**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

**ФАКУЛЬТЕТ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Управление мобильными устройствами

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

«Формирование счета на оплату услуг»

Вариант 10

Работу выполнил:

студент 3 курса

группы N3348

Мосолов А. Н.

Проверил:

Фёдоров Иван Романович

Санкт-Петербург, 2020

**Цель работы:** представить программный модуль для генерации печатной формы счета на оплату и полученный файл .pdf.

**Средства реализации:** в качестве языка программирования был выбран Python в силу его удобства и простоты использования.

**Исходный код:**

import csv  
from num2t4ru import num2text  
from docxtpl import DocxTemplate  
from docx2pdf import convert  
  
def telephonia():  
 buffer = []  
  
 with open('data.csv') as file:  
 reader = csv.reader(file) # csv-файл передаётся функции csv.reader, которая возвращает объект-считыватель,  
 # позволяющий выполнять итерацию над каждым рядом в объекте-считывателе и отобразить строку данных без запятых  
 for row in reader:  
 buffer.append(row) # строка переносится в конец  
  
 call\_duration = 0 # буфер, где хранятся суммы минут звонков  
 sms\_number = -10 # буфер, где хранится количество смс. "-10", так как первые 10 шт. - бесплатно  
  
 for i in range(1, 10): # рассматриваем данные звонков и смс  
 if '933156729' in buffer[i][1]: # проверяется наличие номера  
 call\_duration += float(buffer[i][3]) # складываем минуты  
 sms\_number += float(buffer[i][4]) # складываем количество смс  
 return("%.0f" % (call\_duration\*2+sms\_number\*1))  
  
  
def internet():  
 list\_check = []  
  
 with open('nfcap.csv') as data\_doc:  
 reader = csv.reader(data\_doc)  
 for i in reader:  
 list\_check.append(i)  
  
 ibyt = 0  
 price = 0  
  
 for i in range(len(list\_check)):  
 if '192.0.73.2' in list\_check[i]:  
 ibyt += float(list\_check[i][12])  
  
  
 if ibyt > 200: # Т. к. общий объём трафика по абоненту меньше заявленного во варианте, было решено рассматривать байты.  
 ibyt = ibyt - 200  
 price += 0.5 \* 200  
  
 price += ibyt \* 1  
 return ("%.0f" % price)  
  
cases = ((u'рубль', u'рубля', u'рублей'), 'm')  
telephonia\_price = int(telephonia())  
internet\_price = int(internet())  
total\_price = telephonia\_price+internet\_price  
nds = "%.0f" % (total\_price\*0.2)  
doc = DocxTemplate("blank.docx")  
context ={  
 'telephonia': telephonia\_price,  
 'internet': internet\_price,  
 'total': total\_price,  
 'nds': nds,  
 'word\_price': (num2text(total\_price, cases))  
}  
doc.render(context)  
doc.save("total.docx")  
convert("total.docx")

**Результаты работы программы** отображаются в файлах «total.docx» и «total.pdf».

**Вывод:** таким образом, в результате работы была написана программа, позволяющая генерировать печатную форму счета на оплату и получаемый файл .pdf.