

**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

**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'), 'м')
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.