

**Министерство образования и науки Российской Федерации
Федерально государственное автономное образовательное
учреждение высшего образования
«Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики»**



Мегафакультет: Компьютерных технологий и управления
Факультет: Безопасности информационных технологий
Кафедра: Проектирования и безопасности компьютерных систем
Направление (специальность): 10.03.01 «Информационная безопасность»

**Лабораторная работа №1
на тему
«Обработка и тарификация CDR (Call Detail Record)»
ВАРИАНТ № 3**

Выполнил:

Студент гр.N3352

/Распутина А.А.

Проверил:

Федоров Иван Романович

Санкт-Петербург

2020 г.

Цель работы:

Изучить и программно реализовать простейшее правило тарификации для услуг типа “Телефония” по длительности разговора и “СМС” по общему количеству. Работа включает в себя 2 этапа:

1. Парсинг файла CDR и выборка нужных строк для обработки
2. Тарификация выбранных записей

Задание:

Вариант №3. Протарифицировать абонента с номером 915783624 с коэффициентом k : 2руб/минута исходящие звонки, но 20 минут бесплатно, 0руб/минута входящие, смс - 2руб/шт

Теоретическая часть

При совершении звонка абонентом АТС формирует файлы с данными CDR (call detail records), которые загружаются в биллинг. Далее звонки тарифицируются. В процессе тарификации происходит начисление на операционные счета услуг, после чего выставляются платежные документы (биллинговые счета) и отправляются клиентам.

Правила тарификации услуг “Телефония”:

$$X = T * k,$$

где X - итоговая стоимость всех звонков абонента, T - общая длительность звонков (сумма длительностей всех записей по абоненту в файле), k - множитель тарифного плана (у каждого варианта свой).

Правила тарификации услуг “СМС”:

$$Y = N * k,$$

где Y - итоговая стоимость всех СМС абонента, N - общее количество СМС (сумма числа всех СМС в записях по абоненту в файле), k - множитель тарифного плана (у каждого варианта свой).

Практическая часть

В качестве среды разработки программного обеспечения (консольного приложения) была использована интегрированная среда разработки PyCharm и язык программирования Python.

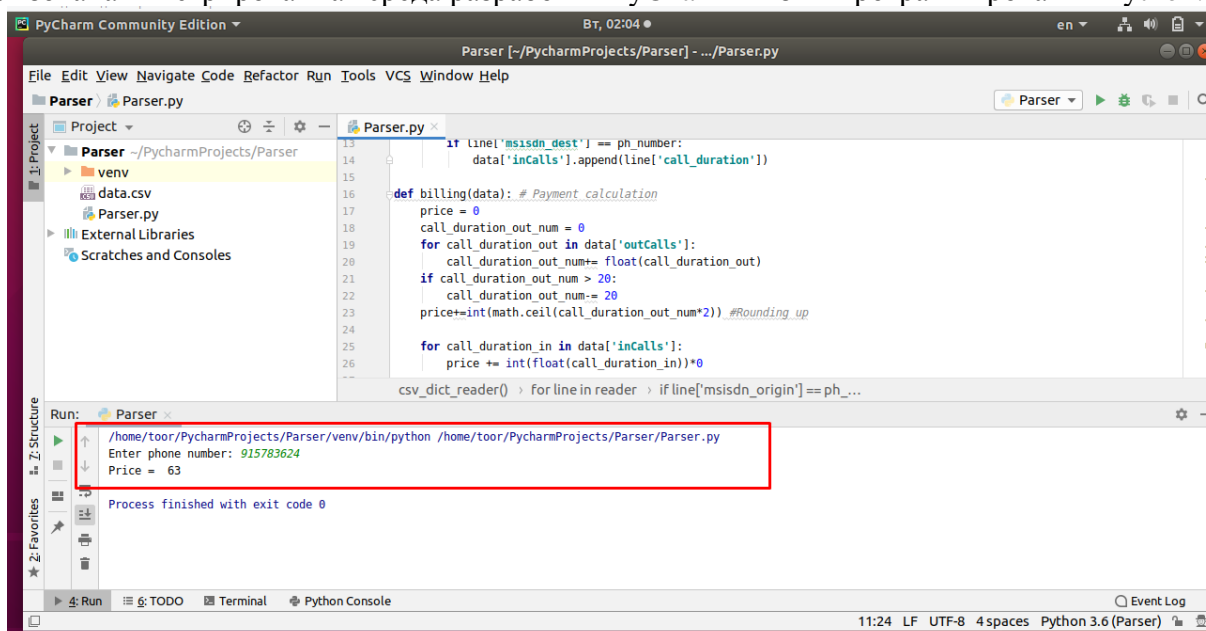


Рис. 1 – Расчет тарифа, отладка приложения в PyCharm

Проверка:

Данные из файла:

timestamp,msisdn_origin,msisdn_dest,call_duration,sms_number

2020-01-01 00:00:00,915783624,911926375,36.23,15

36.23 минут – 20 = 16.23 * 2 = 32.46 – стоимость исходящих, окр. в большую сторону до 33

15 смс стоимостью 2 руб = 30 руб

Итого к оплате: 63 руб

Запуск программы на исполнение производится при помощи команды из директории Parser с файлом Parser.py (файл с данными data.csv необходим для корректной работы программы):

python3 Parser.py

(либо python Parser.py)

После запуска программы необходимо ввести номер телефона.

В результате работы программы будет выведена сумма к оплате (если номер отсутствует в файле, будет выведена цена 0 руб):

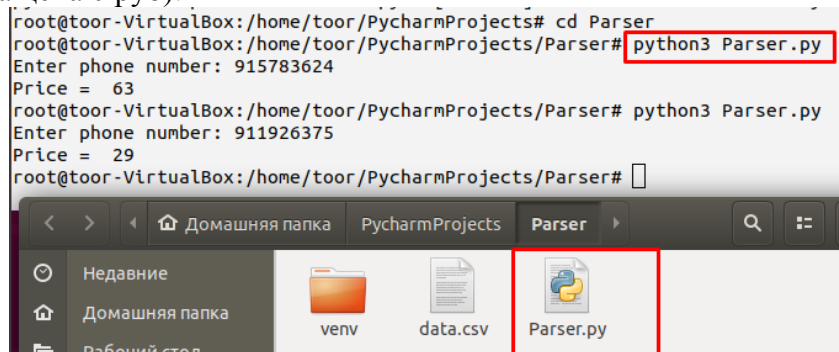


Рис. 2 – Запуск программы на исполнение

Выводы

В результате проделанной работы были изучены и программно реализованы на Python правила тарификации для услуг типа “Телефония” по длительности разговора и “СМС”.

Протарифицирован абонент с номером 915783624 с коэффициентом к: 2руб/минута исходящие звонки, но 20 минут бесплатно, 0руб/минута входящие, смс - 2руб/шт

Приложение Parser.py:

```
import csv
import math

data = {'outCalls': [], 'inCalls': [], 'sms': []} # Array for storing incoming, outgoing
calls and the number of SMS

def csv_dict_reader(file_obj, ph_number): #Function for parsing a csv - file and filling
the array with data
    reader = csv.DictReader(file_obj, delimiter=',')

    for line in reader:
        if line['msisdn_origin'] == ph_number:
            data['outCalls'].append(line['call_duration'])
            data['sms'].append(line['sms_number'])
        if line['msisdn_dest'] == ph_number:
            data['inCalls'].append(line['call_duration'])

def billing(data): # Payment calculation
    price = 0
    call_duration_out_num = 0
    for call_duration_out in data['outCalls']:
        call_duration_out_num += float(call_duration_out)
    if call_duration_out_num > 20:
        call_duration_out_num -= 20
    price += int(math.ceil(call_duration_out_num*2)) #Rounding up

    for call_duration_in in data['inCalls']:
        price += int(float(call_duration_in))*0

    for sms_number in data['sms']:
        price += int(float(sms_number))*2

    return price

with open("data.csv") as f_obj:
    ph_number = input("Enter phone number: ")
    csv_dict_reader(f_obj, ph_number)
    print('Price = ', billing(data))
```