

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего
профессионального образования
«Санкт-Петербургский национальный исследовательский университет информационных
технологий,
механики и оптики»

Факультет безопасности информационных технологий

Управление мобильными устройствами

Отчет
по лабораторной работе №1

Обработка и тарификация CDR (Call Detail Record)
Вариант №6

Студент: Ниткин И.С.
Группа: N3354

Преподаватель: Федоров И.Р.



Санкт-Петербург
2020

Цель работы:

Изучить и программно реализовать простейшее правило тарификации для услуг типа “Телефония” по длительности разговора и “СМС” по общему количеству.

Ход работы:

Работа включает в себя 2 этапа:

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

В качестве средства реализации мною было выбрано написание программного модуля на языке Java. Достоинством языка Java является динамическая компиляция.

Исходный код программы:

```
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.List;

public class Tarificate {

    static String filename = "\\data.csv"; // name of CDR file

    public static void main (String [] args) throws IOException {

        String subNumber;
        if (args.length == 0) subNumber = "";
        else subNumber = args[0];

        String path = new File("").getAbsolutePath() + filename;
        List<cdrRow> cdr = parse (path);

        calcilation(cdr, new Tarif(subNumber));
    }

    private static List<cdrRow> parse (String path) throws IOException {
        List<cdrRow> cdr = new ArrayList<>();
        List<String> fileLines = Files.readAllLines(Paths.get(path));
        fileLines.remove(0);

        for (String fileLine : fileLines) {
            String[] splitedText = fileLine.split(",");

            cdrRow current = new cdrRow (splitedText);
            cdr.add(current);
        }

        return cdr;
    }
}
```

```

        private static void calculation (List<cdrRow> cdr, Tarif tarif) {
            for (cdrRow cdrrow : cdr) {
                if (cdrrow.msisdn_origin.equals(tarif.subNumber)) {
                    tarif.Ti += Math.ceil(cdrrow.call_duration);
                    tarif.Ts += cdrrow.sms_number;
                }
                if (cdrrow.msisdn_dest.equals(tarif.subNumber)) tarif.Tu +=
Math.ceil(cdrrow.call_duration);
            }

            System.out.println(tarif.countNprint());
        }
    }

    class cdrRow {

        String timestamp;
        String msisdn_origin;
        String msisdn_dest;
        double call_duration;
        int sms_number;

        cdrRow (String [] splitedText) {
            this.timestamp = splitedText[0];
            this.msisdn_origin = splitedText[1];
            this.msisdn_dest = splitedText[2];
            this.call_duration = Double.parseDouble(splitedText[3]);
            this.sms_number = Integer.parseInt(splitedText[4]);
        }
    }

    class Tarif {
        String subNumber;

        int ki = 4, ku = 1, ks = 1; // price incoming minutes/upcoming
minutes/sms
        int fi = 0, fu = 5, fs = 5; // free incoming minutes/upcoming minutes/sms

        int Ti, Tu, Ts;
        int It, Ut, St, S;

        Tarif (String subNumber) {
            if (subNumber.length() == 0) this.subNumber = "968247916";
            else this.subNumber = subNumber;
        }

        void count() {
            It = Math.max((Ti-fi), 0)*ki;
            Ut = Math.max((Tu-fu), 0)*ku;
            St = Math.max((Ts-fs), 0)*ks;
            S = It + Ut + St;
        }

        String countNprint() {
            count();
            String printmessage = subNumber + ":\n" +
                "incoming calls:\t" + Ti + " min.\t = " + It + " rub.\n" +
                "upcoming calls:\t" + Tu + " min.\t = " + Ut + " rub.\n" +
                "sms:\t" + Ts + "\t\t = " + St + " rub.\n" +
                "TOTAL\t = " + S + " rub.";

            return printmessage;
        }
    }
}

```

Для запуска:

```
> java -jar L1.jar [phone number]
```

По умолчанию - запуск тарификации для номера 968247916, соответствующего варианту №6

Файл с данными CDR:

data.csv

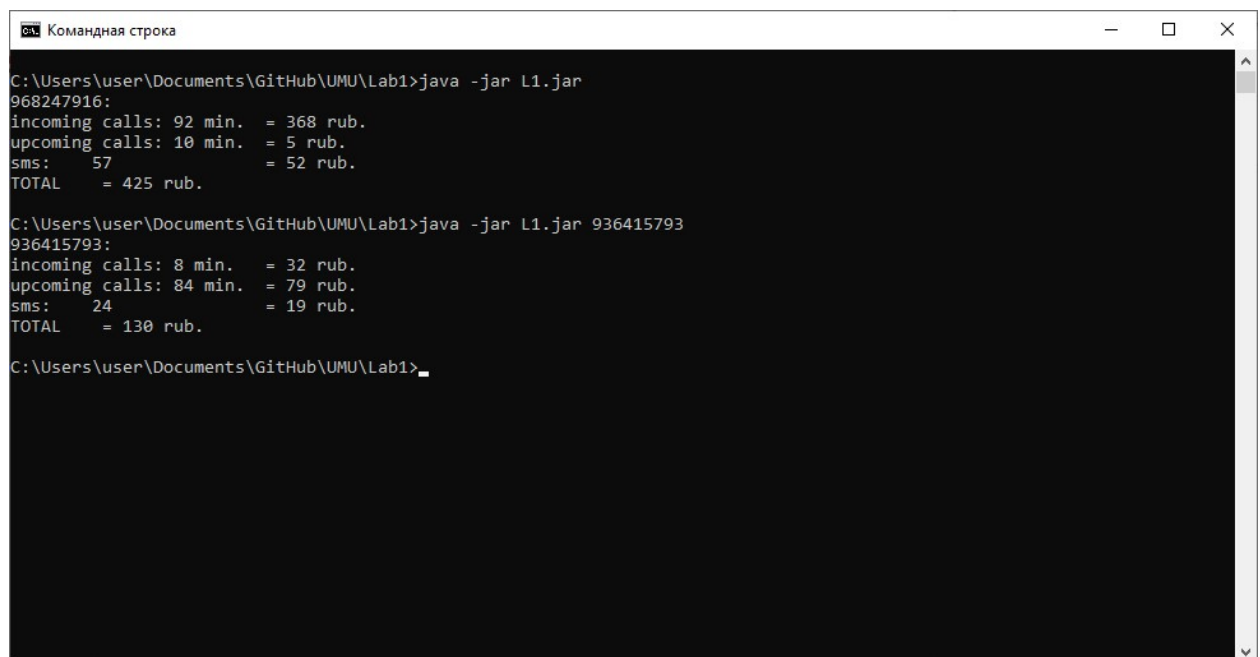
Тарифные опции (соответствуют варианту №6, записаны в код программы):

4руб/минута исходящие звонки,

0руб/минута входящие первые 5 минут, далее 1руб/минута,

смс - первые 5шт бесплатно, далее 1руб/шт.

В результате выполнения программы в консоль выводится сообщение с информацией о тарификации.



```
Командная строка

C:\Users\user\Documents\GitHub\UMU\Lab1>java -jar L1.jar
968247916:
incoming calls: 92 min.   = 368 rub.
upcoming calls: 10 min.  = 5 rub.
sms:      57              = 52 rub.
TOTAL     = 425 rub.

C:\Users\user\Documents\GitHub\UMU\Lab1>java -jar L1.jar 936415793
936415793:
incoming calls: 8 min.    = 32 rub.
upcoming calls: 84 min.   = 79 rub.
sms:      24              = 19 rub.
TOTAL     = 130 rub.

C:\Users\user\Documents\GitHub\UMU\Lab1>
```

Вывод:

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