Практическое занятие № 11 №1

Тема: Составление программ для работы с текстовыми файлами.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с текстовыми файлами в IDE PyCharm Community.

Постановка задачи.

Средствами языка Python сформировать два текстовых файла (.txt), содержащих по одной последовательности из целых положительных и отрицательных чисел. Сформировать новый текстовый файл (.txt) следующего вида, предварительно выполнив требуемую обработку элементов:

1. Элементы первого и второго файлов:
2. Элементы первого файла, отсутствующие во втором:
3. Элементы второго файла, отсутствующие в первом:
4. Количество элементов:
5. Индекс первого минимального элемента:
6. Индекс последнего максимального элемента:

Тип алгоритма: линейный.

Текст программы:

```
import random
from pprint import pprint


def generating_files():
    # Генерируем случайную последовательность чисел для первого файла
    sequence1 = [random.randint(-100, 100) for _ in range(10)]
    with open('file1.txt', 'w') as f1:
        f1.write(' '.join(map(str, sequence1)))

    # Генерируем случайную последовательность чисел для второго файла
    sequence2 = [random.randint(-100, 100) for _ in range(10)]
    with open('file2.txt', 'w') as f2:
        f2.write(' '.join(map(str, sequence2)))


def preprocessor():
    # Чтение последовательностей чисел из файлов
    with open('file1.txt', 'r') as f1:
        sequence1 = list(map(int, f1.read().split()))

    with open('file2.txt', 'r') as f2:
        sequence2 = list(map(int, f2.read().split()))

    result = []

    result.append(f"Элементы первого и второго файлов: {sequence1 +
sequence2}")
    result.append(f"Элементы первого файла, отсутствующие во втором:
{list(set(sequence1) - set(sequence2))}")
    result.append(f"Элементы второго файла, отсутствующие в первом:
{list(set(sequence2) - set(sequence1))}")
    result.append(f"Количество элементов: {len(sequence1 + sequence2)}")
    result.append(f"Индекс первого минимального элемента:
{sequence1.index(min(sequence1))}")
    result.append(f"Индекс последнего максимального элемента:
{len(sequence1) + sequence2[::-1].index(max(sequence2))}")

    # Элементы первого и второго файлов
    print(f"Элементы первого и второго файлов: {sequence1 + sequence2}")

    # Элементы первого файла, отсутствующие во втором
    print(f"Элементы первого файла, отсутствующие во втором:
{list(set(sequence1) - set(sequence2))}")

    # Элементы второго файла, отсутствующие в первом
    print(f"Элементы второго файла, отсутствующие в первом:
{list(set(sequence2) - set(sequence1))}")

    # Количество элементов
    print(f"Количество элементов: {len(sequence1 + sequence2)}")

    # Индекс первого минимального элемента
    print(f"Индекс первого минимального элемента:
{sequence1.index(min(sequence1))}")
```

```
    # Индекс последнего максимального элемента
    print(f"Индекс последнего максимального элемента: {len(sequence1) +
sequence2[::-1].index(max(sequence2))}")


    # Запись результатов в новый файл
    with open('result.txt', 'w') as result_file:
        result_file.write('\n'.join(result))


generating_files()
preprocessor()
```

Протокол работы программы:

Элементы первого и второго файлов: [-65, 23, -8, 85, 66, 60, 80, 43, 94, 76, 27, 65, 98, 26, -37, 77, -78, -98, -58, 79]

Элементы первого файла, отсутствующие во втором: [66, 43, 76, 80, 85, 23, -8, 60, 94, -65]

Элементы второго файла, отсутствующие в первом: [65, 98, -58, 27, 77, 79, -78, 26, -37, -98]

Количество элементов: 20

Индекс первого минимального элемента: 0

Индекс последнего максимального элемента: 17

Process finished with exit code 0

Вывод:

Этот алгоритм представляет собой комбинацию случайной генерации чисел и сравнения между двумя файлами.

Шаги алгоритма:

1. generating_files():
   - Генерируется случайная последовательность из 10 целочисленных чисел от -100 до 100 для каждого из двух файлов 'file1.txt' и 'file2.txt'.

2. preprocessor():
   - Читаются последовательности чисел из файлов 'file1.txt' и 'file2.txt'.
   - Выполняются следующие операции:
   - Объединяются числа из обоих файлов и добавляются в результат.
   - Находятся элементы, которые есть в первом файле, но отсутствуют во втором.
   - Находятся элементы, которые есть во втором файле, но отсутствуют в первом.
   - Считается общее количество элементов в обоих файлах.
   - Находится индекс первого минимального элемента в первом файле.
   - Находится индекс последнего максимального элемента из второго файла.

3. Результаты записываются в файл 'result.txt', который содержит: - Элементы первого и второго файлов.
   - Элементы первого файла, которые отсутствуют во втором.
   - Элементы второго файла, которые отсутствуют в первом.
   - Общее количество элементов.
   - Индекс первого минимального элемента из первого файла.
   - Индекс последнего максимального элемента из второго файла.

Код также выводит результаты на экран с помощью pprint и print. Готовые программные коды выложены на GitHub.

Практическое занятие № 11 №2

Тема: Составление программ для работы с текстовыми файлами.
Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с текстовыми файлами в IDE PyCharm Community.

Постановка задачи.

Из предложенного текстового файла (text18-26.txt) вывести на экран его содержимое, количество знаков препинания. Сформировать новый файл, в который поместить текст в стихотворной форме предварительно заменив все знаки пунктуации на знак «/».

Тип алгоритма: линейный.

Текст программы:

```
# -*- encoding: utf-8 -*-
# Чтение содержимого текстового файла with
open('text18-26.txt', 'r') as text_file:
    text = text_file.read()

# Подсчет количества знаков препинания
punctuation_count = sum(1 for char in text if char in '.,?!:;--')
# Замена знаков пунктуации на '/'
text_with_slash = ''.join(['/' if char in '.,?!:;--' else char for char in
text])

# Вывод информации на экран
print("Содержимое текстового файла:")
print(text_with_slash)
print("Количество знаков препинания:", punctuation_count)
# Запись текста с заменой пунктуации в новый файл
with open('poem.txt', 'w') as poem_file:
    poem_file.write(text_with_slash)
```

Протокол работы программы:

Содержимое текстового файла:

И вот нашли большое поле:

Есть разгуляться где на воле!

Построили редут.

У наших ушки на макушке!

Чуть утро осветило пушки

И леса синие верхушки —

Французы тут как тут.

Количество знаков препинания: 6

Process finished with exit code 0

Вывод:

Этот алгоритм относится к простой обработке текста и счету определенных символов в нем.

Шаги алгоритма:

1. Чтение содержимого текстового файла 'text18-26.txt'.
2. Подсчет количества знаков препинания в тексте.
3. Замена всех знаков пунктуации на символ '/'.
4. Вывод информации на экран:
    - Сначала выводится содержимое текстового файла с заменой пунктуации на '/'.
    - Затем выводится количество знаков препинания в тексте.
5. Запись отредактированного текста с замененной пунктуацией в новый файл 'poem.txt'.

Таким образом, этот алгоритм считывает текст из файла, подсчитывает количество определенных символов в нем, заменяет их на другой символ, выводит результат на экран и сохраняет отредактированный текст в новый файл.

Готовые программные коды выложены на GitHub.