

Практическое занятие №11

Тема: Составление программ со словарями в IDE PyCharm Professional.

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

Постановка задачи:

1) Средствами языка Python сформировать текстовый файл (.txt), содержащий последовательность из целых положительных и отрицательных чисел. Сформировать

новый текстовый файл (.txt) следующего вида, предварительно выполнив требуемую

обработку элементов:

Исходные данные:

Количество элементов:

Минимальный элемент:

Квадраты четных элементов:

Сумма квадратов четных элементов:

Среднее арифметическое суммы квадратов четных элементов:

2) Из предложенного текстового файла (text18-3.txt) вывести на экран его содержимое,

количество знаков пунктуации в первых четырёх строках. Сформировать новый файл, в

который поместить текст в стихотворной форме предварительно заменив символы третей

строки их числовыми кодами.

Тип алгоритма: Ветвящийся

Текст программы:

1)

```
import random

import os


def create_file(numbers: list) -> None:

    with open("File_numbers.txt", "w", encoding='utf-8') as file:

        file.write(str(numbers))


def random_number() -> list:

    num = random.randint(50, 100)

    rnd = [random.randint(-100, 100) for i in range(num)]

    create_file(numbers=rnd)


def read_file() -> list:

    if os.path.isfile("File_numbers.txt"):

        with open("File_numbers.txt", "r", encoding='utf-8') as
file:
```

```
        numbers = eval(file.read())

    return numbers

return None


def save_result(data: str) -> None:

    with open("File_result.txt", "w", encoding='utf-8') as file:

        file.write(data)


def processing_function(numbers: list) -> str:

    square_list = [i ** 2 for i in numbers if i > 0]

    str_data = (f"# Исходные данные: \n{numbers}\n\n"

                f"# Количество элементов: \n{len(numbers)}\n\n"

                f"# Минимальный элемент: \n{min(numbers)}\n\n"

                f"# Квадраты четных чисел: \n{square_list} \n\n"

                f"# Сумма квадратов четных элементов: \n{sum(square_list)}\n\n"

                f"# Среднее арифметическое суммы квадратов четных чисел: \n{round(sum(square_list) / len(square_list), 2)}")

    save_result(str_data)

    return str_data


def main():
```

```
random_number()

numbers = read_file()

print(processing_function(numbers))

if __name__ == "__main__":

    main()
```

Исходные данные:

[-47, 68, 20, -91, -37, 34, -7, 49, -64, -82, -27, 73, 99, -89, 66, 92, -85, -81, -8, -47, 15, -63, 65, -41, -81, 64, -67, -63, -62, 28, -69, -30, -28, -46, -4, 83, 56, -43, 71, 53, -92, -58, -69, 43, -15, -99, -7, 9, -63, -42, -72, -38, 33, 29, -22, 82, 15, -27, -66, -51, 71, 72, -89, 82, -21, -37, 26, -14, -30, -99, -95, 81, -37, 99, 91, 15, -16, 34, 15, 65, -92, -27]

Количество элементов:

82

Минимальный элемент:

-99

Квадраты четных чисел:

[4624, 400, 1156, 2401, 5329, 9801, 4356, 8464, 225, 4225, 4096, 784, 6889, 3136, 5041, 2809, 1849, 81, 1089, 841, 6724, 225, 5041, 5184, 6724, 676, 6561, 9801, 8281, 225, 1156, 225, 4225]

Сумма квадратов четных элементов:

122644

Среднее арифметическое суммы квадратов четных чисел:

3716.48

Process finished with exit code 0

2)

```
import os

from chardet.universaldetector import UniversalDetector


def search_charset(file_name: str) -> str:

    """
    Функция для определения кодировки файла
    :param file_name: путь до файла
    :return: кодировка файла
    """

    detector = UniversalDetector()

    with open(file_name, 'rb') as fh:

        for line in fh:

            detector.feed(line)

            if detector.done:

                break

        detector.close()

    return detector.result['encoding']


def read_file(file_name: str) -> str:

    if os.path.isfile(file_name):

        with open(file_name, "r",
encoding=search_charset(file_name)) as file:

            return file.read()

    return ""
```

```
def proccessing_data_punctuation(text: str) -> int:

    punctuation_library = ['.', ',', ':', ';', '?', '«', '»']

    count_punctuation = 0

    new_text = ' '.join(text.split('\n')[:4])

    for i in new_text:

        if i in punctuation_library:

            count_punctuation += 1

    return count_punctuation


def replace_symbols(text: str) -> str:

    new_text = text.split('\n')

    for i in new_text[2]:

        new_text[2] = new_text[2].replace(i, str(ord(i)))

    return '\n'.join(new_text)


def save_result(text: str):

    with open(r'File_result_2.txt', 'w', encoding='utf-8') as file:

        file.write(text)
```

```

def main() -> None:

    text = read_file("text18-3.txt")

    print(f"Содержимое текстового файла:
\n{text}\n-----\n")

    len_punctuation = processing_data_punctuation(text=text)

    print(f"Кол-во знаков пунктуации: {len_punctuation}")

    save_result(replace_symbols(text))

if __name__ == '__main__':

    main()

```

Содержимое текстового файла:
Мы долго молча отступали,
Досадно было, боя ждали,
Ворчали старики:
«Что ж мы? на зимние квартиры?
Не смеют, что ли, командиры
Чужие изорвать мундиры
О русские штыки?»

Кол-во знаков пунктуации: 7

Process finished with exit code 0

Вывод: В процессе выполнения практического занятия №11 я приобрел навыки составления программ, приобрести навыки составление

программ с текстовыми файлами в IDE PyCharmProfessional.