МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Український державний університет імені Михайла Драгоманова

Факультет математики, інформатики та фізики

Кафедра інформаційних технологій та програмування

**Звіт**

з лабораторної роботи №5

«»

з дисципліни «Програмування»

Виконав:

студент ІІІ курсу групи 31І

Бритвич О.В.

Перевірила:

викладач Устименко О.Б.

Оцінка

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2023

Зміст

Мета роботи 3

1 Постановка задачі 4

2 Основна частина 5

2.1 Опис вхідних та вихідних даних 7

2.1 Блок-схема 8

2.2 Опис вхідних та вихідних даних 12

2.2 Блок-схема 13

Висновки 15

Список літератури 16

Додатки 17

**Мета роботи**

Розробити програму, яка забезпечує правильність введених дат і об’єднує мову програмування та її рейтингові дані у два текстові файли, відображає результати на екрані та зберігає їх у третьому файлі у заданому форматі.

**1 Постановка задачі**

1. Задача №1.

Написати функцію для перевірки правильності введеної дати. Функція приймає 3 аргументи – день, місяць та рік і повертає *True*, якщо така дата є в календарі, і *False* в протилежному випадку.

1. Задача №2.

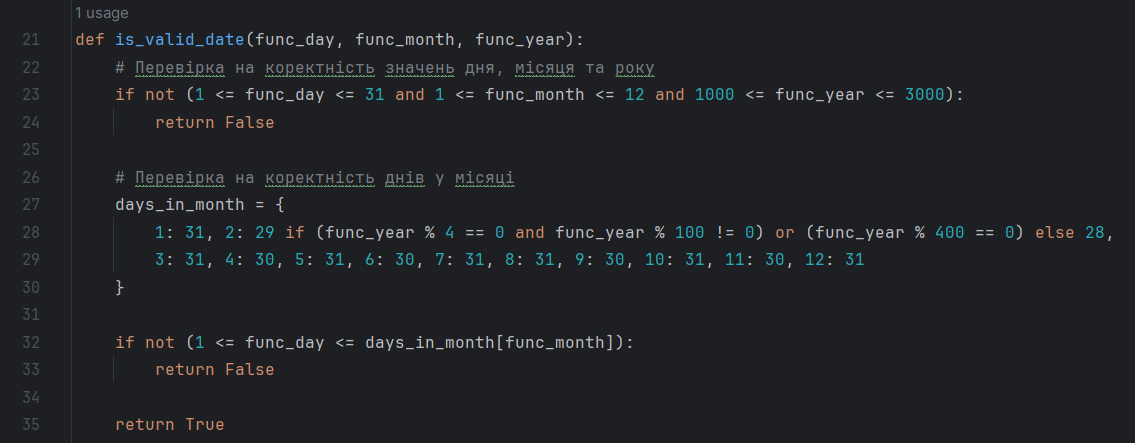
Дано два текстові файли: в першому записані в рядках назви мов програмування, а у іншому на відповідних рядках – рейтинги цих мов у відсотках. Напишіть програму для об’єднання кожного рядка з першого файлу з відповідним рядком у другому файлі. Об’єднані рядки необхідно вивести на екран і, водночас, в такому вигляді як у вихідних даних, записати у третій текстовий файл.

**2 Основна частина**

Задача №1

Код починається з ініціалізації функції *is\_valid\_date*, яка перевіряє, чи є введена дата коректною. Вона приймає значення дня, місяця і року. Ця функція виконує наступні перевірки:

* Перевіряє, чи день знаходиться в межах від 1 до 31, місяць від 1 до 12 і рік від 1000 до 3000.
* Перевіряє, чи введений день не перевищує максимальну кількість днів у відповідному місяці за допомогою словника *days\_in\_month*. У випадку лютого враховується високосний рік.

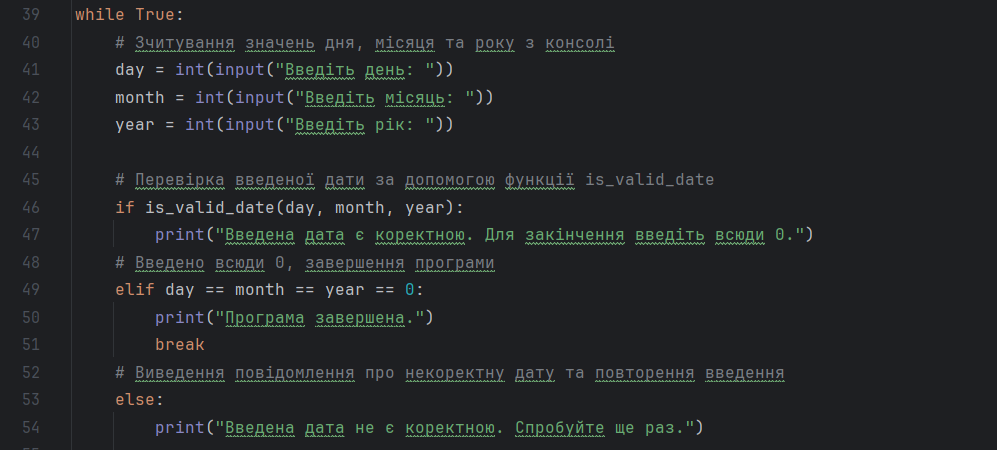


Далі йде цикл *while* – безкінечний цикл, який продовжується до тих пір, поки користувач не введе всюди *0* для дня, місяця та року, позначаючи бажання завершити програму.

Користувачу пропонується ввести значення для дня, місяця та року через виклик *input*. Функція *is\_valid\_date* перевіряє ці значення. Якщо вони відповідають правилам (є коректною датою), програма повідомляє про це.

Якщо користувач вводить всюди *0*, програма завершується.

Якщо введена дата не є коректною, програма повідомляє про це і пропонує спробувати знову:



2.1 Опис вхідних та вихідних даних

Вхідні дані:

*day* – змінна для збереження цілого значення днів;

*month* – змінна для збереження цілого значення місяців;

*year* – змінна для збереження цілого значення років.

Вихідні дані:

Оброблені значення днів, місяців і років згідно функції is\_valid\_date.

2.1 Блок-схема

Початок

def is\_valid\_date(func\_day, func\_month, func\_year)

while True

if not (1 <= func\_day <= 31 and 1 <= func\_month <= 12 and 1000 <= func\_year <= 3000)

if not (1 <= func\_day <= days\_in\_month[func\_month])

return True

days\_in\_month = {}

return False

return False

Увести day

Увести month

Увести year

if is\_valid\_date(day, month, year)

Вивести текст

elif day == month == year == 0

Вивести текст про завершення програми

break

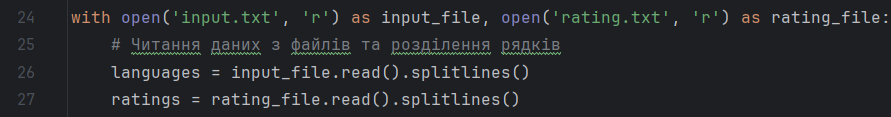
else

Вивести текст про некоректність введених даних

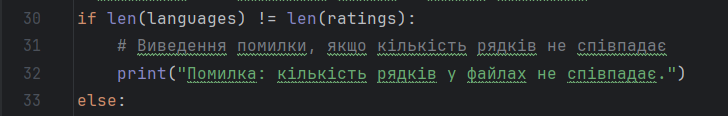
Кінець

Задача №2

Спочатку програма відкриває два файли: *input.txt* та *rating.txt* для читання. Зчитує дані з цих файлів: текстові рядки, які містять мови і оцінки:

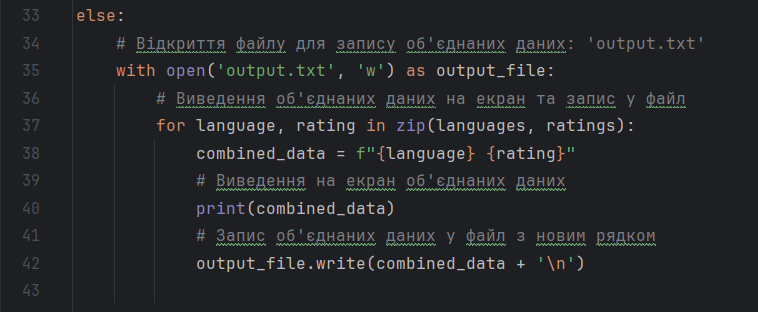


Перевіряє, чи кількість рядків у цих файлах співпадає. Якщо ні, виводить повідомлення про помилку і припиняє виконання програми:



Якщо кількість рядків співпадає, відкриває файл *output.txt* для запису.

Для кожного рядка з файлів *input.txt* і *rating.txt* об’єднує дані у вигляді «мова – оцінка» і виводить ці дані на екран. Записує об’єднані дані у файл output.txt, кожен запис з новим рядком:



2.2 Опис вхідних та вихідних даних

Вхідні дані:

*input.txt* – змінна для зберігання файлу зі списком мов програмування;

*rating.txt* – змінна для збереження файлу зі списком рейтингів мов програмування.

Вихідні дані:

*output.txt* – змінна для збереження файлу, в якому об’єднуються дані з *input.txt* та *rating.txt*.

2.2 Блок-схема

Початок

with open('input.txt', 'r') as input\_file, open('rating.txt', 'r') as rating\_file

Вивести текст про помилку

ratings = rating\_file.read().splitlines()

languages = input\_file.read().splitlines()

for language, rating in zip(languages, ratings)

combined\_data = f"{language} {rating}"

with open('output.txt', 'w') as output\_file

if len(languages) != len(ratings)

else

Вивести об’єднані дані

output\_file.write(combined\_data + '\n')

Кінець

**Висновки**

У даній лабораторній роботі були вирішені дві задачі. Перша – написання функції для перевірки правильності введеної дати, а друга – об’єднання даних з двох текстових файлів.

Програми працюють коректно. Функція для перевірки дати може застосовуватися в програмах, що потребують валідації дат. Програма для об’єднання даних може бути використана для аналізу та обробки подібних даних.

Можливі напрямки модифікацій та розширення включають додавання додаткових перевірок у функцію для перевірки дати та розширення обробки різних форматів у програмі для об’єднання даних.

**Список літератури**

1Вивчаємо Python, В двох томах. 5-е видання. Марк Лутц (комплект з двох книг) – [Режим доступу] : <https://leader-books.com.ua/komp-yuterna-literatura/vivchayemo-python-v-dvoh-tomah-5-e-vidannya-mark-lutc-komplekt-z-dvoh-knig?gclid=CjwKCAiAgeeqBhBAEiwAoDDhnxzTW_ZFWceIMmEQLdDM0XPtB2uG9tUdKMSjvmw6BX7zHEoS49h0gxoC_skQAvD_BwE>

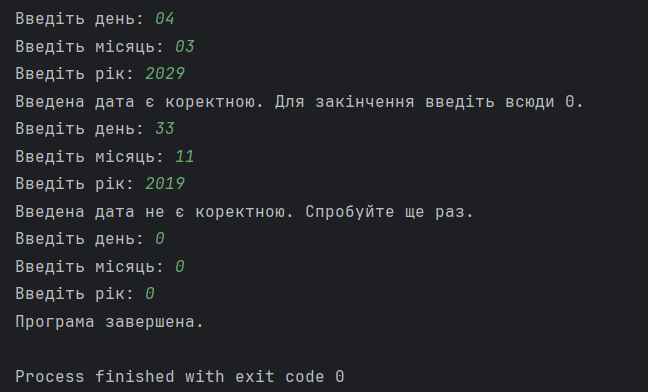
2 Форум з технічних питань – [Режим доступу] : <https://stackoverflow.com/>

**Додатки**

Текст програми №1

# Функція для перевірки коректності введених значень дня, місяця та року  
def is\_valid\_date(func\_day, func\_month, func\_year):  
 # Перевірка на коректність значень дня, місяця та року  
 if not (1 <= func\_day <= 31 and 1 <= func\_month <= 12 and 1000 <= func\_year <= 3000):  
 return False  
  
 # Перевірка на коректність днів у місяці  
 days\_in\_month = {  
 1: 31, 2: 29 if (func\_year % 4 == 0 and func\_year % 100 != 0) or (func\_year % 400 == 0) else 28,  
 3: 31, 4: 30, 5: 31, 6: 30, 7: 31, 8: 31, 9: 30, 10: 31, 11: 30, 12: 31  
 }  
  
 if not (1 <= func\_day <= days\_in\_month[func\_month]):  
 return False  
  
 return True  
  
  
# Безкінечний цикл для введення дати користувачем  
while True:  
 # Зчитування значень дня, місяця та року з консолі  
 day = int(input("Введіть день: "))  
 month = int(input("Введіть місяць: "))  
 year = int(input("Введіть рік: "))  
  
 # Перевірка введеної дати за допомогою функції is\_valid\_date  
 if is\_valid\_date(day, month, year):  
 print("Введена дата є коректною. Для закінчення введіть всюди 0.")  
 # Введено всюди 0, завершення програми  
 elif day == month == year == 0:  
 print("Програма завершена.")  
 break  
 # Виведення повідомлення про некоректну дату та повторення введення  
 else:  
 print("Введена дата не є коректною. Спробуйте ще раз.")

Результат роботи програми №1

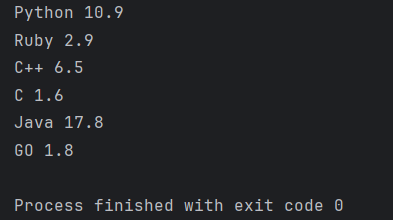


Текст програми №2

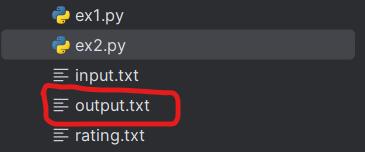
# Відкриття файлів для читання: 'input.txt' та 'rating.txt'  
with open('input.txt', 'r') as input\_file, open('rating.txt', 'r') as rating\_file:  
 # Читання даних з файлів та розділення рядків  
 languages = input\_file.read().splitlines()  
 ratings = rating\_file.read().splitlines()  
  
# Перевірка, чи кількість рядків у файлах співпадає  
if len(languages) != len(ratings):  
 # Виведення помилки, якщо кількість рядків не співпадає  
 print("Помилка: кількість рядків у файлах не співпадає.")  
else:  
 # Відкриття файлу для запису об'єднаних даних: 'output.txt'  
 with open('output.txt', 'w') as output\_file:  
 # Виведення об'єднаних даних на екран та запис у файл  
 for language, rating in zip(languages, ratings):  
 combined\_data = f"{language} {rating}"  
 # Виведення на екран об'єднаних даних  
 print(combined\_data)  
 # Запис об'єднаних даних у файл з новим рядком  
 output\_file.write(combined\_data + '\n')

Результат роботи програми №2

Вивід даних:



Поява нового файлу:



Зміст даного файлу:

