
	<p>Politechnika Bydgoska im. J. J. Śniadeckich</p> <p>Wydział Telekomunikacji, Informatyki i Elektrotechniki</p>		
Przedmiot	Skryptowe języki programowania		
Prowadzący	mgr inż. Martyna Tarczewska		
Temat	Napisy		
Student	Marcin Ogórkiewicz		
Nr ćw.	7	Data wykonania	14.11.2023
Ocena		Data oddania spr.	14.11.2023

Zadanie 1

```
#!C:\Users\koksu\AppData\Local\Programs\Python\Python310

# importy
import typing
# stałe i zmienne globalne

# funkcje

def process_text(file_path: str) -> None:
    with open(file_path, 'r', encoding='utf-8') as file:
        lines = file.readlines()
        total_lines = len(lines)
        total_words = 0
        total_characters = 0
        for i, line in enumerate(lines, start=1):
            line = line.strip()
            num_characters = len(line)
            num_words = len(line.split())
            print(f"Wiersz {i}: Ilość znaków: {num_characters}, Ilość słów: {num_words}")
            total_words += num_words
            total_characters += num_characters
        print("\nIlość wierszy:", total_lines)
        print("Ilość wszystkich słów:", total_words)
        print("Ilość wszystkich znaków:", total_characters)

def main() -> None:
    process_text("inwokacja.txt")
```

```
main()
```

```
C:\Users\koku\PycharmProjects\SkrzytoweJęzykiProgramowania\env\Scripts\python.exe C:\Users\koku\PycharmProjects\SkrzytoweJęzykiProgramowania\Lab07\Zadanie1.py
Wiersz 1: Ilość znaków: 44, Ilość słów: 7
Wiersz 2: Ilość znaków: 42, Ilość słów: 8
Wiersz 3: Ilość znaków: 50, Ilość słów: 9
Wiersz 4: Ilość znaków: 37, Ilość słów: 7
Wiersz 5: Ilość znaków: 43, Ilość słów: 6
Wiersz 6: Ilość znaków: 47, Ilość słów: 9
Wiersz 7: Ilość znaków: 44, Ilość słów: 6
Wiersz 8: Ilość znaków: 44, Ilość słów: 7
Wiersz 9: Ilość znaków: 40, Ilość słów: 7
Wiersz 10: Ilość znaków: 37, Ilość słów: 4
Wiersz 11: Ilość znaków: 44, Ilość słów: 8
Wiersz 12: Ilość znaków: 39, Ilość słów: 6
Wiersz 13: Ilość znaków: 41, Ilość słów: 7
Wiersz 14: Ilość znaków: 39, Ilość słów: 5
Wiersz 15: Ilość znaków: 48, Ilość słów: 8
Wiersz 16: Ilość znaków: 45, Ilość słów: 5
Wiersz 17: Ilość znaków: 40, Ilość słów: 6
Wiersz 18: Ilość znaków: 42, Ilość słów: 4
Wiersz 19: Ilość znaków: 50, Ilość słów: 7
Wiersz 20: Ilość znaków: 44, Ilość słów: 5
Wiersz 21: Ilość znaków: 43, Ilość słów: 6
Wiersz 22: Ilość znaków: 48, Ilość słów: 8

Ilość wierszy: 22
Ilość wszystkich słów: 145
Ilość wszystkich znaków: 951

Process finished with exit code 0
```

Zadanie 2

```
#!C:\Users\koku\AppData\Local\Programs\Python\Python310

# importy
import typing

# stałe i zmienne globalne

# funkcje

def count_whitespace(file_path: str) -> None:
    with open(file_path, 'r', encoding='utf-8') as file:
        file_text = file.read()
        num_newlines = file_text.count('\n')
        num_spaces = file_text.count(' ')
        num_tabs = file_text.count('\t')
        print("Number of newlines:", num_newlines)
        print("Number of spaces:", num_spaces)
        print("Number of tabs:", num_tabs)

def main() -> None:
    count_whitespace("inwokacja.txt")

main()
```

```
C:\Users\koku\PycharmProjects\SkrzytoweJęzykiProgramowania\venv\Scripts\python.exe C:\Users\koku\PycharmProjects\SkrzytoweJęzykiProgramowania\Lab07\Zadanie2.py
Number of newlines: 22
Number of spaces: 123
Number of tabs: 0
Process finished with exit code 0
```

Zadanie 3

```
#!C:\Users\koku\AppData\Local\Programs\Python\Python310

# importy
import typing

# stałe i zmienne globalne

# funkcje

def replace_dead_stops(file_path: str) -> None:
    with open(file_path, 'r', encoding='utf-8') as file:
        original_text = file.read()
        lines = original_text.split('\n')
        modified_lines = []
        for line in lines:
            words = line.split()
            modified_words = []
            for word in words:
                if '.' in word and '...' not in word:
                    word = word.replace('.', '...')
                modified_words.append(word)
            modified_lines.append(' '.join(modified_words))
        modified_text = '\n'.join(modified_lines)
        print(modified_text)

def main() -> None:
    replace_dead_stops("inwokacja.txt")

main()
```

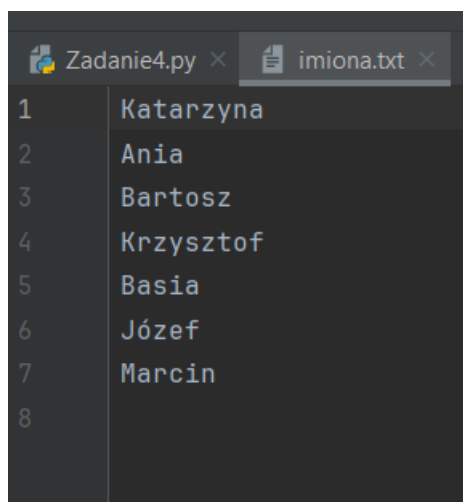
```

C:\Users\koku\PycharmProjects\SkrypoweJęzykiProgramowania\venv\Scripts\python.exe C:\Users\koku\PycharmProjects\SkrypoweJęzykiProgramowania\Lab07\Zadanie3.py
Litwo! Ojczyzno moja! ty jesteś jak zdrowie...
Ile cię trzeba cenić, ten tylko się dowie,
Kto cię stracił... Dziś piękność twą w całej ozdobie
Widzę i opisuję, bo tęsknię po tobie...
Panno Święta, co Jasnej bronisz Częstochowy
I w Ostrej świecisz Bramie! Ty, co gród zamkowy
Nowogródzki ochraniasz z jego wiernym ludem!
Jak mnie dziecko do zdrowia powróciłaś cudem
(Gdy od płaczącej matki pod Twoją opiekę
Ofiarowany, martwą podniosłem powiekę
I zaraz mogłem pieszo do Twych świątyn progu
Iść za wrócone życie podziękować Bogu),
Tak nas powrócisz cudem na Ojczyzny łono...
Tymczasem przenosę moję duszę utęsknioną
Do tych pagórków leśnych, do tych łąk zielonych,
Szeroko nad błękitnym Niemnem rozciągnionych;
Do tych pól malowanych zbożem rozmaitem,
Wyzłacanych pszenicą, posrebrzanych żytem;
Gdzie bursztynowy świerczop, gryka jak śnieg biała,
Gdzie panieńskim rumieńcem dzięcielina pała,
A wszystko przepasane, jakby wstęgą, miedzą
Zieloną, na niej z rzadka ciche grusze siedzą...

Process finished with exit code 0

```

Zadanie 4



```

1 Katarzyna
2 Ania
3 Bartosz
4 Krzysztof
5 Basia
6 Józef
7 Marcin
8

```

```

#!C:\Users\koku\AppData\Local\Programs\Python\Python310

# importy
import typing
import re
# stałe i zmienne globalne

# funkcje

def find_female_names(file_path: str) -> None:
    with open(file_path, 'r', encoding='utf-8') as file:
        text = file.read()
        pattern = r'\b[A-Z][a-z]*a\b'
        female_names = re.findall(pattern, text)
        print(f"Damskie imiona znalezione w pliku: {file_path}")
        for name in female_names:

```

```

        print(name)

def main() -> None:
    find_female_names("imiona.txt")

main()

```

```

C:\Users\koku\PycharmProjects\SkryptoweJęzykiProgramowania\venv\Scripts\python.exe C:\Users\koku\PycharmProjects\SkryptoweJęzykiProgramowania\Lab07\Zadanie4.py
Damskie imiona znalezione w pliku: imiona.txt
Katarzyna
Ania
Basia
Process finished with exit code 0

```

Zadanie 5

```

#!C:\Users\koku\AppData\Local\Programs\Python\Python310

# importy
import typing
import re
# stałe i zmienne globalne

# funkcje

def polish_phone_numbers(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        text = file.readlines()
        pattern = r'(\+48|0048)'
        polish_numbers_list = []
        for number in text:
            if re.match(pattern, number):
                polish_numbers_list.append(number)
        print("Polskie numery telefoniczne znalezione w pliku:")
        for number in polish_numbers_list:
            print(number)

def main() ->None:
    polish_phone_numbers("numery.txt")

main()

```

```

C:\Users\koku\PycharmProjects\SkrptoweJęzykiProgramowania\venv\Scripts\python.exe C:\Users\koku\PycharmProjects\SkrptoweJęzykiProgramowania\Lab07\Zadanie5.py
Polskie numery telefoniczne znalezione w pliku:
+48592712569

+48858745965

0048686748125

0048753963951

+48 501 241 665

Process finished with exit code 0

```

Zadanie 6

```

#!C:\Users\koku\AppData\Local\Programs\Python\Python310

# importy
import typing
import re

# stałe i zmienne globalne

# funkcje

def read_phone_numbers(file_path: str) -> list:
    with open(file_path, 'r', encoding='utf-8') as file:
        numbers = [line.strip() for line in file.readlines()]

    return numbers

def validate_phone_numbers(numbers: list) -> dict:
    validation_patterns = {
        "1. xxx-xxx-xxx": r'^\d{3}-\d{3}-\d{3}$',
        "2. xxxxxxxxx": r'^\d{9}$',
        "3. +48xxxxxxxxxx": r'^\+48\d{9}$',
        "4. 0048xxxxxxxxxx": r'^0048\d{9}$',
        "5. +48 xxx xxx xxx": r'^\+48\s\d{3}\s\d{3}\s\d{3}$',
    }
    counts = {key: 0 for key in validation_patterns}
    for number in numbers:
        for pattern_description, pattern in validation_patterns.items():
            if re.match(pattern, number):
                counts[pattern_description] += 1
                break
    return counts

def main() -> None:
    file_path = 'numery.txt'
    phone_numbers = read_phone_numbers(file_path)
    counts = validate_phone_numbers(phone_numbers)
    print("Ilość numerów telefonu o danym formacie:")
    for pattern_description, count in counts.items():
        print(f"{pattern_description}: {count} numery")

main()

```

```
C:\Users\koku\PycharmProjects\SkrzytoweJęzykiProgramowania\venv\Scripts\python.exe C:\Users\koku\PycharmProjects\SkrzytoweJęzykiProgramowania\Lab07\Zadanie6.py
Ilość numerów telefonu o danym formacie:
1. xxx-xxx-xxx: 2 numery
2. xxxxxxxx: 3 numery
3. +48xxxxxxxx: 2 numery
4. 0048xxxxxxxx: 2 numery
5. +48 xxx xxx xxx: 1 numery

Process finished with exit code 0
```

Zadanie 7

```
r'^[a-z][a-z0-9_.-]*@gmail\.com$'
```

Zadanie 8

```
#!C:\Users\koku\AppData\Local\Programs\Python\Python310

# importy
import typing
import re
# stałe i zmienne globalne

# funkcje

def validate_date(input_date: str) -> None:
    date_pattern = re.compile(r'^(0[1-9]|12)[0-9]|3[01])[-/](0[1-9]|1[0-2])[-/](\d{4})$')
    match = re.match(date_pattern, input_date)

    if match:
        day, month, year = map(int, match.groups())
        if 1 <= day <= 31 and 1 <= month <= 12:
            months = [
                "Styczeń", "Luty", "Marzec", "Kwiecień",
                "Maj", "Czerwiec", "Lipiec", "Sierpień",
                "Wrzesień", "Październik", "Listopad", "Grudzień"
            ]
            print(f"Miesiąc dla podanej daty: {months[month - 1]}")
        else:
            print("Niepoprawna ilość dni lub miesięcy")
    else:
        print("Niepoprawny format daty")

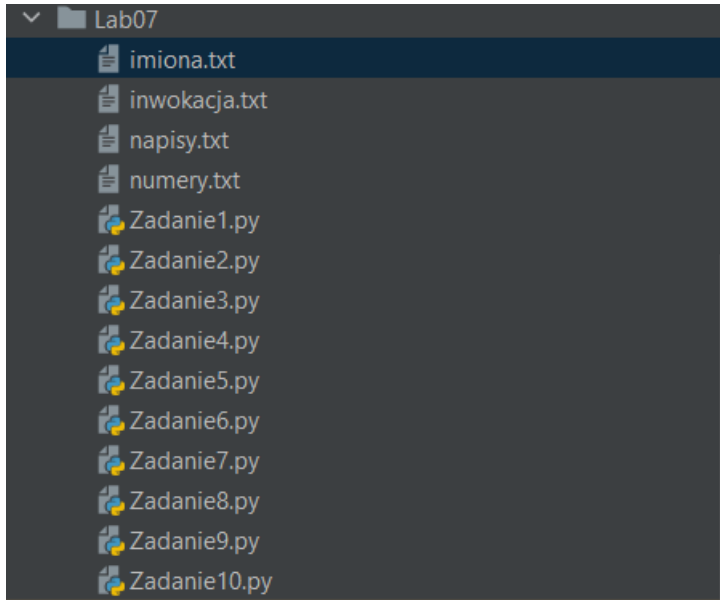
def main() -> None:
    user_input = input("Proszę wpisać datę (przyjmowane formaty to dd-mm-rrrr or dd/mm/rrrr): ")
    validate_date(user_input)

main()
```

```
C:\Users\koku\PycharmProjects\SkrptoweJęzykiProgramowania\venv\Scripts\python.exe C:\Users\koku\PycharmProjects\SkrptoweJęzykiProgramowania\Lab07\Zadanie8.py
Proszę wpisać datę (przyjmowane formaty to dd-mm-rrrrr or dd/mm/rrrrr): 11-11-2021
Miesiąc dla podanej daty: Grudzień

Process finished with exit code 0
```

Zadanie 9



```
#!C:\Users\koku\AppData\Local\Programs\Python\Python310

# importy
import typing
import re
import os

# stałe i zmienne globalne

# funkcje

def list_txt_files(directory_path: str) -> None:
    txt_files_pattern = re.compile(r'.*\.txt$', re.IGNORECASE)
    txt_files = [file for file in os.listdir(directory_path) if
re.match(txt_files_pattern, file)]
    if txt_files:
        print(f"Pliki tekstowe w katalogu {directory_path}:")
        for txt_file in txt_files:
            print(txt_file)
    else:
        print(f"W katalogu {directory_path} nie znaleziono żadnych plików
tekstowych.")

def main() -> None:
    directory_path = input("Wprowadź ścieżkę dostępu: ")
    if os.path.isdir(directory_path):
        list_txt_files(directory_path)
    else:
```



```

        print(f"{directory_path} nie jest poprawną ścieżką dostępu.")

main()

```

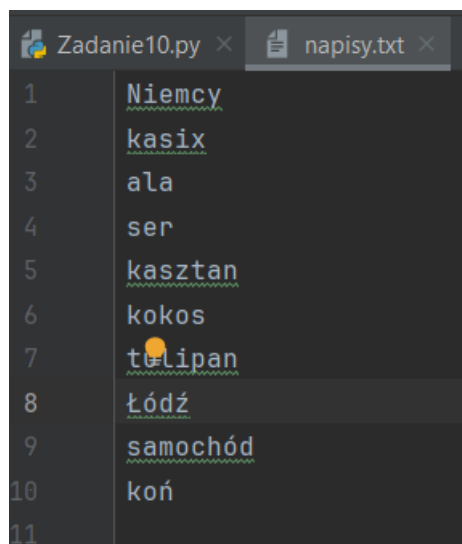
```

C:\Users\koku\PycharmProjects\SkrptoweJęzykiProgramowania\venv\Scripts\python.exe C:\Users\koku\PycharmProjects\SkrptoweJęzykiProgramowania\Lab07\Zadanie9.py
Wprowadź ścieżkę dostępu: C:\Users\koku\PycharmProjects\SkrptoweJęzykiProgramowania\Lab07
Pliki tekstowe w katalogu C:\Users\koku\PycharmProjects\SkrptoweJęzykiProgramowania\Lab07:
imiona.txt
inwokacja.txt
napisy.txt
numery.txt

Process finished with exit code 0

```

Zadanie 10



```

1  Niemcy
2  kasix
3  ala
4  ser
5  kasztan
6  kokos
7  tulipan
8  Łódź
9  samochód
10 koń
11

```

```

#!C:\Users\koku\AppData\Local\Programs\Python\Python310

# importy
import typing
import re

# stałe i zmienne globalne

# funkcje

def process_strings(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        lines = file.readlines()
    end_with_x_or_y = []
    three_chars_start_with_a = []
    starts_with_vowel = []
    for line in lines:
        line = line.strip()
        if re.search(r'[xy]$', line):
            end_with_x_or_y.append(line)

```

```

        if re.match(r'^a\w{2}$', line):
            three_chars_start_with_a.append(line)
        if re.match(r'^[aeiouyAEIOUY]', line):
            starts_with_vowel.append(line)
    print("Elements ending with x or y:")
    print(end_with_x_or_y)
    print("\nThree-character elements starting with a:")
    print(three_chars_start_with_a)
    print("\nElements starting with a vowel:")
    print(starts_with_vowel)

def main() -> None:
    file_path = 'napisy.txt'
    process_strings(file_path)

main()

```

```

C:\Users\koku\PycharmProjects\SkrzytweJęzykiProgramowania\venv\Scripts\python.exe C:\Users\koku\PycharmProjects\SkrzytweJęzykiProgramowania\Lab07\Zadanie10.py
Elements ending with x or y:
['Niemcy', 'kasix']

Three-character elements starting with a:
['ala']

Elements starting with a vowel:
['ala']

Process finished with exit code 0

```

Wnioski

Ćwiczenie nauczyło mnie pracy na ciągach znaków przy pomocy funkcji wbudowanych oraz biblioteki re.