

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО



Факультет Безопасных Информационных Технологий

УПРАВЛЕНИЕ МОБИЛЬНЫМИ УСТРОЙСТВАМИ

Лабораторная работа №2

Обработка и тарификация трафика NetFlow

Вариант 9

Выполнила:

студентка группы N3352

Копосова Ю.Н.



Проверил:

Федоров И.Р.

Санкт-Петербург

2020

Задача работы: разработать программный модуль для обработки трафика NetFlow и тарификации абонента.

В качестве средства реализации выбрала C#, поскольку имеется небольшой опыт работы с ним.

Код:

```
using
System;

using System.Windows.Forms;
using System.IO;
using System.Globalization;

namespace Lab_2 {
    public partial class Form1 : Form {
        public Form1() {
            InitializeComponent();

            private void Form1_Load(object sender, EventArgs e) {
                string[] uniqueTime = new string[100];
                double[] traffics = new double[100];
                string sumTrafficString = "";
                string MString = "";
                _Main.mainFunction(ref uniqueTime, ref traffics, ref
sumTrafficString, ref MString);
                label1.Text = $"Bcero {sumTrafficString} K6";
                label2.Text = $"{MString} рублей";
                for (int i = 0; i < 80; i++) {
                    chart1.Series[0].Points.AddXY(uniqueTime[i], traffics[i]);
                }
            }
        }

        class _Main {
            public static void mainFunction(ref string[] uniqueTime, ref double[]
traffics, ref string sumTrafficString, ref string MString) {
                double sumTraffic = 0;
                Parsing Parser = new Parsing();
                Tariffication Tarificator = new Tariffication();
                Parser.CreateArray();
                Parser.GetValues(ref sumTraffic, ref uniqueTime, ref traffics);
                Tarificator.CalculationService(ref sumTraffic, ref sumTrafficString,
ref MString);
            }
        }

        class Parsing {
```

```

string[] NFR = File.ReadAllLines("NetFlow.csv");
string abonent = "192.168.250.3";
string[] row;
string[,] abonentRecords = new string[4774, 50];
int recordsCounter = 0;

public void CreateArray() {
    for (int i = 0; i < NFR.Length; i++) {
        if (NFR[i].Contains(abonent)) {
            row = NFR[i].Split(',');
            for (int j = 0; j < row.Length; j++) {
                abonentRecords[recordsCounter, j] = row[j];
            }
            recordsCounter++;
        }
    }
}

public void GetValues(ref double sumTraffic, ref string[] uniqueTime, ref
double[] traffics) {
    CultureInfo _invert = CultureInfo.InvariantCulture;
    sumTraffic = 0;
    int count = 0;
    int uniqueTimeCount = 0;
    bool check = false;
    DateTime[] fullTime = new DateTime[4774];
    string[] shortTime = new string[4774];
    for (int i = 0; i < abonentRecords.GetLength(0); i++) {
        if (abonentRecords[i, 3] == abonent || abonentRecords[i, 4] ==
abonent) {
            sumTraffic += Convert.ToDouble(abonentRecords[i, 12],
_invert);

            fullTime[count] = Convert.ToDateTime(abonentRecords[i, 0]);
            shortTime[count] = fullTime[count].ToShortTimeString();
            count++;
        }
    }

    for (int i = 0; i < shortTime.Length; i++) {
        for (int j = 0; j < uniqueTimeCount + 1; j++) {
            if (shortTime[i] == uniqueTime[j]) {
                traffics[j] += Convert.ToDouble(abonentRecords[i, 12],
_invert);

                check = true;
                break;
            }
        }
        if (check == false) {

```

```

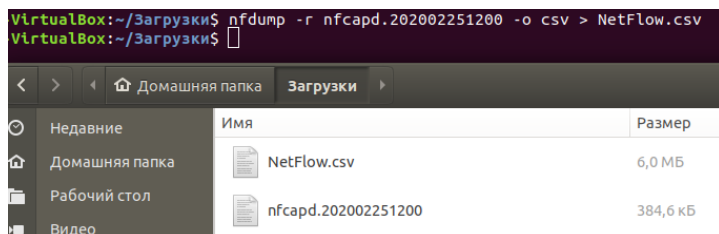
        uniqueTime[uniqueTimeCount] = shortTime[i];
        traffics[uniqueTimeCount] +=
Convert.ToDouble(abonentRecords[i, 12], _invert);
        uniqueTimeCount++;
    }
    check = false;
}
}
}

class Tariffication {
    double kTraffic = 0.5;
    double M;

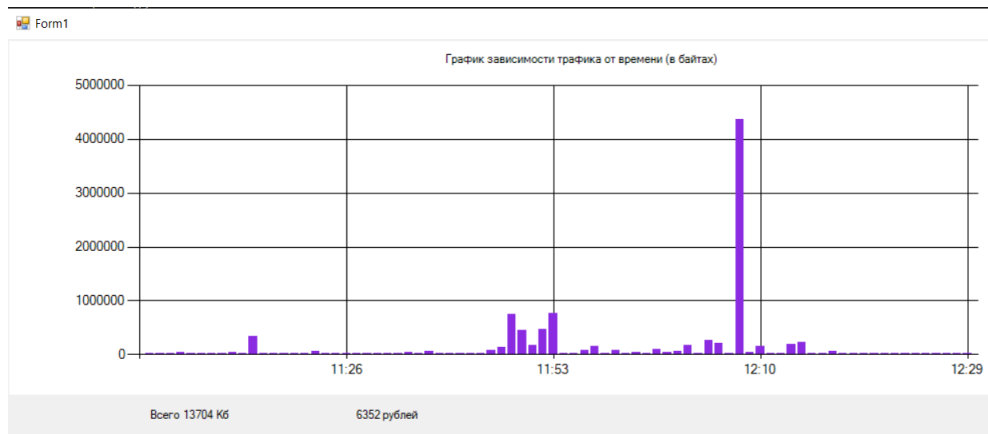
    public void CalculationService(ref double sumTraffic, ref string
sumTrafficString, ref string MString) {
        sumTraffic = sumTraffic / 1024; //в килобайтах
        M = (sumTraffic - 1000) * kTraffic;
        sumTraffic = Math.Round(sumTraffic);
        M = Math.Round(M);
        sumTrafficString = Convert.ToString(sumTraffic);
        MString = Convert.ToString(M);
    }
}
}
}

```

Файл csv сформировался при помощи следующей команды:



Результат:



Выводы: научилась работать с Windows Forms; закрепила навыки работы с C#.