**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ФАКУЛЬТЕТ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**Управление мобильными устройствами**

Отчёт по лабораторной работе №1

«Обработка и тарификация CDR (Call Detail Record)»

**Выполнил:**

Изображение выглядит как рисунок, зеркало

Автоматически созданное описаниеСтудент группы N3347

Светлов А. Д. \_\_\_\_\_\_\_\_\_\_

**Проверил:**

Университет ИТМО

Федоров Иван Романович

\_\_\_\_\_\_\_\_\_\_

Дата защиты: 15.04.20

Санкт-Петербург

2020

**Цель работы**: необходимо реализовать программный модуль для обработки CDR и тарификации абонента АТС, который выполняет обработку файла CDR и два правила тарификации типа “Телефония” по длительности разговора и “СМС” по общему количеству СМС.

Работа включает в себя 2 этапа:

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

Правила тарификации услуг “Телефония”: X = T \* k, где X - итоговая стоимость всех звонков абонента, T - общая длительность звонков, k - множитель тарифного плана.

Правила тарификации услуг “СМС”: Y = N \* k, где Y - итоговая стоимость всех СМС абонента, N - общее количество СМС, k - множитель тарифного плана.

Вариант работы: абонент с номером 915642913 с коэффициентом k: 1руб/минута исходящие звонки, 1руб/минута входящие, смс - первые 5шт бесплатно, далее 1руб/шт.

**Выбранные средства реализации и обоснование выбора:**

В качестве средства реализации был выбран язык программирования Python версии 3.8.2 и среда программирования PyCharm. Выбор пал именно на это средство, так как язык кроссплатформенный, гибкий в реализации различных методов и имеет простой синтаксис.

**Исходный код:**

file = open('data.csv')  
#Раздел "Парсинг"  
  
#объявляем необходимые переменные  
msisdn\_origin = 0  
msisdn\_dest = 0  
call\_durationISH = 0  
call\_durationVH = 0  
sms\_number = 0  
i = 0  
t = 0  
#получаем все строки в виде списка  
l = file.readlines()  
#получаем кол-во строк  
s = len(l)  
#зная нужный номер абонента выбираем нужные строки и данные CDR  
print('Введите номер абонента:', end = ' ')  
nomer = input()  
if nomer == '915642913':  
#проходимся по спискам l и создаём подсписки  
 for i in range(s):  
 x = l[i].split(',')  
#получаем длину подсписка  
 y = len(x)  
#выбираем нужные нам строки и данные CDR  
 for t in range(y):  
 if x[t] == '915642913' and len(x[t - 1]) != 9:  
 msisdn\_origin = x[t]  
 msisdn\_dest = x[t+1]  
 call\_durationISH = x[t+2]  
 if x[t] == '915642913' and len(x[t - 1]) == 9:  
 call\_durationVH = x[t+1]  
 sms\_number = x[t+2]  
else:  
 print('Данный вариант ЛР имеет следующий номер абонента: 915642913')  
 raise SystemExit  
  
#Раздел "Тарификация"  
  
#объявляем необходимые переменные  
call\_duration = 0  
  
#итоговая стоимость всех звонков абонента  
X = 0  
#1 руб/минута исходящие звонки, 1 руб/минута входящие звонки  
k = 1  
#общая длительность звонков  
T = 0  
#тарификация Телефонии  
T = float(call\_durationISH) + float(call\_durationVH)  
X = T \* k  
  
#итоговая стоимость всех CМC абонента  
Y = 0  
#СМС - первые 5шт бесплатно, далее - 1 руб/шт  
kSMS = 1  
freeSMS = 5  
#общее количество СМС  
N = int(sms\_number)  
#тарификация СМС  
Y = (N - freeSMS) \* kSMS  
  
print('Итоговая стоимость звонков абонента 915642913 за текущий период:', X, 'руб.')  
print('Итоговая стоимость СМС абонента 915642913 за текущий период:', Y, 'руб.')

**Скриншоты работы программы:**

Работа программы при неверно введённом номере абонента:

Изображение выглядит как снимок экрана, сидит, монитор, черный

Автоматически созданное описание

Работа программы при верно введённом номере абонента:

**Изображение выглядит как снимок экрана, сидит, монитор, зеленый

Автоматически созданное описание**

**Выводы:** в ходе выполнения лабораторной работы были получены навыки по обработке файла и тарификации CDR абонента по указанным правилам.