

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

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

**ФАКУЛЬТЕТ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ**

Управление Мобильными Устройствами

**Лабораторная работа №3  
«Формирование счета на оплату услуг»  
Вариант 11**

Работу выполнила:  
Студентка группы N3347  
Конева Ксения  
19.04.2020.



Проверено:  
Федоров И.Р.

Санкт-Петербург

2020



**УНИВЕРСИТЕТ ИТМО**

## Цель работы

В рамках работы требуется:

По полученным результатам тарификации услуг «Телефония» и «Интернет» в лабораторных работах 1, 2 сформировать счет на оплату в формате .pdf.  
Все поля печатной формы должны заполняться разработанным программным модулем.

Название банка, имена покупателей и прочие формальные поля можно заполнить какими угодно значениями, зависит от фантазии. Стоимость услуг берется из предыдущих двух работ.

В качестве результата работы необходимо представить программный модуль для генерации печатной формы счета на оплату и полученный файл .pdf.  
Средства реализации выбираются студентом самостоятельно.

## Описание выбранных средств реализации и обоснования выбора.

Для реализации программного модуля я выбрала Python 3.8 (включая библиотеки csv, docx2pdf, docxtpl и num2words), так как владею им наиболее свободно из всех языков, и считаю наиболее быстрым и удобным инструментом для реализации большинства задач.

## Исходный код

```
import csv
from docx2pdf import convert
from docxtpl import DocxTemplate
from num2words import num2words

# Функция для превращения чисел в буквы
def cursive(s):
    end_i = ['2', '3', '4']
    s = s.split(',')
    if len(s) == 2:
        if len(s[1]) == 1:
            s[1] = s[1] + '0'
        if s[0] == '0':
            k = list(s[1])
            if (k[-2] != '1') and (k[-1] == '1'):
                s.append('копейка')
            elif (k[-2] != '1') and (k[-1] in end_i):
                s.append('копейки')
            else:
                s.append('копеек')
            s[0] = 'ноль рублей'
        else:
            r = list(s[0])
            k = list(s[1])
            s[0] = num2words(s[0], lang='ru') # рубли
            if (len(r) == 1) or (r[-2] != '1') and (r[-1] == '1'):
                s.insert(1, 'рубль')
            elif (len(r) == 1) or (r[-2] != '1') and (r[-1] in end_i):
                s.insert(1, 'рубля')
            else:
                s.insert(1, 'рублей')
            if (k[-2] != '1') and (k[-1] == '1'):
                s.append('копейка')
            elif (k[-2] != '1') and (k[-1] in end_i):
                s.append('копейки')
            else:
                s.append('копеек')
    elif len(s) == 1:
        r = list(s[0])
        if s[0] != '0':
            s[0] = num2words(s[0], lang='ru') # рубли
            if (len(r) == 1):
                if (r[0] == '1'):
                    s.insert(1, 'рубль')
                elif (r[0] in end_i):
                    s.insert(1, 'рубля')
```

```

        else:
            s.insert(1, 'рублей')
    else:
        if (r[-2] != '1') and (r[-1] == '1'):
            s.insert(1, 'рубль')
        elif (r[-2] != '1') and (r[-1] in end_i):
            s.insert(1, 'рубля')
        else:
            s.insert(1, 'рублей')
    s.append('00 копеек')
else:
    s[0] = 'ноль рублей 00 копеек'
out = ''.join(s)
return out.capitalize()

```

## # МОБИЛЬНАЯ СВЯЗЬ И СМС

```

def mobil_tarif(lst):
    callcost = 0
    smscost = 0
    cost = 0
    for row in lst:
        callcost += float(row[calldur]) * k
    for row in lst:
        if float(row[smsnum]) <= freesms:
            smscost == 0
        else:
            smscost += (float(row[smsnum]) - 5) * k
    callcost = round(callcost, 2)
    smscost = round(smscost, 2)
    cost = callcost + smscost
    return str(cost)

```

num = '911926375'

k = 1

freesms = 5

origin = 1

dest = 2

calldur = 3

smsnum = 4

mobil = "

mobilf = 0

# Создаем список из нужных для обработки строк (Парсинг CSV)

lst = []

with open('data.csv') as file:

data = csv.reader(file)

for row in data:

for item in row:

```

        if item == num:
            lst.append(row)

# Вывод:
mobil = mobil_tarif(lst)
mobilf = float(mobil)

# ИНТЕРНЕТ
ipadr = '17.248.150.51'
k = 0.5
net = ""
netf = 0
# Функция тарификации интернет-трафика
def net_tarif(lst):
    ibyt = 12
    obyt = 14
    traffic = 0
    cost = 0
    for row in lst:
        traffic += (float(row[ibyt]) + float(row[obyt]))
    traffic /= (1024 ** 2)
    cost = round((traffic * k), 2)
    return str(cost)

lst = []
with open('nfcapd_20200.csv') as file:
    data = csv.reader(file)
    for row in data:
        for item in row:
            if item == ipadr:
                lst.append(row)
net = net_tarif(lst)
netf = float(net)

total = mobilf + netf
tax = total * 0.2
total_cursive = cursive(str(total))
# СЧЕТ

doc = DocxTemplate('template.docx')
context = {
    'provider_bank' : 'ПАО \"Сбербанк\"',
    'provider_bik' : '044525225',
    'provider_bank_acc' : '30101810400000000225',
    'provider_inn' : '6705083723',
    'provider_kpp' : '654602004',
    'provider_acc' : '20441830400020000107',
    'provider' : 'ООО \"MobilNetService\"',
}

```

```
'bill_num' : '116',
'bill_date' : '20 мая 2020',
'provider_address' : 'г. Санкт-Петербург, ул. Кронверкская, д. 116',
'client': 'Конева К.П.',
'client_inn' : '1105083489',
'client_kpp' : '217602189',
'client_address': 'г. Санкт-Петербург, ул. Дунайская, д. 89',
'bill_basis' : '№300678 от 19.05.2019',
'mobile_service': 'Мобильная связь и СМС',
'mob.price' : mobil,
'net_service': 'Интернет',
'net.price': net,
'bill_total' : total,
'bill_tax': tax,
'total_cursive' : total_cursive,
'provider_director' : 'В.А. Александрова',
'provider_accountant' : 'И.Н. Серов'
}
doc.render(context)
doc.save('final.docx')
convert('final.docx', 'final.pdf')
```

## Выходы

Успешно выполнена реализация программного модуля для автоматизированного создания счетов. Вызов программы осуществляется в командной строке. Результатом работы являются файлы final.docx и final.pdf.