

Практическое занятие № 14

Тема: Составление программ в функциональном стиле в IDE PyCharm Community.

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

Постановка задачи.

Приложение СДАЧА В АРЕНДУ ТОРГОВЫХ ПЛОЩАДЕЙ для некоторой организации. БД должна содержать таблицу Клиент со следующей структурой записи: ФИО клиента, код помещения, срок аренды, стоимость аренды за весь срок.

Тип алгоритма: линейный.

Текст программы:

```
# Вариант 31
# Из исходного текстового файла (price.txt) выбрать все цены. Посчитать
# количество полученных элементов.
import re
price = 'Округление розничных цен с примерамиВарианты округления розничных
цен Округление рассчитанной розничной цены до десятков копеек. Например,
необходимо округлить цену до 10 копеек. В этом случае можно выбрать один из
следующих вариантов: до десятков копеек вверх, например, 1 руб. 01 коп. будет
округлено до 1 руб. 10 коп., 1 руб. 52 коп. будет округлено до 1 руб. 60коп.
до десятков копеек вниз, например, 1 руб. 09коп. будет округлено до 1 руб. 00
коп., 1руб. 99 коп. будет округлено до 1руб. 90коп.) до десятков копеек,
например, все цены от 100 руб. 55коп. до 100 руб. 64 коп. будут округлены до
100 руб. 60 коп., от 1 руб. 65 коп. до 1руб. 74 коп. будут округлены до 1
руб. 70 коп.'
```

```
with open('price.txt', 'r', encoding='utf-8') as file:
    text = file.read()
findalldigits = re.findall(r'\d+\.\w+', price)
with open('findalldigits.txt', 'w') as file:
    file.write(','.join(findalldigits))
print(len(findalldigits))
```

Протокол работы программы:

```
10 копеек., 1 руб., 01 коп., 1 руб., 10 коп., 1 руб., 52 коп., 1 руб.,
60коп., 1 руб., 09коп., 1 руб., 00 коп., 1руб., 99 коп., 1руб., 90коп., 100
```

```
руб., 55коп., 100 руб., 64 коп., 100 руб., 60 коп., 1 руб., 65 коп., 1руб.,  
74 коп., 1 руб., 70 коп.
```

29

Process finished with exit code 0

Вывод по проделанной работе:

Используя Python для извлечения цен из текстового файла, я успешно отфильтровал числовые значения из исходного текста. Применяв регулярные выражения, я извлек все целые числа из файла, создав новый файл с отфильтрованными данными.

Этот процесс продемонстрировал мои навыки обработки текста и манипулирования файлами на Python. Кроме того, это задание укрепило мое понимание регулярных выражений и их применения для извлечения и обработки данных. Мой опыт, включая разработку, тестирование и отладку кода, во многом способствовал успешному выполнению этой задачи.

Готовые коды выложены на GitHub.