## МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет ИТМО»

**ФАКУЛЬТЕТ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**

Управление мобильными устройствами

Лабораторная работа №1 Вариант 3

Работу выполнила: Студентка группы N3347 Волкова Елизавета

Проверено:

Таранов С. В.

Описание: Macintosh HD:Users:3wfrer:Documents:ENDY works:ИТМО:ITMO_black-01.png

Санкт-Петербург

2020

1. **Задание**

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.

В данной работе необходимо 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 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 (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).

В качестве результата работы необходимо представить программный модуль для обработки CDR и тарификации абонента.

Протарифицировать абонента с номером 915783624 с коэффициентом k: 2руб/минута исходящие звонки, но 20 минут бесплатно,

0руб/минута входящие, смс - 2руб/шт

1. **Описание выбранных средств реализации и обоснование выбора**

Для реализации программного модуля я выбрала C# так как владею им наиболее свободно из всех языков, и считаю самым удобным инструментом для реализации большинства задач.

1. **Исходный код**

using System;

using System.Collections.Generic; using System.IO;

using System.Linq;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

var data = ProcessCSV("data.csv"); foreach (var ndata in data)

{

int result = String.Compare(ndata.msisdn\_origin, Cost.num); if (result == 0)

{

string x = ndata.call\_duration; x = x.Replace ('.',',');

float d = Convert.ToSingle(x);

float costt = (d \* Cost.origin)-Cost.free; if (costt > 0)

{

Console.WriteLine("Cost for phone:"+"\n"+ costt + "\n");

}

string b = ndata.sms\_number;

float costs = Convert.ToSingle(b) \* Cost.sms; Console.WriteLine("Cost for sms:" + "\n" + costs);

}

}

Console.ReadLine();

}

private static List<Data> ProcessCSV(string path)

{

return File.ReadAllLines(path)

.Skip(1)

.Where(row => row.Length > 0)

.Select(Data.ParseRow).ToList();

}

}

public class Data

{

public string timestamp { get; set; } public string msisdn\_origin { get; set; } public string msisdn\_dest { get; set; } public string call\_duration { get; set; }

public string sms\_number { get; set; }

internal static Data ParseRow (string row)

{

var columns = row.Split(',');

return new Data()

{

timestamp = columns[0], msisdn\_origin = columns[1], msisdn\_dest = columns[2], call\_duration = columns[3], sms\_number = columns[4]

};

}

}

public class Cost

{

internal const float origin = 2; internal const float sms = 2; internal const float free = 40;

internal const string num = "915783624";

}

}

# Результат работы

