

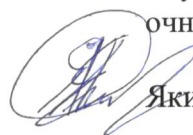
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
Университет ИТМО

**ФАКУЛЬТЕТ БЕЗОПАСНОСТИ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Лабораторная работа №1
по теме «Обработка и тарификация CDR»
по предмету «Управление мобильными устройствами»

Работу выполнил
Студент группы №3347
очного отделения:



Якимов Ярослав 29.03.20

Проверил

Федоров И.Р.



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2020

Цель работы: познакомиться с форматом представления данных – CDR и как он применяется на практике.

Задачи:

- Парсинг файла csv, содержащий Call Detail Records
- Тарификация выбранных записей по своему варианту

Средства реализации: Для реализации был выбран язык программирования Python, как отвечающий всем современным требованиям в разработке. Кроме того, он предоставляет возможность написать быстрый код, как по времени разработки, так и по времени исполнения. Тем более он имеет встроенные модули для работы с файлами csv.

Исходный код (так же по [ссылке](#)):

```
import csv
import time
from math import ceil

balance = 0 # Итоговый счет для номера
input_file = "data.csv"
msisdn_origin = '933156729'
first_sms = 10 # Количество "первых" смс, как только доходит до
0 - считается всё по основному тарифу
first_minutes_out = 0 # Количество "первых" исходящих минут
first_minutes_in = 0 # Количество "первых" входящих минут
in_cost = 0 # Стоимость входящих
out_cost = 2 # Стоимость исходящих
sms_cost = 2 # Стоимость смс

def check(row, msisdn_origin, in_cost, out_cost, sms_cost,
first_sms_cost=0, first_minutes_out_cost=0,
first_minutes_in_cost=0):
    global balance
    global first_sms
    global first_minutes_out
    if msisdn_origin == row['msisdn_origin']: # Если это
исходящие звонки, то тарификация по исходящим
        balance += (ceil(
            float(row['call_duration'])) - first_minutes_out) *
out_cost + first_minutes_out * first_minutes_out_cost
        # Считаем сколько стоят исходящие минуты, и т.к. "первые"
минуты считаются по звонку, то мы их не сбрасываем
        if first_sms != 0: # смс сбрасываются, так что
            if first_sms < int(row['sms_number']):
                balance += (int(row['sms_number']) - first_sms) *
sms_cost + first_sms * first_sms_cost
                first_sms = 0
            else:
                first_sms -= int(row['sms_number'])
        else:
```

```

        balance += (int(row['sms_number'])) * sms_cost
    if msisdn_origin == row['msisdn_dest']: # Входящие звонки
        balance += (ceil(
            float(row['call_duration'])) - first_minutes_in) *
in_cost + first_minutes_in * first_minutes_in_cost
    return balance
start_time = time.time()
data = csv.DictReader(open(input_file))
result = [check(row, msisdn_origin, in_cost, out_cost, sms_cost)
for row in data]

with open('result.txt', 'w') as f:
    f.write(f'Total billing: {balance}\n')
    f.write(f"Took {time.time() - start_time} seconds")

```

Выводы: Я познакомился с форматом CDR и научился тарифицировать абонентов, на основании файлов с записями о звонках и смс