with для автоматичного закриття файлів, а також обробляти текстові файли. Це дозволяє ефективно керувати файлами та забезпечувати коректну роботу програм із зовнішніми даними.

Посилання на Git: : https://github.com/JanRizhenko/Python_Labs

					«Житомирська політехніка».24.121.05.000	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		