

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»

**ФАКУЛЬТЕТ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Лабораторная работа №3  
по теме «Формирование счета на оплату услуг»  
по предмету «Управление мобильными устройствами»

Работу выполнил  
Студент группы N3347  
очного отделения  
Милосердов Дмитрий



Проверил:  
Федоров И.Р.  
Дата защиты: 17.04.2020



Санкт-Петербург 2020

**Цель работы:** создать программный модуль по формированию отчета оплаты

**Задачи:**

1. Подсчет стоимости всех услуг клиента
2. Формирование отчета

**Средства реализации:** для реализации программного модуля был выбран язык программирования Python. Выбран он был, потому что написание кода на данном языке программирования занимает меньше времени, а также обладает удобными готовыми средствами, как docxtpl, docx2pdf, num2t4ru. Также за основу были взяты коды из предыдущих лабораторных работ, что значительно упростило написание кода.

Исходный код (также можно найти его на

[https://github.com/NevSk1y/mobile/blob/master/laba3/Miloserdov\\_laba3.py](https://github.com/NevSk1y/mobile/blob/master/laba3/Miloserdov_laba3.py)):

```
#num = '915783624'
```

```
#ipaddr = "217.15.20.194"
```

```
import csv
```

```
import time
```

```
from docxtpl import DocxTemplate
```

```
from docx2pdf import convert
```

```
from num2t4ru import num2text, decimal2text
```

```
import decimal
```

```
filename = input('Введите путь до файла с интернет трафиком: ')
```

```
filename1 = input('Введите путь до файла с мобильным трафиком: ')
```

```
num = input('Введите номер телефона: ')
```

```
ipaddr = input('Введите IP-адрес: ')
```

```
rows = []
```

```
fields = []
```

```
T_out = 0
```

```
k_out = 2
```

```
T_inc = 0
```

```
k_inc = 0
```

```
N = 0
```

```
k_sms = 1
```

```
with open(filename1, 'r') as file:
```

```
    reader = csv.reader(file)
```

```
    fields = next(reader)
```

```
    for row in reader:
```

```
        rows.append(row)
```

```
def out_calls():
```

```
    global T_out
```

```
    global k_out
```

```
    for row in rows[:10]:
```

```
        if num in row[1]:
```

```
        T_out += float(row[3])
    X_out = T_out*k_out
    return X_out
```

```
def inc_calls():
    global T_inc
    global k_inc
    for row in rows[:10]:
        if num in row[2]:
            T_inc += float(row[3])
    X_inc = T_inc*k_inc
    return X_inc
```

```
def sms():
    global N
    global k_sms
    for row in rows[:10]:
        if num in row[1]:
            N += int(row[4])
    Y = (N - 10)*k_sms
    if Y < 0:
        Y = 0
    return Y
```

```
Total = sms()+out_calls()+inc_calls()
```

```
rows = []
fields = []
inc_time = []
inc_traf = []
out_traf = []
out_time = []
k = 0.5
Q_inc = 0
Q_out = 0
```

```
with open(filename, 'r') as file:
```

```

        reader = csv.reader(file)
        fields = next(reader)
        for row in reader:
            rows.append(row)
def out_traffic():
    global Q_out
    for row in rows[:reader.line_num]:
        if ipaddr in row[3]:
            Q_out += int(row[12])
            out_time.append(row[0])
            out_traf.append(Q_out)

    return Q_out
def inc_traffic():
    global Q_inc
    for row in rows[:reader.line_num]:
        if ipaddr in row[4]:
            Q_inc += int(row[12])
            inc_time.append(row[0])
            inc_traf.append(Q_inc)

    return Q_inc
summ_traffic_Mb = (inc_traffic() + out_traffic()) / 1048576
Total1 = summ_traffic_Mb * k

Total1 = float('{:.2f}'.format(Total1))
nds = float('{:.2f}'.format((Total + Total1) * 0.2))

int_units = ((u'рубль', u'рубля', u'рублей'), 'm')
exp_units = ((u'копейка', u'копейки', u'копеек'), 'f')
translite = str(decimal2text(
    decimal.Decimal(str(Total + Total1)),
    int_units=int_units,
    exp_units=exp_units))

doc = DocxTemplate('schetinput.docx')
context = {
    'sum_tele' : out_calls()+inc_calls(),
    'sum_sms' : sms(),

```

```
        'sum_inet' : Total1,  
        'propis' : translite,  
        'sum_total' : Total + Total1,  
        'nds' : nds  
    }  
doc.render(context)  
doc.save('oplata.docx')  
convert('oplata.docx')
```

**Выводы:** Я научился автоматизировать процесс создания документов и конвертировать в разные форматы.